

Principe : $k=1$ et exploitation des *Suivant*

Un analyseur SLR(1) prend en compte le symbole sous la tête de lecture ($k=1$, cf LL(1)) pour **décider d'une réduction** :

Réduction par $X \rightarrow \alpha$ seulement si **tête lecture** \in *Suivant*(X)

Repose comme l'analyse LR(0) sur l'automate LR-AFD.

Permet d'arbitrer certains conflits LR(0) S/R et R/R.

Conflicts shift/reduce au sens SLR(1)

Un état de LR-AFD provoque un **conflit S/R** au sens **SLR(1)** s'il contient à la fois :

- ▶ un item de la forme $[Y \rightarrow \dots \bullet a \dots]$
- ▶ un item de la forme $[X \rightarrow \alpha \bullet]$ avec $a \in \text{Suivant}(X)$

Comparer avec LR(0) : conflit S/R au sens LR(0) si l'état contient les items $[Y \rightarrow \dots \bullet a \dots]$ et $[X \rightarrow \alpha \bullet]$

Conflits reduce/reduce au sens SLR(1)

Un état de LR-AFD provoque un **conflit R/R** au sens **SLR(1)** s'il contient à la fois :

- ▶ un item de la forme $[Y \rightarrow \beta \bullet]$
- ▶ un item de la forme $[X \rightarrow \alpha \bullet]$
- ▶ avec $Suivant(X) \cap Suivant(Y) \neq \emptyset$

Comparer avec LR(0) : conflit R/R au sens LR(0) si l'état contient les items $[Y \rightarrow \beta \bullet]$ et $[X \rightarrow \alpha \bullet]$

Grammaire SLR(1)

Une grammaire est dite **SLR(1)** si l'automate LR-AFD ne contient pas de conflits au sens SLR(1).

Construction de la table des actions SLR(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 $X \rightarrow \alpha \bullet$;
- ▶ alors, si $a \in \text{Suivant}(X)$, mettre **réduction** $X \rightarrow \alpha$ dans la case (q, a) .

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

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

Exemple : table des actions SLR(1)

$Suivant(S) = \{\#\}$ $Suivant(L) = \{\#\}$

$Suivant(E) = \{",", \#\}$

	$E0$	$E1$	$E2$	$E3$	$E4$	$E5$
x	decale	erreur	erreur	erreur	decale	erreur
$,$	red $E \rightarrow \epsilon$	erreur	decale	red $E \rightarrow x$	red $E \rightarrow \epsilon$	erreur
$\#$	red $E \rightarrow \epsilon$	accepte	red $L \rightarrow E$	red $E \rightarrow x$	red $E \rightarrow \epsilon$	red $L \rightarrow E, L$

Caractérisation d'une grammaire SLR(1)

La grammaire est **SLR(1)** si sa table des actions contient pour chaque case :

- ▶ une seule action
- ▶ ou erreur.

Remarques

Une grammaire LR(0) ou SLR(1) n'est pas ambiguë.

Une grammaire ambiguë n'est ni LR(0) ni SLR(1).

Comparaison SLR(1) - LR(0)

Méthode SLR(1) basée comme LR(0) sur l'automate LR-AFD :

- ▶ les tables des successeurs LR(0) et SLR(1) sont identiques ;
- ▶ les tables LR(0) et SLR(1) ont le même encombrement mémoire.

Comparaison SLR(1) - LR(0)

Grâce au $k = 1$:

- ▶ l'analyse SLR(1) est strictement plus puissante que l'analyse LR(0) ;
- ▶ = elle engendre moins de conflits.

$$\text{LR}(0) \subset \text{SLR}(1)$$

Néanmoins beaucoup de grammaires (non ambiguës) ne sont pas SLR(1).

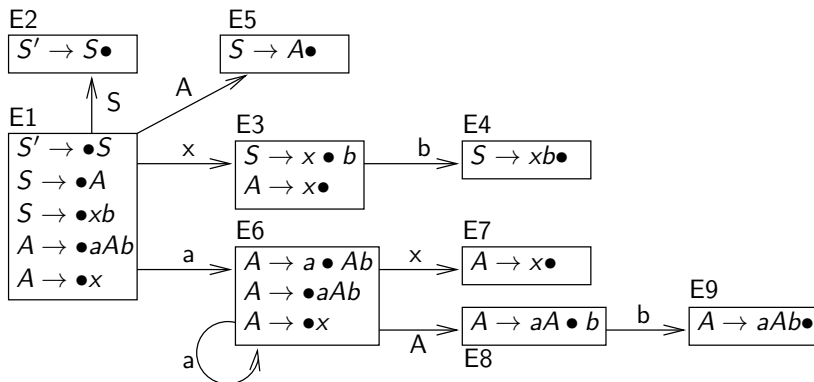
Exemple 1 : G_1

$$\begin{aligned} S &\rightarrow A \mid xb \\ A &\rightarrow aAb \mid x \end{aligned}$$

G_1 grammaire non ambiguë (mais non LL(1)) :

- ▶ si $xb : S \Rightarrow xb$;
- ▶ si $a^nxb^n : S \Rightarrow A \Rightarrow^n a^nAb^n \Rightarrow a^nxb^n$.

Automate LR-AFD de G_1



Conflit pour G_1

L'automate LR-AFD contient un conflit S/R au sens LR(0) dans l'état E3 :

$$\begin{aligned} [S \rightarrow x \bullet b] \\ [A \rightarrow x \bullet] \end{aligned}$$

Pour savoir si c'est un conflit au sens SLR(1), calcul des *Suivant* :

- ▶ $Suivant(S') = Suivant(S) = \{\#\}$;
- ▶ $Suivant(A) = Suivant(S) \cup \{b\} = \{\#, b\}$;

$b \in Suivant(A)$ donc E3 contient un **conflit S/R au sens SLR(1)**.

Conflit SLR(1) pour G_1 : origine

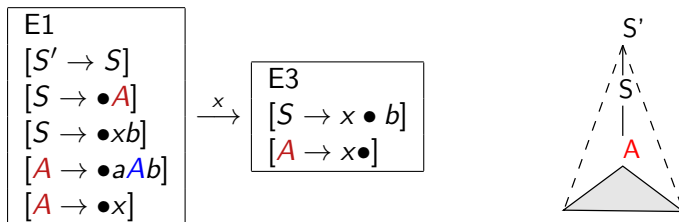
Conflit dans E3 car $b \in \text{Suivant}(A)$. Et pourtant...

... la lecture de b impose la dérivation $S' \Rightarrow S \Rightarrow xb$.

... mais *Suivant* trop imprécis pour le voir.

Comment être plus précis ?

Conflit SLR(1) pour G_1 : solution



Les A de $E1$ et $E3$ ne peuvent être suivis que d'un $\#$, pas d'un b .
 Ce A (suivi par b) n'est pas expansé dans $E1$ et $E3$, mais dans $E6$.
 Si on considère les symboles de $V_T \cup \{\#\}$ qui peuvent suivre A
 dans $E3$, on fait sauter le conflit.

Restriction des symboles de look-ahead

L'analyse LR(1) ne considère pas tous l'ensemble $Suivant(X)$ pour réduire par $X \rightarrow \dots$.

Elle calcule :

- ▶ pour chaque item $[X \rightarrow \alpha]$ d'un état E ;
- ▶ un ensemble $L \subseteq Suivant(X)$;
- ▶ contenant les symboles qui peuvent suivre X dans E .

L peut parfois être égal à $Suivant(X)$.

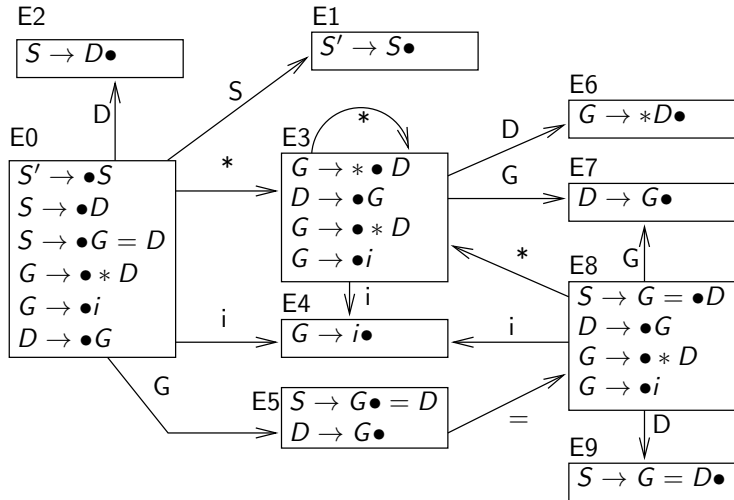
Exemple 2 : G_2

$$\begin{aligned} S &\rightarrow G = D \mid D \\ G &\rightarrow *D \mid i \\ D &\rightarrow G \end{aligned}$$

Grammaire G_2 non ambiguë :

- ▶ la présence ou l'absence du $=$ indique s'il faut choisir $S \rightarrow G = D$ ou $S \rightarrow D$;
- ▶ la grammaire de productions $\{ G \rightarrow *D \mid i, D \rightarrow G \}$ est LL(1).

Automate LR-AFD pour G_2



Conflit pour G_2

L'automate LR-AFD contient un conflit S/R au sens LR(0) dans l'état 5 :

$$\begin{array}{l} E5 \\ [S \rightarrow G\bullet = D] \\ [D \rightarrow G\bullet] \end{array}$$

Pour savoir si c'est un conflit au sens SLR(1), calcul des *Suivant* :

- ▶ $Suivant(S') = Suivant(S) = \{\#\}$;
- ▶ $Suivant(G) = \{=\} \cup Suivant(D)$;
- ▶ $Suivant(D) = Suivant(S) \cup Suivant(G)$;

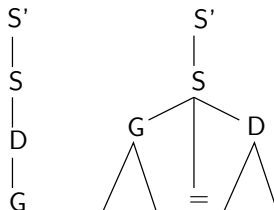
D'où $Suivant(G) = Suivant(D) = \{\#, =\}$.

$\Rightarrow \in Suivant(D)$ donc E_5 contient un **conflit S/R au sens SLR(1)**.

Conflit SLR(1) pour G_2 : origine

Conflit car " $=$ " $\in \text{Suivant}(D)$.

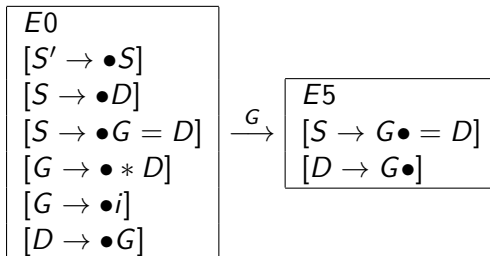
Pourtant il n'existe pas de dérivation t.q. $S \Rightarrow^* w_1 D = w_2$



$\text{Suivant}(D)$ contient ici un " $=$ " jamais rencontré comme look-ahead dans une analyse effective.

Restriction des symboles de look-ahead

Si on particularise les symboles de look-ahead aux états $E0$ et $E5$:



En $E0$ et $E5$, D ne peut être suivi que par $\#$: levée du conflit.