Automates à groupe aléatoires

Cyril Nicaud

5 février 1999

Résumé

Un automate est un automate à groupe quand son monoïde des transitions est un groupe. Le but ici est de montrer qu'asymptotiquement presque tous les automates à groupe à n états sont minimaux et d'en déduire un algorithme de complexité linéaire pour tirer au sort un automate à groupe avec n états.

1 Introduction et notations

Le cadre le plus général de cet exposé est l'étude d'algorithmes ([Knu68, Knu69, Knu73, SF96]) et plus particulièrement leur étude en moyenne. On s'intéresse ici aux automates à groupe qui sont des automates dont le monoïde de transitions est un groupe. Les automates à groupes apparaissent sous diverses formes dans la littérature (par exemple [Pin94]).

On ne connait que peu de choses sur le comportement en moyenne des automates. Le cas général semble très difficile à traiter. Cependant, de part leur structure particulière, les automates à groupe s'étudient assez bien en utilisant des propriétés combinatoires non triviales sur les permutations.

Nous proposons ici principalement deux algorithmes. Le premier teste la minimalité d'un automate à groupe en temps moyen linéaire. Le second tire au sort de façon équiprobable un automate à groupe minimal à n états. Le principal intérêt pratique du second algorithme est de pouvoir étudier de façon empirique la complexité de divers algorithmes sur les automates à groupe.

Un automate à groupe $\mathcal{A}$ est un automate déterministe et complet ([Eil74, HU79]) dont le monoïde de transition est un groupe. Celà revient à dire que la fonction de transition associée à chaque lettre de l'alphabet est une permutation de l'ensemble des états, et donc que pour tout lettre a et tout état q, il y a exactement une transition d'étiquette a qui arrive en q et une qui part de q. Ici on ne donne pas, comme c'est parfois le cas pour les automates à groupes, de restrictions sur le nombre d'états finaux. Il est facile de voir que l'automate minimal d'un automate à groupe est un automate à groupe.

On dira que *presque tous les automates de type* $\mathbf{T}$ *vérifient la propriété* $\mathcal{P}$ quand le rapport du nombre d'automates de type $\mathbf{T}$ à n états vérifiant $\mathcal{P}$ sur le nombre d'automates de type $\mathbf{T}$ à n états tends vers 1 quand n tends vers l'infini.

2 Resultat sur un alphabet à une lettre

Un automate à groupe, connexe , sur un alphabet à une lettre est juste une boucle. Comme on étiquette les états et qu'il faut choisir les états finaux il n'est pas difficile de voir qu'il y a exactement $n!2^n$ tels automates avec n états.

Ce qui nous intéresse ici ce sont les automates minimaux. Le cas des alphabet à une lettre, bien que très particulier, nous sera utile par la suite . La démonstration du théorème suivant revient à énumérer les mots circulaires primitifs à n lettres.

Théorème 1 *Il y a moins de $n!2^{(n/2)+1}$ automates à groupes non minimaux, avec n états, sur un alphabet à une lettre.*

3 Presque tous les automates à groupe sont connexes

On va montrer dans cette partie que la proportion d'automates connexes parmi les automates à n états tends vers 1 quand n tend vers l'infini. C'est un premier pas puisqu'un automate minimal est forcément connexe. De plus, d'un point de vu combinatoire, la connexité d'un automate nous assure de la "rigidité" de l'objet combinatoire correspondant, c'est-à-dire qu'il y a exactement $n!$ façons différentes d'étiqueter les n états d'un automate connexe, ce qui n'est pas vrai dans le cas général.

La démonstration du théorème suivant peut se faire de façon similaire à la démonstration combinatoire du fait que presque tous les graphes sont connexes.

Théorème 2 *Presque tous les automates à groupes sont connexes.*

4 Presque tous les automates à groupes sont minimaux

Traiter la minimalité est plus délicat. Nous donnons ici un critère suffisant de minimalité pour un automate à groupe connexe, puis, avec des arguments de combinatoires, nous montrons que presque tous les automates à groupe vérifient ce critère. Ce critère sera utilisé pour faire un algorithme qui teste la minimalité d'un automate à groupe en temps linéaire (en moyenne).

Pour tout automate à groupe à n états $\mathcal{A}$, sur un alphabet A, et pour toute lettre $a \in A$, un a-cycle de $\mathcal{A}$ est une orbite de l'action canonique du groupe engendré par a sur l'ensemble $\{1, \cdots, n\}$. C'est donc l'ensemble des $q.a^i$, $i \in \mathbb{N}$, pour un état $q \in \{1, \cdots, n\}$.

Le critère de minimalité est le suivant: un automate à groupe connexe $\mathcal{A}$ avec n états vérifie **C** quand il existe une lettre $a \in A$ tel qu'il y ait un unique a-cycle de $\mathcal{A}$ de taille maximale et que cette taille soit supérieure à $\sqrt{n}$; de plus, cet a-cycle vu comme un automate sur un alphabet à une lettre est minimal (il n'est pas nécessaire de spécifier l'état initial puisque la minimalité d'un automate à groupe connexe ne dépend pas de son état initial).

Proposition 1 *Tout automate à groupe connexe vérifiant* **C** *est minimal.*

Le résultat de cette partie est le théorème suivant :

Théorème 3 *Presque tous les automates à groupes connexes vérifient* **C**, *et donc presque tous les automates à groupe sont minimaux.*

Pour la démonstration on utilise le résultat suivant, établi avec la précieuse aide de Philippe Flajolet :

Théorème 4 *Le nombre de permutations de $\{1, \cdots, n\}$ qui possèdent un unique cycle de taille maximale est équivalent à $K(n-1)!$ avec $K = \frac{1}{2}e^{\gamma}$. Où γ est la constante d'Euler. Et donc la probabilité qu'une permutation de $\{1, \cdots, n\}$ ait plusieurs cycles de taille maximale est en $O(\frac{1}{n})$.*

5 Les algorithmes

Le premier algorithme proposé est pour tester la minimalité d'un automate à groupe :

```
Teste_minimalité(A)
  1. si A n'est pas connexe retourner Faux
  2. sinon
     2.1. si A vérifie le critère C retourner Vrai
     2.2. sinon utiliser Hopcroft
fin
```

Hopcroft est un algorithme en $O(n \log n)$ qui construit l'automate minimal d'un automate déterministe. Il permet donc de savoir si un automate est minimal ou non. L'algorithme est de complexité linéaire en moyenne. Pour le prouver, on remarque que le test de 2.1. est linéaire et est vérifié suffisament souvent.

Pour tirer au sort et de façon équiprobable un automate à groupe minimal à n états sur un alphabet à k lettres l'algorithme proposé est le suivant :

```
Automate_à_groupe_minimal(n,k)
  1. répèter
     1.1. Tirer au sort un automate à groupe A
          à n états sur un alphabet à k lettres
     Jusqu'à ce que Teste_minimalité(A) soit Vrai
  2. retourner A
fin
```

L'étape 1.1. se fait en temps linéaire et le fait que presque tous les automates vérifient **C** nous permet d'affirmer qu'en moyenne on ne parcourt la boucle qu'un nombre borné de fois. La compléxité de l'algorithme est donc bien linéaire en moyenne.

Références

[Eil74] S. Eilenberg, *Automata, languages and machines*, vol. A, Academic Press, 1974.

[HU79] J. Hopcroft and J.D. Ullman, *Introduction to automata theoriy, language and computation*, Addison Wesley, 1979.

[Knu68] D. E. Knuth, *The art of computer programming*, vol. 1: fundamental algorithms, Addison-Wesley, Reading, MA, 1968.

[Knu69] D. E. Knuth, *The art of computer programming*, vol. 2: seminumerical algorithms, Addison-Wesley, Reading, MA, 1969.

[Knu73] D. E. Knuth, *The art of computer programming*, vol. 3: sorting and searching, Addison-Wesley, Reading, MA, 1973.

[Pin94] J.-E. Pin, *Polynomial closure of group languages and open sets of the hall topology*, ICALP 1994, Lecture notes in Computer Science **820** (1994), 424–435.

[SF96] R. Sedgewick and P. Flajolet, *An introduction to the analysis of algorithms*, Addison-Wesley Publishing Company, 1996.