

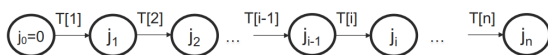
# Lezione 11 17/11/2023

1. Costruzione dell'automa per il pattern P  
→ Calcolo della funzione  $\delta$  in tempo  $\theta(m|\Sigma|)$

2. Uso dell'automa per riconoscere, in un testo T definito su alfabeto  $\Sigma$ , tutte le occorrenze esatte di P  
→ Scansione del testo T in tempo  $\theta(n)$

## Scansione del testo

- Stato iniziale  $\rightarrow j_0 = 0$
- Leggo T[1] e arrivo allo stato  $j_1 = \delta(j_0, T[1])$
- Leggo T[2] e arrivo allo stato  $j_2 = \delta(j_1, T[2])$
- ...
- Leggo T[i] e arrivo allo stato  $j_i = \delta(j_{i-1}, T[i])$
- ...
- Leggo T[n] e arrivo allo stato  $j_n = \delta(j_{n-1}, T[n])$



T

1	2	3	4	5	6	7
c	a	b	a	c	a	b

0

$\delta$	a	b	c
0	1	0	0
1	1	0	2
2	3	0	0
3	1	0	4
4	3	0	0

a	c	a	c
---	---	---	---

P

0 1 2 3 4

↑

Si parte dallo stato iniziale  $j_0 = 0$ , si legge il primo simbolo del testo che viene dato in input all'automa, che fa la transizione tramite T[1]. Quindi legge la funzione di transizione.

Passo poi da  $j_1$  a  $j_2$  e così via.

Ne passo generico ci immaginiamo di essere arrivati allo stato  $j_{i-1}$  e andiamo nello stato  $j_i$  tramite T[n].

Arriviamo allo stato  $j_i$  quando viene letto il carattere  $i$ .

Parto dalla posizione 0, prima dell'inizio del pattern. Leggo la c, guardo la tabella di transizione e vedo che rimango in 0.

Leggo quindi a, vado nella posizione 1 dell'automa.

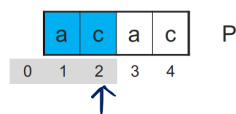
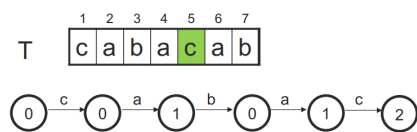
a	c	a	c
---	---	---	---

P

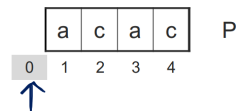
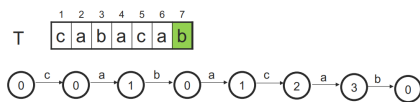
0 1 2 3 4

↑

Quando poi leggo b, la terza posizione della stringa, torno a 0. Poi con a vado nella posizione 1.



Con la c quindi arrivo nello stato 2 dell'automa.



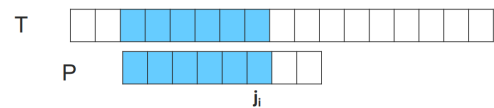
Alla fine mi ritroverò di nuovo nella posizione 0 perché leggo *b*.

Stati dell'automa passati durante la scansione = cardinalità del testo + 1.

Non essendo mai arrivati allo stato 4, non abbiamo trovato il pattern nella stringa.

Successione di stati corrisponde a:

- ➔ successione di posizioni su P
- ➔ successione di lunghezze di prefissi di P

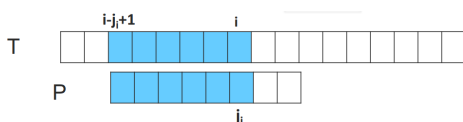


**TEOREMA:**  $j_i$  ( $0 \leq i \leq n$ ) è la lunghezza del più lungo prefisso di P uguale a una sottostringa di T che finisce in posizione  $i$

Lo stato a cui arriviamo nella posizione  $i$  ci dà la lunghezza del più lungo prefisso del pattern uguale ad una sottostringa di T che finisce in  $i$ .

Qui quindi ci fermiamo alla posizione  $i$  e consideriamo il prefisso che finisce in posizione  $i$ , quindi fino alla posizione  $i$ , senza considerare i possibili prefissi successivi più lunghi.

Quindi siamo sicuri che non esiste un prefisso del pattern più lungo, che finisce in posizione  $i$ .



$$T[i-j_i+1, i] = P[1, j_i]$$

Questo è quello che ci dice il teorema.

Se però  $j_i = m$  abbiamo trovato un'occorrenza esatta.

$$j_i = m \Rightarrow T[i-m+1, i] = P[1, m] = P$$

$i-m+1$  è occorrenza esatta

## Dimostrazione



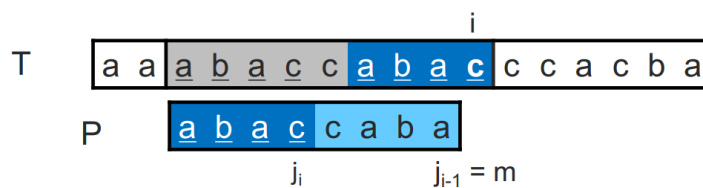
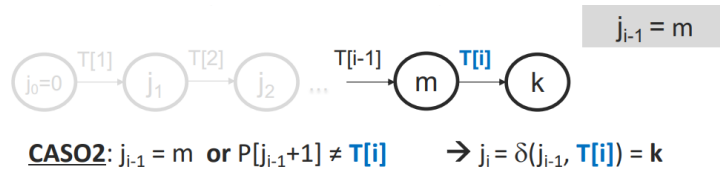
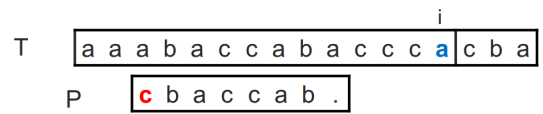
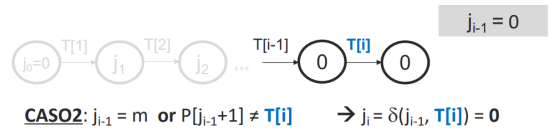
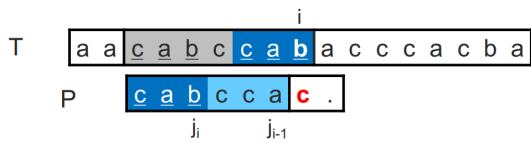
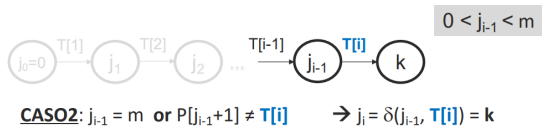
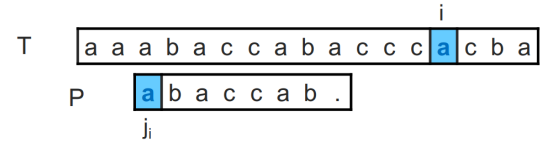
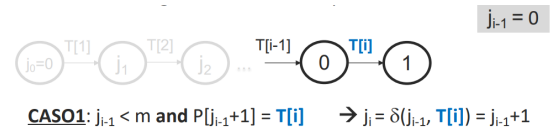
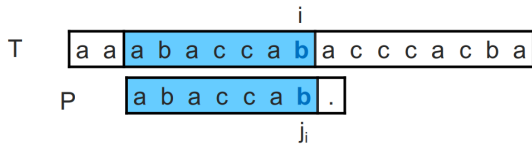
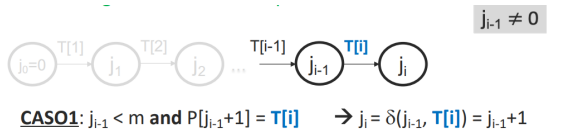
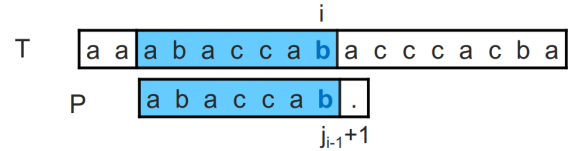
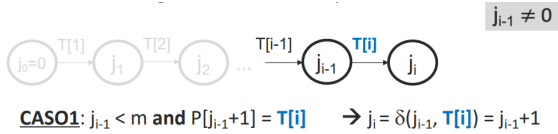
Per lo stato  $j_0 = 0$  il teorema è banalmente dimostrabile



#### PASSO INDUTTIVO

SE  $j_{i-1}$  è la lunghezza del più lungo prefisso di P uguale alla sottostringa di T che finisce in posizione  $i-1$ ,

ALLORA  $j_i$  è la lunghezza del più lungo prefisso di P uguale alla sottostringa di T che finisce in posizione  $i$



1. Si parte dallo stato iniziale 0 e si effettua una scansione di T dal primo all'ultimo simbolo
2. Per ogni posizione i di T si effettua la transizione dallo stato corrente  $j_c$  al nuovo stato  $j_f = \delta(j_c, T[i])$
3. Ogni volta che lo stato  $j_f$  è lo stato accettante m, viene prodotta in output l'occorrenza  $i - m + 1$

```

Procedura ASF_exact_occurrences( $\delta, T, m$ )
begin
  n  $\leftarrow$  |T|
  j  $\leftarrow$  0
  for i  $\leftarrow$  1 to n do
    j  $\leftarrow$   $\delta(j, T[i])$ 
    if j = m then
      output i-m+1 //occorrenza di P in T
  end

```

Tempo  $\rightarrow \theta(n)$

Algoritmo di ricerca esatta con l'automa a stati finiti.

## Esercizio (che può capitare all'esame)

T    

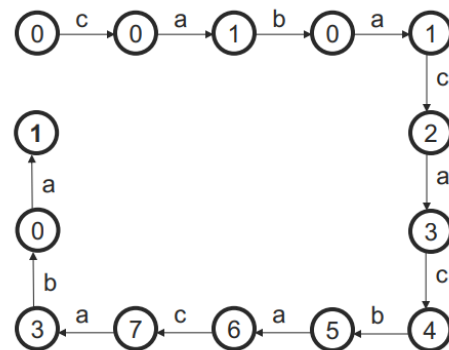
1	2	3	4	5	6	7	8	9	10	11	12	13
c	a	b	a	c	a	c	b	a	c	a	b	a

    m = 7

$\delta$	a	b	c
0	1	0	0
1	1	0	2
2	3	0	0
3	1	0	4
4	3	5	0
5	6	0	0
6	1	0	7
7	3	0	0

P    

a	c	a	c	b	a	c
---	---	---	---	---	---	---



Quando arrivo nello stato 1 so che il più lungo prefisso del pattern che ha un'occorrenza esatta che finisce nella posizione 2 di T è a.

Quando torno sullo 0 significa che non c'è un prefisso esatto del pattern che finisce in quella cella.

Quando arrivo nello stato 7 ho trovato un'occorrenza esatta del pattern che finisce nella posizione 10 della stringa.