Sommario

- **Macchina di Turing universale**
- **il problema dell'accettazione per TM e sua indecidibilità**
- **il problema della fermata e sua indecidibilità**
- **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 il modello delle TM è equivalente a una moderno calcolatore con un linguaggio di programmazione e una riserva illimitata di memoria.

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.

Costruzione UTM: la codifica

- 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 saranno codificati come numeri binari di lunghezza $\lceil \log(|Q| + |\Gamma|) \rceil$

Costruzione UTM: fine codifica

La codifica della TM T in input per U 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 $x = x_1 \dots x_k$, con la virgola ancora usata come separatore degli interi binari che codificano i singoli simboli.

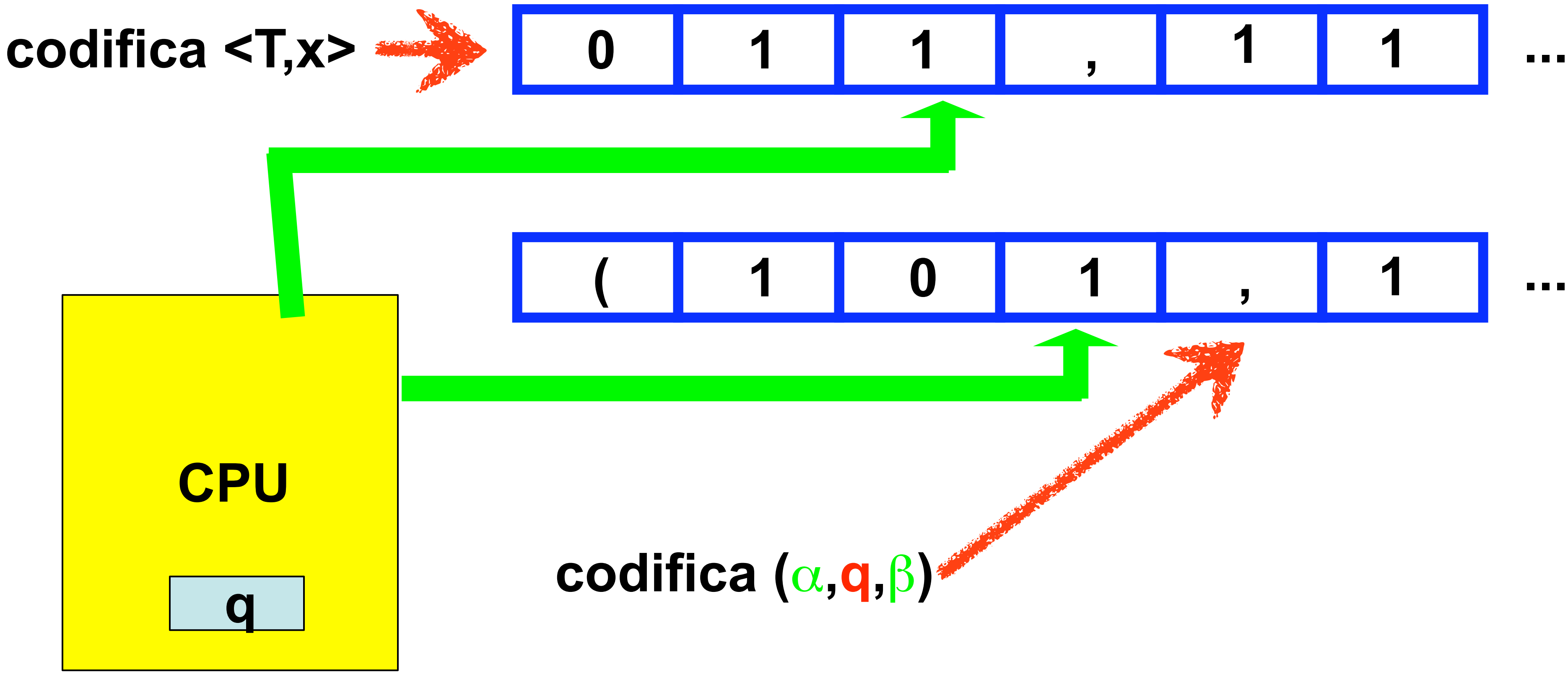
Costruzione UTM - 2

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

- **il primo contiene l'input <$T, x_1 \ldots x_k$>**
- **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, ($q_0, x_1 \ldots x_k$)

Macchina di Turing universale



Costruzione UTM - 4

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 una 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 si ferma o rifiuta.

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?

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 esiste un algoritmo che risolve il problema dell'accettazione per TM o equivalentemente esiste una TM che <u>decide</u> il linguaggio

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

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

Non decidibilità - 1

Teorema. Il problema dell'accettazione per TM non è decidibile.

Prova. La prova utiizza il metodo della diagonalizzazione e l'autoreferenzialità.

Supponiamo per assurdo che esista una TM M che decide A_{TM} = {< T,w> | T è un TM e w $\in$L(T)}.

$$M(\langle T,w\rangle)\begin{cases}\text{accetta se } w \in L(T)\\ \text{altrimenti rifiuta}\end{cases}$$

Complementiamo M e otteniamo la TM M'

Non decidibilità - 2

$M'(<T,w>)$ $\begin{cases} \text{rifiuta se } w \in L(T) \\ \text{accetta altrimenti} \end{cases}$

Costruiamo una tabella con l'esito del calcolo di M' su <T,<T'>>, la codifica di una TM T che prende in input la codifica di una TM T'

M'	$<T_1>$	$<T_2>$	$<T_3>$	$<T_4>$	$<T_5>$	$<T_6>$	...
T_1	a	r	r	a	r	a	...
T_2	r	a	a	a	r	a	...
T_3	a	a	r	a	a	r	...
T_4	r	r	r	r	r	a	...
T_5	a	a	a	r	r	r	...
T_6	r	a	r	r	r	a	...
...							

Costruiamo una TM D(< T >)= M' (< T ,< T >>)

D si comporta come M' su <T,<T>>, una TM T che prende in input la codifica di "se stessa"

Non decidibilità - 3

$$D(\langle T\rangle) = \begin{cases} \text{rifiuta se } \langle T\rangle \in L(T) \\ \text{accetta altrimenti} \end{cases}$$

Come si comporta D su < D >?

se $\langle D \rangle \in L(D)$ allora $D(\langle D \rangle)$ rifiuta, cioè $\langle D \rangle \notin L(D)$,

se $\langle D \rangle \notin L(D)$ allora $D(\langle D \rangle)$ accetta, cioè $\langle D \rangle \in L(D)$,

La contraddizione deriva dall'ipotesi di esistenza della TM M che decide $A_{TM} = \{\langle T,w\rangle \mid T$ è un TM e $w \in L(T)\}$.

Il problema è dunque indecidibile.

Il problema della fermata

Problema della fermata (Halting problem): data una TM T e una parola input w per T, T si ferma su w?

Esiste un algoritmo che risolve il problema della fermata?
In altre parole il linguaggio

$$Halt_{TM} = \{< T, w > \mid T \text{ è un TM e si ferma su } w\}$$

è decidibile?

Teorema. Il problema della fermata è indecidibile.
Prova. Supponiamo per assurdo che M sia una TM che decide $Halt_{TM}$:

$$M(<T,w>) = \begin{cases} \text{accetta se } T \text{ si ferma su } w \\ \text{altrimenti rifiuta} \end{cases}$$

Il problema della fermata: prova

$$M(\langle T,w\rangle)=\begin{cases}\text{accetta se } T \text{ si ferma su } w\\ \text{altrimenti rifiuta}\end{cases}$$

Possiamo allora usare M per costruire una TM M' che prende in input una TM T e non si ferma se M(<T,<T>>) si ferma e si ferma altrimenti:

TM M' :
Sull'input (<T>)
esegui M su (<T,<T>>);
se M accetta, cicla senza fine
altrimenti accetta

$$M'(\langle T\rangle)=\begin{cases}\text{accetta se } T \text{ non si ferma su } \langle T\rangle\\ \text{altrimenti cicla}\end{cases}$$

Ci domandiamo ora se M' si ferma su <M'>:
M' <u>non</u> si ferma su <M'> se M accetta <M',<M'>> e cioè se M' si ferma su <M'> ma M' si ferma altrimenti, cioè se M rifiuta <M',<M'>> e cioè se M' <u>non</u> si ferma su <M'>!

Il problema della fermata - 2° prova

$$M(\langle T,w \rangle)= \begin{cases} \text{accetta se } T \text{ si ferma su } w \\ \text{altrimenti rifiuta} \end{cases}$$

Possiamo allora usare M per costruire una TM M' che decide A_{TM} cosa che sappiamo impossibile.

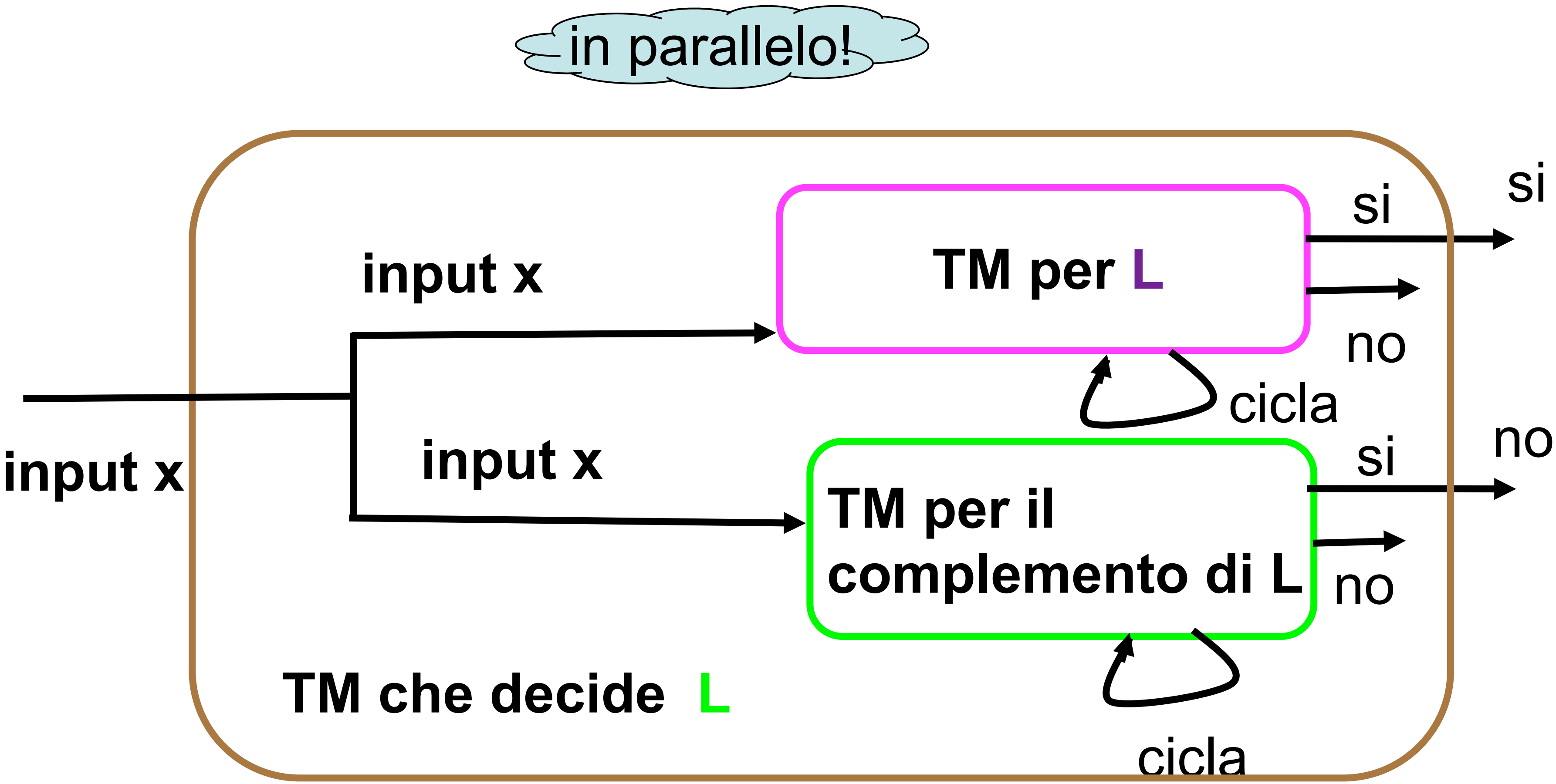
Poichè M accetta, T si ferma su w

TM M' :
Sull'input (<T,w>)
esegui M su (<T,w>);
se M accetta esegui T su w
se T accetta, accetta
e se T rifiuta, rifiuta
se M rifiuta, rifiuta

$$M'(T,w\rangle)= \begin{cases} \text{accetta} \\ \text{se } T \text{ accetta } w \\ \text{altrimenti rifiuta} \end{cases}$$

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:

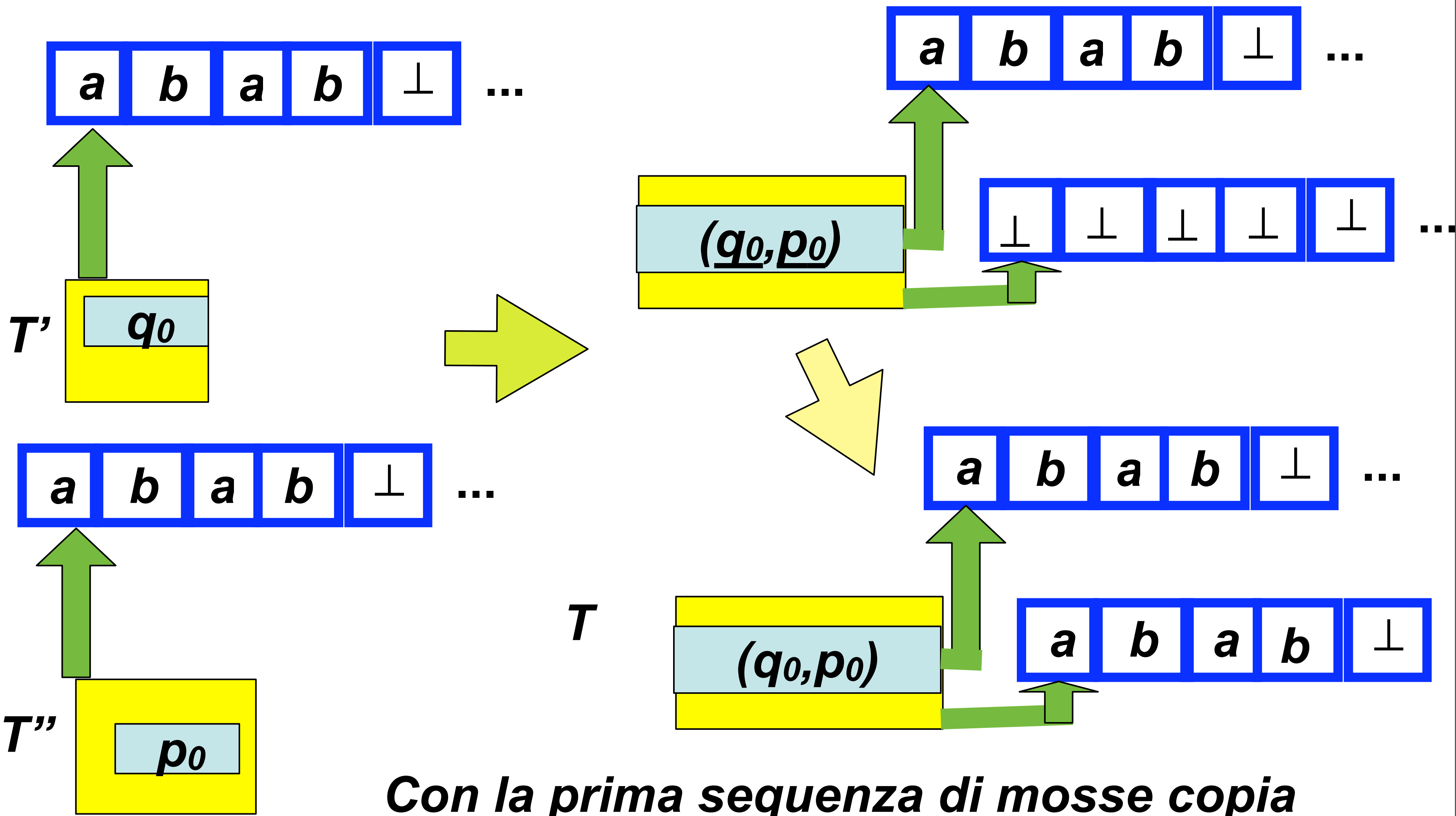
Sia la TM T1 che riconosce L e T2 quella per il suo complemento. Costruiamo una TM T che simula T1 e T2 "in parallelo". T ha due nastri.

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**

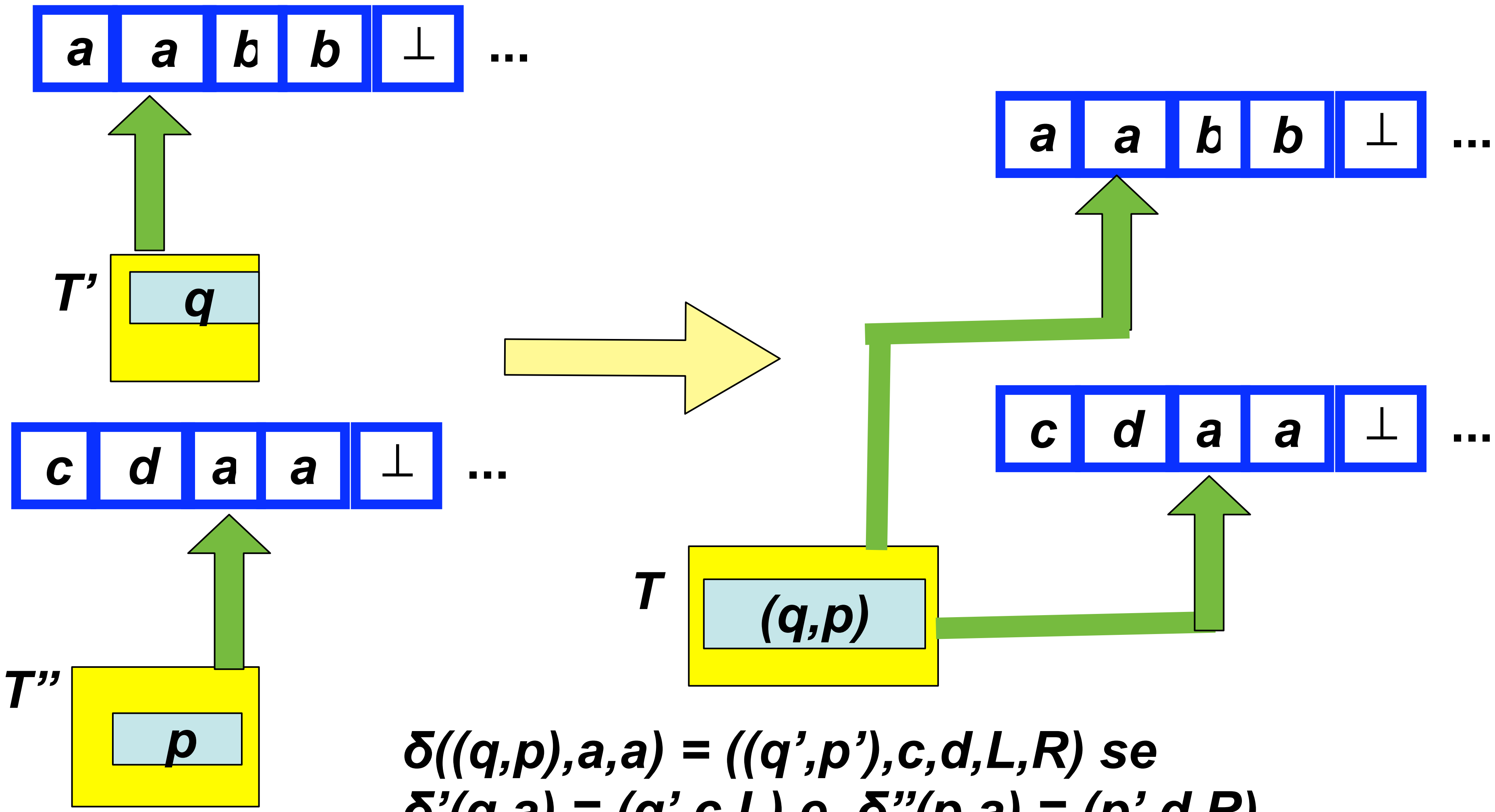
Una parola x è accettata da T1 o da T2, le parole sulle quali T1 non si ferma non sono accettate da T1 quindi sono nel complemento e sono accettate da T2

Una implementazione alternativa



Con la prima sequenza di mosse copia l'input sul secondo nastro ed entra nello stato (q_0,p_0)

Composizione in parallelo



δ((q,p),a,a) = ((q',p'),c,d,L,R) se
δ'(q,a) = (q',c,L) e δ''(p,a) = (p',d,R)
Appena una delle due TM si ferma e accetta allora la TM composta si ferma e accetta

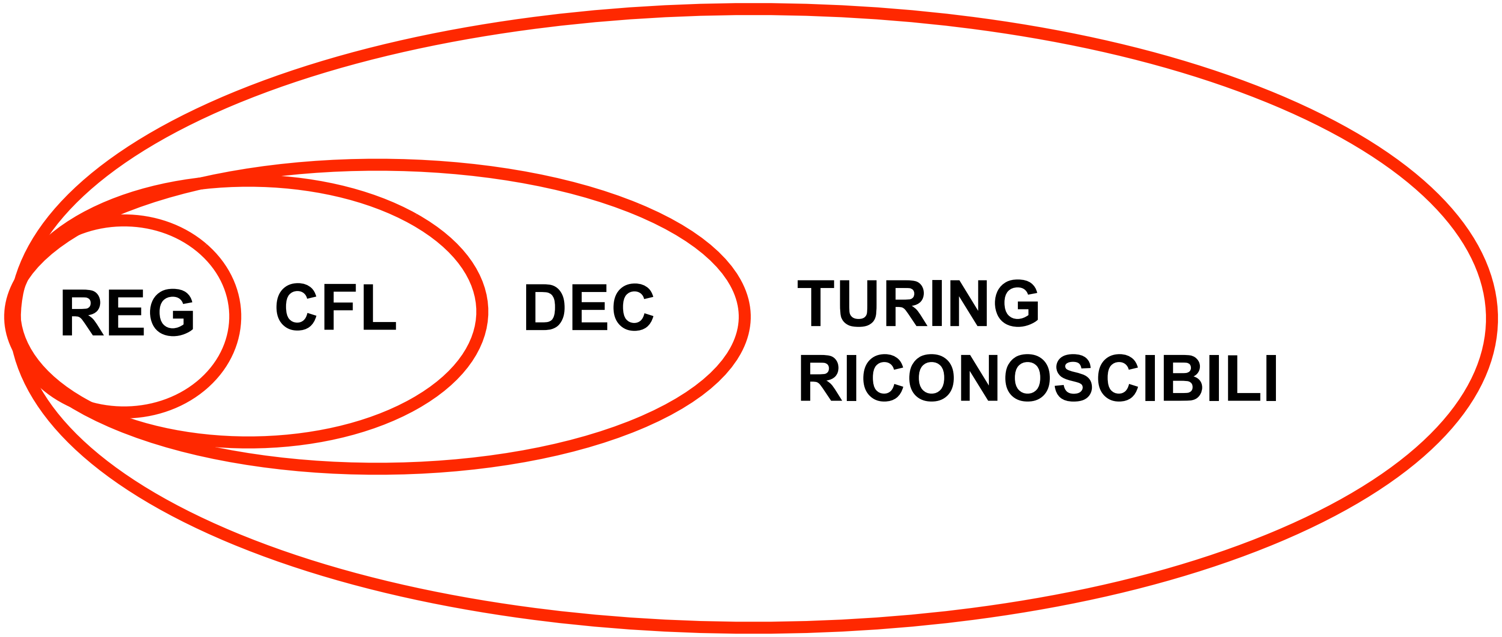
Problemi non Turing riconoscibili

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.

La gerarchia dei linguaggi:



Enumerabilità e Turing calcolablità

Ogni linguaggio enumerabile è Turing riconoscibile? In entrambi i casi si dimostri quanto affermato.

Risposta. NO. Le TM sono codificabili in binario, cosi' some le parole sull'alfabeto di input.
Poichè {0,1}* è enumerabile, il complemento di A_{TM} = {<T,w> | T è una TM e w è in L(T)} è un sottoinsieme infinito di un insieme enumerabile e in quanto tale enumerabile.
Ma si è dimostrato che il complemento di A_{TM} non è Turing riconoscibile.