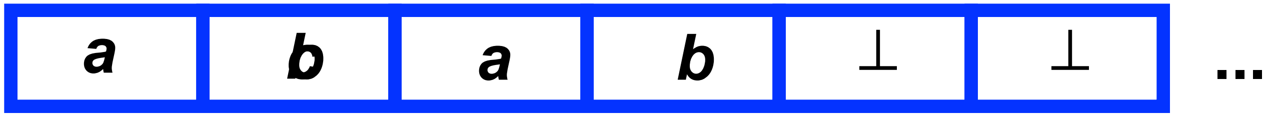
Sommario

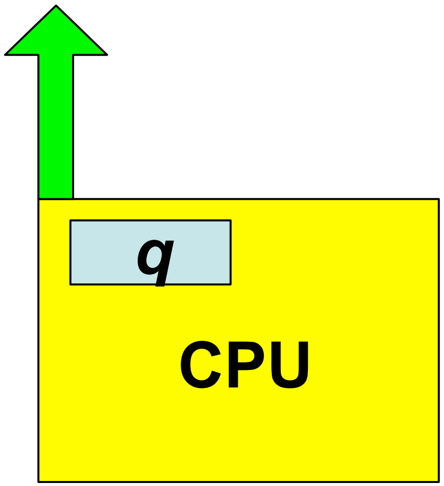
- **Complessità di tempo e spazio di una TM che si ferma sempre.**

- **Relazioni tra le due misure**

- **Analisi complessità delle TM costruite per dimostrare che più nastri o il non determinismo non aumentano il potere computazionale del modello**

Complessità di tempo di una TM M





Da ora in poi consideriamo **solo** TM che si **fermano sempre**

Contiamo i passi!

Definiamo la funzione $\mathrm{Passi}_M : \{a,b\}^* \rightarrow N$ tale che $\mathrm{Passi}_M(x)$ =n° passi su input x

Definiamo la funzione $t_M : N \rightarrow N$ tale che $t_M(n) = \max\{\mathrm{Passi}_M(x)$ per x di lunghezza $n\}$

Il caso peggiore!

Esempio: la TM M che riconosce $\{0^n1^n \mid n\geq 0\}$

su input del tipo 10^{n-1} T termina in un passo rifiutando
Quindi $Passi_M(10^{n-1}) = 1$

su input del tipo 0^m1^m tale che $2m = n$
T compie $m+1$ passi per la prima coppia 01 per ottenere sul nastro $X0^{m-1}Y1^{m-1}$
poi la TM torna indietro di $m+1$ posizioni e compie altri $m+1$ passi per la seconda coppia...

Ogni volta la distanza tra lo 0 e l'1 corrispondente è m quindi complessivamente si compiono $O(m^2)$ passi: $Passi_M(0^m1^m) = O(m^2)$. Nel caso peggiore quindi

$$t_M(n) = O(n^2)$$

Dimensione input

La complessità di tempo o di spazio è sempre misurata in funzione della dimensione dell'input, quindi la rappresentazione dei dati può fare differenza.

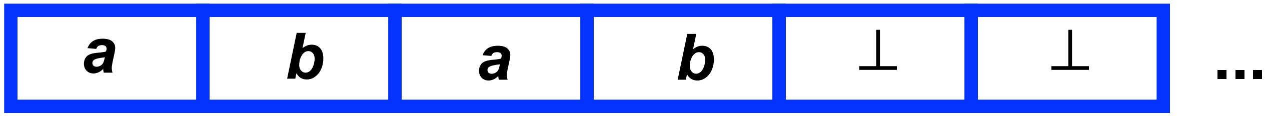
In questo contesto ogni rappresentazione ragionevole va bene.

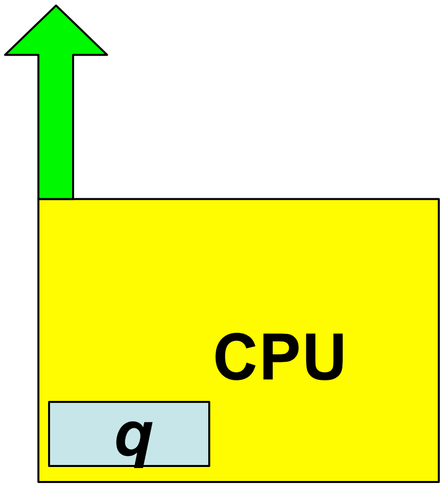
Per un numero
per es 17

vanno bene	non va bene
17	$1^{17} = 11111111111111111$
10001	

N.B. Per ogni T in TM, assumiamo che un input di lunghezza n dovrà essere letto interamente almeno in un caso, e quindi che vale sempre $t_T(n) \geq n$.

Complessità di spazio di una TM M





contiamo il numero di celle utilizzate!

Definiamo la funzione $T_M : \{a,b\}^* \rightarrow N$ tale che

$Celle_M(x)$ = n° celle utilizzate su input x

Definiamo la funzione

$s_M(n) = \max\{Celle_M(x)$ per x di lunghezza $n\}$

Il caso peggiore!

Quindi, nella nostra ipotesi sull'input, $s_M(n) \geq n$

Relazione tra spazio e tempo

Conoscendo la complessità di tempo possiamo limitare superiormente quella di spazio?
Poichè ogni cella visitata comporta un passo di calcolo è evidente che per ogni TM T:

$$s_T(n) \leq t_T(n)$$

E se si conosce la complessità di spazio $s_T(n)$ possiamo limitare superiormente quella di tempo?

Relazione tra spazio e tempo - 2

Se la complessità di spazio è $s_T(n)$ allora il contenuto del nastro in ogni passo di calcolo è una parola di lunghezza al più $s_T(n)$.

Con ogni mossa la TM cambia la configurazione in cui si trova. Il numero di tutte le possibili configurazioni, con contenuto del nastro di lunghezza al più $s_T(n)$, fornisce quindi un limite superiore alla complessità di tempo di una TM.

Relazione tra spazio e tempo - 2

Se d è il numero di simboli dell'alfabeto di nastro, il numero delle parole di lunghezza $s_T(n)$ è $d^{s_T(n)}$

Tenendo conto che in ogni stato si può avere un qualsiasi contenuto di nastro, detto q il numero degli stati, abbiamo

$$q * d^{s_T(n)}$$

e infine tenendo conto che la testina di lettura può trovarsi su una cella qualunque, il numero delle configurazioni è

$$q * s_T(n) * d^{s_T(n)}$$

In conclusione $t_T(n) \leq 2^{O(s_T(n))}$

Spazio, tempo e numero di nastri

La complessità di spazio di una TM T a k nastri, $s_T(n)$, è

$s_T(n) = \max$\{la somma dei n° delle celle utilizzate su input x di lunghezza n, su ogni nastro\}

Teorema. **Se L è riconosciuto da una TM con k nastri con complessità di spazio $s_T(n)$, allora L è riconosciuto da una TM a un solo nastro con complessità di spazio $O(s_T(n))$.**

E per il tempo?

Tempo e numero di nastri

Teorema. Se L è riconosciuto da una TM con k nastri con complessità di tempo $t_T(n)$, allora L è riconosciuto da una TM a un solo nastro con complessità di tempo $O((t_T(n))^2)$.

Prova. Ricordiamo che la TM a un solo nastro inizializza il proprio nastro per predisporre la simulazione della macchina a k nastri.

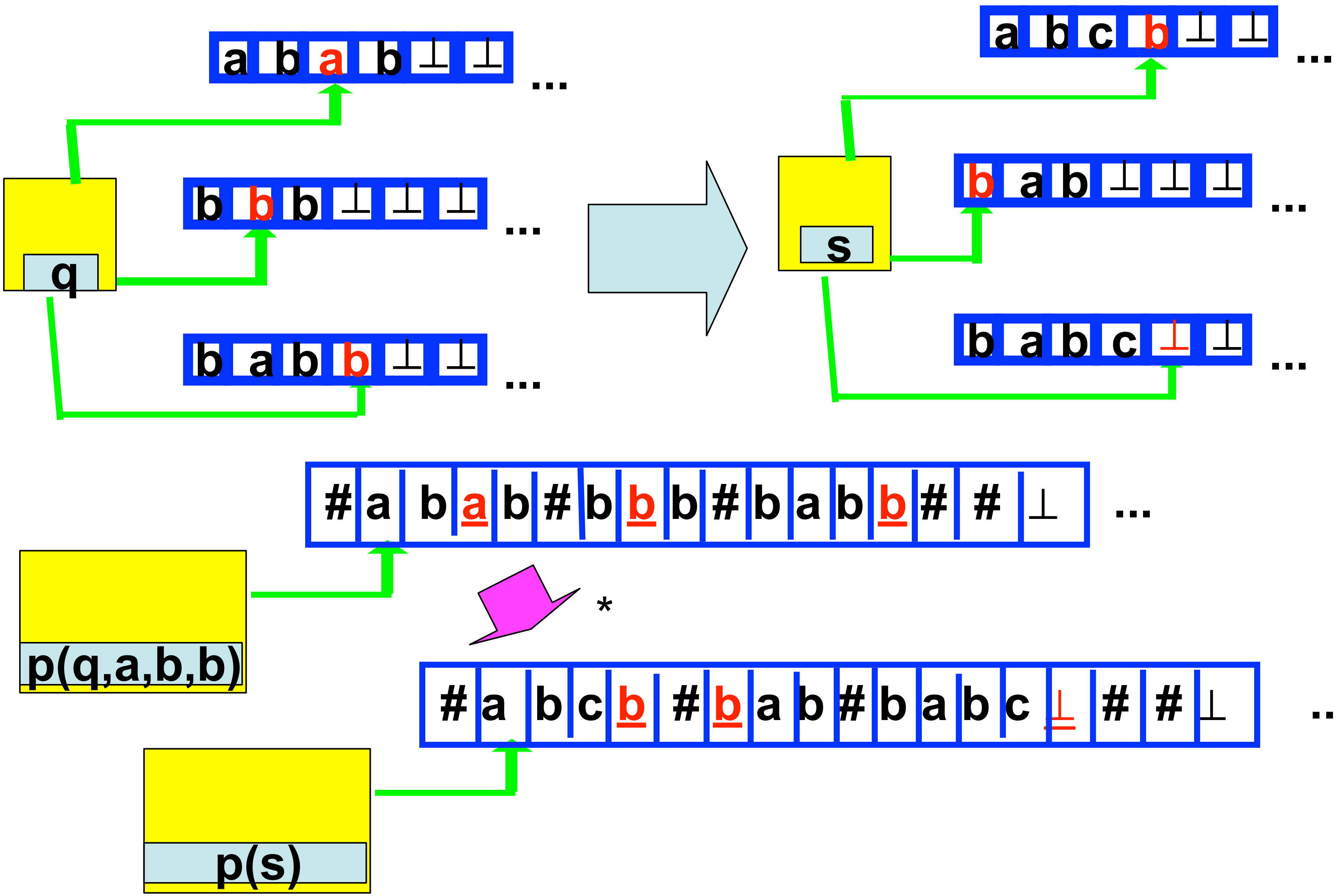
Alla fine di questa fase avremo questo contenuto sul nastro:

$$\$p(q_0)\underline{a_1}\ldots a_n \underbrace{\#\underline{\perp}\# \ldots \underline{\perp}\#\underline{\perp}}_{k\text{-1 volte}}\#\#$$

dove $a_1 \ldots a_n$ è la stringa input. Questo costa O(n).

Simulazione di una mossa

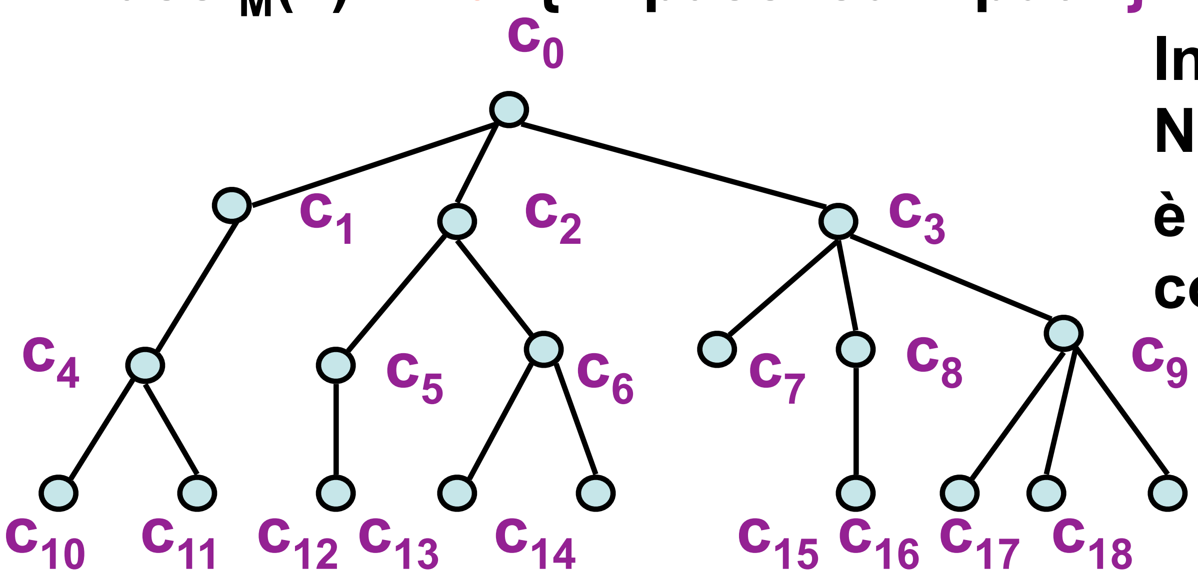
k = 3



Complessità di tempo di una NTM M

Definiamo la funzione $NPassi_M : \{a,b\}^* \rightarrow N$ tale che

$NPassi_M(x) = max\{n°$ passi su input $x\}$



In altre parole $NPassi_M(x) : \{a,b\}^* \rightarrow N$ è l'altezza dell'albero di computazione di x.

Anche qui $t_T(n) \geq n$.

Definiamo la funzione $t_M : N \rightarrow N$ tale che

$t_M(n) = max\{NPassi_M(x)$ per x di lunghezza $n\}$

Il caso peggiore!

In altre parole $t_M(n) : N \rightarrow N$ è l'altezza dell'albero di computazione più alto tra tutti quelli determinati da input di lunghezza n.

Complessità di spazio di una NTM M

Definiamo la funzione $S_M : \{a,b\}^* \rightarrow N$ tale che
$NCelle_M(x)$ = max{n° celle utilizzate in ogni computazione su input x}

Definiamo la funzione $s_M : N \rightarrow N$ tale che:
$s_M(n)$ = max{$NCelle_M(x)$ su input x di lunghezza n}

Anche qui $s_M(n) \geq n$.

Spazio, tempo e nontederminismo

Sappiamo costruire una TM equivalente a una NTM data.

Possiamo mettere in relazione le complessità di tempo e spazio delle macchine?

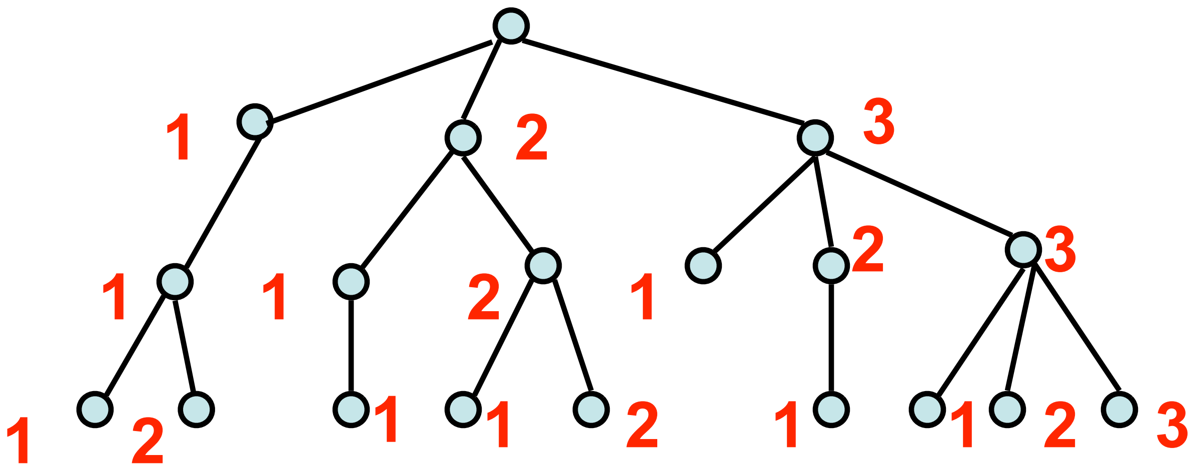
Dobbiamo prima modificare la costruzione della TM in modo che si fermi quando la NTM data si ferma!

Una TM equivalente a una NTM che non si ferma sempre

- Sia **$M = (Q,\Sigma, \Gamma,\delta,q_0, q_a , q_r)$ la NTM da simulare.**
- **Sia d = max {|δ(q,a)| q in Q e a in Γ }**
- **Sia D = {1,2...,d} l'alfabeto delle sequenze guida**
- **Sia M_0 la TM che genera e scrive sul nastro la sequenza successiva a quella data in input.**
- **La TM M' equivalente a M è una 3-TM che esegue tutte le possibili esecuzioni della NTM nell'ordine dettato dalle sequenze guida sul terzo nastro. Queste ultime vengono generate e aggiornate dalla copia della TM M_0 incorporata in M'. Come visto, la TM M' usa il secondo nastro per ogni esecuzione, e conserva l'input sul primo per poterlo ricopiare e avviare la successiva esecuzione. M' si ferma e accetta quando M si ferma e accetta.**
- **In tutti gli altri casi non si ferma, perchè genera la successiva sequenza sul terzo nastro e ricomincia la simulazione.**

Una TM equivalente a una NTM che si ferma sempre: il caso del rifiuto

Una parola è rifiutata se ogni cammino di computazione su di essa è di rifiuto :

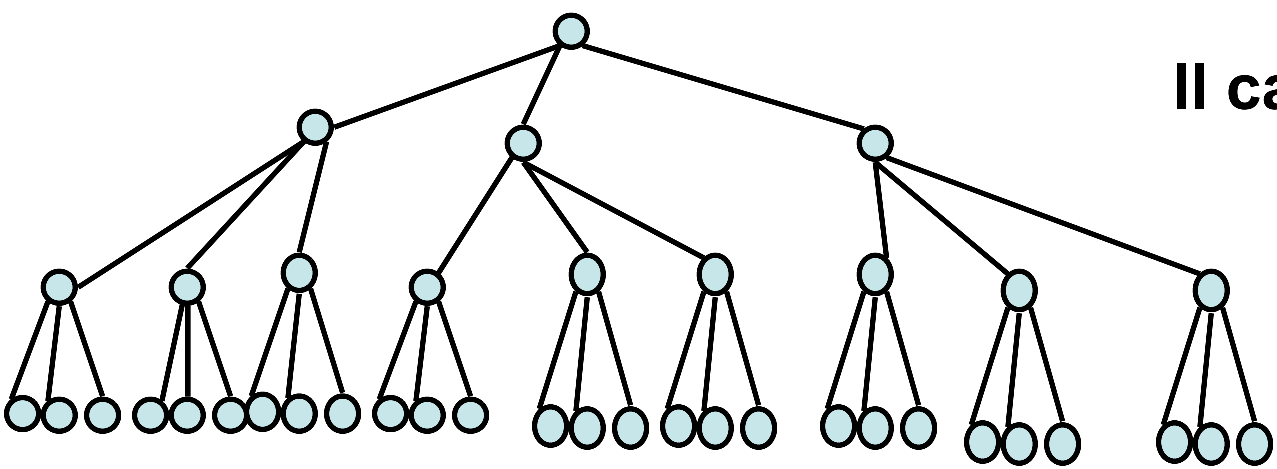


Aggiungiamo un contatore, il IV nastro, che sarà inizializzato a 1 ogni volta che si genera sul III nastro una sequenza del tipo 1^n, cioè quando si simula il cammino più a sinistra.

Il contatore verrà incrementato di 1 ogni volta che la computazione guidata dalla sequenza sul terzo nastro produce uno stato di rifiuto o fallisce. Il contatore viene confrontato con d^n, quando si genera sul IV nastro la sequenza d^n (cioè si simula l'ultimo cammino di computazione di lunghezza n) e se i due valori coincidono si ferma e rifiuta.

Spazio, tempo e nontederminismo -2

Otteniamo un limite superiore alla complessità di tempo della TM deterministica equivalente alla NTM data, considerando il più grande albero di computazione possibile.
Si tratta dell'albero *d*-ario completo di altezza la complessità di tempo $t_T(n)$, dove d è il massimo grado di non determinismo.



Il caso di *d*=3

Spazio, tempo e nontederminismo - 3

Il numero dei nodi di un albero *d*-ario completo di altezza $t_T(n)$ è

$$(d^{t_T(n)+1} - 1)/(d - 1) = O(d^{t_T(n)})$$

Quindi nel passaggio da una NTM alla TM equivalente la complessità di tempo diventa $O(t_T(n)d^{t_T(n)}) = 2^{O(t_T(n))}$.

Teorema. **Se L è riconosciuto da una NTM con complessità di tempo $t_T(n)$, allora L è riconosciuto da una TM con complessità di tempo**

$$2^{O(t_T(n))}$$