2.3 Automates à pile

Un automate à pile (*pushdown automaton*) possède les différents éléments d'un reconnaisseur (bande de lecture, tête de lecture, unité de contrôle), sa particularité est son mode de stockage : une **pile** dans laquelle sont stockés des symboles d'un alphabet particulier appelé alphabet de pile. Comme l'automate sans pile, la tête de lecture ne peut se déplacer que d'une case à chaque mouvement, de gauche à droite.

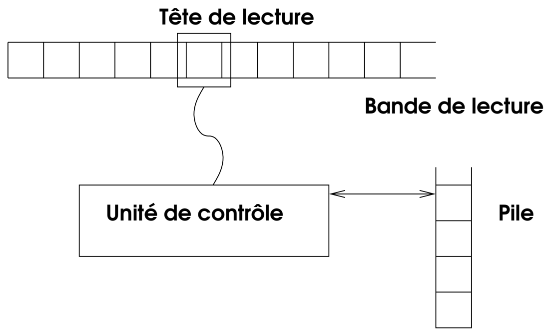

FIG. 2.1 – Eléments d'un automate à pile

L'automate peut stocker et accéder à la pile, mais, comme son nom l'indique, uniquement au sommet. Ce type d'automate est capable, par exemple, de reconnaître a^nb^n parce que la pile peut lui permettre de se souvenir du nombre de a déjà lus.

Le mouvement d'un tel automate va être déterminé par 3 paramètres (seulement 2 pour l'automate sans pile) : (1) le symbole sous la tête de lecture, (2) l'état courant, et (3) le symbole en sommet de pile. Par ailleurs, le « mouvement » d'un tel automate consistera en (1) un déplacement de la tête d'une case à droite (sauf en cas d'ε) (comme l'automate sans pile), et (2) le dépilement du caractère en sommet de pile et l'empilement d'un **mot** sur la pile (avec le cas particulier d'ε).

2.3.1 Définition

Formellement, on définit un automate à pile par un sextuplet $\langle Q, X, \Gamma, \delta, q_0, Z_0, F\rangle$:

- Q est un ensemble fini d'états
- X est l'alphabet d'entrée (souvent noté Σ)
- Γ est l'alphabet de pile (pas nécessairement disjoint de X)
- δ est l'application[6] de transition $\delta : Q \times (X \cup \{\varepsilon\}) \times \Gamma \longrightarrow Q \times \Gamma^*$
- $q_0 \in Q$ est l'état initial
- $Z_0 \in \Gamma$ est le symbole de fond de pile
- $F \subset Q$ est l'ensemble des états d'acceptation

Pour définir de façon précise le comportement d'un tel automate, il est pratique d'introduire la notion de **configuration**. Une configuration représente tous les aspects pertinents de la machine dans une situation donnée. On la définit comme un triplet $(q, m, \alpha) \in Q \times X \times \Gamma$, où :
- q est l'état courant de l'unité de contrôle

[6] i.e. pas nécessairement déterministe.

- m représente la partie du mot à reconnaître non encore lue. Le symbole le plus à gauche de m est le caractère sous la tête de lecture.
- α représente le contenu de la pile (si on la couche sur le côté gauche). Le symbole le plus à gauche d'α est le sommet de la pile.

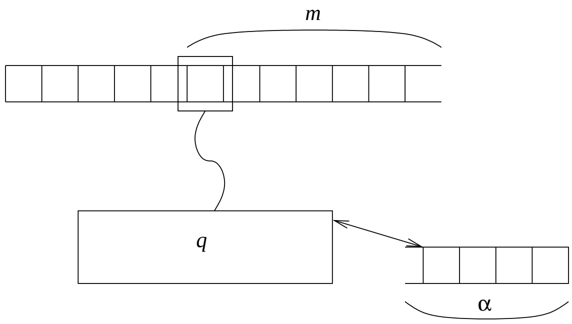

FIG. 2.2 – Représentation de la configuration (q, m, α)

Il est maintenant possible de spécifier le comportement de l'automate. On dira :

$$(q, aw, Z\alpha) \vdash (q', w, \gamma\alpha)$$
$$\text{si}$$
$$\delta(q, a, Z) \ni (q', \gamma)$$

Lors de ce mouvement :

- l'unité de contrôle passe de q à q'
- le symbole a a été lu
- la tête de lecture s'est déplacée d'une case vers la droite
- le symbole Z a été dépilé et le mot γ empilé.

Remarques :
- Si $\gamma = \varepsilon$, la pile a été dépilée (sauf si $Z = \varepsilon$)
- Si $a = \varepsilon$, le changement d'état et la modification de la pile se font sans mouvement de la tête.
- Si $Z = \varepsilon$, il s'agit d'une transition permise **quel que soit le symbole sur la pile**.
- Si $\gamma = \varepsilon$ et $Z = \varepsilon$, alors la pile est inchangée.
- Donc la transition $(q, \varepsilon, \varepsilon) \vdash (q', \varepsilon)$ est un changement d'état sans autre modification, l'équivalent pour l'automate à pile de l'ε-transition des automates finis.

2.3.2 Représentation graphique

Définition

On peut représenter les automates à pile au moyen d'un graphe, comme pour les automates finis. Mais à un arc seront associés, en plus du symbole de transition, les contraintes et opérations sur la pile. Une étiquette d'arc sera de la forme "$a, b \to \alpha$"[7], où a est le symbole lu, b le sommet de pile (si le sommet de pile n'est pas b, la transition ne peut être réalisée), et c est le mot empilé à la place de b. Avec les notations précédentes :

[7]Variante notationelle : $a\ \ b/\alpha$.

$$q_i \xrightarrow{a,b\to\alpha} q_j \equiv \delta(q_i, a, b) \supseteq \{(q_j, \alpha)\}$$

Le symbole ε peut être utilisé dans chaque cas, avec la signification donnée plus haut. Bien entendu, tout automate fini peut être vu comme un automate à pile, sans opérations ni contraintes liées à la pile. Voir par exemple la figure 2.3.

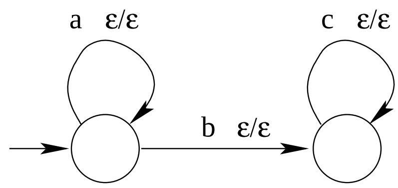

FIG. 2.3 – Un automate à pile pour a^*bc^*

Exemple 1

$A_1 = \langle\{q_0, q_1, q_2\}, \{a, b\}, \delta, q_0, Z, \{q_2\}\rangle$ avec : $\delta(q_0, a, Z)=\{(q_1, aZ)\}$
$\delta(q_1, a, a) =\{(q_1, aa)\}$
$\delta(q_1, b, a) =\{(q_2, \varepsilon)\}$
$\delta(q_2, b, a) =\{(q_2, \varepsilon)\}$

Cet automate (figure 2.4) est un automate déterministe reconnaissant le langage a^nb^n. Il commence par lire une suite de a en les empilant, puis dépile un a pour chaque b lu.

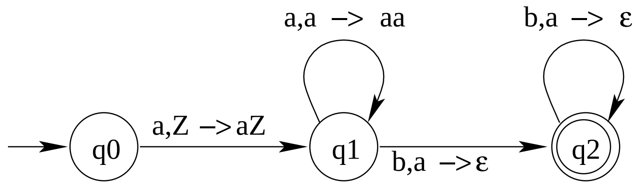

FIG. 2.4 – Un automate à pile pour a^nb^n

2.3.3 Reconnaissance

Une **configuration initiale** d'un automate à pile $A = \langle Q, X, \Gamma, \delta, q_0, Z_0, F\rangle$ est une configuration de la forme (q_0, m, Z_0) avec $m \in X^*$. L'automate est dans l'état initial, la pile ne contient que le "fond", et la tête se trouve sur le premier symbole du mot à reconnaître.

Une **configuration d'acceptation** est une configuration de la forme (q, ε, Z_0) avec $q \in F$. Le mot a été reconnu en entier, l'automate est dans un état terminal et la pile ne contient que le "fond".

Un mot $m \in X^*$ est **accepté** par A s'il existe une suite de configurations (q_i, m_i, α_i), telle que $(q_0, m_0, \alpha_0) = (q_0, m, Z_0)$; $(q_k, m_k, \alpha_k) = (q, \varepsilon, Z_0)$ avec $q \in F$; et pour tout $i \in [O, k[$, $(q_i, m_i\alpha_i) \vdash (q_{i+1}, m_{i+1}, \alpha_{i+1})$.

Le **langage reconnu** par A, noté $L(A)$ est l'ensemble des mots reconnus par A :

$$L(A) = \{m \in X^* \ / \ (q_0, m, Z_0) \overset{*}{\vdash} (q, \varepsilon, Z_0) \text{ avec } q \in F\}$$

Noter que dans un automate à pile non déterministe, il peut y avoir plusieurs suites de configurations vérifiant la définition précédente.

Variantes Il existe de nombreuses variantes d'automate à pile, dont on démontre plus ou moins facilement qu'elles sont équivalentes. Par exemple, on peut n'autoriser l'empilement que d'un symbole à la fois ; on peut décider de s'arrêter sur pile vide, ou simplement sur état d'acceptation ; on peut ne pas introduire de symbole de fond de pile, et définir simplement la notion de pile vide. Les démonstration d'équivalence reposent à chaque fois sur un algorithme de transformation de tout automate d'un type vers un automate "variante" qui reconnaît le même langage.

Exemple 2

$A_2 = \langle \{B, C, D, E, F\}, \{a, b, c\}, \{\$, a\}, \delta, B, \$, \{D, F\} \rangle$ avec (\$ est au fond de la pile) :

$$\begin{array}{lll}
\delta(B, a, \varepsilon) = \{(B, a)\} & \delta(C, b, a) = \{(C, \varepsilon)\} & \delta(E, b, \varepsilon) = \{(E, \varepsilon)\} \\
\delta(B, \varepsilon, \varepsilon) = \{(C, \varepsilon), (E, \varepsilon)\} & \delta(C, \varepsilon, \$) = \{(D, \$)\} & \delta(E, \varepsilon, \varepsilon) = \{(F, \varepsilon)\} \\
 & \delta(D, c, \$) = \{(D, \$)\} & \delta(F, c, a) = \{(F, \varepsilon)\}
\end{array}$$

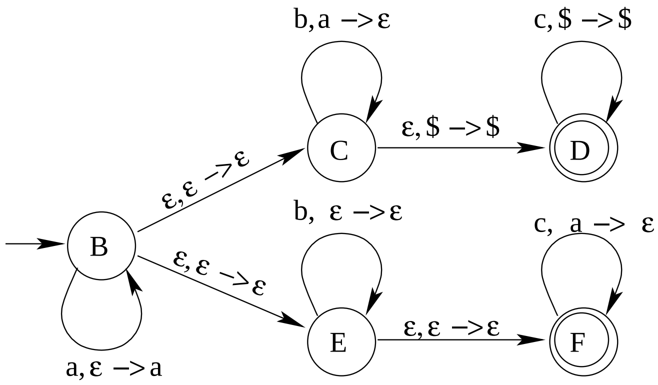

Fig. 2.5 – Automate à pile reconnaissant $a^i b^j c^k$ avec $i = j$ ou $i = k$

L'automate A_2 (figure 2.5) est un automate non déterministe, et reconnaît le langage $\{a^i b^j c^k$, avec $i, j, k \geq 0$ et $i = j$ ou $i = k\}$. Le principe est de lire un suite de a en les empilant, pour pouvoir comparer leur nombre au nombre de b ou de c. Cette comparaison est un peu délicate, et c'est là que le non-déterminisme entre en jeu : chaque branche de l'automate correspond à une alternative. La branche supérieure vérifie qu'il y a autant de b que de a (le nombre de c étant dès lors non contraint), celle du bas vérifie qu'il y a autant de c que de a, en sautant les b qui peuvent être en nombre quelconque.