Algoritmi e Principi dell'Informatica

Tema d'esame del 20 Gennaio 2022

Esercizio 1 (8 punti).
Si consideri la seguente grammatica:

$$G = \begin{cases} S & \to CBS|CS|BS|a \\ CB & \to BC \\ CC & \to c \\ BB & \to b \end{cases}$$

a) Che tipo di grammatica è G?

b) Che linguaggio genera G?

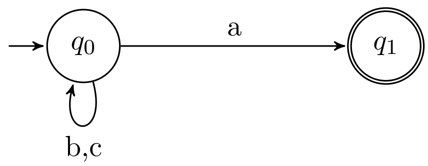
c) Si scriva un automa a potenza minima che accetti il linguaggio generato da G.

Soluzione

a) La grammatica è di tipo non ristretto.

b) Il linguaggio generato da G è regolare; in particolare è quello rappresentato dall'espressione regolare $(b|c)^*a$.

c) L'automa a potenza minima che riconosce $L(G)$ è il seguente automa a stati finiti:



Esercizio 2 (8 punti).
Si consideri una macchina di Turing universale M fissata, a nastro singolo. Sia Σ l'alfabeto del nastro di M.

a) È calcolabile la funzione $f : \begin{cases} 1 & \text{se } M(s) \neq \perp \\ 0 & \text{se } M(s) = \perp \end{cases}$ che riceve in ingresso una qualunque stringa $s \in \bigcup_{i=0}^{100} \Sigma^i$?

b) È calcolabile la funzione $g : \begin{cases} 1 & \text{se } M(s) \neq \perp \\ \perp & \text{se } M(s) = \perp \end{cases}$ che riceve in ingresso una qualunque stringa $s \in \bigcup_{i=2k, k\in\mathbb{N}} \Sigma^i$?

Soluzione

a) Sì. Il dominio della funzione da calcolare è finito, quindi essa è rappresentabile da una tabella di dimensioni finite, contenente la corrispondenza ingresso-uscita tra la stringa s ed il comportamento della macchina M quando s viene data in ingresso.

b) Sì, in particolare g può essere calcolata emulando $M(s)$ sino alla sua eventuale terminazione. Se $M(s)$ termina, allora g restituisce 1, risultando così conforme al comportamento richiesto.

Esercizio 3 (8 punti)
Si consideri un vettore di n interi. Si descriva in pseudocodice la procedura che ordina il vettore in ordine crescente in base al valore del resto della divisione di ognuno degli elementi per 4. Elementi con lo stesso valore di resto della divisione possono trovarsi in qualunque ordine tra loro. Il vettore ordinato avrà quindi alla sua estrema sinistra tutti gli elementi con resto nullo rispetto alla divisione per quattro, seguiti da tutti quelli aventi resto 1, a loro volta seguiti da tutti quelli aventi resto 2, ed infine da tutti gli elementi aventi resto 3.

La procedura di ordinamento deve avere complessità temporale $\mathcal{O}(n)$ e complessità spaziale $\mathcal{O}(1)$.

Soluzione

```
void ordina_mod_4(int *a, int len){
    int dst_idx = 0, tmp;
    for(int resto = 0; resto < 3; resto++){
        for(int pos = 0; pos < len; pos++){
            if(a[pos] % 4 == resto){
                tmp = a[pos];
                a[pos] = a[dst_idx];
                a[dst_idx] = tmp;
                dst_idx++;
            }
        }
    }
}
```

Esercizio 4 (8 punti).
Si voglia indicare con la notazione $f(\mathcal{O}(n))$ l'insieme di funzioni $\{f(g(n)) \mid g(n) \in \mathcal{O}(n)\}$. Si valuti la validità di ciascuna delle seguenti espressioni, giustificando le proprie risposte:

a) $\log(\mathcal{O}(n)) = \mathcal{O}(\log(n))$

b) $2^{\mathcal{O}(n)} = \mathcal{O}(2^n)$

c) $(\mathcal{O}(n))^k = \mathcal{O}(n^k)$

Soluzione

a) Falsa. Si consideri ad esempio $f(n) = 2\log(n)$. Abbiamo chiaramente che $f(n) \in \mathcal{O}(\log(n))$. Tuttavia $f(n) \notin \log(\mathcal{O}(n))$, cioè non è vero che $f(n)$ è una funzione della forma $\log(g(n))$, con $g(n) \in \mathcal{O}(n)$; infatti $f(n) = \log(n^2)$ e $n^2 \notin \mathcal{O}(n)$.

b) Falsa. Si prenda $f(n) = 2n$: certamente $f(n) \in \mathcal{O}(n)$, quindi $2^{f(n)} \in 2^{\mathcal{O}(n)}$, ma $2^{f(n)} = 2^{2n} \notin \mathcal{O}(2^n)$.

c) Vera. L'insieme $\mathcal{O}(n)$ è dato dalle funzioni g per cui esistono $c > 0, n_0 > 0$ tali che $g(n) \leq cn$ per $n > n_0$. Affinché $(g(n))^k \in \mathcal{O}(n^k)$ devono allora esistere $c' > 0, n_0' > 0$ tali che $(g(n))^k \leq c' \cdot n^k$ per $n > n_0'$, che è senz'altro verificata per c' opportuno e $n_0' = n_0$. Infatti, per $n > n_0$, $g(n) \leq cn$ e quindi $(g(n))^k \leq (cn)^k \leq c' \cdot n^k$ se $c' > c^k$.

   L'altra direzione è analoga: sia h una funzione in $\mathcal{O}(n^k)$, cioè tale per cui esistono $c > 0, n_0 > 0$ tali che $h(n) \leq cn^k$ per $n > n_0$. Affinché valga $h(n) \in \mathcal{O}(n)^k$ si deve mostrare che esiste una funzione $g(n) \in \mathcal{O}(n)$ tale che vale $h(n) = g(n)^k$. Tale funzione è $\sqrt[k]{h(n)}$: si ha infatti che $\sqrt[k]{h(n)} \in \mathcal{O}(n)$, poiché $\sqrt[k]{h(n)} \leq \sqrt[k]{c}n$ per $n > n_0$.