CORSO DI LINGUAGGI DI PROGRAMMAZIONE — PARZIALE DI FINE MODULO
PROVA SCRITTA B 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 **esp** e_1 (ovvero esponenziale con e_0 come base ed e_1 come esponente) secondo la disciplina di valutazione esterna-sinistra (ES). Attenzione: il valore di 0^0 è indefinito, cioè la valutazione di 0 **esp** 0 si blocca (o, se preferite, raggiunge uno stato di errore).
2. Costruire una grammatica G che generi il linguaggio $L = \{a^{n+2}b^m a^{m+1}b^n \mid n \geq 0, m \geq 1\}$.
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 aa^*a^*. 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. Sia $L_1 = \{a^i b^j \mid i \geq j \geq 0\}$ e $L_2 = \{a^{2n}b^m \mid n, m \geq 0\}$. Sfruttando le proprietà di chiusura, si può concludere se il linguaggio $\overline{L_1} \cap L_2$ sia regolare, oppure libero, oppure non libero? Giustificare la risposta.
7. Mostrare che $L = \{a^n c^m b^{2n+1} \mid n \geq 0, m \geq 1\}$ 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:

$$\begin{array}{rcl} S & \rightarrow & BCA \mid BD \\ A & \rightarrow & \mathsf{a} \mid \mathsf{ab}A \mid C \\ B & \rightarrow & \epsilon \mid B\mathsf{b} \\ C & \rightarrow & \mathsf{c} \mid \mathsf{c}SC \\ D & \rightarrow & \epsilon \mid \mathsf{d}DB \end{array}$$

   (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:

$$\begin{array}{rcl} S & \rightarrow & B\mathsf{b} \mid S\mathsf{c} \mid \epsilon \\ B & \rightarrow & \mathsf{a} \mid \mathsf{a}B\mathsf{b} \end{array}$$

   (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 il parser LL(1) per G'. (v) Mostrare il funzionamento del parser LL(1) su input abc.
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 abc.
11. Discutere la seguente affermazione: se L è libero e $L' \subseteq L$, allora L' è libero.