Sommario

- Macchina di Turing universale
- Proprietà dei problemi Turing riconoscibili
- Linguaggi non Turing riconoscibili.

UTM: la TM universale

Una TM T che accetta un linguaggio è analoga a un programma che implementa un algoritmo

Ma abbiamo affermato che le TM forniscono un modello di calcolo, cioè un modello astratto di un calcolatore con un linguaggio di programmazione.

Ora faremo vedere come costruire la UTM U, una TM che prende in input la codifica di una TM T e di un suo input w e e la esegue producendo lo stesso risultato di T su w.

Codifica di una TM

- **Per una TM T=$(Q,\Sigma,\Gamma,\delta,q_0,q_a,q_r)$ possiamo assumere :**
- $\Sigma = \{1,2, \ldots, |\Sigma|\}$ **e**
- $\Gamma - \Sigma = \{|\Sigma| + 1, \ldots , |\Gamma|\}$
- $Q = \{|\Gamma| + 1, \ldots , |\Gamma| + |Q|\}$
- $|\Gamma| + 1, |\Gamma| + 2, |\Gamma| + 3$ **sono rispettivamente gli stati q_0,q_a,q_r**
- **Tutti i numeri sono codificati come numeri binari di lunghezza $\lceil \log(|Q| + |\Gamma|) \rceil$**

Fine codifica di una TM con un suo input

La codifica della TM T comincerà con

il numero $|Q|$ in binario, seguito da $|\Sigma|$ e da $|\Gamma|$, separati da virgole, seguiti da

una descrizione di δ in termini di quintuple

$((q,a),(p,b,D))$, con D in $\{L,R\}$ seguiti da

un ; che termina la descrizione di T e la separa dalla

codifica in binario della parola input $x = x_1 \ldots x_k$, con la virgola ancora usata come separatore degli interi binari che codificano i singoli simboli.

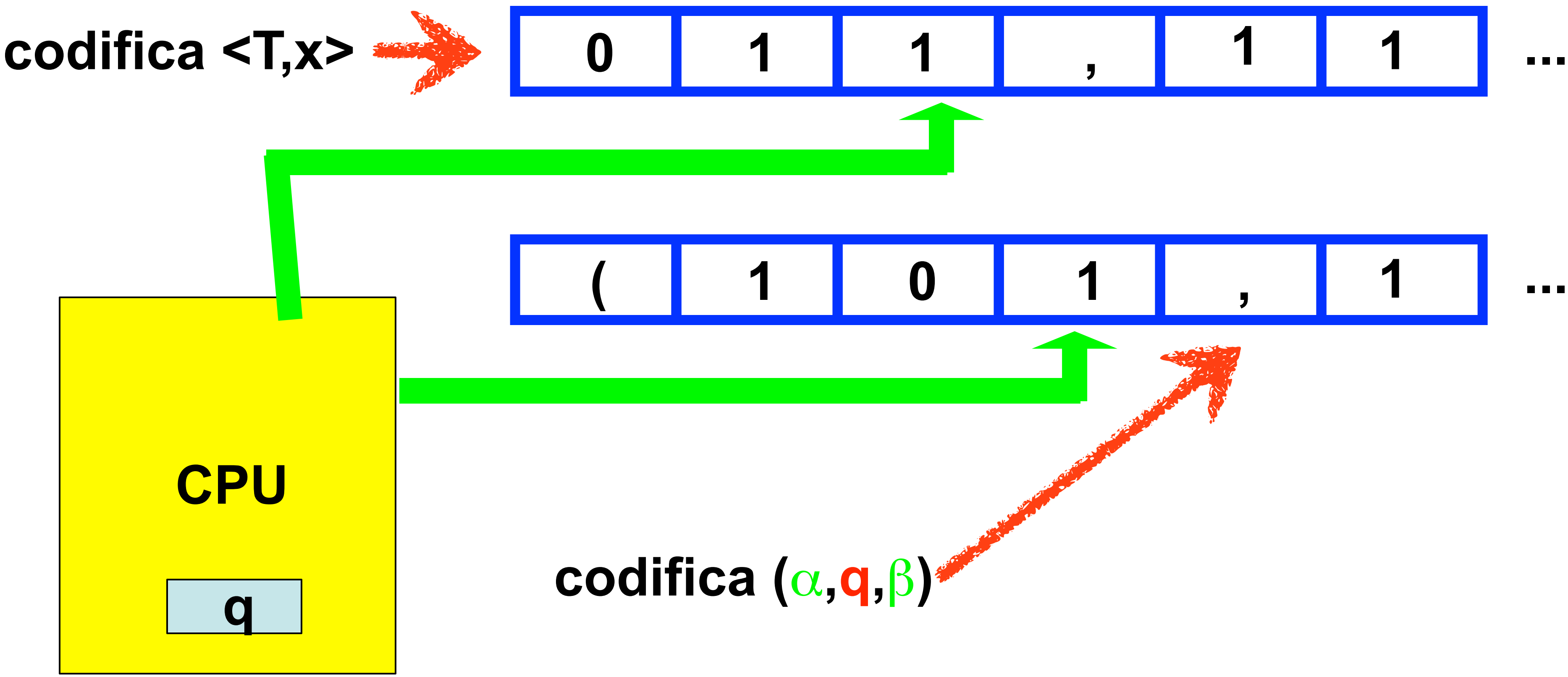
Costruzione UTM: i nastri

La UTM U sull'input <T,x_1 ... x_k > ha due nastri:

- **il primo contiene l'input <T,x_1 ... x_k>**
- **il secondo nastro contiene la (codifica della) configurazione corrente di T, nella forma (α,q,β) dove $\alpha\beta$ è il contenuto del nastro di T a quel punto del suo calcolo, q è lo stato in cui si trova e il simbolo in lettura è il primo di β.**

I primi passi della UTM, dopo aver verificato che l'input è una codifica corretta, servono per scrivere sul secondo nastro la codifica della configurazione iniziale, (q_0,x_1...x_k)

Macchina di Turing universale



Costruzione UTM

Per simulare un passo di T:

- **U esamina il secondo nastro fino a trovare la codifica binaria dello stato corrente q (è un numero tra $|\Gamma| + 1$ e $|\Gamma| + |Q|$)**

- **cerca sul primo nastro la prima regola per q**

- **poi muove la testina del secondo nastro per individuare il simbolo in lettura per T e controlla se la regola in lettura sul primo nastro coinvolge lo stesso simbolo input; se sì la regola viene implementata (cambiando la configurazione sul secondo nastro in corrispondenza) altrimenti si controlla la regola successiva**

Costruzione UTM - fine

U si ferma e accetta o rifiuta quando T accetta o rifiuta, se T non si ferma anche U non si ferma.

Un problema semidecidibile ma non decidibile

Problema dell'accettazione per TM: data una TM T e una parola input w per T, w è accettata da T?

Abbiamo dimostrato che non esiste un algoritmo che, data una TM T e una parola input w per T, risponde sì se w è accettata da T e no altrimenti.

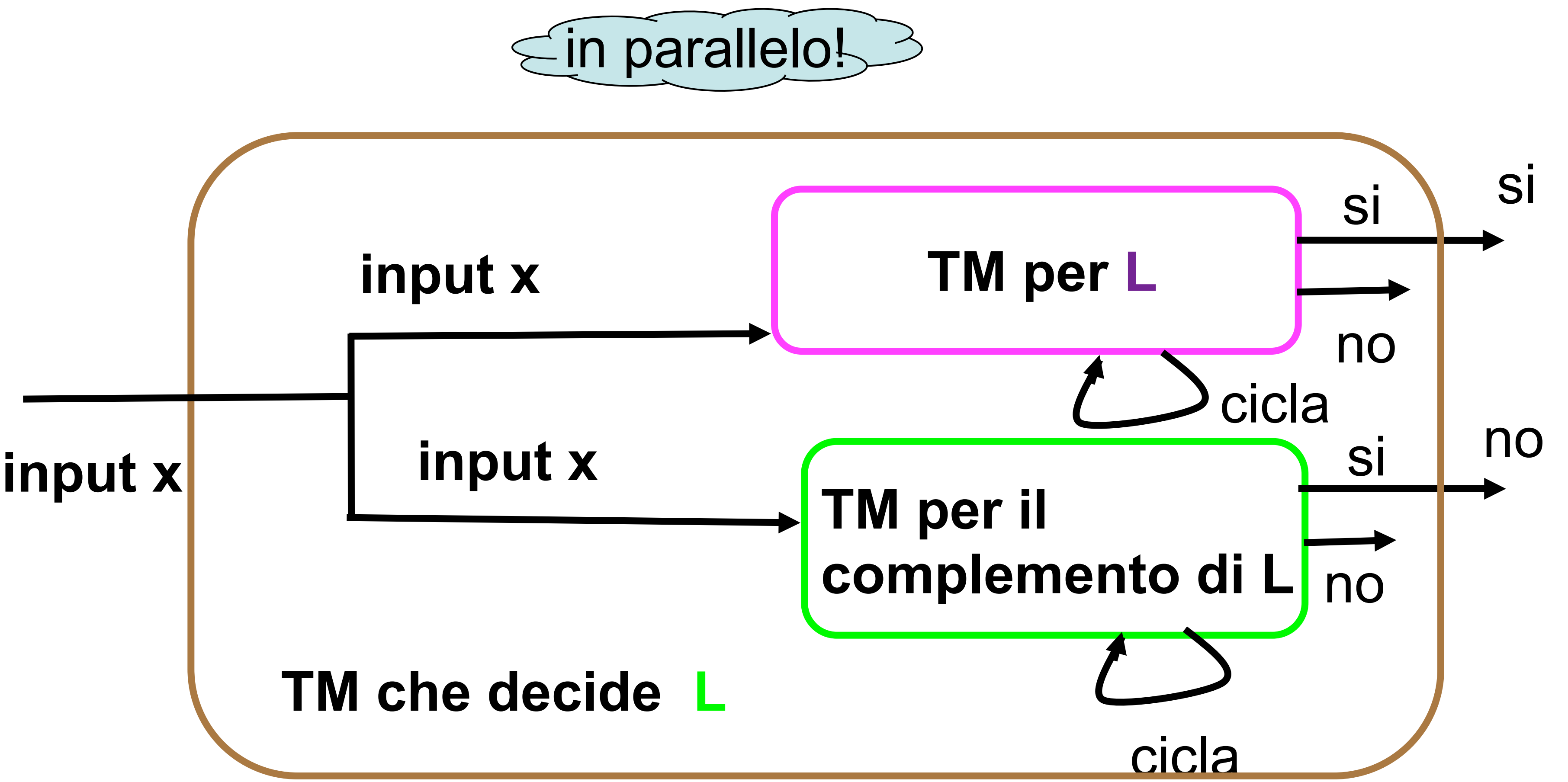
In altre parole abbiamo dimostrato che non esiste un algoritmo che risolve il problema dell'accettazione per TM o equivalentemente che non esiste una TM che decide il linguaggio

$$A_{TM} = \{\langle T, w \rangle \mid T \text{ è una TM e } w \in L(T)\}$$

La UTM U riconosce A_{TM}, ma non lo decide perchè se T non si ferma anche U non si ferma.

Proprietà delle TM

Teorema. **Se L e il suo complemento sono Turing riconoscibili allora L è decidibile.**



Una possibile implementazione

Teorema. **Se L e il suo complemento sono Turing riconoscibili allora L è decidibile.**

Prova:

Sia la TM T1 che riconosce L e T2 quella per il suo complemento.

Costruiamo una TM T che simula T1 e T2 "in parallelo", T ha due nastri:

T sull'input x

1. **copia x sul secondo nastro**
2. **esegui una mossa di T1, sul primo nastro**
   se T1 accetta, accetta
3. **esegui una mossa di T2, sul secondo nastro**
   se T2 accetta, rifiuta
4. **torna al punto 2**

Una parola x è accettata da T1 o da T2, le parole sulle quali T1 non si ferma non sono accettate da T1 quindi sono nel complemento e sono accettate da T2

Problemi non Turing riconoscibili

Teorema. Il complemento del problema dell'accettazione e di quello della fermata non sono Turing riconoscibili.

Prova.

A_{TM} = {< T,w> | T è un TM e w $\in$L(T)} e $Halt_{TM}$ = {< T, w > | T è un TM e si ferma su w} sono Turing riconoscibili.

Infatti per $Halt_{TM}$ basta modificare opportunamente la UTM in modo che accetti quando la TM in input si ferma accettando o rifiutando il suo input.

Ma sono indecidibili, quindi il loro complemento non può essere Turing riconoscibile, per il teorema precedente.

Proprietà di chiusura delle TM

Teorema. $\mathcal{L}$(TM) è chiusa rispetto a unione e intersezione.

Prova. Dati due linguaggi Turing riconoscibili, L1 ed L2, facciamo vedere che anche L1$\cup$ L2 e L1$\cap$ L2 sono Turing riconoscibili.

TM per l' Intersezione:
sull'input x
esegui T1
se T1 accetta esegui T2
se T2 accetta accetta
altrimenti rifiuta

TM per l' Unione:
sull'input x
esegui T1 e T2 in parallelo,
se una delle due accetta,
accetta
se T1 e T2 rifiutano,
rifiuta

Proprietà di chiusura delle TM - 2

E il complemento?

Teorema. La classe dei linguaggi decidibili è chiusa rispetto al complemento.

Prova. Sia T una TM che accetta L. Costruiamo una TM che accetta il complemento.

TM per il complemento:
sull'input x
esegui la TM T
se T accetta rifiuta
altrimenti accetta

T si ferma sempre!