Informatica teorica

Esercizio 1 (8 punti)
Si consideri il linguaggio $L = \{x.y \mid x, y \in \{0,1\}^+, |x| = |y| \text{ oppure } 2|x| = |y|\}$. Si definisca un modello formale a potenza minima che accetti L.

Soluzione

L è un linguaggio regolare, infatti $L = \{x \in \{0,1\}^+ \text{ tale che } |x| \mod 2 = 0 \text{ oppure } |x| \mod 3 = 0\}$.

Esso è generato dalla seguente grammatica regolare:

$$
\begin{aligned}
S &\rightarrow\ 0A \mid 1A \mid 0A' \mid 1A' \\
A' &\rightarrow\ 0 \mid 1 \mid 0B' \mid 1B' \\
B' &\rightarrow\ 0A' \mid 1A' \\
A &\rightarrow\ 0B \mid 1B \\
B &\rightarrow\ 0 \mid 1 \mid 0C \mid 1C \\
C &\rightarrow\ 0A \mid 1A
\end{aligned}
$$

Esercizio 2 (8 punti)

Per ciascuna delle seguenti funzioni si chiarisca, barrando la casella corrispondente e motivando adeguatamente la risposta, se è calcolabile.

1. $h_1(x,y) = \begin{cases} 1 & \text{se } f_y(x) > 1 \\ 0 & \text{altrimenti.} \end{cases}$

2. $h_2(x,y) = \begin{cases} 1 & \text{se } f_y(x) > 1 \\ \perp & \text{altrimenti.} \end{cases}$

3. $h_3(x,y) = \begin{cases} \perp & \text{se } f_y(x) > 1 \\ 0 & \text{altrimenti.} \end{cases}$

4. $h_4(x,y) = \begin{cases} \perp & \text{se } f_y(x) > 1 \\ \perp & \text{altrimenti.} \end{cases}$

Soluzione

1. h_1 non è calcolabile. Fissato x (ad esempio $x = 0$), stabilire se $f_y(0) > 1$ è indecidibile per il teorema di Rice (almeno una funzione calcolabile restituisce un valore > 1 nel punto 0 e almeno una non lo fa). A maggior ragione, è indecidibile stabilire se $f_y(x) > 1$, che generalizza il caso $x = 0$. Se h_1 fosse calcolabile, sarebbe possibile decidere se $f_y(x) > 1$, che abbiamo appena mostrato essere indecidibile.

2. h_2 è calcolabile: ponendo in esecuzione mediante macchina di Turing universale la macchina y-esima con ingresso x, se questa si arresta e il risultato è > 1 posso restituire 1, se il risultato è ≤ 1 entro di proposito in un loop infinito, altrimenti la macchina non si arresta, compatibilmente con la definizione della funzione.

3. h_3 non è calcolabile. Sia P il problema di stabilire se $f_y(x) > 1$. Al punto 1 abbiamo mostrato che P è indecidibile; P è però semidecidibile (usando la tecnica descritta al punto 2). Pertanto il complemento di P è necessariamente non semidecidibile (se fosse semidecidibile, per il teorema "$\frac{1}{2} + \frac{1}{2} = 1$", P sarebbe decidibile, ma non lo è). Se h_3 fosse calcolabile, allora il complemento di P sarebbe semidecidibile, ma abbiamo appena mostrato che non lo è.

4. h_4 è calcolabile, in quanto è una funzione costante.

Algoritmi e Principi dell'Informatica
Soluzioni al Tema d'esame

15-1-2025

Algoritmi e strutture dati

Esercizio 3 (8 punti)

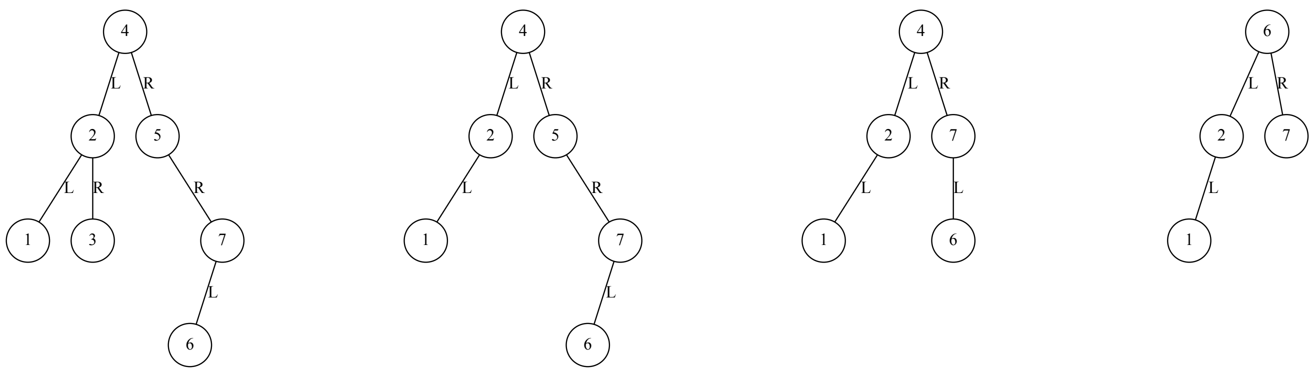
Si consideri il seguente codice che inserisce e cancella nodi in un BST:

popolaBST()
$T :=$ BST vuoto
for $x \in$ insieme di nodi con rispettive chiavi $4, 2, 1, 3, 5, 7, 6$
    tree-insert(T, x)
for $x \in$ insieme di nodi con rispettive chiavi $3, 5, 4$
    tree-delete(T, x)

1. Si disegni il BST risultante alla fine del ciclo di inserimenti e dopo ciascuna cancellazione di nodo.
2. Si valuti la complessità asintotica della procedura popolaBST()

Soluzione

1. I BST risultanti sono mostrati di seguito; l'etichetta indica se il figlio è sinistro ("L") o destro ("R").



2. Poiché c'è un numero di operazioni limitato da una costante e ciascuna operazione interviene su un BST di grandezza limitata da una costante, il costo della procedura è $\mathcal{O}(1)$.

Esercizio 4 (8 punti)

Si consideri una collezione di n elementi, ognuno dei quali rappresenta i seguenti dati contenuti in una tessera del codice fiscale: nome e cognome dell'intestatario, data di nascita e codice fiscale (il codice fiscale è una stringa alfanumerica unica per ogni cittadino italiano).

Si progetti e descriva una struttura dati in grado organizzare la collezione di elementi appena descritta in modo da poter effettuare:

- Inserimenti e cancellazioni nella/dalla collezione in tempo $\mathcal{O}(\log(n))$
- Ricerca in $\mathcal{O}(\log(n) + m)$ di m elementi, con $0 \leq m \leq n$, corrispondenti a persone nate consecutivamente a partire da una data indicata
- Ricerca di un elemento per codice fiscale in $\mathcal{O}(1)$

Si descriva, in modo sintetico e chiaro, l'organizzazione della struttura dati e le operazioni sopra citate.

Soluzione

Una possibile organizzazione è data da un albero rosso-nero, in cui gli elementi sono inseriti utilizzando come chiave la data di nascita. L'albero è affiancato da una tabella hash che utilizza il codice fiscale come chiave e contiene come valore il riferimento al nodo dell'albero avente il detto codice fiscale come chiave.

- Inserimenti / cancellazioni: si inserisce come da prassi degli alberi rosso-neri l'elemento nell'albero, mantenendo un riferimento allo stesso. Il riferimento viene quindi inserito nella tabella hash. Le cancellazioni procedono analogamente, cancellando il riferimento dalla tabella hash ed il nodo dall'albero
- La ricerca viene effettuata cercando il nodo corrispondente alla data indicata (o, se non esistente, si prosegue dal nodo a cui si è arrivati individuando il successore). Si itera la chiamata alla procedura Successore, estraendo i successivi $m - 1$ nodi
- È sufficiente effettuare un look-up nella tabella di hash.