1 Automates à pile

Comme nous l'avons vu en cours, il existe des langages algébriques qui sont non rationnels. Cela implique que le modèle de calcul d'automate fini n'est pas assez puissant pour exprimer l'ensemble des langages algébriques. L'idée principale est qu'un automate fini n'a qu'une mémoire fini, ce qui est insuffisant pour reconnaître des langages définis par induction. On présente dans ce chapitre hors programme un modèle d'automate doté d'une mémoire infinie : une pile non bornée.

Définition 1.1

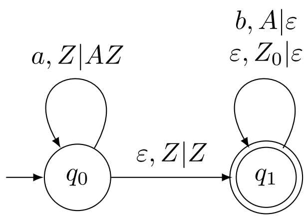
Un **automate à pile non déterministe** (APND) A sur Σ est un septuplet $(Q, \Sigma, \Gamma, \Delta, q_0, Z_0, F)$ où :

- Q est un ensemble fini d'**états** ;
- Σ est un alphabet fini ;
- Γ est un alphabet fini appelé **alphabet de pile** ;
- Δ est une **fonction de transition** de $Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma$ dans $\mathcal{P}(Q \times \Gamma^*)$. Si $(q, \beta) \in \Delta(p, a, Z)$, on note simplement $p \xrightarrow{a,Z|\beta} q$ (notation qui s'inspire de la représentation graphique des transitions) ;
- $q_0 \in Q$ est **l'état initial** ;
- $Z_0 \in \Gamma$ est le **symbole de pile initial** ;
- $F \subseteq Q$ est un ensemble d'**états finaux**.

On utilise une représentation graphique similaire à celle des automates finis pour représenter un APND.

Exemple 1.2

L'automate A_0 suivant est un automate à pile pour $\Sigma = \{a, b\}$ et $\Gamma = \{A, Z_0\}$.



où Z désigne A ou Z_0 indifféremment.

La sémantique des automates à pile est la suivante : la pile ne contient initialement que le symbole de pile initial. Pour effectuer une transition, on dépile le symbole du haut de la pile, et en fonction de ce symbole, de l'état et de la lettre (ou ε) lue, on change d'état et on réempile éventuellement de nouveaux symboles dans la pile. Pour déterminer si un mot est accepté, on peut étudier la pile et l'état atteint après sa lecture.

Définition 1.3

On définit les notions suivantes :

- Une **configuration** de $\mathcal{A}$ est un couple (q, γ) où $q \in Q$ est l'**état** de l'automate et $\gamma \in \Gamma^*$ est la **pile** de l'automate (le haut de la pile correspond à la gauche de γ).
- La configuration **initiale** est (q_0, Z_0).
- Une **transition de configurations** est définie de la manière suivante : lorsque l'automate est dans une configuration (p, γ), avec $\gamma = Z\beta$ où $\beta \in \Gamma^*$ et $Z \in \Gamma$, et lit une lettre du mot d'entrée $a \in \Sigma$ (ou ne lit aucune lettre dans le cas $a = \varepsilon$), alors il peut passer dans une des configurations $(q, \alpha\beta)$ telle que $(q, \alpha) \in \Delta(p, a, Z)$. (on peut avoir $\alpha = \varepsilon$). On note alors $(p, Z\beta) \xrightarrow{a} (q, \alpha\beta)$.

À partir des configurations, on peut définir différents modes d'acceptation d'un mot :

Définition 1.4

Soit $u = a_1 \dots a_n \in \Sigma^*$ et A un APND. S'il existe une suite de configurations $(q_0, Z_0) \xrightarrow{b_1} \dots \xrightarrow{b_m} (q, \gamma)$ telle que $a_1 \dots a_n = b_1 \dots b_m$, on dit que u est reconnu par A par :

- pile vide si $\gamma = \varepsilon$;
- état final si $q \in F$;
- pile vide et état final si $\gamma = \varepsilon$ et $q \in F$.

On note $L_P(A)$, $L_F(A)$ et L_{PF} l'ensemble des mots reconnus par A pour chacun des modes d'acceptation précédents.

Exemple 1.5

Pour l'automate A_0 précédent, on peut montrer que $L_P(A_0) = L_{PF}(A_0) = \{a^n b^n \mid n \in \mathbb{N}\}$ et $L_F(A_0) = \{a^n b^m \mid n \geqslant m\}$.

Exercice 1

Pour chacun des langages suivants sur $\Sigma = \{a, b\}$, déterminer sans justifier un APND qui les reconnaît par pile vide et état final.

1. $\{u\tilde{u} \mid u \in \Sigma^*\}$ où $\tilde{u}$ désigne le miroir de u ;
2. $\{a^n b^m \mid n \leqslant m \leqslant 2n\}$;
3. $\{u \in \Sigma^* \mid |u|_a = |u|_b\}$;
4. le langage de Dyck, c'est-à-dire l'ensemble des mots bien parenthésés, avec a assimilé à une parenthèse ouvrante et b à une parenthèse fermante.

Exercice 2

Soit L un langage rationnel. Montrer qu'il existe un APND reconnaissant L par pile vide et état final.

Corrigé

On construit un automate en transformant chaque transition de l'automate sans pile en une transition de l'automate à pile qui ne change pas le symbole de pile. Pour s'assurer de l'acceptation par pile vide (en plus de l'état final), on rajoute une transition $q \xrightarrow{\varepsilon, Z_0 | \varepsilon} q$ pour chaque $q \in F$.

Théorème 1.6

Soit $L \subseteq \Sigma^*$. Alors il y a équivalence entre les trois propriétés suivantes :

1. il existe un APND A tel que $L = L_F(A)$;
2. il existe un APND A tel que $L = L_P(A)$;
3. il existe un APND A tel que $L = L_{PF}(A)$.

Preuve

- L reconnu par état final $\Rightarrow$ L reconnu par pile vide :

  On rajoute un symbole de pile initial X_0 à l'alphabet de pile, un état initial p_0 et un état puit s. On rajoute une transition $p_0 \xrightarrow{\varepsilon, X_0 | Z_0 X_0} q_0$ (le symbole X_0 permettant d'éviter de vider la pile dans un état non final). On rajoute également des transitions $q \xrightarrow{\varepsilon, Z | \varepsilon} s$, pour Z symbole de pile quelconque et $q \in F \cup \{s\}$ (permettant de vider « gratuitement » la pile une fois qu'on est dans un état final).

– L reconnu par pile vide $\Rightarrow$ L reconnu par état final et pile vide :

On transforme chaque état en état final.

– L reconnu par état final et pile vide $\Rightarrow$ L reconnu par état final :

On rajoute un symbole de pile initial X_0 à l'alphabet de pile, un état initial p_0 et un état final p_f. On rajoute une transition $p_0 \xrightarrow{\varepsilon, X_0 | Z_0 X_0} q_0$ (le symbole X_0 permet de savoir à quel moment on a un pile vide). On rajoute de plus, pour chaque $q \in F$, une transition $q \xrightarrow{\varepsilon, X_0 | \varepsilon} p_f$. Enfin, on ne garde que p_f comme état final.

1.1 Automates déterministes

Comme pour les automates finis, on peut s'intéresser aux automates déterministes, dont l'utilisation est plus naturelle que le non déterminisme.

Définition 1.7

Un automate à pile $(Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ est dit **déterministe** (APD) si pour $q \in Q$, $a \in \Sigma$ et $Z \in \Gamma$:

$$|\delta(q, a, Z) \cup \delta(q, \varepsilon, Z)| \leqslant 1$$

C'est-à-dire qu'il n'existe toujours qu'une seule transition possible, et qu'on n'a jamais à choisir entre lire une lettre ou effectuer une ε-transition. On utilise les mêmes notations et définitions pour les langages reconnus selon les modes d'acceptation.

Exercice 3

Montrer qu'il existe un APD A tel que $L_F(A) = \{a^n b^n \mid n \in \mathbb{N}\}$.

Proposition 1.8

Pour tout langage rationnel L, il existe un APD A tel que $L = L_F(A)$.

Preuve

C'est plus ou moins la preuve de l'exercice 2.

Proposition 1.9

Il existe un langage rationnel L tel qu'il n'existe aucun APD A tel que $L = L_P(A)$.

Preuve

Le langage $L = 0^*$ convient. En effet, si on suppose l'existence d'un tel automate A, comme $\varepsilon \in L$, alors $\delta(q_0, \varepsilon, Z_0)$ est non vide et de la forme (p, ε). Cela implique que $\delta(q_0, 0, Z_0)$ est vide et donc que le mot 0 n'est pas reconnu.

Définition 1.10

Un langage L a la **propriété des préfixes** si $\forall x, y \in L$, x n'est pas un préfixe de y.

Proposition 1.11

Soit $L \subseteq \Sigma^*$. Alors il y a équivalence entre les deux propriétés suivantes :

1. il existe un APD A tel que $L = L_P(A)$;

2. L a la propriété des préfixes et il existe un APD A tel que $L = L_F(A)$.

Preuve

L'interprétation en termes d'automates de la propriété des préfixes est qu'il n'existe pas de chemin d'une configuration acceptante vers une autre configuration acceptante.

– ($\Rightarrow$) Soit $\mathcal{A}$ un automate à pile déterministe reconnaissant un langage L par pile vide. Soit x un mot reconnu par $\mathcal{A}$. L'automate étant déterministe, la lecture de x mène de (q_0, Z_0) à une configuration de la forme (q, ε) acceptante. Cependant, il ne peut pas y avoir de successeur à cette configuration (car la pile est vide).
Pour créer un automate à pile déterministe reconnaissant L par état final, il suffit de rajouter un état q_f, et de modifier chaque transition de la forme $\delta(p, a, Z_0) = \{(q, \varepsilon)\}$ en $\delta(p, a, Z_0) = \{(q_f, \varepsilon)\}$. Enfin, on fait en sorte que seul q_f soit un état final.

– ($\Leftarrow$) Soit L un langage vérifiant la propriété des préfixes et $\mathcal{A}$ un automate à pile déterministe reconnaissant un langage L par état final. Comme L vérifie la propriété des préfixes, on peut supprimer les transitions sortantes des états finaux sans changer le langage reconnu par état final. Dès lors, on peut rajouter, pour chaque état $q \in F$, une transition $\delta(q, \varepsilon, Z) = \{(q, \varepsilon)\}$ pour tout $Z \in \Gamma$ et reconnaître le langage par pile vide. On supprime également toute transition de la forme $\delta(q, a, Z_0) = \{(p, \varepsilon)\}$ où $p \notin F$.

Proposition 1.12

Soit $L = \{u\tilde{u} \mid u \in \{a, b\}^*\}$. Alors il n'existe aucun APD A tel que $L = L_F(A)$.

Preuve

Admis.

2 Équivalence avec les langages algébriques

Dans cette section, on montre un théorème similaire au théorème de Kleene, à savoir :

Théorème 2.1

Soit $L \subseteq \Sigma^*$. Alors il y a équivalence entre :

1. L est algébrique.
2. L est reconnu par un APND.

On commence par montrer le sens direct du théorème :

Théorème 2.2

Soit $G = (\Sigma, V, P, S)$ une grammaire hors-contexte. Alors il existe un APND $A = (Q, \Sigma, \Gamma, \Delta, q_0, Z_0, F)$ tel que $L(G) = L_{PF}(A)$.

Preuve

On pose $Q = \{q_0, q_1, q_2\}$, $F = \{q_2\}$, $\Gamma = \{Z_0\} \cup \Sigma \cup V$ et Δ défini par :

– $\Delta(q_0, \varepsilon, Z_0) = (q_1, SZ_0)$;
– pour $X \to \alpha \in P$, $(q_1, \alpha) \in \Delta(q_1, \varepsilon, X)$;
– pour $a \in \Sigma$, $(q_1, \varepsilon) \in \Delta(q_1, a, a)$;
– $(q_2, \varepsilon) \in \Delta(q_1, \varepsilon, Z_0)$.

Alors $L(G) = L_{PF}(A)$. En effet, par construction, il y a équivalence entre $X \Rightarrow_g^* u\alpha$ et $\forall \beta \in \Gamma^*$, $(q_1, X\beta) \xrightarrow{u}^* (q_1, \alpha\beta)$. L'idée est que dans une dérivation immédiate gauche, la variable X remplacée permet d'ajouter dans la pile le mot α. Lors qu'on lit des lettres de Σ dans le mot u et en haut de la pile, on les dépile en avançant dans le mot. Le symbole Z_0 permet de passer à l'état 2 lorsqu'on a terminé les dérivations.

Théorème 2.3

Soit $A = (Q, \Sigma, \Gamma, \Delta, q_0, Z_0, F)$ un APND. Alors il existe une grammaire hors-contexte $G = (\Sigma, V, P, S)$ telle que $L(G) = L_P(A)$.

Preuve

On pose $V = (Q \times \Gamma \times Q) \cup \{S\}$. L'idée est de créer les règles de production de telle sorte que $(q_0, Z_0, q) \Rightarrow^* u$ si et seulement si $(q_0, Z_0) \xrightarrow{u}^* (q, \varepsilon)$.

Pour cela P contenant les règles de production suivantes :

- $S \to (q_0, Z_0, q)$ pour tout $q \in Q$;
- si $(q, \beta) \in \Delta(p, a, Z)$, avec $a \in \Sigma \cup \{\varepsilon\}$, $Z \in \Gamma$ et $\beta = Z_1 Z_2 \ldots Z_k \in \Gamma^*$, alors pour chaque k-uplet $(q_1, q_2, \ldots, q_k) \in Q^k$, on considère $(p, Z, q_k) \to a(q, Z_1, q_1)(q_1, Z_2, q_2)\ldots(q_{k-1}, Z_k, q_k)$

On peut alors montrer par induction que $(p, Z) \xrightarrow{u}^* (q, \varepsilon)$ si et seulement si $(p, Z, q) \Rightarrow^* u$, ce qui permet de conclure (puisqu'on accepte par pile vide).