

Automates et graphes infinis

Abel Douzal

Table des matières

1	Introduction	2
2	Généralités	2
2.1	Bases, langages, mots	2
2.2	Graphes et automates	2
2.3	Reconnaissance de langages	2
2.4	Déterminisme	3
2.5	Isomorphismes de graphes	3
3	Cardinalité	4
3.1	Conséquences générales des contraintes sur les cardinaux	4
3.2	Graphes finis	4
3.3	Graphes infinis de présentation infinie	4
3.4	Graphes infinis de présentation finie	4
3.4.1	Présentations internes	4
3.4.2	Présentations externes	5
3.4.3	Systèmes d'équations et grammaires de graphes	5
3.5	Hierarchie de graphes infinis de présentation finie	5
4	Grammaires de graphes et graphes réguliers	6
4.1	Grammaires déterministes de graphes	6
4.2	Graphes réguliers	7
4.3	Propriétés des graphes réguliers	7
4.4	Liens avec les automates à pile	7

1 Introduction

Dans ce rapport, nous nous intéressons à une généralisation des automates finis : les automates infinis. Après avoir défini les notions de graphe et d'automate, nous nous focaliserons sur les manières de définir et de représenter des graphes infinis. À cette fin, nous distinguerons les présentations infinies internes et externes de graphes, pour ensuite nous intéresser à un type de présentation externe : les grammaires de graphes. On énoncera quelques propriétés des graphes –les graphes réguliers– définis par ces grammaires, puis on en prouvera deux résultats.

2 Généralités

2.1 Bases, langages, mots

On rappelle quelques définitions :

- Un alphabet est un ensemble fini Σ dont les éléments sont appelés lettres.
- Un mot est une séquence finie de lettres ordonnées, que l'on peut considérer comme étant un n -uplet, c'est à dire un élément de $\bigcup_{n \in \mathbf{N}} \Sigma^n = \Sigma^*$.
- Un langage L est une partie de Σ^* .

2.2 Graphes et automates

On utilise les définitions données dans [6].

Définition : Un Automate A est un graphe orienté dont les arêtes sont étiquetées par les lettres d'un alphabet, et dont on a distingué deux sous-ensembles de sommets. Plus précisément, un automate est un quintuplet (Q, Σ, E, I, T) :

- Q est l'ensemble des états de A
- Σ est un ensemble non vide : l'alphabet
- $I, T \subset Q$ sont les états initiaux et finaux (ou terminaux)
- $E \subset Q \times \Sigma \times Q$ est l'ensemble des transitions de l'automate

Définition : Un graphe simple, orienté et étiqueté, ou graphe sur Σ , est un ensemble $G \subset V \times \Sigma \times V$, où Σ est un ensemble fini d'étiquettes et V un ensemble dénombrable. Un élément (s, a, t) de G est un arc de source s , d'extrémité t et d'étiquette a , et se note $s \xrightarrow[a]{} t$.

2.3 Reconnaissance de langages

On peut associer un langage à un graphe en considérant l'ensemble des mots qui étiquettent ses chemins entre deux ensembles de sommets. On appelle traces d'un graphe

G à étiquettes dans l'alphabet Σ entre les ensembles de sommets I et F , ou simplement langage de G , le langage $L(G, I, F) = \{u \in \Sigma^* \mid \exists s \in I, \exists t \in F, s \xrightarrow[u]{} t\}$.

La différence entre graphe et automate est essentiellement une question de vocabulaire, on dira état au lieu de sommet, transition au lieu d'arc, et calcul à la place de chemin. Dans la suite on utilisera essentiellement la terminologie liée aux graphes.

2.4 Déterminisme

Un graphe est dit déterministe (respectivement co-déterministe) s'il ne contient aucune paire d'arcs de même source (resp. extrémité) et de même étiquette.

2.5 Isomorphismes de graphes

Soit G un graphe de support V_G , c'est à dire l'ensemble des sources et extrémités des arcs de G . Son image par une application ϕ de V_G dans un ensemble V est le graphe $\phi(G) = \{(\phi(s), a, \phi(t)) \mid (s, a, t) \in G\}$.

On dit que ϕ est un morphisme du graphe G dans un graphe H si pour tout arc $(s, a, t) \in G$, $(\phi(s), a, \phi(t)) \in H$.

Si de plus ϕ est bijectif et ϕ^{-1} est un morphisme de H à G , alors ϕ est un isomorphisme et on dit que G et H sont isomorphes.

Intuitivement, deux graphes sont isomorphes si ils sont identiques à un renommage des sommets près.

3 Cardinalité

3.1 Conséquences générales des contraintes sur les cardinaux

L'alphabet Σ étant fini, l'ensemble des mots sur l'alphabet Σ est dénombrable, par conséquent, tout langage $L \subset \Sigma$ est dénombrable.

3.2 Graphes finis

Si le graphe d'un automate est fini, alors l'automate sera dit fini. Le principal résultat est le suivant :

Théorème : (Kleene) Les langages rationnels sur Σ^* sont exactement les langages reconnus par les automates finis.

Ceci a pour conséquence que les graphes qui reconnaissent des langages plus riches sont nécessairement infinis.

3.3 Graphes infinis de présentation infinie

Tout graphe infini $G = (V, E)$ dont l'ensemble de sommets V est dénombrable peut être défini en donnant la liste de ses sommets et arêtes (qui est aussi dénombrable). Comme tout langage L sur Σ est dénombrable, il suffit pour le reconnaître de définir le graphe correspondant à l'arbre de recherche d'arité $|\Sigma|$ des mots de L .

Comme cette notion a peu d'intérêt lorsque il s'agit de représenter des graphes, et qu'elle permet de reconnaître tous les langages, on va s'intéresser aux graphes infinis que l'on peut représenter de façon finie.

3.4 Graphes infinis de présentation finie

On présente deux grandes familles de méthodes permettant de définir finiment des graphes infinis : *les représentations internes*, qui consistent à spécifier directement un ensemble fini de relations binaires sur un certain domaine, une pour chaque lettre d'un alphabet d'étiquettes. *Les représentations externes* consistent à décrire directement la structure des graphes, on s'intéressera en particulier aux graphes issus de transformations structurelles à partir de graphes finis [5, 4].

3.4.1 Présentations internes

Les présentations internes consistent à donner de façon explicite, le nom des sommets du graphe et leurs arêtes. Elle consiste à spécifier directement un ensemble fini de relations binaires, une pour chaque lettre de l'alphabet d'étiquettes, entre les éléments d'un certain domaine, par exemple des mots sur un alphabet.

Par exemple, une machine de Turing $(Q, \Sigma, \Gamma, E, q_0, F, \#)$ constitue une présentation interne de son graphe de configurations, où chaque sommet est une configuration, et il existe un arc entre les configurations C et C' si $C = ucpav$ et $C' = uqcbv$ et $(p, a \rightarrow q, b, \triangleleft) \in E$ ou $C = upav$ et $C' = ubqv$ et $(p, a \rightarrow q, b, \triangleright) \in E$, ce qui définit bien une relation binaire sur l'ensemble des configurations (qui est par ailleurs un ensemble dénombrable).

L'intérêt de ce type de présentation est que l'on peut facilement déterminer si il existe une transition entre deux configurations données, ou obtenir une description explicite de l'ensemble des configurations.

3.4.2 Présentations externes

La présentation externe d'un graphe consiste à définir des graphes par transformation à partir d'un graphe donné, en appliquant (potentiellement une infinité de fois) des transformations parmi un ensemble de transformations en nombre fini.

Les transformations que l'on considèrera dans la suite seront des remplacements d'hyperarcs par des hypergraphes. Cela se fera au sein de grammaires de graphes.

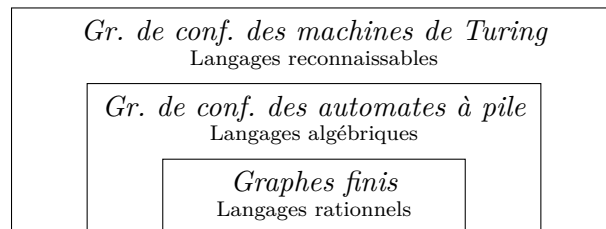
3.4.3 Systèmes d'équations et grammaires de graphes

On peut utiliser des grammaires sur les hypergraphes ayant des règles de dérivation sur les hyperarcs pour définir des familles de graphes : les graphes réguliers. Une étape de dérivation d'une telle grammaire est définie en remplaçant chaque hyperarc étiqueté par un non-terminal A ayant une occurrence dans un hypergraphe G par un hypergraphe H . Une grammaire de graphes est alors constituée d'un ensemble de règles dont le membre gauche est constitué d'un hyperarc et le membre droit d'un hypergraphe.

L'intérêt de ces grammaires réside dans le fait de faciliter l'application de transformations sur le graphe produit, comme le déterminer ou se restreindre à une composante connexe : ce sont *les transformations qui modifient le graphe*. Ou des transformations qui modifient les règles de production dans un but fixé, mais qui laissent invariant (à isomorphisme près) le graphe produit, ce qui est utile lorsque l'on s'intéresse à des propriétés de nature structurelle.

3.5 Hiérarchie de graphes infinis de présentation finie

On obtient la hiérarchie suivante "à la Chomsky" en s'intéressant à certaines familles de graphes infinis de présentation finie :



4 Grammaires de graphes et graphes réguliers

Les notions présentées ici sont essentiellement issues de [5],[4] et [1].

4.1 Grammaires déterministes de graphes

Définition : (hypergraphe) Un hypergraphe H est un couple (V, E) où V est un ensemble non vide et E est une famille de parties non vides de V .

- Les éléments de V sont appelés sommets de H .
- Les éléments de E sont les hyperarcs, arcs, ou encore arêtes de H .
- Si on étiquette les éléments de E par un alphabet Σ on dira que H est un hypergraphe étiqueté.

Définition : (HR-grammaire) Une grammaire d'hyperarcs, ou grammaire à remplacement d'hyperarcs, ou encore HR-grammaire G , est un quadruplet (N, T, H_0, R) , où :

1. N est l'alphabet des symboles non-terminaux
2. T est l'alphabet des symboles terminaux
3. H_0 est l'axiome, c'est à dire un hypergraphe fini formé d'hyperarcs étiquetés par $N \cup T$
4. R est un ensemble fini de règles de la forme $(A, x_1, \dots, x_{\rho(A)}) \rightarrow H_A$ où $(A, x_1, \dots, x_{\rho(A)})$ est un hyperarc qui connecte des sommet finis, et H_A est un hypergraphe fini qui contient $x_1, \dots, x_{\rho(A)}$.

Remarques :

- Une grammaire est dite *déterministe* si elle ne possède qu'une règle de réécriture par non-terminal.
- Comme on cherche à obtenir des graphes, l'alphabet des symboles terminaux se limitera à ceux d'ordre un ou deux (on identifie les hyperarcs d'arité deux à des arcs).

Définition : (Relation de réécriture) Soient M et P deux hypergraphes. et $G = (N, T, H_0, R)$ une HR-grammaire.

- On dit que M se réécrit en P , et on notera $M \xrightarrow{R} P$, si il y a un hyperarc non-terminal $X = (A, v_1, v_2, \dots, v_p)$ dans M et une règle $(A, x_1, x_2, \dots, x_p) \rightarrow H_A$ dans R tels que P est obtenu en remplaçant X par H_A dans M : $P = (M - X) \cup h(H_A)$ pour une injection h , qui associe v_i à x_i pour tout $i \in \{1, \dots, p\}$, et tous les autres sommets de H_A à de nouveaux sommets qui ne sont pas dans M . Une telle réécriture est notée $M \xrightarrow{R, X} P$.

- Cette notion de réécriture s'étend aux ensembles de non-terminaux : pour $E \subset N$ on note la réécriture par $E : M \xrightarrow{R,E} P$.
La réécriture est dite *parallèle* si il s'agit de la réécriture de l'ensemble des non-terminaux : $M \xrightarrow{R,N} P$
- La réécriture complète et parallèle est l'opération de réécriture relative à l'ensemble de tous les arcs non-terminaux de R . Elle est notée $\xrightarrow{R}$.

4.2 Graphes réguliers

Soit une HR-grammaire déterministe $G = (N, T, H_0, R)$, pour un hypergraphe H , on note $[H]$ l'ensemble des arcs terminaux de H . G^ω est défini par :

$$G^\omega = \left\{ \bigcup_{n \geq 0} [H_n] \mid \forall n \geq 0, H_n \xrightarrow{R} H_{n+1} \right\}$$

G^ω est un ensemble de graphes, et on dit qu'un graphe H est engendré par G lorsqu'il appartient à G^ω .

On constate que tous les graphes de G^ω sont isomorphes entre eux. Comme on s'intéresse à des propriétés qui sont invariantes par isomorphisme, on identifiera à isomorphisme près G^ω à un de ses éléments. On qualifiera cet élément de graphe engendré par G .

Définition : On appelle *graphes réguliers* les graphes engendrés par les grammaires déterministes de graphe.

4.3 Propriétés des graphes réguliers

Un graphe régulier peut posséder des sommets de degré infini :

Exemple La grammaire :



Les graphes réguliers peuvent donc avoir des sommets de degré infini. Cependant, l'ensemble des degrés des sommets est toujours fini, il est de plus calculable.

4.4 Liens avec les automates à pile

On obtient la caractérisation suivante, lorsque l'on contraint le degré des sommets à être fini :

Théorème : Les graphes réguliers de degré fini sont les graphes de transition des automates à pile. [2].

On se limite à présent à des grammaires pour lesquelles chaque membre gauche est un arc étiqueté du sommet 1 au sommet 2, et il n'y a pas dans les membres droits d'arc sortant de 2, ni d'arc entrant en 1.

On montre deux résultats réciproques, issus de [3] :

Proposition 1 : On peut transformer toute grammaire algébrique $K = (N_K, T_K, P)$ en une grammaire de graphes $G = (N_G, T_G, (\{1, 2\}, \{1 \xrightarrow{A} 2\}), \{(A, 1, 2) \longrightarrow H_A | A \in N\})$ sans circuit tel que $\underline{L_G(A) = L_K(A)}$ pour tout non-terminal $A \in N_K$ où $L_G(A) = L(G^\omega, 1, 2)$.

Proposition 2 : On peut transformer toute grammaire G de graphes en une grammaire algébrique K telle que $\underline{L_G(A) = L_K(A)}$ pour tout $A \in N_G$.

Preuve 1 : Pour tout non terminal $A \in N_K$, on définit l'hypergraphe :

$$\begin{aligned} H_A = & \{1 \xrightarrow{a} a.v \mid (A, av) \in P \text{ et } a \in N_K \cup T_K \text{ et } v \neq \varepsilon\} \\ & \cup \{u.a \xrightarrow{a} 2 \mid (A, ua) \in P \text{ et } a \in N_K \cup T_K \text{ et } u \neq \varepsilon\} \\ & \cup \{1 \xrightarrow{a} 2 \mid (A, a) \in P \text{ et } a \in N_K \cup T_K\} \\ & \cup \{u.av \xrightarrow{a} ua.v \mid (A, uav) \in P \text{ et } a \in N_K \cup T_K \text{ et } u, v \neq \varepsilon\} \end{aligned}$$

De plus, on définit $N_G = N_K$ et $T_G = T_K$. On a alors défini G grammaire de graphes reconnaissant le même langage que K . $\square$

Preuve 2 : Pour chaque non-terminal $A \in N_G$, on définit un renommage h_A des sommets du membre droit des règles de productions de $R : H_A : h_A(1) = A, h_A(2) = \varepsilon$, et pour tout autre sommet de H_A , un symbole arbitraire, de façon à ce que h_A soit injective, et que pour deux non-terminaux $A, B \in N_G, A \neq B$ on ait $h_A(x) = h_B(y)$ où x est sommet de H_A et y sommet de H_B ssi $h_A(x) = h_B(y) = \varepsilon$.

On définit la grammaire algébrique $K = (N_K, T_K, P)$ par :

$$P = \{(h_A(s), ah_A(t)) \mid A \in N_G \text{ et } (s \xrightarrow{a} t) \in E(H_A)\}$$

$$N_K = N_G$$

$$T_K = \left(\bigcup_{A \in N_G} h_A(V(H_A)) \right) - N_G$$

Comme les langages reconnus par les automates à piles sont exactement les langages algébriques, on a le résultat recherché. $\square$

Références

- [1] Arnaud Carayol. *Automates infinis, logiques et langages*. PhD thesis, Université Rennes 1, 2006.
- [2] Didier Caucal. Deterministic graph grammars. *Logic and automata*, 2 :169–250, 2008.
- [3] Trong Hiêu Dinh. *Grammaires de graphes et langages formels*. PhD thesis, Paris Est, 2011.
- [4] Antoine Meyer. *Graphes infinis de présentation finie*. PhD thesis, Université Rennes 1, 2005.
- [5] Christophe Morvan. *Familles de graphes de présentation finie, propriétés et applications*. PhD thesis, Université Paris-Est, 2014.
- [6] Jacques Sakarovitch. *Éléments de théorie des automates*. Vuibert, 2003.