Théorie des langages formels : Les langages algébriques.

Wadoud Bousdira[1]

[1]LIFO, University of Orléans
Orléans, France

Analyse syntaxique LL(k).

1. Définition
2. Ensembles `Premier` et `Suivant`
3. Récursivité gauche
4. Factorisation
5. Grammaire LL(1)
6. Table d'analyse LL(1)

Analyse syntaxique LL(k).

Dérivation gauche :

$$S \overset{*}{\Rightarrow} \underbrace{a_1 \dots a_j}_{\in \Sigma} \underbrace{A}_{\in N} \alpha$$

le mot à reconnaître est m,

- il faut que m commence par $a_1 \dots a_j$
- on doit dériver A et la grammaire comporte les règles $A \to \gamma_1 \mid \gamma_2 \mid \dots \mid \gamma_n$

Problème : rendre déterministe le choix entre $\gamma_1, \gamma_2, \dots$ et γ_n.

Grammaire LL(k).

Grammaire LL(k)

où k est un entier fixé. Si on sait quelle règle choisir à condition de regarder les k caractères qui suivent (k caractères de prévision).

- On étudiera les grammaires LL(1) : 1 caractère de prévision permet de choisir quelle règle appliquer entre toutes les règles possibles.
- On tente de construire l'arbre de dérivation de façon descendante en ayant un parcours déterministe du mot de gauche à droite à condition de pouvoir regarder 1 caractère à l'avance.

Grammaire LL(1).

Exemple 1 :

$$\begin{cases} S & \rightarrow & aAS \mid b \\ A & \rightarrow & a \mid bSA \end{cases}$$

mot : a^3b^2ab

$S \Rightarrow_g aAS \Rightarrow_g aaS \Rightarrow_g aaaAS \Rightarrow_g a^3bSAS \Rightarrow_g a^3bbAS \Rightarrow_g a^3b^2aS \Rightarrow_g a^3b^2ab$

Dans l'exemple de dérivation, on a besoin de connaître le prochain caractère et seulement celui-ci $\Rightarrow$ la grammaire est LL(1).

Grammaire LL(1).

Exemple 2 :

$$\mathcal{L} = \{a^n 1 b^n,\ n \geq 0\} \cup \{a^n 2 b^{2n},\ n \geq 0\},\ P(\mathcal{L}) = \left\{ \begin{array}{lcl} S & \rightarrow & A \mid B \\ A & \rightarrow & aAb \mid 1 \\ B & \rightarrow & aBbb \mid 2 \end{array} \right.$$

$\mathcal{L}$ n'est pas LL(k), $\forall k$. Il suffit d'avoir $n > k$ de a pour ne pas pouvoir choisir entre $S \rightarrow A$ et $S \rightarrow B$.

Grammaire LL(1).

Premier et Suivant pour une grammaire LL(1)

soit un mot $\alpha \in (\Sigma \cup N)^*$

- $\texttt{Premier}(\alpha) = \{x \in \Sigma \ / \ \alpha \overset{*}{\Rightarrow} x\beta\} \cup \{\epsilon, \text{ si } \alpha \overset{*}{\Rightarrow} \epsilon\}$

soit $A \in N$,

- $\texttt{Suivant}(A) = \{x \in \Sigma \ / \ S \overset{*}{\Rightarrow} \alpha A x \beta, \ \alpha \in \Sigma^*\} \cup \{\$, \text{ si } S \overset{*}{\Rightarrow} \alpha A, \ \alpha \in \Sigma^*\}$

  ou encore

  $$\texttt{Suivant}(A) = \bigcup_{\gamma \neq \epsilon} \texttt{Premier}(\gamma) \ / \ S \overset{*}{\Rightarrow} \alpha A \gamma \cup \{\$, \text{ si } S \overset{*}{\Rightarrow} \alpha A, \ \alpha \in \Sigma^*\}$$

Grammaire LL(1).

Quel rôle ?

pour étudier les productions possibles à partir du non-terminal A :

$$A \to \gamma_1 \mid \gamma_2 \mid \ldots \mid \gamma_n$$

Ça marchera si on sait distinguer les γ_i en fonction des 1^{ers} symboles qu'ils produisent ... $\texttt{Premier}(\gamma_i)$ d'où la notion de $\texttt{Premier}$.

Exemple 1 : $S \to aAS \mid b, \; A \to a \mid bSA$

- $\texttt{Premier}(aAS) = \{a\}$, $\texttt{Premier}(b) = \{b\}$ et $\{a\} \cap \{b\} = \emptyset$
  $\texttt{Premier}(a) = \{a\}$, $\texttt{Premier}(bSA) = \{b\}$ et $\{a\} \cap \{b\} = \emptyset$.

Grammaire LL(1).

Exemple 2 : $S \to A \mid B,\ A \to aAb \mid 1,\ B \to aBbb \mid 2$

$\texttt{Premier}(A) = \{a,\ 1\}$, $\texttt{Premier}(B) = \{a,\ 2\}$ et $\{a,\ 1\} \cap \{a,\ 2\} \neq \emptyset$!

Exemple 3 : $E \to E + T \mid T,\ T \to T * F \mid F,\ F \to 0 \mid 1 \mid x \mid (E)$

$\texttt{Premier}(F) = \{0,\ 1,\ x,\ (\}$
$\texttt{Premier}(F) = \texttt{Premier}(T) = \texttt{Premier}(E) = \{0,\ 1,\ x,\ (\}$.

Grammaire LL(1).

Mais la notion de Premier ne suffit pas !

Exemple : $S \rightarrow aSa \mid \epsilon$
$\texttt{Premier}(aSa) = \{a\}$, $\texttt{Premier}(\epsilon) = \{\epsilon\}$
Les ensembles sont disjoints et pourtant on ne sait pas choisir
lors de la reconnaissance de *aaaa* par ex :
$S \Rightarrow aSa \Rightarrow aaSaa$

- Si on choisit $S \rightarrow aSa$, on produit des 'a' de trop !
- si on choisit $S \rightarrow \epsilon$, on fait aussi apparaître un a qui était derrière S d'où la notion de Suivant.
  Quel est le problème ? a commence l'alternative aSa de S et $a \in \texttt{Suivant}(S)$!

Si $S \rightarrow aSb \mid \epsilon$, et $m = aabb$,
$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aabb$: dérivation déterministe.

Ensembles Premier et Suivant

Algorithmes de calcul

Pour calculer Premier :

$\forall\ X \in (\Sigma \cup N)$, appliquer les règles suivantes jusqu'à ce qu'aucun terminal ni ϵ ne puisse être ajouté aux ensembles Premier

1. si $X \in \Sigma$, $\texttt{Premier}(X) = \{X\}$
2. si $X \to \epsilon \in \mathcal{P}$, ajouter ϵ à $\texttt{Premier}(X)$
3. si $X \in N$, et $X \to Y_1 Y_2 \dots Y_k \in \mathcal{P}$,
   - ajouter $\texttt{Premier}(Y_1)$ à $\texttt{Premier}(X)$,
   - si $\epsilon \in \texttt{Premier}(Y_1)$, ajouter $\texttt{Premier}(Y_2)$ à $\texttt{Premier}(X)$,
   - ...
   - si $\epsilon \in \texttt{Premier}(Y_j)$, $\forall j = 1, \ \dots, \ k$ (i.e. $Y_1 \dots Y_k \stackrel{*}{\Rightarrow} \epsilon$) alors ajouter ϵ à $\texttt{Premier}(X)$.

Ensembles `Premier` et `Suivant`

Conséquence :

On étend la notion de $\texttt{Premier}(X)$, $X \in (\Sigma \cup N)$ à celle de $\texttt{Premier}(\alpha)$, $\alpha \in (\Sigma \cup N)^*$

$\forall$ la chaîne $X_1 X_2 \dots X_n \in (\Sigma \cup N)^*$, $\texttt{Premier}(X_1 X_2 \dots X_n)$ se calcule

- en ajoutant tous les symboles de $\texttt{Premier}(X_1) \neq \epsilon$.
- Si $\epsilon \in \texttt{Premier}(X_1)$, ajouter aussi les symboles de $\texttt{Premier}(X_2) \neq \epsilon$.
- Si $\epsilon \in \texttt{Premier}(X_1)$ et à $\texttt{Premier}(X_2)$, ajouter aussi les symboles de $\texttt{Premier}(X_3) \neq \epsilon$.
- etc ...
- Finalement, si $\forall i$, $\epsilon \in \texttt{Premier}(X_i)$, ajouter ϵ à $\texttt{Premier}(X_1 X_2 \dots X_n)$.

Ensembles Premier et Suivant

Pour calculer $\texttt{Suivant}(A)$, $\forall A \in N$:

Appliquer les règles suivantes jusqu'à ce qu'aucun terminal ne puisse être ajouté aux ensembles Suivant :

1. mettre \$ dans $\texttt{Suivant}(S)$, où S est l'axiome et \$ est un marqueur droit indiquant la fin du texte en entrée.
   $\$ \notin \Sigma$ (par hypothèse),
2. si $A \rightarrow \alpha B \beta \in \mathcal{P}$, ajouter $\texttt{Premier}(\beta) \backslash \{\epsilon\}$ à $\texttt{Suivant}(B)$.
3. si $A \rightarrow \alpha B \in \mathcal{P}$
   ou $A \rightarrow \alpha B \beta \in \mathcal{P}$ tq $\texttt{Premier}(\beta)$ contient ϵ (i.e. $\beta \stackrel{*}{\Rightarrow} \epsilon$),
   ajouter $\texttt{Suivant}(A)$ à $\texttt{Suivant}(B)$.

Ensembles Premier et Suivant.

Exemple 1 : $E \to E + T \mid T,\ T \to T * F \mid F,\ F \to x \mid 0 \mid 1 \mid (E)$

$\texttt{Premier}(x) = \{x\}$, $\texttt{Premier}(0) = \{0\}$ $\texttt{Premier}(1) = \{1\}$
$\texttt{Premier}((E)) = \{(\}$

$\texttt{Premier}(F) = \{x,\ 0,\ 1,\ (\}$
$\texttt{Premier}(F) \subseteq \texttt{Premier}(T) \Rightarrow \{x,\ 0,\ 1,\ (\} \subseteq \texttt{Premier}(T)$
$\texttt{Premier}(T) \subseteq \texttt{Premier}(E) \Rightarrow \{x,\ 0,\ 1,\ (\} \subseteq \texttt{Premier}(E)$

$\texttt{Suivant}(E) = \{\$,\ +,\ )\}$
$\texttt{Suivant}(E) \subseteq \texttt{Suivant}(T) \Rightarrow \{\$,\ +,\ )\} \subseteq \texttt{Suivant}(T)$
$\texttt{Suivant}(T) \subseteq \texttt{Suivant}(F) \Rightarrow \{\$,\ +,\ )\} \subseteq \texttt{Suivant}(F)$

Ensembles Premier et Suivant.

Exemple 2 : $S \rightarrow AaAb \mid BbBa, \ A \rightarrow \epsilon, \ B \rightarrow \epsilon$

$\texttt{Premier}(AaAb) = \{a\}$, $\texttt{Premier}(BbBa) = \{b\}$,
$\texttt{Premier}(S) = \texttt{Premier}(AaAb) \cup \texttt{Premier}(BbBa) = \{a, b\}$

$\texttt{Suivant}(S) = \{\$\}$
$\texttt{Suivant}(A) = \{a, \ b\}$ $\qquad$ $\texttt{Suivant}(B) = \{b, \ a\}$

Exemple 3 : $S \rightarrow aSbS \mid bSaS \mid \epsilon$
$\texttt{Premier}(S) = \texttt{Premier}(aSbS) \cup \texttt{Premier}(bSaS) \cup$
$\texttt{Premier}(\epsilon) = \{a, \ b, \ \epsilon\}$

$\texttt{Suivant}(S) = \{\$, \ a, \ b\}$.

Remarque

ϵ n'est jamais dans un ensemble Suivant.

Récursivité gauche.

Grammaire récursive à gauche

$\mathcal{G}$ est récursive à gauche si $\exists\, A \in N$ tq on peut construire une dérivation de la forme $A \overset{+}{\Rightarrow} A\alpha$.

La récursivité gauche simple se caractérise par :

$$\exists\ A \rightarrow A\,\alpha \in \mathcal{P}$$

Récursivité gauche.

Pb de la récursivité gauche :

les méthodes d'analyse descendante ne peuvent pas fonctionner.

- Si $\exists\, A \to A\,\alpha \mid \beta \in \mathcal{P}$ et si on doit dériver A en une de ses alternatives, $A\alpha$ peut être choisie de façon répétitive sans aucun moyen d'arrêt.
- De $+$, dans ce cas, 1 caractère de prévision ne suffit pas à choisir l'alternative à appliquer.

On prouvera $+$ tard qu'une grammaire récursive à gauche ne peut pas être LL(1).

Supprimer la récursivité gauche.

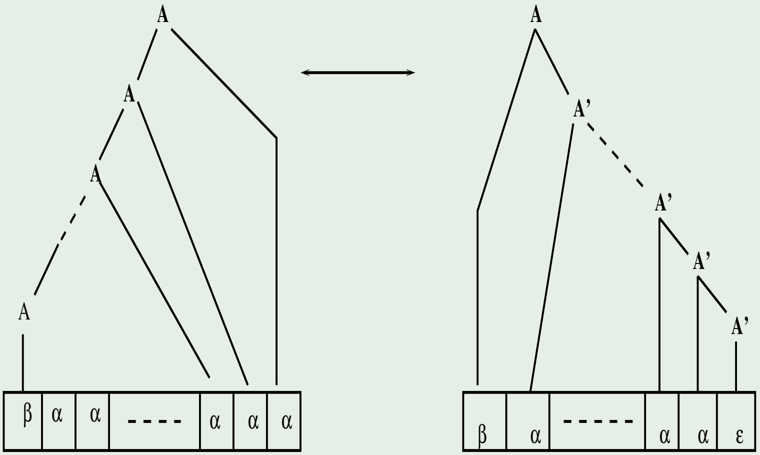
Comment supprimer la récursivité gauche ?

- simple :
  $A \rightarrow A\alpha \mid \beta$ où $\alpha, \ \beta \in (\Sigma \cup N)^*$ ne commencent pas par A.
  On remplace $A \rightarrow A\alpha \mid \beta$ par :

$$\begin{cases} A & \rightarrow & \beta A' \\ A' & \rightarrow & \alpha A' \mid \epsilon \end{cases}$$

- générale : se ramener à une récursivité simple (par expansion de non-terminaux) puis la supprimer.

Supprimer la récursivité gauche.



Supprimer la récursivité gauche.

Exemple 1 :

$$E \rightarrow E + T \mid T \Longleftrightarrow \begin{cases} E & \rightarrow \quad T \ E' \\ E' & \rightarrow \quad +T \ E' \mid \epsilon \end{cases}$$

Exemple 2 :

$$E \rightarrow id \mid id(E) \mid E + id \Longleftrightarrow \begin{cases} E & \rightarrow \quad id\, E' \mid id\,(E)\, E' \\ E' & \rightarrow \quad +id\, E' \mid \epsilon \end{cases}$$

Factorisation.

Grammaire non factorisée

Elle contient au moins 2 productions de la forme

$$A \to \alpha\beta_1 \mid \alpha\beta_2 \text{ tq } \alpha \in (\Sigma \cup N)^+.$$

Problème : le choix de l'alternative à appliquer devient impossible avec 1 caractère de prévision.

Factorisation.

On obtient une grammaire équivalente factorisée avec la méthode suivante :

- pour chaque $A \in N$, trouver le $+$ long préfixe α commun à 2 alternatives ou $+$. Si $\alpha \neq \epsilon$:

$$A \rightarrow \alpha\beta_1 \mid \alpha\beta_2 \mid \ldots \mid \alpha\beta_n \mid \gamma \Leftrightarrow \begin{cases} A \rightarrow \alpha A' \mid \gamma \\ A' \rightarrow \beta_1 \mid \beta_2 \mid \ldots \mid \beta_n \end{cases}$$

  où γ représente toutes les alternatives qui ne commencent pas par α.
- Appliquer cette transformation de façon répétitive jusqu'à plus de préfixe commun pour des alternatives.

Factorisation.

Exemple

$$E \to id\, E \mid (E,\, E) \mid (E) \mid id$$

équivalente à

$$\left\{ \begin{array}{lcl} E & \to & id\, A \mid (E\, B \\ A & \to & E \mid \epsilon \\ B & \to & ,\, E) \mid) \end{array} \right.$$

Grammaires LL(1).

Proposition :

- une grammaire ambiguë n'est pas LL(1).
- une grammaire récursive à gauche n'est pas LL(1).
- une grammaire non factorisée n'est pas LL(1).

Grammaires LL(1).

Proposition :

une grammaire est LL(1) ssi $\forall A \to \alpha \mid \beta \in \mathcal{P}$ tq $\alpha \neq \beta$,

1. $\texttt{Premier}(\alpha) \cap \texttt{Premier}(\beta) = \emptyset$
2. si $\beta \overset{*}{\Rightarrow} \epsilon$, $\texttt{Premier}(\alpha) \cap \texttt{Suivant}(A) = \emptyset$

Conséquence : si $\beta \overset{*}{\Rightarrow} \epsilon$, alors $\alpha \overset{*}{\not\Rightarrow} \epsilon$.

(1) et (2) permettent de tester si une grammaire est LL(1).

Grammaires LL(1).

Exemple 1 :

$E \to E + T \mid T,\ T \to T * F \mid F,\ F \to x \mid 0 \mid 1 \mid (E)$

Récursive à gauche. On dérécursive :

$$\begin{cases} (1) & E \to T\ E' \\ (2) & E' \to +T\ E' \mid \epsilon \\ (3) & T \to F\ T' \\ (4) & T' \to *F\ T' \mid \epsilon \\ (5) & F \to x \mid 0 \mid 1 \mid (E) \end{cases}$$

Exemple 1

(2) : $\texttt{Premier}(+T\ E') \cap \texttt{Premier}(\epsilon) = \{+\} \cap \{\epsilon\} = \emptyset$
$\texttt{Premier}(+T\ E') \cap \texttt{Suivant}(E') = \{+\} \cap \{\$,\ )\} = \emptyset$

(4) : $\texttt{Premier}(*F\ T') \cap \texttt{Premier}(\epsilon) = \{*\} \cap \{\epsilon\} = \emptyset$
$\texttt{Premier}(*F\ T') \cap \texttt{Suivant}(T') = \{*\} \cap \{\$,\ ),\ +\} = \emptyset$

(5) : $\texttt{Premier}(x) \cap \texttt{Premier}(0) = \emptyset$
$\texttt{Premier}(x) \cap \texttt{Premier}(1) = \emptyset$
$\texttt{Premier}(x) \cap \texttt{Premier}((E)) = \emptyset$
$\texttt{Premier}(0) \cap \texttt{Premier}(1) = \emptyset$
$\texttt{Premier}(0) \cap \texttt{Premier}((E)) = \emptyset$
$\texttt{Premier}(1) \cap \texttt{Premier}((E)) = \emptyset$

Donc grammaire LL(1).

Grammaires LL(1).

Exemple 2 : $S \to AaAb \mid BbBa,\ A \to \epsilon,\ B \to \epsilon$
$\texttt{Premier}(AaAb) \cap \texttt{Premier}(BbBa) = \{a\} \cap \{b\} = \emptyset$
Donc grammaire LL(1).

Exemple 3 : $S \to aSbS \mid bSaS \mid \epsilon$

1. $\texttt{Premier}(aSbS) \cap \text{Premier}(bSaS) = \{a\} \cap \{b\} = \emptyset$
2. $\texttt{Premier}(aSbS) \cap \text{Premier}(\epsilon) = \{a\} \cap \{\epsilon\} = \emptyset$
3. $\texttt{Premier}(bSaS) \cap \texttt{Premier}(\epsilon) = \{b\} \cap \{\epsilon\} = \emptyset$
4. $\texttt{Suivant}(S) \cap \texttt{Premier}(aSbS) = \{\$,\ b,\ a\}\ \cap \{a\} \neq \emptyset$!

Donc grammaire non LL(1).

Grammaires LL(1).

Que faire si la grammaire n'est pas LL(1) ?

- dérécursiver les productions récursives à gauche ;
- expanser : remplacer une occurrence d'un non-terminal par ses productions ;
- factoriser ;
- itérer si nécessaire.

Remarque : Transformer la grammaire le − possible : les transformations incorrectes modifient la sémantique d'une grammaire.

Grammaires LL(1).

Exemple 4 :

$I \to LA \mid LB,\ A \to \epsilon,\ B \to \epsilon \mid BL \mid BC$
$L \to a \mid b \mid \ldots \mid z,\ C \to 0 \mid 1 \mid \ldots \mid 9$

- on expanse A dans les productions de I : $I \to L \mid LB$
- on factorise L : $I \to L\ I',\ I' \to \epsilon \mid B$
- on dérécursive les productions de B :
  $B \to B',\ B' \to LB' \mid CB' \mid \epsilon \Rightarrow B \to LB \mid CB \mid \epsilon$
  On arrive à :
  $$\begin{cases} I \to L\ I' \\ I' \to \epsilon \mid B \\ B \to LB \mid CB \mid \epsilon \\ L \to a \mid b \mid \ldots \mid z \\ C \to 0 \mid 1 \mid \ldots \mid 9 \end{cases}$$

Exemple 4

On vérifie LL(1)

1. $\texttt{Premier}(B) \cap \texttt{Premier}(\epsilon) = (\texttt{Premier}(L) \cup \texttt{Premier}(C) \cup \{\epsilon\}) \cap \{\epsilon\} \neq \emptyset$!

On expanse B dans les productions de I', i.e.
$I \rightarrow \epsilon \mid B \Longleftrightarrow I' \rightarrow \epsilon \mid LB \mid CB$
On arrive à :

$$\begin{cases} I \rightarrow L\ I' \\ I' \rightarrow \epsilon \mid LB \mid CB \\ L \rightarrow a \mid b \mid \ \ldots \ \mid z \\ C \rightarrow 0 \mid 1 \mid \ \ldots \ \mid 9 \\ B \rightarrow LB \mid CB \mid \epsilon \end{cases}$$

Exemple 4

I' et B peuvent être confondus.

$$\left\{\begin{array}{l} I \to L\ I' \\ I' \to \epsilon \mid LB \mid CB \\ L \to a \mid b \mid \ \dots\ \mid z \\ C \to 0 \mid 1 \mid \ \dots\ \mid 9 \\ B \to LB \mid CB \mid \epsilon \end{array}\right. \iff \left\{\begin{array}{l} I \to L\ B \\ B \to \epsilon \mid LB \mid CB \\ L \to a \mid b \mid \ \dots\ \mid z \\ C \to 0 \mid 1 \mid \ \dots\ \mid 9 \end{array}\right.$$

Exemple 4

On vérifie LL(1)

1. $\texttt{Premier}(\epsilon) \cap \texttt{Premier}(LB) = \{\epsilon\} \cap \texttt{Premier}(L) = \emptyset$
2. $\texttt{Premier}(\epsilon) \cap \texttt{Premier}(CB) = \{\epsilon\} \cap \texttt{Premier}(C) = \emptyset$
3. $\texttt{Premier}(LB) \cap \texttt{Premier}(CB) = \emptyset$
4. $\texttt{Suivant}(B) \cap \texttt{Premier}(LB) = \{\$\} \ \cap \{a, \ldots, z\} = \emptyset$
5. $\texttt{Suivant}(B) \cap \texttt{Premier}(CB) = \{\$\} \ \cap \{0, \ldots, 9\} = \emptyset$
6. $\texttt{Premier}(a) \cap \texttt{Premier}(b) = \emptyset$, $\texttt{Premier}(a) \cap \texttt{Premier}(c) = \emptyset \ldots$
   $\texttt{Premier}(y) \cap \texttt{Premier}(z) = \emptyset$
7. $\texttt{Premier}(0) \cap \texttt{Premier}(1) = \emptyset$, $\texttt{Premier}(0) \cap \texttt{Premier}(2) = \emptyset \ldots$
   $\texttt{Premier}(8) \cap \texttt{Premier}(9) = \emptyset$

Donc grammaire LL(1).

Table d'analyse LL(1).

- L'analyseur est un automate à pile déterministe sans état.
  L'arrêt est sur pile vide.
- Configuration initiale : (pile = $S\$$, mot à analyser) où S est le sommet de pile et \$ un symbole spécial.
- On construit une table $\mathcal{M}$ qui indique ce qu'il faut faire en fonction du sommet de pile et du caractère de prévision dans le mot à analyser. $\mathcal{M}$ est de la forme

	Symbole $\in \Sigma \cup \{\$\}$
Symbole $\in N$	

Table d'analyse LL(1).

4 actions possibles :

1. remplacer le sommet de pile par autre chose sans consommer de symbole dans le mot (correspond à une action de dérivation gauche).
2. dépiler et avancer dans le mot d'un caractère si le sommet de pile et le caractère de prévision sont identiques.
3. échec : entrée de $\mathcal{M}$ vide.
4. succès : fin de l'analyse.

Table d'analyse LL(1).

Algorithme de construction de la table

Donnée : une grammaire $\mathcal{G}$ LL(1)

Résultat : une table d'analyse $\mathcal{M}$ de $\mathcal{G}$.

1. pour chaque production $A \rightarrow \alpha$ de $\mathcal{G}$, procéder aux étapes suivantes
   - $\forall\, a \in \texttt{Premier}(\alpha)$, ajouter α à $\mathcal{M}[A,\ a]$
   - si $\epsilon \in \texttt{Premier}(\alpha)$, ajouter α à $\mathcal{M}[A,\ b]$, $\forall\, b \in \texttt{Suivant}(A)$.
2. Faire de chaque entrée non définie de $\mathcal{M}$ une erreur.

Table d'analyse LL(1).

Exemple 1 :
$E \to TE', \ E' \to +TE' \mid \epsilon, \ T \to FT', \ T' \to *FT' \mid \epsilon, \ F \to x \mid 0 \mid 1 \mid (E)$

	x	0	1	$+$	$*$	(	)	\$
E	TE'	TE'	TE'			TE'		
E'				$+TE'$			ϵ	ϵ
T	FT'	FT'	FT'			FT'		
T'				ϵ	$*FT'$		ϵ	ϵ
F	x	0	1			(E)		

Exemple 1 (suite) :

Pile	chaîne d'entrée	Action
$E\$$	$x * (x+1)\$$	$E \to TE'$
$TE'\$$	$x * (x+1)\$$	$T \to FT'$
$FT'E'\$$	$x * (x+1)\$$	$F \to x$
$xT'E'\$$	$x * (x+1)\$$	dépiler et avancer
$T'E'\$$	$*(x+1)\$$	$T' \to *FT'$
$*FT'E'\$$	$*(x+1)\$$	dépiler et avancer
$FT'E'\$$	$(x+1)\$$	$F \to (E)$
$(E)T'E'\$$	$(x+1)\$$	dépiler et avancer

Exemple 1 (suite) :

Pile	chaîne d'entrée	Action
$E)T'E'\$$	$x+1)\$$	$E \to TE'$
$TE')T'E'\$$	$x+1)\$$	$T \to FT'$
$FT'E')T'E'\$$	$x+1)\$$	$F \to x$
$xT'E')T'E'\$$	$x+1)\$$	dépiler et avancer
$T'E')T'E'\$$	$+1)\$$	$T' \to \epsilon$
$E')T'E'\$$	$+1)\$$	$E' \to +TE'$
$+TE')T'E'\$$	$+1)\$$	dépiler et avancer
$TE')T'E'\$$	$1)\$$	$T \to FT'$

Exemple 1 (suite) :

Pile	chaîne d'entrée	Action
$FT'E')T'E'\$$	$1)\$$	$F \rightarrow 1$
$1T'E')T'E'\$$	$1)\$$	dépiler et avancer
$T'E')T'E'\$$	$)\$$	$T' \rightarrow \epsilon$
$E')T'E'\$$	$)\$$	$E' \rightarrow \epsilon$
$)T'E'\$$	$)\$$	dépiler et avancer
$T'E'\$$	$\$$	$T' \rightarrow \epsilon$
$E'\$$	$\$$	$E' \rightarrow \epsilon$
$\$$	$\$$	succès

Exemple 2 :

$S \to aaSbb \mid a \mid \epsilon$
LL(1) ?

- $\texttt{Premier}(aaSbb) \cap \texttt{Premier}(a) = \{a\} \neq \emptyset$

On factorise : $S \to aA \mid \epsilon, \;\; A \to aSbb \mid \epsilon$

- $\texttt{Premier}(aA) \cap \texttt{Premier}(\epsilon) = \emptyset$
- $\texttt{Suivant}(S) \cap \texttt{Premier}(aA) = \{\$, \; b\} \cap \{a\} = \emptyset$
- $\texttt{Premier}(aSbb) \cap \texttt{Premier}(\epsilon) = \emptyset$
- $\texttt{Suivant}(A) \cap \texttt{Premier}(aSbb) = \{\$, \; b\} \cap \{a\} = \emptyset$

$\Longrightarrow$ LL(1).

Exemple 2 (suite) :

$S \rightarrow aA \mid \epsilon, \; A \rightarrow aSbb \mid \epsilon$

	a	b	$\$$
S	aA	ϵ	ϵ
A	$aSbb$	ϵ	ϵ

Exemple 2 (suite) :

$S \rightarrow aA \mid \epsilon, \; A \rightarrow aSbb \mid \epsilon$

Pile	chaîne d'entrée	Action
$S\$$	$aab\$$	$S \rightarrow aA$
$aA\$$	$aab\$$	dépiler et avancer
$A\$$	$ab\$$	$A \rightarrow aSbb$
$aSbb\$$	$ab\$$	dépiler et avancer
$Sbb\$$	$b\$$	$S \rightarrow \epsilon$
$bb\$$	$b\$$	dépiler et avancer
$b\$$	$\$$	échec !

Un exemple d'application

On considère la grammaire $\mathcal{G}$ suivante :

$$\mathcal{G} = (\{a,\ b,\ c,\ d\},\ \{S,\ E,\ F,\ G,\ H,\ C,\ D\},\ S,\ \mathcal{P})$$

où $\mathcal{P}$ est l'ensemble des règles suivantes :

$$\begin{cases} S & \rightarrow & aE \mid bF \\ F & \rightarrow & aF \mid aG \mid aHD \\ G & \rightarrow & Gc \mid d \\ H & \rightarrow & Ca \\ C & \rightarrow & Hb \\ D & \rightarrow & ab \end{cases}$$

$\mathcal{G}$ est-elle réduite ?

Un exemple d'application

$\mathcal{P}rod = \{D,\ G,\ F,\ S\}$

$$\left\{\begin{array}{lcl} S & \rightarrow & bF \\ F & \rightarrow & aF \mid aG \\ G & \rightarrow & Gc \mid d \\ D & \rightarrow & ab \end{array}\right.$$

Un exemple d'application

$\mathcal{A}cc = \{S,\ F,\ G\}$

$$\left\{\begin{array}{lcl} S & \rightarrow & bF \\ F & \rightarrow & aF \mid aG \\ G & \rightarrow & Gc \mid d \end{array}\right.$$

$\mathcal{G}'$ est-elle LL(1) ?

Un exemple d'application

$\mathcal{G}'$:

$$\begin{cases} S & \rightarrow & bF \\ F & \rightarrow & aF' \\ F' & \rightarrow & F \mid G \\ G & \rightarrow & dG' \\ G' & \rightarrow & cG' \mid \epsilon \end{cases}$$

Construire la table d'analyse LL(1) de $\mathcal{G}'$.

Un exemple d'application

	a	b	c	d	\$
S		bF			
F	aF'				
F'	F			G	
G				dG'	
G'			cG'		ϵ

Le mot *baadc* appartient-il à $\mathcal{L}(\mathcal{G}')$?
Si oui, déduire de l'analyse du mot une dérivation gauche.

Un exemple d'application

$S\$$	$baadc\$$	$S \rightarrow bF$
$bF\$$	$baadc\$$	dépiler et avancer
$F\$$	$aadc\$$	$F \rightarrow aF'$
$aF'\$$	$aadc\$$	dépiler et avancer
$F'\$$	$adc\$$	$F' \rightarrow F$
$F\$$	$adc\$$	$F \rightarrow aF'$
$aF'\$$	$adc\$$	dépiler et avancer
$F'\$$	$dc\$$	$F' \rightarrow G$
$G\$$	$dc\$$	$G \rightarrow dG'$
$dG'\$$	$dc\$$	dépiler et avancer
$G'\$$	$c\$$	$G' \rightarrow cG'$
$cG'\$$	$c\$$	dépiler et avancer
$G'\$$	$\$$	$G' \rightarrow \epsilon$
$\$$	$\$$	succès

dérivation gauche : 3ème colonne avec $\Rightarrow$ au lieu de $\rightarrow$.