Automi e Linguaggi Formali – 15/6/2023
Seconda prova intermedia – Primo Turno

1. (12 punti) Una macchina di Turing "ecologica" (ETM) è uguale a una normale macchina di Turing deterministica a singolo nastro, ma può leggere e scrivere su entrambi i lati di ogni cella del nastro: fronte e retro. La testina di una TM ecologica può spostarsi a sinistra (L), a destra (R) o passare all'altro lato del nastro (F).

(a) Dai una definizione formale della funzione di transizione di una TM ecologica.

(b) Dimostra che le TM ecologiche riconoscono la classe dei linguaggi Turing-riconoscibili. Usa una descrizione a livello implementativo per definire le macchine di Turing.

Soluzione.

(a) $\delta : Q \times \Gamma \mapsto Q \times \Gamma \times \{L, R, F\}$

(b) Per dimostrare che TM ecologiche riconoscono la classe dei linguaggi Turing-riconoscibili dobbiamo dimostrare due cose: che ogni linguaggio Turing-riconoscibile è riconosciuto da una ETM, e che ogni linguaggio riconosciuto da una ETM è Turing-riconoscibile.

La prima dimostrazione è banale: le TM deterministiche a singolo nastro sono un caso particolare di ETM che usano solamente il lato di fronte del nastro e non effettuano mai la mossa F per passare dall'altro lato del nastro. Di conseguenza, ogni linguaggio Turing-riconoscibile è riconosciuto da una ETM.

Per dimostrare che ogni linguaggio riconosciuto da una ETM è Turing-riconoscibile, mostriamo come convertire una macchina di Turing ecologica M in una TM deterministica a due nastri S equivalente. Il primo nastro di S rappresenta il lato di fronte del nastro di M, il secondo nastro rappresenta il lato di retro.

$S =$ "Su input w:

1. S usa lo stato per memorizzare lo stato di M ed il lato dove si trova la testina di M. All'inizio il lato corrente è "fronte" e lo stato di M è quello iniziale.
2. Se il lato corrente è "fronte": leggi il simbolo sotto la testina del primo nastro per stabilire la mossa da fare. Se il lato corrente è "retro", leggi il simbolo sotto la testina del secondo nastro per stabilire la mossa da fare.
3. La simulazione delle mosse del tipo $\delta(q, a) = (r, b, L)$ scrive b sul primo nastro se il lato corrente è "fronte", e scrive b sul secondo nastro se il nastro corrente è "retro". Poi sposta entrambe le testine di una cella a sinistra.
4. La simulazione delle mosse del tipo $\delta(q, a) = (r, b, R)$ scrive b sul primo nastro se il lato corrente è "fronte", e scrive b sul secondo nastro se il nastro corrente è "retro". Poi sposta entrambe le testine di una cella a destra.
5. Per simulare una mossa del tipo $\delta(q, a) = (r, b, F)$ la TM S scrive b sul primo nastro se il lato corrente è "fronte", poi cambia il valore del lato corrente in "retro". Se il lato corrente è "retro", scrive b sul secondo nastro, poi cambia il valore del lato corrente in "fronte". Sposta entrambe le testine di una cella a destra e poi una cella a sinistra, in modo da ritornare nella cella di partenza.
6. r diventa il nuovo stato corrente della simulazione. Se r è lo stato di accettazione di M, allora S termina con accettazione. Se r è lo stato di rifiuto di M, allora S termina con rifiuto. Negli altri casi continua la simulazione dal punto 2."

Per concludere, siccome sappiamo che le TM multinastro riconoscono i linguaggi Turing-riconoscibili, allora abbiamo dimostrato che ogni linguaggio riconosciuto da una ETM è Turing-riconoscibile.

2. (12 punti) Considera il seguente problema: dato un DFA D e un'espressione regolare R, il linguaggio riconosciuto da D è uguale al linguaggio generato da R?

(a) Formula questo problema come un linguaggio $EQ_{\text{DFA,REX}}$.

(b) Dimostra che $EQ_{\text{DFA,REX}}$ è decidibile.

Soluzione.

(a) $EQ_{\text{DFA,REX}} = \{\langle D, R\rangle \mid D$ è un DFA, R è una espressione regolare e $L(D) = L(R)\}$

(b) La seguente macchina N usa la Turing machine M che decide EQ_{DFA} per decidere $EQ_{\text{DFA,REX}}$:

$N =$ "su input $\langle D, R\rangle$, dove D è un DFA e R una espressione regolare:
1. Converti R in un DFA equivalente D_R
2. Esegui M su input $\langle D, D_R\rangle$, e ritorna lo stesso risultato di M."

Mostriamo che N è un decisore dimostrando che termina sempre e che ritorna il risultato corretto. Sappiamo che esiste un algoritmo per convertire ogni espressione regolare in un ε-NFA, ed un algoritmo per convertire ogni ε-NFA in un DFA. Il primo step di N si implementa eseguendo i due algoritmi in sequenza, e termina sempre perché entrambi gli algoritmi di conversione terminano. Il secondo step termina sempre perché sappiamo che EQ_{DFA} è un linguaggio decidibile. Quindi N termina sempre la computazione.

Vediamo ora che N dà la risposta corretta:

- Se $\langle D, R\rangle \in EQ_{\text{DFA,REX}}$ allora $L(D) = L(R)$, e di conseguenza $L(D) = L(D_R)$ perché D_R è un DFA equivalente ad R. Quindi $\langle D, D_R\rangle \in EQ_{\text{DFA}}$, e l'esecuzione di M terminerà con accettazione. N ritorna lo stesso risultato di M, quindi accetta.
- Viceversa, se $\langle D, R\rangle \notin EQ_{\text{DFA,REX}}$ allora $L(D) \neq L(R)$, e di conseguenza $L(D) \neq L(D_R)$ perché D_R è un DFA equivalente ad R. Quindi $\langle D, D_R\rangle \notin EQ_{\text{DFA}}$, e l'esecuzione di M terminerà con rifiuto. N ritorna lo stesso risultato di M, quindi rifiuta.

3. (12 point) Una macchina di Turing M accetta una stringa unaria se esiste una stringa $x \in \{1\}^*$ tale che M accetta x. Consideria il problema di determinare se una TM M accetta una stringa unaria.

(a) Formula questo problema come un linguaggio UA.

(b) Dimostra che il linguaggio UA è indecidibile.

Soluzione.

(a) $UA = \{\langle M\rangle \mid$ esiste $x \in \{1\}^*$ tale che M accetta $x\}$

(b) La seguente macchina F calcola una riduzione $A_{TM} \leq_m UA$:

$F =$ "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, rifiuta."
2. Ritorna $\langle M'\rangle$."

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

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

- 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 esiste almeno una parola unaria accettata da M', e di conseguenza $\langle M'\rangle \in UA$.
- Viceversa, se $\langle M, w\rangle \notin A_{TM}$ allora la macchina M su input w rifiuta o va in loop. Di conseguenza, la macchina M' rifiuta o va in loop su tutte le stringhe, quindi M' ha linguaggio vuoto e non esiste una stringa unaria che sia accettata da M'. Di conseguenza $\langle M'\rangle \notin UA$.