4
Analyse syntaxique.

1 – Généralités.

Lorsque l'on s'intéresse à un langage algébrique particulier L, on est conduit à choisir :

- une grammaire $G = (\mathcal{V}, \mathcal{A}, R)$,
- $S \in \mathcal{V}$ une variable, appelée *axiome*,

telles que $\mathcal{L}(G, S) = L$.

On suppose que G est *réduite* par rapport à S, c'est–à–dire que toute $X \in \mathcal{V}$ est :

- *productive* : $\mathcal{L}(G, X) \neq \emptyset$.
- *accessible à partir de* S : il existe α et $\beta \in (\mathcal{A} + \mathcal{V})^*$ tels que $S \overset{*}{\underset{G}{\Longrightarrow}} \alpha X \beta$.

Ceci signifie simplement que G ne comporte pas de variable inutile au calcul des éléments de L.

Une analyse syntaxique par G d'une "phrase" $u \in \mathcal{A}^*$ est un algorithme qui doit décider si $u \in \mathcal{L}(G, S)$:

- si tel est le cas, il doit décrire une dérivation $S \overset{k}{\underset{G}{\Longrightarrow}} u$,
- sinon, cet échec, considéré comme le résultat d'une *erreur syntaxique*, doit donner lieu à un diagnostic.

Les algorithmes d'analyse syntaxique qui nous intéressent sont de la classe LR :

- Ils lisent le mot à analyser de la gauche vers la droite (Left to right scanning).
- Ils choisissent, parmi les dérivations équivalentes possibles d'un même mot, la dérivation *à droite* (Rightmost derivation). La construction de cette dérivation s'effectue du mot à analyser vers l'axiome : on parle d'analyse "ascendante".
- Ils sont basés sur l'utilisation d'"automates à pile" : le résultat d'une analyse réussie est une suite de règles qui définit une dérivation $S \overset{k}{\underset{G}{\Longrightarrow}} u$.
- Ils sont déterministes.

Les grammaires “ambiguës” sont *a priori* hors du domaine d’application de ces algorithmes.

Soient $X \in \mathcal{V}$ et $\alpha \in (\mathcal{A} + \mathcal{V})^*$: lorsqu’il existe une dérivation $X \overset{k}{\underset{G}{\Longrightarrow}} \alpha$ dans G, il en existe généralement beaucoup d’autres ! Cependant, on sait que deux dérivations équivalentes font le même calcul, c’est–à–dire, ont le même arbre.

On est conduit à dire qu’*une grammaire n’est pas ambiguë ssi, pour toute $X \in \mathcal{V}$ et tout $\alpha \in (\mathcal{A}+\mathcal{V})^*$, deux dérivations $d : X \overset{k}{\underset{G}{\Longrightarrow}} \alpha$ et $d' : X \overset{k'}{\underset{G}{\Longrightarrow}} \alpha$ sont équivalentes.*

Dans la pratique, l’ambiguïté d’une grammaire provient de conventions d’écriture bien répertoriées, par exemple :

- **Préséance** : dans une grammaire dont certaines constantes représentent des opérateurs, par exemple $\times$ pour un produit et $\oplus$ pour une somme, une expression de la forme $\alpha \oplus \beta \times \gamma$ devra généralement être analysée comme $\alpha \oplus (\beta \times \gamma)$ et non pas comme $(\alpha \oplus \beta) \times \gamma$: on dira alors que $\times$ *a un degré de préséance supérieur à celui de* $\oplus$.
- **Mode d’association** : de même, une expression de la forme $\alpha \oplus \beta \oplus \gamma$ devra généralement être analysée comme $(\alpha \oplus \beta) \oplus \gamma$ et non pas comme $\alpha \oplus (\beta \oplus \gamma)$: on dira alors que $\oplus$ *est associative à (partir de la) gauche.*

Exemple 3.1 (suite).

Notre grammaire habituelle (Exemple 1 du chapitre 3) est ambiguë.

En effet, les dérivations

$$\underline{S} \overset{1}{\Longrightarrow} \underline{S}bS \overset{1}{\Longrightarrow} SbSbS$$
$$\underline{S} \overset{1}{\Longrightarrow} Sb\underline{S} \overset{1}{\Longrightarrow} SbSbS$$

ne sont pas équivalentes.

Plus généralement, considérons une grammaire G disposant de règles $X \longrightarrow X\alpha$ et $X \longrightarrow \alpha' X$ (on parle d'"appels récursifs" à gauche et à droite) alors G est ambiguë, à cause des deux dérivations non équivalentes :

$$\underline{X} \overset{1}{\Longrightarrow} \underline{X}\alpha \overset{1}{\Longrightarrow} \alpha' X\alpha$$
$$\underline{X} \overset{1}{\Longrightarrow} \alpha'\underline{X} \overset{1}{\Longrightarrow} \alpha' X\alpha.$$

Cette cause d'ambiguïté n'est pas très grave car on peut effectivement éliminer tout appel récursif à gauche sans pour cela changer les langages engendrés.

Il y a des cas plus sérieux : il existe des langages algébriques qui ne peuvent pas être engendrés par une grammaire non ambiguë !

2 – Dérivations à droite et analyse ascendante.

Lorsque l'on séquentialise une arborescence, par une méthode descendante, on peut choisir systématiquement d'élaguer la racine qui est le plus à droite.

Dérivations à droite

Une dérivation élémentaire est dite *à droite* lorsqu'elle se présente sous la forme

$$\pi X v \overset{1}{\underset{G}{\Longrightarrow}} \pi\alpha v$$

pour $v \in \mathcal{A}^*$.

Une dérivation est dite *à droite* lorsqu'elle est définie comme un enchaînement de dérivations élémentaires à droite.

Toute dérivation est équivalente à une dérivation à droite.

Une dérivation à droite sera symbolisée par $\beta \overset{k}{\underset{d,G}{\Longrightarrow}} \gamma$ ou plus simplement par $\beta \overset{k}{\underset{d}{\Longrightarrow}} \gamma$.

Exemple 3.1 (suite).

Voici une dérivation à droite $S \overset{11}{\underset{d}{\Longrightarrow}} aabbabaabba$ dans notre grammaire habituelle :

$$\begin{aligned} S &\overset{1}{\underset{d}{\Longrightarrow}} SbS \\ &\overset{1}{\underset{d}{\Longrightarrow}} SbSbS \\ &\overset{1}{\underset{d}{\Longrightarrow}} SbSba \\ &\overset{1}{\underset{d}{\Longrightarrow}} Sbba \\ &\overset{1}{\underset{d}{\Longrightarrow}} SbSbba \\ &\overset{1}{\underset{d}{\Longrightarrow}} SbSbSbba \\ &\overset{1}{\underset{d}{\Longrightarrow}} SbSbaabba \\ &\overset{1}{\underset{d}{\Longrightarrow}} Sbabaabba \\ &\overset{1}{\underset{d}{\Longrightarrow}} SbSbabaabba \\ &\overset{1}{\underset{d}{\Longrightarrow}} Sbbabaabba \\ &\overset{1}{\underset{d}{\Longrightarrow}} aabbabaabba \end{aligned}$$

dont l'arbre a été représenté au chapitre 3.

En énumérant les règles de la façon suivante :

$$\begin{array}{ll} 1 : S \longrightarrow SbS & \quad 3 : S \longrightarrow a \\ 2 : S \longrightarrow \varepsilon & \quad 4 : S \longrightarrow aa \end{array}$$

on obtient la suite $1, 1, 3, 2, 1, 1, 4, 3, 1, 2, 4$ avec laquelle on peut effectivement construire une dérivation à droite.

2.1 – Analyse ascendante.

Nous utiliserons les algorithmes LR.

Ce type d'analyse est le fruit d'une séquentialisation ascendante (on parle d'analyse ascendante) : on va du mot à analyser vers l'axiome.

Pour rendre ces méthodes efficaces, on se permet d'observer, dans la mesure du possible, les k symboles qui sont au début de la partie du mot restant à analyser ; on obtient alors les algorithmes de type $LR(k)$ qui sont "d'autant plus déterministes" que k est grand.

Nous nous limiterons à $LR(0)$et $LR(1)$.

2.2 – Configuration d'analyse partielle et préfixes viables.

L'analyse ascendante de $u \in \mathcal{A}^*$ est la construction d'une dérivation $S \Longrightarrow u$, qui détermine ses règles dans l'ordre inverse de leur enchaînement naturel : on remonte du mot u vers l'axiome S.

On peut appeler "analyse partielle de u" une dérivation $\pi \underset{d}{\Longrightarrow} w$ telle qu'il existe une dérivation $S \underset{d}{\Longrightarrow} \pi v$ pour laquelle $wv = u$.

La première question qui se pose est donc de trouver une condition utilisable, que doit nécessairement satisfaire un "préfixe" $\pi \in (\mathcal{A} + \mathcal{V})^*$ pour qu'il existe une dérivation à droite $S \underset{d}{\Longrightarrow} \pi v$.

Forme générale des dérivations à droite.

Une dérivation à droite $X \underset{d}{\Longrightarrow} \gamma$ de longueur > 0 se décompose de la façon suivante : $X \overset{1}{\Longrightarrow} \alpha'\beta \underset{d}{\Longrightarrow} \alpha' v$ où $v \in \mathcal{A}^*$ et $\alpha' v = \gamma$.
Si $\beta = Y\alpha''$ pour $Y \in \mathcal{V}$, on peut préciser l'analyse précédente en décomposant de la façon suivante :

$$X \overset{1}{\Longrightarrow} \alpha' Y \alpha'' \underset{d}{\Longrightarrow} \alpha' Y v'' \underset{d}{\Longrightarrow} \alpha' \gamma v''$$

Nous dirons que cette dérivation est *la composée* de

$$X \overset{1}{\Longrightarrow} \alpha' Y \alpha'' \underset{d}{\Longrightarrow} \alpha' Y v'' \quad \text{et de} \quad Y \underset{d}{\Longrightarrow} \gamma$$

Un raisonnement par induction sur la longueur montre alors que toute dérivation à droite, non triviale, est une composée de dérivations des types précédents :

$$\begin{array}{llllll}
X_0 & \overset{1}{\Longrightarrow} & \alpha'_1 X_1 \alpha''_1 & \underset{d}{\Longrightarrow} & \alpha'_1 X_1 v_1 & \text{pour } v_1 \in \mathcal{A}^* \\
X_1 & \overset{1}{\Longrightarrow} & \alpha'_2 X_2 \alpha''_2 & \underset{d}{\Longrightarrow} & \alpha'_2 X_2 v_2 & \text{pour } v_2 \in \mathcal{A}^* \\
\ldots & \ldots & \ldots & \ldots & \ldots & \\
X_{n-1} & \overset{1}{\Longrightarrow} & \alpha'_n X_n \alpha''_n & \underset{d}{\Longrightarrow} & \alpha'_n X_n v_n & \text{pour } v_n \in \mathcal{A}^* \\
X_n & \overset{1}{\Longrightarrow} & \alpha'_{n+1} \alpha''_{n+1} & \underset{d}{\Longrightarrow} & \alpha'_{n+1} v_{n+1} & \text{pour } v_{n+1} \in \mathcal{A}^*
\end{array}$$

ce qui, dans le cas où $n = 0$ se présente sous la forme particulière suivante :

$$\begin{array}{llllll}
X_0 & \overset{1}{\Longrightarrow} & \alpha'_1 \alpha''_1 & \underset{d}{\Longrightarrow} & \alpha'_1 v_1 & \text{pour } v_1 \in \mathcal{A}^*
\end{array}$$

Commentaires.

- La composée de ces dérivations a la forme

$$X_0 \underset{d}{\Longrightarrow} \pi X_n v \overset{1}{\Longrightarrow} \pi \alpha'_{n+1} \alpha''_{n+1} v \underset{d}{\Longrightarrow} \pi \alpha'_{n+1} v_{n+1} v$$

où $\pi = \alpha'_1 \ldots \alpha'_n$ et $v = v_n \ldots v_1$.

- Les mots $\pi\alpha'_{n+1}$ obtenus ainsi sont les “préfixes” que nous cherchons à caractériser.

- La dernière dérivation de la liste ci–dessus suppose seulement que α''_{n+1} est un suffixe du membre de gauche d'une règle pour X_n : une dérivation à droite admet donc généralement plusieurs décompositions.

- Un cas particulier intéressant est celui où $\alpha''_{n+1} = \varepsilon$: on a alors $v_{n+1} = \varepsilon$ et $X_n \overset{1}{\Longrightarrow} \alpha'_{n+1}$ est la dernière dérivation élémentaire, ce qui veut dire que $X_n \longrightarrow \alpha'_{n+1}$ est la dernière règle de la liste définissant la dérivation qui nous intéresse.

- La dérivation triviale $S \overset{0}{\Longrightarrow} S$, qui joue un rôle dans notre algorithme, ne rentre pas dans ce cadre!

Augmentation d'une grammaire.

La désignation effective de l'axiome S, se fait en "augmentant" G : on introduit une nouvelle variable $S' \notin \mathcal{V}$ et une seule nouvelle règle $S' \longrightarrow S$. Ainsi, S' *n'est–elle présente dans aucune autre règle que la sienne* et est l'axiome de la grammaire ainsi construite.

Nous préférons utiliser l'augmentation

$$\emptyset \longrightarrow S$$

produisant S à partir de l'ensemble vide (qui n'est pas un élément de $\mathcal{V}$).

Toute dérivation $S \underset{d}{\overset{k}{\Longrightarrow}} \gamma$ devient, une dérivation $\emptyset \underset{d}{\overset{k+1}{\Longrightarrow}} \gamma$ de longueur > 0 qui commence par $\emptyset \overset{1}{\Longrightarrow} S$.

Préfixes viables

$\pi\alpha' \in (\mathcal{A}+\mathcal{V})^*$ est appelé un *préfixe viable* ssi il existe une dérivation à droite

$$\emptyset \underset{d}{\Longrightarrow} \pi X v \overset{1}{\Longrightarrow} \pi\alpha'\alpha'' v$$

(v est alors nécessairement un élément de $\mathcal{A}^*$).
La règle $X \longrightarrow \alpha'\alpha''$ définit alors un *item valide* pour ce préfixe viable, que l'on notera $\cdot X \longrightarrow \alpha' \cdot \alpha''$, où $\cdot$ est un nouveau symbole (cf. ci–dessous).

Ce que nous avons observé au sujet des dérivations à droite peut s'exprimer en disant que toute dérivation à droite détermine un enchaînement d'item, c'est–à–dire :
ou bien :

$$\begin{array}{rcl}
\emptyset & \longrightarrow & \cdot S \\
\cdot S & \longrightarrow & \alpha'_1 \cdot X_1 \alpha''_1 \\
\cdot X_1 & \longrightarrow & \alpha'_2 \cdot X_2 \alpha''_2 \\
\ldots & & \ldots \\
\cdot X_{n-1} & \longrightarrow & \alpha'_n \cdot X_n \alpha''_n \\
\cdot X_n & \longrightarrow & \alpha'_{n+1} \cdot \alpha''_{n+1}
\end{array}$$

ce qui, dans le cas où $n = 0$, se présente de la façon suivante :

$$\begin{array}{rcl}
\emptyset & \longrightarrow & \cdot S \\
\cdot S & \longrightarrow & \alpha'_1 \cdot \alpha''_1
\end{array}$$

ou bien, le cas spécial

$$\emptyset \longrightarrow S \cdot$$

qui provient de l'augmentation de la grammaire.

Réciproquement, tout enchaînement d'item comme ci–dessus permet de construire des dérivations à droite assurant la viabilité d'un préfixe donné.

Cette remarque est importante pour la compréhension et la justification de la suite. Par exemple, on observe que :

Tout préfixe d'un préfixe viable est viable.

en "raccourcissant" l'enchaînement qui définit le préfixe viable en question.

2.3 – Item $LR(0)$ d'une grammaire G.

Pour définir les item $LR(0)$, on introduit un pointeur, c'est–à–dire un nouveau symbole $\cdot \notin \mathcal{A}+\mathcal{V}$.

Les item $LR(0)$

Pour toute $X \in \mathcal{V}+\{\emptyset\}$, et tout couple $\alpha \in (\mathcal{A}+\mathcal{V})^*$ et $\beta \in (\mathcal{A}+\mathcal{V})^*$:

$\cdot X \longrightarrow \alpha \cdot \beta$ est un *item* $LR(0)$ de G
ssi
$X \longrightarrow \alpha\beta$ est une règle de G.

Nous dirons simplement "item" pour "item $LR(0)$".

Pour définir les item de façon plus active, considérons les mots sur l'alphabet $\cdot + \mathcal{A}+\mathcal{V}$ comportant une et une seule occurrence de $\cdot$ (un tel mot peut s'écrire $\alpha \cdot \beta$ pour $\alpha \in (\mathcal{A}+\mathcal{V})^*$ et $\beta \in (\mathcal{A}+\mathcal{V})^*$). On peut appliquer à ces mots l'opération de "décalage" qui est symbolisée par la règle (sensible au contexte)

$$\cdot\,\xi \longrightarrow \xi\,\cdot$$

Alors, pour chaque règle $X \longrightarrow \alpha$ de G :

- $\cdot X \longrightarrow \cdot\,\alpha$ est un item (qui est l'application de $X \longrightarrow \alpha$ au mot $\cdot X$) ;
- si $\cdot X \longrightarrow \beta \cdot \xi\gamma$ est un item où $\xi \in \mathcal{A}+\mathcal{V}$, alors $\cdot X \longrightarrow \beta\xi \cdot \gamma$ est un item (qui est obtenu en appliquant un décalage après $\cdot X \longrightarrow \beta \cdot \xi\gamma$).

Exemples.

Compte tenu du fait que $\cdot\emptyset = \emptyset$, la règle $\emptyset \longrightarrow S$ définit deux item :

- *l'item initial* : $\emptyset \longrightarrow \cdot S$
- *l'item final* : $\emptyset \longrightarrow S\cdot$

Une règle $X \longrightarrow \varepsilon$ définit le seul item $\cdot X \longrightarrow \cdot$ dont le second membre est une occurrence du pointeur.

La règle $X \longrightarrow aXaYb$ définit les item suivants :

$$\begin{array}{ll}
\cdot X \longrightarrow \cdot aXaYb & \cdot X \longrightarrow aXa\cdot Yb \\
\cdot X \longrightarrow a\cdot XaYb & \cdot X \longrightarrow aXaY\cdot b \\
\cdot X \longrightarrow aX\cdot aYb & \cdot X \longrightarrow aXaYb\cdot
\end{array}$$

2.3.1 – AFD des item $LR(0)$.

Considérons l'ε–AF $\mathbf{A}$, sur l'alphabet $\mathcal{A}+\mathcal{V}$, défini par les données suivantes :

- Les états sont les item ;
- l'entrée est $\emptyset \longrightarrow \cdot S$;
- une transition(*) par $\xi \in \mathcal{A}+\mathcal{V}$ est définie sur les états de la forme $\cdot X \longrightarrow \alpha \cdot \xi\beta$ par un décalage

$$\boxed{\cdot X \longrightarrow \alpha \cdot \xi\beta} \xrightarrow{\quad\xi\quad} \boxed{\cdot X \longrightarrow \alpha\xi \cdot \beta}$$

- une ε–transition est définie sur les états de la forme $\cdot X \longrightarrow \alpha \cdot Y\beta$, par

$$\boxed{\cdot X \longrightarrow \alpha \cdot Y\beta} \xrightarrow{\quad\varepsilon\quad} \boxed{\cdot Y \longrightarrow \cdot \gamma}$$

  pour toute règle $Y \longrightarrow \gamma$.

(*) On représente $r \in \delta(q,\xi)$ par l'arête $q \xrightarrow{\quad\xi\quad} r$ du graphe de transition.

Si l'on fait une sortie de chacun de ses états, **A** reconnaît le langage formé des préfixes viables. Plus précisément

ε–AF des item

Pour tout $\pi \in (\mathcal{A} + \mathcal{V})^*$:
$(\emptyset \longrightarrow \cdot S, \pi) \overset{*}{\vdash\!\!-} (\cdot X \longrightarrow \alpha \cdot \beta, \varepsilon)$ (dans **A**)
ssi
π est un préfixe viable et $\cdot X \longrightarrow \alpha \cdot \beta$ est valide pour π.

Un ε–AF **A** est peu maniable et il est préférable de considérer l'AFD $D(\mathbf{A})$ équivalent obtenu en appliquant l'algorithme de détermination, et en négligeant l'état vide.

L'AFD équivalent à **A** est donc la partie accessible de l'AFD défini de la façon suivante :

- les états sont des ensembles clos non vides d'item,
- l'état initial : $cl(\emptyset \longrightarrow \cdot S)$,
- l'action de ξ sur q, $q \bullet \xi$ est la clôture de la réunion des images des éléments de q,
- les sorties ne jouent aucun rôle ici.

Lorsque $q \bullet \pi \neq \emptyset$, π définit un chemin partant de $q \in Q$: $ch(q, \pi) \in Chem(q, q \bullet \pi)$.

- $ch(q, \varepsilon) = q$,
- si $ch(q, \pi) = \chi r$ et si $r \bullet \xi = s$,
  alors $ch(q, \pi\xi) = ch(q, \pi) \circ rs = \chi rs$.

Propriétés de l'AFD des item.

Toutes ces propriétés viennent directement de la définition de l'AFD et des observations qui ont été faites au sujet des dérivations à droite.

- $q \bullet \xi$ est défini ssi il existe $(\cdot X \longrightarrow \alpha \cdot \xi\beta) \in q$.
- $(\cdot X \longrightarrow \alpha \cdot \xi\beta) \in q$ implique $(\cdot X \longrightarrow \alpha\xi \cdot \beta) \in q \bullet \xi$. En conséquence, toutes les transitions aboutissant à un état donné sont étiquetées par le même symbole, c'est–à–dire que, si $q \in \delta(r, \xi)$ et $q \in \delta(s, \eta)$ alors $\xi = \eta$: un chemin détermine donc au plus un mot.
- on peut reformuler la remarque précédente en disant que, la connaissance de $ch(q, \pi)$ et celle de π sont équivalentes, pour un q donné. Dans la pratique, il est cependant intéressant de les considérer tous les deux simultanément : dans les analyses que nous allons maintenant définir, ils seront traités comme des piles avec sommet à droite ; respectivement "la pile d'états" et "la pile" proprement dite.
- Soient q_0 l'état initial, π un préfixe viable et $q = q_0 \bullet \pi$, alors :
  - q est l'ensemble des item valides pour π,
  - si $(\cdot X \longrightarrow \alpha \cdot \xi\beta) \in q$ alors $\pi\xi$ est viable,
  - si $(\cdot X \longrightarrow \alpha \cdot) \in q$ alors $\pi = \pi'\alpha$ pour un $\pi' \in (\mathcal{A} + \mathcal{V})^*$ et $\pi' X$ est un préfixe viable.

Un item de la forme $\cdot X \longrightarrow \alpha \cdot$ est dit *complet*.

Exemple 1.

Soit G_1 la grammaire :

$$\begin{array}{l} 1 : S \longrightarrow (S \wedge S) \\ 2 : S \longrightarrow \neg S \\ 3 : S \longrightarrow \mathbf{id} \end{array}$$

AFD des item $LR(0)$ de G_1.

(Chaque nouvel état I_i est accompagné de l'ensemble Σ_i des symboles ξ tels que $I_i \bullet \xi \neq \emptyset$.)

$$\begin{array}{rcll} I_0 & = & \{\emptyset \longrightarrow \cdot S, & \Sigma_0 = S + (+\neg + \mathbf{id} \\ & & \cdot S \longrightarrow \cdot(S \wedge S), \cdot S \longrightarrow \cdot \neg S, \cdot S \longrightarrow \cdot \mathbf{id}\} & \\ I_1 & = & I_0 \bullet S & \Sigma_1 = \emptyset \\ & = & \{\emptyset \longrightarrow S \cdot\} & \\ I_2 & = & I_0 \bullet (& \Sigma_2 = S + (+\neg + \mathbf{id} \\ & = & \{\cdot S \longrightarrow (\cdot S \wedge S), & \\ & & \cdot S \longrightarrow \cdot(S \wedge S), \cdot S \longrightarrow \cdot \neg S, \cdot S \longrightarrow \cdot \mathbf{id}\} & \\ I_3 & = & I_0 \bullet \neg & \Sigma_3 = S + (+\neg + \mathbf{id} \\ & = & \{\cdot S \longrightarrow \neg \cdot S, & \\ & & \cdot S \longrightarrow \cdot(S \wedge S), \cdot S \longrightarrow \cdot \neg S, \cdot S \longrightarrow \cdot \mathbf{id}\} & \\ I_4 & = & I_0 \bullet \mathbf{id} & \Sigma_4 = \emptyset \\ & = & \{\cdot S \longrightarrow \mathbf{id} \cdot\} & \\ I_5 & = & I_2 \bullet S & \Sigma_5 = \wedge \\ & = & \{\cdot S \longrightarrow (S \cdot \wedge S)\} & \\ I_2 & = & I_2 \bullet (& \end{array}$$

$$I_3 = I_2 \bullet \neg$$

$$I_4 = I_2 \bullet \mathbf{id}$$

$$\begin{aligned} I_6 &= I_3 \bullet S & \Sigma_6 = \emptyset \\ &= \{\cdot\, S \longrightarrow \neg S \cdot\} \end{aligned}$$

$$I_2 = I_3 \bullet ($$

$$I_3 = I_3 \bullet \neg$$

$$I_4 = I_3 \bullet \mathbf{id}$$

$$\begin{aligned} I_7 &= I_5 \bullet \wedge & \Sigma_7 = S + (+\neg + \mathbf{id} \\ &= \{\cdot\, S \longrightarrow (S \wedge \cdot\, S), \\ &\quad \cdot\, S \longrightarrow \cdot(S \wedge S), \cdot\, S \longrightarrow \cdot\,\neg S, \cdot\, S \longrightarrow \cdot\, \mathbf{id}\} \end{aligned}$$

$$\begin{aligned} I_8 &= I_7 \bullet S & \Sigma_8 =) \\ &= \{\cdot\, S \longrightarrow (S \wedge S \cdot)\} \end{aligned}$$

$$I_2 = I_7 \bullet ($$

$$I_3 = I_7 \bullet \neg$$

$$I_4 = I_7 \bullet \mathbf{id}$$

$$\begin{aligned} I_9 &= I_8 \bullet) & \Sigma_9 = \emptyset \\ &= \{\cdot\, S \longrightarrow (S \wedge S) \cdot\} \end{aligned}$$

Exemple 2.

Soit G_2 la grammaire :

$$\begin{array}{l} 1 : E \longrightarrow E \oplus T \\ 2 : E \longrightarrow T \\ 3 : T \longrightarrow T * F \\ 4 : T \longrightarrow F \\ 5 : F \longrightarrow (E) \\ 6 : F \longrightarrow \mathbf{id} \end{array}$$

AFD des item $LR(0)$ de G_2.

$$\begin{array}{rll} I_0 = & \{\emptyset \longrightarrow \cdot E, & \Sigma_0 = E + T + F + (+\mathbf{id} \\ & \cdot E \longrightarrow \cdot E \oplus T, \cdot E \longrightarrow \cdot T, & \\ & \cdot T \longrightarrow \cdot T * F, \cdot T \longrightarrow \cdot F, & \\ & \cdot F \longrightarrow \cdot (E), \cdot F \longrightarrow \cdot \mathbf{id}\} & \end{array}$$

$$\begin{array}{rll} I_0 \bullet E = & I_1 & \Sigma_1 = \oplus \\ = & \{\emptyset \longrightarrow E\, \cdot, & \\ & \cdot E \longrightarrow E \cdot \oplus T\} & \end{array}$$

$$\begin{array}{rll} I_0 \bullet T = & I_2 & \Sigma_2 = * \\ = & \{\cdot E \longrightarrow T\, \cdot, & \\ & \cdot T \longrightarrow T \cdot * F\} & \end{array}$$

$$\begin{array}{rll} I_0 \bullet F = & I_3 & \Sigma_3 = \emptyset \\ = & \{\cdot T \longrightarrow F \cdot\} & \end{array}$$

$$\begin{array}{rll} I_0 \bullet (= & I_4 & \Sigma_4 = E + T + F + (+\mathbf{id} \\ = & \{\cdot F \longrightarrow (\cdot E), & \\ & \cdot E \longrightarrow \cdot E \oplus T, \cdot E \longrightarrow \cdot T, & \\ & \cdot T \longrightarrow \cdot T * F, \cdot T \longrightarrow \cdot F, & \\ & \cdot F \longrightarrow \cdot (E), \cdot F \longrightarrow \cdot \mathbf{id}\} & \end{array}$$

$$\begin{aligned}
I_0 \bullet \mathbf{id} &= I_5 && \Sigma_5 = \emptyset \\
&= \{\cdot F \longrightarrow \mathbf{id}\,\cdot\}
\end{aligned}$$

$$\begin{aligned}
I_1 \bullet \oplus &= I_6 && \Sigma_6 = T + F + (+\mathbf{id} \\
&= \{\cdot E \longrightarrow E \oplus \cdot T, \\
&\quad \cdot T \longrightarrow \cdot T * F, \cdot T \longrightarrow \cdot F, \\
&\quad \cdot F \longrightarrow \cdot(E), \cdot F \longrightarrow \cdot\, \mathbf{id}\}
\end{aligned}$$

$$\begin{aligned}
I_2 \bullet * &= I_7 && \Sigma_7 = F + (+\mathbf{id} \\
&= \{\cdot T \longrightarrow T * \cdot F, \\
&\quad \cdot F \longrightarrow \cdot(E), \cdot F \longrightarrow \cdot\, \mathbf{id}\}
\end{aligned}$$

$$\begin{aligned}
I_4 \bullet E &= I_8 && \Sigma_8 =) + \oplus \\
&= \{\cdot F \longrightarrow (E\,\cdot), \\
&\quad \cdot E \longrightarrow E \cdot \oplus T\}
\end{aligned}$$

$$I_4 \bullet T = I_2$$

$$I_4 \bullet F = I_3$$

$$I_4 \bullet (= I_4$$

$$I_4 \bullet \mathbf{id} = I_5$$

$$\begin{aligned}
I_6 \bullet T &= I_9 && \Sigma_9 = * \\
&= \{\cdot E \longrightarrow E \oplus T\,\cdot, \\
&\quad \cdot T \longrightarrow T \cdot *F\}
\end{aligned}$$

$$I_6 \bullet F = I_3$$

$$I_6 \bullet (= I_4$$

$$I_6 \bullet \mathbf{id} = I_5$$

$$\begin{aligned} I_7 \bullet F &= I_{10} & \Sigma_{10} = \emptyset \\ &= \{\cdot T \longrightarrow T * F \cdot\} \end{aligned}$$

$$I_7 \bullet (= I_4$$

$$I_7 \bullet \mathbf{id} = I_5$$

$$\begin{aligned} I_8 \bullet) &= I_{11} & \Sigma_{11} = \emptyset \\ &= \{\cdot F \longrightarrow (E) \cdot\} \end{aligned}$$

$$I_8 \bullet \oplus = I_6$$

$$I_9 \bullet * = I_7$$

2.4 – L'automate à pile $LR(0)$.

Une configuration d'analyse LR d'un mot $u \in \mathcal{A}^*$ est un couple (π, v) où π est un préfixe viable tel qu'il existe une dérivation $\pi \underset{d}{\Longrightarrow} w$ vérifiant $wv = u$.

Pour avoir accès à l'état de l'AFD dans lequel π est validé, il faut modifier un peu ce couple, en considérant la configuration $(ch(q_0, \pi), v)$: il est intéressant de considérer aussi le mot π, bien que cette information soit redondante.

L'automate à pile pour l'analyse $LR(0)$ de G est défini de la façon suivante :

Une configuration est un couple (χ, v) où

- $\chi = ch(q_0, \pi)$ est traité comme une pile (*la pile des états*) dont le sommet est à droite ;
  le mot π, lui aussi, est traité comme une pile (*la pile* proprement dite) dont le sommet est encore à droite,
- $v \in \mathcal{A}^*$.

Les transitions sont définies par :

(D) $(\chi q, xv) \longmapsto (\chi qr, v)$ si $x \in \mathcal{A}$ et $q \bullet x = r$

(R) $(\chi ch(q, \alpha), v) \longmapsto (\chi qr, v)$ si $(\cdot X \longrightarrow \alpha \cdot) \in q \bullet \alpha$ et $q \bullet X = r$.

La configuration initiale pour l'analyse de u est (q_0, u).

La configuration d'acceptation est $(q_0 q_1, \varepsilon)$ où $q_1 = q_0 \bullet S$.

Les données nécessaires à la définition des transitions sont consignées dans une table $Action(q, \xi)$ définie pour les états q de l'AFD des item $LR(0)$ et $\xi \in \varepsilon + \mathcal{A} + \mathcal{V}$:

Table $LR(0)$

Les $Action(\_, \_)$ sont les plus petits ensembles tels que

- $r \in Action(q, \xi)$ si $q \bullet \xi = r$ pour $\xi \in \mathcal{A} + \mathcal{V}$
- $(X \longrightarrow \alpha) \in Action(q, \varepsilon)$ si $(\cdot X \longrightarrow \alpha \cdot) \in q$ pour $X \in \mathcal{V}$
- $Acc \in Action(q, \varepsilon)$ si $(\emptyset \longrightarrow S \cdot) \in q$

C'est donc la table de transition de l'AFD des item $LR(0)$ de G enrichi d'une colonne pour ε qui comporte, pour chaque état, l'ensemble des "item complets" qu'il contient : on l'appelle *la table* $LR(0)$ *de* G.

Propriété.

Pour tout $u \in \mathcal{A}^*$ *:*

$$u \in \mathcal{L}(G, S) \text{ ssi } (q_0, u) \stackrel{*}{\longmapsto} (q_0 q_1, \varepsilon)$$

où $\stackrel{*}{\longmapsto}$ *signifie l'existence d'un enchaînement de transitions.*

La démonstration se fait par récurrence.

2.5 – Grammaires $LR(0)$.

Une grammaire est dite $LR(0)$ ssi sa table $LR(0)$ vérifie les propriétés suivantes :

pour tout état q de l'AFD

- $Action(q, \varepsilon)$ comporte au plus un élément,
- si $Action(q, \varepsilon) \neq \emptyset$ alors $Action(q, \xi) = \emptyset$ pour tout $\xi \in \mathcal{A}$.

Remarques.

• Lorsque ces conditions sont vérifiées, chaque état q est d'une nature bien déterminée :

- si $Action(q, \varepsilon) \neq \emptyset$, alors $Action(q, \varepsilon)$ contient un seul élément qui est :
  - ou bien une réduction : q est un *état de réduction*,
  - ou bien Acc : $q = q_1 = q_0 \bullet S$ est l'*état d'acceptation*,
- sinon, $Action(q, x)$ contient au plus un décalage pour chaque $x \in \mathcal{A}$: q est un *état de décalage*.

• Les conditions signifient que l'automate à pile est déterministe : une analyse, lorsqu'elle est faisable, l'est de façon unique. En particulier, une grammaire $LR(0)$ n'est pas ambiguë.

2.5.1 – Algorithme d'analyse $LR(0)$.

Soit G une grammaire $LR(0)$ dont l'axiome est S.
L'analyse $LR(0)$ de $u \in \mathcal{A}^*$ s'effectue à partir de la configuration initiale (q_0, u) en tentant d'exécuter une suite de transitions (D) ou (R). Chaque application de (R) ajoute une règle au début de la liste d'analyse, initialament vide.

Soit (χ, v) la configuration courante. Lors de l'exécution de l'algorithme, χ n'est jamais vide : notons $\chi = \chi' q$ pour mettre en valeur le sommet q de cette pile, qui est l'"état courant" de l'automate. Alors :

- si q est un état de réduction :
  si $Action(q, \varepsilon) = (X \longrightarrow \alpha)$, χ est de la forme $\chi'' ch(r, \alpha)$ et si $Action(r, X) = s$: on passe à la configuration $(\chi'' rs, v)$ par application d'une transition (R),
- si q est un état de décalage :
  si $v = xv'$ avec $x \in \mathcal{A}$ et si $Action(q, x) = r$: on passe à la configuration $(\chi' qr, v')$ par application d'une transition (D),
- si $q = q_1 = q_0 \bullet S$ est l'état d'acceptation :
  si $\chi = q_0 q_1$ et si $v = \varepsilon$: on est parvenu à la configuration d'acceptation et l'analyse est terminée de façon satisfaisante.

• Dans les autres cas : l'analyse se termine sur un échec.
Une bonne table d'analyse doit comporter un diagnostic d'"erreur syntaxique" pour chacun de ces cas (les cases vides de la table).

Présentation pratique de la table $LR(0)$.

• Les états sont énumérés $I_0, \ldots, I_n, \ldots$: on note di (décaler en i) au lieu de I_i dans la table de transition.

• Les règles sont codées par des entiers : on note rk (réduire par la règle k), au lieu de la règle elle–même, dans la colonne ε (la réduction par $\emptyset \longrightarrow S$, est codée par Acc).

Ceci donne la description suivante de la table $LR(0)$:

– si $I_i \bullet \xi = I_j$ alors $dj \in Action(i, \xi)$
– si $(\cdot X \longrightarrow \alpha \cdot) \in I_i$ avec $X \in \mathcal{V}$ alors
$$rk \in Action(i, \varepsilon)$$
où k est le numéro de la règle $X \longrightarrow \alpha$
– si $(\emptyset \longrightarrow S \cdot) \in I_i$ alors $Acc \in Action(i, \varepsilon)$.

Exemple.

Table $LR(0)$ de G_1.

	ε	(	$\wedge$	)	$\neg$	**id**	S
0		$d2$			$d3$	$d4$	$d1$
1	Acc						
2		$d2$			$d3$	$d4$	$d5$
3		$d2$			$d3$	$d4$	$d6$
4	r3						
5			$d7$				
6	r2						
7		$d2$			$d3$	$d4$	$d8$
8				$d9$			
9	r1						

L'analyse $LR(0)$ de $\neg(\mathbf{id} \wedge \neg\mathbf{id})$ dans G_1.

Pile	Pile d'états	Entrée	Act.
ε	0	$\neg(\mathbf{id} \wedge \neg\mathbf{id})$	$d3$
$\neg$	03	$(\mathbf{id} \wedge \neg\mathbf{id})$	$d2$
$\neg($	032	$\mathbf{id} \wedge \neg\mathbf{id})$	$d4$
$\neg(\mathbf{id}$	0324	$\wedge\neg\mathbf{id})$	$r3$
$\neg(S$	0325	$\wedge\neg\mathbf{id})$	$d7$
$\neg(S\wedge$	03257	$\neg\mathbf{id})$	$d3$
$\neg(S \wedge \neg$	032573	$\mathbf{id})$	$d4$
$\neg(S \wedge \neg\mathbf{id}$	0325734	$)$	$r3$
$\neg(S \wedge \neg S$	0325736	$)$	$r2$
$\neg(S \wedge S$	032578	$)$	$d9$
$\neg(S \wedge S)$	0325789	ε	$r1$
$\neg S$	036	ε	$r2$
S	01	ε	Acc

3 – Analyse ascendante avec symboles de prévision.

- Lorsque la première condition $LR(0)$ n'est pas vérifiée, on parle d'un "conflit réduction–réduction",
- lorsque la deuxième ne l'est pas, d'un "conflit décalage–réduction".

Ces conflits sont fréquents dès que G n'est plus très simpliste : ceci tient à ce que l'on n'est pas prévoyant.

Exemple de conflit décalage–réduction.

L'état $I_9 = \{\cdot E \longrightarrow E \oplus T \cdot, \cdot T \longrightarrow T \cdot *F\}$ de l'AFD de la grammaire G_2 de l'exemple 2 , contient un item complet, correspondant à une réduction par la règle $E \longrightarrow E \oplus T$ et un item non complet, permettant un décalage sur $*$.

On adjoint des prévisions à 1 caractère pour résoudre ces conflits pour des grammaires assez intéressantes.

Chacun des types d'analyse ainsi obtenu (il y en a trois) est caractérisé par une gestion particulière des "symboles de prévision" lors de la définition de l'AF de ses item.

3.1 – Symboles de prévision.

Pour obtenir des algorithmes déterministes, nous allons les rendre "prévoyants". Le choix de l'action à exécuter est déterminé par la connaissance de quelques caractères du début de la partie du mot restant à analyser. Voici ce que l'on entend par là :

• Si k est le nombre de ces caractères et $v \in \mathcal{A}^*$, on définit $Premier_k(v) \in (\varepsilon + \mathcal{A})^k$ par :

- si $|v| \leq k$: $Premier_k(v) = v$,
- sinon $Premier_k(v)$ est le facteur gauche de v dont la longueur est k.

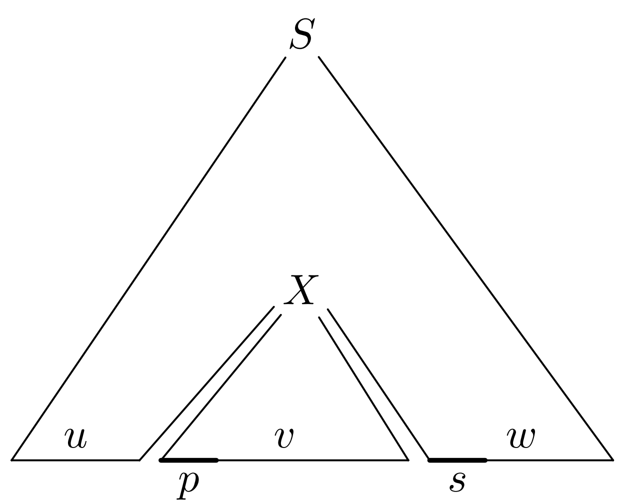
• Il faut étendre cette notion à des mots sur $\mathcal{A} + \mathcal{V}$, en fonction de G et de S, de la façon suivante :

Soit $S \overset{k}{\Longrightarrow} uXw \overset{l}{\Longrightarrow} uvw$ une dérivation de $uvw \in \mathcal{A}^*$, dans laquelle on a mis en évidence une sous–dérivation $X \overset{l}{\Longrightarrow} v$.

Posons :

$p = Premier_k(v)$
$s = Premier_k(w)$.

$Premier_k(X)$ est l'ensemble des p et $Suivant_k(X)$ celui des s, que l'on peut obtenir ainsi.



Premier et *Suivant* désigneront respectivement $Premier_1$ et $Suivant_1$.

3.1.1 – Calcul de *Premier*.

Pour toute $X \in \mathcal{V}$, l'ensemble $Premier(X)$ est formé de facteurs gauches des mots sur $\mathcal{A}$ que l'on peut dériver à partir de X. Plus précisément, $Premier(X) \subseteq \varepsilon + \mathcal{A}$ est le plus petit ensemble tel que :

– si $X \stackrel{*}{\Longrightarrow} x\alpha$ pour $x \in \mathcal{A}$, alors $x \in Premier(X)$;
– si $X \stackrel{*}{\Longrightarrow} \varepsilon$, alors $\varepsilon \in Premier(X)$.

La deuxième clause signifie :

$$\varepsilon \in Premier(X) \text{ ssi } X \in \mathcal{E}ps(G).$$

$Premier(X) \neq \emptyset$ car toute variable est productive.

Plus généralement :

Premier

1) $Premier(\varepsilon) = \varepsilon$
2) pour tout $x \in \mathcal{A}$ et tout $\alpha \in (\mathcal{A} + \mathcal{V})^*$:
$$Premier(x\alpha) = x$$
3) pour toute $X \in \mathcal{V}$ et tout $\alpha \in (\mathcal{A} + \mathcal{V})^*$:
   – si $X \in \mathcal{E}ps(G)$ alors :
$$Premier(X\alpha) = (Premier(X) - \varepsilon) + Premier(\alpha)$$
   – sinon :
$$Premier(X\alpha) = Premier(X).$$
4) pour toute règle globale $X \longrightarrow \mathbf{l}$:
$$Premier(X) = Premier(\mathbf{l}).$$

Exemple.

Appliquons cette construction à la grammaire G :

$$\begin{array}{ll} 1 : E \longrightarrow E \oplus T & 4 : T \longrightarrow F \\ 2 : E \longrightarrow T & 5 : F \longrightarrow (E) \\ 3 : T \longrightarrow T * F & 6 : F \longrightarrow \mathbf{id} \end{array}$$

- Il est clair que $\mathcal{E}ps(G) = \emptyset$;
- En apliquant 4) et 3) on voit que :

$$\begin{aligned} Premier(E) &= Premier(E \oplus T) + Premier(T) \\ &= Premier(E) + Premier(T) \end{aligned}$$

donc

$$Premier(T) \subseteq Premier(E).$$

$$\begin{aligned} Premier(T) &= Premier(T * F) + Premier(F) \\ &= Premier(T) + Premier(F) \end{aligned}$$

d'où

$$Premier(F) \subseteq Premier(T).$$

- En faisant maintenant intervenir 2) :

$$Premier(F) = Premier((E)) + Premier(\mathbf{id}) = (\ + \mathbf{id}.$$

- On peut conclure en prenant les plus petits ensembles vérifiant ces propriétés :

$$Premier(E) = Premier(T) = Premier(F) = (\ + \mathbf{id}.$$

3.1.2 – Calcul de *Suivant*.

Pour tout $X \in \mathcal{V}$, l'ensemble $Suivant(X)$ est formé de facteurs de mots sur $\mathcal{A}$ qui peuvent suivre immédiatement une occurrence de X dans un mot que l'on peut dériver à partir de S. Plus précisément, $Suivant(X) \subseteq \varepsilon + \mathcal{A}$ est le plus petit ensemble tel que :

si $S \stackrel{*}{\Longrightarrow} \alpha X \beta$, alors $Premier(\beta) \subseteq Suivant(X)$.

$Suivant(X) \neq \emptyset$ car toute variable est accessible à partir de S.

Le calcul explicite est basé sur la propriété :

Suivant

Les $Suivant(\_)$ sont les plus petits ensembles tels que :

1) $\varepsilon \in Suivant(S)$
2) si $Y \longrightarrow \alpha X \beta$ pour $Y \in \mathcal{V}$:
   $Premier(\beta) - \varepsilon \subseteq Suivant(X)$
3) si $Y \longrightarrow \alpha X \beta$ pour $Y \in \mathcal{V}$ et si $\varepsilon \in Premier(\beta)$:
   $Suivant(Y) \subseteq Suivant(X)$

Remarques.

- X peut se trouver en plusieurs occurrences dans les clauses 2) et 3).
- On peut avoir $\beta = \varepsilon$ dans la clause 3).

On a évidemment $Suivant_0(X) = \varepsilon$

Exemple.

Reprenons la grammaire de l'exemple précédent.

- $Suivant(E)$: Par 1), $\varepsilon \in Suivant(E)$

En appliquant 2) aux règles contenant E à droite, il vient

$$\oplus \in Suivant(E)$$

par 1 : et

$$) \in Suivant(E)$$

par 5 :.

Comme on a épuisé toutes les possibilités, on peut en conclure que

$$Suivant(E) = \varepsilon + \oplus +)$$

- $Suivant(T)$: En appliquant 3) à 1 : ou 2 :, il vient

$$Suivant(E) \subseteq Suivant(T)$$

En appliquant 2) à la règle 3 :, on a

$$* \subseteq Suivant(T).$$

Finalement

$$Suivant(T) = \varepsilon + \oplus + * +).$$

- $Suivant(F)$: L'application de 3) à 3 : ou 4 : implique que

$$Suivant(T) \subseteq Suivant(F).$$

Comme il n'y a pas d'autre possibilité, on conclut que

$$Suivant(F) = Suivant(T) = \varepsilon + \oplus + * +)$$

3.2 – Automate à pile de type $LR(1)$.

La définition de l'automate à pile s'effectue comme plus haut, à l'exception des transitions qui tiennent compte des symboles de prévision :

(D) $(\chi q, xv) \longmapsto (\chi qr, v)$ si $x \in \mathcal{A}$ et $q.x = r$

(R) $(\chi ch(q, \alpha), v) \longmapsto (\chi qr, v)$ si

$$(\cdot X \longrightarrow \alpha \cdot, Premier(v)) \in q.\alpha \text{ et } q.X = r.$$

Il est commode de représenter les données nécessaires à la définition des transitions dans une "table" $Action(q, \xi)$ définie pour les états q de l'AFD des item de type $LR(1)$ et $\xi \in \varepsilon + \mathcal{A} + \mathcal{V}$, de la façon suivante :

Table de type $LR(1)$

Les $Action(\_, \_)$ sont les plus petits ensembles tels que

- $r \in Action(q, \xi)$ si $q \bullet \xi = r$ pour $\xi \in \mathcal{A} + \mathcal{V}$
- $(X \longrightarrow \alpha) \in Action(q, x)$ si $(\cdot X \longrightarrow \alpha \cdot, x) \in q$ pour $X \in \mathcal{V}$
- $Acc \in Action(q, \varepsilon)$ si $(\emptyset \longrightarrow S \cdot, \varepsilon) \in q$

C'est donc la réunion de la table de transition de l'AFD en question et d'une table caractérisant la position des "item complets".

Il faut remarquer que l'application d'une transition (R) est maintenant sujette à une condition sur $Premier(v) \in \varepsilon + \mathcal{A}$, contrairement au cas $LR(0)$.

3.2.1 – Les trois types $LR(1)$.

$SLR(1)$ (*Simple* $LR(1)$) :

dans ce type, on forme un item pour $\cdot X \longrightarrow \alpha\cdot$ avec **tout** élément de $Suivant(X)$.

$LR(1)$ proprement dit :

l'action de l'AF est définie de la façon suivante :

- une transition(*) par $\xi \in \mathcal{A}+\mathcal{V}$ est définie sur les états de la forme $(\cdot X \longrightarrow \alpha\cdot\xi\beta, x)$ par

$$\boxed{\cdot X \longrightarrow \alpha\cdot\xi\beta, x} \xrightarrow{\quad\xi\quad} \boxed{\cdot X \longrightarrow \alpha\xi\cdot\beta, x}$$

- une ε–transition est définie sur les états de la forme $(\cdot X \longrightarrow \alpha\cdot Y\beta, x)$ par

$$\boxed{\cdot X \longrightarrow \alpha\cdot Y\beta, x} \xrightarrow{\quad\varepsilon\quad} \boxed{\cdot Y \longrightarrow \cdot\gamma, y}$$

  pour chaque $(Y \longrightarrow \gamma) \in R$ et pour chaque $y \in Premier(\beta x)$.

$LALR(1)$ (*Look Ahead* $LR(1)$) :

les états de l'AFD correspondant à ce type s'obtiennent en faisant la réunion des états de l'AFD $LR(1)$ qui ne diffèrent que par les symboles de prévision : ses états sont ceux de l'AFD $LR(0)$, aux éléments desquels on a adjoint des symboles de prévision.

(*) On représente $r \in \delta(q,\xi)$ par l'arête $q \xrightarrow{\quad\xi\quad} r$

3.3 – Grammaires de type $LR(1)$.

Une grammaire est dite $LR(1)$ (resp. $SLR(1)$, $LALR(1)$) ssi sa table $LR(1)$ (resp. $SLR(1)$, $LALR(1)$) est déterministe, c'est–à–dire, ssi chaque $Action(q, \xi)$ comporte toujours au plus un élément.

Remarques.

- Cette condition signifie qu'il ne se produit pas de conflit et assure évidemment que l'automate à pile correspondant est déterministe.
- Contrairement au cas $LR(0)$, la nature de la transition à exécuter n'est pas déterminée par le seul état courant (sommet de la pile d'états) mais aussi par le symbole de prévision.
- Comme dans le cas $LR(0)$, on peut voir qu'une grammaire de type $LR(1)$ n'est pas ambiguë.

3.3.1 – Algorithme d'analyse de type $LR(1)$.

Soit G une grammaire de type $LR(1)$ dont l'axiome est S. L'analyse $LR(1)$ de $u \in \mathcal{A}^*$ s'effectue à partir de la configuration initiale (q_0, u) en tentant d'exécuter une suite de transitions (D) ou (R). Chaque application de (R) ajoute une règle au début de la liste d'analyse, initialement vide.

Soit (χ, v) la configuration courante. Lors de l'exécution de l'algorithme, on peut vérifier que χ n'est jamais vide : notons $\chi = \chi' q$ pour mettre en valeur le sommet q de cette pile, qui est l'"état courant" de l'automate. Alors :

• si $Action(q, Premier(v)) = (X \longrightarrow \alpha)$, χ est alors nécessairement de la forme $\chi'' ch(r, \alpha)$ (comme on peut le vérifier par une induction) et si $Action(r, X) = s$: on passe à la configuration $(\chi'' rs, v)$ par une transition (R),

• si $Action(q, Premier(v)) = r$ où r est un état (alors on a nécessairement $Premier(v) = x \in \mathcal{A}$ et) v s'écrit sous la forme $v = xv'$, et si $Action(q, x) = r$: on passe à la configuration $(\chi' qr, v')$ par une transition (D),

• si $\chi = q_0 q_1$ et si $v = \varepsilon$: on est parvenu à la configuration d'acceptation et l'analyse est terminée de façon satisfaisante.

• Dans les autres cas : l'analyse se termine sur un échec. Un bonne table d'analyse doit comporter un diagnostic d'"erreur syntaxique" pour chacun de ces cas (qui correspondent aux cases vides de la table).

Le codage introduit pour la table $LR(0)$ est encore appliqué ici ; ceci donne la description suivante :

– si $I_i \bullet \xi = I_j$ alors $dj \in Action(i, \xi)$
– si $(\cdot X \longrightarrow \alpha \cdot, x) \in I_i$ avec $X \in \mathcal{V}$ alors $rk \in Action(i, x)$ où k est le numéro de la règle $X \longrightarrow \alpha$
– si $(\emptyset \longrightarrow S \cdot, \varepsilon) \in I_i$ alors $Acc \in Action(i, \varepsilon)$.

Exemple 2 : analyse $SLR(1)$ de G_2.

$\mathcal{E}ps(G) = \emptyset$
$Premier(E) = Premier(T) = Premier(F) = (+\mathbf{id}$
$Suivant(E) = \varepsilon + \oplus +)$
$Suivant(T) = Suivant(F) = \varepsilon + \oplus + *+)$

	ε	$\oplus$	$*$	(	)	**id**	E	T	F
0				$d4$		$d5$	$d1$	$d2$	$d3$
1	Acc	$d6$							
2	$r2$	$r2$	$d7$		$r2$				
3	$r4$	$r4$	$r4$		$r4$				
4				$d4$		$d5$	$d8$	$d2$	$d3$
5	$r6$	$r6$	$r6$		$r6$				
6				$d4$		$d5$		$d9$	$d3$
7				$d4$		$d5$			$d10$
8		$d6$			$d11$				
9	$r1$	$r1$	$d7$		$r1$				
10	$r3$	$r3$	$r3$		$r3$				
11	$r5$	$r5$	$r5$		$r5$				

Table $SLR(1)$ de G_2.

Pile	Pile d'états	Entrée	Act.
ε	0	$\mathbf{id} \oplus \mathbf{id} * \mathbf{id}$	$d5$
$\mathbf{id}$	0 5	$\oplus \mathbf{id} * \mathbf{id}$	$r6$
F	0 3	$\oplus \mathbf{id} * \mathbf{id}$	$r4$
T	0 2	$\oplus \mathbf{id} * \mathbf{id}$	$r2$
E	0 1	$\oplus \mathbf{id} * \mathbf{id}$	$d6$
$E\oplus$	0 1 6	$\mathbf{id} * \mathbf{id}$	$d5$
$E \oplus \mathbf{id}$	0 1 6 5	$*\mathbf{id}$	$r6$
$E \oplus F$	0 1 6 3	$*\mathbf{id}$	$r4$
$E \oplus T$	0 1 6 9	$*\mathbf{id}$	$d7$
$E \oplus T*$	0 1 6 9 7	$\mathbf{id}$	$d5$
$E \oplus T * \mathbf{id}$	0 1 6 9 7 5	ε	$r6$
$E \oplus T * F$	0 1 6 9 7 10	ε	$r3$
$E \oplus T$	0 1 6 9	ε	$r1$
E	0 1	ε	Acc

L'analyse $SLR(1)$ de $\mathbf{id} \oplus \mathbf{id} * \mathbf{id}$ dans G_2.

Exemple 3 : analyses $LR(1)$ et $LALR(1)$.

Considérons la grammaire G_3 :

$$\begin{array}{ll} 1: S \longrightarrow G\texttt{=}D & 3: G \longrightarrow *D \\ 2: S \longrightarrow D & 4: G \longrightarrow \mathbf{id} \\ & 5: D \longrightarrow G \end{array}$$

Le début du calcul de l'AFD des item $LR(0)$ montre que G_3 n'est pas $SLR(1)$, en effet, on a

$$\begin{aligned} I_0 = \; & \{\emptyset \longrightarrow \cdot S, \\ & \cdot S \longrightarrow \cdot G\texttt{=}D, \cdot S \longrightarrow \cdot D, \\ & \cdot G \longrightarrow \cdot *D, \cdot G \longrightarrow \cdot \mathbf{id}, \\ & \cdot D \longrightarrow \cdot G\} \end{aligned}$$

et donc $I_2 = I_0 \bullet G = \{\cdot S \longrightarrow G \cdot \texttt{=}D, \cdot D \longrightarrow G \cdot\}$; or, il est facile de voir que $\texttt{=} \in Suivant(D)$ et donc que la méthode $SLR(1)$ ne résoud pas le conflit décalage–réduction qui se présente dans cet état.

AFD des item $LR(1)$ de G_3.

L'état initial est :

$$\begin{aligned} I_0 = \; & \{(\emptyset \longrightarrow \cdot S, \varepsilon), \\ & (\cdot S \longrightarrow \cdot G\texttt{=}D, \varepsilon), (\cdot S \longrightarrow \cdot D, \varepsilon), \\ & (\cdot G \longrightarrow \cdot *D, \texttt{=}), (\cdot G \longrightarrow \cdot \mathbf{id}, \texttt{=}), \\ & (\cdot D \longrightarrow \cdot G, \varepsilon), \\ & (\cdot G \longrightarrow \cdot *D, \varepsilon), (\cdot G \longrightarrow \cdot \mathbf{id}, \varepsilon)\} \end{aligned}$$

On regroupe les item d'un état qui ne diffèrent que par leur symbole de prévision, en notant (ϱ, P) un ensemble d'item de ce type $\sum_{x \in P} (\varrho, x)$, où $P \subseteq \varepsilon + \mathcal{A}$.

Reprenons notre calcul en utilisant cette notation.

$$
\begin{array}{rll}
I_0 = & \{(\emptyset \longrightarrow \cdot S, \varepsilon), & \Sigma_0 = S + G + D + * + \mathbf{id} \\
& (\cdot S \longrightarrow \cdot G\texttt{=}D, \varepsilon), (\cdot S \longrightarrow \cdot D, \varepsilon), & \\
& (\cdot G \longrightarrow \cdot *D, \varepsilon + \texttt{=}), (\cdot G \longrightarrow \cdot \mathbf{id}, \varepsilon + \texttt{=}), & \\
& (\cdot D \longrightarrow \cdot G, \varepsilon)\} & \\
I_0 \bullet S = & I_1 & \Sigma_1 = \emptyset \\
= & \{(\emptyset \longrightarrow S \cdot, \varepsilon)\} & \\
I_0 \bullet G = & I_2 & \Sigma_2 = \texttt{=} \\
= & \{(\cdot S \longrightarrow G \cdot \texttt{=}D, \varepsilon), (\cdot D \longrightarrow G \cdot, \varepsilon)\} & \\
I_0 \bullet D = & I_3 & \Sigma_3 = \emptyset \\
= & \{(\cdot S \longrightarrow D \cdot, \varepsilon)\} & \\
I_0 \bullet * = & I_4 & \Sigma_4 = D + G + * + \mathbf{id} \\
= & \{(\cdot G \longrightarrow * \cdot D, \varepsilon + \texttt{=}), & \\
& (\cdot D \longrightarrow \cdot G, \varepsilon + \texttt{=}), & \\
& (\cdot G \longrightarrow \cdot *D, \varepsilon + \texttt{=}), (\cdot G \longrightarrow \cdot \mathbf{id}, \varepsilon + \texttt{=})\} & \\
I_0 \bullet \mathbf{id} = & I_5 & \Sigma_5 = \emptyset \\
= & \{(\cdot G \longrightarrow \mathbf{id} \cdot, \varepsilon + \texttt{=})\} & \\
I_2 \bullet \texttt{=} = & I_6 & \Sigma_6 = D + G + * + \mathbf{id} \\
= & \{(\cdot S \longrightarrow G\texttt{=} \cdot D, \varepsilon), & \\
& (\cdot D \longrightarrow \cdot G, \varepsilon), & \\
& (\cdot G \longrightarrow \cdot *D, \varepsilon), (\cdot G \longrightarrow \cdot \mathbf{id}, \varepsilon)\} & \\
I_4 \bullet D = & I_7 & \Sigma_7 = \emptyset \\
= & \{(\cdot G \longrightarrow *D \cdot, \varepsilon + \texttt{=})\} &
\end{array}
$$

$$
\begin{aligned}
I_4 \bullet G = {} & I_8 && \Sigma_8 = \emptyset \\
= {} & \{(\cdot\, D \longrightarrow G\, \cdot, \varepsilon + \texttt{=})\} \\[1ex]
I_4 \bullet * = {} & I_4 \\[1ex]
I_4 \bullet \mathbf{id} = {} & I_5 \\[1ex]
I_6 \bullet D = {} & I_9 && \Sigma_9 = \emptyset \\
= {} & \{(\cdot\, S \longrightarrow G\texttt{=}D\, \cdot, \varepsilon)\} \\[1ex]
I_6 \bullet G = {} & I_{10} && \Sigma_{10} = \emptyset \\
= {} & \{(\cdot\, D \longrightarrow G\, \cdot, \varepsilon)\} \\[1ex]
I_6 \bullet * = {} & I_{11} && \Sigma_{11} = D + G + * + \mathbf{id} \\
= {} & \{(\cdot\, G \longrightarrow * \cdot D, \varepsilon), \\
& (\cdot\, D \longrightarrow \cdot\, G, \varepsilon), \\
& (\cdot\, G \longrightarrow \cdot *D, \varepsilon), (\cdot\, G \longrightarrow \cdot\, \mathbf{id}, \varepsilon)\} \\[1ex]
I_6 \bullet \mathbf{id} = {} & I_{12} && \Sigma_{12} = \emptyset \\
= {} & \{(\cdot\, G \longrightarrow \mathbf{id}\, \cdot, \varepsilon)\} \\[1ex]
I_{11} \bullet D = {} & I_{13} && \Sigma_{13} = \emptyset \\
= {} & \{(\cdot\, G \longrightarrow *D\, \cdot, \varepsilon)\} \\[1ex]
I_{11} \bullet G = {} & I_{10} \\[1ex]
I_{11} \bullet * = {} & I_{11} \\[1ex]
I_{11} \bullet \mathbf{id} = {} & I_{12}
\end{aligned}
$$

Les item $LR(1)$ définissent un AFD sans conflit ; dans l'état I_2 qui était conflictuel dans la version $SLR(1)$, = n'est plus utilisé pour déclencher une réduction par $D \longrightarrow G$: la table $LR(1)$ de G_3 est déterministe.

	ε	=	$*$	**id**	S	G	D
0			$d4$	$d5$	$d1$	$d2$	$d3$
1	Acc						
2	$r5$	$d6$					
3	$r2$						
4			$d4$	$d5$		$d8$	$d7$
5	$r4$	$r4$					
6			$d11$	$d12$		$d10$	$d9$
7	$r3$	$r3$					
8	$r5$	$r5$					
9	$r1$						
10	$r5$						
11			$d11$	$d12$		$d10$	$d13$
12	$r4$						
13	$r3$						

Table $LR(1)$ de G_3.

Lorsque l'on fait la réunion des états de l'AFD $LR(1)$ qui ne diffèrent que par les symboles de prévision, on ne réintroduit pas de conflit : la table $LALR(1)$ a été construite en numérotant les nouveaux états de la façon suivante :

$$I'_4 = I_4 + I_{11} \qquad I'_7 = I_7 + I_{13}$$
$$I'_5 = I_5 + I_{12} \qquad I'_8 = I_8 + I_{10}$$

La grammaire G_3 est donc $LALR(1)$.

	ε	**=**	$*$	**id**	S	G	D
0			$d4$	$d5$	$d1$	$d2$	$d3$
1	Acc						
2	$r5$	$d6$					
3	$r2$						
4			$d4$	$d5$		$d8$	$d7$
5	$r4$	$r4$					
6			$d4$	$d5$		$d8$	$d9$
7	$r3$	$r3$					
8	$r5$	$r5$					
9	$r1$						

Table $LALR(1)$ de G_3.