Macchine di Turing, Varianti ed Esempi

Lezione 7: TM, Varianti ed esempi

Automi e Linguaggi Formali

Gabriel Rovesti

Corso di Laurea in Informatica - Università degli Studi di Padova

Anno Accademico 2024-2025

Contents

1 Introduzione alle Macchine di Turing

1.1 Un Nuovo Modello di Calcolo

Le Macchine di Turing (TM) rappresentano un modello di calcolo più potente rispetto agli automi finiti e agli automi a pila che abbiamo studiato in precedenza.

Concetto chiave

Una macchina di Turing è un modello matematico di calcolo che descrive una macchina astratta che manipola i simboli su una striscia di nastro secondo una tabella di regole. Nonostante la semplicità del modello, è in grado di implementare qualsiasi algoritmo informatico.

Rispetto ai modelli di calcolo precedenti:

- Gli automi finiti sono limitati da una ridotta quantità di memoria
- Gli automi a pila hanno memoria illimitata ma con accesso LIFO (Last-In-First-Out)
- Le macchine di Turing hanno una memoria illimitata e senza restrizioni di accesso

Esistono linguaggi che vanno oltre le capacità degli automi finiti e a pila, e le macchine di Turing sono state progettate per gestire questi linguaggi più complessi.

1.2 Definizione Informale

Una macchina di Turing funziona su un nastro di memoria infinito diviso in celle discrete, ciascuna delle quali può contenere un singolo simbolo tratto da un insieme finito di simboli chiamato alfabeto della macchina.

La macchina è composta da:

- Un nastro infinito come memoria illimitata
- Una testina che legge e scrive simboli sul nastro
- All'inizio il nastro contiene l'input
- La testina si può muovere sia a destra che a sinistra sul nastro
- Stati speciali per l'accettazione e il rifiuto dell'input

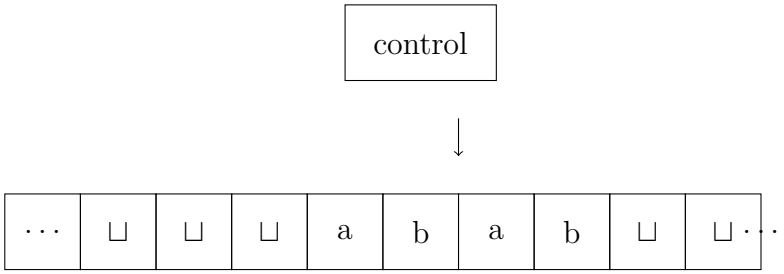

Figure 1: Rappresentazione schematica di una macchina di Turing

2 Definizione Formale delle Macchine di Turing

2.1 Specifica Formale

Una Macchina di Turing (o Turing Machine, TM) è una tupla $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$:

- Q è l'insieme finito di stati
- Σ è l'alfabeto di input che non contiene il simbolo blank $\sqcup$
- Γ è l'alfabeto del nastro che contiene $\sqcup$ e Σ
- $\delta : Q \times \Gamma \mapsto Q \times \Gamma \times \{L, R\}$ è la funzione di transizione
- $q_0 \in Q$ è lo stato iniziale
- $q_{\text{accept}} \in Q$ è lo stato di accettazione
- $q_{\text{reject}} \in Q$ è lo stato di rifiuto (diverso da q_{accept})

2.2 Configurazioni e Computazione

Lo stato corrente, la posizione della testina ed il contenuto del nastro formano la **configurazione** di una TM. Da questa configurazione possiamo determinare la prossima mossa della macchina.

Le configurazioni sono rappresentate da una tripla uqv:

- q è lo stato corrente
- u è il contenuto del nastro prima della testina
- v è il contenuto del nastro dalla testina in poi
- la testina si trova sul primo simbolo di v

Procedimento di risoluzione

La computazione procede secondo la funzione di transizione δ:
Diciamo che una configurazione C_1 produce C_2 se la TM può passare da C_1 a C_2 in un passo.
Se $a, b, c \in \Gamma$, $u, v \in \Gamma^*$ e q_i, q_j sono stati, allora:

- uaq_ibv produce uq_jacv se $\delta(q_i, b) = (q_j, c, L)$
- uaq_ibv produce $uacq_jv$ se $\delta(q_i, b) = (q_j, c, R)$

La configurazione iniziale con input w è q_0w.

2.3 Accettazione e Rifiuto

Una TM M può comportarsi in tre modi diversi quando riceve un input:

- La macchina **accetta**: raggiunge lo stato q_{accept}
- La macchina **rifiuta**: raggiunge lo stato q_{reject}
- La macchina **va in loop**: non si ferma mai, continuando a calcolare indefinitamente

Le configurazioni di accettazione e rifiuto sono configurazioni di arresto, dove la computazione termina.

3 Linguaggi e Decidibilità

3.1 Linguaggi Turing-riconoscibili

Concetto chiave

Una TM M accetta l'input w se esiste una sequenza di configurazioni $C_1, C_2, \ldots, C_k$ tale che:

- C_1 è la configurazione iniziale con input w
- ogni C_i produce C_{i+1}
- C_k è una configurazione di accettazione

Il linguaggio riconosciuto da M è l'insieme delle stringhe accettate da M.

Definizione 1. Un linguaggio è **Turing-riconoscibile** (o anche ricorsivamente enumerabile) se esiste una macchina di Turing che lo riconosce.

3.2 Linguaggi Turing-decidibili

Definizione 2. Un linguaggio è **Turing-decidibile** (o anche ricorsivo) se esiste una macchina di Turing che lo decide, cioè che termina sempre (accettando o rifiutando) per ogni input.

Un **decisore** è una TM che termina sempre la computazione, senza andare in loop. Accetta o rifiuta ogni stringa in input.

Suggerimento

Tutti i linguaggi Turing-decidibili sono anche Turing-riconoscibili, ma il contrario non è vero. Esistono linguaggi Turing-riconoscibili che non sono decidibili.

4 Esempi di Macchine di Turing

4.1 Esempio 1: Linguaggio delle Stringhe di 0 con Lunghezza Potenza di 2

Consideriamo il linguaggio $A = \{0^{2^n} | n \geq 0\}$.

Procedimento di risoluzione

M_1 = "su input w:

1. Scorri il nastro da sinistra a destra, cancellando ogni secondo 0
2. Se il nastro conteneva un solo 0, accetta
3. Se il nastro conteneva un numero dispari di 0, rifiuta
4. Ritorna all'inizio del nastro
5. Vai al passo 1."

L'idea è che se abbiamo 2^n zeri, dopo un'iterazione avremo 2^{n-1} zeri. Ripetendo il processo, alla fine avremo un solo 0 se e solo se la lunghezza iniziale era una potenza di 2.

4.2 Esempio 2: Linguaggio delle Copie di Stringhe

Consideriamo il linguaggio $B = \{w\#w | w \in \{0,1\}^*\}$.

Procedimento di risoluzione

M_2 = "Su input w:

1. Si muove a zig-zag lungo il nastro, raggiungendo posizioni corrispondenti ai due lati di # per controllare se contengono lo stesso simbolo. In caso negativo, o se non trova #, rifiuta. Barra gli elementi già controllati.
2. Se tutti i simboli a sinistra di # sono stati controllati, verifica i simboli a destra di #. Se c'è qualche simbolo ancora da controllare rifiuta, altrimenti accetta."

4.3 Esempio 3: Linguaggio delle Operazioni Aritmetiche

Consideriamo il linguaggio $C = \{a^i b^j c^k | k = i \cdot j \text{ e } i, j, k \geq 1\}$.

Procedimento di risoluzione

M_3 = "su input w:

1. Scorri il nastro da sinistra a destra e controlla se l'input sta in $a^+b^+c^+$. Rifiuta se non lo è.
2. Ritorna all'inizio del nastro
3. Barra una a e scorri a destra fino a trovare una b. Fai la spola tra b e c, barrando le b e le c fino alla fine delle b. Se tutte le c sono barrate e rimangono ancora b, rifiuta
4. Ripristina le b barrate e ripeti 3 finché ci sono a da barrare.
5. Quanto tutte le a sono barrate, controlla se tutte le c sono barrate: in caso positivo accetta, altrimenti rifiuta."

Questo procedimento simula la moltiplicazione: per ogni a, barra j simboli c (tanti quanti sono i b).

4.4 Esempio 4: Linguaggio degli Elementi Distinti

Consideriamo il linguaggio $D = \{\#x_1\#x_2\#\cdots\#x_\ell | x_i \in \{0,1\}^*$ e $x_i \neq x_j$ per ogni $i \neq j\}$.

Procedimento di risoluzione

M_4 = "su input w:

1. Mette un segno sul simbolo del nastro più a sinistra. Se è un blank, accetta. Se è un #, continua con 2. Altrimenti, rifiuta.
2. Scorre a destra fino al successivo # e vi mette sopra un secondo segno. Se nessun # viene trovato, allora era presente solo x_1: accetta.
3. Procede a zig-zag confrontando le due stringhe a destra dei # segnati. Se sono uguali, rifiuta.
4. Sposta il segno più a destra sul successivo # alla sua destra. Se non trova nessun #, sposta il segno più a sinistra sul successivo # alla sua destra, e sposta il segno più a destra sul successivo #. Se non c'è un # dopo il segno più a destra, allora tutte le stringhe sono state confrontate: accetta.
5. Vai alla fase 3."

5 Varianti di Macchine di Turing

5.1 Un Modello Robusto

Concetto chiave

Esistono varie definizioni alternative delle macchine di Turing, chiamate **varianti**. Una caratteristica notevole del modello di Turing è che tutte le varianti "ragionevoli" riconoscono la stessa classe di linguaggi. Per questo motivo, le macchine di Turing sono considerate un modello di calcolo **robusto**.

5.2 Macchine a Nastro Semi-infinito

Una variante è la macchina di Turing con nastro semi-infinito:

- È una TM con un nastro infinito solo verso destra
- L'input si trova all'inizio del nastro
- La testina parte dalla posizione più a sinistra del nastro
- Se la macchina tenta di spostare la testina a sinistra quando si trova nella prima cella del nastro, allora la testina rimane ferma

Teorema 1. Per ogni TM a nastro semi-infinito esiste una TM a nastro infinito equivalente, e viceversa.

5.3 Macchine Multinastro

Un'altra variante è la macchina di Turing multinastro:

- È una TM con k nastri semi-infiniti
- Ha k testine di lettura e scrittura
- L'input si trova sul nastro 1
- Ad ogni passo scrive e si muove simultaneamente su tutti i nastri

La funzione di transizione è: $\delta : Q \times \Gamma^k \mapsto Q \times \Gamma^k \times \{L, R\}^k$

Teorema 2. Per ogni TM multinastro esiste una TM a singolo nastro equivalente.

Procedimento di risoluzione

Idea della dimostrazione:

- Simulare la TM multinastro M con una TM a singolo nastro S
- I nastri di M sono separati da # sul nastro di S
- Un punto sopra un simbolo indica la posizione della testina su quel nastro
- Quando S deve simulare una mossa di M, fa più passaggi sul nastro per aggiornare tutti i simboli necessari

5.4 Macchine Non Deterministiche

Una TM non deterministica ha più scelte possibili a ogni passo della computazione:

- La funzione di transizione è: $\delta : Q \times \Gamma \mapsto 2^{Q \times \Gamma \times \{L,R\}}$
- La computazione è un albero che rappresenta tutte le possibili esecuzioni
- La macchina accetta se almeno un ramo dell'albero porta a uno stato di accettazione

Teorema 3. Per ogni TM non deterministica N esiste una TM deterministica D equivalente.

5.5 Altri Modelli Equivalenti

Esistono altri modelli di computazione universali, alcuni molto simili alle macchine di Turing e altri molto diversi. Tutti hanno però una caratteristica comune: l'accesso senza restrizioni a una memoria illimitata.

Alcuni esempi includono:

- Macchine di Turing con nastro bidirezionale infinito
- Macchine con reset, che possono tornare all'inizio del nastro in un solo passo
- Automi a coda (queue automata)
- Enumeratori e altri formalismi

Concetto chiave

Tutti questi modelli sono equivalenti in termini di potere computazionale, cioè riconoscono esattamente la stessa classe di linguaggi. Questa equivalenza è alla base della Tesi di Church-Turing, che afferma che ogni procedimento di calcolo effettivo può essere realizzato mediante una macchina di Turing.

6 Conclusioni

- Le macchine di Turing rappresentano un modello di calcolo potente, capace di catturare l'essenza dell'algoritmo
- Esistono varianti diverse del modello, ma tutte riconoscono la stessa classe di linguaggi
- I linguaggi Turing-riconoscibili (ricorsivamente enumerabili) sono quelli che possono essere riconosciuti da una TM
- I linguaggi Turing-decidibili (ricorsivi) sono un sottoinsieme dei linguaggi Turing-riconoscibili, dove la TM termina sempre
- Esistono linguaggi Turing-riconoscibili ma non decidibili

Suggerimento

Le macchine di Turing hanno un ruolo fondamentale nella teoria della calcolabilità e nella definizione dei limiti teorici del calcolo. Nonostante siano state definite negli anni '30, molto prima dell'avvento dei computer moderni, rappresentano ancora oggi il modello di riferimento per definire cosa può essere calcolato algoritmicamente.