Linguaggi e Traduttori – Tempo: 2 ore

Prof. Marco Gavanelli

15 giugno 2017

Esercizio 1 (3 punti)

Si consideri il linguaggio $L = \{ab^n ca^n | n \geq 0\} \cup \{a^n b^n ca | n \geq 0\}$. Si scriva una grammatica non ambigua per il linguaggio L.

Lo studente cerchi di trovare una grammatica di livello più basso possibile nella classificazione secondo Chomsky, intendendo il livello 3 come minimo e il livello 0 come massimo.

Si mostri poi l'albero di derivazione della stringa *abca*.

Esercizio 2 (6 punti)

Si consideri la grammatica $G = \langle \{a, b, d\}, \{S, A, B\}, P, S\rangle$, dove:

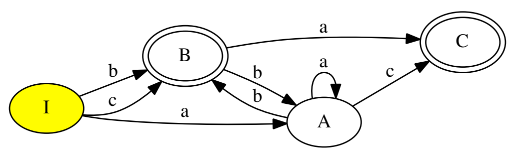
$$P = \begin{array}{l} S \to AbB \mid ASB \\ A \to aA \mid d \\ B \to dB \mid a \end{array}$$

1. Si classifichi la grammatica secondo Chomsky.
2. La grammatica è LL(1)? Se sì, si scriva la parsing table del PDA riconoscitore. Se no, si motivi il perché.
3. qualora la grammatica non sia LL(1), se ne scriva una equivalente G' che sia LL(1)
4. Si scriva la parsing table associata alla grammatica G'. Qualora non si sia ottenuta una grammatica LL(1), si scriva comunque la parsing table, evidenziando i conflitti
5. La grammatica è LR(0)? Se sì, si scriva l'automa a stati finiti ausiliario del PDA riconoscitore. Se no, si motivi il perché.

Qualora nei punti precedenti si sia riusciti a ottenere uno o due riconoscitori deterministici, si mostri come riconoscono la stringa *aadadbdada* mostrando l'evoluzione dello stack.

Esercizio 3 (3 punti)

Si consideri il seguente automa, dove lo stato I è lo stato iniziale e B e C sono stati finali:



Si scriva una grammatica regolare lineare a sinistra che genera il linguaggio riconosciuto dall'automa. La stringa *caa* appartiene al linguaggio generato dalla grammatica? Si mostri il funzionamento dell'automa su tale stringa.

Soluzione 1

Il linguaggio può essere generato dalla seguente grammatica:

$$P = \begin{array}{l} S \rightarrow aB \mid Cca \\ B \rightarrow bBa \mid c \\ C \rightarrow aCb \mid \epsilon \end{array}$$

che però è ambigua, come si può vedere dal fatto che la stringa *abca* ha due possibili derivazioni.

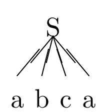
Immaginiamo quindi il linguaggio come costituito da 3 sotto-linguaggi:

$$L = \{ab^n ca^n | n \geq 2\} \cup \{a^n b^n ca | n \geq 2\} \cup \{abca, ac, ca\}$$

da cui la possibile grammatica:

$$P = \begin{array}{l} S \rightarrow aB \mid Cca \mid abca \mid ac \mid ca \\ B \rightarrow bBa \mid bbcaa \\ C \rightarrow aCb \mid aabb \end{array}$$

L'unico albero di derivazione di *abca* diventa:



Soluzione 2

1. La grammatica è di tipo 2 (context-free).

2. La grammatica non è LL(1), come si vede dal fatto che entrambe le alternative per il simbolo S iniziano con A.

3. Per ottenere una grammatica LL(1), raccogliamo a fattor comune le parti comuni delle produzioni per S:

$$\begin{array}{l} S \rightarrow AC \\ A \rightarrow aA \mid d \\ B \rightarrow dB \mid a \\ C \rightarrow bB \mid SB \end{array}$$

4. per scrivere la parsing table, calcoliamo l'insieme $START$ dei non terminali:

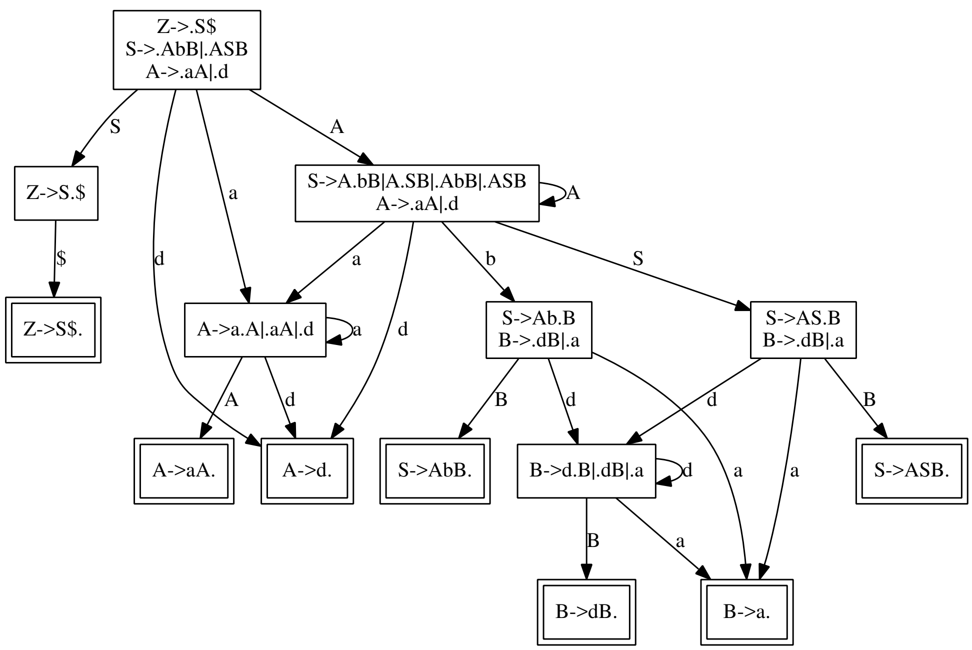
	$START$
S	$\{a, d\}$
A	$\{a, d\}$
B	$\{a, d\}$
C	$\{a, b, d\}$

La parsing table risulta:

	a	b	d	$\$$
S	$S \to AC$		$S \to AC$	
A	$A \to aA$		$A \to d$	
B	$B \to a$		$B \to dB$	
C	$C \to SB$	$C \to bB$	$C \to SB$	

Non presenta conflitti, quindi G' è LL(1).

5. La grammatica è LR(0), come si può vedere dal fatto che l'automa non presenta conflitti:



Riconoscimento :

LL(1)

Input	Stack
aadadbdada$	S$
aadadbdada$	AC$
aadadbdada$	aAC$
adadbdada$	AC$
adadbdada$	aAC$
dadbdada$	AC$
dadbdada$	dC$
adbdada$	C$
adbdada$	SB$
adbdada$	ACB$
adbdada$	aACB$
dbdada$	ACB$
dbdada$	dCB$
bdada$	CB$
bdada$	bBB$
dada$	BB$
dada$	dBB$
ada$	BB$
ada$	aB$
da$	B$
da$	dB$
a$	B$
a$	a$
$	$

LR(0)

Input	Stack
aadadbdada$	
adadbdada$	a
dadbdada$	aa
adbdada$	aad
adbdada$	aaA
adbdada$	aA
adbdada$	A
dbdada$	Aa
bdada$	Aad
bdada$	AaA
bdada$	AA
dada$	AAb
ada$	AAbd
da$	AAbda
da$	AAbdB
da$	AAbB
da$	AS
a$	ASd
$	ASda
$	ASdB
$	ASB
$	S
	S$
	Z

Soluzione 3

$$
\begin{aligned}
A &\to a \mid Aa \mid Bb \\
B &\to c \mid b \mid Ab \\
C &\to Ba \mid Ac
\end{aligned}
$$

Poiché ci sono più stati finali, lo scopo della grammatica è l'unione dei due scopi B e C.

$$S \to B \mid C$$

riscriviamo B e C in tale equazione per riportare la grammatica in forma regolare

$$
\begin{aligned}
S &\to c \mid b \mid Ab \mid Ba \mid Ac \\
A &\to a \mid Aa \mid Bb \\
B &\to c \mid b \mid Ab \\
C &\to Ba \mid Ac
\end{aligned}
$$

La stringa *caa* non appartiene al linguaggio. L'automa con tale input passa allo stato A, poi allo stato C, poi con l'ultimo simbolo a non ha una transizione, si passa quindi in stato di errore.