Langages et Automates : LA3
Partie 8 : Automates a Pile

Automates à Pile

On a vu l'équivalence :
Langages Rationnels <-> Automates Finis <-> Grammaries Régulières.

Automates à Pile

On a vu l'équivalence :
Langages Rationnels <-> Automates Finis <-> Grammaries Régulières.

Pour les Langages Algébriques, on aimerait définir un objet généralisant les Automates finis.
$\Rightarrow$ Automates à Pile.

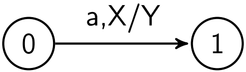
Automates à Pile

Un Automate à Pile = AFND avec ε-transitions + une PILE (avec un alphabet spécifique pour la pile).

Automates à Pile

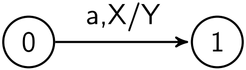
Un Automate à Pile = AFND avec ε-transitions + une PILE (avec un alphabet spécifique pour la pile).

Chaque transition précise en plus de la lettre lue, le symbole de pile X qu'on doit trouver en haut de la pile et le symbole Y par lequel on doit le remplacer.



Un Automate à Pile = AFND avec ε-transitions + une PILE (avec un alphabet spécifique pour la pile).

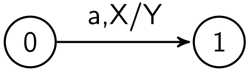
Chaque transition précise en plus de la lettre lue, le symbole de pile X qu'on doit trouver en haut de la pile et le symbole Y par lequel on doit le remplacer.



Si en haut de la pile ce n'est pas le bon symbole, alors on ne peut pas emprunter cette transition. Si c'est le bon symbole, on effectue l'opération sur la pile.

Un Automate à Pile = AFND avec ε-transitions + une PILE (avec un alphabet spécifique pour la pile).

Chaque transition précise en plus de la lettre lue, le symbole de pile X qu'on doit trouver en haut de la pile et le symbole Y par lequel on doit le remplacer.



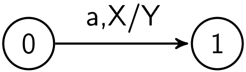
Si en haut de la pile ce n'est pas le bon symbole, alors on ne peut pas emprunter cette transition. Si c'est le bon symbole, on effectue l'opération sur la pile.

X ou Y peuvent être égaux à ε. Si c'est X cela signifie que l'on ne se soucie pas du symbole en haut de la pile et qu'on ne dépile rien. Si c'est Y alors on ne rempile rien.

Automates à Pile

Un Automate à Pile = AFND avec ε-transitions + une PILE (avec un alphabet spécifique pour la pile).

Chaque transition précise en plus de la lettre lue, le symbole de pile X qu'on doit trouver en haut de la pile et le symbole Y par lequel on doit le remplacer.



Si en haut de la pile ce n'est pas le bon symbole, alors on ne peut pas emprunter cette transition. Si c'est le bon symbole, on effectue l'opération sur la pile.

X ou Y peuvent être égaux à ε. Si c'est X cela signifie que l'on ne se soucie pas du symbole en haut de la pile et qu'on ne dépile rien. Si c'est Y alors on ne rempile rien.

La lettre a peut aussi être ε : dans ce cas cela se passe comme pour les ε-transitions dans les automates simples. On peut avancer sans lire de lettre du mot d'entrée (mais on effectue quand même les opérations de pile).

Automates à Pile Non Déterministe

Définition

Un automate à pile non déterministe est un septuplet $M = (\Sigma, \Gamma, Q, q_0, Z_0, F, \delta)$ *où*

- Σ *est l'alphabet d'entrée (fini)*
- Γ *est l'alphabet de pile.*
- Q *est un ensemble fini d'états*
- $q_0 \in Q$ *est l'état initial.*
- $Z_0 \in \Gamma$ *est le symbole de pile initial (ce que contient la pile au début).*
- $F \subset Q$ *est l'ensemble des états acceptants.*
- δ *est la fonction de transition. Elle va de* $Q \times (\Sigma \cup \{\varepsilon\}) \times (\Gamma \cup \{\varepsilon\})$ *vers l'ensemble des parties de* $Q \times (\Gamma \cup \{\varepsilon\})$.

Automates à Pile Non Déterministe

Définition

Un automate à pile non déterministe est un septuplet $M = (\Sigma, \Gamma, Q, q_0, Z_0, F, \delta)$ *où*

- Σ *est l'alphabet d'entrée (fini)*
- Γ *est l'alphabet de pile.*
- Q *est un ensemble fini d'états*
- $q_0 \in Q$ *est l'état initial.*
- $Z_0 \in \Gamma$ *est le symbole de pile initial (ce que contient la pile au début).*
- $F \subset Q$ *est l'ensemble des états acceptants.*
- δ *est la fonction de transition. Elle va de* $Q \times (\Sigma \cup \{\varepsilon\}) \times (\Gamma \cup \{\varepsilon\})$ *vers l'ensemble des parties de* $Q \times (\Gamma \cup \{\varepsilon\})$.

Chaque transition de l'automate associe a un état q, une lettre a, un symbole de pile X à dépiler, un nouvel état q' et un symbole de pile Y à réempiler. L'automate n'est pas déterministe : il peut exister plusieurs transitions pour le meme triplet (q, a, X). (C'est pourquoi $\delta(q, a, X)$ est un ensemble de couples (q', Y).)

Automates à Pile Non Déterministe

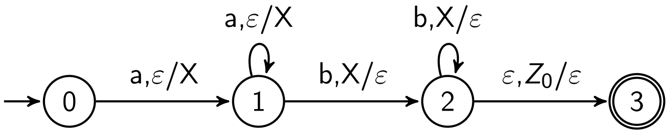
Un mot est reconnu si son calcul finit dans un état acceptant ET que la pile est vide.

Le langage reconnu par l'automate à pile est l'ensemble des mots reconnus.

Les cas des automates traditionnels correspond au cas ou la pile est vide au début et où on ne fait aucune opération sur la pile (transitions $a, \varepsilon/\varepsilon$)

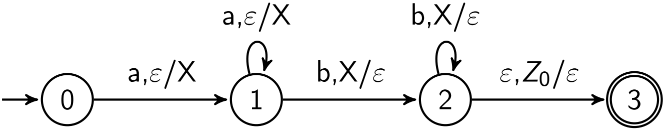
Automates à Pile - Un Exemple

L'automate suivant reconnait les $a^n b^n$



Automates à Pile - Un Exemple

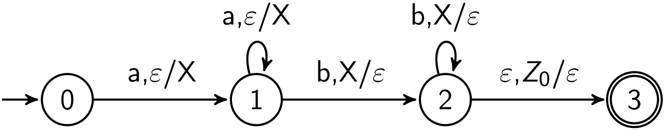
L'automate suivant reconnait les $a^n b^n$



La transition de 1 a 2 doit être comprise ainsi :
On veut lire un b, on cherche a dépiler X, si c'est bien le symbole au dessus de la pile, alors on le dépile et on rempile ε, c'est à dire qu'on ne rempile rien.

Automates à Pile - Un Exemple

L'automate suivant reconnait les $a^n b^n$

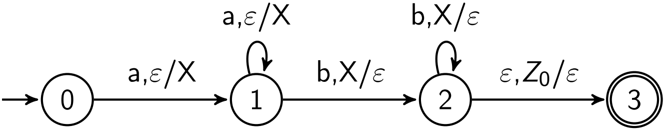


La transition de 1 a 2 doit être comprise ainsi :
On veut lire un b, on cherche a dépiler X, si c'est bien le symbole au dessus de la pile, alors on le dépile et on rempile ε, c'est à dire qu'on ne rempile rien.

La boucle au dessus de 1 :
On veut lire un a, on se moque de ce qu'il ya dessus sur la pile (ε), mais on empile un symbole X.

Automates à Pile - Un Exemple

L'automate suivant reconnait les $a^n b^n$



La transition de 1 a 2 doit être comprise ainsi :
On veut lire un b, on cherche a dépiler X, si c'est bien le symbole au dessus de la pile, alors on le dépile et on rempile ε, c'est à dire qu'on ne rempile rien.

La boucle au dessus de 1 :
On veut lire un a, on se moque de ce qu'il ya dessus sur la pile (ε), mais on empile un symbole X.

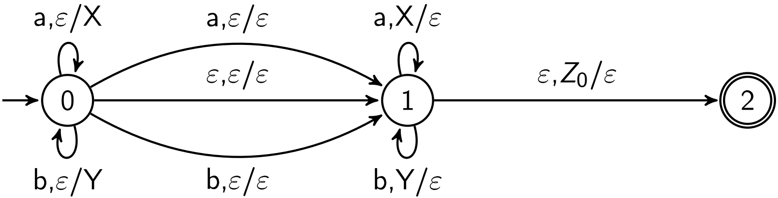
Si on lit le mot aaabbb, on parcourt la suite d'états $0, 1, 1, 1, 2, 2, 2, 3$ et la pile évolue de la façon suivante :

$$Z_0,\ Z_0X,\ Z_0XX,\ Z_0XXX,\ Z_0XX,\ Z_0X, Z_0,\ \textit{vide}$$

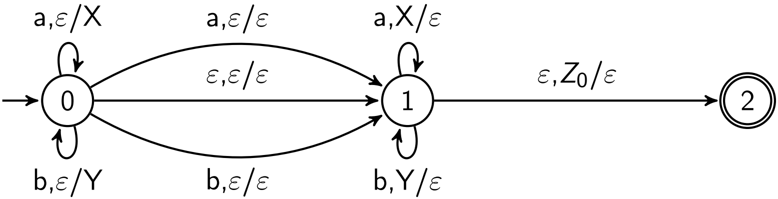
Automates à Pile - Exemples

Trouver des automates à pile pour les langages suivants

1. $L_0 = \{a^n b^m,\ n \geq m \geq 0\}$
2. Sur l'alphabet $(a_1, b_1, a_2, b_2, \dots, a_n, b_n)$, L_1 est le langage des mots bien parenthésés (a_i est la parenthèse gauche associée a la parenthèse droite b_i), appelé n-ième langage de Dyck.
3. $L_2 = \{a^n b^{2n+3},\ n \geq 0\}$.
4. $L_3 = \{u,\ |u|_a = |u|_b \text{ et } |u|_c \text{ pair}\}$ (L'alphabet est $\{a, b, c\}$).
5. $L_4 = \{a^p b^q c^r,\ p \neq q \text{ ou } q \neq r\}$



Quels sont les mots acceptés par cet automate à pile ?



Quels sont les mots acceptés par cet automate à pile ?

Les palindromes

Automates à Pile - Langage accepté

Il existe plusieurs définitions possibles pour un mot accepté :

- Si à la fin du calcul, on est dans un état acceptant
- Si à la fin du calcul, la pile est vide
- Si à la fin du calcul, on est dans un état acceptant ET la pile est vide (celle qu'on a choisie au début)

Automates à Pile - Langage accepté

Il existe plusieurs définitions possibles pour un mot accepté :

- Si à la fin du calcul, on est dans un état acceptant
- Si à la fin du calcul, la pile est vide
- Si à la fin du calcul, on est dans un état acceptant ET la pile est vide (celle qu'on a choisie au début)

Bien sur pour un automate fixé si on change de règle on change le langage accepté, mais toutes les définitions sont équivalentes en ce sens que :

Théoreme

Pour tout langage L les propositions sont équivalentes

- *Il existe un automate à pile qui reconnait L par état acceptant*
- *Il existe un automate à pile qui reconnait L par pile vide*
- *Il existe un automate à pile qui reconnait L par état acceptant ET par pile vide.*

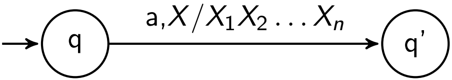
Le résultat important pour nous est le suivant :

Théoreme

Un langage L est algébrique si et seulement si il existe un automate à pile non déterministe qui le reconnait

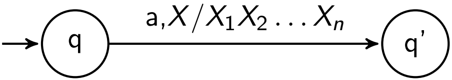
On ne montre qu'un sens : comment passer de la grammaire à l'automate.

Pour simplifier on s'autorise des transitions un peu plus générales : on s'autorise à empiler plusieurs symboles d'un coup :



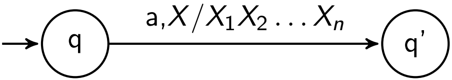
Automates à Pile - généralisation

Pour simplifier on s'autorise des transitions un peu plus générales : on s'autorise à empiler plusieurs symboles d'un coup :



Important : cela ne change rien aux langages qu'on peut décrire.

Pour simplifier on s'autorise des transitions un peu plus générales : on s'autorise à empiler plusieurs symboles d'un coup :



Important : cela ne change rien aux langages qu'on peut décrire.

En effet, pour se ramener à un automate à pile normal il suffit pour chaque transition de ce type de rajouter des nouveaux états $q_1, q_2, \ldots q_{n-1}$ avec des transitions données par

- $\delta(q, a, X) = \{(q_1, X_1)\}$
- $\delta(q_i, \varepsilon, \varepsilon) = \{(q_{i+1}, X_{i+1})\}$ pour tout $1 \leq i \leq n-2$
- $\delta(q_{n-1}, \varepsilon, \varepsilon) = \{(q', X_n)\}$

Automates à Pile et Langages Algébriques

Considérons un langage algébrique décrit par une grammaire $G = (\Sigma, V, S, P)$.

On va construire un automate à pile (généralisé) à un seul état q_0.

Considérons un langage algébrique décrit par une grammaire $G = (\Sigma, V, S, P)$.

On va construire un automate à pile (généralisé) à un seul état q_0.

Son alphabet d'entrée est Σ et son alphabet de pile est V.

Considérons un langage algébrique décrit par une grammaire $G = (\Sigma, V, S, P)$.

On va construire un automate à pile (généralisé) à un seul état q_0.

Son alphabet d'entrée est Σ et son alphabet de pile est V.

On commence par mettre la grammaire sous forme normale de Chomsky et ensuite :

- Pour chaque règle $X \to X_1X_2$, on met une transition $(\varepsilon, X/X_1X_2)$
- Pour chaque règle $X \to a$, on met une transition $(a, X/\varepsilon)$

Cela revient à faire la dérivation la plus à droite du mot

Automates à Pile Déterministe

Un automate à pile M comme défini précédemment est déterministe si dans une situation donnée, on ne peut pas avoir la choix sur la transition à suivre.

Automates à Pile Déterministe

Un automate à pile M comme défini précédemment est déterministe si dans une situation donnée, on ne peut pas avoir la choix sur la transition à suivre.
Cela veut dire :

1. Pour un état q donné, pour un symbole de pile X donné, si il existe une transition partant du triplet (q, ε, X), elle est unique et il n'existe pas de transitions partant de (q, a, X) quelle que soit la lettre a.

Un automate à pile M comme défini précédemment est déterministe si dans une situation donnée, on ne peut pas avoir la choix sur la transition à suivre.
Cela veut dire :

1. Pour un état q donné, pour un symbole de pile X donné, si il existe une transition partant du triplet (q, ε, X), elle est unique et il n'existe pas de transitions partant de (q, a, X) quelle que soit la lettre a.
2. Pour un état q donné, pour une lettre a donnée, pour un symbole de pile X donné, Il existe au plus une transition partant du triplet (q, a, X).

Automates à Pile Déterministe et Non Déterministe

Pour les automates finis, on a vu qu'on pouvait toujours étant donné un automate, en trouver un déterministe qui reconnaissait le même langage.
C'est malheursement faux pour les automates à pile.

Automates à Pile Déterministe et Non Déterministe

Pour les automates finis, on a vu qu'on pouvait toujours étant donné un automate, en trouver un déterministe qui reconnaissait le même langage.
C'est malheursement faux pour les automates à pile.

Il existe des langages L reconnus par des automates à pile non déterministes mais reconnus par aucun automate à pile déterministe.

Automates à Pile Déterministe et Non Déterministe

Pour les automates finis, on a vu qu'on pouvait toujours étant donné un automate, en trouver un déterministe qui reconnaissait le même langage.
C'est malheursement faux pour les automates à pile.

Il existe des langages L reconnus par des automates à pile non déterministes mais reconnus par aucun automate à pile déterministe.

Un exemple est le langage des palindromes.

L'idée est qu'on n'a pas moyen de savoir quand est-ce qu'on est au milieu du mot quand on le lit.