

Esercizio 2

Si consideri la grammatica seguente:

$$\begin{aligned} S &::= A \\ S &::= Bbb \\ A &::= aB \\ B &::= aAb \\ B &::= \varepsilon \end{aligned}$$

Si risponda alle domande seguenti (giustificando formalmente la risposta):

- quale linguaggio genera la grammatica (lo si esprima mediante espressioni su insiemi) ?
- la grammatica è LR(1) ?
- la grammatica è LL(k) per qualche k ?
- esiste una grammatica LL(1) per il linguaggio ?

Se la risposta in d) e' affermativa si dia una tabella di analisi LL(1).

Esercizio2a

Eliminazione mutua ricorsione da A, B

$$\begin{aligned} S &::= A \mid Bbb \\ A &::= aB \\ B &::= aaBb \mid \varepsilon \end{aligned}$$

Sottolinguaggi:

$$\begin{aligned} L(B) &= \{a^{2n} b^n \mid n \geq 0\} \\ L(A) &= \{aa^{2n} b^n \mid n \geq 0\} \\ L(S) &= \{aa^{2n} b^n \mid n \geq 0\} \cup \{a^{2n} b^n bb \mid n \geq 0\} \\ &= \{a^{2n+1} b^n, a^{2n} b^{n+2} \mid n \geq 0\} \end{aligned}$$

Esercizio2b

Esaminiamo la grammatica.

Considerazione1: Se ci e' chiaro il processo di riduzione effettuato da un analizzatore LR, allora osserviamo che la grammtica consente un riconoscimento di tale tipo. Infatti, le due produzioni per S ci dicono che il riconoscimento avverra' sempre cercando, inizialmente, una maniglia ad A o a B. Le produzioni per A e per B ci dicono che il riconoscitore trovando il simbolo *a* come lookahead operera' sempre uno shift, rimandando ogni riduzione alla lettura di un simbolo *b*. Quando cio' avverra', la maniglia e' $B::=\varepsilon$. Dopo, se nello stack sono contenute *a*

applichera', in corrispondenza di b nel lookhaead, prima la maniglia $A ::= aB$ e dopo la maniglia $B ::= aAb$. Se e quando nello stack le a accumulate sono state ridotte, allora o abbiamo $\$$ nel lookaead (maniglia $S ::= A$, applicabile), o abbiamo b . Quest'ultimo caso condurra' dopo due shift alla maniglia $S ::= Bbb$.

Considerazione2: Le osservazioni precedenti, nella forma in cui sono state espote, sebbene indicative, non formano una dimostrazione. Una dimostrazione puo' facilmente essere ottenuta costruendo la collezione canonica LR(0) e mostrando che essa non contiene *stati con conflitti*.

$S' ::= S$
 $S ::= A \mid B b b$
 $A ::= a B$
 $B ::= a A b \mid \varepsilon$

$I_0 = \text{Clos}(S' \rightarrow \cdot S)$	$\{S' \rightarrow \cdot S, S \rightarrow \cdot A, S \rightarrow \cdot Bbb, A \rightarrow \cdot aB, B \rightarrow \cdot aAb, B \rightarrow \cdot\}$
$I_1 = G(0, S)$	$\{S' \rightarrow S \cdot\}$
$I_2 = G(0, A)$	$\{S \rightarrow A \cdot\}$
$I_3 = G(0, B)$	$\{S \rightarrow B \cdot bb\}$
$I_4 = G(0, a) = G(4, a)$	$\{A \rightarrow a \cdot B, B \rightarrow a \cdot Ab, A \rightarrow a \cdot aB, B \rightarrow a \cdot aAb, B \rightarrow \cdot\}$
$I_5 = G(3, b)$	$\{S \rightarrow Bb \cdot b\}$
$I_6 = G(4, B)$	$\{A \rightarrow aB \cdot\}$
$I_7 = G(4, A)$	$\{B \rightarrow aA \cdot b\}$
$I_8 = G(5, b)$	$\{S \rightarrow Bbb \cdot\}$
$I_9 = G(7, b)$	$\{B \rightarrow aAb \cdot\}$

La dimostrazione deve essere completata mostrando che gli stati sopra non contengono conflitti. Occorre quindi calcolare la tabella FOLLOW ed esaminare gli stati con possibili conflitti, ovvero: I_0, I_4 .

<u>Follow</u>	S	$\{\$, \}$
	A	$\{b, \$\}$
	B	$\{b, \$\}$

Esercizio2c

La grammatica non e' LL(K) per alcun K.

Dimostrazione.

Sia K un arbitrario naturale non 0.

Osserviamo

S e' una sentenziale sinistra della grammatica

$S ::= A$ ed $S ::= Bbb$ sono entrambe produzioni applicabili (in una d. sin.) a tale sentenziale:

$S \Rightarrow A$

$S \Rightarrow Bbb$

Ma vediamo che:

a^k incluso in $\text{First}_k(A)$, infatti $A \Rightarrow^* a^k B$

a^k incluso in $\text{First}_k(Bbb)$, infatti $Bbb \Rightarrow^* a^k a^k Bb^k bb$

Cio' conclude la dimostrazione.

Esercizio2d

Possiamo operare in vari modi alternativi tra loro: 1) analizziamo il linguaggio e definiamo una grammatica LL che tenga conto delle [osservazioni fatte](#), 2) cerchiamo una soluzione con trasformazioni conservative del linguaggio generato.

Soluzione 1).

Il linguaggio e':

$$L = \{a^{2n+1} b^n, a^{2n} b^n bb \mid n \geq 0\}$$

Le frasi hanno suffisso diverso secondo che il numero di a nel prefisso sia dispari o pari. Allora possiamo vedere tali frasi come costuite dal solo bb oppure iniziati con a e seguite da frasi di un linguaggio $L' = \{a a^{2n+1} b^n b, a a^{2n} b^n bb b, \epsilon \mid n \geq 0\}$. Ma vediamo che vi e' stretta relazione tra le frasi di L ed L', ottenendo la grammatica seguente

$S ::= a S' \mid bb$

$S' ::= a S b \mid \epsilon$

E' facile vedere che la grammatica e' LL(1). E' altrettanto facile vedere che essa genera lo stesso linguaggio, peraltro il metodo induttivo con cui e' stata derivata fornisce anche una dimostrazione.

Soluzione 2) Richiede maggiore accortezza e si basa essenzialmente sulla relazione che abbiamo tra produzioni ed equazioni tra linguaggi. Mostriamo come puo' operare un metodo basato su una trasformazione conservativa.

Partiremo dalla grammatica ottenuta rimuovendo la mutua ricorsione.

$$\begin{aligned} S &::= A \mid Bbb \\ A &::= aB \\ B &::= aaBb \mid \varepsilon \end{aligned}$$

Rimpiazziamo A ed eliminiamo i non terminali diventati inutili:

$$\begin{aligned} S &::= aB \mid Bbb \\ B &::= aaBb \mid \varepsilon \end{aligned}$$

Fattorizziamo e operiamo l'introduzione di un nuovo linguaggio con S':

$$\begin{aligned} S &::= a(B \mid aBb \mid bb) \mid bb \\ B &::= aaBb \mid \varepsilon \end{aligned}$$
$$\begin{aligned} S &::= a S' \mid bb \\ S' &::= B \mid aBb \mid bb \\ B &::= aaBb \mid \varepsilon \end{aligned}$$

Rimpiazziamo B in S', e fattorizziamo:

$$\begin{aligned} S &::= a S' \mid bb \\ S' &::= aaBb \mid aBb \mid bb \mid \varepsilon \\ B &::= aaBb \mid \varepsilon \end{aligned}$$
$$\begin{aligned} S &::= a S' \mid bb \\ S' &::= a(aB \mid Bb \mid b) \mid \varepsilon \\ B &::= aaBb \mid \varepsilon \end{aligned}$$

Rimpiazziamo in S' la sottoespressione $aB \mid Bb \mid b$ con il non terminale S a cui essa corrisponde vedi produzione $S::=aB \mid Bb \mid b$ occorrente nella seconda delle grammatiche equivalenti prodotte dalle nostre trasformazioni (che hanno sempre mantenuto il significato dei vari simboli)

$$\begin{aligned} S &::= a S' \mid bb \\ S' &::= a S \mid \varepsilon \\ B &::= aaBb \mid \varepsilon \end{aligned}$$

Eliminiamo i terminali diventati inutili:

$$S ::= a S' \mid bb$$
$$S' ::= a S b \mid \varepsilon$$

Abbiamo ottenuto la stessa grammatica LL(1).