Linguaggi e Traduttori – Tempo: 2 ore

Prof. Marco Gavanelli

5 luglio 2017

Esercizio 1 (3 punti)

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

$$P = \begin{array}{r} S \to aAaB \mid bAb \\ aAa \to a \mid b \\ bAb \to bB \mid c \\ B \to a \mid bB \end{array}$$

Si classifichi la grammatica secondo Chomsky.

Si dica qual è il linguaggio $L(G)$ generato dalla grammatica.

Si scriva un'altra grammatica di livello più basso possibile nella classificazione secondo Chomsky (intendendo il livello 3 come minimo e il livello 0 come massimo) che genera lo stesso linguaggio.

Esercizio 2 (6 punti)

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

$$P = \begin{array}{l} S \to AdB \mid ABd \\ A \to bB \mid Af \\ B \to aB \mid Aa \mid c \end{array}$$

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. 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 il riconoscimento delle stringhe $bcfdbca$ e $bcbcfc$ mostrando l'evoluzione dello stack.

Esercizio 3 (3 punti)

Sia data la seguente grammatica: $G = \langle \{a, b, c\}, \{S, A, B\}, P, S \rangle$, dove

$$P = \begin{array}{l} S \to aA \mid aB \mid bA \\ A \to bB \mid aC \\ B \to a \mid bB \\ C \to b \mid aB \end{array}$$

Si disegni il corrispondente automa riconoscitore. L'automa è deterministico? Se non lo è, si produca un automa deterministico equivalente indicando chiaramente gli stati iniziali e finali.

Soluzione 1

La grammatica è di tipo 0, come si vede dalle produzioni $aAa \to a \mid b$ e $bAb \to bB \mid c$.

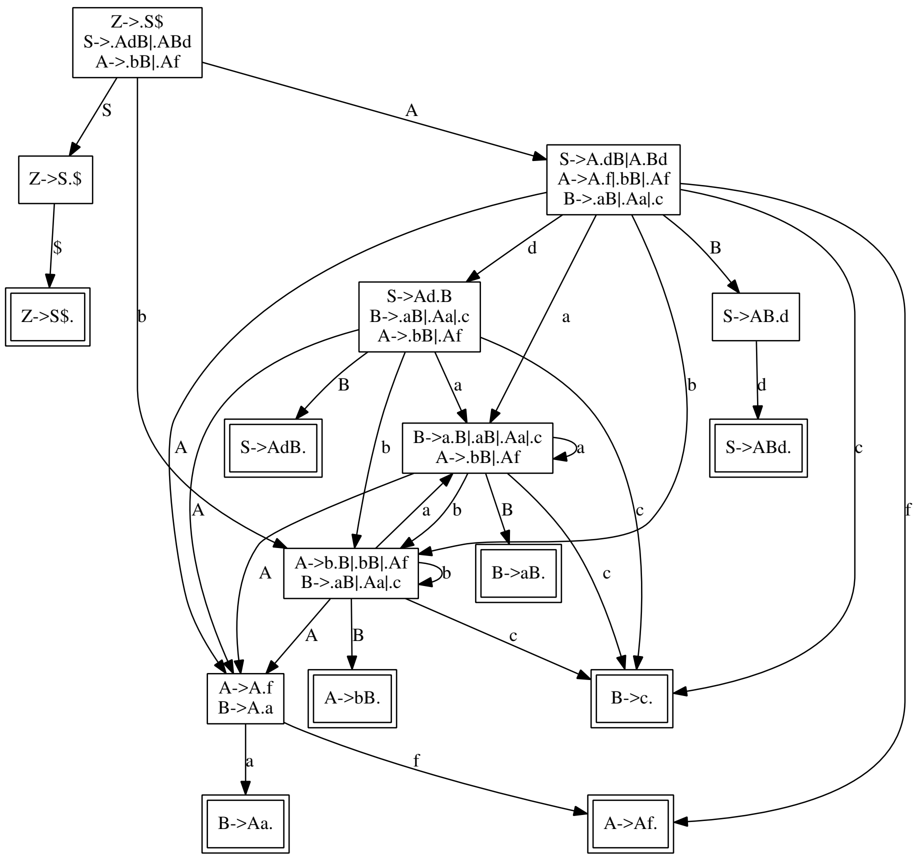
È però possibile sostituire le definizioni di aAa e bAb nella produzione per S, ottenendo:

$$P = \begin{array}{l} S \to aB \mid bB \mid c \\ B \to a \mid bB \end{array}$$

che è una grammatica regolare. Il linguaggio generato è $ab^*a + bb^*a + c$.

Soluzione 2

1. La grammatica è di tipo 2 (context-free).
2. La grammatica non è LL(1), come si vede dal fatto che c'è una ricorsione sinistra per A.
3. La grammatica è LR(0), come si può vedere dal fatto che l'automa non presenta conflitti:


Z->.S$
S->.AdB|.ABd
A->.bB|.Af
Z->S.$
Z->S$.
S->A.dB|A.Bd
A->A.f|.bB|.Af
B->.aB|.Aa|.c
S->Ad.B
B->.aB|.Aa|.c
A->.bB|.Af
S->AB.d
S->AdB.
B->a.B|.aB|.Aa|.c
A->.bB|.Af
S->ABd.
A->b.B|.bB|.Af
B->.aB|.Aa|.c
B->aB.
A->A.f
B->A.a
A->bB.
B->c.
B->Aa.
A->Af.

Riconoscimento:

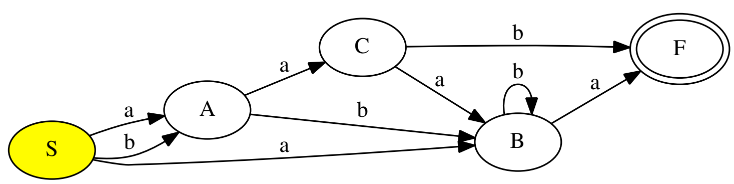
Input	Stack
bcfdbca$	
cfdbca$	b
fdbca$	bc
fdbca$	bB
fdbca$	A
dbca$	Af
dbca$	A
bca$	Ad
ca$	Adb
a$	Adbc
a$	AdbB
a$	AdA
$	AdAa
$	AdB
$	S
	S$
	Z

Stringa riconosciuta.

Input	Stack
bcbcfc$	
cbcfc$	b
bcfc$	bc
bcfc$	bB
bcfc$	A
cfc$	Ab
fc$	Abc
fc$	AbB
fc$	AA
c$	AAf
c$	AA

Stringa non riconosciuta.

Soluzione 3



L'automa non è deterministico, come si vede dal fatto che con input a dallo stato S si può passare sia in A sia in B.

Calcoliamo un automa deterministico che riconosce lo stesso linguaggio.

La tabella di transizione, aggiungendo uno stato di errore E e aggiungendo nuovi stati man mano che si presentano:

	a	b
S	AB	A
A	C	B
B	F	B
C	B	F
F	E	E
AB	CF	B
CF	BE	FE
BE	FE	BE
FE	E	E

L'automa risultante: