Automi e Linguaggi Formali

Parte 12 – Varianti di macchine di Turing

Università degli Studi di Padova

Davide Bresolin

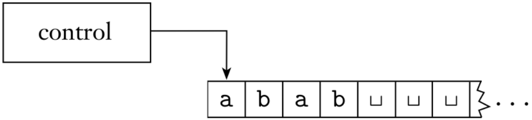
Ultimo aggiornamento: 25 aprile 2021

Sommario

Un modello robusto

- Esistono definizioni alternative delle macchine di Turing
- Chiamiamo varianti queste alternative
- Tutte le varianti "ragionevoli" riconoscono la stessa classe di linguaggi
- Le Turing machine sono un modello robusto

Macchine a nastro semi-infinito



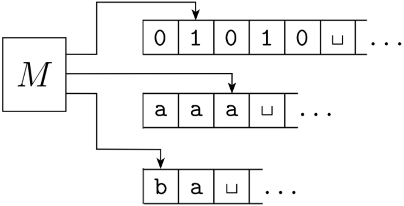
- è una TM con un nastro infinito solo verso destra
- l'input si trova all'inizio del nastro
- la testina parte dalla posizione più a sinistra del nastro
- se M tenta di spostare la testina a sinistra quando si trova nella prima cella del nastro, allora la testina rimane ferma

Macchine a nastro semi-infinito: equivalenza

Theorem

1. *Per ogni TM a nastro semi-infinito esiste una TM a nastro infinito equivalente.*
2. *Per ogni TM a nastro infinito esiste una TM a nastro semi-infinito equivalente.*

Macchine Multinastro



- è una TM con k nastri semi-infiniti
- k testine di lettura e scrittura
- l'input si trova sul nastro 1
- ad ogni passo scrive e si muove simultaneamente su tutti i nastri

- funzione di transizione:

$$\delta : Q \times \Gamma^k \mapsto Q \times \Gamma^k \times \{L, R\}^k$$

$$\delta(q_i, a_1, \ldots, a_k) = (q_j, b_1, \ldots, b_k, L, R, \ldots, L) :$$

- se lo stato è q_i e le testine leggono $a_1, \ldots, a_k$
- allora scrivi $b_1, \ldots, b_k$ sui k nastri
- muovi ogni testina a sinistra o a destra come specificato

Theorem

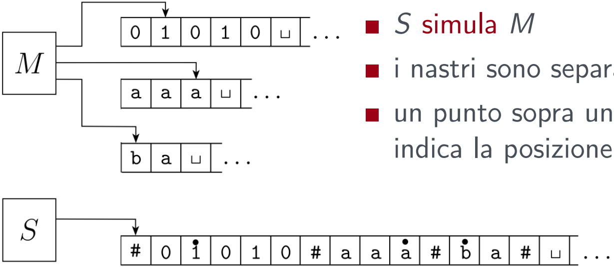
Per ogni TM multinastro esiste una TM a singolo nastro equivalente.

Macchine Multinastro: equivalenza

Theorem

Per ogni TM multinastro esiste una TM a singolo nastro equivalente.

Idea:



- *S* simula *M*
- i nastri sono separati da #
- un punto sopra un simbolo indica la posizione della testina

$S =$ "Su input $w = w_1 \dots w_n$:

Macchine Multinastro: equivalenza

$S =$ "Su input $w = w_1 \dots w_n$:

1. Inizializza il nastro per rappresentare i k nastri:

$$\#\dot{w}_1 w_2 \dots w_n \# \dot{\sqcup} \# \dot{\sqcup} \# \dots \#$$

Macchine Multinastro: equivalenza

$S =$ "Su input $w = w_1 \ldots w_n$:

1. Inizializza il nastro per rappresentare i k nastri:

$$\# \dot{w}_1 w_2 \ldots w_n \# \dot{\sqcup} \# \dot{\sqcup} \# \ldots \#$$

2. Per simulare una mossa di M, scorri il nastro per determinare i simboli puntati dalle testine virtuali

Macchine Multinastro: equivalenza

$S =$ "Su input $w = w_1 \dots w_n$:

1. Inizializza il nastro per rappresentare i k nastri:

$$\# \dot{w}_1 w_2 \dots w_n \# \dot{\sqcup} \# \dot{\sqcup} \# \dots \#$$

2. Per simulare una mossa di M, scorri il nastro per determinare i simboli puntati dalle testine virtuali
3. Fai un secondo passaggio del nastro per aggiornare i nastri virtuali secondo la funzione di transizione di M

Macchine Multinastro: equivalenza

$S =$ "Su input $w = w_1 \dots w_n$:

1. Inizializza il nastro per rappresentare i k nastri:

$$\# \dot{w}_1 w_2 \dots w_n \# \dot{\sqcup} \# \dot{\sqcup} \# \dots \#$$

2. Per simulare una mossa di M, scorri il nastro per determinare i simboli puntati dalle testine virtuali
3. Fai un secondo passaggio del nastro per aggiornare i nastri virtuali secondo la funzione di transizione di M
4. Se S sposta una testina virtuale a destra su un $\#$, allora M ha spostato la testina sulla parte vuota del nastro. Scrivi un $\sqcup$ e sposta il contenuto del nastro di una cella a destra

Macchine Multinastro: equivalenza

$S =$ "Su input $w = w_1 \dots w_n$:

1. Inizializza il nastro per rappresentare i k nastri:

$$\# \dot{w}_1 w_2 \dots w_n \# \dot{\sqcup} \# \dot{\sqcup} \# \dots \#$$

2. Per simulare una mossa di M, scorri il nastro per determinare i simboli puntati dalle testine virtuali
3. Fai un secondo passaggio del nastro per aggiornare i nastri virtuali secondo la funzione di transizione di M
4. Se S sposta una testina virtuale a destra su un $\#$, allora M ha spostato la testina sulla parte vuota del nastro. Scrivi un $\sqcup$ e sposta il contenuto del nastro di una cella a destra
5. Ripeti da 2

Macchine Multinastro: conclusione

Corollary

Un linguaggio è Turing-riconoscibile se e solo se esiste una macchina di Turing multinastro che lo riconosce.

Macchine Multinastro: conclusione

Corollary

Un linguaggio è Turing-riconoscibile se e solo se esiste una macchina di Turing multinastro che lo riconosce.

$\Rightarrow$ Un linguaggio è Turing-riconoscibile se è riconosciuto da una TM con un solo nastro, che è un caso particolare di TM multinastro

Macchine Multinastro: conclusione

Corollary

Un linguaggio è Turing-riconoscibile se e solo se esiste una macchina di Turing multinastro che lo riconosce.

- $\Rightarrow$ Un linguaggio è Turing-riconoscibile se è riconosciuto da una TM con un solo nastro, che è un caso particolare di TM multinastro
- $\Leftarrow$ Costruzione precedente

Macchine non deterministiche

- Una TM non deterministica ha più strade possibili durante la computazione
- Consideriamo macchine con un solo nastro semi-infinito
- La funzione di transizione è:

$$\delta : Q \times \Gamma \mapsto 2^{(Q \times \Gamma \times \{L, R\})}$$

- la computazione è un albero che descrive le scelte possibili
- la macchina accetta se esiste un ramo che porta allo stato di accettazione

Macchine non determinische: equivalenza

Università degli Studi di Padova

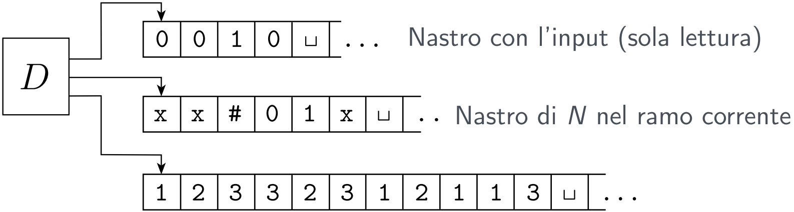
Theorem

Per ogni TM non deterministica esiste una TM deterministica equivalente.

Theorem

Per ogni TM non deterministica esiste una TM deterministica equivalente.

Idea:



Il terzo nastro tiene traccia delle scelte non deterministiche

Corollary

Un linguaggio è Turing-riconoscibile se e solo se esiste una macchina di Turing non deterministca che lo riconosce.

Macchine non determinisitche: conclusione

Corollary

Un linguaggio è Turing-riconoscibile se e solo se esiste una macchina di Turing non deterministca che lo riconosce.

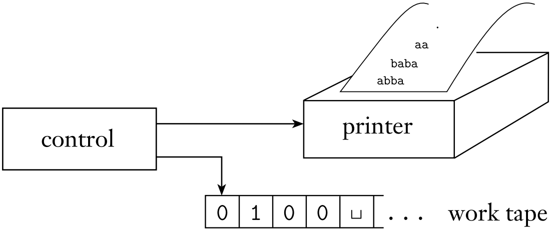
$\Rightarrow$ Un linguaggio è Turing-riconoscibile se è riconosciuto da una TM deterministica, che è un caso particolare di TM non deterministica

Corollary

Un linguaggio è Turing-riconoscibile se e solo se esiste una macchina di Turing non deterministca che lo riconosce.

- $\Rightarrow$ Un linguaggio è Turing-riconoscibile se è riconosciuto da una TM deterministica, che è un caso particolare di TM non deterministica
- $\Leftarrow$ Costruzione precedente

Enumeratori



- Enumeratore: macchina di Turing + stampante
- Un enumeratore E inizia con nastro vuoto
- Di tanto in tanto, invia una striga alla stampante
- Linguaggio enumerato da E: tutte le stringhe stampate
- E può generare le stringhe in qualsiasi ordine, anche con ripetizioni

Enumeratori: equivalenza

Università degli Studi di Padova

Theorem

Un linguaggio è Turing-riconoscibile se e solo se esiste un enumeratore che lo enumera

Enumeratori: equivalenza

Università degli Studi di Padova

Theorem

Un linguaggio è Turing-riconoscibile se e solo se esiste un enumeratore che lo enumera

Idea: dobbiamo mostrare che

- se esiste un enumeratore E, allora esiste una TM M che riconosce lo stesso linguaggio
- se esiste una TM M che riconosce il linguaggio, allora possiamo costruire un enumeratore

- Una macchina di Turing con "resta ferma" invece di "muovi a sinistra"
- Funzione di transizione: $\delta : Q \times \Gamma \mapsto Q \times \Gamma\{S, R\}$
- Ad ogni passo, la TM può lasciare ferma la testina o muoverla a destra
- Non può muoversi a sinistra!

Equivalenza con altri modelli

- Esistono altri modelli di computazione universali
- Alcuni sono molto simili alle macchine di Turing
- Altri sono molto diversi
- Hanno tutti una caratteristica comune:
  - accesso senza restrizioni ad una memoria illimitata
- Sono tutti equivalenti tra loro!