Linguaggi e Traduttori

Prof. Marco Gavanelli

21 settembre 2016

Esercizio 1 (4 punti)

Si consideri il linguaggio $L = \{ab^n | n \geq 0\} \cup \{a^n b^n | n \geq 0\}$. 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 ab.

Esercizio 2 (6 punti)

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

$$P = \begin{array}{l} A \to AAa \mid Ba \\ B \to bAd \mid f \end{array}$$

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.
4. Si mostri come l'automa riconosce la stringa $fabfadaa$ mostrando l'evoluzione dello stack

Esercizio 3 (4 punti)

Si consideri la seguente grammatica:

$$\begin{array}{l} S \to Aa \mid Ba \mid a \\ A \to Ab \mid Bb \\ B \to Sa \end{array}$$

Si disegni l'automa riconoscitore del linguaggio generato. L'automa è deterministico? Si scriva l'espressione regolare che rappresenta il linguaggio generato dallo scopo S.

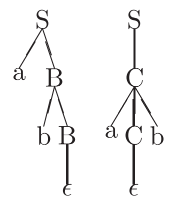
Soluzione 1

$$L = \{ab^n | n \geq 0\} \cup \{a^n b^n | n \geq 0\}$$

Il linguaggio può essere generato da una grammatica di tipo 2 (indipendente dal contesto); ad esempio:

$$S \to aB \mid C$$
$$B \to \epsilon \mid bB$$
$$C \to \epsilon \mid aCb$$

tale grammatica, però, è ambigua, come si vede dal fatto che esistono due alberi di derivazione distinti per la stringa ab:



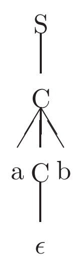
Una possibile grammatica non ambigua è la seguente:

$$S \to a \mid aB \mid C$$
$$B \to bb \mid bB$$
$$C \to \epsilon \mid aCb$$

che corrisponde ad immaginare il linguaggio costituito da 3 sottoinsiemi disgiunti:

$$L = \{a\} \cup \{ab^n | n \geq 2\} \cup \{a^n b^n | n \geq 0\}$$

In questa grammatica, la stringa ab ha un'unica rappresentazione:

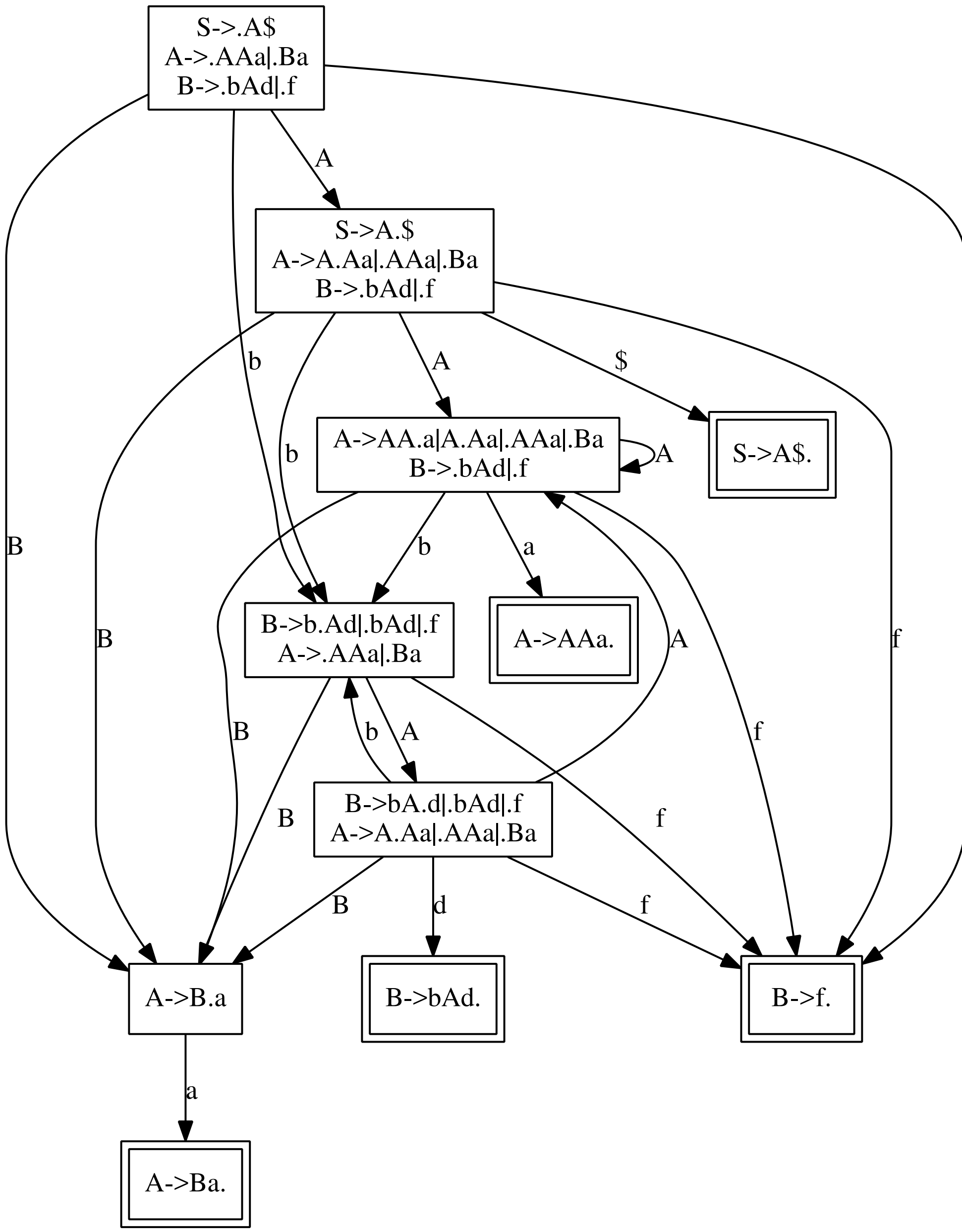


Soluzione 2

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 il simbolo A.

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

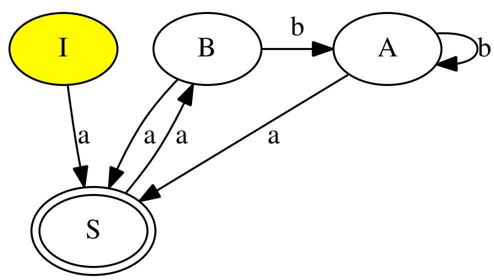

S->.A$
A->.AAa|.Ba
B->.bAd|.f
A
S->A.$
A->A.Aa|.AAa|.Ba
B->.bAd|.f
b
A
$
A->AA.a|A.Aa|.AAa|.Ba
B->.bAd|.f
A
S->A$.
b
B
b
a
B
f
f
B->b.Ad|.bAd|.f
A->.AAa|.Ba
A->AAa.
A
B
b
A
f
B
B->bA.d|.bAd|.f
A->A.Aa|.AAa|.Ba
f
B
d
f
A->B.a
B->bAd.
B->f.
a
A->Ba.

Riconoscimento stringa:

Input	Stack
fabfadaa$	
abfadaa$	f
abfadaa$	B
bfadaa$	Ba
bfadaa$	A
fadaa$	Ab
adaa$	Abf
adaa$	AbB
daa$	AbBa
daa$	AbA
aa$	AbAd
aa$	AB
a$	ABa
a$	AA
$	AAa
$	A
	A$
	S

Soluzione 3

L'automa riconoscitore:



Questo automa è deterministico.

Per calcolare il linguaggio generato tramite espressioni regolari, riscriviamo la grammatica come sistema di equazioni:

$$\begin{cases} S = Aa + Ba + a \\ A = Ab + Bb \\ B = Sa \end{cases}$$

Sostituiamo la B nelle altre equazioni

$$\begin{cases} S = Aa + Saa + a \\ A = Ab + Sab \end{cases}$$

Eliminiamo la ricorsione diretta per A

$$\begin{cases} S = Aa + Saa + a \\ A = Sabb^* \end{cases}$$

Sostituiamo la A nell'equazione per S

$$S = Sabb^*a + Saa + a$$

Raccogliamo la S a destra del simbolo di uguaglianza:

$$S = S(abb^*a + aa) + a$$

A questo punto si può eliminare la ricorsione diretta per S:

$$S = a(abb^*a + aa)^*$$

Ovviamente, altre espressioni regolari possono descrivere lo stesso linguaggio. Ad esempio, si può notare che $abb^*a + aa$ è equivalente a ab^*a, per cui un'espressione equivalente è la sequente:

$$S = a(ab^*a)^*$$