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

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

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

dove L, R indicano gli spostamenti a sinistra e a destra della testina, e I indica l'inserimento di una nuova cella nella posizione corrente della testina. Dopo una operazione di inserimento, la cella inserita contiene il simbolo blank, mentre la cella che si trovava sotto la testina si trova immediatamente a destra della nuova cella.

Dimostra che qualsiasi macchina di Turing con inserimento 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.

$S =$ "Su input w:

1. Inizialmente S mette il suo nastro in un formato che gli consente di implementare l'operazione di inserimento di una cella, segnando con il simbolo speciale # la fine della porzione di nastro usata dalla macchina. Se w è l'input della TM, la configurazione iniziale del nastro è $w\#$.
2. 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.
3. 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 il # che marca la fine del nastro, S scrive un blank al posto del #, e scrive un # nella cella immediatamente più a destra. La simulazione continua con la testina in corrispondenza del blank.
4. Per simulare una mossa del tipo $\delta(q, a) = (r, b, I)$ la TM S scrive un blank marcato nella cella corrente e sposta il contenuto del nastro, dalla cella corrente fino al # di fine nastro, di una cella più a destra. Quindi riporta la testina in corrispondenza del blank marcato, toglie la marcatura e scrive b nella cella immediatamente più a destra. La simulazione continua con la testina in corrispondenza della cella inserita.
5. 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

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

Dimostra che F è indecidibile.

Soluzione. La seguente macchina G calcola una riduzione $\overline{A_{TM}} \leq_m F$:

$G =$ "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 G calcola una funzione di riduzione g da $\overline{A_{TM}}$ a F, cioè una funzione tale che

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

- Se $\langle M, w\rangle \in \overline{A_{TM}}$ 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 $\text{HALTS}(M) = \emptyset$ che è un insieme finito. Di conseguenza $M' \in F$.
- Viceversa, se $\langle M, w\rangle \notin \overline{A_{TM}}$ allora la macchina M accetta w. In questo caso la macchina M' accetta tutte le parole, quindi $\text{HALTS}(M) = \Sigma^*$ che è un insieme infinito. Di conseguenza $M' \notin F$.

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 *bilanciata* se ogni colore è associato ad esattamente $1/3$ dei vertici del grafo.

Balanced-3-Color è il problema di trovare una 3-colorazione bilanciata:

$$\text{Balanced-3-Color} = \{\langle G\rangle \mid G \text{ è un grafo che ammette una 3-colorazione bilanciata}\}$$

(a) Dimostra che Balanced-3-Color è un problema NP
(b) Dimostra che Balanced-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 Balanced-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 ogni colore compare esettamente in $1/3$ 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 Balanced-3-Color è NP-Hard per riduzione polinomiale da 3-Color a Balanced-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 $2n$ 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, e assumiamo che questa colorazione associ k_1 vertici al colore 1, k_2 vertici al colore 2 e k_3 vertici al colore 3. Posso costruire una colorazione bilanciata per il grafo G' come segue:
  - i colori dei vertici di G' che appartengono anche a G sono colorati come in G;
  - $n - k_1$ vertici isolati sono colorati con il colore 1;
  - $n - k_2$ vertici isolati sono colorati con il colore 2;
  - $n - k_3$ vertici isolati sono colorati con il colore 3.

  In questo modo ognuno dei tre colori compare in esattamente n vertici e la colorazione di G' è bilanciata.
- Se $\langle G'\rangle \in$ Balanced-3-Color, allora esiste una colorazione bilanciata 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 $2n$ vertici al grafo, operazioni che si possono fare in tempo polinomiale.