Cours de Langages formels

Laure Danthony - Marianne Delorme

Grammaires LL et LR

Introduction

Il existe deux types d'analyse [1] :

- **L'analyse descendante** : à partir de l'axiome, on construit une dérivation du mot à analyser. Pour l'analyse LL, ces dérivations s'effectuent "le plus à gauche".
- **L'analyse ascendante** : on construit une dérivation qui commence par l'axiome (en partant de la fin) . Pour l'anlyse LR ce sont des dérivations "le plus à droite".

REMARQUE 1 La lecture du mot se fait **toujours** de gauche à droite. D'où la signification du L de LR et LL.

Intuitivement, on note $L?(k)$ une grammaire si le test d'appartenance peut être réalisé avec k "look-ahead". C'est l'idée intuitive du formalisme qui va suivre.

1 Grammaires $LL(k)$

DÉFINITION 1
Si $|w| > k$, $k : w$ désigne le préfixe de longueur k du mot w, sinon $k : w$ désigne w lui-même.

DÉFINITION 2
Une grammaire algébrique $G = (\Sigma, N, P, S)$ est dite $LL(k)$, $k \geq 0$ lorsque la condition suivante est vérifiée :

$$\begin{cases} (1) \quad S \overset{*}{\underset{g}{\longmapsto}} uA\theta \longmapsto u\alpha\theta \overset{*}{\underset{g}{\longmapsto}} uw \\ (2) \quad S \overset{*}{\underset{g}{\longmapsto}} uA\theta \longmapsto u\beta\theta \overset{*}{\underset{g}{\longmapsto}} uw' \Rightarrow \alpha = \beta \\ (3) \quad k : w = k : w' \end{cases}$$

REMARQUE 2 Une telle grammaire n'est pas ambiguë.

EXEMPLE 1

[1] c'est à dire deux manières de savoir si $u \in L(G)$

- $\begin{cases} S \to aAb \mid b \\ A \to a \mid bSA \end{cases}$ *est* $LL(1)$.
- $\begin{cases} S \to \varepsilon \mid abA \\ A \to Saa \mid b \end{cases}$ *est* $LL(2)$.

REMARQUE 3 Un langage $LL(0)$ est soit vide soit réduit à une seule lettre.

EXERCICE 1 *Montrez que les langages rationnels sont* $LL(1)$

REMARQUE 4 Il existe des grammaires algébriques qui ne sont $LL(k)$ pour aucun k, par exemple :

- La grammaire : $\begin{cases} S \to A \mid B \\ A \to aAb \mid 0 \\ B \to aBbb \mid 1 \end{cases}$ qui décrit le langage $\{a^n 0 b^n\} \cup \{a^n 1 b^{2n}\}$
- La grammaire :$S \to b \mid S\ a$ qui décrit le langage $\{ba^n, n \geq 1\}$

2 Les fonctions $Premier_k$, $Suivant_k$

DÉFINITION 3
Soit $G = (\Sigma, N, P, S)$ une grammaire. Si $\alpha \in (\Sigma \cup N)^*$, on définit pour tout $k \geq 0$ les ensembles suivants :

- $Premier_k(\alpha) = \{k : w \mid \alpha \overset{*}{\underset{G}{\vdash}} w\}$
- $Suivant_k(\alpha) = \{w \in \Sigma^* \mid \exists \beta, \gamma, S \longmapsto \beta\alpha\gamma \text{ et } w \in Premier_k(\gamma)\}$

DÉFINITION 4
Si L est un langage, on note :
$Premier_k(L) = \bigcup_{\alpha \in L} Premier_k(\alpha)$ et $Suivant_k(L) = \bigcup_{\alpha \in L} Suivant_k(\alpha)$.

Nous allons maintenant énoncer différentes caractérisations des grammaires $LL(k)$:

PROPOSITION 1 *Une grammaire algébrique est* $LL(k)$ *ssi pour toutes productions distinctes* $A \to \alpha$ *et* $A \to \beta$*, l'intersection des ensembles* $Premier_k(\alpha\gamma)$ *et* $Premier_k(\beta\gamma)$ *est vide, pour tout* γ *tel que* $S \overset{*}{\underset{g}{\vdash}} uA\gamma$ *existe.*

PREUVE :

- : $\Leftarrow$ Si G n'est pas $LL(k)$, d'après la définition, il existe $\alpha \neq \beta$, ...Alors $k : w$ appartient aux deux ensembles $Premier_k(\alpha\theta)$ et $Premier_k(\beta\theta)$ et donc l'intersection est non vide.
- : $\Rightarrow$ Si la condition de droite n'est pas satisfaite, alors il existe un mot u dans l'intersection et à l'aide de ce mot on obtient L qui n'est pas $LL(k)$.

■

COROLLAIRE 1 *Les grammaires* $LL(k)$ *ne sont pas récursives à gauche.*

PREUVE : Supposons que G soit $LL(k)$ pour un certain k et récursive à gauche. Alors $\exists A \to A\alpha$ et $A \to \beta$ dans P (on suppose toujours qu'aucun symbole n'est inutile). Alors considérons les deux dérivations $S \overset{*}{\underset{g}{\vdash}} uA\theta \overset{+}{\underset{g}{\vdash}} uA\alpha^n\theta \longmapsto uA\alpha^{n+1}\theta$ et $S \overset{*}{\underset{g}{\vdash}} uA\theta \overset{+}{\underset{g}{\vdash}} uA\alpha^n\theta \longmapsto u\beta\alpha^n\theta$. Comme G est $LL(k)$, les ensembles $Premier_k(A\alpha^{n+1}\theta)$ et $Premier_k(\beta\alpha^n\theta)$ sont d'intersection nulle. Comme $A \to \beta \in P$, $Premier_k(\beta\alpha^{n+1}\theta) \subseteq Premier_k(A\alpha^{n+1}\theta)$. Donc :

$$Premier_k(\beta\alpha^{n+1}\theta) \cap Premier_k(\beta\alpha^n\theta) = \emptyset.$$

Mais alors, si $\alpha \longmapsto \varepsilon$ et on a une contraction avec l'égalité précédente. Sinon, on considère $n > k$ pour avoir encore une contradiction.

■

Et encore une caractérisation, mais cette fois pour $k = 1$:

PROPOSITION 2 *Une grammaire est $LL(1)$ ssi pour toutes productions distinctes $A \to \alpha$ et $A \to \beta$, on a :*

$$Premier_1\big(\alpha Suivant_1(A)\big) \cap Premier_1\big(\beta Suivant_1(A)\big) = \emptyset.$$

PREUVE : On utilise le fait que :

$$Premier_1(\alpha Suivant_1(A)) = \begin{cases} Premier_1(\alpha) \setminus \{\varepsilon\} \cup Suivant_1(A) & \text{si } \alpha \overset{*}{\underset{g}{\vdash}} \varepsilon \\ Premier_1(\alpha) & \text{sinon.} \end{cases}.$$

- $\Rightarrow$: Soit I l'intersection des deux ensembles. Si I est non vide, alors il existe $a \in I$ et alors :
  - Si α et β ne dérivent pas ε, alors a appartient à l'intersection des ensembles $Premier_1(\alpha)$ et $Premier_1(\beta)$. Donc $\alpha \overset{*}{\underset{g}{\vdash}} aw$ et $\beta \overset{*}{\underset{g}{\vdash}} aw'$.

    Et on peut exhiber deux dérivations à gauche : $S \overset{*}{\underset{g}{\vdash}} uA\theta \longmapsto u\alpha\theta \overset{*}{\underset{g}{\vdash}} uawt$ et $S \overset{*}{\underset{g}{\vdash}} uA\theta \longmapsto u\beta\theta \overset{*}{\underset{g}{\vdash}} uaw't'$ avec $1 : awt = 1 : aw't'$ ce qui est absurde car G est $LL(1)$.
  - autres cas similaires.
- $\Leftarrow$: C'est immédiat en utilisant la définition.

■

On en déduit une autre caractérisation des grammaires $LL(1)$:

PROPOSITION 3 *G est $LL(1)$ ssi pour toutes productions distinctes $A \to \alpha_1 \mid \alpha_2 \mid \ldots \mid \alpha_m$, on a :*

1. *Les ensembles $Premier_1(\alpha_i)$ sont deux à deux disjoints,*
2. *De $\alpha_i \overset{*}{\underset{g}{\vdash}} \varepsilon$ il résulte que $Premier_1(\alpha_j) \cap Suivant_1(\alpha_i)$ pour tout $j \neq i$.*

A ce stade, on a tout ce qu'il faut pour montrer que si G est $LL(k)$, alors $L(G)\$$ est reconnu par un automate à pile déterministe par pile vide.

PROPOSITION 4 *Soit G une grammaire algébrique $LL(1)$. Alors $L(G)\$$, où $\$$ est un caractère n'appartenant pas à $(\Sigma \cup N)$, est reconnu par APD par pile vide.*

PREUVE : On construit à l'aide de G l'automate

$$\mathcal{M} = (\Sigma \cup \{q_0, \$\}, \Sigma \cup \{\$\}, N, \delta, q_0, S)$$

avec δ définie par :
- $\delta(q_0, \$, \$) = (\$, \varepsilon)$ si $\varepsilon \in L(G)$.
- Pour $a \in \Sigma$, $\delta(q_0, a, S) = (a, \alpha)$ si $S \to \alpha \in P$ et $a \in Premier_1(Suivant_1(S))$.
- Pour $a \in \Sigma$, $\delta(a, \varepsilon, A) = (a, \alpha)$ si $A \to \alpha \in P$ et $a \in Premier_1(Suivant_1(A))$.
- Pour $a \in \Sigma$ et $b \in (\Sigma \cup \$)$, $\delta(a, b, a) = (b, \varepsilon)$.

Montrons alors que $L(G)\$ = L_P(G)$. Pour cela, prouvons :

$$A\beta \;\underset{g}{\overset{*}{\longmapsto}}\; a_1 \ldots a_p \text{ ssi } (a_1, a_2 \ldots a_p\$, A\beta) \;\underset{\mathcal{M}}{\overset{*}{\longmapsto}}\; (\$, \varepsilon, \varepsilon)$$

Chaque implication sera prouvée par récurrence :

- $\Rightarrow HR_n =$ "$\forall k \leq n, A\beta \;\underset{g}{\overset{n}{\longmapsto}}\; a_1 \ldots a_p \Rightarrow (a_1, a_2 \ldots a_p\$, A\beta \;\underset{\mathcal{M}}{\overset{*}{\longmapsto}}\; (\$, \varepsilon, \varepsilon)$" :
  - si $n = 1$ Supposons donc que $A\beta \;\underset{g}{\overset{1}{\longmapsto}}\; a_1 \ldots a_p$. Alors il existe $\alpha, A \to \alpha \in P$ et $A\beta \longmapsto \alpha\beta = a_1 \ldots a_p$. Trois cas sont alors possibles :
    - Si $\alpha = \varepsilon$ et $\beta = a_1 \ldots a_p$, alors :

      $$(a_1, a_2 \ldots a_p\$, A\beta \;\underset{\mathcal{M}}{\overset{1}{\longmapsto}}\; (a_1, a_2 \ldots a_p\$, \beta) = (a_1, a_2 \ldots a_p\$, a_1 \ldots a_p) \;\underset{\mathcal{M}}{\overset{*}{\longmapsto}}\; (\$, \varepsilon, \varepsilon),$$

      la deuxième dérivation étant bien définie car $a_1 \in Premier_1(\alpha Suivant_1(A))$.
    - Si $\alpha = a_1 \ldots a_l$ et $\beta = a_{l+1} \ldots a_p$, cela se traite de façon identique.
    - Si $\alpha = a_1 \ldots a_p$ et $\beta = \varepsilon$, idem.
  - soit n quelconque. On va découper la dérivation de taille n dans G en une première dérivation $A\beta \;\underset{G}{\overset{1}{\longmapsto}}\; \alpha\beta$ et les $n-1$ autres : $\alpha\beta \;\underset{G}{\overset{n-1}{\longmapsto}}\; a_1 \ldots a_p$. On a alors $A \to \beta \in P$, et en écrivant $\alpha = u\beta\alpha'$, où $u \in \Sigma^*$, ou bien $u = \varepsilon$ et on écrit $A\beta \;\underset{G}{\overset{1}{\longmapsto}}\; B\alpha'\beta' \;\underset{G}{\overset{n-1}{\longmapsto}}$, ou bien $u = a_1 \ldots a_l$ et on obtient $B\alpha'\beta \;\underset{G}{\overset{m}{\longmapsto}}\; a_{l+1}a_p$ en appliquant le lemme fondamental.

    En appliquant l'hypothèse de récurrence sur la dérivation plus petite (taille $n-1$ ou m), on obtient bien le résultat voulu.

- $\Leftarrow$ On va montrer que $\forall n, (a_1, a_2 \ldots a_p\$, A\beta) \;\underset{\mathcal{M}}{\overset{n}{\longmapsto}}\; (\$, \varepsilon, \varepsilon) \Rightarrow A\beta \;\underset{G}{\overset{*}{\longmapsto}}\; a_1 \ldots a_p$.
  - si $n = 2$ Si on a $(a_1, a_2 \ldots a_p\$, A\beta) \;\underset{\mathcal{M}}{\overset{1}{\longmapsto}}\; (a_1, a_2 \ldots a_p\$, \alpha\beta) \;\underset{\mathcal{M}}{\overset{1}{\longmapsto}}\; (\$\varepsilon, \varepsilon)$ c'est que $A \to \alpha \in P$ et $a_1 \in Premier_1(\alpha Suivant_1(A))$. Donc ou bien $\alpha = a_1$ et $\beta = \varepsilon$, ou bien $\alpha = \varepsilon$ et $\beta = a_1$, ce qui entraîne dans chaque cas $A\beta \;\underset{G}{\overset{*}{\longmapsto}}\; a_1$.
  - soit n quelconque. On découpe là aussi la dérivation en un premier pas, puis les $n-1$ autres à qui on peu appliquer l'HR. Cela se passe bien.

■

REMARQUE 5 C'est ce dernier automate que l'on a construit en cours de compilation, sauf que les états sont utilisés pour la construction de la table.

3 Grammaires $LR(k)$

EXEMPLE 2 *Considérons la grammaire définie par les productions suivantes :*

$$\begin{cases} S' \to Sc \\ S \to A \mid SA \\ A \to aSb \mid ab \end{cases}$$

DÉFINITION 5 (FSD)
On appelera **forme syntaxique droite** tout mot $\gamma \in (\Sigma \cup N)^*$ qui se dérive le plus à droite à partir de S. Dans l'exemple précédent, $Saabbc$ est une FSD.

3.1 Notion de manche

DÉFINITION 6
Soit γ une FSD. Un couple $(A \to \beta, i)$ où $A \to \beta \in P$ et $i \geq 0$ est dite une **poignée** pour γ si $S \underset{d}{\overset{*}{\longmapsto}} \alpha A w \longmapsto \alpha\beta w$ avec $\alpha\beta w = \gamma$ et $i = |\alpha\beta|$.

EXEMPLE 3

- *Soit G définie par les productions suivantes :*

$$\begin{cases} S \to aB \mid ab \\ B \to b \end{cases}$$

  Alors pour la FSD ab, $(S \to ab, 2)$ et $(B \to b, 2)$ sont des poignées. En effet, on a $S \longmapsto ab$ et $S \longmapsto aB \longmapsto ab$.

- *Soit G' définie par les productions suivantes :*

$$\begin{cases} S \to aA \mid Aa \\ A \to a \end{cases}$$

  Alors $(A \to a, 1)$ et $(A \to a, 2)$ sont des poignées pour $\gamma = aa$.

3.2 Définition de $LR(k)$

DÉFINITION 7
Soit G une grammaire algébrique réduite (*i.e.* sans symbole inutile) telle que S ne dérive pas S. Alors G est $LR(k)$ pour $k \geq 0$ si la condition suivante est vérifiée :

$$\begin{cases} (1) \quad S \underset{d}{\overset{*}{\longmapsto}} \alpha A w \longmapsto \alpha\beta w \\ (2) \quad S \underset{d}{\overset{*}{\longmapsto}} \alpha' A' w' \longmapsto \alpha'\beta' w' = \alpha\beta w'' \\ (3) \quad k : w = k : w' \end{cases} \Rightarrow \big(A \to \beta, |\alpha\beta|\big) = \big(A' \to \beta', |\alpha'\beta'|\big),$$

c'est-à-dire $A = A'$, $B = B'$, $|\alpha\beta| = |\alpha'\beta'|$, donc $w' = w''$ et $\alpha' = \alpha$, $\beta' = \beta$.

EXEMPLE 4

- *Soit G définie par les productions suivantes :*

$$\begin{cases} S \to aAc \\ A \to Abb \mid b \end{cases}$$

est $LR(0)$. En effet, les formes syntaxiques droites que l'on peut obtenir sont :
– *aAc de poignée $(S \to aAc, 3)$*
– *$aAb^{2n}c$ de poignée $(A \to Abb, 4)$*
– *$ab^{2n+1}c$ de poignée $(A \to b, 2)$.*

- *Attention, soit G' définie par les productions suivantes :*

$$\begin{cases} S \to aB \mid ab \\ A \to a \end{cases}$$

Alors cette grammaire engendre le même langage que G mais elle n'est pas $LR(0)$. Il faut considérer $S \overset{1}{\underset{d}{\longmapsto}} aAc \overset{1}{\underset{d}{\longmapsto}} abc$ et $S \longmapsto aAc \longmapsto abAbc \longmapsto abbbc$, on obtient les manches pour ab : $(A \to b, 2)$ et $(A \to b, 3)$.

3.3 Non-ambiguité de $LR(k)$

LEMME 1 *Soit G une grammaire algébrique réduite. Supposons que l'on ait les dérivations : $S \overset{*}{\underset{d}{\longmapsto}} \alpha A w \longmapsto \alpha\beta w$ et $S \overset{*}{\underset{d}{\longmapsto}} \alpha' A' w' \longmapsto \alpha'\beta' w' = \alpha\beta w$. Alors on a l'équivalence :*

$$\alpha A w = \alpha' A' w' \ \textit{ssi} \ \big(A \to \beta, |\alpha\beta|\big) = \big(A' \to \beta', |\alpha'\beta'|\big)$$

PREUVE : Facile et laissée au lecteur. ■

LEMME 2 *Soit G une grammaire algébrique réduite. Alors G est non ambiguë si et seulement si toute forme algébrique droite a exactement une poignée à l'exception de S qui n'en a aucune.*

PREUVE :

- $\Rightarrow$: Supposons G non ambiguë :
  – S n'a pas de poignée. Sinon, il existerait une poignée $(A \to \beta, i)$ telle que $S \overset{*}{\underset{d}{\longmapsto}} \alpha A w \longmapsto \alpha\beta w = S$, $i = |\alpha\beta|$ donc pour tout u mot sur Σ^*, on aurait : $S \overset{+}{\underset{d}{\longmapsto}} S \overset{*}{\underset{d}{\longmapsto}} u$ et $S \longmapsto u$, ce qui contredit l'hypothèse.
  – Toute forme syntaxique droite différente de S possède une poignée puisque G est réduite.
  – Montrons qu'il n'y en a qu'une par FSD. Supposons que $(A \to \beta, i)$ et $(A' \to \beta', i')$ soient des poignées pour γ une FSD. En appliquant la définition des poignées, on obtient l'existance des dérivations :

$$S \longmapsto \alpha A w \longmapsto \alpha\beta w \overset{*}{\underset{d}{\longmapsto}} u \text{ et } S \longmapsto \alpha' A' w' \longmapsto \alpha'\beta' w' = \gamma = \alpha\beta w \overset{*}{\underset{d}{\longmapsto}} u$$

, ce qui contredit le fait que G est supposée non ambiguë.

- $\Leftarrow$: Réciproquement, montrons que G n'est pas ambiguë. Supposons le contraire. Alors il existe un mot u qui admet deux dérivations droites distinctes :

$$\alpha_0 = S \longmapsto \alpha_1 \longmapsto \ldots \longmapsto \alpha_m = u \text{ et } \alpha'_0 = S \longmapsto \alpha'_1 \longmapsto \ldots \longmapsto \alpha_{m'} = u$$

  Alors il existe certainement i, $\alpha_{m-i} \neq \alpha'_{m'-i}$ (distinguer les cas $m = m'$ et $m \neq m'$). Soit alors j le plus petit entier vérifiant $\alpha_{m-j} \neq \alpha'_{m'-j}$. Alors on obtient $\alpha_{m-j+1} = \alpha'_{m'-j+1}$. Comme toute forme syntaxique a exactement une poignée, on déduit que $\alpha_{m-j} = \alpha_{m'_j}$. Ce qui est contradictoire avec j minimal.

■

On en déduit immédiatement le corollaire à l'aide du lemme précédent :

COROLLAIRE 2 *Si G est une grammaire $LL(k)$, alors elle n'est pas ambiguë.*

3.4 L'automate caractéristique d'une grammaire algébrique

DÉFINITION 8
On appelle **préfixe viable** de G tout mot $\mu \in (\Sigma \cup N)^*$ tel que

$$S \overset{*}{\underset{d}{\longmapsto}} \alpha A w \longmapsto \alpha\beta w, \text{ et } \mu \text{ préfixe de } \alpha\beta.$$

DÉFINITION 9
On appelle **item** de G tout objet $[A \rightarrow \beta_1 \bullet \beta_2]$ tel que $A \rightarrow \beta_1\beta_2 \in P$.

DÉFINITION 10
On appelle **item complet** un item $[A \rightarrow \bullet]$ ou $[A \rightarrow \beta\bullet]$.

DÉFINITION 11
Un item $[A \rightarrow \beta_1 \bullet \beta_2]$ est **valide** pour un préfixe viable μ de la grammaire si il existe une dérivation de la forme :

$$S \overset{*}{\underset{d}{\longmapsto}} \alpha A w \longmapsto \alpha\beta_1\beta_2 w,$$

telle que $\mu = \alpha\beta_1$.

On a alors la proposition suivante :

PROPOSITION 5 *Pour tout préfixe viable μ, il existe un item valide pour μ.*

PREUVE : facile mais à faire ! ■

DÉFINITION 12 (AUTOMATE CARACTÉRISTIQUE)
L'automate caractéristique $\mathcal{C}_G$ d'une grammaire G a pour états les items de G ($Q_\mathcal{C}$), son alphabet est $\Sigma \cup N$. Son état initial est $[S' \rightarrow \bullet S] = q_{0c}$ et les états d'acceptation sont $F_\mathcal{C}\{[A \rightarrow B\bullet] \mid A \rightarrow B \in P\}$. Enfin, la fonction de transition $\delta_\mathcal{C}$ est donnée par :

- $\delta([A \rightarrow \alpha \bullet X\beta], X) = \{[A \rightarrow \alpha X \bullet \beta]\}$ quand $A \rightarrow \alpha X\beta$ avec $X \in \Sigma \cup N$.

– $\delta([A \to \alpha \bullet B\beta], \varepsilon) = \{[B \to \bullet u], B \to u \in P\}$ pour $B \in N$.

PROPOSITION 6 *Soient $u \in (\sigma \cup N)^*$ et $q \in Q_C$. Alors $q \in \delta_C(q_{0c}, u)$ ssi μ est un préfixe viable de G et q est un item valide pour μ.*

PREUVE :

- $\Rightarrow$. Supposons que $q \in \delta_C(q_{0c}, \mu)$. Il existe donc un chemin de longueur n menant de q_{0c} à q dans l'automate. On va montrer par récurrence sur n la longueur du chemin la proposition.
  - si $n = 1$ Alors $q_{0c} = [S' \to \bullet S] \to_\mu [S \to \bullet\gamma]$ et nécessairement $\mu = eps$ et $S \to \gamma \in P$. Alors $S \underset{C}{\overset{1}{\longmapsto}} \gamma$ et $[S \to \bullet\gamma]$ est valide pour ε.
  - On suppose que c'est vrai pour les chemins de longueur inférieur strict à n. On va montrer que c'est vrai pour n. Si on prend un chemin de longueur n, il se segmente en un chemin de longueur $n-1$ et se termine sur une transition sur ε ou une transition à l'aide d'un élément de $\Sigma \cup N$:
    - Si le chemin est de la forme $q_{0c} \to_\mu q' \to_\varepsilon q$, alors par HR, q' est un item valide pour μ. On note $q' = [A \to \beta_1 \bullet \beta_2]$ et on a alors $S \underset{d}{\overset{*}{\longmapsto}} \alpha A w \longmapsto \alpha\beta_1\beta_2 w$ avec $\mu = \alpha\beta_1$. Comme dans l'automate la transition (q, q') est étiquetée par ε, on a appliqué la seconde règle de la définition pour construire cette transition. Par conséquent, β_2 commence par une variable B et donc s'écrit $\beta_2 = B\beta_2'$ et aussi on a $q = [B \to \bullet\gamma]$. Alors les dérivations $S \underset{d}{\overset{*}{\longmapsto}} \alpha A w \longmapsto \alpha\beta_1 B\beta_2' \longmapsto \alpha\beta_1\gamma\beta_2' w$ prouvent que q est valide pour μ.
- $\Leftarrow$. On suppose que q est un item valide pour le préfixe viable μ. On va montrer par récurrence la longueur de μ que $q \in \delta_C(q_{0c}, \mu)$
  - si $n = 0$: $|\mu| = 0$ signifie $\mu = \varepsilon$, et $q = [A \to \beta_1 \bullet \beta_2$ est un item valide pour ε, ce qui signifie qu'il existe une dérivation la plus à droite du type : $S \underset{d}{\overset{*}{\longmapsto}} \alpha A w \longmapsto \alpha\beta_1\beta_2 w$ tel que $\alpha\beta_1 = \varepsilon = \alpha = \beta_1$. Alors en réécrivant cette dernière dérivation : $S \underset{d}{\overset{*}{\longmapsto}} Aw$ et $A \to \beta_2 \in P$. Cela implique que $S \to X\alpha_1 w \underset{d}{\overset{*}{\longmapsto}} Xw' \longmapsto A\beta' w'$ (X dernière variable avant A). D'où $S \to X\alpha_1 w_1 \in P$ et $X \to a\beta' \in P$. Alors dans l'automate $[S' \to S] \to_\varepsilon [S \to \bullet X\alpha_1 w_1] \to_\varepsilon [X \to \bullet A\beta'] \to_\varepsilon [A \to \bullet\beta_2]$, donc $q_{oc} \to_\varepsilon q$ et c'est gagné.
  - On suppose que c'est vrai pour les mots de longueur inférieur strict à n. On va montrer que c'est vrai pour n. Si μ est un préfixe viable et $q = [A \to \beta_1 \bullet \beta_2]$ est un préfixe viable pour μ, alors par définition on a la dérivation : $S \underset{d}{\overset{*}{\longmapsto}} \alpha A w \longmapsto \alpha\beta_1\beta_2 w$ et $\mu = \alpha\beta_1$. On écrit alors $\mu = \mu' X$ avec $|\mu'| = |mu| - 1 = n - 1$.
    - Si $\beta_1 \neq \varepsilon$, alors $\beta_1 = \beta_1' X$ et par HR $[A \to \beta_1' \bullet X\beta_2]$ est un item valide pour μ'. donc on a :

$$q_{0c} \to_{\mu'} [A \to \beta_1' \bullet X\beta_2] \to_X [A \to \beta_1' X \bullet \beta_2] = [A \to \beta_1 \bullet \beta_2]$$

      et c'est bien ce qu'on veut.
    - Si $\beta_1 = \varepsilon$, alors $q = [A \to \bullet\beta_2]$, $\alpha = \mu' X$, $S \underset{d}{\overset{*}{\longmapsto}} \alpha A w \underset{\beta}{\overset{\alpha}{\longmapsto}}{}_2\, w =$

$\mu' X\beta_2 w$. Donc cela s'écrit encore :

$$S \overset{*}{\underset{d}{\longmapsto}} \theta' B w' \longmapsto \theta\nu X\rho w' \overset{*}{\underset{d}{\longmapsto}} \theta\nu X A w \longmapsto \theta\nu X\beta_2 w = \mu' X\beta_2 w$$

Alors ρ s'écrit forcément $C\rho'$ donc $\rho w' = C\rho' w' \overset{*}{\underset{d}{\longmapsto}} A\nu' w'$, avec $B \to \nu X\rho \in P$. Alors $[B \to \nu \bullet X\rho]$ est valide pour $\theta\nu = \mu'$:

$$q_{0c} \to_{\mu'} [B \to \nu \bullet X\rho] \to_\alpha [B \to \nu X \bullet \rho] \to_\varepsilon [C \to \bullet A\nu'] \to_\varepsilon [A \to \bullet\beta_2]$$

■

On déterminise ensuite l'automate $\mathcal{C}_G$ en $\mathcal{D}_G$. Alors on obtient une caractérisation des grammaires algébriques $LR(0)$:

PROPOSITION 7 *Une grammaire algébrique réduite (i.e. sans caractères inutiles) est $LR(0)$ ssi :*

1. *S ne dérive pas S à droite,*
2. *$\mathcal{D}_G$ ne contient pas d'état impropre (i.e. il ne contient pas plusieurs items complets distincts ni d'item complet accompagné d'items incomplets)*

PREUVE : La preuve, similaire à la précédente, est laissée au lecteur. ■

3.5 Automate reeconnaisseur de $LR(0)$

On va,dans cette section, décrire un automate à pile reconnaissant le langage engendré par une grammaire algébrique $LR(0)$.

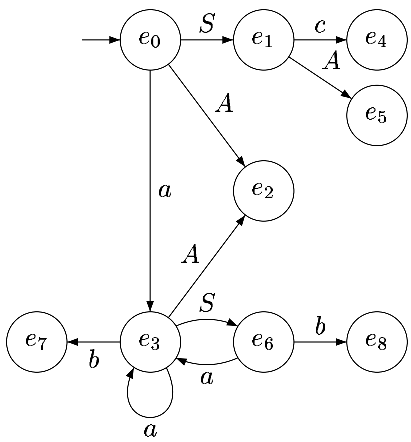
Sa fonction de transition sera notée $\tau : Q \times \gamma \times \Sigma \cup \{\varepsilon\} \to Q \times \Gamma^+$. De plus, dans le mot de pile, la lettre la plus à droite sera ici le sommet de pile. L'alphabet de pile est $Q_\mathcal{D} \cup N \cup \Sigma$, où $Q_\mathcal{D} = \{e_0, e_1, \dots e_k\}$ sont les états de l'automate déterminisé précédent (e_0 était l'état initial). e_0 sera le symbole de fin de pile. L'état initial de cet automate à pile sera q_0. Construisons maintenant la fonction de transition :

- $\tau(q_0, e_0, a) = (q_0, e_0 a \delta_\mathcal{D}(e_0, a))$.
- $\tau(q_0, e, a) = (q_0, e, a) = (q_0, e_0 a \delta_\mathcal{D}(e_0, a))$ si e ne contient pas d'item complet, et $a \in \Sigma$.
- Si e contient un item complet $[A \to \beta_1 \dots \beta_n \bullet]$, chaque $\beta_i \in \Sigma \cup N$, le mot de sommet de pile est alors de la forme $e'\beta_1 e_1 \beta_2 e_2 \dots \beta_n e_n$. Alors on va le manger, d'où les règles suivantes :
  - $\tau(q_0, e_n, \varepsilon) = (q_1, \varepsilon)$ (changement d'état "dépile")
  - $\tau(q_1, \beta_n, \varepsilon) = (q_2, \varepsilon)$
  - ...
  - $\tau(q_1, \beta_1, \varepsilon) = (q_{2n}, \varepsilon)$
  - $\tau(q_{2n}, e', \varepsilon) = \begin{cases} (q_0, e' A \delta_\mathcal{D}(e', A) & \text{si } A \neq S' \\ (q_0, e' S') & \text{si } A = S' \end{cases}$
- $\tau(q_0, S', \varepsilon) = (q^*, \varepsilon)$ (c'est fini)
- $\tau(q^*, e_0, \varepsilon) = (q^*, \varepsilon)$.

EXEMPLE 5 *Pour la grammaire :*

$$G = \begin{cases} S' \to Sc \\ S \to A \mid SA \\ A \to aSb \mid ab \end{cases},$$

on a obtenu l'automate suivant :



En notant les états :
- $e_0 = \{S' \to \bullet Sc, S \to \bullet SA, S \to \bullet A, A \to \bullet aSb, A \to \bullet ab\}$
- $e_1 = \{S' \to S \bullet c, S \to S \bullet A, A \to \bullet aSb, A \to \bullet ab\}$
- $e_2 = \{S \to A\bullet\}$
- $e_3 = \{A \to a \bullet Sb, A \to a \bullet b, A \to \bullet aSb, S \to \bullet A, S \to \bullet SA, A \to \bullet ab\}$
- $e_4 = \{S' \to Sc\bullet\}$
- $e_5 = \{S \to SA\bullet\}$
- $e_6 = \{S \to S \bullet A, A \to aS \bullet b, A \to \bullet ab, A \to \bullet aSb\}$
- $e_7 = \{A \to ab\bullet\}$
- $e_8 = \{A \to aSb\bullet\}$

Alors pour le mot abaabc, on a les mots de pile successifs : $e_0ae_3be_7$*, puis comme* e_7 *contient un item "Reduce" on obtient* e_0Ae_2*, puis* $e_0Se_1ae_3a_3ba_7$*, etc.*

3.6 Grammaires et langages $LR(1)$

On admettra ici que tout langage $LR(1)$ est algébrique déterministe.

THÉORÈME 1 *Tout langage algébrique déterministe est engendré par une grammaire* $LR(1)$.

Le cheminement de la preuve est le suivant :
- Soit L un langage algébrique déterministe. Alors il est reconnu par un APD par état final. Alors $L\$$ est reconnu par un APD par pile vide. $L\$$ a aussi la propriété du préfixe.
- Notion de grammaire algébrique stricte :

  DÉFINITION 13
  Une grammaire algébrique est dite déterministe **stricte** si il existe une partition Π de $N \cup \Sigma$ telle que :

1. $\Sigma \in \Pi$
2. pour tous A, A' de N, pour tous $\alpha, \beta, \beta' \in (\Sigma \cup N)^*$, si $A \to \alpha\beta \in P$, $\alpha \to \alpha\beta' \in P$) et $A \equiv A'[\Pi]$, alors
   - ou bien β et β' sont distincts de ε et $1 : \beta \equiv 1 : \beta'[\Pi]$
   - ou bien $\beta = \beta' = \varepsilon$ et $A = A'$.

- On montre le théorème suivant :

  THÉORÈME 2 *La famille des langages algébriques ayant la propriété su préfixe est celle des langages engendrés par les grammaires algébriques déterministes strictes.*

- Avec la définition suivante :
- Il existe un algorithme qui décide si une grammaire algébrique donnée est déterministe stricte ou non.
- Toute grammaire algébrique stricte est $LR(0)$. Donc $L\$$ est engendré par une grammaire $LR(0)$, et qui plus est par une grammaire $LR(0)$ sans ε-transition.
- On utilise enfin(!) le lemme :

  LEMME 3 *Soit G une grammaire $LR(0)$ de la forme $(N \cup \{\$\}, \Sigma \cup \{\$\}, P, S)$ où $P \subseteq N \times \big((N \cup \Sigma)^* \times (n \cup \Sigma)^*\$\big)$ et $L(G) \subseteq \Sigma^*\$$ et telle qu'il n'existe pas de dérivation pouvant mener de S à $S\$$ dans G. Alors la grammaire $G' = (N, \Sigma, P', S)$ avec $P' = P_1 \cup P_2$, $P_1 = \{A \to \beta \mid A \to \beta \in P, \beta \in (N \cup \Sigma)^*\}$ et $P_2 = \{A \to \beta \mid A \to \beta\$ \in P\}$ est $LR(1)$ et $L(G')\$ = L(G)$.*

THÉORÈME 3 *Tout langage engendré par une grammaire $LR(k)$ est algébrique déterministe.*

On va maintenant construire un analyseur $LR(k)$, c'est-à-dire un automate déterministe pour une grammaire $LR(k)$.

3.7 Analyseur $LR(k)$

DÉFINITION 14 ($LR(k)$-ITEM)
Un **$LR(k)$-item** pour une grammaire algébrique est $[A \to \beta_1 \bullet \beta_2, u]$, où $u \in Suivant_k(A)$ et $A \to \beta_1\beta_2 \in P$.

REMARQUE 6 En cours de compilation, on a noté cela $[A \to \beta_1 \bullet \beta_2]\{u\}$.

DÉFINITION 15
L'item $[A \to \beta_1 \bullet \beta_2, u]$ est dit **valide** pour un préfixe viable μ si

$$S \xmapsto[d]{*} \alpha A w \longmapsto \alpha\beta_1\beta_2 w \text{ avec } \mu = \alpha\beta_1 \text{ et } u = k : w$$

Il existe un algorithme qui permet de calculer les ensembles d'items valides pour les préfixes viables de la grammaire.

On note $\mathcal{S}_G$ l'ensembles des ensembles de $LR(k)$-items valides pour les préfixes viables de G.

DÉFINITION 16
Un ensemble $\mathcal{S}$ d'ensembles de $LR(k)$-items valides est dit **consistant** si pour tout ensemble I de $\mathcal{S}$, $\forall A, A' \in N$, $\forall \beta, \beta_1, \beta_2 \in (\Sigma \cup N)^*$ vérifiant $1 : \beta_2 \notin N$ et pour u, v mots de Σ^* de taille $\leq k$, $u \in Premier_k(\beta_2 v)$, on a :

$$\Big([A \to \beta\bullet, u] \in I, [A \to \beta_1 \bullet \beta_2] \in P\Big) \Rightarrow \Big([A \to \beta\bullet, u] = [A' \to \beta_1 \bullet \beta_2, v]\Big)$$

On a alors :

THÉORÈME 4 *G est $LR(k)$ ssi $\mathcal{S}_{k,G}$ est consistant.*

Tous ces résultats nous permettent de construire l'analyseur $LR(k)$ cherché :

On commence par définir deux "fonctions" f et g dites d'action et de succession. Soit T un ensemble de $LR(k)$-items de G et $u \in \Sigma^{\leq k}$. On définit :

1. $f(T, u)$: **décaler** si il existe $\beta_1 \in (N \cup \Sigma)^*$, $\beta_1 \in \Sigma(\mathbb{N} \cup \Sigma)^*$, $v \in \Sigma^{\leq k}$ et $A \in N$ vérifiant :
$$[A \to \beta_1 \bullet \beta_2] \in T \text{ avec } u \in Premier_k(\beta_2 v)$$
2. $f(T, u)$: **réduire** la production $A \to \beta$ si $[A \to \beta\bullet, u] \in T$.
3. $f(T, u)$: **erreur** partout ailleurs
4. si on note $T = V_k(\gamma)$ (*i.e.* γ préfixe viable et $V_k(\gamma)$ valide pour γ) et $X \in (N \cup \Sigma)$:
   - $g(V_k(\gamma), X) = V_k(\gamma X)$ si $V_k(\gamma) \neq \emptyset$
   - $g(V_k(\gamma), X)$: **erreur** sinon.

PROPOSITION 8 *Si l'ensemble d'ensemble d'items T est consistant, alors f et g sont vraiment des fonctions.*

On numérote les productions de la grammaire. Soit $(\gamma T, z, \rho)$ une configuration, où γT désigne le contenu de la pile (T est au dessus), z l'entrée, ρ la sortie. Quelle est la configuration suivante ?

1. Si $f(T, k : z)$=décaler alors
   - si $z = \varepsilon$ retourner ERREUR
   - si $z \neq \varepsilon$, notons $z = bz', b \in \Sigma$:
     - si $g(T, b)$=erreur alors retourner ERREUR
     - sinon $(\gamma, T, bz', \rho) \longmapsto (\gamma T b g(T), z', \rho)$
2. Si $f(T, k : z)$=réduire(i) alors on veut dépiler $2\,|\beta|$ symboles si la production numéro i est $A \to \beta$. Donc si $\gamma T = \gamma T \gamma''$ avec $|\gamma''| = 2\,|\beta|$, on a deux cas :
   - si $T' = T_0$ (fonds de pile), $S = A$, $z = \varepsilon$, alors $(\gamma T, z, rho) \longmapsto (T_0, \varepsilon, \rho i)$ ACCEPTE
   - si $T \neq T_0$ alors :
     - si $g(T', A)$=erreur alors retourner ERREUR
     - sinon la nouvelle configuration est $(\gamma T' A g(T', A), z, \rho i)$.
3. si $f(T, k : z)$=erreur alors retourner ERREUR