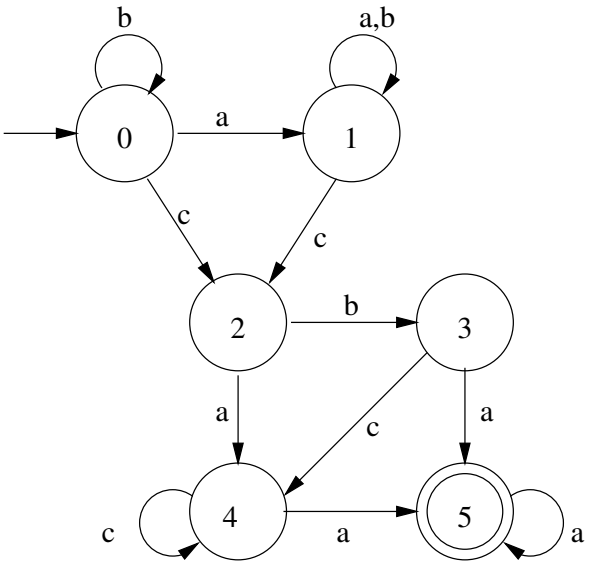
Linguaggi di Programmazione e Compilatori

V° Appello del 17/12/2004

ISTRUZIONI: Scrivere **in stampatello** COGNOME e NOME su ogni foglio. Vanno consegnati **tutti** i fogli: brutta copia e testo compresi. Coloro che non vogliono consegnare possono andarsene, consegnando il testo, dopo un'ora dall'inizio del compito ed entro 15 minuti dalla scadenza del tempo.
NOTA: Nelle espressioni regolari si possono usare le usuali convenzioni di precedenza: l'operatore * lega più della concatenazione che, a sua volta, lega più dell'operatore $|$. Inoltre si può usare l'abbreviazione $+$ con il solito significato.

ESERCIZIO 1 (6 punti)

Si consideri il seguente automa:



1. Si scriva il linguaggio accettato tramite una espressione regolare
2. Si dica se l'automa è minimo e, nel caso non lo fosse, minimizzarlo.

SOLUZIONE

L'espressione regolare più semplice per descrivere il linguaggio è

$$(a \mid b)^* \, c \, (a \mid b) \, c^* \, a^+$$

L'automa è già deterministico: proviamo quindi direttamente a minimizzarlo. Innanzitutto aggiungiamo uno stato fittizio d che funzionerà come *dead state* e non sarà di accettazione. Poi iniziamo ad applicare l'algoritmo di minimizzazione.

La prima suddivisione degli stati in partizioni è la seguente: $(0\,1\,2\,3\,4\,d)$ e (5). Al primo passo quindi dovremmo partizionare il primo gruppo di stati, mentre il secondo (contenente un solo stato) rimarrà invariato fino alla fine dell'algoritmo.

Le seguenti transizioni

$$\begin{array}{lll} 0 & \xrightarrow{a} & 1 \\ 1 & \xrightarrow{a} & 2 \\ 2 & \xrightarrow{a} & 4 \\ d & \xrightarrow{a} & d \end{array} \qquad \begin{array}{lll} 3 & \xrightarrow{a} & \underline{\mathbf{5}} \\ 4 & \xrightarrow{a} & \underline{\mathbf{5}} \end{array}$$

inducono la prima separazione del gruppo $(0\,1\,2\,3\,4\,d)$ in $(0\,1\,2\,d)$ e $(3\,4)$ poiché questi ultimi si differenziano dagli altri per la partizione di stati in cui si arriva prendendo la transizione etichettata a. Per quanto riguarda gli altri simboli dell'alfabeto non si hanno comportamenti differenti. Pertanto il primo passo si conclude con il nuovo partizionamento: $(0\,1\,2\,d)$, $(3\,4)$ e (5).

Al secondo passo ci accorgiamo che gli stati 3 e 4 sono indistinguibili nel partizionamento attuale e quindi il gruppo formato da loro rimane inalterato. Per quanto riguarda il gruppo $(0\,1\,2\,d)$ invece notiamo che:

$$\begin{array}{lll} 0 & \xrightarrow{a} & 1 \\ 1 & \xrightarrow{a} & 2 \\ 2 & \xrightarrow{a} & \underline{\mathbf{4}} \\ d & \xrightarrow{a} & d \end{array} \qquad \begin{array}{lll} 0 & \xrightarrow{b} & 0 \\ 1 & \xrightarrow{b} & 1 \\ 2 & \xrightarrow{b} & \underline{\mathbf{3}} \\ d & \xrightarrow{b} & d \end{array}$$

Quindi lo stato 2 si differenzia dagli altri dello stesso gruppo per quanto riguarda i simboli uscenti a e b. Il simbolo c non induce nessuna differenza. Pertanto la nuova partizione calcolata dal secondo passo è $(0\,1\,d)$, (2), $(3\,4)$ e (5).

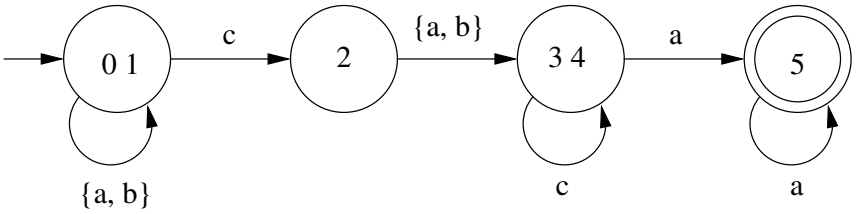
Al terzo passo di nuovo gli stati 3 e 4 risultano indistinguibili. Per il primo gruppo invece abbiamo che:

$$
\begin{array}{lcl}
0 & \xrightarrow{c} & 2 \\
1 & \xrightarrow{c} & 2 \\
d & \xrightarrow{c} & \underline{\mathbf{d}}
\end{array}
$$

Quindi lo stato d si differenzia da 1 e da 2. I simboli a e b non inducono altre differenze. La partizione calcolata al terzo passo è $(0\,1)$, (2), $(3\,4)$, (5) e (d).

Al quarto passo ci accorgiamo che gli stati 0 e 1 sono ancora indistingibili, così come gli stati 3 e 4. Pertanto l'algoritmo, non potendo effettuare ulteriori partizioni, termina.

Togliendo lo stato d fittizio l'automa minimo risulta essere il seguente:



ESERCIZIO 2 (16 punti)

Si consideri la seguente grammatica:

$$
\begin{array}{lcl}
S & \rightarrow & aA \mid B \mid \epsilon \\
A & \rightarrow & aA \mid C \\
C & \rightarrow & aCb \mid \epsilon \\
B & \rightarrow & CD \\
D & \rightarrow & bD \mid b
\end{array}
$$

1. Si indichi il linguaggio generato mediante una espressione su insiemi
2. Si dimostri che la grammatica non è SLR(1)
3. Si dimostri che la grammatica è LALR e si dia la tabella di analisi del relativo analizzatore

SOLUZIONE

Il linguaggio generato dalla grammatica è il seguente:

$$L = \{\epsilon\} \cup \{a^n\, a^m\, b^m \mid n > 0, m \geq 0\} \cup \{a^n\, b^n\, b^m \mid n \geq 0, m > 0\}$$

Proviamo a calcolare l'insieme iniziale di item LR(0).

Si ha $I_0 = \{S' \to \cdot S, S \to \cdot aA, S \to \cdot B, S \to \cdot, B \to \cdot CD, C \to \cdot aCb, C \to \cdot\}$.

Poiché FOLLOW$(S) = \{\$\}$ e FOLLOW$(C) = \{\$, b\}$ abbiamo in I_0 un conflitto reduce/reduce per il $\$$. Difatti l'insieme di item ci dice di ridurre, in presenza del $\$$, sia con la produzione 3 che con la produzione 7 (assumendo la consueta numerazione). Pertanto la grammatica non è SLR(1).

Per dimostrare che la grammatica è LALR dobbiamo calcolare la collezione di insiemi di item LR(1), accorpare gli stati con lo stesso core e controllare che non ci siano conflitti. Iniziamo con il calcolo della collezione.

$I_0 = \begin{array}{lcl} S' & \to & \cdot S, \$ \\ S & \to & \cdot aA, \$ \\ S & \to & \cdot B, \$ \\ S & \to & \cdot, \$ \\ B & \to & \cdot CD, \$ \\ C & \to & \cdot aCb, b \\ C & \to & \cdot, b \end{array}$	$I_1 = \texttt{goto}(I_0, S) = S' \to S\cdot, \$$
$I_2 = \texttt{goto}(I_0, a) = \begin{array}{lcl} S & \to & a \cdot A, \$ \\ C & \to & a \cdot Cb, b \\ A & \to & \cdot aA, \$ \\ A & \to & \cdot C, \$ \\ C & \to & \cdot aCb, b/\$ \\ C & \to & \cdot, b/\$ \end{array}$	$I_3 = \texttt{goto}(I_0, B) = S \to B\cdot, \$$
$I_4 = \texttt{goto}(I_0, C) = \begin{array}{lcl} B & \to & C \cdot D, \$ \\ D & \to & \cdot bD, \$ \\ D & \to & \cdot b, \$ \end{array}$	$I_5 = \texttt{goto}(I_2, A) = S \to aA\cdot, \$$ $I_6 = \texttt{goto}(I_2, C) = \left\{ \begin{array}{lcl} C & \to & aC \cdot b, b \\ A & \to & C\cdot, \$ \end{array} \right.$
$I_7 = \texttt{goto}(I_2, a) = \begin{array}{lcl} A & \to & a \cdot A, \$ \\ C & \to & a \cdot Cb, b/\$ \\ A & \to & \cdot aA, \$ \\ A & \to & \cdot C, \$ \\ C & \to & \cdot aCb, b/\$ \\ C & \to & \cdot, b/\$ \end{array}$	$I_8 = \texttt{goto}(I_4, D) = B \to CD\cdot, \$$ $I_9 = \texttt{goto}(I_4, b) = \left\{ \begin{array}{lcl} D & \to & b \cdot D, \$ \\ D & \to & b\cdot, \$ \\ D & \to & \cdot bD, \$ \\ D & \to & \cdot b, \$ \end{array} \right.$
$I_{10} = \texttt{goto}(I_6, b) = C \to aCb\cdot, b$ $I_{11} = \texttt{goto}(I_7, a) = A \to aA\cdot, \$$	$I_{12} = \texttt{goto}(I_7, C) = \begin{array}{lcl} C & \to & aC \cdot b, b/\$ \\ A & \to & C\cdot, \$ \end{array}$
$\texttt{goto}(I_7, a) = I_7$ $I_{13} = \texttt{goto}(I_9, D) = D \to bD\cdot, \$$	$\texttt{goto}(I_9, b) = I_9$ $I_{14} = \texttt{goto}(I_{12}, b) = C \to aCb\cdot, b/\$$

Notiamo innanzitutto che gli stati non presentano conflitti e quindi possiamo dire che la grammatica è LR(1). Tuttavia è richiesta la dimostrazione che sia LALR(1). Gli insiemi I_6 e I_{12} hanno lo stesso core. Così come gli insiemi I_{10} e I_{14}. Quindi, per calcolare gli stati dell'analizzatore LALR, dobbiamo accorparli e ottenere gli stati 612 e 1014.

La tabella di parsing è la seguente:

	a	b	\$	S	A	B	C	D
0	s2	r7	r3	1		3	4	
1			acc					
2	s7	r7	r7		5		612	
3			r2					
4		s9						8
5			r1					
612		s1014	r5					
7	s7	r7	r7		11		612	
8			r8					
9		s9	r10					13
1014		r6	r6					
11			r4					
13			r9					

ESERCIZIO 3 (9 punti)

Si consideri il linguaggio delle espressioni definite induttivamente nel seguente modo:

i a, b, e c sono espressioni

ii Se e è un'espressione allora anche $a(e)$, $b(e)$ e $c(e)$ sono espressioni

Si dia una grammatica LL(1) per il linguaggio e si mostri la tabella di analisi del relativo parser predittivo.

SOLUZIONE

Come al solito scriviamo la grammatica già con l'ottica di una analisi predittiva. La struttura ricorsiva delle espressioni da definire ci aiuta a scegliere il tipo di ricorsione da usare nella grammatica: senz'altro una ricorsione destra a più casi. Tuttavia, per evitare conflitti sui FIRST delle produzioni con la stessa testa, usiamo il tipico espediente della ricorsione indiretta:

$$\begin{array}{lcl} S & \rightarrow & aS' \mid bS' \mid cS' \\ S' & \rightarrow & (S) \mid \epsilon \end{array}$$

Si ha FOLLOW(S) = FOLLOW(S') = $\{\$,)\}$. La tabella del parser predittivo è la seguente

	a	b	c	$($	$)$	\$
S	$S \rightarrow aS'$	$S \rightarrow bS'$	$S \rightarrow cS'$			
S'				$S' \rightarrow (S)$	$S' \rightarrow \epsilon$	$S' \rightarrow \epsilon$