Construction d'un automate à pile vide à partir d'une grammaire algébrique :

En entrée : La grammaire G<X, V, P, S> sans variables non productives et inaccessibles.

En sortie : l'automate à Pile vide A_ε<X, Y, S_A, S_0, F, II, #>.

Une grammaire ne donne pas la structure du mot mais les règles de production qui permettent de générer les mots. Ces mots sont obtenus en générant leur arbre de dérivation. Pour reconnaître un mot par cet automate, il faut générer son arbre de dérivation dans la pile.

Définition des paramètres de A_ε:

$Y = X \cup V \cup \{\#\}$ (puisque l'on va générer l'arbre de dérivation du mot à reconnaitre dans la pile on doit pouvoir empiler les éléments de X et V)

$S_A = \{S_0, S_1\}$

$F = \{S_1\}$

Procédure de construction de II à partir de P :

1. $II = \{ \# S_0 \rightarrow \# S S_1\}$ /* La première instruction de l'automate est l'empilement de l'axiome pour que l'on puisse générer l'arbre de dérivation dans la pile. On rajoute cette instruction quelque soit la grammaire que l'on transforme en automate. Vous remarquerez que cet automate ne fonctionne pas de la même manière que le premier que nous avons défini. Il n'y a pas toujours lecture de la lettre en entrée.

2. Pour toutes les productions de P ($A \rightarrow \alpha$) faire :

$II = II \cup \{ AS_1 \rightarrow \alpha^R S_1\}$ /* Ces instructions permettent de générer l'arbre de dérivation du mot à reconnaître dans la pile.

3. Pour tous les éléments x_i de X faire :

$II = II \cup \{ x_i S_1 x_i \rightarrow S_1\}$.

Exemple:

Soit G<X, V, P, S> une grammaire où P est défini comme suit :

P : { **$S \rightarrow AB$**
$A \rightarrow aAb / \varepsilon$
$B \rightarrow cBd / \varepsilon$ }

L'automate à Pile vide A_ε<X, Y, S_A, S_0, F, II, #> reconnaissant L(G) est le suivant :

$Y = X \cup V \cup \{\#\}$
$S_A = \{S_0, S_1\}$
$F = \{S_1\}$
$II = \{\# S_0 \rightarrow \# S S_1$
$SS_1 \rightarrow BA S_1$

$AS_1 \rightarrow bAa\ S_1$
$AS_1 \rightarrow S_1$
$BS_1 \rightarrow dBc\ S_1$
$BS_1 \rightarrow S_1$
$a\ S_1\ a \rightarrow S_1$
$b\ S_1\ b \rightarrow S_1$
$c\ S_1\ c \rightarrow S_1$
$d\ S_1\ d \rightarrow S_1$

}

Reconnaissance d'un mot :

1. W= aabbcd $\in L(A_\varepsilon)$?

$\#\ S_0$aabbcd $\vdash$ #SS_1aabbcd $\vdash$ # BAS_1 aabbcd $\vdash$ # $BbAaS_1$ aabbcd $\vdash$ # $BbAS_1$ aabbcd $\vdash$ # $BbbAaS_1$abbcd $\vdash$ # $BbbAS_1$bbcd $\vdash$ # $BbbS_1$bbcd $\vdash$ # BbS_1bcd $\vdash$ # BS_1cd $\vdash$ # dBc S_1cd $\vdash$ # dB S_1d $\vdash$ # dS_1d $\vdash$ # S_1

W= aabbcd fait passer l'automate d'une configuration initiale à une configuration finale alors w $\in L(A_\varepsilon)$.

2. W= aacd $\in L(A_\varepsilon)$

S_0aacd $\vdash$ #SS_1aacd $\vdash$ # BAS_1 aacd $\vdash$ # $BbAaS_1$ aacd $\vdash$ # $BbAS_1$acd $\vdash$ # $BbbAaS_1$acd $\vdash$ # $BbbAS_1$cd $\vdash$ # $BbbS_1$cd A ce niveau, l'automate se bloque car il n'existe, dans II, aucune instruction dont le membre gauche est bS_1c.

W= aacd ne fait pas passer l'automate d'une configuration initiale à une configuration finale alors w $\notin L(A_\varepsilon)$