Svolgimento appello di Linguaggi di Programmazione e Compilatori (Ascoli Piceno) Traccia 1

Lunedì, 1 ottobre 2007

Esercizio 1 - (9 Punti)

Si consideri il seguente linguaggio:

$$\mathcal{L} = \{(ab)^n c^m d^{2n-m} | m, n \in \mathbb{N} \text{ e } m < 2n\} \cup \{(ab)^n c^m d^{m-2n} | m, n \in \mathbb{N} \text{ e } m \geq 2n\}$$

- Si determini la classe di appartenenza del linguaggio in accordo alla classificazione di Chomsky, e si definisca un automa capace di accettare il linguaggio fornendo la definizione di tutte le sue componenti, commentando altresì le scelte effettuate.
- L'automa definito accetta stringhe del tipo $(ab)^n c^n d^n$? Se si, si discuta se questo risultato è in contrasto con il fatto che il linguaggio $\mathcal{L} = \{(ab)^n c^n d^n\}$ non é un linguaggio libero da contesto.
- Si mostrino infine i passi dell'automa sulla stringa $(ab)^2 c^2 d^2 = ababccdd$.

Svolgimento :
Punto 1: Il linguaggio non può essere regolare dato che richiede la capacità di contare per verificare ad esempio la relazione tra m ed n. La struttura fa sospettare che il linguaggio possa essere dio tipo 2 (i.e. libero da contesto). Per provare questo fatto abbiamo due strade più semplici. Costruire un automa a pila che riconosca il linguaggio oppure definire una grammatica di tipo 2 che generi il linguaggio. Procediamo dunque alla costruzione di un automa a pila.

Il linguaggio è formato da due differenti tipi di stringhe corrispondenti ai due insiemi definiti. Nella definizione dell'automa può risultare utile tener conto della relazione che intercorre tra i due naturali m ed n. I seguenti passi forniscono una descrizione informale di un possibile algoritmo per un automa a pila che accetta il linguaggio dato.

- Si inizia la lettura delle occorrenze della sottostringa "ab" memorizzandone il numero ma immagazzinando $2n$ simboli sulla pila. Questo successivamente ci permetterà di decidere se $m \geq 2n$ o se $m < 2n$.

- si procede alla lettura delle occorrenze del simbolo c decrementando il contenuto della pila. Se si termina la lettura di simboli c prima di aver svuotata la pila si utilizzeranno i restanti simboli sulla pila per il conteggio delle occorrenze del simbolo d il valore sulla pila è infatti pari a $2n - m$. Nel caso in cui le occorrenze di simboli c siano maggiori di $2n$ si procede al conteggio dei simboli eccedenti "incrementando" il contatore di pila dopo averla svuotata. In tal caso al termine del conteggio dei simboli c la pila conterrà il valore $m - 2n$. In ogni caso al termine del conteggio delle occorrenze di simboli c la pila conterrà il numero corretto di occorrenze di simboli d che l'automa si aspetta.

- si procede alla lettura dei simboli d e si decrementa il contenuto della pila. Se al termine della lettura della stringa di input la pila risulta essere vuota l'automa accetterà la parola, la rifiuterà altrimenti.

Dalla descrizione dell'algoritmo si può intuire che sarà necessario avere un solo simbolo di pila per poter effetturare il conteggio.

Il seguente automa a pila $A = \langle \Sigma, \Gamma, Z_0, Q, q_0, F, \delta \rangle$ accetta per pila vuota il linguaggio $\mathcal{L}$:

$\Sigma = \{a, b, c, d\}$, $\Gamma = \{C, Z_0\}$, $Q = \{q_1, q_2, q_3, q_4, q_5, q_6, q_f\}$, $F = \{\}$ e la tabella 1 per la funzione di transizione δ.

	Z_0				
	a	b	c	d	ε
q_0	(q_1, ε)				
q_1		(q_0, CC)			
q_2			(q_3, C)		
q_3					
q_4					

	C				
	a	b	c	d	ε
q_0	(q_1, C)		(q_2, ε)	(q_4, ε)	
q_1		(q_0, CCC)			
q_2			(q_2, ε)	(q_4, ε)	
q_3			(q_3, CC)	(q_4, ε)	
q_4				(q_4, ε)	

Tabella 1: Automa a pila che riconosce il linguaggio $\mathcal{L}$

L'automa che abbiamo definito riconosce il linguaggio $\mathcal{L}$ che dunque risulta essere un linguaggio libero da constesto.

Ovviamente molti altri automi a pila possono essere derivati per accettare lo stesso linguaggio.
Punto 2: Certamente l'automa, se corretto, accetterà le stringhe del tipo $(ab)^n c^n d^n$ dato che queste costituiscono un sottoinsieme delle parole del linguaggio corrispondente al caso in cui $m = n$. Ovviamente questo risultato non è in contrasto con il fatto che il linguaggio $\mathcal{L}' = \{(ab)^n c^n d^n | n \in \mathbb{N}\}$ non è un linguaggio context free. L'automa definito non accetta infatti **tutte e sole** le stringhe di questo linguaggio. D'altra parte il linguaggio definito come $\mathcal{L} = \{(ab)^l c^m d^n | l, m, n \in \mathbb{N}\}$ è un linguaggio regolare ma risulta che $\mathcal{L}' \subset \mathcal{L}$
Punto 3 : I passi dell'automa sulla stringa $(ab)^2 c^2 d^2$ sono mostrati nella tabella 2:

Stato	Pila automa	Input	Azione
q_0	Z_0	$ababccdd$	(q_1, ε)
q_1	Z_0	$babccdd$	(q_0, CC)
q_0	Z_0CC	$abccdd$	(q_1, C)
q_1	Z_0CC	$bccdd$	(q_0, CCC)
q_0	Z_0CCCC	$ccdd$	(q_2, ε)
q_2	Z_0CCC	cdd	(q_2, ε)
q_2	Z_0CC	dd	(q_4, ε)
q_4	Z_0C	d	(q_4, ε)
q_4	Z_0	ε	Accetta

Tabella 2: Passi dell'automa sulla stringa *ababccdd*

Esercizio 2 - (15 Punti)

Si consideri la seguente grammatica G:

$$S \longrightarrow aAbC \quad A \longrightarrow aB \quad B \longrightarrow AC \mid \varepsilon \quad C \longrightarrow bC \mid a \tag{1}$$

e si risolvano i seguenti punti, commentando adeguatamente i vari passi attuati:

1. si derivino gli insiemi FIRST, FOLLOW e *nullable* per G. Nella derivazione degli insiemi si annotino i vari simboli con l'indice dell'iterazione e il riferimento alla produzione che hanno richiesto l'aggiunta del simbolo all'insieme;
2. si discuta l'applicabilità del parsing LL(1);
3. si derivi l'automa LR(1) e la corrispondente tabella di parsing LR(1) discutendone la possibile applicabilità al riconoscimento del linguaggio generato dalla grammatica.

Svolgimento :

Punto 1: La seguente tabella mostra la composizione degli insiemi FIRST, FOLLOW e nullable a partire dal seguente ordine delle produzioni:

$$1.C \longrightarrow a, 2.C \longrightarrow bC, 3.B \longrightarrow \varepsilon, 4.B \longrightarrow AC, 5.A \longrightarrow aB, 6.S \longrightarrow aAbC \tag{2}$$

	nullable	FIRST	FOLLOW
S		$a_{1,6}$	
A		$a_{1,5}$	$a_{1,4}$ $b_{1,4}$
B	$y_{1,3}$	$a_{2,4}$	$a_{1,5}$ $b_{1,5}$
C		$a_{1,1}$ $b_{1,2}$	$a_{2,4}$ $b_{2,4}$

Punto 2: Per verificare se il parser LL(1) sia applicabile si può procedere alla derivazione della tabella di parsing:

	\$	a	b
S		$S \longrightarrow aAbC$	
A		$A \longrightarrow aB$	
B		$B \longrightarrow AC$, $B \longrightarrow \varepsilon$	$B \longrightarrow \varepsilon$
C		$C \longrightarrow a$	$C \longrightarrow bC$

Dunque il parsing LL(1) non è applicabile poichè la tabella evidenzia un conflitto per il simbolo non terminale B associato al simbolo terminale a.

Alla stessa conclusione si può giungere immediatamente osservando che per il simbolo non terminale B gli insiemi FIRST e FOLLOW hanno intersezione non vuota.

Punto 3: Per derivare l'automa si procede dapprima ad aumentare la grammatica con la produzione $S' \longrightarrow S\$$. Successivamente partendo dall'item $S' \longrightarrow .S\$$ si procede iterativamente fino a giungere all'automa LR(1) di figura 1. A partire da questo automa è possibile derivare la tabella di parsing LR(1) mostrata in tabella 3.

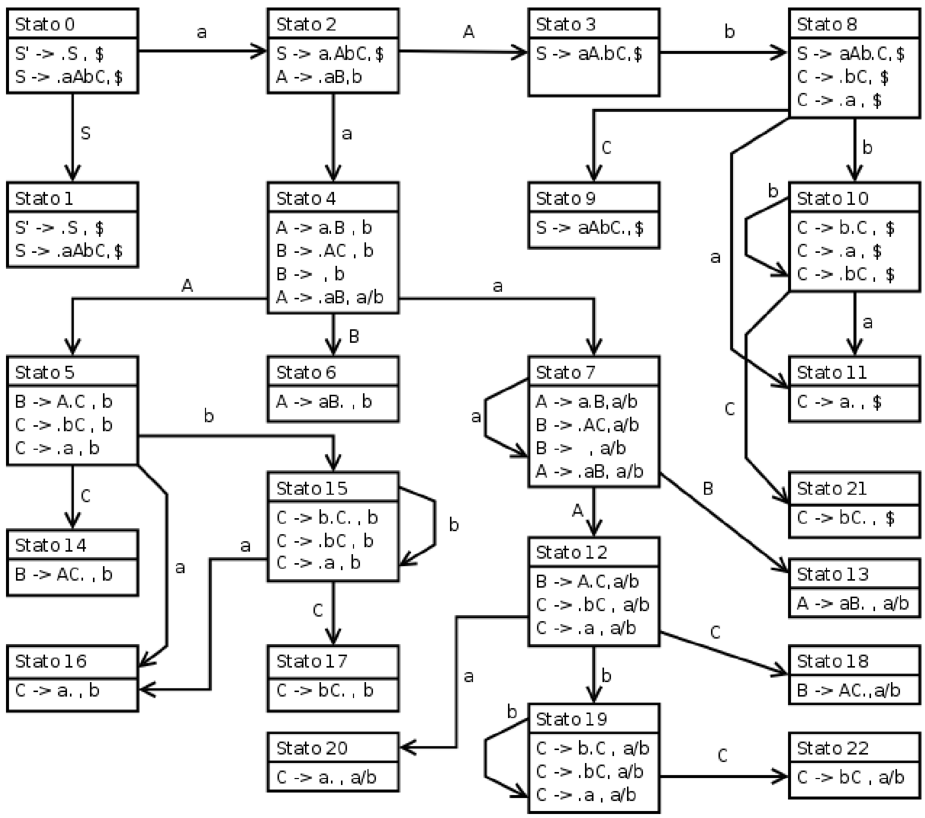

Figura 1: Automa LR(1) per la grammatica G

Osservando la tabella di parsing si può concludere che la grammantica non può essere utilizzata con un parser LR(1). La corrispondente tabella contiene infatti un conflitto di tipo shift/reduce in corrispondenza dello stato 7.

	a	b	\$	S	A	B	C
0	S2			G1			
1			accept				
2	S4				G3		
3		S8					
4	S7	$R(B \longrightarrow \varepsilon)$			G5	G6	
5	S16	S15					G14
6		$R(A \longrightarrow aB)$					
7	$R(B \longrightarrow \varepsilon)$/**S7**	$R(B \longrightarrow \varepsilon)$			G12	G13	
8	S11	S10					G9
9			$R(S \longrightarrow aAbC)$				
10		S11	S10				G21
11			$R(C \longrightarrow a)$				
12	S20	S19					G18
13	$R(A \longrightarrow aB)$	$R(A \longrightarrow aB)$					
14		$R(B \longrightarrow AC)$					
15	S16	S15					G17
16		$R(C \longrightarrow a)$					
17		$R(C \longrightarrow bC)$					
18	$R(B \longrightarrow AC)$	$R(B \longrightarrow AC)$					
19	S20	S19					G22
20	$R(C \longrightarrow a)$	$R(C \longrightarrow a)$					
21			$R(C \longrightarrow bC)$				
22	$R(C \longrightarrow bC)$	$R(C \longrightarrow bC)$					

Tabella 3: Tabella di parsing LR(0)

Esercizio 3 - (5 Punti)

Si fornisca una definizione di automa a pila deterministico e si dimostri che, contrariamente a quanto avviene per gli automi a stati finiti, non è vero che la classe dei linguaggi accettata dagli automi a pila deterministici coincide con quella accettata dagli automa a pila non deterministici.

Svolgimento :
Si consulti il libro di testo alla pagina corrispondente.