Esempio

Supponiamo che $L_{pr} = \{1^p : p \text{ è primo}\}$ sia regolare.
Sia n dato dal pumping lemma.
Scegliamo un numero primo $p \geq n + 2$ e $w = 1^p$.

$$w = \overbrace{111 \cdots\cdots 11111 \cdots 11}^{p}{}_{z}$$

Esempio

Supponiamo che $L_{pr} = \{1^p : p \text{ è primo}\}$ sia regolare.
Sia n dato dal pumping lemma.
Scegliamo un numero primo $p \geq n+2$ e $w = 1^p$.

$$w = \overbrace{\underbrace{111\cdots\cdots}_{x}\underbrace{\cdots1}_{y}\underbrace{1111\cdots11}_{z}}^{p}$$

$$|y|{=}m \;\wedge\; |xz|{=}p{-}m$$

Ora $xy^{p-m}z$ dovrebbe appartenere a L_{pr}

Esempio

Supponiamo che $L_{pr} = \{1^p : p \text{ è primo}\}$ sia regolare.
Sia n dato dal pumping lemma.
Scegliamo un numero primo $p \geq n + 2$ e $w = 1^p$.

$$w = \overbrace{\underbrace{111 \cdots\cdots}_{x}\underbrace{\cdots 1}_{y}\underbrace{1111 \cdots 11}_{z}}^{p}$$

$$|y| = m \wedge |xz| = p - m$$

Ora $xy^{p-m}z$ dovrebbe appartenere a L_{pr}
$|xy^{p-m}z| = |xz| + (p-m)|y| = p - m + (p-m)m = (1+m)(p-m)$
che non è primo a meno che uno dei fattori non sia 1.

- $y \neq \epsilon \Rightarrow 1 + m > 1$
- $m = |y| \leq |xy| \leq n, \quad p \geq n + 2$

$\Rightarrow p - m \geq n + 2 - n = 2.$

Dimostrare che un linguaggio non è regolare come gioco a due

Nel pumping lemma ci sono quattro quantificatori distinti.

- Il giocatore 1 sceglie il linguaggio L
- Il giocatore 2 (l'avversario) sceglie n senza dirlo a 1 (la strategia di 1 deve valere per qualsiasi n [$\forall n$])
- Il giocatore 1 sceglie w tale che $|w| \geq n$ [$\exists w$]
- Il giocatore 2 sceglie come scomporre w rispettando i vincoli (1) e (2) [$\forall x, y, z$]
- Il giocatore 1 "vince" scegliendo k tale che $xy^kz \notin L$ [$\exists k$]

Se il linguaggio è regolare "vince" l'avversario

- $L = \emptyset$: il giocatore non può scegliere w dall'insieme vuoto
- $L = \{00, 11\}$: se l'avversario sceglie $n > 2$, il giocatore non può scegliere w. Analogo ragionamento vale per tutti gli insiemi finiti.
- $L = (\mathbf{00} + \mathbf{11})^*$: scelto n, qualsiasi w scelto dal giocatore è composto da coppie 00 o 11. L'avversario può scegliere una qualsiasi di queste coppie per y. Ma allora per qualsiasi i, xy^iz continua a rimanere dentro L.
- $L = \mathbf{10}^*\mathbf{1}^*\mathbf{0}$ scelto $n > 2$, qualsiasi w scelto dal giocatore è del tipo 10^i1^j0, con $|w| \geq 1$. Ognuna di queste stringhe w può essere pompata, prendendo $x = 1$, y come secondo simbolo della stringa e z come quel che rimane ($|xy| \leq n$ e $|y| \neq 0$), e rimanere dentro L.
  Per ogni stringa, l'avversario trova come decomporre per "vincere".

PL: non è una condizione sufficiente per la regolarità

Il pumping lemma fornisce **soltanto** una condizione necessaria affinché un linguaggio sia regolare. Ovvero:

- L è regolare $\Rightarrow$ L soddisfa il Pumping Lemma
- L **non** soddisfa il Pumping Lemma $\Rightarrow$ L **non** è regolare
- L soddisfa il Pumping Lemma $\not\Rightarrow$ L è regolare

Esistono linguaggi **non** regolari che soddisfano il Pumping Lemma:

$$\{ww^R v | w, v \in \{0,1\}^+\}$$

Per dimostrare la non regolarità dobbiamo usare altri sistemi.

Proprietà di chiusura dei linguaggi regolari

Siano L e M due linguaggi regolari. Allora i seguenti linguaggi sono regolari:

- *Unione:* $L \cup M$
- *Intersezione:* $L \cap M$
- *Complemento:* $\overline{N}$
- *Differenza:* $L \setminus M$
- *Inversione:* $L^R = \{w^R : w \in L\}$
- *Chiusura:* L^*.
- *Concatenazione:* $L.M$

Chiusura rispetto a unione e complemento

Teorema 4.4. Per ogni coppia di linguaggi regolari L e M, $L \cup M$ è regolare.

Chiusura rispetto a unione e complemento

Teorema 4.4. Per ogni coppia di linguaggi regolari L e M, $L \cup M$ è regolare.

Dimostrazione. Sia $L = L(E)$ e $M = L(F)$. Allora $L(E + F) = L \cup M$ per definizione.

Chiusura rispetto a unione e complemento

Teorema 4.4. Per ogni coppia di linguaggi regolari L e M, $L \cup M$ è regolare.
Dimostrazione. Sia $L = L(E)$ e $M = L(F)$. Allora $L(E + F) = L \cup M$ per definizione.

Teorema 4.5. Se L è un linguaggio regolare su Σ, allora che $\overline{L} = \Sigma^* \setminus L$ è regolare.

Chiusura rispetto a unione e complemento

Teorema 4.4. Per ogni coppia di linguaggi regolari L e M, $L \cup M$ è regolare.
Dimostrazione. Sia $L = L(E)$ e $M = L(F)$. Allora $L(E + F) = L \cup M$ per definizione.

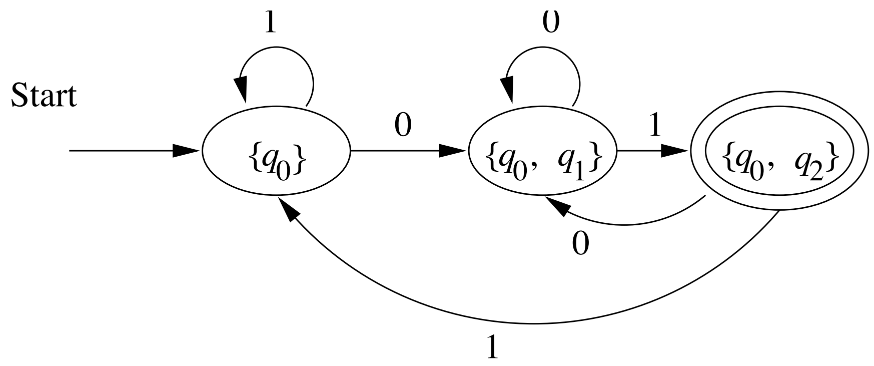
Teorema 4.5. Se L è un linguaggio regolare su Σ, allora che $\overline{L} = \Sigma^* \setminus L$ è regolare.
Dimostrazione. Sia L riconosciuto da un DFA

$$A = (Q, \Sigma, \delta, q_0, F).$$

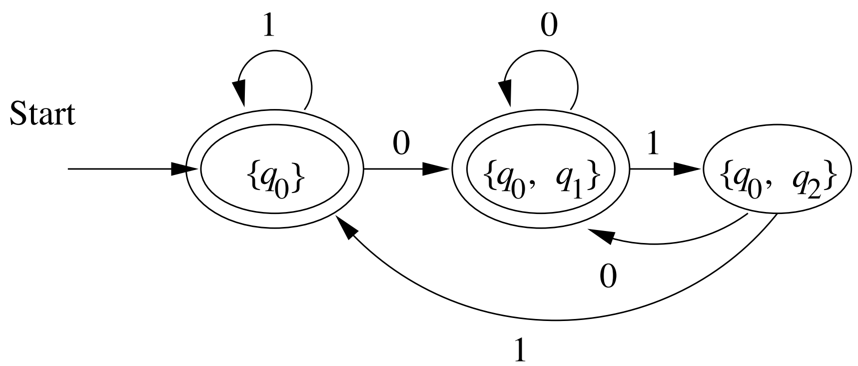
Sia $B = (Q, \Sigma, \delta, q_0, Q \setminus F)$. Allora $L(B) = \overline{L}$.

Esempio

Sia L riconosciuto dal DFA qui sotto:



Allora $\overline{L}$ è riconosciuto da:



Domanda: Quali sono le espressioni regolari per L e $\overline{L}$?

Chiusura rispetto all'intersezione

Teorema 4.8. Se L e M sono regolari, allora anche $L \cap M$ è regolare.

Chiusura rispetto all'intersezione

Teorema 4.8. Se L e M sono regolari, allora anche $L \cap M$ è regolare.

Dimostrazione 1. Per la legge di De Morgan, $L \cap M = \overline{\overline{L} \cup \overline{M}}$. Sappiamo già che i linguaggi regolari sono chiusi sotto il complemento e l'unione.

Chiusura rispetto all'intersezione: un'altra dimostrazione

Se L e M sono regolari, allora anche $L \cap M$ è regolare.
Dimostrazione 2. Sia L il linguaggio di

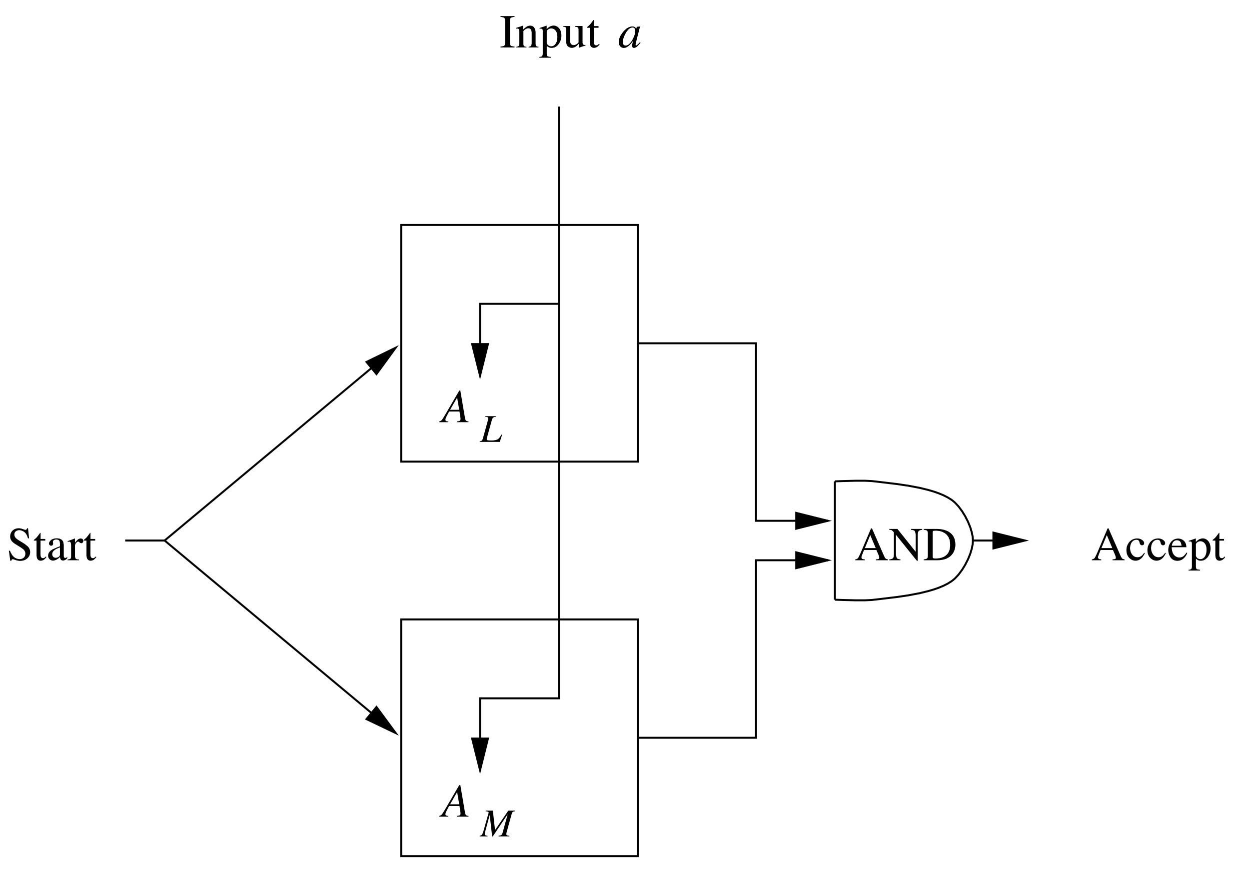
$$A_L = (Q_L, \Sigma, \delta_L, q_L, F_L)$$

e M il linguaggio di

$$A_M = (Q_M, \Sigma, \delta_M, q_M, F_M)$$

Supponiamo senza perdita di generalità che entrambi gli automi siano deterministici.
Costruiremo un automa che simula A_L e A_M in parallelo, e accetta se e solo se sia A_L che A_M accettano.

Se A_L va dallo stato p allo stato s leggendo a, e A_M va dallo stato q allo stato t leggendo a, allora $A_{L \cap M}$ andrà dallo stato (p, q) allo stato (s, t) leggendo a.



Formalmente

$$A_{L\cap M} = (Q_L \times Q_M, \Sigma, \delta_{L\cap M}, (q_L, q_M), F_L \times F_M),$$

dove

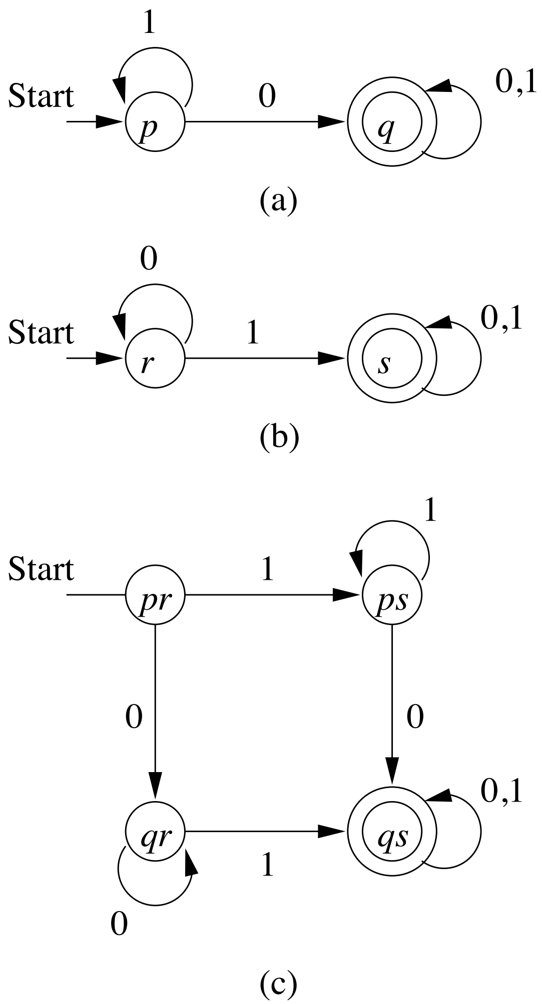
$$\delta_{L\cap M}((p,q),a) = (\delta_L(p,a), \delta_M(q,a))$$

Si può mostrare per induzione su $|w|$ che

$$\hat{\delta}_{L\cap M}((q_L, q_M), w) = \left(\hat{\delta}_L(q_L, w), \hat{\delta}_M(q_M, w)\right)$$

Esempio

$$(c) = (a) \times (b)$$



Chiusura rispetto alla differenza

Teorema 4.10 Se L e M sono linguaggi regolari, allora anche $L \setminus M$ è regolare.

Chiusura rispetto alla differenza

Teorema 4.10 Se L e M sono linguaggi regolari, allora anche $L \setminus M$ è regolare.
Dimostrazione. Osserviamo che $L \setminus M = L \cap \overline{M}$. Sappiamo già che i linguaggi regolari sono chiusi sotto il complemento e l'intersezione.

Chiusura rispetto al ”reverse”

Teorema 4.11 Se L è un linguaggio regolare, allora anche L^R è regolare.

Chiusura rispetto al "reverse"

Teorema 4.11 Se L è un linguaggio regolare, allora anche L^R è regolare.
Dimostrazione 1: Sia L riconosciuto da un FA A. Modifichiamo A per renderlo un FA per L^R:

1. Giriamo tutti gli archi.
2. Rendiamo il vecchio stato iniziale l'unico stato finale.
3. Creiamo un nuovo stato iniziale p_0, con $\delta(p_0, \epsilon) = F$ (i vecchi stati finali).

Chiusura rispetto al "reverse": un'altra dimostrazione

Se L è un linguaggio regolare, allora anche L^R è regolare.
Dimostrazione 2: Sia L descritto da un'espressione regolare E.
Costruiremo un'espressione regolare E^R, tale che
$L(E^R) = (L(E))^R$.

Chiusura rispetto al "reverse": un'altra dimostrazione

Se L è un linguaggio regolare, allora anche L^R è regolare.
Dimostrazione 2: Sia L descritto da un'espressione regolare E.
Costruiremo un'espressione regolare E^R, tale che
$L(E^R) = (L(E))^R$.
Procediamo per induzione strutturale su E.
Base: Se E è ϵ, $\emptyset$, o $\boldsymbol{a}$, allora $E^R = E$.
Induzione:

1. $E = F + G$. Allora $E^R = F^R + G^R$
2. $E = F.G$. Allora $E^R = G^R.F^R$
3. $E = F^*$. Allora $E^R = (F^R)^*$

Proprietà di decisione

1. Convertire tra diverse rappresentazioni dei linguaggi regolari.
2. È $L = \emptyset$?
3. È $w \in L$?
4. Due descrizioni definiscono lo stesso linguaggio?

Da NFA a DFA

- Supponiamo che un ϵ-NFA abbia n stati.
- Per calcolare ECLOSE(p) seguiamo al più n^2 archi. Lo facciamo per n stati, quindi in totale sono n^3 passi.
- Il DFA ha 2^n stati, per ogni stato S e ogni $a \in \Sigma$ calcoliamo $\delta_D(S, a)$ in n^3 passi, consultando l'informazione sulle ϵ-chiusure e la tabella delle transizioni. In totale abbiamo $O(n^3 2^n)$ passi.
- Se calcoliamo δ solo per gli stati raggiungibili, dobbiamo calcolare $\delta_D(S, a)$ solo s volte, dove s è il numero di stati raggiungibili. In totale: $O(n^3 s)$ passi.

Da DFA a NFA

Dobbiamo solo mettere le parentesi graffe attorno agli stati.
Totale: $O(n)$ passi.

Da FA a espressione regolare

Dobbiamo calcolare n^3 cose di grandezza fino a 4^n. Totale: $O(n^3 4^n)$.
L'FA può essere un NFA. Se prima vogliamo convertire l'NFA in un DFA, il tempo totale sarà doppiamente esponenziale.

Da espressioni regolari a FA

Possiamo costruire un albero per l'espressione in n passi.
Possiamo costruire l'automa in n passi.
Eliminare le ϵ-transizioni ha bisogno di $O(n^3)$ passi.
Se si vuole un DFA, potremmo aver bisogno di un numero esponenziale di passi.

Controllare se un linguaggio è vuoto

- $L(A) \neq \emptyset$ per FA A se e solo se uno stato finale è raggiungibile dallo stato iniziale in in A. Totale: $O(n^2)$ passi.
- Oppure, possiamo guardare un'espressione regolare E e vedere se $L(E) = \emptyset$, considerando tutti i casi:
  - $E = F + G$. Allora $L(E)$ è vuoto se e solo se sia $L(F)$ che $L(G)$ sono vuoti.
  - $E = F.G$. Allora $L(E)$ è vuoto se e solo se o $L(F)$ o $L(G)$ sono vuoti.
  - $E = F^*$. Allora $L(E)$ non è mai vuoto, perché $\epsilon \in L(E)$.
  - $E = \epsilon$. Allora $L(E)$ non è vuoto.
  - $E = \mathbf{a}$. Allora $L(E)$ non è vuoto.
  - $E = \emptyset$. Allora $L(E)$ è vuoto.

Controllare l'appartenenza ad un linguaggio

- Per controllare se $w \in L(A)$ per DFA A, simuliamo A su w. Se $|w| = n$, questo prende $O(n)$ passi.
- Se A è un NFA e ha s stati, simulare A su w prende $O(ns^2)$ passi.
- Se A è un ϵ-NFA e ha s stati, simulare A su w prende $O(ns^3)$ passi.
- Se $L = L(E)$, per l'espressione regolare E di lunghezza s, prima convertiamo E in un ϵ-NFA con $2s$ stati. Poi simuliamo w su questo automa, in $O(ns^3)$ passi.

Pumping Lemma per la non regolarità: qualche esercizio

Dimostrare che i seguenti linguaggi non sono regolari.

- L'insieme delle stringhe di parentesi bilanciate.
- $L = \{0^n 1^m | n \leq m\}$
- $L = \{0^n 1^m 2^n | n, m \text{ interi}\}$
- $L = \{0^{n^2} | n \text{ intero}\}$
- $L = \{ww | w \in \{0,1\}^*\}$
- $L = \{ww^R | w \in \{0,1\}^*\}$
- $L = \{0^i 1^j | mcd(i,j) = 1\}$