

Modelli di calcolo

1 Studio di algoritmi

Per studiare il significato (semantica operativa) e la complessità di un algoritmo, è necessaria una descrizione formale del modello su cui esso viene eseguito.

Il modello di riferimento è la **macchina RAM**.

2 Modello RAM

Il **modello RAM (Random Access Machine)** è caratterizzato da

1. memoria ad accesso casuale, formata da un numero infinito **registri** (celle) che possono contenere interi qualsiasi;
2. istruzioni di un linguaggio macchina elementare (input e output, aritmetica, accesso e modifica dei registri, salti e salti condizionati).

La semplicità e trasparenza di questo modello consentono una valutazione diretta delle prestazioni.

2.1 Struttura

- **Programma:** è fissato ed è composto da istruzioni ($istr_1, istr_2, \dots$), ciascuna indicizzata da un'**etichetta** (un numero naturale).
- **Location Counter (lc):** contiene l'etichetta dell'istruzione da eseguire.
- **Nastro di lettura:** è dotato di una testina di sola lettura che legge le infinite celle, in ordine, a partire dalla prima. Ogni cella può contenere un intero.
- **Nastro di scrittura:** è dotato di una testina di sola scrittura, che scrive interi nelle celle, in ordine, a partire dalla prima.
- **Registri:** sono infiniti. Ciascuno è identificato da un indirizzo intero k e può contenere un numero intero di qualsiasi dimensione. Il registro R_0 , chiamato **accumulatore**, è l'unico sul quale si possono svolgere operazioni aritmetiche.

2.2 Programma e istruzioni

Un programma è una sequenza *finita* di istruzioni.

Ogni istruzione, indicizzata da un'etichetta, è una coppia (**opcode**, **indirizzo**), nella quale l'indirizzo può essere un **operando** o un'etichetta.

Tipologia	Opcode	Indirizzo
spostamento dati tra i registri	LOAD, STORE	operando
operazioni aritmetiche	ADD, SUB, MULT, DIV	operando
lettura e scrittura	READ, WRITE	operando
salto	JUMP, JGTZ, JZERO, JBLANK	etichetta
arresto	HALT	n. s.

Un operando può assumere tre forme diverse:

- $=i$: indica l'intero $i \in \mathbb{Z}$ (*indirizzamento immediato*);
- i : indica il contenuto di R_i , con $i \in \mathbb{N}$ (*indirizzamento diretto*);
- $*i$: indica il contenuto di R_j , dove j è il contenuto di R_i , con $i, j \in \mathbb{N}$ (*indirizzamento indiretto*).

2.3 Stato della macchina

Lo **stato della macchina** è una funzione che associa a ogni registro (compreso lc) il suo contenuto e alle testine le loro posizioni sui nastri:

$$S : \{r, w, lc, 0, 1, \dots, k, \dots\} \rightarrow \mathbb{Z}$$

Quindi:

- $S(r)$ è la posizione della testina di lettura;
- $S(w)$ è la posizione della testina di scrittura;
- $S(lc)$ è il contenuto del *location counter*;
- $S(k)$ è il contenuto del registro R_k , per ogni $k \in \mathbb{N}$.

Informalmente, lo stato è una “fotografia” della macchina in un preciso istante.

Lo **stato iniziale** S_0 è:

- $S_0(r) = S_0(w) = S_0(lc) = 1$;
- $S_0(k) = 0 \quad \forall k \in \mathbb{N}$;

- il programma è caricato;
- il nastro di lettura contiene i dati x_1, x_2, \dots, x_n , seguiti da un simbolo di *blank* (b) che ne indica la fine;
- il nastro di scrittura è vuoto.

2.4 Esecuzione di un programma

- Si pone la macchina nello stato iniziale.
- Finché lc non indica l'istruzione **HALT** si esegue:
 1. **fetch**: individua l'istruzione da eseguire mediante lc ;
 2. **decode**: decodifica il comando tramite l'opcode;
 3. **execute**: esegue l'istruzione in base alla sua *semantica*, *aggiornando* lo stato della macchina.

2.5 Valore degli operandi

Il valore di un operando op dipende sia dallo stato S che dalla sua forma. Tale valore si indica con:

$$V_S(op) = \begin{cases} i & \text{se } op \text{ è } =i, \text{ dove } i \in \mathbb{Z} \\ S(i) & \text{se } op \text{ è } i, \text{ dove } i \in \mathbb{N} \\ S(S(i)) & \text{se } op \text{ è } *i, \text{ dove } i \in \mathbb{N} \text{ e } S(i) \geq 0 \\ \perp & \text{altrimenti} \end{cases}$$

2.6 Semantica delle istruzioni

La **semantica** di un'istruzione definisce il tipo di cambiamento di stato causato dalla sua esecuzione.

Per tutte le istruzioni, ad eccezione dei salti, si assume implicitamente che $S(lc) := S(lc) + 1$.

Istruzioni di spostamento dati tra registri

Istruzione	Semantica
LOAD a	$S(0) := V_S(a)$
STORE i	$S(i) := S(0)$
STORE $*i$	$S(S(i)) := S(0)$

Istruzioni aritmetiche

Istruzione	Semantica
ADD a	$S(0) := S(0) + V_S(a)$
SUB a	$S(0) := S(0) - V_S(a)$
MULT a	$S(0) := S(0) \times V_S(a)$
DIV a	$S(0) := S(0) \div V_S(a)$

Istruzioni di lettura e scrittura sui nastri

Istruzione	Semantica
READ i	$S(i) := x_{S(r)}$ e $S(r) := S(r) + 1$
READ *i	$S(S(i)) := x_{S(r)}$ e $S(r) := S(r) + 1$
WRITE a	stampa $V_S(a)$ nella cella $S(w)$ del nastro di scrittura e $S(w) := S(w) + 1$

Istruzioni di salto

Istruzione	Semantica
JUMP b	$S(lc) := b$
JGTZ b	se $S(0) > 0$ allora $S(lc) := b$ altrimenti $S(lc) := S(lc) + 1$
JZERO b	se $S(0) = 0$ allora $S(lc) := b$ altrimenti $S(lc) := S(lc) + 1$
JBLANK b	se $x_{S(r)} = b$ allora $S(lc) := b$ altrimenti $S(lc) := S(lc) + 1$

Istruzione di arresto

Istruzione	Semantica
HALT	arresta la computazione

2.7 Computazione

Una **computazione** è una sequenza (finita o infinita) di stati $S_0, S_1, \dots, S_i, \dots$ nella quale:

- S_0 è lo stato iniziale, con input x_1, x_2, \dots, x_n ;
- per ogni i , S_{i+1} si ottiene eseguendo nello stato S_i l'istruzione di indice $S_i(lc)$ del programma P .

Se la sequenza è finita e S_m è l'ultimo stato, allora $S_m(lc)$ indica un'istruzione **HALT**, oppure un'istruzione che non può essere eseguita (ad esempio $S(lc) \leq 0$ o $V_S(a) = \perp$).

2.8 Semantica del linguaggio RAM

La **semantica del linguaggio RAM** si ottiene associando a ogni programma P la funzione parziale **calcolata** da P :

$$F_P : \bigcup_{n=0}^{\infty} \mathbb{Z}^n \rightarrow \bigcup_{n=0}^{\infty} \mathbb{Z}^n \cup \{\perp\}$$

dove $\bigcup_{n=0}^{\infty} \mathbb{Z}^n$ è l'insieme di tutte le possibili sequenze di interi (e quindi di dati in input o output).

Per ogni $n \in \mathbb{N}$ e $\underline{x} \in \mathbb{Z}^n$

- se la computazione si arresta, $F_P(\underline{x})$ è il vettore di interi risultante sul nastro di uscita;
- altrimenti $F_P(\underline{x}) = \perp$ (che si legge “indeterminato”).