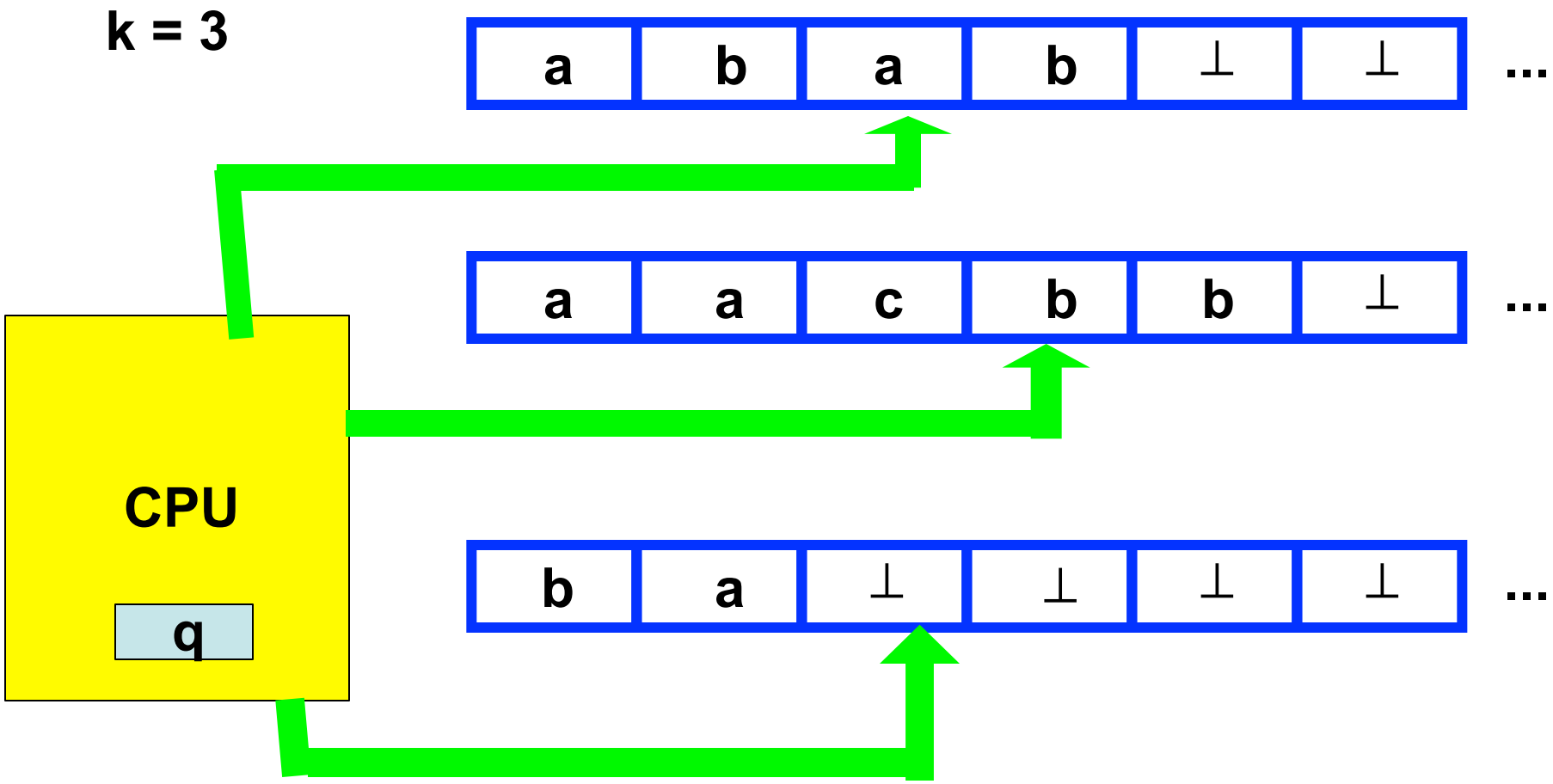
Sommario

- **Variante a più nastri della TM**
- **equivalenza con il modello a un solo nastro**

“Turing showed that such innovations as adding tapes or tape symbols does not increase the set of functions that can be computed by machine”

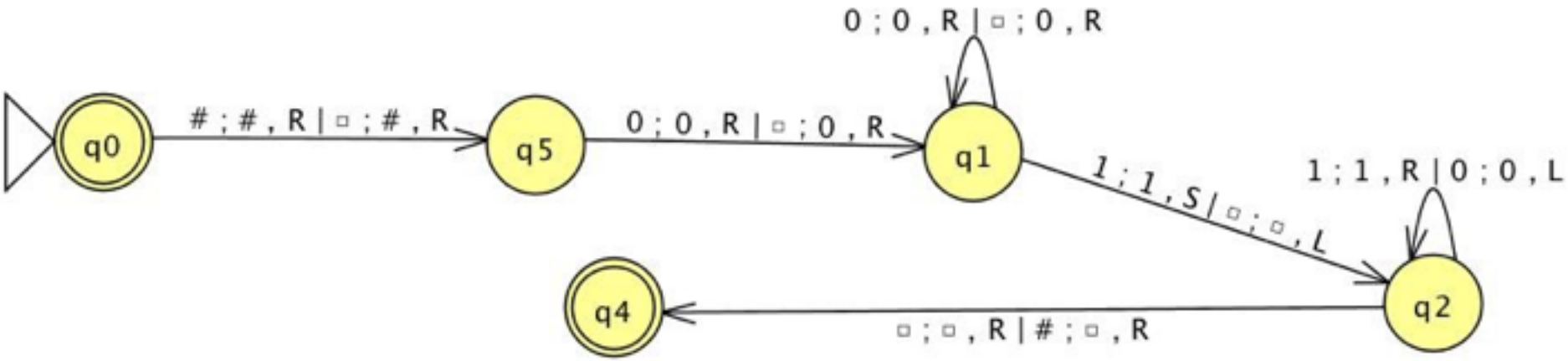
da “On the Computational Complexity of Algorithms” di
J. Hartmanis and R. E. Stearns
pubblicato in Transactions of the American Mathematical Society,
Vol. 117 (May, 1965), pp. 285-306

Macchina di Turing a k nastri



Esempio di TM a due nastri

Una TM a due nastri per $\{0^n1^n \mid n \geq 0\}$:



Macchina di Turing a k nastri

Una Machina di Turing deterministica a k nastri, in breve k-TM , è una settupla

$$M = (Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$$

come nel caso a un nastro dove

$$\delta : (Q - \{ q_a, q_r \}) \times \Gamma^k \longrightarrow Q \times \Gamma^k \times \{L, R\}^k,$$

è la funzione di transizione, che descrive i cambiamenti nelle celle in lettura sui singoli nastri e la direzione di spostamento delle k testine di lettura. Se

$$\delta(q, a_1, \ldots, a_k) = (p, b_1, \ldots, b_k, D_1, \ldots, D_k)$$

con a_i, b_i in Γ, D_i in {L,R}, per i = 1,...,k, si intende che nello stato q leggendo a_1 sul primo nastro,..., a_k sul k-simo, la k-TM entra nello stato p, scrive b_1 sul primo nastro,..., b_k sul k-simo nastro e sposta la prima testina in direzione D_1, ... e la k-sima in direzione D_k.

Configurazioni per una K-TM

Una configurazione deve informare sul contenuto dei nastro, lo stato della macchina e la posizione delle testine di lettura.

Queste informazioni si possono ottenere sinteticamente con una k-pla di sequenze del tipo:

$$\alpha_1 q a_1 \beta_1,\ldots,\alpha_k q a_k \beta_k \text{ in } (\Gamma^* Q \Gamma^*)^k$$

dove α_i e β_i sono stringhe su Γ, a_i è un simbolo di Γ, $\alpha_i a_i \beta_i$ è il contenuto dell'i-simo nastro, a_i è il simbolo in lettura sull'i-simo nastro, per $1 \le i \le k$, e q è lo stato della macchina.

La sequenza $q_0 a_1 \ldots a_n, q_0 \perp, \ldots, q_0 \perp$ in $(\Gamma^* Q \Gamma^*)^k$ dove $a_1 \ldots a_n$ è la stringa input e q_0 è lo stato iniziale è la configurazione iniziale.

Mosse

Un configurazione porta a un'altra se le operazioni di scrittura e di spostamento delle testine sono eseguite in conformità ad una regola descritta dalla funzione di transizione

Una configurazione del tipo $\alpha_1 q_a a_1 \beta_1,\ldots,\alpha_k q_a a_k \beta_k$ è detta di accettazione.

Una configurazione del tipo $\alpha_1 q_r a_1 \beta_1,\ldots,\alpha_k q_r a_k \beta_k$ è detta di rifiuto.

Queste configurazioni sono di terminazione

Linguaggio accettato

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

Il linguaggio accettato (riconosciuto) è

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

Il linguaggio rifiutato è

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

In generale $L(M) \cup R(M) \subseteq \Sigma^*$!

Se $L(M) \cup R(M) = \Sigma^*$ allora vuol dire che la TM si ferma sempre, in tal caso L(M) è il linguaggio deciso dalla TM.

Classe dei linguaggi accettati

L'insieme dei linguaggi che sono accettati da una k-TM è così definito, per ogni k≥1:

$$\mathcal{L}(\text{TMPiùNastri}) =$$

$$\{L \mid \exists\, k \in N, \exists\, M \in k\text{-TM}, \text{ e } L(M) = L\}$$

Macchina di Turing a k nastri

DOMANDA:

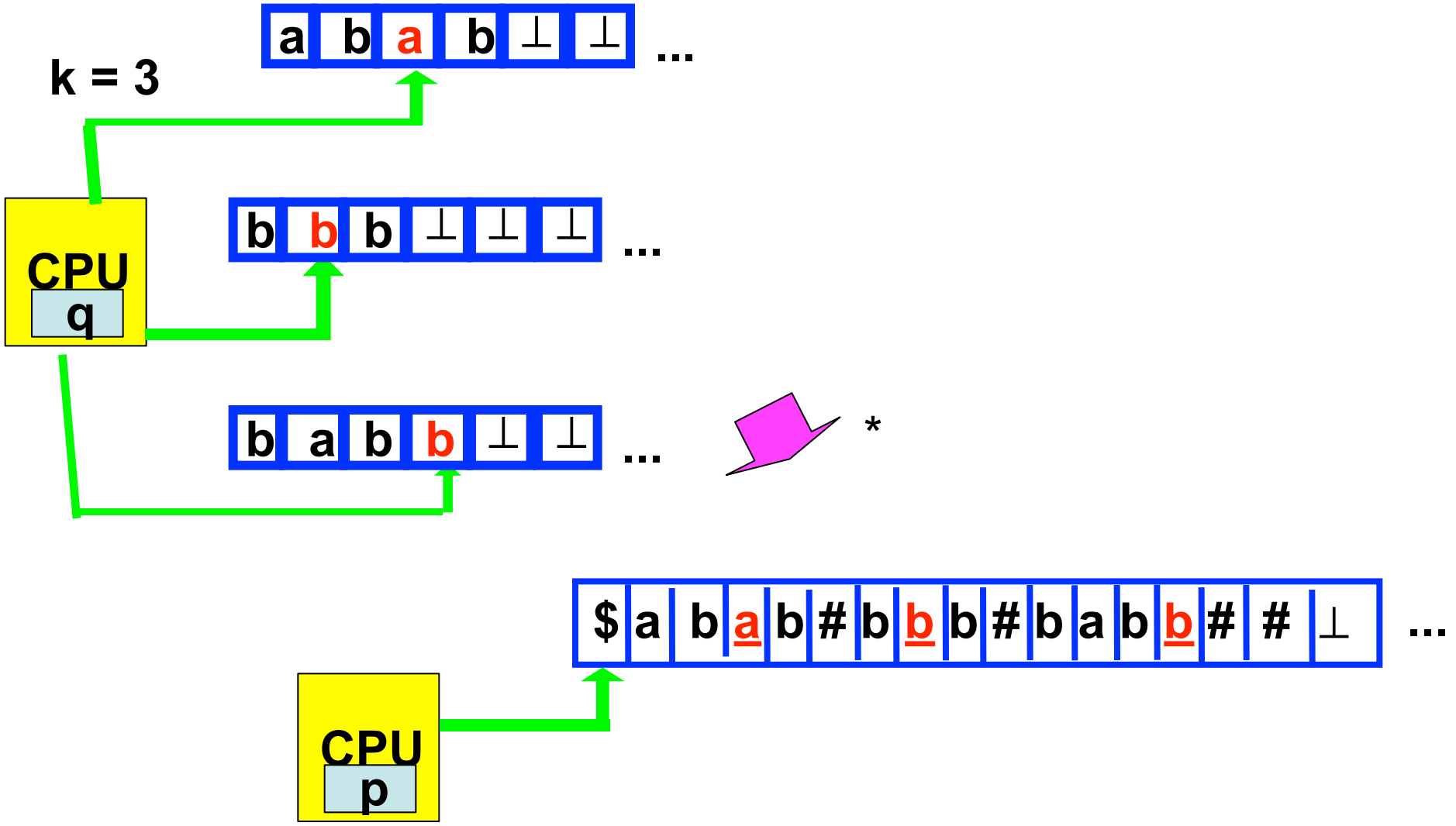
La versione a k nastri ha un potere computazionale maggiore??

RISPOSTA: NO.

Possiamo dimostrare l'esistenza di una TM a un solo nastro equivalente a una TM a k nastri.

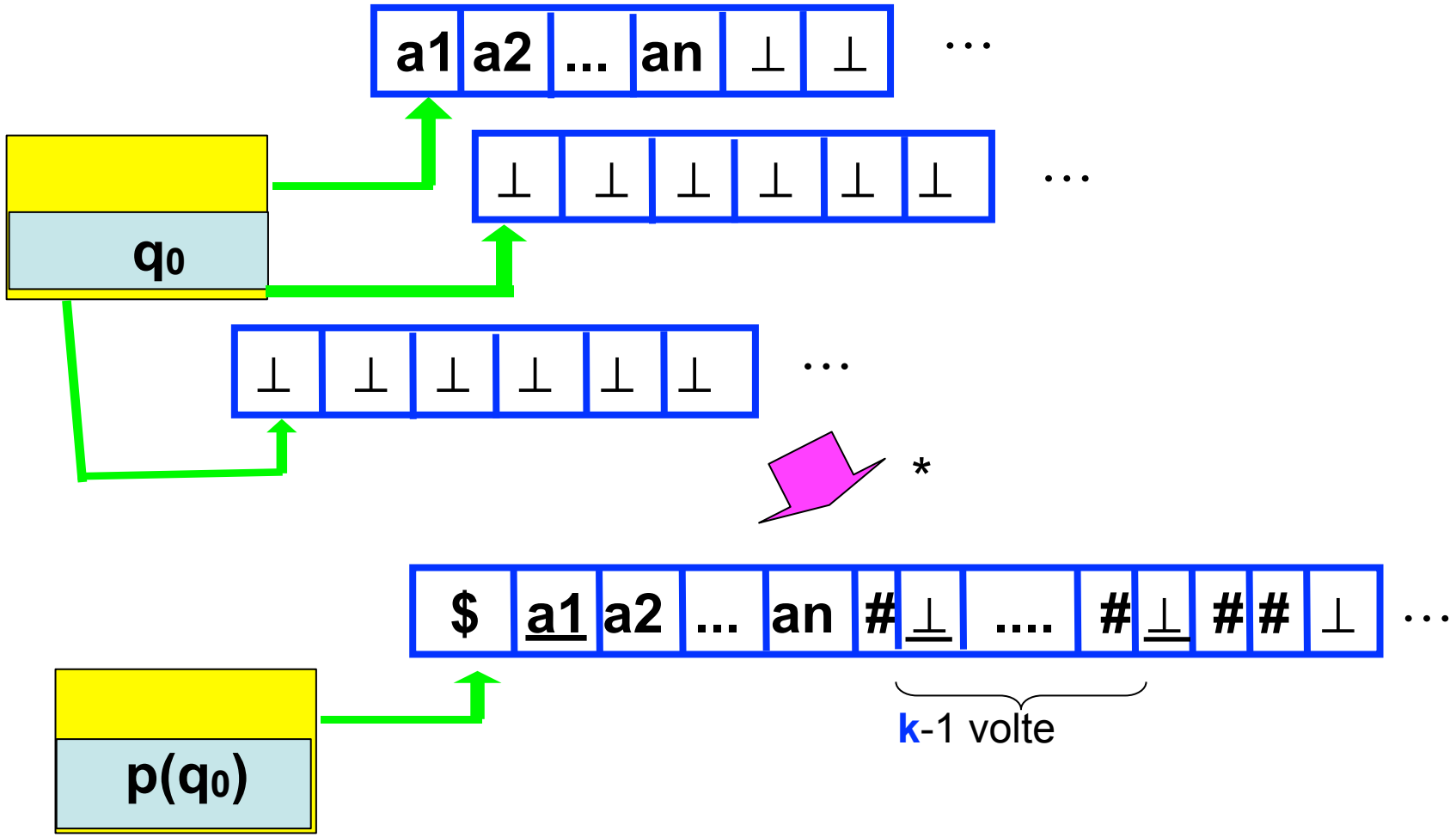
Quindi $\mathcal{L}$(TM) = $\mathcal{L}$(TMPiùNastri).

Costruzione di una TM equivalente a una a k nastri: la rappresentazione dei nastri



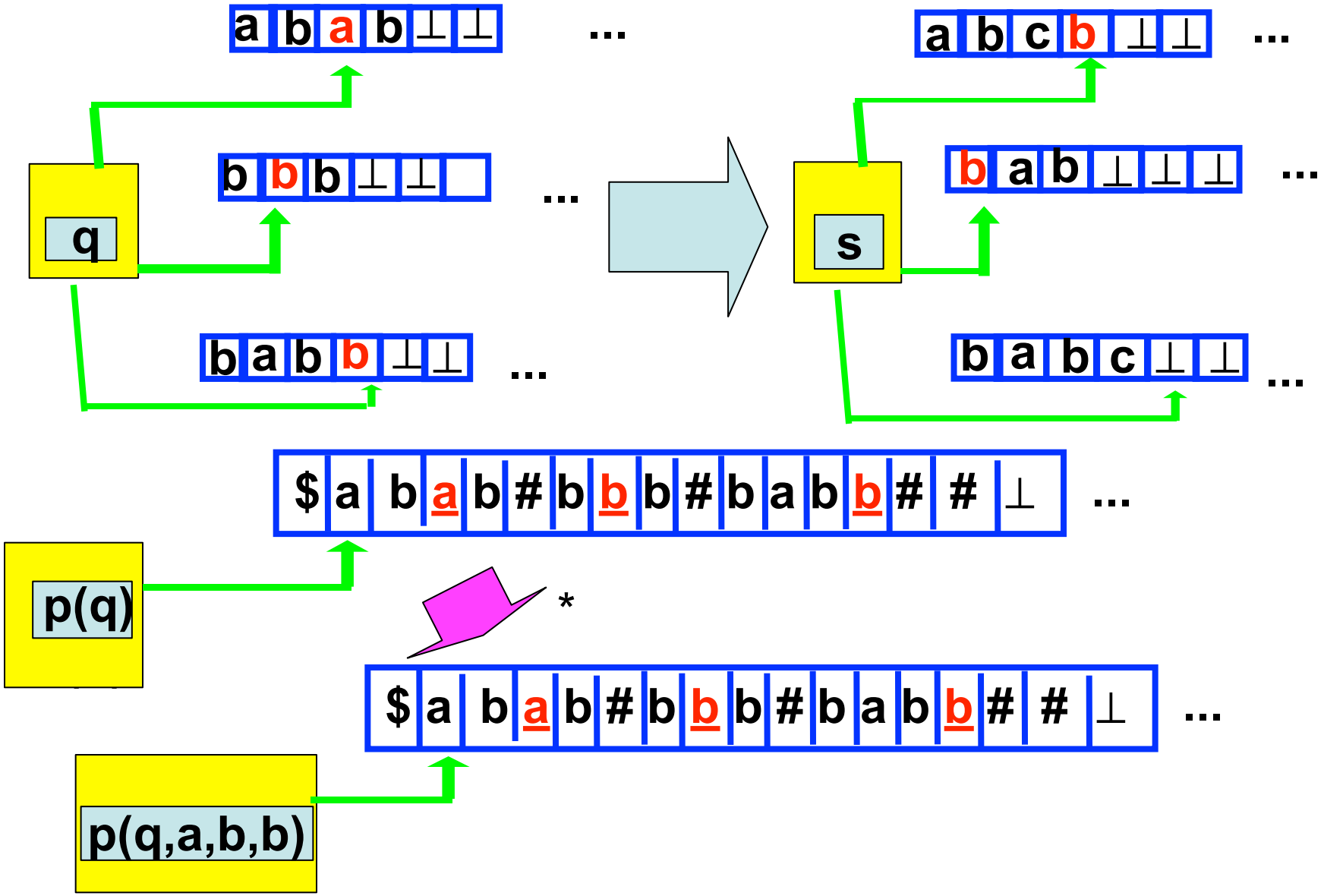
Inizio simulazione

Detta $a_1 \ldots a_n$ la stringa input, la TM simulante deve configurare il nastro in modo da poter eseguire le mosse della TM a k nastri.



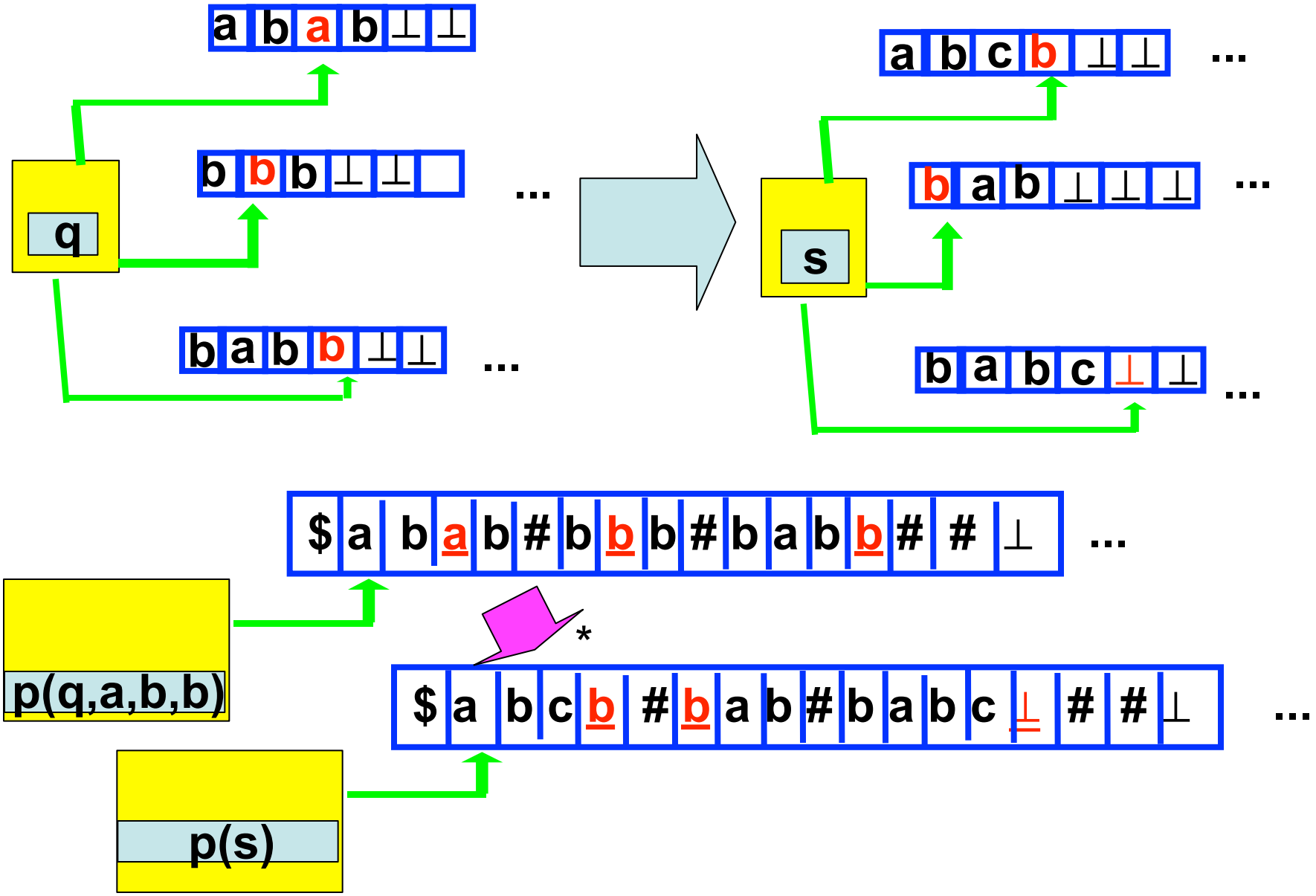
k = 3

Simulazione di una mossa: parte 1



k = 3

Simulazione di una mossa: parte 2



Descrizione a livello di implementazione della costruzione di una TM equivalente a una a k nastri:

input: una TM $M = (Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$ a **k** nastri

output: una TM a un solo nastro equivalente

1. le prime mosse della TM servono a inizializzare il nastro alla configurazione: $\$p(q_0)\underline{a_1}...a_n$ $\#\underline{\perp}\# \ldots \underline{\perp}\#\#$,

dove $a_1...a_n$ è la stringa input

k-1 volte

2. per ogni mossa del tipo $\delta(q,a_1,...,a_k) = (s,b_1,...,b_k,D_1,...,D_k)$ la TM scorre il nastro verso destra, memorizzando nello stato i simboli in lettura sui singoli nastri; riposiziona la testina all'inizio del nastro, poi scorre il nastro di nuovo verso destra eseguendo sulla porzione i-sima di nastro le azioni che simulano scrittura e spostamento della testina sull'i-simo nastro, per i= 1,...,k. Infine la TM entra nello stato che ricorda il nuovo stato s e riposiziona la testina all'inizio del nastro.

Nota alla costruzione di una TM equivalente a una a k nastri

Se la simulazione della mossa porta a aggiungere una cella vuota all'i-sima porzione di nastro sarà necessario spostare di una posizione a destra tutto il contenuto dei nastri, a partire dalla cella a destra della posizione della testina che ha bisogno di una nuova cella.