

Correction du TP 3

Nicolas Monmarché
département informatique – Polytech Tours

1 Construction de la table SLR

Remarque : la correction proposée ci-dessous ne prend pas en compte le fait qu'un texte JSON puisse être directement un tableau (dans la version présentée ci-dessous, le tableau doit être encapsulé dans un objet).

Avec les notations proposées dans le support de TP, la grammaire devient :

$$\mathcal{A} = \{\{'\{'\}'\}', '\['\]''\}, '\{'\}'\}, '\{'\}'\}, '\{'\}'\}, s, n, t, f, u\}$$

$$V_N = \{O, M, P, A, E, V\}$$

$$\text{axiome} = O$$

$$\text{Règles} = \left\{ \begin{array}{l} O \rightarrow \{\} \\ O \rightarrow \{M\} \\ M \rightarrow P \\ M \rightarrow P, M \\ P \rightarrow s : V \\ A \rightarrow [] \\ A \rightarrow [E] \\ E \rightarrow V \\ E \rightarrow V, E \\ V \rightarrow s \\ V \rightarrow n \\ V \rightarrow O \\ V \rightarrow A \\ V \rightarrow t \\ V \rightarrow f \\ V \rightarrow u \end{array} \right.$$

Et si on l'augmente, on a la grammaire augmentée (dont on numérote les règles) :

$$\begin{aligned} \mathcal{A} &= \{\{', '\}, '[', ']', ', ', ':', s, n, t, f, u\} \\ V_N &= \{O, M, P, A, E, V\} \\ \text{axiome} &= S \end{aligned}$$

$$\text{R\`egles} = \left\{ \begin{array}{ll} (0) & S \rightarrow O \\ (1) & O \rightarrow \{\} \\ (2) & O \rightarrow \{M\} \\ (3) & M \rightarrow P \\ (4) & M \rightarrow P, M \\ (5) & P \rightarrow s : V \\ (6) & A \rightarrow \square \\ (7) & A \rightarrow [E] \\ (8) & E \rightarrow V \\ (9) & E \rightarrow V, E \\ (10) & V \rightarrow s \\ (11) & V \rightarrow n \\ (12) & V \rightarrow O \\ (13) & V \rightarrow A \\ (14) & V \rightarrow t \\ (15) & V \rightarrow f \\ (16) & V \rightarrow u \end{array} \right.$$

On construit ensuite l'automate LR en utilisant les fonctions Avancer ($A()$) et Fermeture ($F()$) données dans le cours de compilation :

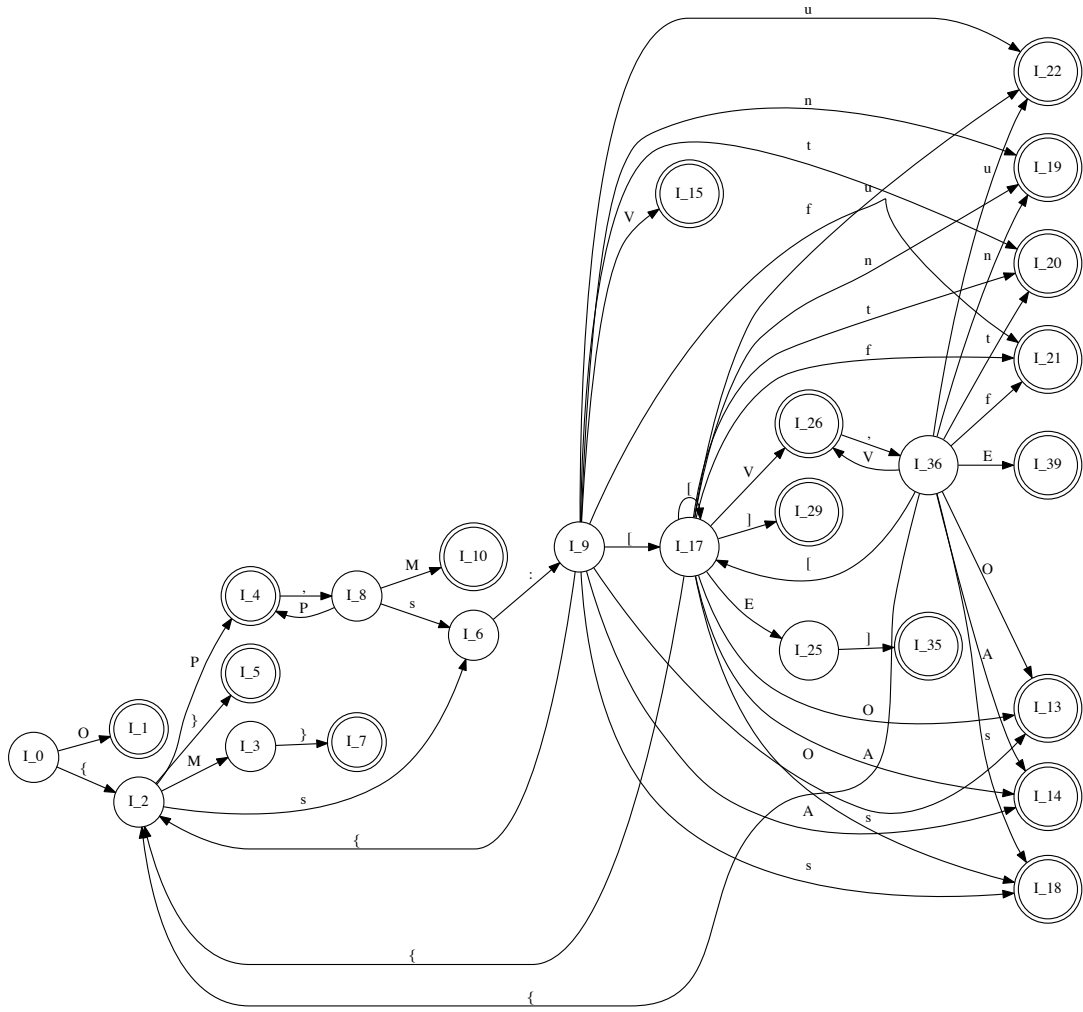
$$\begin{aligned}
- & I_0 = F(\{[S \rightarrow \bullet O]\}) = \{[S \rightarrow \bullet O], [O \rightarrow \bullet \{\}], [O \rightarrow \bullet \{M\}]\} \\
- & I_1 = A(I_0, O) = F(\{[S \rightarrow O \bullet]\}) = \{[S \rightarrow O \bullet]\} \\
- & I_2 = A(I_0, \{\}) = F(\{[O \rightarrow \{\bullet\}], [O \rightarrow \{\bullet M\}]\}) = \left\{ \begin{array}{l} [O \rightarrow \{\bullet\}] \\ [O \rightarrow \{\bullet M\}] \\ [M \rightarrow \bullet P] \\ [M \rightarrow \bullet P, M] \\ [P \rightarrow \bullet s : V] \end{array} \right\} \\
- & I_3 = A(I_2, M) = F(\{[O \rightarrow \{M \bullet\}]\}) = \{[O \rightarrow \{M \bullet\}]\} \\
- & I_4 = A(I_2, P) = F(\{[M \rightarrow P \bullet], [M \rightarrow P \bullet, M]\}) = \{[M \rightarrow P \bullet], [M \rightarrow P \bullet, M]\} \\
- & I_5 = A(I_2, \}) = F(\{[O \rightarrow \{\}\bullet]\}) = \{[O \rightarrow \{\}\bullet]\} \\
- & I_6 = A(I_2, s) = F(\{[P \rightarrow s \bullet : V]\}) = \{[P \rightarrow s \bullet : V]\} \\
- & I_7 = A(I_3, \}) = F(\{[O \rightarrow \{M\}\bullet]\}) = \{[O \rightarrow \{M\}\bullet]\} \\
- & I_8 = A(I_4, ,) = F(\{[M \rightarrow P, \bullet M]\}) = \left\{ \begin{array}{l} [M \rightarrow P, \bullet M] \\ [M \rightarrow \bullet P] \\ [M \rightarrow \bullet P, M] \\ [P \rightarrow \bullet s : V] \end{array} \right\}
\end{aligned}$$

$$\begin{aligned}
- I_9 &= A(I_6, :) = F(\{[P \rightarrow s : \bullet V]\}) = \left\{ \begin{array}{l} [P \rightarrow s : \bullet V] \\ [V \rightarrow \bullet s] \\ [V \rightarrow \bullet n] \\ [V \rightarrow \bullet O] \\ [V \rightarrow \bullet A] \\ [V \rightarrow \bullet t] \\ [V \rightarrow \bullet f] \\ [V \rightarrow \bullet u] \\ [O \rightarrow \bullet \{\}] \\ [O \rightarrow \bullet \{M\}] \\ [A \rightarrow \bullet []] \\ [A \rightarrow \bullet [E]] \end{array} \right\} \\
- I_{10} &= A(I_8, M) = F(\{[M \rightarrow P, M\bullet]\}) = \{[M \rightarrow P, M\bullet]\} \\
- (I_{11}) &= A(I_8, P) = F(\{[M \rightarrow P\bullet], [M \rightarrow P\bullet, M]\}) = \{[M \rightarrow P\bullet], [M \rightarrow P\bullet, M]\} = I_4 \\
- (I_{12}) &= A(I_8, s) = F(\{[P \rightarrow s\bullet : V]\}) = \{[P \rightarrow s\bullet : V]\} = I_6 \\
- I_{13} &= A(I_9, O) = F(\{[V \rightarrow O\bullet]\}) = \{[V \rightarrow O\bullet]\} \\
- I_{14} &= A(I_9, A) = F(\{[V \rightarrow A\bullet]\}) = \{[V \rightarrow A\bullet]\} \\
- I_{15} &= A(I_9, V) = F(\{[P \rightarrow s : V\bullet]\}) = \{[P \rightarrow s : V\bullet]\} \\
- (I_{16}) &= A(I_9, \{\}) = F(\{[O \rightarrow \{\bullet}], [O \rightarrow \{\bullet M\}]\}) = \left\{ \begin{array}{l} [O \rightarrow \{\bullet\}] \\ [O \rightarrow \{\bullet M\}] \\ [M \rightarrow \bullet P] \\ [M \rightarrow \bullet P, M] \\ [P \rightarrow \bullet s : V] \end{array} \right\} = I_2 \\
- I_{17} &= A(I_9, []) = F(\{[A \rightarrow [\bullet]], [A \rightarrow [\bullet E]]\}) = \left\{ \begin{array}{l} [A \rightarrow [\bullet]] \\ [A \rightarrow [\bullet E]] \\ [E \rightarrow \bullet V] \\ [E \rightarrow \bullet V, E] \\ [V \rightarrow \bullet s] \\ [V \rightarrow \bullet n] \\ [V \rightarrow \bullet O] \\ [V \rightarrow \bullet A] \\ [V \rightarrow \bullet t] \\ [V \rightarrow \bullet f] \\ [V \rightarrow \bullet u] \\ [O \rightarrow \bullet \{\}] \\ [O \rightarrow \bullet \{M\}] \\ [A \rightarrow \bullet []] \\ [A \rightarrow \bullet [E]] \end{array} \right\} \\
- I_{18} &= A(I_9, s) = F(\{[V \rightarrow s\bullet]\}) = \{[V \rightarrow s\bullet]\} \\
- I_{19} &= A(I_9, n) = F(\{[V \rightarrow n\bullet]\}) = \{[V \rightarrow n\bullet]\} \\
- I_{20} &= A(I_9, t) = F(\{[V \rightarrow t\bullet]\}) = \{[V \rightarrow t\bullet]\} \\
- I_{21} &= A(I_9, f) = F(\{[V \rightarrow f\bullet]\}) = \{[V \rightarrow f\bullet]\} \\
- I_{22} &= A(I_9, u) = F(\{[V \rightarrow u\bullet]\}) = \{[V \rightarrow u\bullet]\} \\
- (I_{23}) &= A(I_{17}, O) = F(\{[V \rightarrow O\bullet]\}) = I_{13} \\
- (I_{24}) &= A(I_{17}, A) = F(\{[V \rightarrow A\bullet V]\}) = I_{14} \\
- I_{25} &= A(I_{17}, E) = F(\{[A \rightarrow [E\bullet]]\}) = \{[A \rightarrow [E\bullet]]\} \\
- I_{26} &= A(I_{17}, V) = F(\{[E \rightarrow V\bullet], [E \rightarrow V\bullet, E]\}) = \{[E \rightarrow V\bullet], [E \rightarrow V\bullet, E]\} \\
- (I_{27}) &= A(I_{17}, \{\}) = F(\{[O \rightarrow \{\bullet}], [O \rightarrow \{\bullet M\}]\}) = I_2 \\
- (I_{28}) &= A(I_{17}, []) = F(\{[A \rightarrow [\bullet]], [A \rightarrow [\bullet E]]\}) = I_{17} \\
- I_{29} &= A(I_{17}, []) = F(\{[A \rightarrow []\bullet]\}) = \{[A \rightarrow []\bullet]\}
\end{aligned}$$

$$\begin{aligned}
& \text{--- } (I_{30}) = A(I_{17}, s) = I_{18} \\
& \text{--- } (I_{31}) = A(I_{17}, n) = I_{19} \\
& \text{--- } (I_{32}) = A(I_{17}, t) = I_{20} \\
& \text{--- } (I_{33}) = A(I_{17}, f) = I_{21} \\
& \text{--- } (I_{34}) = A(I_{17}, u) = I_{22} \\
& \text{--- } I_{35} = A(I_{25}, \text{ }) = F(\{[A \rightarrow [E] \bullet]\}) = \{[A \rightarrow [E] \bullet]\} \\
& \text{--- } I_{36} = A(I_{26}, \text{ }) = F(\{[E \rightarrow V, \bullet E]\}) = \left\{ \begin{array}{l} [E \rightarrow V, \bullet E] \\ [E \rightarrow \bullet V] \\ [E \rightarrow \bullet V, E] \\ [V \rightarrow \bullet s] \\ [V \rightarrow \bullet n] \\ [V \rightarrow \bullet O] \\ [V \rightarrow \bullet A] \\ [V \rightarrow \bullet t] \\ [V \rightarrow \bullet f] \\ [V \rightarrow \bullet u] \\ [O \rightarrow \bullet \{\}] \\ [O \rightarrow \bullet \{M\}] \\ [A \rightarrow \bullet []] \\ [A \rightarrow \bullet [E]] \end{array} \right\} \\
& \text{--- } (I_{37}) = A(I_{36}, O) = I_{13} \\
& \text{--- } (I_{38}) = A(I_{36}, A) = I_{14} \\
& \text{--- } I_{39} = A(I_{36}, E) = F(\{[E \rightarrow V, E \bullet]\}) = \{[E \rightarrow V, E \bullet]\} \\
& \text{--- } (I_{40}) = A(I_{36}, V) = I_{26} \\
& \text{--- } (I_{41}) = A(I_{36}, \{\}) = I_2 \\
& \text{--- } (I_{42}) = A(I_{36}, []) = I_{17} \\
& \text{--- } (I_{43}) = A(I_{36}, s) = I_{18} \\
& \text{--- } (I_{44}) = A(I_{36}, n) = I_{19} \\
& \text{--- } (I_{45}) = A(I_{36}, t) = I_{20} \\
& \text{--- } (I_{46}) = A(I_{36}, f) = I_{21} \\
& \text{--- } (I_{47}) = A(I_{36}, u) = I_{22}
\end{aligned}$$

Remarque : On pourrait renuméroter les états afin d'éliminer les états vides (ceux qui ont été mis entre parenthèses).

On obtient l'automate fini suivant (les états n'ont pas été renumérotés) :



Remarque L'automate a été dessiné avec Graphviz/dot. Ci-dessous, le code source :

```
digraph finite_state_machine {
    rankdir=LR;
    size="8,5"
    node [shape = doublecircle]; I_1 I_4 I_5 I_7 I_10 I_13 I_14 I_15
    I_18 I_19 I_20 I_21 I_22 I_26 I_29 I_35 I_39;
    node [shape = circle];
    I_0 -> I_1 [ label = "O" ];
    I_0 -> I_2 [ label = "{" ];
    I_2 -> I_3 [ label = "M" ];
    I_2 -> I_4 [ label = "P" ];
    I_2 -> I_5 [ label = "}" ];
    I_2 -> I_6 [ label = "s" ];
    I_3 -> I_7 [ label = "}" ];
    I_4 -> I_8 [ label = "," ];
    I_6 -> I_9 [ label = ":" ];
    I_8 -> I_10 [ label = "M" ];
    I_9 -> I_15 [ label = "u" ];
    I_9 -> I_17 [ label = "l" ];
    I_9 -> I_18 [ label = "s" ];
    I_9 -> I_22 [ label = "n" ];
    I_9 -> I_26 [ label = "v" ];
    I_9 -> I_29 [ label = "l" ];
    I_9 -> I_35 [ label = "l" ];
    I_9 -> I_36 [ label = "l" ];
    I_9 -> I_39 [ label = "E" ];
    I_10 -> I_15 [ label = "u" ];
    I_10 -> I_17 [ label = "l" ];
    I_10 -> I_18 [ label = "s" ];
    I_10 -> I_22 [ label = "n" ];
    I_10 -> I_26 [ label = "v" ];
    I_10 -> I_29 [ label = "l" ];
    I_10 -> I_35 [ label = "l" ];
    I_10 -> I_36 [ label = "l" ];
    I_10 -> I_39 [ label = "E" ];
    I_15 -> I_17 [ label = "l" ];
    I_15 -> I_18 [ label = "s" ];
    I_15 -> I_22 [ label = "n" ];
    I_15 -> I_26 [ label = "v" ];
    I_15 -> I_29 [ label = "l" ];
    I_15 -> I_35 [ label = "l" ];
    I_15 -> I_36 [ label = "l" ];
    I_15 -> I_39 [ label = "E" ];
    I_17 -> I_18 [ label = "s" ];
    I_17 -> I_22 [ label = "n" ];
    I_17 -> I_26 [ label = "v" ];
    I_17 -> I_29 [ label = "l" ];
    I_17 -> I_35 [ label = "l" ];
    I_17 -> I_36 [ label = "l" ];
    I_17 -> I_39 [ label = "E" ];
    I_18 -> I_15 [ label = "u" ];
    I_18 -> I_17 [ label = "l" ];
    I_18 -> I_18 [ label = "s" ];
    I_18 -> I_22 [ label = "n" ];
    I_18 -> I_26 [ label = "v" ];
    I_18 -> I_29 [ label = "l" ];
    I_18 -> I_35 [ label = "l" ];
    I_18 -> I_36 [ label = "l" ];
    I_18 -> I_39 [ label = "E" ];
    I_22 -> I_15 [ label = "u" ];
    I_22 -> I_17 [ label = "l" ];
    I_22 -> I_18 [ label = "s" ];
    I_22 -> I_22 [ label = "n" ];
    I_22 -> I_26 [ label = "v" ];
    I_22 -> I_29 [ label = "l" ];
    I_22 -> I_35 [ label = "l" ];
    I_22 -> I_36 [ label = "l" ];
    I_22 -> I_39 [ label = "E" ];
    I_26 -> I_15 [ label = "u" ];
    I_26 -> I_17 [ label = "l" ];
    I_26 -> I_18 [ label = "s" ];
    I_26 -> I_22 [ label = "n" ];
    I_26 -> I_26 [ label = "v" ];
    I_26 -> I_29 [ label = "l" ];
    I_26 -> I_35 [ label = "l" ];
    I_26 -> I_36 [ label = "l" ];
    I_26 -> I_39 [ label = "E" ];
    I_29 -> I_15 [ label = "u" ];
    I_29 -> I_17 [ label = "l" ];
    I_29 -> I_18 [ label = "s" ];
    I_29 -> I_22 [ label = "n" ];
    I_29 -> I_26 [ label = "v" ];
    I_29 -> I_29 [ label = "l" ];
    I_29 -> I_35 [ label = "l" ];
    I_29 -> I_36 [ label = "l" ];
    I_29 -> I_39 [ label = "E" ];
    I_35 -> I_15 [ label = "u" ];
    I_35 -> I_17 [ label = "l" ];
    I_35 -> I_18 [ label = "s" ];
    I_35 -> I_22 [ label = "n" ];
    I_35 -> I_26 [ label = "v" ];
    I_35 -> I_29 [ label = "l" ];
    I_35 -> I_35 [ label = "l" ];
    I_35 -> I_36 [ label = "l" ];
    I_35 -> I_39 [ label = "E" ];
    I_36 -> I_15 [ label = "u" ];
    I_36 -> I_17 [ label = "l" ];
    I_36 -> I_18 [ label = "s" ];
    I_36 -> I_22 [ label = "n" ];
    I_36 -> I_26 [ label = "v" ];
    I_36 -> I_29 [ label = "l" ];
    I_36 -> I_35 [ label = "l" ];
    I_36 -> I_36 [ label = "l" ];
    I_36 -> I_39 [ label = "E" ];
    I_39 -> I_15 [ label = "u" ];
    I_39 -> I_17 [ label = "l" ];
    I_39 -> I_18 [ label = "s" ];
    I_39 -> I_22 [ label = "n" ];
    I_39 -> I_26 [ label = "v" ];
    I_39 -> I_29 [ label = "l" ];
    I_39 -> I_35 [ label = "l" ];
    I_39 -> I_36 [ label = "l" ];
    I_39 -> I_39 [ label = "E" ];
```

```

I_8 -> I_4 [ label = "P" ];
I_8 -> I_6 [ label = "s" ];
I_9 -> I_13 [ label = "O" ];
I_9 -> I_14 [ label = "A" ];
I_9 -> I_15 [ label = "V" ];
I_9 -> I_2 [ label = "{" ];
I_9 -> I_17 [ label = "[" ];
I_9 -> I_18 [ label = "s" ];
I_9 -> I_19 [ label = "n" ];
I_9 -> I_20 [ label = "t" ];
I_9 -> I_21 [ label = "f" ];
I_9 -> I_22 [ label = "u" ];
I_17 -> I_13 [ label = "O" ];
I_17 -> I_14 [ label = "A" ];
I_17 -> I_26 [ label = "V" ];
I_17 -> I_2 [ label = "{" ];
I_17 -> I_17 [ label = "[" ];
I_17 -> I_29 [ label = "]" ];
I_17 -> I_18 [ label = "s" ];
I_17 -> I_19 [ label = "n" ];
I_17 -> I_20 [ label = "t" ];
I_17 -> I_21 [ label = "f" ];
I_17 -> I_22 [ label = "u" ];
I_17 -> I_25 [ label = "E" ];
I_25 -> I_35 [ label = "]" ];
I_26 -> I_36 [ label = "," ];
I_36 -> I_13 [ label = "O" ];
I_36 -> I_14 [ label = "A" ];
I_36 -> I_39 [ label = "E" ];
I_36 -> I_26 [ label = "V" ];
I_36 -> I_2 [ label = "{" ];
I_36 -> I_17 [ label = "[" ];
I_36 -> I_18 [ label = "s" ];
I_36 -> I_19 [ label = "n" ];
I_36 -> I_20 [ label = "t" ];
I_36 -> I_21 [ label = "f" ];
I_36 -> I_22 [ label = "u" ];
}

```

En calculant les ensembles :

- $Suivant(M) = \{ ' ' \}$
- $Suivant(O) = \{ ' ' , ' ' , ' ' , \# \}$
- $Suivant(V) = \{ ' ' , ' ' , ' ' \}$
- $Suivant(P) = \{ ' ' , ' ' \}$
- $Suivant(E) = \{ ' ' \}$
- $Suivant(A) = \{ ' ' , ' ' , ' ' \}$

on peut remplir la table d'analyse :

État	{	}	[]	,	:	<i>s</i>	<i>n</i>	<i>t</i>	<i>f</i>	<i>u</i>	#	<i>O</i>	<i>M</i>	<i>P</i>	<i>A</i>	<i>E</i>	<i>V</i>
I_0	dI_2												I_1					
I_1												Acc						
I_2		dI_5					dI_6							I_3	I_4			
I_3		dI_7																
I_4		$r3$			dI_8													
I_5		$r1$		$r1$	$r1$							$r1$						
I_6						dI_9												
I_7		$r2$		$r2$	$r2$							$r2$						
I_8							dI_6							I_{10}	I_4			
I_9	dI_2		dI_{17}				dI_{18}	dI_{19}	dI_{20}	dI_{21}	dI_{22}		I_{13}			I_{14}		I_{15}
I_{10}		$r4$																
I_{13}		$r12$		$r12$	$r12$													
I_{14}		$r13$		$r13$	$r13$													
I_{15}		$r5$			$r5$													
I_{17}	dI_2		dI_{17}	dI_{29}			dI_{18}	dI_{19}	dI_{20}	dI_{21}	dI_{22}		I_{13}			I_{14}	I_{25}	I_{26}
I_{18}		$r10$		$r10$	$r10$													
I_{19}		$r11$		$r11$	$r11$													
I_{20}		$r14$		$r14$	$r14$													
I_{21}		$r15$		$r15$	$r15$													
I_{22}		$r16$		$r16$	$r16$													
I_{25}				dI_{35}														
I_{26}				$r8$	dI_{36}													
I_{29}		$r6$		$r6$	$r6$													
I_{35}		$r7$		$r7$	$r7$													
I_{36}	dI_2		dI_{17}				dI_{18}	dI_{19}	dI_{20}	dI_{21}	dI_{22}		I_{13}			I_{14}	I_{39}	I_{26}
I_{39}				$r9$														

et même renuméroter les états (pour éviter les trous) :

État	{	}	[]	,	:	s	n	t	f	u	#	O	M	P	A	E	V
I_0	dI_2												I_1					
I_1												Acc						
I_2		dI_5					dI_6							I_3	I_4			
I_3		dI_7																
I_4		$r3$			dI_8													
I_5		$r1$		$r1$	$r1$							$r1$						
I_6						dI_9												
I_7		$r2$		$r2$	$r2$							$r2$						
I_8							dI_6							I_{10}	I_4			
I_9	dI_2		dI_{14}				dI_{15}	dI_{16}	dI_{17}	dI_{18}	dI_{19}		I_{11}			I_{12}		I_{13}
I_{10}		$r4$																
I_{11}		$r12$		$r12$	$r12$													
I_{12}		$r13$		$r13$	$r13$													
I_{13}		$r5$			$r5$													
I_{14}	dI_2		dI_{14}	dI_{22}			dI_{15}	dI_{16}	dI_{17}	dI_{18}	dI_{19}		I_{11}			I_{12}	I_{20}	I_{21}
I_{15}		$r10$		$r10$	$r10$													
I_{16}		$r11$		$r11$	$r11$													
I_{17}		$r14$		$r14$	$r14$													
I_{18}		$r15$		$r15$	$r15$													
I_{19}		$r16$		$r16$	$r16$													
I_{20}				dI_{23}														
I_{21}				$r8$	dI_{24}													
I_{22}		$r6$		$r6$	$r6$													
I_{23}		$r7$		$r7$	$r7$													
I_{24}	dI_2		dI_{14}				dI_{15}	dI_{16}	dI_{17}	dI_{18}	dI_{19}		I_{11}			I_{12}	I_{25}	I_{21}
I_{25}				$r9$														

2 Autres méthodes de vérification

On peut obtenir la table LR en utilisant `bison` (voir TP de compilation).

On peut aussi utiliser un outil en ligne : <http://jsmachines.sourceforge.net/machines/slrl.html> en lui fournissant la grammaire :

```

0' -> 0
0 -> { }
0 -> { M }
M -> P
M -> P , M
P -> s : V
A -> [ ]
A -> [ E ]
E -> V
E -> V , E
V -> s
V -> n
V -> 0
V -> A

```


V -> t
V -> f
V -> u

3 Automate LR

On peut représenter le fonctionnement de l'analyseur LR basé sur un automate à pile selon l'algorithme suivant :

```
/* initialisation */
Pile : Pile d'Entiers
PileJSon : Pile de void *

Empiler l'état 0 sur la Pile
/* on recupère un symbole en appelant l'analyseur lexicale */
lexeme = Lex()
/* boucle générale */
Fini = faux
TantQue non Fini Faire
    action = TableAction[sommet(Pile)][lexeme]
    Si action==Acceptation Alors
        0 = Depiler un objet JSon de la PileJSon /* l'arbre JSON est sur la pile */
        Fini = vrai
        Retourner (0)
    Sinon Si action==Déplacement Alors
        Empiler TableGoto[sommet(Pile)][lexeme] sur la Pile
        lexeme = Lex() /* on lit le symbole suivant */
    Sinon Si action==ERREUR Alors
        Afficher l'erreur et Sortir
    Sinon /* c'est une réduction, la table donne le numéro de la règle : n */
        /* on depile autant d'états que d'éléments
        * en partie droite de la règle */
        i = 0,
        Tantque i<taillePartieDroiteRegle[n]
            Depiler un état de la Pile
            i=i+1
        /* on empile le nouvel état qui résume la nouvelle situation
        * X est l'auxiliaire en partie gauche de la règle n */
        Empiler TableGoto[sommet(Pile)][X] sur la Pile
        /* la suite de l'algorithme est consacré à la construction
        * de l'arbre JSon en mémoire (si on vérifie la validité de la
        * séquence, on pouvait s'arreter ici */
        /* en fonction de la règle, il faut créer et assembler les structures JSON */
        Si n==1 Alors /* règle 0 -> { } */
            Créer un objet JSon vide et l'empiler sur PileJSon
        Sinon Si n==2 Alors /* règle 0 -> { M } */
            /* il n'y a rien à faire car l'objet M est déjà sur la pile */
        Sinon Si n==3 Alors /* règle M -> P */
            P = dépiler de la PileJSon
            Créer un objet Json, y insérer la paire P, et l'empiler sur PileJSon
```

```

Sinon Si n==4 Alors /* M -> P , M */
    O = dépiler de la PileJSon
    P = dépiler de la PileJSon
    Insérer la paire P dans l'objet O et empiler O sur PileJSon
Sinon Si n==5 Alors /* règle P -> s : V */
    Créer une paire P
    C = dépiler le conteneur de Valeur de la PileJSon
    s = dernier symbole lu par l'analyseur lexical (chaîne dans la
                                                table des symboles)

    insérer s dans P
    empiler P sur PileJSon
    effacer s de la table des symboles
/*
* continuer ainsi pour toutes les règles 6...16
*/

```