Lezione 12 22/11/2023

Il primo passo è quello di trovare le proprietà di un problema, per ottenere un algoritmo efficiente. Riprendiamo l'esempio di un paio di lezioni fa.

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

m = 7 Σ = {a,b,c}

δ(j,σ)	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

δ(j, σ) = j+1

se 0 ≤ j < m **and** P[j+1] = σ

Dovevamo specificare la funzione di transizione per riconoscere questo pattern.

Avevamo riempito prima di tutto le caselle che rappresentano le transizioni da *j* a *j+1* (caselle arancioni).

Poi avevamo scritto l'algoritmo banale: costruisco una tabella vuota di m+1 righe, e poi facciamo un for per riempire le caselle arancioni.

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

m = 7 Σ = {a,b,c}

δ(j,σ)	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

δ(j, σ) = |B(P[1,j] σ)|

se j = m **or** P[j+1] ≠ σ

Le caselle verdi sono invece quelle relative al bordo.

Serve quindi un secondo for.

```
Procedure Trivial-build-transition-function(P)
begin
    m ← |P|
    δ ← empty table (m+1)x|Σ|

O(m)     for j ← 0 to m-1 do
             δ(j, P[j+1]) ← j+1

O(m)     for j ← 0 to m do
             foreach σ in Σ do
O(|Σ|)           δ(j,σ) ← |B(P[1,j] σ)|      O(m²)

    return δ
end
```

Tempo $O(m^3|\Sigma|)$

m^3 non è bello perchè se l'input è molto grande, l'algoritmo (banale) diventa troppo lento.

Non conoscendo bene le proprietà del problema, abbiamo raggiunto un algoritmo troppo lento. Cerchiamo ora di sfruttarle per raggiungere un algoritmo più efficiente.

Definizione di δ_j

$\delta_j \rightarrow$ funzione di transizione di P[1,j]

δ_j: {0, 1, 2,..., j} x $\Sigma \rightarrow$ {0, 1, 2, ..., j}

$\delta_0 \rightarrow$ funzione di transizione di P[1,0] = ε

δ_0: {0} x $\Sigma \rightarrow$ {0}

δ_0	σ_1	σ_2	σ_3	…	$\sigma_{\lvert\Sigma\rvert}$
0	0	0	0	…	0

$\delta_m \rightarrow$ funzione di transizione di P[1,m] $\rightarrow$ funzione δ

δ_m: {0, 1, 2,..., m} x $\Sigma \rightarrow$ {0, 1, 2, ..., m}

Seguendo la prima definizione, la delta 0 cosa sarà? Mettiamo 0 al posto di j, e troviamo che questa è la funzione di transizione di una stringa nulla. Abbiamo quindi un solo stato possibile in cui possiamo trovarci, lo zero.

La delta m (dove poni j=m) è la funzione di transizione che ha nel dominio e nel codominio m.

Ci sono quindi tante funzioni di transizioni parziali.

Calcolo di δ per un pattern P di lunghezza m:

1. Calcolo $\delta_0 \rightarrow$ **CASO BASE**
2. Calcolo δ_1 da δ_0
3. Calcolo δ_2 da δ_1
4. …
5. Calcolo δ_j da $\delta_{j-1} \rightarrow$ **PASSO INDUTTIVO**
6. …
7. Calcolo δ_m da δ_{m-1}

$\delta_m \rightarrow$ funzione di transizione δ

La funzione è quindi definita per induzione.

Se riusciamo a definire un algoritmo che ci permette data delta j-1 di permette di calcolare la successiva delta j abbiamo l'algoritmo.

Dobbiamo calcolare il passo induttivo. Lo vediamo tramite un esempio che poi generalizziamo, senza vedere definizioni formali.

Esempio di δ_4 e δ_5 per P = abccbab

La delta 4 è la funzione di transizione per un prefisso lungo 4, mentre delta 5 per un prefisso lungo 5. Riempiamo ora le due tabelle come se le due fossero scollegate.

Andiamo a vedere le differenze tra le due, e useremo queste per trovare l'algoritmo che trasforma la delta 5 e la delta 4.

Abbiamo quindi una riga in più in delta 5 perché abbiamo uno stato in più.

P_4 = abccbab

δ_4	a	b	c	d
0	1	0	0	0
1	1	2	0	0
2	1	0	3	0
3	1	0	4	0
4	1	0	0	0

P_5 = abccbab

δ_5	a	b	c	d
0	1	0	0	0
1	1	2	0	0
2	1	0	3	0
3	1	0	4	0
4	1	5	0	0
5	1	0	0	0

Vediamo anche che nelle righe da 0 a 3 non ci sono differenze. Questa in blu è la formalizzazione.

$$\delta_4(p, \sigma) = \delta_5(p, \sigma) \text{ per } 0 \leq p < 4 \text{ e } \forall\, \sigma \in \Sigma$$

Per definire la delta 4 p, sigma posso dire che questa è p+1 quando p<4 e il simbolo in posizione p+1 è il simbolo di transizione. La delta 5 p sigma è la stessa cosa, ma con p<5 e il prefisso lungo 5. Però noi siamo nel caso in cui gli stati vanno da 0 a 3, non arriviamo a 4. Quindi p<4 e p<5 sono vere sempre. Di conseguenza le espressioni si riducono solo alla seconda parte che controlla che il simbolo sia quello corretto.

$$\delta_4(p,\sigma) = p+1 \qquad \text{se } p < 4 \textbf{ and } P_4[p+1] = \sigma$$
$$\delta_5(p,\sigma) = p+1 \qquad \text{se } p < 5 \textbf{ and } P_5[p+1] = \sigma$$

$$\delta_4(p,\sigma) = p+1 \qquad \text{se } P_4[p+1] = \sigma$$
$$\delta_5(p,\sigma) = p+1 \qquad \text{se } P_5[p+1] = \sigma$$

Il simbolo successivo però è lo stesso perché stiamo considerando i prefissi lunghi 3. Quindi la delta 5 è uguale alla delta 4 se p<4 e il prossimo simbolo è sigma.

$$\Rightarrow \delta_5(p,\sigma) = \delta_4(p,\sigma) \quad \text{se } p < 4 \textbf{ and } P_5[p+1] = \sigma$$

Dobbiamo dimostrare questa cosa anche per un generico k, non solo il sigma simbolo corretto.

$$\delta_4(p,\sigma) = k_1 \qquad \text{se } p = 4 \textbf{ or } P_4[p+1] \neq \sigma$$
$$\delta_5(p,\sigma) = k_2 \qquad \text{se } p = 5 \textbf{ or } P_5[p+1] \neq \sigma$$

Qui p=4 e p=5 sono sempre false perché siamo in 0 ≤ p < 4.

$$\delta_4(p,\sigma) = k_1 \qquad \text{se } P_4[p+1] \neq \sigma$$
$$\delta_5(p,\sigma) = k_2 \qquad \text{se } P_5[p+1] \neq \sigma$$

Qui però non si può fare lo stesso ragionamento di prima perché sono diversi da sigma.

Dobbiamo calcolare k1 e k2. Sono la lunghezza del bordo di P4 e P5 da 1 a p concatenato sigma.

$$\delta_4(p,\sigma) = |B(P_4[1,p]\sigma)|$$
$$\delta_5(p,\sigma) = |B(P_5[1,p]\sigma)|$$

Se p è strettamente minore di 4, allora il prefisso lungo p di P4 coincide con il prefisso lungo p di P5. Di conseguenza k1 e k2 saranno uguali.

Abbiamo dimostrato che:

$$\Rightarrow \delta_5(p,\sigma) = \delta_4(p,\sigma) \quad \text{se } p < 4 \textbf{ and } P_5[p+1] \neq \sigma$$

Ora dove notiamo un'altra differenza?

P_4 = abccbab

δ_4	a	b	c	d
0	1	0	0	0
1	1	2	0	0
2	1	0	3	0
3	1	0	4	0
4	1	0	0	0

P_5 = abccbab

δ_5	a	b	c	d
0	1	0	0	0
1	1	2	0	0
2	1	0	3	0
3	1	0	4	0
4	1	5	0	0
5	1	0	0	0

$\delta_4(p, \sigma) = \delta_5(p, \sigma)$ per $p = 4$ e $\sigma \neq P[5]$

$\delta_4(4,\sigma) = k_1$

$\delta_5(4,\sigma) = k_2$

Tutti i valori si conservano tranne il simbolo b, che è il simbolo che P5 aggiunge rispetto a P4.

Qui facciamo un discorso analogo a prima: calcoliamo k1 e k2, applichiamo la definizione del bordo.

$$\delta_4(p, \sigma) = \delta_5(p, \sigma) \text{ per } p = 4 \text{ e } \sigma \neq P[5]$$

$$\delta_4(4,\sigma) = |B(P_4[1,p]\sigma)|$$

$$\delta_5(4,\sigma) = |B(P_5[1,p]\sigma)|$$

Anche qui P4[1,p] e P5[1,p] sono uguali quando sigma diverso da P[5]. Quindi le celle colorate conservano il valore.

Vediamo ora la cella che cambia. Questa cambia perché a sinistra P4 non ha un simbolo in posizione 5, quindi non può transitare verso lo stato successivo, ma P5 ce l'ha. Infatti:

$$\delta_4(4,P[5]) = |B(P_4[1,4]P[5]| = |B(abccb)| = |\varepsilon| = 0$$

$$\delta_5(4,P[5]) = 4 + 1 = 5$$

Ci manca vedere come si aggiunge la riga in più. Non vediamo la dimostrazione di com'è fatta questa riga. Diciamo che è banalmente la copiatura di una riga precedente, ovvero la riga 0. La riga da copiare è indicata da quel valore (0) che è cambiato. Questo 0 ci da il puntatore alla riga 0, ovvero la riga che va copiata nell'ultima.

P_4 = abccbab

δ_4	a	b	c	d
0	1	0	0	0
1	1	2	0	0
2	1	0	3	0
3	1	0	4	0
4	1	0	0	0

P_5 = abccbab

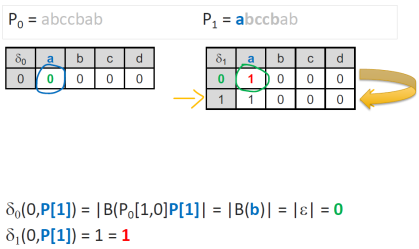
δ_5	a	b	c	d
0	1	0	0	0
1	1	2	0	0
2	1	0	3	0
3	1	0	4	0
4	1	5	0	0
5	1	0	0	0

$\delta_5(5, \sigma) = \delta_5(0, \sigma)$ per $\forall \sigma \in \Sigma$

Trasformazione $\delta_4 -> \delta_5$

1. Prendo il valore **0** contenuto nella cella della riga 4 di δ_4 in corrispondenza del simbolo P[5]
2. Sostituisco il valore **0** con il valore 5 (stato successivo a 4)
3. Aggiungo una nuova riga (corrispondente allo stato 5)
4. Copio la riga che corrisponde allo stato **0** nella riga che corrisponde allo stato 5
5. Rinomino δ_4 in δ_5

Il punto 2 è intercambiabile con i punti 3 e 4, cioè potrei fare prima i punti 3 e 4 prima del 2. In questo caso funziona, ma non sempre, per esempio:



mi son perso

Trasformazione $\delta_{j-1} -> \delta_j$

1. Prendo il valore **k** contenuto nella cella della riga j-1 di δ_{j-1} in corrispondenza del simbolo P[j]
2. Sostituisco il valore **k** con il valore j (stato successivo a j-1)
3. Aggiungo una nuova riga (corrispondente allo stato j)
4. Copio la riga che corrisponde allo stato **k** nella riga che corrisponde allo stato j
5. Rinomino δ_{j-1} in δ_j

Algoritmo

```
Procedure Build-transition-function(P)
begin
    m ← |P|
    δ ← empty table (m+1)x|Σ|

    Calcola δ₀
    for j ← 1 to m do
      Calcola δⱼ da δⱼ₋₁
    return δ
end
```

Recuperiamo la lunghezza del pattern. Inizializziamo una tabella completa, vuota (non aggiungiamo righe man mano, abbiamo già la tabella completa).

Bisogna vedere cosa mettere al posto delle righe blu.

δ	σ_1	σ_2	σ_3	…	…	…	$\sigma_{\|\Sigma\|}$
0	0	0	0	0	0	0	0
1							
2							
…							
…							
…							
j-1							
j							
…							
m							

```
foreach σ in Σ do
  δ(0,σ) ← 0
```

Calcolo δ_0

Questo foreach completa la prima riga.

δ	σ_1	σ_2	σ_3	…	P[j]	…	$\sigma_{\|\Sigma\|}$
0	0	0	0	0	0	0	0
1							
2							
…							
k							
…							
j-1					j		
j							
…							
m							

```
k ← δ(j-1,P[j])
δ(j-1,P[j]) ← j
foreach σ in Σ do
  δ(j,σ) ← δ(k,σ)
```

Calcolo δ_j da δ_{j-1}

```
Procedure Build-transition-function(P)
begin
    m ← |P|
    δ ← empty table (m+1)x|Σ|

    foreach σ in Σ do
      δ(0,σ) ← 0

    for j ← 1 to m do
      k ← δ(j-1,P[j])
      δ(j-1,P[j]) ← j
      foreach σ in Σ do
        δ(j,σ) ← δ(k,σ)

    return δ
end
```

Tempo → θ(m|Σ|)

Dal precedente m^3 siamo passati ora a:

Esercizio

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

m = 7 $\Sigma = \{a,b,c\}$

δ	a	b	c
0			
1			
2			
3			
4			
5			
6			
7			

Esercizio che può essere in esame.

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

m = 7 $\Sigma = \{a,b,c\}$

δ	a	b	c
0	1	0	0
1	1	0	0
2			
3			
4			
5			
6			
7			

k ← 0
Sostituisco 0 con 1
Copio la riga 0 nella riga 1

Salviamo k e poi lo cambiamo in 1 perché sto aggiungendo il primo simbolo del pattern, e quindi prefisso lungo 1.
Terzo passaggio, copio la riga 0 nella riga 1.

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

m = 7 $\Sigma = \{a,b,c\}$

δ	a	b	c
0	1	0	0
1	1	0	2
2	1	0	0
3			
4			
5			
6			
7			

k ← 0
Sostituisco 0 con 2
Copio la riga 0 nella riga 2

Ora di nuovo aggiungo il simbolo del pattern, quindi prendo l'ultima riga che ho riempito e leggo la cella in corrispondenza del simbolo che sto aggiungendo. Salvo quel valore in k e poi lo sostituisco con 2 (secondo simbolo). Infine faccio di nuovo la copiatura della riga 0 nella riga 2.

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

m = 7 $\Sigma = \{a,b,c\}$

δ	a	b	c
0	1	0	0
1	1	0	2
2	3	0	0
3	1	0	2
4			
5			
6			
7			

k ← 1
Sostituisco 1 con 3
Copio la riga 1 nella riga 3

Qui copieremo la riga 1 perché in k ho salvato 1.

P

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

m = 7 Σ = {a,b,c}

δ	a	b	**c**
0	1	0	0
1	1	0	2
2	3	0	0
3	1	0	**4**
4	3	0	0
5			
6			
7			

k ← **2**
Sostituisco **2** con **4**
Copio la riga **2** nella riga 4

P

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

m = 7 Σ = {a,b,c}

δ	a	**b**	c
0	1	0	0
1	1	0	2
2	3	0	0
3	1	0	4
4	3	**5**	0
5	1	0	0
6			
7			

k ← **0**
Sostituisco **0** con **5**
Copio la riga **0** nella riga 5

P

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

m = 7 Σ = {a,b,c}

δ	**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	2
7			

k ← **1**
Sostituisco **1** con **6**
Copio la riga **1** nella riga 6

P

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

m = 7 Σ = {a,b,c}

δ	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

k ← **2**
Sostituisco **2** con **7**
Copio la riga **2** nella riga 7

Altri esercizi

P

1	2	3	4	5	6

m = 6 Σ = {a,b,c,d}

δ(j,σ)	a	b	c	d
0	1			
1			2	
2		3		
3	4			
4	5			
5				6
6				

j → j+1 tramite simbolo **P[j+1]**
(0 ≤ j < m)

P

1	2	3	4	5	6
a					

m = 6 Σ = {a,b,c,d}

δ(j,σ)	**a**	b	c	d
0	1			
1			2	
2		3		
3	4			
4	5			
5				6
6				

0 → 1 tramite simbolo **a**
⇒ **P[1] = a**

P

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

m = 6 Σ = {a,b,c,d}

δ(j,σ)	a	b	c	**d**
0	1			
1			2	
2		3		
3	4			
4	5			
5				6
6				

5 → 6 tramite simbolo **d**
⇒ **P[6] = d**

P

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

m = 6 Σ = {a,b,c,d}

δ(j,σ)	a	b	c	d
0	1			
1			2	
2		3		
3	4			
4	5			
5				6
6				

Completare la tabella

P

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

m = 6   Σ = {a,b,c,d}

δ(j,σ)	a	b	c	d
0	1			
1			2	
2		3		
3	4			
4	5			
5				6
6				

j → **k**
per 0 ≤ j < m e ∀ σ ≠ P[j+1]

P

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

m = 6   Σ = {a,b,c,d}

δ(j,σ)	a	b	c	d
0	1			
1			2	
2		3		
3	4			
4	5			
5				6
6				

j → **k**
per j = m e ∀ σ

P

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

δ(j,σ)	a	b	c	d
0	1	0	0	0
1	1	0	2	0
2	1	3	0	0
3	4	0	0	0
4	5	0	2	0
5	1	0	2	6
6	1	0	0	0

P

1	2	3	4	5	6
?	?	?	?	?	?

m = 6   Σ = {a,b,c}

δ(j,σ)	a	b	c
0			1
1	2		
2			
3			
4			
5			
6			3

Quali simboli di P possono essere specificati?

P

1	2	3	4	5	6
c	?	?	?	?	?

m = 6   Σ = {a,b,c}

δ(j,σ)	a	b	**c**
0			**1**
1	2		
2			
3			
4			
5			
6			3

0 → 1 tramite simbolo **c**
⇒ **P[1] = c**

P

1	2	3	4	5	6
c	a	?	?	?	?

m = 6   Σ = {a,b,c}

δ(j,σ)	**a**	b	c
0			1
1	**2**		
2			
3			
4			
5			
6			3

1 → 2 tramite simbolo **a**
⇒ **P[2] = a**

P

1	2	3	4	5	6	
c	a	?	?	?	?	c

m = 6   Σ = {a,b,c}

δ(j,σ)	a	b	c
0			1
1	2		
2			
3			
4			
5			
6			**3**

6 → 3 tramite simbolo **c**
⇒ **|B(P[1,6]c)| = 3**

	1	2	3	4	5	6	
P	c	a	c	?	c	a	c

m = 6 Σ = {a,b,c}

δ(j,σ)	a	b	c
0			1
1	2		
2			
3			
4			
5			
6			**3**

6 → 3 tramite simbolo **c**
⇒ **|B(Pc)| = 3**

	1	2	3	4	5	6	
P	c	a	c	?	c	a	c

m = 6 Σ = {a,b,c}

δ(j,σ)	a	b	c
0			1
1	2		
2			
3		4	
4	3		
5			
6			3

I due valori in rosso sono compatibili con P?

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

m = 6 Σ = {a,b,c}

δ(j,σ)	a	b	c
0			1
1	2		
2			
3		4	
4	3		
5			
6			3

3 → **4** tramite simbolo **b**
⇒ **P[4] = b**

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

m = 6 Σ = {a,b,c}

δ(j,σ)	a	b	c
0			1
1	2		
2			
3		~~4~~	
4	~~3~~		
5			
6			3

4 → **3** tramite simbolo **a**
⇒ ~~**|B(cacba)|**~~ **= 3**

δ(4,a) = **3** è impossibile per qualsiasi simbolo P[4]