Automi e Linguaggi Formali
Università degli Studi di Padova
Maggio 2024

Homework 6 - Macchine di Turing (TMs), varianti ed esempi

Gabriel Rovesti

1. Per ciascuno dei seguenti linguaggi, si fornisca una descrizione a livello implementativo della TM in grado di accettare e decidere il linguaggio proposto:

   - $\mathrm{L}_1 = \{a^n b^m c^n \mid n, m \geq 1\}$
   - Data una TM $M_1 =$ "Sull'input w:
     (a) Se $w = \epsilon$, rifiuta
     (b) Scansiona fino al primo simbolo 'b'. Se non viene trovato alcun 'b', rifiuta.
     (c) Ripeti i seguenti passi:
       - Scansiona a sinistra e conta il numero di $'a'$ consecutive prima del primo $'b'$. Memorizza questo valore come n.
       - Scansiona a destra e conta il numero di $'b'$ consecutive dopo l'ultimo $'a'$. Memorizza questo valore come m.
       - Scansiona oltre i $'b'$ fino al primo $'c'$. Se non viene trovato alcun $'c'$, rifiuta.
       - Scansiona e conta il numero di $'c'$ consecutive. Memorizza questo valore come p.
       - Se $n = p$, accetta. Altrimenti, rifiuta.
     (d) Se la scansione termina senza accettare/rifiutare, rifiuta.
   - $L_1 = \{w \mid w$ contiene un numero uguale di 0 e di 1$\}$
     Sulla stringa di input w:
     (a) Eseguire la scansione del nastro e contrassegnare il primo 0 che non è stato contrassegnato. Se non viene trovato nessuno 0 non contrassegnato, vai alla fase 4. Altrimenti, sposta il file torna alla parte anteriore del nastro
     (b) Eseguire la scansione del nastro e contrassegnare il primo 1 che non è stato contrassegnato. Se non è stato trovato alcun 1 non contrassegnato, rifiutare

(c) Riporta la testa davanti al nastro e vai alla fase 1

(d) Spostare la testa indietro verso la parte anteriore del nastro. Scansiona il nastro per vedere se rimangono degli 1 non contrassegnati. Se non ne viene trovato nessuno, accetta; altrimenti, rifiuta

- $L_1 = \{w \mid w$ contiene un numero di 0 doppio rispetto al numero degli 1$\}$

  Data una TM M_1 = "Sull'input w:

  (a) Se $w = \epsilon$, accetta

  (b) Ripeti i seguenti passi:

  - Scansiona fino al successivo 1
    * Se ce n'è uno, cancellalo
    * Se non viene trovato alcun 1, scansionare il nastro cercando il simbolo 0. Se viene trovato il simbolo 0, rifiuta. Altrimenti (trovato uno 0), accetta.
  - Eseguire la scansione a destra e a sinistra finché non viene trovato il simbolo 0. Se non viene trovato nessuno 0, rifiuta. Se ce n'è uno, barrare se disattivato.

- $L_1 = \{w \mid w$ non contiene il doppio degli 0 rispetto al numero degli 1$\}$

  Data una TM M_2 = "Sull'input w:

  (a) Avviare la macchina M_1 sull'input w. Rispondere al contrario, cioè se M1 accetta, rifiuta e se M_1 rifiuta, accetta."

  (Si noti che la macchina M2 è la macchina che risolve il problema precedente)

2. Una *Turing machine con un nastro doppiamente infinito* è simile ad una Turing Machine ordinaria, ma il nastro è infinito a sinistra ma anche a destra. Il nastro è inizialmente riempito di blank, eccetto che per la porzione che contiene l'input. La computazione è definita come al solito, eccetto che la testina non incontra mai la fine del nastro finché si muove a sinistra. Dimostra che questo tipo di Turing machine riconosce la classe dei linguaggi Turing-riconoscibili.

   <u>Soluzione</u>

   Una TM con nastro doppiamente infinito può simulare una TM ordinaria. Essa marca l'estremità sinistra dell'ingresso per rilevare e impedire che la testina si sposti da quell'estremità. di sinistra dell'ingresso per rilevare e impedire che la testina si sposti da quell'estremità. Per simulare la TM con nastro doppiamente infinito con una TM ordinaria, mostriamo mostriamo come simularla con una TM a due nastri, già dimostrato essere equivalente in potenza a un TM ordinario.

Il primo nastro della TM a due nastri viene scritto con la stringa di input e il secondo nastro è vuoto (blank). Tagliamo il nastro del TM a doppio nastro infinito in due parti, in corrispondenza della cella iniziale della stringa di input. La parte con la stringa di input e tutti gli spazi vuoti alla sua destra appare sul primo nastro della TM a due nastri. La parte a sinistra della stringa di input appare sul secondo nastro, in ordine inverso.

3. Una *Turing machine con nastro di sola lettura* è una TM che, oltre al nastro di lavoro standard, ha un nastro aggiuntivo di sola lettura che contiene l'input e sul quale la testina può solo leggere, senza scrivere. Dimostra che questo tipo di Turing machine riconosce la classe dei linguaggi Turing-riconoscibili.

   Soluzione

   Per dimostrare che una Turing machine con nastro di sola lettura riconosce la classe dei linguaggi Turing-riconoscibili, dobbiamo mostrare le seguenti:

   (a) Qualsiasi linguaggio riconoscibile da una Turing machine con nastro singolo può essere riconosciuto da una Turing machine con nastro di sola lettura
   (b) Qualsiasi linguaggio riconoscibile da una Turing machine con nastro di sola lettura può essere riconosciuto da una Turing machine con nastro singolo

   Partiamo dal primo punto:

   Dato L un linguaggio riconoscibile da una Turing machine M con nastro singolo, possiamo costruire una TM con nastro di sola lettura M' che riconosce L nel seguente modo:

   (a) M' inizializza il suo nastro di lavoro a vuoto
   (b) M' copia l'input dal nastro di sola lettura al nastro di lavoro
   (c) M' simula il comportamento di M sul nastro di lavoro
   (d) M' accetta se e solo se M accetta

Poiché M riconosce L, e M'simula esattamente il comportamento di M,M' riconosce L.

Per l'implicazione opposta, sia L un linguaggio riconoscibile da una Turing machine con nastro di sola lettura M'. Possiamo costruire una Turing machine con nastro singolo M che riconosce L nel seguente modo:

(a) M inizializza il suo nastro con l'input

(b) M simula il comportamento di M' sul suo nastro, utilizzando l'input come nastro di sola lettura, eseguendo i seguenti passi:

- Per simulare una transizione di M' che legge dal nastro di sola lettura,M legge dal suo nastro nella posizione corrispondente all'input
- Per simulare una transizione di M' che scrive sul nastro di lavoro, M scrive sul suo nastro
- Per simulare gli spostamenti della testina di M' sul nastro di lavoro, M sposta la sua testina di conseguenza sul suo nastr

(c) M accetta se e solo se M' accetta

Poiché M' riconosce L, e M simula esattamente il comportamento di M', M riconosce L.

Quindi, una Turing machine con nastro di sola lettura riconosce esattamente la classe dei linguaggi Turing-riconoscibili.

4. Una *macchina di Turing a singola scrittura* è una TM a nastro singolo che può modificare ogni cella del nastro al più una volta (inclusa la parte di input del nastro). Mostrare che questa variante dimacchina di Turing è equivalente alla macchina di Turing standard.

   Soluzione

   Mostriamo innanzitutto come sia possibile utilizzare una TM a doppia scrittura per simulare una TM ordinaria, e poi costruire una TM a singola scrittura (detta anche write-once) a partire dalla TM a doppia scrittura (detta anche write-twice).

   La TM write-twice simula una fase della macchina originale copiando l'intero nastro su una nuova porzione del nastro a destra di quella utilizzata per l'input. La procedura di copiatura segna ogni carattere mentre viene copiato, quindi questa procedura altera ogni casella del nastro due volte, una volta per scrivere il carattere per la prima volta e un'altra volta per segnare che è stato copiato, cosa che avviene nel passaggio successivo quando il nastro viene ricopiato.

Quando si copiano le celle in corrispondenza o adiacenti alla posizione contrassegnata, il contenuto del nastro viene aggiornato secondo le regole della TM originale, il che consente a questa procedura di copia di simulare un passaggio di una TM ordinaria. (Piccolo dettaglio tecnico: è necessario conoscere anche la posizione della testina del nastro della TM originale sul corrispondente simbolo copiato).

Per effettuare la simulazione con una macchina write-once, si opera esattamente come prima, tranne che per il fatto che ogni cella del nastro precedente è ora rappresentata da due celle. La prima contiene il simbolo del nastro della macchina originale, mentre la seconda contiene il segno utilizzato nella procedura di copiatura. L'input non viene presentato alla macchina nel formato con due celle per simbolo, quindi la prima volta che il nastro viene copiato, i segni di copiatura vengono posti direttamente sopra il simbolo di input.

5. Una macchina di Turing con "ferma" invece di sinistra è simile a una macchina di Turing ordinaria, ma la funzione di transizione ha la forma: $\delta : Q \times \Gamma \leftarrow Q \times \Gamma \times \{R, S\}$. In ogni punto, la macchina può spostare la testa a destra o lasciarla nella stessa posizione. Dimostrare che questa variante della macchina di Turing **non** è equivalente alla versione usuale.Quale classe di linguaggi riconoscono queste macchine?

   Soluzione

   Per dimostrare che una macchina di Turing con "ferma" non è potente come (e quindi non è equivalente a) la versione usuale della classica TM, dobbiamo dimostrare che esiste un linguaggio che può essere riconosciuto da una macchina di Turing standard ma non può essere riconosciuto da una macchina di Turing con stay put invece di left. Uno di questi linguaggi è il linguaggio context-free $\{a^n b^n \mid n \geq 0\}$.

   Una macchina di Turing con "ferma" non può tornare indietro a leggere ciò che ha scritto in precedenza; di conseguenza, non può tenere traccia di quante a ha letto per abbinarle alle b. In effetti, questo tipo di macchina di Turing non è più potente di un DFA e può riconoscere solo la classe dei linguaggi regolari.