

Tempo a disposizione: ore 2.

Svolgere gli esercizi 1-4 e 5-8 su due fogli differenti.

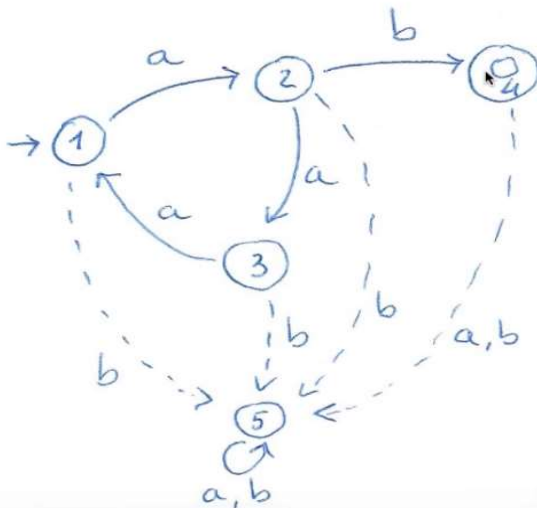
1. Una password è una qualunque sequenza di caratteri (lettere maiuscole, minuscole e numeri) che deve iniziare e terminare con una lettera maiuscola, deve includere anche almeno un numero ma una sola lettera minuscola. Fornire una definizione regolare per le password di questo tipo.
2. Si costruisca il minimo DFA che riconosca il linguaggio  $L = \{a^{3k+1}b \mid k \geq 0\}$ .
3. Si dimostri che il linguaggio  $L = \{a^n b^{2n} \mid n \geq 0\}$  è di classe LL(1).
4. Si consideri la seguente grammatica  $G$  con simbolo iniziale  $S$ :

$$\begin{aligned} S &\rightarrow aSc \mid bAc \\ A &\rightarrow Ab \mid \epsilon \end{aligned}$$

(i) Che linguaggio genera  $G$ ? (ii) Si verifichi se il parser LR(0) presenta conflitti. (iii) Se sono presenti conflitti, costruire il parser SLR(1), altrimenti mostrare il funzionamento del parser LR(0) su input  $abcc$ .

1)  $pswd := \text{maius} \text{ middle } \text{maius}$   
 $\text{maius} := [A - Z]$   
 $\text{middle} := \text{vario}^* (\text{num } \text{vario}^* \text{ minus} \mid \text{minus } \text{vario}^* \text{ num}) \text{ vario}^*$   
 $\text{vario} := \text{maius} \mid \text{num}$   
 $\text{minus} := [a - z]$   
 $\text{num} := [0 - 9]$

2)  $L = \{a^{3k+1}b \mid k \geq 0\} = \{ab, a^4b, a^7b, \dots\}$



2	$X_1$			
3	$X_2$	$X_1$		
4	$X_0$	$X_0$	$X_0$	
5	$X_1$	$X_1$	$X_2$	$X_0$
	1	2	3	4

$$3) \quad L = \{ a^n b^{2n} \mid n \geq 0 \} \text{ è LL(1)?}$$

$$S \rightarrow aSbb \mid \varepsilon \quad ] \quad G \text{ è tale che } L(G) = L$$

Verifichiamo che  $G$  è di classe  $LL(1)$ . Se così è, allora  $L$  è di classe  $LL(1)$ .

$$\bullet \text{ First}(aSbb) \cap \text{First}(\varepsilon) = \emptyset$$

$$\{a\} \cap \{\varepsilon\} = \emptyset$$

$$\bullet \text{ First}(aSbb) \cap \text{Follow}(S) = \emptyset$$

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

$$4) \quad \left. \begin{array}{l} S \rightarrow aSc \mid bAc \\ A \rightarrow Ab \mid \varepsilon \end{array} \right] G$$

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

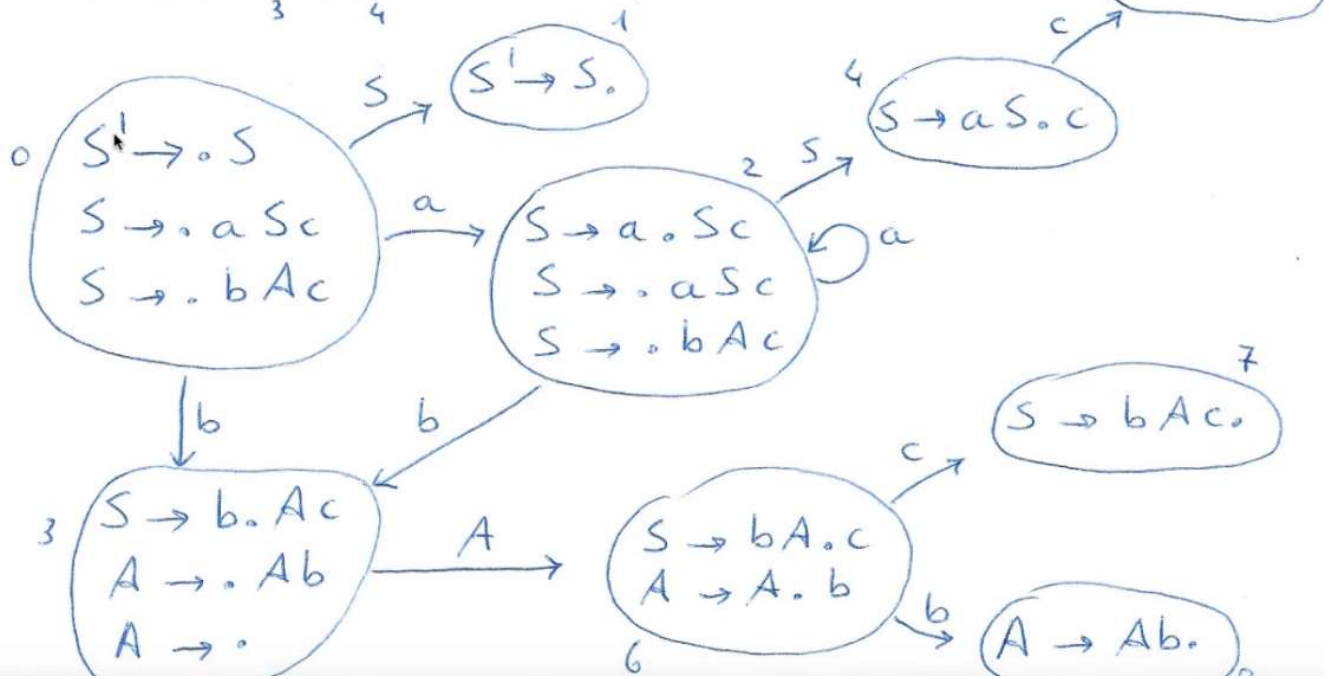
$$L(A) = \{ b^m \mid m \geq 0 \}$$

$$L(S) = \{ a^n b \cdot L(A) \cdot c \cdot c^m \mid n \geq 0 \}$$

$$= \{ a^n b^{m+1} c^{m+1} \mid n, m \geq 0 \}$$

$$S \rightarrow aSc^1 \mid bAc^2$$

$$A \rightarrow Ab_3 \mid \epsilon_4$$



	a	b	c	#	S	A
0	S2	S3			G1	
1				ACC		
2	S2	S3			G4	
3	R4	R4	R4	R4		G6
4			S5			
5	R1	R1	R1	R1		
6		S8	S7			
7	R2	R2	R2	R2		
8	R3	R3	R3	R3		

(0, ε, abcc#)

(02, a, bcc#)

(023, ab, cc#)

(0236, abA, cc#)

(02367, abAc, c#)

(024, aS, c#)

(0245, aSc, #)

(01, S, #)

ACCEPT!

5. Si consideri il seguente frammento in uno pseudolinguaggio con scope dinamico e parametri di ordine superiore:

```

int x = 3;
int n = 7;
void g(){
    write(n+x)
}
void foo (int f(), int x){
    if (x==0) f();
    else
    {int x = 20
    foo(f,0);
    g();
    }
}
{
    int x = 5;
    foo(g,1)
}

```

Si dica cosa stampa il frammento con shallow binding.

6. Un certo linguaggio adotta un'allocazione della memoria completamente statica. Quali delle seguenti caratteristiche *sono compatibili* con tale politica di allocazione? (a) Blocchi annidati; (b) Definizione di funzioni annidate; (c) Definizione di funzioni ricorsive; (d) Passaggio dei parametri per valore; (e) Passaggio dei parametri per riferimento; (f) comandi allocazione esplicita della memoria.
7. Spiegare brevemente (entro le 10 righe), che cos'è il "garbage" nell'ambito dei linguaggi di programmazione, perché si genera e perché è un problema.
8. Si assuma un linguaggio di programmazione a oggetti, con tipi nominali e passaggio per riferimento. Le classi A, B, e C sono tali che B è sottoclasse di A e C è sottoclasse di B. Nel linguaggio, il tipo T[] indica un array di oggetti della classe T con scritture e letture covarianti rispetto ai sottotipi.

Indicare quali istruzioni verrebbero segnate come \*non\* corrette dal controllore dei tipi (e indicare brevemente perché).

```

void f( A a, B b, C c, A[] aa, B[] bb, C[] cc ) {
    c = b; // I1 controvarianza - X
    a = b; // I2
    c = bb[ 0 ]; // I3 controvarianza - X
    aa = bb; // I4
    a = aa; // I5 assegnamento non compatibile - X
    bb = bb; // I6
    a = bb[ 0 ]; // I7
    aa[ 0 ] = cc; // I8 assegnamento non compatibile - X
    b = bb; // I9
    aa[ 0 ] = cc[ 0 ]; // I10
    cc = bb; // I11 controvarianza - X
    c = bb[ 0 ]; // I12 X
}

```

B <: A  
C <: B

X = errato, istruzione non corretta