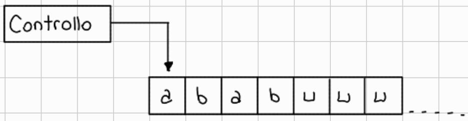
3.1 MACCHINE DI TURING

La Macchina di Turing (MdT) è un modello per calcolare general purpose. Si tratta di un modello molto potente, molto più di quelli visti finora, ed è in grado di fare tutto ciò che un computer reale può fare, quindi risolvere problemi che si trovano entro il limite della calcolabilità. Vediamo le caratteristiche:

1. Usa un nastro potenzialmente infinito come memoria illimitata;
2. Ha una testina che può leggere e scrivere simboli sulle celle del nastro;
3. La testina può muoversi sia a sinistra che a destra;
4. A inizio della computazione il nastro contiene solo la stringa di ingresso, mentre tutte le altre celle sono bianche;
5. La computazione continua finché non produce in uscita un "accetta" o un "rifiuta", che si riferiscono rispettivamente agli stati di accettazione o rifiuto designati.

Lo schema di una possibile MdT è il seguente:



DEFINIZIONE FORMALE DI MACCHINA DI TURING

Definizione 3.3
Una macchina di Turing è una 7-upla $(Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$, dove Q, Σ, Γ sono tutti insiemi finiti e
1. Q è l'insieme degli stati;
2. Σ è l'alfabeto di input non contenente il simbolo blank $\sqcup$;
3. Γ è l'alfabeto del nastro, con $\sqcup \in \Gamma$ e $\Sigma \subseteq \Gamma$;
4. $\delta : Q \times \Gamma \longrightarrow Q \times \Gamma \times \{L, R\}$ è la funzione di transizione;
5. $q_0 \in Q$ è lo stato iniziale;
6. $q_{accept} \in Q$ è lo stato accettante;
7. $q_{reject} \in Q$ è lo stato di rifiuto, con $q_{reject} \neq q_{accept}$

La funzione di transizione δ in pratica dice che quando la macchina si trova in uno stato q e la testina è su una cella che contiene il simbolo b, sostituisce quest'ultimo col simbolo a e si sposta nello stato r. Lo spostamento è verso sinistra se il terzo componente della funzione di transizione è L; mentre se è R si sposta verso destra.

FUNZIONAMENTO DELLA MdT

Siano dati:
- Una MdT $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$;
- Una stringa di input w tale che $w = w_1 w_2 \dots w_n$, dove ogni w_i fa parte dell'alfabeto Σ^*.

Inizialmente w occupa le prime n celle a sinistra del nastro, e la testina si trova sulla prima cella a sinistra. Poiché l'alfabeto Σ non contiene il simbolo blank, il primo $\sqcup$ che appare sul nastro coincide con la fine dell'ingresso. Una volta che M è partita, la computazione procede in accordo alle regole descritte dalla funzione di transizione. Si noti che se ci viene chiesto di spostarci a sinistra del limite sinistro del nastro, rimaniamo fermi sulla prima cella senza sforare. La computazione continua fino a quando non si arriva a un q_{accept} o un q_{reject}, altrimenti va avanti all'infinito.

Durante la computazione, M continua a modificare lo stato corrente, il contenuto del nastro e la posizione della testina. I valori assunti da questi tre elementi ad ogni passo prendono il nome di configurazione della MdT. Queste possono essere rappresentate in una forma molto semplice, che spieghiamo con un esempio: $01000 1 q_3 01011$, significa che il contenuto del nastro è 01000101011, che lo stato corrente è q_3, e che la testina si trova sul carattere all'immediata destra dello stato, ovvero sul quinto 0.

Si dice che una configurazione C_1 produce una configurazione C_2 se la MdT può passare legalmente da C_1 a C_2 in un unico passo. Alcune configurazioni importanti di M sono:

- Configurazione di partenza;
- Configurazione di accettazione, in cui lo stato corrente è q_{accept};
- Configurazione di rifiuto, in cui lo stato corrente è q_{reject};

Tutte queste definizioni ci serviranno per dire che M accetta l'ingresso w se esiste una sequenza di configurazione $C_1, ..., C_k$ tali che:

1. C_1 è la configurazione di partenza di M sull'ingresso w;
2. Ogni C_i produce C_{i+1};
3. C_k è la configurazione di accettazione;

L'insieme di stringhe che M accetta sono chiamate linguaggio di M, o linguaggio riconosciuto da M, scritto anche $L(M)$.

Definizione 3.5

Un linguaggio si dice Turing-riconoscibile se esiste una macchina di Turing che lo riconosce.

Una MdT può accettare, rifiutare o andare in loop su un certo ingresso, dove per loop si intende quel comportamento per cui non si arriva mai a uno stato di terminazione. Dato che è difficile sapere se la fase di loop è temporanea o definitiva, spesso alle MdT generiche si preferisce una variante chiamata decisori, che arrivano sempre a una decisione finale (la stringa è accettata oppure no). Se un decisore riconosce un certo linguaggio, si può dire anche lo decide.

Definizione 3.6

Un linguaggio si dice Turing-decidibile o semplicemente decidibile se esiste una macchina di Turing che la decide.

VARIANTI DI MACCHINE DI TURING

Della MdT vedremo due varianti molto importanti: la MdT multinastro e quella non deterministica. Si noti che ognuna di esse ha le stesse potenzialità dato che riconosce gli stessi linguaggi delle altre (sono quindi equivalenti). Questa proprietà è detta robustezza.

MdT MULTINASTRO

Una MdT multinastro si differenzia da quelle "standard" per le seguenti caratteristiche:

1) Ha più nastri, ognuno ha la propria testina per leggere e scrivere;
2) Inizialmente la stringa di ingresso si trova solo sul primo nastro, mentre tutti gli altri sono inizializzati coi simboli blank (quindi sono considerati vuoti);
3) La funzione di transizione cambia nella forma: $\delta : Q \times \Gamma^k \longrightarrow Q \times \Gamma^k \times \{L, R, S\}^k$, dove gli indici k permettono di compiere più operazioni su k nastri contemporaneamente, mentre l'aggiunta del simbolo S indica che la testina può muoversi a sinistra (L), a destra (R) o stare ferma (S, stay).

Teorema 3.13

Per ogni macchina di Turing multinastro esiste una macchina di Turing a nastro singolo equivalente.

DIM: Ciò che dobbiamo fare è riuscire a convertire una MdT multinastro M nell'equivalente a singolo nastro. L'idea è copiare il contenuto dei k nastri di M sull'unico nastro di S, e separarli da un unico carattere delimitatore, ad esempio il #. Per quanto riguarda le testine di M, possiamo mettere un pallino su ogni carattere in S corrispondente. Nel nostro caso useremo il simbolo °; ad esempio nella stringa $w_1 \mathring{w}_2 w_3$ la testina "virtuale" è sul carattere w_2.

Proviamo a dare una descrizione più completa dell'algoritmo di conversione, che tenga conto anche dei casi critici come lo spostamento su un simbolo #.

S = "su ingresso $w = w_1 w_2 \dots w_n$:

1. Scrivi sul nastro la configurazione iniziale di M nel formato corretto : $\# \mathring{w}_1 w_2 \dots w_n \# \mathring{\sqcup} \# \mathring{\sqcup} \# \dots \#$
2. Per simulare una singola mossa : (a) fai uno scan del nastro partendo dal primo # (limite sinistro) al (K+1)simo # (limite destro), così da capire dove sono posizionate le testine virtuali ; (b) fai un secondo scan per aggiornare il nastro in accordo alle funzioni di transizioni di M.
3) Se, applicando le funzioni di transizioni di M, la testina di S si sposta di un #, significa che in M si sarebbe finiti su una cella blank. Quindi, S deve scrivere un simbolo blank in quella cella e shiftare a destra di una posizione tutti i caratteri rimanenti del nastro. Fatto questo può continuare la simulazione come prima ".

Corollario 3.15

Un linguaggio è Turing-riconoscibile se e solo se qualche macchina di Turing multinastro lo riconosce.

DIM: Il teorema è un "se e solo se" quindi vanno dimostrati entrambi i sensi delle implicazioni.

1. Se un linguaggio è Turing-riconoscibile, allora esiste un MdT multinastro che lo riconosce.
   Dato che un linguaggio è Turing-riconoscibile se esiste un MdT a singolo nastro che lo riconosce, e che questa è un caso particolare di un MdT multinastro (dove il numero di nastri è $K = 1$), l'implicazione è banalmente risolta.

2. Se un linguaggio è riconosciuto da una MdT multinastro, allora è Turing-riconoscibile.
   Dato che per il teorema 3.13 ogni MdT multinastro ha un equivalente MdT a singolo nastro, e che un linguaggio è Turing-riconoscibile se esiste una MdT (a singolo nastro) che la riconosce, anche la seconda implicazione è banalmente risolta.

MACCHINA DI TURING NON DETERMINISTICA

Una MdT non deterministica può procedere ad ogni punto della computazione secondo diverse possibilità. La funzione di transizione assumerà dunque la nuova forma : $\delta : Q \times \Gamma \longrightarrow P(Q \times \Gamma \times \{L, R, S\})$

Teorema 3.16

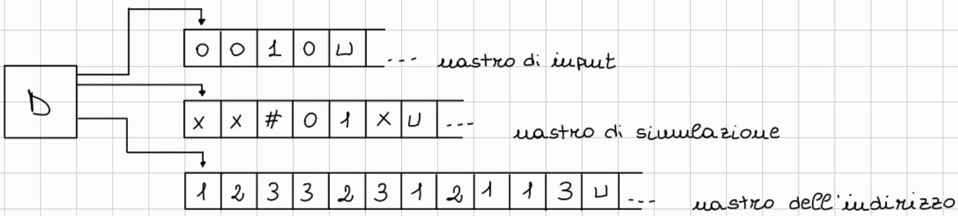
Per ogni macchina di Turing non deterministica esiste una macchina di Turing deterministica equivalente.

IDEA: Dobbiamo simulare che è possibile simulare una MdT non deterministica N con una MdT deterministica D. L'idea è semplice : D dovrà tentare tutti i possibili rami di computazione non deterministica di N, e se arriva a uno stato di accettazione, accetta. Come sappiamo la computazione di N può essere vista come un albero, in cui ogni ramo rappresenta un ramo del nondeterminismo ed ogni nodo una configurazione di N. D dovrà quindi esplorare l'albero alla ricerca di una configurazione di accettazione. Attenzione però : non conviene fare un'esplorazione per profondità (dalla radice alla foglia e poi di nuovo su) perché potrei incappare in un ramo infinito e non uscirne mai più. La strategia giusta è quella in larghezza, in cui prima controllo tutti i nodi sullo stesso livello, poi passo a quello più in basso.

DIM: Costruiamo D come una MdT a tre nastri (tanto il teorema 3.13 ci dice che possiamo sempre convertirla in una a singolo nastro), ognuno dei quali ha un proprio ruolo :

1. Contiene la stringa d'ingresso, ed è in sola lettura ;
2. Mantiene una copia del nastro di N su un certo ramo non deterministico di computazione, su cui fa le simulazioni ;
3. Tiene traccia della posizione di D sull'albero di computazione non deterministica di N. Ad esempio la sequenza 1 2 3 significa che deve partire dalla radice, spostarmi sul suo primo figlio, da qui andare sul suo secondo figlio, e infine da qui al suo terzo figlio.

Ecco uno schema :



Descriviamo D in modo meno formale del solito : D = "

1. Inizialmente abbiamo sul primo nastro l'ingresso w, mentre gli altri due sono vuoti;
2. Copiamo il contenuto del primo nastro sul secondo;
3. Simuliamo sul secondo nastro il comportamento di N con ingresso w su un certo ramo di computazione non deterministico. Prima di ogni passo di N dobbiamo consultare il prossimo simbolo sul terzo nastro, così da capire quale scelta dobbiamo fare tra quelle consentite dalla funzione di transizione di N. Se non ci sono più simboli sul terzo nastro, o quella scelta non é valida, usciamo da questo ramo di computazione e andiamo al passo 4. Stessa cosa se troviamo una configurazione di rifiuto. Se invece troviamo una configurazione di accettazione, D accetta l'ingresso.
4. Sostituisci la stringa sul terzo nastro con quella successiva. Simula il prossimo ramo di computazione di N tornando al passo 2.

Corollario 3.18

Un linguaggio é Turing-riconoscibile se e solo se esiste una macchina di Turing non deterministica che lo riconosce.

DIM : Il teorema é un "se e solo se" quindi vanno dimostrati entrambi i sensi delle implicazioni.

1. Se un linguaggio é Turing-riconoscibile, allora esiste una MdT non deterministica che lo riconosce.
   Dato che un linguaggio é Turing-riconoscibile se esiste una MdT deterministica che lo riconosce, e che questa é un caso particolare di una MdT non deterministica, l'implicazione é banalmente risolta.

2. Se un linguaggio é riconosciuto da una MdT non deterministica, allora é Turing-riconoscibile.
   Dato che per il teorema 3.16 ogni MdT non deterministica ha un'equivalente MdT deterministica, e che un linguaggio é Turing-riconoscibile se esiste una MdT deterministica che lo riconosce, anche la seconda implicazione é banalmente risolta.

Corollario 3.19

Un linguaggio é decidibile se e solo se esiste una macchina di Turing non deterministica che lo decide.