THÈSE

PRÉSENTÉE À

L'UNIVERSITÉ DE BORDEAUX

ÉCOLE DOCTORALE DE MATHÉMATIQUES ET D'INFORMATIQUE

par **Théo Pierron**

POUR OBTENIR LE GRADE DE

DOCTEUR

SPÉCIALITÉ : INFORMATIQUE

Induction Schemes: From Language Separation to Graph Colorings

---

Date de soutenance : 08 juillet 2019

Devant la commission d'examen composée de :

Marthe Bonamy	Chargée de recherche, CNRS, LaBRI	Encadrante
Mireille Bousquet-Mélou	Directrice de recherche, CNRS, LaBRI	Examinatrice
Thomas Colcombet	Directeur de recherche, CNRS, IRIF	Examinateur
Zdeněk Dvořák	Associate Professor, Charles University	Rapporteur
Michał Pilipczuk	Assistant Professor, University of Warsaw	Rapporteur
Jean-Sébastien Sereni	Directeur de recherche, CNRS, ICube	Rapporteur
Éric Sopena	Professeur, Université de Bordeaux	Directeur
Marc Zeitoun	Professeur, Université de Bordeaux	Directeur

Présidée par : Mireille Bousquet-Mélou

– 2019 –

Résumé Cette thèse présente des résultats obtenus dans deux domaines : la théorie des langages, et la théorie des graphes. En théorie des langages, on s'intéresse à des problèmes de caractérisation de classes de langages réguliers. Le problème générique consiste à déterminer si un langage régulier donné peut être défini dans un certain formalisme. Les méthodes actuelles font intervenir un problème plus général appelé séparation. On présente ici deux types de contributions : une généralisation d'un résultat de décidabilité au cadre des langages de mots infinis, ainsi que des bornes inférieures pour la complexité du problème de séparation.

En théorie des graphes, on considère le problème classique de coloration de graphes, où on cherche à attribuer des couleurs aux sommets d'un graphe de sorte que les sommets adjacents reçoivent des couleurs différentes, le but étant d'utiliser le moins de couleurs possible. Dans le cas des graphes peu denses, la méthode de déchargement est un atout majeur. Elle a notamment joué un rôle décisif dans la preuve du théorème des quatre couleurs. Cette méthode peut être vue comme une construction non conventionnelle d'un schéma de preuve par induction, spécifique à la classe de graphes et à la propriété considérées, et où la validité du schéma est rarement immédiate. On utilise des variantes de la méthode de déchargement pour étudier deux types de problèmes de coloration.

Title Induction Schemes: From Language Separation to Graph Colorings

Abstract In this thesis, we present results obtained in two fields: formal language theory and graph theory. In formal language theory, we consider some problems of characterization of classes of regular languages. The generic problem consists in determining whether a given regular language can be defined in a fixed formalism. The current approaches use a more general problem called separation. We present here two types of contributions: a generalization of a decidability result to the setting of infinite words, together with lower bounds for the complexity of the separation problem.

In graph theory, we consider the classical problem of graph coloring, where we assign colors to vertices of a graph in such a way that two adjacent vertices receive different colors. The goal is to use the fewest colors. When the graphs are sparse, a crucial tool for this is the discharging method. It is most notably decisive in the proof of the Four-Color Theorem. This method can be seen as an unconventional construction of an inductive proof scheme, specific to the considered problem and graph class, where arguing the validity of the scheme is rarely immediate. We use variants of the discharging method to study two types of coloring problems.

Keywords Coloration de graphe, Graphe planaire, Méthode de déchargement, Langage régulier, Séparation de langages

Mots-clés Graph coloring, Planar graph, Discharging method, Regular language, Languages separation

Laboratoire d'accueil LaBRI, 351, cours de la Libération F-33405 Talence cedex

Remerciements

Bien qu'individuel, une thèse est loin d'être un travail solitaire. Ainsi, de nombreuses personnes ont contribué, de près ou de loin, et parfois sans s'en rendre compte, à ce manuscrit. La liste est d'autant plus longue que le chemin fut sinueux, il y a de fortes chances qu'elle ne soit pas exhaustive.

Les premières lignes sont naturellement adressées à ma famille, ceux qu'on ne choisit pas mais dont on ne voudrait quand même pas changer. Merci pour votre présence, et votre soutien, surtout dans les moments difficiles. Dédicace spéciale à ma maman qui, même si tout ne s'est pas toujours passé comme elle l'aurait voulu, a été d'un grand secours tout au long de ces quatre ans.

La famille académique n'est pas en reste : merci Marc, pour la liberté que tu m'as laissée ainsi que ta disponibilité malgré ton emploi du temps surchargé. Merci Marthe, pour tes multiples casquettes de dealeuse de gâteaux et thé, amie et encadrante. Accepter le dernier rôle n'a pas dû être une décision facile à prendre, et je ne te remercierai certainement jamais assez de l'avoir prise. C'est aussi valide pour Éric : merci de m'avoir pris sous ton aile, ainsi que pour ton écoute et ta générosité. J'ai énormément apprécié travailler avec vous, et j'espère continuer à l'avenir. Je suis aussi très fier de vous avoir eu comme directeurs, j'ai énormément appris à vos côtés, et je ferai tout mon possible pour ne pas vous faire regretter de m'avoir encadré.

Another trio deserves acknowledgment: Michał, Zdeněk and Jean-Sébastien. Despite all my efforts, reviewing this manuscript is not an easy task, and I am very grateful that you still accepted it. Thank you for all your remarks and comments, even if some of them gave me nightmares. Thanks also for coming to Bordeaux for the defense, I was very pleased to meet you at last. I could not dream of any better referees, and I look forward to meet you again in conferences or even research circumstances.

Mireille, Thomas, merci d'avoir accepté le rôle d'examinateur (et la présidence du jury) pour la soutenance. À ce titre, merci aussi pour votre relecture attentive du manuscrit.

Merci à l'équipe Méthodes Formelles, où je suis arrivé en stage puis en thèse, notamment à Anca, qui m'a accompagné dès le master. Merci aussi à CombAlgo, qui m'a accueilli à bras ouverts. C'est une équipe formidable pour effectuer un doctorat, et j'espère y revenir très vite, pour discuter de recherche ou d'autres choses. Plus généralement, merci aussi à la communauté des chercheurs en théorie des graphes pour m'avoir fait me sentir comme l'un des leurs, et pour faire ressembler chaque conférence à une grande réunion de famille. En particulier, merci à ceux avec qui j'ai pu travailler : Florent (Foucaud), Julien (Bensmail), Pascal (Ochem), Alexandre (Pinlou), Mickaël (Montassier), Borut (Lužar), Édouard (Bonnet), Jean-Florent (Raymond), Riste (Škrekovski), Mirko (Petruševski), Daniel (Paulusma), Matthew (Johnson), Nicolas (Bousquet), Dan (Cranston), Ilkyoo (Choi), François (Dross), Sagnik (Sen), Sandip (Das),... Finally, thanks Dan (Král') for letting me continue in research with a postdoc.

Pendant quatre ans, j'ai pu obtenir un service d'enseignement à l'université (merci Alain). J'ai donc été amené à m'intégrer dans plusieurs équipes pédagogiques. Merci donc à tous ceux avec qui j'ai pris beaucoup de plaisir à préparer et organiser les cours : Olivier, Feri, Giuliana, Carole, Marc, Frédérique, Stefka, Irène, Simon, Philippe (peut-être pourrais-je un jour rentrer dans ton bureau sans devoir régler un problème d'emploi du temps), Luis (ravi de faire affaire avec toi), Pierre, Louis-Marie, Henri, François. Toujours au LaBRI, merci aussi aux gestionnaires, pour votre efficacité et bonne humeur. Je ne doute pas que mes recettes de gâteaux feront encore des heureux après mon départ.

Parmi les membres du LaBRI que je n'ai pas encore cité, merci aussi à tous ceux qui sont devenus plus que des collègues : Niko (merci encore pour le chocolat !), Hervé (pour ta simplicité et ta bonne humeur), Filip (I hope you will come to Brno for tea or beer), Paul Jr (on ne saura jamais qui est le grand frère, mais je suis fier que tu fasses partie de la famille), Feri, Laurent, Claire, Rohan, Jason, Simon, Mathias, Romain, Théo, Henri, Louis-Marie, Mohammed, Nathan, Vincent (bon courage pour les prochaines soirées jeux), Tristan, Matthieu, Jonathan (à ton tour de prendre la relève !) et évidemment Thomas, Dimitri, Alexandre (bienvenue dans la fratrie aussi !), Tobias et Antonin, pour l'ambiance qui règne dans le bureau (j'espère que ça continuera encore longtemps).

Merci enfin aux amis de plus ou moins longue date : d'abord, Thomas et Antonin, pour les nombreuses soirées passées ensemble, puis la troupe de mamies (qui n'ont de mamies que le nom) et le club tarot, que j'espère revoir très vite, et enfin la troupe du lycée, pour me supporter depuis la nuit des temps.

Contents

Introduction (French version)

Cette thèse est divisée en deux parties, chacune d'entre elles considérant des objets et des problèmes différents. Chaque problème fera l'objet d'une introduction détaillée dans le chapitre idoine. Ici, on propose tout d'abord une présentation historique des méthodes inductives. Bien que ces méthodes soient monnaie courante en informatique théorique, cette introduction s'oriente spécifiquement vers une présentation des problèmes et outils considérés dans la première partie. Le domaine de la deuxième partie nécessite plus de pré-requis et sera donc présenté dans le chapitre dédié.

Histoire des méthodes inductives

Selon [Cajori, 1918], la méthode *chakravala*, introduite par Bhāskara II au XII^e^ siècle est un des premiers exemples de ce qu'on pourrait considérer comme une preuve par récurrence. Cette méthode est en réalité un algorithme fournissant une solution entière (x, y) à l'équation diophantienne $x^2 - ny^2 = 1$, qui porte aujourd'hui le nom d'équation de Pell-Fermat. Bien que l'étude de ces équations puisse être retracée jusqu'au VI^e^ siècle, la première solution complète (donnée par la méthode chakravala) fut découverte seulement au XII^e^ siècle. Elle fut ensuite redécouverte bien plus tard (en 1930) en Europe. Entre-temps, le problème fut aussi résolu indépendamment par Lagrange en 1767, au moyen d'un autre algorithme.

Bien qu'ils ne soient pas écrits avec un formalisme actuel, on peut trouver de nombreux exemples de preuves par récurrence au fil du temps et des civilisations. Par exemple, on peut citer les travaux d'Euclide sur les nombres premiers, ou de Al-Karaji, al-Samaw'al et Ibn al-Haytham en combinatoire (sommes d'entiers ou de carrés, formule du binôme...), ou bien encore de Gersonide au Moyen Âge. Au XVII^e^ siècle, on peut trouver de nombreux exemples dans les traités de Pascal, Fermat et Bernoulli. Enfin, le formalisme de ce schéma de preuve devint plus précis au XIX^e^ siècle, grâce aux travaux de Grassmann, puis de Dedekind et Peano.

Supposons qu'on souhaite prouver que tout entier naturel n satisfait une propriété P. Pour parvenir à nos fins, il suffit de montrer dans un premier temps que P est vérifiée au premier niveau, c'est-à-dire pour $n = 0$. Dans un second temps, on montre que si P est satisfaite par un entier n, alors elle est aussi satisfaite par son successeur $n+1$. Par exemple, considérons une échelle infinie que l'on souhaiterait gravir. Si on sait grimper sur le premier barreau, et on sait monter sur le barreau suivant quand on est sur un barreau donné, alors on peut atteindre n'importe quel barreau de l'échelle. Ce schéma de preuve peut se résumer par l'axiome suivant de l'arithmétique de Peano.

$$\left.\begin{array}{l} P(0) \\ \forall n, P(n) \Rightarrow P(n+1) \end{array}\right\} \Rightarrow \forall n, P(n)$$

Les preuves par récurrence ne sont pas réservées aux entiers : elles peuvent s'étendre à des structures plus complexes, généralement récursives, comme les arbres (enracinés). Un arbre est constitué soit d'une feuille, soit d'une racine adjacente à un nombre fini de sous-arbres. On parle alors de preuve par induction. Dans ce cadre, le but est de montrer une propriété sur les feuilles, puis de l'étendre des sous-arbres d'un nœud au nœud lui-même. En fait, on peut se ramener à une preuve par récurrence sur le bon paramètre de l'arbre comme sa hauteur ou son nombre de nœuds. Par extension, on parle aussi de schéma d'induction pour définir des objets récursivement. Les preuves par induction constituent donc des outils adaptés à l'étude d'objets définis par induction.

Les preuves par induction peuvent se généraliser (d'une manière similaire) à des ensembles munis d'un *ordre bien fondé*, c'est-à-dire tel qu'il n'y ait pas de suite infinie strictement décroissante. En effet, on peut souvent traduire une preuve par induction via un schéma de preuve alternatif utilisant des ordres bien fondés. Bien que ce soit seulement une reformulation, cet effet cosmétique permet d'obtenir des preuves généralement plus lisibles. On peut citer deux exemples de ce phénomène. Le premier est la méthode de *descente infinie*. Sa première apparition est parfois attribuée à Euclide pour sa preuve que tout nombre composé est divisible par un nombre premier. Elle fut ensuite popularisée par Fermat pour étudier les équations diophantiennes, dans le but de montrer qu'il n'y a pas de solution. L'argument principal est le suivant : en supposant que l'équation admet une solution s, on peut construire une solution s' telle que $s' < s$ pour un ordre bien fondé $<$. Ainsi, l'existence d'une solution est incompatible avec la propriété que $<$ est bien fondé, ce qui assure que l'équation considérée n'a pas de solution. Dans le cas du résultat d'Euclide, on utilise que l'ordre habituel sur les entiers naturels est bien fondé : à partir d'un entier composé n sans diviseur premier, on construit un entier $n' < n$, lui aussi composé et sans diviseur premier. On obtient alors une suite infinie strictement décroissante d'entiers naturels, une contradiction.

L'autre exemple est donné par la méthode dite *de déchargement*, adaptée à l'étude des graphes, et centrale dans les chapitres 1 et 2. Un *graphe* est un objet déterminé par un ensemble fini de sommets $V(G)$ et par un ensemble $E(G)$ d'arêtes, *i.e.* de paires de sommets de la forme uv pour $u, v \in V(G)$

tels que $u \neq v$. Un des problèmes les plus connus en théorie des graphes est la conjecture des quatre couleurs, énoncée en 1852 et qui est dorénavant un théorème. Un *graphe planaire* est un graphe qu'on peut dessiner dans le plan sans que deux arêtes se croisent. Le théorème des quatre couleurs assure qu'on peut attribuer un entier 1, 2, 3 ou 4 à chaque sommet de tout graphe planaire (cet entier est la *couleur* du sommet) de telle sorte que les sommets adjacents reçoivent des couleurs différentes. La méthode de déchargement fut introduite il y a plus d'un siècle dans [Wernicke, 1904], pour étudier ce problème. Dans ce cadre, l'idée générale est de considérer un contre-exemple de taille minimale à la 4-colorabilité, et d'utiliser cette minimalité pour obtenir des propriétés structurelles du contre-exemple, d'une manière comparable à la méthode de descente infinie. On obtient ensuite une contradiction en montrant que ces propriétés ne sont jamais simultanément satisfaites. Cette méthode est surtout pratique pour étudier des classes de graphes peu denses (comme les graphes planaires), et a permis d'obtenir de nombreux résultats, comme l'attestent [Borodin, 2013; Cranston and West, 2017].

Le cadre général de la méthode de déchargement peut s'exprimer ainsi. Supposons qu'on veuille montrer une propriété P sur une classe de graphes $\mathcal{C}$. Pour simplifier, on considère que le graphe vide appartient à $\mathcal{C}$ et satisfait P. Le but est de trouver des informations structurelles sur les graphes de $\mathcal{C}$. Pour ce faire, on définit certaines structures appelées « configurations ». De nombreux exemples de telles configurations seront présentées dans le chapitre 1. On recherche donc un ensemble $\mathcal{S}$ de configurations vérifiant deux types de propriétés :

- $\mathcal{S}$ est $\mathcal{C}$-inévitable, c'est-à-dire que tout graphe non vide de $\mathcal{C}$ doit contenir au moins une configuration de $\mathcal{S}$.
- Chaque configuration S dans $\mathcal{S}$ est $(P, \mathcal{C})$-réductible, c'est-à-dire que pour tout graphe $G \in \mathcal{C}$ ne satisfaisant pas P et contenant S, on peut construire un graphe G_S plus petit, appartenant toujours à $\mathcal{C}$ et qui ne satisfait pas P (le graphe G_S est généralement construit en supprimant dans G des sommets ou des arêtes formant S).

Si on peut trouver un ensemble $\mathcal{S}$ à la fois $\mathcal{C}$-inévitable et ne contenant que des configurations $(P, \mathcal{C})$-réductibles, alors on peut montrer la propriété P sur la classe $\mathcal{C}$. En effet, considérons un contre-exemple minimal $G \in \mathcal{C}$ à la propriété P. Comme $\mathcal{S}$ est $\mathcal{C}$-inévitable, G doit contenir une configuration de $\mathcal{S}$. Ce graphe est $(P, \mathcal{C})$-réductible, ce qui contredit la minimalité de G.

On peut voir cette méthode sous un autre angle, comme la construction d'un schéma d'induction *ad hoc*, spécifique à la classe $\mathcal{C}$. Soit $\mathcal{S}$ un ensemble $\mathcal{C}$-inévitable de configurations $(P, \mathcal{C})$-réductibles. Comme $\mathcal{S}$ ne contient que des configurations réductibles pour P et $\mathcal{C}$, on a :

- Le graphe vide satisfait P.

- Pour tout graphe G et toute configuration $S \in \mathcal{S}$ tels que G_S satisfait P, le graphe G satisfait P.

En d'autres termes, obtenir des configurations réductibles permet de montrer P par induction via ce schéma. On utilise alors le fait que $\mathcal{S}$ est $\mathcal{C}$-inévitable pour montrer que tout graphe de $\mathcal{C}$ est atteint par ce schéma (on dit alors qu'il est *complet*). Ainsi, on obtient ainsi que tout graphe de $\mathcal{C}$ satisfait P.

Le nom *déchargement* provient de la manière traditionnelle de montrer qu'un ensemble de configurations $\mathcal{S}$ est inévitable. L'idée générale est de considérer un graphe G ne contenant aucune configuration de $\mathcal{S}$, puis d'obtenir une contradiction en utilisant un argument de double comptage. On commence par attribuer un certain poids à des éléments de G, puis on les *décharge* via des règles adaptées et qui préservent le poids total. Ainsi, la somme des poids reste constante, et doit donc être la même avant et après le processus. Cependant, comme G ne contient aucune configuration de $\mathcal{S}$, on peut compter différemment la somme des poids finaux. En considérant une pondération initiale et des règles adaptées, on peut alors obtenir la contradiction recherchée.

Illustrons ce processus de déchargement sur l'exemple de la preuve de la formule d'Euler : pour tout dessin dans le plan d'un graphe planaire connexe G à v sommets, e arêtes et f faces, on a $v-e+f=2$. Dans cet exemple, on ne cherche pas une contradiction, mais seulement une pondération et des règles telles que le poids total initial soit $v-e+f$, et le poids total final soit 2.

Bien que le (même) argument soit valide quand G est un graphe planaire quelconque, il est plus simple de considérer seulement un graphe dont toutes les faces (y compris la face extérieure) sont des triangles. Ce n'est pas une hypothèse restrictive, car on peut toujours ajouter une arête à une face non triangulaire tout en préservant la valeur de $v-e+f$ (on ajoute une arête, et on transforme une face en deux faces). Considérons maintenant la pondération initiale suivante : chaque sommet et chaque face de G reçoit 1, et chaque arête reçoit -1. On peut facilement constater que la somme des poids est $v-e+f$.

On définit ensuite une seule règle de déchargement, à savoir que tout sommet et toute arête donne son poids à la face à sa droite. Ceci est ambigu quand une arête est horizontale, mais on peut éliminer ce cas en tournant G d'un bon angle. Le poids final des sommets et des arêtes est donc 0. Considérons une face interne xyz de G, où x est le sommet le plus bas, et z le sommet le plus haut. On peut distinguer deux cas, selon que y est à gauche ou à droite du segment $[xz]$. Ces deux cas sont illustrés en Figure 1.

Dans le premier cas, y, xy et yz donnent leur poids à xyz, tandis que dans l'autre, seulement xz donne son poids. Ainsi, le poids total transféré est -1 et le poids final de xyz est nul. On peut effectuer la même analyse quand xyz est la face externe, sauf que dans ce cas le poids transféré est 1 (x, z, xz ou x, y, z, xy, yz donnent leur poids), ce qui donne un poids final de 2.

Comme la règle préserve le poids total, on obtient alors que le poids total initial $v-e+f$ est égal au poids total final 2, ce qui achève la preuve de la

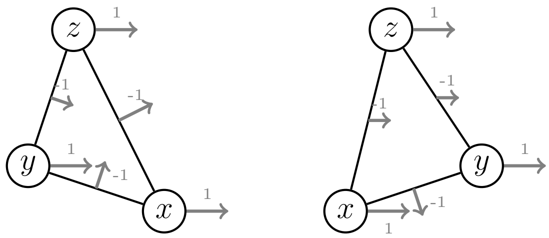


Figure 1 – Application de la règle à deux types de faces

formule d'Euler.

Cet argument est un exemple simple d'utilisation de cette méthode. La preuve du théorème des quatre couleurs repose sur une version beaucoup plus évoluée de ce genre d'arguments. Dans les chapitres 1 et 2, on présentera d'autres exemples, plus élaborés que pour la formule d'Euler (mais moins techniques que pour le théorème des quatre couleurs).

Organisation de la thèse

Cette thèse est divisée en deux parties, chacune d'entre elles s'intéressant à un domaine spécifique d'informatique théorique. Au lieu de présenter une introduction commune, on la divise en plusieurs parties : chaque chapitre contient les motivations et l'historique du problème qui y est considéré. De plus, nous préfaçons chaque chapitre en soulignant les contributions qui y sont présentées, ainsi que les articles qui en découlent et leurs auteurs.

Dans la première partie de cette thèse, on présente des résultats à propos de problèmes de coloration de graphes. Ces résultats sont obtenus en appliquant (des variantes de) la méthode de déchargement présentée ci-dessus.

Dans le chapitre 1, on s'intéresse à la *coloration totale par listes*, une variante de la coloration de graphes où on souhaite aussi colorer les arêtes, ayant aussi des restrictions sur quelles couleurs peuvent être utilisées sur chaque sommet. On s'appuie ici sur la méthode de déchargement, c'est-à-dire qu'on définit un schéma d'induction spécifique pour montrer une borne supérieure sur le nombre de couleurs nécessaires. Plus précisément, on montre que tout graphe planaire de degré maximum $\Delta \geqslant 8$ est totalement $(\Delta + 2)$-colorable par listes (Théorème 1.11). Ceci étend les résultats précédemment connus à l'ensemble des graphes de degré maximum 8. L'intérêt de ce chapitre ne se limite pas à ce résultat : il provient aussi des méthodes génériques utilisées pour montrer la réductibilité des configurations.

Le chapitre 2 considère un autre type de coloration, paramétré par un entier k, où on impose à toute paire de sommets à distance au plus k de recevoir des couleurs différentes. En guise de mise en bouche, on montre d'abord qu'on peut économiser $k - O(1)$ couleurs par rapport à la borne supérieure naïve lorsqu'on colore un graphe quelconque (Théorème 2.3). On s'intéresse ensuite au cas $k = 2$ pour les graphes planaires, et on caractérise quels cycles doivent être interdits pour obtenir une majoration en $\Delta + O(1)$ du nombre de couleurs nécessaires pour colorer à distance 2 les graphes planaires de degré maximum Δ. En particulier, on montre que lorsque Δ est assez grand, tout graphe planaire sans C_4 peut être coloré à distance 2 avec $\Delta + 2$ couleurs (Théorème 2.15). On utilise ici une variante de la méthode de déchargement. On peut toujours y voir la création d'un schéma d'induction adapté. Cependant, cette fois, la partie intéressante est la complétude de ce schéma (*i.e.*, l'inévitabilité de l'ensemble de configurations), dont la preuve n'utilise pas de déplacement de poids.

La seconde partie de cette thèse s'intéresse à des questions d'expressivité de formalismes syntaxiques donnés. Une question emblématique dans ce domaine consiste à déterminer quelles propriétés définies sur une structure discrète peuvent être exprimées dans un formalisme descriptif donné. Cette question dépend donc de deux objets, à savoir :

- Un type de structure discrète, comme par exemple les arbres binaires étiquetés.
- Un formalisme descriptif, comme par exemple la logique du premier ordre.

On cherche alors à déterminer quels ensembles de structures peuvent être définis à l'aide du formalisme choisi. En particulier, l'exemple cité ci-dessus, bien qu'utilisant une structure et un formalisme courants, est déjà une question ouverte majeure dans ce domaine : quels ensembles d'arbres binaires étiquetés peuvent être définis en logique du premier ordre ?

Dans cette thèse, on considère une version plus simple (mais déjà non-triviale) de ce problème, en étudiant des structures discrètes plus simples, mais fondamentales : les mots finis et infinis, au lieu des arbres. Dans ce cadre, le cas de la logique du premier ordre est déjà résolu : on peut caractériser les langages (c'est-à-dire les ensembles de mots) définissables dans cette logique. Cependant, le problème consistant à caractériser les langages définissables par des formules satisfaisant d'autres restrictions syntaxiques est toujours ouvert pour de nombreux types de restrictions naturelles.

Les résultats récents sur ce genre de questions sont obtenus en considérant un problème appelé $\mathcal{C}$-*séparation*, paramétré par une classe [1] de langages $\mathcal{C}$ représentant les langages définissables dans le modèle considéré. Ce problème prend deux langages L_1, L_2 en entrée et teste s'il existe $L \in \mathcal{C}$ qui sépare L_1 de L_2, c'est-à-dire tel que $L_1 \subset L \subset \overline{L_2}$. Comme on le verra plus tard, ce problème est plus général que le problème de définissabilité présenté ci-dessus.

1. La terminologie « classe » est utilisée dans cette thèse pour des raisons historiques, mais doit être comprise comme « ensemble ».

Dans cette thèse, on s'intéresse au problème de séparation sous deux aspects : décidabilité et complexité.

Une introduction plus détaillée de ces questions est fournie au chapitre 3. Le but de ce chapitre est tout d'abord d'introduire le domaine et le problème de séparation d'un point de vue historique. On y illustre le caractère robuste du problème de séparation à l'aide d'une première contribution (Théorème 3.37) à propos de sa complexité dans le cadre des langages dits « réguliers », c'est-à-dire reconnus par un automate fini, ou de manière équivalente, par un monoïde fini. Selon le formalisme choisi pour représenter des langages réguliers (automates ou monoïdes), la complexité des problèmes de définissabilité peut varier. C'est en particulier le cas pour le problème de définissabilité par des formules du premier ordre, qui est PSpace-difficile lorsque les langages d'entrée sont représentés par des automates, mais LogSpace s'ils le sont par des monoïdes. On montre en revanche que le comportement du problème de $\mathcal{C}$-séparation est tout autre : sa complexité ne dépend pas du type de représentation choisi pour ses entrées (quand $\mathcal{C}$ est raisonnable).

On s'intéresse ensuite à l'étude du problème de $\mathcal{C}$-séparation pour des classes $\mathcal{C}$ spécifiques. La plupart des classes considérées historiquement sont construites à partir d'une autre classe plus petite, en la clôturant par certaines opérations. Dans le cadre des langages réguliers, il est naturel d'utiliser des opérations préservant la régularité telles que la concaténation, les opérations booléennes, ou encore l'étoile (ces opérations seront définies dans le chapitre 3).

À partir d'une classe $\mathcal{C}$, on peut par exemple définir sa *clôture booléenne*, c'est-à-dire l'ensemble des langages obtenus comme unions d'intersections de langages de $\mathcal{C}$ ou de complémentaires de langages de $\mathcal{C}$. Une autre opération historique est donnée par la *clôture polynomiale* $\mathrm{Pol}(\mathcal{C})$ d'une classe $\mathcal{C}$. Informellement, il s'agit de l'ensemble des langages obtenus comme unions de concaténations de langages de $\mathcal{C}$. Ces deux opérations ont une importance considérable, d'un point de vue historique comme sémantique, qui sera détaillée dans les chapitres 3 et 4.

Le chapitre 4 s'intéresse exclusivement à l'opération de clôture polynomiale. On y considère le problème de $\mathrm{Pol}(\mathcal{C})$-séparation sous deux angles. On s'intéresse tout d'abord à la décidabilité du problème de $\mathrm{Pol}(\mathcal{C})$-séparation pour une classe $\mathcal{C}$ finie. Dans le cas des langages de mots finis, ce résultat est déjà connu (Corollaire 4.44, provenant de [Place and Zeitoun, 2017d]) : $\mathrm{Pol}(\mathcal{C})$-séparation est décidable quand $\mathcal{C}$ est une classe raisonnable finie. Même si ce résultat se limite à une classe $\mathcal{C}$ finie, sa preuve est non triviale et nécessite l'introduction d'objets et d'arguments élaborés (voir Théorème 4.43). La première contribution de ce chapitre est une borne inférieure générique sur la complexité de $\mathrm{Pol}(\mathcal{C})$-séparation quand $\mathcal{C}$ est une classe assez grande (Théorème 4.10). La seconde contribution consiste à étendre ces résultats lorsqu'on considère des langages de mots infinis (Théorème 4.67).

Introduction

This thesis is divided into two parts, each of them considering different objects and problems. Each problem will have its own detailed introduction in the corresponding chapter. Here, we first give an historical presentation of inductive methods. While these methods are quite common in theoretical computer science, this introduction is specifically oriented towards presenting the problems and tools considered in the first part. Presenting the field of the second part needs more definitions, and thus will be introduced later, in a dedicated chapter.

The tale of induction schemes

Once upon a time, in a country across mountains and seas, there lived a man called Bhāskara II. According to [Cajori, 1918], the *chakravala* method he introduced was one of the first examples of what can be classified as an inductive proof. This method is an algorithm finding an integer solution (x, y) of the Diophantine equations $x^2 - ny^2 = 1$, the so-called Pell-Fermat equations. While studies about this family of equations can be tracked back to the 6th century, the first complete solution (given by the chakravala method) was found only in the 12th century. This method was then rediscovered much later (in 1930) in Europe. In the meantime, Lagrange solved the problem independently, with a different algorithm, in 1767.

While not stated in nowadays formalism, many examples of proofs by induction can be traced throughout history and civilizations. For example, one may cite the work of Euclid on prime numbers, of Al-Karaji, al-Samaw'al and Ibn al-Haytham on combinatorics (sum of integers, squares, binomial theorem...), or even of Gersonides during Middle Ages. In the 17th century, many examples can be found in the books of Pascal, Fermat or Bernoulli. The formalization of this proof scheme became more precise with the work of Grassmann, and then Dedekind and Peano during the 19th century.

Say we want to prove that every natural integer n satisfies a property P. To this end, we first prove that P is verified at first level, i.e. for $n = 0$. Then, we prove that if P is satisfied for some integer n, then it is also satisfied for integer $n + 1$. For example, consider an infinite ladder we want to climb. If we know how to get on the first level, and we know how to climb from every level n to level $n+1$, then we can reach any level of the ladder. This proof scheme can be summarized as the following axiom of Peano's arithmetic.

$$\left.\begin{array}{l} P(0) \\ \forall n, P(n) \Rightarrow P(n+1) \end{array}\right\} \Rightarrow \forall n, P(n)$$

Proofs using induction are not restricted to integers: they can be extended to more complex structures, such as (rooted) trees. A tree is either a leaf, or consists in a root adjacent to an arbitrary but finite number of rooted trees. In this case, the goal is to prove a given property for single-node trees, and then to extend it from the children of a node to the node itself. This can be translated as an inductive proof on some integer parameters of trees, such as height or number of nodes. By extension, we also speak of induction schemes for defining some objects (such as trees). Inductive proofs are well-designed tools for studying inductive objects.

Induction proofs also extend (in a similar way) to sets endowed with a *well-founded order*, i.e. such that there is no strictly decreasing sequence of infinite length. We can give an alternative proof scheme using well-founded orders. While this is only a reformulation of the previous inductive proofs, it can obtain more readable proofs. We consider here two examples of this. The first historical one is the method of *infinite descent*. Its first appearance could be traced back to Euclid for his proof that every composite integer is multiple of some prime number. It was then popularized much later by Fermat to study Diophantine equations, in view of proving there is no solution. The main argument is the following: assuming that there is a solution s, we can construct another solution s' such that $s' < s$ for some well-founded order $<$. Therefore, the existence of a solution is incompatible with well-foundedness of $<$, ensuring that no solution exists. For Euclid's result, we use that the standard order on natural integers is well-founded: starting from a composite number n without any prime divisor, we construct an integer $n' < n$, also composite and without prime divisor. We thus obtain a strictly decreasing sequence of natural integers, a contradiction.

The second example is the so-called *discharging method*. It is well-suited to study graphs, and will be central in Chapters 1 and 2. A *graph* G is a discrete object given by a finite set of vertices $V(G)$ and a set $E(G)$ of edges, i.e. pairs of distinct vertices of the form uv for $u, v \in V(G)$ with $u \neq v$. One of the most famous problems in graph theory is the Four-Color conjecture, stated in 1852, and now a theorem. A *planar graph* is a graph that can be drawn in the plane without crossing edges. The Four-Color theorem states that we can assign 1, 2, 3 or 4 to each vertex of any planar graph (we call this integer the *color* of the vertex) such that the endpoints of each edge have a different color. The discharging method was introduced more than a century ago in [Wernicke, 1904] to tackle this problem. In this case, the idea is to consider a smallest counterexample to 4-colorability, and to use the minimality to get some structural information on this counterexample, in a similar fashion to the infinite descent method. We then reach a contradiction by proving that these information are never simultaneously satisfied. This method is especially well-suited for studying sparse graphs (such as planar graphs), and leads to many results, as shown in [Borodin, 2013; Cranston and West, 2017].

The generic setting of the discharging method can be explained as follows. Assume we want to prove a property P on a graph class $\mathcal{C}$. For simplicity, assume that the empty graph lies in $\mathcal{C}$ and satisfies property P. We look for some structural information about the graphs in $\mathcal{C}$. To this end, we describe some structures in graphs, named "configurations". Many examples of such configurations will be given in Chapter 1. We look for a set $\mathcal{S}$ of configurations satisfying two properties:

- $\mathcal{S}$ is $\mathcal{C}$-unavoidable, i.e. every non-empty graph of $\mathcal{C}$ contains at least one of the configurations in $\mathcal{S}$.
- Each configuration $S \in \mathcal{S}$ is $(P, \mathcal{C})$-reducible, i.e. for every graph G not satisfying P and containing S, we can construct a smaller graph G_S, which still lies in $\mathcal{C}$ and does not satisfies P. (The graph G_S is usually constructed by from G by removing some of the vertices or edges that form S.)

If we can find a $\mathcal{C}$-unavoidable set $\mathcal{S}$ of $(P, \mathcal{C})$-reducible configurations, then we can prove the property P. Indeed, consider a smallest counterexample G for property P in $\mathcal{C}$. Then, since $\mathcal{S}$ is $\mathcal{C}$-unavoidable, G has to contain a configuration in $\mathcal{S}$, which is $(P, \mathcal{C})$-reducible, yielding a contradiction.

This method can be seen as the construction of a custom induction scheme, which is specific to the class $\mathcal{C}$. Let $\mathcal{S}$ be a $\mathcal{C}$-unavoidable set of $(P, \mathcal{C})$-reducible configurations. Since $\mathcal{S}$ contains only reducible configurations for P, we have the following:

- The empty graph satisfies P.
- For each graph G and every configuration $S \in \mathcal{S}$ such that G_S satisfies P, the graph G also satisfies P.

In other words, obtaining reducible configurations allows us to prove P by induction with this custom induction scheme. Moreover, since $\mathcal{S}$ is $\mathcal{C}$-unavoidable, we know that every graph of $\mathcal{C}$ is constructed by this induction scheme (we say that the scheme is *complete*). Therefore, every graph of $\mathcal{C}$ satisfies P.

The name "discharging" comes from the usual way of proving unavoidability of a set of configurations $\mathcal{S}$. The idea is to consider a graph G containing none of the patterns of $\mathcal{S}$, and then to reach a contradiction by double counting a suitable quantity defined on the graph. To this end, we give some weights to some elements of G, and then "discharge" them using suitable rules that are bound to preserve the total weight. The sum of the weights should be the same before and after the process. However, since G does not contain any pattern of $\mathcal{S}$, we may be able to count differently the sum of the final weights. Taking the right initial weighting and suitable rules may yield the requested contradiction.

We illustrate the discharging procedure on the example of Euler's formula: for every drawing of a connected planar graph G with v vertices, e edges and f faces, we have $v - e + f = 2$. In this case, we do not look for a contradiction, but for a weighting and rules such that the initial weighting has total weight $v - e + f$, and the final one has 2.

While the (same) argument actually works when G is any planar graph, it is easier to consider that every face of G is a triangle (even the external one). This is not restrictive since we can always add an edge in a non-triangular face without changing the value of $v - e + f$ (since we add an edge and transform a face into two). Now consider the following initial weighting: each vertex and face of G receives 1, and each edge receives -1. It is easy to check that the sum of all these weights is $v - e + f$.

We now define a single discharging rule, stating that every vertex and edge gives all its weight to the face on its right. This is ambiguous only if there are some horizontal edges, but we can always find a rotation of G such that this is not the case. The final weight of vertices and edges is then 0. Consider an internal face xyz of G, where x is the lowest vertex and z the highest one. There are two cases, depending on whether y is on the left of $[xz]$ or on its right. These are depicted in Figure 2.

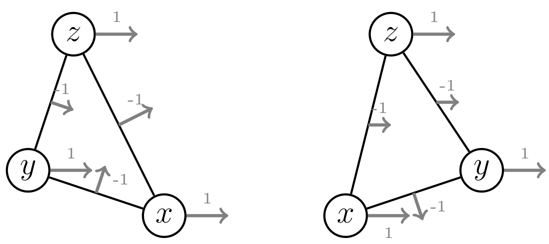


Figure 2 – Applying the rule to two types of faces

In the former, y, xy and yz give their weight to xyz. In the latter, only xz gives its weight to xyz. Thus, the total transferred weight is -1, and the final weight of xyz is 0. The same analysis holds when xyz is the external face, except that in this case, the transfer-ed weight is 1 (x, z, xz or x, y, z, xy, yz give weight), for a final weight of 2.

Since the rule preserves the total weight, we thus obtain that the initial total weight $v - e + f$ equals the final total weight 2. This proves Euler's formula.

This argument is a simple example of how to use this method. The proof of the Four-Color theorem relies on a much more involved version of this kind of arguments. In Chapters 1 and 2, we will present other examples of discharging proofs, more involved than for Euler's formula (but not as technical as for the Four-Color theorem).

And so the tale goes...

This thesis contains two parts, each of them considering a specific field of theoretical computer science. Instead of giving a common introduction, we divide it: each of the chapters will contain the motivations and history of the problem it considers. Moreover, we preface every chapter with a short text highlighting the contributions in it, together with the associated papers and authors.

In the first part of this thesis, we obtain some results about graph coloring problems by applying (variants of) the discharging method presented above.

Chapter 1 considers *total list coloring*, a variant of coloring where we also require for the edges to be colored, with some restrictions on which color can be used on which element. To this end, we rely on the discharging method we presented, i.e. we design a custom induction scheme for proving an upper bound on the number of colors needed. More precisely, we prove that every planar graph of maximum degree $\Delta \geqslant 8$ is totally $(\Delta+2)$-list-colorable (Theorem 1.11). This extends the previously known results to graphs with maximum degree 8. Interest in this chapter is not limited to this result: it also comes from the generic methods we use for proving reducibility of configurations.

Chapter 2 investigates another kind of coloring, parameterized by an integer k, where we require any two vertices within distance k to receive different colors. As a preliminary result, we first prove that one can spare $k-O(1)$ colors from the naive upper bound when coloring any given graph (Theorem 2.3), while the previously known result allowed to spare a constant number of colors. We then investigate the case $k = 2$ on planar graphs, and characterize which cycles have to be forbidden to obtain a $\Delta + O(1)$ upper bound for coloring planar graphs of maximum degree Δ at distance 2. In particular, we prove that for large enough Δ, every C_4-free planar graph can be colored at distance 2 with $\Delta + 2$ colors (Theorem 2.15). We use here a variant of the discharging method. It can still be seen as a custom induction scheme. However, this time, the interesting part comes from the completeness of this scheme (i.e., the unavoidability of the set of configurations), which is proven in a unusual way.

The second part of this thesis consider expressiveness problems of given syntactic formalisms. An emblematic question in this field is to determine which properties of a given discrete structure can be expressed in a given descriptive formalism. This question thus depends on two objects, namely:

- A type of discrete structure, for instance binary labeled trees.
- A descriptive formalism, for instance first-order logic.

We want to determine which sets of these structures can be defined using the chosen formalism. In particular, the aforementioned example, even if it uses a basic structure and formalism, is already a major open question in this field: which sets of binary labeled trees can be defined with first-order logic?

In this thesis, we consider a simpler (yet already non-trivial) version of this problem by studying simpler but fundamental discrete structures: finite and infinite words, instead of trees. In this setting, the case of first-order logic is already solved: we can characterize the languages (i.e., the set of words) definable in this logic. However, the problem of characterizing languages that are definable by sentences satisfying some other syntactic restrictions is still open for many types of natural restrictions.

The recent results about this kind of questions are obtained by considering a problem called *$\mathcal{C}$-separation*, which is parameterized by a class [2] of languages $\mathcal{C}$ representing the languages definable using the considered formalism. This problem takes two languages L_1, L_2 as input an tests whether there exists $L \in \mathcal{C}$ separating L_1 from L_2, i.e. such that $L_1 \subset L \subset \overline{L_2}$. As we will see, this problem is more general than the definability problem stated above. In this thesis, we investigate the separation problem from two points of view: decidability and complexity.

We give a more detailed introduction to these questions in Chapter 3. The goal of this chapter is twofold: first, we introduce the field and the separation problem from a historical point of view. We illustrate the robustness of the separation problem with a first contribution (Theorem 3.37) regarding its complexity when considering "regular" languages, i.e. the ones recognized by a finite automaton, or equivalently by a finite monoid. Depending on the formalism chosen to represent regular languages (automata or monoids), the complexity of definability problems may vary. It is in particular the case for definability by first-order logic, which is PSpace-hard when the input languages are represented by automata, but LogSpace when they are given by monoids. We prove that the behavior of the $\mathcal{C}$-separation problem is different: its complexity does not depend on how its inputs are represented (when $\mathcal{C}$ is nice enough).

We then consider the $\mathcal{C}$-separation problem for specific classes $\mathcal{C}$. Most of the historically considered classes are built from another smaller class by clos-

2. We use here the terminology "class" for historical reasons, but it is to be understood as "set".

ing it under some operations. For regular languages, it is natural to consider operations preserving regularity, such as concatenation, Boolean operations, or star (which will be defined in Chapter 3).

Starting from a class $\mathcal{C}$, we may for example define its *Boolean closure*, i.e. the set of languages obtained as unions of intersection of languages in $\mathcal{C}$ or complements of languages in $\mathcal{C}$. Another historical operation is given by the *polynomial closure* $\mathrm{Pol}(\mathcal{C})$ of a class $\mathcal{C}$. Informally, it is the set of languages obtained as unions of concatenations of languages in $\mathcal{C}$. These two operations are very important, from historical and semantical points of view, and will be detailed in chapters 3 and 4.

Chapter 4 is devoted only to the polynomial closure operation. We consider the problem of $\mathrm{Pol}(\mathcal{C})$-separation under two points of view. We first investigate the decidability of this problem when $\mathcal{C}$ is a finite class. For languages of finite words, this result is already known (Corollary 4.44, coming from[Place and Zeitoun, 2017d]): $\mathrm{Pol}(\mathcal{C})$-separation is decidable when $\mathcal{C}$ is a reasonable finite class. Even if this result considers only a finite class $\mathcal{C}$, its proof is not trivial, and relies on the introduction of involved objects and arguments (see Theorem 4.43). The first contribution of this chapter consists in a generic lower bound on this complexity when $\mathcal{C}$ is a large enough class (Theorem 4.10). A second contribution extends both results when considering languages of infinite words (Theorem 4.67).

Chapter 1

An example of what (not) to do: the raw power of discharging

Contents

This chapter is joint work with Marthe Bonamy and Éric Sopena ([Bonamy et al., 2019b]). It uses the discharging method. We rely on three standard techniques to reduce configurations: the Combinatorial Nullstellensatz, case analysis, and recoloring. For the third one, we use here a specific framework, adapted from [Bonamy, 2015]. This framework (the so-called color-shifting graph) seems to appear only in these two papers.

1.1 Introduction

The chromatic number χ of a graph G is the minimum number of colors needed to color every vertex of G such that no two adjacent vertices receive the same color. Edge coloring is a variant of graph coloring where edges (instead of vertices) are to be colored. We ask for every two incident edges to receive different colors. The minimum number of colors needed to color a graph is called its *chromatic index* and is denoted by χ'. Given a graph G, an edge-coloring of G can actually be seen as a vertex coloring of its line graph, i.e. the graph whose vertices are edges of G, and where the edge ef is present if e and f have a common endpoint in G. Therefore, the edge-coloring problem is a restriction of the coloring problem to a subclass of graphs. However, while there are some interesting characterizations of line graphs in terms of forbidden induced subgraphs (see [Beineke, 1970]), the transformation of a graph into its line graph does not preserve well the properties of G, like planarity. Therefore, we rather consider edge-coloring.

Like its vertex analogue, this kind of coloring has many applications in scheduling problems. Imagine a tournament with n entrants where each player has to play against every other one. One may ask the minimum duration of the tournament, assuming all the games last the same time. This problem is equivalent to finding the chromatic index of the clique K_n. Indeed, each edge of K_n represents a game that has to be played at some point. A color class is a set of games that can be played simultaneously, and the chromatic index is the minimum number of rounds needed to ensure that each game has been played. While the case of K_n is solved (the solution is $n-1$ or n depending on the parity of n), it is not the case when we consider a general graph G.

A first observation about the chromatic index of graphs is that all the edges incident to a given vertex have to receive pairwise distinct colors. This implies that every proper edge-coloring of a graph G uses at least $\Delta(G)$ colors, where $\Delta(G)$ is the maximum degree of G. This is a first difference from the vertex coloring case, since there are 2-colorable graphs of arbitrarily large maximum degree. We may consider how the classical results about (vertex) coloring translate in this new setting. A first result is Brooks' theorem that characterizes the graphs achieving the greedy upper bound for χ.

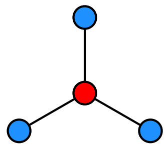
Theorem 1.1 ([Brooks, 1941]). *Every connected graph G is $\Delta(G)$-vertex-colorable, except when G is a clique or an odd cycle.*

In any edge-coloring of a graph G, each edge is adjacent to at most $2\Delta(G)-2$ other edges, thus by a greedy argument, we obtain that $\chi'(G) \leqslant 2\Delta(G)-1$. While a greedy coloring gives a polynomial 2-approximation for the edge coloring problem, Vizing's theorem states that the bound is actually far from being tight.

Theorem 1.2 ([Vizing, 1964]). *Every graph G satisfies $\chi'(G) \leqslant \Delta(G)+1$, and a $(\Delta(G)+1)$-edge-coloring of G can be found in polynomial time.*

Moreover, a Brooks-like result about characterizing equality in Vizing's theorem is hopeless, since deciding whether $\chi'(G)$ is $\Delta(G)$ or $\Delta(G)+1$ is an NP-complete problem, even for cubic graphs (see [Holyer, 1981]).

While maybe less important, an interesting property of edge-colorings is that we may always assume that the color classes are balanced (see Theorem 1.3). This emphasizes again the different behaviors of vertex and edge coloring, since this property is clearly false for vertex coloring (consider the star $K_{1,3}$ on the right).



Theorem 1.3 ([Folkman and Fulkerson, 1966]). *Every graph G has a $\chi'(G)$-edge-coloring where the sizes of every two color classes differ by at most one.*

An interesting variation of the coloring problem can be obtained by considering list coloring: instead of assigning colors from $\{1, \ldots, k\}$ to elements of the graphs, we associate to each element a list of available colors and we color each element with a color from its own list. The question is now to determine the minimum size k of the lists such that G is k-choosable (no matter which lists of size k are considered), i.e. such that G has a proper coloring using only available colors, regardless of the list assignment. We thus obtain two parameters: the choosability number χ_ℓ and the choosability index χ'_ℓ. Again, the situation is quite different from vertex choosability. Indeed, note that there are 2-colorable graphs with arbitrarily large χ_ℓ, see Figure 1.1. However, we do not know any graph G such that $\chi'_\ell(G) \neq \chi'(G)$. This motivates the following conjecture.

Conjecture 1.4 (List edge coloring conjecture [Borodin *et al.*, 1997a; Juvan *et al.*, 1998; Vizing, 1976]). *Every graph G satisfies $\chi'(G) = \chi'_\ell(G)$.*

The following similar conjecture has also been introduced, as an extension of Vizing's theorem.

Conjecture 1.5 ([Vizing, 1976]). *Every graph G satisfies $\chi'_\ell(G) \leqslant \Delta(G)+1$.*

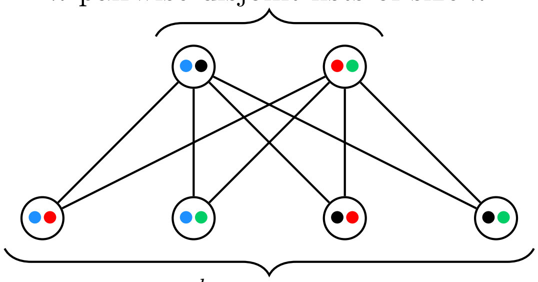


Figure 1.1 – A graph G with $\chi(G) = 2$ and $\chi_\ell(G) > k$ (here $k = 2$).

In the generic case, only few results are known towards these conjectures. The first improvement of the greedy upper bound $2\Delta - 1$ comes from [Hind, 1988] proving that $2\Delta - 2$ colors are enough. Using probabilistic arguments, this bound has been improved to $c\Delta + o(\Delta)$ for several values of c: $\frac{11}{6}$ [Bollobás and Harris, 1985], $\frac{9}{5}$ [Hind, 1988], $\frac{7}{4}$ [Bollobás and Hind, 1989], and finally $c = 1$. In this case, the error term has been refined several times, culminating with the following result.

Theorem 1.6 ([Kahn, 1996; Häggkvist and Janssen, 1997; Molloy and Reed, 2000])**.** *There exists a polynomial P such that, for every graph G, $\chi'_\ell(G) \leqslant \Delta(G) + P(\log \Delta(G))$.*

This implies that $\chi'_\ell(G) \leqslant \chi'(G) + o(\chi'(G))$ when $\chi'(G) \to \infty$, and thus gives an analogous of Theorem 1.6 for the list edge coloring conjecture. Apart from these results, we can consider specific classes of graphs, which allows us to use specific methods to deal with them. For example, for sparse graphs, we can use the discharging method as an additional tool, which leads to many results. A first class to consider is the one of planar graphs. In this case, Vizing's theorem can be strengthened: planar graphs are Δ-edge-colorable whenever $\Delta \geqslant 7$ [Vizing, 1965; Sanders and Zhao, 1999; Zhang, 2000]. However, for $\Delta \leqslant 5$, there are some planar graphs needing $\Delta + 1$ colors, see [Borodin, 2013] and Figure 1.2. The last open case is then planar graphs of maximum degree 6, for which only partial results are known, see [Borodin, 2013] for more details.

The situation is maybe more interesting in the edge choosability setting. For choosability in the generic case, the $\Delta + 1$ upper bound is only conjectured. Even with the new tools coming with planar graphs, this bound is actually still unproved. However, partial results are known. For graphs with small maximum degree, the conjecture holds in the trivial case $\Delta \leqslant 2$. Using Brooks' theorem, it also holds for $\Delta = 3$ [Vizing, 1976]. An ad-hoc incremental construction settles the case $\Delta = 4$ [Juvan *et al.*, 1999]. On the other side, the discharging method settles the case $\Delta \geqslant 9$ [Borodin, 1990; Cohen and Havet, 2010], which has been extended to the following result using a new technique

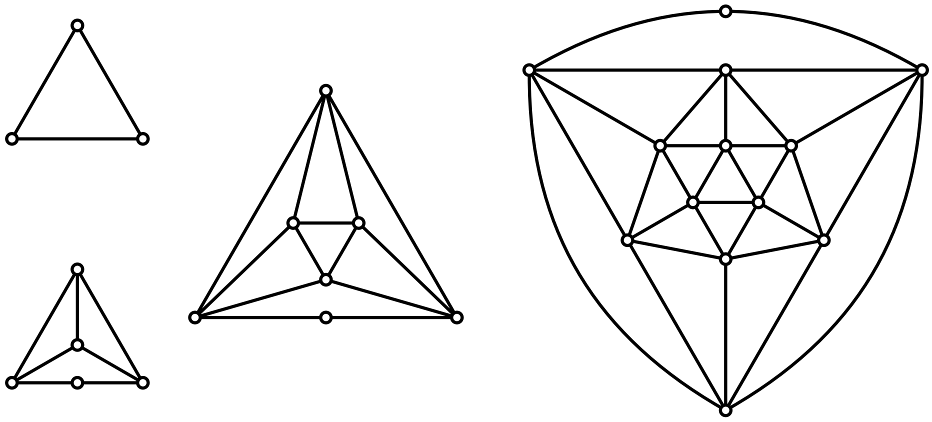


Figure 1.2 – Planar graphs with $2 \leqslant \Delta \leqslant 5$ that are not Δ-edge-colorable

we describe later in this chapter.

Theorem 1.7 ([Bonamy, 2015])**.** *Every planar graph with maximum degree* 8 *is* 9*-edge-choosable.*

Finally, using again discharging, we can even show that χ'_ℓ is at most Δ for planar graphs with $\Delta \geqslant 12$, see [Borodin, 1990; Borodin *et al.*, 1997a]. Many results improve these bounds when considering girth or cycle obstructions, and can be found in [Borodin, 2013].

Vertex and edge colorings can be combined to obtain total coloring. As expected, vertices and edges are to be colored, and every two adjacent or incident elements receive different colors. The parameter we obtain is called total chromatic number and denoted by χ''. We can also consider the list version of the problem, and the corresponding parameter is χ''_ℓ. These parameters share numerous similarities with their edge versions, as we shall see. First note that if we discard the colors of the vertices, we obtain a proper edge coloring, so $\chi' \leqslant \chi''$ and $\chi'_\ell \leqslant \chi''_\ell$. Moreover, we can link these parameters in another way: $\chi''_\ell \leqslant \chi'_\ell + 2$. Indeed, we can obtain a proper total list coloring of a graph G by first constructing greedily a vertex-coloring of G (since each vertex has at least $\chi'_\ell(G)+2 \geqslant \Delta(G)+1$ available colors). Then, for each edge e, we remove the colors of the endpoints of e from its list. The obtained lists have size at least $\chi'_\ell(G)$, thus we can find an edge coloring of G. The resulting coloring is a proper total coloring of G.

We can extend the upper and lower bounds of the edge coloring problem: every graph G needs at least $\Delta(G) + 1$ colors since the edges incident to a vertex of degree $\Delta(G)$ together with the vertex itself require pairwise distinct colors. Moreover, the greedy upper bound is almost unchanged: $2\Delta + 1$ colors are sufficient to color any graph of maximum degree Δ. On the other hand, Vizing's theorem does not extend directly to the total coloring setting. However, we do not know any graph G needing more than $\Delta(G)+2$ colors (i.e. only one more color than the lower bound). This is summarized in the following conjecture.

Conjecture 1.8 ([Behzad, 1965; Vizing, 1976]). *Every graph G satisfies $\chi''(G) \leqslant \Delta(G) + 2$.*

This conjecture is a relaxation of the following one, an analogue of the list edge-coloring conjecture to this new setting.

Conjecture 1.9 (Total list coloring conjecture [Borodin *et al.*, 1997a; Juvan *et al.*, 1998; Vizing, 1976]). *Every graph G satisfies $\chi''(G) = \chi''_\ell(G)$.*

Towards these conjectures, the probabilistic method still gives some results (that imply similar ones for edge colorings). Again, a $\Delta + o(\Delta)$ bound follows from the inequality $\chi''_\ell \leqslant \chi'_\ell + 2$. It has been improved several times [Hind, 1990; Chetwynd and Häggkvist, 1996] up to the following result.

Theorem 1.10 ([Molloy and Reed, 1998]). *There exists a constant C such that for every graph G, $\chi''(G) \leqslant \Delta(G) + C$.*

The constant C is at most 10^{26}, but may be brought down to roughly 100 [Molloy and Reed, 1998]. However, bringing down this bound to tackle Conjecture 1.8 seems to be out of reach from the probabilistic method for now. The situation is again more advanced when we consider planar graphs. We may again separate the results obtained for small Δ from those considering large Δ. Conjecture 1.8 clearly holds when $\Delta \leqslant 2$. Several proofs of the conjecture are known for subcubic graphs, using either custom induction or ad-hoc decompositions involving matchings and 2-factors by Petersen's theorem [Juvan *et al.*, 1998; Rosenfeld, 1971; Vijayaditya, 1971]. This idea of decomposing into 2-factors also settles the case $\Delta = 4$ [Kostochka, 1977], and has then been adapted to $\Delta = 5$ [Kostochka, 1996]. On the other hand, using discharging arguments, Conjecture 1.8 has been proved for $\Delta \geqslant 7$ [Borodin, 1987, 1989; Jensen and Toft, 1995; Sanders and Zhao, 1999].

Moreover, as in the edge-coloring case, planar graphs with large enough maximum degree need only $\Delta + 1$ colors for $\Delta \geqslant 9$, as shown by [Borodin, 1989; Borodin *et al.*, 1997a,b; Wang, 2007; Kowalik *et al.*, 2008].

Apart from those using decomposition in 2-factors and matchings, the proofs cited above are also valid in the list coloring setting. This implies that Conjecture 1.8 also holds for total choosability of planar graphs with $\Delta \leqslant 3$ [Juvan *et al.*, 1998] or $\Delta \geqslant 9$ [Borodin, 1987, 1989]. It can even be extended to $(\Delta + 1)$-choosability when $\Delta \geqslant 12$ [Borodin, 1987, 1989; Borodin *et al.*, 1997a].

Note that the bounds obtained on total (list) chromatic number seem only to be one more than the corresponding ones in the edge case. This may suggest a link between $\chi'(G)$ and $\chi''(G)-1$. However, the inequality $\chi'(G) \leqslant \chi''(G)-1$

is false, even for planar graphs, since $\chi'(K_3) = \chi''(K_3) = 3$. Moreover, we have $\chi'(K_4) = 3$ and $\chi''(K_4) = 5$, hence the converse inequality $\chi'(G) \geqslant \chi''(G) - 1$ does not hold either. However, we can still prove the inequality $\chi''(G) \leqslant \chi'(G) + 2$ when G is a planar graph (using the Four Color Theorem).

This chapter is devoted to the proof of the following result.

Theorem 1.11. *Every planar graph with maximum degree* 8 *is totally* 10-*choosable.*

This extends Theorem 1.7 to the total choosability setting, and settles Conjecture 1.8 for planar graphs with maximum degree 8. We use the discharging method, as done in all the proofs of Conjecture 1.8 for planar graphs with large enough maximum degree. We use several tools to reduce our configurations: the first one is an application of the polynomial method, via the Combinatorial Nullstellensatz. The other one has been introduced in [Bonamy, 2015] and is based on recoloring vertices to obtain more information on the coloring. Section 1.2 is devoted to presenting these two methods. We then apply them to reduce the configurations in Section 1.4 and present the discharging argument in Section 1.5.

1.2 Proof overview

We prove Theorem 1.11 by contradiction. Assuming that it has a counterexample, we consider the one with the smallest number of edges. Our goal is to prove that G satisfies structural properties incompatible with planarity, hence the conclusion. We consider G together with a planar embedding $\mathcal{M}$. Unless specified otherwise, all the faces discussed in the proof are faces in $\mathcal{M}$.

We first introduce a set of configurations in Section 1.3 and prove in Section 1.4 that they are not present in G. To this end, we use several techniques we introduce in this section. Then, we find a contradiction in Section 1.5 using the discharging method. This means that we assign some initial weights to vertices and faces of G, then we redistribute these weights, and obtain a contradiction by double counting the total weight. We present an appropriate collection of discharging rules, and then argue that every element of G ends up with non-negative weight while the total initial weight was negative. We thus reach the required contradiction.

1.2.1 A framework for reducing configurations

We now introduce the generic framework we use to prove that a given configuration is reducible. Reducing a configuration C_i means to take a list assignment L of G, to find a suitable subgraph G' of G (often constructed by removing elements of G creating C_i), and to extend any L-coloring of G' to G.

Since G is a minimum counterexample, we get a contradiction if G contains C_i.

There are two non-immediate steps in this proof scheme: first, we have to find the right subgraph G'. Then, the most difficult part is to extend the coloring. Note that in some cases, we may have to change the given coloring before extending it.

We first introduce some terminology. In the previous setting, a *constraint* for an element x of G (vertex or edge) is an already colored element y such that x and y are incident (or adjacent). The *total graph* of G is the graph denoted by $\mathcal{T}(G)$, whose vertices are $V(G) \cup E(G)$, and there is an edge between any two elements x and y such that x and y are adjacent vertices or incident elements of G. Observe that finding a total L-coloring of G is equivalent to finding an L-coloring of $\mathcal{T}(G)$.

Given an element x of G, we denote by $\widehat{x}$ the list of colors available for x after having colored G'. We denote by $\mathcal{T}(G \setminus G')$ the subgraph of $\mathcal{T}(G)$ induced by the elements that are not already colored, i.e. the elements of $G \setminus G'$. Note that extending the coloring from G' to G is equivalent to producing an L'-coloring of $\mathcal{T}(G \setminus G')$ where L' is defined by $L'(x) = \widehat{x}$ for every element x of $G \setminus G'$.

By definition, for every element x of $G \setminus G'$, we have $|\widehat{x}| \geqslant 10 - c_x$ where c_x is the number of constraints of x. We may only consider the worst case, as stated in the following remark.

Remark 1.12. We may assume that $|\widehat{x}| = 10 - c_x$ for every element x of $G \setminus G'$.

This observation applies every time we compute the number of available colors for each element. A similar observation allows us to assume when appropriate that, when we color an uncolored element x of G, the lists of all its neighbors in $\mathcal{T}(G)$ always lose a color.

Remark 1.13. Let x, y be adjacent elements in $\mathcal{T}(G)$. Unless otherwise stated (i.e. if we assume explicitly that the color of x does not appear in $\widehat{y}$, for example if $\widehat{x}$ and $\widehat{y}$ are disjoint), coloring x makes $|\widehat{y}|$ decrease by 1.

We sometimes *forget* elements. Forgetting x means that for every coloring of its neighbors in $\mathcal{T}(G)$, we can always find an available color for x. For example, this happens when x has more available colors than uncolored neighbors in $\mathcal{T}(G)$. Therefore, when we forget x, we postpone the coloring of x to the end of the coloring process: we implicitly assign a color to x when all the remaining elements are colored. We extend this terminology to lists of elements: forgetting $x_1, \ldots, x_p$ means that we forget x_1, then x_2, ..., then x_p (observe that the order matters). Note that we can always forget uncolored vertices of degree at most 4 in G, since they have at most eight neighbors in $\mathcal{T}(G)$.

1.2.2 Combinatorial Nullstellensatz

Most of the proofs of Section 1.5 rely on more or less involved case analyses, depending on the lists $\widehat{x}$. This may lead to rather long proofs. To deal with this issue, we introduce another approach to reduce the corresponding configurations. As we will see, this method relies on an algebraic criterion that can be computer checked. This leads to much shorter proofs, with the downside of not being human-checkable. We now describe how to reduce a given configuration, or more precisely how to extend a coloring from a subgraph of G to G itself. The method uses the Combinatorial Nullstellensatz stated below.

Theorem 1.14 ([Alon and Tarsi, 1992])**.** *Let $\mathbb{K}$ be a field, and $P \in \mathbb{K}[X_1, \dots, X_n]$ a multivariate polynomial. Let $X_1^{a_1} \cdots X_n^{a_n}$ be a monomial with a non-zero coefficient in P, and of maximal degree. Then, for any family $S_1, \dots, S_n$ of subsets of $\mathbb{K}$ satisfying $|S_i| > a_i$ for $1 \leqslant i \leqslant n$, there exists a non-zero value of P in $S_1 \times \cdots \times S_n$.*

While this result is stated in terms of polynomials and does not seem to be related to graphs, it has many applications in algebra, additive combinatorics, graph theory... Several examples of such applications can be found in [Alon, 1999]. It is one of the main tools of the so-called *polynomial method*. From a high-level point of view, the generic approach is the following:

- We represent the studied combinatorial object (e.g. a coloring) as a set E of values for some polynomial indeterminates.
- We define a polynomial P whose roots are the "bad" objects (e.g. improper colorings).

Showing that a "good" object exists thus reduces to finding a non-root of P in E. This is where Theorem 1.14 is used: it gives a set of sufficient conditions to ensure that such a non-root exists, and reduces the initial combinatorial problem to the search of a suitable monomial in P.

Let us now describe how we use this approach to reduce configurations. With each uncolored element x in G which is neither colored nor forgotten, we associate a polynomial variable X (we use the same letter but capitalized). We denote by $<$ the lexicographic order on the variables. The polynomial P_G is then defined as the product of all $(X - Y)$ when $X < Y$ and x and y are adjacent uncolored vertices of $\mathcal{T}(G)$. Using P_G, we can associate with each coloring of G (where colors are integers) a value, obtained by replacing in P_G each variable X with the color of the corresponding element x.

Moreover, due to the construction of P_G, this value is not 0 if and only if the corresponding coloring is proper, i.e. if the coloring of G' extends to G. Therefore, we now look for a non-zero value of P_G. In particular, applying Theorem 1.14 to the subsets $\widehat{x}$ gives a sufficient condition in terms of the monomials in P_G: to prove that the coloring extends from G' to G, it is sufficient to find a monomial m in P_G such that the three following conditions hold:

1. $\deg(m) = \deg(P_G)$.

2. $\deg_X(m) < |\widehat{x}|$ for every uncolored element x of G.

3. The coefficient of m in P_G is non-zero.

Therefore, proving that a configuration is reducible using the Combinatorial Nullstellensatz amounts to finding a suitable monomial in P_G. For the sake of readability, we do not state the polynomial P_G in each of the reduction proofs.

Note that we do not believe that finding a suitable monomial, as well as checking Condition 3, can be done without a computer. For the former problem, we use an exhaustive search algorithm that produces an output in a reasonable time on most of the instances, but not for all, hence we do not have a reduction proof using Combinatorial Nullstellensatz for each configuration. For checking Condition 3, a Maple code is available here [1].

Finally, observe that Theorem 1.14 is not an equivalence in general: a polynomial may satisfy the conclusion of the theorem even if it has no suitable monomial. However, we do not know whether there exist reducible configurations such that the associated polynomial contains no suitable monomial.

1.2.3 Recoloring approach

For some configurations, both case analysis and Nullstellensatz approaches fail. For these configurations, we use a third technique, introduced in [Bonamy, 2015]. This is based on the following idea. Take φ a coloring of a subgraph of G. Depending on φ, it may not always be possible to extend it to G. However, in this case, we can analyze why the extension fails and deduce some properties of φ. We use these properties to prove that we can first transform φ into another coloring ψ, and then hope for ψ to be easier to extend. The end of this section is devoted to presenting this method in more details.

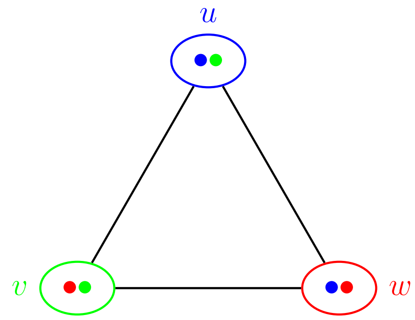
Our approach is based on recoloring some vertices in a given partial coloring, or at least to find sufficient conditions to be able to do so. We start with a preliminary definition. Let L be a list assignment on $\mathcal{T}(G)$ and γ a partial L-coloring of $\mathcal{T}(G)$. Let S be a properly colored clique in $\mathcal{T}(G)$. The *color shifting graph* of S with respect to γ is the loopless digraph $H_{S,\gamma}$ defined as follows (see Figure 1.3 for an example):

- Each element of S is a vertex of $H_{S,\gamma}$.

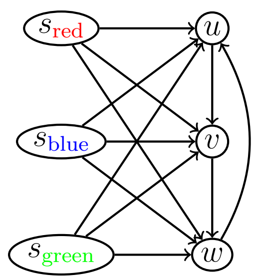
1. http://www.labri.fr/perso/tpierron/Delta8_check.txt

- We add a vertex s_α to $H_{S,\gamma}$ for each color $\alpha \in \cup_{x\in S}\widehat{x}$, where $\widehat{x}$ is the set of available colors for x when we uncolor S.
- If $x, y \in S$ with $x \neq y$, there is an arc $x \to y$ if the color of x lies in $\widehat{y}$ once S is uncolored.
- For any x, α, there is an arc $s_\alpha \to x$ if $\alpha \in \widehat{x}$ and $\alpha \notin \gamma(S)$ which means that the color α could replace the color of x.
- For any x, α, there is an arc $x \to s_\alpha$.
- For any $\alpha \neq \beta$, there is an arc $s_\alpha \to s_\beta$.

In Figure 1.3, we give a set S of three vertices inducing a triangle in $\mathcal{T}(G)$. For each node x, the list of colors depicted inside x is $\widehat{x}$, and the color of x is $\gamma(x)$. Since $H_{S,\gamma}$ is quite dense, we draw its complement in the figure, meaning that $H_{S,\gamma}$ contains all the non-loop arcs that are not present in Figure 1.3 below.



(a) A set S, with a coloring γ



(b) The **complement** of the graph $H_{S,\gamma}$

Figure 1.3 – Example of a color shifting graph

The terminology comes from the fact that any directed cycle in $H_{S,\gamma}$ allows us to shift the colors of the elements of S as stated in the following lemma.

Lemma 1.15. *Let L be a list assignment of $\mathcal{T}(G)$, let γ be a partial L-coloring of $\mathcal{T}(G)$ and S be a colored clique of $\mathcal{T}(G)$. Assume that there is a directed cycle $x_1 \to \cdots \to x_n \to x_1$ in the color shifting graph $H_{S,\gamma}$.*

Then there exists a partial L-coloring γ', defined on the same elements of $\mathcal{T}(G)$ as γ, and that differs from γ exactly on $S \cap \{x_1, \ldots, x_n\}$.

Proof. We define γ' by taking $\gamma'(x) = \gamma(x)$ for all the vertices x of S outside the directed cycle. It remains to define γ' on $S \cap \{x_1, \ldots, x_n\}$.

If none of the x_i's is some s_α, we move the colors following the arrows: for $1 \leqslant i \leqslant n$, we define $\gamma'(x_{i+1}) = \gamma(x_i)$ (the indices are taken modulo n). This is allowed since we have $\gamma(x_i) \in \widehat{x_{i+1}}$ by the definition of the arc $x_i \to x_{i+1}$.

Moreover, γ' is still a proper coloring since the color $\gamma(x_i)$ appears only on x_i in γ since S is a clique in G, hence it appears only on x_{i+1} in γ'.

Otherwise, we decompose the directed cycle into (maximal) directed paths of the form $s_\alpha \to x_1 \to \cdots \to x_p$. We then apply a similar approach to each of these paths: for $2 \leqslant i \leqslant p$, we define $\gamma'(x_i) = \gamma(x_{i-1})$. Similarly, this gives a proper coloring. It remains to color x_1. Note that $s_\alpha \to x_1$, so $\alpha \in \widehat{x_1}$ and $\alpha \notin \gamma(S)$. Therefore, we can take $\gamma'(x_1) = \alpha$ and keep a proper coloring.

Since we consider a cycle, for every α, the vertex s_α is the source of at most one such sub-path of the directed cycle. Therefore, color α appears in γ' on at most one vertex of S, and the coloring γ' is proper.

In both cases, we thus obtain a proper coloring γ' satisfying $\gamma'(x) \neq \gamma(x)$ for each vertex x of the considered directed cycle. □

We are now ready to describe the generic way used to reduce configurations in this approach. The framework is the same as before: our goal is to extend a coloring of a subgraph G' of G to the entire graph G. To this end, we first identify the conditions on the color lists impeding the coloring to extend directly to G. If these conditions are not satisfied, then we can extend the coloring and we are done.

On the contrary, if they are satisfied, we look for some elements of G' to recolor in order to change the available colors of the uncolored elements of G, and hence break the previous conditions. We finally use the previous lemma to reduce the initial problem to finding a suitable directed cycle in the color shifting graph of a well-chosen set of elements.

To find such directed cycles, we first state a simple but useful property of color shifting graphs: if $H_{S,\gamma}$ is the color shifting graph of a set S with respect to γ, then the in-degree of any vertex $x \in S$ of $H_{S,\gamma}$ is at least $|\widehat{x}|$. We often use this property together with the following lemma to find the required directed cycles. Recall that a strong component of $H_{S,\gamma}$ is a maximal set of vertices C such that any two of them are linked by a directed path in C.

Lemma 1.16. *Every directed graph H has a strong component C such that*

$$|C| > \max_{x \in C} d_H^-(x)$$

Proof. Consider the graph $\pi(H)$ obtained by contracting each strong component of H to a single vertex.

Note that $\pi(H)$ is an acyclic digraph, therefore it contains a vertex C of in-degree 0. Take $x \in C$. Then note that due to the definition of $\pi(H)$, for each arc $y \to x$, we also have $y \in C$. Therefore C contains every predecessor of x. Since G is a simple graph, there are $d^-(x)$ such predecessors, and x is not such a predecessor. Thus $|C| > d^-(x)$. This is valid for any $x \in C$, thus we obtain the result. □

Our goal is to prove that the elements we want to recolor are not alone in their strong component in the color shifting graph we consider (so that one of these elements is contained in a suitable directed cycle, and we can recolor it using Lemma 1.15). With the previous result, we have a dichotomy: if there is a strong component containing a vertex with large in-degree, then it is a large component, and it is likely to contain an element we want to recolor. Otherwise, we remove all the vertices with large in-degree and apply recursively the same dichotomy until (hopefully) a suitable directed cycle is found.

In order to applying this method, we need to compute the in-degree of every vertex in a color shifting graph. This is the goal of the last lemma of this section.

Lemma 1.17. *Let L be a list assignment of $\mathcal{T}(G)$, let γ be a partial L-coloring of $\mathcal{T}(G)$ and S be a colored clique of $\mathcal{T}(G)$. Let x be a vertex of $H_{S,\gamma}$. We have*

$$d^-(x) = \begin{cases} |\widehat{x}| - 1 & \textit{if } x \in S \\ |V(H_{S,\gamma})| - 1 & \textit{otherwise.} \end{cases}$$

Proof. Let $x \in V(H_{S,\gamma})$. If x is some s_α, then by definition, there is an arc $y \to x$ for every other vertex y of $H_{S,\gamma}$. Therefore, $d^-(x) = |V(H_{S,\gamma})| - 1$.

Otherwise, assume that $x \in S$. By definition, every predecessor of x is either an element of S colored with some color in $\widehat{x}$, or a vertex s_α with $\alpha \in \widehat{x} \setminus \gamma(S)$. Observe that since γ is proper and S is a clique, then for every $\alpha \in \gamma(S)$, there is exactly one vertex of S colored with α. In particular, there is no vertex $y \neq x$ with $\gamma(y) = \gamma(x)$. Therefore, there is one predecessor of x for every color of $\widehat{x} \setminus \{\gamma(x)\}$.

Conversely, let $\alpha \in \widehat{x} \setminus \{\gamma(x)\}$. If α does not appear on S, i.e. $\alpha \notin \gamma(S)$, then we have an arc $s_\alpha \to x$ in $H_{S,\gamma}$. Otherwise, $\alpha = \gamma(y)$ for some $y \in S \setminus \{x\}$, and we have an arc $y \to x$ in $H_{S,\gamma}$.

Therefore, the number $d^-(x)$ of predecessors of x is $|\widehat{x} \setminus \{\gamma(x)\}| = |\widehat{x}| - 1$. □

1.3 Configurations

In this section, we first introduce some terminology and then present our configurations.

1.3.1 Notation

We say that a vertex is *triangulated* if all the faces containing it are triangles (recall that we fix a planar embedding of G). Given a vertex u and two of its neighbors v_1, v_2, the *triangle-distance between v_1 and v_2 around u*, denoted by $\mathrm{dist}_u(v_1, v_2)$, the (possibly infinite) distance between v_1 and v_2 in the subgraph of G induced by the edges vw such that uvw is a triangular face (see Figure 1.4). This distance is the minimum of the lengths of (at most) two paths in the neighborhood of u, each one turning in one direction. In all the following, a k^-- (resp. k^+-) vertex is a vertex of degree at most (resp. at least) k. Moreover, in the figures, a node containing an integer i represents a vertex with degree i. An empty node is a vertex with no degree constraint. Moreover, observe that all the edges incident to the depicted vertices are not necessarily drawn. Moreover, for configurations, the drawing does not necessarily corresponds to the chosen embedding of the graph. When we reduce a configuration, we give a figure with the name of all the elements we will have to color. It may happen that not every element has a name (meaning that we keep its color from a coloring obtained using minimality). In this case, the corresponding element will be depicted in boldface.

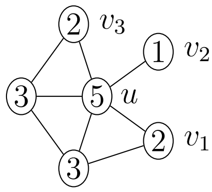


Figure 1.4 – Triangle-distance: $d_u(v_1, v_2) = \infty$ and $d_u(v_1, v_3) = 3$.

Given an edge uv, we say that v is:

- a *weak neighbor* of u if either v is a 4^--vertex and both faces containing the edge uv are triangles, or v is a triangulated 5-vertex (see Figure 1.5).

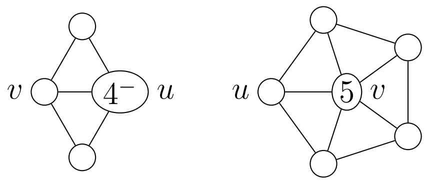


Figure 1.5 – A weak neighbor v of u.

- a *semi-weak neighbor* of u if v is a 4^--vertex and exactly one of the faces containing uv is a triangle (see Figure 1.6).

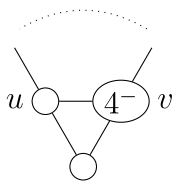


Figure 1.6 – A semi-weak neighbor v of u.

Moreover, if v is a weak neighbor of u, we often consider the degree of the common neighbors of u and v. We thus define the following: for any integers $p \leqslant q$, we say that v is a (p, q)*-neighbor* of u if v is a weak neighbor of u and the two vertices w_1, w_2 such that uvw_1 and uvw_2 are triangular faces have degree p and q, respectively. The same holds with p^+ (resp. p^-), meaning that the degree is at least (resp. at most) p.

We also define special types of 5-vertices. Consider a 7-vertex u with a weak neighbor v of degree 5. We say that v is:

(i) an S_3*-neighbor* of u if one of the following conditions holds (see Figure 1.7):

- v is a $(6, 6^+)$-neighbor of u.
- v is a $(7^+, 7^+)$-neighbor of u and v has two neighbors w_1, w_2 such that $d(w_1) = d(w_2) = 6$ and uvw_1, uvw_2 are not triangular faces.
- v has a neighbor w of degree 5 such that uvw is not a triangular face.

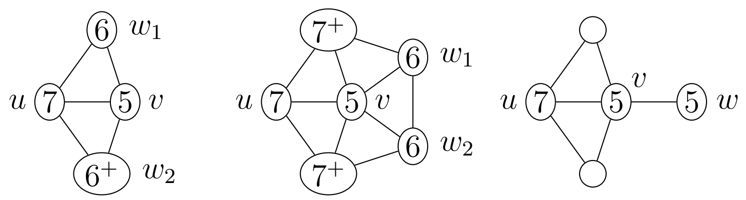


Figure 1.7 – v is an S_3-neighbor of u

(ii) an S_5*-neighbor* of u if every neighbor of v has degree 7.

(iii) an S_6*-neighbor* of u if it is not a $(5, 6)$-neighbor of u, nor an S_3-neighbor nor an S_5-neighbor.

We give a similar definition when u is an 8-vertex with a weak neighbor v of degree 5. We say that v is an E_3*-neighbor* of u if one of the following conditions holds (see Figure 1.8):

- v is a $(6, 7^+)$- or $(7, 7)$-neighbor of u.
- v is a $(7^+, 8)$-neighbor of u and v has two neighbors w_1, w_2 such that $d(w_1) = d(w_2) = 6$ and uvw_1, uvw_2 are not triangular faces.
- v is a $(7^+, 8)$-neighbor of u and v has a neighbor w of degree 5 such that uvw is not a triangular face.

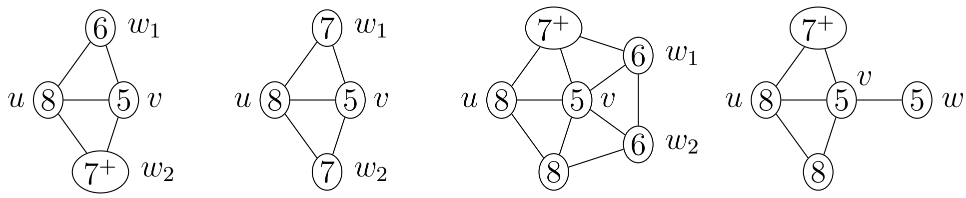


Figure 1.8 – v is an E_3-neighbor of u

1.3.2 The configurations

We now present several configurations, defined as all the sub-configurations of the forthcoming nineteen configurations C_1 to C_{19}. A configuration C is a *sub-configuration* of C' if we can obtain C by decreasing the degree of vertices in C' while preserving the adjacency relation and the triangle-distance: for every vertices x, y, z, the vertices x and y are adjacent in C if and only if they are in C' and $\text{dist}_z(x, y)$ is the same in C and C'. For example, a path uvw where $d(u) = d(v) = d(w) = 4$ is a sub-configuration of C_4 but a path $u_1u_2u_3u_4$ is not a subconfiguration of C_2 even if $d(u_1) = d(u_3) = 3$ and $d(u_2) = d(u_4) = 8$.

- C_1 is an edge (u, v) such that $d(u) + d(v) \leqslant 10$ and $d(u) \leqslant 4$.
- C_2 is an even cycle $v_1 \cdots v_{2n}v_1$ such that for $1 \leqslant i \leqslant n$, $d(v_{2i-1}) \leqslant 4$ and $d(v_{2i-1}) + d(v_{2i}) \leqslant 11$.
- C_3 is a triangle with two vertices of degree 5 and one of degree 6.
- C_4 is a vertex of degree 5 with two neighbors of degree 5.
- C_5 is a 7-vertex u with a $(5, 6)$-neighbor v_1 and a 5-neighbor v_2 such that either $\text{dist}_u(v_1, v_2) = 2$, or v_2 is a $(5, 6)$-neighbor of u with $\text{dist}_u(v_1, v_2) \leqslant 3$, see Figure 1.9.

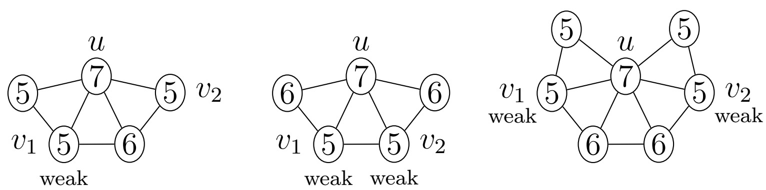


Figure 1.9 – Configuration C_5

- C_6 is a 5-vertex u adjacent to three 6-vertices v_1, v_2, v_3 and two vertices v_4, v_5 such that either there are two edges v_1v_2 and v_2v_3 or u is triangulated and $d(v_4) = d(v_5) = 7$, see Figure 1.10.
- C_7 is a 7-vertex u with a $(5, 6)$-neighbor of degree 5 and a neighbor of degree 4, see Figure 1.11.

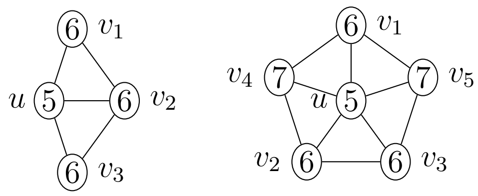


Figure 1.10 – Configuration C_6

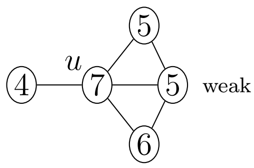


Figure 1.11 – Configuration C_7

- C_8 is a 7-vertex u with an S_3-neighbor v_1, a $(7, 7^+)$-neighbor v_3 of degree 4, and a neighbor v_4 of degree 5 such that $\text{dist}_u(v_1, v_3) = 2$ and the common neighbor v_2 of u, v_1, v_3 has degree 7, see Figure 1.12.

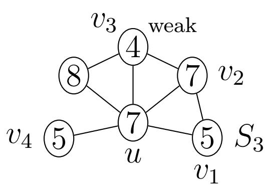


Figure 1.12 – Configuration C_8

- C_9 is a 7-vertex u with a weak neighbor v_1, a $(7, 7^+)$-neighbor v_2 of degree 4 and a weak neighbor $v_3 (\neq v_2)$ such that $\text{dist}_u(v_1, v_2) = \text{dist}_u(v_1, v_3) = 2$ and either v_1 is a S_3-neighbor of u, or it is an S_5-neighbor of u such that the common neighbor of u, v_1, v_2 has degree 7, see Figure 1.13.

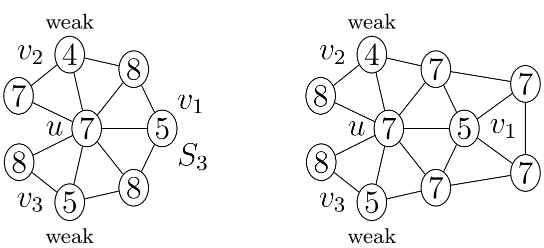


Figure 1.13 – Configuration C_9

- C_{10} is a 7-vertex u with three weak neighbors of degree 4 and a neighbor of degree 7.

- C_{11} is a 7-vertex u with a $(7,7^+)$-neighbor v_1 of degree 4, two weak neighbors v_2 and v_3 of degree 4 and 5 such that $\mathrm{dist}_u(v_1,v_2) = 2$ and either u, v_1 and v_3 have a common neighbor of degree 7, or v_3 is an S_3-neighbor of u such that $\mathrm{dist}_u(v_1,v_3) = 3$, see Figure 1.14.

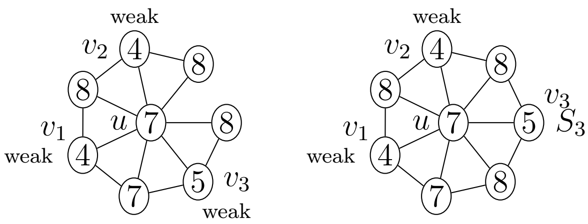


Figure 1.14 – Configuration C_{11}

- C_{12} is a 7-vertex u with two weak neighbors v_1, v_2 of degree 4 satisfying $\mathrm{dist}_u(v_1,v_2) > 2$, and a weak neighbor v_3 of degree 5 such that either v_1 is a $(7,7)$-neighbor of u, or so is v_3, or v_1 is a $(7,7^+)$-neighbor and v_3 is an S_3-neighbor of u, see Figure 1.15.

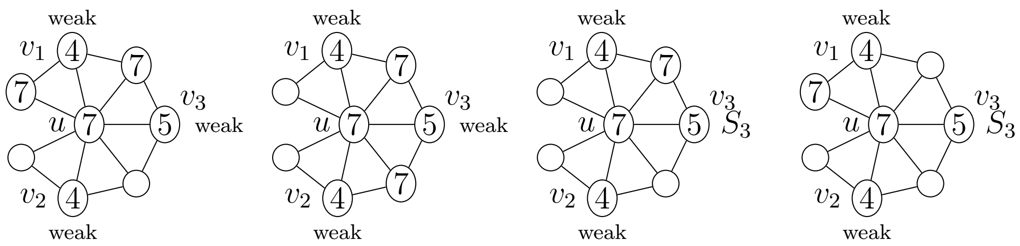


Figure 1.15 – Configuration C_{12}

- C_{13} is an 8-vertex u with either five weak neighbors of degree 5 and three neighbors of degree 6, or two $(6,6)$-neighbors of degree 5 with two weak neighbors of degree 5 and four neighbors of degree 6, see Figure 1.16.

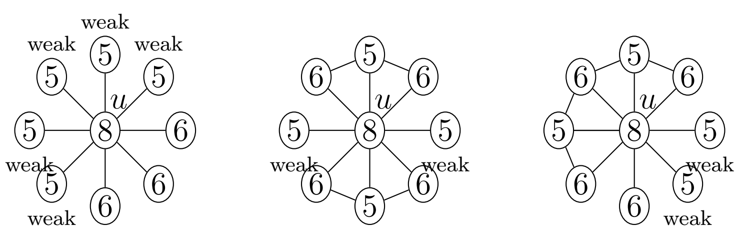


Figure 1.16 – Configuration C_{13}

- C_{14} is an 8-vertex u with four pairwise non-adjacent neighbors v_1, v_2, v_3, v_4 of degree 4 or 5 such that one of the following holds (see Figure 1.17):

- v_1, v_2, v_3, v_4 are weak neighbors of degree 4 and u has a neighbor of degree 7,
- v_1, v_2, v_3, v_4 are $(7,8)$-neighbors, and at most one of them has degree 5
- v_1 is a $(7,7)$-neighbor of degree 4, and v_2 is a weak neighbor of u of degree 4 such that $\text{dist}_u(v_1, v_2) = \text{dist}_u(v_1, v_4) = 2$.

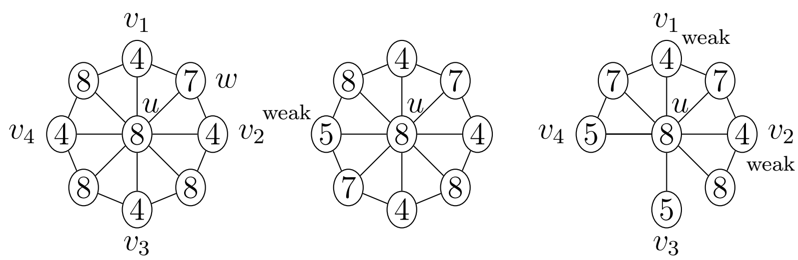


Figure 1.17 – Configuration C_{14}

- C_{15} is an 8-vertex u with a weak neighbor of degree 3 and either two $(6,6)$-vertices of degree 5, or three weak neighbors of degree 5 and two neighbors of degree 6, see Figure 1.18.

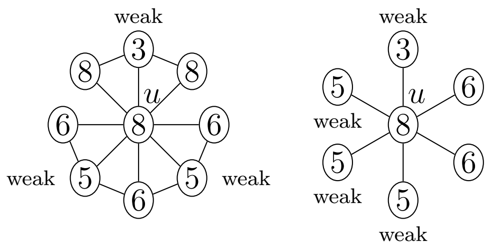


Figure 1.18 – Configuration C_{15}

- C_{16} is an 8-vertex u with a weak neighbor of degree 3 and:
  - either a $(7,8)$-neighbor of u of degree 4 and two weak neighbors of degree 5 not in a triangular face in u
  - or at triangle-distance 2 from a weak neighbor of u of degree 5 with two neighbors of degree 6, see Figure 1.19.
- C_{17} is an 8-vertex u with a weak neighbor of degree 3, a weak neighbor of degree 4, and either a $(6,6)$-neighbor of degree 5, or two weak neighbors of degree 5, one of them being an E_3-neighbor, see Figure 1.20.
- C_{18} is an 8-vertex u with two weak neighbors of degree 3 and another neighbor of degree at most 5, see Figure 1.21.

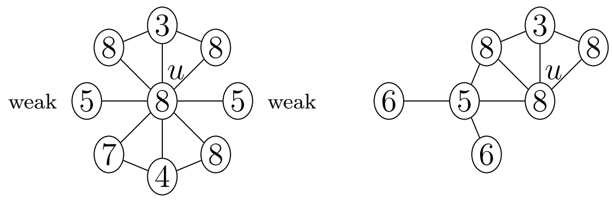


Figure 1.19 – Configuration C_{16}

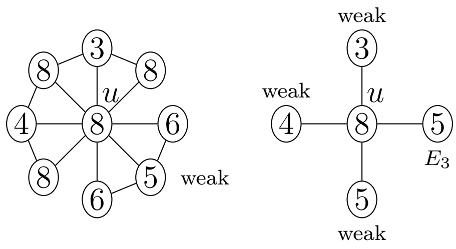


Figure 1.20 – Configuration C_{17}

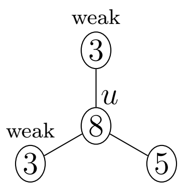


Figure 1.21 – Configuration C_{18}

- C_{19} is an 8-vertex u with a weak and a semi-weak neighbor v_1, v_2 of degree 3 and adjacent to two vertices w_1, w_2 such that $(d(w_1), d(w_2))$ is $(4, 7)$ or $(5, 6)$, see Figure 1.22.

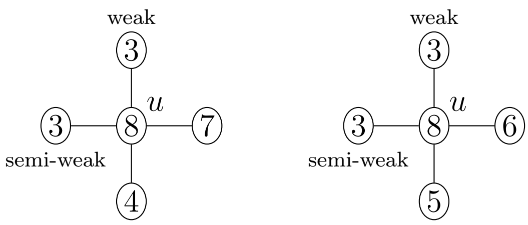


Figure 1.22 – Configuration C_{19}

- C_{20} is an 8-vertex u with a weak neighbor of degree 3 and four neighbors of degree $4, 4, 5, 7$, see Figure 1.23.

- C_{21} is an 8-vertex u with a weak neighbor of degree 3, two weak neighbors of degree 4 and a neighbor of degree 7, see Figure 1.24.

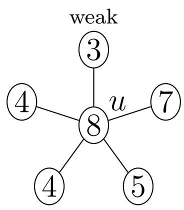


Figure 1.23 – Configuration C_{20}

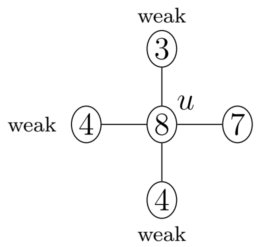


Figure 1.24 – Configuration C_{21}

- C_{22} is an 8-vertex u with a neighbor of degree 7 and either three semi-weak neighbors of degree 3 or two semi-weak neighbors of degree 3 and two neighbors of degree 4, see Figure 1.25.

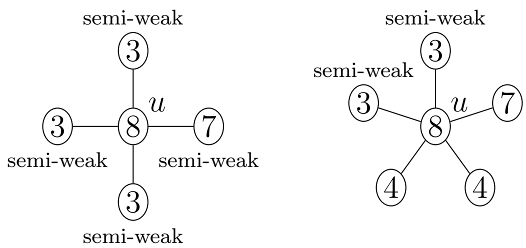


Figure 1.25 – Configuration C_{22}

1.4 Reducing configurations

This section is devoted to the proofs that each configuration is reducible, i.e. that G does not contain them. We first introduce some generic arguments we use to handle small cases.

1.4.1 Generic arguments

Recall that proving that a configuration is reducible amounts to extending a coloring of a subgraph G' of G to the entire graph G. This can be rephrased in terms of *f-choosability*. This variant of the choosability problem is defined as follows. Let H be a graph and $f : V(H) \to \mathbb{N}$. We say that H is f-choosable if we can produce a vertex L-coloring of H from any list assignment L satisfying $|L(v)| \geqslant f(v)$ for every vertex v of G.

To extend a coloring from G' to G, we often prove that $\mathcal{T}(G \setminus G')$ is f-choosable, where $f(x)$ is the number of available colors of the element x (in our case, $f(x)$ is ten minus the number of elements of G' incident to x). This point of view gives another tool to extend colorings, as shown by the following theorem.

Theorem 1.18 ([Borodin *et al.*, 1997a; Erdős *et al.*, 1979])**.** *Let G be a connected graph such that none of its blocks is a complete graph or an odd cycle. For any function $f : V(G) \to \mathbb{N}$ such that $f(v) \geqslant d(v)$ for each vertex v, G is f-choosable.*

Despite the fact that Theorem 1.18 is about vertex choosability while we focus on total choosability, Theorem 1.18 will turn out to be helpful when looking at the constraint graphs.

As a consequence, we get this classical result about choosability of even cycles.

Corollary 1.19. *Any even cycle is 2-choosable.*

We introduce some other useful results. The first one is based on Corollary 1.19.

Lemma 1.20. *Let G be the graph composed of a cycle $v_1 \cdots v_n v_1$ such that v_1, v_n share a common neighbor u, see Figure 1.26. Let L be a list assignment satisfying that for every vertex v, $|L(v)| \geqslant 2$. Then G is L-choosable if either $|L(v_1)| \geqslant 3$ or n is even and $L(v_1) \neq L(u)$.*

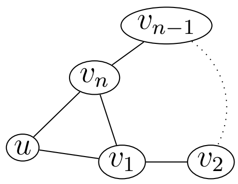


Figure 1.26 – Configuration of Lemma 1.20

Proof. Without loss of generality, we may assume that for any $v \neq v_1$, $|L(v)| = 2$, and that $|L(v_1)|$ is 2 or 3. First assume that the cycle has odd length, thus $|L(v_1)| = 3$. If $L(v_n) = L(u)$, we color v_1 with a color not in $L(u)$, then $v_2, \ldots, v_n, u$. Otherwise, we color v_n with a color not in $L(u)$, then color $v_{n-1}, \ldots, v_1, u$.

If the cycle has even length, we distinguish two cases:

- $L(v_2) = \cdots = L(v_n)$, then we color $v_2, v_4, \ldots, v_n$ with a color, $v_3, v_5 \ldots, v_{n-1}$ with another color. Denote by $\widehat{L}$ the list assignment obtained from L by removing the colors of the neighbors of each vertex. Observe that we have $|\widehat{L}(v_1)| = 1$ or 2. If $|\widehat{L}(v_1)| = 2$, we color u then v_1. Otherwise, since $\widehat{L}(v_1) \neq \widehat{L}(u)$, we can color v_1 then u.

- Otherwise, there exists i such that $L(v_i) \neq L(v_{i+1})$. Color v_{i+1} with a color not in $L(v_i)$, then color $v_{i+1}, \ldots, v_n$. With $\widehat{L}$ defined as previously, we now have $|\widehat{L}(u)| = 1$ and $|\widehat{L}(v_1)| = 1$ or 2. If $|\widehat{L}(v_1)| = 2$, we color $u, v_1, v_2, \ldots, v_i$. Otherwise, since we have $L(u) \neq L(v_1)$, we can color u with a color not in $L(v_1)$, then $v_1, v_2, \ldots, v_i$.

□

The next result is a consequence of Hall's necessary and sufficient condition for a perfect matching to exist in a bipartite graph. Finding an L-coloring of a graph G can be reduced to finding a perfect matching in the following graph. It has one vertex per color c and per vertex x of G, and an edge (c, x) when $c \in L(x)$. Since this graph is bipartite, Hall's criterion gives a condition for an L-coloring to exist.

Theorem 1.21 (Hall's marriage theorem)**.** *Let G be a clique. Then for any list assignment L, the graph G is L-choosable if and only if for all $S \subset V(G)$, $|S| \leqslant |\cup_{x \in S} L(x)|$.*

We end this subsection with a last configuration, depicted in Figure 1.27.

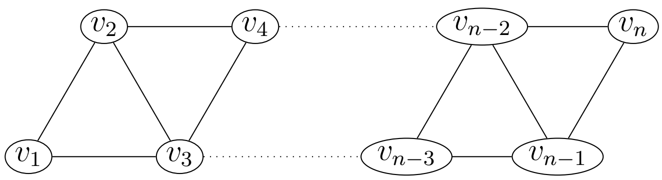


Figure 1.27 – Configuration of Lemma 1.22

Lemma 1.22. *Let $n \geqslant 4$ be an integer such that $n \not\equiv 0 \bmod 3$. Let G be the graph formed by a path $v_1 \ldots v_n$ with additional edges $v_i v_{i+2}$ for $1 \leqslant i \leqslant n-2$ (see Figure 1.27). Let L be a list assignment L such that $|L(v)| \geqslant 2$ for $v \in \{v_1, v_{n-1}, v_n\}$, and $|L(v)| \geqslant 3$ for any other v. Then G is L-choosable.*

Proof. We proceed by induction on n.

- Assume $n = 4$. If $L(v_3) = L(v_4)$, we color v_2 with a color not in $L(v_3)$, then v_1, v_3, v_4. Otherwise, we color v_3 with a color not in $L(v_4)$, then v_1, v_2, v_4.

- Assume $n = 5$. If $L(v_4) = L(v_5)$, we color v_3 with a color not in $L(v_4)$, then v_1, v_2, v_4, v_5. Otherwise, we color v_5 with a color not in $L(v_4)$ and use the case $n = 4$ to color v_1, v_2, v_3, v_4.

- Assume $n > 6$. If $L(v_{n-1}) = L(v_n)$, we color v_{n-2} with a color not in $L(v_n)$, then apply the case $n-3$ to color $v_1, \ldots, v_{n-3}$, then color v_{n-1} and v_n. Otherwise if $n \equiv 2 \mod 3$, we color v_n with a color not in $L(v_{n-1})$, then apply the use case $n-1$ to color $v_1, \ldots, v_{n-1}$. If $n \equiv 1 \mod 3$, color v_{n-1} with a color not in $L(v_n)$, then use case $n-2$ to color $v_1, \ldots, v_{n-2}$, then color v_n. □

We are now ready to prove that all the configurations C_1 to C_{22} are reducible. We devote a subsection to each configuration. We use the recoloring approach to reduce C_{18} to C_{22}. For C_1 to C_{15}, we try to give a case analysis argument and another one using the Combinatorial Nullstellensatz. The first one can be checked by hand, while the second one often requires some computations. However, in some cases, we only present one argument. This is because either the case analysis leads to an unreasonably long proof, or because the (naive) algorithm we use to find a suitable monomial in the Nullstellensatz approach does not output a result on the instance in a reasonable time.

Configuration C_1

Lemma 1.23. *The graph G does not contain C_1.*

Proof. Assume that G has an edge $e = uv$ such that $d_G(u) + d_G(v) \leqslant 10$ and $d_G(u) \leqslant 4$.

Color $G \setminus \{e\}$ by minimality and uncolor u. We may assume that $|\widehat{e}| = 1$ and $|\widehat{u}| = 3$.

We can extend the coloring to G using the following argument. We first forget u. Then uv has at most $d_G(u) + d_G(v) - 1 < 10$ constraints, so we can color it.

We can also conclude using the Nullstellensatz: note that P_G is $E - U$. Then the monomial $m = U$ satisfies:

1. $\deg(m) = 1 = \deg(P)$.

2. $\deg_E(m) = 0 < 1 = |\widehat{e}|$ and $\deg_U(m) = 1 < 3 = |\widehat{u}|$.

3. m has coefficient -1 in P.

Hence we can color G using Theorem 1.14. □

Configuration C_2

Lemma 1.24. *The graph G does not contain C_2.*

Proof. Assume that G has an even cycle $v_1 \cdots v_{2n} v_1$ such that for $1 \leqslant i \leqslant n$, $d(v_{2i-1}) \leqslant 4$ and $d(v_{2i-1}) + d(v_{2i}) \leqslant 11$.

Denote by G' the graph obtained from G by removing the edges of the cycle. Using the minimality of G, we can color G'. Remove the color of vertices with odd subscript, and forget them since they have degree at most 4. Observe that each edge of the cycle has now $d(v_{2i}) - 1 + d(v_{2i+1}) - 2 = 11 - 3 = 8$ constraints.

By Corollary 1.19, we can color the edges of the cycle and obtain a valid coloring of G.

We can also conclude using the Nullstellensatz: we have $P_G = (E_1 - E_2) \cdots (E_{2n-1} - E_{2n})(E_1 - E_{2n})$ and $m = E_1 \cdots E_{2n}$, where $e_1, \ldots, e_{2n}$ are the uncolored edges of G. We have:

1. $\deg(m) = 2n = \deg(P_G)$.
2. For $1 \leqslant i \leqslant 2n$, $\deg_{E_i}(m) = 1 < 2 = |\widehat{e_i}|$.
3. The coefficient of $m = E_1 \cdots E_{2n}$ in P_G is then -2.

Using Theorem 1.14, we can extend the coloring to G. □

Configuration C_3

Lemma 1.25. *The graph G does not contain C_3.*

Proof. We name the elements according to Figure 1.28. By minimality, we color $G' = G \setminus \{a, c, u\}$ and we remove the color of b, v, w.

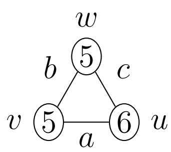


Figure 1.28 – Notation for Lemma 1.25

Observe that u has eight constraints, while a, c have seven and v, w, b have six. Thus, there are at least two colors in $\widehat{u}$, three in $\widehat{a}$ and $\widehat{c}$ and four in $\widehat{v}, \widehat{b}$ and $\widehat{w}$. By Remark 1.12, it is sufficient to treat the worst case: when $|\widehat{a}| = |\widehat{c}| = 3$, $|\widehat{v}| = |\widehat{w}| = |\widehat{b}| = 4$ and $|\widehat{u}| = 2$.

We color c with a color not in $\widehat{u}$. If afterwards, we have $|\widehat{a}| = 2$, then we color b with a color not in $\widehat{a}$ (since $|\widehat{b}|$ is now at least 3). Otherwise, we color b arbitrarily. In both cases, we obtain $|\widehat{a}| = 2$ after coloring c and b. We conclude the proof applying Lemma 1.22 on $\mathcal{T}(G)$ with the path $wvua$.

We can also conclude using the Nullstellensatz: the coefficient of $A^2B^3C^2V^2W^3$ in P_G is 1. Hence, using Theorem 1.14, we can extend the coloring to G. □

Configuration C_4

Lemma 1.26. *The graph G does not contain C_4.*

Proof. Assume that G contains a path uvw such that $d(u) = d(v) = d(w) = 5$. Note that we may assume that $uw \notin E(G)$ due to C_3. We denote by a, b the edges uv and vw.

We color by minimality the graph G' obtained by removing a and b from G. Then we uncolor u, v and w. By Remark 1.12, we may assume that $|\widehat{v}| = 4$, $|\widehat{a}| = |\widehat{b}| = 3$ and $|\widehat{u}| = |\widehat{w}| = 2$. We conclude using Lemma 1.22 on $\mathcal{T}(G)$ with the path $uavbw$.

We can also conclude using the Nullstellensatz: the coefficient of $A^2B^2V^2W$ in P_G is -1. Hence, using Theorem 1.14, we can extend the coloring to G. □

Configuration C_5

To prove that G does not contain C_5, it is sufficient to prove the three following lemmas, one for each possible minimal triangle-distance between neighbors of u satisfying the hypothesis of v_1 and v_2.

Lemma 1.27. *The graph G does not contain a 7-vertex u with two $(5,6)$-neighbors v_1, v_2 such that uv_1v_2 is a triangle.*

Proof. We use the notation depicted in Figure 1.29. By minimality, we color $G' = G \setminus \{a, b, c, d, e, f, g\}$ and uncolor u, v_1, v_2, w_1, w_2. By C_3, there is no edge w_1v_2 nor w_2v_1. Therefore, the only possible edge of G not on the drawing is w_1w_2. By Remark 1.12, we may assume that $|\widehat{d}| = |\widehat{g}| = 3$, $|\widehat{a}| = |\widehat{c}| = |\widehat{u}| = 4$, $|\widehat{e}| = |\widehat{f}| = 5$ and $|\widehat{v_1}| = |\widehat{v_2}| = |\widehat{b}| = 6$. Moreover, if $w_1w_2 \notin E(G)$, we can take $|\widehat{w_1}| = |\widehat{w_2}| = 2$ and $|\widehat{w_1}| = |\widehat{w_2}| = 3$ otherwise. We remove a color $\alpha \in \widehat{e} \setminus \widehat{a}$

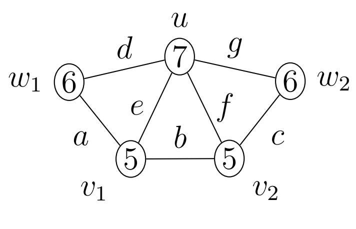


Figure 1.29 – Notation for Lemma 1.27

from $\widehat{u}$ and $\widehat{g}$. If $w_1w_2 \notin E(G)$, we apply Lemma 1.22 on $\mathcal{T}(G)$ with the path w_1dugw_2, otherwise, we color w_2 with a color not in $\widehat{g}$ and apply Lemma 1.22 on $\mathcal{T}(G)$ with the path w_1dug. Due to the choice of α, either d is colored with α and we have $|\widehat{a}| = 3$, or d is not colored with α, hence we still have $\alpha \in \widehat{e} \setminus \widehat{a}$. In the first case, we color e arbitrarily, otherwise, we color e with α. Then we color f and c and apply Lemma 1.22 on $\mathcal{T}(G)$ with the path av_1bv_2.

We can also conclude using the Nullstellensatz approach. Let

$$m = A^2B^5C^3D^2E^4F^4G^2U^3V_1^4V_2^5W_2.$$

If $w_1w_2 \notin E(G)$, m has coefficient 1 in P_G. Otherwise, mW_2 has coefficient -1 in P_G. Hence, using Theorem 1.14, we can extend the coloring to G. $\square$

Lemma 1.28. *The graph G does not contain a 7-vertex u with four neighbors v_1, v_2, v_3, v_4 such that $d(v_1) = d(v_2) = d(v_4) = 5$, $d(v_3) = 6$ and $v_iv_{i+1} \in E(G)$ for $i = 1, 2, 3$.*

Proof. We use the notation depicted in Figure 1.30. By minimality, we color $G' = G \setminus \{a, b, c, d, e, f, g\}$ and uncolor u, v_1, v_2, v_3, v_4. Due to C_4, v_1v_4 and v_2v_4 are not edges of G. Moreover, by C_3, $v_1v_3 \notin E(G)$. Therefore, all the edges between $u, v_1, \dots, v_4$ in G are drawn in the figure. By Remark 1.12, we may assume that $|\widehat{v_1}| = |\widehat{v_3}| = |\widehat{v_4}| = |\widehat{u}| = |\widehat{c}| = |\widehat{d}| = |\widehat{f}| = |\widehat{g}| = 4$, $|\widehat{a}| = |\widehat{b}| = |\widehat{e}| = 5$ and $|\widehat{v_2}| = 6$. We consider three cases:

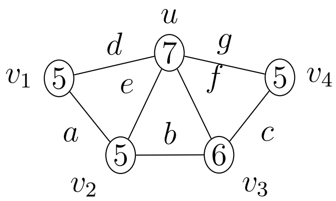


Figure 1.30 – Notation for Lemma 1.28

1. If $\widehat{v_1} \cap \widehat{v_3} \neq \varnothing$, we color v_1 and v_3 with the same color. Then we can forget v_2, a, b, e. We color f arbitrarily, then apply Lemma 1.22 on $\mathcal{T}(G)$ with the path cv_4gud.

2. If $\widehat{v_1} \cap \widehat{v_3} = \varnothing$ and $\widehat{u} \neq \widehat{v_1}$, we color u with a color not in $\widehat{v_1}$, then we can forget v_1, a, v_2, b, e, d. We color c arbitrarily and we apply Corollary 1.19 to the cycle v_3v_4gf in $\mathcal{T}(G)$.

3. Otherwise, we color u arbitrarily. Since $\widehat{u} = \widehat{v_1}$ which is disjoint from $\widehat{v_3}$, this does not affect $\widehat{v_3}$. Then, we color c with a color not in $\widehat{f}$. We again consider three subcases. After each of them, $\{v_1, v_2, v_3, a, b\}$ will remain to be colored. To do it, we apply Lemma 1.22 on $\mathcal{T}(G)$ to the path $v_1av_2bv_3$.

   (a) If $\widehat{v_4} \cap \widehat{f} \neq \varnothing$, then we color them with the same color, and color g, d, e arbitrarily.

   (b) If $\widehat{v_4} \cap \widehat{f} = \varnothing$ and $\widehat{g} \not\subset \widehat{f}$, we color g with a color not in $\widehat{f}$, then we color v_4. If $|\widehat{v_3}| = 2$, we color f with a color in $\widehat{f} \setminus \widehat{v_3}$ (recall that, up to deleting an arbitrary color from $\widehat{v_3}$, we may assume that $|\widehat{v_3}| = 2$ after having colored v_4), otherwise we color f arbitrarily. Then we color d, e.

(c) Otherwise, we can color g with a color not in $\widehat{v_4}$. Then we forget v_4, and color f, d, e arbitrarily.

We can also conclude using the Nullstellensatz: the coefficient of

$$A^4B^4C^3D^2E^4F^3G^3UV_2^5V_3^3V_4^3$$

in P_G is 1. Hence, using Theorem 1.14, we can extend the coloring to G. $\square$

Lemma 1.29. *The graph G does not contain a 7-vertex u with six neighbors $v_1, \dots, v_6$ such that $d(v_1) = d(v_2) = d(v_5) = d(v_6) = 5$, $d(v_3) = d(v_4) = 6$ and $v_iv_{i+1} \in E(G)$ for $1 \leqslant i \leqslant 6$.*

Proof. We use the notation depicted in Figure 1.31. By minimality, we color $G' = G \setminus \{a, \dots, k\}$ and uncolor u, v_1, v_2, v_5, v_6. By Remark 1.12, we may assume that $|\widehat{c}| = 2$, $|\widehat{b}| = |\widehat{d}| = 4$, $|\widehat{a}| = |\widehat{e}| = |\widehat{h}| = |\widehat{i}| = 5$, $|\widehat{u}| = |\widehat{f}| = |\widehat{k}| = 6$ and $|\widehat{g}| = |\widehat{j}| = 7$.

Note that due to C_4, there is no edge v_iv_j for $i = 1, 2$ and $j = 5, 6$. Thus, we may assume that $|\widehat{v_1}| = |\widehat{v_6}| = 4$ and $|\widehat{v_2}| = |\widehat{v_5}| = 5$. We did not succeed

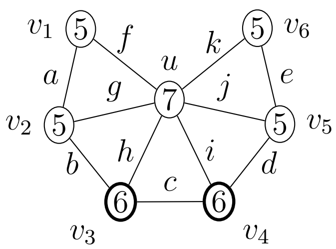


Figure 1.31 – Notation for Lemma 1.29

in finding a suitable monomial for the Nullstellensatz approach, hence we only present a case analysis proof. We color e with a color not in $\widehat{v_6}$. We forget v_6 and v_5, then color d, i and h with colors not in $\widehat{c}$ and forget c. We color j, k, u, f, g, then apply Lemma 1.22 on $\mathcal{T}(G)$ with the path v_1av_2b. $\square$

Configuration C_6

To prove that G does not contain C_6, we prove the two following lemmas.

Lemma 1.30. *The graph G does not contain a 5-vertex u adjacent to three consecutive 6-vertices v_1, v_2, v_3.*

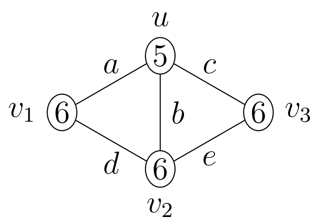


Figure 1.32 – Notation for Lemma 1.30

Proof. We use the notation depicted in Figure 1.32. We color $G \setminus \{a, \ldots, e\}$ by minimality, and then uncolor u, v_1, v_2, v_3.

By Remark 1.12, we may assume that $|\widehat{v_1}| = |\widehat{v_3}| = 3$ or 2 depending on whether $v_1v_3 \in E(G)$, $|\widehat{d}| = |\widehat{e}| = 3$, $|\widehat{v_2}| = |\widehat{a}| = |\widehat{c}| = 4$, $|\widehat{b}| = 5$ and $|\widehat{u}| = 6$. Assume that the coloring cannot be extended to G. We prove several assertions on the color lists.

1. If $v_1v_3 \notin E(G)$, then $\widehat{v_1} \cap \widehat{v_3} = \varnothing$. Otherwise, assign the same color to v_1 and v_3. We can forget u. Recall that now we may assume that $|\widehat{d}| = |\widehat{e}| = 2$ and $|\widehat{v_2}| = 3$. If $\widehat{d} = \widehat{e}$, then color v_2 and b with a color not in $\widehat{d}$, and conclude using Corollary 1.19 on the cycle $aced$ in $\mathcal{T}(G)$. Otherwise, color d with a color not in $\widehat{e}$, and then color a. To conclude, apply Lemma 1.22 on $\mathcal{T}(G)$ with the path $cbev_2$.

2. $\widehat{v_2} \cap \widehat{a} = \varnothing$ (and by symmetry $\widehat{v_2} \cap \widehat{c} = \varnothing$). Otherwise, put the same color on v_2 and a and forget u, b, c. If $v_1v_3 \notin E(G)$, Item 1 ensures that the common color is not in both $\widehat{v_1}$ and $\widehat{v_3}$. If it is not in $\widehat{v_3}$, color v_1, d, e, v_3, otherwise, v_3, e, d, v_1. If $v_1v_3 \in E(G)$, we apply Corollary 1.19 on the cycle v_1v_3ed in $\mathcal{T}(G)$.

3. $\widehat{v_1} \subset \widehat{u}$ (and by symmetry $\widehat{v_3} \subset \widehat{u}$). Otherwise, color v_1 with a color not in $\widehat{u}$ and forget u. Item 2 ensures that this color is not in both $\widehat{v_2}$ and $\widehat{a}$. If it is not in $\widehat{a}$, we forget a, c, b and apply Lemma 1.22 on $\mathcal{T}(G)$ with the path dv_2ev_3. Otherwise, we color e with a color not in $\widehat{v_3}$. Again, by Item 2, this color is not in both $\widehat{v_2}$ and $\widehat{c}$. If it is not in $\widehat{c}$, color d, v_2, a, b, v_3, c, otherwise, observe that $|\widehat{v_2}| = 4$ and color d, c, a, b, v_3, v_2.

4. $\widehat{a} \subset \widehat{u}$ (and by symmetry $\widehat{c} \subset \widehat{u}$). Otherwise, color a with a color not in u and forget u. Note that this does not affect $\widehat{v_1}$. Color d, v_1 such that $\widehat{e} \neq \widehat{v_3}$, then e with a color not in $\widehat{v_3}$. This color is not in both $\widehat{v_2}$ and $\widehat{c}$. If it is not in $\widehat{c}$, then color v_2, v_3, b, c, otherwise apply Corollary 1.19 on the cycle v_2v_3cb in $\mathcal{T}(G)$.

5. $\widehat{v_2} \subset \widehat{u}$. Otherwise we can color v_2 with a color not in $\widehat{u}$ (and hence not in $\widehat{v_1}$ nor in $\widehat{v_3}$ by Items 1 and 3) and then forget u. We consider two cases:

(a) Assume that $v_1v_3 \notin E(G)$. If $\widehat{v_1} \neq \widehat{d}$, then color d with a color not in $\widehat{v_1}$, then color v_3, c, b, a, v_1. Otherwise, we also have $\widehat{v_3} = \widehat{e}$ by symmetry, hence $\widehat{d} \cap \widehat{e} = \widehat{v_1} \cap \widehat{v_3} = \varnothing$ by Item 1. We may thus color d with a color not in $\widehat{e}$, color v_1 and a arbitrarily and apply Lemma 1.22 on $\mathcal{T}(G)$ with the path $bcev_3$.

(b) Assume that $v_1v_3 \in E(G)$. We color d, e, b arbitrarily and apply Corollary 1.19 on the cycle v_1v_3ca in $\mathcal{T}(G)$.

We proved that $\widehat{a}$ and $\widehat{v_2}$ are disjoint and contained in $\widehat{u}$. Since $|\widehat{a}| = |\widehat{v_2}| = 4$ and $|\widehat{u}| = 6$, this is a contradiction. Therefore, the coloring extends to G.

We can also conclude using the Nullstellensatz. Let $m = A^2B^4C^3D^2E^2U^5V_1V_2^3V_3$. If $v_1v_3 \notin E(G)$, the coefficient of m in P_G is 1, otherwise, mV_3 has coefficient -2 in P_G. Hence, using Theorem 1.14, we can extend the coloring to G. □

Lemma 1.31. *The graph G does not contain a triangulated 5-vertex u with neighbors $v_1, \ldots, v_5$ satisfying $d(v_1) = d(v_3) = d(v_5) = 6$ and $d(v_2) = d(v_4) = 7$.*

Proof. We use the notation depicted in Figure 1.33. We color $G' = G \setminus \{a, \ldots, j\}$ by minimality, and then uncolor $u, v_1, \ldots, v_5$. By Remark 1.12,

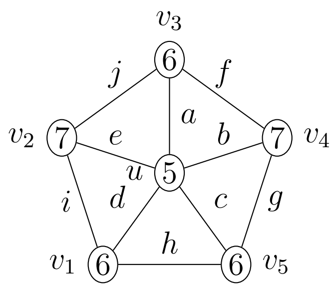


Figure 1.33 – Notation for Lemma 1.31

we may assume that $|\widehat{v_2}| = |\widehat{v_4}| = 2$, $|\widehat{f}| = |\widehat{g}| = |\widehat{i}| = |\widehat{j}| = 3$, $|\widehat{v_1}| = |\widehat{v_3}| = |\widehat{v_5}| = |\widehat{h}| = 4$, $|\widehat{b}| = |\widehat{e}| = 6$, $|\widehat{a}| = |\widehat{c}| = |\widehat{d}| = 7$ and $|\widehat{u}| = 10$. Moreover, for $1 \leqslant i \leqslant 5$, $|\widehat{v_i}|$ may differ depending on the presence of edges between the v_i's that are not on the figure, but we may assume that $|\widehat{v_2}|, |\widehat{v_4}|$ are at least 2 and $|\widehat{v_1}|, |\widehat{v_3}|, |\widehat{v_5}|$ are at least 4.

We do not have a case analysis proof in this case, only the Combinatorial Nullstellensatz approach. Let $m_0 = A^6B^4C^6D^3E^5F^2G^2H^3I^2J^2U^9V_1^3V_2V_3^3V_4V_5^3$. We distinguish several cases depending on the edges between the v_i's that may not be in the figure. In each of them we define a monomial m and give its coefficient in P_G, so we can apply Theorem 1.14 to m in order to extend the coloring to G.

First note that the subgraph H of G induced by the v_i's is an outerplanar graph on five vertices. It has thus at most seven edges, so there are at most two additional edges on the figure. Moreover, these edges must be non-crossing diagonals of the pentagon formed by the v_i's, otherwise, there is a K_4 minor in H. Due to the symmetry, we may only consider the following cases:

1. There is no additional edge. Then we take $m = m_0$ and m has coefficient -1 in P_G.
2. The only additional edge is v_2v_4. Then we take $m = m_0V_2$ and m has coefficient -2 in P_G.
3. The only additional edge is v_1v_3. Then we take $m = m_0V_1$ and m has coefficient 1 in P_G.
4. The only additional edge is v_2v_5. Then we take $m = m_0V_2$ and m has coefficient -1 in P_G.
5. The two additional edges are v_2v_4, v_2v_5. Then we take $m = m_0V_2^2$ and m has coefficient -2 in P_G.
6. The two additional edges are v_1v_3, v_3v_5. Then we take $m = m_0V_3^2$ and m has coefficient 1 in P_G.
7. The two additional edges are v_1v_3, v_1v_4. Then we take $m = m_0V_3V_4$ and m has coefficient 1 in P_G.

In each case, we have $\deg m = \deg P_G$, m has a non-zero coefficient in P_G, and for any element x of the configuration, $\deg_X m < |\widehat{x}|$. Therefore, we can extend the coloring to G. $\square$

Configuration C_7

Lemma 1.32. *The graph G does not contain a 7-vertex u with four neighbors $v_1, \ldots, v_4$ satisfying $d(v_1) = d(v_2) = 5$, $d(v_3) = 6$, $d(v_4) = 4$, and $v_1v_2, v_2v_3 \in E(G)$.*

Proof. We consider the notation depicted in Figure 1.34. By minimality, we color $G' = G \setminus \{a, \ldots, f\}$ and uncolor u, v_1, v_2, v_3, v_4.

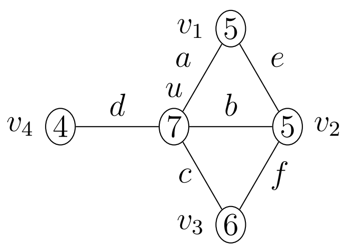


Figure 1.34 – Notation for Lemma 1.32

By Remark 1.12, we may assume that $|\widehat{c}| = 3$, $|\widehat{u}| = |\widehat{v_4}| = |\widehat{a}| = |\widehat{d}| = |\widehat{f}| = 4$, $|\widehat{b}| = |\widehat{e}| = 5$ and $|\widehat{v_2}| = 6$. Moreover, none of v_1, v_2, v_3 are adjacent to v_4 because of C_1, and $v_1v_3 \notin E(G)$ because of C_3. Hence, we have $|\widehat{v_1}| = 4$, $|\widehat{v_3}| = 2$. We forget v_4, then color c and u with colors not in $\widehat{v_3}$, and color a, d, b. We finally apply Lemma 1.22 on $\mathcal{T}(G)$ with the path $v_1ev_2fv_3$.

We can also conclude using the Nullstellensatz. Let

$$m = A^3B^4C^2D^3E^4F^3V_1^2V_2^5V_3.$$

The coefficient of m in P_G is -1. Hence, using Theorem 1.14, we can extend the coloring to G. □

Configuration C_8

According to the definition of a S_3-neighbor, if G contains C_8, v_1 is triangulated and we are in one of the following cases:

- C_{8a}: u and v_1 have a common neighbor w of degree six.
- C_{8b}: v_1 has two neighbors w_1, w_2 of degree six such that uv_1w_1 and uv_1w_2 are not triangular faces. Moreover, due to C_{8a}, we know that $w_1w_2, v_2w_2 \in E(G)$ and $uw_1, uw_2 \notin E(G)$.
- C_{8c}: v_1 has a neighbor w of degree five such that uv_1w is not a triangular face.

We dedicate a lemma to each of these configurations.

Lemma 1.33. *The graph G does not contain C_{8a}.*

Proof. We use the notation depicted in Figure 1.35. By minimality, we color $G \setminus \{a, \dots, j\}$ and uncolor $u, v_1, \dots, v_4, w$.

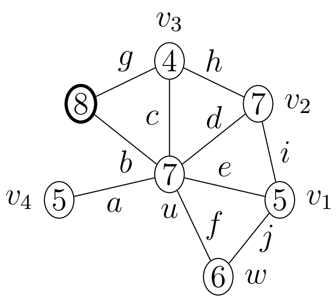


Figure 1.35 – Notation for Lemma 1.33

By Remark 1.12, we may assume that $|\widehat{b}| = |\widehat{g}| = 2$, $|\widehat{i}| = |\widehat{j}| = 4$, $|\widehat{d}| = |\widehat{f}| = |\widehat{a}| = |\widehat{h}| = 5$, $|\widehat{u}| = |\widehat{v_3}| = |\widehat{e}| = 7$ and $|\widehat{c}| = 8$. Moreover, the sizes $|\widehat{v_1}|, |\widehat{v_2}|, |\widehat{w}|, |\widehat{v_4}|$ depend on the presence of edges between the vertices v_1, v_2, v_4, w. Note that v_3 is not adjacent to any of v_1, v_4, w by C_1. Moreover, $|N(v_4) \cap \{v_1, w\}| \leqslant 1$ by C_3. We may thus assume that $|\widehat{v_1}| = 6 + |N(v_1) \cap \{v_4\}|$, $|\widehat{v_2}| = 2 + |N(v_2) \cap \{v_4, w\}|$, $|\widehat{v_4}| = 2 + |N(v_4) \cap \{v_1, v_2, w\}|$ and $|\widehat{w}| = 2 + |N(w) \cap \{v_2, v_4\}|$. We forget v_3.

The following procedure shows how to extend the coloring to G, even if some edges among v_2w, v_2v_4, v_1v_4 and wv_4 are present in G. Note that v_2w and v_2v_4 do not affect the procedure. We separate the first step in three cases:

1. If $\widehat{g} \not\subset \widehat{h}$, we color g with a color not in $\widehat{h}$, then we forget h, c and color b, v_2, d, i arbitrarily.

2. If $\widehat{g} \subset \widehat{h}$ and $\widehat{b} = \widehat{g}$, then we color b and h with the same color, then color g, v_2, d, i arbitrarily and forget c.

3. Otherwise, we color b with a color not in g. Then, we color i with a color not in $\widehat{h} \setminus \widehat{g}$, then v_2, d arbitrarily,. The lists $\widehat{h}, \widehat{g}$ are thus different lists of size 2, so we can color h with a color not in $\widehat{g}$, and forget g and c.

In each case, we are left with the same set of uncolored elements, namely $\{u, v_1, v_4, w, a, e, f, j\}$. Moreover, we have $|\widehat{u}| = |\widehat{e}| = 4$, $|\widehat{a}| = |\widehat{f}| = |\widehat{j}| = 3$, $|\widehat{v_4}| = 2 + |N(v_4) \cap \{v_1, w\}|$ and $|\widehat{v_1}| = 4 + |N(v_1) \cap \{v_4\}|$.

We may assume that $\widehat{e} \cap \widehat{w} = \varnothing$. Otherwise, we color w and e with the same color. Then, we can forget v_1, j and conclude using Lemma 1.22 on $\mathcal{T}(G)$ with the path $fuav_4$.

We now separate three cases depending on the presence of the edges v_1v_4 and wv_4:

1. Assume there is an edge v_1v_4 in G. Then we have $|\widehat{v_1}| = 5$ and $|\widehat{v_4}| \geqslant 3$. We consider two cases:

   (a) If $\widehat{j} \not\subset \widehat{e}$, we color j with a color not in $\widehat{e}$, then color w and f. If $\widehat{u} = \widehat{a}$, we color v_4 and e with a color not in $\widehat{u}$, then color v_1, u and a. If $\widehat{u} \neq \widehat{a}$, we color a with a color not in $\widehat{u}$, and apply Lemma 1.22 on $\mathcal{T}(G)$ with the path ev_1uv_4.

   (b) Otherwise, we color e with a color not in $\widehat{j}$. As we have $\widehat{j} \cap \widehat{w} = \varnothing$, we can forget j, v_1, v_4, and conclude applying Lemma 1.22 on $\mathcal{T}(G)$ with the path $aufw$.

2. Assume there is an edge wv_4 but no edge v_1v_4. Thus we have $|\widehat{v_1}| = 4$, $|\widehat{v_4}| = 3$ and $|\widehat{w}| = 3$. Let $\alpha \in \widehat{u} \setminus \widehat{v_4}$. We separate two cases depending on whether $\widehat{e}$ or $\widehat{w}$ does not contain α. Recall that α cannot be in both $\widehat{e}$ and $\widehat{w}$. In any case, we first color u with α and forget v_4.

   (a) Assume that $\alpha \notin \widehat{w}$. If $\widehat{f} = \widehat{a}$, we color e with a color not in $\widehat{f}$, forget a and apply Lemma 1.22 on $\mathcal{T}(G)$ with the path $fwjv_1$. Otherwise, we color f with a color not in $\widehat{a}$, forget a, and apply Lemma 1.22 on $\mathcal{T}(G)$ with the path ev_1jw.

(b) Assume that $\alpha \notin \widehat{e}$. We color a arbitrarily. Let $\beta \in \widehat{f}$. If $\beta \notin \widehat{e}$, then color f with β and then w, j, v_1, e. Otherwise, recall that $\widehat{w} \cap \widehat{e} = \varnothing$, so $\beta \notin \widehat{w}$, and we color f with β and apply Lemma 1.22 on $\mathcal{T}(G)$ with the path ev_1jw.

3. Assume there is no edge v_1v_4 nor wv_4. We thus have $|\widehat{v_4}| = 2$, so we can color a with a color not in $\widehat{v_4}$, then forget v_4. We consider three cases:

   (a) If $\widehat{f} = \widehat{w}$, we have $\widehat{f} \cap \widehat{e} = \varnothing$. We color j and u with a color not in $\widehat{f}$, then color e. Recall that this does not affect $\widehat{f}$. Then we color v_1, w arbitrarily and f.

   (b) If $\widehat{f} \neq \widehat{w}$ and $\widehat{j} \cap \widehat{u} \neq \varnothing$, we color j and u with the same color. Note that $\widehat{f} \neq \widehat{w}$, hence we can color f and w, then e and v_1.

   (c) Otherwise, $\widehat{f} \neq \widehat{w}$ and $\widehat{j} \cap \widehat{u} = \varnothing$. We color f with a color $\alpha \notin \widehat{w}$. Free to exchange u and j in $\mathcal{T}(G)$, we may assume that $\alpha \notin \widehat{u}$. Since $\widehat{e} \cap \widehat{w} = \varnothing$, we can color either e or w with a color not in $\widehat{u}$. We color j, either w or e, v_1 and u.

We can also conclude using the Nullstellensatz. Let

$$m_0 = A^4BC^6D^4E^6F^4GH^4I^3J^3V_1^5V_4W$$

We distinguish several cases depending on which edges are present between v_1, v_2, v_4 and w. In each case, we define a monomial with non-zero coefficient in P_G, so that Theorem 1.14 ensures that the coloring extends to G. First observe that since G is planar, only one of the edges v_1v_4 and v_2w is present in G.

1. If no additional edge is present in G, then we take $m = m_0CU^2V_2$ which has coefficient 3 in P_G.

2. If the only additional edge is v_2v_4, then we take $m = m_0CUV_2^2V_4$ which has coefficient 2 in P_G.

3. If the only additional edge is v_4w, then we take $m = m_0CUV_2V_4W$ which has coefficient -1 in P_G.

4. If the only additional edge is v_2w, then we take $m = m_0CUV_2^2W$ which has coefficient 1 in P_G.

5. If the only additional edge is v_1v_4, then we take $m = m_0CU^3V_4$ which has coefficient -2 in P_G.

6. If the only additional edges are v_2v_4 and v_4w, then we take $m = m_0CV_2^2V_4^2W$ which has coefficient 3 in P_G.

7. If the only additional edges are v_1v_4 and v_2v_4, then we take $m = m_0CV_1V_2^2V_4^2$ which has coefficient 3 in P_G.

8. If the only additional edges are v_1v_4 and v_4w, then we take $m = m_0CV_1V_2V_4^2W$ which has coefficient -3 in P_G.

9. If the only additional edges are v_2v_4 and v_2w, then we take $m = m_0CV_2^3V_4W$ which has coefficient -3 in P_G.

10. If the only additional edges are v_2w and v_4w, then we take $m = m_0UV_2^2V_4W^2$ which has coefficient -1 in P_G.

11. If the only additional edges are v_1v_4, v_2v_4 and v_4w, then we take $m = m_0CV_1V_2V_4^3W$ which has coefficient 3 in P_G.

12. If the only additional edges are v_2v_4, v_2w and v_4w, then we take $m = m_0CV_2^2V_4^2W^2$ which has coefficient -1 in P_G. □

Lemma 1.34. *The graph G does not contain C_{8b}.*

Proof. We consider the notation of Figure 1.36. By minimality, we color $G \setminus \{a, \ldots, h\}$ and uncolor v_1, v_2, v_3, w_1, w_2. By Remark 1.12, we may assume that

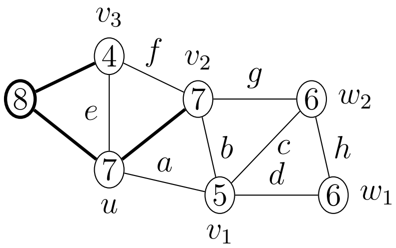


Figure 1.36 – Notation for Lemma 1.34

$|\widehat{e}| = 2$, $|\widehat{a}| = |\widehat{g}| = |\widehat{h}| = 3$, $|\widehat{w_2}| = |\widehat{f}| = 4$, $|\widehat{v_3}| = |\widehat{b}| = |\widehat{d}| = 5$, $|\widehat{c}| = 6$, $|\widehat{v_1}| = 7$. Moreover, $|\widehat{v_2}|, |\widehat{w_1}|$ are 2 or 3 depending on the presence of the edge v_2w_1 in G. We forget v_3.

We color a with a color not in $\widehat{e}$, then forget e, f. The resulting configuration is now the same as in Lemma 1.30. □

Lemma 1.35. *G does not contain C_{8c}.*

Proof. We use the notation depicted in Figure 1.37. Note that we may assume that, if $w \neq v_4$, $uw \notin E(G)$ due to Lemma 1.33. By minimality, we color $G' = G \setminus \{a, \ldots, g\}$ and uncolor u, v_1, v_2, v_3, v_4, w. By Remark 1.12, we may assume that $|\widehat{c}| = 3$, $|\widehat{u}| = |\widehat{e}| = |\widehat{f}| = 4$, $|\widehat{b}| = |\widehat{d}| = 5$ and $|\widehat{v_1}| = |\widehat{v_3}| = 6$.

Moreover, if $w = v_4$, we have $|\widehat{a}| = 4$ and $|\widehat{g}| = 5$, otherwise, $|\widehat{a}| = 3$ and $|\widehat{g}| = 4$. We may also assume that $|\widehat{v_2}|, |\widehat{v_4}|, |\widehat{w}|$ are $2, 3$ or 4 depending on whether $v_2v_4, v_2w \in E(G)$. We forget v_3.

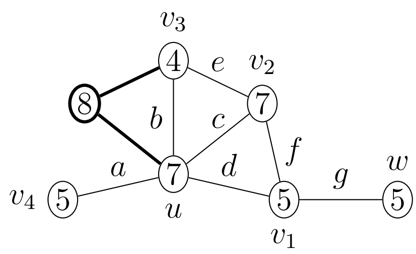


Figure 1.37 – Notation for Lemma 1.35

If $w = v_4$, we color v_2, c and u arbitrarily. Then, we color v_4 with a color not in $\widehat{a}$. We then color a such that $\widehat{g} \neq \widehat{v_1}$ if they have both size 3. Then, we color the even cycle induced by $\{b, e, f, d\}$. We then obtain that $|\widehat{g}| = |\widehat{v_1}| = 1$, but $\widehat{g} \neq \widehat{v_1}$, hence we can color them.

Assume now that $w \neq v_4$. Take $\alpha \in \widehat{d} \setminus \widehat{f}$, and remove it from $\widehat{u}$ and $\widehat{a}$. If $v_2v_4 \notin E(G)$, apply Lemma 1.22 on $\mathcal{T}(G)$ with the path v_2cuav_4, otherwise, color v_4 with a color not in $\widehat{a}$ and apply Lemma 1.22 on $\mathcal{T}(G)$ with the path v_2cua. Due to the choice of α, we now have $\widehat{d} \neq \widehat{f}$ if $|\widehat{d}| = |\widehat{f}| = 2$. We have the following:

1. $|\widehat{d} \cup \widehat{f}| > 3$, otherwise we color $\widehat{v_1}$ with a color not in $\widehat{d}$ nor in $\widehat{f}$, then color w and apply Lemma 1.20 on $\mathcal{T}(G)$ with the cycle $dbef$ and the element g.

2. $\widehat{w} \subset \widehat{v_1}$, otherwise we color w with a color not in $\widehat{v_1}$, then forget v_1, g, and apply Corollary 1.19 to color the cycle $bdfe$ in $\mathcal{T}(G)$.

3. $\widehat{f} \cap \widehat{w} = \varnothing$, otherwise we can color f, w with the same color, and then color e, b, d, v_1, g arbitrarily. Therefore, we may assume that $\widehat{w}$ is disjoint from $\widehat{f}$ and (by symmetry) from $\widehat{d}$.

Therefore, we can color either d or f with a color not in $\widehat{v_1}$. We then forget v_1, g, w and color b, e, f (or e, b, d).

We can also conclude using the Nullstellensatz. Take

$$m_0 = A^2B^4C^2D^4E^3F^3G^3UV_1^4V_2V_4$$

We distinguish several cases depending on whether $w = v_4$, and on the presence of edges between uncolored vertices that are not drawn. Here the only such edges are v_2w and v_2v_4. In each case, we define a monomial with non-zero coefficient in P_G, so that Theorem 1.14 ensures that the coloring extends to G.

1. If $w = v_4$ and $v_2w \notin E(G)$, then we take $m = A^3B^4C^2D^4E^3F^3G^4V_1^4V_2W^3$ which has coefficient -1 in P_G.

2. If $w = v_4$ and $v_2w \in E(G)$, then we take $m = A^3B^4C^2D^4E^3F^3G^4V_1^5V_2W^3$ which has coefficient 2 in P_G.

3. If $w \neq v_4$ and there is no additional edge between v_2, v_4 and w, then we take $m = m_0 U V_1$, which has coefficient -2 in P_G.

4. If $w \neq v_4$ and the only additional edge is $v_2 w$, then we define $m = m_0 V_2 W^2$, which has coefficient -1 in P_G.

5. If $w \neq v_4$ and the only additional edge is $v_2 v_4$, then we define $m = m_0 V_2 V_4 W$, which has coefficient -2 in P_G.

6. If $w \neq v_4$ and the only additional edges are $v_2 v_4$ and $v_2 w$, then we define $m = m_0 U V_2 W^2$, which has coefficient -5 in P_G. $\square$

Configuration C_9

Note that G does not contain C_8, therefore, if G contains C_9, we are in one of the following cases:

- C_{9a}: the common neighbor of v_1, u and v_2 has degree 7 and v_1 is an S_5-neighbor of u.

- C_{9b}: the common neighbor of v_1, u and v_2 has degree 8 and v_1 is a $(6,8)$-neighbor of u.

- C_{9c}: the common neighbor of v_1, u and v_2 has degree 8 and v_1 has two neighbors w_1, w_2 of degree 6 such that uv_1w_1 and uv_1w_2 are not triangular faces

- C_{9d}: the common neighbor of v_1, u and v_2 has degree 8 and v_1 has a neighbor w of degree 5 such that uv_1w is not a triangular face.

We dedicate a lemma to each of these configurations.

Lemma 1.36. *The graph G does not contain C_{9a}.*

Proof. We use the notation depicted in Figure 1.38. By minimality, we color $G \backslash \{a, \ldots, m\}$ and uncolor $u, v_1, v_2, v_3, w_1, w_2$. By Remark 1.12, we may assume that $|\widehat{g}| = |\widehat{h}| = 2$, $|\widehat{a}| = |\widehat{\ell}| = |\widehat{m}| = |\widehat{n}| = 3$, $|\widehat{i}| = 5$, $|\widehat{c}| = |\widehat{e}| = |\widehat{j}| = |\widehat{k}| = 6$, $|\widehat{f}| = |\widehat{v_2}| = 7$, $|\widehat{u}| = |\widehat{v_1}| = 8$, $|\widehat{b}| = 9$ and $|\widehat{d}| = 10$.

Moreover, note that the only edges of G between uncolored vertices that may not be present on the figure are $w_1 w_2$ and $w_2 v_3$. Depending on the presence of these edges, $|\widehat{w_2}|$ is $2, 3$ or 4, $|\widehat{w_1}|$ is 2 or 3 and $|\widehat{v_3}|$ is 4 or 5.

We first forget v_2, and color j with a color not in $\widehat{i}$. Then we color m, n, w_2, and forget i, b, h.

To color the remaining graph G', we do not have a case analysis proof, we use the Combinatorial Nullstellensatz. The coefficient of

$$U^6 V_1^3 V_3^3 W_1 A^2 C^3 D^6 E^3 F^6 G K^2 L^2$$

in P_G is -2. Therefore, by Theorem 1.14, we can extend the coloring to G. $\square$

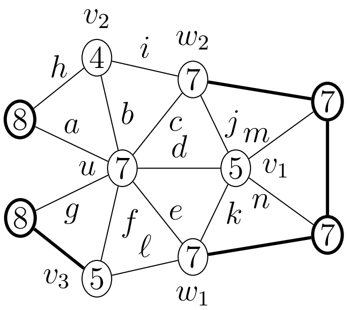


Figure 1.38 – Notation for Lemma 1.36

Lemma 1.37. *The graph G does not contain C_{9b}.*

Proof. We use the notation depicted in Figure 1.39. By minimality, we color $G' = G \setminus \{a, \dots, \ell\}$ and uncolor u, v_1, v_2, v_3, w. By Remark 1.12, we may

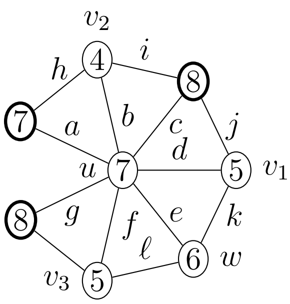


Figure 1.39 – Notation for Lemma 1.37

assume that $|\widehat{g}| = |\widehat{j}| = 2$, $|\widehat{h}| = |\widehat{i}| = 3$, $|\widehat{w}| = |\widehat{a}| = |\widehat{c}| = |\widehat{\ell}| = 4$, $|\widehat{k}| = 5$, $|\widehat{v_2}| = 6$, $|\widehat{u}| = |\widehat{e}| = |\widehat{f}| = 7$, $|\widehat{d}| = 8$ and $|\widehat{b}| = 9$.

Note that, due to C_3, $v_1v_3 \notin E(G)$. Moreover, due to C_1, v_2 is not adjacent to v_1, v_3, w. Since the graph G is simple, all the edges of G between uncolored vertices are drawn in the figure. We may thus assume that $|\widehat{v_1}| = 5$ and $|\widehat{v_3}| = 4$.

We forget v_2, then we color i and c with a color not in $\widehat{j}$. We then color g, then e with a color not in $\widehat{w}$, then a, h, and finally v_3 with a color not in $\widehat{\ell}$. We separate two cases:

1. If $\widehat{u} = \widehat{f}$, we color b and d with some colors not in $\widehat{u}$. Then we color j arbitrarily, and color v_1 with a color not in $\widehat{u}$, then k. We color w with a color α. If $\alpha \in \widehat{f}$, then we can also color f with α, then color u and ℓ. Otherwise, we color ℓ, f, u.

2. If $\widehat{u} \neq \widehat{f}$, we color f with a color not in $\widehat{u}$. We again separate two cases:

   (a) If $\widehat{b} = \widehat{u}$, we color d with a color not in $\widehat{b}$, then j. We forget b and apply Lemma 1.22 on $\mathcal{T}(G)$ with the path $uv_1wk\ell$.

(b) If $\widehat{b} \neq \widehat{u}$, we can color d with a color not in $\widehat{j}$ and still color b and u afterwards. We then apply Lemma 1.22 on $\mathcal{T}(G)$ with the path $jv_1kw\ell$.

We can also conclude using the Nullstellensatz: the coefficient of

$$A^3B^8C^3D^7E^6F^6GH^2I^2JK^4L^3V_1^3V_3^3W^3$$

in P_G is -3. Hence, using Theorem 1.14, we can extend the coloring to G. $\square$

Lemma 1.38. *The graph G does not contain C_{9c}.*

Proof. We use the notation depicted in Figure 1.40. By minimality, we color $G\setminus\{a,\dots,q\}$ and uncolor $u, v_1, v_2, v_3, w_1, w_2$. By Remark 1.12, we may assume

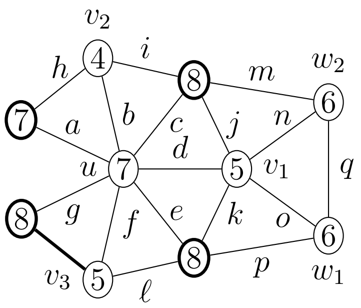


Figure 1.40 – Notation for Lemma 1.38

that $|\widehat{m}| = |\widehat{p}| = |\widehat{\ell}| = 2$, $|\widehat{h}| = 3$, $|\widehat{a}| = |\widehat{i}| = |\widehat{q}| = 4$, $|\widehat{c}| = |\widehat{e}| = |\widehat{j}| = |\widehat{k}| = 5$, $|\widehat{v_2}| = 6$, $|\widehat{f}| = |\widehat{n}| = |\widehat{o}| = 7$, $|\widehat{v_1}| = 8$, $|\widehat{b}| = 9$ and $|\widehat{d}| = 10$.

Moreover, $|\widehat{g}|$, $|\widehat{u}|$, $|\widehat{v_3}|$, $|\widehat{w_1}|$ and $|\widehat{w_2}|$ may differ depending on the presence of edges between these vertices that are not on the figure, and whether g is incident to w_1 or w_2. However, we still have at least 2 colors in $\widehat{g}$, 3 in $\widehat{w_1}, \widehat{v_3}, \widehat{w_2}$ and 6 in $\widehat{u}$.

We did not succeed in finding a suitable monomial for the Nullstellensatz approach, hence we only present a case analysis proof. We forget v_2, then color a with a color not in $\widehat{h}$ and g arbitrarily. Then we forget h, b, i. We color ℓ such that $\widehat{u}$ and $\widehat{f}$ are not the same set of size 4 afterwards, then p, e. We color m such that $\widehat{w_1}, \widehat{w_2}$ are not the same set of size 2, then c, k, j. We then separate three cases:

1. Assume that g is not incident to w_1, w_2 and that $\widehat{f} = \widehat{v_3}$. We color u with a color not in $\widehat{f}$ and forget v_3, f. We have three cases:

   (a) If $\widehat{w_2} = \widehat{q}$ (or $\widehat{w_1} = \widehat{q}$ by symmetry), we color w_1 with a color not in $\widehat{q}$, then o with a color not in $\widehat{q}$, and we apply Lemma 1.22 on $\mathcal{T}(G)$ with the path dv_1nw_2q.

(b) If $\widehat{w_2} \neq \widehat{q}$ and moreover, $\widehat{q} \not\subset \widehat{w_1} \cup \widehat{w_2}$, we color q with a color not in this union. We color w_2 with a color not in $\widehat{w_1}$, color n, and apply Lemma 1.22 on $\mathcal{T}(G)$ with the path dv_1ow_1.

(c) Otherwise, we have $\widehat{w_2} = \{\alpha, \beta\}$, $\widehat{w_1} = \{\gamma, \delta\}$ and $\widehat{q} = \{\alpha, \gamma\}$ (with possibly $\beta = \delta$). Therefore, there are two possible colorings for $\{w_1, w_2, q\}$ hence at least one of them ensures that $\widehat{v_1} \neq \widehat{d}$. We then apply Theorem 1.21 on $\{v_1, d, n, o\}$.

2. Assume that g is not incident to w_1, w_2 and that $\widehat{f} \neq \widehat{v_3}$.

   Since $|\widehat{w_1}| \neq |\widehat{w_2}|$, $\{w_1, w_2, q\}$ is colorable. Moreover, there are at least two different colorings for this set. Therefore, we may always color w_1, w_2, q such that afterwards we have $\widehat{n} \neq \widehat{o}$ if they are lists of size two.

   If $|\widehat{n} \cup \widehat{o}| = 3$, we can color v_1 with a color not in $\widehat{n} \cup \widehat{o}$, then color u. Since $\widehat{f} \neq \widehat{v_3}$, we can color f, v_3, then d. Finally, we can color n and o since $\widehat{n} \neq \widehat{o}$.

   Otherwise, we have $|\widehat{n} \cup \widehat{o}| > 3$. We may thus color v_3, f, u (since $\widehat{v_3} \neq \widehat{f}$) and apply Theorem 1.21 on $\{v_1, d, n, o\}$.

3. Assume that g is incident to w_1 or w_2. Free to exchange w_1 and w_2, we may assume that $g = uw_1$. The situation is depicted on Figure 1.41. We may thus assume that $|\widehat{f}| = |\widehat{q}| = |\widehat{w_2}| = 2$, $|\widehat{u}| = |\widehat{v_3}| = 3$, $|\widehat{d}| = |\widehat{n}| = |\widehat{o}| = |\widehat{w_1}| = 4$ and $|\widehat{v_1}| = 6$.

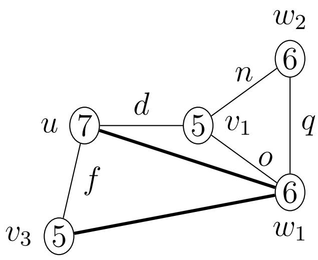


Figure 1.41 – Remaining elements for Lemma 1.38

   If $\widehat{n} = \widehat{o}$, we color w_2 arbitrarily, otherwise, there exists $\alpha \in \widehat{n} \neq \widehat{o}$ and we color w_2 not with α. We then color q and w_1 such that $\widehat{f} \neq \widehat{v_3}$ afterwards.

   Due to the choice for the color of w_2, we now have $\widehat{n} \neq \widehat{o}$ if they have size two. We have $\widehat{n} \neq \widehat{o}$ and $\widehat{f} \neq \widehat{v_3}$, hence may now apply the same procedure as in the previous item. □

Lemma 1.39. *The graph G does not contain C_{9d}.*

Proof. We use the notation depicted in Figure 1.42. By definition, there is an edge m between w and either w_1 or w_2. We separate three cases depending

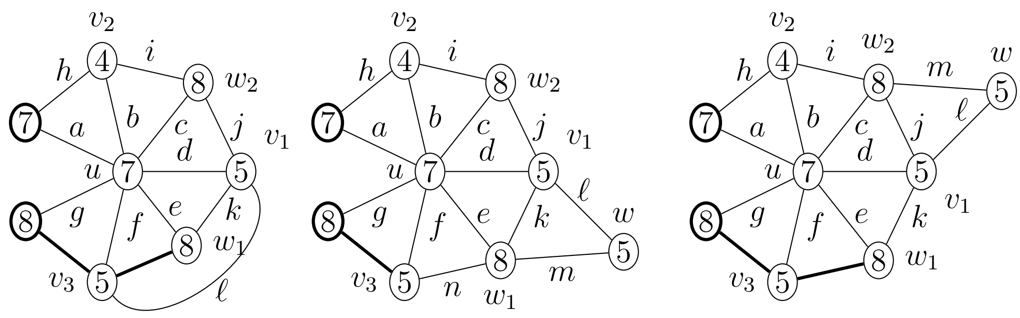


Figure 1.42 – Notation for Lemma 1.39

on whether $w = v_3$, and whether $m = ww_1$ or $m = ww_2$. In each case, we color by minimality the graph G' obtained from G by removing $a, \dots, \ell$ and the labeled edges incident to m if $w \neq v_3$. We then uncolor u, v_1, v_2, v_3 and the endpoints of m if $w \neq v_3$.

Observe that if $w \neq v_3$, there is no edge v_3w nor v_1v_3 in $E(G)$ due to C_4. Moreover, since v_3 is a weak neighbor of u, we cannot have $g = uw$ either (otherwise, v_1wv_3 creates C_4).

By Remark 1.12, we may assume that:

1. If $w = v_3$, $|\widehat{g}| = |\widehat{k}| = 2$, $|\widehat{e}| = |\widehat{h}| = |\widehat{i}| = |\widehat{j}| = 3$, $|\widehat{v_3}| = |\widehat{a}| = |\widehat{c}| = 4$, $|\widehat{u}| = |\widehat{v_1}| = |\widehat{v_2}| = |\widehat{\ell}| = 6$, $|\widehat{f}| = 7$ and $|\widehat{b}| = |\widehat{d}| = 9$.

2. If $m = ww_1$, $|\widehat{w_1}| = |\widehat{g}| = 2$, $|\widehat{h}| = |\widehat{i}| = |\widehat{j}| = |\widehat{m}| = |\widehat{n}| = 3$, $|\widehat{v_3}| = |\widehat{w}| = |\widehat{a}| = |\widehat{c}| = 4$, $|\widehat{k}| = 5$, $|\widehat{v_2}| = |\widehat{e}| = |\widehat{\ell}| = 6$, $|\widehat{u}| = |\widehat{v_1}| = |\widehat{f}| = 7$, and $|\widehat{b}| = |\widehat{d}| = 9$.

3. If $m = ww_2$, $|\widehat{v_3}| = |\widehat{w_2}| = |\widehat{g}| = |\widehat{k}| = 2$, $|\widehat{e}| = |\widehat{h}| = |\widehat{m}| = 3$, $|\widehat{w}| = |\widehat{a}| = 4$, $|\widehat{c}| = |\widehat{i}| = |\widehat{j}| = 5$, $|\widehat{f}| = |\widehat{\ell}| = 6$, $|\widehat{v_1}| = |\widehat{v_2}| = |\widehat{u}| = 7$ and $|\widehat{b}| = |\widehat{d}| = 9$.

In each case, we forget v_2, then color a with a color not in $\widehat{h}$, forget h, b, i and color g.

We consider two cases depending on whether w and v_3 are equal.

1. Assume that $w = v_3$. We color e, c, k, j arbitrarily, then v_3 with a color not in $\widehat{f}$, then u, and we apply Lemma 1.22 on $\mathcal{T}(G)$ with the path $f\ell dv_1$.

2. Assume that $w \neq v_3$. We show how to obtain the same configuration regardless of whether $m = ww_1$ or $m = ww_2$.

   (a) If $m = ww_1$, we color w_1 arbitrarily, then n such that $\widehat{u} \neq \widehat{f}$ if $|\widehat{u}| = |\widehat{f}| = 4$ afterwards and finally color m, e, k, c, j.

   (b) If $m = ww_2$, we color e, k, thencolor w_2 such that $\widehat{u} \neq \widehat{f}$ if $|\widehat{u}| = |\widehat{f}| = 3$ afterwards, and finally we color c, m, j arbitrarily.

In both cases, the same set of uncolored elements remains. If $\widehat{f} = \widehat{v_3}$, we color u with a color not in $\widehat{f}$, then forget v_3, f and apply Lemma 1.22 on $\mathcal{T}(G)$ with the path $dv_1\ell w$. Otherwise, we color u such that $\widehat{v_1} \neq \widehat{\ell}$ if $|\widehat{v_1}| = |\widehat{\ell}| = 3$ afterwards. Then we can color v_3 and f, then d, w and v_1, ℓ.

We can also conclude using the Nullstellensatz. We distinguish several cases depending on whether $w = v_3$, $m = ww_1$ or $m = ww_2$ and whether $v_3w_2 \in E(G)$. In each case, we define a monomial with non-zero coefficient in P_G, so that Theorem 1.14 ensures that the coloring extends to G.

1. If $w = v_3$, then $AB^8C^3D^8E^2F^6GH^2I^2J^2KL^5U^4V_1^4V_3^3$ has coefficient 1 in P_G.
2. If $m = ww_1$, then $AB^8C^3D^7E^5F^6GH^2I^2J^2K^4L^4M^2N^2U^6V_1^6V_3^3W^3W_1$ has coefficient -1 in P_G.
3. If $m = ww_2$ and $v_3w_2 \notin E(G)$, then $AB^8C^4D^8E^2F^5GH^2I^4J^4KL^5M^2U^6V_1^5V_3W^3W_2$ has coefficient 1 in P_G.
4. If $m = ww_2$ and $v_3w_2 \notin E(G)$, then $A^3B^8C^3D^8E^2F^5GH^2I^4J^4KL^5M^2U^4V_1^5V_3^2W^3W_2^2$ has coefficient 1 in P_G. □

Configuration C_{10}

Lemma 1.40. *The graph G does not contain a 7-vertex u with three weak neighbors v_1, v_2, v_3 of degree 4 and a neighbor v_4 of degree 7.*

Proof. As G does not contain C_2, we may assume that v_4 is adjacent to only one vertex among $\{v_1, v_2, v_3\}$. Moreover, due to C_1, we may assume (up to renaming the vertices) that the situation is depicted in Figure 1.43. By minimality, we color $G' = G \setminus \{a, \dots, m\}$ and uncolor u, v_1, v_2, v_3.

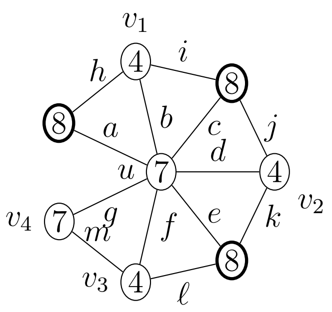


Figure 1.43 – Notation for Lemma 1.40

By Remark 1.12, we may assume that $|\widehat{h}| = 2$, $|\widehat{a}| = |\widehat{i}| = |\widehat{j}| = |\widehat{k}| = |\widehat{\ell}| = |\widehat{m}| = 3$, $|\widehat{c}| = |\widehat{e}| = |\widehat{g}| = 4$, $|\widehat{u}| = |\widehat{v_1}| = |\widehat{v_2}| = |\widehat{v_3}| = 6$ and $|\widehat{b}| = |\widehat{d}| = |\widehat{f}| = 9$.

We forget v_1, v_2, v_3 and color h arbitrarily. Take $\alpha \in \widehat{f} \setminus \widehat{b}$, and remove it from $\widehat{\ell}$ and $\widehat{m}$. Assume that we can color every element excepted b and f. If α appears on a, c, d, e, g or u, we end up with $|\widehat{b}| = 2$ and $|\widehat{f}| = 1$, thus we color f then b. Otherwise, we can color f with α, and then b since $\alpha \notin \widehat{b}$.

We can thus forget b and f. Then we can also forget u and d. We then color g with a color not in $\widehat{a}$, then m and ℓ. As $|\widehat{a}| = |\widehat{i}| = 2$ and $|\widehat{c}| = 3$, we can color i such that $\widehat{a} \neq \widehat{c}$ afterwards. We conclude applying Lemma 1.20 on $\mathcal{T}(G)$ with the cycle $cjke$ and the element a.

We can also conclude using the Nullstellensatz: the coefficient of

$$B^6C^2D^8E^3F^8G^3HI^2J^2K^2L^2M^2U^4$$

in P_G is -2. Hence, using Theorem 1.14, we can extend the coloring to G. □

Configuration C_{11}

Due to the definitions of C_{11} and S_3-neighbor, if G contains C_{11}, then we are in one of the following cases:

- C_{11a}: $\mathrm{dist}_u(v_1, v_3) = 2$ and the common neighbor w of v_1, u and v_3 has degree seven.
- C_{11b}: $\mathrm{dist}_u(v_1, v_3) = 3$ and u, v_3 share a common neighbor w_1 of degree six.
- C_{11c}: $\mathrm{dist}_u(v_1, v_3) = 3$ and v_3 has two neighbors w_2, w_3 of degree six.
- C_{11d}: $\mathrm{dist}_u(v_1, v_3) = 3$ and v_3 has a neighbor w of degree five.

We dedicate a lemma to each of these configurations.

Lemma 1.41. *The graph G does not contain C_{11a}.*

Proof. We use the notation depicted in Figure 1.44. By minimality, we color $G' = G \setminus \{a, \dots, \ell\}$ and uncolor u, v_1, v_2, v_3, w.

By Remark 1.12, we may assume that: $|\widehat{f}| = |\widehat{k}| = |\widehat{w}| = 2$, $|\widehat{e}| = |\widehat{i}| = |\widehat{j}| = |\widehat{\ell}| = 3$, $|\widehat{v_3}| = |\widehat{c}| = 4$, $|\widehat{h}| = 5$, $|\widehat{a}| = |\widehat{v_2}| = 6$, $|\widehat{u}| = |\widehat{v_1}| = |\widehat{g}| = 7$ and $|\widehat{b}| = |\widehat{d}| = 9$.

We forget v_1, v_2, then color a with a color not in $\widehat{w} \cup \widehat{\ell}$, then f, e, k, c, j, i. We then color u with a color not in $\widehat{w}$. Remove a color $\alpha \in \widehat{b} \setminus \widehat{d}$ from $\widehat{h}$, if any. Apply Lemma 1.22 on $\mathcal{T}(G)$ with the path $gv_3\ell wh$. Due to the choice of α, either g has color α and we have $|\widehat{d}| = 2$, $|\widehat{b}| = 1$ and we can color b, d, or we can color b with α and color d since $|\widehat{d}| = 1$.

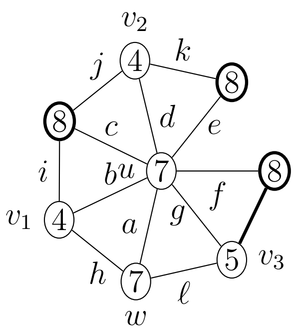


Figure 1.44 – Notation for Lemma 1.41

We can also conclude using the Nullstellensatz: the coefficient of

$$A^5B^7C^2D^8E^2FG^6H^4I^2J^2KL^2U^4V_3^3W$$

in P_G is -1. Hence, using Theorem 1.14, we can extend the coloring to G. $\square$

Lemma 1.42. *The graph G does not contain C_{11b}.*

Proof. We use the notation depicted in Figure 1.45. By minimality, we color $G' = G \setminus \{a, \dots, n\}$ and uncolor $u, v_1, v_2, v_3, w_1, w_2$. By Remark 1.12, we may

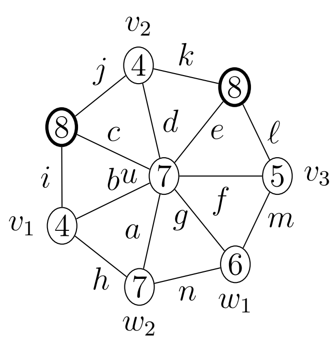


Figure 1.45 – Notation for Lemma 1.42

assume that: $|\widehat{\ell}| = 2$, $|\widehat{i}| = |\widehat{j}| = |\widehat{k}| = |\widehat{n}| = 3$, $|\widehat{w_1}| = |\widehat{c}| = |\widehat{e}| = 4$, $|\widehat{h}| = |\widehat{m}| = 5$, $|\widehat{a}| = |\widehat{v_2}| = 6$, $|\widehat{g}| = |\widehat{v_1}| = 7$, $|\widehat{u}| = |\widehat{f}| = 8$ and $|\widehat{b}| = |\widehat{d}| = 9$.

If $v_3w_2 \in E(G)$, we have $|\widehat{v_3}| = 6$ and $|\widehat{w_2}| = 3$. Otherwise, we may assume that $|\widehat{v_3}| = 5$ and $|\widehat{w_2}| = 2$.

We forget v_1, v_2 and color k, ℓ arbitrarily. Then we remove a color $\alpha \in \widehat{b} \setminus \widehat{d}$ from $\widehat{h}$ and $\widehat{i}$. Assume that we can color every element excepted b and d. Then either α appears on a, c, e, f, g, u and we have $|\widehat{b}| = 1$ and $|\widehat{d}| = 2$, or α is still in $\widehat{b}$ at the end, therefore, we have $\widehat{b} \neq \widehat{d}$. In both cases, we can color b and d.

We may thus forget b and d. We remove a color $\beta \in \widehat{u} \setminus \widehat{f}$ from $\widehat{w_1}$ and $\widehat{w_2}$. We color $w_2, n, w_1, m, v_3, h, i, j, c, e, a, g$. Due to the choice of β, either β

appears on a, c, e, g or v_3 and we have $|\widehat{u}| = 1$ and $|\widehat{f}| = 2$, or β is still in $\widehat{u}$ so $\widehat{u} \neq \widehat{f}$. We can thus color u then f.

We can also conclude using the Nullstellensatz. Let

$$m = B^4C^3D^8E^3F^7G^6H^4I^2J^2K^2LM^4N^2U^7V_3^4W_1^3W_2$$

If $v_3w_2 \notin E(G)$, the coefficient of m in P_G is 3, otherwise, $\frac{mV_3W_2}{B}$ has coefficient -2 in P_G. Hence, using Theorem 1.14, we can extend the coloring to G. □

Lemma 1.43. *The graph G does not contain C_{11c}.*

Proof. We use the notation depicted in Figure 1.46. By minimality, we color $G \setminus \{a, \ldots, q\}$ and uncolor $u, v_1, v_2, v_3, w_1, w_2, w_3$. By Remark 1.12, we may

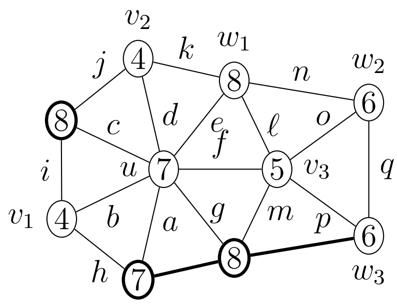


Figure 1.46 – Notation for Lemma 1.43

assume that: $|\widehat{g}| = |\widehat{h}| = |\widehat{i}| = |\widehat{j}| = |\widehat{m}| = |\widehat{n}| = |\widehat{q}| = 3$, $|\widehat{w_2}| = |\widehat{a}| = |\widehat{c}| = 4$, $|\widehat{k}| = 5$, $|\widehat{v_1}| = |\widehat{e}| = |\widehat{\ell}| = |\widehat{p}| = 6$, $|\widehat{u}| = |\widehat{v_2}| = |\widehat{o}| = 7$, $|\widehat{v_3}| = |\widehat{b}| = |\widehat{d}| = 9$ and $|\widehat{f}| = 10$. Moreover, depending on the presence of the edge w_1w_3 in $E(G)$ we may assume that $|\widehat{w_1}|, |\widehat{w_3}|$ are 2 or 3.

We did not succeed in finding a suitable monomial for the Nullstellensatz approach, hence we only present a case analysis proof. We forget v_1, v_2 and we color a with a color not in $\widehat{h}$. Then we forget h, b, i, d, j, k and color c with a color not in $\widehat{g}$.

Note that if $w_1w_3 \notin E(G)$, then we may assume that $\widehat{w_1} \cap \widehat{w_3} = \varnothing$, since otherwise we color them with the same color, then color $g, m, n, q, w_2, e, \ell, o, p, u, f, v_3$.

We remove a color $\alpha \in \widehat{v_3} \setminus \widehat{f}$ from $\widehat{w_1}, \widehat{w_2}, \widehat{w_3}$. Assume that we can color every element excepted v_3 and f. Either α appears on ℓ, m, o, p or u so $|\widehat{v_3}| = 1$ and $|\widehat{f}| = 2$, or α is still in $\widehat{v_3}$ at the end, so $\widehat{v_3} \neq \widehat{f}$. Thus we can extend the coloring to v_3 and f. We may thus forget v_3 and f, and then o, p, ℓ, m, u, e, g also. Considering the edge w_1w_3, we have two cases:

1. If $w_1w_3 \notin E(G)$, $\widehat{w_1}$ and $\widehat{w_3}$ are disjoint, so at most one of them (say w) loses a color when we removed α. We color w, then apply Lemma 1.22 to $\mathcal{T}(G)$ with the path nw_2qw_3 if $w = w_1$ or qw_2nw_1 if $w = w_3$.

2. Assume that $w_1w_3 \in E(G)$. If $\widehat{w_1} = \widehat{w_3}$, we color w_2 with a color not in $\widehat{w_1}$, then apply Corollary 1.19 on the cycle w_1w_3qn in $\mathcal{T}(G)$. Otherwise, we color w_1 with a color not in $\widehat{w_3}$, then apply Lemma 1.22 on $\mathcal{T}(G)$ with the path w_3qw_2n. □

Lemma 1.44. *The graph G does not contain C_{11d}.*

Proof. We use the notation depicted in Figure 1.47. By minimality, we color $G' = G \setminus \{a, \dots, n\}$ and uncolor u, v_1, v_2, v_3, w.

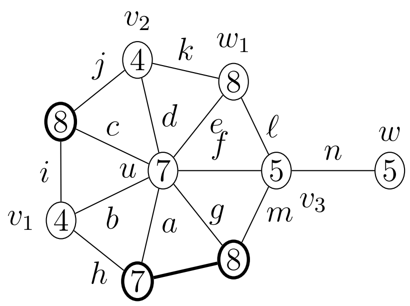


Figure 1.47 – Notation for Lemma 1.44

By Remark 1.12, we may assume that: $|\widehat{w}| = |\widehat{m}| = 2$, $|\widehat{g}| = |\widehat{h}| = |\widehat{i}| = |\widehat{j}| = |\widehat{k}| = |\widehat{\ell}| = 3$, $|\widehat{a}| = |\widehat{c}| = |\widehat{e}| = 4$, $|\widehat{n}| = 5$, $|\widehat{u}| = |\widehat{v_1}| = |\widehat{v_2}| = |\widehat{v_3}| = 6$ and $|\widehat{b}| = |\widehat{d}| = |\widehat{f}| = 9$.

We forget v_1, v_2, then color a with a color not in $\widehat{h}$. We forget h, b, i, d, j, k and color g, m, ℓ, e, c, u arbitrarily. We conclude by applying Lemma 1.22 on $\mathcal{T}(G)$ to the path fv_3nw.

We can also conclude using the Nullstellensatz: the coefficient of

$$B^7C^3D^8E^3F^8G^2H^2I^2JK^2L^2MN^3U^5V_3^5W$$

in P_G is -3. Hence, using Theorem 1.14, we can extend the coloring to G. □

Configuration C_{12}

Note that, when v_3 is an S_3-neighbor of u, it cannot be a $(6, 6^+)$-neighbor of u otherwise we obtain C_1 since the 6-vertex would be adjacent to v_1 or v_2. Thus, due to the definitions of C_{12} and S_3-neighbor, if G contains C_{11}, then we are in one of the following cases:

- C_{12a}: v_1 is a $(7, 7)$-neighbor of u.
- C_{12b}: v_3 is a $(7, 7)$-neighbor of u.
- C_{12c}: v_1 is a $(7, 8)$-neighbor of u and v_3 has two neighbors w_2, w_3 of degree 6 such that uv_3w_2, uv_3w_3 are not triangular faces.

- C_{12d}: v_1 is a $(7,8)$-neighbor of u and v_3 has a neighbor w of degree 5 such that uv_3w is not a triangular face.

We dedicate a lemma to each of these configurations.

Lemma 1.45. *The graph G does not contain C_{12a}.*

Proof. We use the notation depicted in Figure 1.48. By minimality, we color $G' = G \setminus \{a, \dots, n\}$ and uncolor u, v_1, v_2, v_3, w.

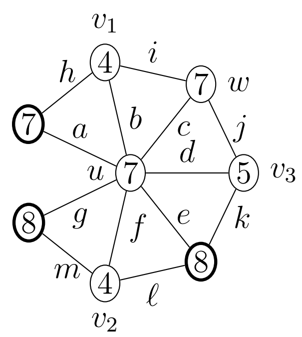


Figure 1.48 – Notation for Lemma 1.45

By Remark 1.12, we may assume that: $|\widehat{w}| = |\widehat{k}| = |\widehat{m}| = 2$, $|\widehat{g}| = |\widehat{h}| = |\widehat{\ell}| = 3$, $|\widehat{a}| = |\widehat{e}| = |\widehat{j}| = 4$, $|\widehat{v_3}| = |\widehat{i}| = 5$, $|\widehat{v_2}| = |\widehat{c}| = 6$, $|\widehat{u}| = |\widehat{v_1}| = 7$, $|\widehat{d}| = 8$ and $|\widehat{b}| = |\widehat{f}| = 9$.

We forget v_1, v_2 and color m in order to obtain $\widehat{\ell} \neq \widehat{k}$. Then we remove from $\widehat{h}$ and $\widehat{i}$ a color $\alpha \in \widehat{b} \setminus \widehat{f}$. We can forget b and f. Indeed, if we can color the remaining elements and α appears on a, c, d, e, g, or u, we end up with $|\widehat{f}| = 2$ and $|\widehat{b}| = 1$. Otherwise, at the end, we still have $\alpha \in \widehat{b}$ but $\alpha \notin \widehat{f}$, thus we may color b with α and end up with $|\widehat{f}| > 0$.

We now color ℓ and e with colors not in $\widehat{k}$. Then we color g, w, a, h arbitrarily. If $\widehat{c} = \widehat{i}$, we color j with a color not in $\widehat{c}$, then k. We forget i and apply Lemma 1.22 on $\mathcal{T}(G)$ with the path $cudv_3$. Otherwise, we color j with a color not in $\widehat{k}$, then i and c (since $\widehat{i} \neq \widehat{c}$) and apply Lemma 1.22 on $\mathcal{T}(G)$ with the path udv_3k.

We can also conclude using the Nullstellensatz: the coefficient of

$$A^3B^7C^5D^7E^3F^8G^2H^2I^4J^3KL^2MUV_3^4W$$

in P_G is 3. Hence, using Theorem 1.14, we can extend the coloring to G. □

Lemma 1.46. *The graph G does not contain C_{12b}.*

Proof. We use the notation depicted in Figure 1.49. By minimality, we color $G' = G \setminus \{a, \dots, m\}$ and uncolor $u, v_1, v_2, v_3, w_1, w_2$.

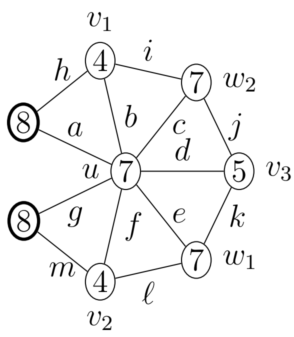


Figure 1.49 – Notation for Lemma 1.46

By Remark 1.12, we may assume that: $|\widehat{w_1}| = |\widehat{w_2}| = |\widehat{h}| = |\widehat{m}| = 2$, $|\widehat{a}| = |\widehat{g}| = 3$, $|\widehat{j}| = |\widehat{k}| = 4$, $|\widehat{i}| = |\widehat{\ell}| = 5$, $|\widehat{v_3}| = |\widehat{c}| = |\widehat{e}| = 6$, $|\widehat{v_1}| = |\widehat{v_2}| = 7$, $|\widehat{u}| = |\widehat{d}| = 8$ and $|\widehat{b}| = |\widehat{f}| = 9$.

We forget v_1, v_2, color m, then remove from $\widehat{h}$ and $\widehat{i}$ a color $\alpha \in \widehat{b} \setminus \widehat{f}$. We then color h, a, g, w_2, i, j, c. We color v_3 with a color not in $\widehat{k}$, then w_1, apply Lemma 1.22 on $\mathcal{T}(G)$ with the path $udek$ and finally color ℓ.

Note that due to the choice of α, if α appears on u, a, c, d, e or g then $|\widehat{f}| = 2$, so we can color b and f. Otherwise, we can put color α on b and then color f (since $\alpha \notin \widehat{f}$).

We can also conclude using the Nullstellensatz. Let

$$m = B^3C^4D^7E^5F^8G^2HI^4J^3K^3L^4MU^7V_3^5W_1W_2$$

If $w_1w_2 \notin E(G)$, the coefficient of m in P_G is 3, otherwise, $\frac{mW_1W_2}{B}$ has coefficient 1 in P_G. Hence, using Theorem 1.14, we can extend the coloring to G. $\square$

Lemma 1.47. *The graph G does not contain C_{12c}.*

Proof. We use the notation depicted in Figure 1.50. Recall that v_1 is a $(7,8)$-neighbor of u, hence w or w_4 has degree 7. By minimality, we color $G \setminus \{a, \dots, r\}$ and uncolor $u, v_1, v_2, v_3, w_1, w_2, w_3, w_4$. By Remark 1.12, we may assume that: $|\widehat{m}| = 2$, $|\widehat{g}| = |\widehat{q}| = 3$, $|\widehat{r}| = 4$, $|\widehat{\ell}| = 5$, $|\widehat{e}| = |\widehat{k}| = 6$, $|\widehat{o}| = |\widehat{p}| = |\widehat{v_1}| = |\widehat{v_2}| = 7$, $|\widehat{u}| = 8$, $|\widehat{b}| = |\widehat{f}| = 9$ and $|\widehat{v_3}| = |\widehat{d}| = 10$. Moreover, depending on the presence of edges between the w_i's, their lists size may vary, but we may assume that $|\widehat{w_1}| \geqslant 2$ and $|\widehat{w_2}|, |\widehat{w_3}|$ are at least 4. We did not succeed in finding a suitable monomial for the Nullstellensatz approach, hence we only present a case analysis proof. We forget v_1, v_2.

We separate two cases depending on the degrees of w_4 and w:

1. We first assume that $d(w_4) = 8$ and $d(w) = 7$. Then we may also assume that $|\widehat{h}| = |\widehat{n}| = 3$, $|\widehat{a}| = 4$, $|\widehat{i}| = 5$, $|\widehat{c}| = |\widehat{j}| = 6$ and $|\widehat{w_4}| \geqslant 2$.

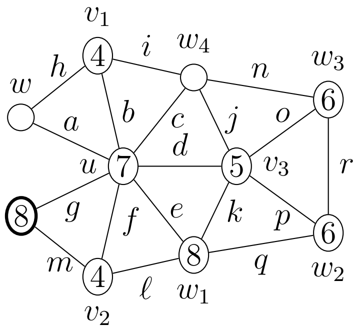


Figure 1.50 – Notation for Lemma 1.47

We remove from $\widehat{w_3}$ and $\widehat{r}$ a color $\alpha \in \widehat{o} \setminus \widehat{j}$, if any. We then color w_2 with a color not in $\widehat{r}$, then w_1 and q, and apply Lemma 1.22 on $\mathcal{T}(G)$ with the path $w_4 n w_3 r$.

Due to the choice of α, we may now color j with a color not in $\widehat{o}$, then color $i, c, h, a, g, e, m, \ell, k$. We color u such that $\widehat{v_3} \neq \widehat{o}$, then b, f, d, p. Since $\widehat{v_3} \neq \widehat{o}$, we can finally color v_3 and o.

2. Assume that $d(w_4) = 7$ and $d(w) = 8$. We may assume that $|\widehat{h}| = 2$, $|\widehat{a}| = 3$, $|\widehat{n}| = 4$, $|\widehat{i}| = 6$, $|\widehat{c}| = |\widehat{j}| = 7$ and $|\widehat{w_4}| \geqslant 4$.

   We color g, ℓ with a color not in $\widehat{m}$. Then we forget m, f, b, then h, i (or i, h, depending on whether w has degree 7 or 8) and color w_1, q, a, e, k, w_2, r, w_3, n, o, p. We then color v_3 with a color not in $\widehat{w_4}$. If $\widehat{w_4} = \widehat{j}$, we color c with a color not in $\widehat{j}$, then apply Corollary 1.19 on the cycle uw_4jd in $\mathcal{T}(G)$. Otherwise, we color w_4 with a color not in $\widehat{j}$, then apply Lemma 1.22 on $\mathcal{T}(G)$ with the path $jdcu$. □

Lemma 1.48. *The graph G does not contain C_{12d}.*

Proof. We use the notation depicted in Figure 1.51. By minimality, we color $G' = G \setminus \{a, \ldots, n\}$ and uncolor u, v_1, v_2, v_3, w_1.

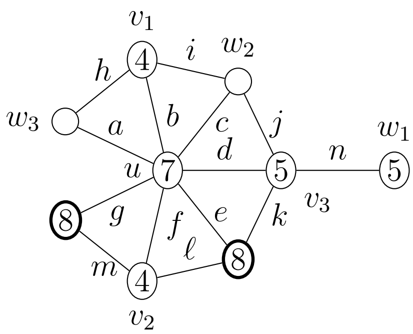


Figure 1.51 – Notation for Lemma 1.48

By Remark 1.12, we may assume that: $|\widehat{w_1}| = |\widehat{m}| = 2$, $|\widehat{g}| = |\widehat{k}| = |\widehat{\ell}| = 3$, $|\widehat{e}| = 4$, $|\widehat{n}| = 5$, $|\widehat{u}| = |\widehat{v_1}| = |\widehat{v_2}| = |\widehat{v_3}| = 6$ and $|\widehat{b}| = |\widehat{d}| = |\widehat{f}| = 9$.

Moreover, if $d(w_2) = 8$ and $d(w_3) = 7$, we have $|\widehat{h}| = |\widehat{i}| = |\widehat{j}| = 3$ and $|\widehat{a}| = |\widehat{c}| = 4$. Otherwise, $d(w_2) = 7$ and $d(w_3) = 8$ so $|\widehat{h}| = 2$, $|\widehat{a}| = 3$, $|\widehat{i}| = |\widehat{j}| = 4$ and $|\widehat{c}| = 5$. We forget v_1, v_2 and color g, ℓ with a color not in $\widehat{m}$, then forget m, f, b, i, h. We color a, e, k, c, j, u and apply Lemma 1.22 on $\mathcal{T}(G)$ with the path dv_3nw_1.

We can also conclude using the Nullstellensatz. Let

$$m = B^7C^2D^8E^3F^8G^2HI^2J^2K^2L^2MN^3V_3^4W_1$$

If $d(w_2) = 8$ and $d(w_3) = 7$, then mHU^5V_3 has coefficient -1 in P_G. Otherwise, $mABC^2IJN$ has coefficient 1 in P_G. Hence, in each case, using Theorem 1.14, we can extend the coloring to G. □

Configuration C_{13}

By definition, if G contains C_{13}, then we are in one of the following cases ($v_1, \ldots, v_8$ denote the neighbors of u in cyclic ordering around u):

- C_{13a}: u has four neighbors of degree 6, and four $(6,6)$-neighbors of degree 5. We may assume that $d(v_{2i}) = 5$ and $d(v_{2i-1}) = 6$ for $1 \leqslant i \leqslant 4$ and that $v_1v_2, \ldots, v_7v_8, v_8v_1$ are in $E(G)$.

- C_{13b}: u has five weak neighbors of degree 5 and three neighbors of degree 6. Due to C_4 and C_3, we may assume that v_1, v_2, v_4, v_6, v_7 have degree 5, v_3, v_5, v_8 have degree 6 and that $v_1v_2, \ldots, v_7v_8, v_8v_1$ are in $E(G)$.

- C_{13c}: u has four neighbors of degree 6, two $(6,6)$-neighbors of degree 5 at triangle-distance 2, and two $(5,6)$-neighbors of degree 5. We may assume that v_2, v_4, v_6, v_7 have degree 5, v_1, v_3, v_5, v_8 have degree 6 and that $v_1v_2, \ldots, v_7v_8$ are in $E(G)$.

- C_{13d}: u has four neighbors of degree 6, two $(6,6)$-neighbors of degree 5 at triangle-distance at least 3, and two $(5,6)$-neighbors of degree 5. We may assume that v_2, v_4, v_5, v_7 have degree 5, v_1, v_3, v_6, v_8 have degree 6 and that $v_1v_2, \ldots, v_7v_8$ are in $E(G)$.

We dedicate a lemma to each of these configurations. In each of them, we did not succeed in finding a suitable monomial for the Nullstellensatz approach, hence we only present case analysis proofs.

Lemma 1.49. *The graph G does not contain C_{13a}.*

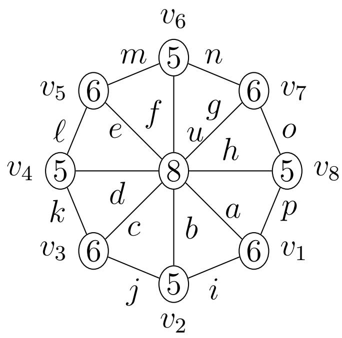


Figure 1.52 – Notation for Lemma 1.49

Proof. We use the notation depicted in Figure 1.52. By minimality, we color $G \backslash \{a, \ldots, p\}$ and uncolor $u, v_1, \ldots, v_8$. First note that there is no edge between 5-vertices excepted maybe v_2v_6 and v_4v_8 since otherwise, it would create C_3.

Using that G is planar, we first show the following:

1. We may assume (up to symmetry) that there is no edge between v_2 and v_5, v_6, v_7.

   Assume that v_6 or v_7 is a neighbor of v_2. Then there is no edge between v_8 and v_3, v_4, v_5, otherwise, $\{\{u\}, \{v_3, v_4, v_5\}, \{v_8\}, \{v_6, v_7\}, \{v_1, v_2\}\}$ is a K_5-minor of G. By exchanging v_2 and v_8, we obtain that v_2 has no neighbor among v_5, v_6, v_7.

   If v_2v_5 is an edge, we obtain the same result by exchanging v_2 and v_4.

2. With such a v_2, we may also assume that v_4 has at most one neighbor among v_1, v_7, v_8. First note that if $v_4v_8 \in E(G)$, then v_1, v_7 are not neighbors of v_4 due to C_3. In this case, v_4 has thus only one neighbor among v_1, v_7, v_8.

   Otherwise, both v_1 and v_7 are neighbors of v_4, so there is no edge between vv_8 with $v \in \{v_3, v_5\}$. Indeed, otherwise, $\{u, v, v_1, v_4, v_7, v_8\}$ would be a $K_{3,3}$ minor of G. Thus, by exchanging v_4 and v_8, we obtain that v_4 has at most one neighbor among v_1, v_7, v_8.

By Remark 1.12, we may thus assume that: $|\widehat{i}| = |\widehat{j}| = |\widehat{k}| = |\widehat{\ell}| = |\widehat{m}| = |\widehat{n}| = |\widehat{o}| = |\widehat{p}| = 5$, $|\widehat{v_2}| = 6$, $|\widehat{a}| = |\widehat{c}| = |\widehat{e}| = |\widehat{g}| = 7$, $|\widehat{b}| = |\widehat{d}| = |\widehat{f}| = |\widehat{h}| = 8$ and $|\widehat{u}| = 10$. Moreover, $\widehat{v_1}, \widehat{v_3}, \widehat{v_5}$ and $\widehat{v_7}$ have size at least 4, and $\widehat{v_6}, \widehat{v_8}$ at least 6.

Due to the previous observations, we may also assume that $|\widehat{v_4}|$ is 6 or 7. We separate three cases:

1. Assume that $\widehat{n} \not\subset \widehat{v_6}$. Then we color n with a color not in $\widehat{v_6}$, d with a color not in $\widehat{v_4}$, g with a color not in $\widehat{o}$ and h with a color not in $\widehat{p}$. We

then color $a, c, e, f, b, u, v_7, m, v_5, \ell$, and forget v_4, v_6. We color v_8 with a color not in $\widehat{o}$, then v_1 and forget o, p. We finally apply Lemma 1.22 on $\mathcal{T}(G)$ with the path iv_2jv_3k.

2. If $\widehat{n} \subset \widehat{v_6}$ (and by symmetry $\widehat{i} \subset \widehat{v_2}$) and v_6 is not a neighbor from both v_1, v_3. Then $|\widehat{v_6}| < 8$ and we can color f with a color not in $\widehat{v_6}$ (hence not in $\widehat{n}$), and b with a color not in $\widehat{v_2}$ (hence not in $\widehat{i}$). Then we color a, c, e, g, h, d, u and forget v_2, i, j, v_6, m, n, and use Theorem 1.18 to color $\{v_1, v_3, v_4, v_5, v_7, v_8, k, \ell, o, p\}$.

3. Otherwise, we color b with a color not in $\widehat{v_2}$, then color $a, c, e, g, d, f, h, u, v_3$ and forget v_2, i, j. We apply Lemma 1.22 on $\mathcal{T}(G)$ with the path $kv_4\ell v_5$, then color m.

   If $v_1v_7 \notin E(G)$, then $|\widehat{v_7}| = 2 = |\widehat{n}|$. If $\widehat{n} = \widehat{v_7}$, we color v_6 and o with a color not in $\widehat{n}$, then color v_1, p, v_8, v_7, n. Otherwise, we color n with a color not in $\widehat{v_7}$, then color v_1 with a color not in v_6, forget v_6 and apply Lemma 1.22 on $\mathcal{T}(G)$ with the path pv_8ov_7.

   Otherwise, $v_1v_7 \in E(G)$. We then color v_1 with a color not in $\widehat{v_6}$, forget v_6 and apply Lemma 1.22 on $\mathcal{T}(G)$ with the path pv_8ov_7n. □

Lemma 1.50. *The graph G does not contain C_{13b} nor C_{13c}.*

Proof. First note that C_{13b} is a sub-configuration of C_{13c}. It is thus sufficient to prove that G does not contain C_{13c}. We use the notation depicted in Figure 1.53. By minimality, we color $G \setminus \{a, \dots, o\}$ and uncolor $u, v_1, \dots, v_8$.

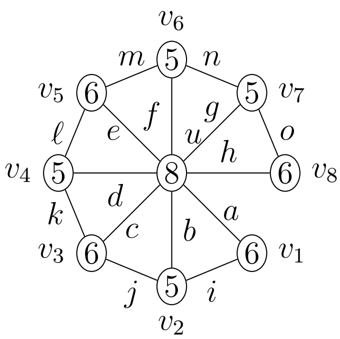


Figure 1.53 – Notation for Lemma 1.50

By Remark 1.12, we may assume that $|\widehat{i}| = |\widehat{o}| = 4$, $|\widehat{j}| = |\widehat{k}| = |\widehat{\ell}| = |\widehat{m}| = 5$, $|\widehat{a}| = |\widehat{h}| = |\widehat{n}| = 6$, $|\widehat{c}| = |\widehat{e}| = 7$, $|\widehat{b}| = |\widehat{d}| = |\widehat{f}| = |\widehat{g}| = 8$ and $|\widehat{u}| = 10$. Moreover, $|\widehat{v_1}|, |\widehat{v_8}|$ are at least 2, $|\widehat{v_3}|, |\widehat{v_5}|$ are at least 4 and $|\widehat{v_2}|, |\widehat{v_4}|, |\widehat{v_6}|, |\widehat{v_7}|$ are at least 6.

We color c with a color not in $\widehat{j}$ and d with a color not in $\widehat{k}$. Then, we color v_1, and v_8 such that $\widehat{u} \neq \widehat{f}$. We color a, h, e, b, g, then u, f since $\widehat{u} \neq \widehat{f}$, and color i, o. We then color v_3 such that $\widehat{v_4} \neq \widehat{\ell}$. Then we color v_2, j, k (if $v_2v_4 \in E(G)$, when we color v_2, we ensure that we still have $\widehat{v_4} \neq \widehat{\ell}$). We then color $\widehat{v_5}$ such that $\widehat{v_6} \neq \widehat{n}$, v_4 and ℓ (since $\widehat{v_4}$ and $\widehat{\ell}$ are different and of size at least one), then m, v_7 arbitrarily, and finally v_6 and n (since again $\widehat{v_6}$ and $\widehat{n}$ are different of size at least one). □

Lemma 1.51. *The graph G does not contain C_{13d}.*

Proof. We use the notation depicted in Figure 1.54. By minimality, we color $G \setminus \{a, \ldots, o\}$ and uncolor $u, v_1, \ldots, v_8$.

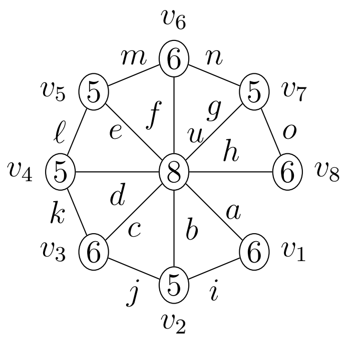


Figure 1.54 – Notation for Lemma 1.51

By Remark 1.12, we may assume that $|\widehat{i}| = |\widehat{o}| = 4$, $|\widehat{j}| = |\widehat{k}| = |\widehat{m}| = |\widehat{n}| = 5$, $|\widehat{a}| = |\widehat{h}| = |\widehat{\ell}| = 6$, $|\widehat{c}| = |\widehat{f}| = 7$, $|\widehat{b}| = |\widehat{d}| = |\widehat{e}| = |\widehat{g}| = 8$ and $|\widehat{u}| = 10$. Moreover, $|\widehat{v_1}|, |\widehat{v_8}|$ are at least 2, $|\widehat{v_3}|, |\widehat{v_6}|$ are at least 4 and $|\widehat{v_2}|, |\widehat{v_4}|, |\widehat{v_5}|, |\widehat{v_7}|$ are at most 6.

We color f with a color not in $\widehat{n}$ and b with a color not in $\widehat{j}$, then we color v_8, and v_1 such that $\widehat{u} \neq \widehat{d}$. We color a, h, c, e, g, then u and d since $\widehat{u} \neq \widehat{d}$, then $i, o, v_6, v_7, n, m, v_5$. We color v_3 such that $\widehat{v_4} \neq \widehat{\ell}$, then v_2, j, k and finally v_4, ℓ since $\widehat{v_4} \neq \widehat{\ell}$. □

Configuration C_{14}

By definition, if G contains C_{14}, then we are in one of the following cases:

- C_{14a}: $v_1, \ldots, v_4$ are weak neighbors of u of degree 4 and u has a neighbor w of degree 7.
- C_{14b}: $v_1, \ldots, v_4$ are $(7,8)$-neighbors of u such that v_1, v_2, v_3 have degree 4 and v_4 has degree at most 5.
- C_{14c}: u has a $(7,7)$-neighbor v_1 of degree 4, a weak neighbor v_2 of degree 4 and two non-adjacent neighbors v_3, v_4 of degree 5 such that $\mathrm{dist}_u(v_1, v_2) = \mathrm{dist}_u(v_1, v_3) = 2$.

We dedicate a lemma to each of these configurations, and we begin with a preliminary lemma, used to edge-color the following graph:

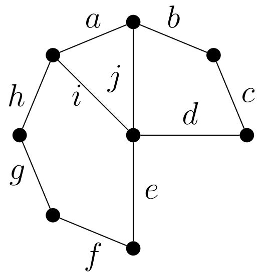


Figure 1.55 – Notation for Lemma 1.52

Lemma 1.52. *We can produce an edge-coloring of the graph H given in Figure 1.55 as soon as $|\widehat{c}|, |\widehat{f}|, |\widehat{g}|, |\widehat{i}|$ are at least 2, $|\widehat{j}| \geqslant 4$ and any other list has length at least 3.*

Proof. We have the following:

1. If $\widehat{i} \not\subset \widehat{e}$, then we color i with a color not in $\widehat{e}$, then d such that $\widehat{j} \neq \widehat{a}$ and c, and finally apply Lemma 1.20 on $\mathcal{T}(H)$ with the cycle $jefgha$ and the element b.

2. If $\widehat{i} \not\subset \widehat{d}$, then we color i with a color not in $\widehat{d}$. Then, if e, f, g, h, a share the same list, we color them putting the same color on e and a, then apply Corollary 1.19 on the cycle $jdcb$ of $\mathcal{T}(H)$. Otherwise, we may color one edge among e, f, g, h, a with a color not in the list of one of its neighbors, then we can color or forget the other edges excepted a or e, and apply Lemma 1.20 on $\mathcal{T}(H)$ with either the cycle $bcdj$ and the element a or the cycle $jbcd$ and the element e.

3. If $\widehat{d} \neq \widehat{e}$, then we color d with a color not in $\widehat{e}$ (thus not in $\widehat{i}$), then i such that $\widehat{a} \neq \widehat{b}$. We then apply Lemma 1.20 on $\mathcal{T}(H)$ with the cycle $ahgfej$ and the element b.

Otherwise, we have $\widehat{i} \subset \widehat{d} = \widehat{e}$ so we can color j with a color not in $\widehat{d} \cup \widehat{e} \cup \widehat{i}$. If $\widehat{i} = \widehat{a}$ afterwards, then we color h with a color not in $\widehat{i}$, then g and f, and apply Lemma 1.20 on $\mathcal{T}(G)$ with the cycle $dcbai$ and the element e. Otherwise, we color i with a color not in $\widehat{a}$, then apply Corollary 1.19 on the cycle $abcdefgh$ of $\mathcal{T}(H)$.

We can also conclude using the Nullstellensatz: the coefficient of $A^2B^2CD^2E^2FGH^2IJ^3$ in P_H is -1. Hence, using Theorem 1.14, we can find a coloring for H. $\square$

Lemma 1.53. *The graph G does not contain C_{14a}.*

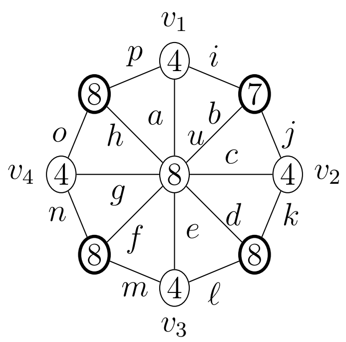


Figure 1.56 – Notation for Lemma 1.53

Proof. We use the notation depicted in Figure 1.56. By minimality, we color $G' = G \setminus \{a, \ldots, p\}$ and uncolor $u, v_1, \ldots, v_4$.

By Remark 1.12, we may assume that: $|\widehat{k}| = |\widehat{\ell}| = |\widehat{m}| = |\widehat{n}| = |\widehat{o}| = |\widehat{p}| = 3$, $|\widehat{d}| = |\widehat{f}| = |\widehat{h}| = |\widehat{i}| = |\widehat{j}| = 4$, $|\widehat{b}| = 5$, $|\widehat{v_1}| = |\widehat{v_2}| = |\widehat{v_3}| = |\widehat{v_4}| = |\widehat{u}| = 6$, and $|\widehat{a}| = |\widehat{c}| = |\widehat{e}| = |\widehat{g}| = 9$.

We forget v_1, v_2, v_3, v_4 and color g with a color not in $\widehat{n} \cup \widehat{o}$. We now distinguish three cases depending on the lists $\widehat{d}, \widehat{f}$ and $\widehat{h}$.

1. Assume that $\widehat{d} = \widehat{f} = \widehat{h}$. Then we remove their colors from $\widehat{a}, \widehat{b}, \widehat{c}, \widehat{e}$ and $\widehat{u}$. Thus we may assume that d, f, h are not adjacent to them in $\mathcal{T}(G)$ anymore. We color b and u. Due to the previous manipulations, coloring G is now equivalent to 3-list-edge-coloring the graph of Figure 1.57. Note

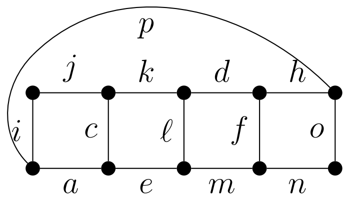


Figure 1.57 – Auxiliary graph for Lemma 1.53

that we now have $\widehat{a} \cap \widehat{h} = \varnothing$, therefore coloring p affects at most one among a, h. Note that Figure 1.57 is symmetrical when we exchange a, c, e, i, j, k with respectively h, f, d, o, n, m. We may thus assume that we can color p with a color not in $\widehat{a}$. We then color n with a color not in $\widehat{o}$ and forget o.

If $\widehat{f} = \widehat{h}$, we color d with a color not in $\widehat{f}$ and forget f, h. Otherwise, we color f with a color not in $\widehat{h}$, then m and we forget h. Up to renaming a, e, m into j, k, d respectively, we may assume we are in the first case (since we obtain the same configuration).

If $\widehat{\ell} = \widehat{m}$, we color e with a color not in $\widehat{\ell}$, then forget m, ℓ and apply Lemma 1.20 on $\mathcal{T}(G)$ with the cycle $caij$ and the element k. Otherwise, we color ℓ with a color not in $\widehat{m}$, forget m, color k and apply Lemma 1.20 on $\mathcal{T}(G)$ with the cycle $aijc$ and the element e.

2. Assume that $\widehat{d} = \widehat{h} \neq \widehat{f}$. We color f with a color not in $\widehat{d}$, u with a color not in $\widehat{d}$, ℓ with a color not in $\widehat{m}$. We distinguish two cases:

   (a) If $|\widehat{b} \cup \widehat{d} \cup \widehat{h}| = 3$, we remove these colors from $\widehat{a}, \widehat{c}$ and $\widehat{e}$. Then, we color k such that $\widehat{c} \neq \widehat{e}$, and color d, b, h.

   If $\widehat{e}, \widehat{m}, \widehat{n}, \widehat{o}, \widehat{p}$ are all the same list of size 2, then, we can color e, n and p with a color not in $\widehat{c}$, then m and o, and apply Corollary 1.19 on the cycle $acji$ in $\mathcal{T}(G)$. Otherwise, denote e, m, n, o, p by $e_1, \ldots, e_5$, and take α as the smallest index such that $\widehat{e_\alpha} \neq \widehat{e_{\alpha+1}}$. We color $e_{\alpha+1}$ with a color not in $\widehat{e_\alpha}$, then forget $e_\alpha, \ldots, e_2$ and color $e_{\alpha+1}, \ldots, e_4$. We then apply Lemma 1.20 on $\mathcal{T}(G)$ with the cycle $aijc$ and the element e.

   (b) Otherwise, we may color b or d with a color not in $\widehat{h}$, then color the other one and k, and we conclude using Lemma 1.52 on $\{p, i, j, c, e, m, n, o, h, a\}$.

3. Assume that $\widehat{d} \neq \widehat{h}$. We color d with a color not in $\widehat{h}$, f such that $\widehat{m} \neq \widehat{\ell}$, ℓ with a color not in $\widehat{m}$, and finally k. We first prove that we may assume that $\widehat{b} = \widehat{h}$ with two cases:

   (a) Assume that $\widehat{b} \neq \widehat{h}$, and if we color b with a color not in $\widehat{h}$, we have $\widehat{u} \neq \widehat{h}$. Then we color u with a color not in $\widehat{h}$ and apply Lemma 1.52 to $\{p, i, j, c, e, m, n, o, h, a\}$.

   (b) Assume that $\widehat{b} \neq \widehat{h}$, and if we color b with a color not in $\widehat{h}$, we have $\widehat{u} = \widehat{h}$. We remove the colors of $\widehat{u}$ from $\widehat{a}, \widehat{c}$ and $\widehat{e}$ and we forget u. This ensures that there is no common color in $\widehat{h}$ and $\widehat{c} \cup \widehat{e}$ anymore. If afterwards, we have $\widehat{c} = \widehat{e}$, we color a with a color not in $\widehat{c}$, then we apply Lemma 1.20 on $\mathcal{T}(G)$ with the cycle $pijcemno$ and the element h (since $\widehat{h} \cap (\widehat{c} \cup \widehat{e}) = \varnothing$). Otherwise, we color c with a color not in $\widehat{e}$, then j and h such that $\widehat{i} \neq \widehat{p}$ and apply Lemma 1.20 on $\mathcal{T}(G)$ with the cycle $aemnop$ and the element i.

   We may thus assume that $\widehat{b} = \widehat{h}$. Then we remove their colors from $\widehat{u}, \widehat{a}, \widehat{c}$ and $\widehat{e}$, and we color u. We conclude with two cases:

   (a) If $\widehat{c} = \widehat{e}$, we color a with a color not in $\widehat{c}$. If $\widehat{h} = \widehat{p}$, we color o with a color not in $\widehat{h}$, then n, m, e, c and apply Lemma 1.20 on $\mathcal{T}(G)$ with the cycle $bhpi$ and the element j. Otherwise, we color h with a color not in $\widehat{p}$, then b and apply Corollary 1.19 on the cycle $cemnopij$ in $\mathcal{T}(G)$.

(b) If $\widehat{c} \neq \widehat{e}$, we color c with a color not in $\widehat{e}$. If $\widehat{b} = \widehat{j}$, we color h such that $\widehat{p} \neq \widehat{i} \backslash \widehat{b}$, then we color b, j and apply Lemma 1.20 on $\mathcal{T}(G)$ with the cycle $aemnop$ and the element i. Otherwise, we color b with a color not in $\widehat{j}$, then forget j, i, color h and apply Corollary 1.19 on the cycle $aemnop$ in $\mathcal{T}(G)$.

We can also conclude using the Nullstellensatz: the coefficient of

$$A^5 B^4 C^8 D^3 E^8 F^3 G^8 H^3 I^3 J^3 K^2 L^2 M^2 N^2 O^2 P^2$$

in P_G is 16. Hence, using Theorem 1.14, we can extend the coloring to G. $\square$

Lemma 1.54. *The graph G does not contain C_{14b}.*

Proof. We use the notation depicted in Figure 1.58. By minimality, we color $G \setminus \{a, \ldots, p\}$ and uncolor $u, v_1, \ldots, v_4, w_1, w_2$.

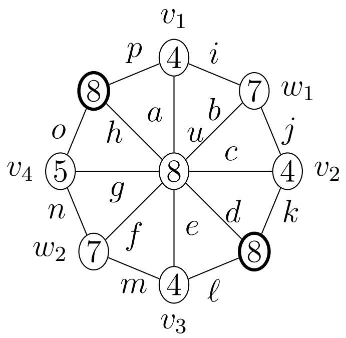


Figure 1.58 – Notation for Lemma 1.54

By Remark 1.12, we may assume that: $|\widehat{o}| = 2$, $|\widehat{k}| = |\widehat{\ell}| = |\widehat{p}| = 3$, $|\widehat{d}| = |\widehat{h}| = |\widehat{n}| = 4$, $|\widehat{v_4}| = |\widehat{i}| = |\widehat{j}| = |\widehat{m}| = 5$, $|\widehat{b}| = |\widehat{f}| = 6$, $|\widehat{v_1}| = |\widehat{v_2}| = |\widehat{v_3}| = 7$, $|\widehat{u}| = |\widehat{g}| = 8$ and $|\widehat{a}| = |\widehat{c}| = |\widehat{e}| = 9$. Moreover, we may also assume that $|\widehat{w_1}|, |\widehat{w_2}|$ are at least 2.

We did not succeed in finding a suitable monomial for the Nullstellensatz approach, hence we only present a case analysis proof. We forget v_1, v_2, v_3, color h with a color not in $\widehat{p}$, then color o. We remove from $\widehat{w_2}, \widehat{f}$ and $\widehat{n}$ a color $\alpha \in \widehat{m} \setminus \widehat{\ell}$. Then, we color w_2, n, f, d. We color u, v_4, w_1, b, g applying Theorem 1.18 on the subgraph of $\mathcal{T}(G)$ they induce. Due to the choice of α, we have $\widehat{m} \neq \widehat{\ell}$ if $|\widehat{m}| = |\widehat{\ell}| = 2$ thus we can color ℓ with a color not in $\widehat{m}$ and forget m. We then color k, then p such that $\widehat{a} \neq \widehat{e}$ and apply Lemma 1.20 on $\mathcal{T}(G)$ with the cycle $aijc$ and the element e. $\square$

Lemma 1.55. *The graph G does not contain C_{14c}.*

Proof. We use the notation depicted in Figure 1.59. By minimality, we color $G' = G \setminus \{a, \ldots, \ell\}$ and uncolor $u, v_1, v_2, v_4, w_1, w_2$.

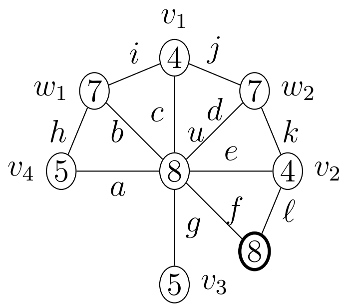


Figure 1.59 – Notation for Lemma 1.55

By Remark 1.12, we may assume that: $|\widehat{f}| = |\widehat{\ell}| = 2$, $|\widehat{h}| = 3$, $|\widehat{v_4}| = |\widehat{g}| = 4$, $|\widehat{b}| = |\widehat{d}| = |\widehat{i}| = |\widehat{j}| = |\widehat{k}| = 5$, $|\widehat{a}| = |\widehat{u}| = 6$, $|\widehat{v_2}| = 7$, and $|\widehat{v_1}| = |\widehat{c}| = |\widehat{e}| = 8$. Moreover, $|\widehat{w_1}|, |\widehat{w_2}|$ are 2 or 3 depending on whether $w_1w_2 \in E(G)$.

We do not have a case analysis proof in this case, only the Combinatorial Nullstellensatz approach. We forget v_1, v_2.

We consider two cases depending on whether w_1w_2 is an edge of G.

1. If $w_1w_2 \notin E(G)$, we have $|\widehat{w_1}| = |\widehat{w_2}| = 2$. The monomial
$$m = U^5V_4^3W_2A^5B^4C^7D^4E^7FG^3H^2I^4J^4K^4$$
has coefficient 1 in P_G.

2. If $w_1w_2 \in E(G)$, then $|\widehat{w_1}| = |\widehat{w_2}| = 3$. The monomial
$$m' = U^5V_4^3W_2^2A^5B^3C^7D^4E^7FG^3H^2I^4J^4K^4L = \frac{W_2L}{B}m$$
has coefficient -1 in P_G.

Therefore, we can extend the coloring to G. □

Configuration C_{15}

Due to C_4 and to the definition of C_{15}, if G contains C_{15} then G contains a subconfiguration of one of the three following cases:

- C_{15a}: u has two $(6,6)$-neighbors of degree 5.
- C_{15b}: u has three weak neighbors of degree 5 and two neighbors of degree 6, such that there is a triangular face containing u and two vertices of degree 5.
- C_{15c}: u has three weak neighbors of degree 5 and two neighbors of degree 6, such that there is no triangular face containing u and two vertices of degree 5.

We dedicate a lemma to each of these configurations. In each of them, we did not succeed in finding a suitable monomial for the Nullstellensatz approach, hence we only present case analysis proofs.

Lemma 1.56. *The graph G does not contain C_{15a}.*

Proof. We use the notation depicted in Figure 1.60. By minimality, we color $G \setminus \{a, \ldots, n\}$ and uncolor u, v_1, v_2, v_3, w_2.

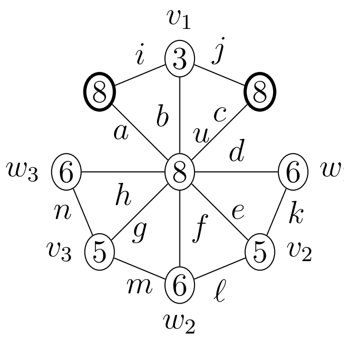


Figure 1.60 – Notation for Lemma 1.56

By Remark 1.12, we may assume that: $|\widehat{a}| = |\widehat{c}| = |\widehat{i}| = |\widehat{j}| = |\widehat{k}| = |\widehat{n}| = 3$, $|\widehat{w_2}| = 4$, $|\widehat{v_2}| = |\widehat{v_3}| = |\widehat{d}| = |\widehat{h}| = |\widehat{\ell}| = |\widehat{m}| = 5$, $|\widehat{u}| = 6$, $|\widehat{f}| = 7$, $|\widehat{v_1}| = |\widehat{e}| = |\widehat{g}| = 8$ and $|\widehat{b}| = 10$. We forget v_1 and consider two cases:

1. Assume that $\widehat{g} \neq \widehat{v_3} \cup \widehat{n}$, and color g with a color not in $\widehat{v_3} \cup \widehat{n}$. Then forget v_3, n, m. We then color a, c such that afterwards we have $\widehat{i} \neq \widehat{j}$ if $|\widehat{i}| = |\widehat{j}| = 2$. We can thus forget i and j (since after coloring every other element, either one of them has 2 choices, or both have one but not the same), then b.

   (a) If $\widehat{d} = \widehat{h}$, we color u, f, e with colors not in $\widehat{d}$, forget h, d and apply Lemma 1.22 on $\mathcal{T}(G)$ with the path $kv_2\ell w_2$.

   (b) Otherwise, if $|\widehat{u} \cup \widehat{d} \cup \widehat{h}| = 3$, we color f, e with a color not in this union, then color d with a color not in $\widehat{h}$, forget h and apply Lemma 1.22 on $\mathcal{T}(G)$ with the path $uv_2w_2\ell$.

   (c) Otherwise, if $|\widehat{u} \cup \widehat{d} \cup \widehat{f} \cup \widehat{h}| = 4$, we color e with a color not in this union, then d with a color not in $\widehat{h}$. If $\widehat{h} = \widehat{u}$, we color f with a color not in $\widehat{u}$, forget h and apply Lemma 1.22 on $\mathcal{T}(G)$ with the path $uv_2w_2\ell$. Otherwise, we color u with a color not in $\widehat{h}$, forget h and apply Lemma 1.22 on $\mathcal{T}(G)$ with the path $v_2\ell w_2 f$.

   (d) Otherwise, we color e with a color not in $\widehat{k}$ and color $\{u, d, f, h\}$ using Theorem 1.21. Then we apply Lemma 1.22 on $\mathcal{T}(G)$ with the path $kv_2\ell w_2$.

2. Otherwise, we can assume by symmetry that $\widehat{v_3} \cap \widehat{n} = \varnothing = \widehat{v_2} \cap \widehat{k}$. Then we can forget v_2, v_3, color g with a color not in $\widehat{m}$ and color a, c such that afterwards we have $\widehat{i} \neq \widehat{j}$ if $|\widehat{i}| = |\widehat{j}| = 2$. Then, we again forget i and j and we color $h, d, u, f, e, k, \ell, w_2, n, m, b$. □

Lemma 1.57. *The graph G does not contain C_{15b}.*

Proof. We use the notation depicted in Figure 1.61. By minimality, we color $G \setminus \{a, \ldots, o\}$ and uncolor $u, v_1, \ldots, v_4, w_1, w_2$.

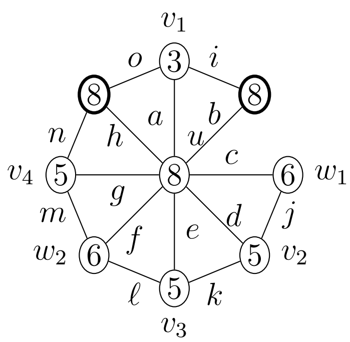


Figure 1.61 – Notation for Lemma 1.57

By Remark 1.12, we may assume that: $|\widehat{n}| = 2$, $|\widehat{b}| = |\widehat{i}| = 3$, $|\widehat{h}| = |\widehat{j}| = |\widehat{o}| = 4$, $|\widehat{\ell}| = |\widehat{m}| = 5$, $|\widehat{c}| = |\widehat{k}| = 6$, $|\widehat{f}| = 7$, $|\widehat{u}| = |\widehat{v_1}| = |\widehat{d}| = |\widehat{e}| = |\widehat{g}| = 8$ and $|\widehat{a}| = 10$. We may moreover assume that $|\widehat{w_1}| \geqslant 2$, $|\widehat{w_2}| \geqslant 4$, $|\widehat{v_4}| \geqslant 5$ and $|\widehat{v_2}|, |\widehat{v_3}| \geqslant 6$.

We forget v_1, then we remove from $\widehat{h}$ and $\widehat{n}$ a color $\alpha \in \widehat{o} \setminus \widehat{i}$. We then color n. Due to the choice of α, we may forget i, o since any coloring of the other elements gives either $|\widehat{o}| > 1$ or $\widehat{o} \neq \widehat{i}$, hence we can always color i then o. We may also forget a.

Note that v_4 has degree 5 hence it is adjacent (in G) to at most four uncolored vertices, hence we may assume that $|\widehat{v_4}| < 7$. We color g with a color not in $\widehat{v_4}$, then h, b. We then color f with a color not in $\widehat{m}$, then w_1, c, u, j, d, e. We forget v_4, m and color v_3, v_2, w_2, k, ℓ using Theorem 1.18 on the subgraph of $\mathcal{T}(G)$ they induce. □

Lemma 1.58. *The graph G does not contain C_{15c}.*

Proof. We use the notation depicted in Figure 1.62. By minimality, we color $G \setminus \{a, \ldots, p\}$ and uncolor $u, v_1, \ldots, v_4, w_1, w_2$.

By Remark 1.12, we may assume that: $|\widehat{j}| = |\widehat{o}| = 2$, $|\widehat{b}| = |\widehat{h}| = |\widehat{i}| = |\widehat{p}| = 4$, $|\widehat{k}| = |\widehat{\ell}| = |\widehat{m}| = |\widehat{n}| = 5$, $|\widehat{d}| = |\widehat{f}| = 7$, $|\widehat{v_1}| = |\widehat{c}| = |\widehat{e}| = |\widehat{g}| = |\widehat{u}| = 8$ and $|\widehat{a}| = 10$. We may also assume that $|\widehat{v_2}|, |\widehat{v_4}|$ are at least 5, $|\widehat{w_1}|, |\widehat{w_2}|$ are at least 4 and $|\widehat{v_3}|$ is at least 6.

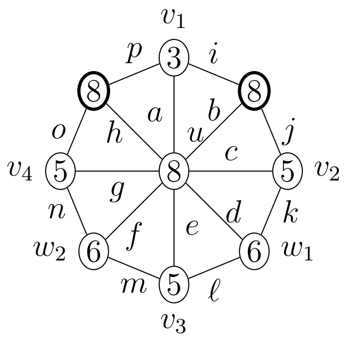


Figure 1.62 – Notation for Lemma 1.58

We forget v_1, color j and remove from $\widehat{h}$ and $\widehat{o}$ a color $\alpha \in \widehat{p} \setminus \widehat{i}$. Then we may forget i, p. Indeed, if we can color every element excepted i, p, then due to the choice of α, if $|\widehat{p}| = |\widehat{i}| = 1$, $\widehat{p} \neq \widehat{i}$, hence we can color them. We may then forget a. We also color o.

Note that there are only six uncolored vertices, hence w_1 has at most 5 uncolored neighbors in G. We thus have $|\widehat{w_1}| \leqslant 6$, hence we can color d with a color not in $\widehat{v_1}$. We then color h, b arbitrarily, and c with a color not in $\widehat{k}$. We color f, g, then u, v_4, w_2, n applying Theorem 1.18 on the subgraph of $\mathcal{T}(G)$ they induce, and then color e, m. We finally apply Lemma 1.22 on $\mathcal{T}(G)$ with the path $v_2 k w_1 \ell v_3$. □

Configuration C_{16}

To prove that G does not contain the configuration C_{16}, we prove that it does not contain any of the configuration below.

- C_{16a} is a 8-vertex u with a weak neighbor v of degree 3, and a $(7, 8)$-neighbor of degree 4 at triangle distance 2 from v.
- C_{16b} is a 8-vertex u with a weak neighbor v of degree 3 and a weak neighbor of degree 5 at triangle distance 2 from v, having two neighbors of degree 6.
- C_{16c} is a 8-vertex u with a weak neighbor v of degree 3, a $(7, 8)$-neighbor of degree 4 at triangle distance at least 3 from v, and two weak neighbors of degree 5.

We dedicate a lemma to each of these configurations.

Lemma 1.59. *The graph G does not contain C_{16a}.*

Proof. We use the notation depicted in Figure 1.63. By minimality, we take a coloring γ of $G \setminus \{a, b, c\}$, and uncolor d, e, f, g, v_1, v_2. We forget v_1, v_2.

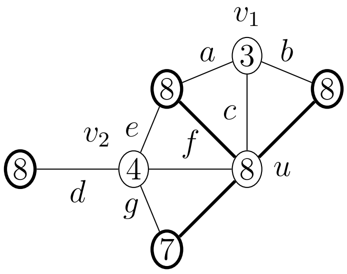


Figure 1.63 – Notation for Lemma 1.59

By Remark 1.12, we may thus assume that: $|\widehat{b}| = |\widehat{d}| = 2$ and $|\widehat{a}| = |\widehat{c}| = |\widehat{e}| = |\widehat{f}| = |\widehat{g}| = 3$.

- If $\widehat{d} \not\subset \widehat{g}$, then we can color d with a color not in $\widehat{g}$, forget g, and apply Lemma 1.20 to $\{b, a, e, f, c\}$. We may thus assume that $\widehat{d} \subset \widehat{g}$.
- If $\widehat{d} \not\subset \widehat{e}$, then we color d with a color not in $\widehat{e}$, color b, and apply Lemma 1.20 to $\{g, e, a, c, f\}$. We may thus assume (by symmetry) that $\widehat{d} \subset \widehat{e}$ and $\widehat{d} \subset \widehat{f}$.
- If $\widehat{f} \neq \widehat{g}$, we color g with a color not in $\widehat{f}$ (hence not in $\widehat{d}$). We color b an apply Lemma 1.20 to color $\{d, f, c, a, e\}$. Therefore, we may assume that $\widehat{f} = \widehat{g}$. By symmetry, we also have $\widehat{e} = \widehat{g}$.

Therefore, we have $|\widehat{d} \cup \widehat{e} \cup \widehat{f} \cup \widehat{g}| = 3$, hence $\{d, e, f, g\}$ is not colorable. This is impossible since γ is a proper coloring. This means that one of the previous cases should happen, hence that we can color G. □

Lemma 1.60. *The graph G does not contain C_{16b}.*

Proof. We use the notation depicted in Figure 1.64. By minimality, we take a coloring γ of $G \setminus \{a, b, c, v_1\}$, uncolor d, e, f, g, h, v_2 and forget v_1.

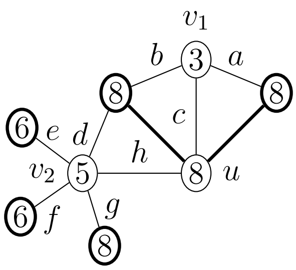


Figure 1.64 – Notation for Lemma 1.60

By Remark 1.12, we may thus assume that: $|\widehat{a}| = |\widehat{g}| = 2$, $|\widehat{b}| = |\widehat{c}| = |\widehat{d}| = |\widehat{h}| = 3$, $|\widehat{e}| = |\widehat{f}| = 4$ and $|\widehat{v_2}| = 5$.

If we color d, e, f, g, h, v_2 with their colors in γ, then the only way for the coloring not to extend to G is to have $\widehat{a}, \widehat{b}$ and $\widehat{c}$ to be the same list of size two. To avoid this, our goal is to find another coloring of d, e, f, g, h, v_2 which differs from γ on either d or h. We consider the color shifting graph H of $\{d, e, f, g, h, v_2\}$. By Lemma 1.16, there exists a strongly connected component C of H such that $|C| > \max_{x \in C} d^-(x)$. By Lemma 1.17, this inequality ensures that $|C| > 1$. We show that C contains either d or h by distinguishing some cases:

1. If C contains a vertex s_α, then $|C| > d^-(s_\alpha) = |V(H)| - 1$. Then $C = V(H)$ and contains d and h.

2. Otherwise, if C contains v_2, then $|C| \geqslant 5$ and C contains either d or h.

3. Otherwise, if C contains e or f, then $|C| \geqslant 4$ and C contains either d or h.

Therefore, C has size at least 2 and is contained in $\{d, g, h\}$, hence it contains d or h. Thus, we can apply Lemma 1.15 to ensure that we can recolor d or h, hence we can extend the coloring to G. □

Lemma 1.61. *The graph G does not contain C_{16c}.*

Proof. We use the notation depicted in Figure 1.65. By minimality, we take a coloring γ of $G \setminus \{b, c, n\}$, and uncolor $a, \ldots, q, v_1, v_2, v_3, v_4$. We forget v_1, v_3.

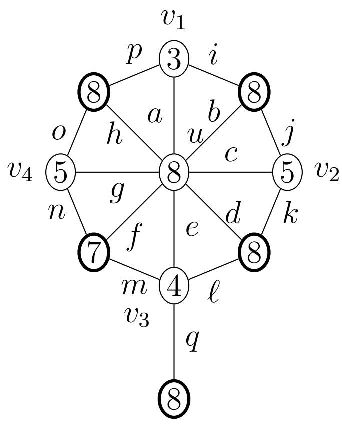


Figure 1.65 – Notation for Lemma 1.61

By Remark 1.12, we may thus assume that: $|\widehat{j}| = |\widehat{k}| = |\widehat{o}| = |\widehat{q}| = 2$, $|\widehat{n}| = 3$, $|\widehat{b}| = |\widehat{d}| = |\widehat{h}| = |\widehat{i}| = |\widehat{\ell}| = |\widehat{p}| = 4$, $|\widehat{f}| = |\widehat{m}| = 5$, $|\widehat{u}| = 6$, $|\widehat{c}| = |\widehat{g}| = 8$ and $|\widehat{a}| = |\widehat{e}| = 10$. Moreover, $\widehat{v_2}$ and $\widehat{v_4}$ have size 4 or 5 depending on whether $v_2v_4 \in E(G)$.

1. Assume that $\widehat{b} \cap \widehat{k} \neq \varnothing$. Then we color b and k with the same color, then color j. Let H be the graph induced by the remaining elements. We conclude using the Nullstellensatz: if $v_2v_4 \notin E(G)$, the coefficient of

   $$A^6C^5D^2E^7F^3G^6H^2IL^2M^4N^2OP^3QU^4V_2V_4^3$$

   in P_H is 1.

   Otherwise, we have $|\widehat{v_2}| = 3$, $|\widehat{v_4}| = 5$ and the coefficient of

   $$A^6C^5D^2E^7F^3G^6H^2IL^2M^4N^2OP^3QU^4V_2^2V_4^3$$

   in $(V_2 - V_4)P_H$ is 1. Using Theorem 1.14, we can find a coloring for H. Therefore, we may assume that $\widehat{b}$ and $\widehat{k}$ are disjoint.

2. Assume that $\widehat{d} \cap \widehat{j} \neq \varnothing$. We color d and j with the same color, then k and ℓ, q arbitrarily. Let H be the graph induced by the remaining elements. We conclude using the Nullstellensatz: if $v_2v_4 \notin E(G)$, the coefficient of

   $$A^7B^2C^5E^6F^3G^6H^2I^2M^2N^2OP^3U^3V_2V_4^3$$

   in P_H is -1.

   Otherwise, we have $|\widehat{v_2}| = 3$, $|\widehat{v_4}| = 5$ and the coefficient of

   $$A^7B^2C^5E^6F^3G^6H^2I^2M^2N^2OP^3U^3V_2^2V_4^3$$

   in $(V_4 - V_2)P_H$ is 1. Using Theorem 1.14, we can find a coloring for H. Therefore, we may assume that $\widehat{q}$ and $\widehat{d}$ are disjoint.

3. Assume that $\widehat{d} \cap \widehat{v_2} \neq \varnothing$. We color d and v_2 with the same color (which hence does not lie in $\widehat{j}$). Then we color k, j, ℓ, q. Let H be the graph induced by the remaining elements. We conclude using the Nullstellensatz: the coefficient of

   $$A^7BC^4E^6F^3G^6H^2I^2M^2N^2OP^3U^4V_4^3$$

   in P_H is -3. Using Theorem 1.14, we can find a coloring for H. Therefore, we may assume that $\widehat{d}$ and $\widehat{v_2}$ are disjoint.

4. Assume that $\widehat{b} \cap \widehat{v_2} \neq \varnothing$. We color b and v_2 with the same color (which hence does not lie in $\widehat{k}$ nor in $\widehat{d}$). Then we color j, k.

   - If $\widehat{o} \not\subset \widehat{p}$, we color o with a color not in $\widehat{p}$, then forget p, i, a. If $\widehat{i} \not\subset \widehat{p}$, we color i with a color not in $\widehat{p}$, then o, and we forget p, a. Finally, if $\widehat{o} \cap \widehat{i} \neq \varnothing$, we color o and i with the same color, then forget p, a. In the three cases, we end up with the same configuration. Let H be the graph induced by the remaining elements. We conclude using the Nullstellensatz: the coefficient of

$$C^4D^2E^8F^3G^5LM^4NQU^4V_4^2$$

in P_H is 1. Using Theorem 1.14, we can find a coloring for H. Therefore, we may assume that $\widehat{o}$ and $\widehat{i}$ are disjoint, and that their union is $\widehat{p}$.

- If $\widehat{h} \not\subset \widehat{p}$, we color h with a color not in $\widehat{p}$ (hence not in $\widehat{o}$), then forget p, i, a. If $\widehat{h} \cap \widehat{i} \not\emptyset$, we color h and i with the same color (hence not in $\widehat{o}$), then forget p, a.

  In both cases, we end up with the same configuration. Let H be the graph induced by the remaining elements. We conclude using the Nullstellensatz: the coefficient of

$$C^3DE^7F^2G^4LM^4N^2OQU^3V_4^3$$

  in P_H is 1. Using Theorem 1.14, we can find a coloring for H. Therefore, if $\widehat{h}$ and $\widehat{i}$ are not disjoint, we can reduce the configuration. Otherwise, since $|\widehat{h}| = 3$, $|\widehat{i}| = 2$ and $|\widehat{p}| = 4$, we have $\widehat{h} \not\subset \widehat{p}$, in which case we can also reduce the configuration.

Therefore, we may assume that $\widehat{b}$ and $\widehat{v_2}$ are disjoint.

5. If $\widehat{j} \not\subset \widehat{v_2}$ (resp. $\widehat{k} \not\subset \widehat{v_2}$), we color j (resp. k) with a color not in $\widehat{v_2}$. We forget v_2, color k (resp. j), then q, ℓ, d arbitrarily. We then end up with the same configuration as in 3., which is reducible. Therefore, we may assume that $\widehat{k}$ contains $\widehat{j}$ and $\widehat{v_2}$.

6. Now observe that $\gamma(b) \in \widehat{b}$. Since $\widehat{b}$ is disjoint from $\widehat{v_2}$, which contains $\widehat{j}$, we have $\gamma(b) \notin \widehat{j}$. Similarly, $\gamma(d) \notin \widehat{k}$. We color $b, d, f, g, h, n, o, u, v_4$ with their color in γ.

   - If $|\widehat{j} \cup \widehat{v_2} \cup \widehat{k}| = 3$, then we can color c with $\gamma(c)$, and this does not affect $\widehat{j}, \widehat{v_2}, \widehat{k}$, hence we can forget v_2. Afterwards, we have $|\widehat{q}| = 2$, $|\widehat{e}| = |\widehat{\ell}| = |\widehat{m}| = 3$.
     - If $\widehat{q} \not\subset \widehat{m}$, then we color q with a color not in $\widehat{m}$, forget m and color p. The remaining elements $\{e, i, j, k, \ell, n\}$ induce an even cycle, which is 2-choosable. We may thus assume that $\widehat{q} \subset \widehat{m}$.
     - If $\widehat{q} \not\subset \widehat{\ell}$, then we color p, and apply Lemma 1.20 to $\{m, e, a, i, j, k, \ell\}$. We may thus assume that $\widehat{q} \subset \widehat{\ell}$.
     - If $\widehat{m} \neq \widehat{e}$, we color e with a color not in $\widehat{m}$ (hence not in $\widehat{q}$). We then color p, i, j, k, ℓ, q, m. Therefore, we may assume that $\widehat{m} = \widehat{e}$.

– Since $\{\ell, q, m, e\}$ has a proper coloring (namely, γ), we know that $\widehat{e} \neq \widehat{\ell}$. We can thus color ℓ with a color not in $\widehat{e}$ (hence not in $\widehat{m}$ nor in $\widehat{q}$). Then, we color k, j, i, p, a, e, q, m.

Therefore, we may assume that $|\widehat{j} \cup \widehat{v_2} \cup \widehat{k}| \geqslant 4$.

- Assume that $|\widehat{\ell} \cup \widehat{q} \cup \widehat{m}| = 3$, then we can color e with $\gamma(e)$, and this does not affect $\widehat{\ell}, \widehat{q}, \widehat{m}$ hence we can forget m, q, ℓ.

  If $\widehat{j} = \widehat{k}$, since $|\widehat{j} \cup \widehat{v_2} \cup \widehat{k}| = 4$, we can remove the colors of $\widehat{j}$ from $\widehat{v_2}$, and $|\widehat{v_2}|$ has size at least 2. Therefore, we can forget v_2. We then color p and apply Lemma 1.20 to color $\{k, c, a, i, j\}$.

  Otherwise, we color c such that $\widehat{p}$ and $\widehat{a}$ are different if they have size 2. Since before we had $\widehat{j} \neq \widehat{k}$ and $|\widehat{j} \cup \widehat{v_2} \cup \widehat{k}| \geqslant 4$, we can then color $\{j, k, v_2\}$, then i and p, a (since $\widehat{p} \neq \widehat{a}$).

- Since $|\widehat{k}| = 2$, we can color k such that $|\widehat{\ell} \cup \widehat{q} \cup \widehat{m}| \geqslant 4$ afterwards. We then color j, v_2, c, i, p, a. Afterwards, we have $|\widehat{\ell} \cup \widehat{q} \cup \widehat{m}| \geqslant 3$, hence we can color $\{\ell, q, m\}$. □

Configuration C_{17}

To prove that G does not contain C_{17}, we prove that G cannot contain the following configurations.

- C_{17a} is a 8-vertex u with two weak neighbors of degree 3 and 4, and a $(6, 6)$-neighbor of degree 5.

- C_{17b} is a 8-vertex u with a weak neighbor v of degree 3, a weak neighbor of degree 4 at triangle distance at least 3 from v, and two weak neighbors of degree 5, one of them having a neighbor of degree 5.

- C_{17c} is a 8-vertex u with a weak neighbor v of degree 3, a weak neighbor of degree 4 at triangle distance 2 from v, and two $(6, 8)$-neighbors of degree 5.

- C_{17d} is a 8-vertex u with a weak neighbor v of degree 3, a weak neighbor of degree 4 at triangle distance 2 from v, and two $(7^+, 8)$-neighbors of degree 5 such that one of them is a triangle-distance 2 from v and has a neighbor of degree 5.

- C_{17e} is a 8-vertex u with a weak neighbor v of degree 3, a weak neighbor of degree 4 at triangle distance 2 from v, and two $(7^+, 8)$-neighbors of degree 5 such that one of them is a triangle-distance at least 3 from v and has a neighbor of degree 5.

- C_{17f} is a 8-vertex u with a weak neighbor v of degree 3, a weak neighbor of degree 4 at triangle distance 2 from v, and two $(7^+, 8)$-neighbors of degree 5 such that one of them is a triangle-distance at least 3 from v and has two neighbors of degree 6.

We dedicate a lemma to each of these configurations.

Lemma 1.62. *The graph G does not contain C_{17a}.*

Proof. We use the notation depicted in Figure 1.66. By minimality, we color $G \setminus \{a, \ldots, n, v_1, \ldots, v_5\}$. We forget v_1, v_2.

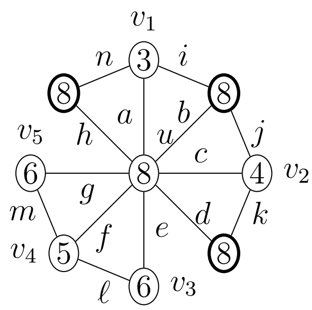


Figure 1.66 – Notation for Lemma 1.62

By Remark 1.12, we may thus assume that: $|\widehat{v_3}| = |\widehat{v_5}| = |\widehat{k}| = 2$, $|\widehat{d}| = |\widehat{j}| = |\widehat{h}| = |\widehat{n}| = 3$, $|\widehat{b}| = |\widehat{i}| = |\widehat{\ell}| = |\widehat{m}| = 4$, $|\widehat{v_4}| = |\widehat{e}| = |\widehat{g}| = 6$, $|\widehat{u}| = 7$, $|\widehat{f}| = 8$, $|\widehat{c}| = 9$ and $|\widehat{a}| = 10$.

We color j and d with colors not in $\widehat{k}$ and forget k.

- If $\widehat{b} = \widehat{h}$, we remove the colors of $\widehat{b}$ from $\widehat{a}, \widehat{c}, \widehat{e}, \widehat{f}, \widehat{g}$ and $\widehat{u}$.

  We then color $\{u, v_3, v_4, v_5, e, f, g, \ell, m\}$ as done in Lemma 1.30. We then color c and a. The remaining elements $\{h, b, i, n\}$ induce an even cycle, which is 2-choosable.

- Otherwise, we color h with a color not in $\widehat{b}$, then i with a color not in $\widehat{n}$ and forget n, a, c. We color b and we again come back to the case of Lemma 1.30. □

Lemma 1.63. *The graph G does not contain C_{17b}.*

Proof. We use the notation depicted in Figure 1.67. By minimality, we color $G \setminus \{a, p, i\}$, and uncolor $a, \ldots, s, v_1, \ldots, v_5$. We forget v_1, v_3.

By Remark 1.12, we may thus assume that: $|\widehat{n}| = |\widehat{o}| = |\widehat{r}| = |\widehat{s}| = |\widehat{v_5}| = 2$, $|\widehat{v_4}| = |\widehat{b}| = |\widehat{d}| = |\widehat{f}| = |\widehat{h}| = |\widehat{i}| = |\widehat{j}| = |\widehat{k}| = |\widehat{\ell}| = |\widehat{m}| = |\widehat{p}| = 4$, $|\widehat{q}| = |\widehat{u}| = 6$, $|\widehat{v_2}| = 7$, $|\widehat{g}| = 8$ and $|\widehat{a}| = |\widehat{c}| = |\widehat{e}| = 10$.

For all items except the last one, we remove from $\widehat{v_2}$ the colors from $\widehat{v_5}$, so that $\widehat{v_5}$ becomes disjoint from $\widehat{v_2}$, so we can forget v_5 then q.

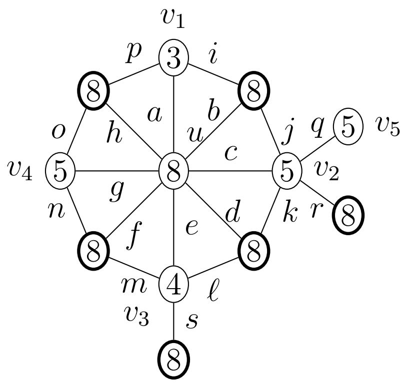


Figure 1.67 – Notation for Lemma 1.63

- If $\widehat{h} \cap \widehat{n} \neq \varnothing$, we color h and n with the same color, then o, r, s. Let H be the graph induced by the remaining elements. We conclude using the Nullstellensatz: the coefficient of

$$A^8B^2C^7D^2E^7F^2G^5I^3J^2K^2L^2U^4V_2^3V_4$$

in P_H is -3. Using Theorem 1.14, we can find a coloring for H. Therefore, we may assume that $\widehat{h}$ and $\widehat{n}$ are disjoint.

- If $\widehat{h} \cap \widehat{v_4} \neq \varnothing$, we color h and v_4 with the same color (hence not in $\widehat{n}$), then o, n, r, s. Let H be the graph induced by the remaining elements. We conclude using the Nullstellensatz: the coefficient of

$$A^8B^2C^7D^2E^7FG^4I^3J^2K^2L^2MU^4V_2^3$$

in P_H is -8. Using Theorem 1.14, we can find a coloring for H. Therefore, we may assume that $\widehat{h}$ and $\widehat{v_4}$ are disjoint.

- If $\widehat{f} \cap \widehat{o} \neq \varnothing$, we color f and o with the same color, then n, m, s, r. Let H be the graph induced by the remaining elements. We conclude using the Nullstellensatz: the coefficient of

$$A^8B^2C^7D^2E^6G^5H^2I^3J^2K^2LU^4V_2^3V_4$$

in P_H is -3. Using Theorem 1.14, we can find a coloring for H. Therefore, we may assume that $\widehat{f}$ and $\widehat{o}$ are disjoint.

- If $\widehat{f} \cap \widehat{v_4} \neq \varnothing$, we color f and v_4 with the same color (hence not in $\widehat{o}$), then n, o, m, s, r. Let H be the graph induced by the remaining elements. We conclude using the Nullstellensatz: the coefficient of

$$A^8B^2C^7D^2E^6G^4HI^3J^2K^2LPU^4V_2^3$$

in P_H is -4. Using Theorem 1.14, we can find a coloring for H. Therefore, we may assume that $\widehat{f}$ and $\widehat{v_4}$ are disjoint.

- If $\widehat{o} \cup \widehat{n} \not\subset \widehat{v_4}$, then we color n or o with a color not in $\widehat{v_4}$, then o or n, then r, s, and we forget v_4.

  Let H be the graph induced by the remaining elements. We conclude using the Nullstellensatz: the coefficient of
  $$A^9B^3C^8D^3E^8F^2G^5HI^3J^2K^2L^2MU^5V_2^3$$
  in P_H is -8. Using Theorem 1.14, we can find a coloring for H. Therefore, we may assume that $\widehat{n}$ and $\widehat{o}$ are contained in v_4.

- If $\widehat{n} \neq \widehat{o}$, we color n with a color not in $\widehat{o}$, then f, m, s, r. We remove $\widehat{v_5}$ from $\widehat{q}$, so that we can forget v_5 and v_2.

  Let H be the graph induced by the remaining elements. We conclude using the Nullstellensatz: the coefficient of
  $$A^8B^2C^7D^2E^6G^5H^2I^3J^2K^2LOP^2Q^2U^4V_4^2$$
  in P_H is -1. Using Theorem 1.14, we can find a coloring for H. Therefore, we may assume that $\widehat{n} = \widehat{o}$.

Now we have $\gamma(h) \in \widehat{h}$, hence not in $\widehat{o}$ since $\widehat{o} \subset \widehat{v_4}$ which is disjoint from $\widehat{h}$. Therefore, $\gamma(h) \notin \widehat{o}$, and similarly, $\gamma(f) \notin \widehat{n}$.

We now color h, b, f and d with their color in γ. Since $\widehat{n} = \widehat{o}$ and $\{g, n, o, v_4\}$ is colorable, coloring g and v_4 with their color in γ does not affect $\widehat{n}$ and $\widehat{o}$. We also color u with its color in γ.

We remove $\widehat{v_5}$ from $\widehat{q}$, so that $\widehat{v_5}$ becomes disjoint from $\widehat{q}$, hence we can forget v_5 and v_3.

Therefore, we obtain $|\widehat{j}| = |\widehat{k}| = |\widehat{r}| = |\widehat{s}| = 2$, $|\widehat{i}| = |\widehat{j}| = |\widehat{k}| = |\widehat{\ell}| = |\widehat{m}| = |\widehat{p}| = 3$, $|\widehat{a}| = |\widehat{c}| = |\widehat{e}| = |\widehat{q}| = 4$. Moreover, $\widehat{j} = \widehat{k}$.

Observe that if we color everything but a, i, p with their color in γ, the only problematic case is when $\widehat{a}, \widehat{i}$ and $\widehat{p}$ are the same list of size 2. Observe then that any recoloring of j or m can break this condition.

Let $\alpha \in \widehat{e} \neq \widehat{m}$. We distinguish two cases.

- Assume that $\{c, j, k, q, r\}$ stays colorable when we remove α from $\widehat{c}$. If $\widehat{\ell} \neq \widehat{s} \cup \{\gamma(k)\}$, then we color c, j, k, q, r with their color from γ, so that $\widehat{e} \neq \widehat{m}$ and $\widehat{\ell} \neq \widehat{s}$. We then color ℓ with a color not in $\widehat{s}$, then i arbitrarily. We then apply Lemma 1.20 to $\{s, m, n, o, p, a, e\}$ since $\widehat{e} \neq \widehat{m}$.

  Let H be the color shifting graph of $\{c, j, k, q, r\}$. By Lemma 1.16, there exists a strongly connected component C of H stable by predecessor. By Lemma 1.17, this ensures that $|C| > d^-(r) = 1$.

  - If C contains j, then we can recolor j by Lemma 1.15, which now breaks $\widehat{a} = \widehat{i} = \widehat{p}$ after having colored every other element. Thus we may assume that C does not contain j.

– If C contains k, then we can recolor k by Lemma 1.15, and the condition $\widehat{\ell} = \widehat{s} \cup \{\gamma(k)\}$ does not hold anymore with the new coloring. Thus we may assume that C does not contain k.
– If C contains some s_β, then it contains j and k.
– Otherwise, $C \subset \{c, q, r\}$. If $q \in C$, then $|C| > 3$, which is not possible.
– Otherwise $C \subset \{c, r\}$, hence $c \in C$ and $|C| > 2$, which is again impossible.

Therefore, we may always recolor either j or k, and then extend the coloring to G.

- Assume that $\{c, j, k, q, r\}$ is not colorable when we remove α from $\widehat{c}$. This means that $\gamma(c) = \alpha$. In particular, when coloring $\{c, j, k, q, r\}$ with their color in γ, we obtain that $\widehat{e}$ and $\widehat{m}$ are the same list of size 3. Since $\{e, \ell, m, s\}$ is colorable, there must exist a color in $\widehat{\ell} \cup \widehat{s}$ not in $\widehat{m}$. We color ℓ or s (say ℓ, by symmetry) with this color, then s. We then apply Lemma 1.20 to $\{i, p, o, n, m, e, a\}$. □

Lemma 1.64. *The graph G does not contain C_{17c}.*

Proof. We use the notation depicted in Figure 1.68. By minimality, we color $G \setminus \{a, \dots, q, v_1, \dots, v_5\}$. We forget v_1, v_5.

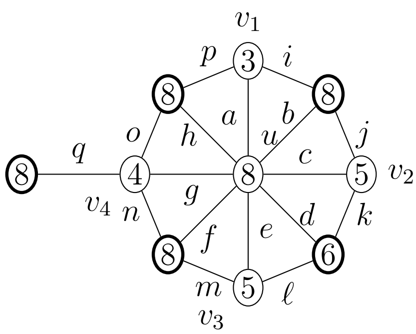


Figure 1.68 – Notation for Lemma 1.64

By Remark 1.12, we may thus assume that: $|\widehat{j}| = |\widehat{m}| = |\widehat{q}| = 2$, $|\widehat{v_3}| = |\widehat{b}| = |\widehat{f}| = |\widehat{h}| = |\widehat{i}| = |\widehat{n}| = |\widehat{o}| = |\widehat{p}| = 4$, $|\widehat{k}| = |\widehat{\ell}| = 5$, $|\widehat{u}| = |\widehat{d}| = 7$, $|\widehat{c}| = |\widehat{e}| = 8$ and $|\widehat{a}| = |\widehat{g}| = 10$. Moreover, $\widehat{v_2}$ and $\widehat{v_4}$ have size 5 or 6 depending on whether $v_2v_4 \in E(G)$.

We color i with a color not in $\widehat{j}$, then o with a color not in $\widehat{p}$, then q, then b with a color not in $\widehat{j}$, then h, f, m, n, v_4.

Let H be the graph induced by the remaining elements. The monomial

$$C^4 D^3 E^2 G^3 J K^4 L^2 U V_2^4 V_3^2$$

has coefficient 1 in P_H. By Theorem 1.14, we can find a coloring for H, and hence color G. $\square$

Lemma 1.65. *The graph G does not contain C_{17d}.*

Proof. We follow here the same approach as for C_{17b}. We use the notation depicted in Figure 1.69. By minimality, we color $G \setminus \{a, p, i\}$, and uncolor $a, \dots, s, v_1, \dots, v_5$. We forget v_1, v_4.

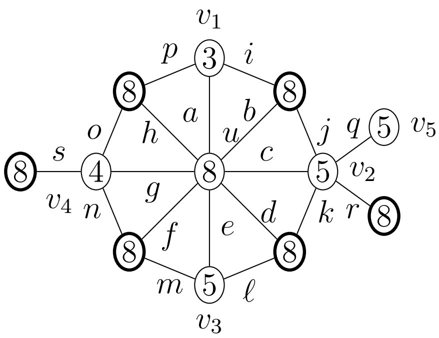


Figure 1.69 – Notation for Lemma 1.65

By Remark 1.12, we may thus assume that: $|\widehat{\ell}| = |\widehat{m}| = |\widehat{r}| = |\widehat{s}| = |\widehat{v_5}| = 2$, $|\widehat{v_3}| = |\widehat{b}| = |\widehat{d}| = |\widehat{f}| = |\widehat{h}| = |\widehat{i}| = |\widehat{j}| = |\widehat{k}| = |\widehat{n}| = |\widehat{o}| = |\widehat{p}| = 4$, $|\widehat{q}| = |\widehat{u}| = 6$, $|\widehat{v_2}| = 7$, $|\widehat{e}| = 8$ and $|\widehat{a}| = |\widehat{c}| = |\widehat{g}| = 10$.

For all items except the last two, we remove from $\widehat{v_2}$ the colors from $\widehat{v_5}$, so that $\widehat{v_5}$ becomes disjoint from $\widehat{v_2}$, so we can forget v_5 then q.

- If $\widehat{f} \cap \widehat{\ell} \neq \varnothing$, we color f and ℓ with the same color, then m, n, s, r. Let H be the graph induced by the remaining elements. We conclude using the Nullstellensatz: the coefficient of
$$A^8 B^2 C^7 D^2 E^5 G^6 H^2 I^3 J^2 K P^2 U^4 V_2^3 V_3$$
in P_H is -1. Using Theorem 1.14, we can find a coloring for H. Therefore, we may assume that $\widehat{f}$ and $\widehat{\ell}$ are disjoint.

- If $\widehat{d} \cap \widehat{m} \neq \varnothing$, we color d and m with the same color, then ℓ, k, r, s. Let H be the graph induced by the remaining elements. We conclude using the Nullstellensatz: the coefficient of
$$A^8 B^2 C^6 E^5 F^2 G^7 H^2 I^3 J N O P^2 U^4 V_2^2 V_3$$
in P_H is -2. Using Theorem 1.14, we can find a coloring for H. Therefore, we may assume that $\widehat{d}$ and $\widehat{m}$ are disjoint.

- If $\widehat{f} \cap \widehat{v_3} \neq \varnothing$, we color f and v_3 with the same color (hence not in $\widehat{\ell}$), then m, ℓ, n, s, r. Let H be the graph induced by the remaining elements. We conclude using the Nullstellensatz: the coefficient of

$$A^8 B^2 C^7 D E^4 G^6 H^2 I^3 J^2 K O P^2 U^4 V_2^3$$

  in P_H is -1. Using Theorem 1.14, we can find a coloring for H. Therefore, we may assume that $\widehat{f}$ and $\widehat{v_3}$ are disjoint.

- If $\widehat{d} \cap \widehat{v_3} \neq \varnothing$, we color d and v_3 with the same color (hence not in $\widehat{m}$), then ℓ, m, k, r, s. Let H be the graph induced by the remaining elements. We conclude using the Nullstellensatz: the coefficient of

$$A^8 B^2 C^6 E^4 F G^7 H^2 I^3 J N O^2 P^2 U^4 V_2^2$$

  in P_H is -2. Using Theorem 1.14, we can find a coloring for H. Therefore, we may assume that $\widehat{d}$ and $\widehat{v_3}$ are disjoint.

- If $\widehat{m} \neq \widehat{\ell}$, we remove the colors of $\widehat{v_5}$ from $\widehat{q}$, so that we can forget v_2 and v_5. We then color m with a color not in $\widehat{\ell}$, then f, n, s, r. We conclude using the Nullstellensatz: the coefficient of

$$A^8 B^2 C^7 D^2 E^5 G^6 H^2 I^3 J^2 K^2 L O P^2 Q^2 U^4 V_3^2$$

  in P_H is -1. Using Theorem 1.14, we can find a coloring for H. Therefore, we may assume that $\widehat{m} = \widehat{\ell}$.

- We remove a color $\alpha \in \widehat{v_2} \setminus \widehat{q}$ from $\widehat{u}$, so that if everything is colored except v_2, v_5, q, we obtain $\widehat{v_2} \neq \widehat{q}$. Therefore, we can forget v_2, v_5, q. The configuration is now symmetric (vertically).

  Assume that $\widehat{m} \not\subset \widehat{v_3}$ so that there exists $\alpha \in \widehat{m} \setminus \widehat{v_3}$. We color m with α and forget v_3. Since $\widehat{m} = \widehat{\ell}$, we have $\alpha \in \widehat{\ell}$, hence $\alpha \notin \widehat{f}$. In this case, we color ℓ, then f with a color not in $\widehat{d}$, then n, s, r.

  We conclude using the Nullstellensatz: the coefficient of

$$A^8 B^2 C^7 D^2 E^4 G^6 H^2 I^3 J^2 K P^2 U^3$$

  in P_H is -1. Using Theorem 1.14, we can find a coloring for H. Therefore, we may assume that $\widehat{m} \subset \widehat{v_3}$ and then $\widehat{\ell} \subset \widehat{v_3}$ since $\widehat{m} = \widehat{\ell}$.

Now we have $\gamma(f) \in \widehat{f}$, hence not in $\widehat{m}$ since $\widehat{m} \subset \widehat{v_3}$ which is disjoint from $\widehat{f}$. Therefore, $\gamma(f) \notin \widehat{m}$, and similarly, $\gamma(d) \notin \widehat{\ell}$.

We now color h, b, f and d with their color in γ. Since $\widehat{\ell} = \widehat{m}$ and $\{e, \ell, m, v_3\}$ is colorable, coloring e and v_3 with their color in γ does not affect $\widehat{\ell}$ and $\widehat{m}$. We also color u with its color in γ.

We remove $\widehat{v_5}$ from $\widehat{q}$, so that $\widehat{v_5}$ becomes disjoint from $\widehat{q}$, hence we can forget v_5 and v_3.

Therefore, we obtain $|\widehat{\ell}| = |\widehat{m}| = |\widehat{r}| = |\widehat{s}| = 2$, $|\widehat{i}| = |\widehat{j}| = |\widehat{k}| = |\widehat{n}| = |\widehat{o}| = |\widehat{p}| = 3$, $|\widehat{a}| = |\widehat{c}| = |\widehat{g}| = |\widehat{q}| = 4$. Moreover, $\widehat{\ell} = \widehat{m}$.

Observe that if we color everything but a, i, p with their color in γ, the only problematic case is when $\widehat{a}, \widehat{i}$ and $\widehat{p}$ are the same list of size 2. Then, any recoloring of j or o can break this condition.

Note that since $\widehat{\ell} = \widehat{m}$, it is sufficient to color the graph obtained by identifying the endpoints of ℓ and m that are not v_3 (so that k and n become incident), and by removing ℓ and m.

Let $\alpha \in \widehat{g} \neq \widehat{o}$. We distinguish two cases.

- Assume that $\{c, j, k, q, r\}$ stays colorable when we remove α from $\widehat{c}$. If $\widehat{n} \neq \widehat{s} \cup \{\gamma(k)\}$, then we color c, j, k, q, r, ℓ, m, so that $\widehat{n} \neq \widehat{s}$ and $\widehat{o} \neq \widehat{g}$. We then color n with a color not in $\widehat{s}$, then color i arbitrarily. We then apply Lemma 1.20 to $\{s, o, p, a, g\}$ since $\widehat{o} \neq \widehat{g}$.

  Let H be the color shifting graph of $\{c, j, k, q, r\}$. By Lemma 1.16, there exists a strongly connected component C of H stable by predecessor. By Lemma 1.17, this ensures that $|C| > d^-(r) = 1$.

  - If C contains j, then we can recolor j by Lemma 1.15, which now breaks $\widehat{a} = \widehat{i} = \widehat{p}$ after having colored every other element. Thus we may assume that C does not contain j.
  - If C contains k, then we can recolor k by Lemma 1.15, and the condition $\widehat{n} = \widehat{s} \cup \{\gamma(k)\}$ does not hold anymore with the new coloring. Thus we may assume that C does not contain k.
  - If C contains some s_β, then it contains j and k.
  - Otherwise, $C \subset \{c, q, r\}$. If $q \in C$, then $|C| > 3$, which is not possible.
  - Otherwise $C \subset \{c, r\}$, hence $c \in C$ and $|C| > 2$, which is again impossible.

  Therefore, we may always recolor either j or k, and then extend the coloring to G.

- Assume that $\{c, j, k, q, r\}$ is not colorable when we remove α from $\widehat{c}$. This means that $\gamma(c) = \alpha$. In particular, when coloring $\{c, j, k, q, r\}$ with their color in γ, we obtain that $\widehat{o}$ and $\widehat{g}$ are the same list of size 3. Since $\{g, n, o, s\}$ is colorable, there must exist a color in $\widehat{n} \cup \widehat{s}$ not in $\widehat{o}$. We color n or s (say n, by symmetry) with this color, then s. We then apply Lemma 1.20 to $\{i, p, o, g, a\}$. □

Lemma 1.66. *The graph G does not contain C_{17e}.*

Proof. We follow here the same approach as for C_{17b}. We use the notation depicted in Figure 1.70. By minimality, there exists a coloring γ of $G\setminus\{a,p,i,v_1\}$. We uncolor $a,\dots,s,v_1,\dots,v_5$ and forget v_1,v_4.

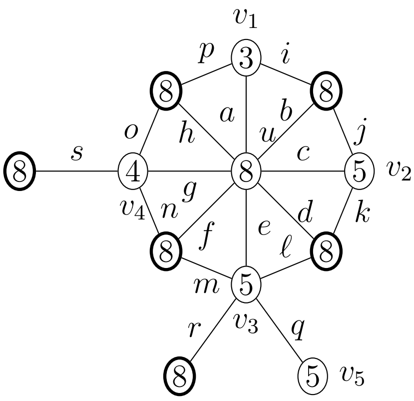


Figure 1.70 – Notation for Lemma 1.66

By Remark 1.12, we may thus assume that: $|\widehat{j}| = |\widehat{k}| = |\widehat{r}| = |\widehat{s}| = |\widehat{v_5}| = 2$, $|\widehat{b}| = |\widehat{d}| = |\widehat{f}| = |\widehat{g}| = |\widehat{i}| = |\widehat{\ell}| = |\widehat{m}| = |\widehat{n}| = |\widehat{o}| = |\widehat{p}| = |\widehat{v_2}| = 4$, $|\widehat{q}| = |\widehat{u}| = 6$, $|\widehat{v_3}| = 7$, $|\widehat{c}| = 8$ and $|\widehat{a}| = |\widehat{e}| = |\widehat{g}| = 10$.

For all items except the last one, we remove from $\widehat{u}$ a color in $\widehat{v_3}\setminus\widehat{q}$, so that if we color everything but q,v_3,v_5, then $\widehat{q} \neq \widehat{v_3}$ if they are lists of size 2. This means that we can forget about q,v_3,v_5.

- If $\widehat{b}\cap\widehat{k} \neq \varnothing$, we color b and k with the same color, then j,r,s. Let H be the graph induced by the remaining elements. We conclude using the Nullstellensatz: the coefficient of

$$A^8C^5DE^7F^2G^7H^2LM^2N^2O^2P^3U^3V_2$$

  in P_H is 8. Using Theorem 1.14, we can find a coloring for H. Therefore, we may assume that $\widehat{b}$ and $\widehat{k}$ are disjoint.

- If $\widehat{d}\cap\widehat{j} \neq \varnothing$, we color d and j with the same color, then k,ℓ,r,s. Let H be the graph induced by the remaining elements. We conclude using the Nullstellensatz: the coefficient of

$$A^8B^2C^5E^6F^2G^7H^2MN^2O^2P^3U^3V_2$$

  in P_H is -4. Using Theorem 1.14, we can find a coloring for H. Therefore, we may assume that $\widehat{d}$ and $\widehat{j}$ are disjoint.

- If $\widehat{d}\cap\widehat{v_2} \neq \varnothing$, we color d and v_2 with the same color (hence not in $\widehat{j}$), then k,j,ℓ,r,s. Let H be the graph induced by the remaining elements. We conclude using the Nullstellensatz: the coefficient of

$$A^8BC^4E^6F^2G^7H^2IMN^2O^2P^3U^3$$

in P_H is -4. Using Theorem 1.14, we can find a coloring for H. Therefore, we may assume that $\widehat{d}$ and $\widehat{v_2}$ are disjoint.

- If $\widehat{b} \cap \widehat{v_2} \neq \varnothing$, we color b and v_2 with the same color (hence not in $\widehat{k}$), then j, k, r, s. Let H be the graph induced by the remaining elements. We conclude using the Nullstellensatz: the coefficient of

$$A^8C^4DE^7F^2G^7H^2LM^2N^2O^2P^3U^3$$

in P_H is -8. Using Theorem 1.14, we can find a coloring for H. Therefore, we may assume that $\widehat{b}$ and $\widehat{v_2}$ are disjoint.

- If $\widehat{k} \not\subset \widehat{v_2}$ or $\widehat{j} \not\subset \widehat{v_2}$, we color k (or j) with a color not in $\widehat{v_2}$, then j (or k), r, s, then forget v_2.

  Let H be the graph induced by the remaining elements. We conclude using the Nullstellensatz: the coefficient of

$$A^9BC^5D^2E^8F^3G^8H^3LM^2N^2O^2P^3U^4$$

in P_H is -8. Using Theorem 1.14, we can find a coloring for H. Therefore, we may assume that $\widehat{k}$ and $\widehat{j}$ are included in $\widehat{v_2}$.

- If $\widehat{j} \neq \widehat{k}$, we color j with a color not in $\widehat{k}$, then i, b and s, r. We remove $\widehat{v_5}$ from $\widehat{q}$, so that $\widehat{v_5}$ becomes disjoint from $\widehat{q}$, hence we can forget v_5 and v_3.

  Let H be the graph induced by the remaining elements. We conclude using the Nullstellensatz: the coefficient of

$$A^7C^5D^2E^7F^2G^7H^2KL^2M^2N^2O^2P^2Q^2U^4V_2^2$$

in P_H is -6. Using Theorem 1.14, we can find a coloring for H. Therefore, we may assume that $\widehat{k} = \widehat{j}$.

Now we have $\gamma(b) \in \widehat{b}$, hence not in $\widehat{j}$ since $\widehat{j} \subset \widehat{v_2}$ which is disjoint from $\widehat{b}$. Therefore, $\gamma(b) \notin \widehat{j}$, and similarly, $\gamma(d) \notin \widehat{k}$.

We now color h, b, f and d with their color in γ. Since $\widehat{j} = \widehat{k}$ and $\{j, k, c, v_2\}$ is colorable, coloring c and v_2 with their color in γ does not affect $\widehat{j}$ and $\widehat{k}$. We also color u with its color in γ.

We remove $\widehat{v_5}$ from $\widehat{q}$, so that $\widehat{v_5}$ becomes disjoint from $\widehat{q}$, hence we can forget v_5 and v_3.

Therefore, we obtain $|\widehat{j}| = |\widehat{k}| = |\widehat{r}| = |\widehat{s}| = 2$, $|\widehat{o}| = |\widehat{p}| = |\widehat{i}| = |\widehat{\ell}| = |\widehat{m}| = |\widehat{n}| = 3$, $|\widehat{a}| = |\widehat{e}| = |\widehat{g}| = |\widehat{q}| = 4$. Moreover, $\widehat{j} = \widehat{k}$.

Observe that if we color everything but a, i, p with their color in γ, the only problematic case is when $\widehat{a}, \widehat{i}$ and $\widehat{p}$ are the same list of size 2. Then, any recoloring of ℓ or o can break this condition.

Let $\alpha \in \widehat{g} \neq \widehat{o}$. We distinguish two cases.

- Assume that $\{e, \ell, m, q, r\}$ stays colorable when we remove α from $\widehat{e}$. If $\widehat{n} \neq \widehat{s} \cup \{\gamma(m)\}$, then we color ℓ, m, q, r, ℓ, k, so that $\widehat{n} \neq \widehat{s}$ and $\widehat{o} \neq \widehat{g}$. We then color n with a color not in $\widehat{s}$, then color i arbitrarily. We then apply Lemma 1.20 to $\{s, o, p, a, g\}$ since $\widehat{o} \neq \widehat{g}$.

  Let H be the color shifting graph of $\{e, \ell, m, q, r\}$. By Lemma 1.16, there exists a strongly connected component C of H stable by predecessor. By Lemma 1.17, this ensures that $|C| > d^-(r) = 1$.

  - If C contains ℓ, then we can recolor ℓ by Lemma 1.15, which now breaks $\widehat{a} = \widehat{i} = \widehat{p}$ after having colored every other element. Thus we may assume that C does not contain ℓ.
  - If C contains m, then we can recolor m by Lemma 1.15, and the condition $\widehat{n} = \widehat{s} \cup \{\gamma(m)\}$ does not hold anymore with the new coloring. Thus we may assume that C does not contain m.
  - If C contains some s_β, then it contains m and ℓ.
  - Otherwise, $C \subset \{e, q, r\}$. If $q \in C$, then $|C| > 3$, which is not possible.
  - Otherwise $C \subset \{e, r\}$, hence $e \in C$ and $|C| > 2$, which is again impossible.

  Therefore, we may always recolor either ℓ or m, and then extend the coloring to G.

- Assume that $\{e, \ell, m, q, r\}$ is not colorable when we remove α from $\widehat{e}$. This means that $\gamma(e) = \alpha$. In particular, when coloring $\{e, \ell, m, q, r\}$ with their color in γ, we obtain that $\widehat{o}$ and $\widehat{g}$ are the same list of size 3. Since $\{g, n, o, s\}$ is colorable, there must exist a color in $\widehat{n} \cup \widehat{s}$ not in $\widehat{o}$. We color n or s (say n, by symmetry) with this color, then s. We then apply Lemma 1.20 to $\{i, p, o, g, a\}$. □

Lemma 1.67. *The graph G does not contain C_{17f}.*

Proof. We use the notation depicted in Figure 1.71. By minimality, there exists a coloring γ of $G \setminus \{a, i, p, v_1\}$.

We uncolor $a, \ldots, s, v_1, v_2, v_3, v_4$ and forget v_1, v_4. By Remark 1.12, we may thus assume that: $|\widehat{s}| = |\widehat{j}| = |\widehat{k}| = 2$, $|\widehat{h}| = |\widehat{b}| = |\widehat{d}| = |\widehat{f}| = |\widehat{o}| = |\widehat{p}| = |\widehat{i}| = |\widehat{\ell}| = |\widehat{m}| = |\widehat{n}| = |\widehat{q}| = |\widehat{r}| = |\widehat{v_2}| = 4$, $|\widehat{v_3}| = |\widehat{u}| = 6$, $|\widehat{c}| = 8$, $|\widehat{a}| = |\widehat{g}| = |\widehat{e}| = 10$.

We first prove that we can color G unless $\widehat{n} = \widehat{s} \cup \{\gamma(f), \gamma(m)\}$. Indeed, otherwise, we color every element but $\{o, p, i, n, a, g, s\}$, and we obtain $|\widehat{s}| = |\widehat{i}| = 2$, $|\widehat{o}| = |\widehat{p}| = |\widehat{a}| = |\widehat{g}| = 3$, and either $|\widehat{n}| = 3$ or $|\widehat{n}| = 2$ and $\widehat{n} \neq \widehat{s}$. We focus on the last case (since we may always remove a color from $\widehat{n}$ in the first case to obtain the second one).

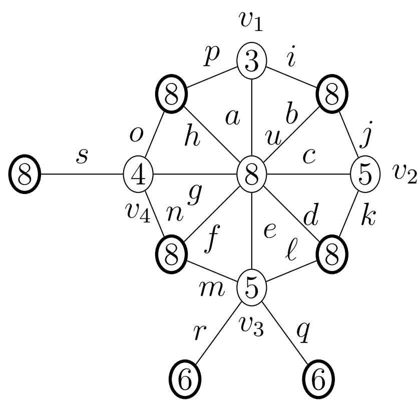


Figure 1.71 – Notation for Lemma 1.67

- If $\widehat{o} \neq \widehat{g}$, we color n with a color not in $\widehat{s}$, then i, and apply Lemma 1.20 to color $\{s, o, p, a, g\}$. Thus we may assume that $\widehat{o} = \widehat{g}$.

- Since $\{o, g, n, s\}$ is colorable, we cannot have both $\widehat{s} \subset \widehat{o}$ and $\widehat{n} \subset \widehat{o}$. By symmetry, assume that we can color s with a color not in $\widehat{o}$ (hence not in $\widehat{g}$). Then we color n and apply Lemma 1.20 to color $\{i, p, o, g, a\}$.

We uncolor the elements of $S = \{m, e, \ell, q, r, v_3\}$. Let H be the color shifting graph of S. By Lemma 1.16, there exists a strongly connected component C of H such that $|C| > \max_{x \in C} d^-(x)$. By Lemma 1.17, this inequality ensures that $|C| > 1$.

- If C contains m, then we can recolor m by Lemma 1.15, and the condition $\widehat{n} = \widehat{s} \cup \{\gamma(f), \gamma(m)\}$ does not hold anymore with the new coloring. Thus we may assume that C does not contain m.

- If C contains some s_α, then $C = V(H)$, hence C contains m.

- Otherwise, $C \subset \{e, \ell, q, r, v_3\}$. If C contains v_3, it has size at least 5, hence $C = \{e, \ell, q, r, v_3\}$. Since C is closed by predecessor, this means that all the colors of $\widehat{e}, \widehat{\ell}, \widehat{q}, \widehat{r}$ and $\widehat{v_3}$ are actually in $\gamma(\{e, \ell, q, r, v_3\})$. In particular, we get that the union of these lists has size 5. Since $\{m, e, \ell, q, r, v_3\}$ is colorable, this means that we can color m with a color not in $\widehat{e} \cup \widehat{\ell} \cup \widehat{q} \cup \widehat{r} \cup \widehat{v_3}$. We may then color $n, s, o, g, p, i, a, e, \ell, q, r$.

- Otherwise, $C \subset \{e, \ell, q, r\}$. Since C has size at least two, it contains one element among e, r, q, hence it has size four and $C = \{e, \ell, q, r\}$.

  Similarly to the previous item, this means that the union $\widehat{e} \cup \widehat{\ell} \cup \widehat{q} \cup \widehat{r}$ has size 4. Since $\{m, e, \ell, q, r, v_3\}$ is colorable, this means that we can color v_3 then m with a color not in $\widehat{e} \cup \widehat{\ell} \cup \widehat{q} \cup \widehat{r}$. We may then color $n, s, o, g, p, i, a, e, \ell, q, r$.

$\square$

Configuration C_{18}

For reducing the remaining configurations, we use the recoloration technique.

Lemma 1.68. *The graph G does not contain C_{18}.*

Proof. First, we consider the notation depicted in Figure 1.72. By minimality, we color $G \backslash \{a, \dots, f\}$ and uncolor v_1, v_2. By Observation 1.12, we may assume

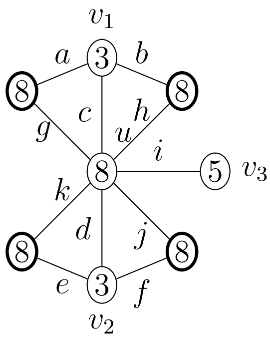


Figure 1.72 – Notation for Lemma 1.68

that $|\widehat{a}| = |\widehat{b}| = |\widehat{e}| = |\widehat{f}| = 2$, $|\widehat{c}| = |\widehat{d}| = 3$ and $|\widehat{v_1}| = |\widehat{v_2}| = 8$. We forget v_1, v_2. In this situation, note that we can extend the coloring to G if and only if one of the following conditions is satisfied:

1. $\widehat{a} \neq \widehat{b}$
2. $\widehat{e} \neq \widehat{f}$
3. $\widehat{c} \setminus \widehat{a} \neq \widehat{d} \setminus \widehat{e}$
4. $|\widehat{c} \setminus \widehat{a}| \neq 1$ or $|\widehat{d} \setminus \widehat{e}| \neq 1$

Indeed, if $\widehat{a} \neq \widehat{b}$ (or similarly $\widehat{e} \neq \widehat{f}$), we color a with a color not in $\widehat{b}$, then color e, f, d, c, b. Otherwise, we color a, b, e, f arbitrarily. If one of the last two conditions holds, then we can color c and d. Therefore, we can extend the coloring to G. Conversely, if none of these conditions holds, then however we color a, b, e, f, we obtain $\widehat{c} = \widehat{d} = \{\alpha\}$ so we cannot produce a coloring of G.

Assume now that none of these conditions holds. We prove that we can recolor some elements among g, h, j, k. This ensures that one of these conditions will hold. If we uncolor g, h, i, j, k, u, we may assume that $|\widehat{g}| = |\widehat{h}| = |\widehat{j}| = |\widehat{k}| = 2$, $|\widehat{u}| = 3$ and $|\widehat{i}| = 4$.

Let H be the color shifting graph of $\{g, h, i, j, k, u\}$. Recall that Lemma 1.17 implies that the in-degree of any vertex $x \neq s_\alpha$ of H is at least $|\widehat{x}| - 1$. By Lemma 1.16, there is a strong component C of H such that $|C| > \max_{x \in C} d^-(x)$. Note that this inequality ensures that $|C| > 1$. We show that C contains g, h, j or k by distinguishing three cases:

1. If C contains a vertex s_α, then we have $|C| > d^-(s_\alpha) = |V(H)| - 1$. Therefore, we have $C = V(H)$, hence C contains g, h, j and k.

2. Otherwise, if C contains i, then we have $|C| > |\widehat{i}| - 1$, so $|C| \geqslant 4$. Hence C also contains g, h, j or k.

3. Otherwise, if C contains u, then $|C| \geqslant 3$ and C contains g, h, j or k.

We thus obtain that C is a strong component of size at least 2 that contains g, h, j or k. Therefore, there is a directed cycle containing at least one of these vertices. Thus, applying Lemma 1.15 gives a valid coloring of $\{g, h, i, j, k, u\}$ where the color of g, h, j or k is different from its color in the previous coloring.

With the initial coloring, we had $\widehat{a} = \widehat{b}$ and $\widehat{d} = \widehat{e}$. Since we recolored at least one element among g, h, j, k, we necessarily have $\widehat{a} \neq \widehat{b}$ or $\widehat{d} \neq \widehat{e}$ with the new coloring. Thus we can extend it to a, b, c, d, e, f. $\square$

Configuration C_{19}

Lemma 1.69. *The graph G does not contain C_{19}.*

Proof. We use the notation depicted in Figure 1.73. By minimality, we color $G \setminus \{a, \dots, f\}$ and uncolor v_1, v_2. Note that we may forget v_1, v_2 and assume

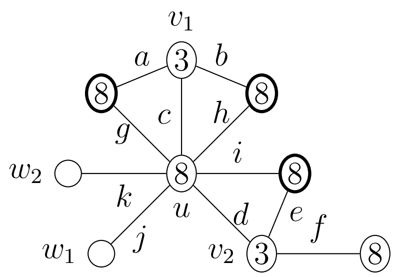


Figure 1.73 – Notation for Lemma 1.69

that $|\widehat{a}| = |\widehat{b}| = |\widehat{e}| = |\widehat{f}| = 2$ and $|\widehat{c}| = |\widehat{d}| = 3$ by Remark 1.12. As in Lemma 1.68, our goal is to obtain one of the following conditions:

1. $\widehat{a} \neq \widehat{b}$

2. $\widehat{e} \neq \widehat{f}$

3. $\widehat{c} \setminus \widehat{a} \neq \widehat{d} \setminus \widehat{e}$

4. $|\widehat{c} \setminus \widehat{a}| \neq 1$ or $|\widehat{d} \setminus \widehat{e}| \neq 1$

Assume that none of them holds. In this case, note that any recoloring of g, h or i is sufficient to ensure that one of these conditions holds. We uncolor u, g, h, i, j, k. We have two cases:

1. If $(d(w_1), d(w_2)) = (4^-, 7^-)$, we uncolor and forget w_1 and we may assume that $|\widehat{g}| = |\widehat{h}| = |\widehat{i}| = |\widehat{k}| = 2$, $|\widehat{u}| = 4$ and $|\widehat{j}| = 6$.

2. If $(d(w_1), d(w_2)) = (5^-, 6^-)$, we may assume that $|\widehat{g}| = |\widehat{h}| = |\widehat{i}| = 2$, $|\widehat{u}| = |\widehat{k}| = 3$ and $|\widehat{j}| = 4$.

Denote by H the color shifting graph of $\{g, h, i, j, k, u\}$. By Lemma 1.16, there exists a strongly connected component C of H such that $|C| > \max_{x \in C} d^-(x)$. By Lemma 1.17, this inequality ensures that $|C| > 1$. We show that C contains g, h or i by distinguishing four cases:

1. If C contains a vertex s_α, then we have $|C| > d^-(s_\alpha) = |V(H)| - 1$. Therefore, $C = V(H)$, and C contains g, h, i.

2. Otherwise, if C contains j, then it has size at least 4, hence it also contains g, h or i.

3. Otherwise, if C contains u, then it has size at least 3, hence contains g, h or i.

4. Otherwise, $C \subset \{g, h, i, k\}$. If C contains k, then its size is at least 2, hence it also contains g, h or i.

We thus obtain that C is a strong component of size at least 2 that contains g, h or i. Therefore, there is a directed cycle containing one of these vertices. Thus, we can apply Lemma 1.15 to ensure that one the conditions is now satisfied, hence we can extend the coloring to G. □

Configuration C_{20}

Lemma 1.70. *The graph G does not contain C_{20}.*

Proof. We use the notation depicted in Figure 1.74. By minimality, we color $G \backslash \{a, b, c\}$ and uncolor v_1, v_2, v_3. Note that we may forget v_1, v_2, v_3 and assume

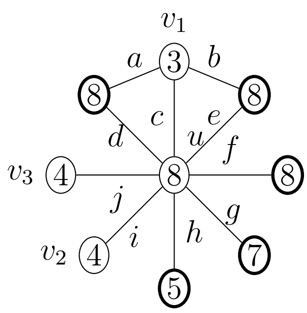


Figure 1.74 – Notation for Lemma 1.70

that $|\widehat{a}| = |\widehat{b}| = |\widehat{c}| = 2$ by Remark 1.12.

If $\widehat{a} \neq \widehat{b}$, we can color c arbitrarily, then a and b. Therefore, assume $\widehat{a} = \widehat{b}$. In this case, note that any recoloring of d or e is sufficient to ensure that $\widehat{a} \neq \widehat{b}$. Denote by H the color shifting graph of $S = \{d, e, f, g, h, i, j, u\}$.

We uncolor the elements of S. Note that we can assume that $|\widehat{f}| = 2$, $|\widehat{d}| = |\widehat{e}| = |\widehat{g}| = 3$, $|\widehat{h}| = |\widehat{u}| = 5$ and $|\widehat{i}| = |\widehat{j}| = 7$.

By Lemma 1.16, there exists a strongly connected component C of H such that $|C| > \max_{x \in C} d^-(x)$. By Lemma 1.17, this inequality ensures that $|C| > 1$. We show that C contains d or e by distinguishing five cases:

1. If C contains a vertex s_α, then we have $|C| > d^-(s_\alpha) = |V(H)| - 1$. Therefore, $C = V(H)$, and C contains d and e.

2. Otherwise, if C contains i or j, then it has size at least 7, hence it also contains d or e.

3. Otherwise, if C contains u or h, then it has size at least 5, hence contains d or e.

4. Otherwise, $C \subset \{d, e, f, g\}$. If C contains g, then its size is at least 3, hence it also contains d or e.

5. Otherwise, $C \subset \{d, e, f\}$. If C contains f, then its size is at least 2, hence it also contains d or e.

We thus obtain that C is a strong component of size at least 2 that contains d or e. Therefore, there is a directed cycle containing one of these vertices. Thus, we can apply Lemma 1.15 to ensure that now $\widehat{a} \neq \widehat{b}$, so that we can extend the coloring to G. $\square$

Configuration C_{21}

To prove that G does not contain C_{21}, we prove that it does not contain the three following configurations:

- C_{21a}: u has a weak neighbor v_1 of degree 3, a $(7, 8)$-neighbor v_2 of degree 4 such that $\text{dist}_u(v_1, v_2) = 2$, and neighbor v_3 of degree 4.

- C_{21b}: u has a weak neighbor v_1 of degree 3, a $(8, 8)$-neighbor v_2 of degree 4 such that $\text{dist}_u(v_1, v_2) = 2$, a neighbor v_3 of degree 4 and a neighbor v_4 of degree 7.

- C_{21c}: u has a weak neighbor v_1 of degree 3, a $(7, 8)$-neighbor v_2 of degree 4 such that $\text{dist}_u(v_1, v_2) \geqslant 3$, and two neighbors of degree 4 and 7.

Lemma 1.71. *The graph G does not contain C_{21a}.*

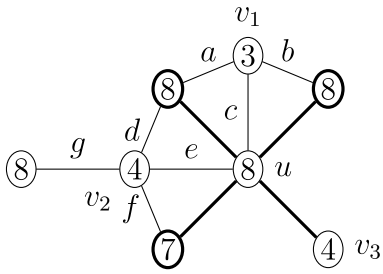


Figure 1.75 – Notation for Lemma 1.71

Proof. We use the notation depicted in Figure 1.75. By minimality, we color $G \setminus \{a, b, c\}$ and uncolor and forget v_1, v_2, v_3. We then uncolor d, e, f, g.

We may thus assume that $|\widehat{b}| = |\widehat{g}| = 2$ and $|\widehat{a}| = |\widehat{c}| = |\widehat{d}| = |\widehat{e}| = |\widehat{f}| = 3$. Moreover, we have $|\widehat{d} \cup \widehat{e} \cup \widehat{f} \cup \widehat{g}| \geqslant 4$ since d, e, f, g were properly colored.

If $\widehat{g}$ is not included in $\widehat{d}$, we color g with a color not in $\widehat{d}$, then b, and apply Lemma 1.20 to color $\{f, d, a, c, e\}$. Therefore, we may assume that $\widehat{g} \subset \widehat{d}$, and similarly, $\widehat{g} \subset \widehat{e}$.

We may also assume that $\widehat{g} \subset \widehat{f}$. Indeed, otherwise, we color g with a color not in $\widehat{f}$, then forget f and apply Lemma 1.20 to $\{b, a, d, e, c\}$.

Now, if $\widehat{f} \not\subset \widehat{d}$, we color f with a color not in $\widehat{d}$ (thus not in $\widehat{g}$), then color b and apply Lemma 1.20 to $\{g, d, a, c, e\}$. If $\widehat{d} \not\subset \widehat{f}$, we color d with a color not in $\widehat{f}$, then color a, b, c, e, g, f. Therefore, we may assume that $\widehat{f} = \widehat{d}$ and similarly $\widehat{f} = \widehat{e}$.

This ensures that

$$|\widehat{f}| = |\widehat{d} \cup \widehat{e} \cup \widehat{f} \cup \widehat{g}| \geqslant 4.$$

We may thus arbitrarily color a, b, c, g, d, e, f. □

Lemma 1.72. *The graph G does not contain C_{21b}.*

Proof. We use the notation depicted in Figure 1.76. By minimality, we color $G \setminus \{a, b, c\}$. We uncolor and forget v_1, v_2, v_3. Denote by α the color of d and β the color of e. We then uncolor d, e, f, g.

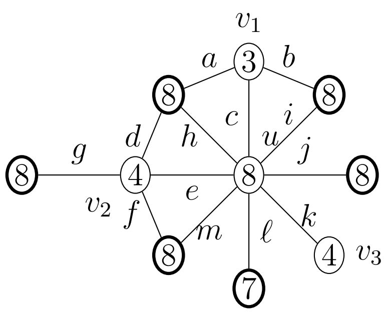


Figure 1.76 – Notation for Lemma 1.72

We may thus assume that $|\widehat{b}| = |\widehat{f}| = |\widehat{g}| = 2$, $|\widehat{a}| = |\widehat{c}| = 3$ and $|\widehat{d}|, |\widehat{e}| \in \{3, 4\}$. Moreover, since d, e, f, g are pairwise incident, they were colored with at least 4 colors before being uncolored. Therefore, we have $|\widehat{d} \cup \widehat{e} \cup \widehat{f} \cup \widehat{g}| \geqslant 4$.

- If $\widehat{d} = \widehat{e}$, then this means there exists $\gamma \in (\widehat{f} \cup \widehat{g}) \setminus (\widehat{d} \cup \widehat{e})$. We color f or g with γ, then g, b. The elements $\{a, c, d, e\}$ induce an even cycle, which is 2-choosable. We may thus assume that $\widehat{d} = \widehat{e}$.

- If $\widehat{f} \neq \widehat{g}$, we color f with $\gamma \notin \widehat{g}$, so that $\widehat{d} \neq \widehat{e}$ if $|\widehat{d}| = |\widehat{e}| = 2$. We may then color b and apply Lemma 1.20 to $\{g, d, a, c, e\}$.

- If $|\widehat{e} \setminus \widehat{f}| = |\widehat{d} \setminus \widehat{f}| = 2$, then we color f and g and apply Lemma 1.20 to $\{b, a, d, e, c\}$. In particular, we have $(|\widehat{d}|, |\widehat{e}|) \neq (4, 4)$.

- Observe that since $\widehat{f} = \widehat{g}$ and d, e, f, g were colored, we have $\alpha, \beta \notin f$. Therefore, d and e are forced to be colored α and β. Hence G is not colorable only if $\widehat{a} \setminus \{\alpha\} = \widehat{c} \setminus \{\beta\} = \widehat{b}$.

- Let γ be the color of h. Observe that $\gamma \in L(a)$. Indeed, since h is adjacent to every colored element incident to a, we could otherwise assume that $|\widehat{a}| = 4$, hence forget a, b, c and color d, e, f, g. Similarly, we have $\gamma \in L(c)$.

We show that we can recolor h, i or m and then extend the coloring with the new available colors of a, b, c, d, e, f, g.

Let H be the color shifting graph of $S = \{h, i, j, k, \ell, m, u\}$. If α appears on S, we remove the arc $s_\alpha \to h$ from H, otherwise α appears on $x \in S$, and we remove the arc $x \to h$ in H. We uncolor the vertices of S, and we have $|\widehat{i}| = |\widehat{j}| = |\widehat{m}| = 2$, $|\widehat{\ell}| = 3$, $|\widehat{h}| = 4$, $|\widehat{u}| = 5$ and $|\widehat{k}| = 7$. Observe that $d^-(h) = 2$.

By Lemma 1.16, there is a strong component C of H such that $|C| > \max_{x \in C} d^-(x)$. We consider two cases, depending on whether C contains a vertex s_δ.

1. Assume first that C does not contain any vertex s_δ.

    - If C contains k, then $|C| \geqslant 7$, hence C contains h, i or m.
    - Otherwise, if C contains u, then $|C| \geqslant 7$ hence C contains h, i or m.
    - Otherwise, $C \subset \{h, i, j, \ell, m\}$. If C contains ℓ, then $|C| \geqslant 3$ hence C contains h, i or m.
    - Otherwise, $C \subset \{h, i, j, m\}$. If C contains j, then $|C| \geqslant 2$ hence C contains h, i or m.

We may thus find a directed cycle in H containing h, i or m and only vertices of S. Shifting the colors along this cycle as done in Lemma 1.15 yields another coloring of S obtained by permuting the colors. Denote by $\widetilde{x}$ the new list of available colors for the element x after the recoloring process.

Observe that since we removed an ingoing arc to h, the edge h cannot be colored with α in the new coloring. This implies that $\alpha \in \widetilde{d}$. Moreover, $\widehat{e} = \widetilde{e}$, hence $\beta \in \widetilde{e}$. We consider three cases, depending on which elements among $\{h, i, m\}$ were recolored.

- Assume that h was recolored. We color d with α and e with β. Since h was recolored, then its former color γ does not appear anymore on a colored incident element of a. Since $\gamma \in L(a)$, we can color a with γ. After this, we may assume that $|b| = |f| = 1, |c| = |g| = 2$, hence we can color b, c, f, g.
- If h was not recolored, then assume that i was. In this case, we may still color d with α and e with β, then color f, g. Since we recolored i, we now have $\widetilde{b} \neq \widetilde{c}$, hence we can color a, b, c.
- Finally, if we only recolored m, then let δ be the former color of m. Note that m is incident to e and $\beta \in \widehat{e}$, hence $\delta \neq \beta$. If $\delta \notin L(f)$, then we could have assumed that $|\widehat{f}| = 3$, and obtained the same situation as in Lemma 1.71. Therefore, we may assume that $\delta \in L(f)$, hence $\delta \in \widetilde{f} \setminus \widehat{f}$. This implies that $\delta \notin \widehat{g}$.

  If $\delta = \alpha$, we color f and a with α, so that we have $|\widetilde{b}| = |\widetilde{c}| = |\widetilde{d}| = |\widetilde{g}| = 2$ and $|\widetilde{e}| = 3$. We then color d, g, e, c, b.

  Otherwise, we color f with δ. Afterwards, we have $\widetilde{g} \neq \widetilde{d}$ or $\widetilde{g} \neq \widetilde{e}$ since $\delta \notin \{\alpha, \beta\}$. We then color b and apply Lemma 1.20 to $\{g, d, a, c, e\}$.

2. Assume now that C contains a vertex s_δ, ensuring that $|C| > |V(H)| - 1$, i.e. H is strongly connected. Note that if $\gamma \notin L(e)$, then $|\widehat{e}| = 4$, hence $|\widehat{d}| = 3$ so $\gamma \in L(d)$.

   We consider a directed path $s_\delta \to \cdots \to h$ in H where each internal vertex lies in S. Since h is colored with γ, s_γ has no successor in H, hence we have $\delta \neq \gamma$. We shift the colors of S along this path, as done in Lemma 1.15. We may then color c and d with γ (this is possible since $\gamma \in L(d)$ and $\gamma \neq \delta$). Note that $\gamma \notin \widehat{d} \cup \widehat{e}$, hence $\gamma \notin \widehat{g} = \widetilde{g}$. Then we may assume that $|\widetilde{f}| = 1$, $|\widetilde{a}| = |\widetilde{b}| = |\widetilde{g}| = 2$ and $|\widetilde{e}| = 3$. We may thus color f, g, e, b, a.

   If $\gamma \in L(e)$, we again shift the colors along the directed path $s_\delta \to \cdots \to h$, then color a, e with γ and d with α (since the new color of h is not

α). Again, we have $\gamma \notin \widetilde{g}$, hence we may assume that $|\widetilde{f}| = 1$ and $|\widetilde{a}| = |\widetilde{b}| = |\widetilde{g}| = 2$. We may thus color f, g, b, c. $\square$

Lemma 1.73. *The graph G does not contain C_{21c}.*

Proof. We use the notation depicted in Figure 1.77. By minimality, we color $G \setminus \{a, b, c\}$. We uncolor and forget v_1, v_2, v_3. Let α, β be the colors of e and f. We then uncolor d, e, f, g. We may thus assume that $|\widehat{a}| = |\widehat{b}| = |\widehat{e}| = |\widehat{f}| = 2$

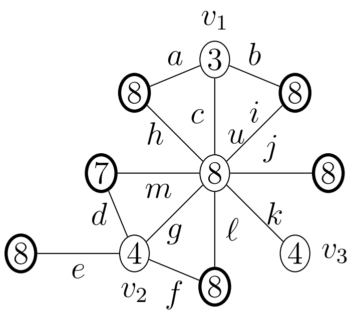


Figure 1.77 – Notation for Lemma 1.73

and $|\widehat{d}| = |\widehat{g}| = |\widehat{i}| = 3$. Moreover, since d, e, f, g are pairwise incident, they were colored with at least 4 colors before being uncolored. Therefore, we have $|\widehat{d} \cup \widehat{e} \cup \widehat{f} \cup \widehat{g}| \geqslant 4$.

- If $\widehat{d} = \widehat{g}$, then this means there exists $\gamma \in (\widehat{e} \cup \widehat{f}) \setminus (\widehat{d} \cup \widehat{g})$. We color e or f with γ, then f, a, b, c, g, d. We may thus assume that $\widehat{d} = \widehat{g}$.

- We also assume that $\widehat{a} = \widehat{b}$. Indeed, otherwise, we color a with a color not in $\widehat{b}$, then forget b, c, and put back the initial colors on d, e, f, g.

- Let γ be the color of m. Observe that $\gamma \in L(c)$. Indeed, since m is adjacent to every colored element incident to c, we could otherwise assume that $|\widehat{c}| = 4$, hence forget c, b, a and color d, e, f, g.

- We now show that we can recolor h or i. Let H be the color shifting graph of $S = \{h, i, j, k, \ell, m, u\}$. We uncolor the vertices of S, and we have $|\widehat{j}| = 2$, $|\widehat{h}| = |\widehat{i}| = |\widehat{\ell}| = 3$, $|\widehat{m}| = 4$, $|\widehat{u}| = 5$ and $|\widehat{k}| = 7$. By Lemma 1.16, there is a strong component C of H such that $|C| > \max_{x \in C} d^-(x)$.

  - If C contains a vertex s_δ, then $|C| = |V(H)|$, hence C contains h and i.
  - If C contains k, then $|C| \geqslant 7$, hence C contains h or i.
  - Otherwise, if C contains u, then $|C| \geqslant 5$ hence C contains h or i.

  - Otherwise, $C \subset \{h, i, j, \ell, m\}$. If C contains m, then $|C| \geqslant 4$ hence C contains h or i.
  - Otherwise, $C \subset \{h, i, j, \ell\}$. If C contains ℓ, then $|C| \geqslant 3$ hence C contains h or i.
  - Otherwise, $C \subset \{h, i, j\}$. If C contains j, then $|C| \geqslant 2$ hence C contains h or i.

  In every case, we can recolor h or i by Lemma 1.15. This allows to obtain new lists $\widetilde{x}$ of available colors for the element x, and we have $\widetilde{a} \neq \widetilde{b}$. However, this may break colorability of d, e, f, g. Proving that this colorability is actually preserved requires a more careful analysis we give in the rest of the proof.

- Let δ, ε be the colors of h and i before recoloring. Observe that $\gamma \in L(d)$ and $\delta, \varepsilon \in L(g)$. Indeed, otherwise, we have $|\widehat{d}| = 4$ or $|\widehat{g}| = 4$ at the beginning. After recoloring, we thus have $|\widetilde{d}| = 4$ or $|\widetilde{g}| = 4$, and $\widetilde{a} \neq \widetilde{b}$. We can therefore extend the coloring to G by coloring a with a color not in $\widetilde{b}$, then forget b, c, and color d, e, f, g.

  We may thus assume that $\gamma \in L(d)$ and that $\delta, \varepsilon \in L(g)$.

- Consider the strong component C of H given by Lemma 1.16. We consider two cases, depending on whether C contains a vertex s_δ.

  - Assume first that C does not contain any vertex s_δ. As we saw, C contains h or i, hence we may find a directed cycle in H containing h or i and only vertices of S.

    The coloring given by applying Lemma 1.15 to this directed cycle uses the same set of colors (the colors are only permuted). Therefore, we have $\widetilde{c} = \widehat{c}$ and $\widetilde{g} = \widehat{g}$. together with $\widetilde{a} \neq \widetilde{b}$.

    If m was not recolored, then we also have $\widetilde{d} = \widehat{d}$. Hence $\widetilde{d}$ and $\widetilde{g}$ are different lists of size at least 3. Otherwise, if m was recolored, then its former color γ lies now in $\widetilde{d}$, but not in $\widetilde{g}$ (since color γ is still present on S).

    Therefore, in both cases, we can hence we can color e, f arbitrarily, then d, g since $\widetilde{d} \neq \widetilde{g}$, then c, and a, b since $\widetilde{a} \neq \widetilde{b}$.

  - Assume now that C contains a vertex s_ζ, ensuring that $|C| > |V(H)| - 1$, i.e. H is strongly connected.

    We consider a directed path $s_\zeta \to \cdots \to h$ in H where each internal vertex lies in S. Since m is colored with γ, s_γ has no successor in H, hence we have $\zeta \neq \gamma$. We shift the colors of S along this path, as done in Lemma 1.15, so that $\widetilde{a} \neq \widetilde{b}$.

Assume that this path goes through m. Then since m is not the last vertex of the path, the color γ is still present on some element of S, hence $\gamma \notin \widetilde{g}$. However, we have $\gamma \in \widetilde{d}$ since $\gamma \in L(d)$ and m is adjacent to every colored element incident to d. Therefore, we have $\widetilde{d} \neq \widetilde{g}$. We can then color e, f, d, g, c, a, b.

– Assume that the path does not go through m. Therefore, the color of m is still γ after the recoloring. Since the initial color δ of h lies in $L(g)$, we now have $\widetilde{g} = (\widehat{g} \setminus \{\zeta\}) \cup \{\delta\}$. If $\widetilde{g} \neq \widetilde{d}$, then we can color a with a color not in $\widetilde{b}$, then forget b, c and color e, f, d, g. Otherwise, we have

$$(\widehat{g} \setminus \{\zeta\}) \cup \{\delta\} = \widetilde{g} = \widetilde{d} = \widehat{d}.$$

We then apply the same argument to a path $s_\eta \to \cdots \to i$. Such a path cannot contain m by the previous item, and we should have

$$(\widehat{g} \setminus \{\eta\}) \cup \{\varepsilon\} = \widehat{d}.$$

Therefore, we can extend the new coloring to G unless we have

$$(\widehat{g} \setminus \{\zeta\}) \cup \{\delta\} = (\widehat{g} \setminus \{\eta\}) \cup \{\varepsilon\} = \widehat{d}.$$

Note that δ and ε are the former colors of h and i, hence they are different. Therefore we also have $\zeta \neq \eta$.

Since $\widehat{d}$ and $\widehat{g}$ are different lists of size 3, there exists $x \in \widehat{g} \setminus \widehat{d}$. If $x \neq \zeta$, then $x \in \widehat{g} \setminus \{\zeta\}$ hence $x \in \widehat{d}$, a contradiction. Otherwise $x = \zeta$, so $x \neq \eta$ and $x \in \widehat{g} \setminus \{\eta\}$. Therefore $x \in \widehat{d}$ and we also get a contradiction. This implies that we can extend the new coloring to G. □

Configuration C_{22}

By definition of C_{22}, if G contains C_{22}, then G contains one of the following:

- C_{22a}: u has three semi-weak neighbors v_1, v_2, v_3 of degree 3 and a neighbor v_4 of degree 7.
- C_{22b}: u has two semi-weak neighbors v_1, v_2 of degree 3, two neighbors w_1, w_2 of degree 4 and a neighbor w_3 of degree 7.

We dedicate a lemma to each of these configurations.

Lemma 1.74. *The graph G does not contain C_{22a}.*

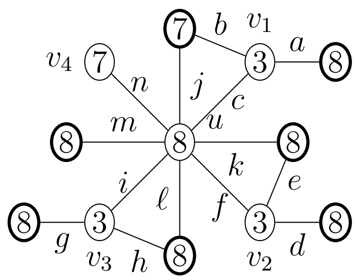


Figure 1.78 – Notation for Lemma 1.74

Proof. We use the notation depicted in Figure 1.78. By minimality, we color $G \setminus \{a, \dots, i\}$ and uncolor v_1, v_2, v_3. By Remark 1.12, we may assume that $|\widehat{a}| = |\widehat{b}| = |\widehat{d}| = |\widehat{e}| = |\widehat{g}| = |\widehat{h}| = 2$, $|\widehat{c}| = |\widehat{f}| = |\widehat{i}| = 4$ and $|\widehat{v_1}| = |\widehat{v_2}| = |\widehat{v_3}| = 7$. We forget v_1, v_2, v_3. We assume that $\widehat{a} = \widehat{b}$, $\widehat{d} = \widehat{e}$, $\widehat{g} = \widehat{h}$, and $\widehat{c} \setminus \widehat{a} = \widehat{f} \setminus \widehat{d} = \widehat{i} \setminus \widehat{g}$ have size 2 (otherwise, we can already extend the coloring to G). Note that any recoloring of j, k or ℓ is sufficient to ensure that this hypothesis does no longer hold. We uncolor j, k, ℓ, m, n, u. We may assume that $|\widehat{m}| = 2$, $|\widehat{n}| = |\widehat{j}| = |\widehat{k}| = |\widehat{\ell}| = 3$ and $|\widehat{u}| = 5$.

Denote by H the color shifting graph of $S = \{j, k, \ell, m, n, u\}$. By Lemma 1.16, there exists a strong component C of H such that $|C| > \max_{x \in C} d^-(x)$. Note that this inequality ensures that $|C| > 1$. We show that C contains j, k or ℓ by distinguishing four cases:

1. If C contains a vertex s_α, then we have $|C| > d^-(s_\alpha) = |V(H)| - 1$. Therefore, $C = V(H)$, and C contains d, e.

2. Otherwise, if C contains u, then it has size at least 5, hence it also contains j, k or ℓ.

3. Otherwise, if C contains n, then its size is at least 3, hence it also contains j, k or ℓ.

4. Otherwise, $C \subset \{j, k, \ell, m\}$. Then, if C contains m, its size is at least 2, hence it also contains j, k or ℓ.

We thus obtain that C is a strong component of size at least 2 that contains j, k or ℓ. Therefore, there is a directed cycle containing one of these vertices. Thus, we can apply Lemma 1.15 to ensure that the starting hypothesis does not hold anymore, hence we can extend the coloring to G. □

Lemma 1.75. *The graph G does not contain C_{22b}.*

Proof. We use the notation depicted in Figure 1.79. By minimality, we color $G \setminus \{a, \dots, f\}$ and uncolor v_1 and v_2. Note that we may assume that $|\widehat{a}| = |\widehat{b}| = |\widehat{e}| = |\widehat{f}| = 2$, $|\widehat{c}| = |\widehat{d}| = 3$ and $|\widehat{v_1}| = |\widehat{v_2}| = 7$. We forget v_1, v_2. As in

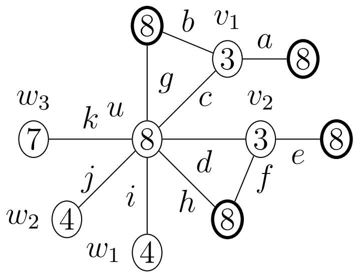


Figure 1.79 – Notation for Lemma 1.75

Lemma 1.68, we assume that $\widehat{a} = \widehat{b}, \widehat{e} = \widehat{f}$ and $\widehat{c} \setminus \widehat{a} = \widehat{d} \setminus \widehat{e} = \{\alpha\}$ (otherwise, we can already extend the coloring to G). Note that any recoloring of g or h is sufficient to ensure that this hypothesis does no longer hold. We uncolor $g, h, i, j, k, u, w_1, w_2$, then forget w_1, w_2.

We may assume that $|\widehat{g}| = |\widehat{h}| = |\widehat{k}| = 2$, $|\widehat{u}| = 5$, and $|\widehat{i}| = |\widehat{j}| = 6$. Denote by H the color shifting graph of $S = \{g, h, i, j, k, u\}$. By Lemma 1.16, there exists a strong component C of H such that $|C| > \max_{x \in C} d^-(x)$. Note that this inequality ensures that $|C| > 1$. We show that C contains g or h by distinguishing three cases:

1. If C contains a vertex s_α, then we have $|C| > d^-(s_\alpha) = |V(H)| - 1$. Therefore, $C = V(H)$, and C contains g, h.

2. Otherwise, if C contains u, i or j, then it has size 5, hence it also contains g or h.

3. Otherwise, $C \subset \{g, h, k\}$. If C contains k, then its size is at least 2, hence it also contains g or h.

We thus obtain that C is a strong component of size at least 2 that contains g or h. Therefore, there is a directed cycle containing one of these vertices. Thus, we can apply Lemma 1.15 to ensure that we can extend the coloring to G. $\square$

1.5 Discharging process

We now strive to reach a contradiction using a double counting argument. To this end, we give an initial weight to every vertex and face such that the total weight is negative. We then introduce a set of discharging rules. Finally, we reach a contradiction by showing that the final weight of each element is non-negative.

1.5.1 The rules

We start with the definition of the initial weighting ω: we set $\omega(v) = d(v) - 6$ and $\omega(f) = 2\ell(f) - 6$ for each vertex v and face f. Using Euler's formula, the total weight is -12.

We then introduce several discharging rules, see Figure 1.80:

- For any 4^+-face f,
  - (R_1) If f is incident to a 5^--vertex u, then f gives 1 to u.
  - (R_2) If f has a vertex v such that $d(v) = 8$ and the neighbors u, w of v along f satisfy $d(u) = 3$ and $d(w) \geqslant 6$, then f gives $\frac{5}{12}$ to v.
  - (R_3) If f has a vertex v such that $d(v) = 7$ and the neighbors u, w of v along f both have degree at least 6, then f gives $\frac{1}{3}$ to v if $d(u) = 6$ or $d(v) = 6$, and $\frac{1}{12}$ otherwise.
  - (R_4) If f has a vertex v such that $d(v) = 7$ and the neighbors u, w of v along f both have degree 5 and 6, then f gives $\frac{1}{6}$ to v, except if $\ell(f) = 4$ and the last vertex of f has degree 5.
- For any 8-vertex u,
  - (R_5) If u has a weak neighbor v of degree 3, then u gives 1 to v.
  - (R_6) If u has a semi-weak neighbor v of degree 3, then u gives $\frac{1}{2}$ to v.
  - (R_7) If u has a (p, q)-neighbor v of degree 4, then u gives ω to v where:

$$\omega = \begin{cases} \frac{2}{3} & \text{if } p = q = 7, \\ \frac{7}{12} & \text{if } p = 7 \text{ and } q = 8, \\ \frac{1}{2} & \text{if } p = q = 8, \\ 0 & \text{otherwise.} \end{cases}$$

  - (R_8) If u has a semi-weak neighbor v of degree 4 and a neighbor w of degree 7 such that uvw is a triangular face, then u gives $\frac{1}{12}$ to v.
  - (R_9) If u has a (p, q)-neighbor v of degree 5 such that $p, q \geqslant 5$, then u gives ω to v where

$$\omega = \begin{cases} \frac{1}{2} & \text{if } p = 5 \text{ and } q = 6, \\ \frac{1}{6} & \text{if } p = 5 \text{ and } q > 6, \\ \frac{2}{3} & \text{if } p = q = 6, \\ \frac{1}{3} & \text{if } v \text{ is an } E_3\text{-neighbor}, \\ \frac{1}{4} & \text{otherwise.} \end{cases}$$

- For any 7-vertex u,

(R_{10}) If u has a (p,q)-neighbor v of degree 4, then u gives ω to v where

$$\omega = \begin{cases} \frac{1}{2} & \text{if } p = q = 7, \\ \frac{5}{12} & \text{if } p = 7 \text{ and } q = 8, \\ \frac{1}{3} & \text{if } p = q = 8, \\ 0 & \text{otherwise.} \end{cases}$$

- For any 7-vertex u with a weak neighbor v of degree 5,

(R_{11}) If v is an $(5,6)$-neighbor of u, then u gives $\frac{1}{2}$ to v.

(R_{12}) If v is an S_3-neighbor of u, then u gives $\frac{1}{3}$ to v.

(R_{13}) If v is an S_5-neighbor of u, then u gives $\frac{1}{5}$ to v.

(R_{14}) If v is not an $(5,6)$-, S_3-, nor S_5-neighbor of u, then u gives $\frac{1}{6}$ to v.

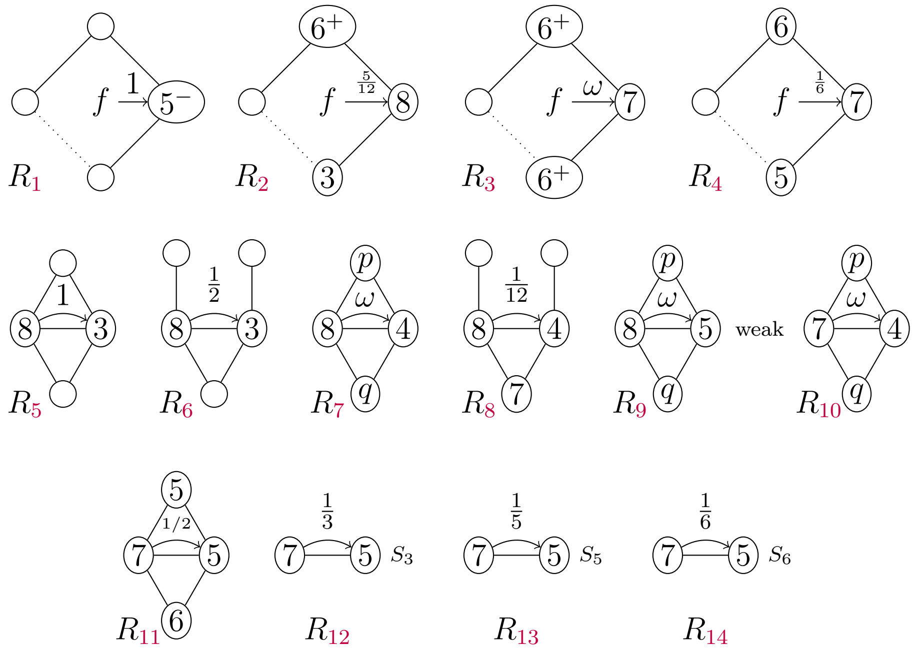


Figure 1.80 – The discharging rules

It then remains to prove that each element ends up with non-negative weight. We first handle the faces, and then distinguish several cases for the vertices, depending on their degree. First note that due to C_1, the minimum degree of G is 3. Moreover, only vertices of degree 7 or 8 lose weight.

1.5.2 Faces

Note that only faces of length at least 4 lose weight. Consider a 4^+-face f. We distinguish some cases, depending on its length ℓ and the minimal degree δ of its incident vertices.

1. $\ell \geqslant 6$: By rules R_1, R_2, R_3 and R_4, the face f loses at most 1 for each of its vertices, hence
$$\omega'(f) \geqslant 2\ell - 6 - \ell = \ell - 6 \geqslant 0$$

2. $\delta = 3$ and $\ell = 4$: Let $f = uu_1vu_2$ where $d(u) = 3$. By C_1, both u_1 and u_2 are 8-vertices. Consider the other neighbor v of these 8-vertices. If v is a 6-vertex or a 8-vertex, then f loses $2 \times \frac{5}{12}$ on u_1 and u_2 by R_2 and f does not lose anything on v since R_2, R_3 and R_4 do not apply.

   If v is a 7-vertex, then f loses $2 \times \frac{5}{12}$ on u_1 and u_2 by R_2 and $\frac{1}{12}$ on v by R_3.

   Otherwise, v is a 5^--vertex and f loses 1 on v by R_1 but nothing on u_1, u_2. Thus the final weight of f is at least $2 - 1 - 1 = 0$.

3. $\delta = 3$ and $\ell = 5$: Let $f = uu_1v_1v_2u_2$ where $d(u) = 3$. By C_1, we have $d(u_1) = d(u_2) = 8$. By R_2, the vertices u_1 and u_2 receive at most $\frac{5}{12}$. The three remaining vertices receive at most 1 by R_1, R_2, R_3 and R_4. Therefore, the final weight of f is at least $4 - 3 \times 1 - 2 \times \frac{5}{12} = \frac{1}{6} > 0$.

4. $\delta = 4$: By C_1, any 4-vertex is adjacent to 7^+-vertices. These vertices do not receive any weight from f. Therefore, f loses at most $(\ell - 2) \times 1$ by R_1, R_2, R_3 and R_4. Hence $\omega'(f) = 2\ell - 6 - (\ell - 2) = \ell - 4 \geqslant 0$.

5. $\delta = 5$ and $\ell = 4$: If there is only one 5-vertex u, then f gives 1 to u and at most $3 \times \frac{1}{3}$ to the other vertices by R_1, R_3 and R_4.

   Otherwise, by C_4, there are two 5-vertices and the two other vertices have degree at least 6. Thus only R_1 applies, and f loses 2, giving a final weight of $2 - 2 = 0$.

6. $\delta = 5$ and $\ell = 5$. By C_4, there are at most three 5-vertices. If there are three such vertices, then R_3 and R_4 do not apply and the final weight of f is $4 - 3 \times 1 = 1 > 0$. If f has two vertices of degree 5, f gives 2×1 to these vertices by R_1 and at most $3 \times \frac{1}{3}$ to the others by R_3 and R_4. Therefore, the final weight is at least $4 - 2 \times 1 - 3 \times \frac{1}{3} = 1 > 0$.

7. $\delta > 5$: Only R_3 applies, so f loses at most $\ell \times \frac{1}{3}$. The final weight is $2\ell - 6 - \frac{\ell}{3} > 0$ since $\ell \geqslant 4$.

1.5.3 3-vertices

Let u be a 3-vertex. Note that due to C_1, each neighbor of u is an 8-vertex. We consider four cases depending on the number n_t of triangular faces containing u. In each case, we show that u receives a weight of 3 during the discharging procedure, so its final weight is 0.

1. $n_t = 0$: by R_1, the vertex u receives 1 from each incident face.

2. $n_t = 1$: the vertex u receives 2 by R_1. Moreover, u is a semi-weak neighbor of two 8-vertices. By R_6, it receives $2 \times \frac{1}{2}$.

3. $n_t = 2$: the vertex u receives 1 by R_1. Moreover, u is a semi-weak neighbor of two 8-vertices, and a weak neighbor of another 8-vertex. By R_5 and R_6, it receives $1 + 2 \times \frac{1}{2}$.

4. $n_t = 3$: the vertex u is a weak neighbor of three 8-vertices. By R_5, it receives 3×1.

1.5.4 4-vertices

Similarly to the previous subsection, we take a 4-vertex u and consider several cases considering the number n_t of triangular faces incident with u. In each case, we show that u receives at least a weight of 2, so ends up with non-negative weight. Recall that, due to C_1, every neighbor of u has degree at least 7.

1. $n_t \leqslant 2$: By R_1, the vertex u receives $(4 - n_t) \times 1 \geqslant 2$ from incident faces.

2. $n_t = 3$: In this case, u receives 1 by R_1. Moreover, u is a weak neighbor of two vertices w_1 and w_2 and a semi-weak neighbor of two other ones s_1 and s_2.

   (a) If $d(w_1) = d(w_2) = 8$ then both w_1 and w_2 give at least $\frac{1}{2}$ to u by R_7, hence u receives 1.

   (b) If $d(w_1) = d(w_2) = 7$, then for $1 \leqslant i \leqslant 2$, either $d(s_i) = 7$ and w_i gives $\frac{1}{2}$ to u by R_{10}, or $d(s_i) = 8$ and u receives $\frac{5}{12} + \frac{1}{12}$ from s_i and w_i by R_{10} and R_8. In both cases, u receives $2 \times \frac{1}{2} = 1$.

   (c) If $d(w_1) = 7$ and $d(w_2) = 8$ (the other case is similar), then w_2 gives at least $\frac{7}{12}$ to u by R_7. Moreover, if $d(s_1) = 7$, the vertex u receives $\frac{5}{12}$ from w_1 by R_{10}. Otherwise, $d(s_1) = 8$, and u receives $\frac{1}{3} + \frac{1}{12}$ from w_1 and s_1 by R_{10} and R_8. In both cases, u receives $\frac{7}{12} + \frac{5}{12} = 1$.

3. $n_t = 4$: In this case, u is a weak neighbor of four 7^+-vertices, say $w_1, \ldots, w_4$, sorted by increasing degree. We show that applying R_7 and/or R_{10} gives a weight of 2 to u in any case.

   (a) If $d(w_1) = 8$, or $d(w_4) = 7$, then each w_i gives $\frac{1}{2}$, hence u receives $4 \times \frac{1}{2} = 2$.

   (b) If $d(w_1) = 7$ and $d(w_2) = 8$, then w_1 gives $\frac{1}{3}$, its two neighbors among $\{w_1, \ldots, w_4\}$ give $2 \times \frac{7}{12}$ and the remaining vertex gives $\frac{1}{2}$.

(c) If $d(w_2) = 7$ and $d(w_3) = 8$, then if $w_3w_4 \in E(G)$, u receives $2 \times \frac{7}{12}$ from w_3 and w_4, and $2 \times \frac{5}{12}$ from w_1 and w_2. Otherwise, u receives $2 \times \frac{2}{3}$ from w_3 and w_4 and $2 \times \frac{1}{3}$ from w_1 and w_2.

(d) If $d(w_3) = 7$ and $d(w_4) = 8$, then w_4 gives $\frac{2}{3}$ to u, its neighbors among the w_i's gives $2 \times \frac{5}{12}$ and the last neighbor of u gives $\frac{1}{2}$.

1.5.5 5-vertices

Take a 5-vertex u. If u is incident to a non-triangular face, then it receives 1 by R_1. Thus, we only have to consider the case where u is triangulated. We denote by $v_1, \ldots, v_5$ the consecutive neighbors of u in the chosen embedding of G.

Note also that due to C_1, the minimum degree δ of the neighborhood of u is at least 5. We distinguish three cases depending on δ. In each case, we show that u receives a weight of at least 1, hence ends up with non-negative weight.

1. If $\delta \geqslant 7$: denote by n_8 the number of 8-vertices adjacent to u. By R_9, the vertex u receives at least $n_8 \times \frac{1}{3}$. We thus may assume that $n_8 < 3$.

   (a) If $n_8 = 0$, by R_{13}, the vertex u receives $5 \times \frac{1}{5} = 1$.

   (b) If $n_8 = 1$, we may assume that $d(v_1) = 8$. Then u is an S_6-neighbor of v_2, v_3, v_4 and v_5. By R_{14}, it receives $\frac{1}{6}$ from each of these vertices. At the end, the received weight is thus $\frac{1}{3} + 4 \times \frac{1}{6} = 1$.

   (c) If $n_8 = 2$, then each of them gives at least $\frac{1}{4}$, while the other neighbors give $\frac{1}{6}$. Thus u receives at least $2 \times \frac{1}{4} + 3 \times \frac{1}{6} = 1$.

   (d) If $n_8 \geqslant 3$, then u receives at least $n_8 \times \frac{1}{4} + (5 - n_8) \times \frac{1}{6} \geqslant 1$.

2. If $\delta = 6$, we consider different cases depending on the number n_6 of 6-vertices in the neighborhood of u. Note that $n_6 \leqslant 3$ because of C_6.

   (a) If $n_6 = 3$, then denote by x and y the two neighbors of u of degree at least 7. Due to C_6, the vertices x and y are not consecutive neighbors of u, and moreover, we cannot have $d(x) = d(y) = 7$. We may thus assume that $d(x) = 8$. Therefore, u receives $\frac{2}{3}$ from x by R_9 and at least $\frac{1}{3}$ from y by R_{12} or R_9.

   (b) If $n_6 = 2$, then for any i such that $d(v_i) > 6$, u is either an S_3 or an E_3-neighbor of v_i. Thus, by R_{12} or R_9, the vertex u receives $3 \times \frac{1}{3}$ from them.

   (c) If $n_6 = 1$, then we may assume that $d(v_1) = 6$. Thus, for $i = 2, 5$, u is an S_3-neighbor or an E_3-neighbor of v_i. Thus, by R_9 or R_{12}, u receives $2 \times \frac{1}{3}$ from v_2 and v_5. Moreover, u receives at least $2 \times \frac{1}{6}$ by R_{14} and R_9 from v_3 and v_4. In any case, u receives at least $2 \times \frac{1}{3} + 2 \times \frac{1}{6} = 1$.

3. If $\delta = 5$, note that u is adjacent to only one 5-vertex (due to C_4). We may thus assume that $d(v_1) = 5$ and $d(v_i) > 5$ for $2 \leqslant i \leqslant 5$. Moreover, we have $d(v_2) > 6$ and $d(v_5) > 6$ due to C_3. We show that v_2 and v_3 give together at least $\frac{1}{2}$ to u. By symmetry, u will receive at least $2 \times \frac{1}{2}$ from v_2, v_3, v_4 and v_5.
   (a) If $d(v_3) = 6$, then u is an $(5,6)$-neighbor of v_2. Thus, u receives $\frac{1}{2}$ by R_9 or R_{11}.
   (b) Otherwise, u is either an S_3-neighbor or an E_3-neighbor of v_3. Therefore, u receives $\frac{1}{3}$ from v_3 by R_9 or R_{12}, and at least $\frac{1}{6}$ from v_2 by R_9 or R_{14}.

1.5.6 6-vertices

Note that 6-vertices do not give nor receive any weight. Moreover, their initial weight is 0. Thus their final weight is 0, hence non-negative.

1.5.7 7-vertices

Let u be a 7-vertex, and denote by $v_1, \ldots, v_7$ its consecutive neighbors in the chosen planar embedding of G.

Note that, due to C_1, the neighbors of u have degree at least 4. Observe also that u gives weight only to its weak neighbors of degree 4 or 5. We show that u loses at most 1 during the discharging phase, thus ends up with a non-negative weight. We distinguish cases depending on several parameters, like the minimum degree δ of the v_i's, or the number of weak neighbors of u of given degree. Note that we may thus assume that δ is 4 or 5. Moreover, due to C_1 and C_4, there are at most four weak neighbors of u.

1. Assume that all the weak neighbors of u have degree 5. We separate three cases depending on the number of triangular faces containing u and two of these neighbors.
   (a) Assume that there are two such triangular faces. Then we may assume that v_2, v_3, v_5 and v_6 have degree 5, and the other neighbors have degree at least 6. By C_5, we have $d(v_4) > 6$, thus v_3 and v_5 are S_6-neighbors of u. Hence u gives $2 \times \frac{1}{6}$ to them by R_{14}.
   If $d(v_1) = d(v_7) = 6$ then by C_5, we have $v_1 v_7 \notin E(G)$. Therefore, u gives at most $\frac{1}{2}$ to v_2 and v_6 by R_{11}, and receives $\frac{1}{3}$ from the face containing v_1 and v_7 by R_3. The final weight loss is then at most $2 \times \frac{1}{6} + 2 \times \frac{1}{2} - \frac{1}{3} = 1$.
   Otherwise, we may assume that $d(v_1) > 6$. In this case, u gives at most $\frac{1}{6}$ to v_2 by R_{14} and at most $\frac{1}{2}$ to v_6 by R_{11}, for a total loss of $2 \times \frac{1}{6} + \frac{1}{6} + \frac{1}{2} = 1$.

(b) Assume that there is only one triangular face containing u and two of its weak neighbors of degree 5 (say v_2 and v_3).

i. Assume that there are two more weak neighbors of degree 5. Due to C_4, the other vertices of degree 5 are v_5 and v_7. By C_5, we have $d(v_1) > 6$ and $d(v_4) > 6$. Then, u gives $2 \times \frac{1}{6}$ to v_2 and v_3 by R_{14}, and at most $2 \times \frac{1}{3}$ to v_5 and v_7 by R_{12} or R_{14}. The total loss is then at most $2 \times \frac{1}{6} + 2 \times \frac{1}{3} = 1$.

ii. Assume that there is only one other weak neighbor of u of degree 5. Due to C_5, then either v_1 or v_4 (say v_1, by symmetry) has degree at least 7 (otherwise v_2 and v_3 would be adjacent $(5,6)$-neighbors of u).
If the other weak neighbor of u of degree 5 is not a $(5,6)$-neighbor, then u gives at most $\frac{1}{3}$ to it, and at most $\frac{2}{3}$ to v_2 and v_3 by R_{11} and R_{14}, for a total loss of 1.
Therefore, assume that the other weak neighbor of u is a $(5,6)$-neighbor. Then, it has to be v_6 by C_5. If $d(v_4) > 6$, then u gives at most $2 \times \frac{1}{6}$ to v_2, v_3 by R_{14}, and $\frac{1}{2}$ to v_6, for a total loss of $\frac{5}{6} < 1$.
In this case, we may assume that $d(v_4) = 6$ and $(d(v_5), d(v_7)) = (6,5)$ or $(5,6)$. In the first case, note that $v_4v_5 \notin E(G)$ due to C_5. Then, by R_3, u receives $\frac{1}{3}$. The total loss is then $\frac{2}{3} + \frac{1}{2} - \frac{1}{3} = \frac{5}{6} < 1$.
Otherwise, we have $d(v_5) = 5$ and $d(v_6) = 6$. Again due to C_5, $v_4v_5 \notin E(G)$. By R_4, u receives $\frac{1}{6}$, and the total loss is $\frac{2}{3} + \frac{1}{2} - \frac{1}{6} = 1$. This holds except if the face containing u, v_4 and v_5 is a 4-face and the last vertex y of this face has degree 5. Due to C_4, this means that $y = v_6$, which creates C_5.

iii. Assume that there is no other weak neighbor of degree 5. Then u only gives weight to its two weak neighbors of degree 5, hence loses at most $2 \times \frac{1}{2} = 1$ by R_{11}.

(c) Assume that there is no triangular face containing u and two of its weak neighbors of u of degree 5. If u has two $(5,6)$-neighbors of degree 5, then by C_5, u does not give weight to any other neighbor. Thus, by R_{11}, the total loss is $2 \times \frac{1}{2} = 1$.

If u has one $(5,6)$-neighbor, say v_2, so that $d(v_1) = 5$ and $d(v_3) = 6$, then v_4, v_7 cannot be weak neighbors of degree 5 of u by C_5. Therefore, by assumption, v_5 and v_6 cannot be both weak neighbors of u. Hence, there are only two weak neighbors of u: v_2 and either v_5 or v_6. The loss is then at most $\frac{1}{2} + \frac{1}{3} = \frac{5}{6} < 1$.

If u has no $(5,6)$-neighbors, then R_{11} never applies and u loses at most $\frac{1}{3}$ for each of its $(6^+,6^+)$-neighbors. Note that u has degree 7, hence there are at most three such neighbors. The total weight loss of u is thus at most $3 \times \frac{1}{3} = 1$.

2. Assume that u has only one weak neighbor of degree 4, say v_4. By C_7, there is no $(5,6)$-neighbor of u of degree 5, hence R_{11} never applies.

   (a) If $d(v_3) = d(v_5) = 8$, then u gives $\frac{1}{3}$ to v_4 by R_{10}, and at most $\frac{1}{3}$ to the weak neighbors of degree 5 among v_1, v_2, v_6 and v_7 by R_{12}, R_{13} or R_{14}. Moreover, due to C_4, there are at most three such neighbors. Therefore, if u has at most two weak neighbors of degree 5, then u loses at most $\frac{1}{3} + 2 \times \frac{1}{3} = 1$ by R_{10} and R_{12}. Otherwise, there are three weak 5-neighbors of u among v_1, v_2, v_5 and v_6, and due to C_5, the remaining vertex has degree at least 7. In this case, u loses at most $\frac{1}{3} + \frac{1}{3} + 2 \times \frac{1}{6} = 1$ by R_{10}, R_{12} and R_{14}.

   (b) Assume that $d(v_3) = 7$ and v_2 is a weak neighbor of u of degree 5.

      i. Assume first that v_1 is a weak neighbor of u of degree 5. In this case, u gives at most $\frac{1}{2}$ to v_4 by R_{10}. Moreover, by C_7, we have $d(v_7) > 6$. Therefore, u gives $2 \times \frac{1}{6}$ to v_1 and v_2 by R_{14}. By C_1, we have $d(v_5) > 6$. If v_6 is weak neighbor of u of degree 5, then it is not a $(5,6)$-neighbor since $d(v_5)$ and $d(v_7)$ are at least 7. Moreover, by C_9, it is not an S_3 nor S_5-neighbor of u. Therefore, by R_{14}, u gives at most $\frac{1}{6}$ to v_6. Therefore, the total loss is at most 1.

      ii. Assume that v_1 is not a weak neighbor of u of degree 5. If there is no other neighbor of u of degree 5, then u loses at most $\frac{1}{2} + \frac{1}{3} = \frac{5}{6} < 1$ by R_{10} and R_{12}. Otherwise, recall that u has no $(5,6)$-neighbor of degree 5, hence by C_8, the vertex v_2 is not an S_3-neighbor of u. Thus $d(v_1) > 6$ and u gives at most $\frac{1}{5}$ to v_2 by R_{13} or R_{14}.
      If $d(v_7) > 5$, then either $d(v_5) = 8$ and u loses at most $\frac{5}{12} + \frac{1}{5} + \frac{1}{3} = \frac{57}{60} < 1$ by R_{10}, R_{13} and R_{12}, or $d(v_5) = 7$, hence by C_8, the vertex v_6 is not an S_3-neighbor of u. Thus u loses at most $\frac{1}{2} + \frac{1}{5} + \frac{1}{5} = \frac{9}{10} < 1$ by R_{10} and R_{13}.
      If $d(v_7) = 5$, then by C_9, the vertex v_2 is not an S_5-neighbor of u, hence u gives at most $\frac{1}{6}$ to v_2 by R_{14}. Since $d(v_1) > 6$, the vertex u gives at most $\frac{1}{3}$ to v_6 and v_7 by R_{12}, R_{13} or R_{14}. The total loss is then at most $\frac{1}{2} + \frac{1}{6} + \frac{1}{3} = 1$

   (c) Assume that $d(v_3) = 7$ and v_2 is not a weak neighbor of u.

      i. If v_1 and v_6 are weak neighbors of u of degree 5, then by C_4, we have $d(v_7) > 5$. If $d(v_5) = 8$, then by C_9, the vertex v_6 is not an S_3-neighbor of u, thus u loses at most $\frac{5}{12} + \frac{1}{3} + \frac{1}{5} = \frac{57}{60} < 1$ by R_{10}, R_{13} and R_{12}. If $d(v_5) = 7$, then by C_9, the vertex v_6 is not an S_3 nor an S_5-neighbor of u, thus u loses at most $\frac{1}{2} + \frac{1}{6} + \frac{1}{3} = 1$ by R_{10}, R_{14} and R_{12}.

ii. If v_1 is a weak neighbor of u of degree 5 but v_6 is not, then we may assume that v_7 is also a weak neighbor of u of degree 5, otherwise u gives at most $\frac{1}{2}$ to v_4 by R_{10} and at most $\frac{1}{3}$ to v_1 by R_{12}, for a total loss of at most $\frac{5}{6}$.
By C_7, we must have $d(v_2) > 6$ and $d(v_6) > 6$. Thus, u gives $2 \times \frac{1}{6}$ to v_1 and v_7 by R_{14} and $\frac{1}{2}$ to v_4 by R_{10} so the total loss is at most $\frac{5}{6}$.

iii. If v_1 is not a weak neighbor of degree 5 of u, then we may assume that v_6 and v_7 are weak neighbors of u of degree 5, otherwise, u gives at most $\frac{1}{2}$ to v_4 and at most $\frac{1}{3}$ to v_6 and v_7, for a loss of at most $\frac{5}{6}$.
In this case, by C_7, we must have $d(v_1) > 6$, therefore u gives $2 \times \frac{1}{6}$ to v_6 and v_7, and again the total loss is at most $\frac{5}{6}$.

3. Assume that u has exactly two weak neighbors of degree 4, and that they are at triangle-distance 2 in the neighborhood of u, say v_2 and v_4. By C_2, we have $d(v_3) = 8$. If there is no weak neighbor of u of degree 5, then the total loss is at most $2 \times \frac{1}{2} = 1$ by R_{10}. By symmetry, we may thus assume that v_6 is a weak neighbor of u of degree 5. Note that, by C_1, we have $d(v_1) \geqslant 7$, hence by C_{11}, we have $d(v_5) = 8$.

   (a) If $d(v_1) = 8$, then u gives $2 \times \frac{1}{3}$ to v_2 and v_4 by R_{10}. Moreover, either $d(v_6) = d(v_7) = 5$ and u loses $2 \times \frac{1}{6}$ by R_{14}, or u gives $\frac{1}{3}$ to v_6 by R_{12}, R_{13} or R_{14}. The total loss is thus $2 \times \frac{1}{3} + \frac{1}{3} = 1$.

   (b) If $d(v_1) = 7$ and $v_1v_7 \in E(G)$, then by C_{11}, we have $d(v_7) > 5$ and v_5 is not an S_3-neighbor of u. Therefore, u gives $\frac{1}{3}$ to v_4, $\frac{5}{12}$ to v_2 and at most $\frac{1}{5}$ to v_6 by R_{10} and R_{13} or R_{14}. The final loss is thus at most $\frac{1}{3} + \frac{5}{12} + \frac{1}{5} = \frac{57}{60} < 1$.

   (c) If $d(v_1) = 7$, $v_1v_7 \notin E(G)$ and $d(v_7) = 5$, then u does not give any weight to v_7 since it is not a weak neighbor. The loss for u can be decomposed as $\frac{1}{3}$ for v_4 by R_{10}, $\frac{5}{12}$ for v_2 by R_{10} and $\frac{1}{6}$ for v_6 by R_{14}, for a total loss of at most $\frac{11}{12} < 1$.

   (d) If $d(v_1) = 7$, $v_1v_7 \notin E(G)$ and $d(v_7) > 5$, then u receives $\frac{1}{12}$ by R_3. Moreover, u gives $\frac{1}{3} + \frac{5}{12}$ to v_2 and v_4 by R_{10} and $\frac{1}{3}$ to v_6 by R_{12}, R_{13} or R_{14}. The final loss is then at most $\frac{1}{3} + \frac{5}{12} + \frac{1}{3} - \frac{1}{12} = 1$.

4. Assume that u has exactly two weak neighbors of degree 4, and that they are at triangle-distance at least 3 in the neighborhood of u, say v_2 and v_6. By C_1, the vertices v_1, v_3, v_5 and v_7 have degree at least 7. We may assume that v_4 is a weak neighbor of u of degree 5, since otherwise u gives at most $2 \times \frac{1}{2}$ to v_2 and v_6 by R_{10} and 0 to its other neighbors. Due to C_{12}, we also know that v_2 and v_6 are $(7^+, 8)$-neighbors of u, hence u gives them at most $2 \times \frac{5}{12}$ by R_{10}.

(a) If v_2 (or similarly v_6) is a $(7,8)$-neighbor of u, then v_4 is not an S_3 nor an S_5-neighbor (by C_{12}), hence u loses at most $2 \times \frac{5}{12} + \frac{1}{6} = 1$ by R_{10} and R_{14}.

(b) Otherwise, v_2 and v_6 are $(8,8)$-neighbors of u, hence u gives $2 \times \frac{1}{3}$ to v_2 and v_6 by R_{10} and at most $\frac{1}{3}$ to v_4 by R_{12}. The total loss is thus at most $3 \times \frac{1}{3} = 1$.

5. Assume that u has three weak neighbors of degree 4, say v_2, v_4 and v_6. Due to C_{10}, the vertices v_1, v_3, v_5 and v_7 all have degree 8. By R_{10}, the vertex u gives $3 \times \frac{1}{3}$ to v_2, v_4 and v_6, for a total loss of 1.

1.5.8 8-vertices

Let u be an 8-vertex, and denote by $v_1, \ldots, v_8$ its neighbors in consecutive order in the chosen planar embedding of G. We show that u loses at most 2 during the discharging phase, thus ends up with non-negative weight. We distinguish several cases depending on the minimum degree δ (resp. δ_w) of the neighbors of u (resp. weak neighbors of u), and several parameters like the number of weak/semi-weak neighbors of u of given degree. Note that $\delta(G) = 3$ and u gives weight only to its weak or semi-weak neighbors, so we may thus assume that δ is $3, 4$ or 5.

1. Assume that $\delta_w = 5$ and $\delta \geqslant 4$. We consider several cases depending on the neighbors of u of degree 5.

(a) If u has at most two weak neighbors of degree 5, then there is room for at most three semi-weak neighbors of degree 4. Hence u loses at most $2 \times \frac{2}{3} + 3 \times \frac{1}{12} = \frac{19}{12} < 2$ by R_9 and R_8.

(b) If u has three weak neighbors of degree 5 then there is room for at most two semi-weak neighbor of degree 4. We may assume that at least one of them shares a neighbor of degree 7 with u, otherwise, u does not give them any weight and u loses at most $3 \times \frac{2}{3} = 2$ by R_9.

Therefore, at least one of the weak neighbors of degree 5 is not a $(6,6)$-neighbor of u, hence u loses at most $2 \times \frac{2}{3} + \frac{1}{2} + 2 \times \frac{1}{12} = 2$ by R_9 and R_8.

(c) If u has four neighbors of degree 5, all being weak, and such that at least two of them are $(6,6)$-neighbors, then by C_{13}, the vertex u has a neighbor of degree at least 7. This implies that u has exactly two $(6,6)$ neighbors, and either two $(6,7^+)$-neighbors, or a $(5,6)$- and a $(5,7^+)$-neighbor of degree 5. Then, u loses at most $2 \times \frac{2}{3} + 2 \times \frac{1}{3} = 2$ or $2 \times \frac{2}{3} + \frac{1}{2} + \frac{1}{6} = 2$ by R_9.

(d) Assume that u has four neighbors of degree 5, all being weak, and such that at most one them is a $(6,6)$-neighbor. Note that there are zero or two $(5,6^+)$-neighbors of u of degree 5. In the first case, the vertex u loses at most $\frac{2}{3}+3\times\frac{1}{3}=\frac{5}{3}<2$. In the latter one, the vertex u loses $\frac{2}{3}+2\times\frac{1}{2}+\frac{1}{3}=2$ by R_9.

(e) If u has at least five neighbors of degree 5 and at least four of them are weak, then by C_4, we may assume that $d(v_1)=d(v_2)=d(v_4)=d(v_6)=d(v_7)=5$. If v_4 is not weak, then u gives at most $4\times\frac{1}{2}=2$ to v_1,v_2,v_6 and v_7 by R_9. Otherwise, by C_{13}, at least one vertex among v_3,v_5 and v_8 must have degree at least 7.

   i. If $d(v_3)>6$ (or similarly if $d(v_5)>6$), then by R_9, the vertex u gives at most $3\times\frac{1}{2}$ to v_1,v_6 and v_7, $\frac{1}{3}$ to v_4 and $\frac{1}{6}$ to v_2, for a total loss of at most 2.

   ii. Otherwise, $d(v_8)>6$ and, by R_9, u gives $\frac{2}{3}$ to v_4, $2\times\frac{1}{2}$ to v_2 and v_6 and $2\times\frac{1}{6}$ to v_1 and v_7, for a total loss of 2.

2. If $\delta=\delta_w=4$ and u has four weak neighbors of degree 4, then by C_{14}, they all are $(8,8)$-neighbors of u. Thus u gives them $4\times\frac{1}{2}=2$ by R_7.

3. If $\delta=\delta_w=4$ and u has three weak neighbors of degree 4, then we may assume that there is also a weak neighbor of degree 5 or a semi-weak neighbor of degree 4 whose common neighbor with u has degree 7, otherwise u gives at most $3\times\frac{2}{3}=2$ by R_7. We may thus assume that $d(v_1)=d(v_3)=d(v_5)=4$ and $d(v_7)=4$ or 5. Moreover, by C_{14}, we may assume that $d(v_2)=8$.

   If $d(v_4)=8$, then u gives $\frac{1}{2}$ to v_3, at most $2\times\frac{7}{12}$ to v_1 and v_5 by R_7, and either $\frac{1}{12}$ or $\frac{1}{3}$ to v_7 by R_8 or R_9. The total loss is thus at most 2.

   Otherwise, if v_7 is a semi-weak neighbor of u of degree 4, then u gives at most $2\times\frac{7}{12}$ to v_1 and v_3, $\frac{2}{3}$ to v_5 by R_7 and $\frac{1}{12}$ to v_7 by R_8, for a total loss of $\frac{23}{12}<2$.

   Otherwise, v_7 is a weak neighbor of u of degree 5. By C_{14}, we have $d(v_6)=8$ and $d(v_8)=8$. Thus u gives $\frac{1}{2}$ to v_1, at most $2\times\frac{7}{12}$ to v_3 and v_5 by R_7 and $\frac{1}{3}$ to v_7 by R_9. The total loss is again at most 2.

4. If $\delta=\delta_w=4$ and u has two weak neighbors of degree 4. Then u gives them at most $2\times\frac{2}{3}$ by R_7. Moreover, u gives weight to at most two other vertices, which could be either semi-weak neighbors of degree 4 or weak neighbors of degree 5. Assume that u gives more than $\frac{1}{3}$ to one of them. Then it is a $(6^-,6)$-neighbor of u. By C_1, there can be only one such neighbor, in which case u loses at most $\frac{2}{3}$ by R_9. Therefore, in each case, u loses at most $2\times\frac{2}{3}+\frac{2}{3}=2$.

5. If $\delta = \delta_w = 4$, and u has one weak neighbor of degree 4 say v_2. Note that u has at most four weak neighbors of degree 5.

   (a) If u has four weak neighbors of degree 5, then we may assume they are v_4, v_5, v_7 and v_8. The vertices v_4 and v_8 are not weak or $(5, 7^+)$-neighbors of u, and v_5 and v_7 are not weak or $(5, 6^+)$-neighbors of u. Thus u loses at most $\frac{2}{3} + 2 \times \frac{1}{6} + 2 \times \frac{1}{2} = 2$ by R_7 and R_9.

   (b) If u has three weak neighbors of degree 5 that do not form a triangular face with u, then they are necessarily v_4, v_6 and v_8. Then v_4 and v_8 are $(5^+, 7^+)$-neighbors of u and v_6 is a $(5^+, 6^+)$-neighbor of u. Thus u loses at most $\frac{2}{3} + 2 \times \frac{1}{3} + \frac{2}{3} = 2$ by R_7 and R_9.

   (c) If u has three weak neighbors of degree 5 and there is a triangular face containing u and two of them, then we may assume up to symmetry that v_4 is one of these vertices. Then, if v_5 is also a weak neighbor of u of degree 5, thus u gives $\frac{1}{6}$ to v_4, at most $\frac{1}{2}$ to v_5 and at most $\frac{2}{3}$ to the remaining neighbor by R_9. Otherwise, if v_4 is a $(5, 6)$-neighbor of u, then v_7 and v_8 are weak neghbors of degree 5 of u. Then u gives at most $2 \times \frac{1}{2}$ to v_4, v_7 and at most $\frac{1}{6}$ to v_8, for a total loss of $\frac{11}{6} < 2$. Otherwise, u gives at most $\frac{1}{3}$ to v_4 and at most $2 \times \frac{1}{2}$ to the other neighbors by R_9. In each case, the total loss for u is at most $\frac{2}{3} + \frac{4}{3} = 2$.

   (d) If u has at most two weak neighbors of degree 5 and they are both $(6, 6)$-neighbor of u, then u loses at most $\frac{2}{3} + 2 \times \frac{2}{3} = 2$ by R_7 and R_9 since there is no room for another weak or semi-weak neighbor. If only one of them is a $(6, 6)$-neighbor, then there is room for one semi-weak neighbor of degree 4, hence u loses at most $\frac{2}{3} + \frac{2}{3} + \frac{1}{2} + \frac{1}{12} = \frac{23}{12} < 2$.

   (e) If u has at most two weak neighbors of degree 5 that are not $(6, 6)$-neighbors. Then u gives them at most $2 \times \frac{1}{2}$ by R_9. Moreover, the neighborhood of u has room for at most two semi-weak vertices of degree 4. The final loss of u is thus at most $\frac{2}{3} + 2 \times \frac{1}{2} + 2 \times \frac{1}{12} = \frac{11}{6} < 2$ by R_7 and R_9.

6. If $\delta = 3$ and u has two weak neighbors of degree 3, then by C_{18}, the vertex u has no other neighbor of degree at most 5. Therefore, the total loss is $2 \times 1 = 2$ by R_5.

7. If $\delta = 3$ and u has one weak neighbor of degree 3 and at least one semi-weak neighbor of degree 3, then u gives them $1 + \frac{1}{2}$ by R_5 and R_6. By C_{19}, we know that u gives weight to at most one other vertex v. Moreover,

   (a) if $d(v) = 3$, it is not a weak neighbor of u by hypothesis, hence u gives at most $\frac{1}{2}$ to v by R_6.

(b) if $d(v) = 4$, then by C_{19}, the other neighbors of u have degree 8, hence u gives at most $\frac{1}{2}$ to v by R_7.

(c) if $d(v) = 5$, then by C_{19}, the other neighbors of u have degree at least 7, hence u gives at most $\frac{1}{3}$ to v by R_9.

The total loss of u is thus at most $1 + \frac{1}{2} + \frac{1}{2} = 2$.

8. Assume that $\delta = 3$ and u has one weak neighbor of degree 3, no semi-weak neighbor of degree 3 and at least one neighbor v of degree 4.

If u has another weak neighbor of degree 4, then C_{21} ensures that all the other neighbors of u have degree 8. Therefore, u loses $1 + 2 \times \frac{1}{2} = 2$ by R_5 and R_7. We may thus assume that u has a single weak neighbor of degree 4.

Note that due to C_4, u has at most two weak neighbors of degree 5.

(a) If u has two weak neighbors of degree 5 w_1, w_2, these are not $(6^-, 6)$-neighbors of u by C_{17}. Assume that they form a triangular face with u.

Therefore, then u gives at most $2\times\frac{1}{6}$ to them by R_9. Moreover, there is no room for another semi-weak neighbor of degree 4. Therefore, the total loss is $1 + \frac{2}{3} + 2 \times \frac{1}{6} = 2$ by R_5, R_7 and R_9.

(b) If u has two weak neighbors of degree 5 w_1, w_2, that do not form a triangular face with u. By C_{16}, v is an $(8, 8)$-neighbor of u, hence receives $\frac{1}{2}$ from u. Moreover, by C_{17}, none of w_1, w_2 are $(6^-, 6)$-neighbors or u, nor E_3-neighbors, hence they receive at most $2 \times \frac{1}{4}$. The total loss is then $1 + \frac{1}{2} + 2 \times \frac{1}{4} = 2$.

(c) If u has a single weak neighbor of degree 5, then it is not a $(5, 6)$-nor a $(6, 6)$-neighbor of u by C_{17}. Moreover, if u has a semi-weak neighbor of degree 4, then by C_{20}, every other neighbor of u has degree 8. Hence R_8 does not apply, and the total loss is $1+\frac{1}{2}+\frac{1}{3} = \frac{11}{6} < 2$ by R_5, R_7 and R_9.
If u has no semi-weak neighbor of degree 4, then u loses $1+\frac{2}{3}+\frac{1}{3} = 2$ by R_5, R_7 and R_9.

(d) If u has no weak neighbor of degree 5, then it may give weight only to its semi-weak neighbors of degree 4. There are at most three of them, hence u loses at most $1 + \frac{2}{3} + 3 \times \frac{1}{12} = \frac{23}{12} < 2$.

9. Assume that $\delta = 3$ and u has one weak neighbor of degree 3 (say v_2) but no semi-weak neighbor of degree 3, and no neighbor of degree 4.

(a) If u has only one weak vertex of degree 5, then u loses at most $1 + \frac{2}{3} = \frac{5}{3} < 2$ by R_5 and R_9.

(b) If u has at most two weak vertices x and y of degree 5, and x is a $(6,6)$-neighbor of u, then by C_{15}, the vertex y is not a $(6^-,6)$-neighbor of u. Thus u loses at most $1+\frac{2}{3}+\frac{1}{3}=2$ by R_5 and R_9.

(c) If u has at most two weak vertices x and y of degree 5 and none of them is a $(6,6)$-neighbor of u, then both receive at most $\frac{1}{2}$ by R_9. Therefore, u loses $1+2\times\frac{1}{2}=2$.

(d) If u has three weak vertices of degree 5, then by C_{15}, none of them is a $(6,6)$-neighbor of u. If none of them is a $(5,6)$-neighbor of u, then the total loss of u is at most $1+3\times\frac{1}{3}=2$ by R_5 and R_9. Otherwise, note that $d(v_1)=d(v_3)=8$ by C_1, hence v_4 and v_8 are not $(5,6)$-neighbors of u.

i. If v_6 is a $(5,6)$-neighbor of u, then we may assume that $d(v_4)=d(v_6)=d(v_7)=5$, $d(v_5)=6$ and $v_iv_{i+1}\in E(G)$ for $1\leqslant i\leqslant 7$. By C_{15}, we have $d(v_8)>6$, so u gives $\frac{1}{2}$ to v_6, $\frac{1}{3}$ to v_4 and $\frac{1}{6}$ to v_7 by R_9.

ii. Assume that v_5 (or v_7 by symmetry) is a $(5,6)$-neighbor of u and $(d(v_4),d(v_6))=(5,6)$. Note that due to C_{15}, v_7 is not a $(5,6)$-neighbor of u, thus v_4 is a weak neighbor of u. Indeed, otherwise, the three weak neighbors of u of degree 5 would be v_5,v_7 and v_8 so v_7 would be a $(5,6)$-neighbor of u. Thus u gives $\frac{1}{6}$ to v_4, $\frac{1}{2}$ to v_5 and $\frac{1}{3}$ to v_7 or v_8 by R_9.

iii. Otherwise, v_5 is a $(5,6)$-neighbor of u and we have $d(v_6)=5$ and $d(v_4)=6$. Then v_6 and v_8 must be weak neighbors of u of degree 5. By C_{15}, we have $d(v_7)>6$. Therefore, u gives $\frac{1}{2}$ to v_5, $\frac{1}{6}$ to v_6 and $\frac{1}{3}$ to v_8 by R_9.

Therefore, in each case, the total loss for u is $1+\frac{1}{2}+\frac{1}{3}+\frac{1}{6}=2$.

(e) If u has four weak vertices of degree 5, then by C_4 we may assume that $v_1\cdots v_8$ is a cycle in G and that $d(v_1)=d(v_3)=8$, $d(v_2)=3$ and $d(v_4)=d(v_5)=d(v_7)=d(v_8)=5$. By C_{15}, we have $d(v_6)>6$, hence u loses $1+4\times\frac{1}{6}<2$ by R_5 and R_9.

10. Assume that $\delta=3$ and u has no weak neighbor of degree 3. By C_{22}, if the vertex u has three semi-weak neighbors of degree 3, then every other neighbor of u has degree 8. In this case, u loses at most $3\times\frac{1}{2}=\frac{3}{2}<2$.

We may thus assume that u has at most two semi-weak neighbors of degree 3. Moreover, due to C_2, two semi-weak neighbors of u of degree 3 are at triangle-distance at least 3 in the neighborhood of u.

(a) If u has two semi-weak neighbors of degree 3, then by C_{22}, the vertex u has at most two neighbors of degree 4. Moreover, if there are exactly two such vertices, the other neighbors of u have degree 8.

i. If u has two neighbors of degree 4, then u loses at most $2 \times \frac{1}{2} + 2 \times \frac{1}{2} = 2$ by R_6 and R_7.

ii. Otherwise, if u has exactly one weak neighbor of degree 4, at triangle-distance 2 from both semi-weak neighbors of degree 3, then we may assume that $d(v_1) = d(v_5) = 3$, $d(v_2) = d(v_4) = 8$, $d(v_3) = 4$, $v_8v_1 \notin E(G)$, $v_5v_6 \notin E(G)$ and $v_iv_{i+1} \in E(G)$ for $1 \leqslant i \leqslant 4$. By (i), u has no other neighbor of degree 4.
Observe that only v_7 can be a weak neighbor of u, and in this case it has degree 5. If v_7 is a $(6^-, 6)$-neighbor of u, then u gives at most $\frac{2}{3}$ to v_7 by R_9 but receives $\frac{5}{12}$ by R_2 thus u loses $2 \times \frac{1}{2} + \frac{1}{2} + \frac{2}{3} - \frac{5}{12} = \frac{7}{4} < 2$. Otherwise, u gives at most $\frac{1}{3}$ to v_7 by R_9 for a total loss of $\frac{5}{3} < 2$.

iii. If u has exactly one weak neighbor of degree 4, at triangle-distance at least 3 from one of the semi-weak neighbors of degree 3, then u gives weight to at most two other vertices. Either there is only one such vertex and it is a $(6^+, 7^+)$-neighbor of degree 5, or there are two such vertices and they are $(5, 7^+)$-neighbors of degree 5. In both cases, u loses at most $2 \times \frac{1}{2} + \frac{2}{3} + \frac{1}{3} = 2$ by R_6, R_7 and R_9.

iv. If u has no weak neighbor of degree 4 and three weak neighbors of degree 5, then they are a $(5, 8)$-, a $(5, 6^+)$- and a $(6^+, 8)$-neighbor of u. Then u loses at most $2 \times \frac{1}{2} + \frac{1}{6} + \frac{1}{2} + \frac{1}{3} = 2$ by R_6 and R_9.

v. Assume that u has no weak neighbor of degree 4 and two weak neighbors of degree 5. If u has no $(6, 6)$-neighbor, it loses at most $2 \times \frac{1}{2} + 2 \times \frac{1}{2} = 2$ by R_6 and R_9. Otherwise u has a $(6, 6)$-neighbor and the other weak neighbor of degree 5 is a $(5^+, 8)$-neighbor of u so u loses at most $2 \times \frac{1}{2} + \frac{2}{3} + \frac{1}{3} = 2$ by R_6 and R_9.

vi. If u has no weak neighbor of degree 4 and at most one weak neighbor of degree 5, then u gives at most $2 \times \frac{1}{2} + \frac{2}{3} = \frac{5}{3} < 2$ by R_6 and R_9.

(b) If u has one semi-weak neighbor of degree 3 and three weak neighbors of degree 4, then one of them is a $(7^+, 8)$-neighbor of u and the two others are $(7^+, 7^+)$-neighbors, hence u gives at most $\frac{1}{2} + \frac{7}{12} + 2 \times \frac{2}{3}$ by R_6 and R_9 but receives $\frac{5}{12}$ by R_2, hence the total loss is at most 2.

(c) If u has one semi-weak neighbor of degree 3 and two weak neighbors w_1 and w_2 of degree 4, then there is room for only one other vertex

v receiving weight from u. If there is no such vertex or if it is a semi-weak neighbor of u, then u loses at most $\frac{1}{2} + 2 \times \frac{2}{3} + \frac{1}{12} = \frac{23}{12} < 2$ by R_6 and R_9.

Otherwise, v is a weak neighbor of u. If w_1 and w_2 are $(7,7)$-neighbors of u, then v is a $(7,8)$-neighbor of u and moreover R_2 applies. Therefore, the total loss is $\frac{1}{2} + 2 \times \frac{2}{3} + \frac{1}{3} - \frac{5}{12} = \frac{21}{12} < 2$.

Otherwise, w_2 is a $(7^+,8)$-neighbor of u. so u gives at most $\frac{7}{12} + \frac{2}{3}$ to w_1, w_2 by R_7.

Note that v cannot be a $(6^-,6)$-neighbor of u. If it is not a $(5,7^+)$-neighbor of u, then u gives it $\frac{1}{3}$ by R_9 but receives $\frac{5}{12}$ by R_2. Otherwise, u gives $\frac{1}{6}$ by R_9. In both cases, u gives at most $\frac{1}{2} + \frac{7}{12} + \frac{2}{3} + \frac{1}{6} = \frac{23}{12} < 2$.

(d) If u has one semi-weak neighbor x of degree 3 and one weak neighbor of degree 4 at triangle-distance 2 from x, then there are at most two weak neighbors of u of degree 5. Moreover, if there is one such neighbor, then u has at most one semi-weak neighbor of degree 4 hence u loses at most $\frac{1}{2} + \frac{7}{12} + \frac{2}{3} + \frac{1}{12} = \frac{11}{6} < 2$ by R_6, R_7, R_9 and R_8.

If there is no weak neighbor of u of degree 5, then u has at most two semi-weak neighbors of degree 4 hence u loses at most $\frac{1}{2} + \frac{7}{12} + 2 \times \frac{1}{12} = \frac{5}{4} < 2$ by R_6, R_7 and R_8. We may thus assume that there are two weak neighbors of u of degree 5.

i. If one of them is a $(6,6)$-neighbor of u, then the other one is a $(6,7^+)$-neighbor and there is no semi-weak neighbor of u of degree 4, hence u loses $\frac{1}{2} + \frac{7}{12} + \frac{2}{3} + \frac{1}{3}$ by R_6, R_7 and R_9 and receives $\frac{5}{12}$ by R_2, for a total loss of at most $\frac{5}{3} < 2$.

ii. If both of them are $(5,6)$-neighbors of u, then u has no semi-weak neighbor of degree 4, hence it loses at most $\frac{1}{2} + \frac{7}{12} + 2 \times \frac{1}{2}$ by R_6, R_7 and R_9 and receives $\frac{5}{12}$ by R_2, for a total loss of $\frac{5}{3} < 2$.

iii. If one of them is a $(5,6)$-neighbor of u, and the other one is a $(5,7^+)$-neighbor of u. Then it gives at most $\frac{1}{2} + \frac{1}{6}$ to them by R_9, and there is at most one semi-weak neighbor of u of degree 4. The total loss is thus at most $\frac{1}{2} + \frac{7}{12} + \frac{1}{2} + \frac{1}{6} + \frac{1}{12} = \frac{11}{6} < 2$. If the other one is a $(6,7^+)$-neighbor, then there is no room for a semi-weak vertex of degree 4, hence the total loss is $\frac{1}{2} + \frac{7}{12} + \frac{1}{2} + \frac{1}{3} = \frac{23}{12} < 2$.

iv. Otherwise, both of them are $(6,7^+)$-neighbors of u, so u has no semi-weak neighbor of degree 4, hence it loses at most $\frac{1}{2} + \frac{7}{12} + 2 \times \frac{1}{3} = \frac{7}{4} < 2$ by R_6, R_7 and R_9.

(e) If u has one semi-weak neighbor x of degree 3 and one weak neighbor

y of degree 4 at triangle-distance at least 3 from x, then u has at most two other weak neighbors of degree 5, and moreover, they are not both $(5,6)$-neighbors of u.

i. If u has a $(6,6)$-neighbor of degree 5, then u gives weight only to x, y and this vertex, and the loss is at most $\frac{1}{2}+\frac{2}{3}+\frac{2}{3}=\frac{11}{6}<2$ by R_6, R_7 and R_9.

ii. If u has a $(5,6)$-neighbor of degree 5, then the other weak neighbor of u of degree 5 is a $(5,7^+)$-neighbor. Hence u loses at most $\frac{1}{2}+\frac{2}{3}+\frac{1}{2}+\frac{1}{6}=\frac{11}{6}<2$ by R_6, R_7 and R_9.

iii. Otherwise, u gives at most $2\times\frac{1}{3}$ to its weak neighbors of degree 5. Moreover, if u has two such neighbors, then it has no semi-weak neighbor of degree 4, hence u loses at most $\frac{1}{2}+\frac{2}{3}+2\times\frac{1}{3}=\frac{11}{6}<2$ by R_6, R_7 and R_9.
If u has only one weak neighbor of degree 5, then it has at most one semi-weak neighbor of degree 4, hence it loses at most $\frac{1}{2}+\frac{2}{3}+\frac{1}{3}+\frac{1}{12}=\frac{19}{12}<2$ by R_6, R_7, R_8 and R_9.
Finally, if u has no weak neighbor of degree 5, it has at most three semi-weak neighbors of degree 4, hence it loses at most $\frac{1}{2}+\frac{2}{3}+3\times\frac{1}{12}=\frac{17}{12}<2$ by R_6, R_7 and R_8.

(f) If u has one semi-weak neighbor of degree 3 and no weak neighbor of degree 4, then note that u has at most four weak neighbors of degree 5.

i. If u has at most one weak neighbor of degree 5, then it has at most four semi-weak neighbors of degree 4, for a loss of $\frac{1}{2}+\frac{2}{3}+4\times\frac{1}{12}=\frac{3}{2}<2$ by R_6, R_8 and R_9.

ii. If u has two weak neighbors of degree 5, then u has at most two semi-weak neighbors of degree 4, hence u loses at most $\frac{1}{2}+2\times\frac{2}{3}+2\times\frac{1}{12}=2$ by R_6, R_8 and R_9.

iii. If u has three weak neighbors of degree 5 such that there is no triangular face containing u and two of them, then we may assume that $d(v_1)=3$, $d(v_2)=8$, $d(v_3)=d(v_5)=d(v_7)=5$ and $v_iv_{i+1}\in E(G)$ for $1\leqslant i\leqslant 7$.
Then u gives $\frac{1}{2}$ to v_1 by R_6, at most $\frac{1}{3}$ to v_3 and at most $\frac{2}{3}$ to v_5 by R_9. Moreover, if $d(v_8)=5$, then u gives at most $\frac{1}{2}$ to v_7 by R_9, hence the total loss is at most 2. Otherwise, u gives at most $\frac{2}{3}$ to v_7 by R_9, but receives $\frac{5}{12}$ by R_2, for a total loss of at most $\frac{7}{4}<2$.

iv. If u has three weak neighbors of degree 5 such that there is a triangular face containing u and two of them, then u has at most one semi-weak neighbor of degree 4.
If u has such a neighbor, then we necessarily have $d(v_1)=3$, $d(v_2)=8$, $d(v_3)=d(v_6)=5$, $d(v_8)=4$ and either $d(v_4)=5$ or

$d(v_5) = 5$. In the first case, u gives $\frac{1}{2}$ to v_1 by R_6, $\frac{1}{2}$ to v_4, $\frac{1}{6}$ to v_3, at most $\frac{2}{3}$ to v_6 by R_9 and at most $\frac{1}{12}$ to v_8 by R_8, for a total loss of $\frac{23}{12} < 2$. In the latter case, u gives $\frac{1}{2}$ to v_1 by R_6, at most $\frac{1}{3}$ to v_3, at most $2 \times \frac{1}{2}$ to v_5 and v_6, by R_9 and at most $\frac{1}{12}$ to v_8 by R_8, for a total loss of $\frac{23}{12} < 2$.

The only case remaining is when u has no semi-weak neighbor of degree 4. Consider the non-triangular face containing u and its semi-weak neighbor of degree 3. If the next neighbor (say v_8) has degree at least 6, then u receives $\frac{5}{12}$ by R_2. In this case, the loss of u is at most $\frac{1}{2} + \frac{2}{3} + 2 \times \frac{1}{2} - \frac{5}{12} = \frac{7}{4} < 2$ by R_6 and R_9.

Note that if $d(v_8) < 6$ and u has a $(6,6)$-neighbor, then we must have $d(v_1) = 3$, $d(v_2) = 8$, $d(v_3) = d(v_4) = d(v_6) = 5$, $d(v_5) = d(v_7) = 6$ and $v_i v_{i+1} \in E(G)$ for $1 \leqslant i \leqslant 7$. Thus u gives $\frac{1}{2}$ to v_1 by R_6, $\frac{1}{6}$ to v_3, at most $\frac{1}{2}$ to v_4 and $\frac{2}{3}$ to v_6 by R_9. The total loss is then at most $\frac{11}{6} < 2$.

If $d(v_8) < 6$ and u has no $(6,6)$-neighbor, then u gives $\frac{1}{2}$ to v_1 by R_6 and at most $3 \times \frac{1}{2}$ to its weak neighbors of degree 5 by R_9. The total loss is at most $4 \times \frac{1}{2} = 2$.

v. If u has four weak neighbors of degree 5, then we may assume that $d(v_1) = 3$, $d(v_2) = 8$, $d(v_3) = d(v_4) = d(v_6) = d(v_7) = 5$ and that for $1 \leqslant i \leqslant 7$, $v_i v_{i+1} \in E(G)$.

Then u gives $\frac{1}{2}$ to v_1 by R_6, $\frac{1}{6}$ to v_3 and at most $\frac{1}{2}$ to v_4, v_6 and v_7 by R_9. Moreover, by R_2, u receives $\frac{5}{12}$, for a total loss of at most $\frac{7}{4} < 2$.

(g) Assume that u has no semi-weak neighbor of degree 3. Since $\delta = 3$, there is a neighbor of u of degree 3, but it is not weak nor semi-weak. Therefore, there are at most three weak neighbors of u of degree 4.

i. If u has three weak neighbors of degree 4, then there is no other weak or semi-weak neighbor of u, hence u loses at most $3 \times \frac{2}{3} = 2$.

ii. If u has two weak neighbors of degree 4, then there is room for one weak vertex of degree 5 or two semi-weak neighbors of degree 4. In both cases, the total loss is at most $2 \times \frac{2}{3} + \frac{2}{3} = 2$ or $2 \times \frac{2}{3} + 2 \times \frac{1}{12} = \frac{3}{2} < 2$.

iii. If u has one weak neighbor of degree 4, there are at most two weak neighbors of u of degree 5, and one of them is not a $(6,6)$-neighbor by C_1. Moreover, there can be at most one semi-weak neighbor of degree 4. Therefore, u loses at most $\frac{2}{3} + \frac{2}{3} + \frac{1}{2} + \frac{1}{12} = \frac{23}{12} < 2$.

iv. If u has no weak neighbor of degree 4, by C_4, there are at most four weak neighbors of u of degree 5.

A. If there are four of them, they cannot be $(6,6)$-neighbors and there is no room for any semi-weak neighbor, hence the total loss is $4 \times \frac{1}{2} = 2$.

B. If there are three weak neighbors of degree 5, and at least two of them are $(6,6)$-neighbors, then there is no room for another semi-weak neighbor, hence the total loss is $3 \times \frac{2}{3} = 2$.

C. If there are three weak neighbors of degree 5, and one of them is a $(6,6)$-neighbor, then there is room for a single semi-weak vertex, hence the total loss is at most $\frac{2}{3} + 2 \times \frac{1}{2} + \frac{1}{12} = \frac{7}{4} < 2$.

D. If there are three weak neighbors of degree 5 that are not $(6,6)$-neighbors, there are at most four semi-weak neighbors of degree 4, for a total loss of at most $3 \times \frac{1}{2} + 4 \times \frac{1}{12} = \frac{11}{6} < 2$.

E. If there are at most two weak neighbors of degree 5, then there are at most five semi-weak neighbors of degree 4, for a total loss of at most $2 \times \frac{2}{3} + 5 \times \frac{1}{12} = \frac{21}{12} < 2$.

1.6 Open questions

To prove our result, we used the discharging method. In this case, we had a lot of configurations to reduce. The key ideas came from the two approaches we used to reduce them. While the Combinatorial Nullstellensatz and recoloring approach have already been used many times in discharging proofs, the framework we present here (the so-called color shifting graph) seems to be quite new. To our knowledge, it was first used in [Bonamy, 2015]. Here we designed a more generic framework to use this idea. This allowed us to reduce some configurations we could not tackle in an usual way. However, it has still limited use since we designed a framework that allows us to recolor only a set of pairwise adjacent elements. It would be interesting to improve it to get rid of this limitation. Moreover, while total 9-choosability seems out of reach from a reasonable discharging proof, it would also be interesting to see if the methods we introduced here can help to prove that $\chi''_\ell = \Delta + 1$ for planar graphs when Δ can be less than 12.

We may also wonder whether our methods may apply to more generic colorings like correspondence coloring. First, the polynomials we designed for the Nullstellensatz approach are designed specifically for list coloring: we handle the colors globally. Indeed, we evaluate the variables on the colors (represented by integers), which means that all the vertices have to agree on which color is represented by each integer. This is something we cannot use for correspondence coloring, since the vertices agree only locally on the definition of the colors. Thus, our polynomial approach does not directly extend to other colorings. However, if we consider other polynomials, we may succeed in encapsulating the new constraints. The price to pay comes from the degree of such polynomials, which may lead to too heavy computations, making this method useless. The recoloring approach suffers from the same kind of "localization" problems in its current definition. However, since we use only local constraints to define color shifting graphs, the definition should extend to the correspondence coloring setting.

From a more algorithmic point of view, we can observe that, like most of the discharging proofs, our result comes along with a linear-time algorithm to find a proper coloring of a graph G given a list assignment. Indeed, we first apply the discharging rules to G in linear time. Using the elements with negative final weight, we can identify the reducible configurations in G. Then, we color recursively the graph obtained by removing one reducible configuration. However, instead of moving again the weights, we keep track of what happens when we remove the configuration.

To extend the obtained coloring, observe that both the case analysis and the recoloring approaches lead to a constant time process. For the Nullstellensatz approach, it is trickier since the proofs are not constructive. However, we can use a preprocessing step that compute a proper coloring for each configuration and for each list assignment just by brute-force. Since the sizes of the configurations are bounded by a constant (except for C_2, but we only used it for removing some cycles of length 4), this also takes constant time. Each recursive step thus takes constant time and we have at most one such step for each element of G. Therefore, if we add the initial discharging phase (which also takes linear time), we obtain a linear time algorithm.

Chapter 2

Discharging without discharging: the power of pigeons

Contents

This chapter contains two results. The first one is a sole-author result ([Pierron, 2019]). We study the girth and the diameter of any graph not satisfying the corresponding bound, in the spirit of [Bonamy and Bousquet, 2014]. However, to improve their result, we consider more relaxed configurations, which require more precise arguments. The second result is joint work with Ilkyoo Choi and Daniel W. Cranston ([Choi et al., 2018]). The case of planar graphs of girth at least 5 was settled by [Bonamy et al., 2019a], here *we consider C_4-free planar graphs. We adapt their approach to allow triangles: the general proof scheme is similar. To handle triangles, we introduce new configurations. We then provide a more careful structural analysis of the regions: both to find large ones (loops are now possible) and to be able to reduce them.*

2.1 Introduction

Graph colorings have classic but still important applications in telecommunication networks optimization. Such a network can be represented by a graph whose vertices are elements (phones, antennas...), and two elements are adjacent if they are able to communicate directly. In this model, all the elements send their messages using some frequencies. However, to avoid interferences, there must be a condition on the distance between two elements using the same frequency. Minimizing the total number of frequencies (and hence the cost of the network) then corresponds to solving a well-designed coloring problem on the graph that models the network.

Several parameters have an influence on interferences. We are interested in two such parameters: the distance between the antennas, and the gap between frequencies. Depending on the distance conditions we require, we obtain several coloring problems. For example, in the original coloring problem, we forbid two adjacent vertices to receive the same color, which corresponds to forbidding interferences at distance 1. A natural refinement of this problem is to forbid interferences at distance k, meaning that any two vertices within distance k of each other have to receive different colors. When the value of k is 2, we obtain the so-called *square coloring* problem. The name comes from the fact that it is equivalent to finding a proper vertex coloring of the *square* of the network, i.e. the graph obtained by adding an edge between every pair of non-adjacent vertices sharing a common neighbor. More generally, we can define the *k-th power G^k* of a graph G, by adding edges between every pair of vertices within distance k in G.

In this chapter, we are interested in colorings of graph powers. First note that the chromatic numbers of powers of paths and cycles are already known (see [Prowse and Woodall, 2003]), hence we may always assume that $\Delta(G) \geqslant 3$ when coloring powers of G.

For coloring squares of graphs, note that $\Delta(G^2) \leqslant \Delta(G)^2$, hence Brooks' theorem (Theorem 1.1) shows that $\chi(G^2) \leqslant \Delta(G)^2$, except when G^2 is a clique. Graphs G such that G^2 is a clique on $\Delta(G)^2+1$ vertices are known as *Moore graphs*, and there are only finitely many of them (see [Hoffman and Singleton, 1960], or [Hoffman and Singleton, 2003] for a more recent version). This $\Delta(G)^2$ bound has been improved by one, first for subcubic graphs [Cranston and Kim, 2008] and then in the generic case, as stated below in the more general setting of list coloring.

Theorem 2.1 ([Cranston and Rabern, 2016]). *If G is not a Moore graph, then $\chi_\ell(G^2) \leqslant \Delta(G)^2 - 1$.*

We thus obtain a gap of 2 from the upper bound $\Delta(G)^2 + 1$, except for at most four graphs. For higher values of k, say $k \geqslant 3$, the maximum possible value of $\Delta(G^k)$ is the number of nodes of a tree of height k whose internal nodes have degree $\Delta(G)$, without counting its root (see Figure 2.1). We denote this number by $f(k, \Delta(G))$, where

$$f(k, \Delta) = \Delta \sum_{i=0}^{k-1} (\Delta - 1)^i = \Delta \frac{(\Delta - 1)^k - 1}{\Delta - 2}.$$

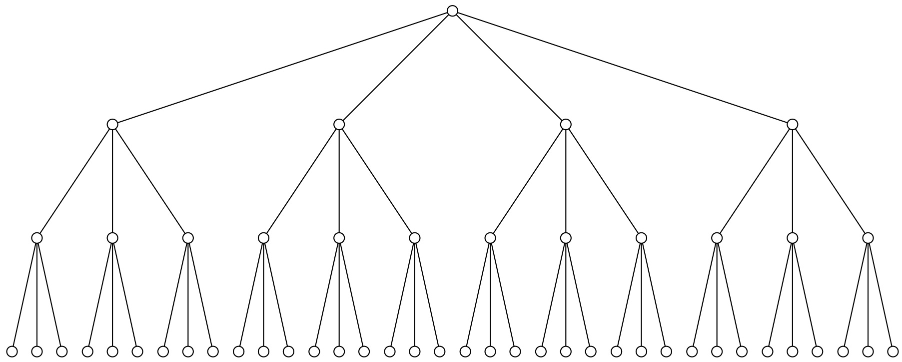


Figure 2.1 – A 4-ary tree of height 3, with $f(4, 3) = 52$ non-root nodes.

By Brooks' theorem, $f(k, \Delta)$ colors are sufficient to color the k-th power of any graph G with maximum degree Δ, as soon as it is not a generalized Moore graph, i.e. G^k is not a clique on $f(k, \Delta) + 1$ vertices. However, such a graph does not exist when $k \geqslant 3$ [Hoffman and Singleton, 1960]. Therefore, the bound $\chi(G^k) \leqslant f(k, \Delta)$ always holds, ensuring a gap of one from the upper bound $f(k, \Delta) + 1$. Moreover, a wider gap of 2 colors holds, see [Bonamy and Bousquet, 2014].

When $k = 2$, note that $f(2, \Delta) = \Delta^2$. Hence, together with Theorem 2.1, this result settles a conjecture of [Miao and Fan, 2014], stating that two colors can be spared from the naive upper bound $f(k, \Delta) + 1$ (i.e. the gap is at least two), except when $k = 2$ and G is a Moore graph. In [Bonamy and Bousquet, 2014], the authors conjecture that we can improve this result further, by obtaining a gap of k colors for higher values of k, except for a finite number of graphs.

Conjecture 2.2 ([Bonamy and Bousquet, 2014]). *For every $k \geqslant 2$, all but finitely many graphs G satisfy $\chi(G^k) \leqslant f(k, \Delta(G)) + 1 - k$.*

As a warm-up, we prove in Section 2.2 the following theorem, stating that most of the time, $k-2$ colors can be spared.

Theorem 2.3. *For every integer $k \in \mathbb{N}^*$ and every $\Delta \geqslant 3$, the k-th power of every graph of maximum degree Δ is $(f(k,\Delta)+3-k)$-colorable, except for finitely many of them.*

Note that even if we consider a counterexample (i.e. a graph of maximum degree $\Delta \geqslant 3$ whose k-th power is not $(f(k,\Delta)+3-k)$-colorable) and prove that it does not contain some configurations, the proof of Theorem 2.3 does not use induction whatsoever: for each configuration, instead of extending a coloring of a subgraph, we actually color the whole graph from scratch. The main result of this chapter is actually introduced hereafter.

When considering the frequency assignment problem, the distance between antennas is not the only parameter to be taken into account. Indeed, the interference between two signals with different frequencies depends on how close the frequencies are. Thus, we may also require the frequencies assigned to close enough vertices to differ by a constant gap. Thus, coloring powers of graphs can be seen as a special instance of the so-called $L(p_1,\ldots,p_n)$-labeling problem, introduced for $n=2$ in the seminal paper [Griggs and Yeh, 1992]. In this generalization, we ask for a vertex labeling where the labels of every pair of vertices at distance exactly i have to differ by at least p_i. The parameter we study is then the *span* of the labeling, i.e. the difference between the maximum label and the minimum one. Given some non-negative real numbers $p_1,\ldots,p_n$ and a graph G, we denote by $\lambda_G(p_1,\ldots,p_n)$ the minimum span of an $L(p_1,\ldots,p_n)$-labeling of G. Such a labeling exists for every finite graph (even for infinite graphs with bounded maximum degree, see [Griggs *et al.*, 2009]). Observe that, in this coloring, colors are integers, and we are interested in their difference. Therefore, colors are not symmetrical anymore: permuting the colors of a valid coloring may result in an invalid one.

Most of the results about $L(p_1,\ldots,p_n)$-labeling are limited to the case when $n=2$ and p_1,p_2 are non-negative integers. We refer to [Griggs *et al.*, 2009] for more results about the generic case. Note that the notion of $L(1,0)$-labeling coincides with proper vertex coloring and, similarly, $L(1,1)$-labeling is equivalent to square coloring. The case of $L(0,1)$-labeling has also been studied: for triangle-free graphs, it corresponds to the notion of injective coloring, introduced in [Hahn *et al.*, 2002] with some applications to error-correcting codes. An *injective coloring* is a vertex coloring which is not necessarily proper, but where every pair of vertices with a common neighbor receive different colors. This means that no vertex has two neighbors with the same color. The motivation for studying such a coloring does not come from interferences, but rather from the communication problem of identifying the sender of a message. With such a coloring, each vertex can identify which of its neighbors sent a message, only by considering its frequency. Many results are known about injective coloring (see [Doyon *et al.*, 2010; Chen *et al.*, 2012; Bu *et al.*, 2009; Lužar *et al.*, 2009; Bu *et al.*, 2015]). However, note that a square coloring is an injective coloring. Hence, most of these results can be strengthened using the results about square coloring we cite below.

Consider two signals with close frequencies. If they are emitted from close enough antennas, they will interfere. However, the further away the antennas are, the less they interfere. This is why we often ask for stronger conditions on closer vertices (with the main exception of injective coloring, where the problem does not come from interferences, but from identification of the sender). Apart from the standard coloring, square coloring and injective coloring problems, the most studied $L(p,q)$-labeling is $L(2,1)$-labeling, also called *radiocoloring*. This problem has a long-standing history, and many results bound $\lambda_G(p,q)$ with respect to several parameters of G, like its maximum degree Δ or its chromatic number χ, see for example [Griggs and Yeh, 1992; Jonas, 1993; Chang and Kuo, 1996; Král' and Škrekovski, 2003; Gonçalves, 2008; Havet *et al.*, 2012]. Just to cite a few results, we have, for any graph G,

$$\lambda_G(2,1) \leqslant \chi(G) + |V(G)| - 2,$$

which is tight for complete k-partite graphs [Griggs and Yeh, 1992]. Moreover, $\lambda_G(2,1) \leqslant \Delta^2 + \Delta - 2$ when $\Delta \geqslant 3$ [Gonçalves, 2008]. We can even strengthen this result by considering $L(p,1)$-labeling [Havet *et al.*, 2012]:

$$\lambda_G(p,1) \leqslant \Delta^2 + C_p$$

where C_p is a constant depending only on p. Moreover, we can take $C_p = 0$ as soon as $\Delta \geqslant \Delta_p$, where Δ_p only depends on p [Havet *et al.*, 2012]. The bound $O(\Delta^2)$ is tight: for every Δ, there exist some graphs all of whose $L(2,1)$-labelings span at least $\Delta^2 - \Delta$ labels, see [Griggs and Yeh, 1992].

Many results are also known for subclasses of graphs, (chordal [Sakai, 1994], outerplanar [Lih *et al.*, 2006], planar [Wegner, 1977; Jonas, 1993; Cranston and West, 2017; Wong, 1996; van den Heuvel and McGuinness, 2003; Agnarsson and Halldórsson, 2003; Borodin *et al.*, 2002; Molloy and Salavatipour, 2005; Amini *et al.*, 2013]...), see [Calamoneri, 2011] for a detailed survey. In this chapter, we also consider colorings of planar graphs. The first known result about $L(p,q)$-coloring of planar graphs comes from [van den Heuvel and McGuinness, 2003], with the upper bound

$$\lambda_G(p,q) \leqslant (4q-2)\Delta + 10p + 38q - 24$$

when $p \geqslant q$, as well as a construction giving a $\frac{3q\Delta}{2} + O(p,q)$ lower bound. This has been refined several times in [Borodin *et al.*, 2002; Molloy and Salavatipour, 2005] up to the actual best known result

$$\lambda_G(p,q) \leqslant q\left\lceil \frac{5\Delta}{3} \right\rceil + 18p + 77q - 18.$$

Since $\chi(G^2) = 1 + \lambda_G(1,1)$, we can rewrite these results for the square coloring problem:

$$\chi(G^2) \leqslant \left\lceil \frac{5\Delta}{3} \right\rceil + 78.$$

However, in this case, this bound can be improved. For example, the constant 78 can be decreased to 25 for large enough Δ [Molloy and Salavatipour, 2005]. This is still far from the conjectured result of [Wegner, 1977].

Conjecture 2.4 ([Wegner, 1977]). *If G is planar, then*

$$\chi(G^2) \leqslant \begin{cases} 7 & \textit{if } \Delta = 3 \\ \Delta + 5 & \textit{if } 4 \leqslant \Delta \leqslant 7 \\ \lfloor \frac{3\Delta}{2} \rfloor + 1 & \textit{otherwise} \end{cases}$$

Except for the case $\Delta = 3$ proved in [Thomassen, 2018], this conjecture remains widely open. However, for large enough Δ, the previous bound has been improved in [Amini *et al.*, 2013; Havet *et al.*, 2007] to the following result.

Theorem 2.5 ([Amini *et al.*, 2013; Havet *et al.*, 2007]). *For every surface S, every graph embeddable in S satisfies $\chi_\ell(G^2) \leqslant \frac{3}{2}\Delta + o(\Delta)$.*

This result thus proves the conjecture for large enough Δ. Moreover, this bound is tight, as shown by the following construction [Wegner, 1977], known as Shannon's triangle.

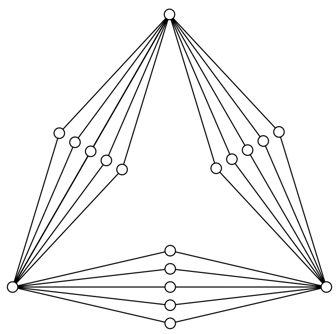


Figure 2.2 – Wegner's construction

To our knowledge, the Shannon triangles seem to be the only family of graphs with unbounded maximum degree achieving the $\frac{3\Delta}{2}$ bound. An interesting question is then whether a better upper bound can be achieved when considering some subclasses of planar graphs. When dealing with planar graphs, natural subclasses are obtained by considering girth restrictions. Wegner's construction has girth 4, then considering triangle-free planar graphs is not enough. However, for higher girths, it is possible to prove that $\Delta + O(1)$ colors are sufficient, as shown in [Wang and Lih, 2003]:

Theorem 2.6 ([Wang and Lih, 2003]). *If G is planar of girth g, then*

- $\chi(G^2) \leqslant \Delta + 5$ *when* $g \geqslant 7$,
- $\chi(G^2) \leqslant \Delta + 10$ *when* $g \geqslant 6$,
- $\chi(G^2) \leqslant \Delta + 16$ *when* $g \geqslant 5$.

These results can be refined for large enough Δ: for $g \geqslant 7$, we have $\chi(G^2) = \Delta + 1$ if $\Delta \geqslant 30$ [Borodin *et al.*, 2004]. Some examples of graphs of girth 5 and 6 are known to require at least $\Delta + 2$ colors, see Figure 2.3.

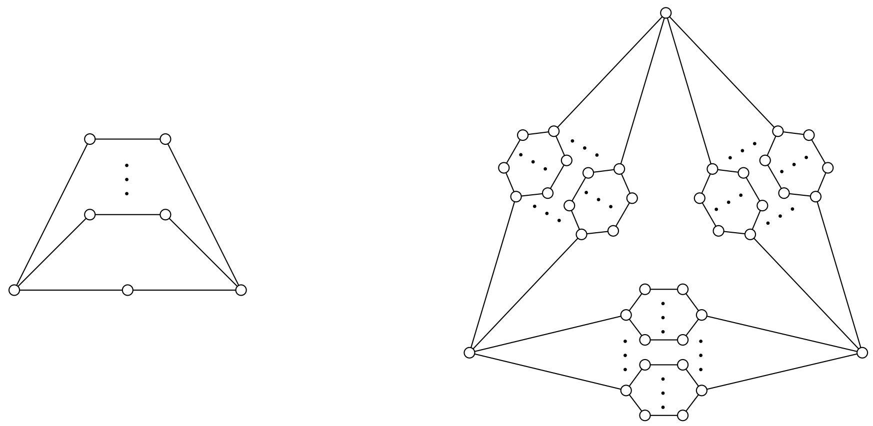


Figure 2.3 – Planar graphs with girth 5 and 6 such that $\chi(G^2) > \Delta + 1$.

However, it is proved in [Dvořák *et al.*, 2008; Borodin and Ivanova, 2009a; Bonamy *et al.*, 2019a] that $\Delta + 2$ are sufficient for planar graphs with girth 5 and 6 and large enough maximum degree, even when considering list coloring [Borodin and Ivanova, 2009b; Bonamy *et al.*, 2019a]. These results have been extended in [Bonamy *et al.*, 2014], by considering restrictions on the maximum average degree instead of the girth.

Some results are known when considering some other cycle obstructions. For example, for planar graphs without C_4 and C_5, a $\Delta + 7$ upper bound holds, see [Zhu *et al.*, 2012], which can be strengthened to $\Delta + 2$ for large enough Δ [Dong and Xu, 2017]. When only cycles of length 4 are forbidden, $\Delta + O(1)$ colors are sufficient, see [Wang and Cai, 2008]. This bound was extended for list coloring in [Choi *et al.*, 2018] using the discharging method.

Proposition 2.7 ([Choi *et al.*, 2018]). *The square of every C_4-free planar graph G is $(\Delta + 72)$-degenerate, and hence $\chi_\ell(G^2) \leqslant \Delta + 73$.*

Looking for cycle obstructions is not a new concept. For example, for proper vertex coloring of planar graphs, four colors are known to be sufficient [Appel *et al.*, 1977]. However, many attempts have been made for finding cycle obstructions to break down this bound to 3. A seminal result comes from [Grötzsch, 1959], and states that triangle-free planar graphs are 3-colorable. Various conjectures were introduced about cycle obstructions for 3-coloring, like the strong Bordeaux conjecture [Borodin and Raspaud, 2003], the Novosibirsk 3-color conjecture [Borodin *et al.*, 2006] and the Steinberg conjecture [Steinberg, 1993]. The latter states that planar graphs without C_4 and C_5 are 3-colorable. All these conjectures have now been disproved in [Cohen-Addad *et al.*, 2017]. Following the Steinberg conjecture, Erdős [Steinberg, 1993] asked the following relaxation: what is the smallest integer n such that $(C_4, \ldots, C_n)$-free planar graphs are 3-choosable? Several results give a partial answer to this question (see [Abbott and Zhou, 1991; Borodin, 1996b,a; Sanders and Zhao, 1995]), culminating with the result of [Borodin *et al.*, 2005] stating that planar graphs without C_4 to C_7 are 3-colorable. It is still open whether the correct answer to Erdős' question is 6 or 7.

In the context of list coloring, a well-known result of [Thomassen, 1994] states that every planar graph is 5-choosable, which is tight by [Voigt, 1993]. While Grötzsch's theorem does not extend to the list coloring setting (see [Voigt, 1995]), similar cycle obstructions are known for enforcing 3-choosability. For example, bipartite graphs [Alon and Tarsi, 1992], graphs with girth 5 [Thomassen, 1995], (C_3, C_5, C_6)-free graphs [Lam *et al.*, 2005] are all 3-choosable. Erdős' question about $(C_4, \ldots, C_n)$-free planar graphs transposes to the list coloring setting. Again, some partial answers are known: $(C_4, \ldots, C_9)$-free planar graphs are 3-choosable [Borodin, 1996a], as well as $(C_4, \ldots, C_8)$-free planar graphs [Dvořák and Postle, 2018] and (C_4, C_i, C_j, C_9)-free planar graphs for any $i, j \in \{5, 6, 7, 8\}$ with $i \neq j$ [Wang *et al.*, 2011].

Except for Section 2.2 containing the proof of Theorem 2.3, this chapter is devoted to the study of cycle obstructions for list-coloring the square of a planar graph with lists of size $\Delta + O(1)$. The results of this chapter have been compiled in [Choi *et al.*, 2018]. We consider classes with a finite number of forbidden cycle lengths. For these classes, we can adapt Wegner's construction to prove that C_4 has to be forbidden. We then prove a dual result: planar C_4-free graphs are $(\Delta + 73)$-choosable. Observe that this bound is worse that the $\Delta + 48$ bound obtained in [Wang and Cai, 2008], but here we consider list coloring instead of coloring.

The main result of this chapter is a strengthening of this bound for C_4-free planar graphs with large enough maximum degree. In this case, we prove a $\Delta + 2$ bound for the list chromatic number, which is tight (see Figure 2.3). To prove this result, we partly use the discharging method: we give a set of reducible configurations, and prove that every C_4-free planar graph with large enough maximum degree has to contain one of them. However, this last part does not use any discharging argument, only the pigeonhole principle.

2.2 A Brooks-like result on graph powers

In this section, we give a proof of Theorem 2.3, stated again below.

Theorem 2.3. *For every integer $k \in \mathbb{N}^*$ and every $\Delta \geqslant 3$, the k-th power of every graph of maximum degree Δ is $(f(k,\Delta)+3-k)$-colorable, except for finitely many of them.*

First note that the case $k=1$ is easy since every graph G can be colored with $\Delta(G)+1 \leqslant f(1,\Delta)+2$ colors. Moreover, the case $k=2$ is already handled by Theorem 2.1. Thus, we only consider the case $k \geqslant 3$. In the following, we denote by G a graph of maximum degree $\Delta \geqslant 3$ such that $\chi(G^k) > f(k,\Delta)+3-k$, if any.

To prove Theorem 2.3, we prove that G cannot have some configurations, until we get to the point where G is proved not to exist at all. For each of these configurations, assuming that G contains it, we design a procedure to give a valid coloring of G, and thus reach a contradiction. This procedure roughly consists in coloring the vertices greedily by decreasing distance to the configuration.

2.2.1 First structural results

To prove Theorem 2.3, we give several properties satisfied by G. All of these are proved using the same technique: by contradiction, we assume the property does not hold. Then we define an ordering of the vertices of G, and we color them greedily in this order. For each vertex, we prove that there is always an available color, thus we reach a contradiction. We first apply this technique to prove that G is Δ-regular.

Proposition 2.9. *The graph G has minimum degree Δ.*

Proof. Assume that G has a vertex u of degree at most $\Delta-1$. Let H be the graph obtained from G by attaching to u a pending path $v_1,\ldots,v_k$. Observe that $\Delta(H)=\Delta$ since u has degree $\Delta-1$ in G. To reach a contradiction, we color vertices of G in H by decreasing distance to v_k.

Note that, usually, we remove elements of G, and use some minimality argument to obtain a coloring to extend. In this case, we instead add some vertices. This is not related to some inductive argument (we do not even color these new vertices). The goal of this modification is to make the gap between the number of forbidden colors and the upper bound easier to find, by counting the uncolored vertices in the neighborhood at distance k instead.

Let w be a vertex of G, at distance d from v_k. Note that the d vertices on a shortest path from w to v_k are uncolored. Therefore, w has at most $f(k,\Delta)-d$ colored neighbors in H^k (thus in G^k), hence w has at least $d-k+3$ available colors. Since $w \in V(G)$, we have $d \geqslant k$, hence w can always be colored.

We thus obtain $\chi(G^k) \leqslant f(k,\Delta)-k+3$, a contradiction. □

By coloring vertices of G by decreasing distance to a given small cycle, we can prove in a similar fashion that G has large girth, as shown by the following result.

Proposition 2.10. *The graph G has girth at least $k+2$.*

The proof of this result again relies on counting the number of available colors of each vertex in some coloring procedure. Given a vertex v in G and a partial coloring of G, we look for the number of available colors for v. In the worst case, the neighborhood of v in G^k induces a Δ-ary tree of height k in G, and the number of available colors is the number of uncolored vertices in this tree. However, we are not always considering worst cases. To mimic this counting argument, we thus consider (in any fixed order) all the $f(k,\Delta)$ non-empty non-backtracking walks of length at most k starting from v, meaning that we allow the same edge to be used twice, but not in a row. The number of available colors thus depends on the number of such walks ending on an uncolored vertex, and ending with vertices with the same color (possibly the same vertices).

We say that such a walk is *nice* if either its endpoint (possibly v) is uncolored, or if it is the endpoint of an already considered walk. The number of forbidden colors is the number of non-nice walk. Therefore, the number of available colors for v is the number of nice walks starting from v, minus $k-3$.

Proof. Assume that G contains a cycle $C = u_1 \cdots u_n$ with $3 \leqslant n \leqslant k+1$. We color vertices of G by decreasing distance to C.

Let v be a vertex of $G \setminus C$ and denote by $v_0 \cdots v_d$ a shortest path from v to C with $v_0 = v$ and $v_d \in C$, say $v_d = u_1$ by symmetry, see Figure 2.4. Then every subwalk of the non-backtracking walk $v_1 \cdots v_d u_2 \cdots u_n u_1 \cdots$ of length k ends up with an uncolored vertex. We thus obtain k nice walks, hence we can always color v.

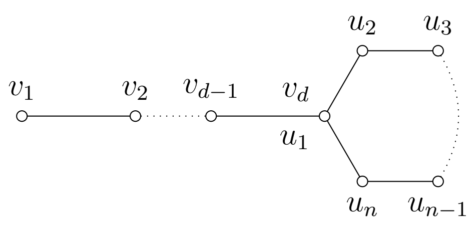


Figure 2.4 – Configuration of Proposition 2.10

Consider then a vertex $v \in C$. Then there are at least $2k$ nice walks starting from v, namely the subwalks of the two non-backtracking walks of length k going around C in the two possible directions. Thus each vertex of C has $k+2$ available colors. Moreover, vertices of C induce a clique in G^k, and there are at most $k+1$ of them. Since K_{k+1} is $(k+1)$-choosable, we can color vertices of C, reaching a contradiction. $\square$

2.2.2 Bounding the diameter

We end the proof of Theorem 2.3 by bounding the diameter of G. Indeed, for every $\Delta \geqslant 3$ and $d \in \mathbb{N}^*$, there is only a finite number of graphs of maximum degree Δ and diameter at most d. Thus, Theorem 2.3 is a consequence of the following proposition.

Proposition 2.11. *The graph G has diameter at most $2k-1$.*

The remainder of this section is devoted to the proof of Proposition 2.11. Assume that G has diameter at least $2k$, and consider a shortest path $P = u_1 \cdots u_k x v_1 \cdots v_k$ between u_1 and v_k, of length $2k$. First, for $1 \leqslant i \leqslant k$, we color both u_i and v_i with color i. Note that this is a proper partial coloring: if $\text{dist}(u_i, v_i) = k$, there is a path from u_1 to v_k of length $2k$, contradicting the hypothesis.

We fix the following ordering of the vertices of P: $u_1 > v_k > u_2 > v_{k-1} > \cdots > u_k > v_1 > x$. Let w be a vertex of G. We define the *root* r_w of w as the largest vertex in P on a shortest path from w to x. We denote by P_w any path obtained by concatenating a shortest path from w to r_w with the subpath of P between r_w and x. By definition, P_w is a shortest path between w and x.

We first prove a generic lemma about these objects.

Lemma 2.12. *Let v be a vertex of G. For every $w \in P_v \setminus P$, $r_w = r_v$.*

Proof. First note that the subpath of P_v between w and x has to be a shortest path. Therefore, since this path contains r_v, we must have $r_w \geqslant r_v$.

Conversely, consider the path obtained by concatenating the subpath of P_v between v and w with P_w. This path has the same length as P_v (since we replaced a shortest path from w to x in P_v by another one), and its largest element in P is r_w. Thus $r_v \leqslant r_w$, which ends the proof. $\square$

We can also have some information about the root of vertices close to P, as shown by the following result.

Lemma 2.13. *Let vw be an edge where $w \in P$ and $v \notin P$. Then $r_v \leqslant w$.*

Proof. Let i be the length of the subpath of P between x and w, so that $w \in \{x, v_i, u_{k+1-i}\}$ (the case $w = x$ corresponding to $i = 0$). The path obtained by concatenating this subpath with the edge wv is a path from x to v of length $i+1$. Thus P_v has length at most $i+1$, and we must have $\text{dist}(x, r_v) < i+1$ since $v \notin P$.

This proves that $r_v \leqslant w$, unless $r_v = u_{k+1-i}$ and $w = v_i$. But in this case, note that either $i = 1$ and we have a 4-cycle $u_k x v_1 w$, a contradiction with Proposition 2.10, or $i \neq 1$, and $u_i w v_{k+1-i}$ is a path shorter than $u_i \cdots u_k x v_1 \cdots v_{k+1-i}$, contradicting that P is a shortest path. $\square$

We are now ready to color the uncolored vertices of G, namely the vertices of G that are not in $P\setminus\{x\}$. We color these vertices by decreasing lexicographic order of $(r_v, \mathrm{dist}(v, r_v))$. Consider a vertex v with $r_v \neq x$. In this case, up to symmetry, we may assume that $r_v = u_i$, and $P_v = P'_v u_{i+1} \cdots u_k x$ where P'_v is a shortest path from v to r_v. We now use the assumption $\Delta \geqslant 3$ together with Proposition 2.9 to find some uncolored vertices.

First note that for $j \in [i+1, k]$, since $\Delta \geqslant 3$, there is a neighbor u'_j of u_j outside P. We define some other vertices by fixing a path $x_0 x_1 \cdots x_{\frac{k}{2}}$ such that $x_0 = x$ and $x_1 \notin P$. Since $\Delta \geqslant 3$, we know that for $1 \leqslant i < \frac{k}{2}$, each x_i has a neighbor x'_i different from x_{i-1} and x_{i+1}.

We now consider three types of uncolored vertices, as shown in Figure 2.5:

1. The internal vertices w of P'_v: by Lemma 2.12, we have $r_w = r_v$. Moreover, $\mathrm{dist}(w, r_w) = \mathrm{dist}(w, r_v) < \mathrm{dist}(v, r_v)$, hence w is uncolored when we consider v.

2. The vertices u'_j for $i < j \leqslant k$: by Lemma 2.13, we have $r_{u'_j} \leqslant u_j < r_v$, hence u'_j is uncolored when we consider v.

3. The vertices x_1 and x_j, x'_{j-1} for $2 \leqslant j \leqslant \frac{k}{2}$: Assume that $r_{x_j} \neq x$. Then P_{x_j} is a path of length at most $\frac{k}{2}$ from x_j to x, which does not use x_1. Therefore, we have two different paths of length at most $\frac{k}{2}$ from x_j to x in G. Hence, G contains a cycle of length at most $2 \times \frac{k}{2} = k$, contradicting Proposition 2.10. The same argument ensures that $r_{x'_j} = x$. Therefore, since $r_v \neq x$, x_1 and all the x_j, x'_{j-1} $(2 \leqslant j \leqslant \frac{k}{2})$ are uncolored when coloring v.

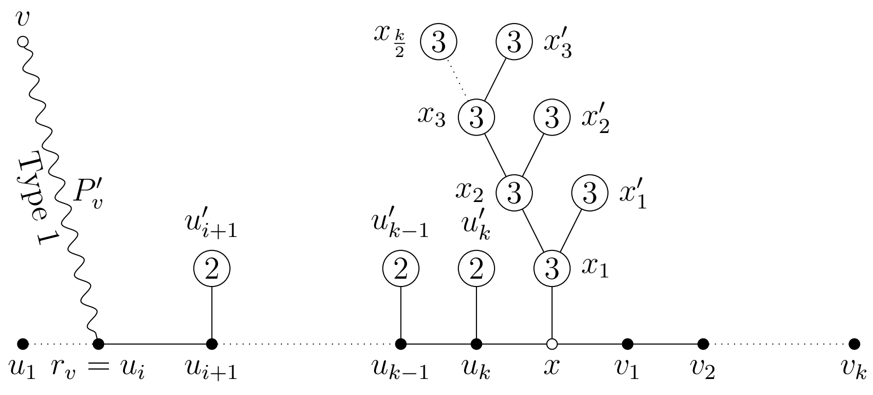


Figure 2.5 – Global picture of the situation when considering v: black vertices are already colored, white ones are uncolored. Integers inside vertices represent types.

We now have to make sure that at least $k-2$ such vertices lie in the neighborhood of v in G^k. Denote by $d = \mathrm{dist}(v, r_v)$ and $d' = \mathrm{dist}(r_v, x)$. For

$i = 1, \dots, d-1$, there is an internal vertex of P'_v at distance i from v. Thus we may assume that $d \leqslant k-2$. In this case, for $i = d+2, \dots, d+d'+1$, there is a vertex at distance i from v (either some u'_j or x_1). There are thus $\min(d+d'+1, k)-2$ uncolored vertices in the neighborhood of v in G^k, hence we may also assume that $d+d'+1 < k$.

For $i = d+d'+2, \dots, d+d'+\frac{k}{2}$, there are two vertices (x_j and x'_{j-1} for some $2 \leqslant j \leqslant \frac{k}{2}$) at distance at most i from v. Then we have $d+d'-1+2\min(k-d-d'-1, \frac{k}{2}-1)$ uncolored vertices at distance at most k from v. Observe that

$$d+d'-1+2(k-d-d'-1) = 2k-d-d'-3 \geqslant k-2$$

and that

$$d+d'-1+2\left(\frac{k}{2}-1\right) = k+d+d'-3 \geqslant k-2$$

since $d+d' \leqslant 0$ is not possible due to $r_v \neq x$. Therefore, we can always find an available color for every vertex v such that $r_v \neq x$.

We now consider the remaining case $r_v = x$. In this case, every vertex of P_v has x as root by Lemma 2.12, hence is uncolored. Write $P_v = x_0 \cdots x_d$ with $x_0 = v$, $d = \mathrm{dist}(x,v)$ and $x_d = x$. Again, since G is Δ-regular and $\Delta \geqslant 3$, each x_i (except possibly x_d) has a neighbor x'_i different from its neighbors in P_v. We now distinguish several cases depending on d. Again, in each of them, we prove that there are at least $k-2$ uncolored vertices or colors appearing twice in the neighborhood of v in G^k. This ensures that v always has an available color.

- If $d \geqslant k-2$, then $x_1, \dots, x_{k-2}$ are uncolored neighbors of v in G^k. Hence we may assume that $d < k-2$.

- If $d < k-2$ and $d \geqslant \frac{k}{2}$, then $x_1, \dots, x_d, x'_2, \dots, x'_{d-1}$ are at distance at most $d-1$ from x, hence they are uncolored. Moreover, they are at distance at most k from v. We thus have $2d-2 \geqslant k-2$ uncolored vertices.

- Otherwise, we have $d < \frac{k}{2}$. In this case, observe that some of the colors $1, \dots, k$ present on $u_1, \dots, u_k$ and $v_1, \dots, v_k$ appear twice in the neighborhood of v in G^k. More precisely, v is at distance at most k from $u_{d+1}, \dots, u_k$ and from $v_1, \dots, v_{k-d}$. Thus, colors $d+1, \dots, k-d$ appear twice in the neighborhood of v in G^k, meaning that we spare $k-2d$ colors. Since there are also $2d-2$ uncolored vertices, v spares $k-2$ colors in its neighborhood in G^k, hence we can color it.

This ends the proof of Theorem 2.3. As a final remark, observe that the proofs of Propositions 2.9 and 2.10 are still valid both in the list coloring setting (since we use only degeneracy arguments) and in the case where we want to spare more colors (say at most k). This is not the case anymore for Proposition 2.11. However, maybe some more involved arguments could bound the diameter of G in these two more general settings.

2.3 Coloring squares of planar graphs

Let $\mathcal{S}$ be a *finite* set of integers, and $\mathcal{C}_{\mathcal{S}}$ be the set of planar graphs with no cycle of length $\ell \in S$ as a subgraph. We first show that removing C_4 is necessary to obtain a $\Delta + O(1)$ upper bound for the square chromatic number.

Proposition 2.14. *If $4 \notin \mathcal{S}$, then for every $C \in \mathbb{N}$, there exists a graph $G \in \mathcal{C}_{\mathcal{S}}$ such that $\chi(G) \geqslant \Delta + C$.*

Proof. Since $\mathcal{S}$ is finite, there exists an odd integer k such that $2k \notin \mathcal{S}$. Begin with a k-cycle and replace each edge vw with a copy of $K_{2,t}$, so that the two vertices of degree t replace v and w. The resulting graph, $G_{k,t}$, has maximum degree $2t$ and has cycles only of lengths 4 and $2k$.

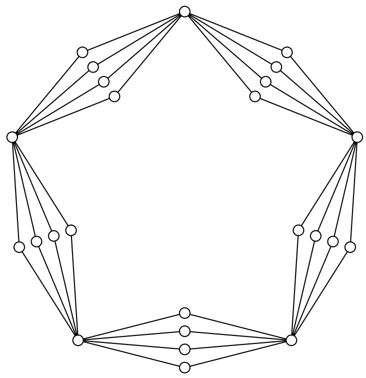


Figure 2.6 – The graph $G_{5,4}$

Consider a proper coloring of $G^2_{k,t}$. Observe that each color class contains at most $(k-1)/2$ vertices of degree 2 in $G_{k,t}$ (by the pigeonhole principle). Since $G_{k,t}$ has kt 2-vertices, we have:

$$\chi(G^2_{k,t}) \geqslant \frac{kt}{(k-1)/2} = \frac{2kt}{k-1} = 2t + \frac{2t}{k-1} = \Delta(G) + \frac{2t}{k-1}$$

Given any constant C, we can choose t sufficiently large so that $\frac{2t}{k-1} > C$. The graph $G_{k,t}$ is then the required graph. □

The goal of this chapter is to prove the following refinement of Proposition 2.7:

Theorem 2.15 ([Choi *et al.*, 2018]). *There exists an integer Δ_0 such that every C_4-free planar graph G with $\Delta \geqslant \Delta_0$ satisfies $\chi_\ell(G^2) \leqslant \Delta + 2$.*

Note that this bound is tight, as shown by several constructions from [Borodin *et al.*, 2004; Dvořák *et al.*, 2008], see Figure 2.3.

As the title of the chapter may suggest, the proof of Theorem 2.15 uses the discharging method. We actually follow the same general approach as in [Bonamy *et al.*, 2019a], which considered planar graphs with girth at least 5; however, we need new ideas to handle the presence of triangles. In Sections 2.4 and 2.5, we design a specific induction scheme allowing to prove $(\Delta+2)$-choosability by reducing some configurations. More precisely, we use the arguments presented in Chapter 1 to reduce some small configurations in Section 2.4. Section 2.5 is devoted to reducing a much larger one.

The second part of the discharging method consists in proving the completeness of the induction scheme, i.e. that every C_4-free planar graph with large enough Δ can be reached by the scheme. In Chapter 1, this part is achieved by moving charges around a graph, which led to a quite involved case analysis. In Section 2.6, we prove this result only using that planar graphs are 5-degenerate, together with the pigeonhole principle.

Remark 2.16. Theorem 2.15 also holds for correspondence coloring, a generalization of list coloring. For the ease of exposition, we present the proof for choosability. We then devote Section 2.7 to the introduction of correspondence coloring and to the proof of the extended version of Theorem 2.15.

2.4 Small reducible configurations

First, take $\Delta_0 = 23769500^2 = 564989130250000$ and fix $k \geq \Delta_0$. We prove by contradiction that if G is a plane graph with no 4-cycles and with $\Delta(G) \leq k$, then G^2 is $(k+2)$-choosable. (By *plane graph*, we mean a planar graph with a fixed embedding in the plane.) Assume this assertion is false and let G be a counterexample that minimizes $|E(G)| + |V(G)|$. Let L be an assignment of lists of size $k+2$ to the vertices of G such that G^2 has no L-coloring. Since G is a plane graph, recall that the embedding of G in the plane is also fixed.

The goal of this section is to prove that G does not contain some configurations. As in Chapter 1, a first step is to prove that G is connected and does not contain vertices of small degree.

Lemma 2.17. *Graph G is connected and has minimum degree at least* 2.

Proof. Note that G is connected, since otherwise one of its components is a smaller counterexample. Now assume there exists a 1-vertex $v \in V(G)$. By the minimality of G, we can L-color $(G \setminus \{v\})^2$. Since $|L(v)| = k+2$, and v has at most $1 + (k-1)$ neighbors in G^2, we can color v with a color not used on its neighbors in G^2, which is a contradiction. □

The next two lemmas essentially show that every vertex of G must be near a vertex of high degree. To formalize this, we use the following terminology:

a vertex $v \in V(G)$ is *big* if $\deg(v) \geq \sqrt{k}$ and *small* otherwise. Denote by B and S the sets of big and small vertices, respectively. To refine the set S, we write S_i for the set of small vertices with exactly i big neighbors.

Remark 2.18. In the figures, we apply the conventions of Chapter 1: we use black circles for vertices with all neighbors shown. Furthermore, we draw small vertices as circles, and big vertices as squares. When we do not know whether a vertex is big or small, we consider (unless stated otherwise) that we are in the worse case, i.e. it has degree Δ, hence we draw it as a big vertex. For example, Figure 2.7 shows the configurations forbidden by Lemma 2.19.

Lemma 2.19. *For each edge $vw \in E(G)$, either $v \in N[B]$ or $w \in N[B]$. Furthermore, if $\deg(v) = \deg(w) = 2$, then $v, w \in N[B]$.*

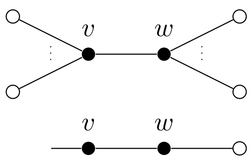


Figure 2.7 – Forbidden configurations of Lemma 2.19.

Proof. Assume to the contrary that some edge vw has $v, w \notin N[B]$. By minimality, we can L-color $(G - vw)^2$. We uncolor v and w. Since $v, w \notin N[B]$, both v and w have less than $\sqrt{k} \times \sqrt{k}$ colored neighbors in G^2. Since $|L(v)| = |L(w)| = k + 2$, we can find distinct available colors for v and w.

Suppose instead that $d(v) = d(w) = 2$ and $v \in N[B]$ and $w \notin N[B]$. Again, by minimality we L-color $(G - vw)^2$, then uncolor v and w. Now v has at most $k + 1$ colored neighbors in G^2, so v has an available color. As before, we can color w. This gives an L-coloring for G^2, a contradiction. $\square$

Lemma 2.20. *If vw is an edge with $\deg(v) = \deg(w) = 2$, then v and w have no common neighbor.*

Proof. Assume there exists a triangle vwx with $\deg(v) = \deg(w) = 2$. By minimality, we can L-color $(G \setminus \{v, w\})^2$. Both v and w have $\deg(x) - 1 \leqslant k - 1$ colored neighbors in G^2. So v and w each have at least 3 available colors, and thus we can color them both. $\square$

Lemma 2.21. *Let vx_1x_2 be a triangle of G such that some vertex $w \in S \setminus \{v, x_1, x_2\}$ has a common 2-neighbor with x_1. If either (a) $d(x_2) = 2$ or (b) $d(x_2) = 3$ and w and x_2 have a common 2-neighbor, then $d(x_1) \geq 4$.*

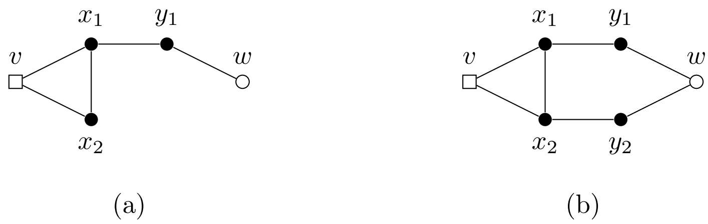


Figure 2.8 – Forbidden configurations of Lemma 2.21.

Proof. Let y_1 and y_2 denote the 2-neighbors of w common with x_1 and x_2 if they exist (in Case (a), only y_1 is defined). Assume that $\deg(x_1) \leqslant 3$. Note that we have $\deg(x_1) = 3$ since otherwise we have $y_1 = v$, hence $\deg(v) = 2$ and G is a triangle. If $vw \in E(G)$, then wvx_1y_1 is a 4-cycle in G, a contradiction. So $vw \notin E(G)$. By assumption, $w \notin \{v, x_1, x_2, y_1\}$. So if $wx_2 \in E(G)$, then $wx_2x_1y_1$ is a 4-cycle in G, again a contradiction. Thus, $wx_2 \notin E(G)$. Since $d(x_1) = 3$ and $v, x_2, y_1 \in N(x_1)$, we must have $w \notin N(x_1)$. Since $N(y_1) = \{x_1, w\}$, also $vy_1 \notin E(G)$. So in both cases $wx_1, wx_2, wv, vy_1 \notin E(G)$. And in (b) also $vy_2 \notin E(G)$.

Let $S = \{x_1, x_2, y_1\}$ in Case (a), and $S = \{x_1, x_2, y_1, y_2\}$ in Case (b). By minimality, we L-color $(G \setminus S)^2$. For each $i \in \{1, 2\}$, the number of colored neighbors in G^2 of x_i is at most:

$$|\{v, w\}| + |N(v) \setminus \{x_1, x_2\}| \leqslant 2 + (k - 2) = k.$$

Thus, x_1 and x_2 both have at least 2 available colors, so we can color them. Further, for each $i \in \{1, 2\}$, the number of colored neighbors of y_i is at most

$$|\{v, w, x_1, x_2\}| + |N(w) \setminus \{y_i\}| \leqslant 4 + \sqrt{k} - 1 = \sqrt{k} + 3.$$

Therefore, y_1 and y_2 (if defined) both have $k - \sqrt{k} - 1$ available colors. Since k is large enough, we can color them to get an L-coloring for G^2, a contradiction. □

We combine Lemmas 2.19 and 2.21 to prove the reducibility of the bigger configuration shown in Figure 2.9.

Lemma 2.22. *Fix $v, w \in V(G)$ such that $w \in S$. Now G cannot contain vertices $y_1, \ldots, y_5$ that are consecutive neighbors of w and that satisfy both conditions below; see Figure 2.9.*

1. *Each y_i has degree two and has a common neighbor x_i with v.*

2. *For each $i \in \{1, \ldots, 4\}$, each vertex inside the cycle $vx_iy_iwy_{i+1}x_{i+1}$ is adjacent to v.*

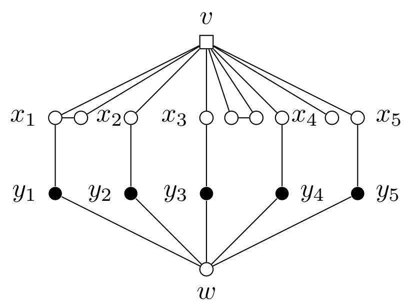


Figure 2.9 – A possible configuration of Lemma 2.22.

Proof. We assume that G contains such a configuration and reach a contradiction, by showing that G contains a configuration forbidden by Lemma 2.21. Since G contains no 4-cycle, all x_i's are distinct. Moreover, every connected component of a graph strictly contained in a cycle $vx_iy_iwy_{i+1}x_{i+1}$ must be of size at most two (otherwise it creates a C_4 with v).

Below when we write a statement about x_i, we mean that it is true for each $i \in \{2, 3, 4\}$. Since $w \in S$, Lemma 2.19 implies that $d(x_i) \geq 3$. Because $y_1, \ldots, y_5$ are consecutive neighbors of w, vertex x_i is not adjacent to w. Since G has no 4-cycle, x_i has at most one common neighbor with v. Thus $d(x_i) = 3$. Define z so that $N(x_3) = \{v, y_3, z\}$. If $z \in \{x_2, x_4\}$, then G contains the second configuration in Lemma 2.21, a contradiction. If z has a neighbor other than x_3 and v, then call it z'; now z' is adjacent to v (by hypothesis 2), so vx_3zz' is a 4-cycle, a contradiction. Thus, z is a 2-vertex with $N(z) = \{x_3, v\}$. Now G contains the first configuration in Lemma 2.21, again a contradiction. □

2.5 Reducing regions

2.5.1 Terminology

In order to present the last configuration, we introduce some terminology. Recall that S is the set of small vertices, and S_i is the set of small vertices with exactly i big neighbors. Let G' denote the multigraph formed from G by suppressing every vertex of degree 2 in $S \setminus N[B]$, and then contracting every edge between S_1 and B. (Suppressing a 2-vertex v means deleting v and adding an edge between its two neighbors.) Note that G' may contain loops. For example, there is a loop in G' around a vertex u if u is a big vertex in G and there is a triangle uvw with $v, w \in S_1$. We say that a vertex of G *disappears* when constructing G' if it is either a suppressed vertex, or a vertex in S_1.

Let G'' denote the multigraph formed from G' by removing every loop, and let G''' denote the underlying multigraph of G'', i.e., the multigraph formed from G'' by deleting the minimal number of edges to remove all faces of length

2. Note that G''' can have parallel edges. For example, suppose v and w have parallel edges, say e_1 and e_2, in G'. If some vertices are embedded inside and outside of the cycle e_1e_2, then in G''' vertices v and w still have parallel edges, with those same vertices embedded inside and outside of the cycle they bound. However, G''' cannot have faces of length 2.

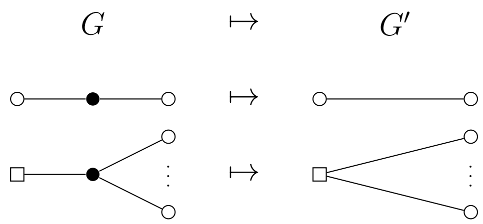


Figure 2.10 – Construction of G'

An *r-region of G''* is a set $\{f_1, \ldots, f_r\}$ of consecutive faces of length 2 such that:

- For $1 \leqslant i < r$, f_i shares one edge with f_{i+1}. (We say that the f_i's are *consecutive*.)
- All the f_i's have the same vertices b_1, b_2 on their boundary, where b_1 and b_2 are distinct vertices of B.

Note that each of the faces in an r-region is constructed from some cycle of G when we apply the construction rules above. By extension, an *r-region of G* is the subgraph of G induced by the vertices of these cycles, together with those lying on the inside of those cycles. (We often simply write *region*, when the specific value of r is less important.) When R is an r-region of G, we denote by $V(R)$ the set of vertices appearing on all faces of R, excluding b_1 and b_2.

To reach a contradiction, we prove the following two propositions.

Proposition 2.23. *G does not contain any r-region for $r \geqslant 475353$.*

Proposition 2.24. *G contains an r-region of size at least $\frac{\sqrt{k}}{50} - 37$.*

Our contradiction now comes quickly. These propositions give that $\frac{\sqrt{k}}{50} - 37 < 475353$. This inequality implies $k < 23769500^2$, contradicting the hypothesis $k \geqslant \Delta_0 = 23769500^2$.

The present section is devoted to the proof of Proposition 2.23. We then prove Proposition 2.24 in Section 2.6. Both these results use some structural properties about regions in G. We thus begin by investigating these structures.

2.5.2 Structural properties of regions

We first classify each edge of G' based on its corresponding path in G. An edge e in G' *corresponds to a path* $x_1 \cdots x_n$ *in* G if $e = x_1 x_n$ and for each $i \in \{2, \ldots, n-1\}$, one of the following holds:

- x_i is a 2-vertex in G and $x_{i-1}, x_{i+1} \in S$, or
- $x_i \in S_1$ and either x_{i-1} or x_{i+1} lies in B.

Due to the construction of G', for every loop (resp. non-loop edge) e of G', there is a unique cycle (resp. path) $x_1 \cdots x_n$ in G corresponding to e (with possibly $n = 2$). Note that we used here that the suppressed 2-vertices are not in $N[B]$, hence every contracted edge (between S_1 and B) is between two adjacent vertices in G.

The following lemma ensures that every edge (resp. loop) of G' corresponds to a short path (resp. cycle) of G. It also gives a classification of all the possible such paths (resp. cycles), depicted in Figure 2.11.

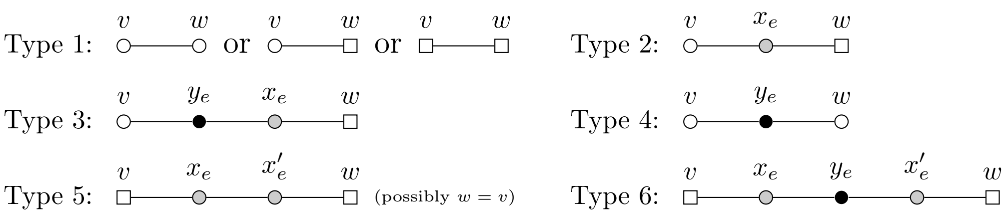


Figure 2.11 – The six types of paths in G that create edges in G' (gray vertices lie in S_1).

Lemma 2.25. *Each edge $e = vw$ of G' corresponds to a path or a cycle in G for which exactly one of the following six conditions holds (up to exchanging v with w). If e satisfies condition i below (for some $i \in \{1, \ldots, 6\}$), then we say that e has* type i. *If $v \in S$, then e has one of types 1–4. If e is a loop of G, then e has type* 5. *Finally, if $v, w \in B$, then e has type 1, 5, or 6.*

1. *$e \in E(G)$.*
2. *$w \in B$ and e corresponds to a path $v x_e w$ in G with $x_e \in S_1$.*
3. *$w \in B$ and e corresponds to a path $v y_e x_e w$ in G with $x_e \in S_1$ and* $\deg(y_e) = 2$.
4. *$w \in S$ and e corresponds to a path $v y_e w$ in G with* $\deg(y_e) = 2$.
5. *e corresponds to a path $v x_e x'_e w$ in G with $x_e, x'_e \in S_1$.*
6. *e corresponds to a path $v x_e y_e x'_e w$ in G with $x_e, x'_e \in S_1$ and* $\deg(y_e) = 2$.

Proof. Due to the construction of G', each edge e in G' between v and w comes from a path (or cycle) P_e in G between v and w. In particular, every internal vertex of P_e is either a 2-vertex in $S \setminus N[B]$ or a vertex of S_1 which is preceded or followed in P_e by a big vertex. This implies that each internal vertex of P_e is small, and that the only vertices of P_e that can be big are v and w.

By Lemma 2.19, no two consecutive vertices of P_e are suppressed. This implies that P_e has length at most four.

- If P_e has length one, then e has type 1.
- If P_e has length two, then we have $v \neq w$ since G is simple. Denote by x the middle vertex of P_e. We must have either $v, w \in S$ and $\deg_G(x) = 2$ (case 4), or $v \in B$, $w \in S$ and $x \in S_1$ (case 2).
- If P_e has length three, then at least one of v, w must be in B and its neighbor in P_e must be in S_1. If both v and w lie in B, then we are in case 5; otherwise, we have $v \neq w$ and we are in case 3.
- Finally, if P_e has length four, then we have $v \neq w$ since G is C_4-free. Moreover, they both have to be big and their neighbors in P_e (say x_e, x'_e) lie in S_1. The other vertex y_e of P_e must have degree two, so we are in case 6.

Observe in particular that if v is small, then cases 5 and 6 cannot occur. Moreover, if v and w are big, then only cases 1, 5, and 6 can occur. Finally, every loop of G' has type 5. □

In what follows, when referring to an edge e with type i, we use x_e, x'_e, and y_e as defined in the corresponding part of Lemma 2.25. This lemma implies the following facts about the structure of regions in G.

Corollary 2.26. *Let R be a region of G. Now $V(R)$ is the disjoint union of three sets B_1, B_2, D such that $B_i \subset N(b_i)$ for some $b_1, b_2 \in B$, and D is an independent set of 2-vertices, each with a neighbor in each of B_1 and B_2.*

Proof. Let R be a region of G. By definition, there exists $b_1, b_2 \in B$ on the boundary of every face of R in G''. Therefore, in G', the edges appearing in R are either loops on b_1 or b_2 or edges between b_1 and b_2.

Note that $V(R)$ is the set of all vertices of G that disappear when we construct the edges of R in G'. For each $i \in \{1, 2\}$, define B_i as the set of vertices v of G such that vb_i is contracted when constructing an edge of R in G'. We also define D as the set of vertices in G that are suppressed when constructing an edge of R in G'. By definition, we have $B_i \subset N(b_i)$.

By Lemma 2.25, since $b_1, b_2 \in B$, each edge e between b_1 and b_2 in G has type 1, 5, or 6, and each loop around b_1, b_2 has type 5. This ensures that

$V(R) = B_1 \cup B_2 \cup D$ and that D contains only vertices of degree 2 in G. By Lemma 2.19, this implies that D is an independent set.

It remains to show that these sets are pairwise disjoint. Assume that there is $x \in B_1 \cap B_2$. Now xb_1 and xb_2 are both contracted when constructing G'. This requires that $x \in S_1$. Since b_1 and b_2 are both big, we must have $b_1 = b_2$, a contradiction. Further, since $b_1 \in B$, no neighbor of b_1 is suppressed during the construction of G'. Since $B_1 \subset N(b_1)$, we thus have $D \cap B_1 = \varnothing$. By symmetry, we also have $D \cap B_2 = \varnothing$. □

In the following, given a region R, we use the notation of Corollary 2.26. We are now ready to prove Proposition 2.23.

2.5.3 Large regions are reducible

In this section, we show that G cannot contain arbitrarily large regions, i.e., for r large enough every r-region is reducible. Note that the square of such r-regions consists of two cliques, with some edges between them. Following the terminology of Corollary 2.26, we denote the vertices of these cliques by B_1 and B_2. As before, D denotes a set of independent 2-vertices, each with one neighbor in B_1 and one neighbor in B_2. We begin by proving that there are only few edges between B_1 and B_2.

Lemma 2.27. *Let R be an r-region of G. Now each $w \in B_1 \cup B_2$ has at most one neighbor in B_1, at most one in B_2, and at most eight in D.*

Proof. Suppose $w \in B_1 \cup B_2$. If w has two neighbors x and y in B_i, then b_ixwy is a 4-cycle in G, a contradiction. So we assume w has at most one neighbor in each of B_1 and B_2. In what follows, we assume by symmetry that $w \in B_1$.

Suppose that w has five consecutive neighbors $x_1, \ldots, x_5$, all in D, and denote by y_i the common neighbor of x_i and b_2. By Lemma 2.22, there is a vertex z inside some cycle $wx_iy_ib_2y_{i+1}x_{i+1}$ that is not adjacent to b_2. Since R is an r-region, z disappears when we construct G'. Since $z \notin N_G(b_2)$, vertex z must be a 2-vertex. By Lemma 2.19, each neighbor of z is adjacent to b_2. So G contains a 4-cycle, a contradiction. Thus, w has at most four consecutive neighbors in D.

Consider an edge wx between these blocks of consecutive neighbors in D where $x \in V(R) \setminus D$. Then x cannot lie in B_1, otherwise b_1wx is a triangle not containing b_2 nor any vertex in B_2. By planarity, there cannot be vertices of D inside and outside of this triangle. Therefore $x \in B_2$.

Since G has no 4-cycle, at most one such neighbor x exists, so w has at most two blocks of consecutive neighbors in D. This proves the final assertion. □

Proving that G does not contain large regions amounts to proving that r-regions of G are square L'-colorable for a suitable assignment L'. To prove this new assertion, we introduce yet another method, different from the ones of

Chapter 1. Here, we use an auxiliary result about choosability, due to Bondy, Boppana, and Siegel (see Remark 2.4 in [Alon and Tarsi, 1992]). This result applies to kernel perfect digraphs. We briefly recall the definition here.

Definition 2.28. A *kernel* K in a digraph D is a subset of $V(D)$ such that every vertex v of D satisfies: $v \in K$ if and only if $N^+(v) \cap K = \varnothing$. A digraph is *kernel perfect* if each of its induced subgraphs has a kernel.

As shown by Bondy, Boppana and Siegel (see [Alon and Tarsi, 1992]), kernel-perfect orientations can be linked to choosability: we can translate the problem of choosability into finding an orientation with nice properties.

Lemma 2.29. *Let D be a kernel perfect digraph D with underlying graph H. If L is a list assignment for $V(H)$ such that for all $v \in V(H)$, $|L(v)| \geqslant d^+(v)+1$, then H is L-colorable.*

Proof. We prove the theorem by induction on the size p of $\cup_{v\in V(H)}|L(v)|$. If $p = 1$, then $d^+(v) = 0$ and $|L(v)| = 1$ for every vertex v of D. Thus, H is an independent set, and assigning to v the unique element in $L(v)$ gives a proper L-coloring.

Assume now that $p > 1$ and take $c \in \cup_{v\in V(H)} L(v)$. Consider the subgraph D' of D induced by $\{v \in V(H), c \in L(v)\}$. By hypothesis, D' has a kernel K. We then color the vertices in K with c.

It remains to color $D \setminus K$. To this end, we apply the induction hypothesis with the list assignment L' defined by $L'(v) = L(v) \setminus \{c\}$ for each $v \in V(H)$. Note that $D \setminus K$ is still kernel perfect, since it is a subgraph of D. We thus have to prove that $|L'(v)| \geqslant d^+_{D\setminus K}(v) + 1$ for each $v \in V(H) \setminus K$. We separate two cases:

- If $c \notin L(v)$, then $|L'(v)| = |L(v)| \geqslant d^+_D(v) + 1 \geqslant d^+_{D\setminus K}(v) + 1$.
- If $c \in L(v)$, then $|L'(v)| = |L(v)| - 1$. However, since K is a kernel of H' and $v \notin K$, v has an out-neighbor in K, hence $d^+_D(v) = d^+_{D\setminus K}(v) + 1$. Finally, we obtain $|L'(v)| \geqslant d^+_{D\setminus K}(v) + 1$. □

We use this lemma to reduce the problem of square L-coloring an r-region to finding a kernel perfect orientation. We apply this method to prove the following generic result about choosability of graphs covered by two cliques with few edges between them.

Lemma 2.30. *Let H be a graph covered by two disjoint cliques, B_1 and B_2, see Figure 2.12. Let L be a list assignment for $V(H)$ and suppose $T_i \subset B_i$ for each $i \in \{1,2\}$. Now H is L-colorable if the following five conditions hold.*

1. $|B_1| \geqslant 52811$ *and* $|B_2| \geqslant 52811$.
2. $|T_1| \leqslant 4400$ *and* $|T_2| \leqslant 4400$.

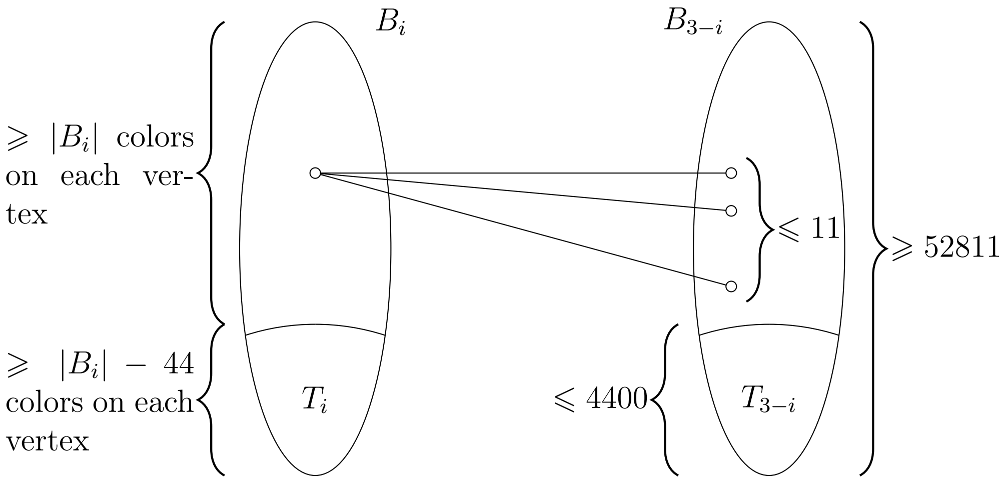


Figure 2.12 – Conditions of Lemma 2.30

3. *For each $v \in B_i$, $|N(v) \cap B_{3-i}| \leqslant 11$.*

4. *For each $v \in T_i$, $|L(v)| \geqslant |B_i| - 44$.*

5. *For each $v \in B_i \setminus T_i$, $|L(v)| \geqslant |B_i|$.*

Proof. To prove this result we construct an orientation D of H such that D satisfies the hypotheses of Lemma 2.29. We first show that we can order the vertices $x_1, \dots, x_{|B_1|}$ and $y_1, \dots, y_{|B_2|}$ of B_1 and B_2 such that $T_1 = \{x_1, \dots, x_{|T_1|}\}$, $T_2 = \{y_1, \dots, y_{|T_2|}\}$ and every path beginning in $\{x_{|B_1|-10}, \dots, x_{|B_1|}\}$ and ending in $\{y_{|B_2|-10}, \dots, y_{|B_2|}\}$ that alternates between B_1 and B_2 has length at least 5. Note that a single edge may be an alternating path, so we require that no edge joins x_i and y_j whenever $i \geqslant |B_1| - 10$ and $j \geqslant |B_2| - 10$.

Definition of the orderings

We now construct the vertex orderings from the previous paragraph. Their only non-trivial property is the absence of short alternating paths between the final 11 vertices in B_1 and those in B_2. So, our goal is to construct $Z_1 \subset B_1$ and $Z_2 \subset B_2$ with $|Z_1| = |Z_2| = 11$ such that no alternating path of length at most 3 begins in Z_1 and ends in Z_2. To this end, we first define Z_2, then count the number of vertices in B_1 reachable from Z_2 with such an alternating path.

If there exists $v \in B_1 \setminus N(T_2)$ with 11 neighbors in B_2, then we take $Z_2 = N_H(v) \cap B_2$. If no such vertex exists, then we swap the roles of B_1 and B_2, take Z_2 as any subset of $B_2 \setminus (T_2 \cup N(T_1))$ of size 11 (this is always possible since $|B_2| \geqslant 52811 \geqslant |T_2| + 11|T_1| + 11$), and let v be any vertex of B_1. Since every element of Z_2 has at most 10 neighbors in $B_1 \setminus \{v\}$, we have $|N_{B_1}(Z_2) \setminus \{v\}| \leqslant 11 \times 10 = 110$. Moreover, each vertex in $N_{B_1}(Z_2) \setminus \{v\}$ has at most 11 neighbors in B_2 (one of them being in Z_2). Since the only neighbors of v in B_2 are in Z_2, we obtain

$$|N_{B_2}(N_{B_1}(Z_2)) \setminus Z_2| \leqslant 11 \times 10^2 = 1100.$$

By the same argument, the set of vertices of B_1 reachable from Z_2 with an alternating path of length exactly 3 has size

$$|N_{B_1}(N_{B_2}(N_{B_1}(Z_2)) \setminus Z_2)| \leqslant 1100 \times 10 = 11000.$$

So the number of vertices of B_1 that are excluded from appearing in Z_1, because of paths to Z_2, is at most

$$|N_{B_1}(N_{B_2}(N_{B_1}(Z_2)) \setminus Z)| + |N_{B_1}(Z_2) \setminus \{v\}| + |\{v\}| = 11000 + 110 + 1 = 11111.$$

Further, we must also remove vertices of T_1. Thus, we can choose Z_1 as desired, since $|B_1| - |T_1| - 11111 \geqslant 11$.

Definition of the orientation

For each edge with both endpoints in the same clique, direct it toward the vertex of lower index. For every other edge, direct it in both directions, unless one of its endpoints is among the last 11 vertices of B_1 or B_2. In this case, direct the edge toward this endpoint.

The orientation is kernel-perfect

Let $A \subset V(H)$, with $A \neq \varnothing$. We look for a kernel of A. Let x_p (resp. y_q) denote the vertex with smallest index in $A \cap B_1$ (resp. $A \cap B_2$), if it exists. If $A \cap B_1 = \varnothing$, then $\{y_q\}$ is a kernel. Similarly, if $A \cap B_2 = \varnothing$, then $\{x_p\}$ is a kernel. So we assume that both x_p and y_q are well defined. We can also assume that $x_p y_q \in E(H)$, since otherwise $\{x_p, y_q\}$ is a kernel.

Let x_r (resp. y_s) denote the vertex with smallest index in $A \cap B_1$ (resp. $A \cap B_2$) that is not a neighbor of y_q (resp. x_p).

We now prove that at least one of $\{x_p\}$, $\{x_p, y_s\}$, $\{y_q\}$ and $\{x_r, y_q\}$ is a kernel. Assume the contrary. Since $\{x_p, y_s\}$ is not a kernel, there exists y_j such that $q \leqslant j < s$ and either there is no edge $x_p y_j$ or it is directed only towards y_j. Due to the choice of s, this edge is present in H and is thus directed only one way. (If y_s is not well defined, i.e. if x_p is adjacent to every vertex in $A \cap B_2$, we can obtain the same result using that $\{x_p\}$ is not a kernel.)

Similarly, using that $\{x_r, y_q\}$ is not a kernel (or only $\{y_q\}$ if y_q is adjacent to every vertex in $A \cap B_1$), we have an edge $x_i y_q$ directed only towards x_i.

Since $x_i y_q$ and $x_p y_j$ are directed towards x_i and y_j, this ensures that x_i and y_j are both among the final 11 vertices of B_1 and B_2. However, this is impossible, since $x_i y_q x_p y_j$ would be a path of length 3 that alternates between B_1 and B_2 and begin and ends in the final 11 vertices of B_1 and B_2. Thus, either $\{x_p, y_s\}$, $\{x_p\}$, $\{x_r, y_q\}$ or $\{y_q\}$ is a kernel of A. So the orientation is kernel-perfect.

The orientation has small out-degrees

We now prove that $|L(v)| \geqslant d^+(v) + 1$ for every $v \in V(H)$. By symmetry, it suffices to prove this for all $v \in B_1$, i.e., $v = x_i$ whenever $i \in \{1, \ldots, |B_1|\}$. If $i \leqslant |T_1|$, i.e., $v \in T_1$, then v has at most $|T_1|-1 \leqslant 4399$ out-neighbors in B_1 and at most 11 out-neighbors in B_2. So $\deg^+(v) + 1 \leqslant 4410 \leqslant |B_1| - 44 \leqslant |L(v)|$. If $|T_1| < i \leqslant |B_1| - 11$, then v has at most $|B_1| - 12$ out-neighbors in B_1 and at most 11 in B_2. So $\deg^+(v) + 1 \leqslant |B_1| \leqslant |L(v)|$. If $i > |B_1| - 11$, then every out-neighbor of v is in B_1, so $\deg^+(v) + 1 \leqslant |B_1| \leqslant |L(v)|$. □

We now use this lemma to prove Proposition 2.23, i.e., that large regions are reducible for square choosability.

Proof of Proposition 2.23. We use proof by contradiction. Assume that G has an r-region R with $r \geqslant 475353$. Let v_1 and v_2 be adjacent vertices of R such that any vertex at distance 2 in G from $\{v_1, v_2\}$ lies in $\{b_1, b_2\} \cup V(R) \cup N(b_1) \cup N(b_2)$. To see that such vertices exist, pick $v_1 \in B_1$ such that each face containing v_1 is in R, and let v_2 be a neighbor of v_1 in $B_2 \cup D$.

Let T denote the set of vertices in $B_1 \cup B_2$ that appear on a face of G not in R. Note that $|T| \leqslant 4$; this is because each vertex of T must lie on the first or last edge of the r-region in G', and each of these edges has exactly one vertex in each of B_1 and B_2. Let $T^{(1)} = N(T) \cap V(R)$, $T^{(2)} = N(T^{(1)}) \cap V(R)$ and $T^{(3)} = N(T^{(2)}) \cap V(R)$, so that for $1 \leqslant i \leqslant 3$, $T \cup \cdots \cup T^{(i)}$ is the set of vertices of $V(R)$ at distance at most i from T (with the distance taken in $V(R)$). By Lemma 2.27, each vertex of T has at most 10 neighbors in $V(R)$, so $|T^{(1)}| \leqslant 40$, $|T^{(2)}| \leqslant 400$ and $|T^{(3)}| \leqslant 4000$.

By minimality, $(G - v_1v_2)^2$ has an L-coloring φ. Let $B_i' = B_i \setminus N[T]$. We uncolor the vertices of $B_1' \cup B_2' \cup D$.

We also define T_i as the set of vertices of B_i' with some colored neighbor from $V(R)$ in R^2, i.e., $T_i = B_i' \cap (T^{(2)} \cup T^{(3)})$. Finally, let $H = G^2[B_1' \cup B_2']$. Note that B_1' and B_2' are cliques in H. Moreover, they are disjoint since $B_1' \cap B_2' \subset B_1 \cap B_2 = \varnothing$.

Our goal is now to apply Lemma 2.30 to L'-color H, where L' is the list assignment formed from L by removing all colors already used on vertices at distance at most 2:

$$L'(v) = L(v) \setminus \{\varphi(w), w \in N^2(v) \setminus (V(H) \cup D)\}.$$

We prove that the hypotheses of Lemma 2.30 are satisfied.

Suppose $v \in B_1'$. Now $|N^2(v) \cap B_2'| = |N(v) \cap B_2'| + \sum_{w \in N(v)} |N(w) \cap B_2'|$. By Lemma 2.27, for each $w \in V(R)$, $|N(w) \cap B_2'| \leqslant 1$. Moreover, if $w \in N(v) \setminus V(R)$, then $|N(w) \cap B_2'| = 0$, unless $w = b_2$. Since B_1 and B_2 are disjoint, we have $b_2 \notin N(v)$, and we get

$$|N^2(v) \cap B_2'| \leqslant 1 + |N(v) \cap V(R)| \leqslant 11.$$

Suppose $v \in B_1' \setminus T_1$. By definition, v is distance at least four from T (in $V(R)$), hence at distance at least three (in $V(R)$) from $N[T]$, the set of colored vertices of $V(R)$. So the only colored neighbors of v in G^2 are in $\{b_1, b_2\} \cup (N(b_1) \setminus B_1')$. Hence, we have

$$|L'(v)| \geqslant k + 2 - (2 + k - |B_1'|) = |B_1'|.$$

Suppose $v \in T_1$. By construction, its colored neighbors in G^2 are in $\{b_1, b_2\} \cup (N(b_1) \setminus B_1') \cup T \cup T^{(1)}$. Since $|T| + |T^{(1)}| \leqslant 44$, we have $|L'(v)| \geqslant |B_1'| - 44$.

We already saw that $|T_1| \leqslant |T^{(2)} \cup T^{(3)}| \leqslant 400 + 4000 = 4400$. There are $r + 1$ edges in the region R (in G'). Every such edge (except b_1b_2 if it exists) corresponds to a path containing a vertex in B_1. By Lemma 2.27, each vertex in B_1 accounts for at most nine of them. Therefore, $|B_1| \geqslant \frac{r}{9}$. Observe also that $|N[T] \cap B_1| \leqslant 6$ since $|T \cap B_1| = 2$ and, by Lemma 2.27, every vertex of $B_1 \cup B_2$ has at most one neighbor in each of B_1 and B_2. We thus obtain:

$$|B_1'| \geqslant |B_1| - |N[T] \cap B_1| \geqslant \frac{r}{9} - 6 \geqslant 52811.$$

We can thus apply Lemma 2.30 to find an L'-coloring of H.

It remains to color the vertices in D. Note that each has $k + 2$ colors and at most $2\sqrt{k}$ neighbors. So we can greedily color the vertices in D. $\square$

2.6 Finding a large region

Our goal in this section is to prove Proposition 2.24, i.e. to find a large region in G. In the traditional discharging proofs, this is done using weight transfers to reach a contradiction whenever G does not contain such a region. However, here, we reach a contradiction only using Euler's formula and the pigeonhole principle.

Recall that, due to the construction of G', finding a large region in G is means finding a large set of consecutive faces of length 2 in G'. Moreover, recall also that faces of length at most 2 disappear when constructing G'''. Therefore, if we manage to find a vertex u with large degree in G' (for example a big vertex), but small degree in G''', then many edges incident to u in G' are spread across few neighbors of u. The pigeonhole principle then implies that these edges must create adjacent 2-faces. The goal of this section is to settle properly this argument.

We first recall a result from [Bonamy *et al.*, 2019a] (Lemma 3.6 in that paper) allowing us to find a vertex in G'' with few neighbors in G'''. Observe that the general context of [Bonamy *et al.*, 2019a] is planar graphs with girth at least 5. However, the proof of Lemma 2.31 uses only that G has no 4-cycles.

Lemma 2.31 ([Bonamy *et al.*, 2019a]). *There exists $b_1 \in B$ such that $\deg_{G'''}(b_1) \leqslant 40$ and $\deg_{G'''[B]}(b_1) \leqslant 10$.*

Our goal is to apply a pigeonhole-like argument to find a large number of consecutive edges between two vertices in G''. To this end, we first need to control the degrees of vertices in G''. We begin with a definition. The *half-edges* of G' are the elements of the multiset of pairs (u, e) where e is an edge incident to u. Note that when e is a loop around u, there are still two half-edges (u, e). Observe also that since we fixed a plane embedding of G, there is a natural cyclic ordering of the half-edges around each fixed vertex u.

Lemma 2.32. *If e is a loop around a vertex v in G', then one of the half-loops induced by e must be followed or preceded by a half-edge (v, vw) with $v \neq w$.*

Proof. By Lemma 2.25, every loop has type 5. So let x_e and x'_e denote the vertices in G that merged into v to form e in G'. By Lemma 2.20, either $\deg(x_e) > 2$ or $\deg(x'_e) > 2$; by symmetry, assume $d(x_e) > 2$. Among all neighbors of x_e in G, other than x'_e and v, choose w to be one that immediately precedes or follows x'_e.

If w is not suppressed in G', then the half-edge (v, vw) precedes or follows (v, e) or (v, e'). Note that $vw \notin E(G)$ since otherwise $vwx_ex'_e$ is a 4-cycle in G. Thus we have $v \neq w$ in G' and the lemma is true. So assume that w is suppressed. Now w has degree 2 in G. Let x be the neighbor of w other than x_e. Since x_e is small, Lemma 2.19 ensures that x has degree at least 3 in G; hence, it is not suppressed in G'. Therefore, the half-edge (v, vx) precedes or follows (v, e) or (v, e'). Again, $vx \notin E(G)$ since otherwise $vxwx_e$ is a 4-cycle in G. Thus $x \neq v$ in G' and the lemma is true. □

Lemma 2.32 implies the following relationship between degrees of vertices in G'' and in G'.

Corollary 2.33. *Every $v \in V(G')$ satisfies $\deg_{G''}(v) \geqslant \frac{\deg_{G'}(v)}{5}$.*

Proof. Suppose $v \in V(G')$ and consider the half-edges around v in G'. By definition, there are $\deg_{G'}(v)$ half-edges around v and $\deg_{G''}(v)$ of them are not half-loops. So it suffices to prove that the number of half-loops around v is at most four times the number of the other half-edges, i.e., at most $4\deg_{G''}(v)$.

Suppose $w \in N_{G'}(v)$. Consider the two half-edges (v, e) and (v, f) such that (v, e), (v, vw) and (v, f) are consecutive around v. Let $F(w)$ be the maximum subset of $\{(v, e), (v, f)\}$ containing only half-loops. Lemma 2.32 ensures that, for every loop, one of its half-loops appears in $F(w)$ for some $w \in N_{G'}(v)$. Therefore, the number of half-loops around v is at most

$$2\left|\cup_{w \in N_{G'}(v)} F(w)\right| \leqslant 4|N_{G'}(v)| = 4\deg_{G''}(v).$$

This concludes the proof, since

$$\deg_{G'}(v) \leqslant \deg_{G''}(v) + 4\deg_{G''}(v) = 5\deg_{G''}(v).$$ □

Consider the vertex b_1 obtained by Lemma 2.31. By Corollary 2.33, we have

$$\deg_{G''}(b_1) \geqslant \frac{\deg_{G'}(b_1)}{5} \geqslant \frac{\deg_G(b_1)}{5} \geqslant \frac{\sqrt{k}}{5}.$$

Using a pigeonhole argument, we will see that b_1 has some neighbor b_2 such that at least $\frac{\sqrt{k}}{5\times 40}$ consecutive edges incident to b_1 end at b_2. Note that Proposition 2.24 almost follows from this result (with $\frac{\sqrt{k}}{50}$ replaced by $\frac{\sqrt{k}}{200}$). We only need to refine this argument to show how to force $b_2 \in B$, i.e., $b_2 \notin S'$, where $S' = V(G') \setminus B$. To this end, we show that small vertices are incident to few consecutive edges in G''.

Lemma 2.34. *If $v \in B$ and $w \in S'$, then (v, w) is on the boundary of at most eight consecutive faces of length 2 in G''.*

Proof. Pick $v \in B$ such that there is an edge $vw \in E(G')$, with $w \in S'$. We consider each possible type of edge in G' between v and w. The type 3 edges are a special case, which we postpone to the end. Since G is simple, at most one edge vw of G' has type 1. Similarly, if G' has two edges e_1 and e_2 of type 2, then $x_{e_1} \neq x_{e_2}$. Thus $vx_{e_1}wx_{e_2}$ is a 4-cycle in G, a contradiction. So G' has at most one edge of type 2. Since $v \in B$ and $w \in S'$, G' has no edge of type 4, 5, or 6.

Only type-3 edges remain. We assume such an edge exists, since otherwise the lemma holds. Note that G' has no edge of type 4 (since $v \in B$), nor of type 1 (since G has no 4-cycle), nor of type 5 or 6 (since $w \in S'$). So G' has at most one edge f not of type 3, and f, if it exists, has type 2. Thus, edge f separates two blocks of consecutive type-3 edges. To prove the lemma, it suffices to prove that each such block has size at most four.

Assume that $e_1, \ldots, e_5$ are edges of type 3 that are consecutive in G''. We now prove that the hypotheses of Lemma 2.22 are satisfied by the subgraph of G induced by the vertices inside the cycle $vx_{e_1}y_{e_1}wy_{e_5}x_{e_5}$. Since each edge e_i has type 3, the first hypothesis holds.

To prove the second hypothesis holds, assume that some vertex x is not adjacent to v, but x lies inside some cycle $C = vx_{e_i}y_{e_i}wy_{e_{i+1}}x_{e_{i+1}}$. Note that x is not a neighbor of y_{e_i} or $y_{e_{i+1}}$, since they both have degree 2; nor of w since e_i and e_{i+1} are consecutive edges in G''. Note that e_i and e_{i+1} bound a face of length 2 in G'' so every vertex inside the cycle C disappears when we construct G'. Thus, all these vertices are small, and either lie in S_1 or lie in $S \setminus N[B]$ and have degree 2 in G. Hence, v is the only big vertex inside or on C and $xv \notin E(G)$; so $x \notin \cup_{i \geq 1} S_i$.

Since $x \notin S_1$, x has degree 2 and its two neighbors, say y and z, lie in S. Applying Lemma 2.19 to edges xy and xz, we get that $y, z \in N[B]$. This implies that both y and z are neighbors of v, so $xyvz$ is a 4-cycle in G, a contradiction. Therefore, no such x exists.

Now Lemma 2.22 yields a contradiction, since G cannot contain this configuration. $\square$

We can now finish the proof of Proposition 2.24.

Proof of Proposition 2.24. Let b_1 be a vertex in G''' guaranteed by Lemma 2.31. For each small neighbor v of b_1 in G''' and edge vb_1, Lemma 2.34 ensures that in G'' edge vb_1 corresponds to at most nine edges between b_1 and v. Since $\deg_{G'''}(b_1) \leqslant 40$, the number of such edges is at most $9 \times 40 = 360$. However, by Corollary 2.33, we have $\deg_{G''}(b_1) \geqslant \frac{\deg_G(b_1)}{5} \geqslant \frac{\sqrt{k}}{5}$. Thus, there must exist a big neighbor b_2 of b_1 in G'' such that there are at least

$$\frac{\frac{\sqrt{k}}{5} - 360}{\deg_{\mathbf{G'''[B]}}(b_1)} \geqslant \frac{\sqrt{k}}{50} - 36$$

consecutive edges b_1b_2 in G''. By definition, these edges form a region of size $\frac{\sqrt{k}}{50} - 37$ in G. $\square$

2.7 Extension to correspondence coloring

The results of Sections 2.4, 2.5 and 2.6, namely Lemmas 2.17 through 2.27 and Propositions 2.23 and 2.24 prove that Theorem 2.15 holds for choosability. In this section, we prove that this result can actually be extended to a stronger notion.

2.7.1 Correspondence coloring

To prove that some configurations are reducible, it is often convenient to identify vertices. This works very well for the original vertex coloring problem, see for example the proof of the following theorem.

Theorem 2.35 (Folklore)**.** *Planar graphs are 5-colorable.*

Proof. Assume that the theorem is false and consider a counterexample G which minimizes $|V(G)|$. Since G is planar, it has a 5^--vertex u.

By minimality $G \setminus u$ is 5-colorable. If only 4 colors are used in $G \setminus u$, then it is easy to extend the coloring to G, since there is always at least one color available for u.

However, this approach does not work when u is a 5-vertex and all the colors are used on $N_G(u)$. To avoid this situation, we identify two non-adjacent vertices u_1, u_2 from $N_G(u)$ in $G \setminus u$ (such vertices exist since G is planar). The resulting graph H is still planar, and has two vertices less than G. Thus, H has a 5-coloring. Unfolding the identification, we obtain a coloring of $G \setminus u$ where only four colors are used on $N_G(u)$, hence we may extend the coloring to G, a contradiction. $\square$

However, in the list coloring setting, this kind of identification does not work anymore since two different vertices may have different lists (even disjoint ones). The notion of correspondence coloring was introduced in [Dvořák and Postle, 2018] to overcome this problem. This new coloring is a generalization of list coloring, as we shall see. Moreover, the authors prove that with this new type of coloring, some identifications can be made, and use it to tackle Erdős' question about 3-choosability of $(C_4, \dots, C_p)$-free planar graphs.

We now give the definition of correspondence coloring. Given a graph G and a function $f : V(G) \to \mathbb{N}$, an *f-correspondence assignment* C is given by a matching C_{vw}, for each $vw \in E(G)$, between $\{v\} \times \{1, \dots, f(v)\}$ and $\{w\} \times \{1, \dots, f(w)\}$. We say that each vertex x has $f(x)$ *available colors*. A *k-correspondence assignment* is an f-correspondence assignment where $f(v) = k$ for all $v \in V(G)$. Given an f-correspondence assignment C, a *C-coloring* is a function $\varphi : V(G) \to \mathbb{N}$ such that $\varphi(v) \leqslant f(v)$ for each $v \in V(G)$, and, for each edge $vw \in E(G)$, the pairs $(v, \varphi(v))$ and $(w, \varphi(w))$ are nonadjacent in C_{vw}. The *correspondence chromatic number* of G is the least integer k such that, for every k-correspondence assignment C of G, the graph G admits a C-coloring. It is denoted by $\chi_{corr}(G)$. Note that if G is k-degenerate, then coloring greedily in an appropriate order shows that $\chi_{corr}(G) \leq k + 1$.

Note also that if L is a k-list assignment for a graph G, we can construct a k-correspondence assignment C such that G has a C-coloring if and only if it has an L-coloring. For every edge vw, C_{vw} contains all the edges between (v, i) and (w, j) when the i-th element of $L(v)$ equals the j-th element of $L(w)$. Therefore, correspondence coloring is a generalization of list coloring, and we have $\chi_\ell(G) \leqslant \chi_{corr}(G)$.

2.7.2 Theorem 2.15 revisited

In this subsection, we extend Theorem 2.15 to the setting of correspondence coloring.

Theorem 2.36. *There exists Δ_0 such that if G is a plane graph with no 4-cycles and with $\Delta \geqslant \Delta_0$, then $\chi_{corr}(G^2) \leqslant \Delta + 2$*

To prove this theorem, we again apply the discharging method, without discharging. Take $\Delta_0 = 2642900^2 = 6984920410000$, and fix $k \geqslant \Delta_0$, as well as a minimum counterexample G. Let C be a $(k+2)$-correspondence assignment for G^2 such that G^2 has no C-coloring. So C assigns, to each pair of vertices (v, w) adjacent in G^2, a partial matching C_{vw} between $\{v\} \times \{1, \dots, k+2\}$ and $\{w\} \times \{1, \dots, k+2\}$.

Note that Proposition 2.24 does not depend on the type of coloring we consider. Hence it is still valid in this new setting: G has a large region. Moreover, we claim that all the results of Section 2.4 (Lemmas 2.17 through 2.27) still hold, since we color vertices using only that they have more available colors than colored neighbors. Therefore, G does not contain any of the small configurations. Thus, it only remains to prove that large regions are also reducible in the new setting, i.e. to extend Proposition 2.23 for G.

Proposition 2.37. *Every r-region of G satisfies $r \leqslant 52821$.*

Assuming this proposition holds, we can conclude. Indeed, Propositions 2.24 and 2.37 imply that $\frac{\sqrt{k}}{50} - 37 < 52821$, i.e., that $k < 2642900^2 = 6984920410000 = \Delta_0$, a contradiction.

It thus remains to prove that large regions are reducible, by generalizing Lemma 2.30. The argument using kernel-perfect orientations is no longer valid, since Lemma 2.29 does not extend to correspondence coloring. Instead of using Lemma 2.29 as a black box, we now have to go more into the details to construct a suitable coloring. As we will see, using low-level arguments allows us to obtain a better bound. The downside is that the proof is much more technical.

Lemma 2.38. *Let H be a graph covered by two disjoint cliques, B_1 and B_2, each of size p. Suppose there exist $T_1 \subset B_1$ and $T_2 \subset B_2$, and a function f satisfying the four properties below. If $p \geq 5863$, then every f-correspondence assignment C admits a C-coloring.*

1. *For each $v \in (B_1 \setminus T_1) \cup (B_2 \setminus T_2)$, we have $f(v) \geqslant p$.*
2. *For each $v \in T_1 \cup T_2$, we have $f(v) \geqslant p - 44$.*
3. *$|T_1| \leqslant 4400$ and $|T_2| \leqslant 4400$.*
4. *$\Delta(H) - p + 1 \leqslant 11$.*

Proof. We begin with a global (and informal) presentation of the proof. Let A be a subset of $B_1 \setminus T_1$ with $|A| = \Delta(H) + 1 - p$. Since each vertex $v \in (B_1 \setminus T_1) \cup (B_2 \setminus T_2)$ has $f(v) \geq p$ and $\Delta(H) - |A| = p - 1$, it is easy to greedily C-color all vertices of $H - A$. For example, greedily color all vertices of T_2, followed by those of $B_2 \setminus T_2$, followed by those of T_1, followed by those of $B_1 \setminus (T_1 \cup A)$. This greedy coloring is possible because at the time we color each vertex it has more available colors than colored neighbors.

We generally follow this approach. However, we modify it so that after we color $H - A$ each vertex in A still has $|A|$ available colors, and we can extend the coloring to A. To do this, for each vertex $v \in A$ we will repeatedly "save a color", before greedily coloring the other vertices. To accomplish this we pick vertices $w \in N(v) \cap B_2$ and $x \in B_1 \setminus N(w)$. Now we color w and x with some colors α and β (possibly with $\alpha = \beta$) such that α and β forbid at most one color on v. For each $v \in A$, we must save a color $|N(v) \cap B_2|$ times. After doing so, we color the remaining vertices greedily (as in the previous paragraph), ending with the vertices of A. The only change is that we must ensure that each of the final 11 vertices we color in B_2 has no colored neighbor in B_1.

We now apply this approach, beginning by the choice of A. In the process of saving colors for vertices in A, we color at most 11^2 vertices in B_1. Each of these forbids at most 11 vertices in B_2 from appearing among the final 11 in B_2, for a total of at most 11^3 vertices in B_2 forbidden. Similarly, we color at most 11^2 vertices in B_2, and these are obviously forbidden from appearing among the final 11 vertices in B_2. Thus, we can choose the desired 11 final vertices in B_2 (after saving colors for the vertices in A), since $|B_2| \geq |T_2| + 11^3 + 11^2 + 11$.

Note that, while saving colors for some vertex $v \in A$, we color all neighbors of v in B_2. As a result, we need that no two vertices in A have a common neighbor in B_2. Each vertex $v \in A$ has at most 11 neighbors in B_2, and each of these neighbors has at most 10 other neighbors in B_1. Thus, each $v \in A$ forbids at most 11(10) other vertices from A. So, to pick the desired A, we need $|B_1| > |T_1| + 10(110 + 1)$.

Now, for each $v \in A$, we repeat the following $|N(v) \cap B_2|$ times. Choose uncolored vertices $w \in N(v) \cap B_2$ and $x \in B_1 \setminus N(w)$. Note that if $N(v) \subset B_1$, there is nothing to do at all, hence we may assume that the vertex w exists. Let $g(v)$, $g(w)$, and $g(x)$ denote the number of remaining available colors for v, w, and x. Without loss of generality, we assume that the bounds of Hypotheses 1. and 2. are tight, so that $f(y) = p - 44$ for all $y \in T_1 \cup T_2$, and $f(y) = p$ otherwise. Since $A \cap T_1 = \varnothing$, we have $f(v) = p \geqslant f(w)$, hence we may assume that C_{vw} saturates $\{w\} \times \{1, \cdots, f(w)\}$ (otherwise, add arbitrary edges until this is the case). Thus, each color available for w forbids a color for v; similarly for colors available for x. By the pigeonhole principle, if $g(w) + g(x) > p$, then there exist colors α and β, available for w and x respectively, that both forbid the same color on v. Suppose that this far we have saved a total of i colors for vertices in A. Therefore, the i colored vertices of B_2 forbid i colors for w, and its neighbors in B_1 forbid at most 11 colors, so that we have $g(w) \geq f(w) - i - 11 \geq p - i - 11 \geq p - 131$. Similarly, we get $g(x) \geq p - 131$. We can assume that $g(v) \leq f(v) \leq p$. And clearly $2(p - 131) > p$. Thus, the desired colors α and β exist. □

This concludes the proof of Theorem 2.36. It is worth noting that the Δ_0 given by our proof of Theorem 2.36, namely 2642900^2, is much smaller than that arising from our proof of Theorem 2.15, namely 23769500^2. This comes from the fact that instead of using the generic argument from Lemma 2.29, we directly construct a suitable coloring. This difference is not so meaningful: these bounds are large, and can certainly be optimized by considering more technical proofs. The real question, which probably requires new ideas, is to bring them down to a more reasonable value, say less than 100.

2.8 Open questions

Observe that Brooks' theorem gives an infinite list of graphs such that $\chi > \Delta$. However, for every $\Delta \geqslant 3$, this list contains only one graph of given maximum degree Δ. In this setting, the first result of this chapter (Theorem 2.3) can be seen as a generalization of Brooks' theorem for k-th powers of graphs. However, observe that the bound we obtain for $k = 1$ is worse than the one given by Brooks' theorem. We can think of several ways for improving this result, the first one consists in proving that $k-1$ or k colors can be spared, instead of $k-2$. This would actually give the statement of Brooks' theorem in the case $k = 1$, and would also generalize the case $k = 2$ proven in [Cranston and Rabern, 2016]. Moreover, we also think that we can spare even more colors using stronger assumptions like $\Delta \geqslant 4$, or $\Delta \geqslant k$.

Another natural question is about the number of exceptions given by Theorem 2.3. Here the proof gives only little information about the structure of any given exception. However, we believe that some similar (but yet more involved) arguments could help to find a better description for exceptions.

The arguments used in the proof of Theorem 2.3 only work with coloring, not with list coloring. However, Conjecture 2.2 is stated in this more general setting. Thus, it would also be interesting to see whether Theorem 2.3 could be extended to the list coloring setting.

Regarding the main contents of this chapter (outside Section 2.2), the methods we present may actually lead to a better bound for Δ_0 in Theorem 2.36. However, obtaining a better bound means increasing the number and possibly the size of the bounded configurations. Since the goal of this chapter is to describe the methods, we decided not to optimize the bound in order to make the arguments as clear as possible. Moreover, we believe that even with some optimizations, this method cannot bring down the bound on Δ_0 to some reasonable integer (say, less than 100).

As shown by the examples depicted in Figures 2.13, 2.14 and 2.15, $\Delta + 2$ colors are not sufficient for small Δ. A natural question is then to ask for the minimum value of Δ_0 needed to ensure a $\Delta + 2$ bound.

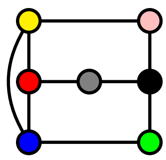


Figure 2.13 – Graph G with $\Delta = 3$ and $\chi(G^2) = 7$

In another direction, we believe that the tools we introduce in this chapter can be useful for extending our result to some more general settings, for example to study $L(p, q)$-labelings. However, it is not clear whether they can be generalized to color any power of planar graphs instead of only squares. For

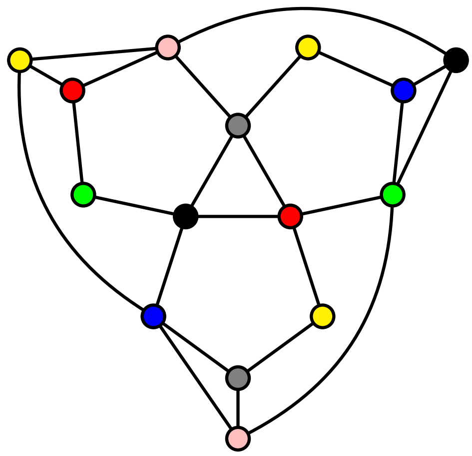


Figure 2.14 – Graph G with $\Delta = 4$ and $\chi(G^2) = 7$

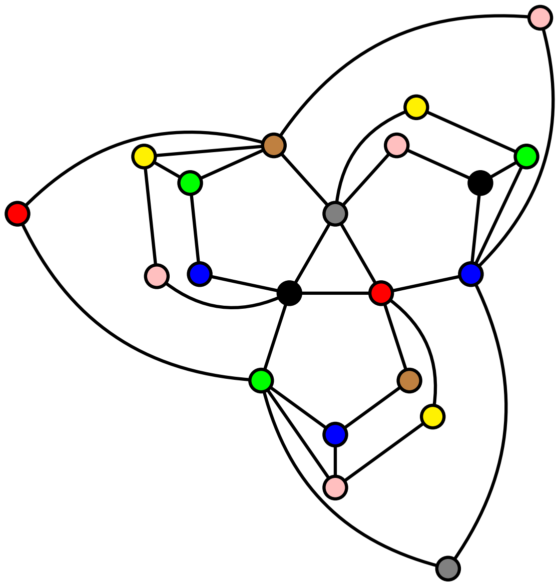


Figure 2.15 – Graph G with $\Delta = 5$ and $\chi(G^2) = 8$

this kind of coloring, the bound from [Agnarsson and Halldórsson, 2003]

$$\chi_\ell(G^k) = O(\Delta^{\lfloor k/2 \rfloor})$$

is known to be tight for example for Δ-ary trees of height $\frac{k}{2}$. However, it seems that the multiplicative constant has not been investigated so far. Extending the notion of regions by considering paths of length $2k$ between two big vertices may help to study $\chi(G^k)$, since the k-th power of a region still consists in two cliques with some edges in-between. However, the key point in our approach is to bound the number of such edges, and it is unclear that planarity is sufficient in this case.

Finally, a last question is whether we can get rid of the hypothesis that only finitely many cycles can be forbidden. If we forget this hypothesis, then C_4 may be allowed. However, the construction of Proposition 2.14 shows that all cycles whose length is 2 modulo 4 have to be forbidden. The resulting class of planar graphs is quite unusual. There are strong constraints on the sizes of adjacent faces. However, these local constraints do not seem sufficient to conclude: locally we cannot distinguish the graphs obtained from an even and an odd cycle by replacing each edge by a copy of $K_{2,t}$. However, the former graph needs only $\Delta + O(1)$ colors while it is not the case for the latter (see Proposition 2.14). Therefore, extending the result (if it holds) for this new class of graphs requires more involved arguments than local discharging.

Chapter 3

Separation of regular languages

Contents

The main part of this chapter is dedicated to an overview of the problems we consider. We present here the history of the field, as well as the basic objects that are involved. Finally, in Section 3.3, we contribute a complexity reduction that we proved with Thomas Place and Marc Zeitoun in 2017. It has since *been extended by them to a reduction between wider problems and published in [Place and Zeitoun, 2018a].*

3.1 Introduction

We begin this chapter with an overview of the questions we consider, whose history spans over several decades. These questions are instances of an emblematic problem of finite model theory, asking what kind of sets can be described using a given formalism. The sets we are interested in are the so-called regular languages of finite words, well-known for the wealth of formalisms describing them. We thus begin with a few words about regular languages.

3.1.1 A brief history of formal languages

The introduction of regular languages comes from the seminal article of [Kleene, 1951]. In this document is proved the famous Kleene's theorem, that establishes an equivalence between recognition by finite automata and regular expressions. Regular expressions are formed from simple languages (containing finitely many words) using three operations:

- Union: if E and E' are regular expressions denoting the languages L, L', then $E + E'$ denotes the language $L \cup L'$.
- Concatenation: if E and E' are regular expressions denoting the languages L, L', then EE' denotes the language $\{uu' \mid u \in L, u' \in L'\}$.
- Kleene's star, also called iteration: if E is a regular expression denoting the language L, E^* denotes the language $\{u^n \mid u \in L, n \geqslant 0\}$.

In particular, note that the set of all words over an alphabet A can be written as A^*. A regular language is thus the language described by a regular expression. For example, the regular expression aA^* denotes the language of all words beginning with the letter a.

Since then, other characterizations of regular languages have been established, showing that regular languages can be constructed in several equivalent ways: with regular expressions, automata, morphisms to finite monoids, or with logical sentences, as shown independently by [Büchi, 1960; Elgot, 1961; Trakhtenbrot, 1961]. In this introduction, we focus on two such formalisms: regular expressions and logical sentences. Two other formalisms, namely finite automata and finite morphisms, will be described later, in Section 3.2.

We now define a logical formalism for describing regular languages. Given an alphabet A, we consider the *first-order* (FO for short) sentences constructed from atomic predicates $a(x)$ for $a \in A$ (representing that the position x is labeled by the letter a) and $x < y$ (representing that position x is before position

y). This means that a sentence is constructed from the atomic predicated by taking conjunctions, negations and by quantifying over variables. Such a sentence defines the language of all the (finite) words satisfying it. For example, the sentence

$$\exists x, a(x) \wedge \neg(\exists y, y < x)$$

can be read as "there is a position x, whose label is a, and such that there is no position y before". Therefore, it denotes the language of words whose first letter is an a.

A famous theorem of Büchi states that regular languages are the ones definable by a *monadic second-order* sentence (MSO for short). MSO is an extension of FO, where we allow also to quantify over *sets* of variables.

Theorem 3.1 ([Büchi, 1960; Elgot, 1961; Trakhtenbrot, 1961]). *A language L is regular if and only if it is definable by an MSO sentence.*

Moreover, this equivalence is effective: we can construct an MSO sentence recognizing a regular language L from any other representation of L, and conversely.

Recall that our goal is to study instances of the following typical problem of finite model theory: determining what kind of sets can be described using a given formalism. Here, we consider this problem for sets of words, i.e. languages. The two formalisms we consider in this introduction are regular expressions and MSO sentences, both of them defining only regular languages. The question thus becomes: which *regular* languages are defined by (restrictions of) these formalisms?

When considering regular expressions, a first example is to ask for languages described by regular expressions that do not use Kleene's star. It is easy to see that these languages are exactly finite languages. A generalization of this question is the so-called *star-height* problem: given a regular language, what is the minimum number of nested stars needed in a regular expression that represents it? This problem was originally stated in [Eggan *et al.*, 1963] in a "restricted" setting (with regular expressions using only union, concatenation and iteration) and in [McNaughton, 1960] in a "generalized" setting (where regular expressions can use complement as an additional operation). While the restricted star-height hierarchy was known to be infinite (see [Eggan *et al.*, 1963]), computing the restricted star-height of a given regular language remained an open problem for roughly 20 years. We know several proofs (with different levels of readability) that the star-height problem is decidable, see for example [Hashiguchi, 1983, 1988; Kirsten, 2005; Bojańczyk, 2015]. However, much less is known for generalized star-height (see [Brzozowski, 1980] for a survey). Right now, it is not known whether there exists a regular language of generalized star-height at least two. Consequently, the generalized star-height problem is not known to be decidable. Even characterizing languages with generalized star-height 0, also known as *star-free* languages, is already a non-trivial problem, which was solved by [Schützenberger, 1965]. We denote by SF the class of star-free languages. The question thus becomes to characterize languages in SF.

Observe that while being a star-free regular expression (or an expression of given star-height) is a syntactical notion, being a star-free language is a semantical one: it corresponds to the existence of a star-free regular expression denoting a given language. In particular, a star-free language can be denoted by a non star-free expression, as long as it is also denoted by another regular expression being star-free.

Similarly to characterizing languages in SF, another question was raised using the logical formalism: which languages can be defined by an FO sentence (i.e. without using second-order quantification)? Note that such languages are necessarily regular, since FO is a fragment of MSO.

Surprisingly, the answer to both these questions is the same: the class SF corresponds to the class of languages definable by an FO sentence. Abusing notations, we denote by FO the class of languages defined by an FO sentence. The previous result can then be rephrased as SF = FO. This equality, as well as a characterization of star-free languages, comes from the forthcoming Theorem 3.2. Due to this correspondence SF = FO, star-free languages then became a milestone in the study of any formalism that represents regular languages. We refer to the surveys of [Diekert *et al.*, 2008; Straubing, 2018] for a more detailed historical presentation of star-freeness.

Theorem 3.2 ([Schützenberger, 1965; McNaughton and Papert, 1971])**.** *Let L be a regular language. The following statements are equivalent:*

- *L is definable in FO.*
- *L is star-free, i.e. $L \in$ SF.*
- *The minimal automaton of L is counter-free.*
- *The syntactic monoid of L is aperiodic.*

This theorem uses many undefined notions: however, their definitions do not really matter for this example. The point is that counter-freeness and aperiodicity are all syntactic properties of automata and monoids, respectively. The other important remark is that the minimal automaton and the syntactic monoid of a regular language L are canonical objects that can be effectively computed from any representation of L. They will be introduced later, in Section 3.2.

Theorem 3.2 thus establishes a link between two semantic properties (being definable in FO, or in SF) and syntactical ones (the last two). In particular, note that the last two properties are decidable: given a language L, we can compute its minimal automaton (or its syntactic monoid), and then test the corresponding property. We thus get an effective characterization of star-free language: given a regular language L, we can decide whether L is star-free.

Deciding whether a given language lies in SF is actually a special instance of the so-called *membership* problem. Given a class of languages $\mathcal{C}$, the $\mathcal{C}$-membership problem takes a regular language as input and asks whether this language lies in $\mathcal{C}$. Note that the $\mathcal{C}$-membership problem is a central problem in theoretical computer science, since proving that the class $\mathcal{C}$ is decidable amounts to deciding the $\mathcal{C}$-membership problem.

The motivation for studying this problem comes from the fact that solving the $\mathcal{C}$-membership problem relies on a deep understanding of the expressive power of $\mathcal{C}$. This is what is hidden in the proof of Theorem 3.2. Indeed, we first need to find (and prove) some properties satisfied by every star-free language. By contrapositive, this requires to understand which properties prevent a language from being star-free. Surprisingly, this is the easiest part of the proof. The harder part actually comes from the converse result: assuming that these properties are satisfied, we have to prove that the considered language is actually star-free. This means that we have to design algorithms to construct a star-free expression denoting the considered language, just by knowing that it has a recognizer satisfying a syntactic property. This step also requires to understand what can be described with star-free expressions.

The approach used here for SF-membership is actually the one we often use to handle $\mathcal{C}$-membership for other classes $\mathcal{C}$. The goal is to prove a link between a semantical property (being definable in $\mathcal{C}$) and a syntactic property of some recognizer, which is easier to decide. The ultimate goal is thus to find a *decidable characterization* of the class $\mathcal{C}$, i.e. a property equivalent to testing membership in $\mathcal{C}$ which has to be decidable.

The last assertion of Theorem 3.2 is actually a typical example of the type of condition we look for. To see this, we need to define aperiodicity: a finite monoid M is *aperiodic* if for every $s \in M$, we have $s^{\omega(M)+1} = s^{\omega(M)}$, where $\omega(M)$ is the least positive integer such that for every $s \in M$, we have $s^{2\omega(M)} = s^{\omega(M)}$. The existence of such an integer will be proven in Section 3.2. This characterization can thus be summarized by the equation $x^{\omega(M)+1} = x^{\omega(M)}$, which has to be satisfied by every element of the syntactic monoid of the considered language. Obtaining an equation (or a set of some equations) characterizing a class is often the approach followed to prove decidability of $\mathcal{C}$-membership for other classes $\mathcal{C}$.

3.1.2 Some examples of classes

The class of FO-definable languages is far from being the the only interesting fragment of regular languages: many attempts have been made to understand the expressiveness of other classes of languages (i.e. to solve the corresponding membership problem). Even if several classes not related to FO were considered, see for example [Margolis and Pin, 1985; Esik and Ito, 2003], we focus here on subclasses of FO. Various hierarchies were defined to stratify FO languages in several classes. We present here three historical examples: the first one is the *quantifier alternation hierarchy*. In this hierarchy, languages are sorted according to syntactic restrictions on sentences defining them. The two other examples are hierarchies stratifying the class SF of star-free languages according to syntactic restrictions on regular expressions. Note that since SF = FO, these three hierarchies are actually three stratifications of the same class.

As the name "quantifier alternation" suggests, the classes of this hierarchy are defined by counting the number of quantifier alternations in a sentence in prenex normal form. A sentence is *prenex* if it is written as $Q_1 x_1 \cdots Q_n x_n \varphi$ with φ quantifier-free and $Q_i \in \{\forall, \exists\}$ for $1 \leqslant i \leqslant n$. It is easy to see that every sentence can be put in prenex form without changing the language it describes. In this context, a *block* of quantifiers is a sequence of consecutive identical quantifiers. We define Σ_i as the class of prenex sentences with:

- either exactly i blocks of quantifiers, the first one being existential,
- or at most $i - 1$ such blocks.

For example, if ψ is a quantifier-free formula, then $\exists x \exists y \psi$ has a single block of two existential quantifiers, hence it is a Σ_1 formula (and also a Σ_2 formula). On the other hand, $\exists x \forall y \forall z \psi$, as well as $\forall x \forall y \psi$, are Σ_2 formulas (but not Σ_1 formulas). Observe that Σ_i is not closed under complement, hence it is convenient to define $\mathcal{B}\Sigma_i$ as its Boolean closure. Similarly to the class FO, we again abuse notation by saying that a language is in one of these classes if it is defined by a sentence in the corresponding class. The same remark applies: being a Σ_i sentence is a syntactical notion, while being a Σ_i language is a semantical one.

The motivation for introducing these classes is twofold. First, an empirical argument is that, usually, the more quantifier alternations a mathematical statement has, the more complex it is, independently of how many quantifiers there are in total. A second motivation comes from complexity questions. For FO-definable languages, many algorithms have an unavoidable non-elementary complexity. Consider for example the translation from a given FO sentence into an automaton recognizing the same language. When restricted to Σ_i, the non-elementary lower bound of the generic case does not hold: we can give an algorithm whose time complexity is bounded by a tower of exponentials of height i. In particular, observe that if we can decide the membership problem for every level of the hierarchy, then we can get an idea of how complex a given language is.

We then present two other examples of hierarchies. These hierarchies stratify the class of star-free languages by considering restrictions on regular expressions. However, recall that $\mathsf{SF} = \mathsf{FO}$, hence these hierarchies can be also seen as stratification of the FO-definable languages. These hierarchies are the *dot-depth hierarchy* [Cohen and Brzozowski, 1971], and the *Straubing-Thérien hierarchy* [Straubing, 1981; Thérien, 1981]. They are often called *concatenation hierarchies*. This comes from the fact that both of them stratify the class of star-free languages by counting the number of alternations between concatenations and complements needed in a star-free regular expression describing a given language. More precisely, these hierarchies contain two kinds of levels: half-levels and full-levels. These are constructed using the same inductive scheme. Starting from a class at level 0, we define the higher levels by applying two kind of operations: the Boolean closure and the polynomial closure.

The *polynomial closure* $\mathrm{Pol}(\mathcal{C})$ of a class $\mathcal{C}$, is the smallest class containing $\mathcal{C}$ and closed under union and *marked concatenation*: if $K, L \in \mathrm{Pol}(\mathcal{C})$ and a is a letter, then $KaL \in \mathrm{Pol}(\mathcal{C})$.

More precisely, the construction scheme is the following: for every integer n,

- The *(half-)level* $n + \frac{1}{2}$ is the polynomial closure of level n.
- The *(full-)level* $n + 1$ is the Boolean closure of level $n + \frac{1}{2}$.

The difference between the dot-depth and the Straubing-Thérien hierarchies comes from the class at level 0. Denoting by ε the empty word, level 0 of the dot-depth hierarchy is $\{\varnothing, \{\varepsilon\}, A^* \setminus \{\varepsilon\}, A^*\}$, while level 0 of the Straubing-Thérien hierarchy is $\{\varnothing, A^*\}$.

Due to the equality $\mathsf{SF} = \mathsf{FO}$, it is not surprising that the three hierarchies we described are linked. In particular, the dot-depth hierarchy is tightly tied to the quantifier alternation hierarchy, as shown by the following result.

Theorem 3.3 ([Thomas, 1982])**.** *For every integer n, a language L lies in the n-th (resp. $(n + \frac{1}{2})$-th) level of the dot-depth hierarchy if and only if it is defined by a $\mathcal{B}\Sigma_n$ (resp Σ_n) sentence using additional predicates:*

- *the nullary predicate ε, satisfied only by the empty word.*
- *the binary predicate $+1$ denoting successor.*
- *the unary predicates $\min(x)$ and $\max(x)$, denoting that x is the leftmost/rightmost position.*

We denote by $\mathcal{B}\Sigma_n(\varepsilon, +1, \min, \max)$ and $\Sigma_n(\varepsilon, +1, \min, \max)$ the corresponding sets of sentences with this enriched signature. A similar connection was also established between logical classes and the Straubing-Thérien hierarchy. This is illustrated by Figure 3.1.

$\boxed{\text{Straubing-Thérien hierarchy}}$

$$\begin{tikzcd} \frac{1}{2} \arrow[r, "\mathrm{Bool}"] \arrow[d, leftrightarrow] & 1 \arrow[r, "\mathrm{Pol}"] \arrow[d, leftrightarrow] & \frac{3}{2} \arrow[r, "\mathrm{Bool}"] \arrow[d, leftrightarrow] & 2 \arrow[r, "\mathrm{Pol}"] \arrow[d, leftrightarrow] & \frac{5}{2} \arrow[d, leftrightarrow] & \cdots & \mathsf{SF} \arrow[d, leftrightarrow] \\ \Sigma_1 \subset & \mathcal{B}\Sigma_1 \subset & \Sigma_2 \subset & \mathcal{B}\Sigma_2 \subset & \Sigma_3 & \cdots & \mathsf{FO} \end{tikzcd}$$

$\boxed{\text{Quantifier alternation hierarchy}}$

Figure 3.1 – Straubing-Thérien and quantifier alternation hierarchies

Theorem 3.4 ([Perrin and Pin, 1986])**.** *For every integer n, a language lies in level n (resp. $n + \frac{1}{2}$) of the Straubing-Thérien hierarchy if and only if it is defined by a $\mathcal{B}\Sigma_n$ (resp. Σ_n) sentence, this time with the usual signature consisting only of ordering and letter predicates.*

Recall that $\mathcal{B}\Sigma_n$ is the Boolean closure of Σ_n, i.e. closure under the logical operations $\wedge, \vee$ and $\neg$. In the Straubing-Thérien hierarchy, the corresponding operation (for obtaining level n from level $n - \frac{1}{2}$) is closure under union, intersection and complement. In particular, observe that each logical operation corresponds to an operation on languages.

A similar connection holds for the other operation. Indeed, observe that Σ_{n+1} is the closure of $\mathcal{B}\Sigma_n$ under conjunction, disjunctions and existential quantifications. In view of the previous result, this closure operation corresponds to the polynomial closure of the corresponding classes of languages. To illustrate this, observe that marked concatenations can be interpreted in a logical setting using existential quantifications: given two languages K, L defined by two sentences φ, ψ and a letter a, we may define KaL by

$$\exists x \quad a(x) \wedge \varphi_{<x} \wedge \psi_{>x}$$

where $\varphi_{<x}$ and $\psi_{>x}$ are obtained from φ and ψ by restricting the scope of their quantifiers. The operations $\varphi \mapsto \varphi_{<x}$ and $\psi \mapsto \psi_{>x}$ are standard and fairly simple: we use the predicate $<$ to compare each variable occurring in φ (resp. ψ) to x, in order to restrict the domain of quantification to positions before (resp. after) x.

3.1.3 The membership problem for hierarchies

For now, we introduced three hierarchies: dot-depth, Straubing-Thérien and quantifier alternation. As we already saw, the last two coincide. We may thus reduce the study of these hierarchies to only two of them. Moreover, note that dot-depth and Straubing-Thérien are defined using the same generic construction. It is then not surprising that decidability of these two hierarchies is linked, as shown by the following theorem.

Theorem 3.5 ([Straubing, 1985; Pin and Weil, 2002]). *The membership problem is decidable for some level of the dot-depth hierarchy if it is decidable for the same level of the Straubing-Thérien hierarchy.*

This was proven for every full-level in [Straubing, 1985], and then extended in [Pin and Weil, 2002] to half-levels, using a generalization of the algebraic tools (introduced in [Pin, 1995]) that handle classes that are not closed under complement. This reduces the study of the three hierarchies to only one: when considering the membership problem, we may consider only the Straubing-Thérien hierarchy unless we consider results older than Theorem 3.5.

The first decidability results came for the lower levels of the hierarchies before the aforementioned reduction was found. The case of level 1 was solved in [Knast, 1983] for the dot-depth hierarchy, and in [Simon, 1975] for the Straubing-Thérien hierarchy. This was pushed up to level $\frac{3}{2}$ in [Arfi, 1991; Pin and Weil, 1997; Glaßer and Schmitz, 2008]. Nothing really new happened about the next levels until recently. Membership for levels 2 and $\frac{5}{2}$ has been proven decidable in [Place and Zeitoun, 2014a]. These results have then been extended in [Place, 2015] for level $\frac{7}{2}$.

These results rely on the introduction of a new problem called *separation*. Given a class $\mathcal{C}$, the $\mathcal{C}$-separation takes two languages L_1 and L_2 as input and asks whether there exists a third language $L \in \mathcal{C}$ such that $L_1 \subset L$ and $L \cap L_2 = \varnothing$. In other words, L_1 and L_2 can be separated by a language of $\mathcal{C}$, see Figure 3.2.

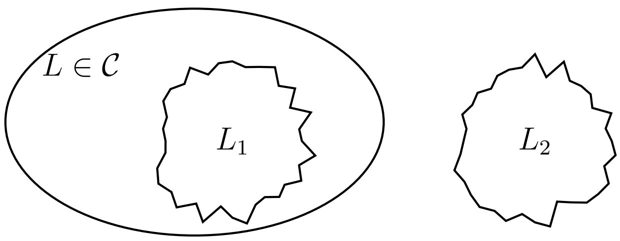


Figure 3.2 – L_1 is $\mathcal{C}$-separated from L_2 by L

Observe that we have a naive reduction from the $\mathcal{C}$-membership problem to the $\mathcal{C}$-separation problem given by the following lemma.

Lemma 3.6. *Let $\mathcal{C}$ be a class of languages and let L be a language. Then $L \in \mathcal{C}$ if and only if L is $\mathcal{C}$-separable from its complement $\overline{L}$.*

Therefore, proving decidability of any separation problem yields an algorithm for deciding the corresponding membership problem. In particular, we can see that $\mathcal{C}$-separation is more general than $\mathcal{C}$-membership. This is not surprising: solving separation requires a deeper understanding of the considered class than for membership. Indeed, for $\mathcal{C}$-membership, we can directly test some properties of $\mathcal{C}$ on the input language. For $\mathcal{C}$-separation, we do not have a language in $\mathcal{C}$ to manipulate. In particular, the input languages may be much more complicated than what can be expressed by $\mathcal{C}$, and still be a positive instance of $\mathcal{C}$-separation. Thus, to solve $\mathcal{C}$-separation, we not only need to know what can be *defined* in $\mathcal{C}$ but we also have to characterize what can be *measured* by $\mathcal{C}$. This observation is far from being cosmetic: for example, the additional information obtained by solving separation for level $n - \frac{1}{2}$ of Straubing-Thérien hierarchy can be used to solve membership at level $n + \frac{1}{2}$, and thus climbing one step in the hierarchy.

Theorem 3.7 ([Place and Zeitoun, 2014a]). *For every integer i, there is reduction from Σ_{i+1}-membership to Σ_i-separation.*

Due to this result, many efforts have been made to solve separation for the lower levels of the hierarchies. Note that Theorem 3.5 does not prove anything regarding separation. However, this theorem was extended in [Place and Zeitoun, 2017a]: decidability of separation for level n does not depend on the considered concatenation hierarchy. This was already proven for full levels with algebraic arguments. Indeed, as shown in [Almeida, 1999], the separation problem is equivalent to an algebraic problem, so-called computability of 2-pointlike sets. For this problem, Theorem 3.5 has been proved in [Steinberg, 2001] for varieties, a kind of classes we define later, and which include the full levels of the concatenation hierarchies.

Level 1 was proven decidable in [Van Rooijen and Zeitoun, 2013; Czerwiński *et al.*, 2013] (but also in [Almeida and Zeitoun, 1997; Almeida *et al.*, 2008] using again the link with algebra established by [Almeida, 1999]). Separation for level $\frac{3}{2}$ was proven decidable in [Place and Zeitoun, 2014a], yielding a membership algorithm for level $\frac{5}{2}$ by Theorem 4.2. In the same paper, the information obtained for solving separation for level $\frac{3}{2}$ was also used to solve membership for level 2, although the result does not rely on a generic reduction between separation for level $n - \frac{1}{2}$ and membership for level n, regardless of n. This result has then been extended to separation for level 2 in [Place and Zeitoun, 2017d]. Moreover, using again Theorem 4.2, decidability of membership for level $\frac{7}{2}$ followed from the decidability of level $\frac{5}{2}$ separation, proven in [Place, 2015]. Note also that the separation problem is also decidable for the class SF, as shown in [Henckell, 1988; Henckell *et al.*, 2010; Place and Zeitoun, 2014b]. The current results about concatenation hierarchies are depicted in Figure 3.3.

We end this section by introducing the two kinds of problems we present in Chapters 3 and 4. The first is to extend these results to the setting of infinite words. The second concerns the complexity of the membership and the separation problem.

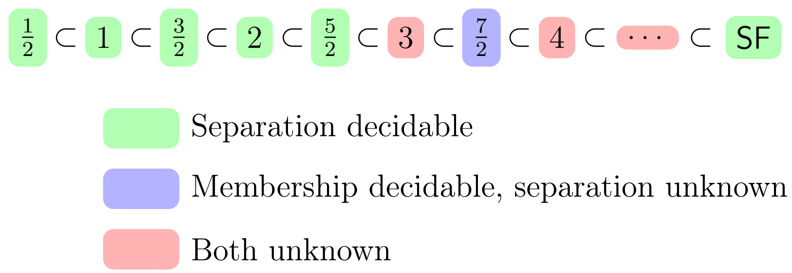


Figure 3.3 – State of the art

3.1.4 The case of infinite words

Solving membership for levels of the Straubing-Thérien and the dot-depth hierarchies is a longstanding open problem. Following Schützenberger's approach for star-free languages, it was first investigated for languages of finite words. However, the question also makes sense for more complex structures, in particular for the most natural extension: infinite words. Schützenberger's result was first generalized to infinite words in [Perrin, 1984], and a suitable algebraic framework for ω-languages (i.e. languages of infinite words) was set up in [Wilke, 1991].

Without any changes, the quantifier alternation hierarchy can be used to define ω-languages instead of only languages of finite words. The definitions of the dot-depth and the Straubing-Thérien hierarchies also extend naturally to this setting, up to some slight adjustments. As shown in [Place and Zeitoun, 2017a], Theorem 3.5 still holds in this setting. Therefore, we may consider only Straubing-Thérien hierarchy, or equivalently the quantifier alternation hierarchy with letter predicates and ordering.

The regular ω-languages are built on top of the regular languages of finite words. Indeed, they are defined using three operations, namely:

- Union: if A, B are regular ω-languages, then $A \cup B$ is a regular ω-language.
- Concatenation: if A is a regular language of finite words and B is a regular ω-language, then AB is a regular ω-language.
- Iteration: if A is a regular language of finite words such that $\varepsilon \notin A$, then A^∞ is a regular ω-language, where $A^\infty = AAA\cdots$.

Therefore, finding a membership or a separation algorithm for ω-languages does not usually require to start over. Instead these algorithms are obtained by building on top of the algorithms for finite words, adding new arguments, specific to infinite words. A flagrant example of this phenomenon is the case of Σ_1-membership. Observe that a Σ_1 sentence can only test whether a (finite or infinite) word w contains some finite words as scattered subwords. This is independent from considering finite or infinite words. Therefore, membership is decidable for Σ_1 on infinite words, using the same criterion as for finite words.

About higher levels of the hierarchies, the decidability of $\mathcal{B}\Sigma_1$-membership from [Simon, 1975] has been generalized from finite to infinite words in [Perrin and Pin, 2004]. This is also the case for Σ_2-membership, whose decidability has been lifted to infinite words in [Bojańczyk, 2008; Diekert and Kufleitner, 2011]. However, few is known for separation on infinite words: except for the easily-solved case of Σ_1, only FO-separation is known to be decidable [Place and Zeitoun, 2014b].

Following the idea of extending algorithms for languages of finite words to this setting, the natural next question is to determine whether the tools developed in [Place and Zeitoun, 2014a] to tackle the lower levels of the hierarchies can be lifted to the case of infinite words. The answer provided in [Pierron *et al.*, 2016] is that we can use the involved algorithms from [Place and Zeitoun, 2014a] as subroutines to prove decidability for lower levels: separation for Σ_2 and Σ_3 and membership for $\mathcal{B}\Sigma_2$ are decidable. This last result was also obtained independently by [Kufleitner and Walter, 2018]. However, no generic decidability transfer result (either from finite to infinite words, or from separation at level $i - \frac{1}{2}$ to membership at level $i + \frac{1}{2}$) is known for now.

3.1.5 Computational complexity of the membership and separation problems

Recall that the hierarchies were introduced following some descriptive complexity motivations: lower levels correspond to languages that are easier to describe. However, if testing membership in a low level class requires an unreasonable amount of computation, then stratifying star-free languages with the hierarchies, and finding the lowest level containing a language may be less useful. This raises the question of the computational complexity of the membership problem for these classes. Moreover, since some membership algorithms rely on separation ones, we also consider complexity of separation.

However, complexity questions can be tricky when considering problems on regular languages, due to the wealth of formalisms representing them. Observe that, a priori, the computational complexity of the membership and the separation problem depends on how the input languages are represented. While converting an automaton into a monoid recognizing the same language is decidable, this may be a costly procedure. In particular, to represent a given language, a monoid can be exponentially less succinct than an automaton. This means that problems on monoids are (a priori) easier than on automata. This is also valid for many possible conversions between the different equivalent descriptions of regular languages. This is not a problem when we only consider decidability, but this can be crucial for complexity, as illustrated for example with membership for level 1. As shown in [Masopust, 2018], this problem is PSpace-complete when the input language is given by a non-deterministic finite automaton (NFA), while it is NLogSpace-complete when the input is a deterministic finite automaton (DFA) [Cho and Huynh, 1991], and in LogSpace when the input is a monoid morphism. However, when considering separability for level 1, the problem is PTime-complete for NFA, DFA and minimal DFA, see [Masopust, 2018].

Similar results are known for other classes, for example first-order definable languages. For NFA and DFA, FO-membership is PSpace-complete [Cho and Huynh, 1991], but for monoids the problem lies in LogSpace. For FO-separation, the exact complexity is still an open question. Using the lower bound for FO-membership for DFA, we can find a lower bound for FO-separation. Indeed, observe that given a DFA recognizing a language L, we can compute in LogSpace a DFA recognizing $\overline{L}$ (by exchanging final and non-final states). Since L is a yes-instance of FO-membership if and only if $(L, \overline{L})$ is a yes-instance of FO-separation, we obtain a LogSpace reduction from FO-membership when the input is a DFA to FO-separation for inputs given by NFAs or DFAs. This yields a PSpace lower bound for FO-separation for NFA and DFA. As we will see, the result of Section 3.3 extends this lower bound to the case of monoids. On the other hand, the best known upper bound is ExpTime, regardless of the input format [Place and Zeitoun, 2014b]. All these results are summarized in Figure 3.4.

	Membership		
	NFA	DFA	Monoid
FO	PSpace-complete	PSpace-complete	LogSpace
Level 1	PSpace-complete	NLogSpace-complete	LogSpace

	Separation		
	NFA	DFA	Monoid
FO	ExpTime PSpace-hard	ExpTime PSpace-hard	ExpTime **PSpace-hard**
Level 1	PTime-complete	PTime-complete	PTime **PTime-hard**
Level $n + 1/2$	**PSpace-hard**	**PSpace-hard**	**PSpace-hard**

Figure 3.4 – Complexity of membership and separation problems (bounds in boldface are consequences of Theorem 3.37 or 4.10, presented later)

3.1.6 Organization of Chapters 3 and 4

This chapter is devoted to a generic complexity result. We prove that when $\mathcal{C}$ is nice enough, $\mathcal{C}$-separation has the same complexity regardless of the input format (NFA or monoid). This is a key difference between separation and membership, for which this property does not hold. We first need to settle down in Section 3.2 the definitions and main properties of these two formalisms. The result itself (namely, Theorem 3.37) is proved in Section 3.3.

Chapter 4 is devoted to the study of the polynomial closure, one of the two fundamental operations for defining the Straubing-Thérien and the dot-depth hierarchies. First, we consider the $\mathrm{Pol}(\mathcal{C})$-separation problem from a complexity point of view, and prove a generic PSpace lower bound when $\mathcal{C}$ is expressive enough (Theorem 4.10). The second result is an extension of the decidability of $\mathrm{Pol}(\mathcal{C})$-separation to the setting of infinite words, when $\mathcal{C}$ is a finite class (Theorem 4.67).

3.2 Preliminaries

This section is devoted to the introduction of some notation, mainly related to the two formalisms describing regular languages that we consider in the rest of this chapter. The first one is algebraic (monoids), while the second one is computational (finite automata). To fix the notation, we begin with basic definitions about words.

Definition 3.8. A *word* over the alphabet A is a finite (possibly empty) sequence of letters of A. We denote by A^* the set of words over A. A *language* on A is a subset of A^*.

The *alphabet* of a word w is the smallest set A such that $w \in A^*$. It is denoted by $\mathrm{alph}(w)$.

If $u, v \in A^*$, we denote by $u \cdot v$ or uv their concatenation.

3.2.1 Monoids and semigroups

Definition 3.9. A *semigroup* is a set S equipped with an associative operation $s \cdot t$ (often written st). A semigroup S with a neutral element 1_S satisfying $1_S s = s 1_S = s$ for every $s \in S$ is called a *monoid*.

Example 3.10. The set A^* is a monoid when endowed with concatenation. Its neutral element is the empty word ε. The set $A^+ = A^* \setminus \{\varepsilon\}$ is a semigroup.

Definition 3.11. A *semigroup morphism* is a product-preserving mapping between two semigroups: $\alpha : S \to T$ is a semigroup morphism if $\alpha(st) = \alpha(s)\alpha(t)$ for all $s, t \in S$. It is a *monoid morphism* if it also satisfies $\alpha(1_S) = 1_T$.

Given a language L and a monoid morphism $A^* \to M$, we say that L is *recognized* by α if there exists $F \subset M$ such that $L = \alpha^{-1}(F)$. It is well-known that a language is regular if and only if it is recognized by a *finite* monoid.

Example 3.12. The set $M = \mathbb{Z}/2\mathbb{Z}$ equipped with addition modulo 2 is a monoid. Moreover, the morphism $\varphi : \{a\}^* \to M$ defined by $\varphi(a) = 1$ recognizes the languages $\varnothing = \varphi^{-1}(\varnothing)$, $a^* = \varphi^{-1}(M)$, $(aa)^* = \varphi^{-1}(\{0\})$ and $a(aa)^* = \varphi^{-1}(\{1\})$.

Every regular language L is recognized by infinitely many morphisms to finite monoids. However, among all of them, there is a "minimal" one. Such a morphism is called the *syntactic morphism* of L. It is constructed as follows.

Definition 3.13. Given a language L, the *syntactic order* of L is the relation defined on A^* by $u \leqslant_L v$ if for all words $x, y \in A^*$, we have $xuy \in L \Rightarrow xvy \in L$.
The *syntactic congruence* of L is defined by $u \sim_L v$ if $u \leqslant_L v$ and $v \leqslant_L u$.

It is easy to check that $\sim_L$ is indeed a congruence. Moreover, when L is regular, this relation has finite order and the *syntactic morphism* of L is the canonical projection $\alpha_L : A^* \to A^*/\sim_L$ associating to each word of A^* its class for $\sim_L$. When L is regular, the relation $\sim_L$ and the morphism α_L can be computed from L, see [Pin, 1997]. We give a construction later, when stating Proposition 3.25.

Example 3.14. We consider two examples:

- Let $L = (aa)^*$. Then it is easy to see that $a^k \sim_L a^\ell$ if and only if k and ℓ have the same parity. There are thus two classes of $\sim_L$: the set of words with even number of a's, namely $(aa)^*$, and the set of words with odd number of a's, namely $a(aa)^*$. Since $(aa)^*(aa)^* = (aa)^*$ and $a(aa)^*a(aa)^* \subset (aa)^*$, the syntactic monoid of $(aa)^*$ is thus isomorphic to $(\mathbb{Z}/2\mathbb{Z}, +)$.

- Let $L = (ab)^*$. Then $\sim_L$ has several classes: $\varepsilon, (ab)^+, (ba)^+, a(ba)^*, b(ab)^*$, and the class of all the other words (containing two consecutive identical letters). We represent these classes respectively by the words $\varepsilon, ab, ba, a, b, aa$.

  The syntactic monoid M of L contains thus six elements, and multiplication is given by the following table:

$\times$	ε	ab	ba	a	b	aa
ε	ε	ab	ba	a	b	aa
ab	ab	ab	aa	a	aa	aa
ba	ba	aa	ba	aa	b	aa
a	a	aa	a	aa	ab	aa
b	b	b	aa	ba	aa	aa
aa	aa	aa	aa	aa	aa	aa

  Indeed, consider the example $ab \cdot a$. This is the class of words obtained as uv where $u \in (ab)^+$ and $v \in a(ba)^*$. We thus have $uv \in a(ba)^*$, i.e. $ab \cdot a = a$. We can find all the other products in the same way.

We may check that the resulting monoid is the set $\{\varepsilon, a, b, ab, ba, aa\}$ endowed with concatenation, using the following relations: $aba = a$, $bab = b$, and $aa = bb$.

Recall that Schützenberger's theorem states that a language is star-free if and only if its syntactic monoid is aperiodic. Aperiodicity is a property of a monoid relying on the notion of idempotent. An element s of a semigroup S is *idempotent* when it satisfies $s^2 = s$. As we will see, idempotency is a key property when studying finite semigroups. This is illustrated by the following result.

Proposition 3.15. *Given a* finite *semigroup S, there is a positive integer ω such that for all s of S, s^ω is idempotent.*

Given a finite semigroup S, the smallest such integer is usually denoted by $\omega(S)$, or by ω when S is understood.

Proof. Let $s \in S$. Since S is finite, the sequence $(s^{2^n})_{n\in\mathbb{N}}$ takes twice the same value. There thus exist two integers i, j such that $s^{2^i} = s^{2^{i+j}}$. Let $\omega(s) = 2^{i+j} - 2^i$. We have

$$(s^{\omega(s)})^2 = s^{2^{i+j+1}-2^{i+1}} = s^{2^{i+j}} s^{2^{i+j}-2^{i+1}} = s^{2^i} s^{2^{i+j}-2^{i+1}} = s^{2^{i+j}-2^{i+1}+2^i} = s^{\omega(s)}$$

We then define $\omega = \prod_{s\in S} \omega(s)$. For every $s \in S$, we have

$$(s^\omega)^2 = s^{2\omega} = (s^{2\omega(s)})^k = (s^{\omega(s)})^k = s^{k\omega(s)} = s^\omega$$

where $k = \frac{\omega}{\omega(s)} \in \mathbb{N}$. $\square$

Example 3.16.

- We have $\omega(\mathbb{Z}/2\mathbb{Z}) = 2$ since 1 is not idempotent, but $2 \times 0 = 2 \times 1 = 0$ is.
- The second monoid of Example 3.14 also satisfies $\omega = 2$.

Observe that given a semigroup S and an element $s \in S$, the set $\{s^{\omega+n}, n \geqslant 0\}$ is actually isomorphic to a subgroup of $(\mathbb{Z}/\omega\mathbb{Z}, +)$, via the application $n \mapsto s^{\omega+n}$. As shown in [Pin, 1997], this is the unique group in the sub-semigroup of S generated by s.

The notion of aperiodicity asks for all these groups to be trivial. Equivalently, it requires the condition $s^{\omega+1} = s^\omega$ to be satisfied for every $s \in S$.

Example 3.17.

- The monoid $\mathbb{Z}/2\mathbb{Z}$ is not aperiodic, since it is a group. Alternatively, we have $\omega(M) = 2$ and $\omega(M) \times 1 = 0 \neq (\omega(M) + 1) \times 1$.

  By Schützenberger's theorem, we thus obtain that $(aa)^*$ is not definable in FO, since its syntactic monoid is $\mathbb{Z}/2\mathbb{Z}$. As a consequence, it is not FO-separable from its complement $a(aa)^*$.

  This example can actually be generalized: every pair of languages of the form $(a^k)^*a^\ell$ (for some positive integers k, ℓ) is FO-inseparable.

- For the second monoid of Example 3.14, we again have $\omega(M) = 2$, but this time M is aperiodic. Therefore, using again Schützenberger's theorem, the language $(ab)^*$ is definable in FO since it is recognized by the morphism $\alpha : \{a, b\}^* \to M$ defined by $\alpha(a) = a$ and $\alpha(b) = b$. We can even construct a FO-sentence describing it:

$$\begin{aligned}&\forall x \forall y, (a(x) \wedge y = x + 1) \Rightarrow b(y)\\&\forall x \forall y, (b(x) \wedge y = x + 1) \Rightarrow a(y)\\&\forall x, (\min(x) \Rightarrow a(x))\\&\forall x, (\max(x) \Rightarrow b(x))\end{aligned}$$

  This sentence describes words such that: each letter following an a is a b, each letter following a b is an a, the first letter is an a and the last letter is a b.

3.2.2 Automata

Definition 3.18. An *automaton* is a 5-tuple $(Q, \Sigma, I, F, \delta)$ where:

- Q is a finite set of states,
- Σ is a finite alphabet,
- $I \subset Q$ is a set of initial states,
- $F \subset Q$ is a set of final states, and
- $\delta \subset Q \times \Sigma \times Q$ is a set of transitions.

We denote a transition (q, a, q') by $q \xrightarrow{a} q'$. A word $a_1 \cdots a_n$ (with $a_1, \ldots, a_n \in A$) is accepted by an automaton $\mathcal{A} = (Q, \Sigma, I, F, \delta)$ if there exists a sequence of states $q_0, \ldots, q_n$ such that $q_0 \in I, q_n \in F$ and $q_{i-1} \xrightarrow{a_i} q_i \in \delta$ for $1 \leqslant i \leqslant n$. The set of such words is the language recognized by $\mathcal{A}$.

Example 3.19. The languages $(aa)^*$ and $(ab)^*$ are recognized by the automata represented in Figure 3.5. Each node corresponds to a state, and each arc to a transition. The initial (resp. final) states are represented by an ingoing (resp. outgoing) arc without source (resp. destination).

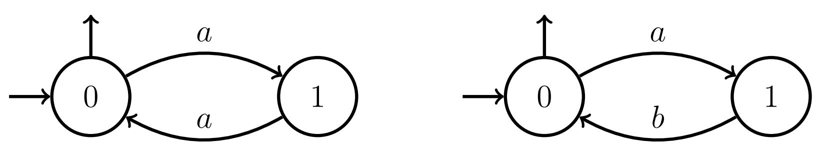


Figure 3.5 – Automata recognizing $(aa)^*$ (left) and $(ab)^*$ (right).

Definition 3.20. An automaton $(Q, \Sigma, I, F, \delta)$ is deterministic (a DFA) when $|I| = 1$ and its set of transitions can be represented as a set of partial functions $\delta_a : Q \to Q$ for $a \in \Sigma$, i.e. if for every $q \in Q$ and $a \in \Sigma$, there is at most one $q' \in Q$ such that $(q, a, q') \in \delta$.

Given an automaton $\mathcal{A} = (Q, \Sigma, I, F, \delta)$, we can compute a deterministic automaton recognizing the same language as $\mathcal{A}$. The usual way of doing so is by considering an automaton whose states are subsets of Q. There is a transition $S \xrightarrow{a} S'$ when $S' = \{q' \in Q \mid \exists q \in S, q \xrightarrow{a} q'\}$. In other words, this means that S' is the set of states obtained by reading a from the states of S. The new initial state is $\{I\}$, and a state S is final when S intersects F. This automaton is deterministic and recognizes the same language as $\mathcal{A}$. However, note that its size is exponential with respect to $\mathcal{A}$.

Given a regular language L, we can prove that there exists a deterministic automaton recognizing L with minimum number of states. This automaton is called the *minimal automaton* of L, and can be effectively constructed from any automaton recognizing L, see [Brzozowski, 1962].

The minimal automaton of a language L is a canonical object: it is unique up to renaming states. Due to its minimality, it also satisfies some more properties. The first one is given using Nerode's congruence.

Definition 3.21. Let $\mathcal{A} = (Q, \Sigma, I, F, \delta)$ be an automaton recognizing a language L. The *Nerode congruence* is the relation defined on Q by $q \sim_{\mathcal{A}} q'$ when the languages recognized by $(Q, \Sigma, \{q\}, F, \delta)$ and $(Q, \Sigma, \{q'\}, F, \delta)$ are the same.

If $\mathcal{A}$ is a minimal automaton, then its Nerode's congruence is the equality. We can exploit this property to design a minimization algorithm: starting from a DFA $\mathcal{A}$, we define a (coarse) partition of its set of states, and refine it successively until we end up with the partition where each class is an equivalence class for the Nerode congruence. To refine a partition, we split a set S of states each time we find $q, q' \in S$ and $a \in A$ such that reading a from q and q' ends in two states not in the same set of the partition. This leads to a polynomial time algorithm. However, we need the initial automaton to be deterministic. Therefore, minimizing an NFA may yield a minimal automaton of exponential size.

Note that the languages used in the definition of Nerode's congruence can be written as $\{v \in A^* \mid uv \in L\}$ where u is a suitable word. For example, assume that $\mathcal{A} = (Q, \Sigma, \{q_i\}, F, \delta)$ is a DFA recognizing L. Let u be a word such that reading u from q_i in $\mathcal{A}$ ends up in the state q. Then the language recognized by $(Q, \Sigma, \{q\}, F, \delta)$ is $\{v \in A^* \mid uv \in L\}$. This language is a special case of the so-called quotients.

Definition 3.22. Given a word $u \in A^*$ and a language L, the left quotient $u^{-1}L$ is defined as $\{v \in A^* \mid uv \in L\}$, and the right quotient Lu^{-1} as $\{v \in A^* \mid vu \in L\}$.

Example 3.23. Let $L = (aa)^*$ and let k be an integer. Then $(a^k)^{-1}L$ is the set of words a^n such that $a^k a^n$ has even length, i.e. with n of same parity as k. Therefore, $(a^k)^{-1}L$ is $a(aa)^*$ when k is odd and $(aa)^*$ when k is even. This also holds for right quotients.

Similarly, the quotients of $L = (ab)^*$ are

- $b(ab)^*$, obtained as $w^{-1}L$ where $w \in a(ba)^*$.
- L, obtained as $w^{-1}L$ or Lw^{-1} where $w \in L$.
- $a(ba)^*$, obtained as Lw^{-1} where $w \in b(ab)^*$.
- $\varnothing$, obtained by taking left or right quotient by any other word.

Observe that, given a deterministic automaton $\mathcal{A} = (Q, \Sigma, I, F, \delta)$ recognizing a language L, the left quotient $u^{-1}L$ is recognized by $(Q, \Sigma, \{q\}, F, \delta)$ where q is the state obtained by reading u from the initial state. Note that, if such a state exists, it is unique since the automaton is deterministic. Otherwise, we have $u^{-1}L = \varnothing$. Similarly, automata recognizing right quotients are obtained by considering other final states. We can thus reformulate the definition of Nerode congruence when $\mathcal{A}$ is deterministic. Assume that reading u (resp. v) from the initial state of $\mathcal{A}$ ends in the state q (resp. q'). Then we have $q \sim_{\mathcal{A}} q'$ if and only if $u^{-1}L = v^{-1}L$.

In particular, there is only a finite number of such quotients. Moreover, if the automaton $\mathcal{A}$ is minimal, then the relation $\sim_{\mathcal{A}}$ is the equality. Therefore, the left quotients of a language are in bijection with the states of its minimal automaton. Note that this also holds for right quotients, by exchanging the role of initial and final states in the previous analysis. This is the well-known result of Myhill-Nerode.

Theorem 3.24 (Myhill-Nerode)**.** *A language is regular if and only if it has finitely many left (resp. right) quotients.*

Automata and monoids are two formalisms that recognize regular languages. Moreover, the conversions are effective in both ways: given an automaton, we can compute a monoid morphism recognizing the same language, and conversely. We end this subsection by presenting the conversions between these formalisms.

First consider a regular language L recognized by a monoid morphism $\alpha : A^* \to M$. Then L is recognized by the automaton whose states are elements of M, alphabet is A, initial state is $\alpha(\varepsilon) = 1_M$, final states are elements of $\alpha(L)$, and transitions are all the $s \xrightarrow{a} s\alpha(a)$ for $s \in M$ and $a \in A$. Indeed, we can prove by induction on the length of any word w that reading w from 1_M ends in the state $\alpha(w)$. Therefore, w is accepted by this automaton if and only if $\alpha(w) \in \alpha(L)$, i.e. $w \in L$ since α recognizes L. Observe that this conversion can be done in $\mathsf{LogSpace}$.

We thus obtain the following proposition.

Proposition 3.25. *Given an monoid morphism $\alpha : A^* \to M$ recognizing a language L, one can construct in* LogSpace *an automaton $\mathcal{A}$ recognizing L.*

Example 3.26. The automata depicted on Figure 3.6 are the ones obtained by the aforementioned construction, starting from the monoids of Example 3.14.

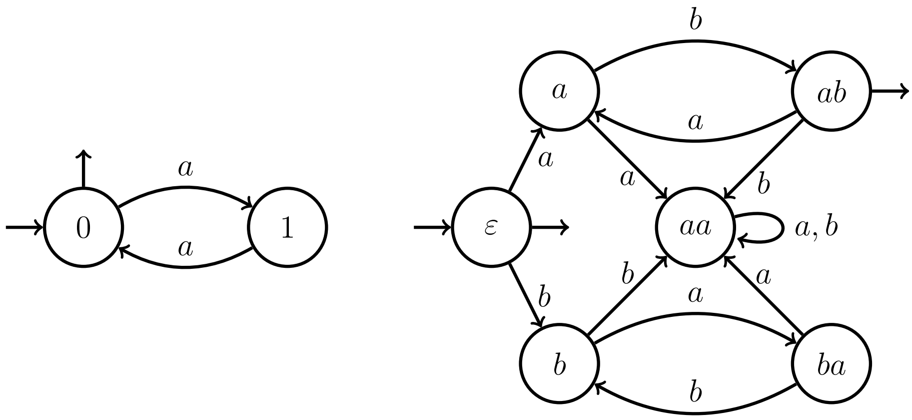


Figure 3.6 – Automata obtained from the syntactic monoids of $(aa)^*$ (left) and $(ab)^*$ (right).

Note in particular that the automaton for $(aa)^*$ is the minimal automaton of this language. However, this is not the case for $(ab)^*$, even if we remove the states labeled aa, b and ba from which no word can be accepted.

One way of computing the other conversion relies on the following notion.

Definition 3.27. Given a deterministic automaton $\mathcal{A} = (Q, \Sigma, I, F, \delta)$, its *transition monoid* $M_{\mathcal{A}}$ is the monoid generated by all the partial functions $\delta_a : Q \to Q$ for $a \in \Sigma$.

Let $\mathcal{A}$ be a deterministic automaton recognizing a language L. Define a monoid morphism $\alpha : A^* \to M_{\mathcal{A}}$ by $\alpha(a) = \delta_a$ for $a \in A$. Then for every word w, we have $w \in L$ if and only if $\alpha(w)$ maps the initial state of $\mathcal{A}$ to a final state. Therefore, L is recognized by α.

Example 3.28. The automata presented in Example 3.19 are deterministic and have the following transition monoids.

- For $(aa)^*$, the function δ_a swaps states 0 and 1. The function $\delta_a \circ \delta_a$ coincides with identity, hence the monoid generated by δ_a is again isomorphic to $\mathbb{Z}/2\mathbb{Z}$.
- For $(ab)^*$, the function δ_a is defined only on state 0, and maps it to state 1, while δ_b is defined only on state 1 and maps it on state 0. The

monoid they generate is exactly the one of Example 3.14. For example, the element ba corresponds to the function $\delta_a \circ \delta_b$, which is defined on state 1 and maps it to itself.

The automata presented in Example 3.19 are actually the minimal automata of $(aa)^*$ and $(ab)^*$, and it appears that their transition monoid are isomorphic to the syntactic monoid of the recognized languages. These examples are not isolated cases, as shown by the following proposition.

Proposition 3.29. *The syntactic monoid of a regular language is isomorphic to the transition monoid of its minimal automaton.*

The construction of the transition monoid of a DFA shows how to translate an automaton into a monoid recognizing the same language: we first determinize it, then we compute its transition monoid. Moreover, with this proposition, we can even compute the syntactic monoid (instead of just some monoid recognizing the right language). To this end, we have to minimize the obtained DFA before computing the transition monoid. This ensures that the syntactic monoid is computable, even if it was unclear from its definition.

However, in contrast to the other conversion, the construction is not LogSpace anymore: the transition monoid $M_{\mathcal{A}}$ of a DFA $\mathcal{A}$ may have exponential size with respect to $\mathcal{A}$. A fortiori, this is also the case for the syntactic monoid of the language recognized by $\mathcal{A}$.

3.2.3 Varieties

In this thesis, we look for decidability and complexity results for the $\mathcal{C}$-separation problem. However, if $\mathcal{C}$ is an arbitrary class, there is no hope of obtaining such results, unless we get some structural hypotheses on $\mathcal{C}$. Moreover, the classes we consider are not chosen randomly: they are defined using some syntactic restrictions. Due to this choice, these classes satisfy some additional properties, like closure by Boolean operations. The most convenient classes to work with are the so-called varieties, defined as follows.

Definition 3.30. Let $\mathcal{C}$ be a class of languages. Given an alphabet A, we denote by $\mathcal{C}_A$ the class of languages in $\mathcal{C}$ over the alphabet A. We say that $\mathcal{C}$ is a *variety* if:

- For every alphabet A, $\mathcal{C}_A$ is closed under Boolean operations (union, intersection and complement).
- For every alphabet A, each word $u \in A^*$ and language $L \in \mathcal{C}_A$, $u^{-1}L$ and Lu^{-1} lie in $\mathcal{C}_A$.
- For every alphabets A, B, if $\varphi : A^* \to B^*$ is a monoid morphism and $L \in \mathcal{C}_B$, then $\varphi^{-1}(L) \in \mathcal{C}_A$.

The three items correspond respectively to closure under Boolean operations, quotients and inverse morphisms. Note that if $u = vw$, we have $u^{-1}L = w^{-1}(v^{-1}L)$ and $Lu^{-1} = (Lw^{-1})v^{-1}$. In particular, this means that we may only require closure under quotient by a letter in the definition of variety.

We will often consider *finite* varieties. A variety $\mathcal{C}$ is finite if for every alphabet A, the class $\mathcal{C}_A$ is finite.

Example 3.31. We present here some classic examples.

- A first example of variety is $\mathsf{Reg} = (\mathsf{Reg}_A)_A$ where for every alphabet A, Reg_A is the class of regular languages over the alphabet A.

- The class SF is also a variety, as well as all the full levels of the Straubing-Thérien and the dot-depth hierarchies.

- The class AT of *alphabet testable languages* is defined as the set of Boolean combinations of all languages A^*aA^* for every alphabet A and every letter $a \in A$.

  It is the class of languages L such that testing if $w \in L$ depends only on the alphabet of w. From a logical point of view, it can be seen as the class of languages defined in $\mathsf{FO}(\varnothing)$, the fragment of first-order logic using only letter predicates (but not ordering).

  This class is a variety. Together with level 0 of the Straubing-Thérien and the dot-depth hierarchies, it is an emblematic example of finite variety.

- Level 1 of Straubing-Thérien hierarchy is also known as the class of *piecewise testable* languages. Due to Theorem 3.4, it corresponds to the fragment $\mathcal{B}\Sigma_1$. It is thus the set of all Boolean combinations of languages $A^*a_1A^*a_2 \cdots A^*a_kA^*$ where k is an integer, A is an alphabet and $a_1, \dots, a_k$ are letters of A. In particular, testing membership of a word w in a piecewise testable language depends only on the set of (scattered) subwords of w.

Given a monoid morphism α, the set of all languages recognized by α has a strong algebraic structure: it is for example closed under Boolean operations and quotients. We can get even more structure on this set when α is a syntactic morphism, by the following proposition.

Proposition 3.32 ([Pin, 1986])**.** *Let $\mathcal{C}$ be a variety, and $\alpha : A^* \to M$ be the syntactic morphism of a language in $\mathcal{C}$. Then all the languages recognized by α lie in $\mathcal{C}$.*

Proof. Let $\alpha : A^* \to M$ be the syntactic morphism of a language $L \in \mathcal{C}$. Let K be a language recognized by α, i.e. $K = \alpha^{-1}(F)$ for $F \subset M$.

Observe that we have

$$K = \bigcup_{s \in F} \alpha^{-1}(s).$$

Since $\mathcal{C}$ is a variety, it is closed under unions, hence it is sufficient to treat the case $F = \{s\}$ for some $s \in M$.

Let $w \in \alpha^{-1}(s)$. Note that such a word exists since otherwise $L = \varnothing$ and we are done: we have $L \in \mathcal{C}$ since $\mathcal{C}$ is closed under Boolean operations. The goal is to write $\alpha^{-1}(s)$ as a Boolean combination of quotients of L. This will conclude that $\alpha^{-1}(s) \in \mathcal{C}$ since $\mathcal{C}$ is a variety.

To this end, consider the set $C(w) = \{(u, v) \in A^* \times A^* \mid uwv \in L\}$. By definition of $\sim_L$, we have, for every word $w' \in A^*$:

$$s = \alpha(w') \Leftrightarrow w \sim_L w' \Leftrightarrow C(w) = C(w').$$

Therefore, $\alpha^{-1}(s) = \{w' \in A^* \mid C(w') = C(w)\}$.

Observe that a pair (u, v) lies in $C(w)$ if and only if $w \in u^{-1}(Lv^{-1})$. We can thus rewrite $\alpha^{-1}(s)$ as

$$\bigcap_{(u,v) \in C(w)} u^{-1}(Lv^{-1}) \setminus \bigcup_{(u,v) \notin C(w)} u^{-1}(Lv^{-1}).$$

This is the expression we look for. However, we cannot conclude directly since the intersection and the union in the above expression are indexed by infinite sets of words.

However, since L is regular, it has a finite number of quotients. Therefore, the union and the intersection above are actually indexed by a finite number of pairs (u, v). This ensures that $\alpha^{-1}(s) \in \mathcal{C}$. $\square$

Observe that the criterion given by Schützenberger's theorem to decide whether a language L is definable in FO depends on the syntactic monoid of L, but not on its accepting set. This is explained by Proposition 3.32: there is nothing special about the accepting set of L since L is definable in FO if and only if all languages recognized by its syntactic morphism are definable in FO. This proposition thus emphasizes that, in order to solve $\mathcal{C}$-membership, we have to focus on properties of syntactic monoids instead of properties of languages.

Due to all their closure properties, varieties are very convenient classes to work with. Note however that many interesting classes, such as the ones obtained using polynomial closure, are often not varieties (for example, consider the half-levels of the Straubing-Thérien and the dot-depth hierarchies). Indeed, they are not closed under complement in general. This observation leads to the introduction of the notion of *positive varieties*.

Definition 3.33. A class $\mathcal{C}$ of languages is a *positive variety* if:

- For every alphabet A, the class $\mathcal{C}_A$ is closed under union and intersection.
- For every alphabet A, each word $u \in A^*$ and language $L \in \mathcal{C}_A$, the quotients $u^{-1}L$ and Lu^{-1} lie in $\mathcal{C}_A$.
- For all alphabets A, B, if $\varphi : A^* \to B^*$ is a monoid morphism and $L \in \mathcal{C}_B$, then $\varphi^{-1}(L) \in \mathcal{C}_A$.

Observe that taking the Boolean closure of a positive variety allows to restore closure under complement, while preserving the other closure properties. In other words, we have the following.

Lemma 3.34. *The Boolean closure of a positive variety is a variety.*

Proof. Let $\mathcal{C}$ be a positive variety and A be an alphabet. Observe that $\mathrm{Bool}(\mathcal{C})_A = \mathrm{Bool}(\mathcal{C}_A)$. It is thus clear that this class is closed under Boolean operations. Moreover, every language $L \in \mathrm{Bool}(\mathcal{C})_A$ is constructed as unions, intersections and complements of languages in $\mathcal{C}_A$.

To prove the result, it is thus sufficient to prove that the Boolean operations commute with quotients and inverse morphisms. Indeed, this will prove that every quotient (resp. inverse image) of a language in $\mathrm{Bool}(\mathcal{C})$ can be written as Boolean combinations of quotients (resp. inverse image) of languages in $\mathcal{C}$, ensuring that $\mathrm{Bool}(\mathcal{C})$ is a variety.

Let K, L be two languages and u, v be two words. Unfolding the definitions, we get:

$$v \in u^{-1}(K \cup L) \Leftrightarrow uv \in K \cup L \Leftrightarrow uv \in K \text{ or } uv \in L \Leftrightarrow v \in u^{-1}K \cup u^{-1}L$$

Similarly, we have $u^{-1}(K \cap L) = u^{-1}K \cap u^{-1}L$.

Moreover, we have

$$v \in u^{-1}\overline{L} \Leftrightarrow uv \notin L \Leftrightarrow v \notin u^{-1}L \Leftrightarrow v \in \overline{u^{-1}L},$$

hence $u^{-1}\overline{L} = \overline{u^{-1}L}$.

Similarly, if A, B are alphabets, $\varphi : A^* \to B^*$ is a morphism and K, L are languages over B, we have

$$\begin{aligned}
&\varphi^{-1}(A \cup B) = \varphi^{-1}(A) \cup \varphi^{-1}(B),\\
&\varphi^{-1}(A \cap B) = \varphi^{-1}(A) \cap \varphi^{-1}(B), \text{ and}\\
&\varphi^{-1}(\overline{A}) = \overline{\varphi^{-1}(A)}.
\end{aligned}$$

This concludes the proof. $\square$

To handle classes that are only positive varieties (but not necessarily varieties), we need to enrich the notion of recognition by monoids: we consider *ordered monoids*, i.e. monoids equipped with an order compatible with the multiplication. A language L is then recognized by a morphism $\alpha : A^* \to (M, \leqslant_M)$

when $L = \alpha^{-1}(\alpha(L))$ and $\alpha(L)$ is upward-closed for $\leqslant_M$. This allows to handle classes that are not closed under complement: even if $\overline{L} = \alpha^{-1}(\overline{\alpha(L)})$, the set $\overline{\alpha(L)}$ may not be upward-closed, hence α may not recognize $\overline{L}$. This will be illustrated in Example 3.35.

Note that, given a regular language L, the syntactic monoid M_L of L is defined as $A^*/\sim_L$. Therefore, the quasi-order $\leqslant_L$ is well defined on M_L. It is even an order by definition of $\sim_L$. Therefore, syntactic monoids are naturally endowed with a structure of ordered monoids.

Example 3.35. We again consider the languages $(aa)^*$ and $(ab)^*$. For each of them, we denote by $\alpha_L : A^* \to M_L$ their syntactic morphism.

- Let $L = (aa)^*$. Its syntactic monoid is $M_L = \mathbb{Z}/2\mathbb{Z}$. Observe that 0 and 1 are incomparable for $\leqslant_L$. Indeed, assuming by symmetry that $0 \leqslant_L 1$, we obtain that $0+1 \leqslant_L 1+1$ since $\leqslant_L$ is compatible with addition. We thus obtain that $0 = 1$, a contradiction.

  Therefore, the relation $\leqslant_L$ is trivial on M_L. In particular, every subset of M_L is upward closed, hence the languages recognized by $(M_L, \leqslant_L)$ as ordered monoid are exactly the ones recognized by M_L as a monoid: $\varnothing, (aa)^*, a(aa)^*$ and a^*.

- Let $L = (ab)^*$. In this case, recall that $M_L = \{\varepsilon, a, b, ab, ba, aa\}$. Observe that for all words $u, v \in A^*$ and $w \in \alpha_L^{-1}(aa)$, we have $uwv \notin (ab)^*$ since w contains two consecutive a's or two consecutive b's. Thus, by definition of $\leqslant_L$, we have $aa \leqslant_L s$ for every $s \in M_L$.

  We can also check that $ab \leqslant_L \varepsilon$ and $ba \leqslant_L \varepsilon$. Except for that, the other pairs are incomparable for $\leqslant_L$.

  Therefore, the language $(ab)^+ = \alpha_L^{-1}(ab)$ is not recognized by the ordered monoid $(M_L, \leqslant_L)$ since $\{ab\}$ is not upward-closed. However, the language $(ab)^* = \alpha_L^{-1}(\{ab, \varepsilon\})$ is still recognized.

  Observe also that the language $\{\varepsilon\} = \alpha_L^{-1}(\{\varepsilon\})$ is recognized, but not its complement $\{a, b\}^+ = \alpha_L^{-1}(\{a, b, ab, ba, aa\})$.

We end this section by extending Proposition 3.32 to this setting, using the more generic framework of [Pin, 1995]. The "missing" hypothesis of closure by complement is balanced by considering recognition by **ordered** monoids.

Proposition 3.36 ([Pin, 1995])**.** *Let $\mathcal{C}$ be a positive variety, and $\alpha : A^* \to (M, \leqslant_M)$ be a the syntactic morphism of a language in $\mathcal{C}$. Then all the languages recognized by α lie in $\mathcal{C}$.*

3.3 Input format vs complexity

In this section, we establish Theorem 3.37, stating that the complexity of the separation problem does not depend on the format of the input languages. We consider here two such formats: (non-deterministic) finite automata, and finite monoids. Given a class $\mathcal{C}$ of languages, we thus introduce the two following variants of the $\mathcal{C}$-separation problem:

- $\mathcal{C}$-separation for automata: given two automata $\mathcal{A}_1$ and $\mathcal{A}_2$, can we separate the languages recognized by $\mathcal{A}_1$ and $\mathcal{A}_2$ by a language in $\mathcal{C}$?
- $\mathcal{C}$-separation for monoids: given two monoid morphisms $\alpha_1 : A^* \to M_1$ and $\alpha_2 : A^* \to M_2$ and two subsets $F_1 \subset M_1$ and $F_2 \subset M_2$, can we separate the languages $\alpha_1^{-1}(F_1)$ and $\alpha_2^{-1}(F_2)$ by a language in $\mathcal{C}$?

First note that there is an easy reduction from each problem to the other one: just convert the automata into monoids recognizing the same languages (or the converse) and apply the suitable algorithm. Recall that constructing an automaton recognizing the same language as a given monoid morphism can be done in $\mathsf{LogSpace}$, using a Proposition 3.25. We thus obtain a $\mathsf{LogSpace}$ reduction from $\mathcal{C}$-separation for languages given by monoids to $\mathcal{C}$-separation for languages given by automata, regardless of the class $\mathcal{C}$ we consider. Intuitively, this means that separation is more costly on automata than on monoids.

On the other hand, converting an automaton into a monoid recognizing the same language may need exponential time, hence the previous naive approach does not lead to a $\mathsf{LogSpace}$ reduction. However, observe that these naive reductions are generic: they are independent from the properties of the languages we want to test. The goal of this section is to use additional properties of separation to establish a $\mathsf{LogSpace}$ reduction in the other direction. We prove that, under suitable assumptions on the class $\mathcal{C}$, the $\mathcal{C}$-separation problem has the same complexity regardless of the kind of input we consider. More precisely, we prove the following result.

Theorem 3.37. *If $\mathcal{C}$ is a positive varietysuch that $\mathrm{Bool}(\mathcal{C}) \neq \mathsf{Reg}$, then there is a $\mathsf{LogSpace}$ reduction from the $\mathcal{C}$-separation problem for automata to the $\mathcal{C}$-separation problem for monoids.*

Together with the previous analysis, this theorem yields the following corollary.

Corollary 3.38. *If $\mathcal{C}$ is a positive variety such that $\mathrm{Bool}(\mathcal{C}) \neq \mathsf{Reg}$, then the $\mathcal{C}$-separation problem has the same complexity regardless on whether its input languages are given by automata or monoids morphisms.*

We can even strengthen this corollary by getting rid of the hypothesis $\mathrm{Bool}(\mathcal{C}) \neq \mathsf{Reg}$, and obtain the following corollary.

Corollary 3.39. *If $\mathcal{C}$ is a positive variety, then the $\mathcal{C}$-separation problem has the same complexity regardless on whether its input languages are given by automata or monoids morphisms.*

Proof. Let $\mathcal{C}$ be a positive variety, and assume that $\mathcal{C} \neq \mathsf{Reg}$. As a consequence of [Almeida and Klíma, 2015, Theorem 9.3], we have $\mathrm{Bool}(\mathcal{C}) \neq \mathsf{Reg}$. Therefore, when $\mathcal{C} \neq \mathsf{Reg}$, Corollary 3.39 is directly obtained from Corollary 3.38.

It remains to consider the case $\mathcal{C} = \mathsf{Reg}$. In this case, the separation problem is equivalent to testing emptiness of intersection. For languages given by automata, this is a well-known NLogSpace-complete problem, see [Jones, 1975]. In the setting of monoids, this is also NLogSpace-complete, as we will see.

Indeed, a NLogSpace algorithm is given by the following: starting from two monoid morphisms, compute two automata recognizing the same languages (by Proposition 3.25, this can be done in LogSpace), and then apply the NLogSpace algorithm for Reg-separation for languages given by automata.

To prove NLogSpace-hardness, we reduce the problem of accessibility: given an oriented graph $G = (V, A)$ and two distinct vertices s, t of G, is there a path from s to t in G? This is a well-known NLogSpace-complete problem.

Given an oriented graph $G = (V, A)$, we define the monoid $M = V^2 \cup \{0, 1\}$, endowed with the multiplication law:

$$(q_1, q_2)(q_1', q_2') = \begin{cases} (q_1, q_2') & \text{if } q_2 = q_1' \\ 0 & \text{otherwise} \end{cases}$$

where 0 is an absorbing element and 1 is the neutral element.

We define two languages over the alphabet A, meaning that letters are arcs in G. They are both recognized by the morphism $\alpha : A^* \to M$, where for every arc $\overrightarrow{uv} \in A$, we set $\alpha(\overrightarrow{uv}) = (u, v)$. By induction, we can see that

$$\alpha(\overrightarrow{u_1v_1} \cdots \overrightarrow{u_nv_n}) = \begin{cases} 1 & \text{if } n = 0 \\ 0 & \text{if } v_i \neq u_{i+1} \text{ for some } 1 \leqslant i < n \\ (u_1, v_n) & \text{otherwise} \end{cases}$$

In particular, for every $u, v \in V^2$, the language $\alpha^{-1}(\{(u, v)\})$ contains all the non-empty sequences of arcs obtained by following paths from u to v.

The two languages we consider are $L_1 = \alpha^{-1}(\{(s, t)\})$ and $L_2 = A^* = \alpha^{-1}(M)$. Observe that since $L_1 \subset L_2$, we have $L_1 \cap L_2 = \varnothing$ if and only if $L_1 = \varnothing$, which is equivalent to the non-existence of a path from s to t in G by definition of α.

Observe that the monoid M, as well as the images $\alpha(a)$ for $a \in A$ can be computed in LogSpace. Therefore, we obtain a LogSpace reduction from the Reg-separation problem for languages given by monoid morphisms to the accessibility problem.

This implies that the Reg-separation problem is NLogSpace-complete for languages given by monoids, which is the same as for automata. □

Before proving Theorem 3.37, we list below a few examples of well-known varieties, as well as some observations regarding Theorem 3.37:

- As shown in [Cho and Huynh, 1991], FO-membership is PSpace-hard even when the input languages are given by their minimal automata. Starting from a DFA $\mathcal{A}$ recognizing a language L, we can construct in polynomial time an automaton recognizing $\overline{L}$ (we just have to swap final states of $\mathcal{A}$ with non-final states). Therefore, the map $L \mapsto (L, \overline{L})$ is a polynomial-time reduction from FO-membership for a language given by a DFA to FO-separation for languages given by automata. In particular, we obtain that FO-separation is PSpace-hard: In view of Theorem 3.37, we obtain that FO-monoids-separation is also PSpace-hard since FO is a variety.

- Similarly, AT-separation is NP-complete when its inputs are given by NFA (even by DFA) according to [Van Rooijen and Zeitoun, 2013]. By Theorem 3.37, we thus obtain that it is also NP-complete when its inputs are given by monoids morphisms.

- Separation for the class of piecewise testable languages is PTime-complete when the input languages are given by automata, as shown by [Masopust, 2018]. Thus, Theorem 3.37 implies that this complexity result transfers to separation for languages given by monoids.

- All the levels of the dot-depth and the Straubing-Thérien hierarchies are positive varieties (and even varieties for the full-levels). Since these hierarchies are infinite [Brzozowski and Knast, 1978], none of their levels is Reg. Therefore, Theorem 3.37 applies to all their levels. This implies that studying complexity of the separation problem only requires to consider one type of recognizer.

3.3.1 Overview of the proof

The remainder of this section is devoted to the proof of Theorem 3.37. We thus fix a positive variety $\mathcal{C}$ such that its Boolean closure Bool($\mathcal{C}$) is different from Reg. We also fix a pair of automata $(\mathcal{A}_1, \mathcal{A}_2)$ recognizing two languages L_1 and L_2. We construct a pair of morphisms (α_1, α_2) recognizing the languages $\alpha_1^{-1}(F_1)$ and $\alpha_2^{-1}(F_2)$ for some accepting sets F_1, F_2.

This construction must be a reduction: we want L_1 to be $\mathcal{C}$-separable from L_2 if and only if so is $\alpha_1^{-1}(F_1)$ from $\alpha_2^{-1}(F_2)$. We also want the reduction to be in LogSpace. In the following, we show how to construct such pairs (α_1, α_2) and (F_1, F_2). For the sake of readability, we only focus on proving that their size is polynomial with respect to $(\mathcal{A}_1, \mathcal{A}_2)$. However, we claim that the construction we present lies actually in LogSpace.

Observe that the naive approach consisting in computing the transition monoid of $\mathcal{A}_1$ and $\mathcal{A}_2$ does not lead to a LogSpace reduction, since some automata $\mathcal{A}$ have transition monoids of exponential size (with respect to $\mathcal{A}$). Thus, we have to look for a construction where $(L_1, L_2) \neq (\alpha_1^{-1}(F_1), \alpha_2^{-1}(F_2))$.

We proceed as follows. We modify L_1, L_2 to obtain two new languages L_1' and L_2' such that:

(1) L_1' is $\mathcal{C}$-separable from L_2' if and only if L_1 is $\mathcal{C}$-separable from L_2.

(2) There is a monoid recognizing L_1' and L_2' of polynomial size with respect to $\mathcal{A}_1, \mathcal{A}_2$.

To construct L_1' and L_2', we actually modify $\mathcal{A}_1$ and $\mathcal{A}_2$ into two new automata $\mathcal{A}_1'$ and $\mathcal{A}_2'$ as described in Subsection 3.3.2. This construction relies on auxiliary languages, whose construction will be presented in Subsection 3.3.3.

We then prove Property (2) in Subsection 3.3.4, ensuring that the reduction is LogSpace. Finally, in Subsection 3.3.5, we prove that $\mathcal{C}$-separability transfers from $\mathcal{A}_1, \mathcal{A}_2$ to $\mathcal{A}_1', \mathcal{A}_2'$ (and conversely), ensuring Property (1) together with the correctness of the reduction. Altogether, this proves Theorem 3.37.

3.3.2 The construction

Our construction is motivated by properties of the transition monoid of a given deterministic automaton. Recall that this monoid recognizes the language accepted by $\mathcal{A}$, but may have exponential size with respect to $\mathcal{A}$. However, a key observation is that it has polynomial size when all the transitions are labeled by different letters, as shown by the following lemma.

Lemma 3.40. *Let $(Q, \Sigma, I, F, \delta)$ be an automaton where each letter labels at most one transition. Then its transition monoid has size at most $|Q|^2 + 2$.*

Proof. Let $a \in \Sigma$. Observe that the partial function δ_a is defined on at most one state. Assume that it is defined on q and that its value is q', i.e. that $q \xrightarrow{a} q'$ is the only transition with label a. Note then that every composition of such functions is defined on at most one state. There are thus two special elements: the function defined nowhere, denoted by 0, the identity δ_ε also denoted by 1, and functions defined on exactly one state. Such a function is entirely defined by q (the state on which it is defined) and q' (its image of q). Thus, the transition monoid is isomorphic to a submonoid of $Q^2 \cup \{0, 1\}$ endowed with the following multiplication:

$$(q_1, q_1') \cdot (q_2, q_2') = \begin{cases} (q_1, q_2') & \text{if } q_1' = q_2, \\ 0 & \text{otherwise,} \end{cases}$$

where 0 is an absorbing element and 1 is a neutral element. This monoid has size $|Q|^2 + 2$, which concludes the proof. $\square$

Observe that this bound is tight, as shown by Example 3.28 for the language $(ab)^*$: its minimal automaton has two states and its transition monoid has size six.

Coming back to the proof of Theorem 3.37, recall that we want to construct $\mathcal{A}'_1$ and $\mathcal{A}'_2$ that recognize languages fulfilling points (1) and (2) (page 191).

A first idea is then to rename all the transitions of $\mathcal{A}_1$ and $\mathcal{A}_2$ to enforce that each letter appears on at most one transition. By doing so, the obtained automata have small transition monoids. However, this kind of renaming may not preserve $\mathcal{C}$-separability. To overcome this problem, we will replace the labels of each transition in order to

- simulate distinct transitions and thus obtain a "small" transition monoid by (a generalization of) Lemma 3.40
- preserve $\mathcal{C}$-separability and $\mathcal{C}$-inseparability of the recognized languages.

To this end, we consider automata where transitions are labeled by regular languages instead of letters. A word w is recognized by such an automaton if there is a path $q_0 \xrightarrow{K_1} q_1 \cdots \xrightarrow{K_n} q_n$ where q_0 is an initial state, q_n a final state, and w can be decomposed as $w_1 \cdots w_n$ where each factor w_i lies in K_i for $1 \leqslant i \leqslant n$. Note in particular that these machines recognize only regular languages: we can recover a classic automaton from such an enriched one by replacing each transition $q \xrightarrow{K} q'$ by an automaton $\mathcal{A}_K$ recognizing K. More precisely, assuming that $\mathcal{A}_K$ has a single initial state q_i and a single final state q_f (which can be done without loss of generality for non-empty languages using standard constructions on automata), we identify q with q_i and q' with q_f. An example of this construction is depicted on Figure 3.7.

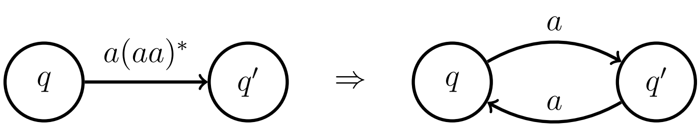


Figure 3.7 – Flattening a transition $q \xrightarrow{a(aa)^*} q'$.

Starting from the automata $\mathcal{A}_1$ and $\mathcal{A}_2$, we construct enriched automata $\mathcal{A}'_1$ and $\mathcal{A}'_2$ by tagging their transitions with (regular) languages. This allows us to simulate distinct transitions (using pairwise disjoint tags), and to transfer separability properties (using tags that cannot be distinguished by $\mathcal{C}$).

The generic construction follows. Given an n-tuple of regular languages $\mathbf{K} = (K_1, \ldots, K_n)$ and an automaton $\mathcal{A}$ with at most n transitions, we denote by $\mathcal{A}[\mathbf{K}]$ the automaton obtained by replacing the i-th transition $q \xrightarrow{a} q'$ of $\mathcal{A}$

by $q \xrightarrow{aK_i} q'$. In this case, we say that $K_1, \dots, K_n$ are the *tagging languages* and $\mathcal{A}[\mathbf{K}]$ is the *tagged automaton*. Note that we may assume that $\mathcal{A}$ comes with an order on its transitions, hence this construction is well-defined. Observe also that $\mathcal{A}$ and $\mathcal{A}[\mathbf{K}]$ share the same sets of states, initial states and final states.

Before giving the value of $\mathbf{K}$ that we use, we state the two main properties of this construction. The first one is a generalization of Lemma 3.40: it replaces the hypothesis of distinct transition labels by pairwise disjoint transition languages.

Proposition 3.41. *Let $\mathbf{K}$ be a tuple of pairwise disjoint languages and let $\mathcal{A}$ be an automaton. Let $\beta : A^* \to N$ be a monoid morphism recognizing all the languages in $\mathbf{K}$. Then we can construct a monoid morphism recognizing the same language as $\mathcal{A}[\mathbf{K}]$ of polynomial size with respect to $\mathcal{A}$ and N.*

This result is proven in Subsection 3.3.4. Let us summarize the reduction: starting from $\mathcal{A}_1$ and $\mathcal{A}_2$, we tag these automata using pairwise disjoint languages. Then, we compute the transition monoid of these tagged automata. By Proposition 3.41, we know that this reduction can be done in polynomial time. However, we still have to prove it is a reduction from $\mathcal{C}$-separation for automata to $\mathcal{C}$-separation for monoids. In other words, the second main property of this reduction is a separability transfer result. It will be stated in Proposition 3.45, but we first need to introduce some terminology before.

Consider two transitions $q_1 \xrightarrow{a} q_1'$ in $\mathcal{A}_1$ and $q_2 \xrightarrow{a} q_2'$ in $\mathcal{A}_2$, labeled by the same letter a. When tagging $\mathcal{A}_1$ and $\mathcal{A}_2$, these transitions become labeled by some languages aK and aK'. To obtain a separability transfer, we need for K, K' to be non $\mathcal{C}$-separable. To see this, for $i = 1, 2$, consider H_i be the language of words labeling a path from q_i to q_i' in $\mathcal{A}_i$. In $\mathcal{A}_1$ and $\mathcal{A}_2$, the considered transitions are labeled with the same letter a. Therefore, the languages H_1 and H_2 intersect and thus are not $\mathcal{C}$-separable. If K and K' are $\mathcal{C}$-separable, then $\mathcal{C}$ may be able to distinguish H_1' from H_2' where, for $i = 1, 2$, H_i' is the language of labels of paths from q_i to q_i' in $\mathcal{A}_i'$. Therefore, $\mathcal{C}$-inseparability may not transfer from $\mathcal{A}_1, \mathcal{A}_2$ to $\mathcal{A}_1', \mathcal{A}_2'$. This is why we need the tagging languages K, K' to be $\mathcal{C}$-inseparable.

However, the letter a may appear on more than one transition in $\mathcal{A}_1$ and $\mathcal{A}_2$, and we still need the tagging languages present on these transitions to be non $\mathcal{C}$-separable. However, the pairwise $\mathcal{C}$-inseparability is not sufficient. We use here a stronger notion: $\mathcal{C}$-coverability.

This variant of the separation problem was introduced in [Place and Zeitoun, 2017b], as a tool to study the separation problem. The intuition motivating the introduction of this problem is the following. In view of Proposition 3.36, solving $\mathcal{C}$-membership for a language L is equivalent to solving it for every language recognized by the syntactic morphism of L. Since the set of all these languages has a robust structure (it is stable by unions, intersections and quotients), it is natural to follow the same approach for separation.

Definition 3.42. A set $\{K_1, \dots, K_n\}$ of languages is $\mathcal{C}$-*coverable* if we can find some languages $L_1, \dots, L_p$ such that:

- $L_1, \dots, L_p \in \mathcal{C}$.
- $K_1 \cup \dots \cup K_n \subset L_1 \cup \dots \cup L_p$.
- No L_i intersects all the languages $K_1, \dots, K_n$.

In this case, the set $\{L_1, \dots, L_p\}$ is said to be a $\mathcal{C}$-*cover* of $\{K_1, \dots, K_n\}$.

The notion of covering is illustrated with Figure 3.8. The covering problem is a generalization of the separation problem for more than two languages, in a way that encapsulates more information than just the one given by pairwise separability of the languages. In particular, when considering coverability of only two languages, we end up back with the initial separation problems, as shown by the following lemma.

Lemma 3.43. *Let $\mathcal{C}$ be a class of regular languages closed under Boolean operations, and let K_1, K_2 be two regular languages. The set $\{K_1, K_2\}$ is $\mathcal{C}$-coverable if and only if K_1 and K_2 are $\mathcal{C}$-separable.*

Proof. Assume that K_1 and K_2 are separated by a language $L \in \mathcal{C}$. Then we can write $K_1 \cup K_2 \subset L \cup \overline{L}$. Observe that both L and $\overline{L}$ are elements of $\mathcal{C}$ (since $\mathcal{C}$ is closed under complement). Moreover, L intersects K_1 but not K_2, and $\overline{L}$ intersects K_2 but not K_1. Thus $\{K_1, K_2\}$ is $\mathcal{C}$-coverable.

Conversely, assume that $\{K_1, K_2\}$ is $\mathcal{C}$-coverable. Then we can find languages $L_1, \dots, L_p$ in $\mathcal{C}$ covering $\{K_1, K_2\}$. Denote by L the union of all L_i intersecting K_1 for $1 \leqslant i \leqslant p$. Observe that $L \in \mathcal{C}$ since $\mathcal{C}$ is closed under union.

Let $w \in K_1$. Since $L_1, \dots, L_p$ is a covering of $\{K_1, K_2\}$, there exists an integer i such that $w \in L_i$. This language L_i intersects K_1, hence it is contained in L. Therefore, $w \in L$ and $K_1 \subset L$.

Moreover, L does not intersect K_2, since otherwise, there would be a language L_i intersecting both K_1 and K_2. Finally, we obtain that L is a separator in $\mathcal{C}$ of K_1 and K_2. $\square$

Note that the property of non-$\mathcal{C}$-coverability refines the notion of pairwise non-separability. In other words, if two languages of $\{K_1, \dots, K_n\}$ are $\mathcal{C}$-separable, then $\{K_1, \dots, K_n\}$ is $\mathcal{C}$-coverable. However, if all the pairs (K_i, K_j) are not $\mathcal{C}$-separable, then $\{K_1, \dots, K_n\}$ may or may not be $\mathcal{C}$-coverable, see Figure 3.8. This is illustrated by the following example.

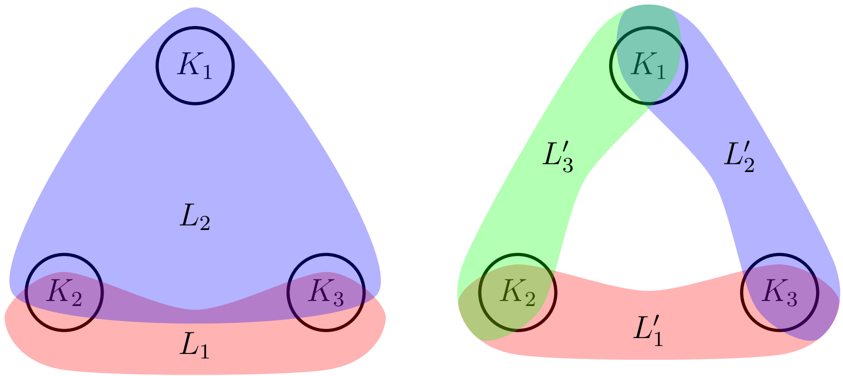


Figure 3.8 – $\{K_1, K_2, K_3\}$ is covered by $L'_1 \cup L'_2 \cup L'_3$, but not by $L_1 \cup L_2$

Example 3.44. Let $K_1 = (aa)^+ + b(bb)^+$, $K_2 = (bb)^+ + c(cc)^+$ and $K_3 = (cc)^+ + a(aa)^+$. Observe that any pair (K_i, K_j) is not FO-separable. Indeed for example, K_1 contains $b(bb)^*$, which is not FO-separable from $(bb)^+$ (contained in K_2). However, $\{K_1, K_2, K_3\}$ is FO-covered by $\{a^*, b^*, c^*\}$.

On the other hand, if we take $L_1 = (aaa)^*$, $L_2 = (aaa)^*a$ and $L_3 = (aaa)^*aa$, they are still pairwise non FO-separable, but $\{L_1, L_2, L_3\}$ is not FO-coverable.

Informally, the reason for which L_1, L_2 and L_3 are pairwise non FO-separable is that all of them contains words a^n with arbitrarily large n. Conversely, K_1 and K_2 are non FO-separable because of arbitrarily long words of the form b^n, and for K_2, K_3 it is because of c^n. Therefore, the set $\{L_1, L_2, L_3\}$ is not FO-coverable since the pairs (L_i, L_j) have the same cause of non-separability, while the set $\{K_1, K_2, K_3\}$ is FO-coverable because these causes are independent.

For more insight on the covering problem and its benefits towards separation, we refer to [Place and Zeitoun, 2017b]. As a final remark, note that the covering problem is the key to solve $\mathcal{C}$-separation for some classes $\mathcal{C}$ such as FO or the lower levels of finitely-based hierarchies: we actually do not know how to solve $\mathcal{C}$-separation directly, but decidability of $\mathcal{C}$-covering yields an algorithm for $\mathcal{C}$-separation.

With this more general problem, we can finally state the separability transfer result we look for. It states that when $\mathcal{C}$ "cannot distinguish" the tagging languages, then the tagging process preserves $\mathcal{C}$-separability of the recognized languages.

Proposition 3.45. *Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be two automata over an alphabet A. Take $\mathbf{K} = (K_1, \ldots, K_n)$ a tuple of languages over an alphabet disjoint from A such that $\{K_1, \ldots, K_n\}$ is not* $\mathrm{Bool}(\mathcal{C})$*-coverable.*

Then the languages recognized by $\mathcal{A}_1$ and $\mathcal{A}_2$ are $\mathcal{C}$-separable if and only if the languages recognized by $\mathcal{A}_1[\mathbf{K}]$ and $\mathcal{A}_2[\mathbf{K}]$ are $\mathcal{C}$-separable.

Observe that this proposition considers Bool($\mathcal{C}$)-coverability instead of $\mathcal{C}$-coverability. This is because we are stating a result about separability, hence we need to recover some separability information from a coverability hypothesis. The only tool we described to do so is given by Lemma 3.43. However, it is only valid when $\mathcal{C}$ is closed under Boolean operation. Since $\mathcal{C}$ is only a positive variety, this may not be the case. Therefore, we use here a stronger notion instead: Bool($\mathcal{C}$)-coverability.

Proposition 3.45 is proven in Subsection 3.3.5. Together with Proposition 3.41, it pinpoints the hypotheses that the tagging languages must satisfy. The existence of such languages is given by the following proposition.

Proposition 3.46. *Let $\mathcal{D}$ be a variety such that $\mathcal{D} \neq \mathsf{Reg}$. For any integer n, one may construct a monoid morphism $\beta : B^* \to N$ of polynomial size in n that recognizes n pairwise disjoint languages $K_1, \dots, K_n$ such that $\{K_1, \dots, K_n\}$ is not $\mathcal{D}$-coverable.*

Observe that this proposition constructs languages satisfying all the hypotheses of Propositions 3.41 and 3.45 except that their alphabet B may not be disjoint from the alphabet of $\mathcal{A}_1$ and $\mathcal{A}_2$. However, the properties of Proposition 3.46 do not depend on which alphabet is used. We can thus rename the letters of B (if needed) to ensure this additional condition while preserving the properties of Proposition 3.46.

Assuming Propositions 3.41, 3.45 and 3.46 hold, we show how to end the proof of Theorem 3.37. Take n as the maximal number of transitions of $\mathcal{A}_1$ and $\mathcal{A}_2$. Take a tuple $\mathbf{K} = (K_1, \dots, K_n)$ given by Proposition 3.46 for $\mathcal{D} = \text{Bool}(\mathcal{C})$ ($\mathcal{D}$ is a variety by Lemma 3.34) and denote by $\beta : B^* \to N$ a monoid morphism recognizing $K_1, \dots, K_n$. Note that this is the only use of the hypothesis $\text{Bool}(\mathcal{C}) \neq \mathsf{Reg}$ in Theorem 3.37. Up to renaming letters, we may assume that B is disjoint from the alphabet A of $\mathcal{A}_1$ and $\mathcal{A}_2$. Using Proposition 3.41, we can effectively construct some monoid morphisms $\alpha_1 : A^* \to M_1$ and $\alpha_2 : A^* \to M_2$ recognizing the same languages as $\mathcal{A}_1[\mathbf{K}]$ and $\mathcal{A}_2[\mathbf{K}]$. These morphisms are the output of our reduction. Since M_1 and M_2 have polynomial size with respect to $\mathcal{A}_1$, $\mathcal{A}_2$ and N (by Proposition 3.41) and N has polynomial size with respect to n (by Proposition 3.46), our reduction has the requested complexity. Moreover, using Proposition 3.45, the reduction is also correct since $\mathcal{C}$-separability and $\mathcal{C}$-inseparability transfers between the languages recognized by $\mathcal{A}_1$ and $\mathcal{A}_2$ and those recognized by $\mathcal{A}_1[\mathbf{K}]$ and $\mathcal{A}_2[\mathbf{K}]$. This concludes the proof of Theorem 3.37.

The end of this section is devoted to the proofs of the three remaining propositions: 3.46, 3.41 and 3.45. A subsection is devoted to each of them.

3.3.3 Construction of tagging languages

In this subsection, we prove Proposition 3.46. We thus fix an integer n and a variety $\mathcal{D} \subsetneq \mathsf{Reg}$. We look for n pairwise disjoint languages $K_1, \dots, K_n$ recognized by a morphism $\beta : B^* \to N$ of polynomial size in n, such that $\{K_1, \dots, K_n\}$ is not $\mathcal{D}$-coverable. Before giving the explicit construction, we present its outline on an example, by taking $\mathcal{D} = \mathsf{FO}$.

Example 3.47. Recall that the language $(aa)^*$ is not FO-definable. Lemma 3.43 thus ensures that $\{(aa)^*, a(aa)^*\}$ is not FO-coverable.

Note that this is valid for every letter a. Therefore, for every integer p, we have p sets of non FO-coverable languages: the $\{(a_ia_i)^*, a_i(a_ia_i)^*\}$ for $1 \leqslant i \leqslant p$.

For every $x \in \{0,1\}^p$, consider the language K_x containing all words $w \in \{a_1, \dots, a_p\}^*$ such that, for $1 \leqslant i \leqslant p$, the number $|w|_{a_i}$ of letters a_i in w equals x_i modulo 2. Observe that there are 2^p such languages. Moreover, they are all recognized by the following morphism:

$$\begin{cases} \{a_1, \cdots, a_p\}^* & \to (\mathbb{Z}/2\mathbb{Z})^p \\ w & \mapsto (|w|_{a_1} \bmod 2, \dots, |w|_{a_p} \bmod 2) \end{cases}$$

In particular, note that the monoid $(\mathbb{Z}/2\mathbb{Z})^p$ has size 2^p, which is linear in the number of languages K_x.

Observe that if $x, y \in \{0,1\}^p$ are different, there exists $1 \leqslant i \leqslant p$ such that $x_i \neq y_i$. By symmetry, assume that $x_i = 0$. In particular, each word of L_x has an even number of a_i's, while every word of L_y has an odd number of a_i's. This implies that no word is contained in both L_x and L_y. As a consequence, the languages L_x are pairwise disjoint.

It remains to show that the set $\{K_x, x \in \{0,1\}^p\}$ is not FO-coverable. To this end, we consider the 2^p languages obtained by considering the concatenations $L'_1 \cdots L'_p$ where each L'_i is $(a_ia_i)^*$ or $a_i(a_ia_i)^*$. Observe that each of these concatenations is contained in exactly one K_x. Therefore, it is sufficient to prove that the set of such concatenations is not FO-coverable. This is actually a generic result about $\mathcal{C}$-covering we will state in Lemma 3.48.

Therefore, we obtain 2^p languages, all of them recognized by the same monoid of size 2^p, and whose set is not FO-coverable. Taking $p = \lceil \log_2(n) \rceil$ thus implies Proposition 3.46 in the case of FO.

We now give the construction in the generic case. Take $p = \lceil \log_2(n) \rceil$. Since $\mathcal{D} \neq \mathsf{Reg}$, there exists a regular language L outside of $\mathcal{D}$. Moreover, since $\mathcal{D}$ is a variety, we can rename the letters of L to obtain p languages $L_1, \dots, L_p$ over some pairwise disjoint alphabets $B_1, \dots, B_p$ such that none of $L_1, \dots, L_p$ lies in $\mathcal{D}$.

We use these languages to construct $K_1, \dots, K_n$. To this end, we introduce the notion of projection over an alphabet. If B and C are alphabets such that

$B \subset C$, the projection $\pi_{C,B}$ is the morphism $C^* \to B^*$ erasing the letters of $C \setminus B$. We fix here $B = \bigcup_{i=1}^{p} B_i$.

For $j \in [1,n]$, we define K_j as an intersection $H_1 \cap \cdots \cap H_p$ where each H_i is $\pi_{B,B_i}^{-1}(L_i)$ or $\pi_{B,B_i}^{-1}(B_i^* \setminus L_i)$. Since we have 2^p choices and $n \leqslant 2^p$, we can thus define n languages $K_1, \dots, K_n$.

Assume that there exist two integers i, j such that $K_i \cap K_j$ contains a word w. Then for $1 \leqslant k \leqslant p$, we have $\pi_{B,B_k}(w) \in \pi_{B,B_k}(K_i) \cap \pi_{B,B_k}(K_j)$. Therefore, $\pi_{B,B_k}(K_i)$ and $\pi_{B,B_k}(K_j)$ are not disjoint. Since these languages are either L_k or $B_k^* \setminus L_k$, we have $\pi_{B,B_k}(K_i) = \pi_{B,B_k}(K_j)$. Since this is valid for every integer k, we have $K_i = K_j$ by construction. This implies that the languages $K_1, \dots, K_n$ are pairwise disjoint.

Moreover, note that $L_1, \dots, L_p$ are the same up to letter renaming. For $1 \leqslant i \leqslant p$, let $\varphi_i : B_i^* \to B_1^*$ be a bijection such that $\varphi_i(L_i) = L_1$, obtained by renaming letters. If $\psi_1 : B_1^* \to M$ is a morphism recognizing L_1, then $\varphi_i \circ \psi_1$ is a morphism $B_i^* \to M$ recognizing L_i. Denote by ψ_i this morphism. The languages $K_1, \dots, K_n$ are then recognized by the single morphism:

$$\varphi\colon \begin{cases} B^* & \to M^p \\ u & \mapsto (\psi_1(\pi_{B,B_1}(u)), \cdots, \psi_p(\pi_{B,B_p}(u))). \end{cases}$$

Due to our choice of p, the size of M^p is polynomial in n.

It remains to show that $\{K_1, \dots, K_n\}$ is not $\mathcal{D}$-coverable. By definition of coverability, observe that it is sufficient to find a non $\mathcal{D}$-coverable set of languages $\{K'_1, \dots, K'_n\}$ such that each K_i contains K'_i. In our case, note that each language K_j contains a concatenation $L'_1 \cdots L'_p$ where for $i \in [1,p]$, $L'_i = L_i$ or $L'_i = B_i^* \setminus L_i$.

Since $L_i \notin \mathcal{D}$ and $\mathcal{D}$ is closed under complement, L_i is not $\mathcal{D}$-separable from $B_i^* \setminus L_i$, hence the set $\{L_i, B_i^* \setminus L_i\}$ is not $\mathcal{D}$-coverable. It is thus sufficient to prove that non $\mathcal{D}$-coverability is preserved when considering concatenations. This is actually a consequence of the following lemma, a simplified version of a result from [Place and Zeitoun, 2017b]. It is an extension of a similar result about inseparability: if K is not $\mathcal{D}$-separable from K' and L is not $\mathcal{D}$-separable from L', then KK' is not $\mathcal{D}$-separable from LL' when $\mathcal{D}$ is a class of regular languages closed under quotients. We will come back to this point in Chapter 4.

Lemma 3.48. *Let $\mathcal{D} \subset$ Reg be a class closed under quotients. Let $\{L_1, \dots, L_n\}$ and $\{L'_1, \dots, L'_p\}$ be two non-$\mathcal{D}$-coverable sets of languages. Then the product set $\{L_iL'_j \mid i \in [1,n], j \in [1,p]\}$ is not $\mathcal{D}$-coverable.*

By Lemma 3.48, we thus obtain that the set of all the products $L'_1 \cdots L'_p$ is not $\mathcal{D}$-coverable, which ensures that neither is $\{K_1, \dots, K_n\}$. This concludes the proof of Proposition 3.46.

For the sake of completeness, we end this subsection by giving the proof of Lemma 3.48. We thus fix a quotient-closed class $\mathcal{D}$ of regular languages, and two non $\mathcal{D}$-coverable sets of languages $\{L_1, \dots, L_n\}$ and $\{L'_1, \dots, L'_p\}$. Let $L = \bigcup_{i=1}^{n} L_i$ and $L' = \bigcup_{j=1}^{p} L'_j$.

Consider a $\mathcal{D}$-covering $K_1, \dots, K_q$ of $\{L_iL'_j \mid i \in [1,n], j \in [1,p]\}$. In particular, observe that $LL' \subset K_1 \cup \cdots \cup K_q$. We have to prove that some K_i intersects all the products $L_iL'_j$.

First assume that we can find two words $u \in L$ and $v \in L'$ such that the following assertions hold for every $w \in A^*$ and $1 \leqslant k \leqslant q$:

(1) If $u \in K_kw^{-1}$, then K_kw^{-1} intersects every L_i for $1 \leqslant i \leqslant n$.

(2) If $v \in w^{-1}K_k$, then $w^{-1}K_k$ intersects every L'_j for $1 \leqslant j \leqslant p$.

Using these two words, we show how to conclude the proof of Lemma 3.48. By construction, we have $uv \in LL'$, hence uv lies in some K_k. We prove that this language intersect every product $L_iL'_j$. Let $i \in [1,n]$ and $j \in [1,p]$.

We have $uv \in K_k$, hence $u \in K_kv^{-1}$. By (1), we obtain that there exists a word $u_i \in K_kv^{-1} \cap L_i$. We thus have $u_iv \in K_k$, i.e. $v \in u_i^{-1}K_k$. Using (2), we obtain that there exists a word $v_j \in u_i^{-1}K_k \cap L'_j$. Finally, we obtain that the word u_iv_j lies in K_k and in $L_iL'_j$.

Therefore, the language K_k intersects all the $L_iL'_j$ for $1 \leqslant i \leqslant n$ and $1 \leqslant j \leqslant p$, ensuring that $\{L_iL'_j \mid i \in [1,n], j \in [1,p]\}$ is not $\mathcal{D}$-coverable.

It remains to prove the existence of u and v. Note that assertions (1) and (2) are symmetrical, hence we only need to prove the first one. It follows from the non-$\mathcal{D}$-coverability of $\{L_1, \dots, L_n\}$. Consider the set $\mathcal{Q} = \{K_kw^{-1} \mid w \in A^*, 1 \leqslant k \leqslant q\}$ of all the possible right quotients of the languages $K_1, \dots, K_q$. Observe that since $K_1, \dots, K_q$ are regular, each of them has a finite number of quotients, hence $\mathcal{Q}$ is finite. Moreover, since $\mathcal{D}$ is closed under quotients, we have $\mathcal{Q} \subset \mathcal{D}$.

Property (1) can be rephrased as follows: there exists a word $u \in L$ such that every $Q \in \mathcal{Q}$ containing u intersects all the L_i's for $1 \leqslant i \leqslant n$. Assume that this is false, i.e. for every $u \in L$, there exists $Q_u \in \mathcal{Q}$ containing u and which does not intersect all the L_i's.

Observe that $\{Q_u, u \in L\}$ is finite and contains only languages of $\mathcal{D}$, since it is a subset of $\mathcal{Q}$. Moreover, we claim that is a $\mathcal{D}$-cover of $\{L_1, \dots, L_n\}$. Indeed, for every $u \in L$, we have $u \in Q_u$ and, by construction of Q_u, none of the Q_u intersect all the L_i's. We thus obtain a contradiction since $\{L_1, \dots, L_n\}$ is not $\mathcal{D}$-coverable, which concludes the proof of Lemma 3.48.

3.3.4 Complexity

In this subsection, we prove Proposition 3.41. We thus fix an automaton $\mathcal{A}$ with alphabet A, n transitions and an n-tuple $\mathbf{K} = (K_1, \dots, K_n)$ of pairwise disjoint languages recognized by a monoid morphism $\beta : B^* \to N$. We construct a monoid morphism $\chi : (A \cup B)^* \to M$ recognizing the same language as $\mathcal{A}[\mathbf{K}]$, and such that the monoid M has polynomial size with respect to $\mathcal{A}$ and N in two steps. Throughout the reduction, we will apply it on the following example.

Example 3.49. We will take $\mathcal{A}$ as the minimal automaton of the language $(aa)^*$. Since it has two transitions, we need two tagging languages: we take $(bb)^*$ and $b(bb)^*$. Note that $\{(bb)^*, b(bb)^*\}$ is not FO-coverable, hence this choice of tagging languages could actually occur when considering FO-separation.

The automata $\mathcal{A}$ and $\mathcal{A}[\mathbf{K}]$ are depicted on Figure 3.9. Observe that $\mathcal{A}[\mathbf{K}]$

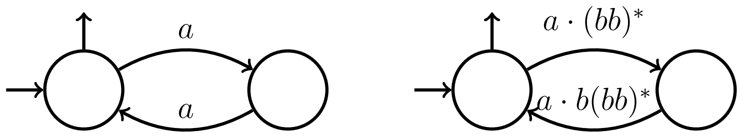


Figure 3.9 – Running example of Subsection 3.3.4

recognizes the language $(a(bb)^*ab(bb)^*)^*$.

The first step is to construct an automaton $\widehat{\mathcal{A}}$ from $\mathcal{A}[\mathbf{K}]$ satisfying the hypothesis of Lemma 3.40. To this end, we construct a monoid C and a morphism $\varphi : (A \cup B)^* \to C$ recognizing all the transition labels aK_i of $\mathcal{A}[\mathbf{K}]$. Observe that, since the languages in $\mathbf{K}$ are pairwise disjoint, the automaton $\widehat{\mathcal{A}}$ obtained by replacing each transition $q \xrightarrow{aK_i} q'$ by all the transitions $q \xrightarrow{s} q'$ for $s \in \varphi(aK_i)$ satisfies the hypothesis of Lemma 3.40: each element of the monoid C appears on at most one transition of $\widehat{\mathcal{A}}$.

We thus obtain that the transition monoid of $\widehat{\mathcal{A}}$ has polynomial size with respect to the number of states of $\mathcal{A}[\mathbf{K}]$, i.e. of $\mathcal{A}$. Our second step then consists in using this transition monoid together with φ to construct another monoid morphism recognizing the same language as $\mathcal{A}[\mathbf{K}]$.

First step: recognizing the tags

The construction of the morphism φ and the monoid C relies on the following lemma.

Lemma 3.50. *Given a morphism $\beta : B^* \to N$ and an alphabet A disjoint from B, the languages aL where $a \in A$ and L is recognized by β are all recognized by a monoid of size $(|A| + 1)|N| + 2$.*

Proof. Define $C = (A \times N) \cup N \cup \{0, 1\}$. We use the different components to remember the number of letters of A we encounter: the component $A \times N$ handles words in AB^*, the component N handles words in B^+ and $0, 1$ take care of the other words. More precisely, we define the multiplication law over C by taking for $a, b \in A$ and $s, t \in N$:

$$(a, s)(b, t) = 0, \qquad (a, s)t = (a, st), \qquad \textit{and} \qquad t(a, s) = 0,$$

where 0 is an absorbing element and 1 is a neutral element. We can check that this multiplication law is associative.

We then define a morphism $\varphi : (A \cup B)^* \to C$ by taking for $a \in A$ and $b \in B$:

$$\varphi(a) = (a, 1) \text{ and } \varphi(b) = \beta(b).$$

Observe that by construction, for every $w \in B^*$:

$$\begin{aligned} \varphi(aw) &= (a, \beta(w)) \\ \varphi(w) &= \beta(w) \text{ if } w \neq \varepsilon \\ \varphi(\varepsilon) &= 1 \end{aligned}$$

and $\varphi(w) = 0$ if $w \notin AB^* \cup B^*$.

As a consequence, we have $\varphi^{-1}(a, s) = a\beta^{-1}(s)$ for any $a \in A$ and $s \in N$. Therefore, the morphism φ recognizes all the languages aL where a is a letter of A and L is recognized by β: the corresponding accepting set is $\{a\} \times \beta(L)$. □

To obtain a monoid morphism φ recognizing the tags in $\mathcal{A}[\mathbf{K}]$, it is then sufficient to apply Lemma 3.50 to the monoid morphism β recognizing the languages in $\mathbf{K}$. It remains to show how to use φ to obtain a monoid recognizing the same language as $\mathcal{A}[\mathbf{K}]$.

Example 3.51. The morphism $\beta : \{b\}^* \to \mathbb{Z}/2\mathbb{Z}$ defined by $\beta(b) = 1$ recognizes the tagging languages $(bb)^*$ and $b(bb)^*$. The monoid constructed by Lemma 3.50 is then $C = (\{a\} \times \mathbb{Z}/2\mathbb{Z}) \cup \mathbb{Z}/2\mathbb{Z} \cup \{0_C, 1_C\}$ with

$\times$	$(a,0)$	$(a,1)$	0	1	0_C	1_C
$(a,0)$	0_C	0_C	$(a,0)$	$(a,1)$	0_C	$(a,0)$
$(a,1)$	0_C	0_C	$(a,1)$	$(a,0)$	0_C	$(a,1)$
0	0_C	0_C	0	1	0_C	0
1	0_C	0_C	1	0	0_C	1
0_C	0_C	0_C	0_C	0_C	0_C	0_C
1_C	$(a,0)$	$(a,1)$	0	1	0_C	1_C

Note that, in order to avoid confusion, 0 and 1 denote elements of $\mathbb{Z}/2\mathbb{Z}$ while the absorbing element and the neutral element of C are denoted by $0_C, 1_C$.

The monoid morphism $\varphi : \{a, b\}^* \to C$ is then defined by $\varphi(a) = (a, 0)$ and $\varphi(b) = 1$. For $n > 0$, we have

$$\varphi(ab^n) = (a, n \bmod 2) \text{ and } \varphi(b^n) = n \bmod 2.$$

Second step: the entire construction

The morphism φ recognizes all the languages labeling transitions in $\mathcal{A}[\mathbf{K}]$. Moreover, its codomain C has polynomial size with respect to A and N. We now show how to use φ to construct a monoid recognizing the same language as $\mathcal{A}[\mathbf{K}]$. The monoid we construct will have polynomial size with respect to $\mathcal{A}[\mathbf{K}]$ and to C, hence with respect to $\mathcal{A}$ and N.

To obtain this monoid, we will use Lemma 3.40, which ensures that an automaton has a small transition monoid when each letter labels at most one transition. Observe that it makes no sense to apply this lemma to $\mathcal{A}[\mathbf{K}]$ since its transitions are labeled by languages instead of letters. However, we know that these languages are pairwise disjoint. We thus construct an auxiliary automaton $\widehat{\mathcal{A}}$ such that:

1. $\widehat{\mathcal{A}}$ satisfies the hypothesis of Lemma 3.40.
2. We can construct a monoid recognizing the same language as $\mathcal{A}[\mathbf{K}]$ from the transition monoid of $\widehat{\mathcal{A}}$.

Remark 3.52. Recall that C is endowed with a monoid structure. To prove the first item, we can actually forget about this structure, we just consider it as a set of letters. However, the second item requires to use also the monoid structure of C: it is used as a building block for constructing the final monoid.

Therefore, C (as a monoid) is equipped with its multiplicative law, and we will consider words over (the alphabet) C, hence we also consider concatenation over C^*. To avoid confusions, we will explicitly write $s \cdot t$ to denote the element of C obtained using the multiplicative law of C, and st to denote the 2-letter word over the alphabet C.

The automaton $\widehat{\mathcal{A}}$ is constructed by applying φ to every transition label of $\mathcal{A}[\mathbf{K}]$. However, this gives an automaton whose transitions are labeled by subsets of C where we would want only elements of C. This is not restrictive: each time we see a transition labeled by $\{s_1, \cdots, s_p\}$, we replace it by p transitions labeled by each of the s_i's. The entire construction is given by the following:

- $\widehat{\mathcal{A}}$ and $\mathcal{A}[\mathbf{K}]$ have the same sets of states, initial states and final states.
- The alphabet of $\widehat{\mathcal{A}}$ is C.
- There is a transition $q \xrightarrow{s} q'$ in $\widehat{\mathcal{A}}$ whenever there exists a transition $q \xrightarrow{K} q'$ in $\mathcal{A}[\mathbf{K}]$ with $s \in \varphi(K)$.

Example 3.53. In our example, the automaton $\widehat{\mathcal{A}}$ is depicted on Figure 3.10.

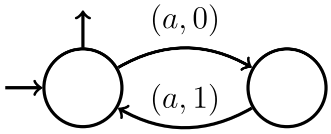


Figure 3.10 – The automaton $\widehat{\mathcal{A}}$

Using that transitions in $\mathcal{A}[\mathbf{K}]$ are labeled by pairwise disjoint languages, we can prove the following claim.

Claim 3.54. *Every $s \in C$ is the label of at most one transition in $\widehat{\mathcal{A}}$.*

Proof. Assume that $s \in C$ appears on two transitions in $\widehat{\mathcal{A}}$. Let K, K' be the corresponding labels of the transitions in $\mathcal{A}[\mathbf{K}]$. We have $s \in \varphi(K) \cap \varphi(K')$. Since φ recognizes K and K', we obtain that K and K' intersect. This is a contradiction since the tagging languages are pairwise disjoint. □

This claim ensures that $\widehat{\mathcal{A}}$ satisfies the hypothesis of Lemma 3.40. Therefore, this automaton has a small transition monoid. In particular, there is a small monoid recognizing the same language as $\widehat{\mathcal{A}}$. To lift this result to the language recognized by $\mathcal{A}[\mathbf{K}]$, we first describe the language recognized by $\widehat{\mathcal{A}}$.

Given a word $u = u_1 \cdots u_p$ where each u_i lies in AB^*, we define $\widehat{\varphi}(u)$ as the word of C^p whose i-th letter is $\varphi(u_i)$ for $1 \leqslant i \leqslant p$. Observe that this is well-defined: since A and B are disjoint, the factorization of u in $(AB^*)^*$ is unique. With this application, we have the following.

Claim 3.55. *The following hold:*

- *If a word $u \in (A \cup B)^*$ is accepted by $\mathcal{A}[\mathbf{K}]$ then $u \in (AB^*)^*$ and $\widehat{\varphi}(u)$ is accepted by $\widehat{\mathcal{A}}$.*
- *If $v \in C^*$ is accepted by $\widehat{\mathcal{A}}$, then there exists $u \in (AB^*)^*$ accepted by $\mathcal{A}[\mathbf{K}]$ such that $v = \widehat{\varphi}(u)$.*

Proof.

- Assume that $u \in (A \cup B)^*$ is accepted by $\mathcal{A}[\mathbf{K}]$. By definition, there is an accepting path in $\mathcal{A}[\mathbf{K}]$ whose labels are $a_1K_1, \cdots, a_pK_p$ and such that u can be decomposed as $u_1 \cdots u_p$ where $u_i \in a_iK_i$ for $1 \leqslant i \leqslant p$. In particular, we have $u \in (AB^*)^*$. By definition of $\widehat{\mathcal{A}}$, the word $\widehat{\varphi}(u) = \varphi(u_1) \cdots \varphi(u_p)$ of C^p labels an accepting path in $\widehat{\mathcal{A}}$. Thus, $\widehat{\varphi}(u)$ is accepted by $\widehat{\mathcal{A}}$.
- Let $v \in C^*$ accepted by $\widehat{\mathcal{A}}$. We write $v = s_1 \cdots s_p$ such that $s_1, \ldots, s_p$ are the letters of v.

By hypothesis, $s_1, \ldots, s_p$ are the labels of an accepting path in $\widehat{\mathcal{A}}$. By construction of $\widehat{\mathcal{A}}$, there is an accepting path labeled by $a_1K_1, \cdots, a_pK_p$ in $\mathcal{A}[\mathbf{K}]$ such that $s_i \in \varphi(a_iK_i)$ for $1 \leqslant i \leqslant p$.

Since φ recognizes every language a_iK_i, for $1 \leqslant i \leqslant p$, there exists $u_i \in a_iK_i$ such that $\varphi(u_i) = s_i$. We define u as $u_1 \cdots u_p$, so that u is accepted by $\mathcal{A}[\mathbf{K}]$. This concludes since

$$\widehat{\varphi}(u) = \varphi(u_1) \cdots \varphi(u_p) = v. \qquad \square$$

Example 3.56. Recall that in our case, φ maps every word ab^i on the element $(a, i \bmod 2)$ of $C = \{(a,0),(a,1),0,1,0_C,1_C\}$.

Let $w \in (ab^*)^*$ and write $w = ab^{i_1} \cdots ab^{i_p}$. Then $\widehat{\varphi}(w)$ is the p-letter word over C whose j-th letter is $(a, i_j \bmod 2)$.

We can check that the language recognized by $\widehat{\mathcal{A}}$ is $(cd)^*$, where c (resp. d) is the letter $(a,0)$ (resp. $(a,1)$) of C.

In particular, observe that the image under $\widehat{\varphi}$ of the language recognized by $\mathcal{A}[\mathbf{K}]$ (namely $(a(bb)^*ab(bb)^*)^*$) is $(cd)^*$, the language recognized by $\widehat{\mathcal{A}}$.

Let L be the language recognized by $\mathcal{A}[\mathbf{K}]$. By Claim 3.55, we obtain that $\widehat{\varphi}(L)$ is recognized by $\widehat{\mathcal{A}}$, and in particular by its transition monoid M. By Claim 3.54 and Lemma 3.40, the monoid M has polynomial size with respect to $\widehat{\mathcal{A}}$ (and thus in $\mathcal{A}$). It remains to lift this result from $\widehat{\varphi}(L)$ to L itself. This is the goal of the following lemma.

Lemma 3.57. *Let φ be a monoid morphism from $(A \cup B)^* \to C$. Denote by $\widehat{\varphi}$ as the monoid morphism $(AB^*)^* \to C^*$ defined previously.*

Let $K \subset C^*$ be a language recognized by a morphism $\psi\colon C^* \to M$. Then we can construct a monoid recognizing $\widehat{\varphi}^{-1}(K)$ of polynomial size with respect to the size of M and C.

Before proving Lemma 3.57, we first show how it concludes the proof of Proposition 3.41. First observe that, due to Claim 3.55, we have $L = \widehat{\varphi}^{-1}(\widehat{\varphi}(L))$. We can then apply Lemma 3.57 with $K = \widehat{\varphi}(L)$. This ensures that L is recognized by a monoid of polynomial size in C and M. Due to Lemma 3.40, M is polynomial in the number of states in $\mathcal{A}$. Moreover, due to Lemma 3.50, C is polynomial in N, which concludes the proof.

We end this subsection by proving Lemma 3.57.

First note that, without loss of generality, we may assume that φ recognizes the language $\{\varepsilon\}$ and that $\psi(1_C) = 1_M$. Indeed, if it is not the case, we can replace M by $M \cup \{1'\}$, where for $s \in M$, we define $s1' = 1's = s$, and extend φ and ψ by $\varphi(\varepsilon) = \psi(1_C) = 1'$, so that $\{\varepsilon\} = \varphi^{-1}(1')$.

Observe also that any word $w \in (A \cup B)^*$ can be decomposed uniquely as $w = w_1w_2w_3$ where $w_1 \in B^*$, and either $w_2 = w_3 = \varepsilon$, or $w_2 \in (AB^*)^*$ and $w_3 \in AB^*$. The monoid structure we define follows this decomposition: we associate an element of C with each part w_1 and w_3. With the part $w_2 \in (AB^*)^*$, we associate an element of M. The base set of the monoid we construct is thus $C \times M \times C$. We define a monoid structure by taking:

$$(r, s, t)(r', s', t') = \begin{cases} (rr', s', t') & \text{if } (s, t) = (1_M, 1_C), \\ (r, s, tr') & \text{if } (s, t) \neq (1_M, 1_C) \text{ and } (s', t') = (1_M, 1_C), \\ (r, s\psi(t \cdot r')s', t') & \text{where } t \cdot r' \in C \text{ otherwise.} \end{cases}$$

Using that φ recognizes $\{\varepsilon\}$, one can check that this law is associative and that $(1_C, 1_M, 1_C)$ is a neutral element. We then define a monoid morphism $\chi : (A \cup B)^* \to C \times M \times C$ by taking $\chi(b) = (\varphi(b), 1_M, 1_C)$ for $b \in B$ and $\chi(a) = (1_C, 1_M, \varphi(a))$ for $a \in A$.

Note that the monoid we constructed has polynomial size with respect to C and M. It thus remains to prove that it recognizes the language $\widehat{\varphi}^{-1}(K)$. More precisely, we prove that its accepting set is

$$F = \{(1_C, s, t) \mid s\psi(t) \in \psi(K)\}.$$

This relies on the following claim, which is a consequence of the definition of χ.

Claim 3.58. *For any word $u \in (AB^*)^*$, we have $\chi(u) = (1_C, s, t)$ where $s\psi(t) = \psi(\widehat{\varphi}(u))$.*

Moreover, for every $u \in (A \cup B)^*$, the first component of $\chi(u)$ is the image under φ of the largest prefix of u in B^*.

Using this claim, we can prove that $\chi^{-1}(F) = \widehat{\varphi}^{-1}(K)$. We separate the proof in two parts, one for each inclusion.

- $\supset$ If $u \in \widehat{\varphi}^{-1}(K)$, then $u \in (AB^*)^*$. Hence, using the previous claim, we know that $\chi(u) = (1_C, s, t)$ where $s\psi(t) = \psi(\widehat{\varphi}(u))$. Therefore, $s\psi(t) = \psi(\widehat{\varphi}(u)) \in \psi(K)$, hence $\chi(u) \in F$.

- $\subset$ Conversely, take $u \in (A \cup B)^*$ such that $\chi(u) \in F$. By definition of F, we can write $\chi(u) = (1_C, s, t)$ with $s\psi(t) \in \psi(K)$.

  Define v as the greatest prefix of u in B^*. Using the previous lemma, we obtain that $\varphi(v) = 1_C$. Since φ recognizes $\{\varepsilon\}$, we have $v = \varepsilon$.

  Thus, u begins with an a, i.e. u lies in $(AB^*)^*$. By Claim 3.58, we have $s\psi(t) = \psi(\widehat{\varphi}(u))$. By definition of F, $\psi(\widehat{\varphi}(u))$ lies in $\psi(K)$. Since ψ recognizes the language K, we obtain that $\widehat{\varphi}(u) \in K$, hence $u \in \widehat{\varphi}^{-1}(K)$.

It remains to prove Claim 3.58. Using the first case in the definition of the multiplicative law on $C \times M \times C$, we have, for $b, b' \in B$:

$$\chi(bb') = \chi(b)\chi(b') = (\varphi(b), 1_M, 1_C)(\varphi(b'), 1_M, 1_C) = (\varphi(bb'), 1_M, 1_C)$$

By induction, we thus obtain that $\chi(w) = (\varphi(w), 1_M, 1_C)$ for every $w \in B^*$.

Let $a \in A$ and $w \in B^*$. We have

$$\chi(aw) = \chi(a)\chi(w) = (1_C, 1_M, \varphi(a))(\varphi(w), 1_M, 1_C)$$

Observe that $\varphi(a) \neq 1_C$, otherwise we have $a \in \varphi^{-1}(1_C)$, a contradiction since φ recognizes $\{\varepsilon\} = \varphi^{-1}(1_C)$. By the second multiplication rule, we thus obtain

$$\chi(aw) = (1_C, 1_M, \varphi(aw))$$

Finally, if $a, b \in A$ and $w, w' \in B^*$, we have by the third multiplication rule:

$$\chi(awbw') = (1_C, 1_M, \varphi(aw))(1_C, 1_M, \varphi(bw')) = (1_C, \psi(\varphi(aw)), \varphi(bw'))$$

where $\varphi(aw)$ is interpreted as a 1-letter word over the alphabet C. We thus have $\chi(awbw') = (1_C, \psi(\widehat{\varphi}(aw)), \varphi(bw'))$.

By induction, we thus obtain that if $u \in (AB^*)^*$ is decomposed as $u_1 \cdots u_p$ where each u_i lies in AB^*, we have

$$\chi(w) = (1_C, \psi(\widehat{\varphi}(u_1 \cdots u_{p-1})), \varphi(u_p))$$

In particular, observe that

$$\psi(\widehat{\varphi}(u_1 \cdots u_{p-1}))\psi(\varphi(u_p)) = \psi(\widehat{\varphi}(u_1 \cdots u_{p-1})\varphi(u_p)) = \psi(\widehat{\varphi}(u))$$

where $\varphi(u_p)$ is interpreted in the middle term as a letter of C. This proves the first part of the lemma.

For the second part, let $u \in (A \cup B)^*$ and write $u = vw$ where v is the largest prefix of u in B^*. In particular, note that $w \in (AB^*)^*$, hence $\chi(w) = (1_C, s, t)$ with $s\psi(t) = \psi(\widehat{\varphi}(w))$. We thus have

$$\chi(u) = \chi(v)\chi(w) = (\varphi(v), 1_M, 1_C)(1_C, s, t) = (\varphi(v), s, t)$$

which proves the second part of the lemma. This concludes the proof of Proposition 3.41.

We now end the construction on the running example.

Example 3.59. The two transitions of the automaton $\widehat{\mathcal{A}}$ have different labels. Denote by c, d the letters $(a, 0)$ and $(a, 1)$ of C. The language recognized by $\widehat{\mathcal{A}}$ is thus $(cd)^*$.

Therefore, we already computed the transition monoid M of $\widehat{\mathcal{A}}$ in Example 3.28: it is isomorphic to $\{\varepsilon, c, d, cd, dc, cc\}$ endowed with concatenation using the relations $cdc = c$, $dcd = d$ and $cc = dd$.

The output of the reduction given by Lemma 3.57 is then the following set of size 216:

$$\{(a, 0), (a, 1), 0, 1, 0_C, 1_C\} \times \{\varepsilon, c, d, cd, dc, cc\} \times \{(a, 0), (a, 1), 0, 1, 0_C, 1_C\}.$$

For every $w \in \{a,b\}^*$, write $w = b^{i_0}ab^{i_1}a\cdots ab^{i_p}$ and define

$$\chi(w) = \begin{cases} (1_C, \varepsilon, 1_C) \text{ if } w = \varepsilon \\ (i_0 \bmod 2, \varepsilon, 1_C) \text{ if } w \in B^+ \\ (1_C, \widehat{\varphi}(ab^{i_1}\cdots ab^{i_{p-1}}), (a, i_p \bmod 2)) \text{ if } w \in (AB^*)^+ \\ (i_0 \bmod 2, \widehat{\varphi}(ab^{i_1}\cdots ab^{i_{p-1}}), (a, i_p \bmod 2)) \text{ otherwise} \end{cases}$$

By Lemma 3.57, χ is a monoid morphism, which recognizes the language $(a(bb)^*ab(bb)^*)^*$ recognized by $\mathcal{A}[\mathbf{K}]$. The corresponding accepting set is

$$\{(1_C, c, (a,1)), (1_C, cd, 1_C), (1_C, \varepsilon, 1_C)\}.$$

3.3.5 Separability transfer

In this subsection, we prove Proposition 3.45. This is the last result we need to prove before concluding about Theorem 3.37. We recall here its statement.

Proposition 3.45. *Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be two automata over an alphabet A. Take $\mathbf{K} = (K_1, \ldots, K_n)$ a tuple of languages over an alphabet disjoint from A such that $\{K_1, \ldots, K_n\}$ is not* Bool($\mathcal{C}$)*-coverable.*

Then the languages recognized by $\mathcal{A}_1$ and $\mathcal{A}_2$ are $\mathcal{C}$-separable if and only if so are the languages recognized by $\mathcal{A}_1[\mathbf{K}]$ and $\mathcal{A}_2[\mathbf{K}]$.

Before proving Proposition 3.45, we illustrate that its hypotheses are needed, with the case $\mathcal{C} = \text{Bool}(\mathcal{C}) = \mathsf{FO}$. We first show that the non-coverability hypothesis is needed to transfer separability from the languages recognizes by the tagged automata to the initial languages.

Example 3.61. Take $\mathcal{A}_1$ and $\mathcal{A}_2$ to be the same as the minimal automaton of $a(aa)^*$. In particular, observe that $a(aa)^*$ is not FO-separable from itself. Assume that the tagging languages are $\mathbf{K} = (b, bb, bbb, bbbb)$, so that $\mathcal{A}_1[\mathbf{K}]$ and $\mathcal{A}_2[\mathbf{K}]$ are depicted on Figure 3.61. Then $\mathcal{A}_1[\mathbf{K}]$ recognizes $ab(abbab)^*$

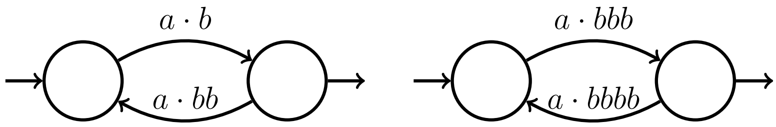


Figure 3.11 – Automata for Example 3.61

and $\mathcal{A}_2[\mathbf{K}]$ recognizes $abbb(abbbbabbb)^*$. These two languages are FO-separable since each word of $ab(abbab)^*$ is either of size 2 or has an a in third position, while each word of $abbb(abbbbabbb)^*$ has length at least 4 and contains a b in third position.

This illustrates that even if the tagging languages use a disjoint alphabet, the non-coverability is necessary for transferring non-separability along the tagging process.

The second example shows that the other hypothesis, namely that the tagging languages use a disjoint alphabet, is also needed. This time, it is used to transfer separability from the initial languages to the ones recognized by the tagged automata.

Example 3.62. Take $\mathcal{A}_1$ and $\mathcal{A}_2$ to be the minimal automata of $(ab)^*$ and $(ab)^*a$. These languages are FO-separable since $(ab)^*$ is FO-definable and $(ab)^* \cap (ab)^*a = \varnothing$. Assume that the tagging languages are $\mathbf{K} = (b(bb)^*, (bb)^*, b(bb)^*, (bb)^*)$, so that $\mathcal{A}_1[\mathbf{K}]$ and $\mathcal{A}_2[\mathbf{K}]$ are depicted on Figure 3.62. Then $\mathcal{A}_1[\mathbf{K}]$ recognizes

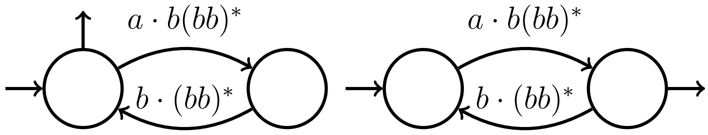


Figure 3.12 – Automata for Example 3.62

$(a(bb)^*)^*$ and $\mathcal{A}_2[\mathbf{K}]$ recognizes $(a(bb)^*)^*b$. These two languages are not FO-separable: they contain respectively $a(bb)^*b$ and $a(bb)^*$. As we already saw, $(bb)^*$ and $(bb)^*b$ are not FO-separable. With a similar argument, we can prove that this is also the case for $a(bb)^*b$ and $a(bb)^*$.

This illustrates that even if the non $\mathrm{Bool}(\mathcal{C})$-coverability hypothesis is fulfilled, we need to use a disjoint alphabet for the tagging languages. This time, this is necessary for transferring separability along the tagging process.

We now come back to the proof of Proposition 3.45. We thus take two automata $\mathcal{A}_1$ and $\mathcal{A}_2$ over an alphabet A. We also fix a tuple of languages $\mathbf{K} = (K_1, \ldots, K_n)$ over an alphabet B disjoint from A such that $\{K_1, \ldots, K_n\}$ is not $\mathrm{Bool}(\mathcal{C})$-coverable.

We show that the languages L_1 and L_2 recognized by $\mathcal{A}_1$ and $\mathcal{A}_2$ are $\mathcal{C}$-separable if and only if so are the languages L'_1 and L'_2 recognized by $\mathcal{A}_1[\mathbf{K}]$ and $\mathcal{A}_2[\mathbf{K}]$. We prove separately the two directions of this equivalence.

Separability and tagging

We first assume that L_1 is $\mathcal{C}$-separated from L_2 by a language Sep. We then use Sep to construct a $\mathcal{C}$-separator for L'_1 and L'_2. As illustrated with Example 3.62, this direction does not use that $\mathbf{K}$ is not $\mathcal{C}$-coverable, only the languages of $\mathbf{K}$ are contained in B^*, hence do not use any letter of A.

For $i = 1, 2$, recall that $\mathcal{A}_i$ and $\mathcal{A}_i[\mathbf{K}]$ only differ from the labels of their transitions: every transition $q \xrightarrow{aK_j} q'$ in $\mathcal{A}_i[\mathbf{K}]$ comes from a transition $q \xrightarrow{a} q'$

in $\mathcal{A}_i$. Since all the K_j's are languages over the alphabet B, erasing the letters in B from a word in L'_i gives a word in L_i. Therefore, if $\pi_{A\cup B,A}$ is the projection $(A \cup B)^* \to A^*$ we have $L'_i \subset \pi^{-1}_{A\cup B,A}(L_i)$.

It is then easy to see that $\pi^{-1}_{A\cup B,A}(\mathsf{Sep})$ separates L'_1 from L'_2. Indeed, since $L_1 \subset \mathsf{Sep}$ and $\mathsf{Sep} \cap \overline{L_2} = \varnothing$, we have:

$$L'_1 \subset \pi^{-1}_{A\cup B,A}(L_1) \subset \pi^{-1}_{A\cup B,A}(\mathsf{Sep})$$

and

$$L'_2 \cap \pi^{-1}_{A\cup B,A}(\mathsf{Sep}) \subset \pi^{-1}_{A\cup B,A}(L_2 \cap \mathsf{Sep}) = \varnothing.$$

Moreover, since $\mathcal{C}$ is a positive variety, it is closed under inverse morphism, hence $\pi^{-1}_{A\cup B,A}(\mathsf{Sep}) \in \mathcal{C}$. Therefore, L'_1 and L'_2 are $\mathcal{C}$-separable.

Non separability and tagging

We now prove the converse direction: we assume that $\mathsf{Sep}' \in \mathcal{C}$ separates L'_1 from L'_2 and we construct a language $\mathsf{Sep} \in \mathcal{C}$ separating L_1 from L_2. Similarly to the other direction, we construct Sep as an inverse image of Sep' by a suitable morphism. As illustrated with Example 3.61, this time the non-coverability hypothesis plays a crucial role: $\mathcal{C}$ must not be able to distinguish the languages in $\mathbf{K}$.

Let $\alpha : (A \cup B)^* \to M$ be the syntactic morphism of $\mathsf{Sep}' \in \mathcal{C}$. Since $\mathrm{Bool}(\mathcal{C})$ is a variety, using Proposition 3.32, all the languages recognized by α lie in $\mathrm{Bool}(\mathcal{C})$. Moreover, the union of all the $\alpha^{-1}(s)$ for $s \in M$ is $(A \cup B)^*$ and hence contains $K_1 \cup \cdots \cup K_n$. Since the set $\{K_1, \ldots, K_n\}$ is not $\mathrm{Bool}(\mathcal{C})$-coverable, there exists an element s of M such that $\alpha^{-1}(s)$ intersects every language K_i.

We thus obtain some words $w_1 \in K_1, \ldots, w_n \in K_n$ mapped on s by α:

$$\alpha(w_1) = \cdots = \alpha(w_n) = s$$

In some sense, these words are not distinguished by $\mathcal{C}$ and α. We may now define our separating language Sep as $\psi^{-1}(\mathsf{Sep}')$ where $\psi : A^* \to (A \cup B)^*$ is the morphism given by $\psi(a) = aw_1$ for any letter $a \in A$.

Again, since $\mathcal{C}$ is a positive variety, it is closed under inverse morphism, hence $\mathsf{Sep} \in \mathcal{C}$. It remains to show that Sep separates L_1 and L_2.

For $i = 1, 2$, consider the word

$$u = a_1 \cdots a_p \in L_i.$$

It labels an accepting path of $\mathcal{A}_i$. By construction of $\mathcal{A}_i[\mathbf{K}]$, there is a path in $\mathcal{A}_i[\mathbf{K}]$ labeled by a language $a_1 K_{i_1} \cdots a_p K_{i_p}$.

Hence, we have

$$v = a_1 w_{i_1} \cdots a_n w_{i_n} \in L'_i.$$

Moreover, we can check that:

$$\alpha(\psi(u)) = \alpha(a_1 w_1 \cdots a_p w_1) = \alpha(a_1 w_{i_1} \cdots a_p w_{i_p}) = \alpha(v).$$

Since α is the syntactic morphism of Sep', we obtain that

$$v \in \mathsf{Sep}' \text{ if and only if } \psi(u) \in \mathsf{Sep}' \text{ if and only if } u \in \mathsf{Sep}.$$

If $u \in L_1$, then $v \in L_1'$, hence $v \in \mathsf{Sep}'$ since $L_1' \subset \mathsf{Sep}'$. Therefore, $u \in \mathsf{Sep}$ and $L_1 \subset \mathsf{Sep}$.

Similarly, if $u \in L_2$, then $v \in L_2'$, hence $v \notin \mathsf{Sep}'$ since $\mathsf{Sep}' \cap L_2 = \varnothing$, so $u \notin \mathsf{Sep}$. Therefore $L_2 \cap \mathsf{Sep} = \varnothing$, which concludes the proof of Proposition 3.45.

3.4 Conclusion

In this chapter, we gave an historical presentation of the membership and the separation problems, which are the main problems considered in this chapter and in the next one. We investigated a complexity question about the separation problem. We proved that its complexity does not depend on the representation of the input languages. This result applies in many contexts, in particular for all the classes we introduced. It has then been adapted in a generalized setting in [Place and Zeitoun, 2018a].

The result presented here exhibits a major difference with the case of the membership problem: the complexity of separation comes from an additional amount of information that we need to compute, and not from the succinctness of the formalisms representing the inputs. This emphasizes that more information is needed to solve separation than to solve membership. In particular, this result also illustrates the robustness of the separation problem: its complexity behaves better than the one of the membership problem, for which Theorem 3.37 does not hold.

Despite its consequences on the robustness of the separation problem, Theorem 3.37 does not provide any complexity bound by itself: it only allows to transfer existing ones. Moreover, it does not help to obtain decidability results, since we do not need this theorem to effectively reduce the membership and separation problems on languages given by several formalisms. These decidability and complexity questions are actually the goal of the next chapter. We will describe there how to obtain some decidability results, as well as complexity lower bounds for the separation problem. In this setting, Theorem 3.37 will help anyway since it will allow us to only consider complexity of the separation problem when the languages are given by automata.

Chapter 4

The polynomial closure operation

Contents

This chapter is organized as follows. Section 4.2 is a presentation of the usual tools for the separation problem. The result of Section 4.3 was obtained with Thomas Place and Marc Zeitoun in 2017. A weaker version (with a lighter proof) was later included in [Place and Zeitoun, 2018a]. Section 4.4 is based on a result obtained with Thomas Place and Marc Zeitoun in 2015 ([Pierron et al., 2016]), which we extended this year to the more generic framework introduced in [Place and Zeitoun, 2017c].

4.1 Introduction

4.1.1 Generic hierarchies

In Chapter 3, we introduced several hierarchies. Among them, the dot-depth hierarchy and the Straubing-Thérien hierarchy were defined to stratify the star-free languages according to expressiveness criteria. Recall that while the initial level of both these hierarchies are different, they follow the same construction pattern.

Both are actually a special case of a generic construction studied in [Place and Zeitoun, 2017c]. Unless stated otherwise, all the results of this introduction come from this paper. Starting from a base class $\mathcal{C}$ (called the *basis*), one may define a concatenation hierarchy by mimicking the construction of the dot-depth hierarchy: level 0 is $\mathcal{C}$, and for every integer n,

- The *(half-)level* $n + \frac{1}{2}$ is the polynomial closure of level n.
- The *(full-)level* $n + 1$ is the Boolean closure of level $n + \frac{1}{2}$.

All of the results mentioned in Chapter 3 actually transpose to this more generic framework whenever the basis has nice properties. In particular, separation is decidable for levels up to $\frac{3}{2}$, and so is membership for to level $\frac{5}{2}$ in every hierarchy with a finite basis satisfying nice properties. Even the link with logic is generic: every concatenation hierarchy with a nice enough basis corresponds to the quantifier alternation hierarchy for FO enriched with some predicates depending on the basis. To state this link properly, we introduce the following predicates: given a language L,

- the binary predicate $I_L(i, j)$ is satisfied by all words $a_1 \cdots a_n$ such that the infix $a_{i+1} \cdots a_{j-1}$ lies in L.
- the unary predicate $P_L(i)$ is satisfied by all words $a_1 \cdots a_n$ such that the prefix $a_1 \cdots a_{i-1}$ lies in L.
- the unary predicate $S_L(i)$ is satisfied by all words $a_1 \cdots a_n$ such that the suffix $a_{i+1} \cdots a_n$ lies in L.

- the nullary predicate N_L is satisfied by all words in L (and not by the words outside L).

We denote by $\mathsf{FO}(\mathcal{C})$ the first-order logic using the predicates I_L, P_L, S_L and N_L for every $L \in \mathcal{C}$. Note that, in contrast to the signatures we defined for the Straubing-Thérien and the dot-depth hierarchies, observe that $\mathsf{FO}(\mathcal{C})$ may contain infinitely many predicates.

Similarly to the previous chapter, we can also define a generic quantifier alternation hierarchy: for every positive integer n, we define $\Sigma_n(\mathcal{C})$ as the fragment of $\mathsf{FO}(\mathcal{C})$ with either less than n blocks of quantifiers, or precisely n blocks of quantifiers, the first one being existential. As previously, its Boolean closure is denoted by $\mathcal{B}\Sigma_n(\mathcal{C})$.

When the class $\mathcal{C}$ is nice enough, the generic concatenation hierarchy with basis $\mathcal{C}$ coincides with the generic quantifier alternation hierarchy created using the predicates associated to $\mathcal{C}$. This is stated in the following result.

Theorem 4.1 ([Place and Zeitoun, 2017c]). *Let $\mathcal{C}$ be a class of languages closed under Boolean operations and quotient. Let n be an integer and L be a language.*

- *L lies in the half-level $n + \frac{1}{2}$ of the generic concatenation hierarchy based on $\mathcal{C}$ if and only if it is defined by a $\Sigma_{n+1}(\mathcal{C})$-formula.*

- *L lies in the full-level n of the generic concatenation hierarchy based on $\mathcal{C}$ if and only if it is defined by a $\mathcal{B}\Sigma_n(\mathcal{C})$-formula.*

This result generalizes the equivalence between the (usual) quantifier alternation hierarchy and the Straubing-Thérien hierarchy given by Theorem 3.4. Indeed, for the basis $\{\varnothing, A^*\}$ the additional predicates satisfy the following:

- $I_\varnothing, P_\varnothing, S_\varnothing$ and $N_\varnothing$ are always false.

- P_{A^*}, S_{A^*} and N_{A^*} are always true.

- I_{A^*} corresponds to the predicate $<$. Indeed, a word $a_1 \cdots a_n$ satisfies $I_{A^*}(i,j)$ when $a_{i+1} \cdots a_{j-1} \in A^*$. This is true as soon as this infix is a (possibly empty) word, i.e. when $j - 1 \geqslant i$. Therefore $I_{A^*}(i,j)$ holds whenever $i < j$.

In the same way, we may recover the link established in Theorem 3.3 between the dot-depth hierarchy and the quantifier alternation hierarchy with predicates $+1, \min, \max$. Indeed, the predicates obtained when taking $\mathcal{C} = \{\varnothing, \{\varepsilon\}, A^+, A^*\}$ satisfy:

- $I_{\{\varepsilon\}}$ is the successor predicate $+1$. Indeed, a word $w = a_1 \cdots a_n$ satisfies $I_{\{\varepsilon\}}(i,j)$ if and only if $a_{i+1} \cdots a_{j-1} = \varepsilon$. This means that $j = i + 1$.

- Using a similar argument, $P_{\{\varepsilon\}}$ and $S_{\{\varepsilon\}}$ corresponds respectively to min and max.
- $N_{\{\varepsilon\}}$ is the predicate ε.
- The additional predicates corresponding to A^+ can be expressed as Boolean combinations of $\varepsilon, +1, \min$ and $\max$.

These generic results thus emphasize how important are the Boolean and polynomial closure operations for studying the previously defined hierarchies. This is also illustrated for polynomial closure with the following result. It gives a generic extension of Theorem 3.7, which again motivates the study of separation for solving membership problems for higher levels of the hierarchies, even for generic ones.

Theorem 4.2 ([Place and Zeitoun, 2018b]). *If $\mathcal{C}$ is a nice enough class of regular languages such that $\mathcal{C}$-separation is decidable, then* $\mathrm{Pol}(\mathcal{C})$*-membership is also decidable.*

The case $\mathcal{C} = \mathsf{AT}$ (see Example 3.31) is a fundamental example. Indeed, AT is a finite class satisfying nice properties. As a consequence, the decidability results for finitely based generic concatenation hierarchies also hold when the basis is AT. In this case, the class $\mathrm{Pol}(\mathsf{AT})$, which is level $\frac{1}{2}$ of the AT-based hierarchy, is also level $\frac{3}{2}$ of the Straubing-Thérien hierarchy, as shown by the following result.

Theorem 4.3 ([Pin and Straubing, 1981]). *Let L be a regular language. The following are equivalent:*

- *L lies in* $\mathrm{Pol}(\mathsf{AT})$.
- *L is defined in* $\Sigma_1(\mathsf{AT})$.
- *L is defined in* $\Sigma_2(\{\varnothing, A^*\})$.

This theorem ensures that the AT-based hierarchy and the Straubing-Thérien are the same but with an offset. In particular, the generic results for the AT-based concatenation hierarchy thus yields decidability of separation for level $\frac{5}{2}$ and membership for level $\frac{7}{2}$ in the Straubing-Thérien hierarchy.

4.1.2 Properties of polynomial closures

The goal of this chapter is to focus on the separation problem for classes obtained as polynomial closures. We first recall the definition of this operation, as well as the historical results about it.

Definition 4.4. The *polynomial closure* $\mathrm{Pol}(\mathcal{C})$ of a class $\mathcal{C}$, is the smallest class containing $\mathcal{C}$ and closed under union and *marked concatenation*: if $K, L \in \mathrm{Pol}(\mathcal{C})$ and a is a letter, then $KaL \in \mathrm{Pol}(\mathcal{C})$.

It is easy to see that when the class $\mathcal{C}$ is closed under quotients, then so is $\mathrm{Pol}(\mathcal{C})$. This is due to the fact that the quotient operation behaves well with respect to union and marked concatenation. Indeed, we have, for all languages K, L and letters a, b:

$$a^{-1}(K \cup L) = a^{-1}K \cup a^{-1}L$$

$$a^{-1}(KbL) = \begin{cases} (a^{-1}K)bL & \text{if } \varepsilon \notin K \text{ or } a \neq b \\ (a^{-1}K)bL \cup L & \text{if } \varepsilon \in K \text{ and } a = b \end{cases}$$

Similar formulas hold for right quotients.

Assuming that $\mathcal{C}$ is also closed under inverse morphisms, we can lift this result to $\mathrm{Pol}(\mathcal{C})$, as shown by [Arfi, 1991], which gives an explicit description of the inverse image of $L \in \mathrm{Pol}(\mathcal{C})$ under a morphism φ using only quotients and inverse images of languages in $\mathcal{C}$. The same paper proves that when $\mathcal{C}$ is closed under quotient and Boolean operations, then $\mathrm{Pol}(\mathcal{C})$ is stable by intersection. This result was then improved in [Branco and Pin, 2009] by getting rid of the hypothesis that $\mathcal{C}$ is closed under complement, and in [Pin, 2013] by giving an explicit formula for the intersection of two languages in $\mathrm{Pol}(\mathcal{C})$. The following theorem summarizes these results.

Theorem 4.5. *Let $\mathcal{C}$ be a class of languages closed under quotients. Then* $\mathrm{Pol}(\mathcal{C})$ *is closed under quotients. Moreover,*

- *If $\mathcal{C}$ is closed under union and intersection, then so is* $\mathrm{Pol}(\mathcal{C})$.
- *If $\mathcal{C}$ is closed under inverse morphisms, then so is* $\mathrm{Pol}(\mathcal{C})$.

Note that this theorem proves that if $\mathcal{C}$ is a positive variety, then so is $\mathrm{Pol}(\mathcal{C})$. In particular, every half-level of any generic hierarchy based on some positive variety is a positive variety, and every full-level is a variety.

4.1.3 A variant of the polynomial closure

When considering closure under operations based on concatenation, a classic restriction asks for the concatenations to be unambiguous. A concatenation KL of two languages is said to be *unambiguous* if for every word $w \in KL$, there exists a unique pair $(u, v) \in K \times L$ such that $w = uv$. In other words, every word w in KL has a unique decomposition witnessing that $w \in KL$.

Note in particular that being an unambiguous concatenation is a semantic property of the concatenation KL and not only of the language $H = KL$. Indeed, the same language can be expressed as two different concatenations, one being unambiguous and not the other one. This is for example the case with $A^* \cdot aA^*$ and $A^*a \cdot (A \setminus \{a\})^*$, which denote the same language.

With this notion, we can define the *unambiguous polynomial closure* $\mathrm{UPol}(\mathcal{C})$ of a class $\mathcal{C}$, as the smallest class containing $\mathcal{C}$ and closed under:

- *disjoint* unions: if $K, L \in \mathrm{UPol}\,\mathcal{C}$ and $K \cap L = \varnothing$ then $K \cup L \in \mathrm{UPol}\,\mathcal{C}$,
- *unambiguous* marked concatenation: if $K, L \in \mathrm{UPol}\,\mathcal{C}$ and a is a letter, then $KaL \in \mathrm{UPol}\,\mathcal{C}$ when KaL is an unambiguous concatenation, i.e. when every word $w \in KaL$ has a unique decomposition as uav with $u \in K$ and $v \in L$.

This operation is not just another artificial construction. For example, when $\mathcal{C} = \mathsf{AT}$, the obtained class $\mathrm{UPol}(\mathsf{AT})$ is the well-studied class of *unambiguous languages*. It enjoys a wealth of equivalent characterizations, for example as the languages definable simultaneously by a Σ_2-sentence and by the negation of a Σ_2-sentence, or with a FO sentence using only two variables (see [Tesson and Thérien, 2002] for more characterizations).

Similarly to $\mathrm{Pol}(\mathcal{C})$, the class $\mathrm{UPol}(\mathcal{C})$ also satisfies some closure properties inherited from $\mathcal{C}$, as shown by the following result.

Theorem 4.6 ([Pin *et al.*, 1988]). *If $\mathcal{C}$ is a class of regular languages closed under Boolean operations and quotients, then so is* $\mathrm{UPol}(\mathcal{C})$*.*

Observe that, while it was not obvious from the definition, $\mathrm{UPol}(\mathcal{C})$ is closed under complement. This is a big difference from $\mathrm{Pol}(\mathcal{C})$, for which this property does not hold in most of the cases. This property can be better seen with the following characterization.

Theorem 4.7 ([Pin and Weil, 1995]). *If $\mathcal{C}$ is a class of regular languages closed under Boolean operations and quotients, then* $\mathrm{UPol}(\mathcal{C})$ *is the class of languages in* $\mathrm{Pol}(\mathcal{C})$ *whose complement also lies in* $\mathrm{Pol}(\mathcal{C})$*.*

Regarding membership and separation, the class $\mathrm{UPol}(\mathcal{C})$ satisfies some nice properties. The first one is generalization of Theorem 4.2: there is a generic reduction from the $\mathrm{UPol}(\mathcal{C})$-membership problem to $\mathcal{C}$-**membership**.

Theorem 4.8 ([Almeida *et al.*, 2015; Place and Zeitoun, 2018c]). *Let $\mathcal{C}$ is a class of regular languages closed under Boolean operations and quotients. If $\mathcal{C}$-membership is decidable, then so is* $\mathrm{UPol}(\mathcal{C})$*-membership.*

This result relies on the fact that if $\mathcal{C}$ is a nice enough class, then every language L in $\mathrm{UPol}(\mathcal{C})$ can be built using disjoint unions and unambiguous marked concatenations starting only from languages recognized by the syntactic morphism of L. In particular, we have $L \in \mathrm{UPol}(\mathcal{C})$ if and only if $L \in \mathrm{UPol}(\mathcal{D})$ where $\mathcal{D}$ is a finite class depending on L. This implies that $\mathrm{UPol}(\mathcal{C})$-membership reduces to $\mathrm{UPol}(\mathcal{D})$-membership for a *finite* class $\mathcal{D}$.

While this result is no longer valid for separation, we can nonetheless decide the UPol($\mathcal{C}$)-separation problem when $\mathcal{C}$ is finite, using an extension of the tools introduced in [Place and Zeitoun, 2017c] to decide the Pol($\mathcal{C}$)-separation problem in this setting. This is summarized in the following result.

Theorem 4.9 ([Place and Zeitoun, 2018c])**.** *If $\mathcal{C}$ is a finite class of regular languages closed under Boolean operations and quotients, then the* UPol($\mathcal{C}$)*-separation problem is decidable.*

4.1.4 Organization of the chapter

The goal of this chapter is to present decidability and complexity results about the Pol($\mathcal{C}$)-separation problem. Before presenting our contributions, we first introduce the tools and the algorithm of [Place and Zeitoun, 2017c] solving Pol($\mathcal{C}$)-separation when $\mathcal{C}$ is finite in Section 4.2. Even if finiteness of $\mathcal{C}$ is a strong hypothesis, the problem is already challenging. This can be illustrated in two ways. The first one relies on the situation for $\mathcal{C} = \mathsf{AT}$: recall that Pol($\mathsf{AT}$) is level $\frac{3}{2}$ of the Straubing-Thérien hierarchy by [Pin and Straubing, 1981]. In this case, the decidability of Pol(AT)-membership was established in 1991, i.e. more than 15 years after that the same problem was solved for piecewise-testable languages.

Another way to illustrate the importance of the case of a finite class $\mathcal{C}$ is given by the restricted operation UPol. Indeed, recall that we can reduce the UPol($\mathcal{C}$)-membership problem to the UPol($\mathcal{D}$)-membership problem for a finite class $\mathcal{D}$. Therefore, the problem is actually harder for finite classes.

(Variants of) the algorithm we present for Pol($\mathcal{C}$)-separation when $\mathcal{C}$ is finite have also been considered in [Place and Zeitoun, 2018a]. In particular, a PSpace upper bound is proved in the case $\mathcal{C} = \mathsf{AT}$. We prove in Section 4.3 a complementary result: when $\mathcal{C}$ is expressive enough, the Pol($\mathcal{C}$)-separation problem is PSpace-hard. This is expressed by the following theorem.

Theorem 4.10. *Let $\mathcal{C}$ be a positive variety of regular languages such that $\mathsf{AT} \subset \mathcal{C}$ and $\mathcal{C} \neq$ Pol($\mathcal{C}$). Then the* Pol($\mathcal{C}$)*-separation problem is PSpace-hard.*

In particular, this theorem applies for $\mathcal{C} = \mathsf{AT}$, which yields the exact complexity of Pol(AT)-separation.

Section 4.4 is devoted to the setting of infinite words. We extend the two types of results presented in Sections 4.2 and 4.3. The first result is a complexity result: it is a corollary of Theorem 4.10 when considering separation for languages of infinite words.

The second result is a decidability result. We generalize the decidability of separation for level $\frac{1}{2}$ of finitely-based hierarchies (obtained in Section 4.2) to the setting of infinite words. This is based on the decidability result of [Pierron *et al.*, 2016] for Pol(AT)-separation on infinite words. The result presented here actually extends the one of [Pierron *et al.*, 2016] since we consider Pol($\mathcal{C}$)-separation for a finite class $\mathcal{C}$ instead of Pol(AT)-separation. Nonetheless, we follow the same approach: the algorithm we present is based on its counterpart on finite words: we reuse some of the tools presented in Section 4.2. For instance, the algorithm for finite words is used as a black box.

4.2 The case of finite words

The goal of this section is to provide an overview of the generic framework and techniques we use to tackle decidability of the Pol($\mathcal{C}$)-separation problem when $\mathcal{C}$ is finite. We give a proof of Corollary 4.44 following the approach of [Place and Zeitoun, 2017d] (leading to Theorem 4.43). As we will see, this requires to introduce a wider framework. In particular, even if we only want to consider classes obtained as polynomial closures, most of the tools we present here actually apply to many classes (with some adjustments, as described in [Place and Zeitoun, 2017c]).

4.2.1 From separation to pairs

In this subsection, we give more details about the approach followed in [Place and Zeitoun, 2014a; Place, 2015; Pierron *et al.*, 2016; Place and Zeitoun, 2017d] to decide membership and separability for some classes. For this subsection, we thus fix a class $\mathcal{C}$ closed under union, intersection and quotients, and which contains $\varnothing$ and A^*.

We begin with an important but simple remark. Observe that the separation problem takes two regular languages as input. The analysis we describe here has to be applied to the recognizers of each language. To avoid duplicates, it is convenient to have a single object recognizing both of them. This is not restrictive: if L_1 is recognized by $\alpha_1 : A^* \to M_1$ and L_2 by $\alpha_2 : A^* \to M_2$, then L_1 and L_2 are both recognized by the morphism $\alpha : A^* \to M_1 \times M_2$ with $\alpha(w) = (\alpha_1(w), \alpha_2(w))$. Observe that α is computable from α_1, α_2.

The goal now becomes to determine whether two languages recognized by a given morphism $\alpha : A^* \to M$ are $\mathcal{C}$-separable. Observe that if we want to solve $\mathcal{C}$-separation in the generic case, we have to be able to test separability of every pair $\alpha^{-1}(s)$ and $\alpha^{-1}(t)$. This case is actually sufficient: determining which pairs $(s,t) \in M^2$ satisfy "$\alpha^{-1}(s)$ is $\mathcal{C}$-separable from $\alpha^{-1}(t)$" is sufficient to recover which pairs of languages recognized by α are $\mathcal{C}$-separable. This is shown by Lemma 4.11. Note that this is not really surprising since every language recognized by α can be written as union of some $\alpha^{-1}(s)$ for suitable values of s.

Lemma 4.11. *Let $\mathcal{C}$ be a class closed under union and intersection. Let α be a monoid morphism recognizing two languages L_1 and L_2. Then L_1 and*

L_2 are $\mathcal{C}$-separable if and only if $\alpha^{-1}(s_1)$ is $\mathcal{C}$-separable from $\alpha^{-1}(s_2)$ for every $s_1 \in \alpha(L_1)$ and $s_2 \in \alpha(L_2)$.

Proof. First assume that L_1 is $\mathcal{C}$-separable from L_2, and fix $s_1 \in \alpha(L_1), s_2 \in \alpha(L_2)$. Then $\alpha^{-1}(s_1) \subset L_1$ and $\alpha^{-1}(s_2) \subset L_2$. Therefore, any language separating L_1 from L_2 also separates $\alpha^{-1}(s_1)$ from $\alpha^{-1}(s_2)$.

Conversely, assume that $\alpha^{-1}(s_1)$ and $\alpha^{-1}(s_2)$ are separated by a language $K_{s_1,s_2} \in \mathcal{C}$ for every $s_1 \in \alpha(L_1)$ and $s_2 \in \alpha(L_2)$, that is:

$$\alpha^{-1}(s_1) \subset K_{s_1,s_2} \subset \overline{\alpha^{-1}(s_2)}.$$

We use these languages K_{s_1,s_2} to prove that there is a language of $\mathcal{C}$ separating L_1 from L_2. As a candidate for this separator, define:

$$L = \bigcup_{s_1 \in \alpha(L_1)} \bigcap_{s_2 \in \alpha(L_2)} K_{s_1,s_2}$$

By hypothesis, $\mathcal{C}$ is closed under union and intersection, hence we have indeed $L \in \mathcal{C}$.

It remains to prove that L separates L_1 from L_2. We first prove that $L_1 \subset L$. By construction, for every $s_1 \in \alpha(L_1), s_2 \in \alpha(L_2)$, we have $\alpha^{-1}(s_1) \subset K_{s_1,s_2}$, hence $\alpha^{-1}(s_1) \subset \bigcap_{s_2 \in \alpha(L_2)} K_{s_1,s_2}$.

Since L_1 is recognized by α, we have $L_1 = \bigcup_{s_1 \in \alpha(L_1)} \alpha^{-1}(s_1)$, hence $L_1 \subset L$.

Next, assume that there is a word $w \in L_2 \cap L$. Then by definition of L, there exists $s_1 \in \alpha(L_1)$ such that for all $s_2 \in \alpha(L_2)$, $w \in K_{s_1,s_2}$. In particular, taking $s_2 = \alpha(w) \in \alpha(L_2)$, we have $w \in K_{s_1,\alpha(w)}$. We thus obtain that $w \in K_{s_1,\alpha(w)} \cap \alpha^{-1}(\alpha(w))$, a contradiction since $K_{s_1,\alpha(w)} \cap \alpha^{-1}(\alpha(w)) = \varnothing$. Finally, we obtain that L is a language in $\mathcal{C}$ separating L_1 from L_2. □

Lemma 4.11 thus reduces $\mathcal{C}$-separation of two languages L_1, L_2 to the computation of all the pairs $(s,t) \in \alpha(L_1) \times \alpha(L_2)$ such that $\alpha^{-1}(s)$ is $\mathcal{C}$-separable from $\alpha^{-1}(t)$. This leads to the following definition.

Definition 4.12. Let $\alpha : A^* \to M$ be a monoid morphism. For any $(s,t) \in M^2$, the pair (s,t) is a *$\mathcal{C}$-pair for α* if $\alpha^{-1}(s)$ is **not** $\mathcal{C}$-separable from $\alpha^{-1}(t)$. We denote by $\mathbb{P}_{\mathcal{C}}[\alpha]$ the set of such pairs.

Example 4.13. In this example, we consider FO-pairs for the syntactic morphisms of the languages $(aa)^*$ and $(ab)^*$ (described in Example 3.14).

- Let $M = \mathbb{Z}/2\mathbb{Z}$ and $\alpha : a^* \to M$ defined by $\alpha(a) = 1$.

  The set $\mathbb{P}_{\mathsf{FO}}[\alpha]$ contains $(0,0)$ since $(aa)^*$ is not FO-separable from itself, and $(1,1)$ for the same reason. Moreover, the languages $(aa)^*$ and $a(aa)^*$ are not FO-separable, hence $(0,1)$ and $(1,0)$ are also FO-pairs for α. In this case, we thus have $\mathbb{P}_{\mathsf{FO}}[\alpha] = M^2$.

- Let $\alpha : \{a,b\}^* \to M$ be the syntactic monoid of $(ab)^*$, where $M = \{\varepsilon, a, b, ab, ba, aa\}$. Recall that this monoid is aperiodic. By Schützenberger's theorem, every language of the form $\alpha^{-1}(s)$ for $s \in M$ is FO-definable. In particular, $\alpha^{-1}(s)$ and $\alpha^{-1}(t)$ are FO-separable as soon as they are disjoint, i.e. $s \neq t$. Therefore, $\mathbb{P}_{\mathsf{FO}}[\alpha]$ contains only trivial pairs: it is $\{(s,s), s \in M\}$.

To illustrate the importance of pairs, we give an application for solving Pol($\mathcal{C}$)-membership. Theorem 4.2, proved in [Place and Zeitoun, 2014a, 2018b] reduces decidability of Pol($\mathcal{C}$)-membership to $\mathcal{C}$-separation. The underlying result is a characterization of Pol($\mathcal{C}$) parameterized by $\mathcal{C}$-pairs, given by the following theorem.

Theorem 4.14 ([Place and Zeitoun, 2018b])**.** *Let $\mathcal{C}$ be a class of regular languages closed under union, intersection and quotient. A language L lies in* Pol($\mathcal{C}$) *if and only if its syntactic morphism $\alpha : A^* \to (M, \leqslant_M)$ satisfies*

$$s^{\omega+1} \leqslant_M s^\omega t s^\omega$$

for every $\mathcal{C}$-pair (s,t) for α.

In view of this result, we obtain that solving $\mathcal{C}$-separation allows us to solve Pol($\mathcal{C}$)-membership. Indeed, recall that due to Lemma 4.11, computing $\mathcal{C}$-pairs is equivalent to solving $\mathcal{C}$-separation. Therefore, Theorem 4.14 implies Theorem 4.2, which states that Pol($\mathcal{C}$)-membership reduces to $\mathcal{C}$-separation.

When $\mathcal{C}$ is finite, observe that $\mathcal{C}$-separation (and thus computing $\mathcal{C}$-pairs) can be solved easily. Indeed, to test separability of two languages L_1, L_2 over an alphabet A, we test for every $L \in \mathcal{C}$ over the alphabet A whether L separates L_1 from L_2. Since there is a finite number of such languages L, this is decidable. Therefore, Theorem 4.14 gives a decidable characterization of Pol($\mathcal{C}$), hence Pol($\mathcal{C}$)-membership is always decidable when $\mathcal{C}$ is finite. In the rest of this section, we show how to lift this result to Pol($\mathcal{C}$)-separation.

Considering non-separability to define pairs is much more robust than separability. Indeed, as we will see, the set of pairs enjoys some nice closure properties, which would not be true for the set of non-pairs. This allows to use fixpoint algorithms for computing sets of pairs: we start from the set of trivial pairs (s,s) for $s \in M$, then compute more pairs using closure under several operations, until we reach a fixpoint. Among these operations, a generic one is the product. Indeed, due to the closure under quotients of $\mathcal{C}$, the set of pairs is a sub-monoid of M^2, as stated in Proposition 4.15.

Proposition 4.15 ([Place and Zeitoun, 2018b])**.** *Let $\mathcal{C}$ be a class of regular languages closed under intersection and quotients. Let $\alpha : A^* \to M$ be a monoid morphism. If (s,t) and (s',t') are $\mathcal{C}$-pairs, then (ss',tt') is also a $\mathcal{C}$-pair.*

This proposition is actually a consequence of the more generic result stating that if K is not $\mathcal{C}$-separable from K' and L is not $\mathcal{C}$-separable from L', then KK' is not $\mathcal{C}$-separable from LL'. Note that a similar statement is already proved for the covering problem in Lemma 3.48, and can then be translated to the separation problem when $\mathcal{C}$ is closed under complement using Lemma 3.43.

We first state a direct proof of Proposition 4.15 (as given in [Place and Zeitoun, 2018b]).

Proof of Proposition 4.15. By contrapositive, assume that (ss', tt') is not a $\mathcal{C}$-pair. By definition, this means that there is a language $K \in \mathcal{C}$ separating $\alpha^{-1}(ss')$ from $\alpha^{-1}(tt')$, i.e.

$$\alpha^{-1}(ss') \subset K \subset \overline{\alpha^{-1}(tt')}$$

We prove that either (s, t) or (s', t') is not a $\mathcal{C}$-pair. To this end, we construct the following candidate for a separator of $\alpha^{-1}(s)$ from $\alpha^{-1}(t)$:

$$H = \bigcap_{w \in \alpha^{-1}(s')} Kw^{-1}$$

By Myhill-Nerode's theorem, the intersection defining H is actually finite since K is regular. Therefore, since $\mathcal{C}$ is closed under quotients, we obtain that $H \in \mathcal{C}$.

By construction, since $\alpha^{-1}(ss') \subset K$, we have $\alpha^{-1}(s) \subset H$. Therefore, if $H \cap \alpha^{-1}(t) = \varnothing$, we get that (s, t) is not a $\mathcal{C}$-pair and the proof is over.

If this is not the case, there exists $u \in H \cap \alpha^{-1}(t)$. We use u to construct a language G separating $\alpha^{-1}(s')$ from $\alpha^{-1}(t')$. This concludes the proof: (s', t') is not a $\mathcal{C}$-pair. Define $G = u^{-1}K$, which lies in $\mathcal{C}$ since $\mathcal{C}$ is closed under quotients. We have to prove that $\alpha^{-1}(s') \subset G$ and $G \cap \alpha^{-1}(t') = \varnothing$.

Let $w \in \alpha^{-1}(s')$. Then, by definition of H, we have $u \in Kw^{-1}$, hence $uw \in K$ and $w \in u^{-1}K$, so $w \in G$. Therefore

$$\alpha^{-1}(s') \subset G.$$

Next, assume that there exists $v \in \alpha^{-1}(t') \cap G$. By definition of G, we have $uv \in K$, hence $uv \in K \cap \alpha^{-1}(tt')$, a contradiction since $K \cap \alpha^{-1}(tt') = \varnothing$. Finally, we have

$$\alpha^{-1}(t') \cap G = \varnothing,$$

hence (t, t') is not a $\mathcal{C}$-pair, which concludes the proof. $\square$

In the following, we will present an alternative proof in the special case of classes obtained as polynomial closures. This proof relies on an equivalent definition of Pol($\mathcal{C}$)-pairs, which will also be helpful for finding other closure operations. We thus devote the two next subsections to proving Proposition 4.15

when considering the class $\mathrm{Pol}(\mathcal{C})$ with $\mathcal{C}$ finite. The proof is based on two ingredients. We first prove that Proposition 4.15 holds for finite classes (closed under intersection and quotients). Then, we apply this result to special classes built on top of $\mathcal{C}$ to deduce Proposition 4.15 for $\mathrm{Pol}(\mathcal{C})$ when $\mathcal{C}$ is finite.

For the rest of this section, we thus fix a finite class $\mathcal{C}$ of regular languages, closed under union, intersection and quotients. We also fix a monoid morphism $\alpha : A^* \to M$ where M is finite.

4.2.2 First step for Proposition 4.15: the case of finite classes

In this subsection, we give an alternative proof of Proposition 4.15 in the special case when $\mathcal{C}$ is a finite class. Note that, in this case, $\mathcal{C}$-separation can be solved directly, without using the formalism of pairs. Indeed, we can just test for each language whether it is a separator. However, the tools we present here will help to tackle infinite classes such as $\mathrm{Pol}(\mathcal{C})$. We begin by introducing the following tool.

Definition 4.16. Given $w, w' \in A^*$, we write $w \leqslant_{\mathcal{C}} w'$ if every language $K \in \mathcal{C}$ containing w also contains w'. We also write $w \sim_{\mathcal{C}} w'$ when $w \leqslant_{\mathcal{C}} w'$ and $w' \leqslant_{\mathcal{C}} w$.

Example 4.17. When $\mathcal{C} = \mathsf{AT}$, the relations $\leqslant_{\mathcal{C}}$ and $\sim_{\mathcal{C}}$ coincide since AT is closed under complement.

Indeed, if two words u, v satisfy $u \leqslant_{\mathsf{AT}} v$, consider $L \in \mathsf{AT}$ containing v and assume that it does not contain u. Then $\overline{L}$ contains u and is still a language of AT (since AT is closed under complement). This implies that $v \in \overline{L}$, a contradiction. We thus obtain that $u \in L$, hence $v \leqslant_{\mathsf{AT}} u$. Therefore, the relations $\leqslant_{\mathsf{AT}}$ and $\sim_{\mathsf{AT}}$ coincide.

Moreover, if $u \sim_{\mathsf{AT}} v$, then for every letter a appearing in u, we have $u \in A^*aA^*$. Since $A^*aA^* \in \mathsf{AT}$, we have $v \in A^*aA^*$, hence a appears in v. By exchanging u and v, we obtain that for all words $u, v \in A^*$, we have $u \sim_{\mathsf{AT}} v$ if and only if u and v have the same alphabet, i.e. $\mathrm{alph}(u) = \mathrm{alph}(v)$.

It is easy to verify that $\leqslant_{\mathcal{C}}$ is reflexive and transitive (it is a quasi-order), and that $\sim_{\mathcal{C}}$ is an equivalence relation. The link between this quasi-order and the separation problem is given by the following lemma.

Lemma 4.18. *Let $\mathcal{C}$ be a finite class of regular languages, closed under union and intersection. Let L_1 and L_2 be two languages. Then L_1 is not $\mathcal{C}$-separable from L_2 if and only if there exist $u_1 \in L_1$ and $u_2 \in L_2$ such that $u_1 \leqslant_{\mathcal{C}} u_2$.*

Proof. Assume that there exist $u_1 \in L_1$ and $u_2 \in L_2$ such that $u_1 \leqslant_{\mathcal{C}} u_2$. Then any language $L \in \mathcal{C}$ containing L_1 contains u_1, hence u_2 by definition of $\leqslant_{\mathcal{C}}$. Therefore, $L \cap L_2$ contains u_2, preventing L from separating L_1 from L_2.

Conversely, assume that L_1 is not $\mathcal{C}$-separable from L_2. We consider the language

$$L = \bigcup_{u \in L_1} \bigcap_{K \in \mathcal{C}, u \in K} K$$

Since $\mathcal{C}$ is finite and closed under union and intersection, we have $L \in \mathcal{C}$.

Observe that for every $u \in L_1$, we have $u \in \bigcap_{K \in \mathcal{C}, u \in K} K$, hence $u \in L$ and $L_1 \subset L$. Since $L \in \mathcal{C}$, it does not separate L_1 from L_2, therefore, there exists $u_2 \in L_2 \cap L$. Then, by definition of L, there exists $u_1 \in L_1$ such that every $K \in \mathcal{C}$ containing u_1 also contains u_2. This ensures that $u_1 \leqslant_{\mathcal{C}} u_2$. □

Applying Lemma 4.18 to $L_1 = \alpha^{-1}(s)$ and $L_2 = \alpha^{-1}(t)$ gives the following alternative definition of pairs.

Definition 4.19. A pair $(s, t) \in M^2$ is a $\mathcal{C}$-pair for α if and only if there exists $u \in \alpha^{-1}(s)$ and $v \in \alpha^{-1}(t)$ such that $u \leqslant_{\mathcal{C}} v$. The words u, v are called *witnesses* of the pair (s, t).

Observe that, even if the relation $\leqslant_{\mathcal{C}}$ is computable on A^* (since $\mathcal{C}$ is finite), the criterion given by Lemma 4.18 involves a quantification over infinitely many words, thus does not give directly a decidable criterion for separability. However, it turns out that, in order to solve separation, it is sufficient to compute the relation $\leqslant_{\mathcal{C}}$ only for a set of representative for the syntactic congruences of the languages we want to separate. This is another motivation to study pairs: this reformulation states that $\mathcal{C}$-pairs are an abstraction of the relation $\leqslant_{\mathcal{C}}$ over a finite monoid, which encapsulates the information needed to solve separation.

This equivalent definition is adapted to prove closure operations on the set of pairs. Indeed, Proposition 4.15 is now a consequence of the fact that $\leqslant_{\mathcal{C}}$ is compatible with concatenations, as shown by Lemma 4.20.

Lemma 4.20. *Let $\mathcal{C}$ be a finite class of regular languages, closed under quotients. Let u, u', v, v' be four words such that $u \leqslant_{\mathcal{C}} u'$ and $v \leqslant_{\mathcal{C}} v'$. Then $uv \leqslant_{\mathcal{C}} u'v'$.*

Note that Lemma 4.20 also applies when $\mathcal{C}$ is infinite. Before proving this lemma, let us show how to use it to conclude about Proposition 4.15 when $\mathcal{C}$ is finite. We will then extend it to $\text{Pol}(\mathcal{C})$ in the following subsection. Assume that (s, t) and (s', t') are $\mathcal{C}$-pairs for α. Then by definition, we can find witnesses for these pairs, i.e. some words u, v, u', v' mapped on s, t, s', t' by α, and such that $u \leqslant_{\mathcal{C}} v$ and $u' \leqslant_{\mathcal{C}} v'$. By Lemma 4.20, we have $uu' \leqslant_{\mathcal{C}} vv'$. Therefore, the words uu' and vv' are witnesses of (ss', tt'), which is thus a $\mathcal{C}$-pair.

We end this subsection with the proof of Lemma 4.20. As we will see, we only use closure under quotients, and not closure under union and intersection.

Proof of Lemma 4.20. Let u, u', v, v' be four words satisfying $u \leqslant_{\mathcal{C}} u'$ and $v \leqslant_{\mathcal{C}} v'$. Consider $K \in \mathcal{C}$ containing uv. We want to show that K contains $u'v'$.

By definition, $u \in Kv^{-1}$ and $Kv^{-1} \in \mathcal{C}$ since $\mathcal{C}$ is closed under quotients. Hence since $u \leqslant_{\mathcal{C}} u'$, we have $u' \in Kv^{-1}$ so $u'v \in K$.

This implies that $v \in u'^{-1}K$. Again, since $u'^{-1}K \in \mathcal{C}$ and $v \leqslant_{\mathcal{C}} v'$, we have $v' \in u'^{-1}K$ hence $u'v' \in K$.

We thus obtain that $uv \leqslant_{\mathcal{C}} u'v'$, which concludes the proof of Lemma 4.20. □

4.2.3 Second step for Proposition 4.15: stratifying Pol($\mathcal{C}$)

Recall that our goal is to decide the separation problem for Pol($\mathcal{C}$) when $\mathcal{C}$ is a finite class closed under union, intersection and quotients. Since Pol($\mathcal{C}$) is infinite, there is no guarantee that Lemma 4.18 is preserved. In particular, the alternative definition of pairs given by definition 4.19 may not extend in this case.

To obtain results about Pol($\mathcal{C}$)-pairs, the approach is to decompose Pol($\mathcal{C}$) as an increasing union of finite classes (we say that we *stratify* Pol($\mathcal{C}$)), and then lift results from these smaller classes to Pol($\mathcal{C}$) itself. Note that these smaller classes have to satisfy some nice closure properties, to make sure that the previous results such as Lemma 4.20 still apply.

Remark 4.21. This approach is similar but different from the one of [Place and Zeitoun, 2017c], where a more generic result is presented, allowing to not explicitly give a decomposition of Pol($\mathcal{C}$). It states that for every finite subclass $\mathcal{C}'$ of a class $\mathcal{C}$ closed under union, intersection and quotients, there exists a finite class $\mathcal{D}$ also closed under union, intersection and quotients such that $\mathcal{C}' \subset \mathcal{D} \subset \mathcal{C}$.

This is valid even (and especially) when $\mathcal{C}$ is infinite. This allows (for example) to remove the finiteness hypothesis on $\mathcal{C}$ in Theorem 4.14, i.e. to characterize Pol($\mathcal{C}$) using $\mathcal{C}$-pairs even when $\mathcal{C}$ is infinite.

Recall that we fixed a finite class $\mathcal{C}$ closed under union, intersection and quotients, as well as a morphism $\alpha : A^* \to M$ where M is a finite monoid. We now present a possible stratification of the class Pol($\mathcal{C}$). The construction relies on the following idea.

Intuitively, even if the class Pol($\mathcal{C}$) is infinite, each language of Pol($\mathcal{C}$) is constructed as a finite union of a finite number of marked concatenations of languages in $\mathcal{C}$. Therefore, we can stratify the class Pol($\mathcal{C}$) in such a way that each stratum is a finite class satisfying nice properties. Intuitively, a language lies in the k-th stratum if it can be built from $\mathcal{C}$ using a bounded number of marked concatenations depending on k.

Given an integer k, we define $\text{Pol}_k(\mathcal{C})$ as follows:

- If $k = 0$, then $\text{Pol}_0(\mathcal{C}) = \mathcal{C}$.

- If $k > 0$, then $\mathrm{Pol}_k(\mathcal{C})$ is the smallest class closed under union and intersection such that
  - $\mathrm{Pol}_{k-1}(\mathcal{C}) \subset \mathrm{Pol}_k(\mathcal{C})$.
  - for every $K, L \in \mathrm{Pol}_{k-1}(\mathcal{C})$ and $a \in A$, $KaL \in \mathrm{Pol}_k(\mathcal{C})$.

For the sake of readability, we write $\leqslant_k$ instead of $\leqslant_{\mathrm{Pol}_k(\mathcal{C})}$, k-pairs instead of $\mathrm{Pol}_k(\mathcal{C})$-pairs and $\mathbb{P}_k[\alpha]$ instead of $\mathbb{P}_{\mathrm{Pol}_k(\mathcal{C})}[\alpha]$.

It is easy to check that the classes $\mathrm{Pol}_k(\mathcal{C})$ are all finite. Moreover, as shown by the following result, they are also closed under quotients, hence the results of the previous subsection hold for them. In particular, the alternative definition of k-pairs using $\leqslant_k$ holds.

Lemma 4.22. *For every integer k, the class $\mathrm{Pol}_k(\mathcal{C})$ is closed under quotients.*

Proof. We use induction on k. If $k = 0$, then $\mathrm{Pol}_k(\mathcal{C}) = \mathcal{C}$, which is closed under quotients.

Assume that $k > 0$ and take $L \in \mathrm{Pol}_k(\mathcal{C})$. To prove closure under quotients, it is sufficient to prove that $a^{-1}L$ and La^{-1} lie in $\mathrm{Pol}_k(\mathcal{C})$ for every letter a. By symmetry, we only consider the left quotient $a^{-1}L$.

Note that the quotient operation commutes with union and intersection, hence it is sufficient to consider the two following cases:

- $L \in \mathrm{Pol}_{k-1}(\mathcal{C})$
- L is a marked concatenation of languages in $\mathrm{Pol}_{k-1}(\mathcal{C})$, i.e. $L = K_1 b K_2$ where b is a letter and $K_1, K_2 \in \mathrm{Pol}_{k-1}(\mathcal{C})$.

In the former, the induction hypothesis ensures that $a^{-1}L \in \mathrm{Pol}_{k-1}(\mathcal{C})$, hence $a^{-1}L \in \mathrm{Pol}_k(\mathcal{C})$.

In the latter, recall that

$$a^{-1}L = \begin{cases} (a^{-1}K_1)bK_2 & \text{if } \varepsilon \notin K_1 \text{ or } a \neq b \\ (a^{-1}K_1)bK_2 \cup K_2 & \text{if } \varepsilon \in K_1 \text{ and } a = b \end{cases}$$

Using again the induction hypothesis, we have $a^{-1}K_1 \in \mathrm{Pol}_{k-1}(\mathcal{C})$, hence $a^{-1}L$ can be written as a (union of) marked concatenation of languages in $\mathrm{Pol}_{k-1}(\mathcal{C})$. Therefore, $a^{-1}L \in \mathrm{Pol}_k(\mathcal{C})$, which concludes the proof. $\square$

This result ensures that the strata are nice classes. In order to use this stratification, we still have to show how to recover results for $\mathrm{Pol}(\mathcal{C})$ using the results about $\mathrm{Pol}_k(\mathcal{C})$. We illustrate this by proving Proposition 4.15 for the class $\mathrm{Pol}(\mathcal{C})$. To this end, we link the set of pairs for these different classes. The following lemma states that we can approximate the set of $\mathrm{Pol}(\mathcal{C})$-pairs using $\mathrm{Pol}_k(\mathcal{C})$-pairs.

Lemma 4.23. *Let $s, t \in M$. We have:*

- *for every integer k, if (s,t) is a $(k+1)$-pair then it is a k-pair: $\mathbb{P}_{k+1}[\alpha] \subset \mathbb{P}_k[\alpha]$.*
- *(s,t) is a $\mathrm{Pol}(\mathcal{C})$-pair if and only if (s,t) is a k-pair for every integer k: we have $\mathbb{P}_{\mathrm{Pol}(\mathcal{C})}[\alpha] = \bigcap_{k\geqslant 0} \mathbb{P}_k[\alpha]$.*

Proof.

- Let (s,t) be a $(k+1)$-pair for α. Then $\alpha^{-1}(s)$ is not $\mathrm{Pol}_{k+1}(\mathcal{C})$-separable from $\alpha^{-1}(t)$. Since $\mathrm{Pol}_k(\mathcal{C}) \subset \mathrm{Pol}_{k+1}(\mathcal{C})$, we obtain that $\alpha^{-1}(s)$ is not $\mathrm{Pol}_k(\mathcal{C})$-separable from $\alpha^{-1}(t)$, i.e. (s,t) is a k-pair.
- Using that $\mathrm{Pol}_k(\mathcal{C}) \subset \mathrm{Pol}(\mathcal{C})$ for every integer k, we can also obtain that every $\mathrm{Pol}(\mathcal{C})$-pair is a k-pair.

  Conversely, let (s,t) be a k-pair for every integer k. Assume that there is $L \in \mathrm{Pol}(\mathcal{C})$ separating $\alpha^{-1}(s)$ from $\alpha^{-1}(t)$. Then there exists an integer k such that $L \in \mathrm{Pol}_k(\mathcal{C})$. We thus obtain that $\alpha^{-1}(s)$ and $\alpha^{-1}(t)$ are $\mathrm{Pol}_k(\mathcal{C})$-separable, a contradiction since (s,t) is a $\mathrm{Pol}_k(\mathcal{C})$-pair. ☐

This lemma implies that the set of k-pairs gets refined when k increases, and that the limit object is the set $\mathbb{P}_{\mathrm{Pol}(\mathcal{C})}[\alpha]$ we want to compute. When the monoid M is finite, the sequence $(\mathbb{P}_k[\alpha])_{k\in\mathbb{N}}$ cannot be endlessly refined. We summarize this in the following result.

Lemma 4.24. *There exists an integer ℓ (depending on α) such that $\mathbb{P}_\ell[\alpha] = \mathbb{P}_{\mathrm{Pol}(\mathcal{C})}[\alpha]$.*

While such an integer ℓ is guaranteed to exist, finding an effective bound on ℓ (depending on α) is still a difficult problem, since it is equivalent to computing the set of $\mathrm{Pol}(\mathcal{C})$-pairs, and thus to solving $\mathrm{Pol}(\mathcal{C})$-separation.

Note that due to Lemma 4.24, the set of $\mathrm{Pol}(\mathcal{C})$-pairs for a morphism α equals the set of ℓ-pairs for α. In particular, it is a submonoid of M^2, which proves Proposition 4.15 for the class $\mathrm{Pol}(\mathcal{C})$.

The methodology described in this subsection to prove Proposition 4.15 can (and will) be applied to find some other closure properties of the set of $\mathrm{Pol}(\mathcal{C})$-pairs. To this end, we summarize the last two results with the following lemma. It gives an alternative definition of $\mathrm{Pol}(\mathcal{C})$-pairs, similar to Definition 4.19. It is obtained by combining Lemmas 4.23 and 4.24 with Definition 4.19 (instantiated for the finite class $\mathrm{Pol}_k(\mathcal{C})$).

Lemma 4.25. *There exists an integer ℓ such that for every $(s,t) \in M^2$, the following are equivalent:*

- *(s,t) is a $\mathrm{Pol}(\mathcal{C})$-pair for α.*

- *For every $k \geqslant 0$, there exists $u_k \in \alpha^{-1}(s)$ and $v_k \in \alpha^{-1}(t)$ such that $u_k \leqslant_k v_k$.*
- *There exists $u_\ell \in \alpha^{-1}(s)$ and $v_\ell \in \alpha^{-1}(t)$ such that $u_\ell \leqslant_\ell v_\ell$.*

4.2.4 Specific closure operations

In the two previous subsections, we proved that the set of $\mathrm{Pol}(\mathcal{C})$-pairs forms a monoid, using a stratification specific to $\mathrm{Pol}(\mathcal{C})$. This method actually applies to less specific classes, and can be used to prove Proposition 4.15, by constructing a generic stratification for every (nice enough) class of regular languages.

Observe that we never use any argument specific to the expressiveness of the class $\mathrm{Pol}(\mathcal{C})$ in any of the two proofs of Proposition 4.15 we presented. Indeed, we did not use any semantical properties of the classes $\mathrm{Pol}_k(\mathcal{C})$: the results rely only on the inclusions between these classes and $\mathrm{Pol}(\mathcal{C})$. In particular, there is no hope to find specific properties of $\mathrm{Pol}(\mathcal{C})$-pairs using such generic arguments: we still miss some closure operations that are specific to $\mathcal{C}$ and $\mathrm{Pol}(\mathcal{C})$. To find such operations, we reuse the framework of the two last subsections. This means that we first find some properties of the quasi-orders $\leqslant_k$ for $k \geqslant 0$, and then lift them to $\mathrm{Pol}(\mathcal{C})$ using the alternative characterization of $\mathrm{Pol}(\mathcal{C})$-pairs given in Lemma 4.25.

Therefore, we first consider the case of a finite class $\mathcal{C}$. We begin with a first example of non trivial words comparable for $\leqslant_\mathcal{C}$ (i.e. a first example of non trivial $\mathcal{C}$-pairs).

Lemma 4.26 ([Place and Zeitoun, 2018b])**.** *If $\mathcal{C}$ is finite, there exists a natural number $p \geqslant 1$, such that for any word $u \in A^*$ and any integers $m, m' \geqslant 1$, we have $u^{pm} \sim_\mathcal{C} u^{pm'}$. The smallest such integer p is called the* period *of $\mathcal{C}$.*

Before giving the proof of this lemma, we state a remark about its consequences in terms of $\mathcal{C}$-pairs.

Remark 4.27. This result is exactly the kind of results we look for. Indeed, it implies that for every $s \in M$, $(s^\omega, s^{\omega+p})$ is a $\mathcal{C}$-pair for α, where $\omega = \omega(M)$. To see this, consider a word $u \in \alpha^{-1}(s)$, and observe that

$$\begin{aligned} \alpha(u^{p\omega}) &= s^{p\omega} = s^\omega \\ \alpha(u^{p\omega+p}) &= s^{p\omega} s^p = s^{\omega+p} \end{aligned}$$

Moreover, Lemma 4.26 gives that $u^{p\omega} \sim_\mathcal{C} u^{p(\omega+1)}$, ensuring that $u^{p\omega}$ and $u^{p\omega+p}$ are witnesses for the $\mathcal{C}$-pair $(s^\omega, s^{\omega+p})$.

Proof of Lemma 4.26. Consider the set $A^*/\sim_\mathcal{C}$. Due to Lemma 4.20, $\sim_\mathcal{C}$ is compatible with concatenation, hence $A^*/\sim_\mathcal{C}$ is a monoid.

Moreover, we can check that it is finite. Indeed, for all words $u, v \in A^*$, we have $u \sim_{\mathcal{C}} v$ if and only if u and v lie in exactly the same languages of $\mathcal{C}$. Therefore, there are at most $2^{|\mathcal{C}|}$ classes for $\sim_{\mathcal{C}}$, corresponding to all the possible subsets of $\mathcal{C}$. Since $\mathcal{C}$ is finite, we obtain that $A^*/\sim_{\mathcal{C}}$ is also finite.

We define p as the integer $\omega(A^*/\sim_{\mathcal{C}})$ obtained by applying Proposition 3.15 to the finite monoid $A^*/\sim_{\mathcal{C}}$. We thus have $u^{2p} \sim_{\mathcal{C}} u^p$ for every word $u \in A^*$. This concludes the proof of Lemma 4.26. □

Observe that the proof of this result relies crucially on the finiteness of $\mathcal{C}$. In particular, the integer p depends on the class $\mathcal{C}$. Therefore, even using stratifications, we cannot extend this result directly to find $\mathrm{Pol}(\mathcal{C})$-pairs. Indeed, applying Lemma 4.26 to each class $\mathrm{Pol}_k(\mathcal{C})$, we obtain a sequence of periods $(p_k)_k$ such that for all integers k, m, m' and every word $u \in A^*$, we have $u^{p_k m} \sim_k u^{p_k m'}$. This translates in the setting of pairs as $(s^\omega, s^{\omega+p_k})$ is a k-pair for every s and k. However, this pair depends on p_k, hence it does not directly yield a $\mathrm{Pol}(\mathcal{C})$-pair. Nonetheless, the period of $\mathcal{C}$ will be fundamental for stating results about $\mathrm{Pol}(\mathcal{C})$.

In order to state our next result on $\mathrm{Pol}(\mathcal{C})$-pairs, we need to investigate the structure of $\mathrm{Pol}(\mathcal{C})$, and its stratification. Besides finiteness and closure under quotient of the strata, we can also find a recurrence relation on the quasi-orders $\leqslant_k$ stated below.

Lemma 4.28 ([Place and Zeitoun, 2017c])**.** *Let k be an integer and w and w' two words. Then $w \leqslant_k w'$ if and only if the two following properties hold:*

- $w \leqslant_{\mathcal{C}} w'$
- *If $k > 0$, for every decomposition $w = uav$ with $a \in A$, we can decompose $w' = u'av'$ in such a way that $u \leqslant_{k-1} u'$ and $v \leqslant_{k-1} v'$.*

The proof of this result relies on the special structure of $\mathrm{Pol}_k(\mathcal{C})$. Indeed, one can witness that a word w lies in a marked concatenation using the letter a by providing a decomposition uav of w. Since $\mathrm{Pol}_k(\mathcal{C})$ is (up to closure under union and intersection) the set of marked concatenations of languages in $\mathrm{Pol}_{k-1}(\mathcal{C})$, it is not surprising that a $\mathrm{Pol}_k(\mathcal{C})$ condition on w translates as $\mathrm{Pol}_{k-1}(\mathcal{C})$ conditions on u and v.

With the recurrence relation given by Lemma 4.28, we can state two last properties of the quasi-order $\leqslant_k$ we need. These are generic examples of non-trivial elements that are comparable for $\leqslant_k$ (and thus of non-trivial k-pairs). These results are proven inductively using Lemma 4.28.

Lemma 4.29 ([Place and Zeitoun, 2017c])**.** *Let p be the period of $\mathcal{C}$, k be an integer and let $u \in A^*$. Then for every m, n at least equal to $2^{k+1}-1$, we have*

$$u^{pm} \leqslant_k u^{pn}.$$

Remark 4.30. We can compare this result to the one obtained by instantiating Lemma 4.26 for the class $\mathrm{Pol}_k(\mathcal{C})$, as done in Remark 4.27. This time, the two witnesses u^{pm} and u^{pn} depend only on the period of $\mathcal{C}$ but not on the period of $\mathrm{Pol}_k(\mathcal{C})$. In particular, we obtain that $(s^\omega, s^{\omega+p})$ is a k-pair for every integer k, i.e. a $\mathrm{Pol}(\mathcal{C})$-pair. Indeed, for every integer k and every word $u \in \alpha^{-1}(s)$, we have:

$$\begin{aligned}\alpha(u^{p\omega 2^{k+1}}) &= s^{p\omega 2^{k+1}} = s^\omega\\ \alpha(u^{p\omega 2^{k+1}+p}) &= s^{p\omega 2^{k+1}} s^p = s^{\omega+p}\end{aligned}$$

together with $u^{p\omega 2^{k+1}} \leqslant_k u^{p\omega 2^{k+1}+p}$ given by Lemma 4.29. Therefore, $u^{p\omega 2^{k+1}}$ and $u^{p\omega 2^{k+1}+p}$ are witnesses of the k-pair $(s^\omega, s^{\omega+p})$. Using Lemma 4.25, we obtain that $(s^\omega, s^{\omega+p})$ is a $\mathrm{Pol}(\mathcal{C})$-pair.

Therefore, Lemma 4.29 allows us to find a first example of non-trivial $\mathrm{Pol}(\mathcal{C})$-pairs, which was not possible using only Lemma 4.26 on each stratum.

We end this section with a last example of non trivial words comparable by $\leqslant_k$, yielding again a non-trivial example of k-pairs and $\mathrm{Pol}(\mathcal{C})$-pairs.

Lemma 4.31 ([Place and Zeitoun, 2017c])**.** *Let p be the period of $\mathcal{C}$, k be an integer and let $u, v \in A^*$ be two words such that $u^p \leqslant_{\mathcal{C}} v$. Then for every m, m_1' and m_2' at least $2^{k+1} - 1$, we have*

$$u^{pm} \leqslant_k u^{pm_1'} v u^{pm_2'}.$$

A final remark is that these results emphasize that the quasi-orders $\leqslant_k$ are much easier to study than $\leqslant_{\mathrm{Pol}(\mathcal{C})}$ since the classes $\mathrm{Pol}_k(\mathcal{C})$ are finite. This justifies to focus on the preorders $\leqslant_k$ in order to obtain results for $\mathrm{Pol}(\mathcal{C})$.

4.2.5 Deciding $\mathrm{Pol}(\mathcal{C})$-separation

Recall that solving $\mathrm{Pol}(\mathcal{C})$-separation amounts to computing the set of $\mathrm{Pol}(\mathcal{C})$-pairs for a given morphism α. As explained previously, the usual approach consists in starting from a set of trivial pairs, and then in constructing inductively new pairs using some closure properties like Proposition 4.15 or (as we will see just after) inspired by Lemma 4.31 (see how close are the equations in Theorem 4.14 and in Lemma 4.31).

This leads to designing fixpoint algorithms by considering closure operations that are specific to the class $\mathrm{Pol}(\mathcal{C})$. Consider Lemma 4.31. The closure operation it suggests is the following: if (r, s) is a $\mathrm{Pol}(\mathcal{C})$-pair, then so is $(r^\omega, s^\omega t s^\omega)$ for any t element of M "compatible" with r^p, where p is the period of $\mathcal{C}$ and $\omega = \omega(M)$.

Morphism completion

To state this operation properly, we need to define more precisely what "compatible" means. To this end, we use the finiteness of $\mathcal{C}$ to extend the relation $\leqslant_{\mathcal{C}}$ on M. The key property we look for is that if two words u, v satisfy $u \leqslant_{\mathcal{C}} v$, we also have $\alpha(u) \leqslant_{\mathcal{C}} \alpha(v)$.

This leads to the following definition.

Definition 4.32. A monoid morphism $\alpha : A^* \to M$ is $\mathcal{C}$*-compatible* if for all words $u, v \in A^*$ such that $\alpha(u) = \alpha(v)$, we have $u \sim_{\mathcal{C}} v$.

Note that when α is $\mathcal{C}$-compatible, for all $s \in M$, the class of s modulo $\sim_{\mathcal{C}}$ is well-defined as the class modulo $\sim_{\mathcal{C}}$ of any element of $\alpha^{-1}(s)$ (if s has no preimage then we simply set its class to $\varnothing$). In particular, the above definition of $\leqslant_{\mathcal{C}}$ gives a quasi-order on M, compatible with its product.

If α is not $\mathcal{C}$-compatible, we can replace it by its $\mathcal{C}$*-completion*. This new morphism $\overline{\alpha}$ recognizes all languages recognized by α and is $\mathcal{C}$-compatible. To construct this completion, observe that $\mathcal{C}$ is finite and contains only regular languages, hence there is a monoid morphism $\beta : A^* \to N$ recognizing all languages in $\mathcal{C}$. We then let $\overline{\alpha} : A^* \to M \times N$ be the morphism defined by $\overline{\alpha}(w) = (\alpha(w), \beta(w))$. Observe that the $\mathcal{C}$-completion of α is computable from α since $\mathcal{C}$ is finite.

The obtained morphism may not be surjective, meaning that some elements of its co-domain are not useful (in terms of language recognition). However, restricting the co-domain to the image of the morphism is harmless, so we may always assume that we consider surjective morphisms. We may summarize the previous transformations in the following lemma.

Lemma 4.33. *Let L_1, L_2 be two regular languages. If $\mathcal{C}$ is a finite class of regular languages, we can compute a surjective $\mathcal{C}$-compatible morphism recognizing both L_1 and L_2.*

Note in particular that the hypothesis that $\mathcal{C}$ is finite is crucial here for ensuring that such a morphism is computable. Moreover, proving Lemma 4.33 without this hypothesis would yield an algorithm for solving $\mathcal{C}$-separation for any class $\mathcal{C}$.

Example 4.34. We illustrate this construction in the case where $\mathcal{C}$ is AT, the class of alphabet testable languages. In this case, we have $u \sim_{\mathsf{AT}} v$ whenever u and v are two words with the same alphabet. Therefore, a morphism α is AT-compatible if any two words with the same image under α use the same alphabet. To obtain the AT-completion of α, observe that 2^A is a monoid when endowed with union and having $\varnothing$ as neutral element. Then $\beta(w) = (\alpha(w), \mathrm{alph}(w))$ is an AT-compatible monoid morphism recognizing the languages recognized by α.

Computing the Pol($\mathcal{C}$)-pairs

Assuming that α is $\mathcal{C}$-compatible, we may state the aforementioned closure property. Recall that we fix a finite class $\mathcal{C}$ of regular languages, closed under union, intersection and quotients. We will proceed the same way with α: in the following, we always assume that $\alpha : A^* \to M$ is a surjective monoid morphism, which is $\mathcal{C}$-compatible and such that M is a finite monoid.

Proposition 4.35. *Let (e, f) be a* Pol($\mathcal{C}$)*-pair for α with $e^2 = e$ and $f^2 = f$. For every $t \in M$ such that $e \leqslant_{\mathcal{C}} t$, the pair (e, ftf) is a* Pol($\mathcal{C}$)*-pair for α.*

Proof. Let (e, f) be a Pol($\mathcal{C}$)-pair with $e^2 = e$ and $f^2 = f$. Let $t \in M$ such that $e \leqslant_{\mathcal{C}} t$. To prove the result, we follow the usual approach: we prove that (e, ftf) is a k-pair for every $k \geqslant 0$.

Let $k \geqslant 0$. Since (e, f) is a Pol($\mathcal{C}$)-pair, there exists $u \in \alpha^{-1}(e)$ and $v \in \alpha^{-1}(f)$ such that $u \leqslant_k v$.

Since α is surjective, there exists $w \in \alpha^{-1}(t)$. Let p be the period of $\mathcal{C}$. Since $\alpha(u^p) = e^p = e \leqslant_{\mathcal{C}} t$ and α is $\mathcal{C}$-compatible, we have $u^p \leqslant_{\mathcal{C}} w$. We can thus apply Lemma 4.31: taking $\ell = p2^{k+1}$, we have

$$u^\ell \leqslant_k u^\ell w u^\ell.$$

Since $u \leqslant_k v$, by Lemma 4.20, we have $u^\ell w u^\ell \leqslant_k v^\ell w v^\ell$. Therefore, we have $u^\ell \leqslant_k v^\ell w v^\ell$. Since $\alpha(u^\ell) = e^\ell = e$ and $\alpha(v^\ell w v^\ell) = f^\ell t f^\ell = ftf$, we obtain that (e, ftf) is a k-pair. □

We may derive a closure operation from this property, and state a first attempt to compute Pol($\mathcal{C}$)-pairs using a fixpoint algorithm. Consider a set $S \subset M^2$ satisfying the following properties:

- S contains all trivial pairs (s, s) for $s \in M$.
- S is closed under *product*: if $(s, s') \in S$ and $(t, t') \in S$, then $(st, s't') \in S$.
- S is closed under the *special* operation: if $(r, s) \in S$, then $(r^\omega, s^\omega t s^\omega) \in S$ for all $t \in M$ such that $r^p \leqslant_{\mathcal{C}} t$ where p is the period of $\mathcal{C}$.

Note that if S and S' satisfy these properties, then so does $S \cap S'$. Therefore, there exists a minimal such set, that we denote by Sat (Sat stands for "saturated"). Observe that this set can be computed using a least fixpoint algorithm: we start from the set of trivial pairs, and close it under both properties until we reach a fixpoint.

Thanks to Propositions 4.15 and 4.35, we know that $\mathsf{Sat} \subset \mathbb{P}_{\mathrm{Pol}(\mathcal{C})}[\alpha]$: each pair (s, t) computed by the algorithm is actually a Pol($\mathcal{C}$)-pair. However, we do not know whether the converse inclusion holds: there may be Pol($\mathcal{C}$)-pairs that are not computed by the algorithm.

To solve this issue, the answer provided in [Place and Zeitoun, 2014a] (in the case $\mathcal{C} = \mathsf{AT}$) consists in considering objects capturing more properties of $\mathrm{Pol}(\mathcal{C})$ than pairs. This approach relies on a refined notion of pairs: *compatible pairs*. In the rest of this section, we define this refined notion, and show how to compute compatible pairs, and how to use them to recover the desired pairs.

Consider a finite class $\mathcal{D}$, and two $\mathcal{D}$-pairs (s, t) and (s, t'). By definition, there exist witnesses (u, v) and (u', v') for these pairs. We want to record whether we can choose such witnesses with the constraint $u = u'$. In other words, we want to determine when the two pairs (s, t) and (s, t') can be "synchronized". We represent such "compatible" pairs as follows.

Definition 4.36. Let $\mathcal{D}$ be a finite class. A $\mathcal{D}$-compatible pair for α is an element $(s, \{t_1, \dots, t_n\}) \in M \times 2^M$ such that there exists $u \in \alpha^{-1}(s)$ such that for all $i \in [1, n]$, there exists $v_i \in \alpha^{-1}(t_i)$ satisfying $u \leqslant_{\mathcal{D}} v_i$.

The set of such compatible pairs is denoted by $\mathbb{C}_{\mathcal{D}}[\alpha]$.

Observe that since $\mathrm{Pol}(\mathcal{C})$ is infinite, the previous definition does not apply. To define $\mathrm{Pol}(\mathcal{C})$-compatible pairs, we mimic Lemmas 4.23 and 4.25: we say that (s, S) is a $\mathrm{Pol}(\mathcal{C})$-compatible pair if (s, S) is a $\mathrm{Pol}_k(\mathcal{C})$-compatible pair for every integer k. We thus take

$$\mathbb{C}_{\mathrm{Pol}(\mathcal{C})}[\alpha] = \bigcap_{k \geqslant 0} \mathbb{C}_{\mathrm{Pol}_k(\mathcal{C})}[\alpha]$$

In particular, observe that Lemmas 4.23 and 4.24 naturally extend to compatible pairs, as summarized by the following result. Note that we also extend the same convention as before: we write $\mathbb{C}_k[\alpha]$ instead of $\mathbb{C}_{\mathrm{Pol}_k(\mathcal{C})}[\alpha]$.

Lemma 4.37. *Let $\alpha : A^* \to M$ be a monoid morphism.*

- *For every integer k, we have $\mathbb{C}_{k+1}[\alpha] \subset \mathbb{C}_k[\alpha]$.*
- *There exists an integer ℓ such that $\mathbb{C}_{\mathrm{Pol}(\mathcal{C})}[\alpha] = \mathbb{C}_\ell[\alpha]$.*

Recovering $\mathrm{Pol}(\mathcal{C})$-pairs from $\mathrm{Pol}(\mathcal{C})$-compatible pairs

The algorithm we present will compute the set of $\mathrm{Pol}(\mathcal{C})$-compatible pairs. Before presenting it, we explain how to recover the set of $\mathrm{Pol}(\mathcal{C})$-pairs from this information. Note that each $\mathrm{Pol}(\mathcal{C})$-pair (s, t) appears in every $\mathbb{C}_k[\alpha]$ as $(s, \{t\})$, hence $(s, \{t\}) \in \mathbb{C}_{\mathrm{Pol}(\mathcal{C})}[\alpha]$. Conversely, for every $\mathrm{Pol}(\mathcal{C})$-compatible pair (s, T) and $t \in T$, the pair (s, t) is a k-pair for every integer k, hence a $\mathrm{Pol}(\mathcal{C})$-pair. Therefore, one can recover the set of pairs from the set of compatible pairs, as shown by the following lemma.

Lemma 4.38. *We have*

$$\mathbb{P}_{\mathrm{Pol}(\mathcal{C})}[\alpha] = \{(s, t) \mid (s, \{t\}) \in \mathbb{C}_{\mathrm{Pol}(\mathcal{C})}[\alpha]\}.$$

Computing Pol($\mathcal{C}$)-compatible pairs

It remains to describe how to compute the set of Pol($\mathcal{C}$)-compatible pairs. The algorithm from [Place and Zeitoun, 2017d] computes the set of Pol($\mathcal{C}$)-compatible pairs for α using a least fixpoint algorithm. It starts from a set of trivial pairs, and saturates it by several closure operations. We devote a lemma for each of these steps. We first consider the trivial pairs.

Lemma 4.39. *For every $s \in M$, $(s, \{s\})$ is a* Pol($\mathcal{C}$)*-compatible pair for α.*

Note that, due to the definition of Pol($\mathcal{C}$)-compatible pairs, the usual approach to prove that some pair (s, S) is a Pol($\mathcal{C}$)-compatible pair consists in showing that it is a k-compatible pair for every integer k. This is the approach followed to prove the Lemma 4.39, as well as the next results which consider closure properties of $\mathbb{C}_{\text{Pol}(\mathcal{C})}[\alpha]$.

Proof of Lemma 4.39. Let $k \geqslant 0$ and $s \in M$. Since α is surjective, there is $u \in \alpha^{-1}(s)$. Since $u \leqslant_k u$, we obtain that (s, s) is a k-pair, hence $(s, \{s\})$ is a k-compatible pair.

Since this is valid for every integer k, we obtain that $(s, \{s\})$ is a Pol($\mathcal{C}$)-compatible pair. ☐

Such a compatible pair $(s, \{s\})$ is called *trivial*. We then consider some closure operations. The first one directly comes from the definition of compatible pairs: their set is closed under inclusion.

Lemma 4.40. *Let (s, S) be a* Pol($\mathcal{C}$)*-compatible pair. Then for every $T \subset S$, (s, T) is also a* Pol($\mathcal{C}$)*-compatible pair.*

Proof. Let $k \geqslant 0$, let (s, S) be a Pol($\mathcal{C}$)-compatible pair and let $T \subset S$.

By hypothesis, (s, S) is a k-compatible pair, hence there are some witnesses for s and for each element of S. Since $T \subset S$, the same witnesses ensure that (s, T) is also a k-compatible pair. Since this holds for every integer k, we obtain that (s, T) is a Pol($\mathcal{C}$)-compatible pair. ☐

Recall that the set of pairs is a sub-monoid of M^2. We can prove a similar result: the set of compatible pairs is also a sub-monoid of $M \times 2^M$, where the multiplicative law is given by

$$(s, S)(t, T) = (st, \{s't' \mid s' \in S, t' \in T\})$$

This closure property is again a consequence of the fact that all the quasi-orders $\leqslant_k$ for $k \geqslant 0$ are compatible with concatenation. Its proof is a slight extension of the one of Proposition 4.15: it relies on the compatibility of $\leqslant_k$ with concatenations.

Lemma 4.41. *Let (s, S) and (t, T) be* Pol($\mathcal{C}$)*-compatible pairs for α. Then (st, ST) is also a* Pol($\mathcal{C}$)*-compatible pair for α.*

Note that Lemmas 4.39, 4.40 and 4.41 actually do not use that we consider the class $\mathrm{Pol}(\mathcal{C})$. They only require that the classes $\mathrm{Pol}_k(\mathcal{C})$ are closed under quotient and that α is surjective. In particular, they actually hold for every class $\mathcal{D}$ closed under union, intersection and quotient.

In order to obtain an algorithm for $\mathrm{Pol}(\mathcal{C})$-separation, we thus require a closure property specific to the $\mathrm{Pol}(\mathcal{C})$ class. This is the goal of the following result, an extension of Proposition 4.35.

Lemma 4.42. *Let (e, E) be an idempotent $\mathrm{Pol}(\mathcal{C})$-compatible pair, i.e. $e^2 = e$ and $\{st, (s,t) \in E^2\} = E$. Let $T = \{t \in M, t \sim_{\mathcal{C}} e\}$. Then (e, ETE) is also a $\mathrm{Pol}(\mathcal{C})$-compatible pair.*

This is actually the only operation requiring that α is $\mathcal{C}$-compatible. This is not surprising, since this operation is the only one which is not generic, but specific to the class $\mathrm{Pol}(\mathcal{C})$.

Proof. We fix a $\mathrm{Pol}(\mathcal{C})$-compatible pair (e, E) with $E = \{s_1, \dots, s_n\}$. Let $T = \{t \in M, t \sim_{\mathcal{C}} e\}$. We prove that (e, ETE) is a $\mathrm{Pol}(\mathcal{C})$-compatible pair by proving that it is a k-compatible pair for every integer k. We fix such an integer k. To show that (e, ETE) is a k-compatible pair, we construct a word u in $\alpha^{-1}(e)$ such that for every $s \in E$, there exists $v \in \alpha^{-1}(s)$ satisfying $u \leqslant_k v$.

By definition, there exists $w, w_1, \dots, w_n$ such that

$$\alpha(w) = e, \quad \alpha(w_i) = s_i \text{ and } w \leqslant_k w_i, \quad 1 \leqslant i \leqslant n.$$

Let p be the period of $\mathcal{C}$ and $\ell = p2^{k+1}$. Let $u = w^\ell$, and observe that $\alpha(u) = e^\ell = e$. We shall use this word u as a witness in order to show that (e, ETE) is a k-compatible pair.

It remains to construct witnesses for every element of ETE. Such an element can be written rts where $r, s \in E$ and $t \in T$. We thus look for a word v such that

$$\alpha(v) = rts \text{ and } u \leqslant_k v.$$

Since α is surjective, we can find $v_t \in A^*$ such that $\alpha(v_t) = t$. Note also that since E is idempotent, we have $E^\ell = E$, hence r and s can both be decomposed as $r_1 \cdots r_\ell$ and $s_1 \cdots s_\ell$ where the r_i's and s_i's lie in E. Using that (e, E) is a $\mathrm{Pol}(\mathcal{C})$-compatible pair, hence a k-compatible pair, we can find words $u_1, \dots, u_\ell, w_1, \dots, w_\ell$ mapped on $r_1, \dots, r_\ell, s_1, \dots, s_\ell$ such that

$$w \leqslant_k u_i \text{ and } w \leqslant_k w_i \text{ for } 1 \leqslant i \leqslant \ell.$$

We define v as $u_1 \cdots u_\ell v_t w_1 \cdots w_\ell$. Observe that by definition

$$\alpha(v) = r_1 \cdots r_\ell t s_1 \cdots s_\ell = rts$$

It thus remains to prove that $u \leqslant_k v$.

Note that we have $\alpha(v_t) = t \sim_{\mathcal{C}} e = \alpha(w^p)$. Therefore, since α is $\mathcal{C}$-compatible, we have $v_t \sim_{\mathcal{C}} w^p$. We may therefore apply Lemma 4.31 and obtain that

$$u = w^\ell \leqslant_k w^\ell v_t w^\ell$$

Moreover, by definition of u_i and w_i for $1 \leqslant i \leqslant \ell$, we have

$$u \leqslant_k w^\ell v_t w^\ell \leqslant_k u_1 \cdots u_\ell v_t w_1 \cdots w_\ell = v.$$

where the second inequality is obtained from the properties $w \leqslant_k u_i$ and $w \leqslant_k w_i$ using that $\leqslant_k$ is compatible with concatenations. This concludes the proof. □

Using Lemma 4.42, we may define a closure operation satisfied by the set of $\mathrm{Pol}(\mathcal{C})$-compatible pairs, and finally state the result of [Place and Zeitoun, 2017d] concluding about decidability of $\mathrm{Pol}(\mathcal{C})$-separation.

Theorem 4.43 ([Place and Zeitoun, 2017d])**.** *Let $\mathcal{C}$ be a finite class of regular languages, closed under union, intersection and quotients. Let $\alpha : A^* \to M$ be a surjective $\mathcal{C}$-compatible morphism. The set of $\mathrm{Pol}(\mathcal{C})$-compatible pairs for α is the smallest set* Sat *satisfying:*

- Sat *contains all trivial pairs $(s, \{s\})$ for $s \in M$.*
- Sat *is stable under* downset*: if $(s, S) \in \mathsf{Sat}$, then $(s, T) \in \mathsf{Sat}$ for every $T \subset S$.*
- Sat *is stable under* product*: if $(s, S), (t, T) \in \mathsf{Sat}$, then $(st, ST) \in \mathsf{Sat}$.*
- Sat *is stable under the* special operation*: if $(e, E) \in \mathsf{Sat}$ is idempotent, then $(e, ETE) \in \mathsf{Sat}$ where $T = \{t \in M, t \sim_{\mathcal{C}} e\}$.*

The proof of Theorem 4.43 is separated in two parts, corresponding to the inclusions $\mathsf{Sat} \subset \mathbb{C}_{\mathrm{Pol}(\mathcal{C})}[\alpha]$ and $\mathbb{C}_{\mathrm{Pol}(\mathcal{C})}[\alpha] \subset \mathsf{Sat}$. The first one is called the *soundness part*: indeed, we check that every element computed in Sat is actually a $\mathrm{Pol}(\mathcal{C})$-compatible pair. This is proved inductively on the several operations defining Sat: we prove that applying each operation on a set of compatible pairs yields a set of compatible pairs. This is a consequence of Lemmas 4.39, 4.40, 4.41 and 4.42.

The converse part is called the *completeness part*: we have to prove that every $\mathrm{Pol}(\mathcal{C})$-compatible pair is actually constructed at some point in Sat. This part is much harder, and requires some involved decomposition arguments. We thus do not present the proof here, but refer to [Place and Zeitoun, 2019] instead.

We end this section by summarizing the algorithm solving $\mathrm{Pol}(\mathcal{C})$-separation. Recall that the first step is to compute a surjective $\mathcal{C}$-compatible morphism

α recognizing the two input languages. Using a least fixpoint algorithm, we then compute the smallest set containing trivial pairs, and stable by downset, product and special operation. By Theorem 4.43, this set is the set of Pol($\mathcal{C}$)-compatible pairs for α. We finally use Lemma 4.38 to recover the set of Pol($\mathcal{C}$)-pairs for α, and solve separation using Lemma 4.11. We thus obtain the following result.

Corollary 4.44. *Let $\mathcal{C}$ be a finite class of regular languages, closed under Boolean operations and quotients. The* Pol($\mathcal{C}$)*-separation problem is decidable.*

4.3 Pol($\mathcal{C}$)-separation is PSpace-hard

The algorithm of the previous section gives an upper bound for the complexity of the Pol($\mathcal{C}$)-separation problem, but this bound depends on the class $\mathcal{C}$. Indeed, since we compute $\mathcal{C}$-compatible morphisms, the size of the obtained monoid depends on $\mathcal{C}$. For example, when $\mathcal{C}$ is level 0 of Straubing-Thérien hierarchy, i.e. $\{\varnothing, A^*\}$, such a monoid has linear size with respect to the ones recognizing the input languages. However, when $\mathcal{C} = \mathsf{AT}$, this monoid has now exponential size with respect to the alphabet. Indeed, as presented in Example 4.34, the AT-completion of a monoid morphism $\alpha : A^* \to M$ has codomain $M \times 2^A$, which has exponential size with respect to A.

As shown in [Place and Zeitoun, 2018a], when $|\mathcal{C}|$ is independent of the alphabet of the considered languages, we can prove a NLogSpace upper bound for the Pol($\mathcal{C}$)-separation problem. In the case $\mathcal{C} = \mathsf{AT}$, this upper bound becomes PSpace.

In this section, we prove the complementary result given by Theorem 4.10: we give a lower bound on the complexity of the Pol($\mathcal{C}$)-separation problem when $\mathcal{C}$ is a sufficiently large positive variety. By "sufficiently large", we mean that $\mathcal{C}$ should be able to distinguish words over different alphabets, i.e. that it contains the variety AT. We recall the statement of this theorem.

Theorem 4.10. *Let $\mathcal{C}$ be a positive variety of regular languages such that $\mathsf{AT} \subset \mathcal{C}$ and $\mathcal{C} \neq$ Pol($\mathcal{C}$). Then the* Pol($\mathcal{C}$)*-separation problem is PSpace-hard.*

Observe that since AT is level 1 of Straubing-Thérien hierarchy, and this hierarchy is strict. Therefore, Theorems 4.10 and 3.37 prove that separation is PSpace-hard for every half-level of the hierarchy greater than $\frac{1}{2}$, regardless of the input format. This is tight for example for level $\frac{3}{2}$ [Place and Zeitoun, 2018a].

Before diving into the proof, we state two additional remarks. The first one illustrates that both the hypotheses are needed.

Remark 4.46. Observe that the two hypotheses $\mathsf{AT} \subset \mathcal{C}$ and $\mathcal{C} \neq$ Pol($\mathcal{C}$) are needed: the Reg-separation problem is NLogSpace when the inputs are given by automata (see the proof of Corollary 3.39), hence certainly not PSpace-hard, but in this case Reg = Pol(Reg). Moreover, the variety $\{\varnothing, A^*\}$ does not contain AT, and its polynomial closure is level $\frac{1}{2}$ of Straubing-Thérien hierarchy, for which separation for languages given by automata is NLogSpace (see [Place and Zeitoun, 2018a]).

The second remark considers a generalization of Theorem 4.10 to the case of unambiguous polynomial closure.

Remark 4.47. The construction we are about to present can be slightly modified to obtain a PSpace lower bound for the UPol($\mathcal{C}$)-separation problem when $\mathcal{C}$ is a positive variety such that $\mathsf{AT} \subset \mathcal{C} \neq \mathrm{UPol}(\mathcal{C})$. However, the proof of Theorem 4.10 is already rather technical, hence we choose to present it only for the class Pol($\mathcal{C}$).

The end of this section is devoted to the proof of Theorem 4.10. We thus fix a positive variety $\mathcal{C}$ of regular languages such that $\mathsf{AT} \subset \mathcal{C} \neq \mathrm{Pol}(\mathcal{C})$. By Theorem 4.5, we know that Pol($\mathcal{C}$) is also a positive variety. Observe that the statement of Theorem 4.10 does not precise the representation of the input languages. In view of Theorem 3.37, since Pol($\mathcal{C}$) is a positive variety, it is sufficient to prove Theorem 4.10 when the input languages are given by automata.

Remark 4.48. This is the only place where we use that $\mathcal{C}$ (and hence Pol($\mathcal{C}$)) is a positive variety. In the rest of the proof, we only use that $\mathcal{C}$ (hence Pol($\mathcal{C}$)) is closed under union, intersection and quotients.

Indeed, even if we use Proposition 3.36 (whose statement requires Pol($\mathcal{C}$) to be positive variety) later, the closure under inverse morphism is actually not used in its proof from [Pin, 1995].

To prove the PSpace-hardness result, we first introduce the problem we reduce, namely the 3-CNF-QBF-SAT problem. We devote a first subsection to presenting this emblematic PSpace-complete problem.

4.3.1 Satisfiability for quantified Boolean formulas

A *Boolean formula* is a logical formula, built from a set of variables using conjunctions, disjunctions and negations. It is a formula in 3-conjunctive normal form (3-CNF) when it has the following shape:

$$\bigwedge_{i=1}^{p} (\ell_{i,1} \vee \ell_{i,2} \vee \ell_{i,3}).$$

where each $\ell_{i,j}$ is a *literal*, i.e. a variable or its negation. Each sub-formula $(\ell_{i,1} \vee \ell_{i,2} \vee \ell_{i,3})$ is called a *clause*. This kind of formulas are well-known due to the famous NP-complete problem 3-SAT, asking for satisfiability of a logical formula in 3-CNF. The 3-CNF-QBF-SAT problem is a generalization of this problem to the setting of quantified formulas.

A *quantified Boolean formula* (QBF) φ is a formula of the form

$$Q_1 x_1 \cdots Q_n x_n \psi$$

where each Q_i is a quantifier among $\{\forall, \exists\}$, each x_i is a variable, and ψ is a Boolean formula. If ψ is in 3-CNF, the formula φ is said to be a 3-CNF quantified Boolean formula (3-CNF-QBF for short).

A variable x is *free* in φ if x occurs in ψ and φ does not contain the quantifier $\forall x$ or $\exists x$. A formula without free variables is said to be a *sentence*.

As explained before, the 3-CNF-QBF-SAT problem asks for satisfiability of a QBF sentence in 3-CNF. To define properly this notion, we introduce valuations.

A *valuation* val is a function $V \to \{\top, \bot\}$ where V is a set of variables. If x is a variable, the equality $\mathsf{val}(x) = \top$ (resp. $\mathsf{val}(x) = \bot$) means that the variable x is set to true (resp. false) by the valuation val. If val is defined on all the free variables of a formula φ, we can define $\mathsf{val}(\varphi)$ inductively as follows:

- $\mathsf{val}(\neg\varphi) = \top$ if and only if $\mathsf{val}(\varphi) = \bot$.
- $\mathsf{val}(\varphi \wedge \psi) = \top$ if and only if $\mathsf{val}(\varphi) = \mathsf{val}(\psi) = \top$.
- $\mathsf{val}(\varphi \vee \psi) = \top$ if and only if $\mathsf{val}(\varphi) = \top$ or $\mathsf{val}(\psi) = \top$.
- $\mathsf{val}(\exists x\varphi) = \top$ if and only if $\mathsf{val}_x(\varphi) = \top$ or $\mathsf{val}_{\overline{x}}(\varphi) = \top$, where val_x (resp. $\mathsf{val}_{\overline{x}}$) are valuations obtained by extending val by setting $\mathsf{val}_x(x) = \top$ (resp. $\mathsf{val}_{\overline{x}}(x) = \bot$).
- $\mathsf{val}(\forall x\varphi) = \top$ if and only if $\mathsf{val}_x(\varphi) = \mathsf{val}_{\overline{x}}(\varphi) = \top$.

We say that a formula φ is *satisfiable* if there exists a valuation val defined on the set of free variables of φ such that $\mathsf{val}(\varphi) = \top$.

Example 4.49.

- The formula $\varphi = \forall y(x \wedge \neg y)$ is a quantified Boolean formula where x is a free variable but not y.

  There are two valuations defined on x: the one setting $\mathsf{val}(x) = \top$ and the other one setting $\mathsf{val}(x) = \bot$. The first one satisfies φ but not the second one.

- The formula $\varphi = \exists x \forall y \exists z(x \vee \neg y \vee z)$ is a QBF sentence since none of the variables x, y, z are free. Moreover, $x \vee \neg y \vee z$ is a 3-CNF formula, hence φ is a 3-CNF-QBF sentence. It is satisfiable since the two valuations defined by $(\mathsf{val}(x), \mathsf{val}(y), \mathsf{val}(z)) = (\top, \top, \top)$ and $(\top, \bot, \top)$ satisfy $x \vee \neg y \vee z$.

We may finally state the problem we consider. The *3-CNF-QBF-SAT problem* takes a 3-CNF-QBF sentence as input and asks whether it is satisfiable. It is a canonical example of a PSpace-complete problem, see [Sipser, 1997].

4.3.2 Outline of the reduction

To prove Theorem 4.10, we reduce the 3-CNF-QBF-SAT problem to the **non-**$\mathrm{Pol}(\mathcal{C})$-separation problem. Note that this will prove that the $\mathrm{Pol}(\mathcal{C})$-separation problem is PSpace-hard since PSpace is closed under complement. We thus fix a 3-CNF quantified Boolean sentence φ. Our goal is to construct in LogSpace two automata recognizing some languages L_φ, L'_φ such that the following holds.

Proposition 4.50. *Given a QBF sentence φ in 3-CNF, φ is satisfiable if and only if L_φ is not $\mathrm{Pol}(\mathcal{C})$-separable from L'_φ.*

Similarly to Section 3.3, we construct two languages of polynomial size in φ, and we claim that they can be computed in LogSpace, although we do not explicitly prove it. The construction is inductive: we start from the quantifier-free part of φ and inductively add each quantifier. To handle this induction, we need a stronger result which takes care of quantified formulas having free variables, not only quantified sentences. As we will see, the free variables of a formula φ will have an influence on the alphabet of L_φ, L'_φ.

The rest of this section is organized as follows. In Subsection 4.3.3, we define the alphabet of L_φ, L'_φ, and state this stronger result. In Subsection 4.3.4, we give the construction for quantifier free formulas and prove the base case of the induction. Then, we devote Subsection 4.3.5 and Subsection 4.3.6 to the inductive steps (one for each type of quantifier).

4.3.3 Encoding valuations into alphabets

Let $X = \{x_1, \dots, x_n\}$ be the variables of φ. For each variable $x \in X$, we create two letters x and $\overline{x}$. We denote by $\overline{X}$ the set $\{\overline{x} \mid x \in X\}$. Together with these letters, our construction needs a constant number p (depending only on the class $\mathcal{C}$) of fresh new letters at each inductive step. Therefore, if φ has n quantifiers, the alphabet of the languages L_φ, L'_φ is a superset of $X \cup \overline{X}$ containing np additional letters. We denote by M the set of these new letters, so that the languages L_φ, L'_φ are defined over the alphabet $A = X \cup \overline{X} \cup M$.

The induction result we prove establishes a link between satisfiability of sub-formulas of φ and separability of some languages. Note that the satisfiability of these sub-formulas may depend of the value of their free variables. Thus, the languages may have to change according to these values. To this end, we consider several languages by considering the intersection of a fixed language with suitable alphabets. With each valuation val, we associate the alphabet

$$A_{\mathsf{val}} = (X \cup \overline{X} \cup M) \setminus (\{x, \mathsf{val}(x) = \bot\} \cup \{\overline{x}, \mathsf{val}(x) = \top\}).$$

With this notation, we may state the generalization of Proposition 4.50 we actually prove.

Proposition 4.51. *Let φ be a QBF-formula and* val *be a valuation defined exactly on the free variables of φ. Then we can construct two languages L_φ, L'_φ over A such that $\mathsf{val}(\varphi) = \top$ if and only if $L_\varphi \cap A^*_{\mathsf{val}}$ is* **not** *Pol($\mathcal{C}$)-separable from $L'_\varphi \cap A^*_{\mathsf{val}}$.*

Observe that when φ is a sentence, we have $A_{\mathsf{val}} = A$ for any valuation val satisfying Proposition 4.51. Then φ is satisfiable if and only if L_φ is not Pol($\mathcal{C}$)-separable from L'_φ, which proves Proposition 4.50. We now prove Proposition 4.51 by induction on φ. The languages L_φ, L'_φ are constructed during the proof. We actually give an inductive construction for regular expressions and automata recognizing these languages, which can be done in LogSpace.

4.3.4 Base case: quantifier-free formulas

Let ψ be a quantifier-free formula in 3-CNF:

$$\psi = \bigwedge_{i=1}^{p} (\ell_{i,1} \vee \ell_{i,2} \vee \ell_{i,3}),$$

where each $\ell_{i,j}$ is a literal.

We define L_ψ as a language whose words witness that the clauses of ψ are satisfied:

$$L_\psi = \prod_{i=1}^{p} (\ell_{i,1} + \ell_{i,2} + \ell_{i,3}).$$

We also define $L'_\psi = (X + \overline{X})^*$. Automata recognizing these languages are given in Figure 4.1.

Let val be a valuation defined on all variables of ψ. Note that finding a word w in $L_\psi \cap A^*_{\mathsf{val}}$ proves that val satisfies ψ. Indeed, since $w \in L_\psi$, it can be written as $\ell_{1,i_1} \cdots \ell_{p,i_p}$. Since $w \in A^*_{\mathsf{val}}$, we have $\mathsf{val}(\ell_{j,i_j}) = \top$ for $1 \leqslant j \leqslant p$. Therefore, every clause of ψ contains a literal which is true for val, hence $\mathsf{val}(\psi) = \top$.

Moreover, observe that $L_\psi \subset L'_\psi$, hence, $L_\psi \cap A^*_{\mathsf{val}}$ is Pol($\mathcal{C}$)-separable from $L'_\psi \cap A^*_{\mathsf{val}}$ if and only if $L_\psi \cap A^*_{\mathsf{val}} = \varnothing$ i.e. if $\mathsf{val}(\psi) = \bot$. This proves that Proposition 4.51 holds for quantifier-free formulas.

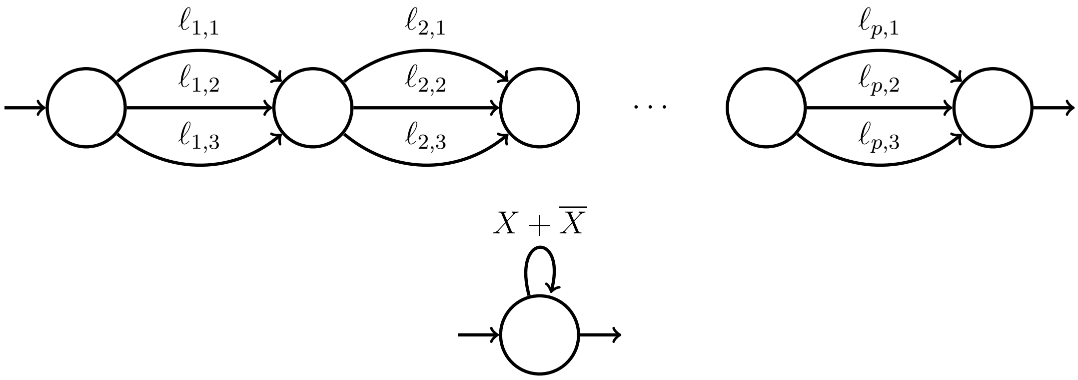


Figure 4.1 – Automata for L_ψ (above) and L'_ψ (below).

4.3.5 Induction step: existential quantifier

Assume that $\varphi = \exists x \xi(x)$, and assume that there are two languages L_ξ, L'_ξ such that Proposition 4.51 holds for ξ, i.e. for every valuation val defined exactly on the free variables of ξ, we have $\mathsf{val}(\xi) = \top$ if and only if $L_\xi \cap A^*_{\mathsf{val}}$ is **not** $\mathrm{Pol}(\mathcal{C})$-separable from $L'_\xi \cap A^*_{\mathsf{val}}$. Our goal is to use L_ξ, L'_ξ to construct two languages L_φ, L'_φ such that the following holds.

Proposition 4.52. *For every valuation* val *defined on the free variables of* φ*,* $L_\varphi \cap A^*_{\mathsf{val}}$ *is not* $\mathrm{Pol}(\mathcal{C})$*-separable from* $L'_\varphi \cap A^*_{\mathsf{val}}$ *if and only if*

- *either* $L_\xi \cap (A_{\mathsf{val}} \setminus \{x\})^*$ *is not* $\mathrm{Pol}(\mathcal{C})$*-separable from* $L'_\xi \cap (A_{\mathsf{val}} \setminus \{x\})^*$*,*
- *or* $L_\xi \cap (A_{\mathsf{val}} \setminus \{\overline{x}\})^*$ *is not* $\mathrm{Pol}(\mathcal{C})$*-separable from* $L'_\xi \cap (A_{\mathsf{val}} \setminus \{\overline{x}\})^*$*.*

We first show that Proposition 4.52 ensures that Proposition 4.51 holds for φ. Take a valuation val defined on the free variables of φ. Recall that we can construct two valuations val_x (resp. $\mathsf{val}_{\overline{x}}$) by setting x to $\top$ (resp. $\bot$) in val. By definition, we have $\mathsf{val}(\varphi) = \top$ if and only if $\mathsf{val}_x(\xi) = \top$ or $\mathsf{val}_{\overline{x}}(\xi) = \top$.

Then note that $A_{\mathsf{val}_x} = A_{\mathsf{val}} \setminus \{\overline{x}\}$ and $A_{\mathsf{val}_{\overline{x}}} = A_{\mathsf{val}} \setminus \{x\}$. Using induction hypothesis, we get that $\mathsf{val}(\varphi) = \top$ if and only if either $L_\xi \cap (A_{\mathsf{val}} \setminus \{x\})^*$ is not $\mathrm{Pol}(\mathcal{C})$-separable from $L'_\xi \cap (A_{\mathsf{val}} \setminus \{x\})^*$ or $L_\xi \cap (A_{\mathsf{val}} \setminus \{\overline{x}\})^*$ is not $\mathrm{Pol}(\mathcal{C})$-separable from $L'_\xi \cap (A_{\mathsf{val}} \setminus \{\overline{x}\})^*$. Using Proposition 4.52, this is equivalent to say that $L_\varphi \cap A^*_{\mathsf{val}}$ is $\mathrm{Pol}(\mathcal{C})$-separable from $L'_\varphi \cap A^*_{\mathsf{val}}$, and Proposition 4.51 holds.

It remains to construct L_φ, L'_φ and prove Proposition 4.52. We first consider a language $K \in \mathrm{Pol}(\mathcal{C}) \setminus \mathcal{C}$, and we rename its letters so that it uses only fresh letters. We also introduce a fresh letter #. We then define:

$$\begin{aligned} L_\varphi &= (K\#(x+\overline{x})L_\xi(x+\overline{x})\#K\#)^*, \\ L'_\varphi &= (K\#(x+\overline{x})L'_\xi(x+\overline{x})\#K\#)^*(T_x + T_{\overline{x}}), \end{aligned}$$

where, for $y \in \{x, \overline{x}\}$,

$$T_y = (\overline{K}\#y(A \setminus \overline{y})^* y\#\overline{K}\#) \cdot (K\#y(A \setminus \overline{y})^* y\#K\#)^*.$$

Automata for L_φ and L'_φ can be constructed from automata recognizing L_ξ and L'_ξ as shown in Figure 4.2.

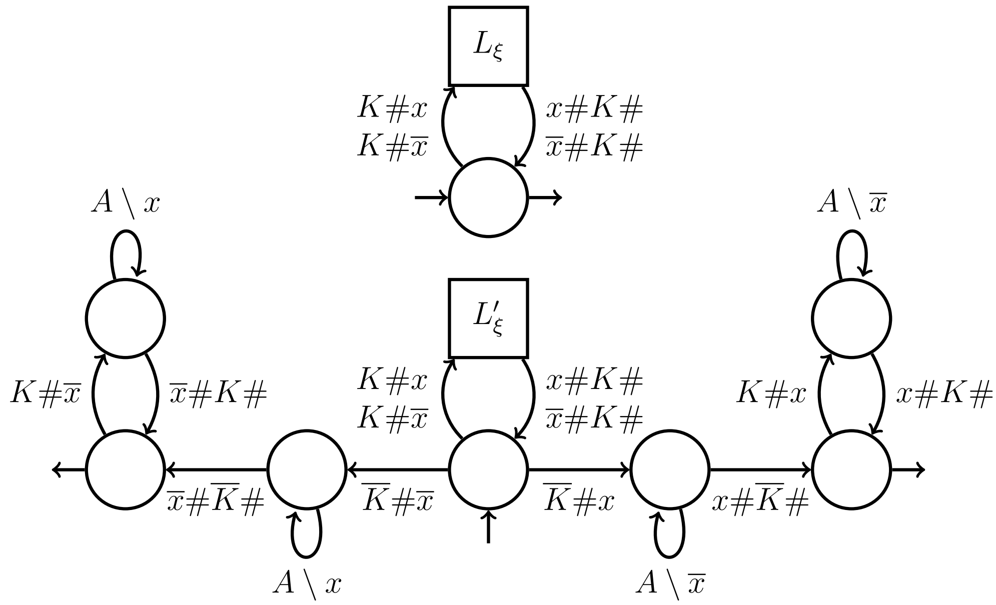


Figure 4.2 – Existential case: Automata for L_φ (above) and L'_φ (below).

Before proving Proposition 4.52, we give some intuition about this construction. First observe that L'_φ is constructed as a union $L'_x \cup L'_{\overline{x}}$ where, for $y \in \{x, \overline{x}\}$,

$$L'_y = (K\#(x + \overline{x})L'_\xi(x + \overline{x})\#K\#)^* T_y.$$

Hence L_φ is not Pol($\mathcal{C}$)-separable from L'_φ if and only if L_φ is not Pol($\mathcal{C}$)-separable from L'_x or from $L_{\overline{x}}$. In our case, L'_x and $L'_{\overline{x}}$ follow the same construction, with x and $\overline{x}$ exchanged. We can actually prove that for $y \in \{x, \overline{x}\}$, $L_\varphi \cap A^*_{\mathsf{val}}$ is not Pol($\mathcal{C}$)-separable from $L'_y \cap A^*_{\mathsf{val}}$ if and only if $L_\xi \cap (A_{\mathsf{val}} \setminus \{y\})^*$ is not Pol($\mathcal{C}$)-separable from $L'_\xi \cap (A_{\mathsf{val}} \setminus \{y\})^*$.

Recall that the characteristic property of the class Pol($\mathcal{C}$) given by Theorem 4.14 states (in essence) that Pol($\mathcal{C}$) cannot distinguish two languages of the form L^* and $(L')^*K(L')^*$ when:

- L is not Pol($\mathcal{C}$)-separable from L', and
- K is not $\mathcal{C}$-separable from L.

Observe in particular that the languages $L_\varphi, L'_x, L'_{\overline{x}}$ follow almost this pattern: up to removing the technical details, we have, for $y \in \{x, \overline{x}\}$,

$$
\begin{aligned}
L_\varphi &= (KL_\xi K)^* \\
L'_y &= (KL'_\xi K)^* \cdot (\overline{K}(A \setminus \{y\})^*\overline{K})^* \cdot (K(A \setminus \{y\})^* K)^*
\end{aligned}
$$

Observe that since $K \in \mathrm{Pol}(\mathcal{C}) \setminus \mathcal{C}$, we know that K is $\mathrm{Pol}(\mathcal{C})$-separable from $\overline{K}$ but not $\mathcal{C}$-separable from $\overline{K}$ by Lemma 3.6. Therefore, $\overline{K}(A \setminus \{x\})^*\overline{K}$ is not $\mathcal{C}$-separable from $K(L_\xi \cap (A \setminus \{x\})^*)K$.

Assuming that $L_\xi \cap (A_{\mathsf{val}} \setminus \{x\})^*$ is not $\mathrm{Pol}(\mathcal{C})$-separable from $L_\xi \cap (A_{\mathsf{val}} \setminus \{x\})^*$, we may thus obtain that $L_\varphi \cap A^*_{\mathsf{val}}$ is not $\mathrm{Pol}(\mathcal{C})$-separable from $L'_x \cap A^*_{\mathsf{val}}$.

Conversely, if $L_\xi \cap (A_{\mathsf{val}} \setminus \{x\})^*$ is $\mathrm{Pol}(\mathcal{C})$-separable from $L'_\xi \cap (A_{\mathsf{val}} \setminus \{x\})^*$, we can separate $L_\varphi \cap A^*_{\mathsf{val}}$ from $L'_x \cap A^*_{\mathsf{val}}$ with $\mathrm{Pol}(\mathcal{C})$. Indeed, given $u \in L_\varphi \cap A^*_{\mathsf{val}}$ and $u' \in L'_x \cap A^*_{\mathsf{val}}$, we can either pinpoint factors v, v' of u, u' with $v \in K$ and $v' \in \overline{K}$ or with $v \in L_\xi \cap (A_{\mathsf{val}} \setminus \{x\})^*$ and $v' \in L'_\xi \cap (A_{\mathsf{val}} \setminus \{x\})^*$.

We now prove each direction of Proposition 4.52 in a separate part. First fix a valuation val defined on the free variables of φ.

Inseparation transfer from L_φ, L'_φ to L_ξ, L'_ξ.

By contrapositive, assume that for $y \in \{x, \overline{x}\}$, Sep_y is a language in $\mathrm{Pol}(\mathcal{C})$ separating $L_\xi \cap (A_{\mathsf{val}} \setminus \{\overline{y}\})^*$ from $L'_\xi \cap (A_{\mathsf{val}} \setminus \{\overline{y}\})^*$. We use the two languages Sep_x and $\mathsf{Sep}_{\overline{x}}$ to construct a language $\mathsf{Sep} \in \mathrm{Pol}(\mathcal{C})$ separating $L_\varphi \cap A^*_{\mathsf{val}}$ from $L'_\varphi \cap A^*_{\mathsf{val}}$.

Since $(A_{\mathsf{val}} \setminus \{\overline{y}\})^*$ is alphabet testable and $\mathsf{AT} \subset \mathcal{C}$, we may assume that $\mathsf{Sep}_y \subset (A_{\mathsf{val}} \setminus \{\overline{y}\})^*$. Indeed, otherwise, we may replace Sep_y with $\mathsf{Sep}_y \cap (A_{\mathsf{val}} \setminus \{\overline{y}\})^*$. This language still separates $L_\xi \cap (A_{\mathsf{val}} \setminus \{\overline{y}\})^*$ from $L'_\xi \cap (A_{\mathsf{val}} \setminus \{\overline{y}\})^*$ since these languages are contained in $(A_{\mathsf{val}} \setminus \{\overline{y}\})^*$. We then construct a language $\mathsf{Sep} \in \mathrm{Pol}(\mathcal{C})$ as follows:

$$
\begin{aligned}
\mathsf{Sep} = \{\varepsilon\} & \\
\cup \quad & A^*\#(A^*xA^* \cap A^*\overline{x}A^* \setminus A^*\#A^*)\#K\# \\
\cup \quad & K\#x\mathsf{Sep}_x x\#(A \setminus \overline{x})^* \\
\cup \quad & K\#\overline{x}\mathsf{Sep}_{\overline{x}}\overline{x}\#(A \setminus x)^* \\
\cup \quad & A^*\#(A^*\overline{x}A^* \setminus A^*\#A^*)\#K\#K\#x\mathsf{Sep}_x x\#(A \setminus \overline{x})^* \\
\cup \quad & A^*\#(A^*xA^* \setminus A^*\#A^*)\#K\#K\#\overline{x}\mathsf{Sep}_{\overline{x}}\overline{x}\#(A \setminus x)^*.
\end{aligned}
$$

First note that Sep is constructed from $K \in \mathrm{Pol}(\mathcal{C})$ and alphabet testable languages using marked concatenations and unions. Therefore, $\mathsf{Sep} \in \mathrm{Pol}(\mathcal{C})$. It remains to prove that it is a separator. We divide the proof in two lemmas.

Lemma 4.53. $L_\varphi \cap A^*_{\mathsf{val}} \subset \mathsf{Sep}$.

Proof. Let $w \in L_\varphi \cap A^*_{\mathsf{val}}$. Then we can write $w = u_1\#v_1\#w_1\# \cdots u_k\#v_k\#w_k$ with $u_i, w_i \in K \cap A^*_{\mathsf{val}}$ and $v_i \in (x+\overline{x})L_\xi(x+\overline{x}) \cap A^*_{\mathsf{val}}$ for $1 \leqslant i \leqslant k$.

If $k = 0$, then $w = \varepsilon$ lies in Sep. We thus assume $k \geqslant 1$. If $\overline{x}$ does not appear in w, then $v_1 \in x(L_\xi \cap (A_{\mathsf{val}} \setminus \overline{x})^*)x$ hence it lies in $x\mathsf{Sep}_x x$. Thus, $w \in K\#x\mathsf{Sep}_x x\#(A \setminus \overline{x})^* \subset \mathsf{Sep}$. By symmetry, the same holds when w does not contain x. We may thus assume that w contains both x and $\overline{x}$.

Moreover, since $\#$ is a fresh letter, if v_k contains both x and $\overline{x}$, then we have $v_k \in A^*xA^* \cap A^*\overline{x}A^* \setminus A^*\#A^*$, hence

$$w \in A^*\#(A^*xA^* \cap A^*\overline{x}A^* \setminus A^*\#A^*)\#K\# \subset \mathsf{Sep}.$$

By symmetry, assume that v_k contains x but not $\overline{x}$. Consider the rightmost occurrence of $\overline{x}$. Since K and $\overline{K}$ do not use x nor $\overline{x}$, this occurrence lies in some v_i, and by hypothesis, $i < k$. Then we obtain that $v_{i+1} \in x(L_\xi \cap (A_{\mathsf{val}} \setminus \overline{x})^*)x$, hence $v_{i+1} \in x\mathsf{Sep}_x x$. Moreover, $w_{i+1}\#u_{i+2}\# \cdots \#w_k \in (A \setminus \overline{x})^*$ by definition of i. Therefore, w lies in $A^*\#(A^*\overline{x}A^* \setminus A^*\#A^*)\#K\#K\#x\mathsf{Sep}_x x\#(A \setminus \overline{x})^* \subset \mathsf{Sep}$. ☐

Lemma 4.54. $(L'_\varphi \cap A^*_{\mathsf{val}}) \cap \mathsf{Sep} = \varnothing$.

Proof. By contradiction, assume that there is a word $w \in (L'_\varphi \cap A^*_{\mathsf{val}}) \cap \mathsf{Sep}$. By construction of L'_φ,

$$w = u_1\#v_1\#w_1\# \cdots u_k\#v_k\#w_k\# \cdot w',$$

with $u_i, w_i \in K \cap A^*_{\mathsf{val}}$, $v_i \in (x+\overline{x})L'_\xi(x+\overline{x}) \cap A^*_{\mathsf{val}}$ and $w' \in T_x \cup T_{\overline{x}}$ for $1 \leqslant i \leqslant k$.

By symmetry, we may assume that $w' \in T_x$ and write

$$w' = \overline{u}\#\overline{v}\#\overline{w}\#u'_1\#v'_1\#w'_1\# \cdots u'_\ell\#v'_\ell\#w'_\ell\#,$$

with $\overline{u}, \overline{w} \in \overline{K}$, $u'_i, w'_i \in K$, and $\overline{v}, v'_i \in x(A \setminus \overline{x})^*x$ for $1 \leqslant i \leqslant \ell$.

Note that Sep is defined as a union of several languages. We consider several cases depending on which of these languages contains w.

- We have $w \neq \varepsilon$ since $\varepsilon \notin T_x$.
- If $w \in A^*\#(A^*xA^* \cap A^*\overline{x}A^* \setminus A^*\#A^*)\#K\#$, then either $\ell = 0$ and $\overline{w} \in K$, or $\ell > 0$ and v'_ℓ contains both x and $\overline{x}$. In both cases, we obtain a contradiction.
- If $w \in K\#x\mathsf{Sep}_x x\#(A \setminus \overline{x})^*$, then either $k = 0$ and $\overline{u} \in K$ or $k > 0$ and $v_1 \in x\mathsf{Sep}_x x$. Since $v_1 \in (x+\overline{x})L'_\xi(x+\overline{x})$, there is a word in $L'_\xi \cap \mathsf{Sep}_x$. Recall that $\mathsf{Sep}_x \subset (A \setminus \{\overline{x}\})^*$, hence $L'_\xi \cap \mathsf{Sep}_x = \varnothing$ and we obtain a contradiction. The case $K\#\overline{x}\mathsf{Sep}_{\overline{x}}\overline{x}\#(A \setminus x)^*$ is similar.

- Assume that $w \in A^*\#(A^*\overline{x}A^* \backslash A^*\#A^*)\#K\#K\#x\mathsf{Sep}_x x\#(A\backslash\overline{x})^*$. Thus, both x and $\overline{x}$ appear in w. First observe that $\overline{x}$ may only appear in some v_i. Let v_j be the factor containing the rightmost occurrence of $\overline{x}$. If $j = k$, then $\overline{u} \in K$, a contradiction. Otherwise, $v_{j+1} \in x\mathsf{Sep}_x x$, hence again we have $L'_\xi \cap \mathsf{Sep}_x \neq \varnothing$.

- Assume that $w \in A^*\#(A^*xA^* \backslash A^*\#A^*)\#K\#K\#\overline{x}\mathsf{Sep}_{\overline{x}}\overline{x}\#(A\backslash x)^*$. Then the letter x does not appear in w after the last occurrence of $\overline{x}$. However, the suffix w' contains x but no $\overline{x}$, a contradiction.

In each case, we obtain a contradiction, hence $(L'_\varphi \cap A^*_{\mathsf{val}}) \cap \mathsf{Sep} = \varnothing$. $\square$

Separation transfer from L_φ, L'_φ to L_ξ, L'_ξ

Assume that the language $\mathsf{Sep} \in \mathrm{Pol}(\mathcal{C})$ separates $L_\varphi \cap A^*_{\mathsf{val}}$ from $L'_\varphi \cap A^*_{\mathsf{val}}$. Assume also by contradiction that $L_\xi \cap (A_{\mathsf{val}} \setminus \{y\})^*$ is not $\mathrm{Pol}(\mathcal{C})$-separable from $L'_\xi \cap (A_{\mathsf{val}} \setminus \{y\})^*$ for some $y \in \{x, \overline{x}\}$. By symmetry, we assume that $y = \overline{x}$. We want to reach a contradiction with the fact that Sep separates $L_\varphi \cap A^*_{\mathsf{val}}$ from $L'_\varphi \cap A^*_{\mathsf{val}}$.

Let $\alpha : A^* \to (N, \leqslant_N)$ be the syntactic morphism of Sep. In particular, $\alpha(\mathsf{Sep})$ is upward-closed and $\mathsf{Sep} = \alpha^{-1}(\alpha(\mathsf{Sep}))$. We use the non-separability of $L_\xi \cap (A_{\mathsf{val}} \setminus \{\overline{x}\})^*, L'_\xi \cap (A_{\mathsf{val}} \setminus \{\overline{x}\})^*$ to construct two words $w \in L_\varphi \cap A^*_{\mathsf{val}}$ and $w' \in L'_\varphi \cap A^*_{\mathsf{val}}$ such that $\alpha(w) \leqslant_N \alpha(w')$.

First assume that these words are constructed, and let us conclude the proof of Proposition 4.52. Observe that since $w \in L_\varphi \cap A^*_{\mathsf{val}}$, we have $w \in \mathsf{Sep}$, hence $\alpha(w) \in \alpha(\mathsf{Sep})$. Since Sep is recognized by α, the set $\alpha(\mathsf{Sep})$ is upward closed for $\leqslant_N$. Therefore, $\alpha(w') \in \alpha(\mathsf{Sep})$, hence $w' \in \mathsf{Sep}$. This is a contradiction since $\mathsf{Sep} \cap (L'_\varphi \cap A^*_{\mathsf{val}})$ is empty and contains w'. Therefore, we conclude that $L_\xi \cap (A_{\mathsf{val}} \setminus \{\overline{x}\})^*$ is $\mathrm{Pol}(\mathcal{C})$-separable from $L'_\xi \cap (A_{\mathsf{val}} \setminus \{\overline{x}\})^*$. This concludes the proof of Proposition 4.52.

Construction of w, w'. It remains to construct the words w, w'. Recall that we have:

$$\begin{aligned} L_\varphi &= (K\#(x+\overline{x})L_\xi(x+\overline{x})\#K\#)^*, \\ L'_\varphi &= (K\#(x+\overline{x})L'_\xi(x+\overline{x})\#K\#)^*(T_x + T_{\overline{x}}). \end{aligned}$$

In view of the shape of L_φ, L'_φ, we look for two words w, w' of the form:

$$\begin{aligned} w &= (v\#xux\#v\#)^k, \text{ and} \\ w' &= (v\#xu'x\#v\#)^\ell(v'\#xu'x\#v'\#)(v\#xu'x\#v\#)^\ell. \end{aligned}$$

for some words u, u', v, v' and some integers k, ℓ satisfying the following:

- $u \in L_\xi \cap (A_{\mathsf{val}} \setminus \{\overline{x}\})^*$

- $u' \in L'_\xi \cap (A_{\mathsf{val}} \setminus \{\overline{x}\})^*$
- $v \in K$
- $v' \in \overline{K}$.

To prove that $w \leqslant_N w'$, we want to use Theorem 4.14. We thus take $k = \omega(N) + 1$, $\ell = \omega(N)$, and we want that

$$\begin{aligned}&(\alpha(v\#xux\#v\#), \alpha(v'\#xux\#v'\#)) \text{ is a } \mathcal{C}\text{-pair},\\ &\alpha(v\#xux\#v\#) \leqslant_N \alpha(v\#xu'x\#v'\#) \text{ and}\\ &\alpha(v'\#xux\#v'\#) \leqslant_N \alpha(v'\#xu'x\#v'\#).\end{aligned}$$

By Proposition 4.15, $(\alpha(v\#xux\#v\#), \alpha(v'\#xux\#v'\#))$ is a $\mathcal{C}$-pair as soon as $(\alpha(v), \alpha(v'))$ is a $\mathcal{C}$-pair. Moreover, using that $\leqslant_N$ is compatible with product, the two last conditions are implied by $\alpha(u) \leqslant_N \alpha(u')$. We thus require that u, u', v, v' also satisfy the following:

$$(\alpha(v), \alpha(v')) \text{ is a } \mathcal{C}\text{-pair} \tag{4.1}$$
$$\alpha(u) \leqslant_N \alpha(u') \tag{4.2}$$

Assume that these hypotheses are satisfied and define the following notation:

$$s = \alpha(v\#), \quad s' = \alpha(v'\#), \quad t = \alpha(xux\#) \text{ and } t' = \alpha(xu'x\#).$$

Since $(\alpha(v), \alpha(v'))$ is a $\mathcal{C}$-pair, Proposition 4.15 yields that $(sts, s'ts')$ is also a $\mathcal{C}$-pair. We may thus apply Theorem 4.14: we have

$$\alpha(w) = (sts)^{\omega+1} \leqslant_N (sts)^\omega s'ts'(sts)^\omega$$

Using that $\alpha(u) \leqslant_N \alpha(u')$, we have $t \leqslant_N t'$, hence

$$(sts)^\omega s'ts'(sts)^\omega \leqslant_N (st's)^\omega s't's'(st's)^\omega = \alpha(w').$$

Therefore, we have $\alpha(w) \leqslant_N \alpha(w')$. It remains to construct the words u, u', v, v' satisfying Properties (4.1) and (4.2).

Construction of v, v'. Observe that $K \notin \mathcal{C}$, hence K and is not $\mathcal{C}$-separable from $\overline{K}$. Since $\mathcal{C}$ is closed under quotients, $K\#$ is also not $\mathcal{C}$-separable from $\overline{K}\#$. By Lemma 4.11, we can find $s \in \alpha(K\#)$ and $s' \in \alpha(\overline{K}\#)$ such that (s, s') is a $\mathcal{C}$-pair. We then take $v \in K$ and $v' \in \overline{K}$ such that $v\# \in \alpha^{-1}(s)$ and $v'\# \in \alpha^{-1}(s')$.

Construction of u, u'. The construction of u, u' relies on the following lemma, applied with $L = \mathsf{Sep}$, $L_1 = L_\xi \cap (A_{\mathsf{val}} \setminus \{\overline{x}\})^*$ and $L_2 = L'_\xi \cap (A_{\mathsf{val}} \setminus \{\overline{x}\})^*$.

Lemma 4.55. *Let L_1, L_2 be two languages, and $\alpha_L : A^* \to (M_L, \leqslant_L)$ be the syntactic morphism of a language $L \in \mathrm{Pol}(\mathcal{C})$.*

If L_1 is not $\mathrm{Pol}(\mathcal{C})$-separable from L_2, then there exist $u_1 \in L_1$ and $u_2 \in L_2$ such that $\alpha_L(u_1) \leqslant_L \alpha_L(u_2)$.

The end of this subsection is devoted to the proof of this lemma. Consider the set F defined as the upward closure of $\alpha_L(L_1)$, i.e.

$$F = \{s \in M_L \mid \exists u \in L_1 \text{ s.t. } \alpha_L(u) \leqslant_L s\}$$

Observe that F is upward closed, hence $\alpha_L^{-1}(F)$ is recognized by α_L. Since α_L is the syntactic morphism of L which lies in $\mathrm{Pol}(\mathcal{C})$, Proposition 3.36 ensures that $\alpha_L^{-1}(F) \in \mathrm{Pol}(\mathcal{C})$. Note also that $\alpha_L^{-1}(F)$ contains L_1 by construction.

Since L_1 is not $\mathrm{Pol}(\mathcal{C})$-separable from L_2, we obtain that there is a word u_2 in $\alpha_L^{-1}(F) \cap L_2$. By definition of F, there exists $u_1 \in L_1$ such that $\alpha_L(u_1) \leqslant_L \alpha_L(u_2)$, giving the requested words u_1, u_2.

This ends the proof of Proposition 4.52.

4.3.6 Induction step: universal quantifier

Assume that $\varphi = \forall x \xi(x)$, and assume that there are two languages L_ξ, L'_ξ such that Proposition 4.51 holds for ξ, i.e. for every valuation val defined exactly on the free variables of ξ, we have $\mathsf{val}(\xi) = \top$ if and only if $L_\xi \cap A^*_{\mathsf{val}}$ is **not** $\mathrm{Pol}(\mathcal{C})$-separable from $L'_\xi \cap A^*_{\mathsf{val}}$. Our goal is to use L_ξ, L'_ξ to construct two languages L_φ, L'_φ such that Proposition 4.56 holds. The approach followed in this subsection is roughly the same than in Subsection 4.3.5: Proposition 4.51 follows from the following result.

Proposition 4.56. *For every valuation val defined on the free variables of φ, $L_\varphi \cap A^*_{\mathsf{val}}$ is not $\mathrm{Pol}(\mathcal{C})$-separable from $L'_\varphi \cap A^*_{\mathsf{val}}$ if and only if*

- *$L_\xi \cap (A_{\mathsf{val}} \setminus \{x\})^*$ is not $\mathrm{Pol}(\mathcal{C})$-separable from $L'_\xi \cap (A_{\mathsf{val}} \setminus \{x\})^*$, and*
- *$L_\xi \cap (A_{\mathsf{val}} \setminus \{\overline{x}\})^*$ is not $\mathrm{Pol}(\mathcal{C})$-separable from $L'_\xi \cap (A_{\mathsf{val}} \setminus \{\overline{x}\})^*$.*

Similarly to the existential case, Proposition 4.56 ensures that Proposition 4.51 holds for φ. Indeed, take a valuation val defined on the free variables of φ. Recall that $\mathsf{val}(\varphi) = \top$ if and only if $\mathsf{val}_x(\xi) = \top$ and $\mathsf{val}_{\overline{x}}(\xi) = \top$.

Recall that $A_{\mathsf{val}_x} = A_{\mathsf{val}} \setminus \{\overline{x}\}$ and $A_{\mathsf{val}_{\overline{x}}} = A_{\mathsf{val}} \setminus \{x\}$. Using induction hypothesis, we get that $\mathsf{val}(\varphi) = \top$ if and only if either $L_\xi \cap (A_{\mathsf{val}} \setminus \{x\})^*$ is not $\mathrm{Pol}(\mathcal{C})$-separable from $L'_\xi \cap (A_{\mathsf{val}} \setminus \{x\})^*$ and $L_\xi \cap (A_{\mathsf{val}} \setminus \{\overline{x}\})^*$ is not $\mathrm{Pol}(\mathcal{C})$-separable from $L'_\xi \cap (A_{\mathsf{val}} \setminus \{\overline{x}\})^*$. Using Proposition 4.56, this is equivalent to say that $L_\varphi \cap A^*_{\mathsf{val}}$ is not $\mathrm{Pol}(\mathcal{C})$-separable from $L'_\varphi \cap A^*_{\mathsf{val}}$, and Proposition 4.51 holds.

It remains to construct L_φ, L'_φ and prove Proposition 4.56. We reuse the notation $K, \#, T_x, T_{\overline{x}}$ defined in Subsection 4.3.5. Recall that $\#$ is a fresh letter and K is a language in $\mathrm{Pol}(\mathcal{C}) \setminus \mathcal{C}$ using only fresh letters. We define:

$$L_\varphi = (K\#(x+\overline{x})L_\xi(x+\overline{x})\#K\#)^*,$$
$$L'_\varphi = T_{\overline{x}}(K\#(x+\overline{x})L'_\xi(x+\overline{x})\#K\#)^* T_x.$$

where

$$T_x = (\overline{K}\#x(A\setminus\overline{x})^*x\#\overline{K}\#)\cdot(K\#x(A\setminus\overline{x})^*x\#K\#)^*$$
$$T_{\overline{x}} = (K\#\overline{x}(A\setminus x)^*\overline{x}\#K\#)^*\cdot(\overline{K}\#\overline{x}(A\setminus x)^*\overline{x}\#\overline{K}\#)$$

Note that T_x is the same as in the existential case, but $T_{\overline{x}}$ is a mirrored version of the former $T_{\overline{x}}$. Automata for L_φ and L'_φ are given in Figure 4.3.

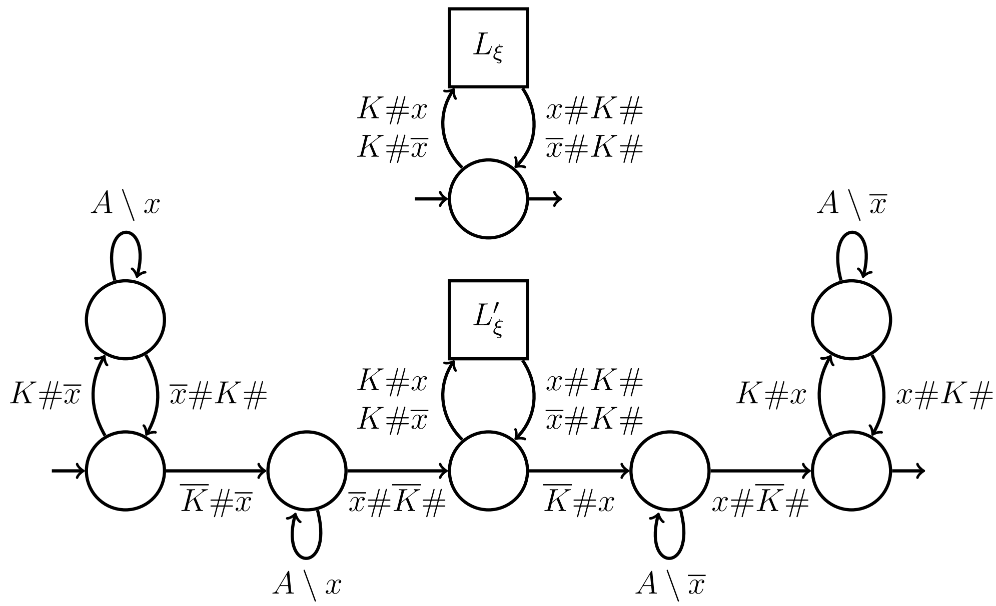


Figure 4.3 – Universal case: Automata for L_φ (above) and L'_φ (below).

Before proving Proposition 4.56, we also give some intuition about this construction. First observe that L'_φ is this time constructed as a concatenation $L'_{\overline{x}}L'_x$ where

$$L'_{\overline{x}} = T_{\overline{x}}(K\#(x+\overline{x})L'_\xi(x+\overline{x})\#K\#)^*$$
$$L'_x = (K\#(x+\overline{x})L'_\xi(x+\overline{x})\#K\#)^* T_x.$$

A second note is that $L_\varphi^2 = L_\varphi$. Therefore, the generic result of Proposition 4.15 ensures that if L_φ is not Pol($\mathcal{C}$)-separable from L_x and from $L_{\overline{x}}$, then L_φ is not Pol($\mathcal{C}$)-separable from L'_φ. Due to the special shape of these languages, we will actually prove the converse statement also holds (which is not true in general).

Finally, observe that L'_x and $L'_{\overline{x}}$ follow a similar construction as before, allowing to use Theorem 4.14 to prove that for $y \in \{x, \overline{x}\}$, $L_\varphi \cap A^*_{\mathsf{val}}$ is not $\mathrm{Pol}(\mathcal{C})$-separable from $L'_y \cap A^*_{\mathsf{val}}$ if and only if $L_\xi \cap (A_{\mathsf{val}} \setminus \{y\})^*$ is not $\mathrm{Pol}(\mathcal{C})$-separable from $L'_\xi \cap (A_{\mathsf{val}} \setminus \{y\})^*$.

We now prove each direction of Proposition 4.56 in a separate part. First fix a valuation val defined on the free variables of φ.

Inseparation transfer from L_φ, L'_φ to L_ξ, L'_ξ.

By contrapositive, assume that, for some $y \in \{x, \overline{x}\}$, there is a language $\mathsf{Sep}_y \in \mathrm{Pol}(\mathcal{C})$ separating $L_\xi \cap (A_{\mathsf{val}} \setminus \{\overline{y}\})^*$ from $L'_\xi \cap (A_{\mathsf{val}} \setminus \{\overline{y}\})^*$. By symmetry, we assume that $y = x$. We use the language Sep_x to construct a language $\mathsf{Sep} \in \mathrm{Pol}(\mathcal{C})$ separating $L_\varphi \cap A^*_{\mathsf{val}}$ from $L'_\varphi \cap A^*_{\mathsf{val}}$.

Since $(A_{\mathsf{val}} \setminus \{\overline{x}\})^*$ is alphabet testable and $\mathsf{AT} \subset \mathcal{C}$, we may assume that $\mathsf{Sep}_x \subset (A_{\mathsf{val}} \setminus \{\overline{x}\})^*$ (up to replacing Sep_x by $\mathsf{Sep}_x \cap (A_{\mathsf{val}} \setminus \{\overline{x}\})^*$). We then construct Sep as follows:

$$\begin{aligned}
\mathsf{Sep} = {} & \{\varepsilon\} \\
\cup \quad & K\#x\mathsf{Sep}_x x\#(A \setminus \overline{x})^* \\
\cup \quad & A^*\#(A^*\overline{x}A^* \setminus A^*\#A^*)\#K\# \\
\cup \quad & A^*\#(A^*\overline{x}A^* \setminus A^*\#A^*)\#K\#K\#x\mathsf{Sep}_x x\#(A \setminus \overline{x})^*.
\end{aligned}$$

Again, Sep is constructed from $K \in \mathrm{Pol}(\mathcal{C})$ and alphabet testable languages using marked concatenations and unions. Therefore $\mathsf{Sep} \in \mathrm{Pol}(\mathcal{C})$. It remains to prove that it is a separator. We separate the proof in two lemmas.

Lemma 4.57. $L_\varphi \cap A^*_{\mathsf{val}} \subset \mathsf{Sep}$.

Proof. Let $w \in L_\varphi \cap A^*_{\mathsf{val}}$. Then $w = u_1\#v_1\#w_1\# \cdots \#u_k\#v_k\#w_k\#$ where $u_i, w_i \in K$ and $v_i \in (x + \overline{x})L_\xi(x + \overline{x})$ for $1 \leqslant i \leqslant k$.

If $k = 0$, then $w = \varepsilon$ and $w \in \mathsf{Sep}$. We may thus assume $k > 0$. If w does not contain $\overline{x}$, then $v_1 \in x\mathsf{Sep}_x x$, hence w in $K\#x\mathsf{Sep}_x x\#(A \setminus \overline{x})^* \subset \mathsf{Sep}$.

We thus assume that $\overline{x}$ appears in w and we denote by v_j the rightmost factor containing $\overline{x}$. If $j = k$, then $w \in A^*\#(A^*\overline{x}A^* \setminus A^*\#A^*)\#K\#$ hence $w \in \mathsf{Sep}$. Otherwise, $j < k$ and $v_{j+1} \in x\mathsf{Sep}_x x$, hence

$$w \in A^*\#(A^*\overline{x}A^* \setminus A^*\#A^*)\#K\#K\#x\mathsf{Sep}_x x\#(A \setminus \overline{x})^* \subset \mathsf{Sep}.$$

In each case, we obtain that $w \in \mathsf{Sep}$. □

Lemma 4.58. $(L'_\varphi \cap A^*_{\mathsf{val}}) \cap \mathsf{Sep} = \varnothing$.

Proof. By contradiction, assume that there is a word $w \in (L'_\varphi \cap A^*_{\mathsf{val}}) \cap \mathsf{Sep}$. By construction of L'_φ, we write $w = w_{\overline{x}} \cdot u_1\#v_1\#w_1\# \ \cdots \ u_k\#v_k\#w_k\# \cdot w_x$ where $w_x \in T_x$, $w_{\overline{x}} \in T_{\overline{x}}$, $u_i, w_i \in K$ and $v_i \in (x + \overline{x})L'_\xi(x + \overline{x})$ for $1 \leqslant i \leqslant k$.

We may also decompose w_x as $\overline{u}\#\overline{v}\#\overline{w}\# \cdot u'_1\#v'_1\#w'_1\# \ \cdots \ u'_\ell\#v'_\ell\#w'_\ell\#$ where $\overline{u}, \overline{w} \in \overline{K}$, $\overline{v}, v'_i \in x(A \setminus \overline{x})^* x$ and $u'_i, w'_i \in K$ for $1 \leqslant i \leqslant \ell$.

We consider several cases, one for each language of the union defining Sep.

- We have $w \neq \varepsilon$ since $\varepsilon \notin T_x$.
- If $w \in K\#x\mathsf{Sep}_x x\#(A \setminus \overline{x})^*$, then $w_{\overline{x}}$ does not contain $\overline{x}$, a contradiction by definition of $T_{\overline{x}}$.
- If $w \in A^*\#(A^*\overline{x}A^* \setminus A^*\#A^*)\#K\#$, then either $\ell = 0$ and $\overline{w} \in K$ or $\ell > 0$ and v'_ℓ contains $\overline{x}$, a contradiction in both cases.
- If $w \in A^*\#(A^*\overline{x}A^* \setminus A^*\#A^*)\#K\#K\#x\mathsf{Sep}_x x\#(A \setminus \overline{x})^*$, then consider the rightmost occurrence of $\overline{x}$ in w. If it is in $w_{\overline{x}}$, then $v_1 \in x\mathsf{Sep}_x x \cap xL'_\xi x$, a contradiction. Otherwise, it is in some v_j. If $j = k$, then $\overline{u} \in K$, otherwise $v_{j+1} \in x\mathsf{Sep}_x x \cap xL'_\xi x$, again a contradiction.

In each case, we obtain a contradiction, hence $(L'_\varphi \cap A^*_{\mathsf{val}}) \cap \mathsf{Sep} = \varnothing$. □

Separation transfer from L_φ, L'_φ to L_ξ, L'_ξ

Assume that the language $\mathsf{Sep} \in \mathrm{Pol}(\mathcal{C})$ separates $L_\varphi \cap A^*_{\mathsf{val}}$ from $L'_\varphi \cap A^*_{\mathsf{val}}$. By contradiction, assume also that for all $y \in \{x, \overline{x}\}$, $L_\xi \cap (A_{\mathsf{val}} \setminus \{y\})^*$ is not $\mathrm{Pol}(\mathcal{C})$-separable from $L'_\xi \cap (A_{\mathsf{val}} \setminus \{y\})^*$. We want to reach a contradiction with the fact that Sep separates $L_\varphi \cap A^*_{\mathsf{val}}$ from $L'_\varphi \cap A^*_{\mathsf{val}}$.

Let $\alpha : A^* \to (N, \leqslant_N)$ be the syntactic morphism of Sep, i.e. $\alpha(\mathsf{Sep})$ is upward-closed and $\mathsf{Sep} = \alpha^{-1}(\alpha(\mathsf{Sep}))$. Similarly to the existential case, we use that, $L_\xi \cap (A_{\mathsf{val}} \setminus \{y\})^*$ is not $\mathrm{Pol}(\mathcal{C})$-separable from $L'_\xi \cap (A_{\mathsf{val}} \setminus \{y\})^*$ for $y \in \{x, \overline{x}\}$ to construct two words $w \in L_\varphi \cap A^*_{\mathsf{val}}$ and $w' \in L'_\varphi \cap A^*_{\mathsf{val}}$ such that $\alpha(w) \leqslant_N \alpha(w')$. Assuming that these two words are constructed, we reach a contradiction since Sep has to contain both w and w'.

Construction of w, w'. It thus remains to construct the words w, w'. Due to the construction of L_φ, L'_φ, we look for two words w, w' of the form:

$$\begin{aligned} w &= (v\#\overline{x}u_{\overline{x}}\overline{x}\#v\#)^k (v\#xu_x x\#v\#)^k, \text{ and} \\ w' &= (v\#\overline{x}u'_{\overline{x}}\overline{x}\#v\#)^\ell (v'\#\overline{x}u'_{\overline{x}}\overline{x}\#v'\#)(v\#\overline{x}u'_{\overline{x}}\overline{x}\#v\#)^\ell \\ &\quad \cdot (v\#xu'_x x\#v\#)^\ell (v'\#xu'_x x\#v'\#)(v\#xu'_x x\#v\#)^\ell. \end{aligned}$$

for some words $u_x, u'_x, u_{\overline{x}}, u'_{\overline{x}}, v, v'$ and some integers k, ℓ satisfying the following: for $y \in \{x, \overline{x}\}$,

- $u_y \in L_\xi \cap (A_{\mathsf{val}} \setminus \{\overline{y}\})^*$
- $u'_y \in L'_\xi \cap (A_{\mathsf{val}} \setminus \{\overline{y}\})^*$
- $v \in K$

- $v' \in \overline{K}$.

Similarly to the existential case, we choose $k = \omega(N) + 1$ and $\ell = \omega(N)$, and we ask for the two additional properties:

- $(\alpha(v), \alpha(v'))$ is a $\mathcal{C}$-pair.
- $\alpha(u_y) \leqslant_N \alpha(u'_y)$ for $y \in \{x, \overline{x}\}$.

These words are obtained using the same approach as in the existential case. First, the existence of $u_x, u'_x, u_{\overline{x}}$ and $u'_{\overline{x}}$ is obtained by applying twice Lemma 4.55: for $y \in \{x, \overline{x}\}$, the words u_y and u'_y come from the lemma applied with $L_\xi \cap (A_{\mathsf{val}} \setminus \{\overline{y}\})^*$ and $L_\xi \cap (A_{\mathsf{val}} \setminus \{\overline{y}\})^*$. Second, the words v, v' are obtained using that $K \notin \mathcal{C}$. We may now conclude the proof.

For $y \in \{x, \overline{x}\}$, denote by $(t_y, t'_y) = (\alpha(yu_y y\#), \alpha(yu'_y y\#))$ and by $(s, s') = (\alpha(v), \alpha(v'))$.

By Proposition 4.15, $(st_y s, s't_y s')$ is a $\mathcal{C}$-pair. Then, applying Theorem 4.14 and using that $\leqslant_N$ is compatible with concatenations, we have

$$(st_y s)^{\omega+1} \leqslant_N (st_y s)^\omega s' t_y s' (st_y s)^\omega \leqslant_N (st'_y s)^\omega s' t'_y s' (st'_y s)^\omega.$$

We thus obtain that

$$(st_{\overline{x}} s)^{\omega+1} (st_x s)^{\omega+1} \leqslant_N (st'_{\overline{x}} s)^\omega s' t'_{\overline{x}} s' (st'_{\overline{x}} s)^\omega \cdot (st'_x s)^\omega s' t'_x s' (st'_x s)^\omega,$$

ensuring that $\alpha(w) \leqslant_N \alpha(w')$, which concludes the proof of Proposition 4.56.

4.4 Extension to infinite words

For now, we only considered a single structure: finite words. However, the logical formalism is not syntactically restricted to considering only finite words. It can thus be transposed without any change in the more generic setting of infinite words, i.e. sequences of letters indexed by $\mathbb{N}$.

Definition 4.59. An *infinite word* over the alphabet A is a sequence of letters of A indexed by $\mathbb{N}$. We denote by A^ω the set of infinite words over A. A *language of infinite words*, or *ω-language* is thus a subset of A^ω.

Remark 4.60. Observe that we cannot concatenate two infinite words. However, we may still construct infinite words using finite ones using the two following operations:

- If $u \in A^*$ and $v \in A^\omega$, then we can concatenate u with v and obtain an infinite word $uv \in A^\omega$.
- If $u \in A^+$, then the infinite concatenation $u^\omega = uu\cdots$ is an infinite word.

As we will see, these operations will be the ones we consider for defining an algebraic structure recognizing ω-languages.

Most of the framework extends to infinite words, up to some slight modifications we illustrate below. This is the goal of the two following remarks. The first one is an important note about the terminology used in this thesis.

Remark 4.61. The languages we consider contain only finite words (i.e., are subsets of A^*), while ω-languages contain only infinite words (i.e., are subsets of A^ω). As we will see in Remark 4.65, we in fact no longer consider languages in A^* but in A^+.

We may say that L is an (ω-)language if L is either a language of finite words or an ω-language. Note in particular that we do not consider "mixed languages", i.e. languages containing both finite and infinite words.

We state a consequence of this remark for classes of languages.

Remark 4.62. For languages of infinite words, we have to redefine the notion of quotients: given an ω-language L, a word u and an infinite word v, we have three kind of quotients. The first two are natural generalizations of the case of finite words, while the third one is specific to infinite words (see [Perrin and Pin, 2004]).

- the left quotient $u^{-1}L$ is the ω-language $\{w \in A^\omega \mid uw \in L\}$.
- the right quotient Lv^{-1} is the language of finite words $\{w \in A^+ \mid wv \in L\}$.
- the quotient $Lu^{-\omega}$ is the language of finite words $\{w \in A^+ \mid (xu) \in L\}$.

Note that the two last operations yield languages in A^+, not in A^*. This is harmless and will be motivated in Remark 4.65.

Since quotients of an ω-language may be languages of finite words, considering classes closed under quotient requires to handle classes containing both languages of finite words and of infinite words at the same time.

In this case, observe that we also have to be careful with closure under Boolean operations. Indeed, since we do not consider mixed languages, these operations have to be understood as restricted to languages of the same type. In other words, the class $\mathcal{C}$ is closed under Boolean operations if the class containing all the languages of finite words (resp. ω-languages) of $\mathcal{C}$ is closed under Boolean operations.

Similarly, the polynomial closure operation is still defined on classes containing ω-languages. However, we have to extend carefully the marked concatenation: to consider the language KaL, we need K to be a language of finite words.

The goal of this section is to generalize both the decidability result of Section 4.2 and the complexity result of Section 4.3 to the setting of infinite words. We begin by the latter, and generalize the PSpace lower bound given by Theorem 4.10 to the setting of infinite words with the following reduction.

Let K_1, K_2 be two languages of finite words and a letter $\#$ not appearing in K_1, K_2. We define $L_1 = K_1\#^\omega$ and $L_2 = K_2\#^\omega$.

Let $\mathcal{C}$ be a class containing AT and closed under quotients. Observe that AT is understood here as a class containing languages of finite words (the Boolean combinations of A^*aA^*) and ω-languages (the Boolean combinations of A^*aA^ω). This extends the initial definition to the setting of infinite words: a language L lies in AT if and only if testing whether $w \in L$ depends only on the alphabet of w (regardless of whether w is finite or infinite).

We claim that the language of finite words K_1 is $\text{Pol}(\mathcal{C})$-separable from K_2 if and only if the ω-language L_1 is $\text{Pol}(\mathcal{C})$-separable from L_2.

- If $\mathsf{Sep} \in \text{Pol}(\mathcal{C})$ separates K_1 from K_2, then the marked concatenation $\mathsf{Sep}\#\#^\omega$ also lies in $\text{Pol}(\mathcal{C})$ since $\#^\omega$ is an alphabet-testable ω-language, thus an ω-language in $\mathcal{C}$. Moreover, this languages separates L_1 from L_2.
- Conversely, if $\mathsf{Sep} \in \text{Pol}(\mathcal{C})$ is an ω-language separating L_1 from L_2, then $\mathsf{Sep}(\#^\omega)^{-1}$ is also a language of $\text{Pol}(\mathcal{C})$ (since it is closed under quotients). This language separates K_1 from K_2.

We thus obtain a LogSpace reduction from the $\text{Pol}(\mathcal{C})$-separation problem for languages on finite words, to the same problem for ω-languages. We thus obtain the following extension of Theorem 4.10.

Theorem 4.63. *Let $\mathcal{C}$ be a positive variety of regular (ω-)languages, containing AT and such that $\mathcal{C} \neq \text{Pol}(\mathcal{C})$. Then the $\text{Pol}(\mathcal{C})$-separation problem is PSpace-hard.*

The rest of this section is dedicated to extending the result of Section 4.2. A first step is to extend the notions of pairs to the setting of infinite words. We begin with a few words on how to extend recognition by monoids to this setting.

4.4.1 Algebraic framework: ω-semigroups

To follow the approach of the previous section, we first need to extend the algebraic framework to the infinite words. The canonical notion to consider is ω-semigroups, an extension of semigroups able to handle infinite products. We follow here the presentation given in [Perrin and Pin, 2004]. In particular, in order to avoid confusions between the ordinal ω and the idempotent power ω of a monoid, we rename the latter as π, according to the convention given in [Perrin and Pin, 2004].

Definition 4.64. An *ω-semigroup* is a pair (S_+, S_ω), where S_+ is a semigroup and S_ω is a set. Moreover, (S_+, S_ω) is equipped with two additional products: a *mixed product* $S_+ \times S_\omega \to S_\omega$ mapping $s, t \in S_+, S_\omega$ to an element st of S_ω, and an *infinite product* $(S_+)^\omega \to S_\omega$ mapping an infinite sequence $s_1, s_2, \dots \in (S_+)^\omega$ to an element $s_1 s_2 \cdots$ of S_ω. We require these products to satisfy all possible forms of associativity:

- for every $r, s, t \in S_+$, we have $(rs)t = r(st)$,
- for every $r, s \in S_+$ and $t \in S_\omega$, we have $(rs)t = r(st)$,
- for every $s_0, s_1, s_2, \ldots \in S_+$, and every increasing sequence of integers $(k_n)_{n\in\mathbb{N}}$, we have $(s_0 \cdots s_{k_0})(s_{k_0+1} \cdots s_{k_1}) \cdots = s_0 s_1 s_2 \cdots$,
- for every $s_0, s_1, s_2, \ldots \in S_+$, we have $s_0(s_1 s_2 \cdots) = (s_0 s_1) s_2 \cdots$,

For $s \in S_+$, we let s^ω be the infinite product $sss \cdots \in S_\omega$.

Remark 4.65. In the case of infinite words, we use ω-semigroups instead of "ω-monoids". This aims to avoid indeterminations (or artificial conventions) regarding the value of 1^ω. This is not restrictive. Indeed, all the previous results apply when replacing monoid morphisms by semigroup morphisms.

Moreover, the classes we consider are always expressive enough to detect the language $\{\varepsilon\}$ reduced to the empty word, so we could also assume that the regular languages we consider do not contain ε, i.e. are subsets of A^+.

A first example of ω-semigroup is the free ω-semigroup (A^+, A^ω), endowed with concatenation.

We say that (S_+, S_ω) is *finite* if both S_+ and S_ω are. Note that even if a given ω-semigroup is finite, it is not clear how to represent the infinite product, since the set of infinite sequences of S_+ is uncountable. However, it has been shown in [Wilke, 1991] that the infinite product is fully determined by the mapping $s \mapsto s^\omega$. This makes it possible to finitely represent any finite ω-semigroup.

Morphisms of ω-semigroups are defined in the natural way, as an extension of morphisms of semigroups: $(\varphi_+, \varphi_\omega)$ is a morphism of ω-semigroups from (S_+, S_ω) to (T_+, T_ω) if:

- φ_+ is a morphism of semigroups from S_+ to T_+,
- φ_ω is an application from S_ω to T_ω,
- the mixed product is preserved: for every $s \in S_+$ and $t \in S_\omega$, we have $\varphi_\omega(st) = \varphi_+(s)\varphi_\omega(t)$.
- the infinite product is preserved: for every sequence $s_1, s_2, \dots \in S_+$, we have $\varphi_\omega(s_1 s_2 \cdots) = \varphi_+(s_1)\varphi_+(s_2) \cdots$.

In particular, observe that any morphism of ω-semigroups $\alpha : (A^+, A^\omega) \to (S_+, S_\omega)$ defines two maps: a semigroup morphism $\alpha_+ : A^+ \to S_+$ and a map $\alpha_\omega : A^\omega \to S_\omega$ (when there is no ambiguity, we shall write $\alpha(w)$ to mean $\alpha_+(w)$ if $w \in A^+$ or $\alpha_\omega(w)$ if $w \in A^\omega$). Therefore, a morphism recognizes both languages of finite words (the languages $\alpha_+^{-1}(F_+)$ for $F_+ \subset S_+$) and ω-languages (the ω-languages $\alpha_\omega^{-1}(F_\omega)$ for $F_\omega \subset S_\omega$). An ω-language is regular if and only if it may be recognized by a morphism into a *finite* ω-semigroup.

Example 4.66. We present two examples of ω-semigroups, based on Example 3.14.

- Recall that $S_+ = \mathbb{Z}/2\mathbb{Z}$ is a monoid, and hence a semigroup. We can extend its structure naively into an ω-semigroup by setting $S_\omega = \{\omega\}$ and defining $0 \cdot \omega = 1 \cdot \omega = 0^\omega = 1^\omega = \omega$.

  In particular, the morphism $\alpha : (a^+, a^\omega) \to (S_+, S_\omega)$ defined by $\alpha(a) = 1$ recognizes the languages of finite words $(aa)^*$ and $a(aa)^*$, and the ω-language a^ω.

- The set $S_+ = \{a, b, ab, ba, aa\}$ endowed with the law presented in Example 3.14 is a semigroup. It is the set obtained by removing the neutral element from the monoid given in Example 3.14.

  Let $S_\omega = \{ab_\omega, ba_\omega, aa_\omega\}$, and define the mixed product as:

$\times$	ab_ω	ba_ω	aa_ω
ab	ab_ω	aa_ω	aa_ω
ba	aa_ω	ba_ω	aa_ω
a	aa_ω	ab_ω	aa_ω
b	ba_ω	aa_ω	aa_ω
aa	aa_ω	aa_ω	aa_ω

  Moreover, we define $a^\omega = b^\omega = (aa)^\omega = aa_\omega$, $(ab)^\omega = ab_\omega$ and $(ba)^\omega = ba_\omega$. Then (S_+, S_ω) is an ω-semigroup. Moreover, the morphism $\alpha : (A^+, A^\omega) \to (S_+, S_\omega)$ defined by $\alpha(a) = a$ and $\alpha(b) = b$ recognizes the ω-languages obtained as Boolean combinations of $(ab)^\omega$ and $(ba)^\omega$.

Similarly to the finite words case, for any regular ω-language L, there exists a canonical morphism $\alpha_L : (A^+, A^\omega) \to (S_+, S_\omega)$ recognizing L. This object is called the *syntactic morphism* of L, and can be computed from any ω-semigroup morphism recognizing L. We refer the reader to [Perrin and Pin, 2004] for the detailed definition of this object.

4.4.2 Computing pairs

The goal of this section is to prove the following result. It generalizes the decidability of Pol(AT)-separation on infinite words proven in [Pierron *et al.*, 2016]. It also extends the result of Section 4.2 to the setting of infinite words.

Theorem 4.67. *Let $\mathcal{C}$ be a finite class containing regular languages of finite words and regular ω-languages, stable under Boolean operations and quotients. Then* $\text{Pol}(\mathcal{C})$*-separation is decidable on infinite words.*

In this section, we fix a class $\mathcal{C}$ satisfying the hypothesis of Theorem 4.67. To prove this theorem, we follow the same approach as in Section 4.2, meaning that we will again compute the set of $\text{Pol}(\mathcal{C})$-pairs for an ω-semigroup morphism recognizing both input ω-languages. The first step is thus to construct this morphism. This can be done with a construction similar to the one on finite words. If $L_0 \subset A^\omega$ is recognized by $\alpha_0 : (A^+, A^\omega) \to (S_+, S_\omega)$ and $L_1 \subset A^\omega$ by $\alpha_1 : (A^+, A^\omega) \to (T_+, T_\omega)$, then L_0 and L_1 are both recognized by $\alpha : (A^+, A^\omega) \to (S_+ \times T_+, S_\omega \times T_\omega)$ with $\alpha(w) = (\alpha_0(w), \alpha_1(w))$.

We thus fix a morphism of ω-semigroups $\alpha : (A^+, A^\omega) \to (S_+, S_\omega)$ recognizing both the ω-languages given as input of the $\text{Pol}(\mathcal{C})$-separation problem. As previously, we may assume that α is surjective and $\mathcal{C}$-compatible: the $\mathcal{C}$-completion construction still holds since $\mathcal{C}$ is finite (as well as the co-domain restriction).

We now look for a generalization of the notion of $\text{Pol}(\mathcal{C})$-pairs when α is an ω-semigroup morphism. This is the goal of the following definition.

Definition 4.68. Let α be an ω-semigroup morphism. The $\mathcal{C}$-pairs for α are given by two sets:

- the set of all $(s, t) \in S_+^2$ such that $\alpha_+^{-1}(s)$ is not $\mathcal{C}$-separable from $\alpha_+^{-1}(t)$, denoted by $\mathbb{P}_\mathcal{C}[\alpha_+]$.
- the set of all $(s, t) \in S_\omega^2$ such that $\alpha_\omega^{-1}(s)$ is not $\mathcal{C}$-separable from $\alpha_\omega^{-1}(t)$, denoted by $\mathbb{P}_\mathcal{C}[\alpha_\omega]$.

We denote by $\mathbb{P}_\mathcal{C}[\alpha]$ the pair $(\mathbb{P}_\mathcal{C}[\alpha_+], \mathbb{P}_\mathcal{C}[\alpha_\omega])$.

Observe that, since $\text{Pol}(\mathcal{C})$-separation is decidable on finite words, the set $\mathbb{P}_{\text{Pol}(\mathcal{C})}[\alpha_+]$ is already known to be computable by Theorem 4.43. This is where we use [Place and Zeitoun, 2017d] (i.e. Section 4.2) as a black box: we show how to construct $\text{Pol}(\mathcal{C})$-pairs for α_ω starting from $\text{Pol}(\mathcal{C})$-pairs for α_+, but we do not need any information about how exactly these pairs are constructed. In particular, we do not need to consider compatible pairs. Computing $\text{Pol}(\mathcal{C})$-pairs for α_ω is the goal of the actual main theorem of this section, stated below.

Theorem 4.69. *Let $\mathcal{C}$ be a finite class containing regular languages and ω-languages, closed under Boolean operations and quotients.*

The set of $\mathrm{Pol}(\mathcal{C})$*-pairs for α_ω consists in all the pairs*

$$(r_1 s_1^\omega, r_2 s_2^\pi t_2)$$

where (r_1, r_2) and (s_1, s_2) are $\mathrm{Pol}(\mathcal{C})$*-pairs for α_+, and $t_2 \in S_\omega$ satisfies $s_1^\omega \leqslant_\mathcal{C} t_2$.*

Observe that the condition $s_1^\omega \leqslant_\mathcal{C} t_2$ is well-defined since α is $\mathcal{C}$-compatible. Before proving Theorem 4.69, we first explain how it gives an algorithm for deciding $\mathrm{Pol}(\mathcal{C})$-separation on infinite words.

Since $\mathcal{C}$ is finite, we can decide whether $u \leqslant_\mathcal{C} v$ for any two given infinite words u, v. Since α is $\mathcal{C}$-compatible, we can also decide whether $s \leqslant_\mathcal{C} t$ for every $(s,t) \in S_\omega^2$ by taking two infinite words u, v such that $\alpha(u) = s$ and $\alpha(v) = t$, and deciding whether $u \leqslant_\mathcal{C} v$. We can thus compute the relation $\leqslant_\mathcal{C}$ on S_ω.

Therefore, since we can also compute the $\mathrm{Pol}(\mathcal{C})$-pairs for α_+, Theorem 4.69 ensures that the $\mathrm{Pol}(\mathcal{C})$-pairs for α_ω are also computable. Using Lemma 4.18, we obtain the following corollary.

Corollary 4.70. *Let $\mathcal{C}$ be a finite class containing regular languages and ω-languages, closed under Boolean operations and quotients.*

Then $\mathrm{Pol}(\mathcal{C})$*-separation is decidable on infinite words.*

The rest of this section is devoted to the proof of Theorem 4.69. We thus define by $\mathsf{Sat}(\alpha)$ the set described in Theorem 4.69, i.e.

$$\begin{aligned}\mathsf{Sat}(\alpha) = \{(r_1 s_1^\omega, r_2 s_2^\pi t_2) \mid & (r_1, r_2) \in \mathbb{P}_{\mathrm{Pol}(\mathcal{C})}[\alpha_+], \\ & (s_1, s_2) \in \mathbb{P}_{\mathrm{Pol}(\mathcal{C})}[\alpha_+], \\ & t_2 \in S_\omega, s_1^\omega \leqslant_\mathcal{C} t_2\}.\end{aligned}$$

The goal is then to prove that $\mathbb{P}_{\mathrm{Pol}(\mathcal{C})}[\alpha_\omega] = \mathsf{Sat}(\alpha)$. We separate this equality in two propositions, and prove each one in a different subsection. While computing $\mathsf{Sat}(\alpha)$ does not require a least fixpoint algorithm, we still use the vocabulary introduced for the case of finite words. The first proposition thus considers the soundness inclusion.

Proposition 4.71. *The algorithm is sound: we have $\mathsf{Sat}(\alpha) \subset \mathbb{P}_{\mathrm{Pol}(\mathcal{C})}[\alpha_\omega]$.*

This proves that every constructed pair is indeed a valid $\mathrm{Pol}(\mathcal{C})$-pair. The other proposition is devoted to the dual result, and states that the algorithm is complete: every $\mathrm{Pol}(\mathcal{C})$-pair is actually obtained by some r_1, r_2, s_1, s_2, t_2. Together, these two results imply Theorem 4.69, and yield an algorithm for deciding $\mathrm{Pol}(\mathcal{C})$-separation for infinite words.

Proposition 4.72. *The algorithm is complete: we have $\mathbb{P}_{\mathrm{Pol}(\mathcal{C})}[\alpha_\omega] \subset \mathsf{Sat}(\alpha)$.*

4.4.3 Soundness of the algorithm

In this subsection, we prove Proposition 4.71. The analogous statement for finite words is proved using Lemmas 4.39, 4.40, 4.41 and 4.42. These lemmas are actually refined versions of Lemmas 4.20 and 4.31 designed to handle Pol($\mathcal{C}$)-compatible pairs.

Since we consider only Pol($\mathcal{C}$)-pairs instead of the compatible ones, we may only consider these two last lemmas. Therefore, to prove soundness of the algorithm, we first have to lift these results to the setting of infinite words. This is done by the following lemmas.

The first lemma generalizes that the relation $\leqslant_\mathcal{C}$ (now defined on $A^+ \cup A^\omega$) behaves well with concatenation. Its proof is actually the same as for Lemma 4.20. As a consequence, we may also mimic the proof of Proposition 4.15 to deduce that $\mathbb{P}_\mathcal{C}[\alpha]$ has an ω-semigroup structure.

Lemma 4.73. *Let $\mathcal{C}$ be a class of (ω-)languages closed under quotients. Let u, u' be two words and v, v' two infinite words such that $u \leqslant_\mathcal{C} u'$ and $v \leqslant_\mathcal{C} v'$. Then $uv \leqslant_\mathcal{C} u'v'$.*

The second result is a counterpart of Lemma 4.31 in the setting of infinite words. It gives an example of non-trivial comparable infinite words for $\leqslant_k$.

Lemma 4.74. *Let p be the period of $\mathcal{C}$, k be an integer and let u be a word and v an infinite word such that $u^\omega \leqslant_\mathcal{C} v$. Then for every m greater than $2^{k+1}-1$, we have*

$$u^\omega \leqslant_k u^{pm} v.$$

Before proving Lemma 4.74, we show how to use it to conclude about Proposition 4.71. We thus take two pairs (r_1, r_2) and (s_1, s_2) in $\mathbb{P}_{\text{Pol}(\mathcal{C})}[\alpha_+]$, as well as $t_2 \in S_\omega$ such that $s_1^\omega \leqslant_\mathcal{C} t_2$. Our goal is to prove that $(r_1(s_1)^\omega, r_2(s_2)^\pi t_2)$ is a pair in $\mathbb{P}_{\text{Pol}(\mathcal{C})}[\alpha_\omega]$.

Recall that $\mathbb{P}_{\text{Pol}(\mathcal{C})}[\alpha_\omega]$ is the intersection of all the $\mathbb{P}_k[\alpha_\omega]$. It is therefore sufficient to prove that $(r_1(s_1)^\omega, r_2(s_2)^\pi t_2)$ lies in each of these sets. Let $k \geqslant 0$. By definition, we need to find two infinite words w_1 and w_2 such that $w_1 \leqslant_k w_2$, $\alpha(w_1) = r_1(s_1)^\omega$ and $\alpha(w_2) = r_2(s_2)^\pi t_2$.

By hypothesis, (r_1, r_2) is a Pol($\mathcal{C}$)-pair for α_+. In particular it is a k-pair, hence we can find $x_1, x_2 \in A^+$ such that $x_1 \leqslant_k x_2$ and $\alpha_+(x_1) = r_1$, $\alpha_+(x_2) = r_2$. Similarly, there exist $y_1, y_2 \in A^+$ such that $y_1 \leqslant_k y_2$, $\alpha(y_1) = s_1$ and $\alpha(y_2) = s_2$.

Moreover, since α is $\mathcal{C}$-compatible, the inequality $s_1^\omega \leqslant_\mathcal{C} t_2$ implies that for every $u, v \in A^\omega$ such that $\alpha(u) = s_1^\omega$ and $\alpha(v) = t_2$, we have $u \leqslant_\mathcal{C} v$. By choosing u as y_1^ω and z as any infinite word in $\alpha^{-1}(t_2)$, we obtain that $y_1^\omega \leqslant_\mathcal{C} z$.

Let p be the period of $\mathcal{C}$. We define $w_1 = x_1(y_1)^\omega$ and $w_2 = x_2(y_2)^{p2^{k+1}\pi} z$. Observe that, by definition, we have $\alpha(w_1) = r_1(s_1)^\omega$ and $\alpha(w_2) = r_2(s_2)^\pi t_2$. Therefore, it remains to prove $w_1 \leqslant_k w_2$.

By Lemma 4.74 applied to $u = y_1$ and $v = z$, we obtain $(y_1)^\omega \leqslant_k (y_1)^{p2^{k+1}\pi}z$. Moreover, using $y_1 \leqslant_k y_2$ and $z \leqslant_k z$ together with Lemma 4.73, we obtain $(y_1)^{p2^{k+1}\pi}z \leqslant_k (y_2)^{p2^{k+1}\pi}z$. Therefore, by transitivity, $(y_1)^\omega \leqslant_k (y_2)^{p2^{k+1}\pi}z$. Finally, we use the inequality $x_1 \leqslant_k x_2$ and Lemma 4.73 to conclude that we have $x_1(y_1)^\omega \leqslant_k x_2(y_2)^{p2^{k+1}\pi}z$, i.e. $w_1 \leqslant_k w_2$.

Therefore, $(\alpha(w_1), \alpha(w_2)) \in \mathbb{P}_{\mathrm{Pol}(\mathcal{C})}[\alpha_\omega]$, which proves the soundness inclusion and concludes the proof of Proposition 4.71.

It remains to prove Lemma 4.74. The proof relies on the recurrence relation on $\leqslant_k$ given by the following generalization of Lemma 4.28 to the case of infinite words. The proof of this generalization is similar to the original proof of Lemma 4.28 in [Place and Zeitoun, 2017c].

Lemma 4.75. *Let k be an integer and w and w' two infinite words. Then $w \leqslant_k w'$ if and only if the two following properties hold:*

- *$w \leqslant_\mathcal{C} w'$*
- *If $k > 0$, for every decomposition $w = uav$ with $a \in A$, we can decompose $w' = u'av'$ in such a way that $u \leqslant_{k-1} u'$ and $v \leqslant_{k-1} v'$.*

Using Lemma 4.75, we can then prove Lemma 4.74 by induction on the integer k.

First consider the case $k = 0$. Note that the relation $\leqslant_0$ coincides with $\leqslant_\mathcal{C}$. Consider a word u and an infinite word v such that $u^\omega \leqslant_\mathcal{C} v$. Let m be a positive integer. By hypothesis, we have $u^\omega \leqslant_\mathcal{C} v$, hence by Lemma 4.73, we have $u^{pm}u^\omega \leqslant_\mathcal{C} u^{pm}v$, initializing the induction.

Assume now that Lemma 4.74 holds for some $k > 0$. Let u be a word and v be an infinite word such that $u^\omega \leqslant_\mathcal{C} v$. Fix an integer $m \geqslant 2^{k+1} - 1$. To prove that $u^\omega \leqslant_k u^{pm}v$, we apply Lemma 4.75.

We can prove that $u^\omega \leqslant_\mathcal{C} u^{pm}v$ similarly to the case $k = 0$. Consider now a decomposition of u^ω as u_1au_2 where $a \in A$, $u_1 \in A^*$ and $u_2 \in A^\omega$. We want to find a decomposition of $u^{pm}v$ as v_1av_2 where $u_1 \leqslant_{k-1} v_1$ and $u_2 \leqslant_{k-1} v_2$.

By construction, the letter a falls into a factor u^p of u^ω. We can thus rewrite the previous decomposition as $u^{p\ell}u_1'au_2'u^\omega$ where u_1', u_2' are finite words such that $u_1'au_2' = u^p$. We separate two cases:

- Assume that $\ell < 2^k - 1$. Then we decompose $u^{pm}v$ as $u^{p\ell}u_1'au_2'u^{p(m-\ell-1)}v$. Define $v_1 = u^{p\ell}u_1' \in A^*$ and $v_2 = u_2'u^{p(m-\ell-1)}v \in A^\omega$. Note that $v_1 = u_1$, hence we have $u_1 \leqslant_{k-1} v_1$. Observe also that $m - \ell - 1 \geqslant 2^k - 1$, hence by induction hypothesis, we have $u^\omega \leqslant_{k-1} u^{p(m-\ell-1)}v$. Moreover, using Lemma 4.73, we thus obtain that $u_2 \leqslant_{k-1} v_2$.
- Conversely, assume that $\ell \geqslant 2^k - 1$. This time, we decompose $u^{pm}v$ as $u^{p(2^k-1)}u_1'au_2'u^{p(m-2^k)}v$, and define $v_1 = u^{p(2^k-1)}u_1'$ and $v_2 = u_2'u^{p(m-2^k)}v$.

By Lemma 4.29, we have $u^{p\ell} \leqslant_{k-1} u^{p(2^k-1)}$. Using Lemma 4.73, we thus obtain $u_1 \leqslant_{k-1} v_1$. Moreover, we have $m - 2^k \geqslant 2^k - 1$, hence by induction hypothesis, we get $u^\omega \leqslant_{k-1} u^{p(m-2^k)}v$. Therefore $u_2 \leqslant_{k-1} v_2$ by Lemma 4.73.

In both cases, we obtain a decomposition of $u^{pm}v$ satisfying the hypothesis of Lemma 4.75. We thus obtain that $u^\omega \leqslant_k u^{pm}v$, ensuring that Lemma 4.74 holds.

4.4.4 Completeness of the algorithm

The end of this section is now devoted to the proof of Proposition 4.72 (corresponding to completeness: all pairs are computed). Before we start the proof, we require an additional result that we will use. The result we need is a standard decomposition lemma stated below.

Lemma 4.76. *Let $\gamma : A^+ \to S$ be a morphism into a finite semigroup S. Then for every infinite word $w \in A^\omega$, there exists an idempotent $e \in S$ and a decomposition $w = u_0u_1u_2u_3\cdots$ of w into infinitely many factors $u_0, u_1, u_2, \cdots \in A^+$ satisfying $\gamma(u_j) = e$ for all $j \geqslant 1$ (there is no constraint on u_0).*

The proof of Lemma 4.76 is standard and is a consequence of Ramsey Theorem over infinite graphs (see [Wilke, 1991] for example).

We may now prove Proposition 4.72.

Proof of Proposition 4.72. To prove the result, we exhibit a number $\ell \geqslant 1$ such that $\mathbb{P}_\ell[\alpha_\omega] \subset \mathsf{Sat}(\alpha)$. Since every Pol($\mathcal{C}$)-pair is an ℓ-pair, this will prove that $\mathbb{P}_{\mathrm{Pol}(\mathcal{C})}[\alpha_\omega] \subset \mathsf{Sat}(\alpha)$.

We begin with the choice of the number $\ell \geqslant 1$. We know from Lemma 4.24 that there exists a number ℓ_+ such that $\mathbb{P}_{\mathrm{Pol}(\mathcal{C})}[\alpha_+] = \mathbb{P}_{\ell_+}[\alpha_+]$. We then define $\ell = \ell_+ + |S_+| + 1$.

It now remains to prove that $\mathbb{P}_\ell[\alpha_\omega] \subset \mathsf{Sat}(\alpha)$. Let $(q, q') \in \mathbb{P}_\ell[\alpha_\omega]$, we have to prove that $(q, q') \in \mathsf{Sat}(\alpha)$. By definition of $\mathsf{Sat}(\alpha)$, this means that we have to find $r_1, r_2, s_1, s_2 \in S_+$ and $t_2 \in S_\omega$ such that

$$\begin{array}{l} (r_1, r_2) \in \mathbb{P}_{\mathrm{Pol}(\mathcal{C})}[\alpha_+] \\ (s_1, s_2) \in \mathbb{P}_{\mathrm{Pol}(\mathcal{C})}[\alpha_+] \\ s_1^\omega \leqslant_{\mathcal{C}} t_2 \end{array} \quad \text{and} \quad \begin{array}{rcl} q & = & r_1(s_1)^\omega \\ q' & = & r_2(s_2)^\pi t_2 \end{array} \tag{4.3}$$

We proceed as follows. First, we use the definition of $\mathbb{P}_\ell[\alpha_\omega]$ to obtain two infinite words w and w' with images q and q' such that $w \leqslant_\ell w'$. We then use the hypothesis $w \leqslant_\ell w'$ together with our decomposition lemma, Lemma 4.28, to split w and w' into factors. Finally, we use this decomposition to find the appropriate r_1, r_2, s_1, s_2 and t such that (4.3) holds.

Using Lemma 4.76 (with α_+ as the morphism γ) we may decompose w as an infinite product $w = u_0u_1u_2\cdots$ ($u_0, u_1, u_2, \ldots \in A^+$) such that $\alpha(u_1) = \alpha(u_2) = \alpha(u_3) = \cdots$ is an idempotent e of S_+.

We now apply Lemma 4.28 m times to the infinite words $w \leqslant_\ell w'$ where $m = |S_+|+1$. This yields a decomposition $w' = u'_0u'_1\cdots u'_{m-1}v$ ($u'_0, u'_1, \ldots, u'_{m-1} \in A^+$ and $v \in A^\omega$) which satisfies the following (recall that $\ell = \ell_+ + m$):

- For all $j \leqslant m-1$, $u_j \leqslant_{\ell_+} u'_j$, and
- $u_pu_{p+1}\cdots \leqslant_{\ell_+} v$.

We may now use the decomposition of w and w' to construct the appropriate r_1, r_2, s_1, s_2 and t_2 such that (4.3) holds.

Since $m = |S_+|+1$, by the pigeonhole principle, we obtain $i < j \leqslant m-1$ such that $\alpha_+(u'_0\cdots u'_i) = \alpha_+(u'_0\cdots u'_j) = \alpha_+(u'_0\cdots u'_i)\alpha_+(u'_{i+1}\cdots u'_j)$. Hence, $\alpha_+(u'_0\cdots u'_i)$ is stable by right multiplication by $\alpha_+(u'_{i+1}\cdots u'_j)$. Iterating this equality, we get

$$\alpha_+(u'_0\cdots u'_i) = \alpha_+(u'_0\cdots u'_i)(\alpha_+(u'_{i+1}\cdots u'_j))^\pi.$$

Let $x_1 = u_0\cdots u_i \in A^+$, $x_2 = u'_0\cdots u'_i \in A^+$, $y_1 = u_{i+1}\cdots u_j \in A^+$ and $y_2 = u'_{i+1}\cdots u'_j \in A^+$. Moreover, we let $r_1 = \alpha_+(x_1)$, $r_2 = \alpha_+(x_2)$, $s_1 = \alpha_+(y_1)$ and $s_2 = \alpha_+(y_2)$. Note that by the equality above, we have

$$r_2 = r_2(s_2)^\pi.$$

Finally, we let $z = u'_{i+1}\cdots u'_m v$ and $t_2 = \alpha_\omega(z)$.

It remains to prove that (4.3) holds. By definition, $s_1 = \alpha_+(u_{i+1}\cdots u_j)$ is the idempotent e, therefore

$$q = \alpha_\omega(w) = r_1(s_1)^\omega.$$

Moreover, we have $w' = x_2z$, therefore,

$$q' = r_2t_2 = r_2(s_2)^\pi t_2.$$

To conclude that (4.3) holds, it remains to prove that $(r_1, r_2), (s_1, s_2) \in \mathbb{P}_{\mathrm{Pol}(\mathcal{C})}[\alpha_+]$ and $s_1^\omega \leqslant_\mathcal{C} t_2$. This is what we do now.

Using Lemma 4.20, we have $x_1 \leqslant_{\ell_+} x_2$ and $y_1 \leqslant_{\ell_+} y_2$. This exactly says that $(r_1, r_2), (s_1, s_2) \in \mathbb{P}_{\ell_+}[\alpha_+]$. Therefore, by choice of ℓ_+, we have $(r_1, r_2), (s_1, s_2) \in \mathbb{P}_{\mathrm{Pol}(\mathcal{C})}[\alpha_+]$.

Moreover, by Lemma 4.28, the inequality $u_{i+1}u_{i+2}\cdots \leqslant_{\ell_+} z$ implies that $u_{i+1}u_{i+2}\cdots \leqslant_\mathcal{C} z$, therefore $s_1^\omega \leqslant_\mathcal{C} z$. This terminates the proof of Proposition 4.72. $\square$

4.5 Conclusion

In this chapter, we study the separation problem for classed defined by polynomial closures. We first presented a simplified version of the generic framework introduced in [Place and Zeitoun, 2017d] designed to handle Pol($\mathcal{C}$)-separation when $\mathcal{C}$ is finite. This framework can be extended to prove decidability of Bool(Pol($\mathcal{C}$))-separation on finite words (see again [Place and Zeitoun, 2014a, 2017d]). The main idea here is to understand how pairs are constructed during the fixpoint algorithm, in order to isolate specific ones characterizing Bool(Pol($\mathcal{C}$)). This approach is also used to obtain complexity results in [Place and Zeitoun, 2018a].

We then presented in Section 4.3 a generic lower bound for the separation problem. As soon as we consider the polynomial closure of classes recognizing alphabets, the problem becomes PSpace-hard. This is a generalization of the result of [Place and Zeitoun, 2018a]: it applies for almost all levels of the hierarchies excepted for the lower ones. Moreover, up to some slight modifications, the reduction we present also applies to unambiguous polynomial closures (a restriction of polynomial closure where we ask for unions to be disjoint and for marked concatenations to be unambiguous), and thus gives a PSpace lower bound for such classes. While it is tight for Pol(AT), we do not know whether this bound is tight for higher levels of the hierarchies, even for the ones with decidable separation.

We believe this is not the case, and conjecture that if separation is decidable for all the levels of an infinite hierarchy, the complexity of separation must (strictly) grow when considering increasing levels of the hierarchy. Indeed, we only used here that we can detect alphabets. When considering higher levels, the classes can distinguish much more properties, which should allow to encode more information and obtain reductions to greater complexity classes.

A final note about complexity comes from the last reduction from [Place and Zeitoun, 2018a], from Pol(AT)-separation to Bool(Pol(AT))-separation. Combined with the previous result, it proves that Bool(Pol(AT))-separation is PSpace-hard. We believe that this reduction can also be generalized into a generic reduction from Pol($\mathcal{C}$)-separation to Bool(Pol($\mathcal{C}$))-separation whenever $\mathcal{C}$ is variety. This would settle a PSpace lower bound for every high enough level of a hierarchy with decidable separation.

In a last part, we presented an extension of the framework to the setting of infinite words. This allowed us to extend the PSpace lower bound to this setting. Note in particular that, while our reduction is only stated for the class Pol($\mathcal{C}$), we did not use any property specific to this class, except that it is closed under quotients, and contains AT. This reduction is actually generic, and proves that for a nice enough class $\mathcal{C}$, the $\mathcal{C}$-separation problem is harder on ω-languages than on languages of finite words.

We also considered decidability of the separation problem for infinite words, and extended a result from [Pierron *et al.*, 2016]: when $\mathcal{C}$ is a finite nice class, then the Pol($\mathcal{C}$)-separation problem is decidable for infinite words.

Many questions remain open regarding infinite words. A first direction would be to investigate which results of [Place and Zeitoun, 2017c] extend to this setting. In particular, the characterization of Pol($\mathcal{C}$) in terms of $\mathcal{C}$ seems to be an interesting result to look for, as a decisive step for extending Theorem 4.2 to infinite words and for finding a generic transfer result from $\mathcal{C}$-separation to Pol($\mathcal{C}$)-membership in the infinite words setting.

Note that it is likely that this transfer result (if it exists) also uses $\mathcal{C}$-separation on finite words. Indeed, considering finite words is useful when studying problems in the setting of infinite words: given an ω-semigroup morphism $\alpha = (\alpha_+, \alpha_\omega)$, the set of pairs for α_ω has no special structure, but the set of pairs for α is an ω-semigroup. It is then natural that separation on finite words plays a role to solve separation on infinite words, as shown by Theorem 4.69.

A final remark is that while the recent results tend to encapsulate many proofs in a unified framework, the classes we introduced are far from being an exhaustive list of interesting classes. First of all, the hierarchies we consider are all finitely-based. While this allows to derive nice properties of each level, there are interesting hierarchies for which there is no finite base, for example the one constructed starting from group languages. In [Place and Zeitoun, 2019], the authors considered this kind of hierarchies, and solved separation for the lower levels. This proves that there is still hope for infinitely-based hierarchies, and thus raises the question of what can be extended in this context.

Conclusion

This thesis presents results from two fields. The first two chapters are devoted to graph theory, and more precisely to the study of graph colorings. In Chapter 1, we use a standard discharging argument to establish a bound on total coloring of planar graphs. In Chapter 2, we color powers of graphs, and we especially focus on squares of planar graphs. We use a variant of the discharging method to characterize cycle obstructions for obtaining a constant difference between upper and lower bounds. The results presented in this thesis are parts of several papers: [Choi *et al.*, 2018], [Bonamy *et al.*, 2019b], [Pierron, 2019], [Pierron *et al.*, 2016] and [Place and Zeitoun, 2018a]. The last one generalizes the result of Section 3.3, while Section 4.3 generalizes another result of this paper.

We recall briefly here some of the main open problems raised in the first two chapters, starting from the ones that seem the most reachable.

Question 5.1. *For every* $k \geqslant 2$*, do all but finitely many graphs* G *satisfy* $\chi(G^k) \leqslant f(k, \Delta(G)) + 1 - k$*? Can we improve this when considering graphs* G *with* $\Delta(G) \geqslant g(k)$*?*

Question 5.2. *Does every planar graph* G *without cycles of length 2 modulo 4 satisfy* $\chi(G^2) \leqslant \Delta(G) + O(1)$*?*

While the class of planar graphs with such restrictions seems quite artificial, it contains the set of all graphs obtained from a bipartite planar multigraph where each edge is subdivided once. In this case, coloring the square of such a graph is equivalent to totally color the initial multigraph. However, this can be done with $\Delta + O(1)$ colors due to [Borodin *et al.*, 1997a]. Therefore, to disprove the statement of Question 5.2, we have to look for some other kind of graphs.

Question 5.3. *Is the square of every* C_4*-free planar graph* $(\Delta+O(1))$*-choosable when* $\Delta \geqslant \Delta_0$*, when* Δ_0 *is less than (say) 100?*

Question 5.4. *For which values of* Δ *every planar graph of maximum degree* Δ *can be totally* $(\Delta + 1)$*-colored? (Open for* $4 \leqslant \Delta \leqslant 8$*.)*

Question 5.5. *Can every planar graph of maximum degree 7 be list totally 9-colored?*

The last questions are special cases of the aforementioned (but still seemingly unreachable) conjectures: the list edge coloring conjecture, its total coloring version (total list coloring conjecture) and their relaxations introduced in [Vizing, 1976].

Conjecture 5.6 ([Vizing, 1976]). *Every simple graph G satisfies $\chi'_\ell(G) \leqslant \Delta(G) + 1$.*

Conjecture 5.7 ([Behzad, 1965; Vizing, 1976]). *Every simple graph G satisfies $\chi''(G) \leqslant \Delta(G) + 2$.*

The second part of this thesis concerns regular languages. The goal here is to understand the expressiveness of some classes of languages. This is done by deciding the so-called membership problems. To this end, we investigate a stronger problem called separation. In Chapter 3, we present these two problems, together with a first complexity result. We show that separation is a more robust problem than membership: its complexity does not depend on the representation of the input languages.

In Chapter 4, we study a special operation: the polynomial closure. We first consider some decidability questions, and present a generic framework to develop algorithms solving $\mathrm{Pol}(\mathcal{C})$-separation when $\mathcal{C}$ is a finite class. We then investigate the complexity of $\mathrm{Pol}(\mathcal{C})$-separation, by proving a PSpace lower bound when $\mathcal{C}$ is a sufficiently expressive class. We finally extend the framework to handle the case of infinite words, and we prove that the previous results also apply to the setting of infinite words.

We recall again some of the open questions we introduced in Chapter 4.

Question 5.8. *Is there a generic transfer result from $\mathcal{C}$-separation to* $\mathrm{Pol}(\mathcal{C})$*-membership in the setting of infinite words?*

Question 5.9. *When the membership/separation problem is decidable for all the levels of a given hierarchy, does it complexity (strictly) increases with the levels?*

Question 5.10. *Can we extend the results we presented to classes that are not necessarily positive varieties?*

Bibliography

Abbott, H.L. and Zhou, B., 1991. On small faces in 4-critical planar graphs. *Ars Combinatoria*, 32:203–207.

Agnarsson, G. and Halldórsson, M.M., 2003. Coloring powers of planar graphs. *SIAM Journal on Discrete Mathematics*, 16(4):651–662.

Almeida, J., 1999. Some algorithmic problems for pseudovarieties. *Publicationes Mathematicae Debrecen*, 54(1):531–552.

Almeida, J., Bartoňová, J., Klíma, O. and Kunc, M., 2015. On decidability of intermediate levels of concatenation hierarchies. In *International Conference on Developments in Language Theory*, pages 58–70. Springer.

Almeida, J., Costa, J.C. and Zeitoun, M., 2008. Pointlike sets with respect to r and j. *Journal of Pure and Applied Algebra*, 212(3):486–499.

Almeida, J. and Klíma, O., 2015. Representations of relatively free profinite semigroups, irreducibility, and order primitivity. *arXiv preprint arXiv:1509.01389*.

Almeida, J. and Zeitoun, M., 1997. The pseudovariety is hyperdecidable. *RAIRO-Theoretical Informatics and Applications*, 31(5):457–482.

Alon, N., 1999. Combinatorial Nullstellensatz. *Combinatorics, Probability and Computing*, 8(1-2):7–29.

Alon, N. and Tarsi, M., 1992. Colorings and orientations of graphs. *Combinatorica*, 12(2):125–134.

Amini, O., Esperet, L. and Van Den Heuvel, J., 2013. A unified approach to distance-two colouring of graphs on surfaces. *Combinatorica*, 33(3):253–296.

Appel, K., Haken, W. *et al.*, 1977. Every planar map is four colorable. Part I: Discharging. *Illinois Journal of Mathematics*, 21(3):429–490.

Arfi, M., 1991. Opérations polynomiales et hiérarchies de concaténation. *Theoretical Computer Science*, 91(1):71–84.

BEHZAD, M., 1965. *Graphs and their chromatic numbers.* PhD thesis, Michigan State University.

BEINEKE, L.W., 1970. Characterizations of derived graphs. *Journal of Combinatorial theory*, 9(2):129–135.

BOJAŃCZYK, M., 2008. The common fragment of ACTL and LTL. In Roberto Amadio, editor, *Foundations of Software Science and Computational Structures, FoSSaCS'08*, volume 4962 of *Lecture Notes in Computer Science*, pages 172–185. Springer-Verlag. doi:10.1007/978-3-540-78499-9_13.
URL http://dx.doi.org/10.1007/978-3-540-78499-9_13

BOJAŃCZYK, M., 2015. Star height via games. In *Proceedings of the 2015 30th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 214–219. IEEE Computer Society.

BOLLOBÁS, B. and HARRIS, A.J., 1985. List-colourings of graphs. *Graphs and Combinatorics*, 1(1):115–127.

BOLLOBÁS, B. and HIND, H.R.F., 1989. A new upper bound for the list chromatic number. *Discrete Mathematics*, 74(1-2):65–75.

BONAMY, M., 2015. Planar graphs with $\Delta \geqslant 8$ are $(\Delta + 1)$-edge-choosable. *SIAM Journal on Discrete Mathematics*, 29(3):1735–1763.

BONAMY, M. and BOUSQUET, N., 2014. Brooks' theorem on powers of graphs. *Discrete Mathematics*, 325:12–16.

BONAMY, M., CRANSTON, D.W. and POSTLE, L., 2019a. Planar graphs of girth at least five are square $(\Delta + 2)$-choosable. *Journal of Combinatorial Theory, Series B*, 134:218–238.

BONAMY, M., LÉVÊQUE, B. and PINLOU, A., 2014. List coloring the square of sparse graphs with large degree. *European Journal of Combinatorics*, 41:128–137.

BONAMY, M., PIERRON, T. and SOPENA, É., 2019b. Every planar graph with $\Delta \geqslant 8$ is totally $(\Delta + 2)$-choosable. *In preparation.*

BORODIN, O.V., 1987. Coupled colorings of graphs on a plane (in Russian). *Metody Diskretnogo Analiza*, 45:21–27.

BORODIN, O.V., 1989. On the total coloring of planar graphs. *Journal für die reine und angewandte Mathematik*, 394:180–185.

BORODIN, O.V., 1990. Generalization of a theorem of Kotzig and a prescribed coloring of the edges of planar graphs. *Mathematical Notes of the Academy of Sciences of the USSR*, 48(6):1186–1190.

Borodin, O.V., 1996a. Structural properties of plane graphs without adjacent triangles and an application to 3-colorings. *Journal of Graph Theory*, 21(2):183–186.

Borodin, O.V., 1996b. To the paper of H.L. Abbott and B. Zhou on 4-critical planar graphs. *Ars Combinatoria*, 43:191–192.

Borodin, O.V., 2013. Colorings of plane graphs: A survey. *Discrete Mathematics*, 313(4):517–539.

Borodin, O.V., Broersma, H.J., Glebov, A. and Van den Heuvel, J., 2002. Stars and bunches in planar graphs. Part II: General planar graphs and colourings. *CDAM Research Report*, 5:2002.

Borodin, O.V., Glebov, A.N., Ivanova, A.O., Neustroeva, T.K. and Tashkinov, V.A., 2004. Sufficient conditions for planar graphs to be 2-distance $(\Delta + 1)$-colourable. *Sibirskie Ehlektronnye Matematicheskie Izvestiya [electronic only]*, 1:129–141.

Borodin, O.V., Glebov, A.N., Jensen, T.R. and Raspaud, A., 2006. Planar graphs without triangles adjacent to cycles of length from 3 to 9 are 3-colorable. *Sibirskie Ehlektronnye Matematicheskie Izvestiya [electronic only]*, 3:428–440.

Borodin, O.V., Glebov, A.N., Raspaud, A. and Salavatipour, M.R., 2005. Planar graphs without cycles of length from 4 to 7 are 3-colorable. *Journal of Combinatorial Theory, Series B*, 93(2):303–311.

Borodin, O.V. and Ivanova, A.O., 2009a. 2-distance $(\Delta + 2)$-coloring of planar graphs with girth six and $\Delta \geqslant 18$. *Discrete Mathematics*, 309(23-24):6496–6502.

Borodin, O.V. and Ivanova, A.O., 2009b. List 2-distance $(\Delta + 2)$-coloring of planar graphs with girth six. *European Journal of Combinatorics*, 30(5):1257–1262.

Borodin, O.V., Kostochka, A.V. and Woodall, D.R., 1997a. List edge and list total colourings of multigraphs. *Journal of Combinatorial Theory, Series B*, 71(2):184–204.

Borodin, O.V., Kostochka, A.V. and Woodall, D.R., 1997b. Total colorings of planar graphs with large maximum degree. *Journal of Graph Theory*, 26(1):53–59.

Borodin, O.V. and Raspaud, A., 2003. A sufficient condition for planar graphs to be 3-colorable. *Journal of Combinatorial Theory, Series B*, 88(1):17–27.

Branco, M.J.J. and Pin, J.-É., 2009. Equations defining the polynomial closure of a lattice of regular languages. In *International Colloquium on Automata, Languages, and Programming*, pages 115–126. Springer.

Brooks, R.L., 1941. On colouring the nodes of a network. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 37, pages 194–197. Cambridge University Press.

Brzozowski, J.A., 1962. Canonical regular expressions and minimal state graphs for definite events. *Mathematical theory of Automata*, 12(6):529–561.

Brzozowski, J.A., 1980. Open problems about regular languages. In *Formal Language Theory: Perspectives and Open Problems*, pages 23–47. Academic Press.

Brzozowski, J.A. and Knast, R., 1978. The dot-depth hierarchy of star-free languages is infinite. *Journal of Computer and System Sciences*, 16(1):37–55.

Bu, Y., Chen, D., Raspaud, A. and Wang, W., 2009. Injective coloring of planar graphs. *Discrete Applied Mathematics*, 157(4):663–672.

Bu, Y., Lu, K. and Yang, S., 2015. Two smaller upper bounds of list injective chromatic number. *Journal of Combinatorial Optimization*, 29(2):373–388.

Büchi, J.R., 1960. Weak second-order arithmetic and finite automata. *Mathematical Logic Quarterly*, 6(1-6):66–92.

Cajori, F., 1918. Origin of the name "mathematical induction.". *The American Mathematical Monthly*, 25(5):197–201.

Calamoneri, T., 2011. The $L(h, k)$-labelling problem: an updated survey and annotated bibliography. *The Computer Journal*, 54(8):1344–1371.

Chang, G.J. and Kuo, D., 1996. The $L(2, 1)$-labeling problem on graphs. *SIAM Journal on Discrete Mathematics*, 9(2):309–316.

Chen, M., Hahn, G., Raspaud, A. and Wang, W., 2012. Some results on the injective chromatic number of graphs. *Journal of Combinatorial Optimization*, 24(3):299–318.

Chetwynd, A. and Häggkvist, R., 1996. An improvement of Hind's upper bound on the total chromatic number. *Combinatorics, Probability and Computing*, 5(2):99–104.

Cho, S. and Huynh, D.T., 1991. Finite-automaton aperiodicity is PSPACE-complete. *Theoretical Computer Science*, 88(1):99–116.

Choi, I., Cranston, D.W. and Pierron, T., 2018. Painting and correspondence coloring of squares of planar graphs with no 4-cycles. *Combinatorica (Accepted with minor revision)*.

Cohen, N. and Havet, F., 2010. Planar graphs with maximum degree $\Delta \geqslant 9$ are $(\Delta + 1)$-edge-choosable—A short proof. *Discrete Mathematics*, 310(21):3049–3051.

Cohen, R.S. and Brzozowski, J.A., 1971. Dot-depth of star-free events. *Journal of Computer and System Sciences*, 5(1):1–16.

Cohen-Addad, V., Hebdige, M., Li, Z., Salgado, E. *et al.*, 2017. Steinberg's conjecture is false. *Journal of Combinatorial Theory, Series B*, 122:452–456.

Cranston, D.W. and Kim, S.-J., 2008. List-coloring the square of a subcubic graph. *Journal of Graph Theory*, 57(1):65–87.

Cranston, D.W. and Rabern, L., 2016. Painting squares in $\Delta^2 - 1$ shades. *The Electronic Journal of Combinatorics*, 23(2):P2–50.

Cranston, D.W. and West, D.B., 2017. An introduction to the discharging method via graph coloring. *Discrete Mathematics*, 340(4):766–793.

Czerwiński, W., Martens, W. and Masopust, T., 2013. Efficient separability of regular languages by subsequences and suffixes. In *Fomin F.V., Freivalds R., Kwiatkowska M., Peleg D. (eds) Automata, Languages, and Programming. ICALP 2013. Lecture Notes in Computer Science*, volume 7966, pages 150–161. Springer, Berlin, Heidelberg.

Diekert, V., Gastin, P. and Kufleitner, M., 2008. A survey on small fragments of first-order logic over finite words. *International Journal of Foundations of Computer Science*, 19(03):513–548.

Diekert, V. and Kufleitner, M., 2011. Fragments of first-order logic over infinite words. *Theory of Computing Systems*, 48(3):486–516.

Dong, W. and Xu, B., 2017. 2-distance coloring of planar graphs without 4-cycles or 5-cycles. *Manuscript*.

Doyon, A., Hahn, G. and Raspaud, A., 2010. Some bounds on the injective chromatic number of graphs. *Discrete Mathematics*, 310(3):585–590.

Dvořák, Z., Nejedlý, P., Škrekovski, R. *et al.*, 2008. Coloring squares of planar graphs with girth six. *European Journal of Combinatorics*, 29(4):838–849.

DVOŘÁK, Z. and POSTLE, L., 2018. Correspondence coloring and its application to list-coloring planar graphs without cycles of lengths 4 to 8. *Journal of Combinatorial Theory, Series B*, 129:38–54.

EGGAN, L.C. *et al.*, 1963. Transition graphs and the star-height of regular events. *The Michigan Mathematical Journal*, 10(4):385–397.

ELGOT, C.C., 1961. Decision problems of finite automata design and related arithmetics. *Transactions of the American Mathematical Society*, 98(1):21–51.

ERDŐS, P., RUBIN, A.L. and TAYLOR, H., 1979. Choosability in graphs. In *Proceedings of West Coast Conference on Combinatorics, Graph Theory and Computing, Congressus Numerantium*, volume 26, pages 125–157.

ESIK, Z. and ITO, M., 2003. Temporal logic with cyclic counting and the degree of aperiodicity of finite automata. *Acta Cybernetica*, 16(1):1–28.

FOLKMAN, J. and FULKERSON, D.R., 1966. Edge colorings in bipartite graphs. Technical report, RAND Corp. Santa Monica, CA.

GLASSER, C. and SCHMITZ, H., 2008. Languages of dot-depth 3/2. *Theory of Computing Systems*, 42(2):256–286.

GONÇALVES, D., 2008. On the $L(p, 1)$-labelling of graphs. *Discrete Mathematics*, 308(8):1405–1414.

GRIGGS, J.R. and YEH, R.K., 1992. Labelling graphs with a condition at distance 2. *SIAM Journal on Discrete Mathematics*, 5(4):586–595.

GRIGGS, J.R. *et al.*, 2009. Graph labellings with variable weights, a survey. *Discrete Applied Mathematics*, 157(12):2646–2658.

GRÖTZSCH, H., 1959. Ein Dreifarbensatz fur dreikreisfreie Netze auf der Kugel. *Wissenschaftliche Zeitschrift / Martin-Luther-Universität, Halle-Wittenberg Mathematisch-naturwissenschaftliche Reihe*, 8:109–120.

HÄGGKVIST, R. and JANSSEN, J., 1997. New bounds on the list-chromatic index of the complete graph and other simple graphs. *Combinatorics, Probability and Computing*, 6(3):295–313.

HAHN, G., KRATOCHVIL, J., ŠIRÁŇ, J. and SOTTEAU, D., 2002. On the injective chromatic number of graphs. *Discrete Mathematics*, 256(1-2):179–192.

HASHIGUCHI, K., 1983. Representation theorems on regular languages. *Journal of computer and system sciences*, 27(1):101–115.

HASHIGUCHI, K., 1988. Algorithms for determining relative star height and star height. *Information and Computation*, 78(2):124–169.

HAVET, F., REED, B. and SERENI, J.-S., 2012. Griggs and Yeh's conjecture and $L(p, 1)$-labelings. *SIAM Journal on Discrete Mathematics*, 26(1):145–168.

HAVET, F., VAN DEN HEUVEL, J., MCDIARMID, C. and REED, B., 2007. List colouring squares of planar graphs. *Electronic Notes in Discrete Mathematics*, 29:515–519.

HENCKELL, K., 1988. Pointlike sets: the finest aperiodic cover of a finite semigroup. *Journal of Pure and Applied Algebra*, 55(1-2):85–126.

HENCKELL, K., RHODES, J. and STEINBERG, B., 2010. Aperiodic pointlikes and beyond. *International Journal of Algebra and Computation*, 20(02):287–305.

HIND, H.R., 1990. An upper bound for the total chromatic number. *Graphs and Combinatorics*, 6(2):153–159.

HIND, H.R.F., 1988. *Restricted edge-colourings.* PhD thesis, University of Cambridge.

HOFFMAN, A.J. and SINGLETON, R.R., 1960. On moore graphs with diameters 2 and 3. *IBM Journal of Research and Development*, 4(5):497–504.

HOFFMAN, A.J. and SINGLETON, R.R., 2003. On moore graphs with diameters 2 and 3. In *Selected Papers Of Alan J Hoffman: With Commentary*, pages 377–384. World Scientific.

HOLYER, I., 1981. The NP-completeness of edge-coloring. *SIAM Journal on Computing*, 10(4):718–720.

JENSEN, T.R. and TOFT, B., 1995. *Graph Coloring Problems. Series in Discrete Mathematics and Optimization.* Wiley-Interscience, John Wiley, New York,.

JONAS, K., 1993. *Graph Coloring Analogues with a Condition at Distance Two: $L(2, 1)$-labellings and List λ-labellings.* PhD thesis, University of South Carolina.

JONES, N.D., 1975. Space-bounded reducibility among combinatorial problems. *Journal of Computer and System Sciences*, 11(1):68–85.

JUVAN, M., MOHAR, B. and ŠKREKOVSKI, R., 1998. List total colourings of graphs. *Combinatorics, Probability and Computing*, 7(2):181–188.

Juvan, M., Mohar, B. and Škrekovski, R., 1999. Graphs of degree 4 are 5-edge-choosable. *Journal of Graph Theory*, 32(3):250–264.

Kahn, J., 1996. Asymptotically good list-colorings. *Journal of Combinatorial Theory, Series A*, 73(1):1–59.

Kirsten, D., 2005. Distance desert automata and the star height problem. *RAIRO-Theoretical Informatics and Applications*, 39(3):455–509.

Kleene, S.C., 1951. Representation of events in nerve nets and finite automata. Technical report, RAND PROJECT AIR FORCE SANTA MONICA CA.

Knast, R., 1983. A semigroup characterization of dot-depth one languages. *RAIRO – Theoretical Informatics and Applications*, 17(4):321–330.

Kostochka, A.V., 1977. The total coloring of a multigraph with maximal degree 4. *Discrete Mathematics*, 17(2):161–163.

Kostochka, A.V., 1996. The total chromatic number of any multigraph with maximum degree five is at most seven. *Discrete Mathematics*, 162(1-3):199–214.

Kowalik, Ł., Sereni, J.-S. and Škrekovski, R., 2008. Total-coloring of plane graphs with maximum degree nine. *SIAM Journal on Discrete Mathematics*, 22(4):1462–1479.

Král', D. and Škrekovski, R., 2003. A theorem about the channel assignment problem. *SIAM Journal on Discrete Mathematics*, 16(3):426–437.

Kufleitner, M. and Walter, T., 2018. Level two of the quantifier alternation hierarchy over infinite words. *Theory of Computing Systems*, 62(3):467–480.

Lam, P.C.B., Shiu, W.C. and Song, Z.M., 2005. The 3-choosability of plane graphs of girth 4. *Discrete Mathematics*, 294(3):297–301.

Lih, K.-W., Wang, W.-F. *et al.*, 2006. Coloring the square of an outerplanar graph. *Taiwanese Journal of Mathematics*, 10(4):1015–1023.

Lužar, B., Škrekovski, R. and Tancer, M., 2009. Injective colorings of planar graphs with few colors. *Discrete Mathematics*, 309(18):5636–5649.

Margolis, S.W. and Pin, J.-É., 1985. Products of group languages. In *International Conference on Fundamentals of Computation Theory*, pages 285–299. Springer.

MASOPUST, T., 2018. Separability by piecewise testable languages is PTime-complete. *Theoretical Computer Science*, 711:109–114.

MCNAUGHTON, R., 1960. *Symbolic logic and automata*. Wright Air Development Division, Air Research and Development Command.

MCNAUGHTON, R. and PAPERT, S.A., 1971. *Counter-Free Automata (MIT research monograph no. 65)*. The MIT Press.

MIAO, L.Y. and FAN, Y.Z., 2014. The distance coloring of graphs. *Acta Mathematica Sinica, English Series*, 30(9):1579–1587.

MOLLOY, M. and REED, B., 1998. A bound on the total chromatic number. *Combinatorica*, 18(2):241–280.

MOLLOY, M. and REED, B., 2000. Near-optimal list colorings. *Random Structures & Algorithms*, 17(3-4):376–402.

MOLLOY, M. and SALAVATIPOUR, M.R., 2005. A bound on the chromatic number of the square of a planar graph. *Journal of Combinatorial Theory, Series B*, 94(2):189–213.

PERRIN, D., 1984. Recent results on automata and infinite words. In *International Symposium on Mathematical Foundations of Computer Science*, pages 134–148. Springer.

PERRIN, D. and PIN, J.-É., 1986. First-order logic and star-free sets. *Journal of Computer and System Sciences*, 32(3):393–406.

PERRIN, D. and PIN, J.-É. (Eds), 2004. *Infinite words: automata, semigroups, logic and games*, volume 141 of *Pure and Applied Mathematics Series*. Elsevier.

PIERRON, T., 2019. A Brooks-like result for graph powers. In *European Conference on Combinatorics, Graph Theory and Applications, Eurocomb 2019*.

PIERRON, T., PLACE, T. and ZEITOUN, M., 2016. Quantifier alternation for infinite words. In *Jacobs B., Löding C. (eds) Foundations of Software Science and Computation Structures. FoSSaCS 2016. Lecture Notes in Computer Science*, volume 9634, pages 234–251. Springer, Berlin, Heidelberg.

PIN, J.-É., STRAUBING, H. and THÉRIEN, D., 1988. Locally trivial categories and unambiguous concatenation. *Journal of Pure and Applied Algebra*, 52(3):297–311.

PIN, J.-É., 1986. *Varieties of formal languages*. Plenum Publishing Co.

PIN, J.-É., 1995. A variety theorem without complementation. *Russian Mathematics (Izvestija vuzov. Matematika)*, 39:80–90.

PIN, J.-É., 1997. Syntactic semigroups. In *Handbook of formal languages*, pages 679–746. Springer.

PIN, J.-É., 2013. An explicit formula for the intersection of two polynomials of regular languages. In *International Conference on Developments in Language Theory*, pages 31–45. Springer.

PIN, J.-É. and STRAUBING, H., 1981. Monoids of upper triangular boolean matrices. In *Semigroups. Structure and Universal AIgebraic Problems*, volume 39, pages 259–272. North-Holland.

PIN, J.-É. and WEIL, P., 1995. Polynomial closure and unambiguous product. In *International Colloquium on Automata, Languages, and Programming*, pages 348–359. Springer.

PIN, J.-É. and WEIL, P., 1997. Polynomial closure and unambiguous product. *Theory of Computing Systems*, 30(4):383–422.

PIN, J.-É. and WEIL, P., 2002. The wreath product principle for ordered semigroups. *Communications in Algebra*, 30(12):5677–5713.

PLACE, T., 2015. Separating regular languages with two quantifiers alternations. In *Proceedings of the 2015 30th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 202–213. IEEE Computer Society. doi:10.1109/LICS.2015.28.

PLACE, T. and ZEITOUN, M., 2014a. Going higher in the first-order quantifier alternation hierarchy on words. In *Esparza J., Fraigniaud P., Husfeldt T., Koutsoupias E. (eds) Automata, Languages, and Programming. ICALP 2014. Lecture Notes in Computer Science*, volume 8573, pages 342–353. Springer, Berlin, Heidelberg.

PLACE, T. and ZEITOUN, M., 2014b. Separating regular languages with first-order logic. In *Proceedings of the Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, CSL-LICS '14, pages 75:1–75:10. Association for Computing Machinery, New York, NY, USA. ISBN 978-1-4503-2886-9. doi:10.1145/2603088.2603098. URL http://doi.acm.org/10.1145/2603088.2603098

PLACE, T. and ZEITOUN, M., 2017a. Adding successor: A transfer theorem for separation and covering. *arXiv preprint arXiv:1709.10052*.

PLACE, T. and ZEITOUN, M., 2017b. The covering problem. *arXiv preprint arXiv:1707.03370*.

PLACE, T. and ZEITOUN, M., 2017c. Generic results for concatenation hierarchies. *Theory of Computing Systems*, pages 1–53.

PLACE, T. and ZEITOUN, M., 2017d. Separation for dot-depth two. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–12. IEEE.

PLACE, T. and ZEITOUN, M., 2018a. The complexity of separation for levels in concatenation hierarchies. *arXiv preprint arXiv:1810.09287*.

PLACE, T. and ZEITOUN, M., 2018b. A generic characterization of Pol(C). *arXiv preprint arXiv:1802.06141*.

PLACE, T. and ZEITOUN, M., 2018c. Separating without any ambiguity. In *45th International Colloquium on Automata, Languages and Programming, ICALP 2018*.

PLACE, T. and ZEITOUN, M., 2019. Separation and covering for group based concatenation hierarchies. *arXiv preprint arXiv:1902.04957*.

PROWSE, A. and WOODALL, D.R., 2003. Choosability of powers of circuits. *Graphs and Combinatorics*, 19(1):137–144.

ROSENFELD, M., 1971. On the total coloring of certain graphs. *Israel Journal of Mathematics*, 9(3):396–402.

SAKAI, D., 1994. Labeling chordal graphs: distance two condition. *SIAM Journal on Discrete Mathematics*, 7(1):133–140.

SANDERS, D.P. and ZHAO, Y., 1995. A note on the three color problem. *Graphs and Combinatorics*, 11(1):91–94.

SANDERS, D.P. and ZHAO, Y., 1999. On total 9-coloring planar graphs of maximum degree seven. *Journal of Graph Theory*, 31(1):67–73.

SCHÜTZENBERGER, M.P., 1965. On finite monoids having only trivial subgroups. *Information and Control*, 8(2):190–194.

SIMON, I., 1975. Piecewise testable events. In *Automata Theory and Formal Languages 2nd GI Conference Kaiserslautern, May 20–23, 1975*, pages 214–222. Springer.

SIPSER, M., 1997. *Introduction to the theory of computation.* PWS Publishing.

STEINBERG, B., 2001. A delay theorem for pointlikes. *Semigroup Forum*, 63(3):281–304. doi:10.1007/s002330010051.
URL https://doi.org/10.1007/s002330010051

STEINBERG, R., 1993. The state of the three color problem. In *Annals of Discrete Mathematics*, volume 55, pages 211–248. Elsevier.

STRAUBING, H., 1981. A generalization of the Schützenberger product of finite monoids. *Theoretical Computer Science*, 13(2):137–150.

STRAUBING, H., 1985. Finite semigroup varieties of the form $V * D$. *Journal of Pure and Applied Algebra*, 36:53–94.

STRAUBING, H., 2018. First-order logic and aperiodic languages: a revisionist history. *ACM SIGLOG News*, 5(3):4–20.

TESSON, P. and THÉRIEN, D., 2002. Diamonds are forever: The variety DA. In *Semigroups, Algorithms, Automata and Languages*, pages 475–499. World Scientific.

THÉRIEN, D., 1981. Classification of finite monoids: the language approach. *Theoretical Computer Science*, 14(2):195–208.

THOMAS, W., 1982. Classifying regular events in symbolic logic. *Journal of Computer and System Sciences*, 25(3):360–376.

THOMASSEN, C., 1994. Every planar graph is 5-choosable. *Journal of Combinatorial Theory, Series B*, 62(1):180–181.

THOMASSEN, C., 1995. 3-list-coloring planar graphs of girth 5. *Journal of Combinatorial Theory, Series B*, 64(1):101–107.

THOMASSEN, C., 2018. The square of a planar cubic graph is 7-colorable. *Journal of Combinatorial Theory, Series B*, 128:192–218.

TRAKHTENBROT, B.A., 1961. Finite automata and the logic of single-place predicates. In *Doklady Akademii Nauk*, volume 140, pages 326–329. Russian Academy of Sciences.

VAN DEN HEUVEL, J. and MCGUINNESS, S., 2003. Coloring the square of a planar graph. *Journal of Graph Theory*, 42(2):110–124.

VAN ROOIJEN, L. and ZEITOUN, M., 2013. The separation problem for regular languages by piecewise testable languages. *arXiv preprint arXiv:1303.2143*.

VIJAYADITYA, N., 1971. On total chromatic number of a graph. *Journal of the London Mathematical Society*, 2(3):405–408.

VIZING, V.G., 1964. On an estimate of the chromatic index of a p-graph (in Russian). *Metody Diskretnogo Analiza*, 3:25–30.

VIZING, V.G., 1965. Critical graphs with given chromatic class (in Russian). *Metody Diskretnogo Analiza*, 5:9–17.

VIZING, V.G., 1976. Vertex colourings with given colours (in Russian). *Metody Diskretnogo Analiza*, 29:3–10.

VOIGT, M., 1993. List colourings of planar graphs. *Discrete Mathematics*, 120(1-3):215–219.

VOIGT, M., 1995. A not 3-choosable planar graph without 3-cycles. *Discrete Mathematics*, 146(1-3):325–328.

WANG, W., 2007. Total chromatic number of planar graphs with maximum degree ten. *Journal of Graph Theory*, 54(2):91–102.

WANG, W. and CAI, L., 2008. Labelling planar graphs without 4-cycles with a condition on distance two. *Discrete Applied Mathematics*, 156(12):2241–2249.

WANG, W.-F. and LIH, K.-W., 2003. Labeling planar graphs with conditions on girth and distance two. *SIAM Journal on Discrete Mathematics*, 17(2):264–275.

WANG, Y., WU, Q. and SHEN, L., 2011. Planar graphs without cycles of length 4, 7, 8, or 9 are 3-choosable. *Discrete Applied Mathematics*, 159(4):232–239.

WEGNER, G., 1977. Graphs with given diameter and a coloring problem. Technical report, University of Dortmund.

WERNICKE, P., 1904. Über den kartographischen vierfarbensatz. *Mathematische Annalen*, 58(3):413–426.

WILKE, T., 1991. An Eilenberg theorem for ∞-languages. In *International Colloquium on Automata, Languages, and Programming*, pages 588–599. Springer.

WONG, S.A., 1996. *Colouring graphs with respect to distance*. PhD thesis, University of Waterloo.

ZHANG, L., 2000. Every planar graph with maximum degree 7 is of class 1. *Graphs and Combinatorics*, 16(4):467–495.

ZHU, H.-Y., LU, X.-Z., WANG, C.-Q. and CHEN, M., 2012. Labeling planar graphs without 4, 5-cycles with a condition on distance two. *SIAM Journal on Discrete Mathematics*, 26(1):52–64.