Riducibilità, Classe P

Tutorato 10: Problemi in P, Complessità e Riducibilità

Automi e Linguaggi Formali

Riassunto delle lezioni del Prof. Davide Bresolin
Corso di Laurea in Informatica - Università degli Studi di Padova

28 Maggio 2025

Contents

1 La Classe P: Problemi Trattabili

1.1 Tempo Polinomiale e Equivalenza dei Modelli

Concetto chiave

Il concetto di **tempo polinomiale** è fondamentale per distinguere tra algoritmi efficienti e non efficienti. Caratteristiche principali:

- Una differenza di tempo **polinomiale** tra TM a nastro singolo e multi-nastro è considerata piccola
- Une differenza di tempo **esponenziale** tra TM deterministiche e non deterministiche è considerata grande
- Tutti i modelli di calcolo deterministici "ragionevoli" sono polinomialmente equivalenti

Definizione

Un modello di calcolo è **ragionevole** se assomiglia molto ai computer reali. Questo include:

- Macchine di Turing deterministiche
- Linguaggi di programmazione concreti (Java, C++, Python)
- Macchine multi-nastro
- Computer con accesso ad array

1.2 Definizione della Classe P

Definizione

P è la classe di linguaggi che sono decidibili in tempo polinomiale da una TM deterministica a singolo nastro:

$$P = \bigcup_k \text{TIME}(n^k)$$

Concetto chiave

Proprietà fondamentali della classe P:

- P è **invariante** per i modelli di calcolo polinomialmente equivalenti ad una TM deterministica
- P corrisponde approssimativamente ai problemi che sono **realisticamente risolvibili** da un computer
- La differenza esponenziale rappresenta la complessità degli approcci "a forza bruta"

1.3 Metodologia per Dimostrare che un Problema è in P

Procedimento di risoluzione

Per dimostrare che un problema/algoritmo è in P:

1. **Descrivi l'algoritmo per fasi numerate**
2. **Dai un limite superiore polinomiale** al numero di fasi che l'algoritmo esegue per un input di lunghezza n
3. **Assicurati che ogni fase** possa essere completata in tempo polinomiale su un modello di calcolo deterministico ragionevole
4. **L'input deve essere codificato** in modo ragionevole

1.4 Esempi di Problemi in P

Raggiungibilità in un Grafo:

$$\text{PATH} = \{\langle G, s, t\rangle \mid G \text{ è un grafo che contiene un cammino da } s \text{ a } t\}$$

Numeri Relativamente Primi:

$$\text{RELPRIME} = \{\langle x, y\rangle \mid 1 \text{ è il massimo comune divisore di } x \text{ e } y\}$$

Teorema

Linguaggi Context-Free in P: Ogni linguaggio context-free è un elemento di P.

- Abbiamo già dimostrato che ogni CFL è decidibile, ma l'algoritmo nella dimostrazione è esponenziale
- La soluzione polinomiale usa la **programmazione dinamica**
- La complessità è $O(n^3)$

2 Riducibilità: Trasformare Problemi

2.1 Concetto Fondamentale di Riduzione

Definizione

Una **riduzione** è un modo per trasformare un problema in un altro problema tale che una soluzione al secondo problema può essere usata per risolvere il primo problema.

Concetto chiave

Principi fondamentali della riducibilità:

- Se A è riducibile a B, e B è decidibile, allora A è decidibile
- Se A è riducibile a B, e A è indecidibile, allora B è indecidibile

2.2 Schema delle Dimostrazioni per Riduzione

Procedimento di risoluzione

Le dimostrazioni per riduzione sono usate per dimostrare che un problema è indecidibile:

1. **Assumi che B sia decidibile**
2. **Riduci A al problema B**
   - Costruisci una TM che usa B per risolvere A
3. **Se A è indecidibile**, allora questa è una contraddizione
4. **L'assunzione è sbagliata** e B è indecidibile

3 Problemi Classici Indecidibili

3.1 Il Problema della Fermata

Definizione

$$\text{HALT}_{TM} = \{\langle M, w\rangle \mid M \text{ è una TM che si ferma su input } w\}$$

Procedimento di risoluzione

Dimostrazione dell'indecidibilità di HALT_{TM} per riduzione da A_{TM}:

1. Assumiamo che esista un deciditore R per HALT_{TM}
2. Costruiamo una TM S che decide A_{TM} usando R:
   - S su input $\langle M, w\rangle$ usa R per verificare se M si ferma su w
   - Se M si ferma, S simula M su w e risponde di conseguenza
   - Se M non si ferma, S rifiuta
3. Ma A_{TM} è indecidibile, quindi abbiamo una contraddizione

3.2 Il Problema del Vuoto

Definizione

$$E_{TM} = \{\langle M\rangle \mid M \text{ è una TM tale che } L(M) = \emptyset\}$$

La dimostrazione dell'indecidibilità di E_{TM} avviene per riduzione da A_{TM}. La costruzione richiede di creare una TM ausiliaria il cui linguaggio è vuoto se e solo se la TM originale non accetta l'input dato.

3.3 Altri Problemi Indecidibili

Regolarità:

$$\text{REGULAR}_{TM} = \{\langle M \rangle \mid M \text{ è una TM tale che } L(M) \text{ è regolare}\}$$

Equivalenza:

$$\text{EQ}_{TM} = \{\langle M_1, M_2 \rangle \mid L(M_1) = L(M_2)\}$$

Suggerimento

Per REGULAR_{TM}, capire come usare il deciditore ipotetico per implementare una soluzione ad A_{TM} è meno ovvio rispetto ai casi precedenti. La dimostrazione richiede una costruzione più sofisticata che sfrutta le proprietà dei linguaggi regolari.

4 Riducibilità mediante Funzione

4.1 Formalizzazione delle Riduzioni

Definizione

Funzione Calcolabile: $f : \Sigma^* \to \Sigma^*$ è una funzione calcolabile se esiste una TM M che su input w, termina la computazione avendo solo $f(w)$ sul nastro.

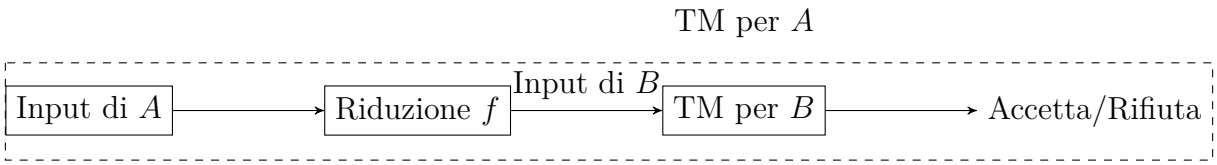
Definizione

Riduzione mediante Funzione: Un linguaggio A è riducibile mediante funzione al linguaggio B ($A \leq_m B$), se esiste una funzione calcolabile $f : \Sigma^* \to \Sigma^*$ tale che:

$$\text{per ogni } w : w \in A \text{ se e solo se } f(w) \in B$$

La funzione f è detta **riduzione** da A a B.

4.2 Schema di Funzionamento delle Riduzioni



4.3 Proprietà delle Riduzioni mediante Funzione

Teorema

Proprietà di Decidibilità:

- Se $A \leq_m B$ e B è **decidibile**, allora A è **decidibile**
- Se $A \leq_m B$ e A è **indecidibile**, allora B è **indecidibile**

Teorema

Proprietà di Riconoscibilità:

- Se $A \leq_m B$ e B è **Turing-riconoscibile**, allora A è **Turing-riconoscibile**
- Se $A \leq_m B$ e A non è **Turing-riconoscibile**, allora B non è **Turing-riconoscibile**

5 Esempi Dettagliati di Riduzioni

5.1 $A_{TM} \leq_m \mathbf{HALT}_{TM}$

Procedimento di risoluzione

Costruzione della riduzione:

1. **Input della riduzione**: $\langle M, w \rangle$
2. **Output della riduzione**: $\langle M', w' \rangle$ dove:
   - M' è una TM che su input w' simula M su w
   - Se M accetta w, allora M' si ferma (accettando)
   - Se M rifiuta w, allora M' si ferma (rifiutando)
   - Se M non si ferma su w, allora M' non si ferma
3. **Proprietà**: M accetta w se e solo se M' si ferma su w'

5.2 $E_{TM} \leq_m \mathbf{EQ}_{TM}$

Procedimento di risoluzione

Costruzione della riduzione:

1. **Input della riduzione**: $\langle M \rangle$
2. **Output della riduzione**: $\langle M_1, M_2 \rangle$ dove:
   - M_1 è una TM che rifiuta ogni input (quindi $L(M_1) = \emptyset$)
   - M_2 è la TM originale M
3. **Proprietà**: $L(M) = \emptyset$ se e solo se $L(M_1) = L(M_2)$

5.3 Una Riduzione Impossibile: $A_{TM} \not\leq_m E_{TM}$

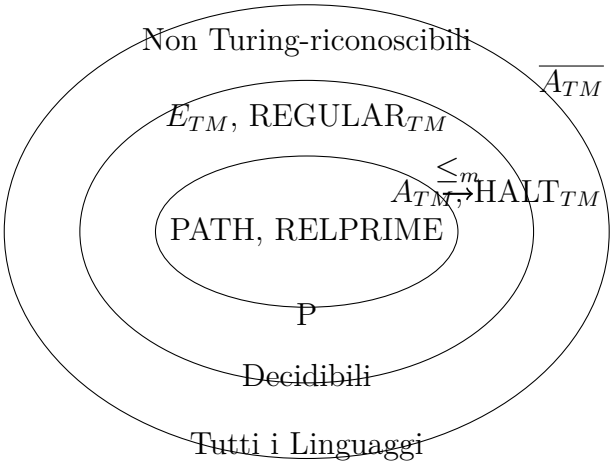
Errore comune

Non tutte le riduzioni sono possibili! Per esempio, non possiamo ridurre A_{TM} a E_{TM} perché:

- Avremmo bisogno che: M accetta w se e solo se $L(M') = \emptyset$
- Ma questo richiederebbe: M accetta w se e solo se $L(M') \neq \emptyset$
- La direzione corretta è: $A_{TM} \leq_m \overline{E_{TM}}$ (il complemento di E_{TM})

6 Gerarchia dei Problemi

6.1 Classificazione per Riconoscibilità



6.2 Il Problema dell'Equivalenza: Un Caso Speciale

Concetto chiave

EQ$_{TM}$ è un problema particolarmente interessante perché:

- Non è né Turing-riconoscibile né co-Turing-riconoscibile
- Questo può essere dimostrato mostrando riduzioni da entrambi A_{TM} e $\overline{A_{TM}}$
- Rappresenta una classe di problemi ancora più difficili dei normali problemi indecidibili

7 Implicazioni Teoriche e Pratiche

7.1 Utilità delle Riduzioni

Suggerimento

Le riduzioni sono strumenti potenti per:

1. **Classificare la difficoltà** dei problemi computazionali
2. **Trasferire algoritmi** da un problema ad un altro
3. **Dimostrare limiti inferiori** sulla complessità computazionale
4. **Identificare famiglie** di problemi con difficoltà equivalente

7.2 Esempi di Funzioni Calcolabili

Concetto chiave

Esempi importanti di funzioni calcolabili:

- **Operazioni aritmetiche** sugli interi (addizione, moltiplicazione, etc.)
- **Trasformazioni di macchine di Turing** (modificare stati, aggiungere transizioni)
- **Manipolazioni di stringhe** (concatenazione, sostituzione)
- **Codifica e decodifica** di strutture dati

8 Esercizi e Approfondimenti

8.1 Problemi Proposti

1. **Linguaggio Universale**: Dimostrare che $\text{ALL}_{TM} = \{\langle M \rangle \mid L(M) = \Sigma^*\}$ è indecidibile usando una riduzione mediante funzione.
2. **TM che non modificano l'input**: Sia $X = \{\langle M, w \rangle \mid M$ è una TM a nastro singolo che non mo
   Dimostrare che X è indecidibile.
3. **Riduzioni e linguaggi regolari**: Se $A \leq_m B$ e B è un linguaggio regolare, ciò implica che A è un linguaggio regolare? Analizzare con controesempi.
4. **Impossibilità di riduzione**: Mostrare che A_{TM} non è riducibile mediante funzione a E_{TM}.
5. **Decidibilità tramite autoriduzione**: Mostrare che se A è Turing-riconoscibile e $A \leq_m \overline{A}$, allora A è decidibile.

8.2 Connessioni con la Teoria della Complessità

9 Conclusioni

La teoria della computabilità e delle riduzioni ci fornisce strumenti essenziali per comprendere i limiti fondamentali della computazione. Attraverso la classe P identifichiamo i problemi efficientemente risolvibili, mentre le riduzioni ci permettono di classificare sistematicamente la difficoltà dei problemi indecidibili.

Concetto chiave

Lezioni fondamentali:

1. Il **tempo polinomiale** è la soglia per l'efficienza computazionale
2. Le **riduzioni** permettono di trasferire difficoltà tra problemi
3. Esistono **gerarchie** di problemi con diversi gradi di indecidibilità
4. La **formalizzazione** mediante funzioni calcolabili rende rigorose le intuizioni

Questi concetti costituiscono la base teorica per comprendere sia i limiti intrinseci della computazione sia le tecniche per affrontare problemi computazionalmente difficili nella pratica.