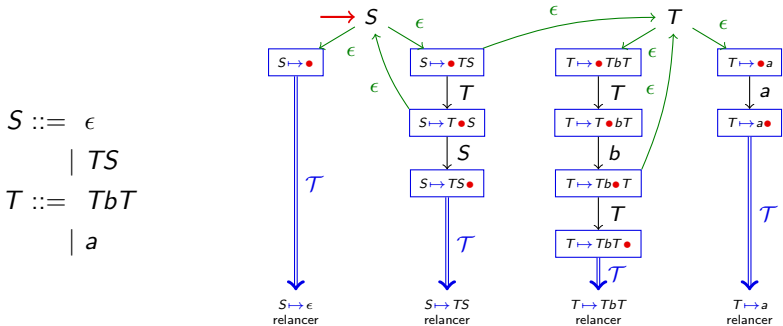


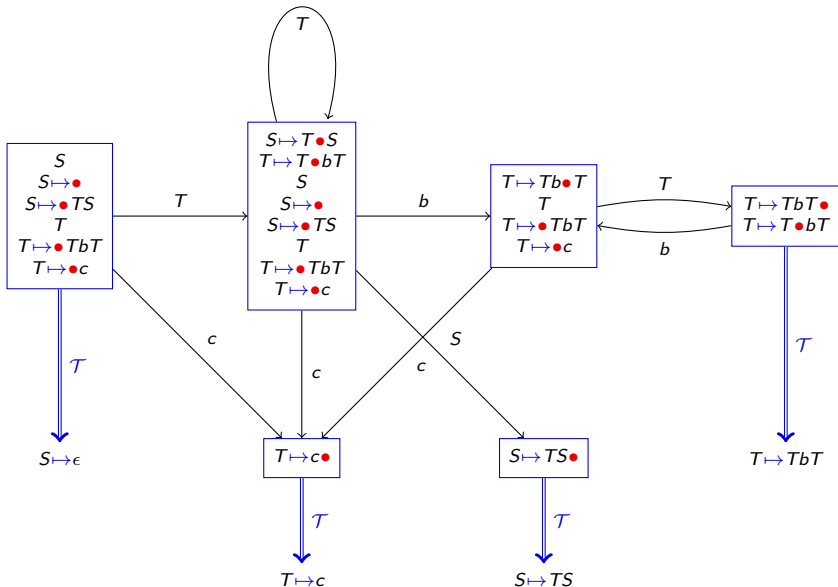
Construction de l'automate non déterministe

Idée rapide

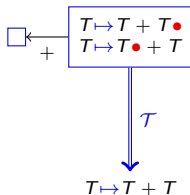
- un état pour chaque NT puis reconnaissance du pattern de chaque règle indépendamment,
- ajout de ϵ -transitions du NT à la règle et de la règle partiel vers un NT si celui-ci est attendu par la règle,
- au bout de chaque pattern, action de réduction de la règle et de relance de l'automate, ce après n'importe quel symbole lu,



Pseudo-déterminisation : Parseur LR₀



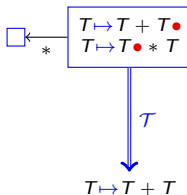
Quelques conflits standards



Associativité

associativité droite :
choisir le shift

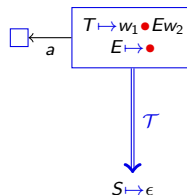
associativité gauche :
choisir l'action



Priorité

* prioritaire :
choisir le shift

+ prioritaire :
choisir l'action

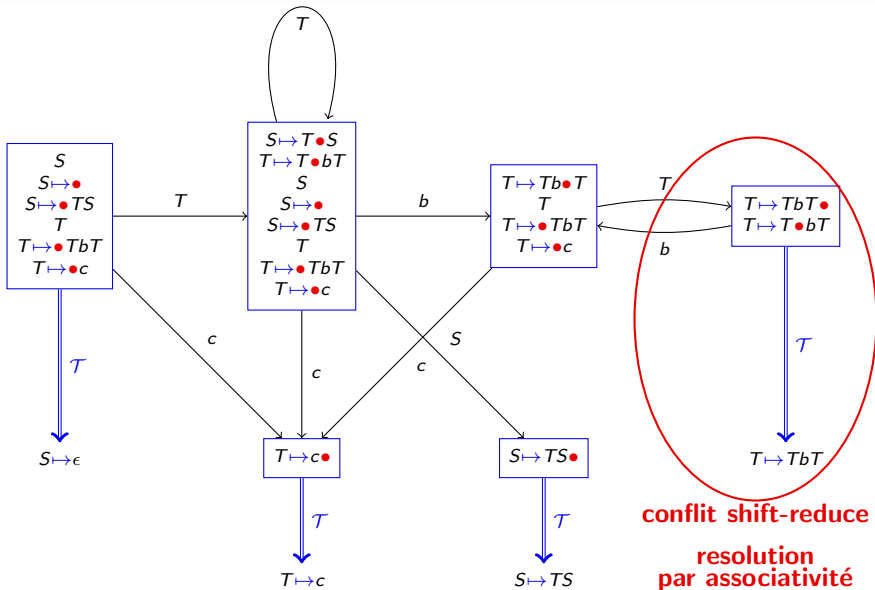


Règle vide

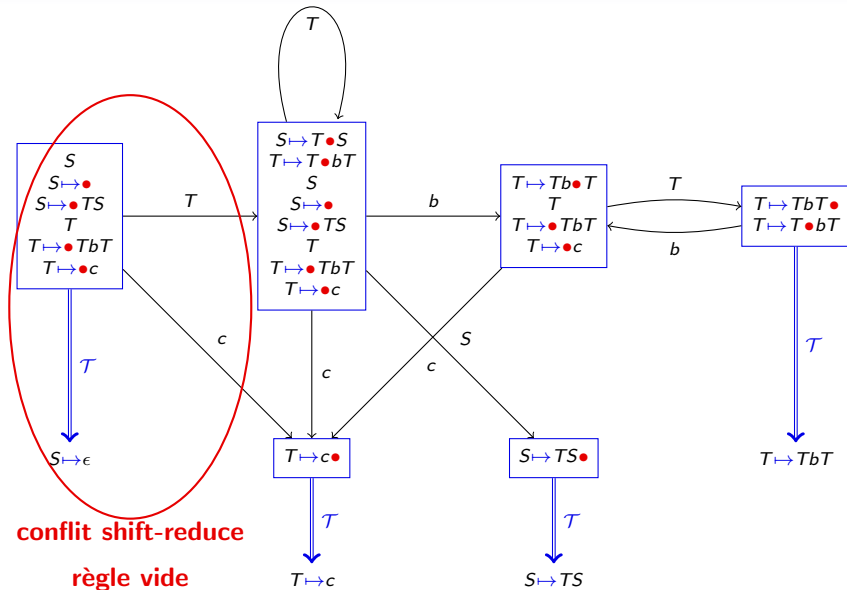
conflit artificiel :
choisir le shift

Automatiquement
résolu avec SLR

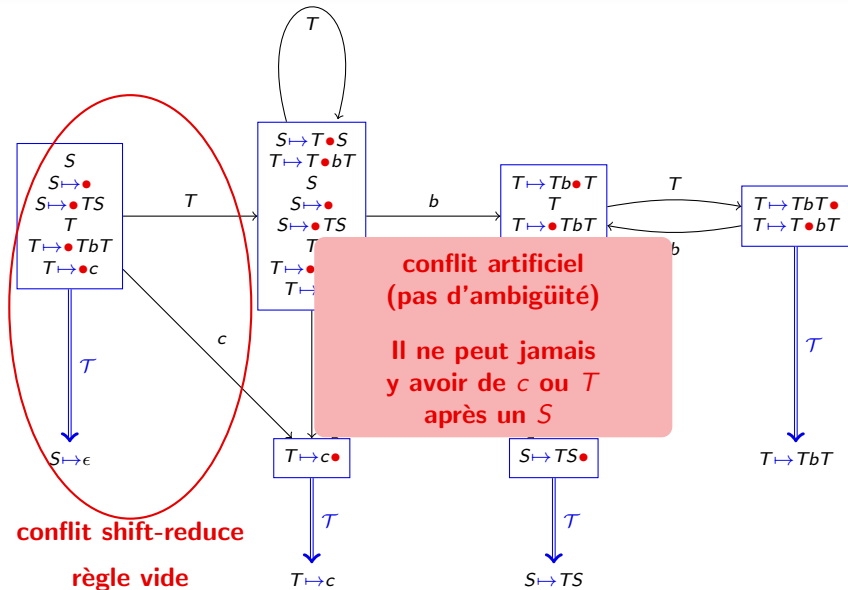
Conflicts



Conflicts



Conflicts



Enlever d'autres conflits : restreindre les conditions d'actions

Si le non-terminal créé n'est jamais suivi du symbole lu
l'action n'a pas de raison d'être prise

Pour ça on calcule les "follows" :

les terminaux qui peuvent suivre un non-terminal.

Pour calculer les "follow" correctement on va avoir besoin des "firsts" :

les terminaux qui peuvent débiter un non-terminal.

Le calcul des *firsts* et *follows* se fait par saturation : on parcourt la grammaire en cherchant à ajouter des nouveaux first/follow et on s'arrête lorsque l'on n'arrive plus à rien ajouter.

Les *Firsts* sans règle vide

Firsts : les terminaux pouvant débiter un mot de non-terminaux et terminaux w

$$\text{First}(w \in (\mathcal{N} \cup \mathcal{T})^*) \subseteq \mathcal{T}$$

First est défini par le plus petit ensemble tel que :

- si $t \in \mathcal{T}$, $\text{First}(t) = \{t\}$,
- pour toute règle $N \mapsto w$, $\text{First}(N) \supseteq \text{First}(w)$,
- $\text{First}(w_1 w_2) = \text{First}(w_1)$.

Exemple de *First*s sans règle vide Algébriquement

Grammaire

$$\begin{aligned} \langle \text{exp} \rangle &::= \langle \text{exp} \rangle - \langle \text{terme} \rangle \\ &| \langle \text{terme} \rangle \\ \langle \text{terme} \rangle &::= \langle \text{terme} \rangle * \langle \text{fac} \rangle \\ &| \langle \text{fac} \rangle \\ \langle \text{fac} \rangle &::= (\langle \text{exp} \rangle) \\ &| - \langle \text{fac} \rangle \\ &| \langle \text{NB} \rangle \end{aligned}$$

Firsts

$$\begin{aligned} \text{First}(\langle \text{exp} \rangle) &= \\ &\quad \text{First}(\langle \text{exp} \rangle - \langle \text{terme} \rangle) \\ &\quad \cup \text{First}(\langle \text{terme} \rangle) \\ \text{First}(\langle \text{term} \rangle) &= \\ &\quad \text{First}(\langle \text{terme} \rangle * \langle \text{fac} \rangle) \\ &\quad \cup \text{First}(\langle \text{fac} \rangle) \\ \text{First}(\langle \text{fac} \rangle) &= \\ &\quad \text{First}(\langle \text{exp} \rangle) \\ &\quad \cup \text{First}(- \langle \text{fac} \rangle) \\ &\quad \cup \text{First}(\langle \text{NB} \rangle) \end{aligned}$$

Exemple de *First*s sans règle vide Algébriquement

Grammaire

$$\begin{aligned} \langle \text{exp} \rangle &::= \langle \text{exp} \rangle - \langle \text{terme} \rangle \\ &| \langle \text{terme} \rangle \\ \langle \text{terme} \rangle &::= \langle \text{terme} \rangle * \langle \text{fac} \rangle \\ &| \langle \text{fac} \rangle \\ \langle \text{fac} \rangle &::= (\langle \text{exp} \rangle) \\ &| - \langle \text{fac} \rangle \\ &| \langle \text{NB} \rangle \end{aligned}$$

Firsts

$$\begin{aligned} \text{First}(\langle \text{exp} \rangle) &= \\ &\text{First}(\langle \text{exp} \rangle) \cup \text{First}(-) \cup \text{First}(\langle \text{terme} \rangle) \\ &\cup \text{First}(\langle \text{terme} \rangle) \\ \text{First}(\langle \text{term} \rangle) &= \\ &\text{First}(\langle \text{terme} \rangle) \cup \text{First}(*) \cup \text{First}(\langle \text{fac} \rangle) \\ &\cup \text{First}(\langle \text{fac} \rangle) \\ \text{First}(\langle \text{fac} \rangle) &= \\ &\text{First}() \cup \text{First}(\langle \text{exp} \rangle) \cup \text{First}() \\ &\cup \text{First}(-) \cup \text{First}(\langle \text{fac} \rangle) \\ &\cup \text{First}(\langle \text{NB} \rangle) \end{aligned}$$

Exemple de *First*s sans règle vide Algébriquement

Grammaire

$$\begin{aligned} \langle \text{exp} \rangle &::= \langle \text{exp} \rangle - \langle \text{terme} \rangle \\ &\quad | \langle \text{terme} \rangle \\ \langle \text{terme} \rangle &::= \langle \text{terme} \rangle * \langle \text{fac} \rangle \\ &\quad | \langle \text{fac} \rangle \\ \langle \text{fac} \rangle &::= (\langle \text{exp} \rangle) \\ &\quad | - \langle \text{fac} \rangle \\ &\quad | \langle \text{NB} \rangle \end{aligned}$$

Firsts

$$\begin{aligned} \text{First}(\langle \text{exp} \rangle) &= \\ &\quad \text{First}(\langle \text{exp} \rangle) \\ &\quad \cup \text{First}(\langle \text{terme} \rangle) \\ \text{First}(\langle \text{term} \rangle) &= \\ &\quad \text{First}(\langle \text{terme} \rangle) \\ &\quad \cup \text{First}(\langle \text{fac} \rangle) \\ \text{First}(\langle \text{fac} \rangle) &= \\ &\quad \text{First}() \\ &\quad \cup \text{First}(-) \\ &\quad \cup \text{First}(\langle \text{NB} \rangle) \end{aligned}$$

Exemple de *First*s sans règle vide Algébriquement

Grammaire

$$\begin{aligned}\langle \text{exp} \rangle &::= \langle \text{exp} \rangle - \langle \text{terme} \rangle \\ &| \langle \text{terme} \rangle \\ \langle \text{terme} \rangle &::= \langle \text{terme} \rangle * \langle \text{fac} \rangle \\ &| \langle \text{fac} \rangle \\ \langle \text{fac} \rangle &::= (\langle \text{exp} \rangle) \\ &| - \langle \text{fac} \rangle \\ &| \langle \text{NB} \rangle\end{aligned}$$

Firsts

$$\begin{aligned}\text{First}(\langle \text{exp} \rangle) &= \\ &\text{First}(\langle \text{exp} \rangle) \\ &\cup \text{First}(\langle \text{terme} \rangle) \\ \text{First}(\langle \text{term} \rangle) &= \\ &\text{First}(\langle \text{terme} \rangle) \\ &\cup \text{First}(\langle \text{fac} \rangle) \\ \text{First}(\langle \text{fac} \rangle) &= \{ (, -, \langle \text{NB} \rangle)\}\end{aligned}$$

Exemple de *First*s sans règle vide Algébriquement

Grammaire

$$\begin{aligned}\langle \text{exp} \rangle &::= \langle \text{exp} \rangle - \langle \text{terme} \rangle \\ &\quad | \langle \text{terme} \rangle \\ \langle \text{terme} \rangle &::= \langle \text{terme} \rangle * \langle \text{fac} \rangle \\ &\quad | \langle \text{fac} \rangle \\ \langle \text{fac} \rangle &::= (\langle \text{exp} \rangle) \\ &\quad | - \langle \text{fac} \rangle \\ &\quad | \langle \text{NB} \rangle\end{aligned}$$

Firsts

$$\begin{aligned}\text{First}(\langle \text{exp} \rangle) &= \\ &\quad \text{First}(\langle \text{exp} \rangle) \\ &\quad \cup \text{First}(\langle \text{terme} \rangle) \\ \text{First}(\langle \text{term} \rangle) &= \{ (, -, \langle \text{NB} \rangle \} \\ \text{First}(\langle \text{fac} \rangle) &= \{ (, -, \langle \text{NB} \rangle \}\end{aligned}$$

Exemple de *First*s sans règle vide Algébriquement

Grammaire

$$\begin{aligned}\langle \text{exp} \rangle &::= \langle \text{exp} \rangle - \langle \text{terme} \rangle \\ &\quad | \langle \text{terme} \rangle \\ \langle \text{terme} \rangle &::= \langle \text{terme} \rangle * \langle \text{fac} \rangle \\ &\quad | \langle \text{fac} \rangle \\ \langle \text{fac} \rangle &::= (\langle \text{exp} \rangle) \\ &\quad | - \langle \text{fac} \rangle \\ &\quad | \langle \text{NB} \rangle\end{aligned}$$

Firsts

$$\text{First}(\langle \text{exp} \rangle) = \{ (, -, \langle \text{NB} \rangle) \}$$
$$\text{First}(\langle \text{term} \rangle) = \{ (, -, \langle \text{NB} \rangle) \}$$
$$\text{First}(\langle \text{fac} \rangle) = \{ (, -, \langle \text{NB} \rangle) \}$$

Exemple de *Firsts* sans règle vide

Par point fixe

Grammaire

```
 $\langle \text{exp} \rangle ::= \langle \text{exp} \rangle - \langle \text{terme} \rangle$   
          |  $\langle \text{terme} \rangle$   
 $\langle \text{terme} \rangle ::= \langle \text{terme} \rangle * \langle \text{fac} \rangle$   
          |  $\langle \text{fac} \rangle$   
 $\langle \text{fac} \rangle ::= (\langle \text{exp} \rangle)$   
          |  $- \langle \text{fac} \rangle$   
          |  $\langle \text{NB} \rangle$ 
```

Firsts

$\text{First}(\langle \text{exp} \rangle) \supseteq \{ \}$

$\text{First}(\langle \text{term} \rangle) \supseteq \{ \}$

$\text{First}(\langle \text{fac} \rangle) \supseteq \{ \}$

Exemple de *First*s sans règle vide

Par point fixe

Grammaire

```
 $\langle \text{exp} \rangle ::= \langle \text{exp} \rangle - \langle \text{terme} \rangle$   
          |  $\langle \text{terme} \rangle$   
 $\langle \text{terme} \rangle ::= \langle \text{terme} \rangle * \langle \text{fac} \rangle$   
          |  $\langle \text{fac} \rangle$   
 $\langle \text{fac} \rangle ::= (\langle \text{exp} \rangle)$   
          |  $- \langle \text{fac} \rangle$   
          |  $\langle \text{NB} \rangle$ 
```

Firsts

$\text{First}(\langle \text{exp} \rangle) \supseteq \{ \}$

$\text{First}(\langle \text{term} \rangle) \supseteq \{ \}$

$\text{First}(\langle \text{fac} \rangle) \supseteq \{ (\}$

Exemple de *Firsts* sans règle vide

Par point fixe

Grammaire

```
 $\langle \text{exp} \rangle ::= \langle \text{exp} \rangle - \langle \text{terme} \rangle$   
          |  $\langle \text{terme} \rangle$   
 $\langle \text{terme} \rangle ::= \langle \text{terme} \rangle * \langle \text{fac} \rangle$   
          |  $\langle \text{fac} \rangle$   
 $\langle \text{fac} \rangle ::= (\langle \text{exp} \rangle)$   
          |  $- \langle \text{fac} \rangle$   
          |  $\langle \text{NB} \rangle$ 
```

Firsts

$\text{First}(\langle \text{exp} \rangle) \supseteq \{ \}$

$\text{First}(\langle \text{term} \rangle) \supseteq \{ \}$

$\text{First}(\langle \text{fac} \rangle) \supseteq \{ (, - \}$

Exemple de *Firsts* sans règle vide

Par point fixe

Grammaire

$$\begin{aligned} \langle \text{exp} \rangle &::= \langle \text{exp} \rangle - \langle \text{terme} \rangle \\ &| \langle \text{terme} \rangle \\ \langle \text{terme} \rangle &::= \langle \text{terme} \rangle * \langle \text{fac} \rangle \\ &| \langle \text{fac} \rangle \\ \langle \text{fac} \rangle &::= (\langle \text{exp} \rangle) \\ &| - \langle \text{fac} \rangle \\ &| \langle \text{NB} \rangle \end{aligned}$$

Firsts

$$\text{First}(\langle \text{exp} \rangle) \supseteq \{ \}$$

$$\text{First}(\langle \text{term} \rangle) \supseteq \{ \}$$

$$\text{First}(\langle \text{fac} \rangle) \supseteq \{ (, -, \langle \text{NB} \rangle \}$$

Exemple de *First*s sans règle vide

Par point fixe

Grammaire

$$\begin{aligned} \langle \text{exp} \rangle &::= \langle \text{exp} \rangle - \langle \text{terme} \rangle \\ &| \langle \text{terme} \rangle \\ \langle \text{terme} \rangle &::= \langle \text{terme} \rangle * \langle \text{fac} \rangle \\ &| \langle \text{fac} \rangle \\ \langle \text{fac} \rangle &::= (\langle \text{exp} \rangle) \\ &| - \langle \text{fac} \rangle \\ &| \langle \text{NB} \rangle \end{aligned}$$

Firsts

$$\text{First}(\langle \text{exp} \rangle) \supseteq \{ \}$$

$$\text{First}(\langle \text{term} \rangle) \supseteq \{ (, -, \langle \text{NB} \rangle \}$$

$$\text{First}(\langle \text{fac} \rangle) \supseteq \{ (, -, \langle \text{NB} \rangle \}$$

Exemple de *First*s sans règle vide

Par point fixe

Grammaire

```
 $\langle \text{exp} \rangle ::= \langle \text{exp} \rangle - \langle \text{terme} \rangle$   
          |  $\langle \text{terme} \rangle$   
 $\langle \text{terme} \rangle ::= \langle \text{terme} \rangle * \langle \text{fac} \rangle$   
          |  $\langle \text{fac} \rangle$   
 $\langle \text{fac} \rangle ::= (\langle \text{exp} \rangle)$   
          |  $- \langle \text{fac} \rangle$   
          |  $\langle \text{NB} \rangle$ 
```

Firsts

$\text{First}(\langle \text{exp} \rangle) \supseteq \{ (, -, \langle \text{NB} \rangle) \}$

$\text{First}(\langle \text{term} \rangle) \supseteq \{ (, -, \langle \text{NB} \rangle) \}$

$\text{First}(\langle \text{fac} \rangle) \supseteq \{ (, -, \langle \text{NB} \rangle) \}$

Exemple de *First*s sans règle vide

Par point fixe

Grammaire

```
⟨exp⟩ ::= ⟨exp⟩ - ⟨terme⟩  
        | ⟨terme⟩  
⟨terme⟩ ::= ⟨terme⟩ * ⟨fac⟩  
           | ⟨fac⟩  
⟨fac⟩ ::= (⟨exp⟩)  
        | -⟨fac⟩  
        | ⟨NB⟩
```

Firsts

$\text{First}(\langle \text{exp} \rangle) = \{ (, -, \langle \text{NB} \rangle) \}$

$\text{First}(\langle \text{term} \rangle) = \{ (, -, \langle \text{NB} \rangle) \}$

$\text{First}(\langle \text{fac} \rangle) = \{ (, -, \langle \text{NB} \rangle) \}$

Exemple de *Firsts* avec règle vide Algébriquement

Grammaire

$$\begin{aligned}
 S &::= T;S \\
 &\quad | ! \\
 T &::= T+T \\
 &\quad | a \\
 &\quad | \epsilon
 \end{aligned}$$

Firsts ??

$$\text{First}(S) = \{a, !\}$$

$$\text{First}(T) = \{a\}$$

Faux : ';' doit être dans $\text{First}(S)$

Les *Firsts*

Firsts : les terminaux pouvant débiter un mot de non-terminaux et terminaux w

Attention, on utilise ε comme symbole supplémentaire pour signifier que w peut reconnaître le mot vide :

$$\text{First}(w \in (\mathcal{N} \cup \mathcal{T})^*) \subseteq \mathcal{T} \cup \{\varepsilon\}$$

First est défini par le plus petit ensemble tel que :

- si $t \in \mathcal{T}$, $\text{First}(t) = \{t\}$,
- pour toute règle $N \mapsto w$, $\text{First}(N) \supseteq \text{First}(w)$,
- $\text{First}(\varepsilon) = \{\varepsilon\}$,
- $\text{First}(w_1 w_2) = \text{First}(w_1) \dot{\cup} \text{First}(w_2)$.

Notation : Soient $A, B \subseteq \mathcal{T} \cup \{\varepsilon\}$, alors on note

$$A \dot{\cup} B := \begin{cases} A & \text{si } \varepsilon \notin A \\ (A - \{\varepsilon\}) \cup B & \text{si } \varepsilon \in A \end{cases}$$

Exemple de *First*s avec règle vide Algébriquement

Grammaire

$$\begin{aligned} S &::= T;S \\ &| ! \\ T &::= T+T \\ &| a \\ &| \epsilon \end{aligned}$$

Firsts

$$\begin{aligned} \text{First}(S) &= \\ &\quad \text{First}(T;S) \\ &\quad \cup \text{First}(!) \\ \text{First}(T) &= \\ &\quad \text{First}(T+T) \\ &\quad \cup \text{First}(a) \\ &\quad \cup \text{First}(\epsilon) \end{aligned}$$

Exemple de *First*s avec règle vide Algébriquement

Grammaire

$$\begin{aligned}
 S &::= T;S \\
 &| ! \\
 T &::= T+T \\
 &| a \\
 &| \epsilon
 \end{aligned}$$

Firsts

$$\begin{aligned}
 \text{First}(S) &= \\
 &\quad \text{First}(T) \cup \text{First}(;) \cup \text{First}(S) \\
 &\quad \cup \text{First}(!) \\
 \text{First}(T) &= \\
 &\quad \text{First}(T) \cup \text{First}(+) \cup \text{First}(T) \\
 &\quad \cup \text{First}(a) \\
 &\quad \cup \{\epsilon\}
 \end{aligned}$$

Exemple de *First*s avec règle vide Algébriquement

Grammaire

$$\begin{aligned}
 S &::= T;S \\
 &\quad | ! \\
 T &::= T+T \\
 &\quad | a \\
 &\quad | \epsilon
 \end{aligned}$$

Firsts

$$\begin{aligned}
 \text{First}(S) &= \\
 &\quad \text{First}(T) \cup \text{First}(;) \\
 &\quad \cup \{!\} \\
 \text{First}(T) &= \\
 &\quad \text{First}(T) \cup \text{First}(+) \\
 &\quad \cup \{a, \epsilon\}
 \end{aligned}$$

Exemple de *First*s avec règle vide Algébriquement

Grammaire

$$\begin{aligned} S &::= T;S \\ &| ! \\ T &::= T+T \\ &| a \\ &| \epsilon \end{aligned}$$

Firsts

$$\begin{aligned} \text{First}(S) &= \\ &\quad \text{First}(T) \cup \text{First}(;) \\ &\quad \cup \{!\} \\ \text{First}(T) &= \{a, \epsilon, +\} \end{aligned}$$

Exemple de *First*s avec règle vide Algébriquement

Grammaire

$$\begin{aligned}
 S &::= T;S \\
 &\quad | ! \\
 T &::= T+T \\
 &\quad | a \\
 &\quad | \epsilon
 \end{aligned}$$

Firsts

$$\text{First}(S) = \{a, +, ;, !\}$$

$$\text{First}(T) = \{a, \epsilon, +\}$$

Exemple de *Firsts* avec règle vide Par point fixe

Grammaire

$$\begin{aligned} S &::= T;S \\ &| ! \\ T &::= T+T \\ &| a \\ &| \epsilon \end{aligned}$$

Firsts

$$\text{First}(S) \supseteq \{ \}$$
$$\text{First}(T) \supseteq \{ \}$$

Exemple de *First*s avec règle vide

Par point fixe

Grammaire

$$\begin{aligned} S &::= T;S \\ &| ! \\ T &::= T+T \\ &| a \\ &| \epsilon \end{aligned}$$

Firsts

$$\text{First}(S) \supseteq \{!\}$$
$$\text{First}(T) \supseteq \{\}$$

Exemple de *Firsts* avec règle vide

Par point fixe

Grammaire

$$\begin{aligned} S &::= T;S \\ &| ! \\ T &::= T+T \\ &| a \\ &| \epsilon \end{aligned}$$

Firsts

$$\text{First}(S) \supseteq \{!\}$$
$$\text{First}(T) \supseteq \{a\}$$

Exemple de *First*s avec règle vide

Par point fixe

Grammaire

$$\begin{aligned} S &::= T;S \\ &| ! \\ T &::= T+T \\ &| a \\ &| \epsilon \end{aligned}$$

Firsts

$$\text{First}(S) \supseteq \{!\}$$
$$\text{First}(T) \supseteq \{a, \epsilon\}$$

Exemple de *Firsts* avec règle vide

Par point fixe

Grammaire

$$\begin{aligned} S &::= T;S \\ &| ! \\ T &::= T+T \\ &| a \\ &| \epsilon \end{aligned}$$

Firsts

$$\text{First}(S) \supseteq \{!, a, ;\}$$
$$\text{First}(T) \supseteq \{a, \epsilon\}$$

Exemple de *Firsts* avec règle vide

Par point fixe

Grammaire

$$\begin{aligned}
 S &::= T;S \\
 &| ! \\
 T &::= T+T \\
 &| a \\
 &| \epsilon
 \end{aligned}$$

Firsts

$$\text{First}(S) \supseteq \{!, a, ;\}$$

$$\text{First}(T) \supseteq \{a, \epsilon, +\}$$

Exemple de *Firsts* avec règle vide Par point fixe

Grammaire

$$\begin{aligned} S &::= T;S \\ &| ! \\ T &::= T+T \\ &| a \\ &| \epsilon \end{aligned}$$

Firsts

$$\text{First}(S) \supseteq \{!, a, ;, +\}$$
$$\text{First}(T) \supseteq \{a, \epsilon, +\}$$

Exemple de *Firsts* avec règle vide

Par point fixe

Grammaire

$$\begin{aligned} S &::= T;S \\ &| ! \\ T &::= T+T \\ &| a \\ &| \epsilon \end{aligned}$$

Firsts

$$\text{First}(S) = \{!, a, ;, +\}$$
$$\text{First}(T) = \{a, \epsilon, +\}$$

Les Follows

Follows : les terminaux pouvant suivre non-terminal N

Attention, on utilise \$ comme symbole supplémentaire pour signifier que N peut être à la fin du fichier :

$$\text{Follow}(N \in \mathcal{N}) \subseteq \mathcal{T} \cup \{\$\}$$

Follow est défini par le plus petit ensemble tel que :

- si S est le non terminal principal,

$$\text{Follow}(S) \ni \$,$$

- pour toute règle $N' \mapsto w_1 N w_2$,

$$\text{Follow}(N) \supseteq \text{First}(w_2) \cup \text{Follow}(N')$$

Exemple de *Follow* sans ϵ

Grammaire

```
 $\langle \text{exp} \rangle ::= \langle \text{exp} \rangle - \langle \text{terme} \rangle$   
          |  $\langle \text{terme} \rangle$   
 $\langle \text{terme} \rangle ::= \langle \text{terme} \rangle * \langle \text{fac} \rangle$   
          |  $\langle \text{fac} \rangle$   
 $\langle \text{fac} \rangle ::= (\langle \text{exp} \rangle)$   
          |  $- \langle \text{fac} \rangle$   
          |  $\langle \text{NB} \rangle$ 
```

Follows

$\text{Follow}(\langle \text{exp} \rangle) \supseteq \{\$ \}$

$\text{Follow}(\langle \text{term} \rangle) \supseteq \{ \}$

$\text{Follow}(\langle \text{fac} \rangle) \supseteq \{ \}$

Exemple de *Follow* sans ϵ

Grammaire

```
 $\langle \text{exp} \rangle ::= \langle \text{exp} \rangle - \langle \text{terme} \rangle$   
          |  $\langle \text{terme} \rangle$   
 $\langle \text{terme} \rangle ::= \langle \text{terme} \rangle * \langle \text{fac} \rangle$   
          |  $\langle \text{fac} \rangle$   
 $\langle \text{fac} \rangle ::= (\langle \text{exp} \rangle)$   
          |  $- \langle \text{fac} \rangle$   
          |  $\langle \text{NB} \rangle$ 
```

Follows

$\text{Follow}(\langle \text{exp} \rangle) \supseteq \{ \$, - \}$

$\text{Follow}(\langle \text{term} \rangle) \supseteq \{ \}$

$\text{Follow}(\langle \text{fac} \rangle) \supseteq \{ \}$

Exemple de *Follow* sans ϵ

Grammaire

```
 $\langle \text{exp} \rangle ::= \langle \text{exp} \rangle - \langle \text{terme} \rangle$   
          |  $\langle \text{terme} \rangle$   
 $\langle \text{terme} \rangle ::= \langle \text{terme} \rangle * \langle \text{fac} \rangle$   
          |  $\langle \text{fac} \rangle$   
 $\langle \text{fac} \rangle ::= (\langle \text{exp} \rangle)$   
          |  $- \langle \text{fac} \rangle$   
          |  $\langle \text{NB} \rangle$ 
```

Follows

$\text{Follow}(\langle \text{exp} \rangle) \supseteq \{ \$, - \}$

$\text{Follow}(\langle \text{term} \rangle) \supseteq \{ \$, - \}$

$\text{Follow}(\langle \text{fac} \rangle) \supseteq \{ \}$

Exemple de *Follow* sans ϵ

Grammaire

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle - \langle \text{terme} \rangle$
| $\langle \text{terme} \rangle$

$\langle \text{terme} \rangle ::= \langle \text{terme} \rangle * \langle \text{fac} \rangle$
| $\langle \text{fac} \rangle$

$\langle \text{fac} \rangle ::= (\langle \text{exp} \rangle)$
| $- \langle \text{fac} \rangle$
| $\langle \text{NB} \rangle$

Follows

$\text{Follow}(\langle \text{exp} \rangle) \supseteq \{ \$, - \}$

$\text{Follow}(\langle \text{term} \rangle) \supseteq \{ \$, -, * \}$

$\text{Follow}(\langle \text{fac} \rangle) \supseteq \{ \}$

Exemple de *Follow* sans ϵ

Grammaire

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle - \langle \text{terme} \rangle$
| $\langle \text{terme} \rangle$

$\langle \text{terme} \rangle ::= \langle \text{terme} \rangle * \langle \text{fac} \rangle$
| $\langle \text{fac} \rangle$

$\langle \text{fac} \rangle ::= (\langle \text{exp} \rangle)$
| $- \langle \text{fac} \rangle$
| $\langle \text{NB} \rangle$

Follows

$\text{Follow}(\langle \text{exp} \rangle) \supseteq \{ \$, - \}$

$\text{Follow}(\langle \text{term} \rangle) \supseteq \{ \$, -, * \}$

$\text{Follow}(\langle \text{fac} \rangle) \supseteq \{ \$, -, * \}$

Exemple de *Follow* sans ϵ

Grammaire

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle - \langle \text{terme} \rangle$
| $\langle \text{terme} \rangle$

$\langle \text{terme} \rangle ::= \langle \text{terme} \rangle * \langle \text{fac} \rangle$
| $\langle \text{fac} \rangle$

$\langle \text{fac} \rangle ::= (\langle \text{exp} \rangle)$
| $- \langle \text{fac} \rangle$
| $\langle \text{NB} \rangle$

Follows

$\text{Follow}(\langle \text{exp} \rangle) \supseteq \{ \$, -,) \}$

$\text{Follow}(\langle \text{term} \rangle) \supseteq \{ \$, -, * \}$

$\text{Follow}(\langle \text{fac} \rangle) \supseteq \{ \$, -, * \}$

Exemple de *Follow* sans ϵ

Grammaire

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle - \langle \text{terme} \rangle$
| $\langle \text{terme} \rangle$

$\langle \text{terme} \rangle ::= \langle \text{terme} \rangle * \langle \text{fac} \rangle$
| $\langle \text{fac} \rangle$

$\langle \text{fac} \rangle ::= (\langle \text{exp} \rangle)$
| $- \langle \text{fac} \rangle$
| $\langle \text{NB} \rangle$

Follows

$\text{Follow}(\langle \text{exp} \rangle) \supseteq \{ \$, -,) \}$

$\text{Follow}(\langle \text{term} \rangle) \supseteq \{ \$, -, *,) \}$

$\text{Follow}(\langle \text{fac} \rangle) \supseteq \{ \$, -, * \}$

Exemple de *Follow* sans ϵ

Grammaire

$$\begin{aligned} \langle \text{exp} \rangle &::= \langle \text{exp} \rangle - \langle \text{terme} \rangle \\ &\quad | \langle \text{terme} \rangle \\ \langle \text{terme} \rangle &::= \langle \text{terme} \rangle * \langle \text{fac} \rangle \\ &\quad | \langle \text{fac} \rangle \\ \langle \text{fac} \rangle &::= (\langle \text{exp} \rangle) \\ &\quad | - \langle \text{fac} \rangle \\ &\quad | \langle \text{NB} \rangle \end{aligned}$$

Follows

$$\text{Follow}(\langle \text{exp} \rangle) \supseteq \{ \$, -,) \}$$

$$\text{Follow}(\langle \text{term} \rangle) \supseteq \{ \$, -, *,) \}$$

$$\text{Follow}(\langle \text{fac} \rangle) \supseteq \{ \$, -, *,) \}$$

Exemple de *Follow* sans ϵ

Grammaire

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle - \langle \text{terme} \rangle$
| $\langle \text{terme} \rangle$

$\langle \text{terme} \rangle ::= \langle \text{terme} \rangle * \langle \text{fac} \rangle$
| $\langle \text{fac} \rangle$

$\langle \text{fac} \rangle ::= (\langle \text{exp} \rangle)$
| $- \langle \text{fac} \rangle$
| $\langle \text{NB} \rangle$

Follows

$\text{Follow}(\langle \text{exp} \rangle) = \{ \$, -,) \}$

$\text{Follow}(\langle \text{term} \rangle) = \{ \$, -, *,) \}$

$\text{Follow}(\langle \text{fac} \rangle) = \{ \$, -, *,) \}$

Exemple de *Follow* complexe (avec ϵ)

Grammaire

$$\begin{array}{l}
 S ::= TS; \\
 \quad | \epsilon \\
 T ::= T+T \\
 \quad | a \\
 \quad | \epsilon
 \end{array}$$

Follows

$$\text{Follow}(S) \supseteq \{\$\}$$

$$\text{Follow}(T) \supseteq \{\}$$

Sachant que:

$$\text{First}(S) = \{\epsilon, a, +, ;\}$$

$$\text{First}(T) = \{\epsilon, a, +\}$$

Exemple de *Follow* complexe (avec ϵ)

Grammaire

$$\begin{array}{l}
 S ::= TS; \\
 \quad | \epsilon \\
 T ::= T+T \\
 \quad | a \\
 \quad | \epsilon
 \end{array}$$

Follows

$$\text{Follow}(S) \supseteq \{\$\}$$

$$\text{Follow}(T) \supseteq \{a, +, ;\}$$

Sachant que:

$$\text{First}(S) = \{\epsilon, a, +, ;\}$$

$$\text{First}(T) = \{\epsilon, a, +\}$$

Exemple de *Follow* complexe (avec ϵ)

Grammaire

$$\begin{array}{l}
 S ::= TS; \\
 \quad | \epsilon \\
 T ::= T+T \\
 \quad | a \\
 \quad | \epsilon
 \end{array}$$

Follows

$$\text{Follow}(S) \supseteq \{\$, ;\}$$

$$\text{Follow}(T) \supseteq \{a, +, ;\}$$

Sachant que:

$$\text{First}(S) = \{\epsilon, a, +, ;\}$$

$$\text{First}(T) = \{\epsilon, a, +\}$$

Exemple de *Follow* complexe (avec ϵ)

Grammaire

$$\begin{aligned} S &::= TS; \\ &| \epsilon \\ T &::= T+T \\ &| a \\ &| \epsilon \end{aligned}$$

Follows

$$\text{Follow}(S) = \{\$, ;\}$$
$$\text{Follow}(T) = \{a, +, ;\}$$

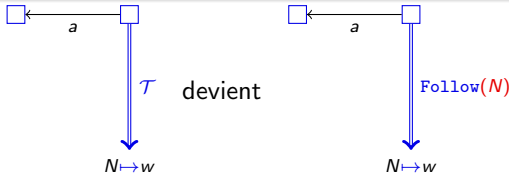
Sachant que:

$$\text{First}(S) = \{\epsilon, a, +, ;\}$$
$$\text{First}(T) = \{\epsilon, a, +\}$$

Parseur SLR

Idée : Restreindre les actions aux *Follows*

- On construit l'automate LR_0
- sauf qu'une action de réduction $N \mapsto w$ ne peut être prise que en lisant les lettres de $\text{Follow}(N)$.



LRO
○○○○

SLR
○○○○○○○○○○○○●

Optimisations
○○

LR1†
○○○○○

LALR†
○○

LL†
○○○○○

Au delà de l'ambiguïté
○○○

Exemple

Optim 1 : Abolir les boucles

L'exécution peut boucler

la grammaire $S ::= S$ produit l'automate:



Solution

On ajoute systématiquement un non-terminal $S_0 ::= S$ qui est le nouveau non-terminal principal

Cette solution marche car elle introduit un conflit action-action là où on aurait bouclé.

Optim 2 : temps linéaire

L'exécution peut être quadratique

la grammaire $S := aS | \epsilon$ produit l'automate:



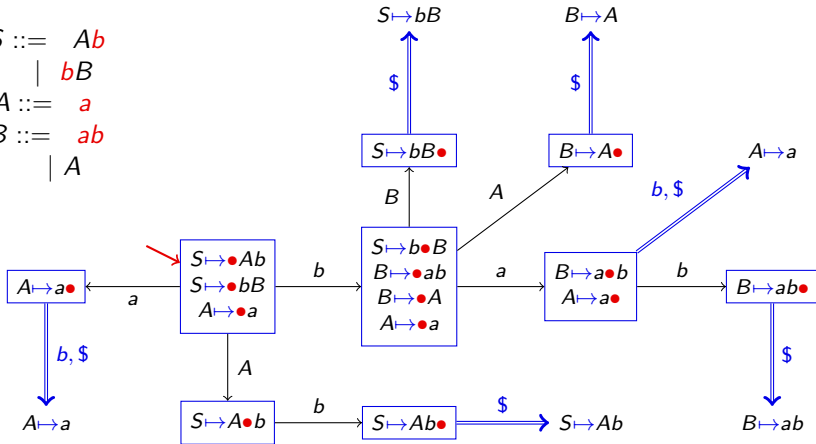
Solution

- Dans le buffer (qui est maintenant une pile), on intercale l'identifiant de l'état d'où l'on vient en plus de la lettre lue à chaque fois que l'on prend une transition.
- lors de l'action de réduction, on retire les éléments réduits et les noms d'états au sommet de pile, et on revient sur le dernier état non réduit.

Pour ceux à qui ça dit quelque chose : cela signifie que l'on utilise un automate à pile.

Encore quelques conflits

$S ::= Ab$
 $\quad | bB$
 $A ::= a$
 $B ::= ab$
 $\quad | A$



Automate LR₁ : Un non-terminal différent pour chaque follow

On raffine la grammaire en dupliquant les non-terminaux

- Pour chaque $N \in \mathcal{N}$ et chaque $t \in \text{Follow}(N)$ on crée N_t ,
- Les règles de N_t sont celles de N , mais démultipliées lorsqu'il y a des non-terminaux
- Après ça on calcul l'automate SLR sur cette nouvelle grammaire
- Puis on efface les annotations partout.

$$S ::= Ab$$

$$| bA$$

$$A ::= a$$

$$B ::= ab$$

$$| A$$

devient

$$S_{\$} ::= A_b b$$

$$| bB_{\$}$$

$$A_{\$} ::= a$$

$$A_b ::= a$$

$$B_{\$} ::= ab$$

$$| A_{\$}$$

Les parseurs LR₁ sont très gros

$$S ::= AbB \\ | bAB$$

$$A ::= Ba$$

$$B ::= aS \\ | bA$$

devient

$$S_{\S} ::= A_b b \\ | bA_a B_{\S} \\ | bA_b B_{\S}$$

$$A_{\S} ::= a$$

$$A_a ::= a$$

$$B_{\S} ::= ab \\ | bA_{\S}$$

$$S_a ::= A_b b \\ | bA_a B_a \\ | bA_b B_a$$

$$A_b ::= a$$

$$B_a ::= ab \\ | bA_a$$

La taille de la grammaire final est en T^P

où T est le nombre de terminaux et p le nombre maximum de non-terminaux dans une règle... et il faut encore calculer l'automate résultant.

Utilisé dans les générateurs modernes

Version minimisée

Il est possible de faire la minimisation de l'automate à la volée, ce qui réduit drastiquement la taille de l'automate et le temps de calcul.

Les outils les plus modernes qui utilise le parsing LR sont en LR₁.
On ne les utilise pas car ils sont un peu plus complexe à installer.

LR_k

Il est possible de faire de même mais avec un “look-ahead” de taille k arbitraire, c'est à dire en décomposant selon les mots de k terminaux pouvant suivre le non terminal (comme les follows, mais avec plus de lettre).

C'est encore plus gros et très rapidement inutilisable.

Universalité de LR₁

Toute grammaire LR_k peut être modifiée automatiquement pour devenir une (grosse) grammaire LR₁.

Toute grammaire non-ambigüe peut se réécrire (non constructivement) en une grammaire LR₁

Taille de SLR mais expressivité proche de LR1

Principe

- On crée une première fois l'automate pseudo-déterministe SLR,
- on repère les différentes utilisations des non-terminaux,
- on duplique les non-terminaux utilisés différemment,
- on recalcule SLR sur la nouvelle grammaire,
- on efface les annotations.

Plus de précisions dans le poly.

Voire son parseur en pratique

En Bison

- Compilez avec l'option `-v`,
- ouvrez le document `y.output`,
- en compilant avec `-g` vous pouvez aussi avoir le graphe (en format `.dot`)

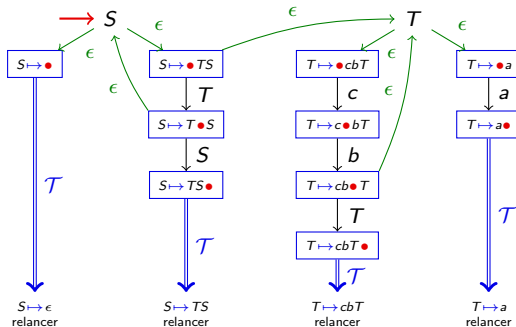
En CamelYacc

- Compilez avec l'option `-v`
- ouvrez le document `y.output`

Et en Java ???

Idée : Exécute l'automate non déterministe avec un oracle pour les epsilon-transition

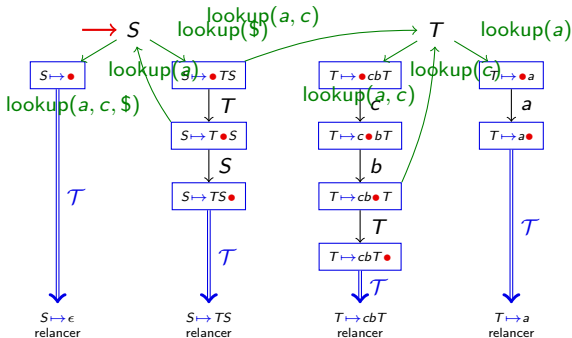
- remplacer les ϵ -transition par des look-ahead (on regarde la lettre suivante sans la consommer),
- utiliser les first/follow pour ces look-ahead (comme SLR).



Et en Java ???

Idée : Exécute l'automate non déterministe avec un oracle pour les epsilon-transition

- remplacer les ϵ -transition par des look-ahead (on regarde la lettre suivante sans la consommer),
- utiliser les first/follow pour ces look-ahead (comme SLR).



Plus techniquement

Changement de paradigme

Il ne s'agit plus d'un automate mais de fonctions s'appelant récursivement (une pour chaque non terminal).

Principe de LL₁

La fonction de parsing du non-terminal N va

- regarder le premier terminal,
- “deviner” quel règle est utilisée,
- appeler successivement les parseurs pour les éléments du patterns.

(Pseudo-)code généré en TP (exo 1)

```
public void expression() { terme();
    while (true) { switch (word.peak()) {
        case '+' : word.pop();term();
        case '-' : word.pop();term();
        case '\n', ')' : return;
        default : raise ParseError;
    } } }
public void term() { facteur();
    while (true) { switch (word.peak()) {
        case '*' : word.pop();facteur();
        case '\n', '+', '-', ')' : return;
        default : raise ParseError;
    } } }
public void facteur() {
    switch (word.peak()) {
        case '(' : word.pop();expression(); assert(word.pop()=='
        case '-' : word.pop();facteur();
        case <NOMBRE> : word.pop();
        default : raise ParseError;
```

Deviner grâce aux Firsts et Follows

Rappel : en Java, un non-terminal est associé à une regexp de terminaux+NT

Sur une disjonction de cas

Les différents "cas" correspondent aux Firsts des différents patterns

ex : $\text{First}(+\langle\text{term}\rangle) = \{+\}$

Sur une étoile de Kleen (non-terminal récursif)

Idem plus un cas de plus qui sort de la boucle à l'aide des Follows.

ex : $\text{Follow}(\langle\text{term}\rangle) = \{\backslash n, +, -,)\}$

Conflit

S'il y a deux fois le même "cas" dans un switch alors la grammaire n'est pas LL₁

Les limites des grammaires LL

On ne dispose que d'un caractère pour deviner sa généalogie

Cela impose d'avoir des "couches" de structures par défaut, comme dans le TP.

LL_k

On peut augmenter le "lookahead" et regarder k caractères au lieu d'un. Contrairement à LR_k , cela n'explode pas la taille du code, mais c'est rarement très utile.

La récursion gauche est impossible

Il y a plusieurs situations où il faudrait un k non bornée, comme la récursivité gauche $S := Sa|b$.

Pourquoi ne pas utiliser un algorithme unique

L'ambiguïté d'une grammaire est indecidable

Elle est uniquement récursivement énumérable, c'est à dire que l'on peut chercher une preuve d'ambiguïté.

La preuve est une réduction directe du problème de correspondance de Post.

Si on veut traiter toutes les grammaires non-ambigües, il faut traiter toutes les grammaires hors-contextes.

Chart parsers :

GLR, Earley, CYK, Packrat...

Ce sont des parseurs “universels” : ils parsent toutes les grammaires hors-contexte, et fournissent potentiellement plusieurs arbres (si elles sont ambiguës)

Complexité en $o(n^{2+\epsilon})$

Peut être rendu linéaire si la grammaire entrée est non-ambigüe mais la constante est grosse.

Complexité semblable à la multiplication de matrice

Utilisé en linguistique

Les grammaires “naturelles” sont toujours ambiguës

Générateurs de parseurs en patates

