Automi e Linguaggi Formali
Università degli Studi di Padova
Maggio 2024

Homework 8 - Indecidibilità e riducibilità

Gabriel Rovesti

1. Considera il seguente problema: data una TM M a nastro semi-infinito, determinare se esiste un input w su cui M sposta la testina a sinistra partendo dalla cella numero 2023 (ossia se in qualche momento durante la computazione la testina si muove dalla cella 2023 alla cella 2022).

   (a) Formula questo problema come un linguaggio 2023_{TM}.

   (b) Dimostra che il linguaggio 2023_{TM} è indecidibile.

   Soluzione

   $2023_{TM} = \{\langle M \rangle \mid M$ è una TM a nastro semi-infinito che, per qualche input w, sposta la testina dalla cella 2023 alla cella 2022 durante la computazione$\}$

   Per dimostrare che 2023_{TM} è indecidibile, utilizzeremo una riduzione da $HALT_{TM}$ che sappiamo essere indecidibile.

   Definiamo la seguente funzione F: $F(\langle M, w \rangle) = \langle M' \rangle$ dove M' è una TM definita come segue:

   M' = "Su input x:

   (a) Scrivi w nelle prime $|w|$ celle del nastro.

   (b) Sposta la testina alla cella 2023.

   (c) Simula M su input w a partire dalla cella 2023.

   (d) Se M termina, sposta la testina a sinistra di una cella (dalla cella 2023 alla cella 2022).

   (e) Termina."

   Mostriamo che F è una riduzione da $HALT_{TM}$ a 2023_{TM}:

- Se $\langle M, w\rangle \in HALT_{TM}$, allora M termina su input w. Quindi M', su qualsiasi input x, scrive w sul nastro, sposta la testina alla cella 2023, simula M su w fino al termine, e poi sposta la testina a sinistra di una cella, dalla cella 2023 alla cella 2022. Quindi $\langle M'\rangle \in 2023_{TM}$.
- Se $\langle M, w\rangle \notin HALT_{TM}$, allora M non termina su input w. Quindi M', su qualsiasi input x, scrive w sul nastro, sposta la testina alla cella 2023, e poi entra in un loop infinito simulando M su w, senza mai spostare la testina dalla cella 2023 alla cella 2022. Quindi $\langle M'\rangle \notin 2023_{TM}$.

Quindi $\langle M, w\rangle \in HALT_{TM}$ sse $\langle M'\rangle \in 2023_{TM}$, ovvero F riduce $HALT_{TM}$ a 2023_{TM}. Poiché $HALT_{TM}$ è indecidibile, per riduzione anche 2023_{TM} è indecidibile.

2. Considera il problema di determinare se un PDA accetta qualche stringa nella forma $\{ww \mid w \in \{0,1\}^*\}$. Dimostra che questo problema è indecidibile.

   <u>Soluzione</u>

   Per dimostrare l'indecidibilità del problema, utilizzeremo una riduzione dal problema E_{PDA} che sappiamo essere indecidibile. Sia F la seguente funzione: $F(\langle P\rangle) = \langle P'\rangle$ dove P' è un PDA definito come segue: $P' =$ "Su input x:

   (a) Simula il PDA P su input x.
   (b) Se P accetta, controlla se x è nella forma ww con $w \in 0, 1^*$.
      - Se sì, accetta.
      - Altrimenti, rifiuta.
   (c) Se P rifiuta, rifiuta."

   Mostriamo che F è una riduzione da E_{PDA} al problema di determinare se un PDA accetta qualche stringa nella forma $\{ww \mid w \in 0, 1^*\}$: $\langle P\rangle \in E_{PDA}$ sse $L(P) = \emptyset$ sse $\forall x, P$ rifiuta x sse $\forall x, P'$ rifiuta x (perché rifiuta sempre al passo 3) sse P' non accetta alcuna stringa, incluse quelle nella forma ww sse $\langle P'\rangle$ non è un PDA che accetta qualche stringa nella forma ww.

   Al contrario: $\langle P\rangle \notin E_{PDA}$ sse $L(P) \neq \emptyset$ sse $\exists x$ tale che P accetta x sse $\exists x$ tale che P' accetta x se $x = ww$ e rifiuta altrimenti sse $\langle P'\rangle$ è un PDA che accetta qualche stringa nella forma ww (ma potrebbe anche accettare altre stringhe).

Quindi abbiamo dimostrato che: $\langle P \rangle \in \mathrm{E}_{PDA}$ sse $\langle P' \rangle$ non è un PDA che accetta qualche stringa nella forma ww, ovvero E_{PDA} si riduce al complemento del nostro problema. Poiché E_{PDA} è indecidibile, per riduzione anche il nostro problema è indecidibile.

3. Una variabile A in una CFG G *necessaria* se appare in tutte le derivazioni di qualche stringa $w \in G$. Sia $NECESSARY_{CFG} = \{\langle G, A \rangle \mid A$ è una variabile necessaria in $G\}$.

   (a) Si mostri $NECESSARY_{CFG}$ è Turing-riconoscibile

   (b) Si mostri che $NECESSARY_{CFG}$ è indecidibile

   Soluzione

   Per dimostrare che $NECESSARY_{CFG}$è Turing-riconoscibile, costruiamo una TM M che accetta $NECESSARY_{CFG}$:

   $M =$ "Su input $\langle G, A \rangle$:

   (a) Verifica che A sia una variabile di G. Se non lo è, rifiuta.

   (b) Enumera tutte le derivazioni di G. Per ogni derivazione:
      - Se la derivazione produce una stringa $w \in \Sigma^*$ (cioè composta solo da simboli terminali) e A appare nella derivazione, accetta.
      - Se la derivazione produce una stringa $w \in \Sigma^*$ e A non appare nella derivazione, rifiuta.

   (c) Se tutte le derivazioni sono state enumerate senza raggiungere una conclusione, vai al passo 2.

   Se $\langle G, A \rangle \in NECESSARY_{CFG}$, allora A apparirà in tutte le derivazioni di qualche stringa w, quindi M troverà una di queste derivazioni e accetterà. Se $\langle G, A \rangle \in NECESSARY_{CFG}$, allora o A non apparirà in nessuna derivazione di stringhe (e quindi M non accetterà mai), oppure per ogni stringa esistono alcune derivazioni in cui A non appare (e quindi M troverà una di queste derivazioni e rifiuterà).

   Quindi $L(M) = NECESSARY_{CFG}$, ovvero è Turing-riconoscibile.

   Per dimostrare che $NECESSARY_{CFG}$ è indecidibile, riduciamo da E_{CFG} che sappiamo essere indecidibile.

   Sia F la seguente funzione: $F(\langle G \rangle) = \langle G', S' \rangle$ dove G' è una CFG definita come segue:

- Tutte le variabili e le produzioni di G sono incluse in G'
- Viene aggiunta una nuova variabile iniziale S' con la produzione $S' \to S$, dove S è la variabile iniziale di G.

Mostriamo che F è una riduzione da E_{CFG} a $NECESSARY_{CFG}$:

- Se $\langle G \rangle \in E_{CFG}$, allora $L(G) = \emptyset$. Quindi $L(G') = \emptyset$ e S' non appare in nessuna derivazione di stringhe. Quindi $\langle G', S' \rangle \notin$.
- Se $\langle G \rangle \notin E_{CFG}$, allora $L(G) \neq \emptyset$. Quindi esiste qualche $w \in L(G')$ e tutte le derivazioni di w in G' devono iniziare con $S' \to S$. Quindi S' appare in tutte le derivazioni di w e $\langle G', S' \rangle \in NECESSARY_{CFG}$.

4. Una CFG è *minimale* se nessuna delle regole può essere rimossa senza cambiare il linguaggio generato. Sia $MIN_{CFG} = \{\langle G, A \rangle \mid A$ è una variabile necessaria in $G\}$.

   (a) Si mostri MIN_{CFG} è Turing-riconoscibile

   (b) Si mostri che MIN_{CFG} è indecidibile

   <u>Soluzione</u>

   Per dimostrare che MIN_{CFG} è Turing-riconoscibile, costruiamo una TM M che accetta MIN_{CFG}:

   $M =$ "Su input $\langle G \rangle$:

   (a) Per ogni produzione p in G:
      i. Costruisci una nuova CFG G' rimuovendo la produzione p da G.
      ii. Enumera tutte le possibili derivazioni in G e G' in parallelo:
         - Se trovi una stringa w che può essere derivata in G ma non in G', continua con la prossima produzione (vai al passo 1).
         - Se trovi una stringa w che può essere derivata sia in G che in G', rifiuta.

   (b) Se tutte le produzioni sono state verificate senza rifiutare, accetta.

   - Se $\langle G \rangle \in MIN_{CFG}$, allora per ogni produzione p, rimuovere p cambierà il linguaggio generato. Quindi per ogni p, M troverà una stringa che può essere derivata in G ma non in G' e alla fine accetterà.

- Se $\langle G\rangle \notin MIN_{CFG}$, allora esiste una produzione p che può essere rimossa senza cambiare il linguaggio. Quindi M troverà una stringa che può essere derivata sia in G che in G' e rifiuterà.

Quindi $L(M) = MIN_{CFG}$, ovvero MIN_{CFG} è Turing-riconoscibile.

Per dimostrare che MIN_{CFG} è indecidibile, riduciamo da che sappiamo essere indecidibile.

Sia F la seguente funzione: $F(\langle G\rangle) = \langle G'\rangle$ dove G' è una CFG definita come segue:

- Tutte le variabili e le produzioni di G sono incluse in G'
- Per ogni produzione $A \to \alpha$ in G, aggiungiamo una nuova variabile A' e le produzioni $A \to A'$ e $A' \to \alpha$ in G'.

Mostriamo che F è una riduzione da ALL_{CFG} a MIN_{CFG}:

- Se $\langle G\rangle \in ALL_{CFG}$, allora $L(G) = \Sigma^*$. Quindi anche $L(G') = \Sigma^*$, ma G' non è minimale perché rimuovendo qualsiasi coppia di produzioni $A \to A'$ e $A' \to \alpha$ si ottiene ancora una grammatica che genera Σ^*. Quindi $\langle G'\rangle \notin MIN_{CFG}$.
- Se $\langle G\rangle \notin ALL_{CFG}$, allora $L(G) \neq \Sigma^*$. Quindi esistono stringhe in Σ^* che non possono essere generate da G. Queste stringhe non possono essere generate nemmeno da G', e rimuovere qualsiasi produzione da G' può solo ridurre ulteriormente il linguaggio generato. Quindi G' è minimale e $\langle G'\rangle \in MIN_{CFG}$.

Quindi $\langle G\rangle \in ALLCFG$ sse $\langle G'\rangle \notin MIN_{CFG}$, ovvero F riduce ALL_{CFG} al complemento di MIN_{CFG}. Poiché ALL_{CFG} è indecidibile, per riduzione anche MIN_{CFG} è indecidibile.

5. Considera il linguaggio $FORTY-TWO = \{\langle M, w\ |M$ termina la computazione su w avendo solo 42 sul nastro $\}$. Dimostra che $FORTY-TWO$ è indecidibile.

   Soluzione

   Per dimostrare che $FORTY - TWO$ è indecidibile, utilizzeremo una riduzione da che sappiamo essere indecidibile. Definiamo la seguente funzione F: $F(\langle M, w\rangle) = \langle M', w\rangle$ dove M' è una TM definita come segue: $M' =$ "Su input x:

   (a) Simula M su input w.

(b) Se M termina, cancella tutto il contenuto del nastro e scrivi '42'.

(c) Accetta."

Mostriamo che F è una riduzione da $HALT_{TM}$ a $FORTY - TWO$

- Se $\langle M, w\rangle \in$, allora M termina su input w. Quindi M' simulerà M fino al termine, poi cancellerà il contenuto del nastro, scriverà '42' e accetterà. Quindi $\langle M', w\rangle \in FORTY - TWO$.
- Se $\langle M, w\rangle \notin$, allora M non termina su input w. Quindi M' entrerà in un loop infinito simulando M e non scriverà mai '42' sul nastro. Quindi $\langle M', w\rangle \notin FORTY - TWO$.

Quindi $\langle M, w\rangle \in HALT_{TM}$ sse $\langle M', w\rangle \in FORTY - TWO$, ovvero F riduce $HALT_{TM}$ a $FORTY - TWO$. Poiché $HALT_{TM}$ è indecidibile, per riduzione anche $FORTY - TWO$ è indecidibile.

6. Una Turing Machine moltiplica correttamente se, dati in input due numeri binari separati da #, termina la computazione con la loro moltiplicazione (in binario) sul nastro. (Non importa cosa fa sugli altri input.)

   Considera il problema di determinare se una TM moltiplica correttamente

   (a) Formula questo problema come un linguaggio MUL_{TM}.

   (b) Dimostra che il linguaggio MUL_{TM} è indecidibile.

   Soluzione

   1. $MUL_{TM} = \{\langle M\rangle \mid M$ è una TM che, su input $w = x\#y$ con $x, y \in \{0,1\}^*$, termina la computazione con z sul nastro, dove z è la rappresentazione binaria del prodotto tra i numeri rappresentati in binario da x e $y\}$

   2. Per dimostrare che MUL_{TM} è indecidibile, utilizzeremo una riduzione da $HALT_{TM}$ che sappiamo essere indecidibile.

   Definiamo la seguente funzione F: $F(\langle M, w\rangle) = \langle M'\rangle$ dove M' è una TM definita come segue:

   $M' =$ "Su input $x\#y$:

   (a) Simula M su input w.

   (b) Se M termina, calcola il prodotto tra i numeri rappresentati in binario da x e y e scrivi il risultato z sul nastro.

(c) Termina."

Mostriamo che F è una riduzione da $HALT_{TM}$ a MUL_{TM}:

Se $\langle M, w\rangle \in HALT_{TM}$, allora M termina su input w. Quindi M', su qualsiasi input della forma $x\#y$, simula M su w fino al termine, poi calcola correttamente il prodotto tra i numeri rappresentati da x e y e scrive il risultato sul nastro. Quindi $\langle M'\rangle \in MUL_{TM}$.

Se $\langle M, w\rangle \notin HALT_{TM}$, allora M non termina su input w. Quindi M', su qualsiasi input della forma $x\#y$, entra in un loop infinito simulando M su w e non calcola mai il prodotto. Quindi $\langle M'\rangle \notin MUL_{TM}$.

Quindi $\langle M, w\rangle \in HALT_{TM}$ sse $\langle M'\rangle \in MUL_{TM}$, ovvero F riduce $HALT_{TM}$ a MUL_{TM}. Poiché $HALT_{TM}$ è indecidibile, per riduzione anche MUL_{TM} è indecidibile.