7 Macchine di turing

lunedì 11 novembre 2024 13:43

Automata, Languages and Computing

Macchine di Turing (MT)

1

1

Macchine di Turing

- la *macchina di Turing* è un automa con una testina di scrittura/lettura su nastro illimitato bidirezionale
- ad ogni istante la macchina si trova in uno stato di un insieme finito K
- la funzione di transizione δ porta la macchina da uno stato ad un altro, facendo anche scrivere e spostare la testina
- possono essere deterministiche o non deterministiche

2

2

Macchine di touring

Registrazione audio avviata: 15:02 lunedì 11 novembre 2024

Qui posso a scrivere e leggere in qualunque posizione del nastro (diverso dagli automi a pila).

Macchine di Turing

"*...è incredibile quanto poco occorra per calcolare tutto ciò che è calcolabile...*"

(C. Papadimitriou)

- descrivono formalmente il concetto di algoritmo
- sono in grado di simulare qualsiasi linguaggio di programmazione

3

3

Un problema di Hilbert

- nel 1900 David Hilbert, in un celebre intervento ad un congresso, presentò 23 problemi come sfida per il secolo successivo
- il decimo problema riguardava gli algoritmi: dato un polinomio trovare un algoritmo che stabilisca se esso ha una radice intera
  - esempio: $6x^3yz^2+3xy^2-x^3-10$
    - ha una radice intera con x=5, y=3 e z=0
- nel suo intervento Hilbert assumeva che l'algoritmo esistesse, il problema era trovarlo

4

4

Un problema di Hilbert

Un problema di Hilbert

- ora sappiamo (dal 1970) che non esiste nessun algoritmo per risolvere il problema posto da Hilbert = non si può risolvere col computer
- ma in quel periodo non c'era nessuno strumento idoneo a trattare la questione
- mancava proprio la *definizione* di algoritmo

Per noi algoritmo e macchina di turing coinciderà

5

5

x^2+4x+9? Problema risolto?

- ma se per esempio il polinomio è x^2+4x+9 ?
  - è facile stabilire se ha radici intere, lo insegnano nella scuola secondaria superiore!
- il fatto che non esista un algoritmo che risolva il problema generale non implica che non esistano algoritmi che risolvano sotto-problemi particolari
- per le equazioni di secondo grado in una sola variabile il problema di Hilbert diventa facile

6

6

Macchina di Turing (MT)

macchina di Turing: $M=<\Sigma,\underline{b},K,q_0,F,\delta>$ (deterministica)

- Σ alfabeto di simboli
- $\underline{b}$ carattere speciale, spazio bianco
- K insieme finito di stati con q_0 stato iniziale
- F insieme di stati finali
- δ funzione $\boxed{\text{(non necessariamente totale)}}$ di transizione

A che serve: è il carattere di "spazio bianco"

Non per forza totale perché possiamo anche non specificarla completamente

$\delta : K \times (\Sigma \cup \{\underline{b}\}) \rightarrow K \times (\Sigma \cup \{\underline{b}\}) \times \{d,s,i\}$

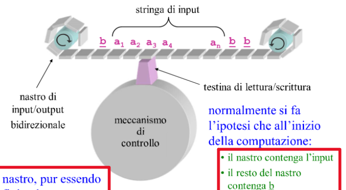
- d,s,i indicano spostamento a destra, a sinistra e immobilità della testina
- talvolta si indica con $\Sigma_{\underline{b}} = \Sigma \cup \{\underline{b}\}$
- le macchine di Turing sono usate sia per accettare stringhe sia per calcolare funzioni: *riconoscitori* e *trasduttori*

Fdt: è in funzione di K (stato corrente) e il carattere letto (di sigma)che può però essere anche lo spazio bianco. -> stato in cui si troverà l'automa e:
1) il carattere che la mt scrive sulla posizione del nastro sulla posizione in cui ho la testina (dove sto guardando) sigma o spazio bianco. Ci dice dove è posizionata la testina. Può scrivere un carattere di sigma oppure uno spazio bianco;
2) {d,s,i}: indica lo spostamento della testina: a destra, a sx, oppure rimani immobile (di una posizione). A dx o sx DI **UNA POSIZIONE** E BASTA.

7

7

Macchina di Turing



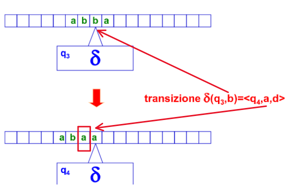
normalmente si fa l'ipotesi che all'inizio della computazione:
- il nastro contenga l'input
- il resto del nastro contenga $\underline{b}$
- la testina sia sul primo carattere dell'input

il nastro, pur essendo infinito, ha un numero finito di caratteri $\neq \underline{b}$

Posso guardare anche porzioni di nastro in presenza di spazi bianchi

8

Una transizione della MT



al posto di b scrivi a e ti sposti a dx.
d sta per: spostati a dx

9

9

Macchine di touring

Registrazione audio avviata: 09:07 mercoledì 13 novembre 2024

Configurazioni delle MT

- configurazione di una MT:
  contenuto del nastro + posizione della testina + stato corrente
  – la conoscenza di una configurazione e della funzione di transizione implica la conoscenza della configurazione successiva
- rappresentazione di una configurazione (es.):
  abb<u>b</u>bq$_i$ab<u>b</u>bbb
  q_i è immediatamente prima del carattere su cui è posizionata la testina

10

10

Configurazione: contenuto del nastro: non tutto, ma ci metto quello che c'è in termini di caratteri diverso dallo spazio bianco. Gli spazi bianchi che metto sono quelli tra le lettere. Qi spezza la stringa, lo stato corrente e si trova prima del carattere su cui è posizionata la testina.

Configurazioni iniziali e accettanti

- configurazione di una MT:
  stringa appartenente al linguaggio
  $(\Sigma_{\underline{b}})^* \circ K \circ (\Sigma_{\underline{b}})^+$
- configurazione iniziale:
  stringa appartenente al linguaggio
  $\{q_0\} \circ (\Sigma_{\underline{b}})^+$
- configurazione accettante:
  stringa appartenente al linguaggio
  $(\Sigma_{\underline{b}})^* \circ F \circ (\Sigma_{\underline{b}})^+$

11

11

Seconde parentesi -> Stringa di caratteri che va esaminata

Config iniziale: la mt si trova nello q0, e l'input da processare è sul nastro quindi la testina è posizionata sul primo carattere da processare (seconda parte della concatenazione). Ci deve essere almeno un carattere, non deve essere vuota. Anche <u>b</u>.
accanto a q0 qualcosa ci devo sempre mettere.
Se l'input è una stringa vuota? Q0<u>b</u> -> sul nastro ci sono solo spazi bianchi, e la testina è posizionata su uno degli spazi bianchi

È una configurazione nella quale lo stato rappresentato appartiene a F, insieme degli stati finali.
Nelle mt, la convenzione è: la computazione termina su uno stato finale (come gli automi normali). Quando scrivo uno stato nella rappresentazione della configurazione a destra ci deve essere qualcosa convenzionalmente. A destra di F potrei avere ancora parte della stringa. Qui nella condizione accettante ci può essere qualunque cosa nel nastro, l'importante è che lo stato sia uno stato di F, intorno possono esserci caratteri in modo qualunque.

Computazione di una MT

- osservazione: la funzione di transizione si presta ad una rappresentazione tabellare o ad una rappresentazione mediante grafo di transizione
- l'applicazione della funzione di transizione ad una configurazione si chiama *passo* o *mossa*
- relazione di transizione per macchina di Turing: relazione binaria sulle configurazioni $\vdash$
  $c_i \vdash c_{i+1}$
- computazione per macchina di Turing: sequenza
  $c_1 \vdash c_2 \vdash \dots \vdash c_i \vdash \dots$
  <u>eventualmente</u> <u>infinita</u> di configurazioni

12

12

Fdt -> 2 modi per rappresentarla: tabellare o grafo di transizione.

-> la mt quando sta in una certa config ci, se la funzione di transizione glielo consente va in c i+1.
Quel simbolo rosa è una "MOSSA" -> passaggio da una config all'altra grazie alla fdt.

-> computazione: può essere eventualmente INFINITA.
Negli automi-> computazione di lunghezza FINITA. Quindi per la prima volta parliamo di computazioni che possono essere INFINITE.
ESEMPIO DI MT che effettui una computazione infinita: la mt ha una stringa composta solo da a sul nastro: e la mt nella fdt c'è scritto: <u>"se sei in q0 e hai a, scrivi a al posto di a e rimani immobile".</u> E questa cosa viene fatta infinite volte -> computazione di lunghezza infinita.

Computazione massimale di una MT

- usiamo la chiusura riflessiva e transitiva di $\vdash$, indicata con $\vdash^*$, per denotare computazioni finite
- c' $\vdash^*$ c" indica che c" è ottenibile da c' tramite un numero *finito* di transizioni da c' a c"
- una computazione finita $c_1 \vdash c_2 \vdash \dots \vdash c_n$ è *massimale* se non esiste una configurazione c tale che $c_n \vdash c$
  – la configurazione c_n è la configurazione finale della computazione massimale

13

13

Mettendo l'asterisco -> denotiamo computazioni finite.

Una computazione finita è massimale se arrivati alla computazione cn non puoi fare nulla, ovvero la fdt non ti dice cosa puoi fare in cn.
Se la computazione non è estendibile ulteriormente allora la computazione è finita e massimale. (cioè non puoi andare oltre)

Un giro in macchina con Turing

funzionamento delle MT
MT multinastro
MT non deterministiche
descrizione linearizzata delle MT
MT universale

14

14

Macchine di touring

Registrazione audio avviata: 09:30 mercoledì 13 novembre 2024

Linguaggio accettato da una MT

- una MT *accetta* una stringa x se, quando ha come configurazione iniziale q_0x, ha luogo una computazione massimale la cui ultima configurazione è accettante
- il *linguaggio accettato* da una macchina di Turing M è l'insieme L(M) delle stringhe accettate da M
- un linguaggio è *Turing-riconoscibile* se esiste una MT che lo accetta

15

15

Accettazione di una stringa: voglio che la mt guardi la stringa e poi si fermi in una configurazione in cui lo stato è finale. Deve avere una computazione di lunghezza finita, non ulteriore estendibile, e si ferma in una configurazione accettante.

Linguaggio accettato: L(M). Cosa manca nella def? -> Cosa fa la mt M per stringhe che NON appartengono al linguaggio? La stringa non verrà accettata ma cosa succede alla stringa? Se la stringa non è del linguaggio non mi importa
La mt è fatta solo per riconoscere e accettare un linguaggio

esiste sempre? Se esiste allora il linguaggio è turing-riconoscibile

Stringhe rifiutate da una MT

- una MT *rifiuta* una stringa x se, quando ha come configurazione iniziale q_0x, ha luogo una computazione massimale la cui ultima configurazione *non* è accettante
- quando facciamo partire una MT su una stringa x ci sono tre possibilità:
  - la MT accetta x
  - la MT rifiuta x
  - la MT computa per sempre
    - non si ferma
    - non ha luogo una computazione massimale

16

16

Una mt rifiuta la stringa quando calcola, poi si ferma, la computazione è massimale, ma lo stato in cui si ferma non appartiene a F.

1) Accetta x: vuol dire che ha luogo una computazione finita, massimale accettante
2) Rifiuta x: ha luogo una computazione finita, massimale, non accettante
3) Computa per sempre

MT che "decidono" linguaggi

- una MT M *decide* il linguaggio L(M) che accetta se, quando le proponiamo una stringa che non è di L(M), termina sempre la computazione
  - ogni stringa di Σ^* viene accettata o rifiutata da M
- una MT che decide un linguaggio è un *decisore*

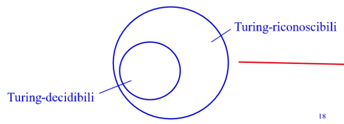
17

17

La mt che decide un linguaggio è un DECISORE.
Una mt prende un L e lo accetta/riconosce quando si esprime positivamente per tutte le stringhe che appartengono al linguaggio.
Una mt **decide** un linguaggio se qualunque stringa gli proponi di sigma asteriscato o accetta o rifiuta.

Linguaggi Turing-decidibili

- un linguaggio è *Turing-decidibile* (o più semplicemente *decidibile*) quando c'è una MT che lo decide
- ogni linguaggio Turing-decidibile è anche Turing-riconoscibile ma non è detto il viceversa



18

Se un ling è touring decidibile è anche riconoscibile;
non è necessariamente vero il contrario.
I linguaggi che ammettono una mt che li decide sono un sottoinsieme dei linguaggi che vengono riconosciuti da una mt.

18

MT trasduttrici

- macchina di Turing trasduttrice:
  - le configurazioni finali sono di tipo *xqy*, dove *x* è il contenuto del nastro all'inizio della computazione
- convenzione:
  - una macchina trasduttrice calcola la funzione *f*(*x*) tale che

$$q_0 x \vdash^* x q_F f(x)$$

19

19

Mt trasduttrice: sono mt che servono per calcolare funzioni:
La fai partire con input qox, avviene una computazione massimale, e restituisce f(x) sul nastro. Mette la testina sul primo carattere di f(x).
X può essere una stringa binaria che rappresenta un numero e la mt per esempio prende il numero e ne calcola il quadrato

Esempio: MT che accetta 0^n1^n ($n \geq 1$)

- nastro all'inizio: b**0000011111**b
- la MT esegue ripetutamente le seguenti operazioni:
  - rimpiazza lo **0** più a sinistra con una **X**: b**X000011111**b
  - si muove a destra verso l'**1** più a sinistra
  - rimpiazza l'**1** più a sinistra con una **Y**: b**X0000Y1111**b
  - si muove a sinistra verso la **X** più a destra
  - si muove di una cella sullo **0** più a sinistra
  - se cercando un **1** trova un b
    - allora rifiuta
  - se dopo aver modificato un **1** in **Y** non trova nessuno **0**
    - allora se non c'è rimasto nessun 1 accetta

20

20

Quello che segue l'ultimo 0

In questo procedimento che sto facendo:
sto MARCANDO oggetti già visitati. È un algoritmo praticamente: devo verificare che ci sia uno 0 per ogni 1 e viceversa: allora si marcano gli 0 e gli uno alternati e fa avanti e indietro matchando 0 con 1. gli 0 già processati ci mette sopra una X, ogni volta che visito l'1 ci metto sopra la y. Se trovo tanti 0 quanti 1 -> top. Altrimenti la mt si arrabbia

Esempio: MT che accetta 0^n1^n ($n \geq 1$)

- q_0 stato usato prima della sostituzione $\boxed{\text{0 X}}$
- q_1 per muoversi a destra verso il primo 1
- q_2 per muoversi a sinistra verso le X
- q_3 per verificare che non rimane nessun 1
- q_4 stato di accettazione

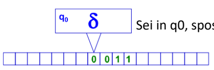
21

21

"Di uno 0 con una X"

Esempio di MT

$\delta(q_0,0)=\{<q_1,X,d>\}$	$\delta(q_0,Y)=\{<q_3,Y,d>\}$
$\delta(q_1,0)=\{<q_1,0,d>\}$	$\delta(q_1,1)=\{<q_2,Y,s>\}$
$\delta(q_1,Y)=\{<q_1,Y,d>\}$	$\delta(q_2,0)=\{<q_2,0,s>\}$
$\delta(q_2,X)=\{<q_0,X,d>\}$	$\delta(q_2,Y)=\{<q_2,Y,s>\}$
$\delta(q_3,Y)=\{<q_3,Y,d>\}$	$\delta(q_3,\underline{b})=\{<q_4,\underline{b},d>\}$



Sei in q0, spostati a dx (d) ma prima di farlo sostituisci 0 con X.

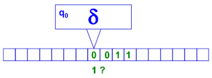
22

22

C'è un modo per realizzare la mt senza fare uso di y e x?
Prendo il primo 0 e diventa uno spazio bianco, poi vado alla ricerca del primo spazio bianco a dx e levo l'1 e così via sperando che #0=#1

Esempio di MT

$\underline{\delta(q_0,0)=\{<q_1,X,d>\}}$	$\delta(q_0,Y)=\{<q_3,Y,d>\}$
$\delta(q_1,0)=\{<q_1,0,d>\}$	$\delta(q_1,1)=\{<q_2,Y,s>\}$
$\delta(q_1,Y)=\{<q_1,Y,d>\}$	$\delta(q_2,0)=\{<q_2,0,s>\}$
$\delta(q_2,X)=\{<q_0,X,d>\}$	$\delta(q_2,Y)=\{<q_2,Y,s>\}$
$\delta(q_3,Y)=\{<q_3,Y,d>\}$	$\delta(q_3,\underline{b})=\{<q_4,\underline{b},d>\}$



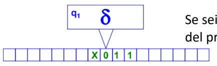
Se ci fosse stato 1 invece? Me lo deve dire la fdt. È una computazione massimale rifiutante -> La stringa viene rifiutata

23

23

Esempio di MT

$\delta(q_0,0)=\{<q_1,X,d>\}$	$\delta(q_0,Y)=\{<q_3,Y,d>\}$
$\underline{\delta(q_1,0)=\{<q_1,0,d>\}}$	$\delta(q_1,1)=\{<q_2,Y,s>\}$
$\delta(q_1,Y)=\{<q_1,Y,d>\}$	$\delta(q_2,0)=\{<q_2,0,s>\}$
$\delta(q_2,X)=\{<q_0,X,d>\}$	$\delta(q_2,Y)=\{<q_2,Y,s>\}$
$\delta(q_3,Y)=\{<q_3,Y,d>\}$	$\delta(q_3,\underline{b})=\{<q_4,\underline{b},d>\}$



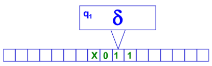
Se sei in q1 e ho 0, rimango in q1, lascio lo 0 dove sta e spostati a dx alla ricerca del primo 1 da marcare

24

24

Esempio di MT

$\delta(q_0,0)=\{<q_1,X,d>\}$	$\delta(q_0,Y)=\{<q_3,Y,d>\}$
$\delta(q_1,0)=\{<q_1,0,d>\}$	$\underline{\delta(q_1,1)=\{<q_2,Y,s>\}}$
$\delta(q_1,Y)=\{<q_1,Y,d>\}$	$\delta(q_2,0)=\{<q_2,0,s>\}$
$\delta(q_2,X)=\{<q_0,X,d>\}$	$\delta(q_2,Y)=\{<q_2,Y,s>\}$
$\delta(q_3,Y)=\{<q_3,Y,d>\}$	$\delta(q_3,\underline{b})=\{<q_4,\underline{b},d>\}$



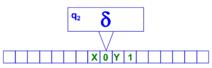
Trovato l'1, lo marco con una Y, mi sposto a sx stavolta per cercare un altro 0 e vado nello q2 per riavvolgere il nastro praticamente

25

25

Esempio di MT

$\delta(q_0,0)=\{<q_1,X,d>\}$	$\delta(q_0,Y)=\{<q_3,Y,d>\}$
$\delta(q_1,0)=\{<q_1,0,d>\}$	$\delta(q_1,1)=\{<q_2,Y,s>\}$
$\delta(q_1,Y)=\{<q_1,Y,d>\}$	$\underline{\delta(q_2,0)=\{<q_2,0,s>\}}$
$\delta(q_2,X)=\{<q_0,X,d>\}$	$\delta(q_2,Y)=\{<q_2,Y,s>\}$
$\delta(q_3,Y)=\{<q_3,Y,d>\}$	$\delta(q_3,\underline{b})=\{<q_4,\underline{b},d>\}$

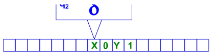


26

26

Esempio di MT

$\delta(q_0,0)=\{<q_1,X,d>\}$	$\delta(q_0,Y)=\{<q_3,Y,d>\}$
$\delta(q_1,0)=\{<q_1,0,d>\}$	$\delta(q_1,1)=\{<q_2,Y,s>\}$
$\delta(q_1,Y)=\{<q_1,Y,d>\}$	$\delta(q_2,0)=\{<q_2,0,s>\}$
$\underline{\delta(q_2,X)=\{<q_0,X,d>\}}$	$\delta(q_2,Y)=\{<q_2,Y,s>\}$
$\delta(q_3,Y)=\{<q_3,Y,d>\}$	$\delta(q_3,\underline{b})=\{<q_4,\underline{b},d>\}$

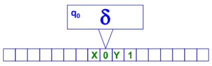


27

27

Esempio di MT

$\delta(q_0,0)=\{<q_1,X,d>\}$	$\delta(q_0,Y)=\{<q_3,Y,d>\}$
$\delta(q_1,0)=\{<q_1,0,d>\}$	$\delta(q_1,1)=\{<q_2,Y,s>\}$
$\delta(q_1,Y)=\{<q_1,Y,d>\}$	$\delta(q_2,0)=\{<q_2,0,s>\}$
$\delta(q_2,X)=\{<q_0,X,d>\}$	$\delta(q_2,Y)=\{<q_2,Y,s>\}$
$\delta(q_3,Y)=\{<q_3,Y,d>\}$	$\delta(q_3,b)=\{<q_4,b,d>\}$



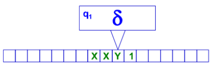
Ricomincio marcando gli 0

28

28

Esempio di MT

$\delta(q_0,0)=\{<q_1,X,d>\}$	$\delta(q_0,Y)=\{<q_3,Y,d>\}$
$\delta(q_1,0)=\{<q_1,0,d>\}$	$\delta(q_1,1)=\{<q_2,Y,s>\}$
$\delta(q_1,Y)=\{<q_1,Y,d>\}$	$\delta(q_2,0)=\{<q_2,0,s>\}$
$\delta(q_2,X)=\{<q_0,X,d>\}$	$\delta(q_2,Y)=\{<q_2,Y,s>\}$
$\delta(q_3,Y)=\{<q_3,Y,d>\}$	$\delta(q_3,b)=\{<q_4,b,d>\}$

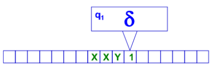


29

29

Esempio di MT

$\delta(q_0,0)=\{<q_1,X,d>\}$	$\delta(q_0,Y)=\{<q_3,Y,d>\}$
$\delta(q_1,0)=\{<q_1,0,d>\}$	$\delta(q_1,1)=\{<q_2,Y,s>\}$
$\delta(q_1,Y)=\{<q_1,Y,d>\}$	$\delta(q_2,0)=\{<q_2,0,s>\}$
$\delta(q_2,X)=\{<q_0,X,d>\}$	$\delta(q_2,Y)=\{<q_2,Y,s>\}$
$\delta(q_3,Y)=\{<q_3,Y,d>\}$	$\delta(q_3,b)=\{<q_4,b,d>\}$



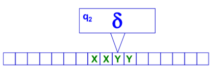
Momento in cui trovo l'1 da marcare con una Y e mi sposto a sx

30

30

Esempio di MT

$\delta(q_0,0)=\{<q_1,X,d>\}$	$\delta(q_0,Y)=\{<q_3,Y,d>\}$
$\delta(q_1,0)=\{<q_1,0,d>\}$	$\delta(q_1,1)=\{<q_2,Y,s>\}$
$\delta(q_1,Y)=\{<q_1,Y,d>\}$	$\delta(q_2,0)=\{<q_2,0,s>\}$
$\delta(q_2,X)=\{<q_0,X,d>\}$	$\delta(q_2,Y)=\{<q_2,Y,s>\}$
$\delta(q_3,Y)=\{<q_3,Y,d>\}$	$\delta(q_3,b)=\{<q_4,b,d>\}$



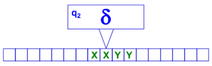
Rivado a sinistra stavolta e arrivo su una x

31

31

Esempio di MT

$\delta(q_0,0)=\{<q_1,X,d>\}$ $\delta(q_0,Y)=\{<q_3,Y,d>\}$
$\delta(q_1,0)=\{<q_1,0,d>\}$ $\delta(q_1,1)=\{<q_2,Y,s>\}$
$\delta(q_1,Y)=\{<q_1,Y,d>\}$ $\delta(q_2,0)=\{<q_2,0,s>\}$
$\delta(q_2,X)=\{<q_0,X,d>\}$ $\delta(q_2,Y)=\{<q_2,Y,s>\}$
$\delta(q_3,Y)=\{<q_3,Y,d>\}$ $\delta(q_3,\underline{b})=\{<q_4,\underline{b},d>\}$



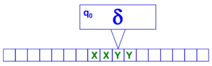
32

Mi sposto a dx ma gli 0 sono finiti

32

Esempio di MT

$\delta(q_0,0)=\{<q_1,X,d>\}$ $\delta(q_0,Y)=\{<q_3,Y,d>\}$
$\delta(q_1,0)=\{<q_1,0,d>\}$ $\delta(q_1,1)=\{<q_2,Y,s>\}$
$\delta(q_1,Y)=\{<q_1,Y,d>\}$ $\delta(q_2,0)=\{<q_2,0,s>\}$
$\delta(q_2,X)=\{<q_0,X,d>\}$ $\delta(q_2,Y)=\{<q_2,Y,s>\}$
$\delta(q_3,Y)=\{<q_3,Y,d>\}$ $\delta(q_3,\underline{b})=\{<q_4,\underline{b},d>\}$



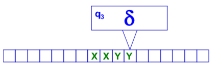
Verifico che non ci siano troppi 1

33

33

Esempio di MT

$\delta(q_0,0)=\{<q_1,X,d>\}$ $\delta(q_0,Y)=\{<q_3,Y,d>\}$
$\delta(q_1,0)=\{<q_1,0,d>\}$ $\delta(q_1,1)=\{<q_2,Y,s>\}$
$\delta(q_1,Y)=\{<q_1,Y,d>\}$ $\delta(q_2,0)=\{<q_2,0,s>\}$
$\delta(q_2,X)=\{<q_0,X,d>\}$ $\delta(q_2,Y)=\{<q_2,Y,s>\}$
$\delta(q_3,Y)=\{<q_3,Y,d>\}$ $\delta(q_3,\underline{b})=\{<q_4,\underline{b},d>\}$



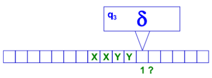
Trovate tutte le Y

34

34

Esempio di MT

$\delta(q_0,0)=\{<q_1,X,d>\}$ $\delta(q_0,Y)=\{<q_3,Y,d>\}$
$\delta(q_1,0)=\{<q_1,0,d>\}$ $\delta(q_1,1)=\{<q_2,Y,s>\}$
$\delta(q_1,Y)=\{<q_1,Y,d>\}$ $\delta(q_2,0)=\{<q_2,0,s>\}$
$\delta(q_2,X)=\{<q_0,X,d>\}$ $\delta(q_2,Y)=\{<q_2,Y,s>\}$
$\delta(q_3,Y)=\{<q_3,Y,d>\}$ $\boxed{\delta(q_3,\underline{b})=\{<q_4,\underline{b},d>\}}$



La stringa viene rifiutata se la ci fosse 1, nel nostro caso finisco in q4 la computazione termina e la stringa è accettata dalla mt.
Questo linguaggio viene riconosciuto dalla mt.

35

Esempio di MT che accetta 0^n1^n $(n\geq1)$

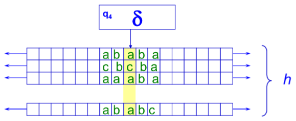
la MT proposta, oltre ad accettare decide?

Si, per come sono state fatte le cose non si finisce in una computazione infinita.
E si può accettare o rifiutare la stringa

36

Macchine di Turing multitraccia

- se il nastro ha *h* tracce la testina può leggere/scrivere *h* caratteri contemporaneamente
  - la posizione della testina è la stessa per tutte le tracce



37

Macchine di touring

Registrazione audio avviata: 10:22 mercoledì 13 novembre 2024

MT e macchine di Turing multitraccia

- la corrispondenza tra MT e MT a nastro suddiviso in tracce è immediata
- osservazione: se sulle tracce sono usati gli alfabeti $\Sigma_1, \Sigma_2, \ldots, \Sigma_h$, una MT *corrispondente* ha un alfabeto Σ con $|\Sigma| \geq |\Sigma_1 \times \Sigma_2 \times \ldots \times \Sigma_h|$

38

C'è una corrispondenza nelle mt.
Su ogni traccia posso utilizzare un alfabeto diverso -> posso creare una mt corrispondente con un alfabeto del tipo: vedi slide
Corrispondente -> ovvero, dato un linguaggio riconosciuto da una mt tradizionale, esiste una mt multitraccia che lo riconosce e viceversa.

Se ho una mt monotraccia, costruirne una multitraccia che fa la stessa cosa è banale, le tracce dalla 2 fino all'ultima non le usi.
Se invece ho a che fare con una mt multitraccia, posso assumere che all'istate iniziale la stringa da riconoscere sia nella prima traccia, poi nella seconda computazione posso usare anche le altre tracce per esempio, utilizzate per la marcatura.
Per costruire la corrispondente monotraccia dalla multi, bisogna codificare opportunamente ogni stringa appartenente alla multitraccia in monotraccia.

Macchine di Turing multinastro (MTM)

macchina di Turing a k nastri:

$M^k = <\Sigma, \underline{b}, K, q_0, F, \delta^{(k)}>$

- Σ alfabeto di simboli
- $\underline{b}$ carattere speciale, spazio bianco
- K insieme finito di stati, con q_0 stato iniziale
- F insieme di stati finali
- $\delta^{(k)}$ funzione di transizione

$\delta^{(k)} : K \times (\Sigma_b)^k \rightarrow K \times (\Sigma_b)^k \times \{d,s,i\}^k$

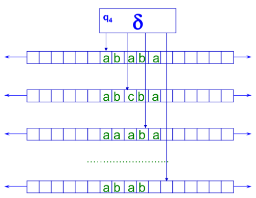
39

Macchine di touring

Registrazione audio avviata: 17:15 giovedì 14 novembre 2024
Ha K nastri in cui fa cose indipendenti in ognuno.

Fdt: stato K x i diversi caratteri sui k nastri -> stato x carattere da scrivere sui vari nastri x mossa da fare sui k nastri.

MTM



40

Esempio: diversi nastri. Ho più testine in base ai nastri che devo leggere.

MTM - configurazioni, transizioni e computazioni

- configurazione:

  $$q\#\alpha_1\uparrow\beta_1\#\alpha_2\uparrow\beta_2\#....\#\alpha_k\uparrow\beta_k$$

  q è lo stato, la stringa $\alpha_i\uparrow\beta_i$ rappresenta la situazione sul nastro i

  α_i è eventualmente vuota e il primo carattere di β_i è il carattere attualmente osservato
- configurazione accettante: q appartiene a F
- configurazione iniziale:

  $$q_0\#\uparrow\beta_1\#\uparrow Z_0\#....\#\uparrow Z_0$$

  l'input sta sul primo nastro e gli altri contengono il carattere Z_0
- transizioni e computazioni: analoghi alle MT

41

41

Configurazione multitraccia:
se ho tanti nastri -> separo con i cancelletti le situazioni sui diversi nastri. Gli alfa e i beta son i caratteri scritti in questo momento sul nastro i-esimo. E la freccetta mi dice la posizione della testina: a dx ho il carattere su cui la i-esima testina è posizionata.

Accettante: quando 1 appartiene ad F

Configurazione iniziale:
devo avere la stringa di input, e tipicamente parto da q0. e poi la stringa di input (1 copia) la metto sul 1° nastro e gli altri li lasciamo vuoti (possiamo anche copiarla successivamente).
La freccia mi dice che il carattere dopo ha la testina sopra di lui. Dal secondo, terzo ecc.. Nastro invece? La testina la metto su uno spazio bianco qualunque: ciò dice che il nastro è vuoto e io sono su uno spazio bianco qualunque. Per convenzione usiamo Z0 per dire che il nastro è vuoto e la testina la impostiamo in quella posizione.

Accettare , rifiutare o ciclare per sempre

Macchine di touring

Macchine di Turing multinastro equivalenza tra MTM e MT

teorema: data una MTM $M=\langle\Sigma,\underline{b},K,q_0,F,\delta^{(k)}\rangle$ a k nastri esiste una MT che simula t passi di M in $\boxed{O(t^2)}$ passi usando un alfabeto di cardinalità $O((2|\Sigma|)^k)$

dimostrazione:

- costruiamo una MT $M'=\langle\Sigma',\underline{b},K',q_0',F',\delta'\rangle$ con nastro suddiviso in 2k tracce che simula M
  - le k tracce di posto pari di M' rappresentano i k nastri di M
  - le k tracce di posto dispari di M' rappresentano con il carattere "↓" la posizione delle testine sui k nastri di M
- poi costruiamo una MT M" equivalente a M'

42

42

Registrazione audio avviata: 17:57 giovedì 14 novembre 2024

"Simula" : la mt che sto costruendo è in grado di fare la stessa cosa passando per diverse configurazioni.

-> ne costruisco 1 multitraccia che simula M

Nastro a 2k tracce per simulare una macchina di Turing a k nastri

1									↓											
2	$\underline{b}$	a	b	c	b	b	b	b	c	c	c	c	c	b	b	b	b	b	b	$\underline{b}$
3		↓																		
4	$\underline{b}$	c	b	b	b	a	b	a	b	b	b	b	b	c	b	b	b	b	b	$\underline{b}$
						↓														
	$\underline{b}$	b	b	b	b	c	b	b	b	b	b	a	b	c	b	b	$\underline{b}$	b	b	$\underline{b}$
										↓										
	$\underline{b}$	b	b	b	b	a	c	b	b	a	b	b	b	b	b	b	b	b	b	$\underline{b}$
														↓						
2k	$\underline{b}$	b	b	a	b	b	b	b	b	a	a	a	a	b	b	b	b	b	b	$\underline{b}$

43

43

Che succede sulle diverse tracce:
Le prime 2 rappresentano il contenuto del primo nastro e vedo dove è posizionata la testina sul primo nastro, ecc..

Simuliamo il comportamento del multi-nastro con una multitraccia.

Dimostrazione (MTM→MT)

- il nastro di M' all'inizio della computazione si presenta con tutte le tracce pari "vuote" tranne la prima
- per simulare la funzione di transizione di M che è del tipo:

  $$\delta^{(k)}(q_i,\sigma_{i1},..,\sigma_{ik}) = \langle q_j,\sigma_{j1},..,\sigma_{jk},z_{j1},..,z_{jk}\rangle$$

  la δ' deve:

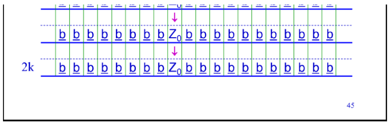
  rintracciare le posizioni dei marcatori,
  scrivere e spostare i marcatori,
  cambiare stato
- quindi per ogni passo di M, M' deve eseguire una sequenza di un numero di passi proporzionale alla distanza (numero di caselle) tra i due marcatori più lontani

44

44

Nastro iniziale di M' con input acca

1									↓											
2	$\underline{b}$	$\underline{b}$	$\underline{b}$	$\underline{b}$	$\underline{b}$	$\underline{b}$	$\underline{b}$	$\underline{b}$	a	c	c	a	$\underline{b}$	$\underline{b}$	$\underline{b}$	$\underline{b}$	$\underline{b}$	$\underline{b}$	$\underline{b}$	$\underline{b}$
3									↓											
4	$\underline{b}$	$\underline{b}$	$\underline{b}$	$\underline{b}$	$\underline{b}$	$\underline{b}$	$\underline{b}$	$\underline{b}$	Z_0	$\underline{b}$	$\underline{b}$	$\underline{b}$	$\underline{b}$	$\underline{b}$	$\underline{b}$	$\underline{b}$	$\underline{b}$	$\underline{b}$	$\underline{b}$	$\underline{b}$
									↓											
	b	b	b	b	b	b	b	b	Z_0	b	b	b	b	b	b	b	b	b	b	b



45

Dimostrazione (MTM→MT)

- dopo t passi due marcatori possono essersi allontanati di al più O(t) caselle
- se M esegue t passi, M' ne esegue $O(t^2)$
- M'' esegue gli stessi passi di M'

- per ciò che riguarda la cardinalità dell'alfabeto di M'', abbiamo da codificare con un solo alfabeto stringhe di 2k simboli così composte:
  - k simboli appartengono a $\{\underline{b},\downarrow\}$
  - 1 simbolo appartiene a $\Sigma\cup\{\underline{b}\}$
  - k-1 simboli appartengono a $\Sigma\cup\{\underline{b},Z_0\}$
  - $|\Sigma''| = 2^k(|\Sigma|+1)(|\Sigma|+2)^{k-1} = O((2|\Sigma|)^k)$

46

46

M è una multinastro
M' è una MT multitraccia mononastro
M'' è una MT mononastro monotraccia che fa lo stesso numero di passi di M' in ordine di t^2 passi.

Su 2K tracce

In K di queste tracce ho solo 2 simboli

Traccia che rappresenta l'input

Simboli che mi servono nella M'' (mononastro monotraccia)

090-macchine-di-turing-24

Macchine di touring

Registrazione audio avviata: 17:26 giovedì 14 novembre 2024

1) Copiamo x sul nastro di lavoro (fino a che arrivo a c)
2) Confronto $\underline{x}$ con x sul nastro di lavoro

Esempi di MTM

esempio: MTM per riconoscere $xc\underline{x}$ con $x\in\{a,b\}^*$

usiamo 2 nastri: uno di input monodirezionale a sola lettura e uno di lavoro che usiamo come pila

durante la scansione di x, fino a c, x viene copiata sul nastro di lavoro

durante la scansione di $\underline{x}$ si confrontano i caratteri con quelli sul nastro di lavoro

configurazione iniziale della MTM:

$q_0\#\uparrow z\#\uparrow Z_0$

dove z è la stringa di input (all'inizio $z = xc\underline{x}$)

47

47

Esempio (continua): MTM per riconoscere $xc\underline{x}$ con $x\in\{a,b\}$

- 3 stati: q_0 per scandire x, q_1 per scandire $\underline{x}$ e q_2 stato finale
- convenzione per le configurazioni:
  <stato, carattere da scrivere sul 2° nastro, movimento del 2° nastro>
- copiatura iniziale: $\delta(q_0,a,Z_0)=<q_0,a,d>$
  $\delta(q_0,b,Z_0)=<q_0,b,d>$
- copiatura a regime: $\delta(q_0,a,\underline{b})=<q_0,a,d>$
  $\delta(q_0,b,\underline{b})=<q_0,b,d>$
- passaggio dalla copiatura alla verifica:
  $\delta(q_0,c,\underline{b})=<q_1,\underline{b},s>$

osservazione: la funzione di transizione è descritta in forma semplificata a causa della natura del primo nastro

48

48

Usiamo 3 stati.
Convenzione per la descrizione della fdt e le configurazioni:
Sto in q0, sul primo nastro ho scritto a e sul secondo Z0, rimani in q0 scrivi a sul secondo nastro e vai a dx. ma di nastri ne ho 2. ma il 1 nastro lo uso sempre per andare a dx lasciando invariato il contenuto che c'è sul nastro. Quello che succede sul 1 nastro non lo scrivo perché è di "sola lettura": vado sempre a dx senza modificare il contenuto.

090-macchine-di-turing-24

Esempio (continua): MTM per riconoscere $xc\underline{x}$ con $x\in\{a,b\}$

1° nastro 2° nastro 2° nastro

- verifica positiva: $\delta(q_1,a,a)=<q_1,a,s>$
  $\delta(q_1,b,b)=<q_1,b,s>$
- accettazione: $\delta(q_1,\underline{b},\underline{b})=<q_2,\underline{b},i>$
- computazione con input bacab:

q_0	$\#\uparrow bacab$	$\#\uparrow Z_0$	$\vdash$	
q_0	$\#b\uparrow acab$	$\#b\uparrow \underline{b}$	$\vdash$	
q_0	$\#ba\uparrow cab$	$\#ba\uparrow \underline{b}$	$\vdash$	Cambio direzione
q_1	$\#bac\uparrow ab$	$\#b\uparrow a$	$\vdash$	
q_1	$\#baca\uparrow b$	$\#\uparrow ba$	$\vdash$	
q_1	$\#bacab\uparrow \underline{b}$	$\#\uparrow \underline{b}ba$	$\vdash$	
q_2	$\#bacab\underline{b}\uparrow \underline{b}$	$\#\uparrow \underline{b}ba$		Stato di accettazione

1° nastro 2° nastro

49

49

q2 stato di accettazione.

Esempio (continua): MTM per riconoscere

Esempio (continua): MTM per riconoscere $xc\underline{x}$ con $x \in \{a,b\}$

- computazione con input acb:

$q_0 \quad \#\uparrow acb \quad \#\uparrow Z_0 \quad \vdash$
$q_0 \quad \#a\uparrow cb \quad \#a\uparrow \underline{b} \quad \vdash$
$q_1 \quad \#ac\uparrow b \quad \#\uparrow a$

50

50

Esempi di MTM

esempio: MTM per riconoscere $x\underline{x}$ con $x \in \{a,b\}$
usiamo 3 nastri: uno di input monodirezionale e due di lavoro
copiamo la stringa sui due nastri di lavoro
poi scandiamo i due nastri in senso contrario ed effettuiamo i confronti
configurazione iniziale della MTM:
$q_0\#\uparrow z\#\uparrow Z_0\#\uparrow Z_0$
copiatura iniziale:
$\delta(q_0,\overset{1}{a},\overset{2}{Z_0},\overset{3}{Z_0})=<q_0,\overset{2}{a},\overset{3}{a},\overset{2}{d},\overset{3}{d}>$
$\delta(q_0,b,Z_0,Z_0)=<q_0,b,b,d,d>$
copiatura a regime:
$\delta(q_0,a,\underline{b},\underline{b})=<q_0,a,a,d,d>$
$\delta(q_0,b,\underline{b},\underline{b})=<q_0,b,b,d,d>$

51

51

Macchine di touring

Registrazione audio avviata: 17:42 giovedì 14 novembre 2024

Copiatura sui due nastri:
- Iniziale: utilizzo la stessa convenzione precedente. Ho 3 nastri: il primo di input e gli altri 2 di lavoro. Ciò che faccio sul primo nastro non sta nelle fdt perché è solo di copiatura

Esempio (continua): MTM per riconoscere $x\underline{x}$ con $x \in \{a,b\}$

termine della copiatura:
$\delta(q_0,\underline{b},\underline{b},\underline{b})=<q_0,\underline{b},\underline{b},s,s>$
riposizionamento della testina:
$\delta(q_0,\underline{b},a,a)=<q_0,a,a,s,i>$
$\delta(q_0,\underline{b},b,a)=<q_0,b,a,s,i>$
$\delta(q_0,\underline{b},a,b)=<q_0,a,b,s,i>$
$\delta(q_0,\underline{b},b,b)=<q_0,b,b,s,i>$
fine del riposizionamento della testina:
$\delta(q_0,\underline{b},\underline{b},a)=<q_1,\underline{b},a,d,i>$
$\delta(q_0,\underline{b},\underline{b},b)=<q_1,\underline{b},b,d,i>$
verifica:
$\delta(q_1,\underline{b},a,a)=<q_1,a,a,d,s>$
$\delta(q_1,\underline{b},b,b)=<q_1,b,b,d,s>$
$\delta(q_1,\underline{b},\underline{b},\underline{b})=<q_2,\underline{b},\underline{b},i,i>$

52

52

Se sul 1 nastro ho lo spazio bianco-> fine copiatura

Riposizionamento testina: queste 4 dft sul secondo nastro vado a dx e sul 3 sto fermo sull'ultimo carattere, ovvero non faccio niente. Mentre sul secondo mi sposto sempre a sinistra: mi posiziono all'inizio della stringa: lo trovo quando a sx stavolta ho spazio bianco (fine riposizionamento della testina).

Verifica: scandisco al contrario per fare il check: il secondo nastro scandisco in avanti, il 3 all'indietro e devo trovare le stesse lettere.
Stato di accettazione se tutto va bene

Esempio (continua): MTM per riconoscere $x\underline{x}$ con $x \in \{a,b\}$

siamo sicuri che la soluzione proposta sia corretta?

53

53

Abba
<u>A</u>bba
Abb<u>a</u>

Se avessi aaa da riconoscere, la riconoscerebbe perché è palindroma. PERO' non è pari. Ma io vorrei riconoscere solo quelle pari con x<u>x</u>.
<u>QUINDI QUESTA SOLUZIONE E' SBAGLIATA.</u>
RICONOSCE IL LINGUAGGIO DI STRINGHE PALINDROME SIA DI LUNGHEZZA PARI CHE DI LUNGHEZZA DISPARI.
ESERCIZIO: modificare la mt affinchè decida x<u>x</u> (sulle stringhe palindrome di lunghezza pari sull'alfabeto {a,b})

Macchine di Turing non deterministiche (MTND)

macchina di Turing non deterministica:
$M=<\Sigma,\underline{b},K,q_0,F,\delta_N>$
Σ alfabeto di simboli
$\underline{b}$ carattere speciale, spazio bianco
K insieme finito di stati con q_0 stato iniziale
F insieme di stati finali
δ_N funzione parziale di transizione
$\delta_N: K\times(\Sigma\cup\{\underline{b}\}) \rightarrow P(K \times (\Sigma\cup\{\underline{b}\}) \times \{d,s,i\})$
d,s,i indicano spostamento a destra, a sinistra e immobilità della testina;

Macchine di turing

Registrazione audio avviata: 14:45 lunedì 18 novembre 2024

La MT va nell'insieme delle parti -> fa diverse cose contemporaneamente

$\Sigma_b = \Sigma \cup \{b\}$

- ogni configurazione può dare luogo a più transizioni verso altre configurazioni

54

54

MTND - funzionamento

- la macchina può eseguire contemporaneamente più transizioni
- grado di non determinismo di una macchina M

  $\nu(M) = \max|\delta_N(q_i,\sigma_j)|$
- il comportamento di una macchina non deterministica può essere rappresentato con un albero di computazione

  nodi: configurazioni

  archi: transizioni
- osservazione: il grado di non determinismo coincide con il massimo numero di figli di un nodo dell'albero di computazione

55

55

Grado di non detrminismo-> quante cose può fare contemporaneamente
Se deterministica -> grado = 1.

MTND - funzionamento

- una MTND M *accetta* una stringa x se nell'albero di computazione di M su x è possibile trovare almeno una computazione deterministica accettante
- una MTND M *rifiuta* una stringa x se tutte le computazioni deterministiche associate all'albero di computazione di M su x sono massimali e non accettanti
- una MTND è un *decisore* (*decide* un linguaggio) se tutte le computazioni deterministiche associate ai sui alberi di computazione sono massimali

56

56

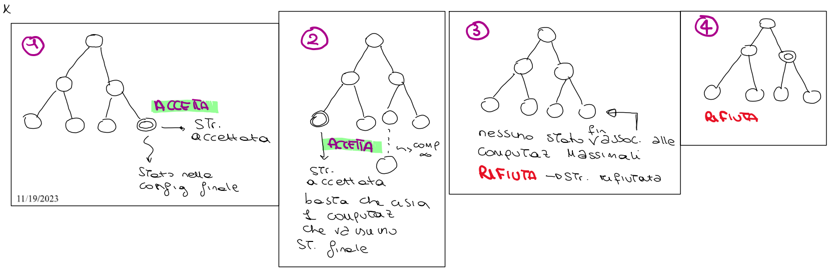
NB:
Che può fare una mt rispetto alla stringa? 1) si ferma: computazione massimale accettando, 2) si ferma: computazionale rifiutando, 3) può continuare a calcolare per un tempo indefinito.
Definizioni:

-> accetta (come per gli automi a stati finiti non det).

-> rifiuta (")

Esempi:

M , x



Equivalenza tra MT e MTND

le MTND non sono più potenti computazionalmente delle MT

teorema: per ogni MTND M con grado di non determinismo d esiste una MT M' equivalente che simula k passi di M in $O(kd^k)$ passi

dimostrazione:

- l'albero di computazione di M viene visitato in ampiezza da M' (perché non in profondità?)

57

57

Macchine di turing

Registrazione audio avviata: 15:03 lunedì 18 novembre 2024

Non si fa in profondità perché sennò vedo una computazione alla volta. E se vedo per esempio una computazione che non termina mai, rimango all'interno del path e non tornerò mai indietro a vedere altri cammini. Vedi esempio 2 qui sopra.

Dimostrazione (MTND→MT)

- M' ha 3 nastri

  nastro 1: contiene l'input   Ovvero la stringa x

  nastro 2: viene usato per generare tutte le sequenze finite composte da cifre comprese tra 1 e d; le sequenze più corte sono generate prima; le sequenze con lo stesso numero di cifre sono generate in ordine numerico crescente

  nastro 3: nastro di lavoro

58

58

Per ipotesi d=2.

Sul nastro 2 ci scrivo tutte le sequenze composte da 1 carattere, 2 caratteri ecc..
(in ordine lessicografico):
ES, Se d=2:
Sul nastro numero 2 ci scrivo: prima tutte le sequenze composte da 1 carattere, poi tutte quelle composte da 2 caratteri in ordine lessicografico
1,2 11,12,21,22.

Dimostrazione (MTND→MT)

- simulazione:

  per ogni sequenza generata sul nastro 2, M' copia l'input sul nastro 3

  le transizioni di ogni insieme $\delta_N(q,\sigma)$ sono numerate da 1 a d

  ogni sequenza di lunghezza s sul nastro 2 è in corrispondenza con una computazione di M di s passi

  gli s numeri di ogni sequenza (compresi tra 1 e d) sono usati per scegliere ad ogni passo una transizione tra le d possibili

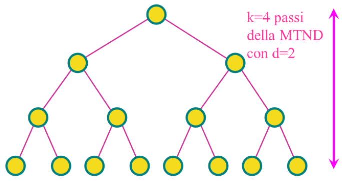
  es. se s=4 e d=2 e la sequenza è 2122 M' sceglie per la prima mossa la seconda transizione disponibile, per la seconda mossa la prima, ecc....

59

59

Le numero: prendo la fdt e le numero una ad una

Costo della simulazione



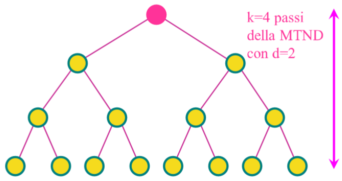
60

60

Costo: qui ho 4 passi con grado di det=2.

1) Guarda tutti i cammini di lunghezza 1; 2 passo
2) Tutti quelli di lunghezza 2; 3 passo
3) Quelli di lunghezza 3 -> 4 passo

Costo della simulazione

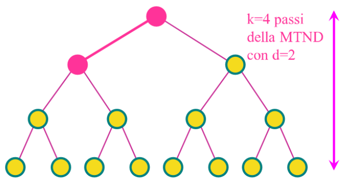


per simulare il primo passo:
tutti i cammini di lunghezza 0

61

61

Costo della simulazione

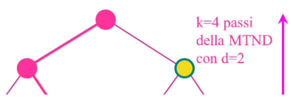


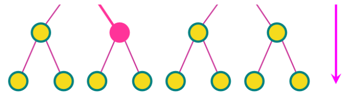
per simulare il secondo passo:
tutti i cammini di lunghezza 1

62

62

Costo della simulazione



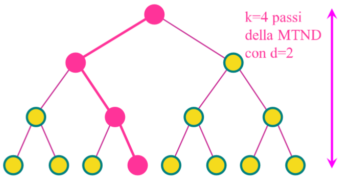


per simulare il terzo passo:
tutti i cammini di lunghezza 2

63

63

Costo della simulazione



per simulare il quarto passo:
tutti i cammini di lunghezza 3

64

64

Dimostrazione (MTND→MT)

- se su qualche foglia dell'albero di computazione di M c'è uno stato finale, allora M' lo raggiunge in tempo finito
  - altrimenti M' non raggiunge mai uno stato finale
- se M termina in k passi M' ha bisogno di

$$O(\sum_{j=0}^{k} jd^j) = O(kd^k) \text{ passi}$$

Tempo per simulare la MTND in modo deterministico

d^j-> numero di cammini che guardo, j è la lunghezza

- osservazione
  - polinomialità ed esponenzialità nella simulazione

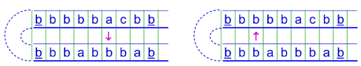
65

65

Simulazione: Da una risoluzione polinomiale in una MTND si passa ad una risoluzione esponenziale. Al momento nessuno è stato in grado di fare di meglio

Alla ricerca del modello di calcolo minimale: ulteriori riduzioni delle MT

- teorema: per ogni MT esiste una MT con nastro semi-infinito equivalente
- dimostrazione
  - simulazione con nastro a 3 tracce
  - il nastro infinito viene "piegato in due"



66

66

Depotenziare le MT per trovare un modello di calcolo minimale.
Teorema:

1) Se invece del nastro infinito ho a disposizione uno semi-infinito (ho una frontiera dove oltre non si può andare), la macchina di turing continua ancora a fare ciò che normalmente fa.
2) Utilizziamo per la dim, la costruzione di una mt con nastro semi-infinito ma diviso in tracce.
3) Parto da una mt normale e ne voglio costruire una equivalente divisa in tracce con nastro semi-infinito.
4) Ho un punto in cui ho la piega -> multinastro con un seminastro di sopra e seminastro di sotto.
5) La parte alta per la piega di sopra; la pare bassa nella parte sotto e in mezzo rappresento la freccetta che indica, attraverso la sua posizione e orientazione, quale è la posizione corrispondente della testina sul nastro infinito.
6) La testina è posizionata su b (disegno a sx).
7) Se la computazione si sposta nella parte alta del nastro la testina cambia orientazione (disegno a dx)
8) Questo tentativo però è senza successo.

Alla ricerca del modello di calcolo minimale: ulteriori riduzioni delle MT

- teorema: per ogni MT M=<Σ,b̲,K,q_0,F,δ> esiste una equivalente MT M'=<Σ',b̲,K',q_0',F',δ'> con |Σ'|=1
- dimostrazione
  - codifica dei caratteri di $Σ_{b̲}$ con i caratteri di $Σ'_{b̲}$

67

67

Altro tentativo: per ogni mt M, ne esiste una equivalente M' con la cardinalità del suo alfabeto =1. quello che puoi fare con una mt che ha un alfabeto qualunque lo puoi anche fare con una mt M' che ha solo 1 carattere.

Dim:
Rappresento ciascuno dei caratteri di sigma con una sequenza di caratteri di sigma primo.
Es:
Sigma': 1
A=1
B=11
C=111
Ma la sequenza ab come la riconosci? Tra i vari caratteri rappresentati con sequenze di 1 inserisco gli spazi bianchi cosi riesco a fare la codifica non ambigua.
ab sarà: 1b̲11
Però la fdt diventerà lunghissima.
Il rapporto temporale tra le 2 macchine per ora non ci interessa.

Codifica delle MT

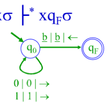
- Concetti principali
  - concatenazione delle MT
  - diramazione condizionata di MT
  - MT elementari
  - descrizione linearizzata delle MT
  - MT universale

1) Concatenazione delle mt: già vista per la chiusura rispetto alla concatenazione dei linguaggi regolari negli automi a stati finiti. La ripetiamo per le mt.
2) Vogliamo, date due mt, farne eseguire una oppure un'altra in base ad una condizione che si deve verificare.
3) Mt elementari: lo facciamo con riferimento a mt molto piccole
4) Rappresentiamo la mt come una stringa -> così posso costruire linguaggi delle mt
5) Mt che prende una mt come stringa sul nastro e la sa eseguire

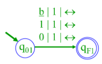
68

Esempi di MT semplici

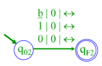
- sia M una MT che calcola $q_0x\sigma \vdash^* xq_F\sigma$
  - con $\sigma\in\Sigma$, $\Sigma=\{0,1\}$ e $x\in\Sigma^*$



- sia M1 una MT che scrive 1



- sia M2 una MT che scrive 0



69

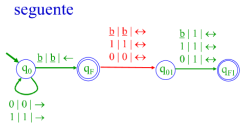
Definiamo queste macchine di turing semplici:
1) Cosa fa: la testina è sul primo carattere della stringa, e alla fine la testina è sull'ultimo carattere della stringa sul dispositivo d'ingresso (sigma minuscolo). questa mt scandisce una stringa.
2) Scrive 1 qualunque cosa gli capita

3) Scrive 0 qualunque cosa gli capita davanti

69

Concatenazione di MT

- la concatenazione di due MT si ottiene introducendo transizioni dagli stati finali della prima MT allo stato iniziale della seconda MT
- esempio: la concatenazione di M ed M1 è la seguente



rappresentazione schematica

$M\ M_1$

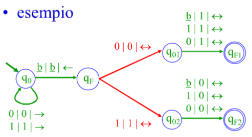
70

70

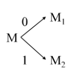
Concateniamo:
1) Parte la prima macchina e poi uniamo qf della M con q iniziale della M1 così passiamo da qf a q01. complessivamente la macchina: scandisce la stringa e sull'ultimo carattere scrive 1.
   Prese M e M' -> il nome della concatenazione diventa: MM'.

Diramazione condizionata di MT

- una diramazione condizionata di una MT si ottiene introducendo transizioni dagli stati finali della prima MT agli stati iniziali delle successive MT dipendenti dal carattere sotto la testina
- esempio



rappresentazione schematica



71

71

Diramazione condizionata:
Ho M che fa la scansione e si posiziona sull'ultimo carattere: se è uno 0 ci scrivo un 1, se è 1 ci scrivo 0.
L'effetto è di far partire M1 o M2 in base a cosa sto osservando sul nastro.
Posso anche rappresentarla schematicamente come il disegno a sx.
È come se avessimo un --> If-then-else

Diramazione verso varie macchine

- siano $M=\langle\Sigma,\underline{b},K,q_0,F,\delta\rangle$ e $M_i=\langle\Sigma,\underline{b},K_i,q_{0i},F_i,\delta_i\rangle$, con $(i=1,\dots,n)$, $n\le|\Sigma|$, $\Sigma=\{\sigma_1,\dots,\sigma_r\}$, n+1 MT con K ed i K_i disgiunti a coppie
- $M'=\langle\Sigma,\underline{b},K',q_0',F',\delta'\rangle$ è ottenuta dalle M e M_i per diramazione se
  - $K' = K \cup K_1 \cup \dots \cup K_n$

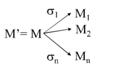
Diramazione verso varie macchine: come se fosse una istruzione switch (?)

– $q_0' = q_0$
– $F' = F_1 \cup \ldots \cup F_n$
– δ' è l'unione delle funzioni di transizione
– per ogni $q \in F$ $\delta'(q,\sigma_j) = \langle q_{0j}, \sigma_j, i \rangle$

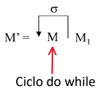
72

Diramazione verso varie macchine

- rappresentazione schematica



- un interessante caso particolare
  – M coincide con una delle macchine verso cui si dirama e tutte le altre sono la stessa macchina M_1
  – rappresentazione schematica



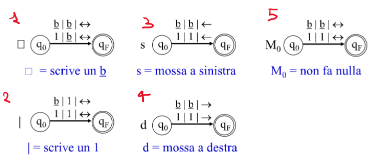
73

Ho M e se leggo sigma 1 nell'ultimo carattere vado in m1, altrimenti se leggo sigma 2 vado su M2 e così via…

Un caso particolare sono i cicli:
Eseguo M, dopo se vedo sigma rieseguo M, se invece il carattere non è sigma vado avanti ed eseguo M1.

MT elementari

- MT elementari con un nastro e due caratteri



74

74

1) Mt quadratino: scrive uno spazio bianco sull'alfabeto ho solo un carattere: 1
2) Mt | : scrive un 1
3) Mt mossa a sx-> sposta di una posizione a sx la testina sul nastro
4) Mt mossa a dx -> analogo a quella prima
5) M0-> non fa nulla

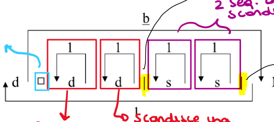
Non stiamo vedendo cosa riconoscono queste mt, ma voglio vedere cosa fanno indipendentemente dall'obbiettivo finale.
Queste mt possono essere usate come pezzi di mt che attraverso la diramazione condizionata contribuisce alla costruzione di una mt più ampia.

Descrizione linearizzata delle MT

esempio: sintesi di una macchina complessa con macchine elementari

sintetizziamo la MT che esegue

$\boxed{q_0\underline{b}x \vdash^* \underline{b}xq_F\underline{b}x}$



scrive un 1 al termine della 2^ scansione di 1 verso dx

2 seq. di 1 scandite verso sx

b sul 1° 1 della 1^ seq.

serve a fare una marcatura per dire che su quell'1 ci siamo già passati

scrivo un 1 all'inizio della 1^ seq. di 1

finché trova 1 si sposta a dx: scandisce una seq. di 1 e si sposta a dx finché non l'ha scandita tutta

scandisce una seconda seq. di 1

scandiscono 2 seq di 1 consecutive e tra loro ho uno b che le separa.

75

75

Assumiamo di partire dalla machina più a sinistra: eseguendo una d
Prendo gli 1 e progressivamente li sto copiando e sono separati da uno spazio bianco.
Fa una copia di una sequenza di uno indipendentemente dal numero di 1 stessi.

Descrizione linearizzata delle MT

teorema: ogni MT è ottenibile per composizione di macchine elementari e ogni macchina ottenuta per composizione di macchine elementari è una MT

dimostrazione:

- la composizione di diagrammi di stato di MT restituisce diagrammi di stato di MT, ciò prova la seconda parte
- per provare la prima parte associamo ad ogni stato la composizione di alcune macchine elementari (| □ d s M_0)

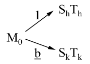
76

76

Le macchine elementari appena viste sono quanto serve per costruire una mt qualunque. Posso costruire tutte e sole le mt.

Dim. (MT→macchine elementari)

- allo stato q_i della MT con $\delta(q_i,1)=<q_h,\sigma_h,t_h>$ e $\delta(q_i,\underline{b})=<q_k,\sigma_k,t_k>$ $\sigma_h,\sigma_k \in \{1,\underline{b}\}$ $t_h,t_k \in \{s,d,i\}$ associamo:



σ_h	S_h
1	|
b	□

σ_k	S_k
1	|
b	□

dove S_h, S_k, T_h e T_k sono come mostrato nelle tabelle

t_h	T_h
d	d
s	s
i	M_0

t_k	T_k
d	d
s	s
i	M_0

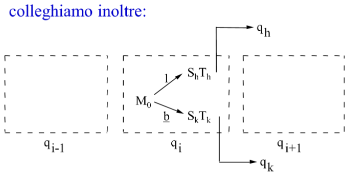
77

Parto dalla mt e cosa fa: si trova sullo stato qi. Se sono in qi o leggo 1 o leggo b
Posso andare in qk o qh e scrivo un carattere 1 o spazio bianco e poi fa una mossa tk o th (sx,dx,immobile).
Associamo a qi il disegno schematico a sx.

Prendo qi, sostituiscigli lo schema e quando sei in qi e leggi un 1 e il carattere da scrivere è 1, sh che scelgo è la pipe. Se ci si sposta a asinistra con Th......
S-> scrivo
T-> sposta la testina

Dim. (MT→macchine elementari)

colleghiamo inoltre:



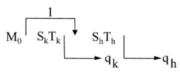
78

78

Posso descrivere la mt come una stringa

Descrizione linearizzata delle MT

- osservazione: per realizzare una macchina con n stati occorrono al più 5n macchine elementari
- osservazione: ogno stato è rappresentabile nella forma linearizzata
  quindi ogni macchina è rappresentabile in forma lienarizzata



- convenzione: l'ultima macchina in un diagramma è M_0
- a questo punto è possibile attribuire un nome ad ogni macchina ed usare i nomi per comporre altre macchine

79

79

Simboleggia il fatto che la computazione si arresta

Quante sono le mt? Sono infinite, ma insieme numerabile: possiamo metterle in corrispondenza biunivoca con i numeri naturali

MT universale

- una MT $U=<\Sigma,\underline{b},K',q_0',F',\delta'>$, che calcola la funzione u, si dice universale se, data una qualunque MT $M=<\Sigma,\underline{b},K,q_0,F,\delta>$ che calcola
$$m: (\Sigma^*)^n \to \Sigma^*$$
con
$$\begin{cases} m(x_1,\ldots,x_n)=y & \text{se } f_M(x_1\underline{b}\ldots\underline{b}x_n\underline{b})=\underline{b}y\underline{b} \\ m(x_1,\ldots,x_n) \text{ indefinita} & \text{altrimenti} \end{cases}$$
si ha
$$u(c_M,x_1,\ldots,x_n) = m(x_1,\ldots,x_n)$$
dove c_M è una codifica di M in Σ

80

80

Mt universale: U
Supponi di avere una mt che calcoli una certa f.
Calcolare è più generale di decidere o riconoscere.
Data M che calcola una certa f, la posso trasformare in una stringa Cm. Poi con Cm che è una stringa la posso mettere sul nastro.
Accanto ci scrivo l'input x, quindi ho 2 stringhe. Posso fare partire la U: esegue M la cui codifica sta sul nastro, sull'input x. E restituisce sul nastro lo stesso valore che avrebbe calcolato M.
Quando calcola la stessa funzione di una mt normale.

MT universale

- U può simulare qualunque MT
- dimostriamo l'esistenza di una U che sa simulare MT con $|\Sigma|=1$ e nastro seminfinito

MT con $|\Sigma|=1$ e nastro semiinfinito

– questa ipotesi non è restrittiva!

Perché qualunque mt ne ha una equivalente con nastro semi infinito e alfabeto di cardinalità =1

81

MT universale

- teorema: esiste una MT $U=\langle\Sigma',\underline{b},K',q_0',F',\delta'\rangle$ tale che data ogni MT $M=\langle\{1\},\underline{b},K,q_0,F,\delta\rangle$
  $f_U(c_M\underline{b}x) = \underline{b}^h f_M(x)\underline{b}^k$ con $h,k\geq 0$ $x\in\{1,\underline{b}\}^*$
- dimostrazione: ingredienti
  – codifica di M con $\Sigma=\{1\}$
  – ulteriore riduzione sull'insieme delle macchine necessarie
  – eliminazione dei salti condizionati su $\underline{b}$
  – simulazione

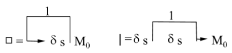
Ne elimino ancora qualcuna

82

Dimostrazione (simulazione di M con U)

ulteriore riduzione sull'insieme delle macchine necessarie
due macchine fondamentali:
δ cambia il carattere letto e va a destra
s va a sinistra
abbiamo che

$d = \delta\ s\ \delta\ M_0$



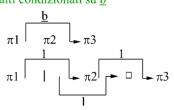
inoltre eliminiamo M_0 ovunque ma non in ultima posizione
rimangono solo δ e s

83

83

Dimostrazione (simulazione di M con U)

eliminazione dei salti condizionati su $\underline{b}$
ad esempio



c_M è quindi organizzata come segue
1 per s
11 per δ Complementa il carattere e si sposta a dx
111 rappresenta la fine delle istruzioni
$1\ldots.1^{(n+3)}$ per salto all'istruzione n

84

84

Dimostrazione (simulazione di M con U)

inoltre

- un'istruzione è separata con $\underline{b}$ dalla successiva
- caratteri speciali per indicare l'istruzione attualmente in esecuzione ed il carattere attualmente esaminato da M: 1' e $\underline{b}$'
- un salto ad istruzione non presente arresta la computazione
  – in particolare 111 termina la sequenza delle istruzioni
- configurazione iniziale

$1\ldots.1\underline{b}1\ldots.1\underline{b}1\ldots.1\underline{b}111\underline{b}\ldots\ldots\ldots..\underline{b}$
c_M x

- configurazione finale

$\underline{b}\ldots\ldots\ldots\ldots\ldots\underline{b}\underline{b}\ldots\ldots\ldots..\underline{b}$
$f_M(x)$

85

85

MT universale

- osservazione: possiamo interpretare la MT universale come un calcolatore ed M e x come un programma e i suoi dati
- osservazione: possiamo interpretare la MT universale come un interprete di un linguaggio di programmazione
- osservazione: possiamo pensare alle macchine di Turing come alle stringhe di un linguaggio, composto dalle codifiche che le rappresentano