Automi e Linguaggi Formali

a.a. 2017/2018

LT in Informatica
21 Marzo 2018

Università degli Studi di Padova

Primo compitino

Giovedì 12 Aprile – ore 14:30

Aule LuM250 e LuF1

- La lista su uniweb è aperta
- Si chiude Martedì 10 aprile
- Mercoledì 11 verrà pubblicata sul moodle la ripartizione tra le due aule degli iscritti

Proprietà dei linguaggi regolari

- **Pumping Lemma.** Ogni linguaggio regolare soddisfa il pumping lemma. Se qualcuno vi presenta un falso linguaggio regolare, l'uso del pumping lemma mostrerà una contraddizione.
- **Proprietà di chiusura.** Come costruire automi da componenti usando delle operazioni, ad esempio dati L e M possiamo costruire un automa per $L \cap M$.
- **Proprietà di decisione.** Analisi computazionale di automi, cioè quanto costa controllare varie proprietà, come l'equivalenza di due automi.

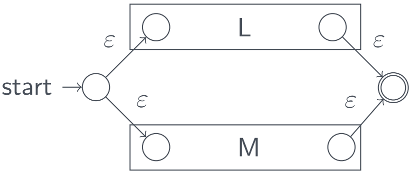
Proprietà di chiusura

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

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

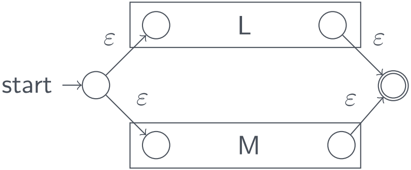
Unione, concatenazione e chiusura di Kleene

- $\mathbf{L} + \mathbf{M}$



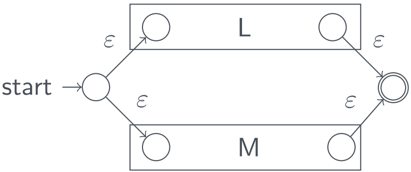
Unione, concatenazione e chiusura di Kleene

- $\mathbf{L + M}$



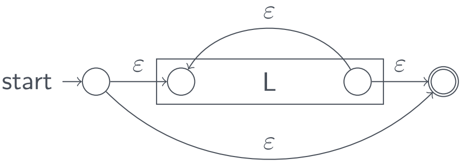
- $\mathbf{L.M}$

- $L + M$



- $L.M$

- L^*



Chiusura per intersezione

Theorem

Se L e M sono regolari, allora anche $L \cap M$ è un linguaggio regolare.

Chiusura per intersezione

Theorem

Se L e M sono regolari, allora anche $L \cap M$ è un linguaggio regolare.

Dimostrazione. Sia L il linguaggio di

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

e M il linguaggio di

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

Possiamo assumere 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.

Dimostrazione (continua).
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.

Dimostrazione (continua).
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

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

Dimostrazione (continua).
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

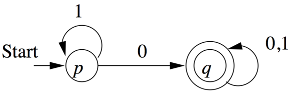
$$\delta_{L\cap M}((p, q), a) = (\delta_L(p, a), \delta_M(q, a))$$

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

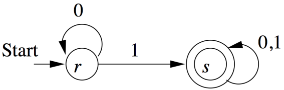
$$\hat{\delta}_{L\cap M}((q_L, q_M), w) = \hat{\delta}_L(q_L, w), \hat{\delta}_M(q_M, w)$$

Esempio

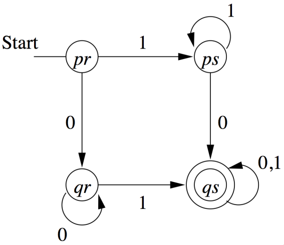
Costruiamo l'automa che rappresenta l'intersezione di (a) e (b)



(a)



(b)



Chiusura per complementazione

Università degli Studi di Padova

Theorem

Sia L un linguaggio regolare sull'alfabeto Σ, allora il linguaggio $\overline{L} = \Sigma^* \setminus L$ è regolare.

Chiusura per complementazione

Theorem

Sia L un linguaggio regolare sull'alfabeto Σ, allora il linguaggio $\overline{L} = \Sigma^* \setminus L$ è regolare.

Dimostrazione. Sia L il linguaggio riconosciuto dal DFA

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

Chiusura per complementazione

Theorem

Sia L un linguaggio regolare sull'alfabeto Σ, allora il linguaggio $\overline{L} = \Sigma^* \setminus L$ è regolare.

Dimostrazione. Sia L il linguaggio riconosciuto dal DFA

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

Scambiamo gli stati finali e non finali di A, ottenendo l'automa

$$B = (Q, \Sigma, \delta, q_0, Q \setminus F)$$

Chiusura per complementazione

Theorem

Sia L un linguaggio regolare sull'alfabeto Σ, allora il linguaggio $\overline{L} = \Sigma^* \setminus L$ è regolare.

Dimostrazione. Sia L il linguaggio riconosciuto dal DFA

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

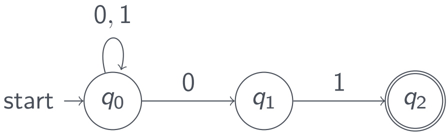
Scambiamo gli stati finali e non finali di A, ottenendo l'automa

$$B = (Q, \Sigma, \delta, q_0, Q \setminus F)$$

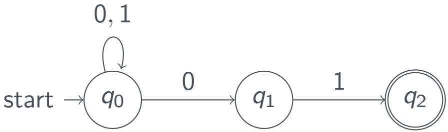
Allora $L(B) = \overline{L}$

Esempio

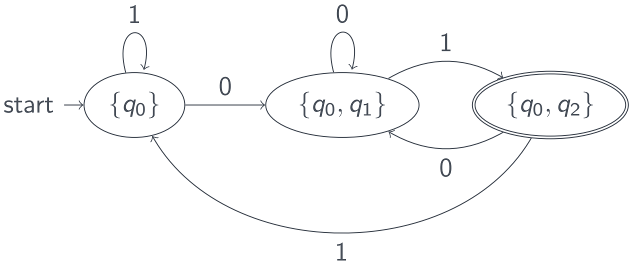
- Costruiamo il complementare dell'NFA:



- Costruiamo il complementare dell'NFA:

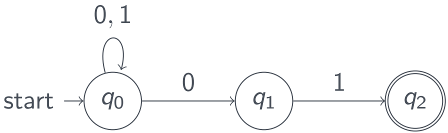


- Per prima cosa determinizziamo l'automa

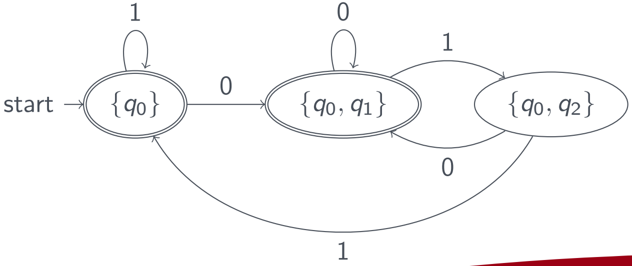


Esempio

- Costruiamo il complementare dell'NFA:



- Per prima cosa determinizziamo l'automa
- e poi scambiamo stati finali e non finali



Chiusura per differenza

Università degli Studi di Padova

Theorem

Se L e M sono linguaggi regolari, allora il linguaggio $L \setminus M$ è regolare.

Chiusura per differenza

Theorem

Se L e M sono linguaggi regolari, allora il linguaggio $L \setminus M$ è regolare.

Dimostrazione. Osserviamo che $L \setminus M = L \cap \overline{M}$. Sappiamo già che i linguaggio regolari sono chiusi per intersezione e complementazione. Quindi $L \cap \overline{M}$ è un linguaggio regolare.

Chiusura per inversione

Università degli Studi di Padova

Theorem

Se L è un linguaggio regolare, allora il linguaggio L^R è regolare.

Chiusura per inversione

Theorem

Se L è un linguaggio regolare, allora il linguaggio L^R è regolare.

Dimostrazione. Sia L il linguaggio riconosciuto dall'FA A. Modifichiamo A per fargli riconoscere il linguaggio L^R:

Chiusura per inversione

Theorem

Se L è un linguaggio regolare, allora il linguaggio L^R è regolare.

Dimostrazione. Sia L il linguaggio riconosciuto dall'FA A. Modifichiamo A per fargli riconoscere il linguaggio L^R:

1. Giriamo tutte le transizioni dell'automa

Chiusura per inversione

Theorem

Se L è un linguaggio regolare, allora il linguaggio L^R è regolare.

Dimostrazione. Sia L il linguaggio riconosciuto dall'FA A.
Modifichiamo A per fargli riconoscere il linguaggio L^R:

1. Giriamo tutte le transizioni dell'automa
2. Il vecchio stato iniziale diventa l'unico stato finale

Chiusura per inversione

Università degli Studi di Padova

Theorem

Se L è un linguaggio regolare, allora il linguaggio L^R è regolare.

Dimostrazione. Sia L il linguaggio riconosciuto dall'FA A.
Modifichiamo A per fargli riconoscere il linguaggio L^R:

1. Giriamo tutte le transizioni dell'automa
2. Il vecchio stato iniziale diventa l'unico stato finale
3. Creiamo un nuovo stato iniziale p_0 tale che $\delta(p_0, \varepsilon) = F$ (c'è una ϵ-transizione da p_0 verso i vecchi stati finiali)

Proprietà di decisione

- Convertire tra diverse rappresentazioni dei linguaggio regolari
- Il linguaggio L è vuoto?
- La parola w appartiene al linguaggio L?
- Due descrizioni definiscono lo stesso linguaggio?

Controllare se un linguaggio è vuoto

- $L(A) \neq \emptyset$ se e solo se esiste uno stato finale raggiungibile dallo stato iniziale.

Controllare se un linguaggio è vuoto

- $L(A) \neq \emptyset$ se e solo se esiste uno stato finale raggiungibile dallo stato iniziale.
- Oppure possiamo analizzare l'espressione regolare $\mathbf{E}$ e vedere se $L(\mathbf{E}) = \emptyset$ ragionando per casi:
  - $\mathbf{E} = \mathbf{F} + \mathbf{G}$. Allora $L(\mathbf{E})$ è vuoto se e solo se sia $L(\mathbf{F})$ e $L(\mathbf{G})$ sono vuoti;
  - $\mathbf{E} = \mathbf{F}.\mathbf{G}$. Allora $L(\mathbf{E})$ è vuoto se e solo $L(\mathbf{F})$ è vuoto oppure $L(\mathbf{G})$ è vuoto;
  - $\mathbf{E} = \mathbf{F}^*$. Allora $L(\mathbf{E})$ non è vuoto perche $\varepsilon \in L(\mathbf{E})$
  - $\mathbf{E} = \varepsilon$. Allora $L(\mathbf{E})$ non è vuoto perche $\varepsilon \in L(\mathbf{E})$
  - $\mathbf{E} = \mathbf{a}$. Allora $L(\mathbf{E})$ non è vuoto perche $a \in L(\mathbf{E})$
  - $\mathbf{E} = \emptyset$. Allora $L(\mathbf{E})$ è vuoto

Controllare l'appartenenza

- Per controllare se $w \in L(A)$ per un FA A dobbiamo simulare A su w:
  - se $|w| = n$ dobbiamo fare n passi di simulazione
  - se A è un DFA ogni passo costa $O(1)$
  - se A è un NFA con s stati ogni passo costa $O(s^2)$
  - se A è un ε-NFA con s stati ogni passo costa $O(s^3)$

Controllare l'appartenenza

- Per controllare se $w \in L(A)$ per un FA A dobbiamo simulare A su w:
  - se $|w| = n$ dobbiamo fare n passi di simulazione
  - se A è un DFA ogni passo costa $O(1)$
  - se A è un NFA con s stati ogni passo costa $O(s^2)$
  - se A è un ε-NFA con s stati ogni passo costa $O(s^3)$
- Se L è rappresentato con un'espressione regolare **E**, prima convertiamo **E** in ε-NFA e poi simuliamo w su questo automa.

Controllare l'equivalenza

- $L = M$ se e solo se:
  - $L \cap \overline{M}$ è vuoto e
  - $\overline{L} \cap M$ è vuoto

Controllare l'equivalenza

- $L = M$ se e solo se:
  - $L \cap \overline{M}$ è vuoto e
  - $\overline{L} \cap M$ è vuoto
- I linguaggi regolari sono chiusi per intersezione e complementazione
- Sappiamo come controllare se un linguaggio regolare è vuoto