Automi e Linguaggi Formali – 9/6/2022
Seconda prova intermedia – Secondo Turno

1. (12 punti) Una *Macchina di Turing con eliminazione* è una macchina di Turing deterministica a nastro singolo che può eliminare celle dal nastro. Formalmente la funzione di transizione è definita come

$$\delta : Q \times \Gamma \mapsto Q \times \Gamma \times \{L, R, D\}$$

dove L, R indicano i normali spostamenti a sinistra e a destra della testina, e D indica l'eliminazione della cella sotto la posizione corrente della testina. Dopo una operazione di eliminazione, la testina si muove nella cella che si trovava immediatamente a destra della cella eliminata.

Dimostra che qualsiasi macchina di Turing con eliminazione può essere simulata da una macchina di Turing deterministica a nastro singolo.

Soluzione. Mostriamo come convertire una macchina di Turing con Inserimento M in una TM deterministica a nastro singolo S equivalente. La simulazione usa il simbolo speciale # per segnare le celle che vengono eliminate.

$S =$ "Su input w:

1. La simulazione delle mosse del tipo $\delta(q, a) = (r, b, L)$ procede come nella TM standard: S scrive b sul nastro e muove la testina di una cella a sinistra. Se lo spostamento a sinistra porta la testina sopra un #, allora continua a spostare la testina a sinistra finché non si arriva in una cella che contiene un simbolo diverso dal #.
2. La simulazione delle mosse del tipo $\delta(q, a) = (r, b, R)$ procede come nella TM standard: S scrive b sul nastro e muove la testina di una cella a destra. Se lo spostamento a destra porta la testina sopra un #, allora continua a spostare la testina a destra finché non si arriva in una cella che contiene un simbolo diverso dal #.
3. Per simulare una mossa del tipo $\delta(q, a) = (r, b, D)$ la TM S scrive # nella cella corrente e e muove la testina di una cella a destra. Se lo spostamento a destra porta la testina sopra un #, allora continua a spostare la testina a destra finché non si arriva in una cella che contiene un simbolo diverso dal #.
4. Se in qualsiasi momento la simulazione raggiunge lo stato di accettazione di M, allora S termina con accettazione. Se in qualsiasi momento la simulazione raggiunge lo stato di rifiuto di M, allora S termina con rifiuto. Negli altri casi continua la simulazione dal punto 2."

2. (12 punti) Data una Turing Machine M, definiamo

$$\text{HALTS}(M) = \{w \mid M \text{ termina la computazione su } w\}.$$

Considera il linguaggio

$$I = \{\langle M \rangle \mid \text{HALTS}(M) \text{ è un insieme infinito}\}.$$

Dimostra che I è indecidibile.

Soluzione. La seguente macchina F calcola una riduzione mediante funzione $A_{TM} \leq_m I$:

$F =$ "su input $\langle M, w \rangle$, dove M è una TM e w una stringa:

1. Costruisci la seguente macchina M':

   $M' =$ "Su input x:
   1. Esegue M su input w.
   2. Se M accetta, *accetta*.
   3. Se M rifiuta, va in loop."

2. Ritorna $\langle M' \rangle$."

Mostriamo che F calcola una funzione di riduzione f da A_{TM} a I, cioè una funzione tale che

$$\langle M, w \rangle \in A_{TM} \text{ se e solo se } M' \in I.$$

- Se $\langle M, w\rangle \in A_{TM}$ allora la macchina M accetta w. In questo caso la macchina M' accetta tutte le parole, quindi HALTS(M) $= \Sigma^*$ che è un insieme infinito. Di conseguenza $M' \in I$.
- Viceversa, se $\langle M, w\rangle \notin A_{TM}$, allora la macchina M su input w rifiuta oppure va in loop. In entrambi i casi la macchina M' va in loop su tutte le stringhe, quindi HALTS(M) $= \emptyset$ che è un insieme finito. Di conseguenza $M' \notin I$.

3. (12 punti) Una 3-colorazione di un grafo non orientato G è una funzione che assegna a ciascun vertice di G un "colore" preso dall'insieme $\{1, 2, 3\}$, in modo tale che per qualsiasi arco $\{u, v\}$ i colori associati ai vertici u e v sono diversi. Una 3-colorazione è *sbilanciata* se esiste un colore che colora più di metà dei vertici del grafo.

UNBALANCED-3-COLOR è il problema di trovare una 3-colorazione sbilanciata:

$$\text{UNBALANCED-3-COLOR} = \{\langle G\rangle \mid G \text{ è un grafo che ammette una 3-colorazione sbilanciata}\}$$

(a) Dimostra che UNBALANCED-3-COLOR è un problema NP
(b) Dimostra che UNBALANCED-3-COLOR è NP-hard, usando 3-COLOR come problema NP-hard di riferimento.

Soluzione.

(a) Il certificato è un vettore c dove ogni elemento $c[i]$ è il colore assegnato al vertice i. Il seguente algoritmo è un verificatore V per UNBALANCED-3-COLOR:

$V =$ "Su input $\langle\langle G\rangle, c\rangle$:
1. Controlla se c è un vettore di n elementi, dove n è il numero di vertici di G.
2. Controlla se $c[i] \in \{1, 2, 3\}$ per ogni i.
3. Controlla se per ogni arco (i, j) di G, $c[i] \neq c[j]$.
4. Controlla se esiste un colore che compare in più di metà degli elementi di c.
5. Se tutte le condizioni sono vere, *accetta*, altrimenti *rifiuta*."

Ognuno dei passi dell'algoritmo si può eseguire in tempo polinomiale.

(b) Dimostriamo che UNBALANCED-3-COLOR è NP-Hard per riduzione polinomiale da 3-COLOR a UNBALANCED-3-COLOR. La funzione di riduzione polinomiale f prende in input un grafo $\langle G\rangle$ e produce come output un grafo $\langle G'\rangle$. Se n è il numero di vertici di G, G' è costruito aggiungendo $n + 1$ vertici isolati a G. Un vertice è isolato se non ci sono archi che lo collegano ad altri vertici.

Dimostriamo che la riduzione è corretta:

- Se $\langle G\rangle \in$ 3-COLOR, allora esiste un modo per colorare G con tre colori $1, 2, 3$. Sia n il numero di vertici di G. Posso costruire una colorazione sbilanciata per il grafo G' come segue:
  - i colori dei vertici di G' che appartengono anche a G sono colorati come in G;
  - i vertici isolati sono colorati tutti con il colore 1.

  In questo modo il colore 1 è assegnato ad almeno metà dei vertici di G' e la colorazione è sbilanciata.
- Se $\langle G'\rangle \in$ UNBALANCED-3-COLOR, allora esiste una colorazione sbilanciata di G'. Se elimino da G' i vertici isolati aggiunti dalla riduzione ottengo una 3-colorazione di G.

La funzione di riduzione deve contare i vertici del grafo G e aggiungere $n + 1$ vertici al grafo, operazioni che si possono fare in tempo polinomiale.