Théorie des Langages

Analyse syntaxique ascendante

Claude Moulin

Université de Technologie de Compiègne

Printemps 2013

Sommaire

Principe

- L'analyse ascendante ou par décalage réduction a pour but de construire un arbre d'analyse à partir de ses feuilles pour une chaîne d'entrée donnée w.
- La racine est construite après que tous les nœuds inférieurs ont été construits.
- A chaque étape de réduction une sous-chaîne correspondant à la partie droite d'une règle de production est remplacée par la variable de la partie gauche.

Exemple

1. $S \longrightarrow aSbT$
2. $S \longrightarrow cT$
3. $S \longrightarrow d$
4. $T \longrightarrow aT$
5. $T \longrightarrow bS$
6. $T \longrightarrow c$

$accbbadbc$
$acTbbadbc$
$aSbbadbc$
$aSbbaSbc$
$aSbbaSbT$
$aSbbS$
$aSbT$
S

Illustration

Pile	Reste	Opération	Pile	Reste	Opération
	accbbadbc	d	aSbbad	bc	r
a	ccbbadbc	d	aSbbaS	bc	d
ac	cbbadbc	d	aSbbaSb	c	d
acc	bbadbc	r	aSbbaSbc		r
acT	bbadbc	r	aSbbaSbT		r
aS	bbadbc	d	aSbbS		r
aSb	badbc	d	aSbT		r
aSbb	adbc	d	S		r
aSbba	dbc	d			

Dérivation :

$S \Longrightarrow aSbT \Longrightarrow aSbbS \Longrightarrow aSbbaSbT \Longrightarrow aSbbaSbc$

$\Longrightarrow aSbbadbc \Longrightarrow acTbbadbc \Longrightarrow accbbadbc$

Méthodes

- LR(0) : importante d'un point de vue théorique mais trop faible en pratique
- SLR(1) : version renforcée de LR(0) mais assez faible
- LR(1) : très puissante mais nécessite une mémoire importante
- LALR(1) : version affaiblie de LR(1) puissante et exécutable

Manche

- Un manche est une suite de nœuds correspondant aux fils d'un nœud supérieur dans l'arbre de dérivation.
- Un manche d'un syntagme $\gamma = \alpha\beta w$ est une sous-chaîne β correspondant à un choix d'une règle de production $A \longrightarrow \beta$

Si $S \underset{rm}{\overset{*}{\Longrightarrow}} \alpha A w \underset{rm}{\Longrightarrow} \alpha\beta w$ alors β est un manche de $\alpha\beta w$

1. $T \longrightarrow c$ est le manche de *accbbadbc* (position 3).
2. $S \longrightarrow cT$ est le manche de *acTbbadbc* (position 2).
3. $S \longrightarrow aSbT$ est le manche de *aSbbaSbT* (position 5).

Préfixe viable

Les préfixes viables sont les préfixes des syntagmes qui ne s'étendent pas au-delà de l'extrémité droite du manche le plus à droite. Ces préfixes sont ceux que l'on peut trouver sur la pile d'un analyseur.

Lors de l'analyse de l'exemple précédent, le syntagme $aSbb\underline{a}dbc$ est considéré lorsque le buffer d'entrée ne contient plus que $adbc$. le prochain manche est $S \longrightarrow d$. Les symboles a, S, b, b constituent un préfixe viable et sont sur la pile de l'analyseur.

Sommaire

Item LR(0)

Les items LR(0) ont la forme $A \longrightarrow \alpha_1 \cdot \alpha_2$ où $A \longrightarrow \alpha_1\alpha_2$ est une règle de production.

La forme $A \longrightarrow \alpha_1 \cdot \alpha_2$ représente l'hypothèse que $\alpha_1\alpha_2$ est un manche possible et qu'il sera réduit à A.

Le point représente l'avancement de l'analyseur quand il lit la règle de gauche à droite : α_1 a déjà été reconnu et α_2 doit encore être analysé lorsque la production est utilisée.

La forme $A \longrightarrow \alpha_1\alpha_2 \cdot$ signifie que le pivot est atteint puisque α_1 et α_2 ont été reconnus.

Exemple

$S \longrightarrow aSbT$ fournit 5 items :

$S \longrightarrow \cdot\, aSbT$
$S \longrightarrow a\cdot\, SbT$
$S \longrightarrow aS\cdot\, bT$
$S \longrightarrow aSb\cdot\, T$
$S \longrightarrow aSbT\cdot$

Une règle telle que $A \longrightarrow \epsilon$ ne donne qu'un seul item : $A \longrightarrow \cdot$.

Fermeture d'un ensemble d'items

- Soit *I* un ensemble d'items pour une grammaire *G*
- Fermeture(I) est l'ensemble des items construit par l'algorithme suivant :
  - *Placer chaque item de I dans Fermeture(I)*
    Si $A \longrightarrow \alpha_1 \cdot B\alpha_2 \in$ *Fermeture(I) et*
    $B \longrightarrow \gamma$ *est une production de G*
    ajouter $B \longrightarrow \cdot\, \gamma$ *dans Fermeture(I)*
    Recommencer tant qu'un nouvel item est ajouté

Exemples

1. Si $I = \{S \longrightarrow a \cdot SbT\}$
   Fermeture(I) =
   $\{S \longrightarrow a \cdot SbT, S \longrightarrow \cdot aSbT, S \longrightarrow \cdot cT, S \longrightarrow \cdot d\}$
2. Si $I = \{S \longrightarrow aS \cdot bT\}$
   Fermeture(I) = $\{S \longrightarrow aS \cdot bT\}$
3. Si $I = \{S \longrightarrow aSb \cdot T\}$
   Fermeture(I) =
   $\{S \longrightarrow aSb \cdot T, T \longrightarrow \cdot aT, T \longrightarrow \cdot bS, T \longrightarrow \cdot c\}$

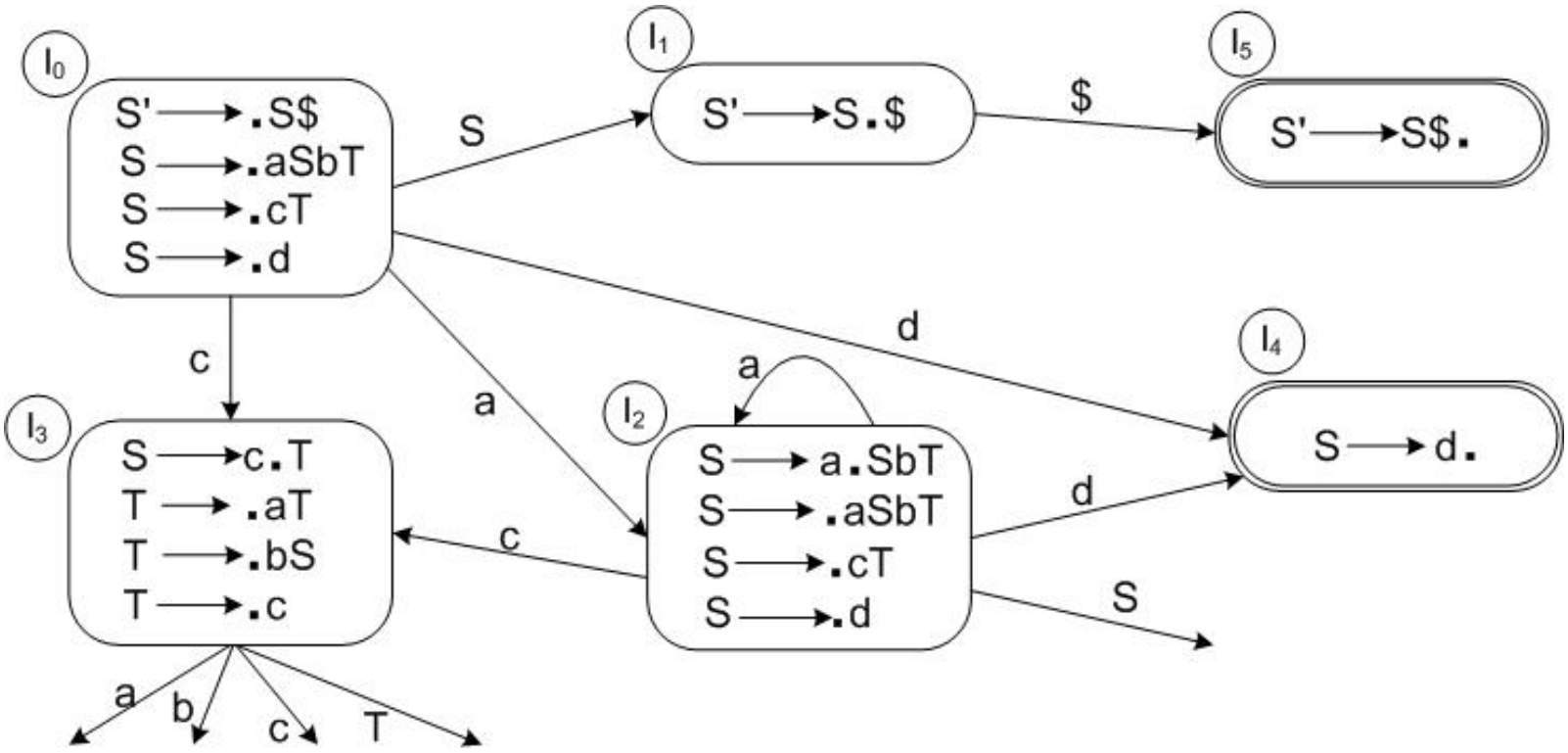
Automate caractéristique canonique

- Son alphabet est $V \cup T$.
- Cet automate fini déterministe reflète les décisions que prend l'analyseur lorsqu'il choisit une règle à appliquer.
- Analyser B dans $A \longrightarrow \alpha_1 \cdot B\alpha_2$ se traduit par l'analyse d'une règle issue de B.
- Chaque fermeture d'ensemble d'items forme un état de l'automate.
- Les items de I, tels que $S \longrightarrow aSb \cdot T$ forment le noyau de *Fermeture(I)*.

Transitions

- La fonction de transition détermine l'état successeur d'un état donné après l'analyse d'un symbole terminal ou non.
- Transition(I,X), I est un ensemble d'items ; $X \in T \cup V$
  - *Pour chaque item* $A \longrightarrow \alpha_1 \cdot X \alpha_2$ *de I, déterminer l'item* $A \longrightarrow \alpha_1 X \cdot \alpha_2$.
  - La fermeture de l'ensemble de ces items détermine Transition(I,X).

Représentation



Collection des items

1. Ajouter l'axiome S' et la production $S' \longrightarrow S\$$
2. $I_0 \longleftarrow$ Fermeture($\{S' \longrightarrow \cdot\, S\$\}$)
3. $C \longleftarrow \{I_0\}$
4. Pour chaque $I \in C$
   Pour chaque $X \in V \cup T$ tel que
   Transition(I,X) $\neq \emptyset$
   Ajouter Transition(I,X) à C
5. Recommencer tant que C n'est pas stable

Construction - 1

- $S \longrightarrow aSbT\,;\, S \longrightarrow cT\,;\, S \longrightarrow d$
- $T \longrightarrow aT\,;\, T \longrightarrow bS\,;\, T \longrightarrow c$

$$I_0 = \{S' \longrightarrow \cdot\, S\$, S \longrightarrow \cdot\, aSbT, S \longrightarrow \cdot\, cT, S \longrightarrow \cdot\, d\}$$

$$\begin{aligned}
I_1 &= \delta(I_0, S) = \{S' \longrightarrow S\cdot\,\$\} \\
I_2 &= \delta(I_0, a) = \{S \longrightarrow a\cdot\, SbT, S \longrightarrow \cdot\, aSbT, \\
&\qquad S \longrightarrow \cdot\, cT, S \longrightarrow \cdot\, d\} \\
I_3 &= \delta(I_0, c) = \{S \longrightarrow c\cdot\, T, T \longrightarrow \cdot\, aT, \\
&\qquad T \longrightarrow \cdot\, bS, T \longrightarrow \cdot\, c\} \\
I_4 &= \delta(I_0, d) = \{S \longrightarrow d\cdot\,\}
\end{aligned}$$

Construction - 2

$$
\begin{array}{lll}
I_1 = & \delta(I_0, S) = & \{S' \longrightarrow S\cdot \$\} \\
I_2 = & \delta(I_0, a) = & \{S \longrightarrow a\cdot SbT, S \longrightarrow\cdot aSbT, \\
 & & S \longrightarrow\cdot cT, S \longrightarrow\cdot d\}
\end{array}
$$

$$
\begin{array}{lll}
I_5 = & \delta(I_1, \$) = & \{S' \longrightarrow S\$\cdot\} \\
I_6 = & \delta(I_2, S) = & \{S \longrightarrow aS\cdot bT\} \\
 & \delta(I_2, a) = & \{S \longrightarrow a\cdot SbT, S \longrightarrow\cdot aSbT, \\
 & & S \longrightarrow\cdot cT, S \longrightarrow\cdot d\} = I_2 \\
 & \delta(I_2, c) = & \{S \longrightarrow c\cdot T, T \longrightarrow\cdot aT, \\
 & & T \longrightarrow\cdot bS, T \longrightarrow\cdot c\} = I_3 \\
 & \delta(I_2, d) = & \{S \longrightarrow d\cdot\} = I_4
\end{array}
$$

Construction - 3

$$I_3 = \quad \delta(I_0, c) = \quad \{S \longrightarrow c\cdot T, T \longrightarrow\cdot aT, \; T \longrightarrow\cdot bS, T \longrightarrow\cdot c\}$$

$$I_7 = \quad \delta(I_3, T) = \quad \{S \longrightarrow cT\cdot\}$$

$$I_8 = \quad \delta(I_3, a) = \quad \{T \longrightarrow a\cdot T, T \longrightarrow\cdot aT, \; T \longrightarrow\cdot bS, T \longrightarrow\cdot c\}$$

$$I_9 = \quad \delta(I_3, b) = \quad \{T \longrightarrow b\cdot S, S \longrightarrow\cdot aSbT, \; S \longrightarrow\cdot cT, S \longrightarrow\cdot d\}$$

$$I_{10} = \quad \delta(I_3, c) = \quad \{T \longrightarrow c\cdot\}$$

Construction - 4

$$I_6 = \quad \delta(I_2, S) = \quad \{S \longrightarrow aS \cdot bT\}$$

$$I_{11} = \quad \delta(I_6, b) = \quad \{S \longrightarrow aSb \cdot T, T \longrightarrow \cdot aT, \\ T \longrightarrow \cdot bS, T \longrightarrow \cdot c\}$$

Construction - 5

$$I_8 = \quad \delta(I_3, a) = \quad \{T \longrightarrow a\cdot T, T \longrightarrow\cdot aT, \; T \longrightarrow\cdot bS, T \longrightarrow\cdot c\}$$

$$\begin{aligned} I_{12} = \quad & \delta(I_8, T) = && \{T \longrightarrow aT\cdot\} && \\ & \delta(I_8, a) = && \{T \longrightarrow a\cdot T, T \longrightarrow\cdot aT, \; T \longrightarrow\cdot bS, T \longrightarrow\cdot c\} && = I_8 \\ & \delta(I_8, b) = && \{T \longrightarrow b\cdot S, S \longrightarrow\cdot aSbT, \; S \longrightarrow\cdot cT, S \longrightarrow\cdot d\} && = I_9 \\ & \delta(I_8, c) = && \{T \longrightarrow c\cdot\} && = I_{10} \end{aligned}$$

Construction - 6

$$I_9 = \quad \delta(I_3, b) = \quad \{T \longrightarrow b\cdot S, S \longrightarrow \cdot\, aSbT, \; S \longrightarrow \cdot\, cT, S \longrightarrow \cdot\, d\}$$

$$\begin{aligned} I_{13} = \quad \delta(I_9, S) &= \{T \longrightarrow bS\cdot\} \\ \delta(I_9, a) &= \{S \longrightarrow a\cdot SbT, S \longrightarrow \cdot\, aSbT, \; S \longrightarrow \cdot\, cT, S \longrightarrow \cdot\, d\} = I_2 \\ \delta(I_9, c) &= \{S \longrightarrow c\cdot T, T \longrightarrow \cdot\, aT, \; T \longrightarrow \cdot\, bS, T \longrightarrow \cdot\, c\} = I_3 \\ \delta(I_9, d) &= \{S \longrightarrow d\cdot\} = I_4 \end{aligned}$$

Construction - 7

$$I_{11} = \quad \delta(I_6, b) = \quad \{S \longrightarrow aSb\cdot T, T \longrightarrow\cdot aT, \; T \longrightarrow\cdot bS, T \longrightarrow\cdot c\}$$

$$\begin{aligned}
I_{14} = \quad & \delta(I_{11}, T) = && \{S \longrightarrow aSbT\cdot\} && \\
& \delta(I_{11}, a) = && \{T \longrightarrow a\cdot T, T \longrightarrow\cdot aT, \; T \longrightarrow\cdot bS, T \longrightarrow\cdot c\} && = I_8 \\
& \delta(I_{11}, b) = && \{T \longrightarrow b\cdot S, S \longrightarrow\cdot aSbT, \; S \longrightarrow\cdot cT, S \longrightarrow\cdot d\} && = I_9 \\
& \delta(I_{11}, c) = && \{T \longrightarrow c\cdot\} && = I_{10}
\end{aligned}$$

Table des actions

Pour tout état s

si s contient un item tel que $A \longrightarrow \alpha \cdot a\beta$ *alors*

Action[s] $\leftarrow$ *Décalage*

sinon

si s contient $A \longrightarrow \alpha \cdot$ *(règle i) et* $A \neq S'$ *alors*

Action[s] $\leftarrow$ *Réduction i*

si s contient $S' \longrightarrow S\$ \cdot$ *alors*

Action[s] $\leftarrow$ *Acceptation*

Grammaire LR(0)

Définition

Une grammaire est LR(0) si aucune entrée de la table des actions ne contient plus d'une action.

La table des successeurs n'est qu'une simple table de transition d'un automate. La définition des transitions entre les états permet de remplir cette table.

Tables - 1

Etats	Action	Successeurs						
		a	b	c	d	S	T	$
0	Décalage	2		3	4	1		
1	Décalage							5
2	Décalage	2		3	4	6		
3	Décalage	8	9	10			7	
4	Réduction 3							
5	Acceptation							
6	Décalage		11					
7	Réduction 2							

$$I_0 = \{S' \longrightarrow \cdot\, S\$, S \longrightarrow \cdot\, aSbT, S \longrightarrow \cdot\, cT, S \longrightarrow \cdot\, d\}$$

Tables - 2

Etats	Action	Successeurs						
		a	b	c	d	S	T	$
8	Décalage	8	9	10			12	
9	Décalage	2		3	4	13		
10	Réduction 6							
11	Décalage	8	9	10			14	
12	Réduction 4							
13	Réduction 5							
14	Réduction 1							

Analyse - Décalage

Si Action[s] = Décalage

$(s, ax, s\gamma, y) \vdash (s', x, s'as\gamma, y)$
avec s' = successeur[s,a]

L'automate :

- consomme le caractère d'entrée (a),
- empile le caractère et le successeur de l'état actuel
- se positionne dans l'état successeur s'

Analyse - Réduction

Si Action[s] = Réduction j

Règle j : $A \longrightarrow \alpha$ et $\alpha = X_1 X_2 \cdots X_n$

$(s, x, sX_n \cdots s_2 X_2 s_1 X_1 t\gamma, y) \vdash (s', x, s' A t\gamma, yj)$

avec avec s' = successeur[t,A]

L'automate :

- dépile les symboles correspondant à la partie droite de la règle j, ainsi que les sommets intermédiaires,
- empile le symbole gauche de la règle j et le successeur de l'état apparu sur la pile,
- se positionne dans cet état.

Analyse - Acceptation - Erreur

Si Action[s] = Acceptation
    chaîne syntaxiquement correcte.

Si Sucesseur[s] = $\emptyset$
    erreur

Simulation - chaîne : accbbadbc

Entrée	Pile	Action	Sortie
accbbadbc$	0⊣	D	
ccbbadbc$	2a0⊣	D	
cbbadbc$	3c2a0⊣	D	
bbadbc$	10c3c2a0⊣	R 6	6
bbadbc$	7T3c2a0⊣	R 2	. . . 2
bbadbc$	6S2a0⊣	D	. . . 2
badbc$	11b6S2a0⊣	D	. . . 2
adbc$	9b11b6S2a0⊣	D	. . . 2
dbc$	2a9b11b6S2a0⊣	D	. . . 2

Tables - 1

Etats	Action	Successeurs						
		a	b	c	d	S	T	$
0	Décalage	2		3	4	1		
1	Décalage							5
2	Décalage	2		3	4	6		
3	Décalage	8	9	10			7	
4	Réduction 3							
5	Acceptation							
6	Décalage		11					
7	Réduction 2							
8	Décalage	8	9	10			12	
9	Décalage	2		3	4	13		
10	Réduction 6							
11	Décalage	8	9	10			14	
12	Réduction 4							
13	Réduction 5							
14	Réduction 1							

Simulation - chaîne : accbbadbc

Entrée	Pile	Action	Sortie
dbc$	2a9b11b6S2a0⊣	D	...2
bc$	4d2a9b11b6S2a0⊣	R 3	...3
bc$	6S2a9b11b6S2a0⊣	D	...3
c$	11b6S2a9b11b6S2a0⊣	D	...3
$	10c11b6S2a9b11b6S2a0⊣	R 6	...6
$	14T11b6S2a9b11b6S2a0⊣	R 1	...1
$	13S9b11b6S2a0⊣	R 5	...5
$	14T11b6S2a0⊣	R 1	...1
$	1S0⊣	D	...1
ϵ	5$1S0⊣	A	...1

Conflits

- Conflit Décaler - Réduire
  - Certains états peuvent contenir un item de la forme $A \longrightarrow \alpha \cdot a\beta$ et un item de la forme $A \longrightarrow \alpha \cdot$.
- Conflit Réduire - Réduire
  - Certains états peuvent contenir deux items de la forme $A \longrightarrow \alpha \cdot$, représentant deux situations de réduction par deux productions différentes.

Exercice

1. $S \longrightarrow BA$
2. $A \longrightarrow aA$
3. $A \longrightarrow B$
4. $B \longrightarrow b$

Items

$$\{S' \longrightarrow \cdot\, S\$, S \longrightarrow \cdot\, BA, B \longrightarrow \cdot\, b\} = I_0$$
$$\{S' \longrightarrow S \cdot \$\} = I_1$$
$$\{S \longrightarrow B \cdot A, A \longrightarrow \cdot\, aA, A \longrightarrow \cdot\, B, B \longrightarrow \cdot\, b\} = I_2$$
$$\{B \longrightarrow b \cdot\} = I_3$$
$$\{S' \longrightarrow S\$ \cdot\} = I_4$$
$$\{S \longrightarrow BA \cdot\} = I_5$$
$$\{A \longrightarrow a \cdot A, A \longrightarrow \cdot\, aA, A \longrightarrow \cdot\, B, B \longrightarrow \cdot\, b\} = I_6$$
$$\{A \longrightarrow B \cdot\} = I_7$$
$$\{A \longrightarrow aA \cdot\} = I_8$$

Tables

Etats	Action	Successeurs					
		a	b	A	B	S	$
0	Décalage		3		2	1	
1	Décalage						4
2	Décalage	6	3	5	7		
3	Réduction 4						
4	Acceptation						
5	Réduction 1						
6	Décalage 1	6	3	8	7		
7	Réduction 3						
8	Réduction 2						

Simulation : baaab - 1

Entrée	Pile	Action	Sortie
baaab$	0⊣	D	
aaab$	3b0⊣	R 4	4
aaab$	2B0⊣	D	4
aab$	6a2B0⊣	D	4
ab$	6a6a2B0⊣	D	4
b$	6a6a6a2B0⊣	D	4
$	3b6a6a6a2B0⊣	R 4	4 4
$	7B6a6a6a2B0⊣	R 3	4 4 3
$	8A6a6a6a2B0⊣	R 2	4 4 3 2
$	8A6a6a2B0⊣	R 2	4 4 3 2 2

Simulation : baaab - 2

Entrée	Pile	Action	Sortie
aab$	6a2B0⊣	D	4
ab$	6a6a2B0⊣	D	4
b$	6a6a6a2B0⊣	D	4
$	3b6a6a6a2B0⊣	R 4	4 4
$	7B6a6a6a2B0⊣	R 3	4 4 3
$	8A6a6a6a2B0⊣	R 2	4 4 3 2
$	8A6a6a2B0⊣	R 2	4 4 3 2 2
$	8A6a2B0⊣	R 2	4 4 3 2 2 2
$	5A2B0⊣	R 1	4 4 3 2 2 2 1
$	1S0⊣	D	4 4 3 2 2 2 1
ϵ	4$1S0⊣	A	4 4 3 2 2 2 1

Sommaire

LR(0) insuffisant

Très peu de grammaires sont LR(0). Une grammaire avec une production vide ne peut être LR(0).

- Règles :
  - $T \longrightarrow \epsilon$
  - $A \longrightarrow \alpha T \beta$
- Ensemble d'items :
  - $\{ A \longrightarrow \alpha \cdot T\beta,\ T \longrightarrow \cdot, \ldots \}$

Conflit : Décaler - Réduire.

Table LR(0)

Etats	Action	Successeurs						
		a	b	c	d	S	T	$
0	Décalage	2		3	4	1		
1	Décalage							5
2	Décalage	2		3	4	6		
3	Décalage	8	9	10			7	
4	Réduction 3							
5	Acceptation							
6	Décalage		11					
7	Réduction 2							

Principe

- La méthode LR(0) ne considère aucun caractère de prévision.
- La méthode SLR(1) prend en compte les suivants globaux de chaque variable pour construire les tables d'actions et de successeurs.
- Dans la méthode SLR(1) un manche ne doit pas être réduit à un non terminal A si le caractère de prévision n'appartient pas à *Suiv(A)*.
- Règle $S \longrightarrow cT$; un manche cT ne doit conduire à une réduction dans un syntagme $wcTx\alpha$ que si x est un suivant de S.

Table SLR(1)

- Le diagramme de transition de l'automate SLR(1) est le même que celui de l'automate LR(0).
- Le nombre d'états est le même.
- La table des successeurs est la même.
- La table d'actions est différente : table à deux dimensions.

Table SLR(1) - Description

Etats	Actions					Succ	
	a	b	c	d	$	S	T
0	D 2		D 3	D 4		1	
1					D 5		
2	D 2		D 3	D 4		6	

- La table des actions fusionne avec la table des successeurs pour les terminaux et $.
- La table des successeurs ne comporte plus que les variables.

Algorithme

1. Définir les ensembles d'items LR(0) : $\{I_0, \cdots I_n\}$
   I_p permet de construire l'état p de l'analyseur.
2. Pour chaque $\delta(I_i, a) = I_j$, $a \in T$
   mettre Décalage j dans la case $M_{Actions}[i, a]$
3. Pour chaque $\delta(I_i, A) = I_j$,
   mettre j dans la case $M_{Successeurs}[i, A]$
4. Si $A \longrightarrow \alpha\cdot \in I_p$ (règle k) (sauf pour $A = S'$),
   mettre Réduction k dans chaque case $M_{Actions}[p, a]$ où $a \in$ SUIV(A).
5. Si $S' \longrightarrow S\$\cdot \in I_p$,
   mettre Acceptation dans $M_{Actions}[p]$.

Exemple

1. $S \longrightarrow aSbT$
2. $S \longrightarrow cT$
3. $S \longrightarrow d$
4. $T \longrightarrow aT$
5. $T \longrightarrow bS$
6. $T \longrightarrow c$

	Premiers
S	$\{a, c, d\}$
SbT	$\{a, c, d\}$
T	$\{a, b, c\}$

	Suivants
S	$\{\$, b\}$
T	$\{\$, b\}$

Table - 1

$$
\begin{aligned}
I_0 &= \{S' \longrightarrow \cdot\, S\$, S \longrightarrow \cdot\, aSbT, S \longrightarrow \cdot\, cT, S \longrightarrow \cdot\, d\} \\
I_1 &= \{S' \longrightarrow S \cdot\, \$\} \\
I_2 &= \{S \longrightarrow a \cdot\, SbT, S \longrightarrow \cdot\, aSbT, S \longrightarrow \cdot\, cT, S \longrightarrow \cdot\, d\} \\
I_3 &= \{S \longrightarrow c \cdot\, T, T \longrightarrow \cdot\, aT, T \longrightarrow \cdot\, bS, T \longrightarrow \cdot\, c\}
\end{aligned}
$$

Etats	Action					Succ	
	a	b	c	d	$	S	T
0	D 2		D 3	D 4		1	
1					D 5		
2	D 2		D 3	D 4		6	
3	D 8	D 9	D 10				7

Table - 2

$I_4 = \{S \longrightarrow d\cdot\}$ $I_5 = \{S' \longrightarrow S\$\cdot\}$ Suivant(S) = $\{\$, b\}$

$I_6 = \{S \longrightarrow aS\cdot bT\}$ $I_7 = \{S \longrightarrow cT\cdot\}$

Etats	Action					Succ	
	a	b	c	d	$	S	T
4		R 3			R 3		
5	Acceptation						
6		D 11					
7		R 2			R 2		

Table - 3

Etats	Action					Succ	
	a	b	c	d	$	S	T
8	D 8	D 9	D 10				12
9	D 2		D 3	D 4		13	
10		R 6			R 6		
11	D 8	D 9	D 10				14
12		R 4			R 4		
13		R 5			R 5		
14		R 1			R 1		

Grammaire SLR(1), non LR(0)

$T = \{id, +, (,), [,]\}$ $V = \{E, T\}$

1. $E \longrightarrow T$
2. $E \longrightarrow E + T$
3. $T \longrightarrow id$
4. $T \longrightarrow (E)$
5. $T \longrightarrow id[E]$: élément de tableau

Conflit Décalage - Réduction en analyse LR(0)

- Après la rencontre du symbole *id*, on peut soit réduire en appliquant la règle 3 soit décaler pour compléter le choix de la règle 5.

Solution du conflit

- L'analyse SLR(1), prend en compte les caractères pouvant suivre chacune des variables.
- Le terminal [n'apparaît pas dans les suivants possibles de la variable T.
- Ceci indique un décalage obligatoire lorsqu'a été reconnue la séquence *id* [.
- Un syntagme tel que $\alpha id[$ ne pourrait se réduire en $\alpha T[$ que si [était un suivant possible de T.

Items LR(0)

$$I_0 = \{S' \longrightarrow \cdot E\$, E \longrightarrow \cdot T, E \longrightarrow \cdot E + T, T \longrightarrow \cdot id, T \longrightarrow \cdot (E), T \longrightarrow \cdot id[E]\}$$

$I_1 = \delta(I_0, E) = \{S' \longrightarrow E \cdot \$, E \longrightarrow E \cdot + T\}$

$I_2 = \delta(I_0, T) = \{E \longrightarrow T \cdot\}$

$I_3 = \delta(I_0, id) = \{T \longrightarrow id \cdot, T \longrightarrow id \cdot [E]\}$

Conflit Décaler - Réduire en LR(0)

$I_4 = \delta(I_0, () = \{T \longrightarrow (\cdot E), E \longrightarrow \cdot T, E \longrightarrow \cdot E + T, T \longrightarrow \cdot id, T \longrightarrow \cdot (E), T \longrightarrow \cdot id[E]\}$

Premiers - Suivants

1. $E \longrightarrow T$
2. $E \longrightarrow E + T$
3. $T \longrightarrow id$
4. $T \longrightarrow (E)$
5. $T \longrightarrow id[E]$

	Premiers
E	$\{id, (\}$
T	$\{id, (\}$
E + T	$\{id, (\}$

	Suivants
E	$\{\$, +,),]\}$
T	$\{\$, +,),]\}$

Tables d'analyse

$$
\begin{array}{lll}
I_0 = & & \{S' \longrightarrow \cdot\, E\$, E \longrightarrow \cdot\, T, E \longrightarrow \cdot\, E + T, \\
 & & T \longrightarrow \cdot\, id, T \longrightarrow \cdot\, (E), T \longrightarrow \cdot\, id[E]\} \\
I_1 = & \delta(I_0, E) = & \{S' \longrightarrow E \cdot \$, E \longrightarrow E \cdot + T\} \\
I_2 = & \delta(I_0, T) = & \{E \longrightarrow T \cdot\}
\end{array}
$$

Etats	Actions							Succ	
	id	+	(	)	[	]	$	E	T
0	D 3		D 4					1	2
1		D 6					D 5		
2		R 1		R 1		R 1	R 1		

Tables d'analyse

$I_3 = \quad \delta(I_0, id) = \quad \{T \longrightarrow id\cdot\,, T \longrightarrow id\cdot\,[E]\}$

Suivants(T)	$\{\$, +,),]\}$

Etats	Actions							Succ	
	id	+	(	)	[	]	$	E	T
3		R 3		R 3	D 7	R 3	R 3		
4	D 3		D 4					8	2
5	Acceptation								
6	D 3		D 4						9

Conflits

- Conflit Décaler - Réduire
  - Certains états peuvent contenir un item de la forme $A \longrightarrow \alpha\cdot a\beta$ et un item de la forme $A \longrightarrow \alpha\cdot$.
- Conflit Réduire - Réduire
  - Certains états peuvent contenir deux items de la forme $A \longrightarrow \alpha\cdot$, représentant deux situations de réduction par deux productions différentes.

Sommaire

Exemple

T = $\{=, *, id\}$ V = $\{S, G, D\}$
P = $\{ S \longrightarrow G = D ; S \longrightarrow D ; G \longrightarrow *D ; G \longrightarrow id ; D \longrightarrow G \}$

$=$ appartient à $SUIV(D)$, puisque : $S \Rightarrow G = D \Rightarrow * D = D$

$I_0 = \{S' \longrightarrow \cdot S\$,$
$S \longrightarrow \cdot G = D,$
$S \longrightarrow \cdot D,$
$D \longrightarrow \cdot G,$
$G \longrightarrow \cdot *D,$
$G \longrightarrow \cdot id\}$

I_1 = Transition(I_0, G)
$I_1 = \{S \longrightarrow G \cdot = D, D \longrightarrow G \cdot \}$
I_1 indique : réduire par $D \longrightarrow G$
en présence d'un suivant de D ($=$).
$G = * * id$ se réduirait en $D = * * id$
or un syntagme $D = \alpha$ ne peut exister.

Solution

- Si un ensemble d'items I_i contient $A \longrightarrow \alpha\cdot$
  et si $a \in SUIV(A)$ alors la méthode *SLR* indique de réduire.
- Or, il peut exister des préfixes $\beta\alpha$ ne pouvant pas être suivis de a.
- Des conflits Décaler-Réduire peuvent survenir car la méthode SLR ne mémorise pas assez de contexte gauche pour décider de l'action à effectuer.
- Il faut attacher plus d'information aux items LR(0)
- Il faut que chaque état d'un analyseur LR indique exactement les symboles d'entrée qui peuvent suivre un manche $A \longrightarrow \alpha$ pour lesquels une réduction vers A est possible.

Items LR(1)

- Un item LR(1) est une forme : $[A \longrightarrow \alpha \cdot \beta, a]$
  - $A \longrightarrow \alpha\beta$ est une production de la grammaire
  - a est un symbole terminal ou \$.
- Un item $[A \longrightarrow \alpha \cdot , a]$ signifie :
  Réduire par la production $A \longrightarrow \alpha$ uniquement si le prochain symbole d'entrée est *a*.
- Un item $[A \longrightarrow \alpha \cdot b\beta, a]$ implique un décalage.

Préfixe viable

- Un item LR(1) $[A \longrightarrow \alpha \cdot \beta, a]$ est valide pour un préfixe viable $\delta\alpha$, s'il existe une dérivation droite telle que : $S \underset{rm}{\overset{*}{\Longrightarrow}} \delta A a w \underset{rm}{\Longrightarrow} \delta\alpha\beta a w$
  - $\delta\alpha$ est une chaîne qui peut commencer un syntagme.
  - $\delta\alpha\beta a w$ peut se réduire en $\delta A a w$ car a peut suivre A dans un syntagme commençant par δ.

Fermeture d'un ensemble d'items

- $[A \longrightarrow \alpha \cdot B\beta, a]$, valide pour un préfixe $\delta\alpha$.
- Il existe une dérivation : $S \underset{rm}{\overset{*}{\Longrightarrow}} \delta Aaw \underset{rm}{\Longrightarrow} \delta\alpha B\beta aw$.
- Pour une production $B \longrightarrow \eta$, la dérivation devient : $S \underset{rm}{\overset{*}{\Longrightarrow}} \delta Aaw \underset{rm}{\Longrightarrow} \delta\alpha B\beta aw \underset{rm}{\Longrightarrow} \delta\alpha\eta\beta aw$.
- Les items valides pour le même préfixe $\delta\alpha$ sont les items $[B \longrightarrow \cdot\, \eta, b]$ tels que $b \in PREM(\beta aw)$.
- Remarque : $PREM(\beta aw) = PREM(\beta a)$
  - Si β est nullifiable alors $a \in PREM(\beta a)$
  - Si β n'est pas nullifiable alors $PREM(\beta a) = PREM(\beta)$

Algorithme de Fermeture

Soit *I* un ensemble d'items d'une grammaire G.

- Placer chaque item de *I* dans Fermeture(I).
- Si l'item $[A \longrightarrow \alpha \cdot B\beta, a] \in$ Fermeture(I) et si $B \longrightarrow \gamma$ est une production de G
  - Pour chaque terminal b $\in$ PREM(βa) Ajouter l'item $[B \longrightarrow \cdot \gamma, b]$ à Fermeture(I).
- Recommencer tant que Fermeture(I) n'est pas stable.

Transition

Transition(I,X), où *I* est un ensemble d'items d'une grammaire G et X un élément de $T \cup V$.

- Pour les items de la forme $[A \longrightarrow \alpha_1 \cdot X \alpha_2, a] \in I$
  - Construire J l'ensemble des items de la forme $[A \longrightarrow \alpha_1 X \cdot \alpha_2, a]$
- Transition(I,X) = Fermeture(J)

Ensemble des items

- Ajouter l'axiome S' et la production $S' \longrightarrow S\$$.
- $I_0 \longleftarrow$ Fermeture($\{[S' \longrightarrow \cdot\, S\$, \epsilon]\}$)
- $C \longleftarrow \{I_0\}$
- Pour chaque I $\in C$
  - Pour tout $X \in V \cup T$ tel que Transition(I,X) $\neq \emptyset$, Ajouter Transition(I,X) à C
- Recommencer tant que C n'est pas stable

Exemple : Items LR(1)

G la grammaire telle que T = $\{c, d\}$, V = $\{S, C\}$,
P = $\{\ S \longrightarrow CC,\ C \longrightarrow cC,\ C \longrightarrow d\ \}$

$$I_0 = \{[S' \longrightarrow \cdot\, S\$, \epsilon], [S \longrightarrow \cdot\, CC, \$], [C \longrightarrow \cdot\, cC, c], [C \longrightarrow \cdot\, cC, d], [C \longrightarrow \cdot\, d, c], [C \longrightarrow \cdot\, d, d]\}$$

$$I_1 = \delta(I_0, S) = \{[S' \longrightarrow S\cdot\, \$, \epsilon]\}$$
$$I_2 = \delta(I_0, C) = \{[S \longrightarrow C\cdot\, C, \$], [C \longrightarrow \cdot\, cC, \$], [C \longrightarrow \cdot\, d, \$]\}$$
$$I_3 = \delta(I_0, c) = \{[C \longrightarrow c\cdot\, C, c], [C \longrightarrow c\cdot\, C, d], [C \longrightarrow \cdot\, cC, c], [C \longrightarrow \cdot\, cC, d], [C \longrightarrow \cdot\, d, c], [C \longrightarrow \cdot\, d, d]\}$$
$$I_4 = \delta(I_0, d) = \{[C \longrightarrow d\cdot\, , c][C \longrightarrow d\cdot\, , d]\}$$

Exemple : Items LR(1)

G la grammaire telle que T = $\{c, d\}$, V = $\{S, C\}$,
P = $\{ S \longrightarrow CC,\ C \longrightarrow cC,\ C \longrightarrow d \}$

$I_1 =$	$\delta(I_0, S) =$	$\{[S' \longrightarrow S\cdot \$, \epsilon]\}$
$I_2 =$	$\delta(I_0, C) =$	$\{[S \longrightarrow C\cdot C, \$], [C \longrightarrow \cdot\, cC, \$], [C \longrightarrow \cdot\, d, \$]\}$
$I_3 =$	$\delta(I_0, c) =$	$\{[C \longrightarrow c\cdot C, c], [C \longrightarrow c\cdot C, d], [C \longrightarrow \cdot\, cC, c],$ $[C \longrightarrow \cdot\, cC, d], [C \longrightarrow \cdot\, d, c], [C \longrightarrow \cdot\, d, d]\}$
$I_4 =$	$\delta(I_0, d) =$	$\{[C \longrightarrow d\cdot\,, c][C \longrightarrow d\cdot\,, d]\}$
$I_5 =$	$\delta(I_1, \$) =$	$\{[S' \longrightarrow S\$\cdot\,, \epsilon]\}$

Exemple : Items LR(1)

G la grammaire telle que T = $\{c, d\}$, V = $\{S, C\}$,
P = $\{ S \longrightarrow CC,\ C \longrightarrow cC,\ C \longrightarrow d \}$

$I_6 =$	$\delta(I_2, C) =$	$\{[S \longrightarrow CC\cdot\ , \$]\}$	
$I_7 =$	$\delta(I_2, c) =$	$\{[C \longrightarrow c\cdot\ C, \$], [C \longrightarrow\cdot\ cC, \$], [C \longrightarrow\cdot\ d, \$]\}$	
$I_8 =$	$\delta(I_2, d) =$	$\{[C \longrightarrow d\cdot\ , \$]\}$	
$I_9 =$	$\delta(I_3, C) =$	$\{[C \longrightarrow cC\cdot\ , c], [C \longrightarrow cC\cdot\ , d]\}$	
	$\delta(I_3, c) =$	$\{[C \longrightarrow c\cdot\ C, c], [C \longrightarrow c\cdot\ C, d], [C \longrightarrow\cdot\ cC, c],$	
		$[C \longrightarrow\cdot\ cC, d], [C \longrightarrow\cdot\ d, c], [C \longrightarrow\cdot\ d, d]\}$	$= I_3$
	$\delta(I_3, d) =$	$\{[C \longrightarrow d\cdot\ , c][C \longrightarrow d\cdot\ , d]\}$	$= I_4$
$I_{10} =$	$\delta(I_7, C) =$	$\{[C \longrightarrow cC\cdot\ , \$]\}$	
	$\delta(I_7, c) =$	$\{[C \longrightarrow c\cdot\ C, \$], [C \longrightarrow\cdot\ cC, \$], [C \longrightarrow\cdot\ d, \$]\}$	$= I_7$
	$\delta(I_7, d) =$	$\{[C \longrightarrow d\cdot\ , \$]\}$	$= I_8$

Algorithme

1. Construire C = $\{I_0, I_1, \ldots I_n\}$ l'ensemble des items LR(1). I_i permet de construire l'état i de l'analyseur.
2. Si $[A \longrightarrow \alpha \cdot a\beta, b] \in I_i$
   mettre Décalage j dans $M_{Action}[i, a]$ (b ignoré) où $I_j = \delta(I_i, a)$.
3. Si $\delta(I_i, A) = I_j$, mettre j dans $M_{Successeur}[i, A]$.
4. Si $[A \longrightarrow \alpha \cdot , a] \in I_i$ et $A \neq S'$
   mettre Réduction p (où p est le rang de $A \longrightarrow \alpha$) dans $M_{Action}[i, a]$.
5. Si $[S' \longrightarrow S\$ \cdot , \epsilon] \in I_i$
   mettre Acceptation dans $M_{Action}[i]$.

Construction - 1

$I_1 = \delta(I_0, S)$ $I_2 = \delta(I_0, C)$ $I_3 = \delta(I_0, c)$ $I_4 = \delta(I_0, d)$

$I_5 = \delta(I_1, \$)$ $I_6 = \delta(I_2, C)$ $I_7 = \delta(I_2, c)$ $I_8 = \delta(I_2, d)$

$I_9 = \delta(I_3, C)$ $I_3 = \delta(I_3, c)$ $I_4 = \delta(I_3, d)$

Etats	Action			Succ	
	c	d	$	S	C
0	D 3	D 4		1	2
1			D 5		
2	D 7	D 8			6
3	D 3	D 4			9

Construction - 2

$I_4 = \{[C \longrightarrow d\cdot\,, c][C \longrightarrow d\cdot\,, d]\}$
$I_5 = \{[S' \longrightarrow S\$\cdot\,, \epsilon]\}$
$I_6 = \{[S \longrightarrow CC\cdot\,, \$]\}$

Etats	Action			Succ	
	c	d	$	S	C
4	R 3	R 3			
5	Acceptation				
6			R 1		

Analyse de ccdcd

Etats	Action			Succ	
	c	d	$	S	C
0	D 3	D 4		1	2
1			D 5		
2	D 7	D 8			6
3	D 3	D 4			9

Entrée	Pile	Action	Sortie
ccdcd$	0⊣	Décalage	
cdcd$	3 c 0⊣	Décalage	
dcd$	3 c 3 c 0⊣	Décalage	
cd$	4 d 3 c 3 c 0⊣	Réduction	3
cd$	9 C 3 c 3 c 0⊣	Réduction	3 2

Conflits

- Une grammaire est LR(1) si aucune case de la table des actions ne contient plus d'une action.
- Lorsqu'un état contient des items tels que : $[A \longrightarrow \alpha\cdot\,, a]$ et $[A \longrightarrow \beta\cdot\, a\gamma, b]$, il dénote un conflit Décaler - Réduire à la rencontre du caractère a.
- Lorsqu'un état contient des items tels que : $[A \longrightarrow \alpha\cdot\,, a]$ et $[B \longrightarrow \alpha\cdot\,, a]$, il dénote un conflit Réduire - Réduire à la rencontre du caractère *a* commun aux suivants de *A* et de *B*.

Sommaire

Analyse LALR(1) - Introduction

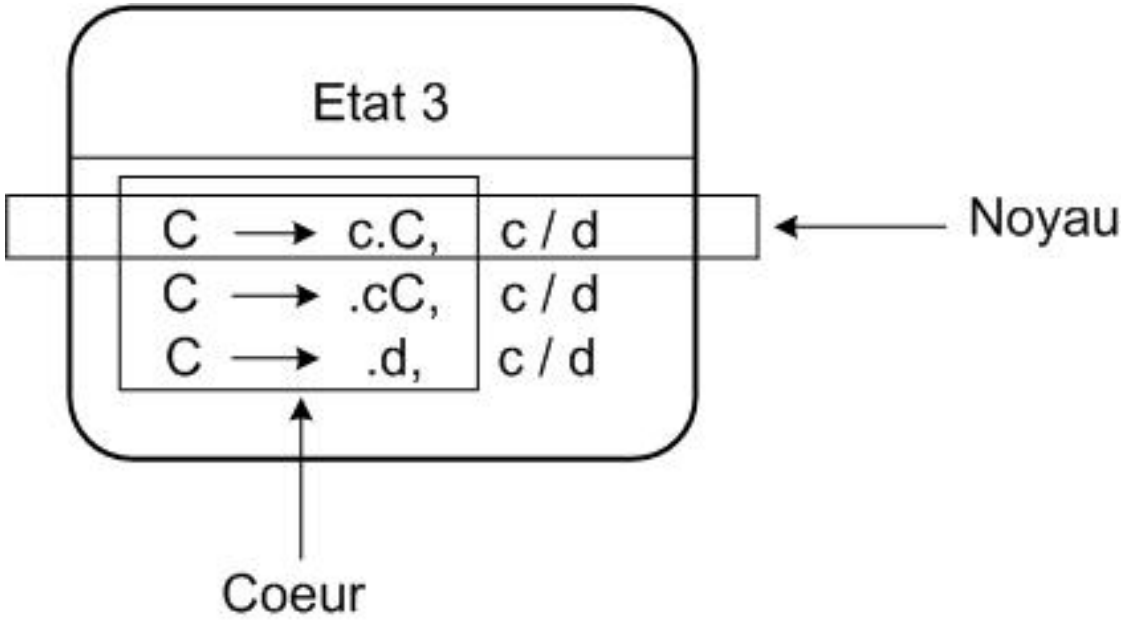
- La méthode LALR(1) (LookAheadLR) est presque aussi puissante que la méthode LR(1).
- Elle est aussi compacte que la méthode SLR(1).
- Les tables SLR et LALR ont le même nombre d'états.
- Les états LALR sont construits à partir des états LR(1) bien qu'en pratique un algorithme permette de les générer à partir des items LR(0).

Réécriture

$$
\begin{aligned}
I_3 = \{ & \\
& [C \longrightarrow c\cdot C, c], [C \longrightarrow c\cdot C, d] \\
& [C \longrightarrow \cdot\, cC, c], [C \longrightarrow \cdot\, cC, d] \\
& [C \longrightarrow \cdot\, d, c], [C \longrightarrow \cdot\, d, d] \\
\} &
\end{aligned}
$$

$$
\begin{aligned}
I_3 = \{ & \\
& [C \longrightarrow c\cdot C, c/d] \\
& [C \longrightarrow \cdot\, cC, c/d] \\
& [C \longrightarrow \cdot\, d, c/d] \\
\} &
\end{aligned}
$$

Cœur, Noyau d'un état



Fusion de deux états LR(1)

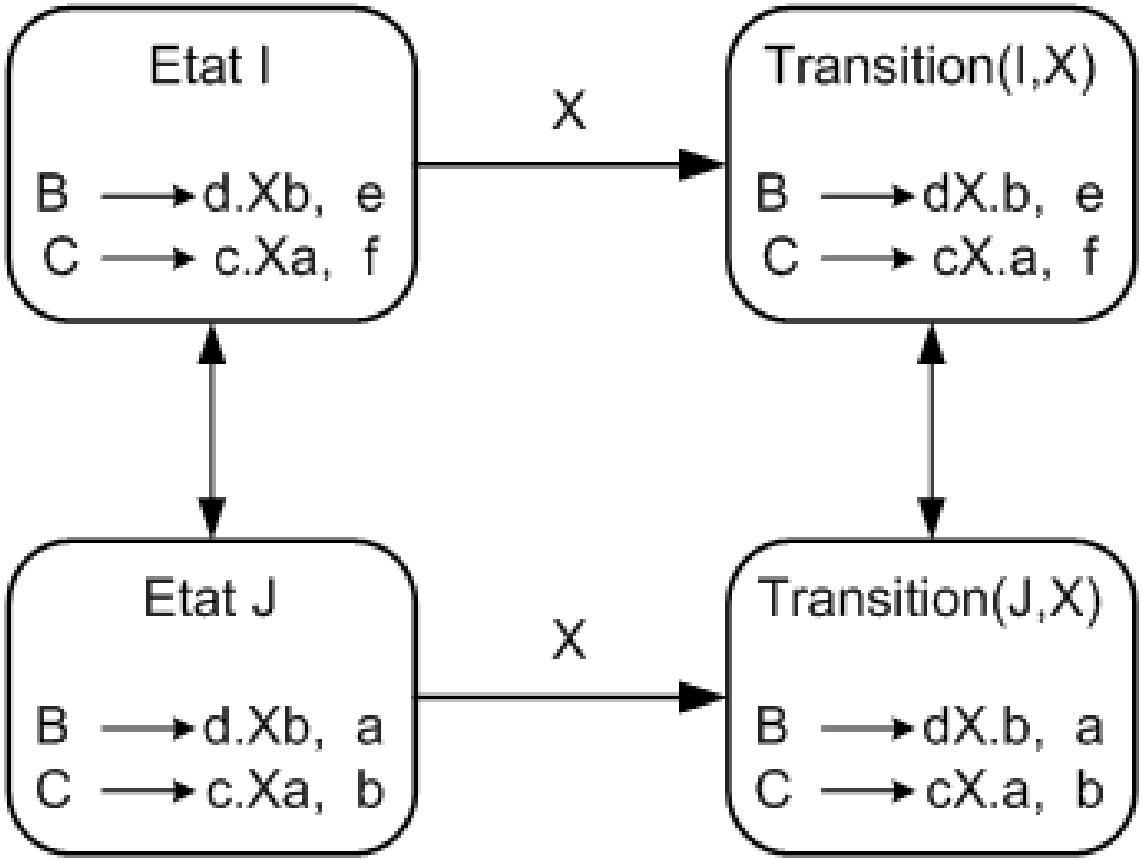
- Il est possible de fusionner deux états ayant le même cœur en réunissant les listes de symboles de prévision pour chaque item LR(0) correspondant.

$$I_3 = \{ [C \longrightarrow c\cdot C, c/d] ; [C \longrightarrow \cdot cC, c/d] ; [C \longrightarrow \cdot d, c/d] \}$$
$$I_7 = \{ [C \longrightarrow c\cdot C, \$], [C \longrightarrow \cdot cC, \$], [C \longrightarrow \cdot d, \$] \}$$
$$J_{37} = \{ [C \longrightarrow c\cdot C, c/d/\$] ; [C \longrightarrow \cdot cC, c/d/\$] ; [C \longrightarrow \cdot d, c/d/\$] \}$$

$$I_4 = \{[C \longrightarrow d\cdot, c][C \longrightarrow d\cdot, d]\} \qquad I_8 = \{[C \longrightarrow d\cdot, \$]\}$$
$$J_{48} = \{ [C \longrightarrow d\cdot, c/d/\$] \}$$

Consistance par transition

- Si *I* et *J* ont le même cœur alors Transition(I,X) et Transition(J,X) ont le même cœur puisqu'ils ne diffèrent que sur les symboles de prévision.



Algorithme - 1

- Construire C = $\{I_0, I_1, \ldots I_n\}$ l'ensemble des items LR(1) pour la grammaire augmentée.
- Fusionner les états de même cœur.
  C' = $\{J_0, J_1, \ldots J_m\}$ l'ensemble des items LALR(1) résultant ($m \leq n$). J_i permet de construire l'état i.
- Soient δ et Δ les fonctions de transition respectives sur les états LR(1) et LALR(1).
  - Si I_k n'a pas fusionné, il est renommé J_k et $\Delta(J_k, X) = \delta(I_k, X)$ pour tout X.
  - Si $\delta(I_k, X) = I_p$ et si I_k a fusionné en J_l alors I_p a fusionné en J_q et $\Delta(J_l, X) = J_q$

Algorithme - 2

- Si $[A \longrightarrow \alpha\cdot a\beta, b] \in J_i$
  mettre Décalage j dans $M_{Action}[i, a]$ (b ignoré)
  où $J_j = \Delta(J_i, a)$.
- Si $\Delta(J_i, A) = J_j$
  mettre j dans $M_{Successeur}[i, A]$
- Si $[A \longrightarrow \alpha\cdot\,, a] \in J_i$ et $A \neq S'$
  mettre Réduction p (où p est le rang de $A \longrightarrow \alpha$)
  dans $M_{Action}[i, a]$
- Si $[S' \longrightarrow S\$\cdot\,, \epsilon] \in J_i$
  mettre Acceptation dans $M_{Action}[i]$.

Exemple

$I_3 = \{[C \longrightarrow c\cdot C, c/d] ; [C \longrightarrow \cdot cC, c/d] ; [C \longrightarrow \cdot d, c/d]\}$
$I_7 = \{[C \longrightarrow c\cdot C, \$], [C \longrightarrow \cdot cC, \$], [C \longrightarrow \cdot d, \$]\}$
$J_{37} = \{[C \longrightarrow c\cdot C, c/d/\$] ; [C \longrightarrow \cdot cC, c/d/\$] ; [C \longrightarrow \cdot d, c/d/\$]\}$

$I_4 = \{[C \longrightarrow d\cdot , c][C \longrightarrow d\cdot , d]\}$ $\quad$ $I_8 = \{[C \longrightarrow d\cdot , \$]\}$
$J_{48} = \{ [C \longrightarrow d\cdot , c/d/\$] \}$

$I_9 = \{[C \longrightarrow cC\cdot , c], [C \longrightarrow cC\cdot , d]\}$ $\quad$ $I_{10} = \{[C \longrightarrow cC\cdot , \$]\}$
$J_{910} = \{[C \longrightarrow cC\cdot , c/d/\$]\}$

Tables

Analyse LR					
Etats	Action			Succ	
	c	d	$	S	C
0	D 3	D 4		1	2
1			D 5		
2	D 7	D 8			6
3	D 3	D 4			9
4	R 3	R 3			
5	Acceptation				
6			R 1		
7	D 7	D 8			10
8			R 3		

Analyse LALR					
Etats	Action			Succ	
	c	d	$	S	C
0	D 37	D 48		1	2
1			D 5		
2	D 37	D 48			6
37	D 37	D 48			910
48	R 3	R 3	R 3		
5	Acceptation				
6			R 1		
910	R 2	R 2	R 2		

Conflits

- La fusion de deux états ne peut entraîner de nouveaux conflits Décaler - Réduire absents des états LR(1).
  - Soit J_{kl} un état LALR présentant un conflit Décaler - Réduire.
    $J_{kl} = \{[A \longrightarrow \alpha\cdot\,, a], [B \longrightarrow \beta\cdot a\gamma, b], \cdots\}$
  - Soit I_k l'état LR contenant $[A \longrightarrow \alpha\cdot\,, a]$.
    Puisque I_k et I_l ont le même cœur, il contient également un item tel que $[B \longrightarrow \beta\cdot a\gamma, c]$.
    I_k présente donc déjà un conflit Décaler - Réduire.
- La fusion de deux états peut entraîner des conflits Réduire - Réduire qui n'auraient pas figuré dans les états LR(1).
- Il existe des grammaires LR(1) non LALR(1).

Grammaire LR(1) non LALR(1)

G la grammaire :
T = $\{a, b, c, d, e\}$, V = $\{S, A, B\}$
P = $\{S \longrightarrow aAd/bBd/aBe/bAe, A \longrightarrow c, B \longrightarrow c\}$

Grammaire LR(1) non LALR(1)

G la grammaire :
T = $\{a, b, c, d, e\}$, V = $\{S, A, B\}$
P = $\{S \longrightarrow aAd/bBd/aBe/bAe, A \longrightarrow c, B \longrightarrow c\}$

$I_0 = \{[S' \longrightarrow \cdot S\$, \epsilon], [S \longrightarrow \cdot aAd, \$], [S \longrightarrow \cdot bBd, \$]$
$[S \longrightarrow \cdot aBe, \$], [S \longrightarrow \cdot bAe, \$]\}$

$I_2 = \{[S \longrightarrow a \cdot Ad, \$], [S \longrightarrow a \cdot Be, \$], [A \longrightarrow \cdot c, d], [B \longrightarrow \cdot c, e]\}$
$I_3 = \{[S \longrightarrow b \cdot Bd, \$], [S \longrightarrow b \cdot Ae, \$], [B \longrightarrow \cdot c, d], [A \longrightarrow \cdot c, e]\}$

$I_k = \delta(I_2, c) = \{[A \longrightarrow c \cdot , d], [B \longrightarrow c \cdot , e]\}$
$I_l = \delta(I_3, c) = \{[A \longrightarrow c \cdot , e], [B \longrightarrow c \cdot , d]\}$
I_k et I_l ne présentent pas de conflit Réduire - Réduire.

Grammaire LR(1) non LALR(1)

G la grammaire :
T = $\{a, b, c, d, e\}$, V = $\{S, A, B\}$
P = $\{S \longrightarrow aAd/bBd/aBe/bAe, A \longrightarrow c, B \longrightarrow c\}$

$I_k = \{[A \longrightarrow c\cdot\ , d], [B \longrightarrow c\cdot\ , e]\}$
$I_l = \{[A \longrightarrow c\cdot\ , e], [B \longrightarrow c\cdot\ , d]\}$

Le préfixe *ac* est viable et les chaînes *acd* et *ace* sont reconnues :
$S \underset{rm}{\Longrightarrow} aAd \underset{rm}{\Longrightarrow} acd \qquad S \underset{rm}{\Longrightarrow} aBe \underset{rm}{\Longrightarrow} ace$

Après fusion : $J_{kl} = \{[A \longrightarrow c\cdot\ , d/e], [B \longrightarrow c\cdot\ , e/d]\}$

J_{kl} présente un conflit Réduire - Réduire à la rencontre des symboles d et e.

Grammaires des langages de programmation

Langage	Nb règles	Nb états LALR	Conflits LR
C	210	375	38
ECMA Script	270	513	38
C#	1200	2021	457

TABLE: Exemples de grammaires

Sommaire

Examen 2007

```
E -> while E do E      E -> id := E    E -> id
```

- Construire les 8 premiers états de l'automate canonique associé dans l'analyse LR(1) :
  $I_1 = \delta(I_0, E)$; $I_2 = \delta(I_0, \mathit{while})$; $I_3 = \delta(I_0, \mathit{id})$; $I_4 = \delta(I_1, \$)$;
  $I_5 = \delta(I_2, E)$; $I_6 = \delta(I_2, \mathit{while})$; $I_7 = \delta(I_2, \mathit{id})$
- Calculer $\delta(I_6, \mathit{while})$ et $\delta(I_6, \mathit{id})$; A-t-on des conflits LR(1) ; Construire les tables d'analyse jusqu'à I_7. On suppose :
  $I_8 = \delta(I_3, :=)$; $I_9 = \delta(I_5, \mathit{do})$; $I_{10} = \delta(I_6, E)$; $I_{11} = \delta(I_7, =)$
- Montrer que deux couples d'états peuvent fusionner en analyse LALR(1). En déduire que deux autres couples fusionnent.
- Construire le début de la table d'analyse LALR(1).