

Fondamenti di Programmazione con Laboratorio CdL in MATEMATICA

I Appello 16 Gennaio 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 l'automa non deterministico A sull'alfabeto $\Sigma = \{0, 1, 2, 3\}$ descritto dalla seguente tabella di transizione:

		0	1	2	3
I	q_0	$\{q_0, q_1\}$	$\{q_0\}$	\emptyset	\emptyset
	q_1	\emptyset	\emptyset	$\{q_2\}$	\emptyset
	q_2	\emptyset	\emptyset	\emptyset	$\{q_1, q_3\}$
F	q_3	$\{q_3\}$	\emptyset	\emptyset	\emptyset

- Descrivere formalmente il linguaggio $L(A)$, accettato da A .
- Con la tecnica di *costruzione per sottoinsiemi* ricavare il corrispondente automa a stati finiti deterministico.
- Costruire la corrispondente **grammatica regolare** a partire dal DFA.

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. Nel caso si fornisca una grammatica, si dimostri che la grammatica genera tutte e sole le stringhe del linguaggio.

- $L_{pal} = \{w \in \{0, 1\}^* \mid w \text{ palindroma, ovvero } w = w^R, \}$;
- $L_{eq} = \{w \in \{0, 1, 2\}^* \mid |w|_0 = |w|_1 = |w|_2, \text{ con } |w|_i \text{ numero di occorrenze di } i \text{ in } w\}$.

ESERCIZIO 3

Si scriva in C una funzione **ricorsiva** (non occorre scrivere tutto il `main`, ma solo la chiamata alla funzione con gli opportuni parametri) che, dato in ingresso un array **ordinato** A di lunghezza n e contenente solo 0 e 1, restituisca un intero che corrisponde alla differenza tra il numero di occorrenze di 1 e il numero di occorrenze di 0 presenti nell'array. Ad esempio, dato $A = [0, 0, 1, 1, 1, 1, 1]$, l'output atteso è 3 ovvero 5-2, dato che il numero di occorrenze di 1 è 5, mentre il numero di occorrenze di 0 è 2.

La complessità in tempo dell'algoritmo **deve** essere $O(\log n)$ e deve essere argomentata.

ESERCIZIO 4

Si descriva l'algoritmo MergeSort in non più di 5 righe. Se ne indichi poi la complessità in tempo, motivando adeguatamente la risposta.

ESERCIZIO 5

Si definisce *min* di un linguaggio L come l'insieme delle stringhe w in L tali che nessun prefisso proprio (ovvero diverso da ϵ e da w stessa) di w sia in L .

$$\min(L) = \{w \in L : \nexists u \in L, v \in \Sigma^+, t.c. w = uv\}$$

Ad esempio, se $L' = \{00, 001, 0011, 101\}$ allora $\min(L') = \{00, 101\}$. Dimostrare, in generale, che se L è regolare, allora lo è anche $\min(L)$. (Partire da un DFA per costruire un DFA per $\min(L)$).

ESERCIZIO 6

• Date due espressioni booleane b_0, b_1 e uno stato σ , scrivere la valutazione dell'espressione composta $b_0 \vee b_1$, in cui \vee rappresenta l'OR logico (o disgiunzione), ovvero completare la regola, scrivendo a cosa corrispondono le premesse A e B nella della regola della semantica operativa, dove t è un generico valore booleano (può essere cioè sia *vero* che *falso*):

$$\frac{A \quad B}{(b_0 \vee b_1, \sigma) \rightarrow t}$$

- come cambierebbe la regola con la valutazione così detta *lazy*?