Macchine di Turing

Corso di Fondamenti di Informatica - modulo 1
Corso di Laurea in Informatica
Università di Roma "Tor Vergata"
a.a. 2020-2021

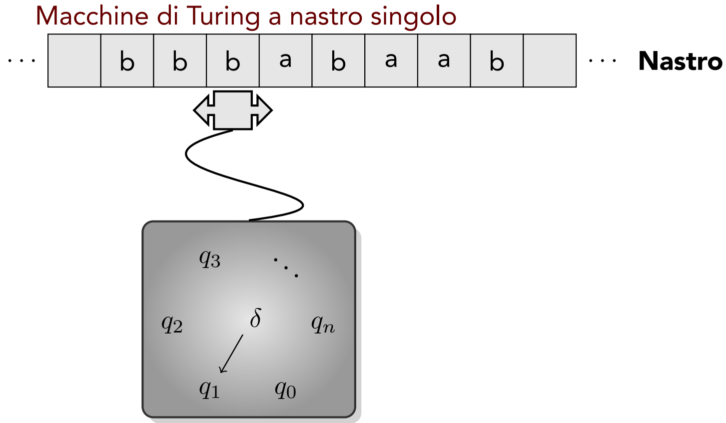
Giorgio Gambosi

Macchine di Turing a nastro singolo

Dispositivo che accede ad un *nastro* potenzialmente illimitato diviso in celle contenenti ciascuna un simbolo appartenente a un alfabeto Γ, ampliato con il carattere speciale $\square$ (blank) che rappresenta la situazione di cella non contenente caratteri.

All'inizio del calcolo solo una porzione finita del nastro contiene simboli di Γ. La macchina di Turing opera su tale nastro tramite una testina, la quale può scorrere su di esso in entrambe le direzioni Su ogni cella la testina può leggere o scrivere caratteri appartenenti all'alfabeto Γ oppure il simbolo $\square$.

Macchine di Turing a nastro singolo



Macchina di Turing deterministica

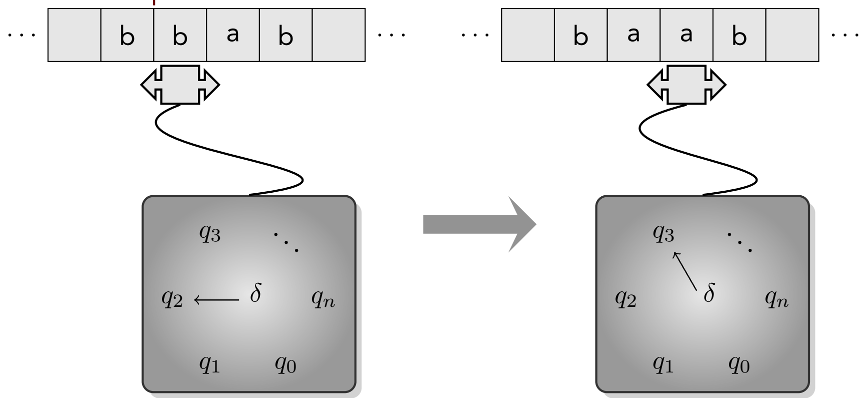
Sestupla $\mathcal{M} = \langle \Gamma, \square, Q, q_0, F, \delta \rangle$, dove :

- Γ: alfabeto dei simboli di nastro
- $\square \notin \Gamma$: carattere speciale denominato blank
- Q: insieme finito e non vuoto di stati
- $q_0 \in Q$: stato iniziale
- $F \subseteq Q$: insieme degli stati finali
- δ: funzione di transizione definita come

$$\delta : (Q - F) \times (\Gamma \cup \{\square\}) \mapsto Q \times (\Gamma \cup \{\square\}) \times \{\curvearrowright, \curvearrowleft, \circ\}$$

  in cui $\curvearrowright$, $\curvearrowleft$ e $\circ$ indicano, rispettivamente, lo spostamento a destra, lo spostamento a sinistra e l'immobilità della testina.

Automi a pila



Transizione determinata da $\delta(q_2, b) = (q_3, a, \curvearrowright)$

DTM

- DTM utilizzabili per calcolo di funzioni, o per riconoscere o accettare stringhe su un alfabeto di input $\Sigma \subseteq \Gamma$
- DTM usate per accettare stringhe vengono dette di tipo *riconoscitore*
- DTM usate per calcolare funzioni vengono dette di tipo *trasduttore*
- In entrambi i casi, all'inizio del calcolo, solo una porzione finita del nastro contiene simboli diversi da blank che costituiscono l'input del calcolo stesso

Configurazioni di DTM

Si definisce configurazione istantanea o configurazione di una macchina di Turing con alfabeto di nastro Γ ed insieme degli stati Q, una stringa $c = xqy$, con (assumendo $\bar{\Gamma} = \Gamma \cup \{\square\}$):

1. $x \in \Gamma\bar{\Gamma}^* \cup \{\varepsilon\}$
2. $q \in Q$
3. $y \in \bar{\Gamma}^*\Gamma \cup \{\square\}$

L'interpretazione data ad una stringa xqy è che xy rappresenti il contenuto della sezione non vuota del nastro, che lo stato attuale sia q e che la testina sia posizionata sul primo carattere di y. Nel caso in cui $x = \varepsilon$ abbiamo che a sinistra della testina compaiono solo simboli $\square$, mentre se $y = \square$ sulla cella attuale e a destra della testina compaiono soltanto simboli $\square$.

Configurazione iniziale

La configurazione iniziale di una MT rispetto a una stringa di input σ prevede che:

- il nastro contenga la stringa σ in una sequenza di celle
- tutte altre celle del nastro siano vuote (contengano $\square$)
- lo stato attuale sia lo stato iniziale q_0
- la testina si trovi sulla cella contenente il primo carattere di σ

Una configurazione xqy è quindi iniziale se $x = \varepsilon$, $q = q_0$, $y = \sigma$.

Configurazione finale

Una configurazione $c = xqy$, con si dice finale se $q \in F$.

Quindi, una macchina di Turing si trova in una configurazione finale se il suo stato attuale è uno stato finale, indipendentemente dal contenuto del nastro e dalla posizione della testina.

Matrice di transizione

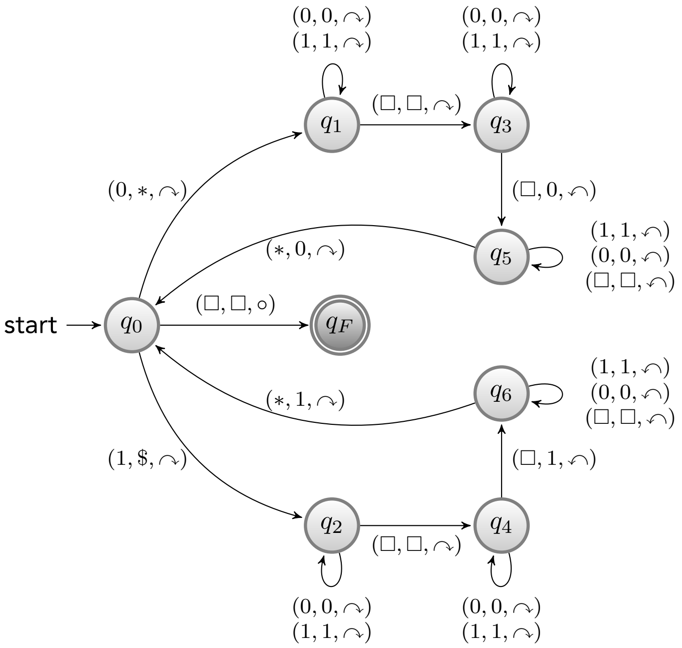
La funzione di transizione può essere rappresentata mediante matrici di transizione e grafi di transizione.

Esempio

	0	1	*	$	$\square$
q_0	$(q_1, *, \curvearrowright)$	$(q_2, \$, \curvearrowright)$	-	-	$(q_F, \square, \circ)$
q_1	$(q_1, 0, \curvearrowright)$	$(q_1, 1, \curvearrowright)$	-	-	$(q_3, \square, \curvearrowright)$
q_2	$(q_2, 0, \curvearrowright)$	$(q_2, 1, \curvearrowright)$	-	-	$(q_4, \square, \curvearrowright)$
q_3	$(q_3, 0, \curvearrowright)$	$(q_3, 1, \curvearrowright)$	-	-	$(q_5, 0, \curvearrowleft)$
q_4	$(q_4, 0, \curvearrowright)$	$(q_4, 1, \curvearrowright)$	-	-	$(q_6, 1, \curvearrowleft)$
q_5	$(q_5, 0, \curvearrowleft)$	$(q_5, 1, \curvearrowleft)$	$(q_0, 0, \curvearrowright)$	-	$(q_5, \square, \curvearrowleft)$
q_6	$(q_6, 0, \curvearrowleft)$	$(q_6, 1, \curvearrowleft)$	-	$(q_0, 1, \curvearrowright)$	$(q_6, \square, \curvearrowleft)$
q_F	-	-	-	-	-

In generale, assumiamo che uno stato finale non abbia transizioni uscenti definite.

Grafo di transizione



Esercizio

Considerata la macchina di Turing deterministica definita sopra e assumendo la configurazione iniziale $q_0 10$:

1. determinare la computazione effettuata dalla macchina, indicando la configurazione finale che viene raggiunta;
2. descrivere informalmente il comportamento della macchina su un input generico.

Accettazione e rifiuto di stringhe

- Computazione massimale: computazione che non può prolungarsi (non esistono transizioni applicabili alla configurazione raggiunta)
- Computazione di accettazione: computazione massimale che termina in una configurazione finale
- Computazione di rifiuto: computazione massimale che si conclude in una configurazione non finale

Dato un alfabeto di input $\Sigma \subseteq \Gamma$, una stringa $x \in \Sigma^*$ è *accettata* (*rifiutata*) da una MT $\mathcal{M}$ se esiste una computazione di accettazione (di rifiuto) di $\mathcal{M}$ con $c_0 = q_0 x$.

Accettazione e rifiuto di stringhe

- Terza possibilità: non esiste alcuna computazione massimale con $c_0 = q_0 x$; in altre parole, la computazione di $\mathcal{M}$ su input x non termina

Data un MT $\mathcal{M}$ con alfabeto di input Σ, l'insieme Σ^* è partizionato in tre linguaggi:

- L'insieme $L(\mathcal{M})$ delle stringhe *accettate* da $\mathcal{M}$
- L'insieme $\overline{L}(\mathcal{M})$ delle stringhe *rifiutate* da $\mathcal{M}$
- L'insieme $\Sigma^* - (L(\mathcal{M}) \cup \overline{L}(\mathcal{M}))$ delle stringhe sulle quali la computazione effettuata da $\mathcal{M}$ non termina

Definizioni equivalenti

1. esistono due soli stati finali q_1, q_2, tutte le computazioni massimali terminano in uno stato finale ed una stringa x è accettata se $q_0x \overset{*}{\underset{\mathcal{M}}{\vdash}} wq_1z$, mentre è rifiutata se $q_0x \overset{*}{\underset{\mathcal{M}}{\vdash}} wq_2z$
2. esiste un solo stato finale q_F, l'alfabeto di nastro contiene due simboli speciali $\mathcal{Y}, \mathcal{N} \notin \Sigma$, tutte le computazioni massimali terminano nello stato finale ed una stringa x è accettata se $q_0x \overset{*}{\underset{\mathcal{M}}{\vdash}} q_F\mathcal{Y}$, mentre è rifiutata se $q_0x \overset{*}{\underset{\mathcal{M}}{\vdash}} q_F\mathcal{N}$.

Riconoscimento di linguaggi

- Data una MT deterministica $\mathcal{M} = \langle \Gamma, \Box, Q, q_0, F, \delta \rangle$
- Dato un alfabeto di input $\Sigma \subseteq \Gamma$
- $\mathcal{M}$ riconosce (decide) un linguaggio $L \in \Sigma^*$ se e solo se per ogni $x \in \Sigma^*$:
  - esiste una computazione massimale $q_0x \overset{*}{\underset{\mathcal{M}}{\vdash}} wqz$
  - $q \in F$ se e solo se $x \in L$
  - $w \in \Gamma\bar{\Gamma}^* \cup \{\varepsilon\}$ e $z \in \bar{\Gamma}^*\Gamma \cup \{\Box\}$ rappresentano il contenuto delle porzioni di nastro significative prima e dopo la posizione della testina
- Affinché un linguaggio sia riconosciuto, $\mathcal{M}$ deve fermarsi per ogni $x \in \Sigma^*$

Accettazione di linguaggi

- Data una MT deterministica $\mathcal{M} = \langle \Gamma, \Box, Q, q_0, F, \delta \rangle$
- Dato un alfabeto di input $\Sigma \subseteq \Gamma$
- $\mathcal{M}$ accetta un linguaggio $L \in \Sigma^*$ se e solo se $L = \{x \in \Sigma^* \mid q_0x \overset{*}{\underset{\mathcal{M}}{\vdash}} wqz; q \in F\}$
- Quindi, L è l'insieme delle stringhe per le quali la computazione effettuata da $\mathcal{M}$ termina in uno stato finale
- Che succede per $x \notin L$? La computazione effettuata da $\mathcal{M}$ può:
  1. terminare in uno stato $q \in Q - F$
  2. non terminare

Esercizio

i) Definire una macchina di Turing deterministica che riconosce il linguaggio $L = \{w\tilde{w} \mid w \in \{a,b\}^+\}$.

ii) Definire una macchina di Turing deterministica che accetta il linguaggio L sopra definito e che per qualche stringa $x \in \{a,b\}^* - L$ cicla indefinitamente.

Turing-decidibilità

- Un linguaggio L è detto Turing-decidibile se esiste una macchina di Turing deterministica che lo riconosce
- Un linguaggio è detto Turing-semidecidibile se esiste una macchina di Turing deterministica che lo accetta.

MT a più nastri

Una MTM (multi-tape Turing machine) a k nastri ($k \geq 2$) è una sestupla $\mathcal{M}^{(k)} = \langle \Gamma, \Box, Q, q_0, F, \delta^{(k)} \rangle$ dove:

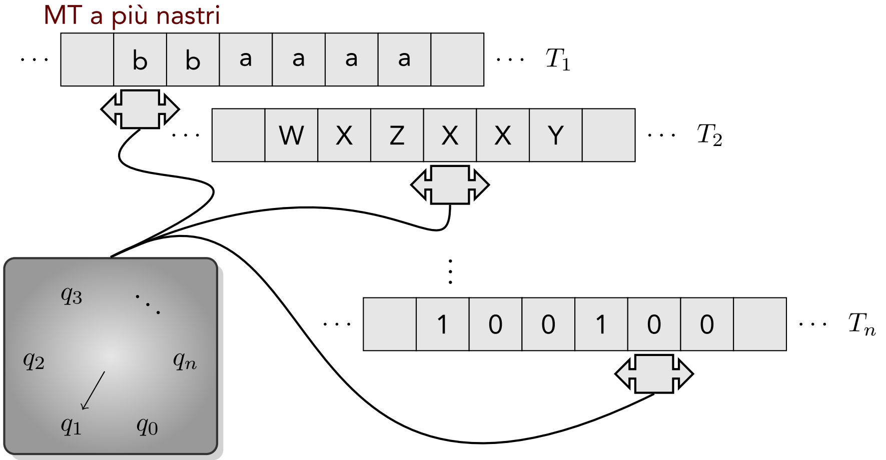
- $\Gamma = \bigcup_{i=1}^{k} \Gamma_i$ è l'unione dei k alfabeti di nastro $\Gamma_1, \ldots, \Gamma_k$ non necessariamente distinti
- Q, q_0 ed F hanno lo stesso significato che nel caso della macchina di Turing ad 1 nastro
- la funzione di transizione $\delta^{(k)}$ è definita come

$$\delta^{(k)} : (Q - F) \times \bar{\Gamma}_1 \times \ldots \times \bar{\Gamma}_k \mapsto Q \times \bar{\Gamma}_1 \times \ldots \times \bar{\Gamma}_k \times \{\curvearrowright, \curvearrowleft, \circ\}^k$$

MT a più nastri

- Una $\mathcal{M}$ esegue una transizione a partire da uno stato interno q_i e con le k testine — una per nastro — posizionate sui caratteri $a_{i_1}, \ldots, a_{i_k}$
- se $\delta^{(k)}(q_i, a_{i_1}, \ldots, a_{i_k}) = (q_j, a_{j_1}, \ldots, a_{j_k}, z_{j_1}, \ldots, z_{j_k})$
  - si porta nello stato q_j,
  - scrive i caratteri $a_{j_1}, \ldots, a_{j_k}$ sui rispettivi nastri
  - fa compiere alle testine i rispettivi spostamenti — a destra, a sinistra o nessuno spostamento, come specificato dagli $z_{j_\ell} \in \{\curvearrowright, \curvearrowleft, \circ\}$, $\ell = 1, \ldots, k$

MT a più nastri



Configurazioni di MTM

Una *configurazione istantanea* di una macchina di Turing multinastro può essere rappresentata da una stringa del tipo

$$q\,\#\,\alpha_1 \uparrow \beta_1\,\#\,\alpha_2 \uparrow \beta_2\,\#\,\ldots\,\#\,\alpha_k \uparrow \beta_k$$

- q è lo stato attuale
- il contenuto significativo del nastro T_k è $\alpha_k \cdot \beta_k$
- la testina del nastro T_k è posizionata sulla cella contenente il primo carattere di β_k

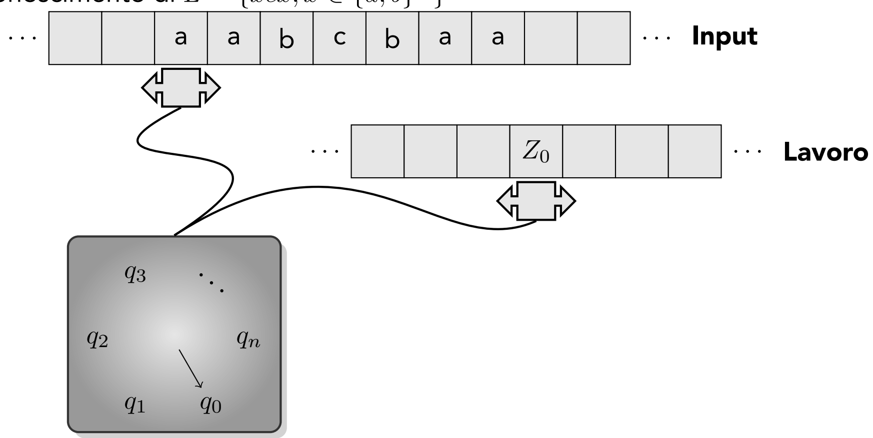
Configurazioni di MTM

Una configurazione di una MTM $q\,\#\,\alpha_1\uparrow\beta_1\,\#\,\alpha_2\uparrow\beta_2\,\#\,\dots\,\#\,\alpha_k\uparrow\beta_k$ è:

- finale se $q \in F$, quindi se lo stato attuale è finale, indipendentemente dal contenuto dei nastri
- iniziale (con stringa di input x) se $q = q_0$, $\alpha_i = \varepsilon, i = 1, \dots, k$, $\beta_1 = x$, $\beta_i = \Box, i = 2, \dots, k$, quindi se il primo nastro contiene l'input con la testina sul primo carattere, e gli altri nastri sono vuoti

Esempio di MTM

Riconoscimento di $L = \{xc\tilde{x}, x \in \{a, b\}^+\}$



Esempio di MTM

- Operazioni:
  1. input scandito da sx verso dx fino a quando si incontra il separatore c: simboli copiati sul nastro di lavoro da sx verso dx
  2. resto dell'input scandito da sx verso dx, nastro di lavoro scandito da dx verso sx, confrontano i caratteri in input con quelli presenti sul nastro di lavoro
- Alfabeto di input $\Sigma = \{a, b, c\}$
- Alfabeto del nastro di lavoro è $\Gamma = \{a, b\}$
- Configurazione iniziale: $q_0\,\#\uparrow xc\tilde{x}\,\#\uparrow\Box$.
- Tre stati: q_0 (scansione di x), q_1 (scansione di $\tilde{x}$), q_2, stato finale. Quindi $Q = \{q_0, q_1, q_2\}$ e $F = \{q_2\}$.

Esempio di MTM

Funzione di transizione:

- Lettura e copiatura di x: $\delta(q_0, a, \Box) = (q_0, a, a, \curvearrowright, \curvearrowright)$, $\delta(q_0, b, \Box) = (q_0, b, b, \curvearrowright, \curvearrowright)$
- Lettura separatore: $\delta(q_0, c, \Box) = (q_1, c, \Box, \curvearrowright, \curvearrowleft)$
- Lettura e verifica di $\tilde{x}$:
  - Caratteri uguali sui due nastri: $\delta(q_1, a, a) = (q_1, a, a, \curvearrowright, \curvearrowleft)$, $\delta(q_1, b, b) = (q_1, b, b, \curvearrowright, \curvearrowleft)$
  - Caratteri diversi sui due nastri: in questo caso la stringa non viene accettata. Nessuna transizione definita.

- Terminazione della verifica: $\delta(q_1, \Box, \Box) = (q_2, \Box, \Box, \circ, \circ)$

Esempio di MTM

Computazioni massimali corrispondenti ai due input $bacab$ e acb.

$$
\begin{array}{l}
q_0 \,\#\uparrow bacab \,\#\uparrow \Box \\
q_0 \,\# b\uparrow acab \,\# b\uparrow \Box \\
q_0 \,\# ba\uparrow cab \,\# ba\uparrow \Box \\
q_1 \,\# bac\uparrow ab \,\# b\uparrow a \\
q_1 \,\# baca\uparrow b \,\#\uparrow ba \\
q_1 \,\# bacab\uparrow \Box \,\#\uparrow \Box ba \\
q_2 \,\# bacab\Box\uparrow \Box \,\#\uparrow \Box ba
\end{array}
$$

$$
\begin{array}{l}
q_0 \,\#\uparrow acb \,\#\uparrow \Box \\
q_0 \,\# a\uparrow cb \,\# a\uparrow \Box \\
q_1 \,\# ac\uparrow b \,\#\uparrow a
\end{array}
$$

Equivalenza tra MTM e MT

È possibile dimostrare l'equivalenza tra MTM e MT:

- per ogni MT $\mathcal{M}$ esiste una MTM $\mathcal{M}'$ equivalente, tale cioé che $L(\mathcal{M}) = L(\mathcal{M}')$ (si tratta della stessa $\mathcal{M}$)
- per ogni MTM $\mathcal{M}$ esiste una MT $\mathcal{M}'$ equivalente, tale cioé che $L(\mathcal{M}) = L(\mathcal{M}')$
  - dimostrazione mediante *simulazione* di $\mathcal{M}'$ su $\mathcal{M}$: mostrando come ad ogni computazione di $\mathcal{M}$ corrisponda una computazione di $\mathcal{M}'$ con stesso esito (accettazione, rifiuto, non termina)

MT non deterministica

Una macchina di Turing non deterministica (NDTM) $\mathcal{M}$ a k nastri è una sestupla $\mathcal{M} = \langle \Gamma, \Box, Q, q_0, F, \delta_{\mathsf{N}} \rangle$, in cui:

- $\Gamma = \bigcup_{i=1}^{k} \Gamma_i$
- δ_{N} è una funzione parziale

$$\delta_{\mathsf{N}} : (Q - F) \times \bar{\Gamma}_1 \times \ldots \bar{\Gamma}_k \mapsto \mathcal{P}(Q \times \bar{\Gamma}_1 \times \ldots \times \bar{\Gamma}_k \times \{\curvearrowright, \curvearrowleft, \circ\}^k)$$

Esempio di NDTM

Consideriamo una macchina di Turing non deterministica $\mathcal{M}$ avente $\Gamma = \{a, b, c, d\}$, $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10},$ $F = \{q_{11}\}$ e funzione δ_{N} definita come segue:

	a	b	c	d	$\Box$
q_0	$\{(q_0, a, \curvearrowright), (q_1, c, \curvearrowright)\}$	$\{(q_0, b, \curvearrowright), (q_2, c, \curvearrowright)\}$	–	–	–
q_1	$\{(q_1, a, \curvearrowright), (q_3, d, \curvearrowleft)\}$	$\{(q_1, b, \curvearrowright)\}$	–	–	–
q_2	$\{(q_2, a, \curvearrowright)\}$	$\{(q_2, b, \curvearrowright), (q_3, d, \curvearrowleft)\}$	–	–	–
q_3	$\{(q_3, a, \curvearrowleft)\}$	$\{(q_3, b, \curvearrowleft)\}$	$\{(q_4, c, \curvearrowright)\}$	–	–
q_4	$\{(q_5, c, \curvearrowright)\}$	$\{(q_6, c, \curvearrowright)\}$	–	–	–
q_5	$\{(q_5, a, \curvearrowright)\}$	$\{(q_5, b, \curvearrowright)\}$	–	$\{(q_7, d, \curvearrowright)\}$	–
q_6	$\{(q_6, a, \curvearrowright)\}$	$\{(q_6, b, \curvearrowright)\}$	–	$\{(q_8, d, \curvearrowright)\}$	–
q_7	$\{(q_9, d, \curvearrowleft)\}$	–	–	$\{(q_7, d, \curvearrowright)\}$	–
q_8	–	$\{(q_9, d, \curvearrowleft)\}$	–	$\{(q_8, d, \curvearrowright)\}$	–
q_9	$\{(q_{10}, a, \curvearrowleft)\}$	$\{(q_{10}, b, \curvearrowleft)\}$	$\{(q_{11}, c, \circ)\}$	$\{(q_9, d, \curvearrowleft)\}$	–
q_{10}	$\{(q_{10}, a, \curvearrowleft)\}$	$\{(q_{10}, b, \curvearrowleft)\}$	$\{(q_4, c, \curvearrowright)\}$	–	–
q_{11}	–	–	–	–	–

Esempio di NDTM

La macchina di Turing $\mathcal{M}$

- ha grado di non determinismo 2
- data una stringa di input $x \in \{a, b\}^*$, la accetta se e solo se esiste una stringa $y \in \{a, b\}^*$ con $|y| \geq 2$ tale che $x = uyyv$, con $u, v \in \{a, b\}^*$

Esempio di computazioni di $\mathcal{M}$

Possibili computazioni su input $abab \in L(\mathcal{M})$

1. $q_0abab \longmapsto cq_1bab \longmapsto cbq_1ab \longmapsto cbaq_1b \longmapsto cbabq_1\square$
2. $q_0abab \longmapsto aq_0bab \longmapsto acq_2ab \longmapsto acaq_2b \longmapsto acabq_2\square$
3. $q_0abab \longmapsto aq_0bab \longmapsto abq_0ab \longmapsto abcq_1b \longmapsto abcbq_1\square$
4. $q_0abab \longmapsto aq_0bab \longmapsto abq_0ab \longmapsto abaq_0b \longmapsto abacq_2\square$
5. $q_0abab \longmapsto aq_0bab \longmapsto abq_0ab \longmapsto abaq_0b \longmapsto ababq_0\square$
6. $q_0abab \longmapsto aq_0bab \longmapsto acq_2ab \longmapsto acaq_2b \longmapsto acq_3ad \longmapsto aq_3cad \longmapsto acq_4ad \longmapsto accq_5d \longmapsto accdq_7\square$
7. $q_0abab \longmapsto cq_1bab \longmapsto cbq_1ab \longmapsto cq_3bdb \longmapsto q_3cbdb \longmapsto cq_4bdb \longmapsto ccq_6db \longmapsto ccdq_8b \longmapsto ccq_9dd \longmapsto cq_9cdd \longmapsto cq_{11}cdd$

Equivalenza tra MT ed MTND

È possibile dimostrare l'equivalenza tra MTND e MT:

- per ogni MT $\mathcal{M}$ esiste una MTND $\mathcal{M}'$ equivalente, tale cioé che $L(\mathcal{M}) = L(\mathcal{M}')$ (si tratta della stessa $\mathcal{M}$)
- per ogni MTND $\mathcal{M}$ esiste una MT $\mathcal{M}'$ equivalente, tale cioé che $L(\mathcal{M}) = L(\mathcal{M}')$
  - dimostrazione mediante *simulazione* di $\mathcal{M}'$ su $\mathcal{M}$: mostrando come ad ogni computazione di $\mathcal{M}$ corrisponda una computazione di $\mathcal{M}'$ con stesso esito (accettazione, rifiuto, non termina)

Linguaggi di tipo 0 e Macchine di Turing

Teorema 1. *Se $\mathcal{G}$ è una grammatica di tipo 0 e $L = L(\mathcal{G})$ è il linguaggio da essa generato, esiste una macchina di Turing non deterministica a due nastri $\mathcal{M}_L$ che accetta L.*

Linguaggi di tipo 0 e Macchine di Turing

Sia $\mathcal{G} = \langle V_N, V_T, P, S\rangle$, la macchina $\mathcal{M}_L$ opera nel seguente modo.

- Data una stringa $w \in V_T^*$, la configurazione iniziale di $\mathcal{M}_L$ è $q_0\#\uparrow w\#\uparrow S$.
- Ad ogni passo, in modo non deterministico $\mathcal{M}_L$ applica sulla forma di frase ϕ presente sul secondo nastro tutte le possibili produzioni in P, rimpiazzando ϕ con una nuova forma di frase ϕ' derivabile da ϕ. Quindi verifica se ϕ' coincide con w: solo se la verifica dà esito positivo la macchina entra in uno stato finale di accettazione.

Linguaggi di tipo 0 e Macchine di Turing

Corollario:

I linguaggi di tipo 0 sono Turing-semidecidibili

Linguaggi di tipo 0 e Macchine di Turing

Teorema 2. *Se $\mathcal{M}$ è una macchina di Turing che accetta il linguaggio L allora esiste una grammatica $\mathcal{G}_L$ di tipo 0 tale che $L = L(\mathcal{G}_L)$.*