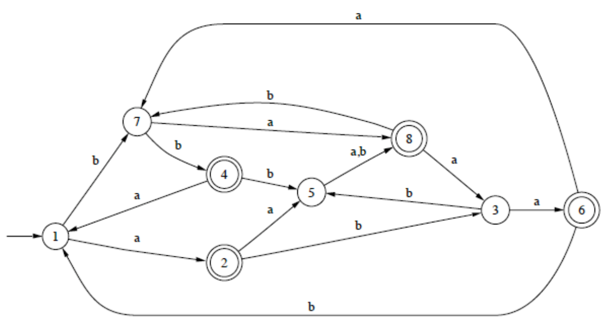
Fondamenti di Programmazione con Laboratorio
CdL in MATEMATICA

Appello 6 Febbraio 2020

N.B.: Negli esercizi di programmazione, vengono valutati anche l'uso dei costrutti appropriati, l'uso delle condizioni booleane, la leggibilità e l'efficienza del codice proposto. Inoltre, non è consentito l'uso di variabili globali o di variabili statiche. Laddove è utilizzato, il tipo `boolean` è definito da `typedef enum {false, true} boolean`.

ESERCIZIO 1
Dato il seguente automa a stati finiti deterministici.



- Si fornisca il corrispondente automa minimo M.
- Si derivi da M la grammatica regolare che generi il linguaggio riconosciuto da M.

ESERCIZIO 2
Dire se i seguenti linguaggi sono regolari o liberi dal contesto, giustificando la risposta e usando, se necessario, il pumping lemma per i linguaggi regolari e il pumping lemma per i linguaggi liberi.

- $L_1 = \{b^2 a^n b^m a^3 | m, n \geq 0\}$;
- $L_2 = \{a^n w | w \in \{a, b\}^*$ e la lunghezza di w è n, con $n \geq 0\}$, ad esempio $aaabab \in L_2$

* $L_3 = \{a^i b^j c^k | i, j \geq 0, k = max(i, j)\}$

ESERCIZIO 3
Si consideri il seguente problema P: dato un array `A` di n interi e la sua dimensione `n`, restituire `1` se `A` contiene *almeno* due elementi uguali, e `0` altrimenti.

1. Si scriva in C una funzione che risolva P e che abbia complessità quadratica al caso pessimo.
2. Si calcoli la complessità al caso ottimo e al caso pessimo dell'algoritmo scritto per il punto 1., motivando la risposta.
3. Si può risolvere il problema P in modo più efficiente? Se sì, esibire in C un algoritmo migliore, motivandone la complessità; se no, si dica il perché.

ESERCIZIO 4
Si descriva a parole, in non oltre 10 righe, l'algoritmo INSERTON-SORT, se ne indichi la complessità al caso ottimo e al caso pessimo, e se ne scriva l'*invariante* di ciclo.

ESERCIZIO 5

Dato un linguaggio L, si definisce $max(L) = \{w | w \in L \wedge \nexists x \neq \epsilon : wx \in L\}$.

- Dimostrare che se L è regolare anche $max(L)$ lo è (partire da un DFA per L e costruire un DFA per $max(L)$).

** Dimostrare che se L è libero $max(L)$ **non** lo è (basta trovare un controesempio).