

# Linguaggi di Programmazione

## Corso C

### Parte n.6

## Linguaggi Regolari ed Espressioni Regolari

Nicola Fanizzi (*fanizzi@di.uniba.it*)

**Dipartimento di Informatica**  
**Università degli Studi di Bari**

# Linguaggi Regolari

Dato un alfabeto finito  $X$

un linguaggio  $L$  su  $X$  è un **linguaggio regolare** sse:

- $L$  è finito  
*oppure*
- $L$  può essere ottenuto induttivamente mediante le operazioni:
  1.  $L = L_1 \cup L_2$  con  $L_1, L_2$  regolari
  2.  $L = L_1 \cdot L_2$  con  $L_1, L_2$  regolari
  3.  $L = L_1^*$  con  $L_1$  regolare

Osservazioni:

- $\emptyset$  e  $\{\lambda\}$  sono linguaggi regolari

Definiamo la **classe dei linguaggi regolari**

denotata con

$$\mathcal{L}_{REG}$$

l'insieme di tali linguaggi

# Espressioni Regolari

Dato un alfabeto finito  $X$  una stringa  $R$  su  $X \cup \{\lambda, +, *, \cdot, \emptyset, (, )\}$  è una **espressione regolare** di alfabeto  $X$  sse vale una delle seguenti condizioni:

- $R = \emptyset$
- $R = \lambda$
- $R = a$   
con  $a \in X$
- $R = (R_1 + R_2)$   
con  $R_1, R_2$  espressioni regolari di alfabeto  $X$
- $R = (R_1 \cdot R_2)$   
con  $R_1, R_2$  espressioni regolari di alfabeto  $X$
- $R = (R_1)^*$   
con  $R_1$  espressione regolare di alfabeto  $X$

Denotiamo con  $\mathcal{R}$

l'insieme delle espressioni regolari di alfabeto  $X$

## Corrispondenza tra Linguaggi ed Espressioni Regolari

Ad ogni espressione regolare  $R$   
corrisponde un linguaggio regolare  $S(R)$   
nel modo seguente:

espressione regolare	linguaggio regolare
$\emptyset$	$\emptyset$
$\lambda$	$\{\lambda\}$
$a$	$\{a\}$
$(R_1 + R_2)$	$S(R_1) \cup S(R_2)$
$(R_1 \cdot R_2)$	$S(R_1) \cdot S(R_2)$
$(R)^*$	$(S(R))^*$

**Proposizione** Un linguaggio su  $X$  è regolare sse esso corrisponde ad una espressione regolare di alfabeto  $X$  ossia definita la funzione  $S : \mathcal{R} \longrightarrow \wp(X^*)$ , si ha che:

$$\mathcal{L}_{REG} = \{L \in \wp(X^*) \mid \exists R \in \mathcal{R}: L = S(R)\}$$

### osservazioni.

- un linguaggio regolare può essere descritto da più di una espressione regolare, cioè
- la funzione  $S : \mathcal{R} \longrightarrow \wp(X^*)$  non è inettiva

Due **espressioni regolari**  $R_1$  e  $R_2$  si dicono **equivalenti**, e si scrive  $R_1 = R_2$ , sse

$$S(R_1) = S(R_2)$$

### Esempio.

$L$  linguaggio delle parole su  $X = \{a, b\}$  con  $a$  e  $b$  alternate, inizianti e terminanti con  $b$

descrivibile con l'espressione

$$R_1 = b(ab)^*$$

ovvero con

$$R_2 = (ba)^*b$$

pertanto  $R_1 = R_2$

# Proprietà dei Linguaggi Regolari

- |    |   |                                |
|----|---|--------------------------------|
| 1. | $(R_1 + R_2) + R_3 = R_1 + (R_2 + R_3)$ | prop. associativa di +         |
| 2. | $R_1 + R_2 = R_2 + R_1$                 | prop. commutativa di +         |
| 3. | $R + \emptyset = \emptyset + R = R$     | $\emptyset$ elem. neutro per + |
| 4. | $R + R = R$                             | Idempotenza                    |
- 
- |    |   |                                    |
|----|---|------------------------------------|
| 5. | $(R_1 \cdot R_2) \cdot R_3 = R_1 \cdot (R_2 \cdot R_3)$ | prop. associativa di ·             |
| 6. | $R_1 \cdot R_2 \neq R_2 \cdot R_1$                      | non commutatività di ·             |
| 7. | $R \cdot \lambda = \lambda \cdot R = R$                 | $\lambda$ elem. neutro per ·       |
| 8. | $R \cdot \emptyset = \emptyset \cdot R = \emptyset$     | $\emptyset$ elem. assorbente per · |
- 
- |     |   |                                   |
|-----|---|-----------------------------------|
| 9.  | $R_1 \cdot (R_2 + R_3) = (R_1 \cdot R_2) + (R_1 \cdot R_3)$ | prop. distributiva di · risp. a + |
| 10. | $(R_1 + R_2) \cdot R_3 = (R_1 \cdot R_3) + (R_2 \cdot R_3)$ | prop. distributiva di · risp. a + |
- 
- |     |   |  |
|-----|---|--|
| 11. | $(R)^* = (R)^* \cdot (R)^* = ((R)^*)^* = (\lambda + R)^*$ |  |
| 12. | $(\emptyset)^* = (\lambda)^* = \lambda$                   |  |
| 13. | $(R)^* = \lambda + R + \dots + R^n + (R^{n+1} \cdot R^*)$ |  |
- 
- |     |  |                        |
|-----|--|------------------------|
| 14. | $(R_1 + R_2)^* = (R_1^* + R_2^*)^* = (R_1^* \cdot R_2^*)^* =$<br>$= (R_1^* \cdot R_2^*)^* \cdot R_1^* = R_1^* \cdot (R_1^* \cdot R_2^*)^*$ |                        |
| 15. | $(R_1 + R_2)^* \neq R_1^* + R_2^*$   | in generale            |
| 16. | $R \cdot R^* = R^* \cdot R$  |                        |
| 17. | $R_1 \cdot (R_2 \cdot R_1)^* = (R_1 \cdot R_2)^* \cdot R_1$  |                        |
| 18. | $(R_1^* \cdot R_2)^* = \lambda + (R_1 + R_2)^* \cdot R_2$  |                        |
| 19. | $(R_1 \cdot R_2^*)^* = \lambda + R_1 \cdot (R_1 + R_2)^*$  |                        |
| 20. | $R_1 = R_2 \cdot R_1 + R_3$ <b>sse</b> $R_1 = R_2^* \cdot R_3$<br>$R_1 = R_1 \cdot R_2 + R_3$ <b>sse</b> $R_1 = R_3 \cdot R_2^*$           | con $R_2 \neq \lambda$ |

# Teorema di Kleene

## Teorema.

Vale la seguente equivalenza:

$$\mathcal{L}_3 \equiv \mathcal{L}_{FSL} \equiv \mathcal{L}_{REG}$$

**Dimostrazione.** Lo schema di dimostrazione è il seguente:

- $\mathcal{L}_3 \subset \mathcal{L}_{FSL}$  (ed anche  $\mathcal{L}_{FSL} \subset \mathcal{L}_3$ )
- $\mathcal{L}_{FSL} \subset \mathcal{L}_{REG}$
- $\mathcal{L}_{REG} \subset \mathcal{L}_3$

# Teorema di Kleene (I)

**Tesi (I):**  $\mathcal{L}_3 \subset \mathcal{L}_{FSL}$

Sia  $L \in \mathcal{L}_3$  cioè  $\exists G = (X, V, S, P)$ ,  $G$  di tipo 3:  $L(G) = L$

Si deve costruire un FSA  $M = (Q, \delta, q_0, F)$

tale che  $T(M) = L(G)$

---

## Algoritmo.

Input:  $G = (X, V, S, P)$ ,  $G$  di tipo 3

Output:  $M = (Q, \delta, q_0, F) \in FSA$

---

1.  $X$  alfabeto di ingresso per  $M$
2.  $Q = V \cup \{q\}$ ,  $q \notin V$
3.  $q_0 = S$
4.  $F = \{q\} \cup \{B \mid B\lambda \in P\}$
5.  $\delta : Q \times X \longrightarrow \wp(X^*)$ 
  - (a)  $\forall B \longrightarrow aC \in P: C \in \delta(B, a)$
  - (b)  $\forall B \longrightarrow a \in P: q \in \delta(B, a)$

## Osservazione:

l'algoritmo può generare (passi 5.a e 5.b) automi non deterministici

- $L(G) \subseteq T(M)$ : sia  $w = x_1x_2 \cdots x_n \in L(G)$

$w$  può essere generata con una derivazione:

$$S \Longrightarrow x_1X_2 \Longrightarrow x_1x_2X_3 \Longrightarrow \cdots \Longrightarrow x_1x_2 \dots X_k \Longrightarrow x_1x_2 \dots x_k$$

Per la sua def. l'automa  $M$ , avendo in input  $w$ , compie una serie di transizioni che portano da  $S$  a  $X_1, X_2, \dots, X_k$  fino a  $q$ .

- $T(M) \subseteq L(G)$ : analogamente

Dimostriamo anche in maniera costruttiva

$$\mathcal{L}_{FSL} \subset \mathcal{L}_3$$

---

**Algoritmo.**

Input:  $M = (Q, \delta, q_0, F) \in FSA$

Output:  $G = (X, V, S, P)$  lineare

---

1.  $X$  alfabeto di ingresso per  $M$
2.  $V = Q$
3.  $S = q_0$
4.  $P = \{q \longrightarrow xq' \mid q' \in \delta(q, x)\} \cup \{q \longrightarrow x \mid \delta(q, x) \in F\}$

Per esercizio:  $L(G) = T(M)$

## Teorema di Kleene (II)

**Tesi (II):**  $\mathcal{L}_{FSL} \subset \mathcal{L}_{REG}$

Sia  $L \in \mathcal{L}_{FSL}$  cioè  $\exists M = (Q, \delta, q_0, F) \in FSA$  tale che:  $T(M) = L$

Supponiamo  $Q = \{q_0, q_1, \dots, q_n\}$

Si definisca il linguaggio:

$$R_{ij} = \{w \in X^* \mid \delta^*(q_i, w) = q_j\}$$

contenente le stringhe che fanno transitare  $M$  da  $q_i$  a  $q_j$ .

Per def. di linguaggio accettato da FSA, risulta:

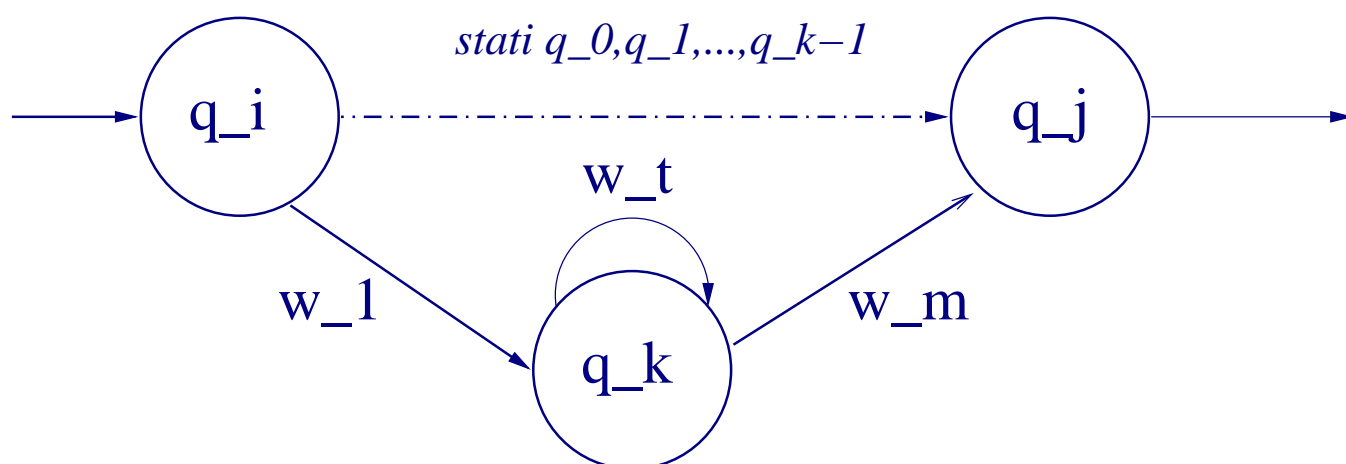
$$T(M) = \bigcup_{q_j \in F} R_{0j}$$

quindi basta dimostrare che ogni  $R_{ij}$  è un linguaggio regolare  $0 \leq i, j \leq n$

Sia  $R_{ij}^k = \{w \in X^* \mid \delta^*(q_i, w) = q_j \text{ senza transitare in } q_k, q_{k+1}, \dots, q_n\}$

Osservato che:  $R_{ij}^{n+1} = R_{ij}$  (1)

dimostriamo per induzione su  $k$  che  $R_{ij}^k \in \mathcal{L}_{REG} \quad \forall i, j : 0 \leq i, j \leq n$



( $k = 0$ )  $R_{ij}^0 = \{w \in X^* \mid \delta(q_i, w) = q_j\} \in \mathcal{L}_{REG}$  perchè linguaggio finito

( $k > 0$ ) per ipotesi:  $R_{ij}^k \in \mathcal{L}_{REG} \quad \forall i, j : 0 \leq i, j \leq n$

Dimostriamo che  $R_{ij}^{k+1} \in \mathcal{L}_{REG}$

Sia  $w \in R_{ij}^{k+1}$  per definizione la lettura di  $w$  non fa transitare  $M$  in nessuno degli stati  $q_{k+1}, q_{k+2}, \dots, q_n$ .

Si possono avere 2 casi:

1.  $w$  non fa transitare  $M$  nemmeno in  $q_k$ , quindi  $w \in R_{ij}^k$  ed  $R_{ij}^k \in \mathcal{L}_{REG}$  cioè regolare, per ipotesi.
2.  $w$  fa transitare  $M$  in  $q_k$ . In tal caso riscriviamo  $w$  come concatenazione di  $m > 1$  sottostringhe:  $w = w_1 w_2 \dots w_{m-1} w_m$  ove:

$$w_1 \in R_{ik}^k \quad w_t \in R_{kk}^k \quad \forall t: 1 < t < m \quad w_m \in R_{kj}^k$$

Data la genericità di  $w$  si può scrivere:

$$w \in R_{ik}^k \cdot (R_{kk}^k)^{m-2} \cdot R_{kj}^k \quad m > 1 \text{ per cui:}$$

$$w \in R_{ik}^k \cdot (R_{kk}^k)^* \cdot R_{kj}^k$$

Si conclude allora che:  $R_{ij}^{k+1} \subseteq R_{ij}^k \cup R_{ik}^k \cdot (R_{kk}^k)^* \cdot R_{kj}^k$

ma ovviamente:  $R_{ij}^k \cup R_{ik}^k \cdot (R_{kk}^k)^* \cdot R_{kj}^k \subseteq R_{ij}^{k+1}$

quindi  $R_{ij}^{k+1} = R_{ij}^k \cup R_{ik}^k \cdot (R_{kk}^k)^* \cdot R_{kj}^k$

sarà regolare perchè espresso come unione, concatenazione e iterazione di linguaggi regolari (per ipotesi induttiva)

Risulta dimostrato che  $\forall k \in [0, n] R_{ij}^k \in \mathcal{L}_{REG}$  perciò

$$T(M) = \bigcup_{q_j \in F} R_{0j} \stackrel{(1)}{=} \bigcup_{q_j \in F} R_{0j}^{n+1}$$

sarà regolare perchè unione di linguaggi regolari

## Teorema di Kleene (III)

**Tesi (III):**  $\mathcal{L}_{REG} \subset \mathcal{L}_3$

Sia  $L \in \mathcal{L}_{REG}$  quindi  $L$  è finito oppure:

- $L = L_1 \cup L_2$  con  $L_1, L_2$  regolari
- $L = L_1 \cdot L_2$  con  $L_1, L_2$  regolari
- $L = (L_1)^*$  con  $L_1$  regolare

Dimostriamo la tesi per induzione sulla costruzione di  $L$

**base**  $L$  finito allora  $L$  è banalmente lineare destro.

Si può scrivere come unione di linguaggi lineari che generano ognuno una stringa di  $L$  e  $\mathcal{L}_3$  è chiusa rispetto all'unione

$$L = \{w_1, w_2, \dots, w_n\} \quad n \geq 0$$

quindi

$$L = \bigcup_{i=0}^n L_i$$

Si può facilmente dimostrare per induzione che  $\forall i \in [0, n]: L_i \in \mathcal{L}_3$

**passo** In tutti i tre casi possiamo considerare i linguaggi  $L_1$  e  $L_2$  come lineari per ipotesi di induzione.

Anche la loro unione/concatenazione/iterazione costituisce un linguaggio lineare destro per i teoremi di chiusura rispetto a queste operazioni

Quindi  $L \in \mathcal{L}_3$

# Pumping Lemma per Linguaggi Regolari

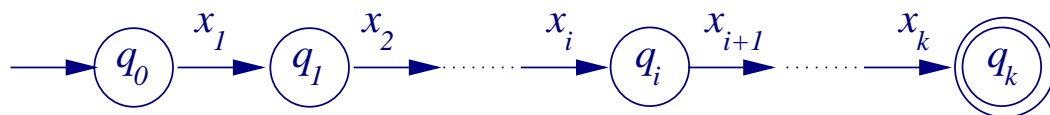
## Teorema.

Sia  $M = (Q, \delta, q_0, F) \in \text{FSA}$  con  $n = |Q|$  e sia  $z \in T(M)$ ,  $|z| \geq n$ . Allora  $z = uvw$  e  $uv^t w \in T(M)$  (cioè  $\forall t \geq 0: uv^t w \in T(M)$ )

## Dimostrazione.

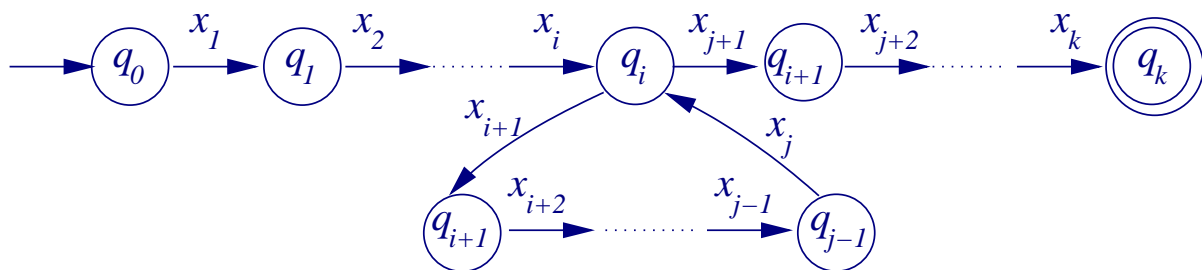
Sia  $z = x_1 x_2 \cdots x_k \in T(M)$  con  $k \geq n$

Si rappresenta il riconoscimento di  $z$  con:



Se si ha  $|z| \geq n$  si deve passare per almeno  $n + 1$  stati

ma  $n = |Q|$  quindi c'è uno stato ripetuto:  $\exists i, j, 0 \leq i < j \leq n: q_i = q_j$



Si osservi che  $z$  si può scrivere come  $uvw$  ove:

- $u = x_1 x_2 \cdots x_i$
- $v = x_{i+1} x_{i+2} \cdots x_j$
- $w = x_{j+1} x_{j+2} \cdots x_k$

Avendo in ingresso  $z \in T(M)$ ,  $M$  si porta in uno stato finale:  $\delta^*(q_0, z) \in F$  ma questo è lo stesso stato in cui si giunge tramite le stringhe:  $uvvw$ ,  $uvvww$  etc. ... (ciclando  $i$  volte tra  $q_i$  e  $q_j$ )

Dunque:  $\forall t \geq 0: uv^t w \in T(M)$

# Esercizi

- Determinare la grammatica di tipo 3 che genera il linguaggio descritto da  $b^* + (ab)^*$
- Data la grammatica lineare  $G = (X, V, S, P)$  con  
 $X = \{a, b, c\}$ ,  $V = \{S, A, B\}$  e  
 $P = \{S \rightarrow bA|aS|b, \quad A \rightarrow aB|cS|a, \quad B \rightarrow bA|cB|c\}$   
determinare un'espressione regolare per  $L(G)$

- Sia  $L = S(R)$  ove  $R = (aa + aaa)^*$

1. costruire un automa che riconosce  $L$
2. trasformare l'NDA del punto 1. in FSA

- determinare una grammatica  $G$  di tipo 3 tale che:

$$L(G) = \{w \in \{a, b\}^* \mid w \text{ ha un numero pari di } a \text{ e dispari di } b\}$$

- $L = S(R)$  ove  $R = ab(bb)^*c$

1. trovare un automa (NDA) per riconoscere  $L$
2. trasformare l'automa NDA nell'FSA equivalente

- Data la grammatica lineare  $G = (X, V, S, P)$  con

$$X = \{a, b\}, \quad V = \{S, B\} \text{ e}$$

$$P = \{S \rightarrow aB \quad B \rightarrow aB|bS|a\}$$

determinare un automa FSA  $M$  tale che:  $T(M) = L(G)$

- Determinare una grammatica lineare  $G$  tale che:

$$L(G) = \{w \in \{a, b\}^* \mid w \neq \alpha a a \beta, \alpha, \beta \in \{a, b\}^*\}$$

- Dimostrare che non sono regolari i linguaggi:

$$- L_1 = \{a^k b^k \mid k > 0\}$$

$$- L_2 = \{a^n b^m c^k \mid n > k, n, m, k > 0\}$$

- Dimostrare che il linguaggio non è regolare:

$$L = \{a^n b^m c^k \mid m > k, n, m, k > 0\}$$

# Esercizio 1

Determinare la grammatica di tipo 3 che genera il linguaggio descritto da

$$b^* + (ab)^*$$

vediamo qual é il linguaggio corrispondente:

$$S(b^* + (ab)^*) = S(b^*) \cup S((ab)^*) = (S(b))^* \cup (S(ab))^* = \{b\}^* \cup (S(a) \cdot S(b))^* = \{b\}^* \cup (\{a\} \cdot \{b\})^* = \{b\}^* \cup \{ab\}^*$$

$$G_1 = (X_1, V_1, S_1, P_1) \text{ con } X_1 = \{b\} \quad V_1 = \{S_1\}$$

$$P_1 = \{S_1 \longrightarrow bS_1 \mid \lambda\} \text{ e}$$

$$G_2 = (X_2, V_2, S_2, P_2) \text{ con } X_2 = \{a, b\} \quad V_2 = \{S_2, B_2\}$$

$$P_2 = \{S_2 \longrightarrow aB_2, B_2 \longrightarrow b\}$$

$$G_3 = (X_3, V_3, S_3, P_3) \text{ con } X_3 = X_2 = \{a, b\} \quad V_3 = V_2 \cup \{S_3\}$$

$$P = \{S_3 \longrightarrow \lambda\} \cup (P_2 \setminus \{S_2 \longrightarrow \lambda\}) \cup$$

$$\cup \{S_2 \longrightarrow w \in P_2\} \cup \{A \longrightarrow bS_3 \mid A \longrightarrow b \in P_2\} =$$

$$= \{S_3 \longrightarrow \lambda\} \cup P_2 \cup \{S_3 \longrightarrow aB_2\} \cup \{B_2 \longrightarrow bS_3\} =$$

$$= \{S_3 \longrightarrow \lambda, S_2 \longrightarrow aB_2, B_2 \longrightarrow b, S_3 \longrightarrow aB_2, B_2 \longrightarrow bS_3\} =$$

$$= \{S_3 \longrightarrow \lambda \mid aB_2, S_2 \longrightarrow aB_2, B_2 \longrightarrow b \mid bS_3\}$$

$$\text{Ma } S_2 \text{ è inutile: } P_3 = \{S_3 \longrightarrow \lambda \mid aB_2, B_2 \longrightarrow b \mid bS_3\}$$

$$\text{Quindi } L(G_3) = \{ab\}^*$$

Per chiudere:  $G = (X, V, S, P)$  ove:

$$X = X_1 \cup X_3 \quad V = V_1 \cup V_3 \cup \{S\} = \{S, S_1, B_2, S_3\}$$

$$P = \{S \longrightarrow w \mid S_1 \longrightarrow w \in P_1\} \cup \{S \longrightarrow w \mid S_3 \longrightarrow w \in P_3\} \cup P_1 \cup P_3 =$$

$$= \{S \longrightarrow bS_3 \mid aB_2 \mid \lambda, S_1 \longrightarrow bS_1 \mid \lambda, S_3 \longrightarrow aB_2 \mid \lambda, B_2 \longrightarrow bS_3 \mid b\}$$

## Esercizio 3

Sia  $L = S(R)$  ove  $R = (aa + aaa)^*$

1. costruire un automa che riconosce  $L$
2. trasformare l'NDA del punto 1. in FSA

Svolgimento:

1. Dato  $X = \{a\}$ , determiniamo le grammatiche  $G_1$  per  $L_1 = \{aa\}$  e  $G_2$  per  $L_2 = \{aaa\}$ :

$G_1 = (X, V_1, S_1, P_1)$  con  $V_1 = \{S_1, A\}$  e  $P_1 = \{S_1 \rightarrow aA, A \rightarrow a\}$

$G_2 = (X, V_2, S_2, P_2)$  con  $V_2 = \{S_2, B, C\}$  e

$P_2 = \{S_2 \rightarrow aB, B \rightarrow aC, C \rightarrow a\}$

Sia  $G_3 = (X, V_3, S_3, P_3)$  la grammatica per  $L_3 = L_1 \cup L_2$ :

con  $V_3 = V_1 \cup V_2 \cup \{S_3\} = \{S_3, S_1, S_2, A, B, C\}$  e

$P_3 = \{S_3 \rightarrow w \mid S_1 \rightarrow w \in P_1\} \cup \{S_3 \rightarrow w \mid S_2 \rightarrow w \in P_2\} \cup P_1 \cup P_2 =$   
 $= \{S_3 \rightarrow aA \mid aB\} \cup P_1 \cup P_2$

che contiene produzioni eliminabili e quindi NT superflui ( $S_1, S_2$ )

Per l'iterazione di  $L_3$  costruisco:

$G = (X, V, S, P)$  con

$V = V_3 \cup \{S\} = \{S, S_3, A, B, C\}$

$P = \{S \rightarrow \lambda\} \cup (P_3 \setminus \{S_3 \rightarrow \lambda\}) \cup$

$\cup \{S \rightarrow w \mid S_3 \rightarrow w \in P_3\} \cup$

$\cup \{N \rightarrow bS \mid N \rightarrow b \in P_3\}$

$\cup \{N \rightarrow bS \mid N \rightarrow bM, b \neq \lambda, M \rightarrow \lambda \in P_3\} =$

$= \{S \rightarrow \lambda\} \cup \{S_3 \rightarrow aA \mid aB, A \rightarrow a, B \rightarrow aC, C \rightarrow a\} \cup$

$\cup \{S \rightarrow aA \mid aB\} \cup$

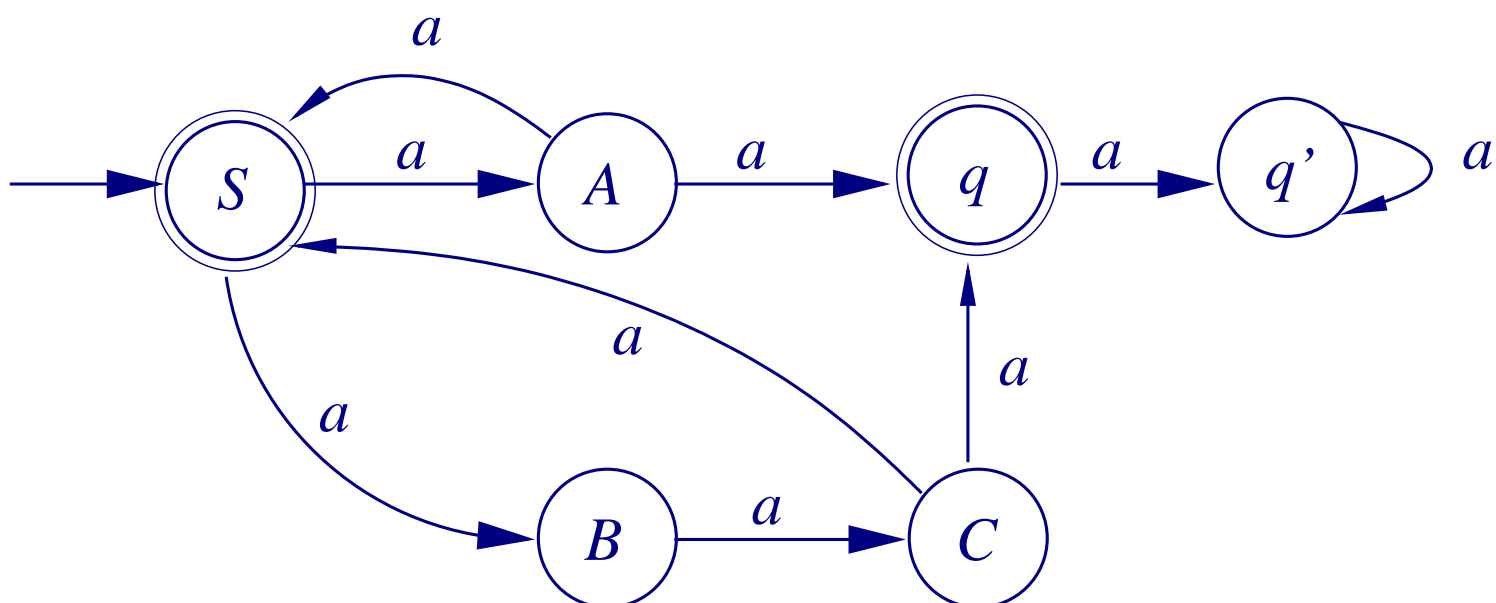
$\cup \{A \rightarrow aS, C \rightarrow aS\}$

$= \{S \rightarrow aA \mid aB \mid \lambda, A \rightarrow aS \mid a, B \rightarrow aC, C \rightarrow aS \mid a\}$

2. L'automa a stati finiti si ottiene mediante l'algoritmo dato nella dimostrazione del teorema di Kleene:

- $Q = V \cup \{q\} = \{S, A, B, C, q\}$
- $q_0 = S$
- $F = \{q, S\}$
- $\delta$  definita dalla matrice di transizione:

$\delta$	$S$	$A$	$B$	$C$	$q$
$a$	$\{A, B\}$	$\{S, q\}$	$\{C\}$	$\{S, q\}$	$\{q\}$



## Esercizio 8

Dimostrare che non è regolare il linguaggio:  $L_1 = \{a^k b^k \mid k > 0\}$

Supponiamo che  $L$  sia regolare. Per il teorema di Kleene sarà anche in  $\mathcal{L}_{FSL}$  quindi  $\exists M = (Q, \delta, q_0, F)$  tale che  $T(M) = L$

Supponiamo anche che  $|Q| = n$ . Considero  $w = a^n b^n \in L$

L'automa  $M$  leggendo le  $a$  di  $w$  deve passare per almeno  $n + 1$  stati, quindi ci devono essere almeno due stati che coincidono.

Chiamiamoli  $q_i$  e  $q_j$  con  $i < j$ .

Ciò significa che nel cammino da  $q_0$  ad uno stato finale  $q_k$  c'è un ciclo di lunghezza  $j - i$ .

Poichè si può restare a ciclare un numero arbitrario di volte, ad ogni ciclo aggiungiamo una sottostringa  $a^{j-i}$

Per il Pumping lemma dovrebbe essere  $a^{n+(j-i)} b^n \in T(M)$  ed anche  $a^{n+k(j-i)} b^n \in T(M), \forall k \geq 0$

Ma queste non sono stringhe di  $L$ .

Supposto  $L$  regolare abbiamo un assurdo, pertanto  $L$  non è regolare