Corso di Linguaggi di Programmazione — Parziale A-L di fine modulo
Prova scritta del 20 Dicembre 2023

Tempo a disposizione: 2 ore e 30 minuti.

1. Descrivere le regole SOS per l'espressione booleana b_0 or b_1, secondo la disciplina di valutazione esterna-parallela (EP). Mostrare un esempio di una espressione di quel tipo tale che la valutazione EP e quella ES (esterna-sinistra, vista a lezione) non forniscono sempre lo stesso risultato.

2. Fornire una definizione regolare per *password*, che deve essere una qualunque sequenza di lettere e/o cifre, che deve iniziare con una cifra, terminare con una cifra o una lettera minuscola, e che deve contenere almeno una lettera maiuscola.

3. Classificare $L = \{a^n b^m c^k \mid n \geq m \geq k \geq 0\}$, ovvero dire se L è regolare, oppure libero ma non regolare, oppure non libero, giustificando adeguatamente la risposta.

4. È vero che i linguaggi di classe LR(0) sono chiusi per unione? Motivare la risposta.

5. Si costruisca il DFA minimo che riconosca il linguaggio denotato dall'espressione regolare ab^*a^*b.

6. Si ricavi dal DFA minimo ottenuto nell'esercizio precedente la grammatica regolare associata; inoltre, si semplifichi la grammatica rimuovendo i simboli inutili; infine, si ricavi da quella grammatica semplificata l'espressione regolare associata.

7. Mostrare che $L_1 = \{a^n b^m c^n \mid 0 \leq m, 0 \leq n\}$ è libero deterministico, costruendo un opportuno DPDA. Considerando $L_2 = \{a^{2n} b^{2m} c^{2k} \mid n, m, k \geq 0\}$, è vero che $L_1 \cap L_2$ è un linguaggio libero? Motivare la risposta.

8. Si consideri la seguente grammatica G con simbolo iniziale S:

$$\begin{array}{lll@{\qquad}lll} S & \to & CBA \mid BAE & A & \to & C \mid \mathtt{a}E\mathtt{c} \mid \mathtt{a}S\mathtt{b} \\ B & \to & \mathtt{c} \mid A\mathtt{c}B & C & \to & \epsilon \mid C\mathtt{d} \\ D & \to & \mathtt{d}D & E & \to & D \mid \mathtt{d}E \end{array}$$

(*i*) Si calcolino i First e i Follow per tutti i nonterminali. (*ii*) Si rimuovano i simboli inutili, ottenendo una grammatica semplificata G'. (*iii*) Si rimuova la produzione epsilon per ottenere una grammatica G'' senza produzioni epsilon, che sia equivalente a G'. (*iv*) Si rimuovano le produzioni unitarie da G'' per ottenere una grammatica G''' senza produzioni unitarie equivalente a G''. (*v*) Si rimuova la ricorsione sinistra immediata per C per ottenere una $\overline{G}$ equivalente a G'''.

9. Data la seguente grammatica G con simbolo iniziale S:

$$\begin{array}{lll} S & \to & \epsilon \mid AB \\ A & \to & \epsilon \mid \mathtt{a}A\mathtt{b} \mid \mathtt{ab} \\ B & \to & \epsilon \mid \mathtt{b} \mid \mathtt{b}B \end{array}$$

(*i*) Si verifichi che G è ambigua e la si disambigui, ottenendo una grammatica non ambigua G' equivalente. (*ii*) Verificare se G' è di classe LL(1); se non lo fosse, la si manipoli per ottenere una grammatica equivalente G'' di classe LL(1). (*iii*) Si costruisca quindi la tabella di parsing LL(1). (*iv*) Si nostri il funzionamento del parser LL(1) sull'input *abb*.

10. Data la seguente grammatica G con simbolo iniziale S:

$$\begin{array}{lll} S & \to & \mathtt{a} \mid AS\mathtt{c} \\ A & \to & \mathtt{a} \end{array}$$

si determini $L(G)$. Inoltre, si costruisca l'automa canonico LR(1).

1)

$$\frac{\langle b_0,\sigma\rangle \to \langle b_0',\sigma'\rangle}{\langle b_0 \text{ or } b_1,\sigma\rangle \to \langle b_0' \text{ or } b_1,\sigma'\rangle}$$

Esterna Parallela (EP)

$$\frac{\langle b_1,\sigma\rangle \to \langle b_1',\sigma'\rangle}{\langle b_0 \text{ or } b_1,\sigma\rangle \to \langle b_0 \text{ or } b_1',\sigma'\rangle}$$

$$\frac{}{\langle tt \text{ or } b_1,\sigma\rangle \to \langle tt,\sigma\rangle}$$

$$\frac{}{\langle ff \text{ or } b_1,\sigma\rangle \to \langle b_1,\sigma\rangle}$$

$$\frac{}{\langle b_0 \text{ or } tt,\sigma\rangle \to \langle tt,\sigma\rangle}$$

$$\frac{}{\langle b_0 \text{ or } ff,\sigma\rangle \to \langle b_0,\sigma\rangle}$$

- $\to_{EP} \neq \to_{ES}$ Ad esempio,

$$\langle (2-5)=8 \text{ or } tt\rangle \not\to_{ES}$$

$$\to_{EP} \langle tt,\sigma\rangle$$

2) password := cifra tutto* maius tutto* (cifra | maius)

tutto := lettera | cifra

cifra := [0-9]

lettera := minus | maius

minus := [a-z]

maius := [A-Z]

3) $L = \{ a^n b^m c^k \mid n \geq m \geq k \geq 0 \}$ non è libero!

Non è possibile costruire un PDA perché debbo "contare" 2 volte. Prima impilo le "a", poi le confronto con le "b" (e spilo le "a" corrispondenti) ma poi non riesco più a confrontare le "b" con le "c": mi servirebbe un secondo stack su cui avrei dovuto impilare le "b"!!

Non è possibile usare il pumping lemma a rovescio! Infatti

- Fissiamo $N > 0$ generico
- Scegliamo $z = a^N b^N c^N$
- Per ogni $uvwxy$ tali che $z = uvwxy$
  - $|vwx| \leq N$
  - $|vx| \geq 1$

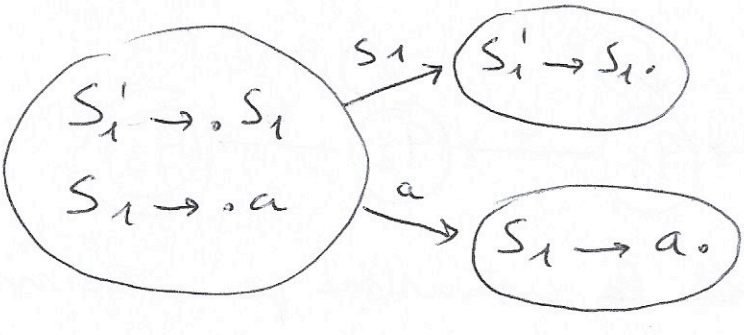
possiamo solo concludere che vwx deve stare

1) o solo nel blocco a^N
2) o a cavallo del blocco $a^N b^N$
3) o solo nel blocco b^N
4) o a cavallo del blocco $b^N c^N$
5) o solo nel blocco c^N

Ma nel caso 1) non arrivo alla contraddizione, perché se aumento solo le "a", ottengo una stringa che sta ancora in L!!

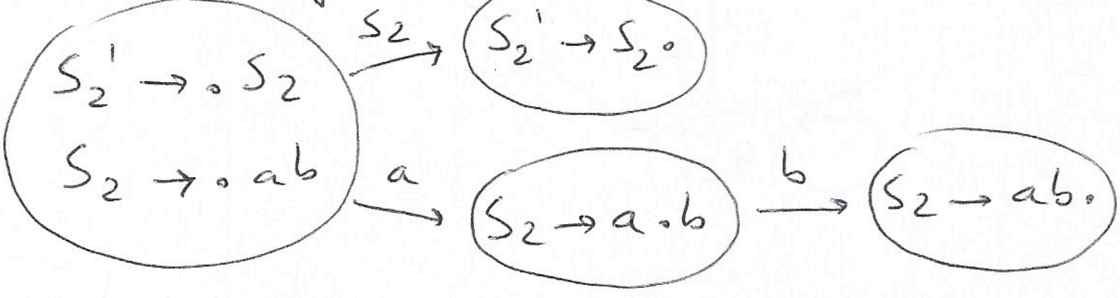
4) I linguaggi LR(0) non sono chiusi per unione.

Infatti, $L_1 = \{a\}$ è LR(0): basta prendere la grammatica $G_1 = [S_1 \to a$



automa canonico LR(0) (Nessun conflitto)

Similmente, $L_2 = \{ab\}$ è LR(0): basta prendere la grammatica $G_2 = S_2 \to ab$



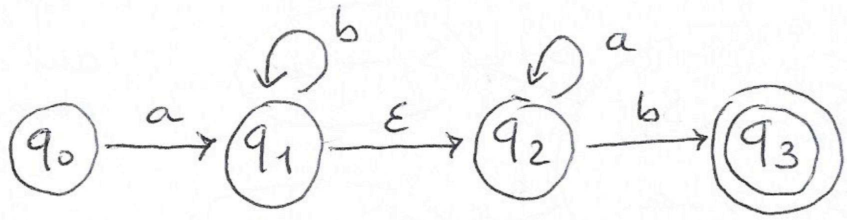
automa canonico LR(0)

Nessun conflitto.

Ma il linguaggio $L = L_1 \cup L_2 = \{a, ab\}$ non è LR(0) perché un linguaggio finito che non gode della prefix property non può essere di classe LR(0).

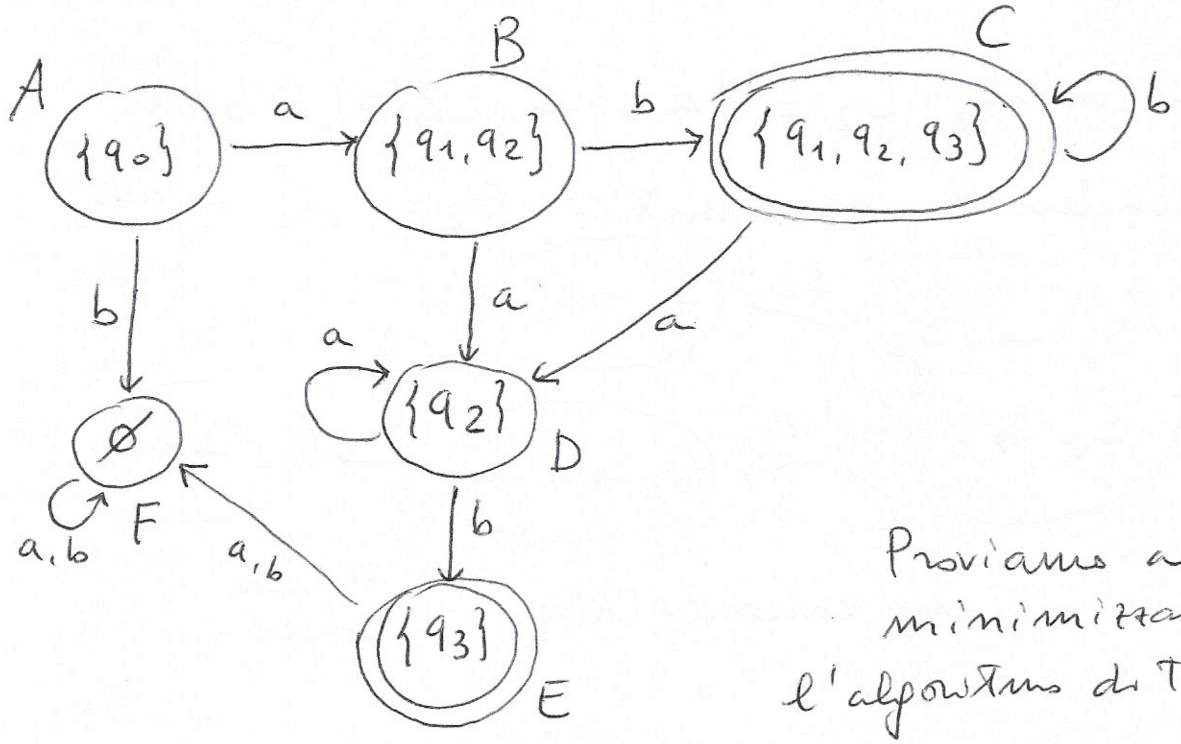
5) ab^*a^*b

Un semplice NFA che riconosce il linguaggio regolare denotato da ab^*a^*b potrebbe essere



NFA

Applicando la costruzione per sottoinsiemi, otteniamo



DFA

Proviamo a minimizzarlo, usando l'algoritmo di Tabelle a scala

	A	B	C	D	E
B	x_1				
C	x_0	x_0			
D	x_1	x_2	x_0		
E	x_0	x_0	x_1	x_0	
F	x_2	x_1	x_0	x_1	x_0

Tutte le entrate vengono marcate (entro il round 2) e quindi il DFA è già minimo!

6)

$$\left.\begin{aligned}
A &\to aB \mid bF \\
B &\to aD \mid bC \\
C &\to \varepsilon \mid bC \mid aD \\
D &\to aD \mid bE \\
E &\to \varepsilon \mid aF \mid bF \\
F &\to aF \mid bF
\end{aligned}\right]$$

- G associata al DFA dell'esercizio 5
- Ci sono simboli inutili: F non è un generatore

$\Downarrow$

G' semplificata

$$\left[\begin{aligned}
A &\to aB \\
B &\to aD \mid bC \\
C &\to \varepsilon \mid aD \mid bC \\
D &\to aD \mid bE \\
E &\to \varepsilon
\end{aligned}\right.$$

$D \to aD \mid b$

$D \rightsquigarrow a^*b$

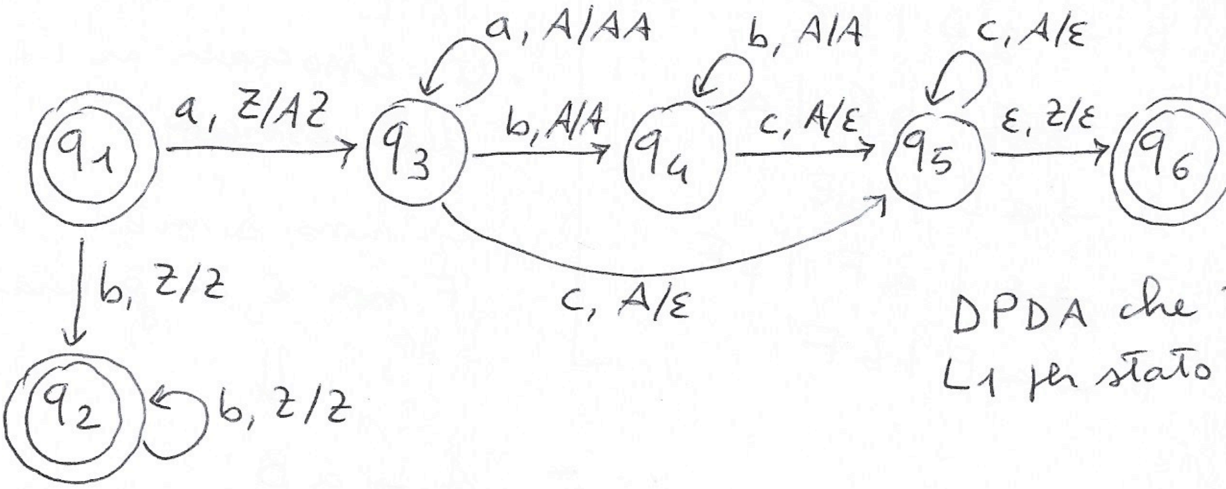
$C \to \varepsilon \mid aa^*b \mid bC$

$\rightsquigarrow b^*(\varepsilon \mid aa^*b)$

$B \to aa^*b \mid bb^*(\varepsilon \mid aa^*b)$

$A \to a(aa^*b \mid bb^*(\varepsilon \mid aa^*b))$

7) $L_1 = \{ a^n b^m c^n \mid n, m \geq 0 \}$ è libero deterministico



DPDA che riconosce L_1 per stato finale

$L_2 = \{ a^{2n} b^{2m} c^{2k} \mid n, m, k \geq 0 \}$ è regolare

perché è denotato dall'espressione regolare

$$(aa)^*(bb)^*(cc)^*$$

A lezione abbiamo dimostrato che l'intersezione di un linguaggio libero con uno regolare è un ling. libero

$\Downarrow$

$$\underset{\text{libero}}{L_1} \cap \underset{\text{regolare}}{L_2} \text{ è libero!}$$

$$L_1 \cap L_2 = \{ a^{2n} b^{2m} c^{2n} \mid n, m \geq 0 \}$$

8) G
$$S \to CBA \mid BAE \qquad A \to C \mid aEc \mid aSb$$
$$B \to c \mid AcB \qquad C \to \varepsilon \mid Cd$$
$$D \to dD \qquad E \to D \mid dE$$

	First	Follow
S	c, a, d	\$, b
A	ε, a, d	\$, b, d, c
B	c, a, d	a, d, \$, b
C	ε, d	d, c, a, \$, b
D	d	\$, b, c
E	d	\$, b, c

- D non è generatore
- E non è generatore

⇓

$$S \to CBA$$
$$A \to C \mid aSb$$
$$B \to c \mid AcB$$
$$C \to \varepsilon \mid Cd$$

G' semplificata

$N(G) = \{C, A\}$

$$S \to CBA \mid BA \mid CB \mid B$$
$$A \to C \mid aSb$$
$$B \to c \mid AcB \mid cB$$
$$C \to Cd \mid d$$

G'' senza ε-prod.

$U(G) = \{(S,S), (A,A), (B,B), (C,C), (A,C), (S,B)\}$

$$S \to CBA \mid BA \mid CB \mid \underbrace{c \mid AcB \mid cB}$$
$$A \to \underbrace{Cd \mid d} \mid aSb$$
$$B \to c \mid AcB \mid cB$$
$$C \to Cd \mid d$$

G''' senza produzioni unitarie

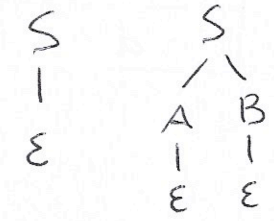
Ricorsione Sx: $C \to Cd \mid d$

$$
\begin{aligned}
S &\to CBA \mid BA \mid CB \mid c \mid AcB \mid cB \\
A &\to Cd \mid d \mid aSb \\
B &\to c \mid AcB \mid cB \\
C &\to dC' \\
C' &\to dC' \mid \varepsilon
\end{aligned}
$$

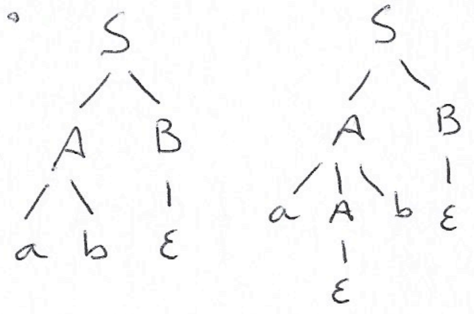
---

9) $$\left.\begin{aligned}
S &\to \varepsilon \mid AB \\
A &\to \varepsilon \mid aAb \mid ab \\
B &\to \varepsilon \mid b \mid bB
\end{aligned}\right]$$ G è ambigua. Ad es.,

- $\varepsilon \in L(G)$



- $ab \in L(G)$



Osserviamo che

$A \to \varepsilon \mid aAb$ è non ambigua (abbiamo rimosso $A \to ab$)

$B \to \varepsilon \mid bB$ è non ambigua (abbiamo rimosso $B \to b$)

$S \to AB$ è non ambigua (abbiamo rimosso $A \to \varepsilon$)

Quindi $$\left.\begin{aligned}
S &\to AB \\
A &\to \varepsilon \mid aAb \\
B &\to \varepsilon \mid bB
\end{aligned}\right]$$ G' è non ambigua!

G' è LL(1)! Infatti

1) - $First(\varepsilon) \cap First(aAb) = \emptyset$
$\{\varepsilon\} \cap \{a\}$

- $Follow(A) \cap First(aAb) = \emptyset$
$\{b, \$\} \cap \{a\} = \emptyset$

2) - $First(\varepsilon) \cap First(bB) = \emptyset$
$\{\varepsilon\} \cap \{b\}$

- $Follow(B) \cap First(bB) = \emptyset$
$\{\$\} \cap \{b\} = \emptyset$

	a	b	\$
S	$S \to AB$	$S \to AB$	$S \to AB$
A	$A \to aAb$	$A \to \varepsilon$	$A \to \varepsilon$
B		$B \to bB$	$B \to \varepsilon$

$First(AB) = \{a, b, \varepsilon\}$
$Follow(S) = \{\$\}$
$Follow(A) = \{b, \$\}$
$Follow(B) = \{\$\}$

Tabella di parsing LL(1)
(senza conflitti)

$\underline{a}bb\$$	$S\$$
	$AB\$$
	$\underline{a}AbB\$$
	$AbB\$$
$\underline{b}b\$$	$\underline{b}B\$$
	$B\$$
$\underline{b}\$$	$\underline{b}B\$$
	$B\$$
$\$$	$\$$

ok, la stringa è accettata

Soluzione alternativa:

$$\left.\begin{array}{l} S \to \varepsilon \mid AB \mid A \mid B \\ A \to ab \mid aAb \\ B \to b \mid bB \end{array}\right]$$

G'' non ambigua, ottenuta utilizzando l'algoritmo per eliminare le prod. epsilon $A \to \varepsilon$ e $B \to \varepsilon$

Per ottenerne una equivalente LL(1), è possibile fattorizzare:

$$\left.\begin{array}{l} S \to \varepsilon \mid AS' \mid B \\ S' \to B \mid \varepsilon \\ A \to aA' \\ A' \to b \mid Ab \\ B \to bB' \\ B' \to \varepsilon \mid B \end{array}\right]$$

G''' è equivalente a G'' e si può verificare che è di classe LL(1)

1)
$$\left.\begin{array}{l} - \text{First}(\varepsilon) \cap \text{First}(AS') = \emptyset \\ - \text{First}(\varepsilon) \cap \text{First}(B) = \emptyset \\ - \text{Follow}(S) \cap \text{First}(AS') = \emptyset \\ - \text{Follow}(S) \cap \text{First}(B) = \emptyset \end{array}\right\} \text{per } S$$

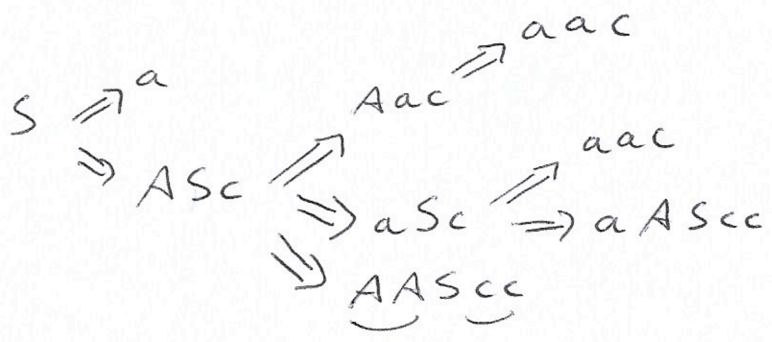
2)
$$\left.\begin{array}{l} - \text{First}(B) \cap \text{First}(\varepsilon) = \emptyset \\ - \text{First}(B) \cap \text{Follow}(S') = \emptyset \end{array}\right\} \text{per } S'$$

3)
$$- \text{First}(b) \cap \text{First}(Ab) = \emptyset \;\} \text{ per } A'$$

4)
$$\left.\begin{array}{l} - \text{First}(\varepsilon) \cap \text{First}(B) = \emptyset \\ - \text{Follow}(B') \cap \text{First}(B) = \emptyset \end{array}\right\} \text{per } B'$$

10)

$$\left.\begin{array}{l} S \to a \mid ASc \\ A \to a \end{array}\right] G$$



$A^n S c^n \quad n \geq 0$

a

$$L(G) = \{ a^{n+1} c^n \mid n \geq 0 \}$$