

Tables d'un analyseur LR(0)

Un analyseur LR(0) est défini par 2 tables :

- ▶ la **table des successeurs** ;
- ▶ la **table des actions**.

Table des actions LR(0)

Indique quelle action effectuer :

- ▶ dans un état $q \in Q$;
- ▶ si $x \in V_T \cup \{\#\}$ est sous la tête de lecture.

$Q \times (V_T \cup \{\#\}) \rightarrow$ ensemble d'actions

Une action peut être :

- ▶ la lecture du terminal x (**decale**) ;
- ▶ la réduction par une production p (**red** par p) ;
- ▶ l'acceptation (**acc**).

Exemple, table des actions

	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}	E_{11}	E_{12}
a	d	e	d	red $A \rightarrow b$	e	red $A \rightarrow aBb$	red $B \rightarrow aB$	e	red $S \rightarrow AD$	red $D \rightarrow e$	red $S \rightarrow B$	red $B \rightarrow c$
b	d	e	d	red $A \rightarrow b$	d	red $A \rightarrow aBb$	red $B \rightarrow aB$	e	red $S \rightarrow AD$	red $D \rightarrow e$	red $S \rightarrow B$	red $B \rightarrow c$
c	d	e	d	red $A \rightarrow b$	e	red $A \rightarrow aBb$	red $B \rightarrow aB$	e	red $S \rightarrow AD$	red $D \rightarrow e$	red $S \rightarrow B$	red $B \rightarrow c$
e	e	e	e	red $A \rightarrow b$	e	red $A \rightarrow aBb$	red $B \rightarrow aB$	d	red $S \rightarrow AD$	red $D \rightarrow e$	red $S \rightarrow B$	red $B \rightarrow c$
#	e	a	e	red $A \rightarrow b$	e	red $A \rightarrow aBb$	red $B \rightarrow aB$	e	red $S \rightarrow AD$	red $D \rightarrow e$	red $S \rightarrow B$	red $B \rightarrow c$

a : acceptation, d : décale, e : erreur, red : réduction par p

Table des actions, remplissage

Pour tout $a \in V_T$ et $q \in Q$:

si q contient un item de la forme $[X \rightarrow \dots \bullet a \dots]$
alors mettre **decale** dans la case (q, a)

Pour tout $q \in Q$, $Q \neq q_f$:

- ▶ si q contient un item terminal de la forme $[X \rightarrow \alpha \bullet]$;
- ▶ alors, pour tout $a \in V_T \cup \{\#\}$, mettre **réduction** $X \rightarrow \alpha$ dans la case (q, a) .

Mettre **acceptation** dans la case $(q_f, \#)$.

Mettre **erreur** dans les cases encore vides.

Table des actions, remarque

Pour un automate LR(0), cas dégénéré pour le remplissage de la table par une réduction.

$k=0$: aucun symbole de prédiction (pas de *Premier*, *Suivant*).

Une réduction est effectuée quelque soit la tête de lecture.

⇒ colonnes remplies de la même réduction.

Le cas général est : pour tout $a \in V_T \cup \{\#\}$ et $q \in Q$:

- ▶ si q contient un item terminal de la forme $X \rightarrow \alpha \bullet$ et que la réduction peut se faire **avec a sous la tête de lecture** ;
- ▶ alors, mettre **réduction $X \rightarrow \alpha$** dans la case (q, a) .

Caractérisation d'une grammaire LR(0)

Une grammaire est LR(0) si sa table des actions contient pour chaque case :

- ▶ une seule action
- ▶ ou erreur.

Exemple, table des actions et conflits

E

$[A \rightarrow b\bullet]$
$[B \rightarrow b\bullet]$

	E	...
c	red $A \rightarrow b$ red $B \rightarrow b$	
b	red $A \rightarrow b$ red $B \rightarrow b$	
...		

E

$[A \rightarrow \bullet b]$
$[B \rightarrow c\bullet]$

	E	...
c	red $B \rightarrow c$	
b	red $B \rightarrow c$ decale	
...		

Quand une grammaire n'est pas LR(0)

C'est peut-être à cause du 0.

On peut essayer une analyse LR(1) : beaucoup plus puissante.

C'est plus facile d'expliquer d'abord les grammaires SLR(1) :
Simple LR(1).

Exemple

Soit la grammaire $S' \rightarrow S, S \rightarrow a \mid \epsilon$.

Conflit shift/reduce dans l'état initial (lire a , réduire par $S \rightarrow \epsilon$) :

$S' \rightarrow \bullet S$
$S \rightarrow \bullet a$
$S \rightarrow \bullet$

Mais si la tête de lecture est :

- ▶ dans $\{a\}$, alors lire a ;
- ▶ dans $\{\#\} = \text{Suivant}(S)$ alors réduire par $S \rightarrow \epsilon$.

Un automate SLR(1) exploite cette information.