Fondamenti d'Informatica: Le Macchine di Turing

Barbara Re, Phd

Agenda

- Introdurremo …
  - … il formalismo delle Macchine di Turing nelle varie versioni
  - … la nozione di calcolabilità e di decidibilità

La Macchina di Turing (MdT)

- E' un modello di calcolo introdotto dall'ing. Alan Turing nel 1936, per **simulare il processo di calcolo umano**
- Rappresenta il primo tentativo di **definizione di procedura effettiva e di programma** eseguito in automatico da una macchina

Alan Turing, 1912-1954

Si prende a esplicito modello **l'impiegato diligente** che svolge con ordine e cura gli incarichi assegnatigli ma non fa nulla di più!

Analisi dei processi di calcolo

- Studio dei vincoli ai quali sottostà un generico agente razionale C impiegato in un processo di calcolo
  - C dispone di una memoria e di capacità percettive limitate
  - C dispone, inoltre, di un supporto (un nastro di dimensioni potenzialmente infinite) su cui scrivere e leggere

- Quali operazioni può svolgere C?
  - C può **scrivere** sul nastro dei simboli tratti da un alfabeto finito
  - C può **osservare** delle caselle sul nastro (è ragionevole assumere che, a causa delle limitate capacità percettive e di memoria esiste un **limite** al numero di caselle osservabili simultaneamente)
  - C è in grado di **ricordare risultati determinati da passi precedenti** della computazione e di **utilizzare** tale informazione nel seguito

Verso le MdT

- Queste condizioni vengono incorporate nella MdT
- La MdT è quindi una macchina che, se concordiamo con quanto detto sino ad ora, **riproduce gli aspetti essenziali dei processi di calcolo**

Elementi della MdT

- Un nastro di lunghezza infinita suddiviso in celle
  - Ogni cella contiene un simbolo dell'alfabeto oppure un simbolo bianco *
  - Il nastro contiene un numero finito di simboli tutto il resto del nastro contiene simboli bianchi

Una Memoria

- Una testina di **lettura** e **scrittura** che si muove lungo il nastro in entrambe le direzioni possibili
  - La testina ad ogni istante può indicare una sola cella del nastro
- Una unità di controllo a stati finiti, cioè controllata da una macchina a stati
  - Contiene il programma secondo cui viene eseguito il calcolo e mantiene lo stato della macchina

Una CPU

stato corrente: q_1

…	*	a	b	*	*	*	…

Elementi della MdT

- **All'inizio del processo di elaborazione** sul nastro si trova la sequenza dei simboli di input e, **al termine si trova l'output del procedimento eseguito**

- La computazione della MdT avviene per passi discreti a partire da **uno stato iniziale prefissato q_0**

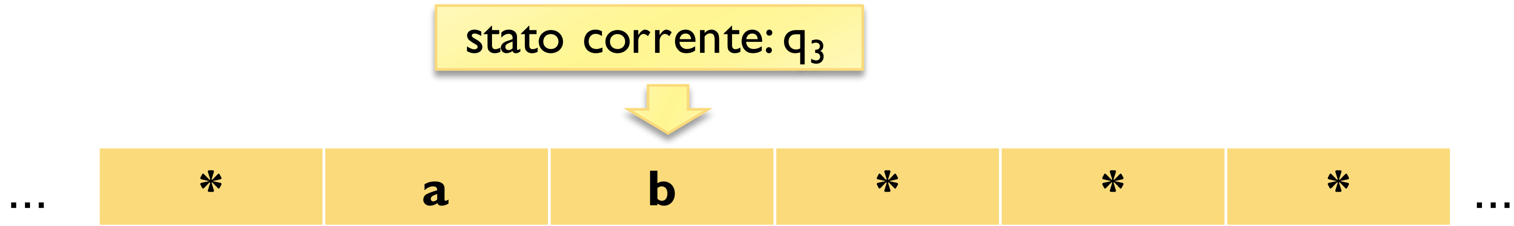
- Il meccanismo di controllo può essere identificato come un **automa**, perché è in grado di assumere uno tra un numero finito di stati e di svolgere una delle seguenti operazioni elementari

- Ad oggi passo l'unità di controllo prende atto dello stato in cui si trova e del simbolo contenuto nella cella e:
  - Rivede il suo stato comandando la sostituzione dello stato attuale con quello successivo
  - Scrive un simbolo nella cella indicata dalla testina, sostituendo il simbolo esistente (compreso *)
  - Sposta la testina di una posizione a sinistra o destra oppure arresta il movimento

In sostanza, durante il suo funzionamento **la macchina evolve da una configurazione all'altra** in corrispondenza
- Del simbolo letto sul nastro
- Dello stato in cui si trova

Allora: viene determinato il simbolo che viene scritto sul nastro, lo stato successivo della macchina e il movimento della testina

Esempio di istruzione di una MdT



- Se ti trovi nello stato q_3 e leggi il simbolo b sul nastro, allora rimpiazza b con a, cambia il tuo stato in q_2, e spostati a destra

Programma di una MdT

- Un programma di una MdT può essere pensato come un insieme di quintuple

  (q, a, q', a', x)

  - *q* stato dell'unità di controllo
  - *a* il simbolo della cella indicata dalla testina
  - *q'* il nuovo stato dell'unità di controllo
  - *a'* il simbolo da scrivere nella cella esaminata dalla testina
  - *x* che può assumere il valore "dx", "sx" o "i" fa riferimento allo spostamento della testina (dx – destra, sx – sinistra e i - immobile)

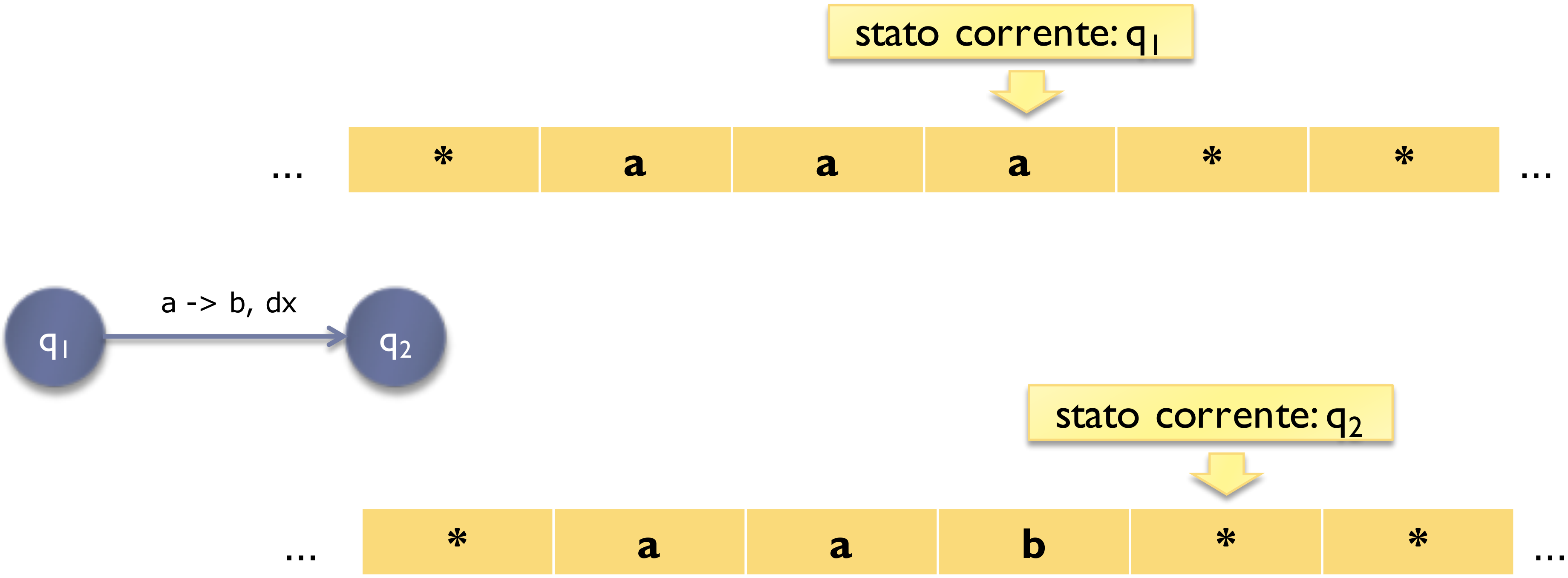
(q, a) $\longrightarrow$ (q', a', x)

Si può ammettere che le istruzioni manchino ma non che si moltiplichino, la corrispondenza deve essere **deterministica** e lontana da ogni ambiguità

Esempio 1

- $\delta\ (q_1, a) = (q_2, b, dx)$

- Se ti trovi nello stato q_1 e leggi il simbolo a sul nastro, assumi lo stato q_2, rimpiazza a con b, e spostati a destra



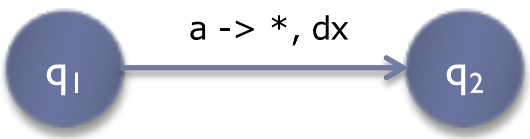
Esempio 2

- $\delta\ (q_1, a) = (q_2, *, dx)$

- Se ti trovi nello stato q_1 e leggi il simbolo a sul nastro, assumi lo stato q_2, rimpiazza a con *, e spostati a destra

stato corrente: q_1

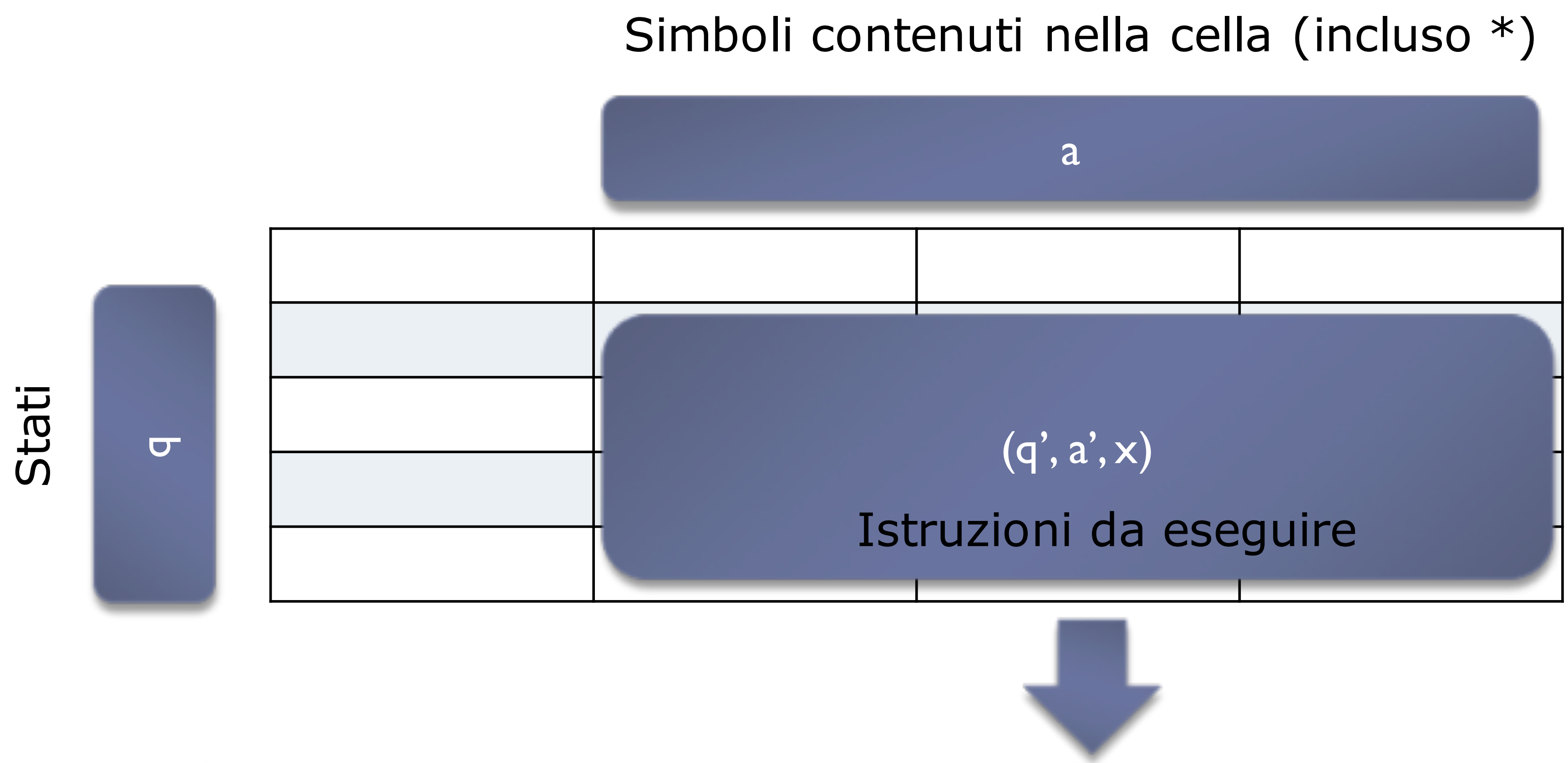
…	*	a	a	a	*	*	…



stato corrente: q_2

…	*	a	a	*	*	*	…

Matrice Funzionale

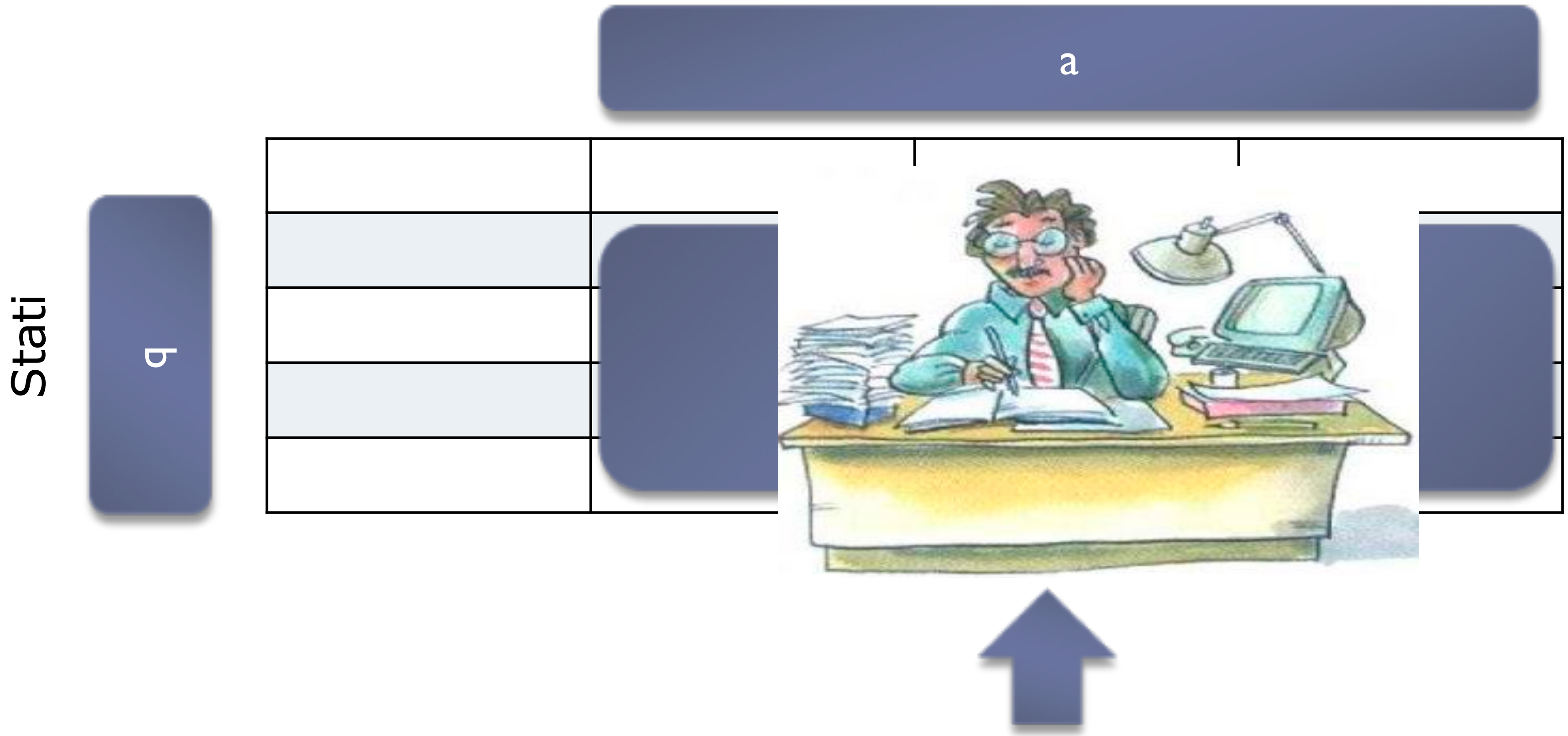


Nel caso una casella della matrice funzionale è vuota significa che la macchina non ha azioni da compiere e quindi termina il calcolo

Una MdT così come definita è detta deterministica a sottolineare che ogni tripla (q’, a’, x), se esiste, è unica e univocamente determinata dalla coppia (q, a)

Matrice Funzionale

Simboli contenuti nella cella (incluso *)



SONO QUESTE LE AZIONI CHE IL NOSTRO **IMPIEGATO DILIGENTE** COMPIE QUOTIDIANAMENTE

MdT è definita da una quintupla

- Una macchina di Turing M deterministica è una quintupla

$$(Q,\ A,\ \delta,\ q_0\ ,\ q_F)$$

- Q è l'insieme finito e non vuoto di stati della macchina
- A è un alfabeto, cui si aggiunge il carattere speciale detto **blank** (rappresentato da un asterisco) che rappresenta la situazione di cella non contenente alcun carattere
- $q_0 \in Q$ è lo stato iniziale
- $q_F \in Q$ è lo stato finale (possono essere definiti anche più di uno stato finale)
- δ è una **funzione di transizione** che fa evolvere la computazione della macchina da

$$\delta = (Q - q_F) \times (A \cup \{*\}) \rightarrow Q \times (A \cup \{*\}) \times \{dx, sx, i\}$$

Gli elementi (q, a, q', a', x) sono chiamati **regole di transizione** o **istruzioni di M**

Trasduttori e Riconoscitori

- Le macchine di Turing possono essere utilizzate:
  - Per riconoscere o accettare stringhe su un alfabeto, in questo caso vengono dette **riconoscitrici**
  - Per calcolare funzioni, in questo caso vengono dette **trasduttori**

Si noti che, le macchine di Turing riconoscitrici calcolano **la funzione caratteristica** dell'insieme delle stringhe che appartengono al linguaggio che si vuole riconoscere

funzione: $\Sigma^* \rightarrow \{0, 1\}$

Vale 1 se per ogni stringa del linguaggio, vale 0 altrimenti

Configurazione di una MdT

Definizione: Si definisce **configurazione** di una macchina di Turing una stringa

$$s \in A^* \circ Q \circ A^*$$

del tipo

$$s = a_{i1} \ldots a_{in}\, q_r\, a_{j1} \ldots a_{jm} \quad n \geq 0, m \geq 0$$

ove:

- A* e l'insieme di tutte le stringhe appartenenti all'alfabeto, il simbolo $\circ$ è l'operatore di concatenazione sulle stringhe
- $a_{i1} \ldots a_{in}\, q\, a_{j1} \ldots a_{jm}$ rappresenta la porzione di nastro contenente caratteri diversi da *, mentre il resto del nastro contiene solo il carattere *
- a_{j1} e il contenuto della cella puntata dalla testina
- q_r e lo stato in cui si trova la macchina
- <u>Configurazioni particolari sono quella **iniziale** e quella **finale**</u>

Configurazione Iniziale e Finale di una MdT

Definizione: Una configurazione **s = xqy** e detta **iniziale** se

$$x = \lambda; \quad q = q_0; \quad y \in A^+$$

- Cioè una configurazione iniziale e una situazione in cui lo stato della macchina e q_0 e la testina si trovi posizionata sul primo carattere a sinistra della stringa di input (non vuota)

Definizione Una configurazione **s = xqy** e detta **finale** se

$$x \in A \quad q = q_F; \quad y \in A^+$$

- Cioè una configurazione finale è una situazione in cui lo stato della macchina e q_F indipendentemente dal contenuto del nastro e dalla posizione della testina

Funzione di Transizione di una MdT

- Data una configurazione c_i una singola applicazione della funzione δ a c_i permette di ottenere una nuova configurazione c_j secondo le seguenti regole:

- $\delta(q_r; a_{j1}) = (q_k; a_h; dx)$ significa compiere la seguente transizione di configurazione:

  $$a_{i1} \ldots a_{in}\ q_r\ a_{j1} \ldots a_{jm} \rightarrow a_{i1} \ldots a_{in}\ a_h\ q_k\ a_{j2} \ldots a_{jm}$$

- $\delta(q_r; a_{j1}) = (q_k; a_h; sx)$ significa compiere la seguente transizione di configurazione:

  $$a_{i1} \ldots a_{in}\ q_r\ a_{j1} \ldots a_{jm} \rightarrow a_{i1} \ldots a_{in-1}\ q_k\ a_{in}\ a_h\ a_{j2} \ldots a_{jm}$$

- $\delta(q_r; a_{j1}) = (q_k; a_h; i)$ significa compiere la seguente transizione di configurazione:

  $$a_{i1} \ldots a_{in}\ q_r\ a_{j1} \ldots a_{jm} \rightarrow a_{i1} \ldots a_{in}\ q_k\ a_h\ a_{j2} \ldots a_{jm}$$

Computazioni di una MdT

Definizione: Una successione di configurazione $c_1,\ldots,c_n$ eventualmente infinita, tale che c_1 e uno stato iniziale e $\forall$ I, $c_i \rightarrow c_{i+1}$ definisce una computazione per la macchina di Turing.

- Usiamo la scrittura $c_i \rightarrow^* c_j$ per denotare l'esistenza di una computazione che da c_i porta a c_j tramite un numero finito, eventualmente 0, di transizioni.
- Ogni computazione può avere al più una configurazione iniziale, e una computazione infinita non ha configurazioni finali.

Definizione Una computazione finita $c_1,\ldots,c_n$ e detta massimale se non esiste una configurazione c tale che $c_n \rightarrow^* c$. Ovvero, una computazione massimale si conclude o con una configurazione finale o con una configurazione in cui non e definita la funzione di transizione δ.

Definizione Una computazione massimale $c_1,\ldots,c_n$ e detta accentante se c_n e uno stato finale.

Definizione Una computazione massimale $c_1,\ldots,c_n$ e detta rifiutante se c_n e uno stato NON finale.

Linguaggio Riconosciuto da una MdT

Una MT puo essere utilizzata per riconoscere o accettare un linguaggio $L \subseteq A^*$

Definizione Data una MdT $M=(Q, A, \delta, q_0, q_F)$ diciamo che M riconosce o decide un linguaggio $L \subseteq A^*$ se e solo se per ogni stringa $x \in A^*$, stringa dell'alfabeto, esiste una computazione massimale accettante o rifiutante per la macchina M:

$$\forall\ x \in A^*, q_0x \rightarrow^* wqz, q = q_F \Leftrightarrow x \in L$$

Cioè una MdT riconosce un linguaggio L se per ogni stringa definita sull'alfabeto A e in grado di stabilire se la stringa x appartiene o no al linguaggio L

Definizione Un linguaggio riconosciuto da una MdT e detto **T-decidibile**

Linguaggio Accettato da una MdT

Definizione: Data una MdT $M=(Q, A, \delta, q_0, q_F)$ diciamo che M accetta un linguaggio $L \subseteq A^*$ se e solo se per ogni stringa $x \in A^* L$ esiste una **computazione massimale accettante** per la macchina M, cioè

$$L = \{x \in A^* | q_0x \rightarrow^* wq_Fz\}$$

- Una MdT accetta un linguaggio L se per tutte le stringhe del linguaggio e in grado di stabilire tale appartenenza, cioe esiste una computazione massimale accettante. Non e in grado di dire nulla per tutte le stringhe non appartenenti al linguaggio.

Definizione: Un linguaggio accettato da una MdT è detto **T-semidecidibile**

Un linguaggio e T-decidibile e anche T-semidecidibile, ma il viceversa non e vero.

Macchine di Turing Trasduttori

Data una mdT $M=(Q, A, \delta, q_0, q_F)$ ed una funzione $f: A^* \rightarrow A^*$, diremo che M calcola la funzione f se e solo se

$$\forall \ x \in A^*, f(x) \downarrow, f(x)=y \Rightarrow q_0 x \rightarrow^* x \{*\}\, q_F\, y$$

Se $x \notin A^*$, o $f(x) \uparrow$ allora o la macchina termina in uno stato non finale o non termina.

In generale:

- Per funzioni arbitrarie $f : D \rightarrow C$ ci si puo ricondurre a MdT operanti su stringhe
- Per funzioni a piu argomenti per esempio $f : D1 \times D2 \rightarrow C$ si può pensare di codicare opportunamente le coppie d1; d2 con simboli da usare sul nastro

Tipi di Macchine di Turing

- Esistono vari tipi di Macchine di Turing
  - **singolo nastro mono traccia deterministiche**
  - singolo nastro multi traccia deterministiche
  - multi nastro mono/multi traccia deterministiche
  - singolo/multi nastro mono/multi traccia NON deterministiche

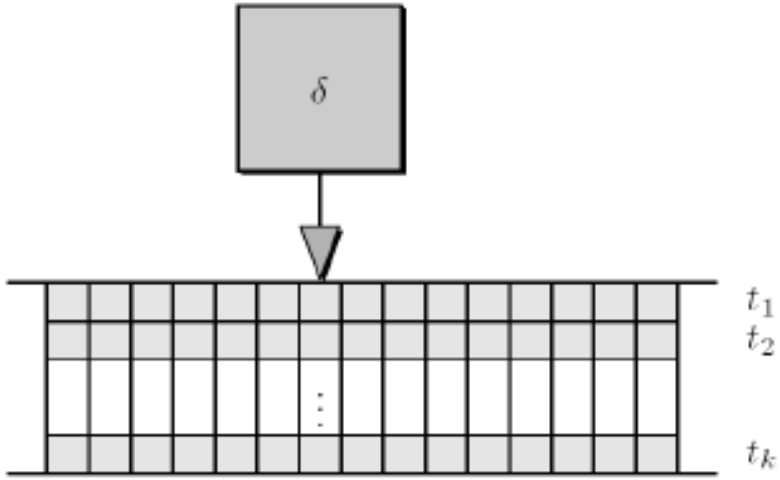
Macchine di Turing Multi-traccia

- Una macchina di Turing multi-traccia consiste di un nastro suddiviso in tracce disposte in modo tale che la testina, con una singola operazione può accedere a tutte le celle di tutte le tracce in corrispondenza della testina

- Possiamo considerare la macchina multi-traccia come una macchina che anzichè operare su simboli scalari opera su simboli vettoriali

- Quindi, una macchina di Turing multitraccia e in grado di scrivere e leggere caratteri vettoriali ma la testina si sposta contemporaneamente su tutte le tracce

- Da un punto di vista fisico si puo immaginare di avere una macchina composta da un nastro suddiviso in m tracce ed una singola testina



MdT deterministica multi-traccia

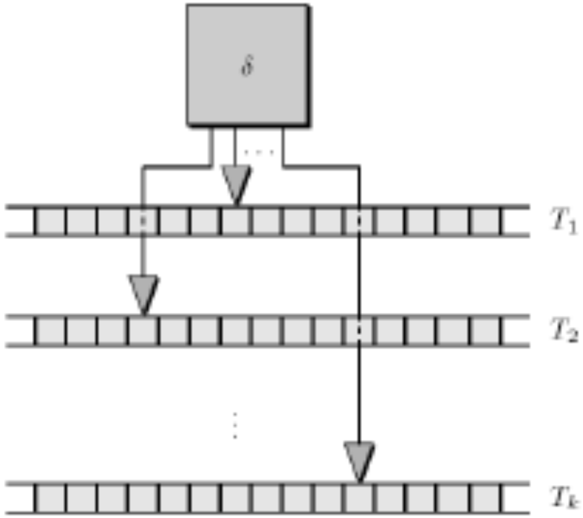
- Una macchina di Turing M deterministica multi-traccia con m tracce è una quadrupla

$$(Q, A, \delta, q_0, q_F)$$

- Q è l'insieme finito e non vuoto di stati della macchina
- $A = A_1 X \dots X A_m$ è un alfabeto, cui si aggiunge il simbolo bianco
- δè una **funzione di transizione** che fa evolvere la computazione della macchina
  da $Q \times (A^m \cup \{*\})$ a $Q \times (A^m \cup \{*\}) \times \{-1, +1\}$
  - Gli elementi (q, a, q', a', x) sono chiamati regole di transizione o istruzioni di M
- $q_0 \in Q$ è lo stato iniziale

MdT deterministica multi-nastro

- Da un punto di vista fisico si può immaginare una macchina composta da m nastri e m testine, una per ogni nastro
- La MdT in base allo stato interno ed ai caratteri letti dalle testine decide in quale stato transire, quali caratteri scrivere sul nastro e come spostare le testine.



MdT deterministica multi-nastro

- Una macchina di Turing M deterministica multi-nastro con m mastri è una quadrupla

$$(Q, A, \delta, q_0, q_F)$$

- Q è l'insieme finito e non vuoto di stati della macchina
- $A = A_1 X \ldots X A_m$ è un alfabeto, cui si aggiunge il simbolo bianco
- δ è una **funzione di transizione** che fa evolvere la computazione della macchina da $Q \times (A^m \cup \{*\})$ a $Q \times (A^m \cup \{*\}) \times \{-1, +1\}^m$
  - Gli elementi (q, a, q', a', x) sono chiamati regole di transizione o istruzioni di M
- $q_0 \in Q$ è lo stato iniziale
- $q_F \in Q$ è lo stato finale (possono essere definiti anche più di uno stato finale)

Espressività

- L'uso di MT multi-traccia permette di avere un maggior potere computazionale ?

- L'uso di MT multi-nastro permette di avere un maggior potere computazionale ?

Espressività

- L'uso di MT multi-traccia permette di avere un maggior potere computazionale ?

- **Teorema** - Una Macchine di Turing singolo nastro multi-traccia M con m tracce puo essere simulata da una macchina di Turing singolo nastro mono-traccia N.

- L'uso di MT multi-nastro permette di avere un maggior potere computazionale ?

- **Teorema** - Sia data una macchina di Turing multi-nastro M con k nastri, allora esiste una macchina di Turing N a singolo nastro che la simula.

Macchine di Turing Deterministiche

- Tutti i modelli di calcolo sinora esaminati sono tutti deterministici, cioè la transizione da un passo della computazione al successivo è sempre univocamente determinata.
- Il determinismo è un concetto molto vicino al modo di funzionamento dei calcolatori i quali ad ogni istante determinano lo stato successivo della computazione in base alla semantica della istruzione da eseguire.
- Si possono definire modelli di calcolo in cui si rilascia l'univocità della funzione di transizione della computazione da un stato al successivo.

MdT non deterministica

- Una macchina di Turing M non deterministica è una quadrupla

$$(Q,\ A, \delta, q_0, q_F)$$

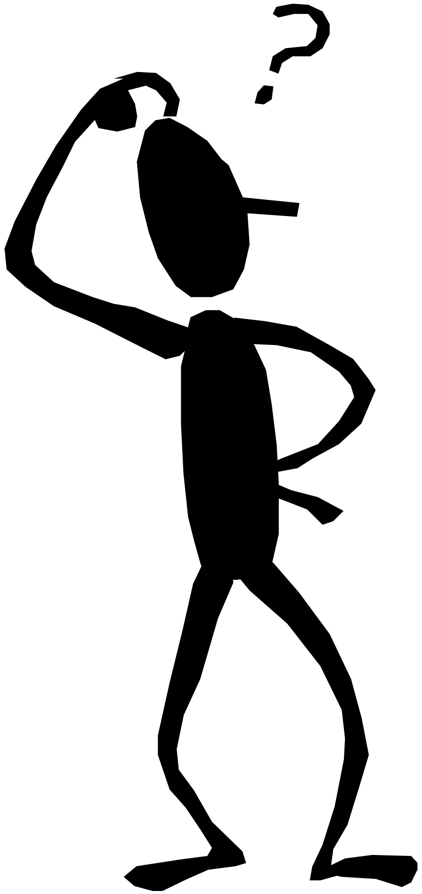
- Q è l'insieme finito e non vuoto di stati della macchina
- A è un alfabeto, cui si aggiunge il simbolo bianco
- δ è una **funzione di transizione** che fa evolvere la computazione della macchina
  da $Q \times (A \cup \{*\})$ a $P(Q \times (A \cup \{*\}) \times \{-1, +1\})$
- $q_0 \in Q$ è lo stato iniziale
- $q_F \in Q$ è lo stato finale (possono essere definiti anche più di uno stato finale)

MdT non deterministica

- La configurazione successiva non e univocamente determinata e la computazione non e piu una successione di configurazioni ma secondo un albero di congurazioni
- Il grado di non determinismo corrisponde, dato una generica configurazione, al massimo numero di configurazioni generate dalla funzione δ.
- Una MT non deterministica si comporta come se ad ogni passo instanziasse nuove MT, ognuna delle quali elabora una delle configurazione diverse prodotte dalla funzione di transizione δ.
- **Teorema** - Per ogni MTND M esiste una MT deterministica $M^{(3)}$ deterministica a 3 nastri equivalente.

Uno dei tanti!

http://kangourou.di.unimi.it/2012/turing/myturing.html



Questions?