

Tempo a disposizione: 2 ore e 30 minuti.

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

$$\mathcal{I}_X^{L_0}(C_{Y, L_3}^{L_1}, C_{X, L_2}^Z)$$

ha senso? E cosa viene calcolato?

2. Descrivere le regole di semantica operativa strutturata per il comando **repeat c until b** di Pascal. (Ricordo che il comando c viene eseguito almeno una volta e che l'iterazione termina quando b vale vero. *Suggerimento*: ricondursi al **while**.)
3. Fornire una definizione regolare per *password*, che deve essere una qualunque sequenza di lettere e/o cifre che deve iniziare con una lettera maiuscola e deve contenere almeno una lettera minuscola ed almeno una cifra (in qualsiasi ordine).
4. Classificare il linguaggio $L = \{a^{n+1}b^m c^k \mid n, k \geq 0, m \geq 1\}$, ovvero dire se L è regolare, oppure libero ma non regolare, oppure non libero, giustificando adeguatamente la risposta.
5. Si consideri l'espressione regolare $b(a|\epsilon)^*b$. 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 due linguaggi L_1 ed L_2 , il primo regolare e il secondo libero deterministico, a quale classe appartiene il linguaggio $L_1 - L_2 = \{w \mid w \in L_1 \wedge w \notin L_2\}$? Può $L_1 - L_2$ essere finito?
8. È vero che, per ogni linguaggio regolare L , esiste una grammatica *non ambigua* G tale che $L = L(G)$? Motivare la risposta.
9. Mostrare che $L = \{a^{n+1}b^n \mid n \geq 0\}$ è libero, costruendo un semplice parser 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 :

$$\begin{array}{ll} S \rightarrow BAC \mid A & A \rightarrow a \mid bSD \\ B \rightarrow C \mid bDB & C \rightarrow \epsilon \mid Cd \\ D \rightarrow c \mid dD \end{array}$$

(i) Si calcolino i First e i Follow per tutti i nonterminali. (ii) Si rimuova la produzione epsilon per ottenere una grammatica G' senza produzioni epsilon, che sia equivalente a G . (iii) Si rimuovano le produzioni unitarie da G' per ottenere una grammatica G'' senza produzioni unitarie equivalente a G' . (iv) Si rimuova la ricorsione sinistra immediata per C per ottenere una G''' equivalente a G'' .

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

$$\begin{array}{l} S \rightarrow aSb \mid cAd \\ A \rightarrow \epsilon \mid cAd \end{array}$$

(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 $aabd$.

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 $aabd$.