

Théorie des Langages Automates à pile

Yannick Le Nir Gaspard Férey

Contributions de :
Taisa Guidini Goncalves

CY TECH

yannick.lenir@eisti.fr gaspard.ferey@inria.fr

Langages décidables et hiérarchie de classes

- ▶ Les langages de type 3 : rationnels ou réguliers
- ▶ Les langages de type 2 : algébriques ou hors contexte
- ▶ Les langages de type 1 : sensibles au contexte
- ▶ Les langages de type 0 : tous les autres décidables

Chaque ensemble est strictement inclus dans ceux de numéro inférieur.

Grammaire de type 2

Définition

Une grammaire $G = \langle T, N, S, P \rangle$ est de type 2 si les règles de production sont de la forme :

$A \rightarrow \alpha$ où $A \in N$ et $\alpha \in (N \cup T)^*$

Langage associé

Un langage est de type 2 s'il peut être engendré par une grammaire de type 2.

Grammaires de type 2

Exemple de grammaire

Le fameux langage $a^n b^n$ n'est pas régulier mais peut être engendré par la grammaire hors contexte suivante :

- ▶ $T = \{a, b\}$
- ▶ $N = \{S\}$
- ▶ $S = S$
- ▶ $P = \left\{ \begin{array}{l} S \rightarrow a S b \\ S \rightarrow a b \end{array} \right\}$

Domaines d'application

- ▶ Langages de programmation
- ▶ La plupart des constructions des langues naturelles

Présentation

- ▶ Classe de machines abstraites plus puissantes que les automates d'états finis.
- ▶ Capables de reconnaître les expressions parenthésées et les imbrications de blocs `begin-end`.
- ▶ Augmentation de la puissance des automates finis en ajoutant une pile simulant une mémoire.

Définition

Un automate à pile **non déterministe** est défini par un septuplet $M = (Q, \Sigma, \Gamma, \Delta, Z, S, F)$

- ▶ Q est un ensemble fini d'états
- ▶ Σ est un alphabet d'entrée
- ▶ Γ est un alphabet de pile
- ▶ $\Delta \subset ((Q * \Sigma^* * \Gamma^*) * (Q * \Gamma^*))$ est la relation de transition
- ▶ $Z \in \Gamma$ est le symbole initial de pile
- ▶ $S \in Q$ est l'état initial
- ▶ $F \subseteq Q$ est l'ensemble des états terminaux (acceptants)

Automates à pile

Opérations

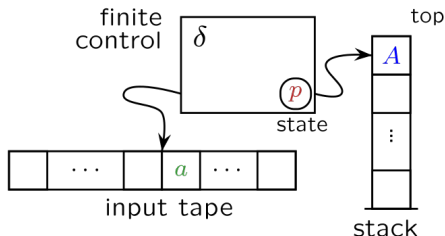
- ▶ Lire un symbole (et avancer la tête de lecture)
- ▶ Dépiler un symbole
- ▶ Empiler un (ou plusieurs) symboles
- ▶ Changer d'état

Contributions de :
Taisa Guidini
Goncalves

Langages
algébriques

Automates à pile

Langage reconnu



Transition

$(p, x, y; q, z)$

- ▶ p est l'état courant
- ▶ x est le symbole en entrée
- ▶ y est le symbole dépilé
- ▶ q est le nouvel état
- ▶ z est le symbole empilé

On autorise $z = z_1 \dots z_n$ qui empile plusieurs symboles.

Remarques

- ▶ Le y est dépilé avant que le z ne soit empilé.
- ▶ L'automate n'est pas obligé de faire à la fois une lecture, un empilement et un dépilement à chaque transition.
- ▶ Si aucun symbole n'est lu (resp. dépilé) (resp. empilé) , x (resp. y) (resp. z) est remplacé par λ .

Note :

Ces conventions (λ -transitions, empilements multiples) sont autorisées car on peut prouver qu'elles ne permettent pas de calculer plus de choses.

De la même façon, on autorise le non-détermisme, les ε -transitions et l'incomplétude pour les automate finis.

Exemples

- La transition $(1, a, t; 2, b)$ signifie que l'automate passe de l'état 1 à l'état 2, en lisant un a , en dépilant un t et en empilant un b .

Exemples

- ▶ La transition $(1, a, t; 2, b)$ signifie que l'automate passe de l'état 1 à l'état 2, en lisant un a , en dépilant un t et en empilant un b .
- ▶ La transition $(p, \lambda, \lambda; q, \lambda)$ signifie que l'automate passe de l'état p à l'état q (aucune lecture, ni empilement, ni déplacement).

Exemples

- ▶ La transition $(1, a, t; 2, b)$ signifie que l'automate passe de l'état 1 à l'état 2, en lisant un a , en dépilant un t et en empilant un b .
- ▶ La transition $(p, \lambda, \lambda; q, \lambda)$ signifie que l'automate passe de l'état p à l'état q (aucune lecture, ni empilement, ni déplacement).
- ▶ La transition $(p, \lambda, 2; q, \lambda)$ signifie que l'automate passe de l'état p à l'état q en enlevant le symbole 2 de la pile.

Exemples

- ▶ La transition $(1, a, t; 2, b)$ signifie que l'automate passe de l'état 1 à l'état 2, en lisant un a , en dépilant un t et en empilant un b .
- ▶ La transition $(p, \lambda, \lambda; q, \lambda)$ signifie que l'automate passe de l'état p à l'état q (aucune lecture, ni empilement, ni déplacement).
- ▶ La transition $(p, \lambda, 2; q, \lambda)$ signifie que l'automate passe de l'état p à l'état q en enlevant le symbole 2 de la pile.
- ▶ La transition $(p, a, \lambda; q, c)$ signifie que l'automate passe de l'état p à l'état q en lisant le symbole a et en plaçant c sur la pile.

Représentation

- ▶ Les diagrammes de transition des automates à pile sont identiques à ceux des automates d'états finis sauf que l'étiquette d'un arc est plus complexe.
- ▶ Ces étiquettes ont la forme $x, y; z$ avec :
 - ▶ x le symbole lu
 - ▶ y le symbole dépilé
 - ▶ z le symbole empilé

Exemple

Soit le Langage $a^n b^n$ avec $n \geq 1$.

- ▶ $Q = \{q_0, q_1\}$
- ▶ $\Sigma = \{a, b\}$
- ▶ $\Gamma = \{A\}$ alphabet de pile
- ▶ $Z \in \Gamma$ est le symbole initial de pile
- ▶ $S = q_0$
- ▶ Il n'y a pas d'état final. La reconnaissance est par pile vide. Il y a un seul symbole de alphabet de pile, noté A.

Automates à pile

Exemple

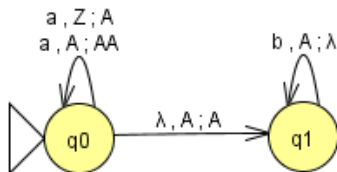
Les transitions sont :

1 : (q_0, a, Z, q_0, A)

2 : (q_0, a, A, q_0, AA)

3 : $(q_0, \varepsilon, A, q_1, A)$

4 : $(q_1, b, A, q_1, \varepsilon)$



Propriétés

- ▶ L'automate à pile est démarré dans l'état initial avec la pile vide.
- ▶ Il accepte la séquence d'entrée ssi il *peut* atteindre un état final *après* l'avoir lue en entier.
 - ▶ "*peut*" : car la définition d'un automate à pile n'exige pas que l'ensemble des transitions représente une fonction ; les automates à pile peuvent être non déterministes.
 - ▶ "*après*" : il n'est pas nécessaire que l'automate à pile atteigne un état final tout de suite après avoir lu le dernier symbole d'entrée. Après avoir terminé la lecture de la séquence d'entrée, l'automate peut effectuer des transitions sans lecture, en manipulant sa pile et atteindre de cette façon un état final.

Langage accepté

Le langage accepté par un automate à pile M est noté $L(M)$.

C'est l'ensemble des séquences acceptées

(i.e. $L(M) = \{w \in \Sigma^* \mid w \text{ est accepté}\}$)

Théorème

Tout langage accepté par un automate à pile est algébrique.

Théorème

Tout langage algébrique est accepté par un automate à pile.