Deciding classes of regular languages: the covering approach

Thomas Place

LaBRI, Université de Bordeaux, Institut Universitaire de France
tplace@labri.fr

Abstract. We investigate the *membership problem* that one may associate to every class of languages $\mathcal{C}$. The problem takes a regular language as input and asks whether it belongs to $\mathcal{C}$. In practice, finding an algorithm provides a deep insight on the class $\mathcal{C}$. While this problem has a long history, many famous open questions in automata theory are tied to membership. Recently, a breakthrough was made on several of these open questions. This was achieved by considering a more general decision problem than membership: *covering*. In the paper, we investigate how the new ideas and techniques brought about by the introduction of this problem can be applied to get new insight on earlier results. In particular, we use them to give new proofs for two of the most famous membership results: Schützenberger's theorem and Simon's theorem.

Keywords: Regular languages · automata · covering · membership · star-free languages · piecewise testable languages

1 Introduction

Historical context. A prominent question in formal languages theory is to solve the membership problem for classes of regular languages. Given a fixed class $\mathcal{C}$, one must find an algorithm which decides whether an input regular language belongs to $\mathcal{C}$. Such a procedure is called a $\mathcal{C}$-*membership algorithm*. What motivates this question is the deep insight on the class $\mathcal{C}$ that is usually provided by a solution. Intuitively, being able to formulate an algorithm requires a *solid understanding* of all languages contained in the class $\mathcal{C}$. In other words, membership is used as a mathematical tool whose purpose is to analyze classes.

This research effort started with a famous theorem of Schützenberger [36] which describes the class of *star-free languages* (SF). These are the languages that can be expressed by a regular expression using union, concatenation and *complement*, but *not Kleene star*. This is a prominent class which admits natural alternate definitions. For example, the star-free languages are those which can be defined in first-order logic [15] or equivalently in linear temporal logic [11]. Schützenberger's theorem yields an algorithm which decides whether an input regular language is star-free (*i.e.* an SF-membership algorithm). This provides insight on SF not because of the algorithm itself, but rather because of its proof.

Indeed, it includes a generic construction which builds an expression witnessing membership in SF for every input language on which the algorithm answers positively. This result was highly influential and pioneered a very successful line of research. The theorem itself was often revisited [16,8,14,17,41,23,10,7,5,21] and researchers successfully obtained similar results for other prominent classes of languages. Famous examples include the locally testable languages [42,4] or the piecewise testable languages [38]. However, membership is a difficult question and despite years of investigation, there are still many open problems.

Among these open problems, a famous one is the *dot-depth problem*. Brzozowski and Cohen [2] defined a natural classification of the star-free languages: the *dot-depth hierarchy*. Each star-free language is assigned a "complexity level" (called dot-depth) according to the number of alternations between concatenations and complements that are required to define it with an expression. It is known that this hierarchy is strict [3]. Hence, a natural question is whether membership is decidable for each level. This has been a very active research topic since the 70s (see [20,28,32] for surveys). Yet, only the first two levels are known to be decidable so far. An algorithm for dot-depth one was published by Knast in 1983 [13]. Despite a lot of partial results along the way, it took thirty more years to solve the next level: the decidability of dot-depth two was shown in 2014 [26,33]. This situation is easily explained: in practice, getting new membership results always required new conceptual ideas and techniques. In the paper, we are interested in the ideas that led to a solution for dot-depth two. The key ingredient was a new more general decision problem called *covering*.

Covering. The problem was first considered implicitly in [26] and properly defined later in [31]. Given a class $\mathcal{C}$, the $\mathcal{C}$-covering problem is as follows. The input consists in two objects: a regular language L and a finite set of regular languages $\mathbf{L}$. One must decide whether there exists a $\mathcal{C}$-cover $\mathbf{K}$ of L (a finite set of languages in $\mathcal{C}$ whose union includes L) such that no language in $\mathbf{K}$ intersects all languages in $\mathbf{L}$. Naturally, this definition is more involved than the one of membership and it is more difficult to find an algorithm for $\mathcal{C}$-covering than for $\mathcal{C}$-membership. Yet, covering was recently shown to be decidable for many natural classes (see for example [6,25,24,30,35,34]) including the star-free languages [29].

At the time of its introduction, there were two motivations for investigating this new question. First, while harder, covering is also more rewarding than membership: it yields a more robust understanding of the classes. Indeed, a $\mathcal{C}$-membership algorithm only yields benefits for the languages of $\mathcal{C}$: we manage to detect them and to build a description witnessing this membership. On the other hand, a $\mathcal{C}$-covering algorithm applies to *arbitrary* languages. One may view $\mathcal{C}$-covering as an approximation problem: on inputs L and $\mathbf{L}$, we want to over-approximate L with a $\mathcal{C}$-cover while $\mathbf{L}$ specifies what an acceptable approximation is. A second key motivation was the application to the dot-depth hierarchy. It turns out that all recent membership results for this hierarchy rely heavily on covering arguments. More precisely, they are based on techniques that allow to lift covering results for a level in the hierarchy as membership results for a higher level (see [32] for a detailed explanation).

Contribution. In the paper, we are not looking to provide new covering algorithms. Instead, we look at a slightly different question. As we explained, finding an algorithm for $\mathcal{C}$-covering is even harder than for $\mathcal{C}$-membership. Consequently, the recent breakthroughs that were made on this question required developing new ideas, new techniques and new ways to formulate intricate proof arguments. In the paper, we look back at the original membership problem and investigate how these new developments can be applied to get new insight on earlier results. We prove that even if one is only interested in membership, reasoning in terms of "covers" is quite natural and rather intuitive when presenting proof arguments. In particular, $\mathcal{C}$-covers are a very powerful tool for presenting generic constructions which build descriptions of languages in the class $\mathcal{C}$. We illustrate this point by using covers to give new intuitive proofs for two of the most important membership results in the literature: Schützenberger theorem [36] for the star-free languages and Simon's theorem [38] for the picewise testable languages.

Organization of the paper. We first recall standard terminology about regular languages and define membership in Section 2. We introduce covering in Section 3 and explain why reasoning in terms of covers is intuitive and relevant even if one is only interested in membership. We illustrate this point in Section 4 with a new proof of Schützenberger's theorem. Finally, we present a second example in Section 5 with a new proof of Simon's theorem.

2 Preliminaries

In this section, we briefly recall standard terminology about finite words and classes regular languages. Moreover, we introduce the membership problem.

Regular languages. An alphabet is a finite set A. As usual, A^* denotes the set of all words over A, including the empty word ε. For $w \in A^*$, we write $|w| \in \mathbb{N}$ for the *length* of w (*i.e.* the number of letters in w). Moreover, for $u, v \in A^*$, we denote by uv the word obtained by concatenating u and v.

Given an alphabet A, a *language* (over A) is a subset of A^*. Abusing terminology, we shall often denote by u the singleton language $\{u\}$. We lift concatenation to languages: for $K, L \subseteq A^*$, we let $KL = \{uv \mid u \in K \text{ and } v \in L\}$. Finally, we use Kleene star: if $K \subseteq A^*$, K^+ denotes the union of all languages K^n for $n \geq 1$ and $K^* = K^+ \cup \{\varepsilon\}$. In the paper, we only consider *regular languages*. These are the languages that can be equivalently defined by regular expressions, monadic second-order logic, finite automata or finite monoids. We shall use the definition based on monoids which we briefly recall now (see [21] for details).

A *monoid* is a set M endowed with an associative multiplication $(s, t) \mapsto s \cdot t$ (also denoted by st) having a neutral element 1_M. An *idempotent* of a monoid M is an element $e \in M$ such that $ee = e$. It is folklore that for any *finite* monoid M, there exists a natural number $\omega(M)$ (denoted by ω when M is understood) such that s^ω is an idempotent for every $s \in M$. Observe that A^* is a monoid whose multiplication is concatenation (the neutral element is ε). Thus, we may consider monoid morphisms $\alpha : A^* \to M$ where M is an arbitrary monoid. Given such

a morphism and $L \subseteq A^*$, we say that L is *recognized* by α when there exists a set $F \subseteq M$ such that $L = \alpha^{-1}(F)$. A language L is *regular* if and only if it is recognized by a morphism into a *finite* monoid.

Classes. We investigate classes of languages. Mathematically speaking, a *class of languages* $\mathcal{C}$ is a correspondence $A \mapsto \mathcal{C}(A)$ which associates a (possibly infinite) set of languages $\mathcal{C}(A)$ over A to every alphabet A. For the sake of avoiding clutter, we shall often abuse terminology and omit the alphabet when manipulating classes. That is, whenever A is fixed and understood, we directly write $L \in \mathcal{C}$ to indicate that some language $L \subseteq A^*$ belongs to $\mathcal{C}(A)$.

While this is the mathematical definition, in practice, the term "class" is used to indicate that $\mathcal{C}$ is presented in a specific way. Typically, classes are tied to a particular *syntax* used to describe all the languages they contain. For example, the regular languages are tied to regular expressions and monadic second-order logic. Consequently, the classes that we consider in practice are natural and have robust properties that we present now.

A *lattice* is a class $\mathcal{C}$ which is closed under finite union and intersection: for every alphabet A, we have $\emptyset, A^* \in \mathcal{C}(A)$ and for every $K, L \in \mathcal{C}(A)$, we have $H \cup L, H \cap L \in \mathcal{C}(A)$. Moreover, a *Boolean algebra* is a lattice $\mathcal{C}$ which is additionally closed under complement: for every alphabet A and $K \in \mathcal{C}(A)$, we have $A^* \setminus K \in \mathcal{C}(A)$. Finally, we say that a class $\mathcal{C}$ is *quotient-closed* when for every alphabet A, every $L \in \mathcal{C}(A)$ and every $w \in A^*$, the following two languages belong to $\mathcal{C}(A)$ as well:

$$w^{-1}L \stackrel{\text{def}}{=} \{u \in A^* \mid wu \in L\},$$
$$Lw^{-1} \stackrel{\text{def}}{=} \{u \in A^* \mid uw \in L\}.$$

The techniques that we discuss in the paper are meant to be applied for classes that are quotient-closed lattices and contain *only regular languages*. The two examples that we detail are quotient-closed Boolean algebras of regular languages.

Membership. When encountering a new class $\mathcal{C}$, a natural objective is to precisely understand the languages it contains. In other words, we want to understand what properties can be expressed with the syntax defining $\mathcal{C}$. Of course, this is an informal objective. In practice, we rely on a decision problem called membership which we use as a mathematical tool to approach this question.

The problem is parameterized by an arbitrary class of languages $\mathcal{C}$: we speak of $\mathcal{C}$-*membership*. It takes as input a regular language L and asks whether L belongs to $\mathcal{C}$. The key idea is that obtaining an algorithm for $\mathcal{C}$-membership is not possible without a solid understanding of $\mathcal{C}$. In the literature, such an algorithm is also called a *decidable characterization of* $\mathcal{C}$.

Remark 1. We are not only interested in $\mathcal{C}$-membership algorithms themselves but also in their correctness proofs. In practice, the deep insight that we obtain on the class $\mathcal{C}$ comes from these proofs. Typically, the difficult part in such an argument is to prove that a membership is sound: when it answers positively, prove that the input language does belong to $\mathcal{C}$. Typically, this requires a generic

construction for building a syntactic description of the language witnessing its membership in $\mathcal{C}$. □

Finding membership algorithms has been an important quest for a long time in formal languages theory. The solutions that were obtained for important classes are milestones in the theory of regular languages [36,38,13,22,40,33]. In the paper, we prove two of them: Schützenberger's theorem [36] and Simon's theorem [38]. We frame these proofs using a new formalism based on a more general problem which was recently introduced [31]: *covering*.

3 The covering problem

The covering problem generalizes membership. It was first considered implicitly in [26,27] and was later formalized in [31] (along with a detailed framework designed for handling it). At the time, its introduction was motivated by two reasons. First, an algorithm for covering is usually more rewarding than an algorithm for membership as the former provides more insight on the investigated class of languages. Second, covering was introduced as a key ingredient for handling difficult membership questions. For several important classes, membership is effectively reducible to covering for another simpler class. Recently, this idea was applied to prominent hierarchies of classes called "concatenation hierarchies" (see the surveys [28,32] for details on these results).

In the paper, we are interested in covering for a slightly different reason. In particular, we do not present any covering algorithm. Instead, we look at how the new ideas that were recently introduced with covering in mind can be applied in the simpler membership setting. It turns out that even for the early membership results, reasoning in terms of covers is quite natural and allows to present arguments in a very intuitive way. We manage to formulate new proof arguments for two famous membership algorithms.

We first define covering and explain why it generalizes membership as a decision problem. Then, we come back to membership and briefly recall the general approach that is usually followed in order to handle it. We show that this approach can actually be formulated in a convenient and natural way with covering. For the sake of avoiding clutter, we fix an arbitrary alphabet A for the presentation: all languages that we consider are over A.

3.1 Definition

Similarly to membership, covering is parameterized by an arbitrary class of languages $\mathcal{C}$: we speak of $\mathcal{C}$-*covering*. It is designed with the same objective in mind: it serves as a mathematical tool for investigating the class $\mathcal{C}$.

For a class $\mathcal{C}$, the $\mathcal{C}$-covering takes a language L and a *finite set* of languages $\mathbf{L}$ as input. It asks whether there exists a $\mathcal{C}$-*cover of* L which is *separating for* $\mathbf{L}$. Let us first define these two notions.

Given a language L, a *cover of* L is a **finite** set of languages $\mathbf{K}$ such that $L \subseteq \bigcup_{K \in \mathbf{K}} K$. Additionally, given some class $\mathcal{C}$, a $\mathcal{C}$-*cover of* L is a cover $\mathbf{K}$ of L such that every $K \in \mathbf{K}$ belongs to $\mathcal{C}$.

Moreover, given two finite sets of languages $\mathbf{K}$ and $\mathbf{L}$, we say that $\mathbf{K}$ is *separating for* $\mathbf{L}$ if for every $K \in \mathbf{K}$, there exists $L \in \mathbf{L}$ which satisfies $K \cap L = \emptyset$. In other words, there exists no language in $\mathbf{K}$ which intersects all languages in $\mathbf{L}$. Given a class $\mathcal{C}$, the $\mathcal{C}$-covering problem is now defined as follows:

INPUT: A regular language L and a finite set of regular languages $\mathbf{L}$.
OUTPUT: Does there exist a $\mathcal{C}$-cover of L which is separating for $\mathbf{L}$?

A simple observation is that covering generalizes another well-known decision problem called *separation*. Given a class $\mathcal{C}$ and two languages L_1 and L_2, we say that L_1 is $\mathcal{C}$-*separable* from L_2 when there exists a third language $K \in \mathcal{C}$ such that $L_1 \subseteq K$ and $K \cap L_2 = \emptyset$. We have the following lemma (see [31] for a proof).

Lemma 2 *Let $\mathcal{C}$ be a lattice and L_1, L_2 two languages. Then L_1 is $\mathcal{C}$-separable from L_2, if and only if there exist a $\mathcal{C}$-cover of L_1 which is separating for $\{L_2\}$.*

Lemma 2 proves that $\mathcal{C}$-covering generalizes $\mathcal{C}$-membership as a decision problem. Indeed, given as input a regular language L, it is immediate that L belongs to $\mathcal{C}$ if and only if L is $\mathcal{C}$-separable from $A^* \setminus L$ (which is also regular). Thus, there exists an effective reduction from $\mathcal{C}$-membership to $\mathcal{C}$-covering.

Yet, this not the only connection between membership and covering. More importantly, this is not how we use covering in the paper. While each membership algorithm existing in the literature is based on unique ideas (specific to the class under investigation), most of them are formulated and proved within a standard common framework. It turns out that this framework boils down to a particular kind of covering question: this is the property that we shall exploit in the paper.

3.2 Application to membership

We first summarize the standard general approach that is commonly used to handle membership questions and formulate solutions. Historically, this approach was initiated by Schützenberger who applied it to obtain the first known membership algorithm [36] (for the class of star-free languages). We shall detail and prove this result in Section 4.

The syntactic approach. Obtaining a membership algorithm for a given class $\mathcal{C}$ is intuitively hard, as it requires to decide a semantic property which may not be apparent on the piece of syntax that defines the input regular language L (be it a regular expression, an automaton or a monoid morphism). To palliate this issue, the syntactic approach relies on the existence of a *canonical recognizer* for any given regular language. The idea is that while belonging to $\mathcal{C}$ may not be apparent on an arbitrary syntax for L, it should be apparent on a canonical representation of L. Typically, the *syntactic morphism* of L serves as this canonical representation. As the name suggests, this object is a canonical morphism into a

finite monoid which recognizes L (and can be computed from any representation of L).

Let us first define the syntactic morphism properly. Consider a language L. One may associate a canonical equivalence relation $\equiv_L$ over A^* to L. Given two words $u, v \in A^*$, we write,

$$u \equiv_L v \qquad \text{if and only if} \qquad \text{for every } x, y \in A^*,\ xuy \in L \Leftrightarrow xvy \in L$$

Clearly, $\equiv_L$ is an equivalence relation and one may verify that it is a *congruence for word concatenation*: for every $u, v, u', v' \in A^*$, if $u \equiv_L v$ and $u' \equiv_L v'$, then $uu' \equiv_L vv'$. Consequently, the quotient set $A^*/\equiv_L$ is a monoid called the *syntactic monoid of* L. Moreover, the map $\alpha : A^* \to A^*/\equiv_L$ which maps each word to its $\equiv_L$-class is a monoid morphism called the *syntactic morphism* of L. In particular, this morphism recognizes the language L: $L = \alpha^{-1}(F)$ where F is the set of all $\equiv_L$-classes which intersect L. It is well-known and simple to verify that L is regular if and only if its syntactic monoid is *finite*. Moreover, in that case, one may compute the syntactic morphism of L from any representation of L (such as an automaton or an arbitrary monoid morphism recognizing L).

We are ready to present the key result behind the syntactic approach: for every quotient-closed Boolean algebra $\mathcal{C}$, membership of an arbitrary regular language in $\mathcal{C}$ depends only on its syntactic morphism. This claim is formalized with the following standard result.

Proposition 3 *Let $\mathcal{C}$ be a quotient-closed Boolean algebra, L a regular language and α its syntactic morphism. Then L belongs to $\mathcal{C}$ if and only if every language recognized by α belongs to $\mathcal{C}$.*

Proof. The right to left implication is immediate since L is recognized by its syntactic morphism. We concentrate on the converse one. Assume that $L \in \mathcal{C}$. We show that every language recognized by α belongs to $\mathcal{C}$ as well. By definition, these languages are exactly the unions of $\equiv_L$-classes. Thus, since $\mathcal{C}$ is closed under union, it suffices to show that every $\equiv_L$-class belongs to $\mathcal{C}$. Observe that the definition of $\equiv_L$ can be reformulated as follows. Given $u, v \in A^*$, we have,

$$u \equiv_L v \quad \text{if and only if} \quad u \in x^{-1}Ly^{-1} \Leftrightarrow v \in x^{-1}Ly^{-1} \text{ for every } x, y \in A^*.$$

Let $x, y \in A^*$. Since L is recognized by α, it is clear that whether some word $w \in A^*$ belongs to $x^{-1}Ly^{-1}$ depends only on its image $\alpha(w)$. In other words, $x^{-1}Ly^{-1}$ is recognized by α. Moreover, since L is regular, its syntactic monoid is finite which implies that α recognizes finitely many languages. Thus, while there are infinitely many words $x, y \in A^*$, there are finitely many languages $x^{-1}Ly^{-1}$.

Altogether, we obtain that every $\equiv_L$-class is a *finite* Boolean combination of languages $x^{-1}Ly^{-1}$ where $x, y \in A^*$. Since $L \in \mathcal{C}$ and $\mathcal{C}$ is quotient-closed, every such language belongs to $\mathcal{C}$. Hence, since $\mathcal{C}$ is a Boolean algebra, we conclude that every $\equiv_L$-class belongs to $\mathcal{C}$, completing the proof. □

Proposition 3 implies that membership of a regular language L in some fixed quotient-closed Boolean algebra is equivalent to some property of an algebraic

abstraction of L: its syntactic morphism. In particular, this is independent from the accepting set $F = \alpha(L)$. By itself, this is a simple result. Yet, it captures the gist of the syntactic approach.

Naturally, the proposition tells nothing about the actual the property on the syntactic morphism that one should look for. This question is specific to each particular class $\mathcal{C}$: one has to find the right decidable property characterizing $\mathcal{C}$.

Remark 4. This may seem counterintuitive. We replaced the question of deciding whether a *single language* belongs to the class $\mathcal{C}$ by an intuitively harder one: deciding whether *all languages* recognized by a given monoid morphism belong to $\mathcal{C}$. The idea is that the set of languages recognized by a morphism has a structure which can be exploited in membership arguments. □

Remark 5. Proposition 3 is restricted quotient-closed Boolean algebras. This excludes quotient-closed lattices that are not closed under complement. One may generalize the syntactic approach to such classes (as done by Pin [19]). We do not discuss this as our two examples are quotient-closed Boolean algebras. □

Back to covering. We proved that for every quotient-closed Boolean algebra $\mathcal{C}$, the associated membership problem boils down to deciding whether *all languages* recognized by an input morphism belong to $\mathcal{C}$. It turns out that this new question is a particular instance of $\mathcal{C}$-covering. In order to explain this properly, we require a last definition.

Consider a morphism $\alpha : A^* \to M$ into a finite monoid M and a finite set of languages $\mathbf{K}$. We say that $\mathbf{K}$ is *confined* by α if it is separating for the set $\{\alpha^{-1}(M \setminus \{s\}) \mid s \in M\}$. The following fact can be verified from the definitions and reformulates this property in a way that is easier to manipulate.

Fact 6 *Let $\alpha : A^* \to M$ be a morphism into a finite monoid and $\mathbf{K}$ a finite set of languages. Then $\mathbf{K}$ is confined by α if and only if for every $K \in \mathbf{K}$, there exists $s \in M$ such that $K \subseteq \alpha^{-1}(s)$.*

Proof. By definition $\mathbf{K}$ is confined by α if and only if for every $K \in \mathbf{K}$, there exists $s \in M$ such that $K \cap \alpha^{-1}(M \setminus \{s\}) = \emptyset$. Since $\alpha^{-1}(M \setminus \{s\}) = A^* \setminus \alpha^{-1}(s)$, the fact follows. □

We show that given a lattice $\mathcal{C}$ and a morphism $\alpha : A^* \to M$ into a finite monoid, all languages recognized by α belong to $\mathcal{C}$ if and only if there exists a $\mathcal{C}$-cover of A^* which is confined by α. The latter question is a particular case of $\mathcal{C}$-covering. In fact, we prove a slightly more general result that we shall need later when dealing with our two examples.

Proposition 7 *Let $\mathcal{C}$ be a lattice, $\alpha : A^* \to M$ a morphism into a finite monoid and $H \in \mathcal{C}$ a language. The two following properties are equivalent:*

1. *For every language L recognized by α, we have $L \cap H \in \mathcal{C}$.*
2. *There exists a $\mathcal{C}$-cover of H which is confined by α.*

Proof. Assume first that $L \cap H \in \mathcal{C}$ for every language L recognized by α. We define $\mathbf{K} = \{\alpha^{-1}(s) \cap H \mid s \in M\}$. Clearly, $\mathbf{K}$ is a cover of H and it is a $\mathcal{C}$-cover by hypothesis. Moreover, it is clear from Fact 6 that $\mathbf{K}$ is confined by α.

For the converse direction, assume that there exists a $\mathcal{C}$-cover $\mathbf{K}$ of H which is confined by α. Let L be a language recognized by α, we show that,

$$L \cap H = \left(\bigcup_{\{K \in \mathbf{K} \mid K \cap L \neq \emptyset\}} K \right) \cap H$$

This implies that $L \cap H \in \mathcal{C}$ since $H \in \mathcal{C}$, every language in $\mathbf{K}$ belongs to $\mathcal{C}$ and $\mathcal{C}$ is a lattice. The left to right inclusion is immediate since $\mathbf{K}$ is a cover of H. We prove the converse one. Let $K \in \mathbf{K}$ such that $K \cap L \neq \emptyset$, we show that $K \cap H \subseteq L \cap H$. Let $u \in K \cap H$. Consider $v \in K \cap L$ (which is nonempty by definition of K). Since $u, v \in K$ and $\mathbf{K}$ is confined by α, we have $\alpha(u) = \alpha(v)$ by Fact 6. Thus, since $v \in L$ and L is recognized by α, it follows that $u \in L$, concluding the proof: we obtain $K \cap H \subseteq L \cap H$. □

Let us combine Proposition 3 and Proposition 7. When put together, they imply that for every quotient-closed Boolean algebra $\mathcal{C}$, a regular language L belongs to $\mathcal{C}$ if and only if there exists a $\mathcal{C}$-cover of A^* which is confined by the syntactic morphism of L.

The key point is that this formulation is very convenient when writing proof arguments. As we explained in Remark 1, the technical core of membership proofs consists in generic constructions which build descriptions of languages in $\mathcal{C}$. It turns out that building a $\mathcal{C}$-cover which is confined by some input morphism is an objective that is much easier to manipulate than directly proving that all languages recognized by the morphism belong to $\mathcal{C}$. We illustrate this point in the next section with new proofs for two well-known membership algorithms: the star-free languages and the piecewise testable languages.

4 Star-free languages and Schützenberger's theorem

We now illustrate the discussion of the previous section with a first example: Schützenberger's theorem [36]. This result is important as it started the quest for membership algorithms. It provides such an algorithm for a very famous class: the *star-free languages* (SF). Informally, these are the languages which can be defined by a regular expression in which the Kleene star is disallowed (hence the name "star-free") but a new operator for the complement operation is allowed instead. This class is important as it admits several natural alternate definitions. For example, the star-free languages are those which can be defined in first-order logic [15] or equivalently in linear temporal logic [11].

Schützenberger's theorem states an algebraic characterization of SF: a regular language is star-free if and only if its syntactic monoid is *aperiodic*. This yields an algorithm for SF-membership as aperiodicity is a decidable property of finite monoids. Historically, Schützenberger's theorem was the first result of

its kind. It motivated the systematic investigation of the membership problem for important classes of languages. It is often viewed as one of the most important results of automata theory. This claim is supported by the number of times this theorem has been revisited over the years and the wealth of existing proofs [16,8,14,17,41,23,10,7,5,21].

In this section, we present our own proof, based on SF-covers. Let us point out that while the formulation is new, the original ideas behind the argument can be traced back to the proof of Wilke [41]. We first recall the definition of the star-free languages. Then, we state the theorem properly and present the proof.

4.1 Definition

Let us define the class of star-free languages (SF). For every alphabet A, $\mathrm{SF}(A)$ is the least set containing $\emptyset$ and all singletons $\{a\}$ for $a \in A$, which is closed under union, complement and concatenation. That is, for every $K, L \in \mathrm{SF}(A)$, the languages $K \cup L$, $A^* \setminus K$ and KL belong to $\mathrm{SF}(A)$ as well.

Example 8. For every sub-alphabet $B \subseteq A$, we have $B^* \in \mathrm{SF}(A)$. Indeed, by closure under complement, $A^* = A^* \setminus \emptyset \in \mathrm{SF}(A)$. We then get $A^*aA^* \in \mathrm{SF}(A)$ by closure under concatenation. Finally, this yields,

$$B^* = A^* \setminus \left(\bigcup_{a \in A \setminus B} A^*aA^* \right) \in \mathrm{SF}(A)$$

Another standard example is $(ab)^*$ (where a, b are two *distinct* letters of A). Indeed, $(ab)^*$ is the complement of $bA^* \cup A^*aaA^* \cup A^*bbA^* \cup A^*a$ (provided that $A = \{a, b\}$) which is clearly star-free. □

By definition, SF is a Boolean algebra and one may verify that it is quotient-closed (the details are left to the reader). We complete the definition with a standard property that we require to prove the "easy" direction of Schützenberger's theorem (every star-free language has an aperiodic syntactic monoid). Another typical application of this property is to show that examples of languages are *not* star-free. For example, $(AA)^*$ (words with even length) is not star-free since since it does not satisfy the following lemma.

Lemma 9 *Let A be an alphabet and $L \in \mathrm{SF}(A)$. There exists a number $k \geq 1$ such that for every $\ell \geq k$ and $w \in A^*$, we have $w^\ell \equiv_L w^{\ell+1}$.*

Proof. We proceed by structural induction on the definition of L as a star-free language. When $L = \emptyset$, it is clear that the lemma holds for $k = 1$. When $L = \{a\}$ for $a \in A$, one may verify that the lemma holds for $k = 2$. We turn to the inductive cases. Assume first that $L = L_1 \cup L_2$ where $L_1, L_2 \in \mathrm{SF}$ are simpler languages. Induction yields $k_1, k_2 \geq 1$ such that for $i = 1, 2$, if $\ell \geq k_i$ and $w \in A^*$, we have $w^\ell \equiv_{L_i} w^{\ell+1}$. Hence, the lemma holds for $k = \max(k_1, k_2)$ in that case. We turn to complement: $L = A^* \setminus H$ where $H \in \mathrm{SF}$ is a simpler language.

By induction, we get $h \geq 1$ such that for every $w \in A^*$ and $\ell \geq h$, we have $w^\ell \equiv_H w^{\ell+1}$. Clearly, the lemma holds for $k = h$.

We now consider concatenation: $L = L_1 L_2$ where $L_1, L_2 \in \mathrm{SF}$ are simpler languages. Induction yields $k_1, k_2 \geq 1$ such that for $i = 1, 2$, if $\ell \geq k_i$ and $w \in A^*$, we have $w^\ell \equiv_{L_i} w^{\ell+1}$. Let m be the maximum between k_1 and k_2. We prove that the lemma holds for $k = 2m + 1$. Let $w \in A^*$ and $\ell \geq k$, we have to show that $w^\ell \equiv_L w^{\ell+1}$, *i.e.* $xu^\ell y \in L \Leftrightarrow xu^{\ell+1}y \in L$ for every $x, y \in A^*$. We concentrate on the right to left implication (the converse one is symmetrical). Assume that $xu^{\ell+1}y \in L$. Since $L = L_1 L_2$, we get $w_1 \in L_1$ and $w_2 \in L_2$ such that $xu^{k+1}y = w_1 w_2$. Since $k \geq 2m+1$, it follows that either xu^{m+1} is a prefix of w_1 or $u^{m+1}y$ is a suffix of w_2. By symmetry, we assume that the former property holds: we have $w_1 = xu^{m+1}z$ for some $z \in A^*$. Observe that since $xu^{k+1}y = w_1 w_2$, it follows that $zw_2 = u^{k-m}y$. Moreover, we have $m \geq k_1$ by definition of m. Since $xu^{m+1}z = w_1 \in L_1$, we know therefore that $xu^m z \in L_1$ by definition of k_1. Thus, $xu^m zw_2 \in L_1 L_2 = L$. Since $zw_2 = u^{k-m}y$, this yields $xu^k y \in L$, concluding the proof. □

4.2 Schützenberger's theorem

We may now present and prove Schützenberger's theorem. Let us first define aperiodic monoids. There are several equivalent definitions in the literature. We use an equational one based on the idempotent power ω available in finite monoids. A *finite* monoid M is aperiodic when it satisfies the following property:

$$\text{for every } s \in M, \quad s^\omega = s^{\omega+1} \tag{1}$$

We are ready to state Schützenberger's theorem.

Theorem 10 (Schützenberger [36]) *A regular language is star-free if and only if its syntactic monoid is aperiodic.*

Theorem 10 illustrates of the syntactic approach presented in Section 3. It validates Proposition 3: the star-free languages are characterized by a property of their syntactic morphism. In fact, for this particular class, one does not even need the full morphism, the syntactic monoid suffices.

The main application is a membership algorithm for the class of star-free languages. Given as input a regular language L, one may compute its syntactic monoid and check whether it satisfies Equation (1): this boils down to testing all elements in the monoid. By Theorem 10, this decides whether L is star-free. However, as we explained in Remark 1 when we first introduced membership, this theorem is also important for the arguments that are required to prove it. Indeed, providing these arguments requires a deep insight on SF. The right to left implication is of particular interest: "given a regular language whose syntactic monoid is aperiodic, prove that it is star-free". This involves devising a generic way to construct a star-free description for *every* regular language recognized by a monoid satisfying a *syntactic* property. This is the implication that we handle with covers. On the other hand, the converse implication is simple and standard (essentially, we already proved it with Lemma 9).

Proof. We fix an alphabet A and a regular language $L \subseteq A^*$ for the proof. Let $\alpha : A^* \to M$ be the syntactic morphism of L. We prove that $L \in \mathrm{SF}(A)$ if and only if M is aperiodic. Let us first handle the left to right implication.

From star-free languages to aperiodicity. Assume that $L \in \mathrm{SF}(A)$. We prove that M is aperiodic, *i.e.* that (1) is satisfied. Let $s \in M$, we have to show that $s^\omega = s^{\omega+1}$.

Since α is a syntactic morphism, it is surjective and there exists $w \in A^*$ such that $\alpha(w) = s$. Moreover, since $L \in \mathrm{SF}(A)$, Lemma 9 yields $k \geq 1$ such that $w^{k\omega} \equiv_L w^{k\omega+1}$. By definition of the syntactic morphism, this implies that $\alpha(w^{k\omega}) = \alpha(w^{k\omega+1})$. Since $\alpha(w) = s$, this yields $s^\omega = s^{\omega+1}$ as desired

From aperiodicity to star-free languages. Assume that M is aperiodic. We show that L is star-free. We rely on the notions introduced in the Section 3 and directly prove that *every* language recognized by α is star-free.

Remark 11. Intuitively, this property is stronger than L being star-free. Yet, since SF is a quotient-closed Boolean algebra, it is equivalent by Proposition 3. □

The argument is based on Proposition 7: we use induction to construct an SF-cover $\mathbf{K}$ of A^* which is confined by α. By the proposition, this implies that every language recognized by α belongs to $\mathrm{SF}(A)$. We start with a preliminary definition that we require to formulate the induction.

Let B be an arbitrary alphabet, $\beta : B^* \to M$ a morphism and $s \in M$. We say that a finite set of languages $\mathbf{K}$ (over B) is (s, β)-*safe* if for every $K \in \mathbf{K}$ and every $w, w' \in K$, we have $s\beta(w) = s\beta(w')$.

Lemma 12 *Let B be an alphabet. Consider a morphism $\beta : B^* \to M$, $C \subseteq B$ and $s \in M$. There exists an* SF*-cover of C^* which is (s, β)-safe.*

We first use Lemma 12 to conclude the main argument. We apply the lemma for $B = A$, $\beta = \alpha$ and $s = 1_M$. This yields an SF-cover $\mathbf{K}$ of A^* which is $(1_M, \alpha)$-safe. By definition, it follows that for every $K \in \mathbf{K}$, we have $\alpha(w) = \alpha(w')$ for all $K \in \mathbf{K}$. By Fact 6, this implies that $\mathbf{K}$ is confined by α, completing the main argument.

It remains to prove Lemma 12. Let B be an alphabet, $\beta : B^* \to M$ a morphism, $C \subseteq B$ and $s \in M$. We build an SF-cover $\mathbf{K}$ of C^* which is (s, β)-safe using induction on the three following parameters listed by order of importance:

1. The size of $\beta(C^+) \subseteq M$.
2. The size of C.
3. The size of $s\beta(C^*) \subseteq M$.

Remark 13. The aperiodic monoid M remains fixed throughout the whole proof. On the other hand, the alphabets B and C, the morphism $\beta : B^* \to M$ and $s \in M$ may change when applying induction. □

We distinguish two cases depending on the following property of β, C and s. We say that s *is* (β, C)*-stable* when the following holds:

$$\text{for every } c \in C, \quad s\beta(C^*) = s\beta(C^*c). \tag{2}$$

We first consider the case when s is (β, C)-stable. This is the base case which we handle using the hypothesis that M is aperiodic.

Base case: s is (β, C)-stable. In that case, we define $\mathbf{K} = \{C^*\}$ which is clearly an SF-cover of C^* (we have $C^* \in \mathrm{SF}(B)$ as seen in Example 8). It remains to show that $\mathbf{K}$ is (s, β)-safe. For $w, w' \in C^*$, we have to show that $s\beta(w) = s\beta(w')$. We actually prove that $s\beta(w) = s$ for every $w \in C^*$ which implies the desired result. Since s is (β, C)-stable, we have the following fact.

Fact 14 *For every $u \in C^*$, there exists $t \in \beta(C^*)$ such that $st\beta(u) = s$.*

Proof. We use induction on the length of $u \in C^*$. If $u = \varepsilon$, the fact holds for $t = 1_M$. Assume now that $u \in C^+$. We have $u = cu'$ for $u' \in C^*$ and $c \in C$. Induction yields $t' \in \beta(C^*)$ such that $st'\beta(u') = s$. Moreover, since s is (β, C)-stable, (2) yields $t \in \beta(C^*)$ such that $st\beta(c) = st'$. Altogether, we obtain that $st\beta(u) = st\beta(c)\beta(u') = st'\beta(u') = s$ which concludes the proof. □

Consider the word $w^\omega \in C^*$ (with ω as the idempotent power of M). We apply Fact 14 for $u = w^\omega$. This yields $t \in \beta(C^*)$ such that $s = st(\beta(w))^\omega$. Since M is aperiodic, we have $(\beta(w))^\omega = (\beta(w))^{\omega+1}$ by Equation 1. This yields $s\beta(w) = st(\beta(w))^{\omega+1} = st(\beta(w))^\omega = s$, concluding the base case.

Inductive case: s is not (β, C)-stable. By hypothesis, there exists a letter $c \in C$ such that the following *strict* inclusion holds $s\beta(C^*c) \subsetneq s\beta(C^*)$. We fix $c \in C$ for the remainder of the argument.

Let D be the sub-alphabet $D = C \setminus \{c\}$. By definition, $|D| < |C|$. Hence, induction on our second parameter in Lemma 12 (*i.e.*, the size of C) yields an SF-cover $\mathbf{H}$ of D^* which is $(1_M, \beta)$-safe. Note that it is clear that our first induction parameter (the size of $\alpha(C^+)$) has not increased since $D \subseteq C$.

We distinguish two independent sub-cases. Clearly, we have $\beta(C^*c) \subseteq \beta(C^+)$. The argument differs depending on whether this inclusion is strict or not.

Sub-case 1: $\beta(C^*c) = \beta(C^+)$. Consider a language $H \in \mathbf{H}$. Since $\mathbf{H}$ is a cover of D^* which is $(1_M, \beta)$-safe by definition, there exists some element $t_H \in \beta(D^*)$ such that $\beta(w) = t_H$ for every $w \in H$. The construction of the desired SF-cover $\mathbf{K}$ of C^* is based on the following fact which we prove using induction on our third parameter (the size of $s\beta(C^*)$).

Fact 15 *For every language $H \in \mathbf{H}$, there exists an* SF*-cover $\mathbf{U}_H$ of C^* which is $(st_H\beta(c), \beta)$-safe.*

Proof. Since $t_H \in \beta(D^*)$, it is immediate that $st_H\beta(c) \in s\beta(D^*c)$. Hence, $st_H\beta(c)\beta(C^*) \subseteq s\beta(C^+)$. Moreover, $\beta(C^*c) = \beta(C^+)$ by hypothesis in Sub-case 1. Thus, $st_H\beta(c)\beta(C^*) \subseteq s\beta(C^*c)$. Finally, recall that the letter c satisfies

$s\beta(C^*c) \subsetneq s\beta(C^*)$ by definition. Consequently, we have the **strict** inclusion $st_H\beta(c)\beta(C^*) \subsetneq s\beta(C^*)$. Hence, we may apply induction on our third parameter in Lemma 12 (*i.e.* the size of $s\beta(C^*)$) to obtain the desiredn SF-cover $\mathbf{U}_H$ of C^* which is $(st_H\beta(c), \beta)$-safe. Note that here, our first two parameters have not increased (they only depend on β and C which remain unchanged). □

We may now use Fact 15 to build the desired cover $\mathbf{K}$ of C^*. We define $\mathbf{K} = \mathbf{H} \cup \{HcU \mid H \in \mathbf{H} \text{ and } U \in \mathbf{U}_H\}$. Clearly, $\mathbf{K}$ is an SF-cover of C^* by hypothesis on $\mathbf{H}$ and $\mathbf{U}_H$ since $D = C \setminus \{c\}$ and SF is closed under concatenation. We need to show that $\mathbf{K}$ is (s, β)-safe. Let $K \in \mathbf{K}$ and $w, w' \in K$, we need to show that $s\beta(w) = s\beta(w')$. By definition of $\mathbf{K}$, there are two cases. When $K \in \mathbf{H}$, the result is immediate since $\mathbf{H}$ is $(1_M, \beta)$-safe by definition. Otherwise, $K = HcU$ for $H \in \mathbf{H}$ and $U \in \mathbf{U}_H$. Thus, we get $x, x' \in H$ and $u, u' \in U$ such that $w = xcu$ and $w' = x'cu'$. By definition, $\beta(x) = \beta(x') = t_H$. Moreover, since $\mathbf{U}_H$ is $(st_H\beta(c), \beta)$-safe by definition in Fact 15, we have $st_H\beta(cu) = st_H\beta(cu')$. Altogether, this yields $s\beta(xcu) = s\beta(x'cu')$, *i.e.* $s\beta(w) = s\beta(w')$ as desired.

Sub-case 2: $\beta(C^*c) \subsetneq \beta(C^+)$**.** Let us first explain informally how the cover $\mathbf{K}$ of C^* is built in this case. Let $w \in C^*$. Since $D = C \setminus \{c\}$, w admits a unique decomposition $w = uv$ such that $u \in (D^*c)^*$ and $v \in D^*$ (*i.e.*, v is the largest suffix of w in D^* and u is the corresponding prefix). Using induction, we construct SF-covers of the possible prefixes and suffixes. Then, we combine them to construct a cover of the whole set C^*. Actually, we already covered the suffixes: we have an SF-cover $\mathbf{H}$ of C^* which is $(1_M, \beta)$-safe. It remains to cover the prefixes. We do so this in the following lemma which we prove using induction on our first parameter (the size of $\beta(C^+)$).

Lemma 16 *There exists an* SF*-cover* $\mathbf{V}$ *of* $(D^*c)^*$ *which is* $(1_M, \beta)$*-safe.*

Proof. Let $E = \beta(D^*c)$. Using E as a new alphabet, we apply induction on the first parameter in Lemma 12 (*i.e.*, the size of $\beta(C^+)$) to build an auxiliary SF-cover of E^* which we then use to construct $\mathbf{V}$.

Since $E = \beta(D^*c) \subseteq M$, there exists a natural morphism $\gamma : E^* \to M$ defined by $\gamma(e) = e$ for every $e \in E$. Clearly, $\gamma(E^+) \subseteq \beta(C^*c)$. Since $\beta(C^*c) \subsetneq \beta(C^+)$ by hypothesis of Sub-case 2, this implies $\gamma(E^+) \subsetneq \beta(C^+)$ and induction on the first parameter in Lemma 12 yields an SF-cover $\mathbf{W}$ of E^* which is $(1_M, \gamma)$-safe. We use $\mathbf{W}$ to construct $\mathbf{V}$. First, we define a map $\mu : (D^*c)^* \to E^*$.

We let $\mu(\varepsilon) = \varepsilon$. Otherwise, let $w \in (D^*c)^+$ be a nonempty word. Since $c \notin D$, w admits a **unique** decomposition $w = w_1 \cdots w_n$ with $w_1, \ldots, w_n \in D^*c$. Hence, we may define $\mu(w_1 \cdots w_n) = e_1 \cdots e_n$ with $e_i = \beta(w_i)$ for every $i \leq n$ (recall that $E = \beta(D^*c)$ by definition). We are ready to define $\mathbf{W}$. We let,

$$\mathbf{V} = \{\mu^{-1}(W) \mid W \in \mathbf{W}\}$$

It remains to show that $\mathbf{V}$ is an SF-cover of $(D^*c)^*$ which is $(1_M, \beta)$-safe. It is immediate that $\mathbf{V}$ is a cover of $(D^*c)^*$ since $\mathbf{W}$ was a cover of E^*.

Let us prove that $\mathbf{V}$ is $(1_M, \beta)$-safe. Let $V \in \mathbf{V}$ and $v, v' \in V$. We prove that $\beta(v) = \beta(v')$. By definition, there exists $w \in \mathbf{W}$ such that $V = \mu^{-1}(W)$. Thus,

$\mu(v), \mu(v') \in W$ which implies that $\gamma(\mu(v)) = \gamma(\mu(v'))$ since $\mathbf{W}$ is $(1_M, \gamma)$-safe by definition. One may now verify from the definitions that $\gamma(\mu(v)) = \beta(v)$ and $\gamma(\mu(v')) = \beta(v')$. Thus, we obtain $\beta(v) = \beta(v')$ as desired.

It remains to show that every $V \in \mathbf{V}$ is star-free. By definition of $\mathbf{V}$, it suffices to show that for every $W \in \mathrm{SF}(E)$, we have $\mu^{-1}(W) \in \mathrm{SF}(B)$. We proceed by induction on the definition of W as a star-free language. When $W = \emptyset$, it is clear that $\mu^{-1}(W) = \emptyset \in \mathrm{SF}(B)$. Assume now that $W = \{e\}$ for some $e \in E$. By definition, $\mu^{-1}(e) = \{w \in D^*c \mid \beta(w) = e\}$. This may be reformulated as follows: $\mu^{-1}(e) = Uc$ with $U = \{u \in D^* \mid \beta(uc) = e\}$. Clearly, U is the intersection of D^* with a language recognized by β. Recall that we have an SF-cover $\mathbf{H}$ of D^* which is $(1_M, \beta)$-safe (and therefore confined by β). Hence, Proposition 7 implies that $U \in \mathrm{SF}(B)$. It follows that $\mu^{-1}(e) = Uc \in \mathrm{SF}(B)$ as desired. We turn to the inductive cases.

First, assume that there are simpler languages $W_1, W_2 \in \mathrm{SF}(E)$ such that either $W = W_1W_2$ or $W = W_1 \cup W_2$. By induction, $\mu^{-1}(W_i) \in \mathrm{SF}(B)$ for $i = 1, 2$. Moreover, the definition of μ implies that $\mu^{-1}(W_1W_2) = \mu^{-1}(W_1)\mu^{-1}(W_2)$ and $\mu^{-1}(W_1 \cup W_2) = \mu^{-1}(W_1) \cup \mu^{-1}(W_2)$. Hence, we obtain $\mu^{-1}(W) \in \mathrm{SF}(B)$. Finally, assume that $W = E^* \setminus W'$ for a simpler language $W' \in \mathrm{SF}(E)$. By induction, $\mu^{-1}(W') \in \mathrm{SF}(E)$. Moreover, $\mu^{-1}(W) = (D^*c)^* \setminus \mu^{-1}(W')$. Clearly, $(D^*c)^* = C^* \setminus (C^*D) \in \mathrm{SF}(B)$. Thus, we get $\mu^{-1}(W) \in \mathrm{SF}(B)$ as desired. □

We are ready to construct the desired SF-cover $\mathbf{K}$ of C^*. Let $\mathbf{V}$ be the $(1_M, \beta)$-safe SF-cover of $(D^*c)^*$ given by Lemma 16 and consider our $(1_M, \beta)$-safe SF-cover $\mathbf{H}$ of D^*. We define $\mathbf{K} = \{VH \mid V \in \mathbf{V} \text{ and } H \in \mathbf{H}\}$. It is immediate by definition that $\mathbf{K}$ is an SF-cover of C^* since $D = C \setminus \{c\}$ and SF is closed under concatenation. It remains to verify that $\mathbf{K}$ is (s, β)-safe (it is in fact $(1_M, \beta)$-safe). Let $K \in \mathbf{K}$ and $w, w' \in K$, we show that $\beta(w) = \beta(w')$ (which implies $s\beta(w) = s\beta(w')$). By definition, $K = VU$ with $V \in \mathbf{V}$ and $U \in \mathbf{U}$. Therefore, $w = vu$ and $w' = v'u'$ with $u, u' \in U$ and $v, v' \in V$. Since U and V are both $(1_M, \beta)$-safe by definition, we have $\beta(u) = \beta(u')$ and $\beta(v) = \beta(v')$. It follows that $\beta(w) = \beta(w')$. This concludes the proof of Lemma 12. □

5 Piecewise testable languages and Simon's theorem

We turn to our second example: Simon's theorem [38]. This results states an algebraic characterization of another prominent class of regular languages: the *piecewise testable languages* (PT). It is quite important in the literature as it was among the first results of this kind after Schützenberger's theorem (which we proved in Section 4). Over the years, many different proofs have been found (examples include [38,18,39,1,9,12]). We present a new proof, based on PT-covers and entirely independent from previously known arguments. It relies on a concatenation principle for the piecewise testable languages that can only be formulated with PT-covers.

We first recall the definition of piecewise testable languages. Then, we state the theorem properly and present the proof.

5.1 Definition

Let us define the class of piecewise testable languages (PT). Given an alphabet A and $u, v \in A^*$, we say that u is a *piece* of v and write $u \preceq v$ when u can be obtained from v by removing letters and gluing the remaining ones together. More precisely, $u \preceq v$ when there exist $a_1, \ldots, a_n \in A$ and $v_0, \ldots, v_n \in A^*$ such that,

$$u = a_1 a_2 \cdots a_n \quad \text{and} \quad v = v_0 a_1 v_1 a_2 v_2 \cdots v_{n-1} a_n v_n.$$

For instance, acb is a piece of $bb\underline{abc}b\underline{cb}ba$. Note that by definition, the empty word "ε" is a piece of every word (this is the case $n = 0$). Furthermore, it is clear that the relation $\preceq$ is a preorder on A^*.

For every word $u \in A^*$, we write $\uparrow u \subseteq A^*$ for the language consisting of all words v such that u is a piece of v. If $u = a_1 \cdots a_n$, we have by definition:

$$\uparrow u = \{v \in A^* \mid u \preceq v\} = A^* a_1 A^* a_2 A^* \cdots a_{n-1} A^* a_n A^*.$$

We may now define PT. A language $L \subseteq A^*$ is *piecewise testable* (*i.e.* $L \in \mathrm{PT}(A)$) when L is a (finite) Boolean combination of languages $\uparrow w$ for $w \in A^*$.

Example 17. We let $A = \{a, b\}$ as the alphabet. Then $a^+b^+ \in \mathrm{PT}(A)$. Indeed, $a^+b^+ = A^*aA^*bA^* \setminus A^*bA^*aA^*$. Moreover, observe that every finite language is piecewise testable. Since PT is closed under union, it suffices to show that every singleton is piecewise testable. Consider a word $w = a_1 \cdots a_n$. By definition, w is the only word belonging to $A^*a_1A^*a_2A^* \cdots a_{n-1}A^*a_nA^*$ but not to $A^*b_1A^*b_2A^* \cdots b_nA^*b_{n+1}A^*$, where $b_1, \ldots, b_{n+1}$ denotes any sequence of $n+1$ letters. Hence, $\{w\}$ is piecewise testable. □

Clearly PT is a Boolean algebra and one may verify that it is quotient-closed (the details are left to the reader). We complete the definition with two properties of PT. The first one is standard and we shall need it to prove that "easy" direction of Simon's theorem (every piecewise testable language satisfies the characterization).

Lemma 18 *Let A be an alphabet and $L \in \mathrm{PT}(A)$. There exists $k \geq 1$ such that for every $\ell \geq k$ and $u, v \in A^*$, we have $(uv)^\ell u \equiv_L (uv)^\ell \equiv_L v(uv)^\ell$.*

Proof. Since $L \in \mathrm{PT}$, there exists $k \geq 1$ such that L is a Boolean combinations of language $\uparrow w$ with $w \in A^*$ such that $|w| \leq k$ (*i.e.* w has length at most k). We prove that the lemma holds for this number k. Let $u, v \in A^*$ and $\ell \geq k$. We show that $(uv)^\ell u \equiv_L (uv)^\ell \equiv_L v(uv)^\ell$. By symmetry, we concentrate on $(uv)^\ell u \equiv_L (uv)^\ell$: given $x, y \in A^*$, we show that $x(uv)^\ell uy \in L \Leftrightarrow x(uv)^\ell y \in L$. Since $\ell \geq k$, one may verify that for every $w \in A^*$ such that $|w| \leq k$, we have $w \preceq x(uv)^\ell uy \Leftrightarrow w \preceq x(uv)^\ell y$. In other words, $x(uv)^\ell uy \in \uparrow w \Leftrightarrow x(uv)^\ell y \in \uparrow w$. Since L is a Boolean combination of such languages, this implies the equivalence $x(uv)^\ell uy \in L \Leftrightarrow x(uv)^\ell y \in L$ as desired. □

The second result is specific to our covering-based approach for proving Simon's theorem. It turns out that elegant proof arguments for membership algorithms often apply to classes that are closed under concatenation (or some weak variant thereof). As seen in the previous section, the star-free languages are an example. Unfortunately, PT is *not* closed under concatenation. For example, consider the alphabet $A = \{a, b\}$. We have $A^* \in \mathrm{PT}$ and $\{a\} \in \mathrm{PT}$ as seen in Example 17. Yet, one may verify with Lemma 18 that $A^*a \notin \mathrm{PT}$.

We solve this issue with a "weak concatenation principle" for piecewise testable languages. This result can only be formulated using PT-covers. While its proof is rather technical, an interesting observation is that it characterizes the piecewise testable languages. In the proof of Simon's theorem, we only use this concatenation principle and the hypothesis that PT is a Boolean algebra (we never come back to the original definition of PT).

Proposition 19 *Let $u, v \in A^*$ and $a \in A$. Moreover, let $\mathbf{K}_u$ and $\mathbf{K}_v$ be* PT*-covers of $\uparrow u$ and $\uparrow v$ respectively. There exists a* PT*-cover $\mathbf{K}$ of $\uparrow(uav)$ such that for every $K \in \mathbf{K}$ we have $K_u \in \mathbf{K}_u$ and $K_v \in \mathbf{K}_v$ satisfying $K \subseteq K_u a K_v$.*

Proof. We start with standard definitions that we need to describe $\mathbf{K}$. For every $k \in \mathbb{N}$, we associate a preorder $\preceq_k$ over A^*. For $w, w' \in A^*$, we write $w \preceq_k w'$ to indicate that for every $x \in A^*$ such that $|x| \leq k$, we have $x \preceq w \Rightarrow x \preceq w'$. Clearly, $\preceq_k$ is a preorder which is coarser than $\preceq$: for every w, w' such that $w \preceq w'$, we have $w \preceq_k w'$. Moreover, we write $\sim_k$ for the equivalence generated by this preorder: $w \sim_k w'$ if and only if $x \preceq w \Leftrightarrow x \preceq w'$ for every $x \in A^*$ such that $|x| \leq k$. Clearly, $\sim_k$ has finite index.

Since $\mathbf{K}_u$ and $\mathbf{K}_v$ are PT-covers, there exists some number $k \in \mathbb{N}$ every language $K \in \mathbf{K}_u \cup \mathbf{K}_v$ is a finite Boolean combination of languages $\uparrow x$ for $x \in A^*$ such that $|x| \leq k$. In other words, every such language K is a union of $\sim_k$-classes. Moreover, we may choose k so that $|u| \leq k$ and $|v| \leq k$. We shall define the cover $\mathbf{K}$ as a set of $\sim_h$-classes for an appropriate number h that we choose using the following technical lemma.

Lemma 20 *Let $h \geq 2|A|^{k+1} + 1$, $a \in A$ and $u', v', w \in A^*$ such that $u'av' \preceq_h w$. There exist $u'', v'' \in A^*$ such that $w = u''av''$, $u' \preceq_k u''$ and $v' \preceq_k v''$.*

Proof. We claim that there exist $y, z \in A^*$ with length at most $|A|^{k+1}$ such that $y \preceq u' \preceq_k y$ and $z \preceq v' \preceq_k z$. We first use this claim to prove the lemma. Clearly, $|yaz| \leq 2|A|^{k+1} + 1 \leq h$ and $yaz \preceq u'av'$. Therefore, since $u'av' \preceq_h w$, it follows that $yaz \preceq w$. This yields a decomposition $w = u''av''$ such that $y \preceq u''$ and $z \preceq v''$. Since $u' \preceq_k y$ and $v' \preceq_k z$, this implies $u' \preceq_k u''$ and $v' \preceq_k v''$ as desired.

It remains to prove the claim. We only construct a piece $y \in A^*$ such that $|y| \leq |A|^{k+1}$ and $y \preceq u' \preceq_k y$, as the construction of z is analogous. Let F be the set of all pieces of u' of size at most k, that is,

$$F = \{u'' \in A^* \mid u'' \preceq u' \text{ and } |u''| \leq k\}.$$

Clearly, $|F| \leq |A|^{k+1}$. For $x \in A^*$, let $L_F(x)$ be the set of words of F that are pieces of x. Let $u' = u_1 a u_2$ be some decomposition of u'. Note that $L_F(u_1) \subseteq$

$L_F(u_1a)$. We say that the occurrence of a given by the decomposition $u' = u_1au_2$ is *bad* if $L_F(u_1) = L_F(u_1a)$. Let y be the word obtained from u' by deleting all bad letters (and keeping the other ones). By construction, $y \preceq u'$ and $L_F(y) = L_F(u')$. The latter property implies that $u' \preceq y$ for every $u' \in F$. By definition of F, this means that $u' \preceq_k y$. Furthermore, letters of y are not bad, and one may verify that there are at most $|L_F(u')| = |F|$ such letters. Therefore, $|y| \leq |F| \leq |A|^{k+1}$, which concludes the proof. □

We define $h = 2|A|^{k+1} + 1$. It is immediate that every $\sim_h$-class is a language of PT (it is a Boolean combination of languages $\uparrow x$ for $x \in A^*$ such that $|x| \leq h$). Hence, the set $\mathbf{K}$ containing all $\sim_h$-classes which intersect $\uparrow(uav)$ is a PT-cover of $\uparrow(uav)$. It remains to show that for every $K \in \mathbf{K}$, there exist $K_u \in \mathbf{K}_u$ and $K_v \in \mathbf{K}_v$ such that $K \subseteq K_uaK_v$. We fix the language $K \in \mathbf{K}$ for the proof. We need the following result.

Lemma 21 *Let $H \subseteq K$ be a* finite *language. There exist $K' \in \mathbf{K}_u$ and $K'' \in \mathbf{K}_v$ such that $H \subseteq K'aK''$.*

Proof. Let $w_1, \ldots, w_n \in A^*$ be the words in H, *i.e.*, $H = \{w_1, \ldots, w_n\}$. Our goal is to find $K' \in \mathbf{K}_u$ and $K'' \in \mathbf{K}_v$ such that $w_i \in K'aK''$ for all $i = 1, \ldots, n$. Therefore, we first have to find a suitable decomposition of each word w_i as u_iav_i, and then to show that all u_i's belong to some $K' \in \mathbf{K}_u$ and all v_i's belong to some $K'' \in \mathbf{K}_v$.

By definition, K is a $\sim_h$-class and it intersects $\uparrow(uav)$. This yields a word $x \in \uparrow(uav)$ such that $x \sim_h w_1 \sim_h \cdots \sim_h w_n$. Since $x \in \uparrow(uav)$, there exist $u' \in \uparrow u$ and $v' \in \uparrow v$ such that $x = u'av'$. Let $\ell = |w_1| + 1$. We may write the relations $x \sim_h w_1 \sim_h \cdots \sim_h w_n$ as follows:

$$u'av' \preceq_h \underbrace{\underbrace{w_1 \preceq_h \cdots \preceq_h w_n}_{\text{block 1}} \preceq_h \underbrace{w_1 \preceq_h \cdots \preceq_h w_n}_{\text{block 2}} \preceq_h \cdots \preceq_h \underbrace{w_1 \preceq_h \cdots \preceq_h w_n}_{\text{block } \ell}}_{n\ell \text{ words}}.$$

Since $h \geq 2|A|^{k+1} + 1$ by definition, may apply Lemma 20 $n\ell$ times to get $u_{1,1}, \ldots, u_{n,1}, \ldots, u_{1,\ell}, \ldots, u_{n,\ell} \in A^*$ and $v_{1,1}, \ldots, v_{n,1}, \ldots, v_{1,\ell}, \ldots, v_{n,\ell} \in A^*$ such that,

- for every $i \leq n$ and $j \leq \ell$, we have $w_i = u_{i,j}av_{i,j}$, and,
- $u' \preceq_k u_{1,1} \preceq_k \cdots \preceq_k u_{n,1} \preceq_k \cdots \preceq_k u_{1,\ell} \preceq_k \cdots \preceq_k u_{n,\ell}$, and,
- $v' \preceq_k v_{1,1} \preceq_k \cdots \preceq_k v_{n,1} \preceq_k \cdots \preceq_k v_{1,\ell} \preceq_k \cdots \preceq_k v_{n,\ell}$.

Since $\ell = |w_1|+1$, the first property and the pigeonhole principle yield $j_1 < j_2 \leq \ell$ such that $u_{1,j_1} = u_{1,j_2}$ and $v_{1,j_1} = v_{1,j_2}$. For every $i \leq n$, we let $u_i = u_{i,j_1}$ and $v_i = v_{i,j_1}$. Therefore, for all $i = 1, \ldots, n$, we have $w_i = u_iav_i$.

The second and third properties now yield $u' \preceq_k u_1 \preceq_k \cdots \preceq_k u_n \preceq_k u_1$ and $v' \preceq_k v_1 \preceq_k \cdots \preceq_k v_n \preceq_k v_1$, whence:

$$u' \preceq_k u_1 \sim_k \cdots \sim_k u_n \quad \text{and} \quad v' \preceq_k v_1 \sim_k \cdots \sim_k v_n.$$

Recall that $|u| \leq k$ by definition of k. Since $u' \in \uparrow u$ and $u' \preceq_k u_1$, it follows that $u_1 \in \uparrow u$. Since $\mathbf{K}_u$ is a cover of $\uparrow u$, this yields $K' \in \mathbf{K}_u$ such that $u_1 \in K'$. Since K' is a union of $\sim_k$-classes by choice of k and since $u_1 \sim_k \cdots \sim_k u_n$, we deduce that $u_1, \ldots, u_n \in K'$. Symmetrically, we obtain $K'' \in \mathbf{K}_v$ such that $v_1, \ldots, v_n \in K''$. Finally, since $w_i = u_i a v_i$ for every $i \leq n$, this yields $H = \{w_1, \ldots, w_n\} \subseteq K' a K''$, as desired. □

We may now finish the proof. For every $n \in \mathbb{N}$, we let $H_n \subseteq K$ be the (finite) language containing all words of length at most n in K. Clearly, $K = \bigcup_{n \in \mathbb{N}} H_n$ and $H_n \subseteq H_{n+1}$ for every $n \in \mathbb{N}$. Moreover, Lemma 21 implies that for every $n \in \mathbb{N}$, we have $K'_n \in \mathbf{K}_u$ and $K''_n \in \mathbf{K}_v$ such that $H_n \subseteq K'_n a K''_n$. Since $\mathbf{K}_u$ and $\mathbf{K}_v$ are finite sets, there exist $K_u \in \mathbf{K}_u$ and $K_v \in \mathbf{K}_v$ such that $K'_n = K_u$ and $K''_n = K_v$ for infinitely many n. Since $H_n \subseteq H_{n+1}$ for every $n \in \mathbb{N}$, it then follows that $H_n \subseteq K_u a K_v$ for every $n \in \mathbb{N}$. Finally, since $K = \bigcup_{n \in \mathbb{N}} H_n$, this implies $K \subseteq K_u a K_v$ which concludes the proof. □

5.2 Simon's theorem

We may now present and prove Simon's theorem. It characterizes the star-free languages as those whose syntactic monoid is *$\mathcal{J}$-trivial*. The original definition of this notion is based on the Green relation $\mathcal{J}$ defined on every finite monoid. Here, we do not consider this relation. Instead, we use an equational definition. A *finite* monoid M is $\mathcal{J}$-trivial when it satisfies the following property:

$$\text{for every } s, t \in M \quad (st)^\omega s = (st)^\omega = t(st)^\omega. \tag{3}$$

Theorem 22 (Simon [38]) *A regular language is piecewise testable if and only if its syntactic monoid is $\mathcal{J}$-trivial.*

As expected, the main application of Simon's theorem is the decidability of PT-membership. Given a regular language L as input, one may compute its syntactic monoid and check whether it satisfies Equation (3) by testing all possible combinations. By Theorem 22, this decides whether L is piecewise testable. Yet, as for the star-free languages in Section 4, this theorem is also important for the arguments that are required to prove it. We present such a proof now.

Proof. We fix an alphabet A and a regular language $L \subseteq A^*$ for the proof. Let $\alpha : A^* \to M$ be the syntactic morphism of L. We prove that $L \in \mathrm{PT}(A)$ if and only if M is $\mathcal{J}$-trivial. We start with the left to right implication which is essentially immediate from Lemma 18. As expected, the difficult and most interesting part of the proof is the converse implication.

From piecewise testable languages to $\mathcal{J}$-triviality. Assume that we have $L \in \mathrm{PT}(A)$. We prove that M is $\mathcal{J}$-trivial: (3) holds. Let $s, t \in M$, we have to show that $(st)^\omega s = (st)^\omega = t(st)^\omega$.

Since α is a syntactic morphism, it is surjective and there exists $u, v \in A^*$ such that $\alpha(u) = s$ and $\alpha(v) = t$. Moreover, since $L \in \mathrm{SF}(A)$, Lemma 18

yields $k \geq 1$ such that $(uv)^{k\omega}u \equiv_L (uv)^{k\omega} \equiv_L v(uv)^{k\omega}$. By definition of the syntactic morphism, this implies that $\alpha((uv)^{k\omega}u) = \alpha((uv)^{k\omega}) = \alpha(v(uv)^{k\omega})$. Since $\alpha(u) = s$ and $\alpha(v) = t$, this yields $(st)^\omega s = (st)^\omega = t(st)^\omega$ as desired

From $\mathcal{J}$-triviality to piecewise testable languages. Assume that M is $\mathcal{J}$-trivial. We show that L is piecewise testable. We rely on the notions introduced in the Section 3 and directly prove that *every* language recognized by α is piecewise testable. The argument is based on Proposition 7: we use induction to construct a PT-cover $\mathbf{K}$ of A^* which is confined by α. By the proposition, this implies that every language recognized by α belongs to $\mathrm{PT}(A)$. We start with a preliminary definition that we require to formulate the induction.

Given a finite set of languages $\mathbf{K}$, and $s, t \in M$, we say that $\mathbf{K}$ is (s,t)-*safe* if for every $K \in \mathbf{K}$ and $w, w' \in K$, we have $s\alpha(w)t = s\alpha(w')t$. The argument is based on the following lemma.

Lemma 23 *Let $s, t \in M$ and $w \in A^*$. There exists a* PT*-cover of $\uparrow w$ which is (s,t)-safe.*

We first use Lemma 23 to complete the main argument. We apply the lemma for $s = t = 1_M$ and $w = \varepsilon$. Since $\uparrow\varepsilon = A^*$, this yields a PT-cover $\mathbf{K}$ of A^* which is $(1_M, 1_M)$-safe. Thus, for every $K \in \mathbf{K}$ and $w, w' \in A^*$, we have $\alpha(w) = \alpha(w')$. By Fact 6, this implies that $\mathbf{K}$ is confined by α, concluding the proof.

It remains to prove Lemma 23. Let $s, t \in M$ and $w \in A^*$. We construct a PT-cover $\mathbf{K}$ of $\uparrow w$ which is (s,t)-safe. We write $P[s,w,t] \subseteq M \times M$ for the following set:

$$P[s,w,t] = \big\{(s\alpha(x), \alpha(y)t) \mid x, y \in A^* \text{ and } xy \in \uparrow w\big\}.$$

We proceed by induction on the two following parameters, listed by order of importance:

1. The size of $P[s,w,t]$.
2. The length of w.

We consider two cases depending on whether w is empty or not. We first assume that this property holds.

First case: $w = \varepsilon$. We handle this case using induction on our first parameter. Let $H \subseteq A^*$ be the language of all words $v \in A^*$ such that $(s,t) \notin P[s,v,t]$. We use induction to build a PT-cover of H (note that it may happen that H is empty in which case we do not need induction).

Fact 24 *There exists a* PT*-cover $\mathbf{K}_H$ of H which is (s,t)-safe.*

Proof. One may verify with a pumping argument that there exists a *finite* set $F \subseteq H$ such that $H \subseteq \bigcup_{v \in F} (\uparrow v)$ (this is also an immediate consequence of Higman's lemma). Hence, it suffices to prove that for every $v \in H$, there exists a PT-cover $\mathbf{K}_v$ of $\uparrow v$ which is (s,t)-safe. Indeed, one may then choose $\mathbf{K}_H$ to be the union of all covers $\mathbf{K}_v$ for $v \in F$. We fix $v \in H$ for the proof.

Since $w = \varepsilon$, we have $\uparrow w = A^*$. Since α is surjective (it is a syntactic morphism), it follows that $P[s, w, t] = \{(sq, rt) \mid q, r \in M\}$. Therefore, we have $P[s, v, t] \subseteq P[s, w, t]$ and $(s, t) \in P[s, w, t]$. Since $(s, t) \notin P[s, v, t]$ by definition of H, we get $|P[s, v, t]| < |P[s, w, t]|$. Hence, induction on the first parameter in Lemma 23 (the size of $P[s, w, t]$) yields a PT-cover $\mathbf{K}_v$ of $\uparrow v$ which is (s, t)-safe, as desired. □

We let $\mathbf{K}_H$ be the PT-cover $\mathbf{K}_H$ of H given by Fact 24. We define,

$$K_\perp = A^* \setminus \left(\bigcup_{K \in \mathbf{K}_H} K \right).$$

Finally, we let $\mathbf{K} = \{K_\perp\} \cup \mathbf{K}_H$. It is immediate that $\mathbf{K}$ is a PT-cover of $A^* = \uparrow \varepsilon$ since PT is a Boolean algebra. It remains to verify that $\mathbf{K}$ is (s, t)-safe. Consider $K \in \mathbf{K}$ and let $u, u' \in K$. We prove that $s\alpha(u)t = s\alpha(u')t$. If $K \in \mathbf{K}_H$, this is immediate since $\mathbf{K}_H$ is (s, t)-safe by construction. Hence, it suffices to show that $K_\perp$ is (s, t)-safe. This is a direct consequence of the following fact. Note that this is the only place in the proof where we use the hypothesis that M satisfies (3).

Fact 25 *For every word $v \in K_\perp$, we have $s\alpha(v)t = st$.*

Proof. Let $v \in K_\perp$. By definition of $K_\perp$, $v \notin K'$ for every $K' \in \mathbf{K}_H$. Since $\mathbf{K}_H$ is a cover of H, it follows that $v \notin H$. By definition of H, it follows that $(s, t) \in P[s, v, t]$. By definition, this yields $x, y \in A^*$ such that $s\alpha(x) = s$, $t = \alpha(y)t$ and $xy \in \uparrow v$. The latter property yields $x', y' \in A^*$ such that $v = x'y'$, $x \in \uparrow x'$ and $y \in \uparrow y'$. We prove that $s\alpha(x') = s$ and $t = \alpha(y')t$, which yields as desired that $s\alpha(v)t = s\alpha(x'y')t = st$. By symmetry, we only show that $s = s\alpha(x')$.

Since $s = s\alpha(x)$, we have $s = s(\alpha(x))^\omega$. Moreover, since $x \in \uparrow x'$, we have $x_0, \ldots, x_n \in A^*$ and $a_1, \ldots, a_n \in A$ such that $x' = a_1 \cdots a_n$ and $x = x_0 a_1 x_1 \cdots a_n x_n$. It follows from (3) that for every $1 \leq i \leq n$, we have:

$$(\alpha(x))^\omega = (\alpha(x))^\omega \alpha(x_0 a_1 x_1 \cdots x_{i-1}) = (\alpha(x))^\omega \alpha(x_0 a_1 x_1 \cdots x_{i-1} a_i).$$

This yields $(\alpha(x))^\omega = (\alpha(x))^\omega \alpha(a_i)$. Therefore, since we know that $s = s(\alpha(x))^\omega$, we obtain $s\alpha(a_i) = s(\alpha(x))^\omega \alpha(a_i) = s(\alpha(x))^\omega = s$. Finally, this yields,

$$s = s\alpha(a_n) = s\alpha(a_{n-1}a_n) = \cdots = s\alpha(a_1 \cdots a_{n-1}a_n) = s\alpha(x').$$

This concludes the proof. □

Second case: $w \in A^+$. In that case, we have $u, v \in A^*$ and $a \in A$ such that $w = uav$ (the choice of u, v and a is arbitrary). Consider the two following subsets of M:

$$M_u = \{\alpha(xa) \mid x \in \uparrow u\} \quad \text{and} \quad M_v = \{\alpha(ay) \mid y \in \uparrow v\}.$$

Moreover, we say that a cover $\mathbf{K}$ of some language H is *tight* when $K \subseteq H$ for every $K \in \mathbf{K}$. We use induction to prove the following fact.

Fact 26 *There exist tight* PT*-covers* $\mathbf{K}_u$ *and* $\mathbf{K}_v$ *of* $\uparrow u$ *and* $\uparrow v$ *which satisfy the following properties:*

- *for every* $r \in M_u$*, the cover* $\mathbf{K}_v$ *of* $\uparrow v$ *is* (sr, t)*-safe.*
- *for every* $r \in M_v$*, the cover* $\mathbf{K}_u$ *of* $\uparrow u$ *is* (s, rt)*-safe.*

Proof. We construct $\mathbf{K}_v$ (the construction of $\mathbf{K}_u$ is symmetrical). Let $M_u = \{r_1, \ldots, r_n\}$. For every $i \leq n$, assume that we already have a PT-cover $\mathbf{H}_i$ of $\uparrow v$ which is (sr_i, t)-safe. We define,

$$\mathbf{K}_v = \{\uparrow v \cap H_1 \cap \cdots \cap H_n \mid H_i \in \mathbf{H}_i \text{ for every } i \leq n\}.$$

Since PT is a Boolean algebra, it is immediate that $\mathbf{K}_v$ is a tight PT-cover of $\uparrow v$ which is (sr, t)-safe for every $r \in M_u$. Thus, it remains to build for every $i \leq n$ such a PT-cover $\mathbf{H}_i$.

We fix $i \leq n$ for the proof. By definition of M_u, we have $r_i = \alpha(u_i a)$ for some word $u_i \in \uparrow u$. Observe that since $w = uav$, we have $P[sr_i, v, t] \subseteq P[s, w, t]$ by definition: our first induction parameter (*i.e.*, the size of $P[s, w, t]$) has **not** increased. Hence, since $|v| < |w|$, it follows by induction on our second parameter in Lemma 23 (the length of w) that there exists a PT-cover $\mathbf{H}_i$ of $\uparrow v$ which is (sr_i, t)-safe. This concludes the proof. □

We are ready to construct the desired PT-cover $\mathbf{K}$ of $\uparrow w$. Consider the tight PT-covers $\mathbf{K}_u$ and $\mathbf{K}_v$ of $\uparrow u$ and $\uparrow v$ described in Fact 26. Since $w = uav$, Proposition 19 yields a PT-cover $\mathbf{K}$ of $\uparrow w$ such that for every $K \in \mathbf{K}$, there exist $K_u \in \mathbf{K}_u$ and $K_v \in \mathbf{K}_v$ satisfying $K \subseteq K_u a K_v$. It remains to prove that $\mathbf{K}$ is (s, t)-safe. Let $K \in \mathbf{K}$ and $x, x' \in K$. We prove that $s\alpha(x)t = s\alpha(x')t$.

By definition, $K \subseteq K_u a K_v$ for $K_u \in \mathbf{K}_u$ and $K_v \in \mathbf{K}_v$. Hence, there exist $y, y' \in K_u$ and $z, z' \in K_v$ such that $x = yaz$ and $x' = y'az'$. Since $\mathbf{K}_u$ is a tight cover of $\uparrow u$, we know that $y \in \uparrow u$, which implies that $\alpha(ya) \in M_u$ by definition. It follows that $\mathbf{K}_v$ is $(s\alpha(ya), t)$-safe by Fact 26. Therefore, since $z, z' \in K_v$ and $K_v \in \mathbf{K}_v$, we obtain $s\alpha(yaz)t = s\alpha(yaz')t$. Symmetrically, one may verify that $s\alpha(yaz')t = s\alpha(y'az')t$. Altogether, it follows that $s\alpha(yaz)t = s\alpha(y'az')t$, meaning that $s\alpha(x)t = s\alpha(x')t$. This concludes the proof of Lemma 23. □

6 Conclusion

We explained how covering provides a natural and convenient framework for handling membership questions. We illustrated this point by using covers to formulate new proofs for Schützenberger's theorem and Simon's theorem. We chose these two examples as they are arguably the two most famous characterization theorems of this kind. However, this approach is also relevant for other prominent characterization theorems. A first promising example is the class of *unambiguous languages*. It was also characterized by Schützenberger [37] and it also famous as the class of languages that can be define in two-variable first-order logic (this was shown by Thérien and Wilke [40]). Another interesting example is Knast's theorem [13] which characterizes the languages of dot-depth one. This class is natural generalization of the piecewise testable languages.

References

1. Almeida, J.: Implicit operations on finite **j**-trivial semigroups and a conjecture of I. Simon. Journal of Pure and Applied Algebra **69**, 205–218 (1990)
2. Brzozowski, J.A., Cohen, R.S.: Dot-depth of star-free events. Journal of Computer and System Sciences **5**(1), 1–16 (1971)
3. Brzozowski, J.A., Knast, R.: The dot-depth hierarchy of star-free languages is infinite. Journal of Computer and System Sciences **16**(1), 37–55 (1978)
4. Brzozowski, J.A., Simon, I.: Characterizations of locally testable events. Discrete Mathematics **4**(3), 243–271 (1973)
5. Colcombet, T.: Green's relations and their use in automata theory. In: Proceedings of the 5th International Conference on Language and Automata Theory and Applications. pp. 1–21. LATA'11 (2011)
6. Czerwiński, W., Martens, W., Masopust, T.: Efficient separability of regular languages by subsequences and suffixes. In: Proceedings of the 40th International Colloquium on Automata, Languages, and Programming. pp. 150–161. ICALP'13, Springer-Verlag, Berlin, Heidelberg (2013)
7. Diekert, V., Gastin, P.: First-order definable languages. In: Flum, J., Grädel, E., Wilke, T. (eds.) Logic and Automata: History and Perspectives, Texts in Logic and Games, vol. 2, pp. 261–306. Amsterdam University Press (2008)
8. Eilenberg, S.: Automata, Languages, and Machines, vol. B. Academic Press, Inc., Orlando, FL, USA (1976)
9. Higgins, P.: A proof of simon's theorem on piecewise testable languages. Theoretical computer science **178**(1), 257–264 (1997)
10. Higgins, P.M.: A new proof of Schützenberger's theorem. International Journal of Algebra and Computation **10**(02), 217–220 (2000)
11. Kamp, H.W.: Tense Logic and the Theory of Linear Order. Phd thesis, Computer Science Department, University of California at Los Angeles, USA (1968)
12. Klima, O.: Piecewise testable languages via combinatorics on words. Discrete Mathematics **311**(20), 2124–2127 (2011)
13. Knast, R.: A semigroup characterization of dot-depth one languages. RAIRO - Theoretical Informatics and Applications **17**(4), 321–330 (1983)
14. Lucchesi, C.L., Simon, I., Simon, I., Simon, J., Kowaltowski, T.: Aspectos teóricos da computação. IMPA, São Paulo (1979)
15. McNaughton, R., Papert, S.A.: Counter-Free Automata. MIT Press (1971)
16. Meyer, A.R.: A note on star-free events. J. ACM **16**(2), 220–225 (1969)
17. Perrin, D.: Finite automata. In: Formal Models and Semantics. Elsevier (1990)
18. Pin, J.E.: Varieties Of Formal Languages. Plenum Publishing Co. (1986)
19. Pin, J.E.: A variety theorem without complementation. Russian Mathematics (Izvestija vuzov.Matematika) **39**, 80–90 (1995)
20. Pin, J.E.: The dot-depth hierarchy, 45 years later, chap. 8, pp. 177–202. World Scientific (2017)
21. Pin, J.E.: Mathematical foundations of automata theory (2019), in preparation `https://www.irif.fr/~jep/PDF/MPRI/MPRI.pdf`
22. Pin, J.E., Weil, P.: Polynomial closure and unambiguous product. Theory of Computing Systems **30**(4), 383–422 (1997)
23. Pippenger, N.: Theories of computability. Cambridge University Press (1997)
24. Place, T.: Separating regular languages with two quantifier alternations. Logical Methods in Computer Science **14**(4) (2018)

25. Place, T., van Rooijen, L., Zeitoun, M.: Separating regular languages by piecewise testable and unambiguous languages. In: Proceedings of the 38th International Symposium on Mathematical Foundations of Computer Science. pp. 729–740. MFCS'13, Springer-Verlag, Berlin, Heidelberg (2013)
26. Place, T., Zeitoun, M.: Going higher in the first-order quantifier alternation hierarchy on words. In: Proceedings of the 41st International Colloquium on Automata, Languages, and Programming. pp. 342–353. ICALP'14, Springer-Verlag, Berlin, Heidelberg (2014)
27. Place, T., Zeitoun, M.: Separating regular languages with first-order logic. In: Proceedings of the Joint Meeting of the 23rd EACSL Annual Conference on Computer Science Logic (CSL'14) and the 29th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS'14). pp. 75:1–75:10. ACM, New York, NY, USA (2014)
28. Place, T., Zeitoun, M.: The tale of the quantifier alternation hierarchy of first-order logic over words. SIGLOG News **2**(3), 4–17 (2015)
29. Place, T., Zeitoun, M.: Separating regular languages with first-order logic. Logical Methods in Computer Science **12**(1) (2016)
30. Place, T., Zeitoun, M.: Separation for dot-depth two. In: Proceedings of the 32th Annual ACM/IEEE Symposium on Logic in Computer Science, (LICS'17). pp. 202–213. IEEE Computer Society (2017)
31. Place, T., Zeitoun, M.: The covering problem. Logical Methods in Computer Science **14**(3) (2018)
32. Place, T., Zeitoun, M.: Generic results for concatenation hierarchies. Theory of Computing Systems (ToCS) **63**(4), 849–901 (2019), selected papers from CSR'17
33. Place, T., Zeitoun, M.: Going higher in first-order quantifier alternation hierarchies on words. Journal of the ACM **66**(2), 12:1–12:65 (2019)
34. Place, T., Zeitoun, M.: On all things star-free. In: Proceedings of the 46th International Colloquium on Automata, Languages, and Programming. pp. 126:1–126:14. ICALP'19 (2019)
35. Place, T., Zeitoun, M.: Separation and covering for group based concatenation hierarchies. In: Proceedings of the 34th Annual ACM/IEEE Symposium on Logic in Computer Science. pp. 1–13. LICS'19 (2019)
36. Schützenberger, M.P.: On finite monoids having only trivial subgroups. Information and Control **8**(2), 190–194 (1965)
37. Schützenberger, M.P.: Sur le produit de concaténation non ambigu. Semigroup Forum **13**, 47–75 (1976)
38. Simon, I.: Piecewise testable events. In: Proceedings of the 2nd GI Conference on Automata Theory and Formal Languages. pp. 214–222. Springer-Verlag, Berlin, Heidelberg (1975)
39. Straubing, H., Thérien, D.: Partially ordered finite monoids and a theorem of I. Simon. Journal of Algebra **119**(2), 393–399 (1988)
40. Thérien, D., Wilke, T.: Over words, two variables are as powerful as one quantifier alternation. In: Proceedings of the 30th Annual ACM Symposium on Theory of Computing. pp. 234–240. STOC'98, ACM, New York, NY, USA (1998)
41. Wilke, T.: Classifying discrete temporal properties. In: Proceedings of the 16th Annual Conference on Theoretical Aspects of Computer Science. pp. 32–46. STACS'99, Springer-Verlag, Berlin, Heidelberg (1999)
42. Zalcstein, Y.: Locally testable languages. Journal of Computer and System Sciences **6**(2), 151–167 (1972)