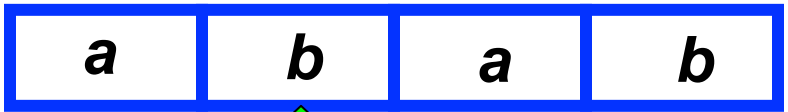
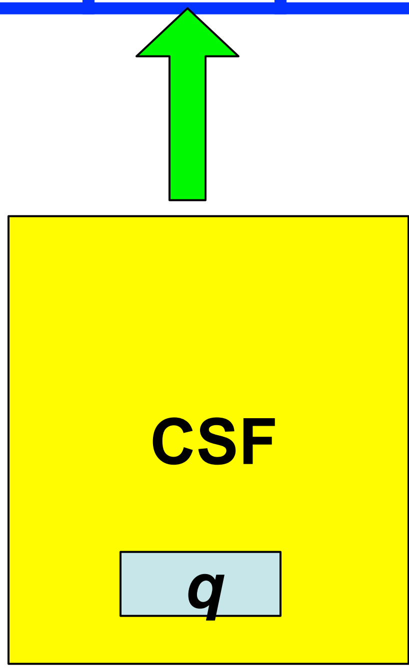
Sommario

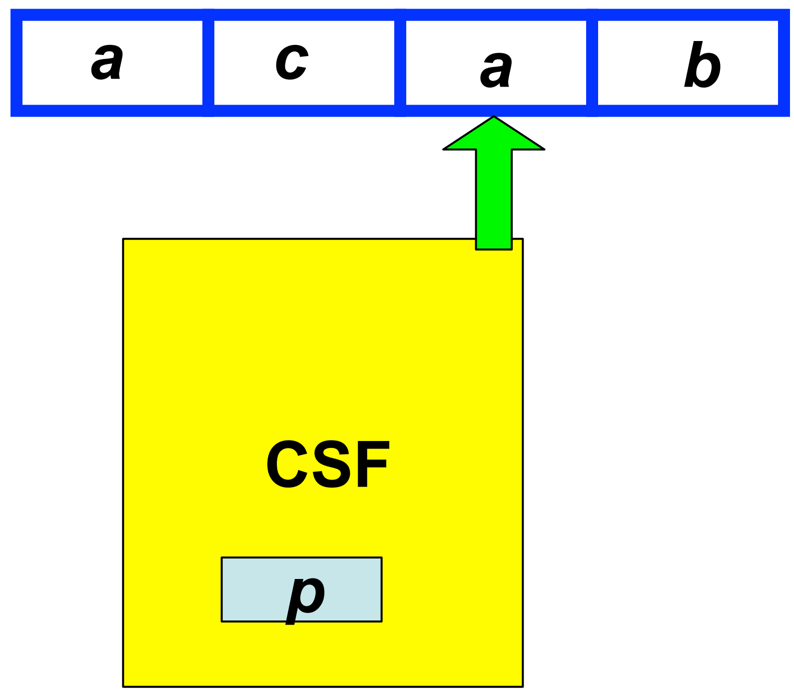
- Versione non deterministica della TM
- equivalenza con la deterministica

Macchine di Turing non deterministica

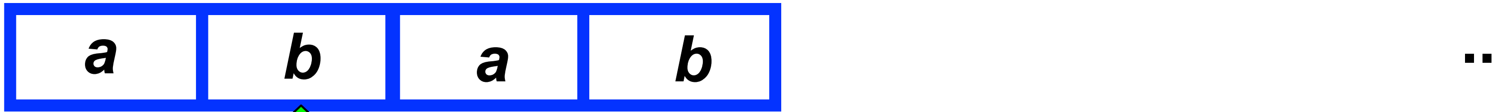


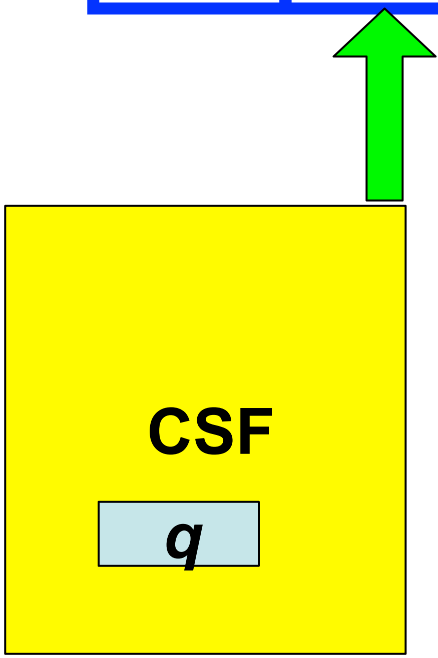


leggendo *b* nello stato *q* la TM può passare nello stato *p* scrivere *c* nella casella e spostare la testina di una posizione a destra, oppure

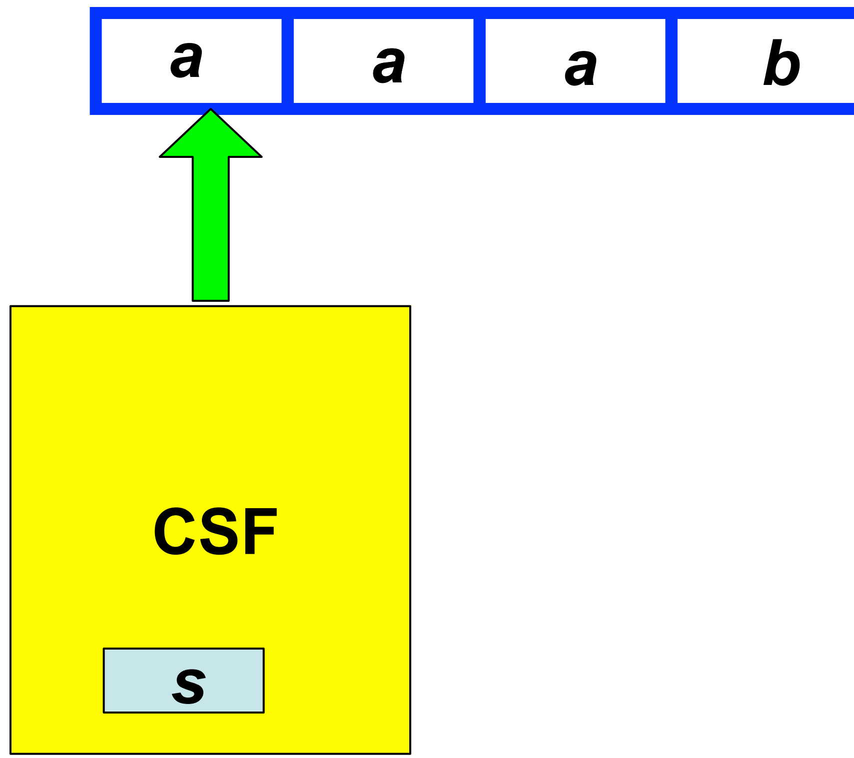


Macchine di Turing non deterministica





oppure passare nello stato s, scrivere a nella casella e spostare la testina di una posizione a sinistra, oppure...



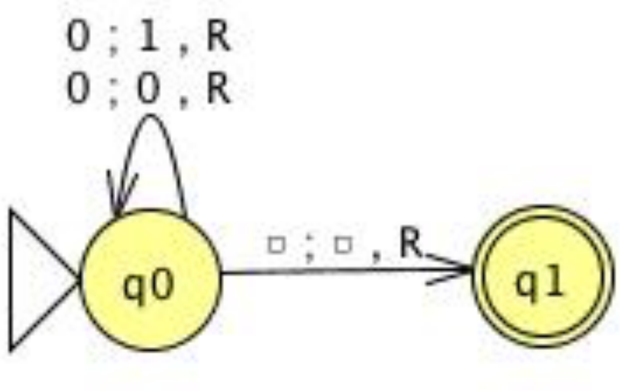
Esempio di una TM Nondeterministica

La NTM M verifica se un numero è composto.

M:
l'input è un numero binario n

1. scrivi sul secondo nastro, non deterministicamente un numero binario, m, di lunghezza pari a quella di n,
2. verifica se m=0, se m=1 e se è maggiore di $n^{1/2}$
e se uno dei test ha successo fermata e fallimento
3. (altrimenti) dividi n per m e se il resto è maggiore di zero fermata e fallimento altrimenti fermata e successo

Esempio di una TM Nondeterministica.



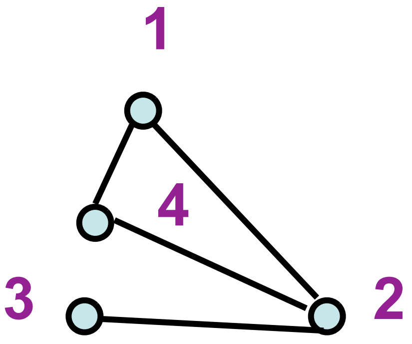
Genera una stringa binaria qualunque della lunghezza della sequenza di 0 data in input.

Esempio 2 di una TM Nondeterministica

La TM M a due nastri verifica se un grafo non diretto ha un ciclo hamiltoniano.

Per l'input prevediamo una codifica di questo tipo:
se il grafo ha n vertici possiamo supporre che sia V= {1,2,...,n} e descrivere gli archi come coppie di vertici, per esempio:

<G> = (1,2,3,4),({1,2},{1,4},{2,3},{2,4}) se G è il grafo del disegno



M:
sull'input <G>
stadio 1. scrivi sul secondo nastro non deterministicamente una permutazione dell'insieme dei vertici
stadio 2. verifica, sul secondo nastro, se tra ogni coppia di vertici c'è un arco e se c'è tra il primo e l'ultimo
se le verifiche dello stadio 2 sono andate a buon fine fermata e successo
altrimenti fermata e fallimento

Modello formale

Una Machina di Turing non deterministica, in breve NTM (Nondeterministic Turing Machine), è una settupla $M = (Q,\Sigma,\Gamma,\delta,q_0,q_a,q_r)$ le cui componenti sono come nel caso della TM ma la cui funzione di transizione è così definita

$$\delta : Q \times \Gamma \rightarrow \mathcal{P}(Q \times \Gamma \times \{L, R\})$$

dove $\mathcal{P}(Q \times \Gamma \times \{L, R\})$ è l'insieme dei sottoinsiemi di $Q \times \Gamma \times \{L, R\}$.

Con il vincolo $\delta(q_a,a) = \delta(q_r,a) = \varnothing$

Linguaggio accettato

Sia $M = (Q,\Sigma, \Gamma,\delta,q_0, q_a , q_r)$ una NTM, x un input per M e sia C(x) l'insieme delle configurazioni raggiungibili da quella iniziale $c_0 = q_0x$.

Il linguaggio accettato è

$L(M) = \{x \mid x \in \Sigma^*$ e $\exists\ c \in C(x)$ e c è di accettazione$\}$

$\alpha q_a \beta$

Il linguaggio rifiutato è

$R(M) = \{x \mid x \in \Sigma^*$ e $\forall c \in C(x)$ e terminale, c è senza sbocco o di rifiuto$\}$

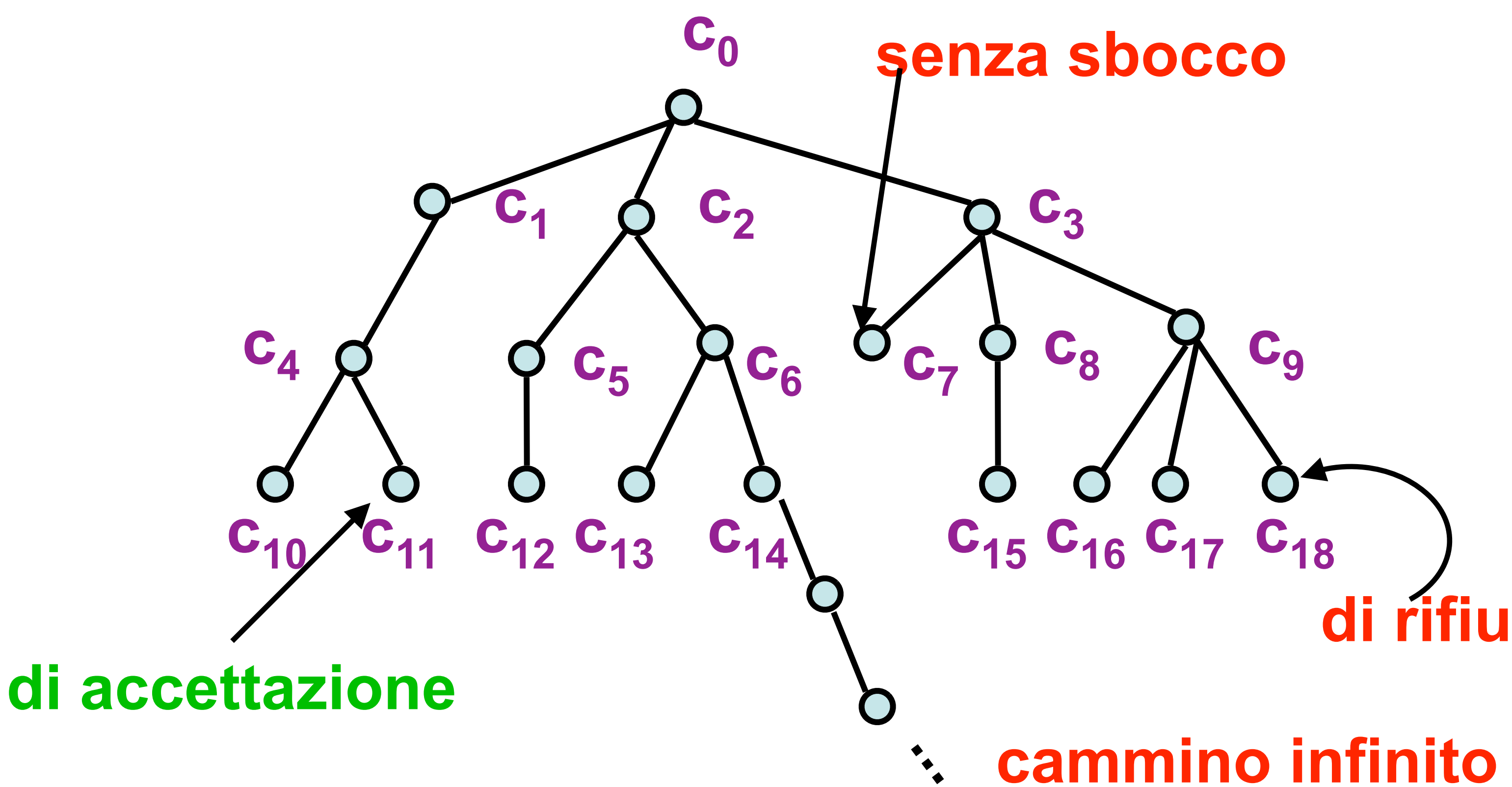
$\alpha q_r \beta$

Non sempre $L(M) \cup R(M) = \Sigma^*$!

NTM: passi di calcolo

Un calcolo di una NTM T è ben descritto da un albero, in cui le etichette dei nodi sono le configurazioni raggiungibili da quella iniziale, $c_0=q_0$x, dove x è l'input.

Il massimo numero di figli di un nodo è il massimo grado di nondeterminismo della TM



$c_0 \Rightarrow_T c_1$

$c_0 \Rightarrow_T c_2$

$c_0 \Rightarrow_T c_3$

$c_1 \Rightarrow_T c_4$

$c_2 \Rightarrow_T c_5$

$c_2 \Rightarrow_T c_6$

...

Classe dei linguaggi accettati

L'insieme di tutti i linguaggi che sono accettati da una NTM è così definito:

$$\mathcal{L}(\mathbf{NTM}) = \{\mathbf{L} \mid \exists\, \mathbf{M} \in \mathbf{NTM} \text{ e } \mathbf{L}(\mathbf{M}) = \mathbf{L}\}$$

Equivalenza tra NTM e TM

DOMANDA:

La versione nondeterministica ha un potere computazionale maggiore??

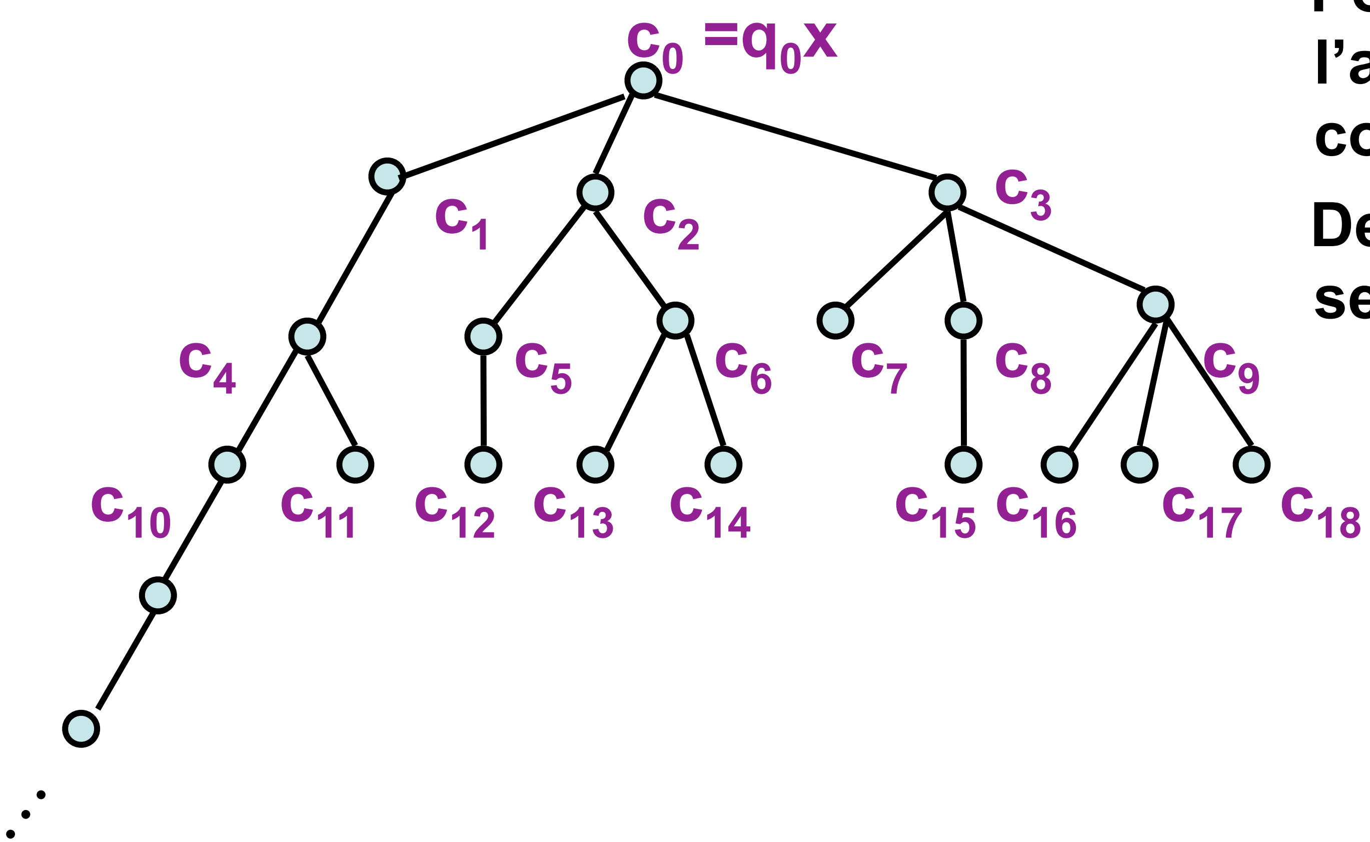
RISPOSTA: NO.

Possiamo dimostrare l'esistenza di una TM equivalente a una NTM data e cioè che $\mathcal{L}(\text{NTM}) \subseteq \mathcal{L}(\text{TM})$,

Quindi $\mathcal{L}(\text{NTM}) = \mathcal{L}(\text{TM})$, visto che la versione deterministica è un caso particolare di quella nondeterministica e cioè che banalmente vale $\mathcal{L}(\text{TM}) \subseteq \mathcal{L}(\text{NTM})$.

UnaTM equivalente a una NTM: l'idea

Una TM M' equivalente alla NTM M deve eseguire in sequenza e in un qualche ordine tutte le possibili esecuzioni di M su un input x e accettare quando M accetta.



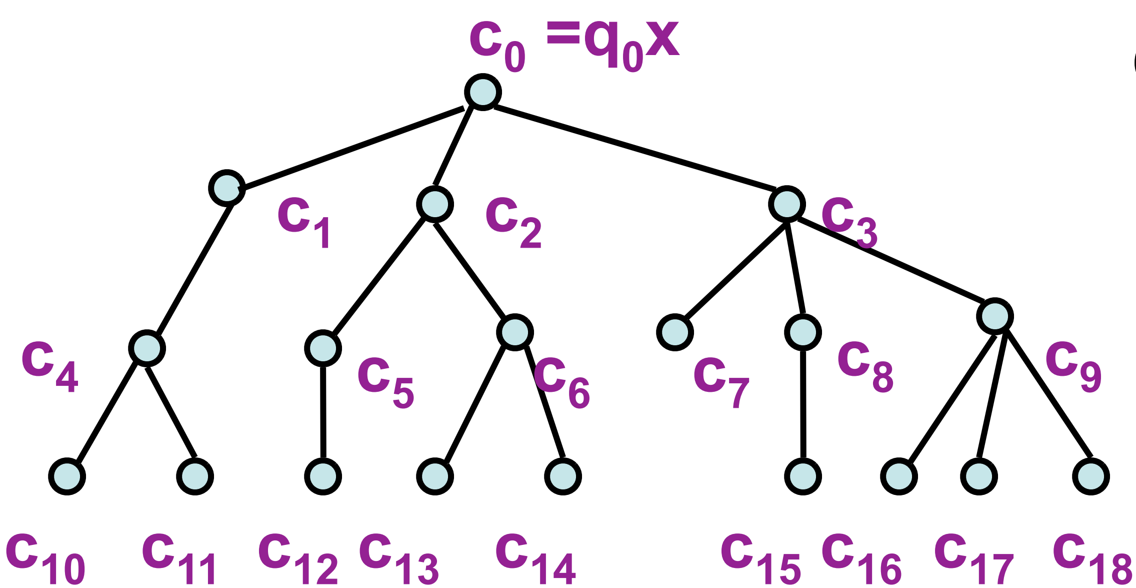
Potremmo visitare l'albero delle configurazioni.

Depth-first-search?

No

UnaTM equivalente a una NTM: l’idea

Una TM M' equivalente alla NTM M deve eseguire in sequenza e in un qualche ordine tutte le possibili esecuzioni di M su un input x e accettare quando M accetta.

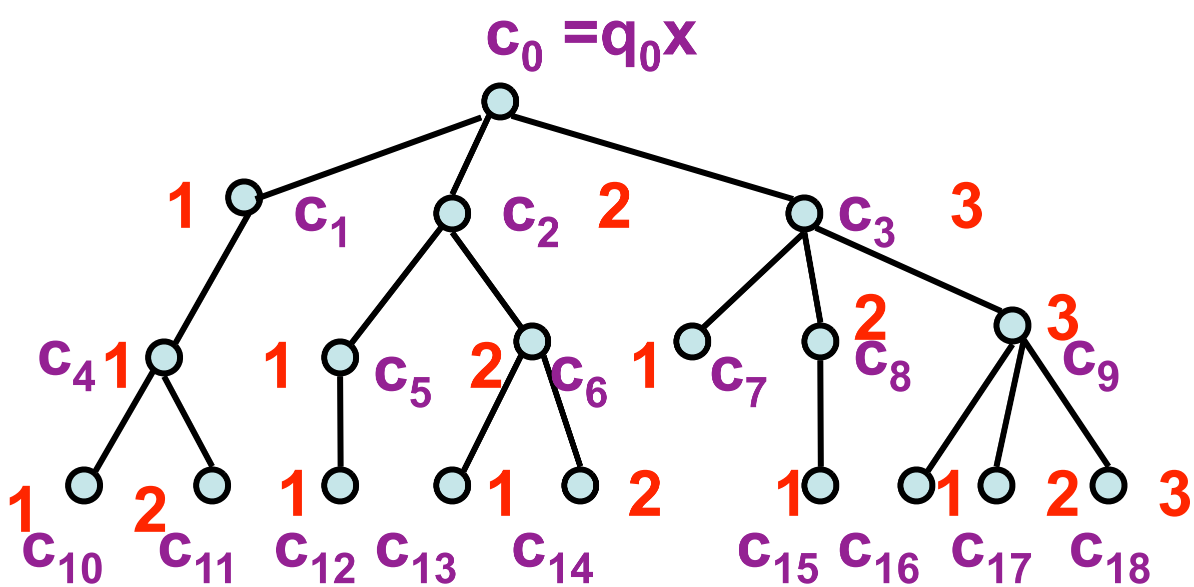


Come visitiamo l’albero delle configurazioni?

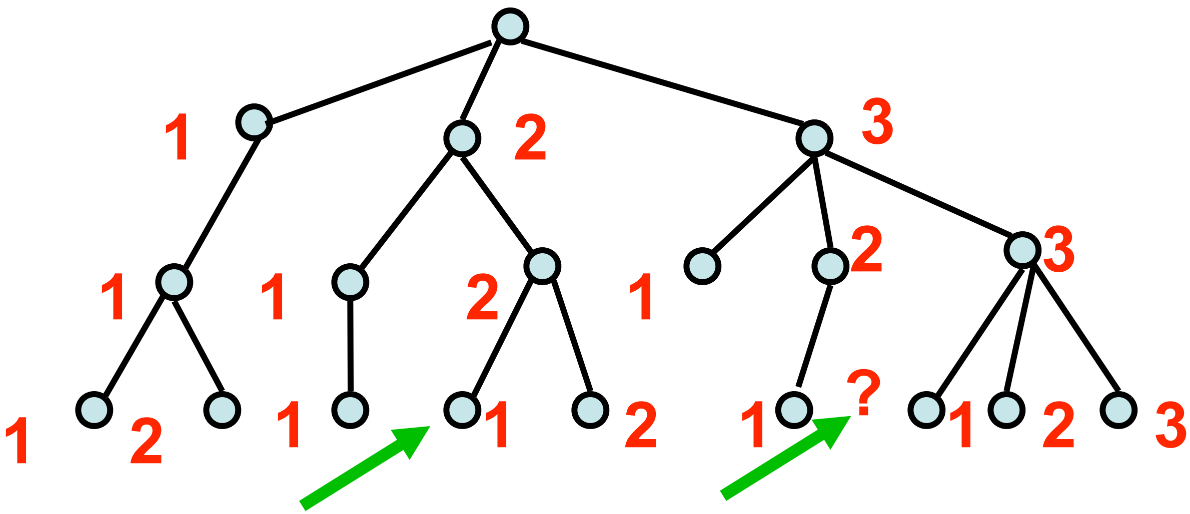
L’albero delle configurazioni raggiungibili da quella iniziale per un input x, q_0x, può essere “visitato” per livelli, per eseguire in sequenza tutte le mosse nondeterministicamente eseguite dalla NTM data.

UnaTM equivalente a una NTM: l'idea

Per essere in grado di eseguire una sequenza di mosse bisogna saper risalire a un cammino radice-nodo di un certo livello, per questo numeriamo in sequenza i figli di ogni nodo da sinistra a destra.



Una TM equivalente a una NTM



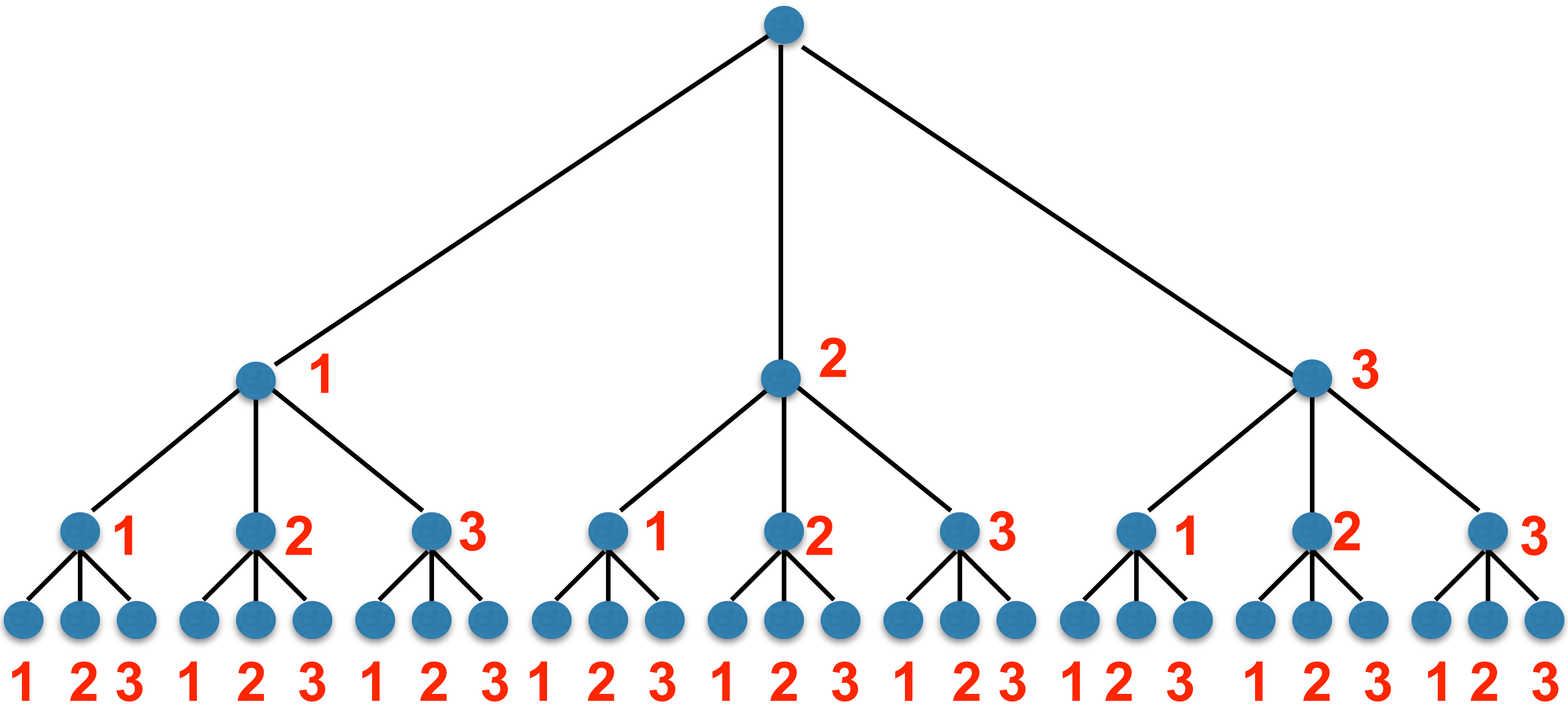
Ora ogni nodo è individuato dalla sequenza delle etichette sui nodi del cammino dalla radice.

Per esempio la sequenza 221 individua il nodo segnalato dalla freccia. Questa sequenza dice che la configurazione corrispondente al nodo si raggiunge compiendo la seconda scelta tra quelle disponibili a partire dalla configurazione iniziale, poi di nuovo la seconda e infine la prima.

La sequenza 322 non corrisponde a una computazione, perché non c'è una seconda scelta possibile dopo la terza e la seconda.

Visitare l’albero leggendo le sequenze

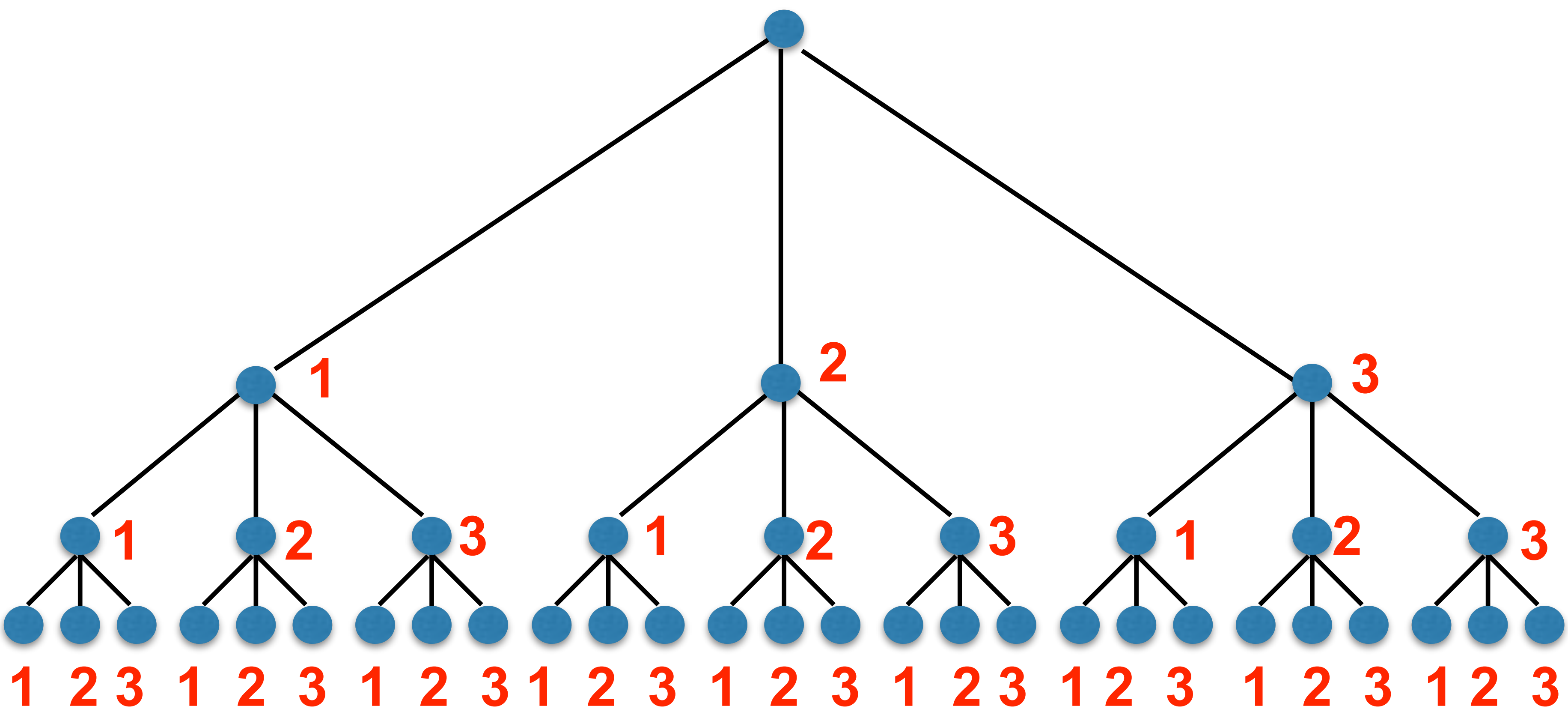
Sia d il massimo grado di non determinismo di una NTM. Se numeriamo i figli dei nodi di un albero d-ario pieno, in ordine progressivo, allora le sequenze che individuano cammini radici - nodo per i nodi di un livello, lette da sinistra verso destra, sono in ordine quasi lessicografico.



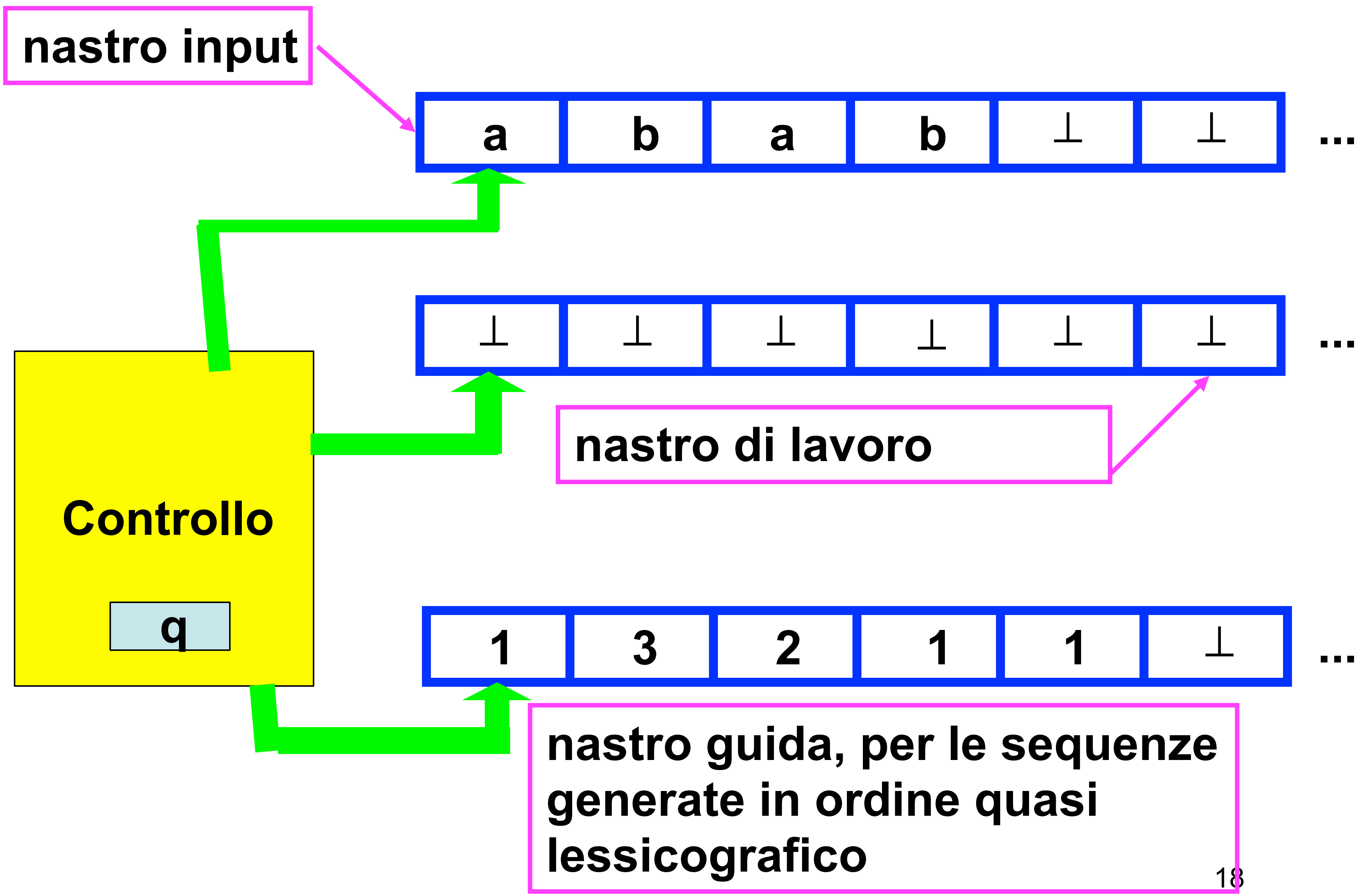
111,112,113,121, ….

Visitare l'albero leggendo le sequenze

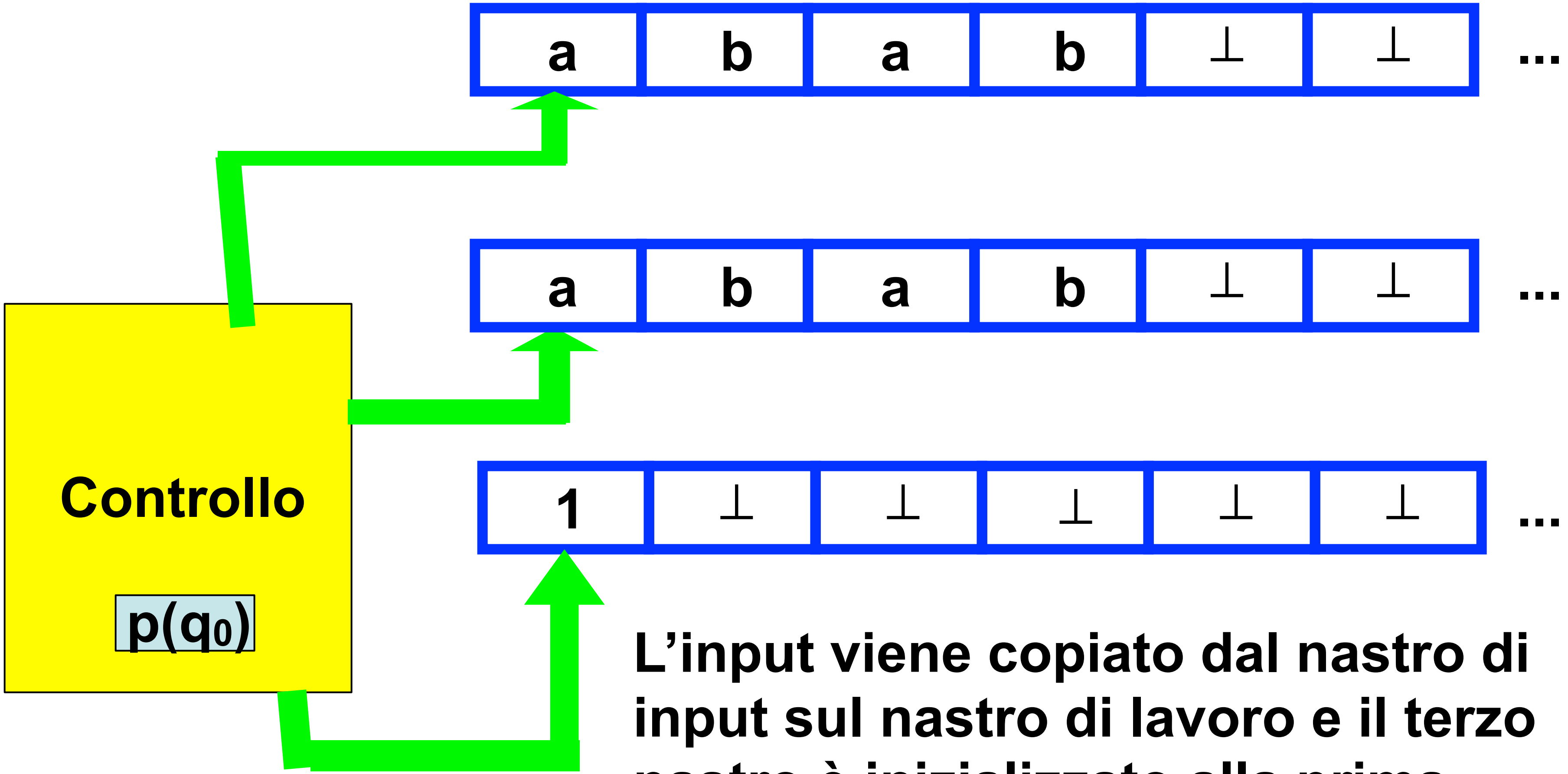
Se generiamo tutte le sequenze in ordine lessicografico sull'alfabeto {1,2,…,d}, generiamo <u>anche</u> tutte le sequenze che corrispondono a computazioni su una TM non deterministica con massimo grado di non determinismo pari a d.



La 3TM M' equivalente alla NTM M

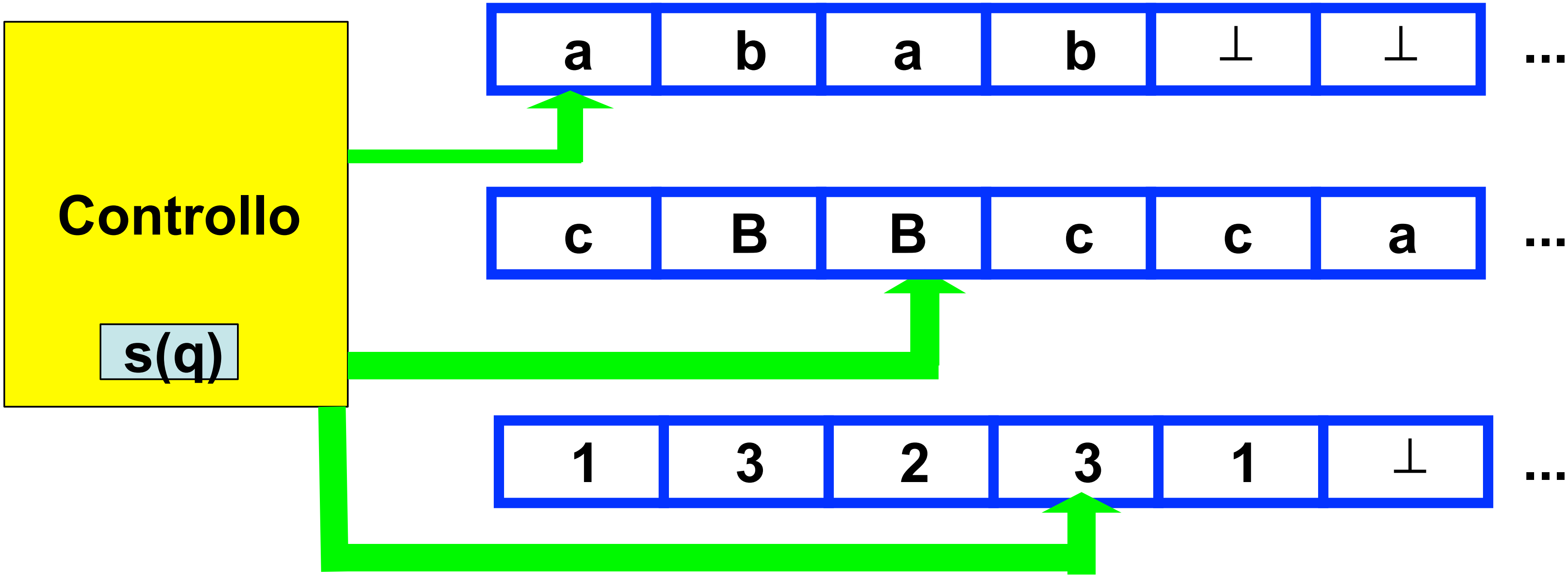


La TM M' equivalente alla NTM M: inizializzazione



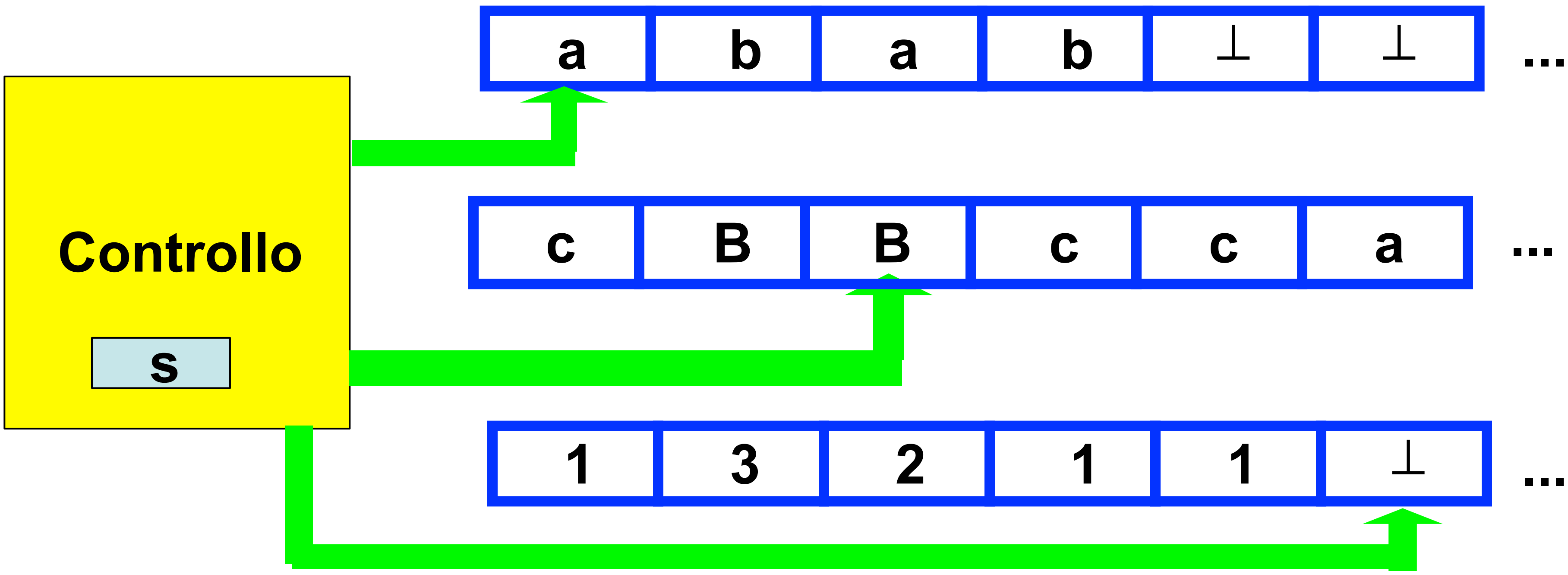
L'input viene copiato dal nastro di input sul nastro di lavoro e il terzo nastro è inizializzato alla prima sequenza tra quelle su {1,..,d}, se d è il massimo grado di non determinismo di M

Come M’ esegue una mossa della NTM M



Nell’esempio viene eseguita la terza tra le mosse possibili nella configurazione raggiunta dopo aver eseguito la sequenza di mosse 132. Se la mossa non c’è, perché |δ(q,B)|< 3, M’ sostituisce la sequenza nel terzo nastro con la sua successiva in ordine lessicografico, ripulendo il secondo nastro, ricopiandovi l’input e riportando le testine sulla prima cella per una nuova esecuzione

La TM M’ equivalente alla NTM M: fase finale dell’esecuzione di una computazione della NTM



Al termine della sequenza di mosse la TM M’ procede in funzione dello stato della macchina simulata M: se è di accettazione la TM M’ si ferma e accetta, altrimenti, anche in caso lo stato sia di rifiuto, M’ genera la sequenza successiva sul terzo nastro, ripulisce il secondo nastro, vi ricopia l’input, e riporta le testine sulla prima cella per cominciare una nuova esecuzione.

La TM M' equivalente a una NTM M

- Sia $M = (Q,\Sigma, \Gamma,\delta,q_0, q_a , q_r)$ la NTM da simulare.
- Sia d = max $\{|\delta(q,a)|$ q in Q e a in Γ }, il grado di non determinismo di M
- Sia G = {1,2…,d}, l'alfabeto delle sequenze guida
- Sia M_0 la TM che genera e scrive sul nastro la sequenza sull'alfabeto G successiva a quella data in input.
- numeriamo ordinandole le k≤d scelte di mosse possibili (p_1,a_1,D_1),…, (p_k,a_k,D_k) in $\delta(q,a)$, con D_1, …,D_k in {L,R}, a in Γ e q in Q, in modo che dato un valore i compreso tra 1 e r questo individui la i-sima mossa.

La TM M’ equivalente a una NTM M

M’: input: $a_1...a_n$

la configurazione iniziale è: $q_0a_1\ldots a_nq_0\perp q_0\perp$, input sul primo nastro e gli altri due vuoti

1. M’ copia l’input sul secondo nastro e inizializza il terzo nastro alla stringa 1, configurazione: $q_0a_1\ldots a_nq_0a_1\ldots a_nq_01$

2. M’ esegue una computazione di M sul nastro 2 facendo la mossa determinata dal simbolo in lettura sul terzo nastro. Se la mossa porta a una configurazione di accettazione di M, anche M’ accetta, se porta a una di rifiuto M’ va alla fase 3. Se il simbolo letto sul terzo nastro non corrisponde a una scelta o è il simbolo di cella vuota M’ va alla fase 3.

3.esegui la TM M_0 per generare sul terzo nastro la successiva sequenza guida e torna al passo 2

La TM M’ equivalente a una NTM M

M' si ferma e accetta quando M si ferma e accetta.

In tutti gli altri casi non si ferma.

Quindi

$$L(M') = L(M)$$