Linguaggi di Programmazione e Compilatori

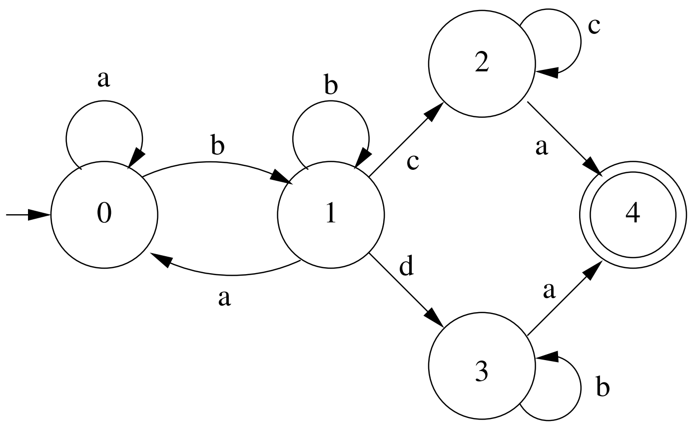
VI° Appello del 10/01/2005

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 (7 punti)

Si dia un automa minimo che accetta il linguaggio $(a|b)^*\, b\, c^+\, a \mid (a|b)^*\, b\, d\, b^*\, a$.

SOLUZIONE



ESERCIZIO 2 (12 punti)

Si consideri la seguente grammatica:

$$
\begin{array}{lcl}
S & \rightarrow & B \mid Caa \\
B & \rightarrow & bC \\
C & \rightarrow & bbCa \mid \epsilon
\end{array}
$$

1. Si indichi il linguaggio generato mediante una espressione su insiemi
2. La grammatica è LR(1)? Se sì, si dia la tabella di un parser shift-reduce e si mostri il parsing della stringa $bbba$.

SOLUZIONE

$$L(S) = \{b^{2n+1}\, a^n \mid n \geq 0\} \cup \{b^{2n}\, a^{n+2} \mid n \geq 0\}$$

Proviamo dapprima a vedere se la grammatica è SLR(1). In tal caso sarà anche LR(1). Calcoliamo gli insiemi di item LR(0):

$I_0 = \begin{array}{lcl} S' & \rightarrow & \cdot S \\ S & \rightarrow & \cdot B \\ S & \rightarrow & \cdot Caa \\ B & \rightarrow & \cdot bC \\ C & \rightarrow & \cdot bbCa \\ C & \rightarrow & \cdot \end{array}$	$I_1 = \texttt{goto}(I_0, S) = S' \rightarrow S\cdot$
$I_2 = \texttt{goto}(I_0, B) = S \rightarrow B\cdot$ $I_3 = \texttt{goto}(I_0, C) = S \rightarrow C \cdot aa$	$I_4 = \texttt{goto}(I_0, b) = \begin{array}{lcl} B & \rightarrow & b \cdot C \\ C & \rightarrow & b \cdot bCa \\ C & \rightarrow & \cdot bbCa \\ C & \rightarrow & \cdot \end{array}$
$I_5 = \texttt{goto}(I_3, a) = S \rightarrow Ca \cdot a$ $I_6 = \texttt{goto}(I_4, C) = B \rightarrow bC\cdot$	$I_7 = \texttt{goto}(I_4, b) = \begin{array}{lcl} C & \rightarrow & bb \cdot Ca \\ C & \rightarrow & b \cdot bCa \\ C & \rightarrow & \cdot bbCa \\ C & \rightarrow & \cdot \end{array}$
$I_8 = \texttt{goto}(I_5, a) = S \rightarrow Caa\cdot$ $I_9 = \texttt{goto}(I_7, C) = C \rightarrow bbC \cdot a$	$\texttt{goto}(I_7, b) = I_7$ $I_{10} = \texttt{goto}(I_9, a) = C \rightarrow bbCa\cdot$

Poiché FOLLOW(S') = FOLLOW(S) = FOLLOW(B) = $\{\$\}$ e FOLLOW(C) = $\{a, \$\}$ non ci sono conflitti negli insiemi di item LR(0) e quindi la grammatica è SLR(1) (e quindi anche LR(1)). La tabella di parsing è la seguente (le produzioni sono numerate nella maniera consueta):

	a	b	$\$$	S	B	C
0	r5	s4	r5	1	2	3
1			acc			
2			r1			
3	s5					
4	r5	s7	r5			6
5	s8					
6			r3			
7	r5	s7	r5			9
8			r2			
9	s10					
10	r4		r4			

Il parsing della stringa $bbba$ è rappresentato nella seguente tabella:

STACK	INPUT	AZIONE
$\$0$	$bbba\$$	shift 4
$\$0b4$	$bba\$$	shift 7
$\$0b4b7$	$ba\$$	shift 7
$\$0b4b7b7$	$a\$$	reduce 5
$\$0b4b7b7C9$	$a\$$	shift 10
$\$0b4b7b7C9a10$	$\$$	reduce 4
$\$0b4C6$	$\$$	reduce 3
$\$0B2$	$\$$	reduce 1
$\$0S1$	$\$$	accept

ESERCIZIO 3 (12 punti)

Si consideri il seguente linguaggio:

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

1. Dare una grammatica che generi tutte e sole le parole del linguaggio
2. Il linguaggio è LL(k) per qualche k?
3. Esiste un parser predittivo per il linguaggio? Se sì, se ne dia la tabella e si esegua il parsing della stringa bcc.

SOLUZIONE

Come di consueto cerchiamo subito di costruire una grammatica LL(1). Se questo è possibile, infatti, la risposta alla seconda domanda è sì, per ogni $k > 0$ e la risposta alla terza domanda è anch'essa affermativa.

Osservando il linguaggio si vede che i due casi sono quasi completamente separati. L'unico problema si ha quando la n del primo caso è zero e quindi le stringhe iniziano con b, come quelle del secondo caso. Per risolvere il problema, come al solito, isoliamo queste stringhe e consideriamo il problema di scrivere una grammatica per il linguaggio $\{a^n\, b\, c^k \mid n > 0, k > 0\} \cup \{b^n\, a\, c^k \mid n > 0, k \geq 0\} \cup \{b\, c^k \mid k > 0\}$.

Vedendo le cose in questo modo si ha che il primo caso è disgiunto dagli altri due e il secondo e terzo caso si differenziano subito dopo aver guardato la prima b. Scriviamo quindi la grammatica in modo tale che, dopo aver letto una b, si passa ad esaminare il simbolo successivo (che può essere o un'altra b o una a o una c) con un altro simbolo di categoria sintattica:

$$\begin{array}{lcl} S & \rightarrow & aA \mid bB \\ A & \rightarrow & aA \mid bcC \\ C & \rightarrow & cC \mid \epsilon \\ B & \rightarrow & cC \mid B' \\ B' & \rightarrow & bB' \mid aC \end{array}$$

Si ha FOLLOW(S) = FOLLOW(A) = FOLLOW(B) = FOLLOW(C) = FOLLOW(B') = $\{\$\}$. La grammatica è LL(1) e la tabella per il parser predittivo è la seguente:

	a	b	c	$\$$
S	$S \rightarrow aA$	$S \rightarrow bB$		
A	$A \rightarrow aA$	$A \rightarrow bcC$		
C			$C \rightarrow cC$	$C \rightarrow \epsilon$
B	$B \rightarrow B'$	$B \rightarrow B'$	$B \rightarrow cC$	
B'	$B' \rightarrow aC$	$B' \rightarrow bB'$		

Il parsing predittivo della stringa bcc è rappresentato nella seguente tabella:

STACK	INPUT	AZIONE
$\$S$	$bcc\$$	$S \rightarrow bB$
$\$Bb$	$bcc\$$	match
$\$B$	$cc\$$	$B \rightarrow cC$
$\$Cc$	$cc\$$	match
$\$C$	$c\$$	$C \rightarrow cC$
$\$Cc$	$c\$$	match
$\$C$	$\$$	$C \rightarrow \epsilon$
$\$$	$\$$	accept