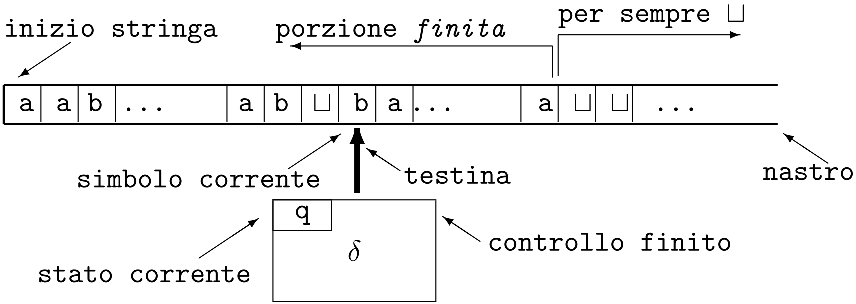
MACCHINE DI TURING

Abbiamo visto che gli automi finiti sono un modello inadeguato per descrivere la nozione intuitiva di algoritmo. Introdurremo il modello delle Macchine di Turing. È comunemente accettato che sia la definizione matematica della nozione di algoritmo.

Schema di una Macchina di Turing.



Descrizione informale di una MdT

- Una macchina di Turing utilizza un nastro semi-infinito.
- Inizialmente il nastro contiene solo la stringa di input.
- La macchina di Turing ha una testina di lettura e scrittura, che può muoversi lungo il nastro.
- In funzione dello stato in cui si trova e del simbolo input letto, la macchina cambia stato, scrive un simbolo nella cella su cui era posizionata, sposta la testina di una posizione a destra o a sinistra.
- La macchina continua a computare finché non **accetta** o **rifiuta**.
- Se non raggiunge uno stato corrispondente a uno delle due situazioni precedenti, la computazione **andrà avanti per sempre**.

Differenze tra una MdT e un automa a stati finiti

La definizione di Macchina di Turing (in seguito abbreviata con MdT o con TM) che daremo, metterà in evidenza le seguenti differenze con il modello automa finito:

- Una MdT ha un nastro semi-infinito (infinito a destra), diviso in celle, ciascuna contenente un carattere.
- Una MdT ha una testina che può muoversi a sinistra o a destra (ma non può oltrepassare il limite a sinistra del nastro).
- La testina può leggere o scrivere caratteri (testina di lettura-scrittura).
- Una MdT ha due stati speciali q_{accept}, q_{reject} rispettivamente corrispondenti all'accettazione e al rifiuto della stringa input. Se la MdT si trova in uno di tali stati non può effettuare ulteriori passi di computazione. (La frase sul testo di Sipser è: **Gli stati speciali di accettazione e rifiuto hanno effetto immediato.**)
- Non c'è limite ai passi di computazione sulla stringa input (in particolare il numero di tali passi non è limitato dalla lunghezza dell'input).

Esempio di Macchina di Turing

Una macchina di Turing che accetta $L = \{0^n 1^n \mid n > 0\}$.
(Descrizione ad alto livello)

(passo 1) Se legge 0 lo sostituisce con X, se legge 1 rifiuta, se legge Y va al passo 4.

(passo 2) Scorre il nastro verso destra, se trova 1 lo sostituisce con Y, altrimenti rifiuta.

(passo 3) Scorre il nastro a sinistra fino a incontrare X, si sposta a destra e ripete dal passo 1.

(passo 4) Scorre il nastro a destra. Se legge solo delle Y e poi il carattere $\sqcup$, accetta. Altrimenti rifiuta.

$000111 \rightarrow X00111 \ldots \rightarrow X00Y11 \ldots \rightarrow XX0Y11 \ldots \rightarrow XXXYYY$

Definizione

Una Macchina di Turing **deterministica** *è una settupla*

$$(Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$$

dove:

- Q*:* **Insieme finito degli stati**
- Σ*:* **Alfabeto dei simboli input** *($\sqcup \notin \Sigma$)*
- Γ*:* **Alfabeto (finito) dei simboli di nastro** *($\sqcup \in \Gamma$, $\Sigma \subset \Gamma$, $L, R \notin \Gamma$)*
- $\delta : (Q \setminus \{q_{accept}, q_{reject}\}) \times \Gamma \to Q \times \Gamma \times \{L, R\}$*:* **funzione transizione**
- $q_0 \in Q$*: stato* **iniziale**
- $q_{accept} \in Q$*: stato di* **accettazione**
- $q_{reject} \in Q$*: stato di* **rifiuto***,* $q_{accept} \neq q_{reject}$

NOTA: Spesso (nel libro, sui lucidi) l'aggettivo "deterministica" è sottinteso.

NOTA: Q, Σ e Γ sono insiemi finiti.
Σ è un sottoinsieme di Γ.
Σ è un sottoinsieme proprio di Γ perché $\sqcup \in \Gamma$ ma $\sqcup \notin \Sigma$.
Q non è mai vuoto.
Γ può contenere altri simboli oltre a quelli di Σ e a $\sqcup$.

NOTA: Esistono molte varianti di questa definizione che hanno lo stesso potere computazionale (o espressivo), cioè riconoscono la stessa classe di linguaggi.

Vedremo due varianti: la macchina di Turing **multinastro** e la macchina di Turing **non deterministica**.

Funzione di transizione di una MdT

Sia $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$ una MdT deterministica.
Il suo comportamento è descritto dalla funzione di transizione

$$\delta : (Q \setminus \{q_{accept}, q_{reject}\}) \times \Gamma \to Q \times \Gamma \times \{L, R\}.$$

Se $\delta(q, \gamma) = (q', \gamma', d)$, sappiamo che $q, q' \in Q$, $\gamma, \gamma' \in \Gamma$, $d \in \{L, R\}$.

Funzione di transizione di una MdT

Se $\delta(q, \gamma) = (q', \gamma', d)$ e se M si trova nello stato q con la testina posizionata su una cella contenente γ, alla fine della transizione:

- M si troverà nello stato q',
- $\gamma' \in \Gamma$ sarà il simbolo scritto sulla cella del nastro su cui la testina si trovava ALL'INIZIO della transizione (contenente γ),
- la testina si sarà spostata sulla cella di sinistra se (tale cella esiste e se) $d = L$, sulla cella di destra se $d = R$.

NOTA: Se $\delta(q, \gamma) = (q', \gamma', L)$, se M si trova nello stato q con la testina posizionata su una cella contenente γ e se tale cella è quella più a sinistra del nastro, la testina non si sposta, resta sulla cella del nastro su cui si trovava all'inizio della transizione.

Spesso nel progetto di una Macchina di Turing si definisce una transizione che scrive un carattere speciale nella cella più a sinistra del nastro e che serve a individuarla.

Diagramma di stato di una MdT

Come nel caso degli automi, spesso preferiamo utilizzare il diagramma di stato di una specifica Macchina di Turing piuttosto che la descrizione formale della settupla.

Il diagramma di stato di una Macchina di Turing è un grafo i cui nodi hanno come nomi gli stati della macchina.

Le etichette degli archi hanno la forma

$$\text{“simbolo} \rightarrow \text{simbolo}, d\text{”}$$

dove “simbolo” è un simbolo di Γ e $d \in \{L, R\}$. Il simbolo a sinistra della freccia rappresenta il simbolo letto, quello dopo la freccia il simbolo (eventualmente lo stesso) che sostituisce il simbolo letto per effetto della transizione, d lo spostamento per effetto della transizione.

Ad esempio se sull'arco dal nodo q_1 al nodo q_2 compare l'etichetta

$$0 \to \sqcup, R$$

questo corrisponde a

$$\delta(q_1, 0) = (q_2, \sqcup, R)$$

Alle volte si usa l'abbreviazione di non indicare il carattere dopo la freccia se esso non è modificato dalla transizione.

Ad esempio se sull'arco dal nodo q_3 al nodo q_4 troviamo l'etichetta

$$0 \to R$$

questo vuol dire che

$$\delta(q_3, 0) = (q_4, 0, R)$$

Esercizio: passare dalla descrizione ad alto livello della MdT per $\{0^n1^n \mid n > 0\}$ al diagramma di stato della MdT.

Esercizio: passare poi dal diagramma di stato alla descrizione della settupla che definisce la MdT.

Computazione di una MdT - descrizione informale

Una MdT M inizia la computazione:

- partendo dallo stato iniziale q_0,
- con l'input $w \in \Sigma^*$ posizionato sulla parte più a sinistra del nastro: le prime n celle a sinistra, se $n = |w|$ è la lunghezza dell'input (il resto delle celle conterrà il carattere $\sqcup$),
- con la testina posizionata sulla cella contenente il primo simbolo di input (quello più a sinistra).

Computazione di una MdT - descrizione informale

- Poiché Σ non contiene il carattere $\sqcup$, all'inizio della computazione il primo simbolo blank sul nastro individua la fine dell'input.
- La computazione di M procede secondo le regole descritte dalla funzione di transizione.
- La computazione di M procede fino a quando non viene raggiunto uno stato di accettazione o rifiuto. Se nessuno dei due stati viene raggiunto, la computazione di M continua per sempre.

Computazione di una MdT - descrizione informale

La computazione termina se M raggiunge

- lo stato di accettazione q_{accept}: **Accetta input**
- lo stato di rifiuto q_{reject}: **Rifiuta input**

NOTA: La computazione può non terminare.

Computazione di una MdT - descrizione informale

NOTA: La computazione può non terminare.
(Analogia: `while C do I` dove C è una condizione sempre vera).

Esempio: $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$, con
$Q = \{q_0, q_{accept}, q_{reject}\}$, $\Sigma = \{a, b\}$, $\Gamma = \{a, b, \sqcup\}$,
$\delta(q_0, a) = (q_0, a, R)$, $\delta(q_0, b) = (q_0, b, L)$,
$\delta(q_0, \sqcup) = (q_{accept}, \sqcup, L)$.

NOTA: "M non accetta l'input w" non è equivalente a "M rifiuta l'input w".

M può accettare un input w o non accettare w. Se non accetta w, questo vuol dire che M rifiuta w oppure che la computazione di M su w non termina.

Computazione di un automa finito deterministico

La computazione di un automa finito deterministico su un input w può essere rappresentata da una sequenza di stati (gli stati che rappresentano i nodi del cammino di etichetta w nel diagramma di stato che rappresenta l'automa).

La posizione della testina è implicitamente rappresentata nella computazione poiché a ogni transizione si sposta di una cella a destra.

Il nastro è di sola lettura, il contenuto del nastro non viene modificato durante la computazione e quindi non è necessario considerarlo.

Configurazioni

Nella computazione di una macchina di Turing su un input w invece abbiamo bisogno di considerare tutti questi elementi:

- stato
- posizione della testina
- la parte iniziale del nastro contenente tutti i caratteri diversi da $\sqcup$ (eventualmente separati da $\sqcup$).
  Nota: è sempre costituita da un numero finito di celle.

Un'impostazione di questi tre elementi è chiamata una configurazione della macchina di Turing.

Configurazioni

Intuitivamente una configurazione di una MdT è l'insieme delle informazioni costituito da una porzione "abbastanza lunga" del nastro, dalla posizione della testina e dallo stato della MdT a un dato istante.

Definizione

Una **configurazione** *C di una MdT $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$ è una stringa $C = uqv \in \Gamma^* Q \Gamma^*$, dove:*

- *$q \in Q$ è lo* **stato corrente** *di M,*
- *$uv \in \Gamma^*$ è il* **contenuto del nastro** *(con la convenzione di aver eliminato tutti i simboli $\sqcup$ dopo v, cioè dopo l'ultimo carattere di v il nastro contiene solo simboli blank),*
- *la* **testina è posizionata** *sul primo simbolo di v.*

Ad esempio

$$1011q_701111$$

rappresenta la configurazione in cui il nastro contiene

$$101101111 \sqcup \ldots \sqcup \ldots$$

lo stato corrente è q_7, la testina è posizionata sul secondo 0.

Configurazione di una MdT

Nota: se $C = uqv$ è una configurazione di una macchina di Turing,

- u è la stringa di caratteri a sinistra del simbolo su cui è posizionata la testina.
- Sia u che v possono essere uguali a ϵ.

Configurazioni: casi particolari

- Se $C = qv$ è una configurazione di una MdT allora la testina è posizionata sulla prima cella del nastro **(posizione della testina a inizio nastro)**.
- Se $C = uq$ è una configurazione di una MdT allora la testina è posizionata sulla prima cella della porzione del nastro contenente solo $\sqcup$ **(posizione della testina sulla porzione del nastro contenente solo $\sqcup$)**.

Passo di computazione

Intuitivamente un passo di computazione è la trasformazione della configurazione C_1 nella configurazione C_2 per effetto di una applicazione della funzione di transizione.

È una relazione tra coppie di configurazioni definita dalla funzione di transizione.

Passo di computazione

Sia $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$ una MdT deterministica.
Siano $q_i, q_j \in Q$, $a, b, c \in \Gamma$ e $u, v \in \Gamma^*$.

Diremo che

$$uaq_ibv \text{ produce } uq_jacv$$

se $\delta(q_i, b) = (q_j, c, L)$.

Diremo che

$$uaq_ibv \text{ produce } uacq_jv$$

se $\delta(q_i, b) = (q_j, c, R)$.

Nota: uaq_ibv, uq_jacv, $uacq_jv$ sono configurazioni di M.

La definizione generale è più complessa perché bisogna considerare anche i casi particolari ($C = qv$, $C = uq$ con u, v eventualmente uguali a ϵ).

Ad esempio $q_i bv$ produce $q_j cv$
se $\delta(q_i, b) = (q_j, c, L)$.

$q_i bv$ produce $cq_j v$ se $\delta(q_i, b) = (q_j, c, R)$.

Esercizio: determinare le configurazioni prodotte da uq_i.
Occorre distinguere $u = \epsilon$ da $u \neq \epsilon$.

Siano C_1, C_2 due configurazioni di una MdT M.

Se C_1 produce C_2, scriveremo

$$C_1 \rightarrow C_2$$

La trasformazione $\rightarrow$ di C_1 in C_2 prende il nome di **passo di computazione**.

Corrisponde a un'applicazione della funzione di transizione di M.

Definizione

Una configurazione C si dice:

- **iniziale** *(con input w) se $C = q_0 w$, con $w \in \Sigma^*$,*
- **di arresto** *se $C = uqv$, con $u, v \in \Gamma^*$ e $q \in \{q_{accept}, q_{reject}\}$ (non esiste nessuna configurazione C' tale che $C \to C'$),*
- **di accettazione** *se $C = uqv$, con $u, v \in \Gamma^*$ e $q = q_{accept}$,*
- **di rifiuto** *se $C = uqv$, con $u, v \in \Gamma^*$ e $q = q_{reject}$.*

Nota: Le configurazioni di arresto sono configurazioni di accettazione o di rifiuto.

Definizione

Siano C, C' configurazioni.
$C \rightarrow^* C'$ *se esistono configurazioni* $C_1, \ldots, C_k$, $k \geq 1$ *tali che*

1. $C_1 = C$,
2. $C_i \rightarrow C_{i+1}$, *per* $i \in \{1, \ldots, k-1\}$,
   (ogni C_i produce C_{i+1})
3. $C_k = C'$.

Diremo che $C \rightarrow^* C'$ è una **computazione** *(di lunghezza $k-1$).*

Sia M una MdT e sia C una configurazione di M. Ci sono tre possibili casi:

1. $C \rightarrow^* C'$ con $C' = uq_{accept}v$ configurazione di accettazione (M si ferma in q_{accept}).
2. $C \rightarrow^* C'$ con $C' = uq_{reject}v$ configurazione di rifiuto (M si ferma in q_{reject}).
3. Per ogni configurazione C' tale che $C \rightarrow^* C'$ esiste una configurazione C'' tale che $C \rightarrow^* C' \rightarrow C''$ (M non si arresta).

Definizione

Una MdT M accetta una parola $w \in \Sigma^*$ se esiste una computazione $C \rightarrow^* C'$, dove $C = q_0 w$ è la configurazione iniziale di M con input w e $C' = uq_{accept}v$ è una configurazione di accettazione.

Quindi M accetta $w \in \Sigma^*$ se e solo se esistono configurazioni $C_1, C_2, \ldots, C_k$ di M, tali che

1. $C_1 = q_0 w$ è la configurazione iniziale di M con input w
2. $C_i \rightarrow C_{i+1}$ per ogni $i \in \{1, \ldots, k-1\}$
3. C_k è una configurazione di accettazione.

Una MdT M rifiuta una parola $w \in \Sigma^*$ se esiste una computazione $C \rightarrow^* C'$, dove $C = q_0 w$ è la configurazione iniziale di M con input w e $C' = u q_{reject} v$ è una configurazione di rifiuto.

Quindi M si ferma su $w \in \Sigma^*$ se M accetta w oppure rifiuta w, cioè se solo se esistono configurazioni $C_1, C_2, \ldots, C_k$ di M tali che

1. $C_1 = q_0 w$ è la configurazione iniziale di M con input w
2. $C_i \rightarrow C_{i+1}$ per ogni $i \in \{1, \ldots, k-1\}$
3. C_k è una configurazione di arresto.

Parola accettata da una MdT

Data una MdT M e una stringa $w \in \Sigma^*$,

1. M accetta w
   oppure
2. M non accetta w

Se M non accetta w, M rifiuta w OPPURE la computazione di M non si arresta su w.

Parola accettata da una MdT

Sia M una MdT. Sia $w = a_1 \dots a_n$, con $a_j \in \Sigma$. Supponiamo che w è accettata da M.

Quindi esistono $u, v \in \Gamma^*$ tali che

$$q_0 w \rightarrow^* u q_{accept} v.$$

Possiamo concludere che in questa computazione M **deve** aver letto tutti i caratteri a_j di w?
E quindi che la computazione deve avere lunghezza almeno $n = |w|$?

Esercizio: Si consideri il linguaggio $L = \{aw \mid w \in \{a, b\}^*\}$.

- Definire una macchina di Turing che accetta tutte e sole le stringhe di L ma che non si arresta su ogni input.
- Definire una macchina di Turing che accetta tutte e sole le stringhe di L, che si arresta su ogni input, con tre stati e senza cicli nel diagramma di stato.

Definizione

Sia $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$ una MdT. Il linguaggio $L(M)$ riconosciuto da M, è l'insieme delle stringhe che M accetta:

$$L(M) = \{w \in \Sigma^* \mid \exists u, v \in \Gamma^* \; q_0 w \to^* u q_{accept} v\}.$$

Quindi

$$L(M) = \{w \in \Sigma^* \mid M \text{ accetta } w\}.$$

A differenza del caso degli automi finiti, per i quali una stringa input è accettata o rifiutata, nel caso delle macchine di Turing c'è una terza possibilità: che per un dato input la computazione non termini.

Data una configurazione di una MdT che non è di arresto, non possiamo sapere se, a partire da tale configurazione, la computazione terminerà in qualche istante futuro o meno.

Sarebbe utile avere un algoritmo che, dato in input una MdT M e una stringa w, determini in un tempo finito se la computazione eseguita da M su w termini o meno (**Problema della fermata**).

Vedremo che un tale algoritmo non esiste.

Sia $R(M) = \{w \in \Sigma^* \mid M \text{ rifiuta } w\}$.

In generale $L(M) \cup R(M)$ non coincide con Σ^*.

Se $L(M) \cup R(M) = \Sigma^*$, allora M si arresta su ogni input.

In tal caso M è chiamata un decisore (o decider) ed $L(M)$ è il linguaggio deciso da M.

- Dal punto di vista delle macchine:

Definizione

Una MdT $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$ è un **decisore** *(o* **decider***) se, per ogni $w \in \Sigma^*$, esistono $u, v \in \Gamma^*$ e $q \in \{q_{accept}, q_{reject}\}$ tali che*

$$q_0 w \rightarrow^* uqv$$

Definizione

Un linguaggio L è **deciso** *da una MdT $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$ se M è un decisore ed $L = L(M)$.*

Linguaggi Turing riconoscibili e linguaggi decidibili

Dal punto di vista dei linguaggi occorre definire due famiglie di linguaggi:

Definizione

Un linguaggio $L \subseteq \Sigma^*$ è Turing riconoscibile se esiste una macchina di Turing $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$ tale che:

1. *M riconosce L*
   (cioè $L = L(M) = \{w \in \Sigma^* \mid \exists u, v \in \Gamma^* \; q_0 w \to^* u q_{accept} v\}$).

Definizione

Un linguaggio $L \subseteq \Sigma^*$ è decidibile se esiste una macchina di Turing $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$ tale che:

1. *M riconosce L*
   (cioè $L = L(M) = \{w \in \Sigma^* \mid \exists u, v \in \Gamma^* \; q_0 w \to^* u q_{accept} v\}$).
2. *M si arresta su ogni input (cioè per ogni $w \in \Sigma^*$, $q_0 w \to^* uqv$ con $q \in \{q_{accept}, q_{reject}\}$).*

Linguaggi Turing riconoscibili e linguaggi decidibili

Vedremo che l'insieme dei linguaggi decidibili è un sottoinsieme proprio dell'insieme dei linguaggi Turing riconoscibili.

Come conseguenza delle definizioni, un linguaggio L è Turing riconoscibile ma non decidibile se:

1. esiste una MdT che riconosce L (quindi accetta tutte e sole le stringhe di L)
2. non esiste nessuna MdT M tale che M accetta tutte le stringhe in L e rifiuta tutte le stringhe che appartengono al complemento $\overline{L}$ di L.

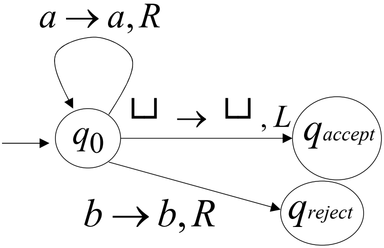
- Una MdT $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$ decide un linguaggio L se per ogni $w \in \Sigma^*$, $q_0 w \to^* C$ con $C = uqv$ configurazione di arresto ed $L = L(M)$.
- In questo caso diciamo che L è il linguaggio deciso da M.
- Nota: data una MdT M, esiste sempre il linguaggio riconosciuto da M ma non è detto che esista il linguaggio deciso da M, ciò è possibile se e solo se M è un **decider**.
- Se M è un decider, esiste il linguaggio deciso da M e coincide con il linguaggio riconosciuto da M.
- Se M non è un decider, esiste il linguaggio riconosciuto da M ma non il linguaggio deciso da M.

Non confondere la proprietà di un linguaggio (essere o non essere Turing riconoscibile, essere o non essere decidibile) con la proprietà di una MdT (essere o non essere un decider).

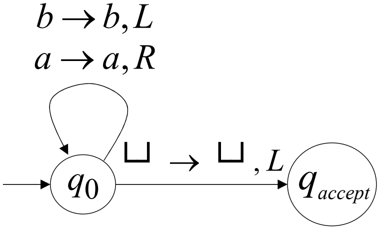
Esempio: $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$, con $Q = \{q_0, q_{accept}, q_{reject}\}$, $\Sigma = \{a, b\}$, $\Gamma = \{a, b, \sqcup\}$, $\delta(q_0, a) = (q_0, a, R)$, $\delta(q_0, b) = (q_0, b, L)$, $\delta(q_0, \sqcup) = (q_{accept}, \sqcup, L)$.

M non è un decider ma $L(M) = a^*$ è decidibile.

Una macchina di Turing che decide a^*



Una macchina di Turing che non è un decisore e che riconosce a^*



Le MdT possono essere utilizzate per il calcolo di funzioni.

Definizione

Una funzione $f : \Sigma^* \to \Sigma^*$ *è calcolabile se esiste una macchina di Turing* $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$ *tale che*

$$\forall w \in \Sigma^* \quad q_0 w \to^* q_{accept} f(w)$$

Quindi, una funzione $f : \Sigma^* \to \Sigma^*$ è calcolabile se esiste una macchina di Turing M che, su qualsiasi input w, si ferma avendo solo $f(w)$ sul nastro.

- Nota: in questo caso la MdT deve arrestarsi su ogni input.
- Possibile variante: nessun vincolo sulla posizione della testina nella configurazione di arresto.
- Possibile variante: nessuna distinzione tra q_{accept} e q_{reject}.

- Definire una macchina di Turing deterministica M che calcoli la funzione $f(x, y) = x + y$, con x, y interi positivi.
- Si assuma che l'input sia della forma

$$w0z$$

dove w è la rappresentazione unaria di x e z è la rappresentazione unaria di y.

- Si definisca M in modo che l'output sia della forma

$$wz0$$

Per esempio se $x = 3$ e $y = 2$, l'input sarà 111011 e M dovrà fermarsi con 111110 sul nastro.

Macchina di Turing per $f(x,y) = x + y$