

Tempo a disposizione: 2 ore e 30 minuti.

1. Per quali valori delle variabili  $X$  e  $Y$  la seguente espressione

$$I_X^{L_0}(C_{Y,L_1}^{L_1}, I_{L_1}^{L_0})$$

ha senso? Se sì, calcola qualcosa di utile?

2. Descrivere le regole di semantica operativa strutturata per il comando `for i:=1 to n do c`. (Si suppone che la variabile di controllo  $i$  ed anche la variabile  $n$  non vengano mai modificate durante l'esecuzione del corpo  $c$ . *Suggerimento*: Ricondursi al caso del `while`.)
3. Fornire una definizione regolare per *password*, che deve essere una qualunque sequenza di lettere e/o cifre che deve terminare con una cifra e deve contenere almeno una lettera minuscola ed almeno una lettera maiuscola (in qualsiasi ordine).
4. Classificare il linguaggio  $L = \{b^{m+k}a^n \mid n, k \geq 0, m \geq 1\}$ , ovvero dire se  $L$  è regolare, oppure libero ma non regolare, oppure non libero, giustificando adeguatamente la risposta.
5. Si consideri l'espressione regolare  $a^*(a\emptyset)^*b$ . Si costruisca l'automa NFA  $M$  associato, secondo la costruzione vista a lezione. Si trasformi l'NFA  $M$  nell'equivalente DFA  $M'$ , secondo la costruzione per sottoinsiemi vista a lezione.
6. Preso il DFA  $M'$  calcolato al punto precedente, si verifichi se è minimo; se non lo fosse, lo si minimizzi per ottenere un DFA  $M''$ ; poi si ricavi da  $M''$  la grammatica regolare associata, seguendo la costruzione vista a lezione; quindi si semplifichi la grammatica ottenuta, eliminando i simboli inutili; infine, si ricavi dalla grammatica l'espressione regolare associata.
7. Dati i linguaggi  $L_1$  ed  $L_2$ , il primo libero, ma non regolare, e il secondo regolare, a quale classe appartiene il linguaggio  $L_1 \cup L_2 = \{w \mid w \in L_1 \vee w \in L_2\}$ ? Può  $L_1 \cup L_2$  essere finito?
8. È vero che, per ogni linguaggio finito  $L$ , esiste un DPDA  $N$  tale che  $L = L[N]$ ? Motivare la risposta.
9. Mostrare che  $L = \{a^n b^n \mid n \geq 2\}$  è libero, costruendo un semplice parser ~~Top-down~~ *Top-down* nondeterministico (come PDA con un solo stato che riconosca  $L$  per pila vuota).
10. Si consideri la seguente grammatica  $G$  con simbolo iniziale  $S$ :

$$\begin{array}{ll} S \rightarrow AB \mid C \mid aE & A \rightarrow \epsilon \mid aSB \\ B \rightarrow a \mid bBC & C \rightarrow A \mid Cd \\ D \rightarrow c \mid dS & E \rightarrow aDE \end{array}$$

- (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  $G'$  (a meno di  $\epsilon$ ). (iv) Si rimuovano le produzioni unitarie da  $G''$  per ottenere una grammatica  $G'''$  senza produzioni unitarie equivalente a  $G''$ .
11. Si consideri la grammatica  $G$  con simbolo iniziale  $S$ :

$$\begin{array}{l} S \rightarrow AB \\ A \rightarrow \epsilon \mid aAb \\ B \rightarrow \epsilon \mid cBd \end{array}$$

- (i) Determinare il linguaggio generato  $L(G)$ . (ii) Verificare che  $G$  è di classe LL(1). (iii) Costruire la tabella di parsing LL(1) per  $G$ . (iv) Mostrare il funzionamento del parser LL(1) su input  $abcc$ .
12. Si consideri la grammatica  $G$  del punto precedente. (i) Costruire l'automa canonico LR(0). (ii) Costruire la tabella di parsing SLR(1) e verificare che non ci sono conflitti. (iii) Mostrare il funzionamento del parser SLR(1) per l'input  $abcc$ .