La macchina di Turing Universale

Fabio Zanasi[1]

[1] *Università di Bologna*

Abstract

Una Macchina di Turing Universale è in grado di emulare il comportamento di una qualsiasi macchina di Turing codificata sul suo nastro.
Keywords: UTM, MTU, Turing

Contents

1 Descrizione

1.1 Struttura

La macchina di Turing universale ($\mathfrak{U}$) è in grado di emulare una qualsiasi macchina di Turing ($\mathfrak{M}$) il cui insieme di simboli consiste di tre soli simboli : ␣(blank), **0** e **1**.

La configurazione corrente della macchina ($\mathfrak{M}$) è codificata sul nastro di ($\mathfrak{U}$) come segue:

$$\ldots \;␣\;␣\; \textit{sinistra-nastro} \;[\; \hat{\mathfrak{M}} \;]\; \boxdot \; \textit{destra-nastro} \;␣\;␣\; \ldots$$

- *sinistra-nastro* è il contenuto del nastro di $\mathfrak{M}$ alla sinistra della cella corrente.
- $\hat{\mathfrak{M}}$ è la codifica (notare ˆ!) delle regole di transizione per $\mathfrak{M}$, ove è segnato lo stato corrente.
- $\boxdot$ è il contenuto della cella corrente di $\mathfrak{M}$.
- *destra-nastro* è il contenuto del nastro di $\mathfrak{M}$ alla destra della cella corrente.

1.2 La codifica delle regole di transizione per $\mathfrak{M}$

Gli stati di $\mathfrak{M}$ sono numerati q_1 , . . . , q_m. Nel seguito denotiamo con $\hat{q}_i$ la codifica dello stato q_1 (notare ˆ!) . Se q_C è lo stato corrente, La codifica delle regole di transizione per $\mathfrak{M}$ ha la forma:

$$\hat{q}_1 : \hat{q}_2 : \ldots\; \hat{q}_{C-1} : \hat{q}_C \;!\; \hat{q}_{C+1} : \ldots\; \hat{q}_m$$

dove $\hat{q}_i$ codifica (notare ˆ!) la regola di transizione per lo stato $\hat{q}_i$, la codifica di ogni stato termina con il i due punti (:) eccetto che per lo stato corrente che è individuato per terminare con un punto esclamativo.

Data la regola di transizione per lo stato $\hat{q}_i$:

- $q_{i,␣} \rightarrow s_j, D_j, q_j$

- q_i,0 $\rightarrow s_k,D_k,q_k$
- q_i,1 $\rightarrow s_l,D_l,q_l$

dove $s_j,s_k,s_l \in \{ ␣,0,1\}$, $D_j,D_k,D_k \} \in \{$ **L**,**R** $\}$ (la testina si muove **sempre**) e $1 \leqslant$ j,k,l $\leqslant$ m (numero degli stati), allora
la codifica $\hat{q_i}$ ha la forma

$$s_j\ D_j\ \sigma_{j-i}\ ,\ s_k\ D_k\ \sigma_{k-i}\ ,\ s_l\ D_l\ \sigma_{l-i}$$

dove le tre parti della codifica della regola di transizione sono separate dalla virgola (,) e

$$\sigma_\delta = \begin{cases} . & se\ \delta = 0, \\ +^\delta & se\ \delta > 0, \\ -^{|\delta|} & se\ \delta < 0, \end{cases}$$

gli esponenti (p.e. δ in $+^\delta$) denotano ripetizione. Così una sequenza di $|\delta|$ simboli (+) o (-) codificano il relativo cambiamento nel numero di stato, il punto (.) denota nessun cambiamento di stato. Se non c'è una regola di transizione per (q_i,s) (interpretata come una configurazione di arresto), allora la codifica $\hat{q_i}$ è la stringa vuota.
La dimensione di $\hat{\mathfrak{M}}$ è O(m^2), dove m è il numero di stati di $\mathfrak{M}$.

1.3 Esempio

La macchina di Turing $\mathfrak{M}$ a due stati per incrementare un numero in formato binario :

- q_1,␣ $\rightarrow$ ␣,**L**,q_2
- q_1,0 $\rightarrow$ **0**,**R**,q_1
- q_1,1 $\rightarrow$ **1**,**R**,q_1

- q_2,␣ $\rightarrow$ **1**,**L**,q_3
- q_2,0 $\rightarrow$ **1**,**L**,q_3
- q_2,1 $\rightarrow$ **0**,**L**,q_2

- q_3,␣ $\rightarrow$ ␣ ,**L**,$halt$
- q_3,0 $\rightarrow$ **0**,**L**,q_3
- q_3,1 $\rightarrow$ **1**,**L**,q_3

con stato iniziale q_1, input iniziale **1011**, ha questa codifica $\hat{\mathfrak{M}}$

[**L**+, **0R**., **1R**.!**1L**+, **1L**+, **0L**. :, **0L**., **1L**. :]**1011**

2 Implementazione

2.1 Insieme dei simboli

$\mathfrak{U}$ fa uso di un insieme di 16 simboli { ␣ **0** **1** [ ] , : **!** **L** **R** . − + # < > }

2.2 Algoritmo

$\mathfrak{U}$ inizia posizionandosi alla prima cella a destra di] , ove si trova il primo simbolo di input di $\mathfrak{M}$.

1. Legge il simbolo, s, dalla cella corrente.
2. Trova lo stato corrente, muovendosi a sinistra fino a quando non incontra il simbolo !
3. Trova la regola di transizione corrente muovendosi a sinistra oltre 2, 1 o 0 virgole (,) se s è ␣ **0** **1** rispettivamente.
4. Se la regola corrente è vuota , $\mathfrak{U}$ si ferma.
5. Altrimenti, modifica la codifica dello stato corrente, cambiando il punto . in #, ogni $-$ in $<$ e ogni $+$ in $<$
6. Legge il simbolo da scrivere dalla regola corrente , si muove verso destra e lo scrive nella cella corrente di $\mathfrak{M}$
7. Muove a sinistra alla regola corrente (riconoscibile dalla presenza di #, $<$ $>$) e legge la direzione di spostamento (**L** o **R**)
8. Sposta l'intera codifica di $\mathfrak{M}$ a sinistra o a destra di una cella, poi aggiorna la cella corrente di $\mathfrak{M}$.
9. Cambia lo stato se necessario sostituendo ogni $<$ con $-$ o ogni $>$ con $+$, e ogni volta , muovendo il simbolo ! alla posizione del successivo : alla sinistra o alla destra rispettivamente. Se non è richiesto un cambiamento di stato, il simbolo # è sostituito con un .
10. Muove a destra alla cella corrente di $\mathfrak{M}$ per ricominciare dal passo 1.