Tabelle SLR: un esempio

1

Verificare se la seguente grammatica è una grammatica SLR

$$
\begin{aligned}
&S \rightarrow AC \\
&S \rightarrow aaB \\
&A \rightarrow aA \\
&A \rightarrow e \\
&B \rightarrow bB \\
&B \rightarrow d \\
&C \rightarrow cC \\
&C \rightarrow \varepsilon
\end{aligned}
$$

Per verificare se la grammatica G in input è una grammatica SLR dobbiamo: (1) costruire la tabella SLR per G e (2) controllare se tale tabella ha entrate multidefinite o meno. Per costruire la tabella SLR abbiamo inanzitutto bisogno della collezione canonica di items LR(0).

1. Grammatica aumentata: aggiungiamo un nuovo non terminale iniziale S' e una produzione $S' \rightarrow S$ ottenendo:

$$
\begin{aligned}
&S' \rightarrow S \\
&S \rightarrow AC \\
&S \rightarrow aaB \\
&A \rightarrow aA \\
&A \rightarrow e \\
&B \rightarrow bB \\
&B \rightarrow d \\
&C \rightarrow cC \\
&C \rightarrow \varepsilon
\end{aligned}
$$

2. Costruiamo la canonica di item LR(0) a partire da $I_0 = closure(\{S' \rightarrow \bullet S\})$

1.1 Piccola nota per il calcolo delle funzioni *closure* e *goto*

Per calcolare la $closure(I)$ dove I è un insieme di items LR(0) dobbiamo:

1. inanzitutto, aggiungere in $closure(I)$ ogni item in I
2. ricorsivamente, cercare in $closure(I)$ item con un non terminale immediatamente a destra del punto, cioè item della forma $A \rightarrow \alpha \bullet B\beta$ e aggiungere un kernel item per ognuna delle produzioni di B

Per calcolare la $goto(I, X)$ dove I è un insieme di items LR(0) e X è un simbolo della grammatica dovete:

1. cercare in I item con il simbolo X immediatamente a destra del punto, e quindi item della forma $A \to \alpha \bullet X\beta$
2. spostare il punto ottenendo così un insieme di item della forma $A \to \alpha X \bullet \beta$
3. calcolare la chiusura dell'insieme di item ottenuto

$$
\begin{array}{ll}
I_0 = closure(\{S' \to \bullet S\}) = \{ & \\
\quad S' \to \bullet S, & \\
\quad S \to \bullet AC, & \textit{un kernel item per ognuna della produzioni di } S \\
\quad S \to \bullet aaB, & \\
\quad A \to \bullet aA, & \textit{un kernel item per ognuna della produzioni di } A \\
\quad A \to \bullet e & \\
\} &
\end{array}
$$

In generale, dato un insieme di item I, la funzione *goto* deve essere calcolata per tutti i simboli della grammatica (terminali o non) che si trovano immediatamente a destra del punto in qualche item di I. Nel caso di I_0 calcoliamo la goto dei simboli S, A, a, e

goto(I_0, S): l'insieme I_0 contiene un solo item con S immediatamente a destra del punto, l'item $S' \to \bullet S$, spostiamo in avanti il $\bullet$ e calcoliamo la chiusura ottenendo:

$$goto(I_0, S) = closure(\{S' \to S\bullet\}) = \{S' \to S\bullet\} = I_1$$

goto(I_0, A): l'unico item in I_0 che contiene A a a destra del punto è $S \to \bullet AC$. Di nuovo spostiamo il punto e calcoliamo la chiusura. Quindi:

$$
\begin{array}{lll}
goto(I_0, A) & = closure(\{S \to A \bullet C\}) = & \\
\qquad \{ & & \\
 & S \to A \bullet C, & \\
 & C \to \bullet cC & \textit{un kernel item per ognuna della produzioni di } C \\
 & C \to \bullet & \\
\qquad \} & = I_2 &
\end{array}
$$

In maniera simile calcoliamo la *goto*(I_0, a) e la *goto*(I_0, e)

$$
\begin{array}{ll}
goto(I_0, a) & = closure(\{S \to a \bullet aB, A \to a \bullet A\}) = \\
\qquad \{ & \\
 & S \to a \bullet aB, \quad A \to a \bullet A, \\
 & A \to \bullet aA, \quad A \to \bullet e \\
\qquad \} & = I_3
\end{array}
$$

$$goto(I_0, e) = closure(\{A \to e\bullet\}) = \{A \to e\bullet\} = I_4$$

Per ogni simbolo X della grammatica, *goto*$(I_1, X) = \emptyset$ (l'unico item in I_1, $S' \to S\bullet$, non ha simboli dopo il punto). Consideriamo I_2 e calcoliamo la *goto* rispetto ai simboli C e c (simboli a destra del punto degli item in I_2)

$goto(I_2, C) = closure(\{S \to AC\bullet\}) = \{S \to AC\bullet\} = I_5$

$$
\begin{aligned}
goto(I_2, c) = \quad & closure(\{C \rightarrow c \bullet C\}) = \\
\{ & \\
& C \rightarrow c \bullet C, \quad C \rightarrow \bullet cC, \quad C \rightarrow \bullet \\
\} & \; = I_6
\end{aligned}
$$

$$goto(I_3, A) = closure(\{A \rightarrow aA\bullet\}) = \{A \rightarrow aA\bullet\} = I_7$$

$$
\begin{aligned}
goto(I_3, a) = \quad & closure(\{S \rightarrow aa \bullet B, A \rightarrow a \bullet A\}) = \\
\{ & \\
& S \rightarrow aa \bullet B, \quad A \rightarrow a \bullet A \\
& B \rightarrow \bullet bB, \quad B \rightarrow \bullet d \\
& A \rightarrow \bullet aA, \quad A \rightarrow \bullet e \\
\} & \; = I_8
\end{aligned}
$$

$goto(I_3, e) = closure(\{A \rightarrow e\bullet\}) = I_4$

La $goto(I_4, X) = goto(I_5, X) = \emptyset$ per ogni simbolo X della grammatica (nessun simbolo dopo il punto – vedi quanto detto per l'insieme I_1).

$goto(I_6, C) = closure(\{C \rightarrow cC\bullet\}) = \{C \rightarrow cC\bullet\} = I_9$
$goto(I_6, c) = closure(\{C \rightarrow c \bullet C\}) = I_6$

$goto(I_8, B) = closure(\{S \rightarrow aaB\bullet\}) = \{S \rightarrow aaB\bullet\} = I_{10}$

$goto(I_8, A) = closure(\{A \rightarrow aA\bullet\}) = I_7$

$$
\begin{aligned}
goto(I_8, b) = \quad & closure(\{B \rightarrow b \bullet B\}) = \\
\{ & \\
& B \rightarrow b \bullet B, \\
& B \rightarrow \bullet bB, \; B \rightarrow \bullet d \\
\} & \; = I_{11}
\end{aligned}
$$

$$
\begin{aligned}
goto(I_8, a) = \quad & closure(\{A \rightarrow a \bullet A\}) = \\
\{ & \\
& A \rightarrow a \bullet A, \\
& A \rightarrow \bullet aA, \; A \rightarrow \bullet e \\
\} & \; = I_{12}
\end{aligned}
$$

$goto(I_8, d) = closure(\{B \rightarrow d\bullet\}) = \{B \rightarrow d\bullet\} = I_{13}$

$goto(I_8, e) = closure(\{A \rightarrow e\bullet\}) = I_4$

$goto(I_{11}, B) = closure(\{B \rightarrow bB\bullet\}) = I_{14}$

$goto(I_{11}, b) = closure(\{B \rightarrow b \bullet B\}) = I_{11}$

$goto(I_{11}, d) = closure(\{B \rightarrow d\bullet\}) = I_{13}$

$goto(I_{12}, A) = closure(\{A \rightarrow aA\bullet\}) = I_7$

$goto(I_{12}, a) = closure(\{A \rightarrow a \bullet A\}) = I_{12}$

$goto(I_{12}, d) = closure(\{A \to e\bullet\}) = I_4$

1.2 Calcolo della FOLLOW

L'ultimo ingrediente di cui abbiamo bisogno è la FOLLOW (e quindi la FIRST) di tutti i non terminali della grammatica aumentata

FIRST$(C) = \{c, \varepsilon\}$
FIRST$((B) = \{b, d\}$
FIRST$(A) = \{a, e\}$
FIRST$(S) = \{a, e\} =$ FIRST(S')

FOLLOW$(S') = \{\$\} =$ FOLLOW(S) (nessuna produzione con S' a destra e quindi la FOLLOW(S') contiene solo il \$; l'unica produzione con S a destra è $S' \to S$: aggiungiamo a FOLLOW(S) tutti i simboli in FOLLOW(S') e quindi solo il \$)

FOLLOW(A): l'unica produzione con A a destra significativa è $S \to AC$ (l'altra $A \to aA$ non ci fornisce informazioni utili). Inanzitutto, aggiungiamo in FOLLOW(A) ogni terminale in FIRST$(C) = \{c, \varepsilon\}$ (quindi aggiungiamo c). Inoltre, visto $\varepsilon \in$ FIRST(C) (e quindi $C \stackrel{*}{\Rightarrow} \varepsilon$) aggiungiamo in FOLLOW$(A)$ tutti i simboli in FOLLOW$(S) = \{\$\}$. Ricapitolando: FOLLOW$(A) = \{c, \$\}$. In maniera simile abbiamo che FOLLOW$(B) =$ FOLLOW$(C) = \{\$\}$

2 Costruzione effettiva della tabella

I possibili stati del parser sono 15, denotati da $s_0, \ldots, s_{14}$. Lo stato s_j corrisponde all'insieme I_j della collezione canonica. La tabella di un parsing è divisa in due parti: una parte *action* e una parte *goto*. Le righe della tabella sono indicizzate (sia nella parte *action* che in quella *goto*) da stati; le colonne della parte *action* sono indicizzate da **terminali** e dal \$, le colonne della parte goto sono indicizzate da **non terminali**. Ripertiamo le regole per la costruzione della tabella SLR:

1. **parte goto**: si costruisce direttamente dalla funzione *goto* ed in particolare dalla *goto* di insiemi di item e *non terminali* in base alla seguente regola:

   se $goto(I_j, A) = I_k$ (qui A è un non terminale) allora poni $goto[s_j, A] = s_k$

   Per brevità sulla tabella scriveremo semplicemente l'indice dello stato. Poichè:

   $goto(I_0, S) = I_1$, poniamo $goto[s_0, S] = 1$

   $goto(I_0, A) = I_2$, poniamo $goto[s_0, A] = 2$

   $goto(I_2, C) = I_5$, poniamo $goto[s_2, C] = 5$

   $goto(I_3, A) = I_7$, poniamo $goto[s_3, A] = 7$

   $goto(I_6, C) = I_9$, poniamo $goto[s_6, C] = 9$

   $goto(I_8, B) = I_{10}$, poniamo $goto[s_8, B] = 10$

   $goto(I_8, A) = I_7$, poniamo $goto[s_8, A] = 7$

   $goto(I_{11}, B) = I_{14}$, poniamo $goto[s_{11}, B] = 14$

   $goto(I_{12}, A) = I_7$, poniamo $goto[s_{12}, A] = 7$

2. **parte action – shift**: di nuovo, si costruisce direttamente dalla funzione *goto*; in questo caso dalla *goto* di insiemi di item e *terminali*

se $goto(I_j, a) = I_k$ (qui a è un terminale) allora poni $action[s_j, a] =$ shift s_k

Per brevità sulla tabella scriveremo semplicemente S (per shift) e l'indice k dello stato (quindi scriveremo Sk). Poichè:

$goto(I_0, a) = I_3$, poniamo $action[s_0, a] = \mathrm{S3}$
$goto(I_0, e) = I_4$, poniamo $action[s_0, e] = \mathrm{S4}$
$goto(I_2, c) = I_6$, poniamo $action[s_2, c] = \mathrm{S6}$
$goto(I_3, a) = I_8$, poniamo $action[s_3, a] = \mathrm{S8}$
$goto(I_3, e) = I_4$, poniamo $action[s_3, e] = \mathrm{S4}$
$goto(I_6, c) = I_6$, poniamo $action[s_6, c] = \mathrm{S6}$
$goto(I_8, b) = I_{11}$, poniamo $action[s_8, b] = \mathrm{S11}$
$goto(I_8, a) = I_{12}$, poniamo $action[s_8, a] = \mathrm{S12}$
$goto(I_8, d) = I_{13}$, poniamo $action[s_8, d] = \mathrm{S13}$
$goto(I_8, e) = I_4$, poniamo $action[s_8, e] = \mathrm{S4}$
$goto(I_{11}, b) = I_{11}$, poniamo $action[s_{11}, b] = \mathrm{S11}$
$goto(I_{11}, d) = I_{13}$, poniamo $action[s_{11}, d] = \mathrm{S13}$
$goto(I_{12}, a) = I_{12}$, poniamo $action[s_{12}, a] = \mathrm{S12}$
$goto(I_{12}, d) = I_4$, poniamo $action[s_{12}, d] = \mathrm{S4}$

3. **parte action – reduce**: per capire se, e in corrispondenza di quali simboli, nello stato s_j eseguiamo una riduzione dobbiamo controllare se l'insieme di item I_j (l'insieme di item corrispondente ad s_j) contiene item della forma $A \to \alpha\bullet$ (ossia item con un punto in fondo) se $A \to \alpha\bullet$ appartiene all'insieme I_j allora, per ogni simbolo a in FOLLOW(A), poni $action[s_j, a] =$ reduce $A \to \alpha$

   Per brevità sulla tabella scriveremo R (per reduce) e il numero progresso della produzione $A \to \alpha$. Assumiamo di numerare le produzioni della nostra grammatica aumentata come segue:

$$
\begin{array}{ll}
 & S' \to S \\
(1) & S \to AC \\
(2) & S \to aaB \\
(3) & A \to aA \\
(4) & A \to e \\
(5) & B \to bB \\
(6) & B \to d \\
(7) & C \to cC \\
(8) & C \to \varepsilon
\end{array}
$$

   Ora:

   I_0 non contiene items con il punto in fondo. I_1 contiene un item di tale forma, ma questo item è quello che corrisponde all'entrata di accettazione (vedi avanti).

   poichè $C \to \bullet \in I_2$ e FOLLOW$(C) = \{\$\}$, poniamo $action[s_2, \$] = \mathrm{R8}$

   poichè $A \to e\bullet \in I_4$ e FOLLOW$(A) = \{\$, c\}$, poniamo $action[s_4, \$] = action[s_4, c] = \mathrm{R4}$

   poichè $S \to AC\bullet \in I_5$ e FOLLOW$(S) = \{\$\}$ poniamo $action[s_5, \$] = \mathrm{R1}$

poichè $C \to \bullet \in I_6$ e FOLLOW$(C) = \{\$\}$, poniamo $action[s_6, \$] = $ R8

poichè $A \to aA\bullet \in I_7$ e FOLLOW$(A) = \{\$, c\}$ poniamo $action[s_7, \$] = action[s_7, c] = $ R3

I_8 non contiene items della forma $A \to \alpha\bullet$

poichè $C \to cC\bullet \in I_9$ e FOLLOW$(C) = \{\$\}$ poniamo $action[s_9, \$] = $ R7

poichè $S \to aaB\bullet \in I_{10}$ e FOLLOW$(S) = \{\$\}$ poniamo $action[s_{10}, \$] = $ R2

I_{11} e I_{12} non contiengono items della forma $A \to \alpha\bullet$

poichè $B \to d\bullet \in I_{13}$ e FOLLOW$(B) = \{\$\}$ poniamo $action[s_{13}, \$] = $ R6

poichè $B \to bB\bullet \in I_{14}$ e FOLLOW$(B) = \{\$\}$ poniamo $action[s_{14}, \$] = $ R5

Manca una sola entrata della tabella tabella, quella che definisce in quale stato accettare eventualmente la stringa in input. Questa entrata viene settata in base alla seguente regola:

$$\text{Se } S' \to S\bullet \in I_j \text{ allora } action[s_i, \$] = \text{acc}$$

Nel nostro caso poichè $S' \to S\bullet \in I_1$ allora $action[s_1, \$] = $ acc. Ricapitolando la tabella SLR per la grammatica in input è:

	a	b	c	d	e	\$	S	A	B	C
s_0	S3				S4		1	2		
s_1						acc				
s_2			S6			R8				5
s_3	S8				S4			7		
s_4			R4			R4				
s_5						R1				
s_6			S6							9
s_7			R3			R3				
s_8	S12	S11		S13	S4			7	10	
s_9						R7				
s_{10}						R2				
s_{11}		S11		S13					14	
s_{12}	S12				S4			7		
s_{13}						R6				
s_{14}						R5				

La tabella non ha entrate multidefinite e quindi è una tabella SLR

3

Esercizio 3.1 *Si dimostri che la seguente grammatica non è SLR:*

$$\begin{aligned}
S &\to AB \\
S &\to bB \\
A &\to aAa \\
A &\to \varepsilon \\
B &\to bB \\
B &\to c
\end{aligned}$$

Aumentiamo la grammatica in input ottenendo:

$$\begin{array}{ll}
 & S' \to S \\
1 & S \to AB \\
2 & S \to bB \\
3 & A \to aAa \\
4 & A \to \varepsilon \\
5 & B \to bB \\
6 & B \to c
\end{array}$$

Calcoliamo le funzioni FIRST e FOLLOW per la grammatica aumentata.

	FIRST	FOLLOW
S	$\{a, b, c\}$	$\{\$\}$
A	$\{a, \varepsilon\}$	$\{a, b, c\}$
B	$\{b, c\}$	$\{\$\}$

Stato iniziale:

$$I_0 = closure(\{S' \to \bullet S\}) = \{S' \to \bullet S,\ S \to \bullet AB\ S \to \bullet bB,\ A \to \bullet aAa,\ A \to \bullet\}$$

Inanzitutto, $A \to \bullet \in I_0$ e $\text{FOLLOW}(A) = \{a, b, c\}$ implica

$$action[s_0, a] = action[s_0, b] = action[s_0, c] = R4$$

Inoltre:

$$goto(I_0, a) = closure(\{A \to a \bullet Aa\}) = \{A \to a \bullet Aa, A \to \bullet aAa,\ A \to \bullet\} = I_1$$

da cui otteniamo

$$action[s_0, a] = S1$$

La tabella presenta un conflitto shift/reduce in corrispondenza della entry $action[s_0, a]$. La grammatica in input non è SRL