Algoritmi e Principi dell'Informatica
Soluzioni al Tema d'esame
30-1-2025

Informatica teorica

Esercizio 1 (8 punti)
Si considerino i seguenti linguaggi definiti sull'alfabeto $\{a, b\}$ e, per ognuno dei due, si costruisca un automa riconoscitore o una grammatica a potenza minima. Si rammenta che $\lfloor x \rfloor$ indica la parte intera del valore x.

1. $L_1 = \{a^n b^{\lfloor mn \rfloor} \mid n \geq 0, 0 < m < 5, \text{ con } m \text{ e } n \text{ numeri naturali}\}$.

2. $L_2 = \{a^n b^{\lfloor mn \rfloor} \mid n \geq 0, m > 0, \text{ con } m \text{ numero reale e } n \text{ numero naturale}\}$.

Soluzione

1. L_1 è costruito come $a^n b^n \cup a^n b^{2n} \cup a^n b^{3n} \cup a^n b^{4n}$, e somiglia strutturalmente al consueto $a^n b^n \cup a^n b^{2n}$. Il riconoscitore a potenza minima è quindi un automa a pila non deterministico o una grammatica non contestuale. Per esempio:
$$\begin{array}{l} S \to A \mid B \mid C \mid D \mid \varepsilon \\ A \to aAb \mid ab \\ B \to aBbb \mid abb \\ C \to aCbbb \mid abbb \\ D \to aDbbbb \mid abbbb \end{array}$$

2. L_2 ammette tutte le stringhe dove una sequenza ininterrotta di a inizia la stringa e (se c'è almeno una a) una sequenza ininterrotta di b la segue, terminando la stringa. Si noti che il fatto che il rapporto tra il numero di a e b sia un qualunque razionale rende possibile una scelta arbitraria dei numeri di a e b. L_2 è quindi $a^* \cup a^+ b^+$, linguaggio regolare. Una grammatica regolare che lo genera è la seguente:
$$\begin{array}{l} S \to aA \mid aB \mid a \mid \varepsilon \\ A \to aA \mid aB \mid a \\ B \to bB \mid b \end{array}$$

Esercizio 2 (8 punti)
Siete stati incaricati di valutare l'efficacia di strumenti di Intelligenza Artificiale (IA) generativa per codice, per esempio Copilot. Per questo motivo, vi viene chiesto di creare un insieme finito e non vuoto di programmi C di riferimento, e per ognuno di questi programmi di definire un prompt da dare a Copilot per generare il programma richiesto (per esempio "genera un programma C che calcola la radice quadrata intera di un numero intero ricevuto in ingresso"). **NB**: Un programma di IA generativa come Copilot restituisce risposte (per esempio, programmi) diverse anche a fronte dello stesso prompt, quando invocato più volte.

1. L'obiettivo è di valutare se, per ognuno dei programmi di riferimento, il programma generato da Copilot funziona come desiderato (nell'esempio precedente, se calcola veramente la radice quadrata del numero ricevuto in ingresso) oppure no. E' fattibile il compito che vi hanno assegnato? Motivare opportunamente la risposta.

Soluzione

Basta considerare il caso in cui l'insieme di programmi contenga un unico elemento. In questo caso è necessario essere in grado di valutare se, dato un programma restituito da Copilot a fronte di un prompt, questo calcoli la stessa funzione calcolata dal corrispondente programma dell'insieme di riferimento. Chiaramente il problema diventa di equivalenza tra programmi, non decidibile per il teorema di Rice, dove l'insieme F di funzioni è la funzione calcolata dal programma di riferimento.

2. Supponiamo che, invece che studiare la capacità di generare programmi C, vi sia richiesto di studiare la capacità di generare automi a stati finiti (quindi, in questo caso si tratterebbe di creare un insieme di FSA di riferimento, ognuno con il relativo prompt, e chiedere allo strumento di IA generativa di produrre l'automa richiesto). E' fattibile il compito che vi hanno assegnato? Motivare opportunamente la risposta.

Soluzione

In questo caso il compito è fattibile: è sufficiente confrontare il linguaggio riconosciuto dall'FSA restituito da Copilot con quello di ogni singolo FSA appartenente all'insieme di riferimento. Due metodi operativi per effettuare il confronto, dati A_{rif} e A_{Cop}, sono:

1. Costruire l'automa $A_{\overline{rif}}$ che riconosce il complemento del linguaggio riconosciuto da A_{rif}. Costruire l'automa che riconosce il linguaggio intersezione tra quelli riconosciuti da $A_{\overline{rif}}$ e A_{Cop} e verificare se esso riconosce il linguaggio vuoto
2. Calcolare la forma minima di A_{rif} e A_{Cop} e verificare se gli automi ottenuti sono identici.

Algoritmi e Principi dell'Informatica

Soluzioni al Tema d'esame

30-1-2025

Algoritmi e strutture dati

Esercizio 3 (8 punti)

Si consideri una tabella hash di dimensione $m = 8$ e la funzione $h(k, i) = (k \mod m + \frac{1}{2}i + \frac{1}{2}i^2) \mod m$ (hash con ispezione quadratica). Lo stato iniziale della tabella è: $\boxed{7}\boxed{9}\boxed{-}\boxed{11}\boxed{12}\boxed{5}\boxed{-}\boxed{15}$.

1. Inserire le chiavi seguenti e mostrare la tabella dopo ciascun inserimento, se possibile: 14, 23, 20.

2. Dopo gli inserimenti, cancellare le chiavi seguenti e mostrare la tabella dopo ciascuna rimozione: 12, 7.

3. Mostrare la complessità asintotica, nel **caso pessimo**, rispetto a m e n, dell'inserimento di n chiavi in una tabella hash di dimensione m con la funzione hash $h(k, i)$ mostrata sopra, partendo da una tabella piena a metà. Motivare.

Soluzione

1.
   - $h(14, 0) = 6 \rightarrow$ ok. $\boxed{7}\boxed{9}\boxed{-}\boxed{11}\boxed{12}\boxed{5}\boxed{14}\boxed{15}$.
   - $h(23, 0) = 7 \rightarrow$ collisione! $h(23, 1) = 0 \rightarrow$ collisione! $h(23, 2) = 2 \rightarrow$ ok. $\boxed{7}\boxed{9}\boxed{23}\boxed{11}\boxed{12}\boxed{5}\boxed{14}\boxed{15}$.
   - La tabella è piena, quindi ci sarà per forza un overflow (dopo 8 collisioni)!
2.
   - $h(12, 0) = 4 \rightarrow$ ok. $\boxed{7}\boxed{9}\boxed{23}\boxed{11}\boxed{\text{DEL}}\boxed{5}\boxed{14}\boxed{15}$.
   - $h(7, 0) = 7 \rightarrow$ collisione! $h(7, 1) = 0 \rightarrow$ ok. $\boxed{\text{DEL}}\boxed{9}\boxed{23}\boxed{11}\boxed{\text{DEL}}\boxed{5}\boxed{14}\boxed{15}$.
3. Nel caso pessimo, ciascun inserimento sarà soggetto a tante collisioni quante sono le chiavi presenti in tabella (inizialmente $m/2$ e al più m, qualora la tabella diventasse completamente piena, il che si verifica se $n \geq m/2$). Questo significa che ogni singolo inserimento ha un costo pari a $\mathcal{O}(m)$ e quindi, complessivamente, il costo di n inserimenti è $\mathcal{O}(m \cdot n)$.

Esercizio 4 (8 punti)

Si consideri il linguaggio $L = \{a^n b^{mn} \mid 0 < m < 5, n \geq 0 \text{ numeri naturali}\}$.

1. Si descriva una macchina di Turing a k nastri che accetta il linguaggio L, e se ne calcolino le complessità spaziale e temporale.

2. Si descriva una macchina RAM che accetta il linguaggio L, e se ne calcolino le complessità spaziale e temporale, sia a costo costante, che a costo logaritmico.

Soluzione

1. La MT legge la stringa in ingresso, contando in binario le a sul primo nastro di memoria T_1 e le b sul secondo nastro T_2. La macchina poi controlla se i due contatori sono uguali, oppure se uno è il doppio, il triplo o il quadruplo dell'altro (ad es. calcolando $m \cdot T_1$ in un terzo nastro di memoria e confrontando con T_2). Al primo caso positivo, termina accettando.

   La complessità spaziale, data dalla somma del contenuto dei nastri, risulta chiaramente $\Theta(\log N)$, con N lunghezza della stringa in ingresso. La complessità temporale richiede la lettura della stringa con l'incremento dei contatori ($\Theta(N)$), poi il controllo del rapporto tra i contatori ($\Theta(\log N)$); quindi la complessità temporale risulta $\Theta(N)$. Si ricordi che incrementare (e decrementare) k volte un contatore binario ha complessità $\Theta(k)$, se il contatore viene scritto a partire dalla cifra meno significativa, e se ad ogni modifica del contatore vengono visitate solo le cifre strettamente necessarie.

2. La macchina RAM può inizialmente operare la lettura in maniera analoga alla MT, con il vantaggio di poter contare le a e le b usando solo 2 celle. In seguito la macchina deve solamente controllare il rapporto tra il contenuto dei due contatori, in tempo costante.

   Complessità con il criterio di costo costante: Spazio: $\Theta(1)$, Tempo: $\Theta(N)$.

   Complessità con il criterio di costo logaritmico: Spazio: $\Theta(\log N)$, Tempo: $\Theta(N + \log N)$ (scansione + controllo dei rapporti) $= \Theta(N)$.