Tabelle LR

Costruzione delle tabelle di parsing LR canoniche

Maria Rita Di Berardini

Dipartimento di Matematica e Informatica
Universitá di Camerino
mariarita.diberardini@unicam.it

Limite della metodologia SLR

- In uno stato s_i la tabella indica una riduzione con una produzione $A \to \alpha$ se l'insieme di item I_i contiene l'item $A \to \alpha\bullet$ e il simbolo corrente è un $a \in \text{FOLLOW}(A)$
- Tuttavia, in alcune situazioni, quando lo stato s_i appare in testa allo stack, il viable prefix $\beta\alpha$ sullo stack non può essere seguito da a in nessuna forma sentenziale destra
- Quindi la riduzione $A \to \alpha$ sarebbe sbagliata in corrispondenza del simbolo in input a
- Esempio: consideramo di nuovo la grammatica G'

$$
\begin{aligned}
S' &\to S \\
S &\to L = R \\
S &\to R \\
L &\to *R \\
L &\to \mathbf{id} \\
R &\to L
\end{aligned}
$$

Limite della metodologia SLR

- La collezione canonica di item LR(0) di G' contiene in insieme di items

$$I_2 = \{S \to L\bullet = R, R \to L\bullet\}$$

- La presenza di $R \to L\bullet$ in I_2 implica che nello stato s_2 il parser effettua una riduzione in base alla produzione $R \to L$ per ogni simbolo in FOLLOW$(R) = \{\$, =\}$
- Il problema è che, effettuando la suddetta riduzione, arriviamo in uno stato in cui il prossimo handle dovrebbe cominciare con $R =$
- Ma nessuna produzione della grammatica che inizia per $R =$
- Quindi nello stato s_2 non dovrebbe essere indicata una riduzione in corrispondenza del simbolo $=$

Soluzione: aggiungiamo informazione agli stati

- Ogni stato deve contenere, oltre agli item validi, anche i simboli di input che possono seguire un handle α riducibile al non terminale A
- Gli item vengono ridefiniti: sono delle coppie $[A \to \alpha \bullet \beta, a]$ dove:
  - $A \to \alpha\beta$ è una produzione (quindi, $A \to \alpha \bullet \beta$ è un "vecchio" item LR(0)) ed
  - a è un terminale oppure il \$
- Questi item che contengono anche un simbolo terminale vengono chiamati item LR(1) (qui 1 indica che si considera un simbolo di lookahead, cioè la seconda componente è un solo simbolo)
- Il lookahead non ha effetti su items della forma $[A \to \alpha \bullet \beta, a]$ dove $\beta \neq \varepsilon$
- Se l'item è della forma $[A \to \alpha\bullet, a]$ allora si esegue una riduzione solo se il corrente simbolo in input è a: non riduciamo per tutti i simboli in FOLLOW(A), ma per un sottoinsieme (proprio) di questi

Items LR(1) validi

- Un item LR(1) $[A \to \alpha \bullet \beta, a]$ è valido per il viable prefix γ se e solo se esiste una derivazione rightmost

$$S \overset{*}{\Rightarrow}_{rm} \delta A w \Rightarrow_{rm} \delta \alpha \beta w$$

tale che:

1. $\gamma = \delta\alpha$ e
2. $w = aw'$ ($w \neq \varepsilon$ ed a è il primo simbolo di w) oppure $w = \varepsilon$ ed $a = \$$

La condizione

1. ci dice come completare il viable prefix per formare il prossimo handle
2. ci dice che il simbolo a deve seguire A nel momento in cui rimpiazziamo l'handle

Items LR(1) validi: un esempio

- Consideramo la grammatica

$$S \to BB$$
$$B \to aB \mid b$$

- E una possibile derivazione rightmost:

$$S \overset{*}{\Rightarrow}_{rm} aa\ B\ ab \Rightarrow_{rm} aa\ aB\ ab$$

- L'item $[B \to a \bullet B, a]$ è un item valido per il viable prefix $\gamma = aaa$ ponendo $\delta = aa$, $A = B$, $\alpha = a$ e $\beta = B$ nella definizione

Items LR(1) validi

- Un item LR(1) $[A \to \alpha \bullet \beta, a]$ è valido per il viable prefix γ se e solo se esiste una derivazione rightmost

$$S \stackrel{*}{\Rightarrow}_{rm} \delta A w \stackrel{*}{\Rightarrow}_{rm} \delta\alpha\beta w$$

tale che: (1) $\gamma = \delta\alpha$ e (2) $w = aw'$ oppure $w = \varepsilon$ ed $a = \$$

Riscriviamo la condizione di validità per item della forma $[A \to \bullet\beta, a]$, ossia in casi in cui $\alpha = \varepsilon$

- Un item LR(1) $[A \to \bullet\beta, a]$ è valido per il viable prefix γ se e solo se esiste una derivazione rightmost

$$S \stackrel{*}{\Rightarrow}_{rm} \delta A w \stackrel{*}{\Rightarrow}_{rm} \delta\beta w$$

tale che: (1) $\gamma = \delta$ e (2) $w = aw'$ oppure $w = \varepsilon$ ed $a = \$$

Costruzione della collezione canonica

- La costruzione della collezione canonica di item LR(1) è la stessa vista per la collezione canonica LR(0)

- Chiaramente cambiano le definizioni delle funzioni *closure* e *goto*

  function *goto*(I, X);
  dove I è un insieme di item LR(1) ed X è un simbolo della gram.
  begin
  Sia J l'insieme di item LR(1) della forma $[A \rightarrow \alpha X \bullet \beta, a]$ tali che $[A \rightarrow \alpha \bullet X\beta, a] \in I$
  return *closure*(J);
  end;

- In realtà, l'unica modifica sostanziale è quella della funzione *closure*

Cosa sappiamo sulla *closure* di item LR(0)

- La collezione canonica di item LR(0) è un insieme $C = \{I_0, I_1, \ldots, I_n\}$
- $I_0 = \mathit{closure}(\{S' \to \bullet S\})$
- Un generico $I_k = \mathit{goto}(I_j, X) = \mathit{closure}(J)$, dove

$$J = \{A \to \alpha X \bullet \beta \mid A \to \alpha \bullet X\beta \in I_j\}$$

- Ciascuno degli insiemi I_j contiene tutti gli item validi per un qualche viable prefix γ (la stringa associata al cammino da I_0 a I_j nella DFA definito dalla funzione *goto*)
- Se $A \to \alpha \bullet B\beta \in \mathit{closure}(I)$ e $B \to \beta$ è una produzione per B allora $B \to \bullet\beta \in \mathit{closure}(I)$
- Il che significa che se $A \to \alpha \bullet B\beta \in \mathit{closure}(I)$ è valido per un qualche viable prefix γ e $B \to \beta$ è una produzione per B allora anche $B \to \bullet\beta$ è valido per γ

closure di item LR(1)

- Sia a un terminale (quindi $a \neq \$$) e assumiamo $[A \to \alpha \bullet B\beta, a]$ valido per $\gamma = \delta\alpha$; dalla definizione di validità, esiste una derivazione

$$S \overset{*}{\Rightarrow}_{rm} \delta A w \Rightarrow_{rm} \delta\alpha B\beta w$$

  con $w = aw'$ (a è il primo simbolo di w). Quindi: $S \overset{*}{\Rightarrow}_{rm} \delta\alpha B\beta a w'$
- Assumiamo che $\beta a w' \overset{*}{\Rightarrow}_{rm} by$ con by stringa di <u>terminali</u>; allora

$$S \overset{*}{\Rightarrow}_{rm} \delta\alpha B\beta a w' \overset{*}{\Rightarrow}_{rm} \delta\alpha B by$$

- Quindi, per ogni produzione $B \to \eta$ di B, $S \overset{*}{\Rightarrow}_{rm} \delta\alpha B by \Rightarrow_{rm} \delta\alpha\eta by$
- Per definizione, l'item $[B \to \bullet\eta, b]$ è valido per $\gamma = \delta\alpha$

closure di item LR(1)

- Chi è b; per il momento sappiamo che $\beta a w' \overset{*}{\Rightarrow}_{rm} by$
- Potrebbe essere il primo terminale della stringa di terminali derivati da β
- Oppure, nel caso in cui $\beta \overset{*}{\Rightarrow}_{rm} \varepsilon$, il terminale a
- In generale, $\beta \in \text{FIRST}(\beta a w') = \text{FIRST}(\beta a)$

Calcolo della *closure*(I)

begin
  closure(I) := I
  repeat
    for each ($[A \to \alpha \bullet B\beta, a] \in I$, $B \to \eta \in G'$ and
      $b \in \text{FIRST}(\beta a)$) **do**
        aggiungi $[B \to \bullet\eta, b]$ in *closure*(I)
  until (nessun nuovo item può essere aggiunto)
end;

Calcolo della *closure*(I): un esempio

$S' \to S$
$S \to CC$
$C \to cC \mid d$

e calcoliamo $I_0 = \mathit{closure}(\{[S' \to \bullet S, \$]\})$

- Inanzittuo, aggiungiamo l'item $[S' \to \bullet S, \$]$. Questo item matcha con il template $[A \to \alpha \bullet B\beta, a]$ ponendo $A = S'$, $\alpha = \varepsilon$, $B = S$, $\beta = \varepsilon$ e $a = \$$. Poichè FIRST(βa) = FIRST$(\$) = \{\$\}$, aggiungiamo solo l'item $[S \to \bullet CC, \$]$
- L'item $[S \to \bullet CC, \$]$ matcha con il template $[A \to \alpha \bullet B\beta, a]$ ponendo $A = S$, $\alpha = \varepsilon$, $B = C$, $\beta = C$ e $a = \$$.
  Poichè FIRST(βa) = FIRST$(C\$)$ = FIRST$(C) = \{c, d\}$, aggiungiamo i seguenti quattro items:
  $[C \to \bullet cC, c]$, $[C \to \bullet cC, d]$
  $[C \to \bullet d, c]$, $[C \to \bullet d, d]$

Calcolo della *closure*(I): un esempio

Ricapitolando:

$$
\begin{aligned}
I_0 \quad = \{ & \\
& [S' \to \bullet S, \$], \\
& [S \to \bullet CC, \$] \\
& [C \to \bullet cC, c], \\
& [C \to \bullet cC, d] \\
& [C \to \bullet d, c], \\
& [C \to \bullet d, d] \\
\} &
\end{aligned}
$$

Abbreviazioni

$$
\begin{aligned}
& [S' \to \bullet S], \$ \\
& [S \to \bullet CC, \$], \\
& [S \to \bullet cC, c/d], \\
& [S \to \bullet d, c/d]
\end{aligned}
$$

La funzione *goto*

function *goto*(I, X);
dove I è un insieme di item LR(1) ed X è un simbolo della gram.
begin
Sia J l'insieme di item LR(1) della forma $[A \rightarrow \alpha X \bullet \beta, a]$
tali che $[A \rightarrow \alpha \bullet X\beta, a] \in I$
return *closure*(J);
end;

Calcolo degli Items LR(1)

- Utilizziamo la stessa identica procedura che abbiamo visto per la collezione canonica LR(0), ma con le funzioni *closure* e *goto* nuove
- Anche in questo caso il risultato è un automa deterministico i cui stati sono insiemi di item LR(1)
- Continuiamo con l'esempio

Calcolo della *closure*(I): un esempio

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

$$goto(I_0, C) = closure(\{[S \to C \bullet C, \$]\}) =$$
$$\{$$
$$\quad [S \to C \bullet C, \$]$$
$$\}$$

Aggiungiamo un kernel item per ognuna della produzioni di C, e quindi qualcosa della forma $[C \to \bullet cC, ??]$ e $[C \to \bullet d, ??]$ in cui dobbiamo identicare i simboli di lookahead.

Dall'item $[\, S \to \overbrace{C}^{\alpha} \bullet \overbrace{C}^{B}, \overbrace{\$}^{a} \,]$, i simboli di lookahead appartengo all'insieme $\text{FIRST}(\beta a) = \text{FIRST}(\$) = \{\$\}$. Aggiungiamo gli items $[C \to \bullet cC, \$]$ e $[C \to \bullet d, \$]$

$$goto(I_0, C) = closure(\{[S \to C \bullet C, \$]\}) = \{$$
$$\quad [S \to C \bullet C, \$], [C \to \bullet cC, \$], [C \to \bullet d, \$]$$
$$\} = I_2$$

Calcolo della *closure*(I): un esempio

$$goto(I_0, c) = closure(\{[C \to c \bullet C, c], C \to c \bullet C, d]\}) =$$
$$\{$$
$$\quad [C \to c \bullet C, c], C \to c \bullet C, d]$$
$$\}$$

Aggiungiamo un kernel item per ognuna della produzioni di C, e quindi: $[C \to \bullet cC, ??], [C \to \bullet d, ??]$. Dobbiamo identicare i simboli di lookahead.

Dal primo item, $[\ C \to \overbrace{c}^{\alpha} \bullet \overbrace{C}^{B}, \overbrace{c}^{a}\ ]$, i simboli di lookahead sono in $\mathsf{FIRST}(\beta a) = \mathsf{FIRST}(c) = \{c\}$. Aggiungiamo gli items $[C \to \bullet cC, c]$ e $[C \to \bullet d, c]$

Per il secondo item, $[\ C \to \overbrace{c}^{\alpha} \bullet \overbrace{C}^{B}, \overbrace{d}^{a}\ ]$ i simboli di lookahead sono in $\mathsf{FIRST}(\beta a) = \mathsf{FIRST}(d) = \{d\}$. Aggiungiamo gli items $[C \to \bullet cC, d]$ e $[C \to \bullet d, d]$

Calcolo della *closure*(I): un esempio

Ricapitolando:

$$goto(I_0, c) = closure(\{[C \to c \bullet C, c], C \to c \bullet C, d]\}) =$$
$$\{$$
$$[C \to c \bullet C, c], C \to c \bullet C, d],$$
$$[C \to \bullet cC, c], [C \to \bullet d, c]$$
$$[C \to \bullet cC, d], [C \to \bullet d, d]$$
$$\}$$

o in forma abbreviata

$$goto(I_0, c) = closure(\{[C \to c \bullet C, c/d]\}) =$$
$$\{$$
$$[C \to c \bullet C, c/d],$$
$$[C \to \bullet cC, c/d], [C \to \bullet d, c/d]$$
$$\} = I_3$$

Calcolo della *closure*(I): un esempio

In maniera del tutto simile:

$$goto(I_0, d) = closure(\{[C \to d\bullet, c/d]\}) =$$
$$\{$$
$$\quad [C \to d\bullet, c/d]$$
$$\} = I_4$$

Questo conclude il calcolo della *goto* "uscenti" da I_0; $I_1 = \{[S' \to S\bullet, \$]\}$

Possiamo passare a I_2

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

Calcolo della *closure*(I): un esempio

$$goto(I_2, c) = closure(\{[C \to c \bullet C, \$]\}) =$$
$$\{$$
$$\quad [C \to c \bullet C, \$],$$
$$\quad [C \to \bullet cC, \$],$$
$$\quad [C \to \bullet d, \$]$$
$$\} = I_6$$

$$goto(I_2, d) = closure(\{[C \to d \bullet, \$]\}) =$$
$$\{$$
$$\quad [C \to d \bullet, \$]$$
$$\} = I_7$$

Calcolo della *closure*(I): un esempio

$$goto(I_3, C) = closure(\{[C \to cC\bullet, c], C \to cC\bullet, d]\}) =$$
$$\{$$
$$\quad [C \to cC\bullet, c], C \to cC\bullet, d]$$
$$\} = I_8$$

$$goto(I_3, c) = closure(\{[C \to c \bullet C, c/d]\}) = I_3$$

$$goto(I_3, d) = closure(\{[C \to d\bullet, c/d]\}) = I_4$$

$$goto(I_6, C) = closure(\{[C \to cC\bullet, \$]\}) = \{[C \to cC\bullet, \$]\} = I_9$$
$$goto(I_6, c) = closure(\{[C \to c \bullet C, \$]\}) = I_6$$
$$goto(I_6, d) = closure(\{[C \to d\bullet, \$]\}) = I_7$$

Alcune considerazioni

- Consideramo gli insiemi di item LR(1) $I_4 = \{[C \rightarrow d\bullet, c/d]\}$ ed $I_7 = \{[C \rightarrow d\bullet, \$]\}$
- Questi insiemi sono molto particolari perchè differiscono solo per la seconda componente
- Questo fenomeno è tipico per gli insiemi di item LR(1)
- Un insieme della collezione canonica LR(0), ad esempio $\{C \rightarrow d\bullet\}$, può corrispondere in generale a più insiemi di item LR(1), in questo caso I_4 ed I_7
- È come se gli stati del parser SLR venissero sdoppiati

Costruzione di tabelle LR canoniche

- Il procedimento è analogo a quello visto per la costruzione di tabelle SLR, anzi, è più semplice
- Per le riduzioni dobbiamo guardare il simbolo di lookahead dell'item e non tutti i simboli in FOLLOW(A) dove A è la parte sinistra della produzione con cui ridurre

Algoritmo

1. Costruisci la collezione canonica $C = \{I_0, I_1, \ldots, I_n\}$ per G'
2. Lo stato s_j del parser corrisponde all'insieme di items I_j
3. La parte *goto* della tabella, per ogni stato s_j e per ogni non terminale A, è costruita dalla funzione *goto* come segue:

   se $goto(I_j, A) = I_k$ (funzione) allora $goto[s_j, A] = s_k$ (tabella)
4. La parte *action* della tabella è costruita come segue:
   - se $[A \to \alpha \bullet a\beta, b] \in I_j$ e $goto(I_j, a) = I_k$ allora $action[s_j, a] = \text{shift } s_k$ (qui a è un terminale)
   - se $[A \to \alpha\bullet, a] \in I_j$, allora poni $action[s_j, a] = \text{reduce } A \to \alpha$
   - se $[S' \to S\bullet, \$] \in I_j$, allora poni $action[s_j, \$] = \text{accept}$
5. Ogni entrata indefinita corrisponde ad un errore
6. Lo stato iniziale corrisponde all'insieme che contiene $[S' \to \bullet S, \$]$

Costruzione di tabelle LR canoniche

- Come nel caso SLR, la parte goto della tabella LR è definita dalla funzione *goto*
- Le entrate "shift" si costruiscono esattamente allo stesso modo
- Come detto, in caso di riduzione, guardiamo il simbolo di lookahead dell'item
- Lo stato iniziale corrisponde all'insieme che contiene $[S' \rightarrow \bullet S, \$]$
- Tutte le entrate vuote corrispondono a "error"
- La tabella così costruita si chiama **tabella di parsing LR(1) canonica**
- Se nella tabella non ci sono entrate multidefinite allora la grammatica è una grammatica LR(1)
- Un parser LR che utilizza questa tabella si chiama parser LR(1) canonico

Costruzione della tabella: un esempio

Numeriamo le produzioni della grammatica e costruiamo la tabella

$$(0)\ S' \to S \qquad (1)\ S \to CC \qquad (2)\ C \to cC \qquad (3)\ C \to d$$

$$goto(I_0, S) = \{[S' \to S\bullet, \$]\} = I_1$$
$$goto(I_0, C) = \{[S \to C \bullet C, \$], [C \to \bullet cC, \$], [C \to \bullet d, \$]\} = I_2$$
$$goto(I_0, c) = \{[C \to c \bullet C, c/d], [C \to \bullet cC, c/d], [C \to \bullet d, c/d]\} = I_3$$
$$goto(I_0, d) = \{[C \to d\bullet, c/d]\} = I_4$$

$$goto(I_2, C) = \{[S \to CC\bullet, \$]\} = I_5$$
$$goto(I_2, c) = \{[C \to c \bullet C, \$], [C \to \bullet cC, \$], [C \to \bullet \$, \$]\} = I_6$$
$$goto(I_2, d) = \{[C \to d\bullet, \$]\} = I_7$$

$$goto(I_3, C) = \{[S \to cC\bullet, c/d]\} = I_8$$
$$goto(I_3, c) = I_3, \quad goto(I_3, d) = I_4$$

$$goto(I_6, C) = \{[C \to cC\bullet, \$]\} = I_9$$
$$goto(I_6, c) = I_6, \quad goto(I_6, d) = I_7$$

La tabella

	c	d	\$	S	C
s_0	$S3$	$S4$		1	2
s_1			ACC		
s_2	S6	S7			5
s_3	S3	S4			8
s_4	R3	R3			
s_5			R1		
s_6	S6	S7			9
s_7			R3		
s_8	R2	R2			
s_9			R2		

Parsing della stringa *ccdcd*

	c	d	\$	S	C
s_0	$S3$	$S4$		1	2
s_1			ACC		
s_2	S6	S7			5
s_3	S3	S4			8
s_4	R3	R3			
s_5			R1		
s_6	S6	S7			9
s_7			R3		
s_8	R2	R2			
s_9			R2		

Stack	Input	Azione
s_0	$ccdcd\$$	shift S3
$s_0\, c\, s_3$	$cdcd\$$	shift S3
$s_0\, c\, s_3\, c\, s_3$	$dcd\$$	shift S4
$s_0\, c\, s_3\, c\, s_3\, d s_4$	$cd\$$	red. $C \to d$
$s_0\, c\, s_3\, c\, s_3\, C s_8$	$cd\$$	red. $C \to cC$
$s_0\, c\, s_3\, C s_8$	$cd\$$	red. $C \to cC$
$s_0\, C s_2$	$cd\$$	shift S6
$s_0\, C s_2\, c\, s_6$	$d\$$	shift S7
$s_0\, C s_2\, c\, s_6\, d\, s_7$	\$	red. $C \to d$
$s_0\, C s_2\, c\, s_6\, C\, s_9$	\$	red. $C \to cC$
$s_0\, C s_2\, C s_5$	\$	red. $S \to CC$
$s_0 S s_1$	\$	acc