Corso di Linguaggi di Programmazione — Parziale M-Z di fine modulo
Prova scritta del 20 Dicembre 2022

Tempo a disposizione: 2 ore e 30 minuti.

1. Descrivere le regole di semantica operazionale strutturata per l'espressione booleana b_0 and b_1, secondo la disciplina di valutazione esterna-parallela (EP). Argomentare che la valutazione EP e quella IS (interna-sinistra) 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 lettera maiuscola o una cifra, terminare con una cifra, e che deve contenere almeno una lettera minuscola.

3. Classificare il linguaggio $L = \{a^{2n+1}b^{2m+1}c^{2k+1} \mid m \geq n \geq 0, k \geq 0\}$, ovvero dire se L è regolare, oppure libero ma non regolare, oppure non libero, giustificando adeguatamente la risposta.

4. È vero che possono esistere linguaggi liberi nondeterministici non ambigui? Motivare la risposta.

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

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 l'espressione regolare associata.

7. Mostrare che $L_1 = \{a^n b^m c^n \mid n \geq 0, m \geq 0\}$ è libero deterministico, costruendo un opportuno DPDA. Sapendo che anche $L_2 = \{a^n b^n c^m \mid 0 \leq n, 0 \leq m\}$ è libero deterministico, 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:

$$\begin{array}{lcl@{\qquad}lcl} S & \rightarrow & ABC\mathtt{c} \mid E\mathtt{a} & A & \rightarrow & \epsilon \mid \mathtt{a}BE \mid BS\mathtt{a} \\ B & \rightarrow & \mathtt{d} \mid \mathtt{b}C\mathtt{d} & C & \rightarrow & A \mid C\mathtt{d} \\ D & \rightarrow & \mathtt{c} \mid \mathtt{d}S\mathtt{a} & E & \rightarrow & \mathtt{a}ED \mid \mathtt{b}E \end{array}$$

(*i*) Si calcolino i First e i Follow per tutti i nonterminali. (*ii*) Si rimuovano i simboli inutili per ottenere una grammatica G' senza simboli inutili, che sia equivalente a 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. Si consideri la grammatica G con simbolo iniziale S:

$$\begin{array}{lcl} S & \rightarrow & \epsilon \mid AB \\ A & \rightarrow & \epsilon \mid \mathtt{c}A \mid \mathtt{c} \\ B & \rightarrow & \epsilon \mid \mathtt{ab} \mid \mathtt{a}B\mathtt{b} \end{array}$$

(*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 *cab*.

10. Si consideri la grammatica G con simbolo iniziale S:

$$\begin{array}{lcl} S & \rightarrow & \mathtt{a}S\mathtt{c} \mid \mathtt{a}A\mathtt{b} \\ A & \rightarrow & \mathtt{a}A\mathtt{b} \mid \epsilon \end{array}$$

(*i*) Calcolare $L(G)$. (*ii*) Verificare se G sia di classe SLR(1).

1)

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

Esterna Parallela (EP)

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

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

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

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

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

$$\to_{EP} \neq \to_{IS}$$

$$\langle (2-5)=0 \text{ and } ff, \sigma \rangle \not\to_{IS}$$

perchè l'argomento di sinistra è bloccato (errore)

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

2)

ide := (maius | cifra) Tutto* minus Tutto* cifra

Tutto := lettera | cifra

cifra := [0 - 9]

lettera := minus | maius

minus := [a - z]

maius := [A - Z]

3) $L = \{ a^{2n+1} b^{2m+1} c^{2k+1} \mid m \geq n \geq 0, \ k \geq 0 \}$

è libero perché è generato dalla grammatica libera

$$S \to BAC$$
$$A \to bbA \mid \varepsilon$$
$$B \to ab \mid aaBbb$$
$$C \to ccC \mid c$$

Infatti, $L(A) = \{ b^{2n} \mid n \geq 0 \}$

$$L(B) = \{ a^{2n+1} b^{2n+1} \mid n \geq 0 \}$$
$$L(C) = \{ c^{2n+1} \mid n \geq 0 \}$$

$$L(S) = L(B) \cdot L(A) \cdot L(C)$$
$$= \{ a^{2n+1} b^{2n+1+2m} c^{2k+1} \mid n, m, k \geq 0 \}$$
$$= \{ a^{2n+1} b^{2m+1} c^{2k+1} \mid m \geq n \geq 0, \ k \geq 0 \}$$

L <u>non è regolare</u> e lo dimostriamo usando il pumping lemma a rovescio

- Fissiamo $N > 0$ generico $(\forall N > 0)$
- Scegliamo $z = a^{2N+1} b^{2N+1} c$ $(\exists z \in L, |z| \geq N)$
- Per ogni u, v, w tali che
  - $z = uvw$
  - $|uv| \leq N$
  - $|v| \geq 1$

  deve essere $v = a^J$ con $J \geq 1$.
- Allora per $k_0 = 2$ abbiamo che

$$uv^2w = a^{2N+1+J} b^{2N+1} c \notin L$$

perché ora ho più "a" che "b"

$\Rightarrow$ L non è regolare

4) Sì, un linguaggio libero nondeterministico non ambiguo è il ling. delle palindrome di lungh. pari

$$L = \{ w w^R \mid w \in (a|b)^* \}$$

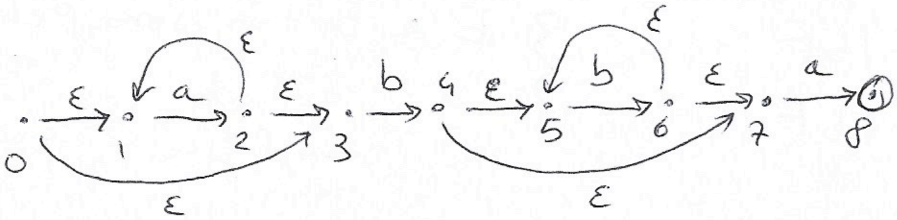
che è generato dalla grammatica

$$S \to \varepsilon \mid aSa \mid bSb \quad \Big] \, G$$

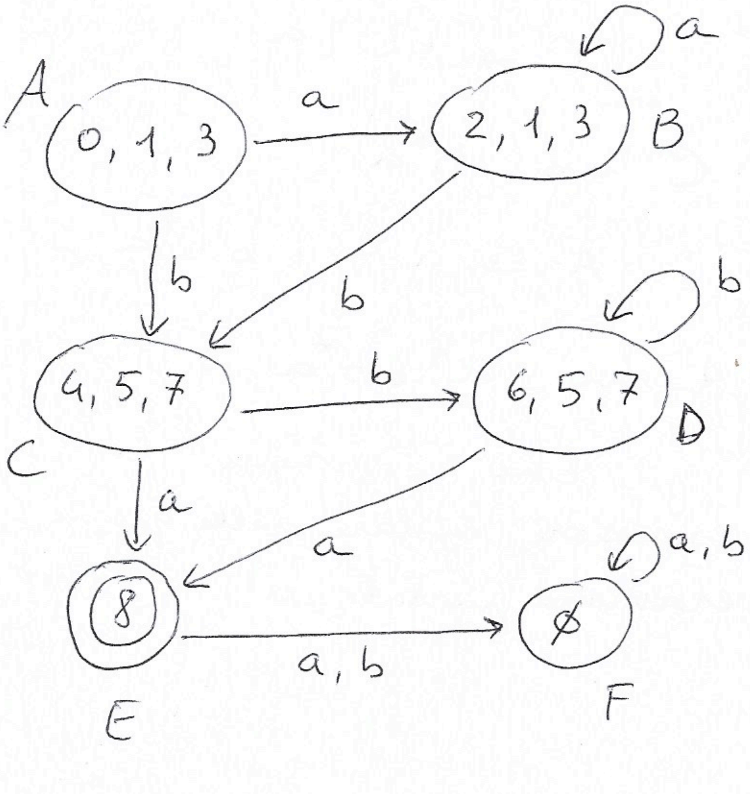
G è libera e anche non ambigua

---

5) $(a^* b)(b^* a)$



NFA associato secondo la costruzione canonica



DFA ottenuto con costruzione per sottoinsiemi

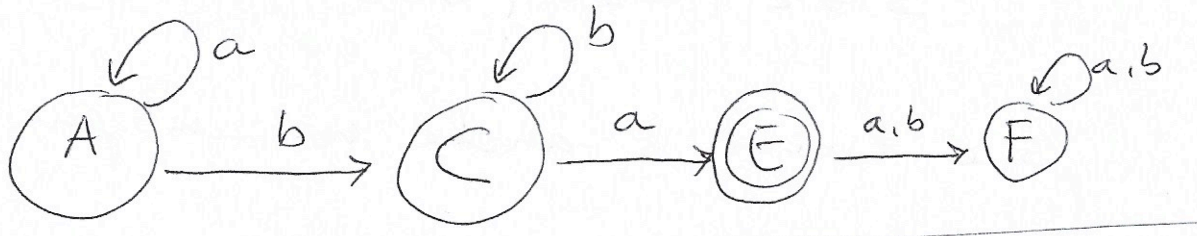
Vediamo se è minimo, utilizzando l'algoritmo di tabelle a scala

B					
C	X_1	X_1			
D	X_1	X_1			
E	X_0	X_0	X_0	X_0	
F	X_2	X_2	X_1	X_1	X_0
	A	B	C	D	E

$A \sim B$

$C \sim D$

quindi posso fondere insieme questi stati per ottenere il DFA minimo



6)

$$
\left.\begin{aligned}
A &\to aA \mid bC \\
C &\to bC \mid aE \mid a \\
E &\to aF \mid bF \\
F &\to aF \mid bF
\end{aligned}\right\}
$$

E e F non sono generatori

$\Downarrow$

semplifico la grammatica

$A \to aA \mid bC$

$C \to bC \mid a$

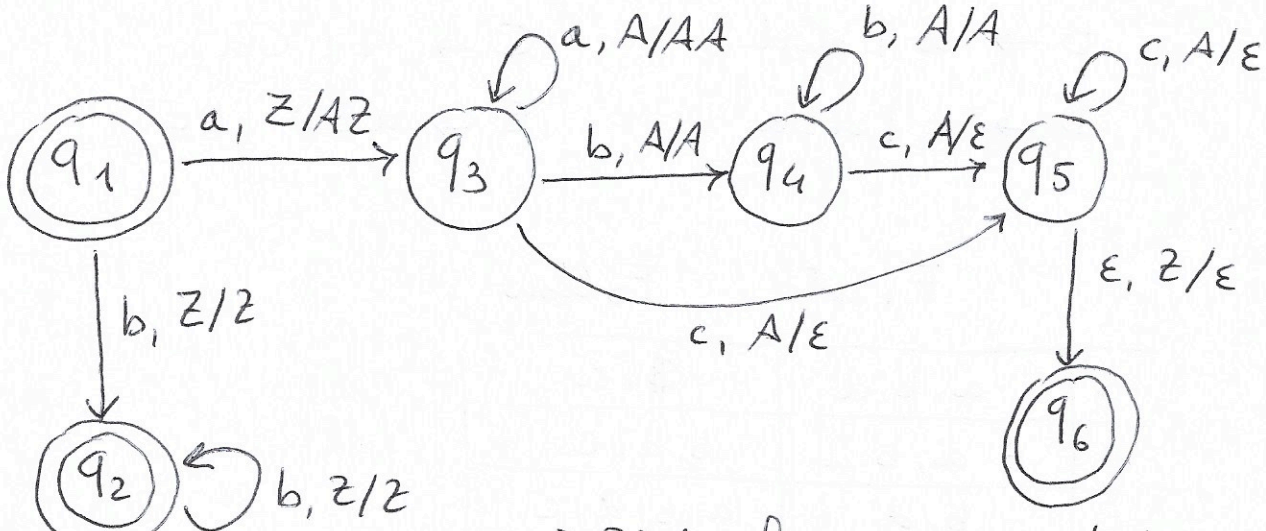
$C \rightsquigarrow b^*a$

$A \to aA \mid bb^*a$

$A \rightsquigarrow a^*bb^*a$

che è proprio la espressione regolare dalla quale eravamo partiti

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



DPDA che riconosce L_1 per stato finale

$L_2 = \{ a^n b^n c^m \mid n, m \geq 0 \}$ è libero deterministico

$L_1 \cap L_2$ non è libero. Infatti,

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

che in classe è stato dimostrato non libero usando il pumping theorem a rovescio.

8)

$$G \begin{cases} S \to ABCc \mid Ea & A \to \varepsilon \mid aBE \mid BSa \\ B \to d \mid bCd & C \to A \mid Cd \\ D \to c \mid dSa & E \to aED \mid bE \end{cases}$$

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

- E non è generatore, e, rimuovendo E, D non è più raggiungibile

$$G' \begin{cases} S \to ABCc \\ A \to \varepsilon \mid BSa \\ B \to d \mid bCd \\ C \to A \mid Cd \end{cases}$$

- per rimuovere la prod-ε, è necessario calcolare i simboli annullabili $N(G) = \{A, C\}$

$$G'' \begin{cases} S \to ABCc \mid BCc \mid ABc \mid Bc \\ A \to BSa \\ B \to d \mid bCd \mid bd \\ C \to A \mid Cd \mid d \end{cases}$$

• di produzioni unitarie c'è solo $C \to A$.
Non ci sono altre coppie unitarie (non banali)

$$G''' \left[\begin{array}{l} S \to ABC_c \mid BC_c \mid AB_c \mid B_c \\ A \to BSa \\ B \to d \mid bCd \mid bd \\ C \to \underbrace{BSa} \mid Cd \mid d \end{array}\right.$$

• Infine, la ricorsione sx è presente solo in $C \to Cd$
Quindi la grammatica G^{IV} sostituisce le produzioni per C in G''' con le seguenti:

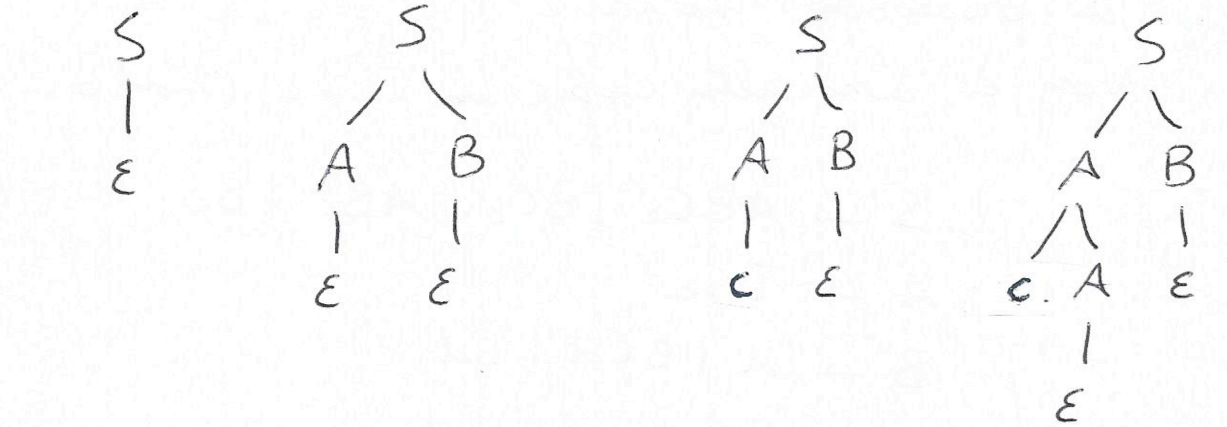
$$C \to BSaC' \mid dC'$$
$$C' \to dC' \mid \varepsilon$$

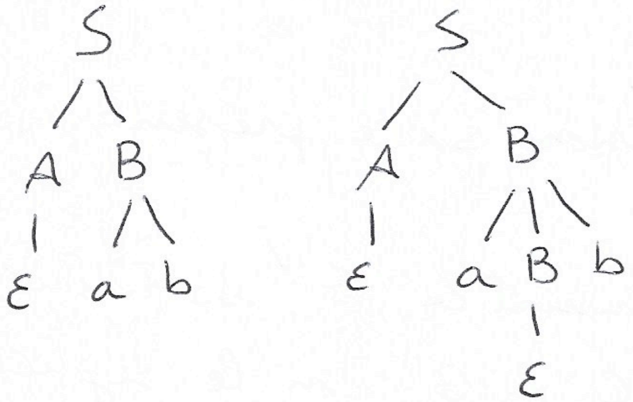
---

9)
$$\left.\begin{array}{l} S \to \varepsilon \mid AB \\ A \to \varepsilon \mid cA \mid c \\ B \to \varepsilon \mid ab \mid aBb \end{array}\right] G$$

G è ambigua perché ci sono parole che ammettono più di un albero di derivazione, ad esempio

$$\varepsilon, \quad c, \quad ab$$





per cui tutti e 3 i nonterminali presentan problemi per ambiguità

- $A \to \varepsilon \mid cA$ è non ambiguo e genera ancora il linguaggio c^*

$B \to \varepsilon \mid aBb$ è non ambiguo e genera ancora il linguaggio $\{a^n b^n \mid n \geq 0\}$

$S \to AB$ è non ambiguo e genera ε in modo univoco

Quindi
$$\left.\begin{array}{l} S \to AB \\ A \to \varepsilon \mid cA \\ B \to \varepsilon \mid aBb \end{array}\right]$$
G' è equivalente a G ma non ambigua

- G' è di classe LL(1). Infatti

(1) - $First(\varepsilon) \cap First(cA) = \emptyset$
$\{\varepsilon\} \cap \{c\}$

- $Follow(A) \cap First(cA) = \emptyset$
$\{a, \$\}$ $\{c\}$

(2) $- \; First(\varepsilon) \cap First(aBb) = \emptyset$

$\{\varepsilon\} \cap \{a\}$

$- \; Follow(B) \cap First(aBb) = \emptyset$

$\{b, \$\} \cap \{a\}$

	a	b	c	\$
S	$S \to AB$		$S \to AB$	$S \to AB$
A	$A \to \varepsilon$		$A \to cA$	$A \to \varepsilon$
B	$B \to aBb$	$B \to \varepsilon$		$B \to \varepsilon$

$First(AB) = \{a, c, \varepsilon\}$

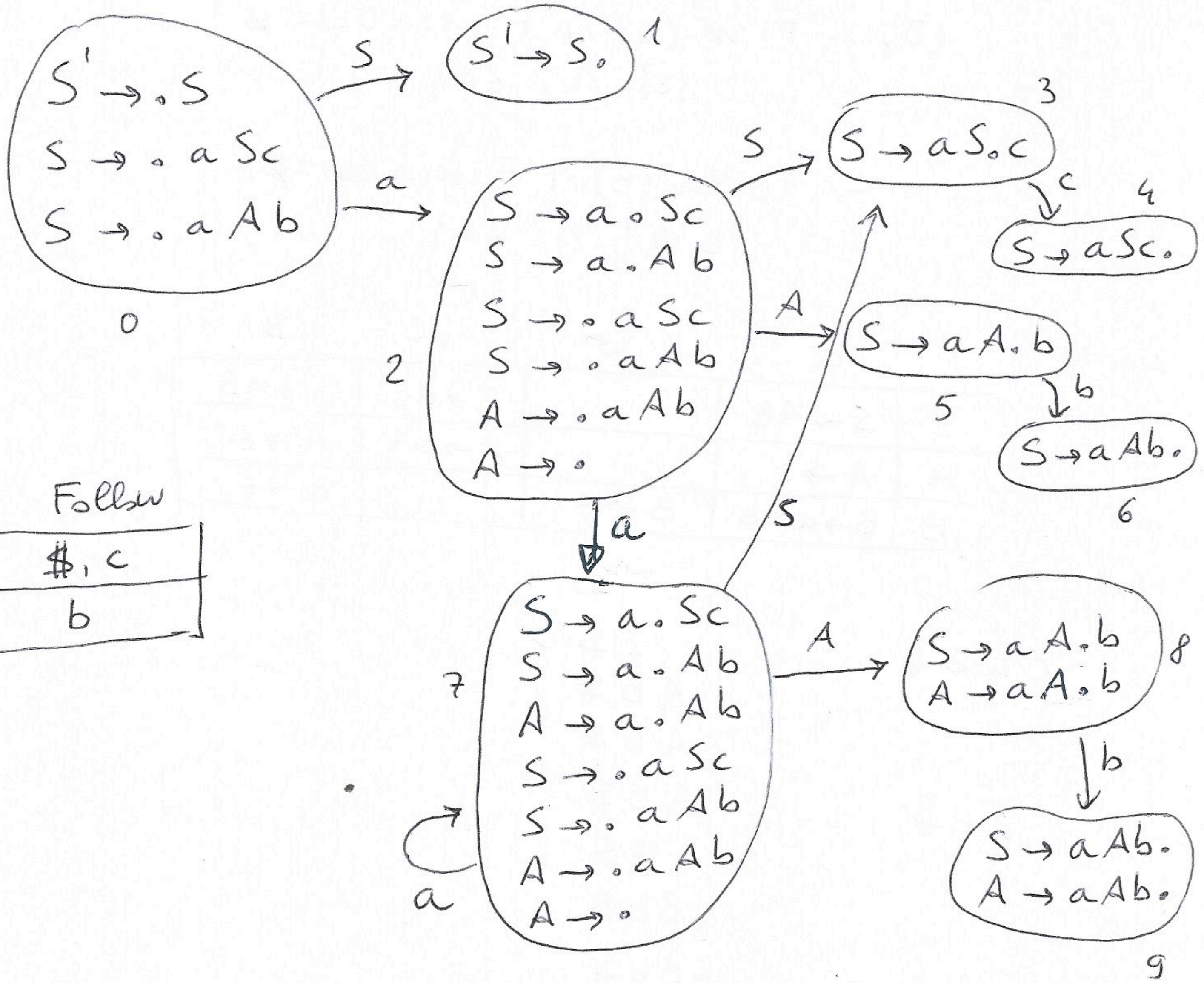
cab\$	S\$
	AB\$
	cAB\$
ab\$	AB\$
	B\$
	aBb\$
b\$	Bb\$
	b\$
\$	\$

ok, accept!

---

10) $G \left[\begin{array}{l} S \to \overset{(1)}{aSc} \mid \overset{(2)}{aAb} \\ A \to \underset{(3)}{aAb} \mid \underset{(4)}{\varepsilon} \end{array} \right.$

$L(G) = \{ a^n b^m c^k \mid n = m + k, \; m \geq 1, \; k \geq 0 \}$



Follow

S	$, c
A	b

	a	b	c	$	S	A
0	S2				G1	
1				ACC		
2	S7	R4			G3	G5
3			S4			
4			R1	R1		
5		S6				
6			R2	R2		
7	S7	R4			G3	G8
8		S9				
9		R3	R2	R2		

La Tabella SLR(1) non presenta conflitti, quindi G è una grammatica di classe SLR(1)