CORSO DI LINGUAGGI DI PROGRAMMAZIONE — PARZIALE DI FINE MODULO
PROVA SCRITTA A DEL 26 MAGGIO 2020

Tempo a disposizione: 2 ore e 30 minuti.

1. Descrivere le regole di semantica operazionale strutturata per l'espressione aritmetica e_0 **div** e_1 (ovvero il quoziente della divisione del dividendo e_0 col divisore e_1) secondo la disciplina di valutazione esterna-destra (ED). Attenzione: se il valore di e_1 è 0, allora la valutazione di e_0 **div** 0 si blocca (o, se preferite, raggiunge uno stato di errore).

2. Costruire una grammatica G che generi il linguaggio $L = \{wa^n b^{n+1} w^R \mid w \in (c|d)^*, n \geq 0\}$.

3. Classificare il linguaggio L del punto precedente, ovvero dire se L è regolare, oppure libero ma non regolare, oppure non libero, giustificando adeguatamente la risposta.

4. Si consideri l'espressione regolare a^*aa^*. 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.

5. 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''; quindi si ricavi da M'' la grammatica regolare associata, seguendo la costruzione vista a lezione; infine, si ricavi dalla grammatica l'espressione regolare associata.

6. Siano $L_1 = \{a^{2n}b^{2n} \mid n \geq 0\}$ ed $L_2 = \{a^{3n}b^{3n} \mid n \geq 0\}$. Classificare il linguaggio $L_1 \cap L_2$. Sarebbe stato possibile sfruttare le proprietà di chiusura per determinare a quale classe appartenga il linguaggio $L_1 \cap L_2$? Giustificare la risposta.

7. Mostrare che $L = \{a^{n+1}b^{m+1}c^m \mid n, m \geq 0\}$ è libero deterministico, costruendo un opportuno DPDA che riconosca $L\$$ per pila vuota. È possibile costruire un DPDA che riconosca L per pila vuota?

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

$$\begin{array}{lcl} S & \to & AB \mid AC \\ A & \to & \epsilon \mid \mathtt{ab}A \mid C \\ B & \to & \mathtt{b} \mid \mathtt{b}DB \\ C & \to & \epsilon \mid \mathtt{c}C \\ D & \to & \mathtt{d} \mid \mathtt{d}SD \end{array}$$

(*i*) Si calcolino i First e i Follow per tutti i nonterminali. (*ii*) La grammatica G è di classe LL(1)? (*iii*) Si rimuovano le produzioni epsilon per ottenere una grammatica G' senza produzioni epsilon, che sia equivalente a G. (*iv*) Si rimuovano le produzioni unitarie, ottenendo una grammatica equivalente G''.

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

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

(*i*) Determinare il linguaggio generato $L(G)$. (*ii*) Verificare che G non è di classe LL(1). (*iii*) Manipolare la grammatica per ottenerne una equivalente G' di classe LL(1). (*iv*) Costruire la tabella di parsing LL(1) per G'. (*v*) Mostrare il funzionamento del parser LL(1) su input *cba*.

10. 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 se ci sono conflitti. (*iii*) Mostrare il funzionamento del parser SLR(1) per l'input *cba*.

11. Discutere la seguente affermazione: se L è regolare e $L' \subseteq L$, allora L' è regolare.