

Construction de la table des actions LR(1)

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 $q \in Q$, $q \neq q_f$ et tout $a \in V_T \cup \{\#\}$:

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

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

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

Caractérisation d'une grammaire LR(1)

Une grammaire est LR(1) si sa table des actions contient pour chaque case :

- ▶ une seule action
- ▶ ou erreur.

Au delà des grammaires LR(1)

Beaucoup de grammaires sont LR(1).

Mais les tables sont rapidement trop grosses pour tenir en mémoire.

L'analyse utilisée en pratique est l'analyse LALR(1) (Look-Ahead LR(1)), avec :

$$\text{LR}(0) \subseteq \text{SLR}(1) \subseteq \text{LALR}(1) \subseteq \text{LR}(1)$$

L'analyse LALR(1) est un bon compromis entre puissance et encombrement mémoire.

CUP est un analyseur LALR(1).