Tutorato di Automi e Linguaggi Formali

Homework 9: Indecidibilità e Riducibilità

Gabriel Rovesti
Corso di Laurea in Informatica - Università degli Studi di Padova

Tutorato 9 - 19-05-2025

1 Problemi Indecidibili e Diagonalizzazione

Esercizio 1. Il metodo della diagonalizzazione di Cantor è utilizzato per dimostrare l'esistenza di linguaggi non riconoscibili da macchine di Turing.

a) Spiegare formalmente perché l'insieme di tutte le macchine di Turing è numerabile. Fornire una funzione di enumerazione che associa ciascuna macchina di Turing a un numero naturale univoco.

b) Dimostrare, utilizzando il metodo della diagonalizzazione, che l'insieme di tutti i linguaggi su un alfabeto Σ è non numerabile. Spiegare chiaramente dove viene applicata la diagonalizzazione nella dimostrazione.

c) Basandosi sui risultati precedenti, spiegare perché deve esistere almeno un linguaggio che non è riconoscibile da alcuna macchina di Turing. Questa è una dimostrazione non costruttiva. Quale linguaggio specifico è stato introdotto nel corso come esempio di linguaggio non Turing-riconoscibile?

Esercizio 2. Consideriamo 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) Formulare questo problema come un linguaggio 2023_{TM}.

b) Dimostrare che il linguaggio 2023_{TM} è indecidibile mediante una riduzione da un problema noto. Specificare chiaramente la funzione di riduzione e verificare che soddisfi le proprietà necessarie.

c) Discutere se 2023_{TM} è Turing-riconoscibile, co-Turing-riconoscibile, o nessuno dei due. Giustificare la risposta.

Esercizio 3. Un linguaggio L viene definito co-Turing-riconoscibile se il suo complemento $\overline{L}$ è Turing-riconoscibile.

a) Dimostrare formalmente il seguente teorema: un linguaggio è decidibile se e solo se è sia Turing-riconoscibile che co-Turing-riconoscibile.

b) Dato che $A_{TM} = \{\langle M, w\rangle \mid M$ è una TM che accetta la stringa $w\}$ è Turing-riconoscibile ma non decidibile, dimostrare che $\overline{A_{TM}}$ non può essere Turing-riconoscibile.

c) Descrivere un linguaggio che non è né Turing-riconoscibile né co-Turing-riconoscibile. Giustificare la risposta.

2 Riducibilità e Dimostrazione di Indecidibilità

Esercizio 4. Consideriamo il concetto di riducibilità mediante funzione:

a) Definire formalmente cosa significa che un linguaggio A è riducibile mediante funzione a un linguaggio B (notazione: $A \leq_m B$). Spiegare il ruolo della funzione di riduzione f e quali proprietà deve soddisfare.

b) Dimostrare che se $A \leq_m B$ e B è decidibile, allora A è decidibile. Spiegare come costruire un decisore per A utilizzando un decisore per B e la funzione di riduzione f.

Esercizio 5. Consideriamo il problema di determinare se un PDA accetta qualche stringa nella forma $\{ww \mid w \in \{0,1\}^*\}$.

a) Formulare questo problema come un linguaggio WW_{PDA}.

b) Dimostrare che il linguaggio WW_{PDA} è indecidibile mediante una riduzione appropriata. Quale problema utilizzereste come punto di partenza e perché?

c) Spiegare perché questo risultato è interessante nel contesto dei linguaggi context-free, dato che è noto che $\{ww \mid w \in \{0,1\}^*\}$ non è un linguaggio context-free.

Esercizio 6. Sia $E_{TM} = \{\langle M\rangle \mid M$ è una TM tale che $L(M) = \emptyset\}$ il problema del vuoto per macchine di Turing.

a) Dimostrare che $A_{TM} \leq_m \overline{E_{TM}}$ (dove $\overline{E_{TM}}$ è il complemento di E_{TM}) fornendo una riduzione mediante funzione esplicita. Spiegare come questa riduzione trasforma un'istanza di A_{TM} in un'istanza di $\overline{E_{TM}}$.

b) Utilizzando la riduzione precedente e il fatto che A_{TM} è indecidibile, dimostrare che E_{TM} è indecidibile.

c) Dimostrare che E_{TM} è co-Turing-riconoscibile ma non Turing-riconoscibile.

3 Analisi di Problemi di Indecidibilità avanzati

Esercizio 7. Una CFG è minimale se nessuna delle regole può essere rimossa senza cambiare il linguaggio generato. Sia $MIN_{CFG} = \{\langle G \rangle \mid G \text{ è una CFG minimale}\}$.

a) Dimostrare che MIN_{CFG} è Turing-riconoscibile. Descrivere una macchina di Turing che riconosce questo linguaggio.

b) Dimostrare che MIN_{CFG} è indecidibile. Suggerimento: utilizzare una riduzione da un problema indecidibile noto.

Esercizio 8. Consideriamo i seguenti problemi per macchine di Turing:

$$ALL_{TM} = \{\langle M \rangle \mid M \text{ è una TM tale che } L(M) = \Sigma^*\}$$
$$FINITE_{TM} = \{\langle M \rangle \mid M \text{ è una TM tale che } L(M) \text{ è un linguaggio finito}\}$$

a) Dimostrare che ALL_{TM} è indecidibile mediante una riduzione da un problema noto (a vostra scelta). Specificare chiaramente la funzione di riduzione e verificare che soddisfi le proprietà necessarie.

b) Dimostrare che $FINITE_{TM}$ è indecidibile utilizzando una riduzione appropriata. Quale problema utilizzate come punto di partenza e perché?

c) Discutere se ALL_{TM} e $FINITE_{TM}$ sono Turing-riconoscibili o co-Turing-riconoscibili, giustificando le risposte.

Esercizio 9. Sia $SELF_{TM} = \{\langle M \rangle \mid M \text{ è una TM che accetta la propria codifica } \langle M \rangle\}$.

a) Dimostrare che $SELF_{TM}$ è indecidibile utilizzando una tecnica di diagonalizzazione. Fornire una dimostrazione dettagliata.

b) Dimostrare che $SELF_{TM}$ è Turing-riconoscibile. Descrivere una macchina di Turing che riconosce questo linguaggio.

c) Descrivere una macchina di Turing U che, per ogni macchina di Turing M, ha la proprietà che U accetta $\langle M \rangle$ se e solo se M non accetta $\langle M \rangle$. Spiegare perché l'esistenza di U porta a una contraddizione.