

Calcolabilità e Complessità

Lezione 3

Ripasso di Riconoscimento, Rifiuto, Non-terminazione
↓
Stato di accettazione in un numero finito

Ogni MT accetta un linguaggio

Una MT può riconoscere un L ma non terminare. Un decodificatore termina sempre

Un linguaggio è decidibile se esiste una MT che lo riconosce

INIZIO LEZ 3

Esercizio

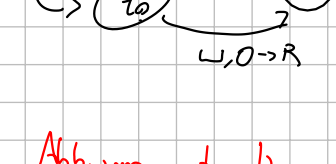
Ordine lessico-grafico su stringhe binarie (Ordine messo)

// Inizio se trova solo 1 aggiunge 0 alla fine

e se ha tutti gli 1 a 0

// Se trova 0 metti 1, Se trova 1 metti 0

0
1
0 0
0 1
1 0
1 1
0 0 0
:
:



Abbiamo visto la macchina di Turing a più registri

Macchine di Turing Non-deterministiche

Def $\delta: Q \times \Gamma \rightarrow \mathcal{P}(Q \times \Gamma \times \{L, R\})$ // MT restituisce tutte le possibili terne: (Un insieme) $(q_i, a_i, \{L, R\})$

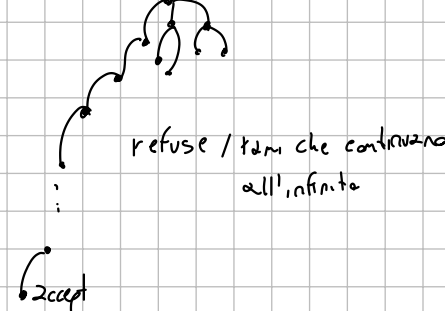
↓
Insieme delle parti

Possiamo immaginare la computazione non-det. come un albero:

Questa macchina è sempre un decodificatore positivo!

Se non ha informazione usa calcolo non deterministico

MANCANZA DI INFORMAZIONI



Come lo implemento? Scelta arbitraria ad ogni biforcazione

↳ Dipende dalla probabilità di trovare

la strada di accettazione

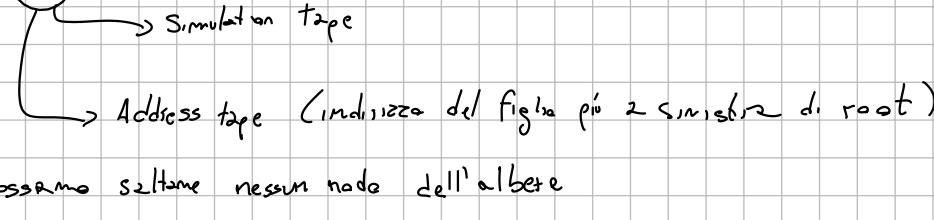
Calcolo parallelo: Sdoppiamo il processore ad ogni ramo (Troppi processori / numero aumentato a potenza esponenziale)

Macchina deterministica \equiv Macchina non deterministica

Se non trovasse soluzioni con rami infiniti? Non terminerà mai!

Vedi Teorema dei 4 colori

MDT Non-Deterministica simulata su 3 Tape



Non possiamo saltare nessun nodo dell'albero

Potremmo usare ricerca in ampiezza: DFS

Come capiamo se terminare? Quando siamo arrivati alla fine.

Cosa aggiungere alla simulazione come rifiuto?

Se non riesce a finire nessun address tape so che non è accettato

↳ Questo criterio è molto costoso, ma rifiuta o accetta o non termina

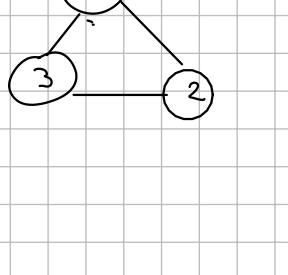
"Finché non posso dalla lunghezza di una tape alla successiva"

Quali problemi posso risolvere con un algoritmo

Esempio: Stabilire se un grafo è connesso

$\langle G \rangle = (1, 2, 3, 4) / ((1, 2), (2, 3), (3, 1), (1, 4)) \rightarrow$ Inserisco questa stringa in una Tape

ed una MT può risolverlo!



Esiste un modo più semplice per sapere se un

algoritmo è Turing