

Théorie des langages : THL

CM 7

Uli Fahrenberg

EPITA Rennes

S5 2023

Aperçu

Programme du cours

- ① Langages rationnels, automates finis
- ③ Langages algébriques, grammaires hors-contexte, automates à pile
 - TP 1 : flex
 - QCM 1 : langages rationnels
- ⑤ Parsage LL
- ⑦ Parsage LR, partie 1
 - TP 2 : parsage LL
- ⑧ **Parsage LR, partie 2**
 - QCM 2 : parsage LL
- ⑨ Parsage LR, partie 3
- ⑩ Introduction flex & bison
- ⑪ TP 3, 4 : flex & bison

Re : passage ascendant : the basics

```
function BULRP( $\alpha$ )  
  if  $\alpha = S$  then  
    return True  
  for  $i \leftarrow 1$  to  $|\alpha|$  do  
    for  $j \leftarrow i$  to  $|\alpha|$  do ▷ décalage / SHIFT  
      for  $A \in N$  do  
        if  $A \rightarrow \alpha_i \dots \alpha_j$  then ▷ réduction / REDUCE  
          return BULRP( $\alpha_1 \dots \alpha_{i-1} A \alpha_{j+1} \dots \alpha_n$ )  
  return False
```

Définition (8.8)

Soit G une grammaire hors-contexte. Une **production pointée** de G est une paire $(A, \alpha \bullet \beta)$ telle que $A \rightarrow \alpha \beta$ est une production de G .

Re : automate de parsage LR(0)

Définition (8.10)

Soit G une grammaire hc et \mathcal{I} un ensemble de productions pointées de G . La **clôture** de \mathcal{I} est le plus petit ensemble $\text{cl}(\mathcal{I})$ t.q. $\mathcal{I} \subseteq \text{cl}(\mathcal{I})$ et

- si $(A, \alpha \bullet B \beta) \in \text{cl}(\mathcal{I})$ et $B \rightarrow \gamma$ est une production de G , alors $(B, \bullet \gamma) \in \mathcal{I}$.

Définition

L'**automate de parsage LR(0)** d'une grammaire hors-contexte G est l'automate fini déterministe (Q, q_0, F, δ) avec

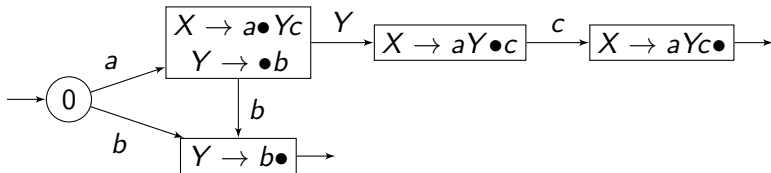
- $Q = \{\text{cl}(\mathcal{I}) \mid \mathcal{I} \text{ ensemble de productions pointées de } G\}$;
- $q_0 = \text{cl}(\{(Z, \bullet S \$)\})$;
- $F = \{q \in Q \mid \exists \text{ production } X \rightarrow w \text{ de } G \text{ t.q. } (X, w \bullet) \in q\}$
- et $\delta : Q \times V \rightarrow Q$ donnée par

$$\delta(q, \beta) = \text{cl}(\{(X, \alpha \beta \bullet \gamma) \mid (X, \alpha \bullet \beta \gamma) \in q\}).$$

Re : exemple

$$X \rightarrow aYc \quad (1)$$

$$Y \rightarrow b \quad (2)$$



Dans le poly

- ① Langages rationnels 2.1, 2.2, 2.3.1, 2.4, 3.1.1, 3.1.2, 3.2
- ② Automates finis 4.1, 4.2.2
- ③ Langages algébriques, grammaires hors-contexte, automates à pile
5.1, 5.2.3, 5.2.4, 5.3.6, 6.2, plus Sipser 2.2
- ④ Parsage LL 7, 8.1
- ⑤ Parsage LR 8.2

Parsage LR(0)

Algorithme de parsing

- ① empiler q_0
- ② repeat
 - ① $q \leftarrow$ état en haut de la pile
 - ② si $q =$ état final $X \rightarrow w\bullet$:
 - ① dépiler $|w|$ états
 - ② $q' \leftarrow$ état en haut de la pile
 - ③ empiler $\delta(q', X)$
 - ③ sinon :
 - ① $a \leftarrow$ next(input)
 - ② empiler $\delta(q, a)$
- ③ until $q =$ état final $Z \rightarrow S\$$ (✓) ou échec (✗)

REDUCE

SHIFT

Algorithme de parsing

- ① empiler q_0
- ② repeat
 - ① $q \leftarrow$ état en haut de la pile
 - ② si $q =$ état final $X \rightarrow w\bullet$:
 - ① dépiler $|w|$ états
 - ② $q' \leftarrow$ état en haut de la pile
 - ③ empiler $\delta(q', X)$ ← possible **X**
 - ③ sinon :
 - ① $a \leftarrow$ next(input) ← possible **X**
 - ② empiler $\delta(q, a)$ ← possible **X**
- ③ until $q =$ état final $Z \rightarrow S\$ \bullet$ (✓) ou échec (✗)

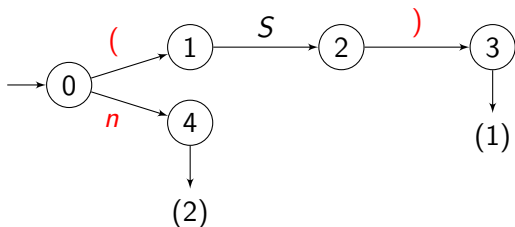
REDUCE

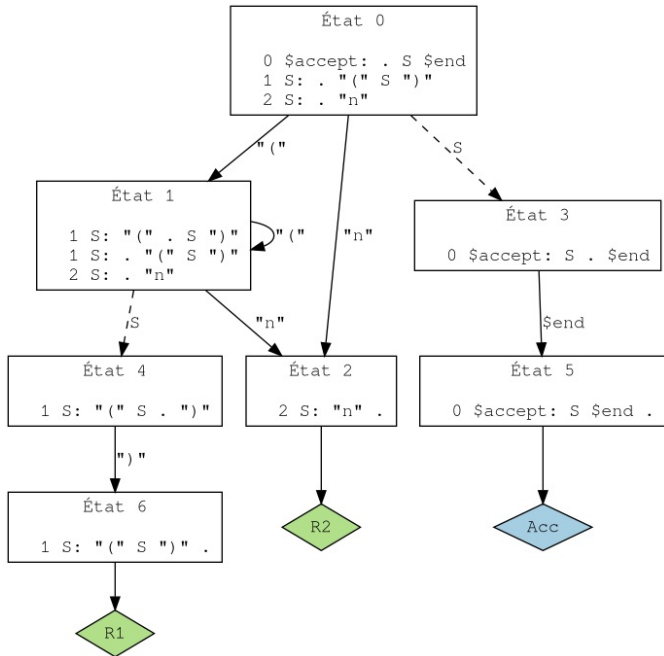
SHIFT

Exemple

$$S \rightarrow (S) \quad (1)$$

$$| n \quad (2)$$



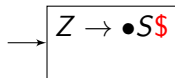


Exemple : automate de parsing

$$Z \rightarrow S\$ \quad (0)$$

$$S \rightarrow (S) \quad (1)$$

$$| n \quad (2)$$

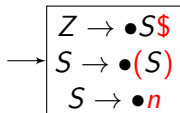


Exemple : automate de parsing

$$Z \rightarrow S\$ \quad (0)$$

$$S \rightarrow (S) \quad (1)$$

$$| n \quad (2)$$

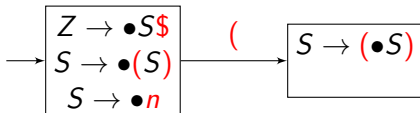


Exemple : automate de parsing

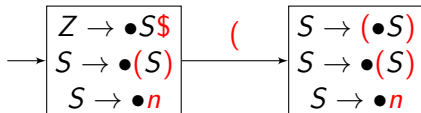
$$Z \rightarrow S\$ \quad (0)$$

$$S \rightarrow (S) \quad (1)$$

$$| n \quad (2)$$



Exemple : automate de parsing

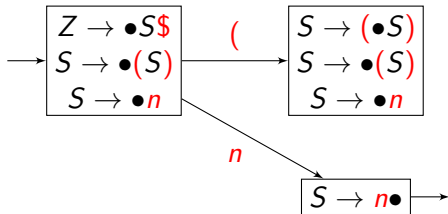
 $Z \rightarrow S\$ \quad (0)$ $S \rightarrow (S) \quad (1)$ $| n \quad (2)$ 

Exemple : automate de parsage

$$Z \rightarrow S\$ \quad (0)$$

$$S \rightarrow (S) \quad (1)$$

$$| n \quad (2)$$

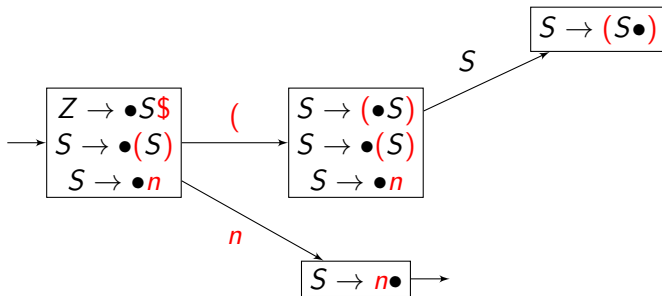


Exemple : automate de parse

$$Z \rightarrow S\$ \quad (0)$$

$$S \rightarrow (S) \quad (1)$$

$$| n \quad (2)$$

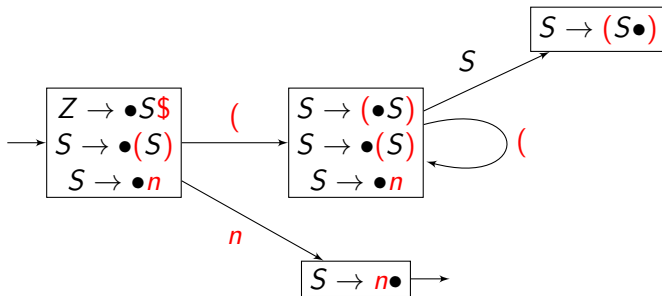


Exemple : automate de parse

$$Z \rightarrow S\$ \quad (0)$$

$$S \rightarrow (S) \quad (1)$$

$$| n \quad (2)$$

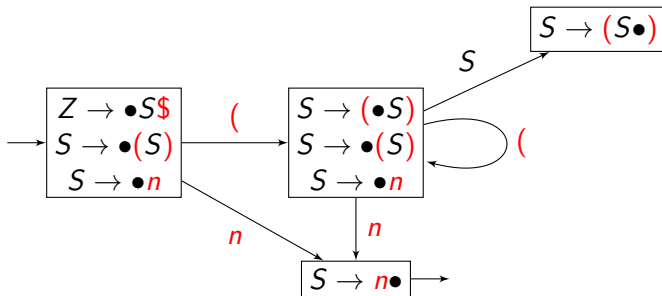


Exemple : automate de parse

$$Z \rightarrow S\$ \quad (0)$$

$$S \rightarrow (S) \quad (1)$$

$$| n \quad (2)$$

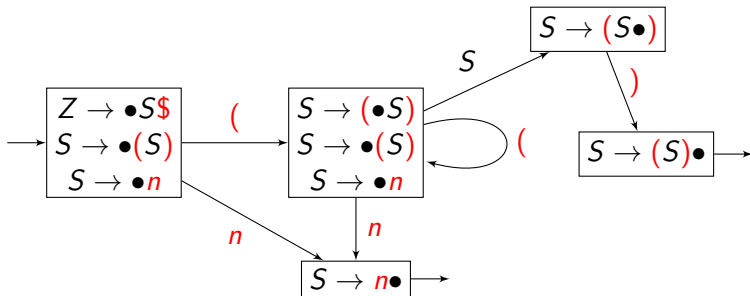


Exemple : automate de parse

$$Z \rightarrow S\$ \quad (0)$$

$$S \rightarrow (S) \quad (1)$$

$$| n \quad (2)$$

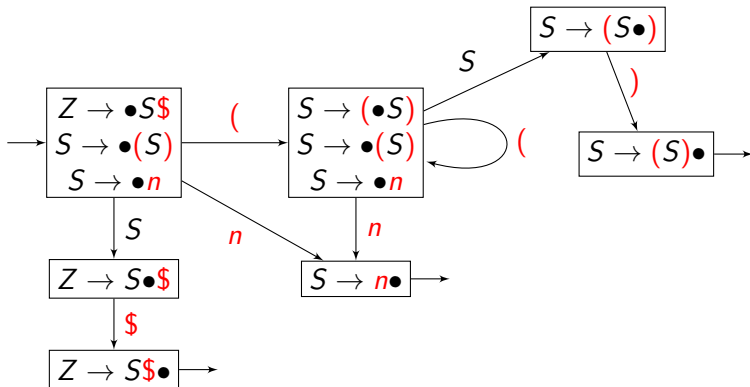


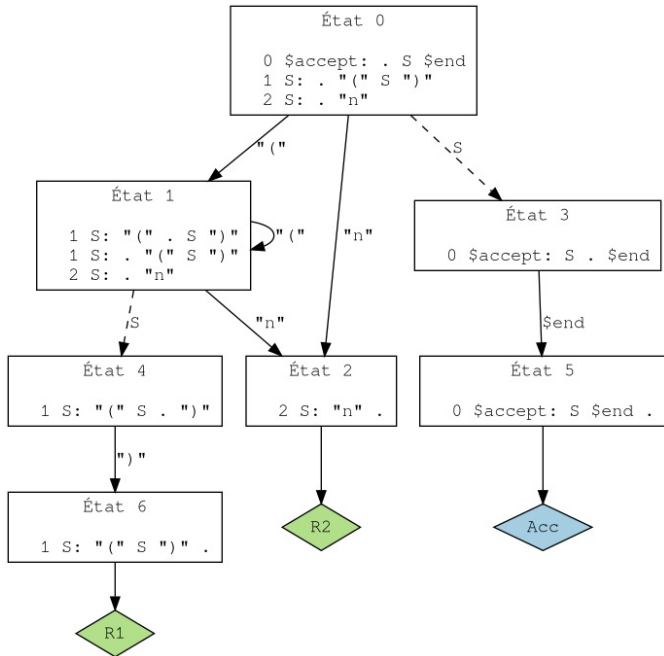
Exemple : automate de parsage

$$Z \rightarrow S\$ \quad (0)$$

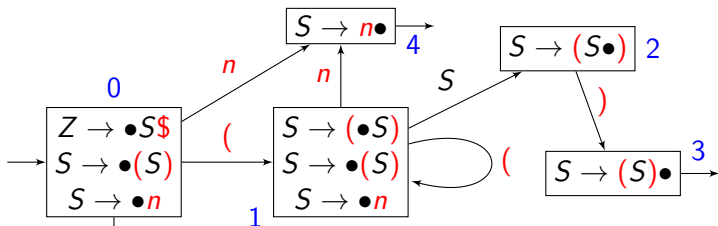
$$S \rightarrow (S) \quad (1)$$

$$| n \quad (2)$$





Exemple : table de parse

 $Z \rightarrow S\$$ (0) $S \rightarrow (S)$ (1) $| n$ (2)

state	action	(n)	\$	S
0	shift	1	4			5
1	shift	1	4			2
2	shift			3		
3	reduce 1					
4	reduce 2					
5	shift				6	
6	accept					

Autre exemple

$Z \rightarrow S\$$ (0)
 $S \rightarrow S-n$ (1)
 $\quad | n$ (2)

état	productions pointées
0	$Z \rightarrow \bullet S \$, S \rightarrow \bullet S - n, S \rightarrow \bullet n$
1	$S \rightarrow n \bullet$
2	$Z \rightarrow S \bullet \$, S \rightarrow S \bullet - n$
3	$Z \rightarrow S \$ \bullet$
4	$S \rightarrow S - \bullet n$
5	$S \rightarrow S - n \bullet$

état	action	n	$-$	$\$$	S
0	décaler	1			2
1	réduire 2				
2	décaler		4	3	
3	accepter				
4	décaler	5			
5	réduire 1				

Autre exemple

$$Z \rightarrow S\$ \quad (0)$$

$$S \rightarrow S-n \quad (1)$$

$$| n \quad (2)$$

état	action	n	$-$	$\$$	S
0	décaler	1			2
1	réduire 2				
2	décaler		4	3	
3	accepter				
4	décaler	5			
5	réduire 1				

Autre exemple

$$Z \rightarrow S\$ \quad (0)$$

$$S \rightarrow S-n \quad (1)$$

$$| n \quad (2)$$

parser $n - n\$$:

entrée	pile	action
$n - n\$$	$\perp 0$	décaler
$-n\$$	$\perp 01$	réduire 2
$-n\$$	$\perp 02$	décaler
$n\$$	$\perp 024$	décaler
$\$$	$\perp 0245$	réduire 1
$\$$	$\perp 02$	décaler
	$\perp 023$	✓

état	action	n	$-$	$\$$	S
0	décaler	1			2
1	réduire 2				
2	décaler		4	3	
3	accepter				
4	décaler	5			
5	réduire 1				

Autre exemple

$$Z \rightarrow S\$ \quad (0)$$

$$S \rightarrow S-n \quad (1)$$

$$| n \quad (2)$$

parser $n - n\$$:

entrée	pile	action
$n - n\$$	$\perp 0$	décaler
$-n\$$	$\perp 01$	réduire 2 $S \rightarrow n$
$-n\$$	$\perp 02$	décaler
$n\$$	$\perp 024$	décaler
$\$$	$\perp 0245$	réduire 1 $S \rightarrow S-n$
$\$$	$\perp 02$	décaler
	$\perp 023$	✓

état	action	n	$-$	$\$$	S
0	décaler	1			2
1	réduire 2				
2	décaler		4	3	
3	accepter				
4	décaler	5			
5	réduire 1				

Autre exemple

$$Z \rightarrow S\$ \quad (0)$$

$$S \rightarrow S-n \quad (1)$$

$$| n \quad (2)$$

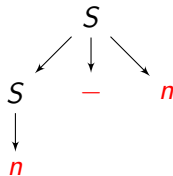
parser $n - n\$$:

entrée	pile	action
$n - n\$$	$\perp 0$	décaler
$-n\$$	$\perp 01$	réduire 2
$-n\$$	$\perp 02$	décaler
$n\$$	$\perp 024$	décaler
$\$$	$\perp 0245$	réduire 1
$\$$	$\perp 02$	décaler
	$\perp 023$	✓

état	action	n	$-$	$\$$	S
0	décaler	1			2
1	réduire 2				
2	décaler		4	3	
3	accepter				
4	décaler	5			
5	réduire 1				

$$S \rightarrow n$$

$$S \rightarrow S-n$$



Parsage LR(0)

- lire l'entrée de gauche à droite (**L**)
- approche ascendant
- construire une dérivation droite (**R**)
- pas de regard avant (**0**)

Parsage SLR(1)

Encore un exemple

 $Z \rightarrow S\$$ (0) $S \rightarrow n-S$ (1) $| n$ (2)

état	productions pointées
0	$Z \rightarrow \bullet S \$$, $S \rightarrow \bullet n - S$, $S \rightarrow \bullet n$
1	$Z \rightarrow S \bullet \$$
2	$S \rightarrow n \bullet - S$, $S \rightarrow n \bullet$
3	$S \rightarrow n - \bullet S$, $S \rightarrow \bullet n - S$, $S \rightarrow \bullet n$
4	$Z \rightarrow S \$ \bullet$
5	$S \rightarrow n - S \bullet$

état	action	n	$-$	$\$$	S
0	décaler	2			1
1	décaler			4	
2	réduire 2, décaler		3		
3	décaler	2			5
4	accepter				
5	réduire 1				

Encore un exemple

 $Z \rightarrow S\$$ (0) $S \rightarrow n-S$ (1) $| n$ (2)

état	productions pointées
0	$Z \rightarrow \bullet S \$$, $S \rightarrow \bullet n - S$, $S \rightarrow \bullet n$
1	$Z \rightarrow S \bullet \$$
2	$S \rightarrow n \bullet - S$, $S \rightarrow n \bullet$
3	$S \rightarrow n - \bullet S$, $S \rightarrow \bullet n - S$, $S \rightarrow \bullet n$
4	$Z \rightarrow S \$ \bullet$
5	$S \rightarrow n - S \bullet$

état	action	n	$-$	$\$$	S
0	décaler	2			1
1	décaler			4	
2	réduire 2, décaler		3		conflit SHIFT/REDUCE
3	décaler	2			5
4	accepter				
5	réduire 1				

Conflits

- **SHIFT/REDUCE**

- faut **réduire** avec $Y \rightarrow u$
- ou **décaler** en attendant v ?

$$X \rightarrow u \bullet v$$

$$Y \rightarrow u \bullet$$

- **REDUCE/REDUCE**

- faut **réduire** avec $X \rightarrow u$
- ou **réduire** avec $Y \rightarrow u$?

$$X \rightarrow u$$

$$Y \rightarrow u$$

- utiliser **FOLLOW** pour résoudre

- (et pourquoi des conflits SHIFT/SHIFT n'existent pas ?)

Re : FOLLOW

Calculer des terminaux qui peuvent **suivre** un symbole dans une dérivation :

Définition

Soit $x \in V$, alors $\text{FOLLOW}(x) \subseteq \Sigma$ est défini par

$$\text{FOLLOW}(x) = \{a \in \Sigma \mid \exists B \in N, \alpha, \beta \in V^* : B \Rightarrow^* \alpha x a \beta\}.$$

Algorithme :

- ① pour chaque $x \in V$: $\text{FOLLOW}(x) = \emptyset$
- ② répéter jusqu'au point fixe :
 - ① pour chaque $B \rightarrow \alpha x \beta \gamma$ avec $\beta \in \text{NULL}^*$:
 - ① si $\gamma \notin \text{NULL}^*$: $\text{FOLLOW}(x) += \text{FIRST}(\gamma)$
 - ② si $\gamma \in \text{NULL}^*$: $\text{FOLLOW}(x) += \text{FOLLOW}(B)$

Simple LR(1)

- calculer la table LR(0)
- si conflits : **conditionner** l'action par le FOLLOW

Exemple : $Z \rightarrow S\$$ (0)
 $S \rightarrow n-S$ (1)
 $\quad \quad | n$ (2)

état	action	n	$-$	$\$$	S		état	n	$-$	$\$$	S
0	décaler	2			1	\Rightarrow	0	d.2			d.1
1	décaler			4			1			d.4	
2	réd. 2, déc.		3				2		d.3	r.2	
3	décaler	2			5		3	d.2			d.5
4	accepter						4	— accepter —			
5	réduire 1						5			r.1	

Simple LR(1)

- 1 calculer la table LR(0)
- 2 si conflits : **conditionner** l'action par le FOLLOW
- 3 passer du type état \rightarrow action \rightarrow entrée
au type état \rightarrow entrée \rightarrow action

The image features a central graphic consisting of several concentric circles. The innermost circle is a solid dark blue. Surrounding it are several rings of varying shades of red, from a deep, dark red to a bright, vibrant red. The text "That's all Folks!" is written in a white, elegant cursive font, slanted slightly upwards from left to right, and is positioned across the middle of the concentric circles.

That's all Folks!