

# Sommario

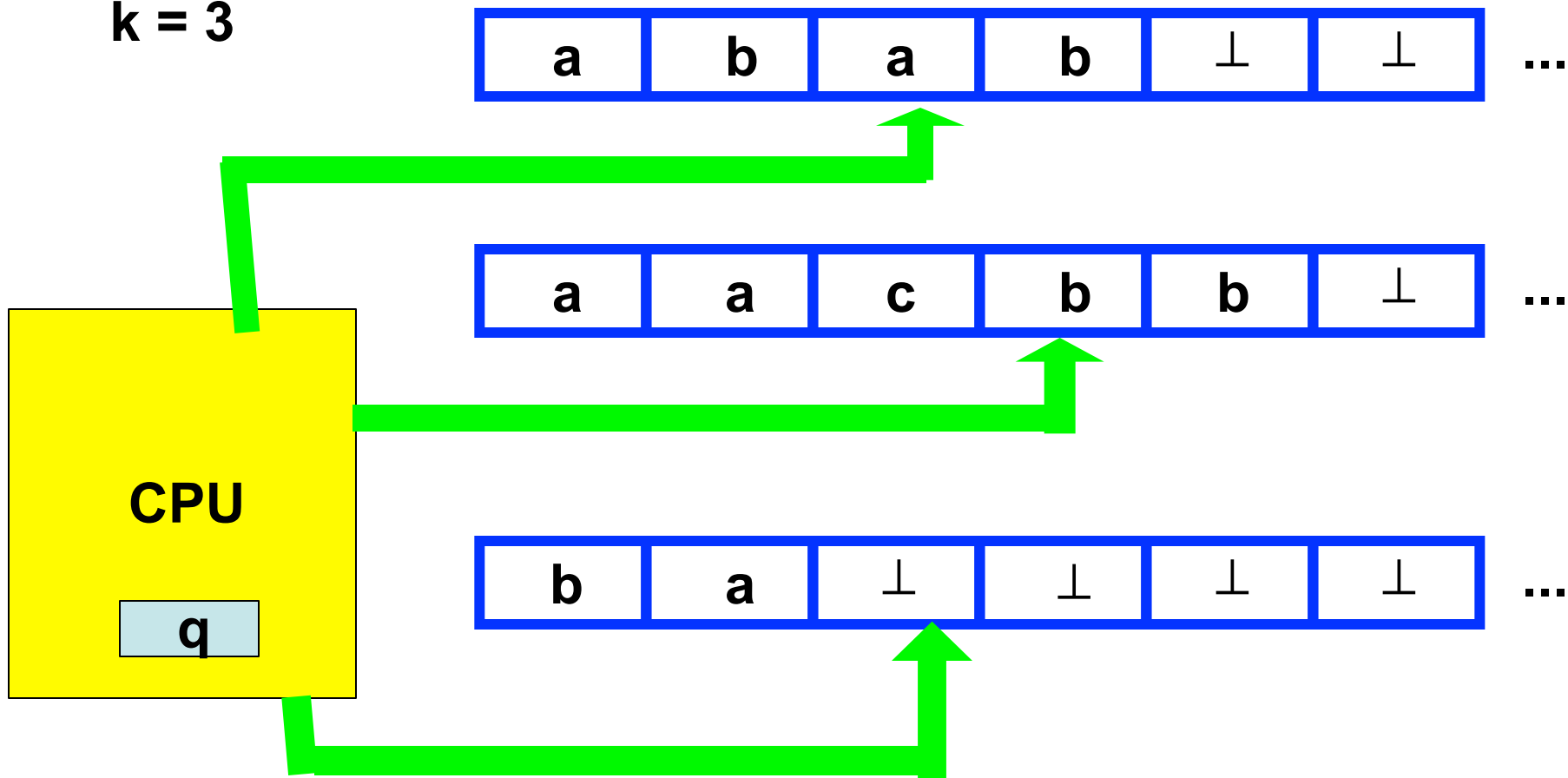
- **Variante a più nastri della TM**
- **equivalenza con il modello a un solo nastro**

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

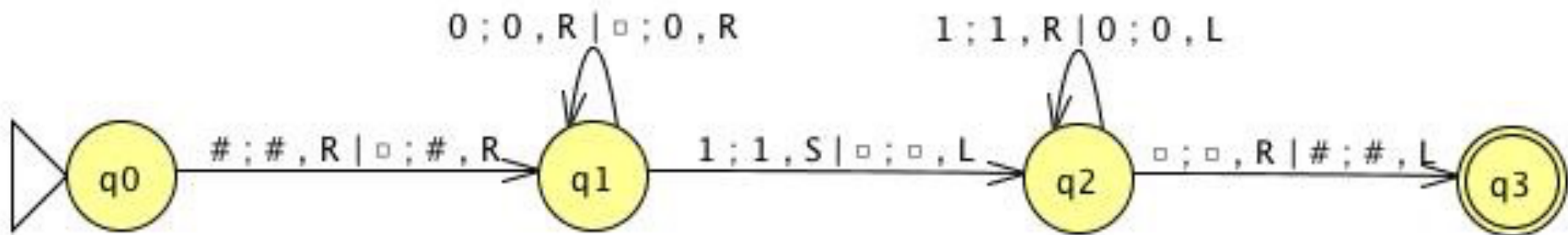
da “On the Computational Complexity of Algorithms” di  
J. Hartmanis and R. E. Stearns  
pubblicato in Transactions of the American Mathematical Society,  
Vol. 117 (May, 1965), pp. 285-306

# Macchina di Turing a k nastri

$k = 3$



2TM per  $L = \{0^n 1^n \mid n \geq 0\}$



# Macchina di Turing a k nastri

Una **Macchina di Turing deterministica a k nastri**, in breve **kTM**, è una settupla

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

come nel caso a un nastro dove

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

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

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

con  $a_i, b_i$  in  $\Gamma$ ,  $D_i$  in  $\{L, R\}$ , per  $i = 1, \dots, k$ , 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 del nastro, lo stato della macchina e la posizione delle testine di lettura.

Queste informazioni si possono ottenere sinteticamente con  $k$  sequenze del tipo

$$\alpha_1 qa_1 \beta_1, \dots, \alpha_k qa_k \beta_k \text{ in } (\Gamma^* Q \Gamma^*)^k$$

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

# Mosse

Una 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

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

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

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

Queste configurazioni sono di terminazione

# Linguaggio accettato

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

Il linguaggio **accettato** (riconosciuto) è

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

Il linguaggio **rifiutato** è

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

**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  $k \geq 1$ :

$$\mathcal{L}(\text{TM Più Nastri}) = \{L \mid \exists k \in \mathbb{N}, \exists M \in k\text{TM}, e L(M) = L\}$$



# Macchina di Turing a k nastri

## DOMANDA:

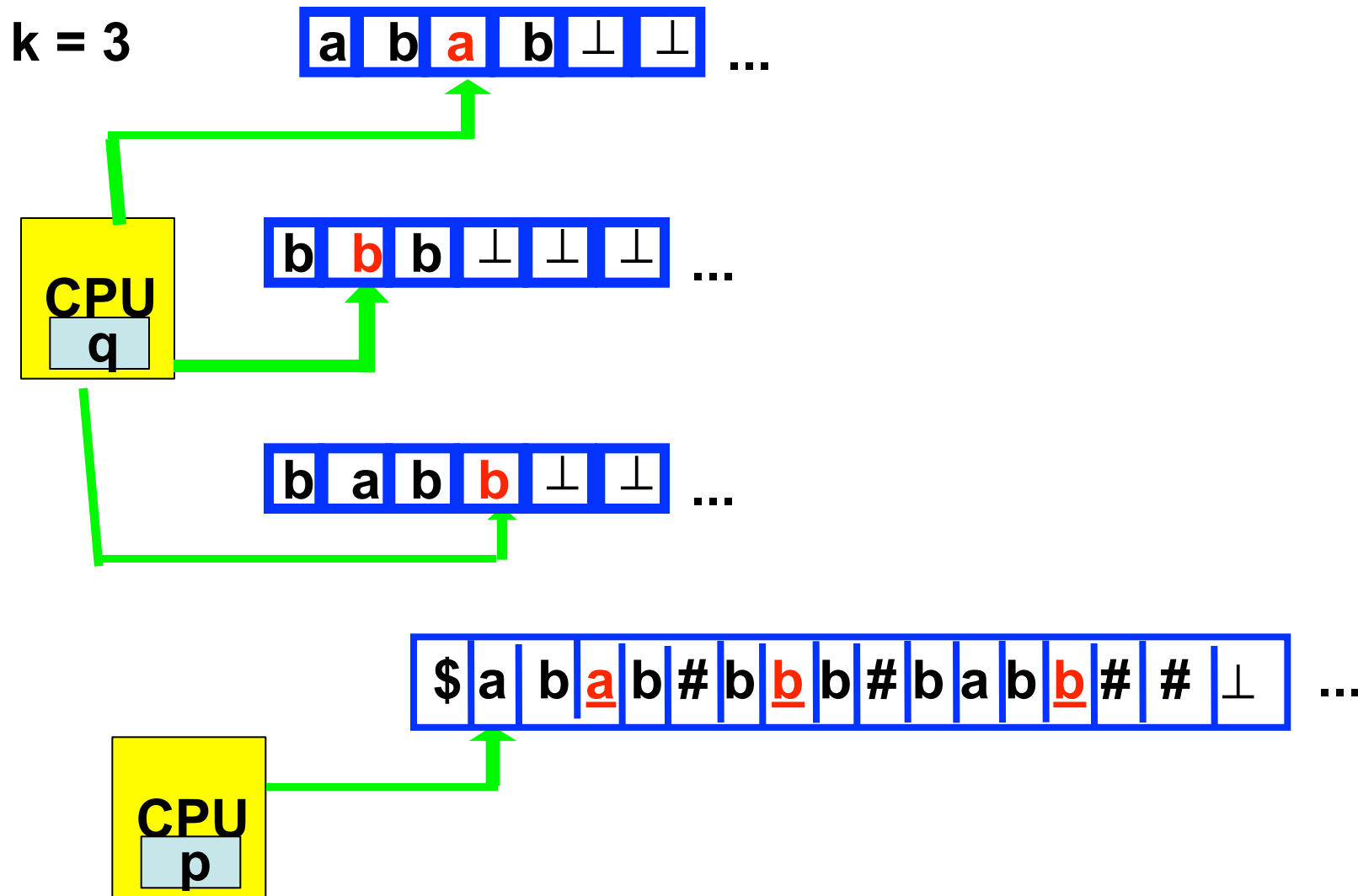
La versione a k nastri ha un potere computazionale maggiore??

## RISPOSTA: NO.

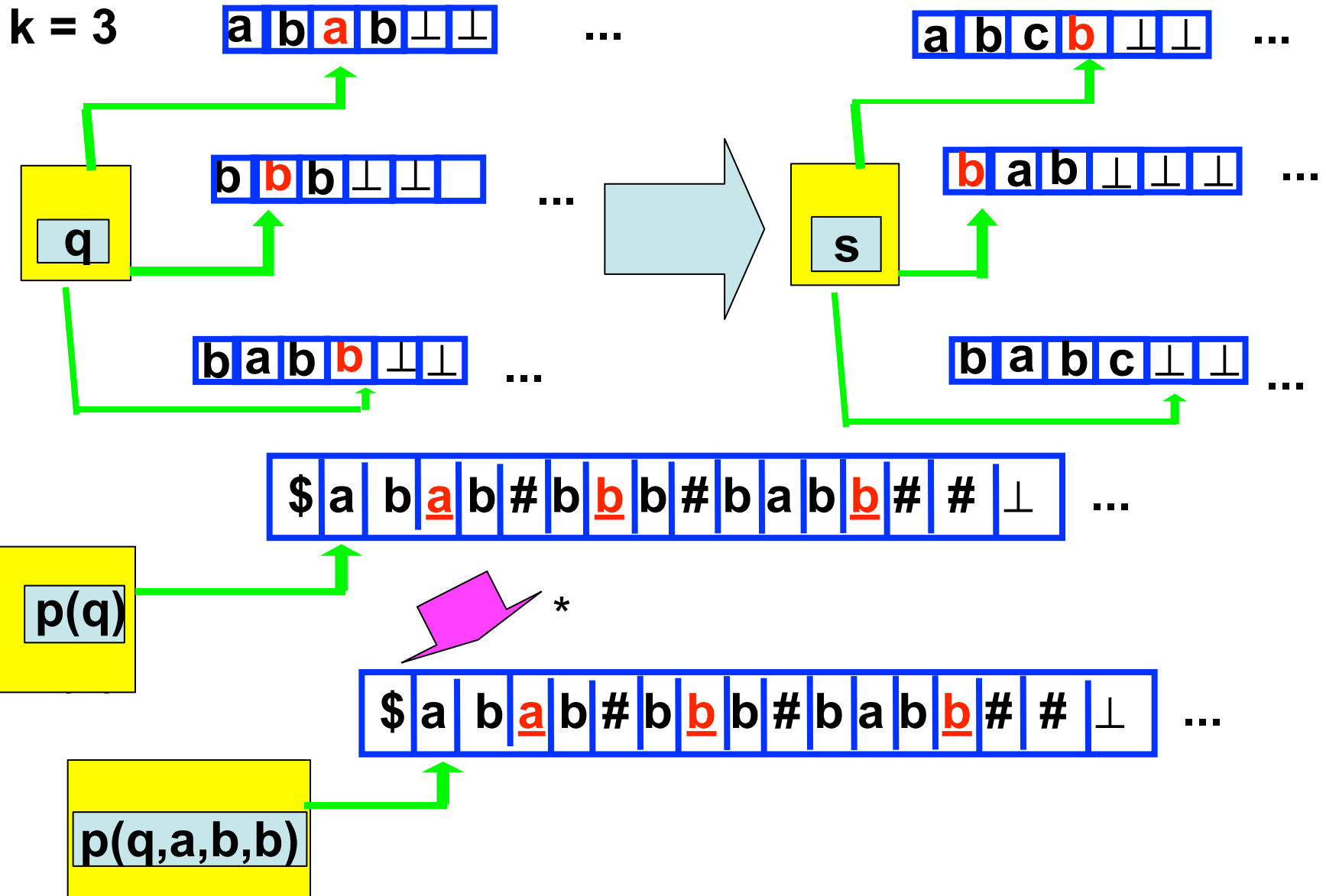
Possiamo dimostrare l'esistenza di una TM a un solo nastro equivalente a una TM a k nastri.

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

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

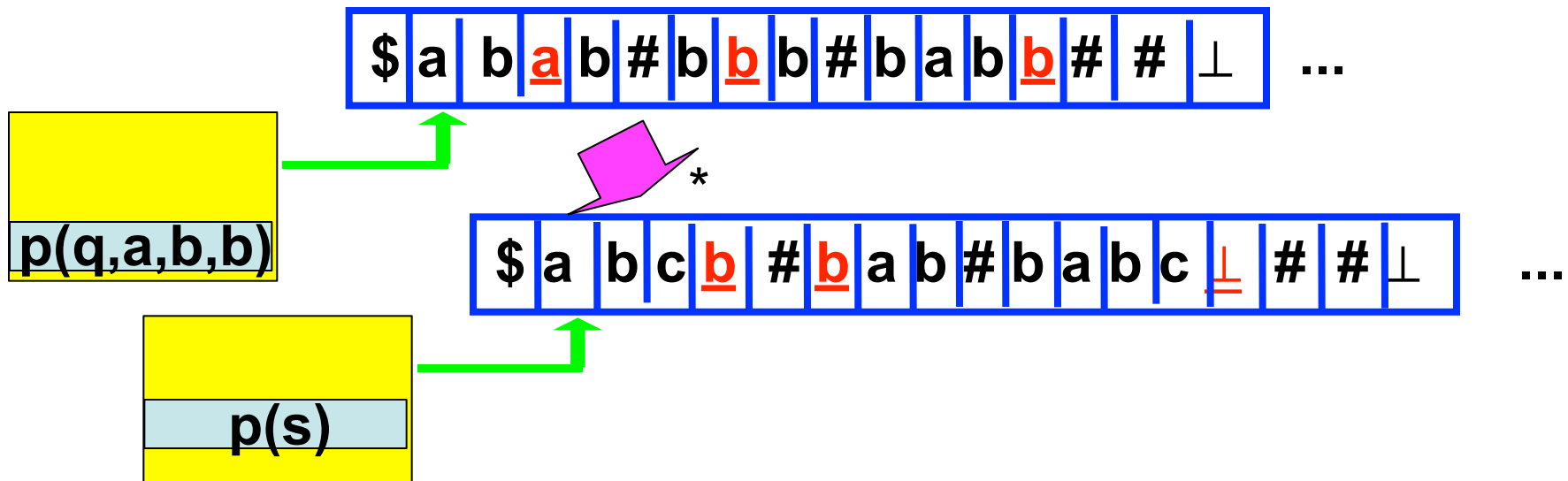
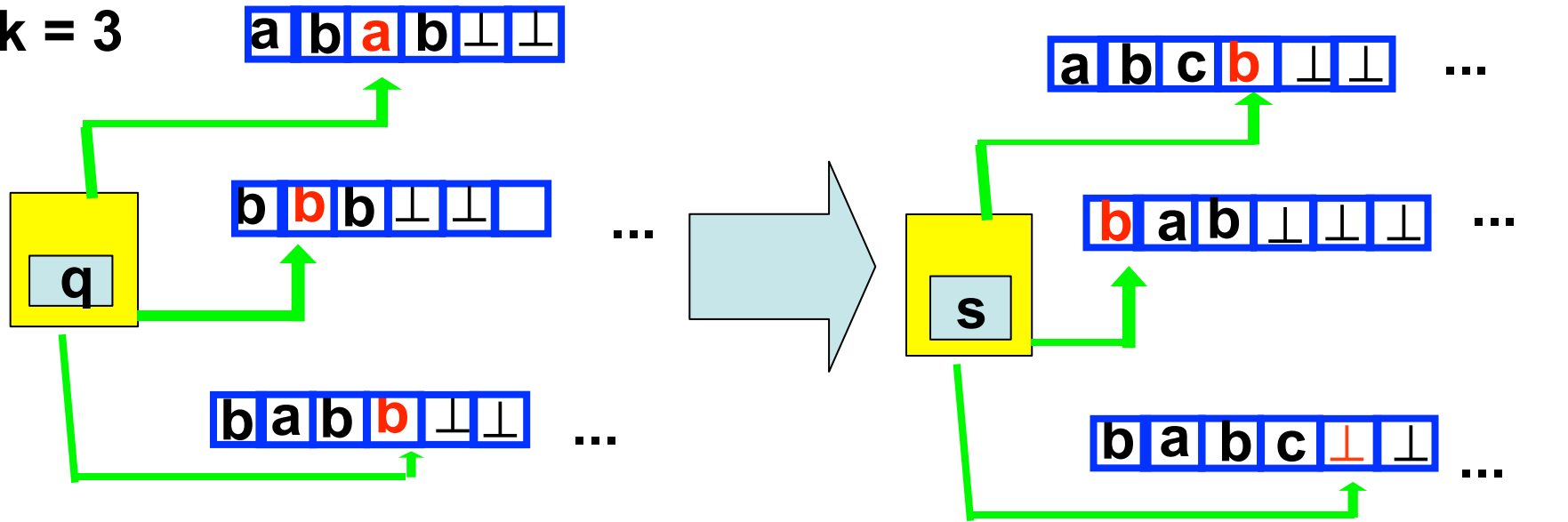


# Simulazione di una mossa: parte 1



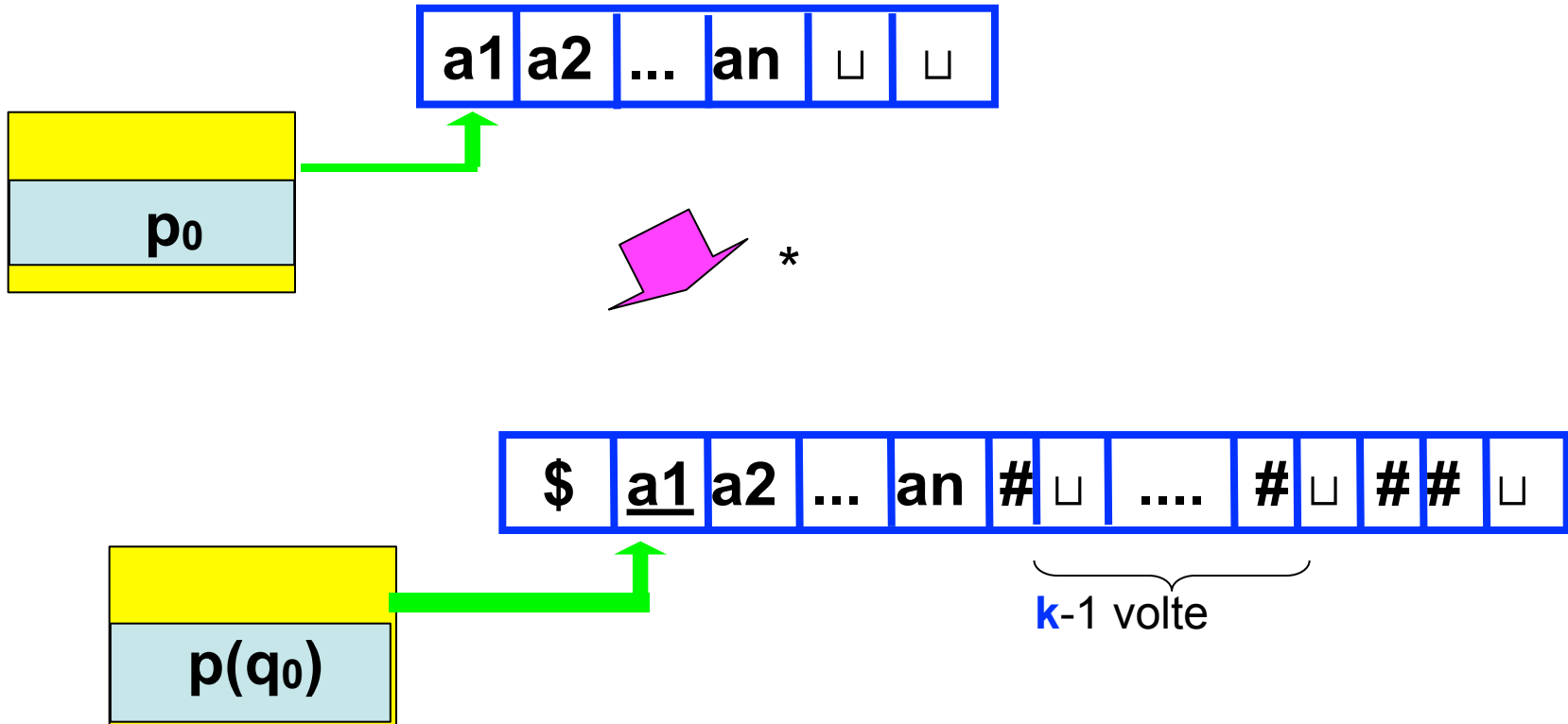
# Simulazione di una mossa: parte 2

$k = 3$



# Inizio simulazione

Detta  $a_1 \dots a_n$  la stringa input, la TM simulante deve configurare il nastro in modo da poter eseguire le mosse della TM a  $k$  nastri. Lo stato iniziale della TM simulante è  $p_0$  e  $q_0$  quello della  $k$ TM.





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

**Se la simulazione della mossa porta a aggiungere una cella vuota a una delle porzioni di nastro tra due # sarà necessario spostare di una posizione a destra tutto il contenuto dei nastri, a partire dalla cella a destra della posizione della testina che deve spostarsi a destra.**