

Documentazione: Universal Turing Machine

Stefano Bavaro

June 3, 2021

1 Introduzione alla Universal Turing Machine

La macchina di Turing Universale è una macchina riprogrammabile che simula ogni altra macchina di Turing M . L'input della macchina di Turing Universale è quindi la descrizione della macchina M da simulare e la stringa che M deve computare.

Per descrivere la macchina M da simulare, sarà necessario trovare un modo per codificarla (in particolare, per codificare le sue transizioni). Tale metodo di codifica è soggettivo, in quanto l'unico fattore rilevante è che la codifica venga riconosciuta dalla Universal Turing Machine; infatti, poichè l'alfabeto delle macchine di Turing può essere scelto arbitrariamente, si può scegliere un qualsiasi metodo di codifica.

2 Codifica scelta

Seppur, come detto nella sezione precedente, la codifica di una Macchina di Turing sia arbitraria, bisogna osservare che all'aumentare dei caratteri nell'alfabeto della Universal Turing Machine (e quindi all'aumentare dei caratteri utilizzati per la codifica) aumenta anche la difficoltà della UTM stessa.

Per questo motivo, in continuità con quanto visto a lezione, si è scelto di utilizzare la codifica binaria, dove ogni carattere/stato/operazione viene indicato in notazione unaria e i caratteri separatori sono invece rappresentati da 0: un solo 0 per separare all'interno della transizione, doppio 0 (00) per separare tra diverse transizioni.

Le regole utilizzate per la codifica sono quindi:

1. Per la codifica dei caratteri, bisogna considerare che si è scelto di codificare $\text{blank} = \lambda = 1$. Quindi ogni codifica di caratteri effettivi potrà avere un numero di 1 maggiore o uguale a due.
(ES: $\text{Blank} = 1$, $0 = 11$, $1 = 111$, $2 = 1111\dots$)
2. Per la codifica degli stati, non c'è nessuna restrizione particolare: la codifica unaria può comprendere un qualsiasi numero di 1.
(ES: $q_0 = 1$, $q_1 = 11$, $q_2 = 111\dots$)
3. Per la codifica delle operazioni, si è scelto: $L = 1$, $R = 11$, $S = 111$.

3 Funzionamento della Universal Turing Machine

3.1 Formato dell'input

Si è organizzata la UTM in tre nastri:

1. Sul primo nastro viene inserita la codifica della macchina di Turing che si vuole eseguire.
2. Sul secondo nastro viene inserita la stringa che la macchina descritta sul primo nastro deve computare: ogni carattere codificato dev'essere separato dal successivo con un separatore 0. È permesso anche non inserire nulla nel secondo nastro, il che verrà considerato come la stringa vuota λ .
3. Sul terzo nastro viene inserito lo stato iniziale della macchina di Turing descritta sul primo nastro.

3.2 Algoritmo utilizzato

Inizialmente si verifica se il secondo nastro sia vuoto, ovvero se si è scelto di inserire la stringa vuota in input: in tal caso viene automaticamente inserita la codifica di λ sul secondo nastro e si procede allora alla normale computazione (una soluzione simile è adottata anche per casi limite delle operazioni R/L/S). Si consideri la prima funzione di transizione descritta sul primo nastro: sia, a titolo di esempio, $\delta(q_i, a) = (q_j, b, R)$.

La computazione si articola in 5 differenti "fasi":

1. Si controlla che lo stato q_i sul primo nastro sia uguale allo stato indicato sul terzo nastro, contandone il numero di 1; se gli stati non coincidono, si entra nel "blocco" in cui si passa alla successiva funzione di transizione sul primo nastro, e si ricomincia la computazione (considerando ovviamente tale nuova funzione di transizione); altrimenti si prosegue.
2. Si controlla che il carattere a sul primo nastro sia uguale al carattere "puntato" sul secondo nastro, contandone il numero di 1: se i caratteri non coincidono, si entra nel "blocco" in cui si passa alla successiva funzione di transizione sul primo nastro, e si ricomincia la computazione (considerando ovviamente tale nuova funzione di transizione); altrimenti si procede.
3. Si passa ad osservare lo stato "d'arrivo" q_j della funzione di transizione. Viene dunque cancellato lo stato attuale sul terzo nastro e sostituito con quello indicato dalla funzione di transizione sul primo nastro (q_j).
4. Bisogna ora sostituire il carattere "puntato" sul secondo nastro col carattere b indicato dalla funzione di transizione. Si confronta il numero di 1, e dipendentemente dal risultato si potranno avere diversi cammini possibili:
 - (a) Il numero di 1 coincide, quindi i caratteri sono uguali e il secondo nastro non viene modificato;

- (b) Il numero di 1 è maggiore sul primo nastro (bisognerà quindi aggiungerne sul secondo nastro) e il carattere "puntato" sul secondo nastro è l'ultimo della stringa: allora si potranno semplicemente aggiungere gli 1 necessari in coda alla stringa, senza effettuare traslazioni.
- (c) Il numero di 1 è maggiore sul primo nastro (bisognerà quindi aggiungerne sul secondo nastro) e il carattere "puntato" sul secondo nastro non è l'ultimo della stringa: allora bisognerà "traslare" a destra tutto il seguito della stringa sul secondo nastro per fare spazio agli 1 aggiuntivi.

Per fare ciò, si è scelto di copiare sul terzo nastro (utilizzato come appoggio ma lasciando invariato lo stato ivi presente) la parte di stringa da "traslare", inserire gli 1 necessari sul secondo nastro e poi ricopiare la parte di stringa "salvata" sul terzo nastro nuovamente sul secondo nastro. Si sarebbe potuta svolgere tale operazione interamente sul secondo nastro, ma per semplice questione di comodità si è preferito utilizzare un nastro di appoggio, facendo attenzione che le informazioni lì contenute non venissero modificate.

- (d) Il numero di 1 è minore sul primo nastro (bisognerà quindi eliminarne sul secondo nastro) e il carattere "puntato" sul secondo nastro è l'ultimo della stringa: allora si potranno semplicemente eliminare gli 1 in eccesso in coda alla stringa, senza effettuare alcuna traslazione.
- (e) Il numero di 1 è minore sul primo nastro (bisognerà quindi eliminarne sul secondo nastro) e il carattere "puntato" sul secondo nastro non è l'ultimo della stringa: allora bisognerà eliminare gli 1 in eccesso e "traslare" a sinistra tutto il seguito della stringa sul secondo nastro per non lasciare caratteri blank nel mezzo.

In continuità con quanto spiegato per il passo (c), anche in questo caso si è scelto di utilizzare il terzo nastro come appoggio, copiando lì la sottostringa da traslare a sinistra, eliminando poi gli 1 in eccesso dal carattere "puntato" e ricopiando sul secondo nastro quanto presente sul terzo.

5. Infine bisogna eseguire l'operazione di spostamento sul nastro indicata dalla funzione di transizione. A seconda della tipologia di operazione (R/L/S), ci si sposta sul secondo nastro all'inizio della codifica del nuovo carattere da "puntare".

Casi particolari si hanno quando, muovendosi a destra o a sinistra, non esistono già caratteri "da puntare", ma si hanno dei blank (quindi si stava considerando o il primo o l'ultimo carattere sulla stringa). In tal caso, semplicemente viene aggiunta (rispettivamente a destra o a sinistra) la codifica (ovvero 1) del carattere blank (λ), in modo da poter proseguire correttamente la computazione.

Al termine di questi passaggi, la UTM si riposiziona all'inizio del primo nastro per esaminare nuovamente tutte le funzioni di transizione e termina se nessuna di esse soddisfa le condizioni sul secondo e terzo nastro.

3.3 Esempio di una semplice MdT

A titolo di esempio, si fornisce la macchina di Turing $M = \{Q, \Sigma, \Gamma, \delta, q_0, \square, F\}$ che scambia tutti gli 0 di una stringa in 1 e viceversa, solo per stringhe che alternano caratteri 0 e 1.

La funzione δ sarà: $\delta(q_0, 0) = (q_1, 1, R)$ e $\delta(q_1, 1) = (q_0, 0, R)$; la sua codifica sarà quindi 101101101110110011011101011011.

Se si pone come input la stringa 01010, questa sarà codificata quindi come 1101110110111011; poichè q_0 è lo stato iniziale di M, la sua codifica da inserire nel terzo nastro sarà 1.

Inseriti tali input, la UTM restituisce, correttamente, sul terzo nastro, la stringa 111011011101101101, ovvero la stringa correttamente invertita con, al termine, il carattere blank.