

Proprietà dei linguaggi regolari

a.a. 2020-2021

Corso di Fondamenti di Informatica - 1 modulo

Corso di Laurea in Informatica

Università di Roma "Tor Vergata"

Prof. Giorgio Gambosi



Proprietà di chiusura

Proprietà di chiusura dei linguaggi regolari

Dato l'insieme dei linguaggi regolari, ci chiediamo quali siano le sue proprietà di chiusura rispetto a varie operazioni.

Ci chiediamo quindi se, data una qualunque operazione \diamond :

- se \diamond è unaria, $\diamond L$ è un linguaggio regolare per ogni L regolare
- se \diamond è binaria, $L_1 \diamond L_2$ è un linguaggio regolare per ogni coppia L_1, L_2 di linguaggi regolari
- non consideriamo operatori di "arità" maggiore

Le dimostrazioni di tali proprietà presentano tutte uno stesso schema, in cui, a partire dagli ASFD che riconoscono i linguaggi regolari dati, viene derivato un automa (deterministico o non deterministico) che riconosce il linguaggio risultante.

Chiusura dei linguaggi regolari: unione

Dati L_1 e L_2 , la loro unione $L_1 \cup L_2$ è un linguaggio regolare.

Infatti, siano $\mathcal{A}_1 = \langle \Sigma_1, Q_1, \delta_{N_1}, q_{0_1}, F_1 \rangle$ e $\mathcal{A}_2 = \langle \Sigma_2, Q_2, \delta_{N_2}, q_{0_2}, F_2 \rangle$, due ASFD che accettano L_1 e L_2 . Costruiamo da \mathcal{A}_1 e \mathcal{A}_2 un automa $\mathcal{A} = \langle \Sigma, Q, \delta_N, q_0, F \rangle$ che riconosce il linguaggio $L_1 \cup L_2$.

Chiusura dei linguaggi regolari: unione

Costruzione:

1. $\Sigma = \Sigma_1 \cup \Sigma_2$.
2. $Q = Q_1 \cup Q_2 \cup \{q_0\}$.
3. $F = F_1 \cup F_2$, oppure $F = F_1 \cup F_2 \cup \{q_0\}$ se uno dei due automi \mathcal{A}_1 , \mathcal{A}_2 riconosce anche la stringa vuota.
4. La funzione di transizione δ_N è definita come:

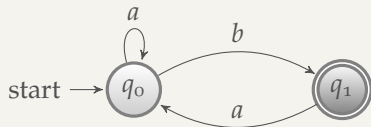
$$\delta_N(q, a) = \delta_{N_1}(q, a) \text{ se } q \in Q_1, a \in \Sigma_1$$

$$\delta_N(q, a) = \delta_{N_2}(q, a) \text{ se } q \in Q_2, a \in \Sigma_2$$

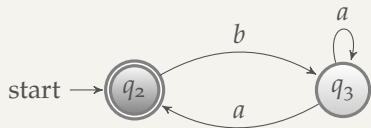
$$\delta_N(q_0, a) = \delta_{N_1}(q_{0_1}, a) \cup \delta_{N_2}(q_{0_2}, a), a \in \Sigma.$$

Esempio

\mathcal{A}_1

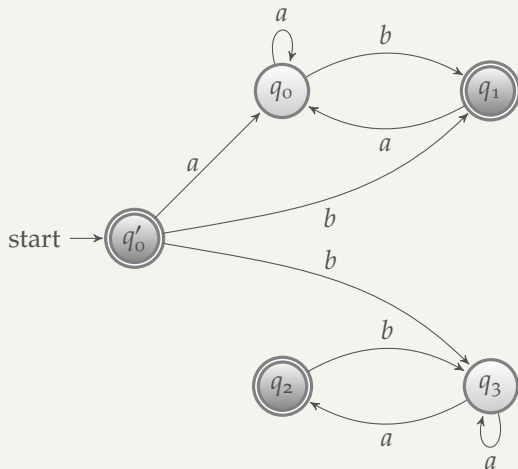


\mathcal{A}_2



Esempio

\mathcal{A}



Chiusura dei linguaggi regolari: complemento

Dato un linguaggio regolare L , il suo complemento \bar{L} è un linguaggio regolare.

Infatti, sia $\mathcal{A} = \langle \Sigma, Q, \delta, q_0, F \rangle$ un automa deterministico che riconosce L , con δ funzione totale: l'automata

$$\bar{\mathcal{A}} = \langle \Sigma, Q, \delta, q_0, \{Q - F\} \rangle;$$

riconosce allora il linguaggio \bar{L} .

Infatti, ogni stringa che porta l'automata \mathcal{A} in uno stato finale porta l'automata $\bar{\mathcal{A}}$ in uno stato non finale e, viceversa, ogni stringa che porta $\bar{\mathcal{A}}$ in uno stato finale porta \mathcal{A} in uno stato non finale

Chiusura dei linguaggi regolari: intersezione

Dati due linguaggi regolari L_1 e L_2 , la loro intersezione $L = L_1 \cap L_2$ è un linguaggio regolare.

È sufficiente osservare che, per la legge di De Morgan,

$$L = L_1 \cap L_2 \equiv \overline{\overline{L_1} \cup \overline{L_2}}.$$

Chiusura dei linguaggi regolari: concatenazione

Dati due linguaggi regolari L_1 e L_2 , la loro concatenazione $L = L_1 \circ L_2$ è un linguaggio regolare.

Infatti, siano $\mathcal{A}_1 = \langle \Sigma_1, Q_1, \delta_{N_1}, q_{0_1}, F_1 \rangle$ e $\mathcal{A}_2 = \langle \Sigma_2, Q_2, \delta_{N_2}, q_{0_2}, F_2 \rangle$, due ASFD che accettano L_1 e L_2 . Costruiamo da \mathcal{A}_1 e \mathcal{A}_2 un automa $\mathcal{A} = \{\Sigma, Q, \delta_N, q_0, F\}$ che riconosce il linguaggio $L_1 \circ L_2$.

Chiusura dei linguaggi regolari: concatenazione

Costruzione:

1. $\Sigma = \Sigma_1 \cup \Sigma_2$;
2. $Q = Q_1 \cup Q_2$;
3. $F = \begin{cases} F_2 & \text{se } \varepsilon \notin L(\mathcal{A}_2), \\ F_1 \cup F_2 & \text{altrimenti;} \end{cases}$
4. $q_0 = q_{01}$;
5. δ_N è definita come:

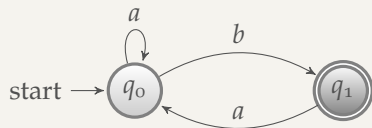
$$\delta_N(q, a) = \delta_1(q, a), \forall q \in Q_1 - F_1, a \in \Sigma_1$$

$$\delta_N(q, a) = \delta_1(q, a) \cup \delta_2(q_{02}, a), \forall q \in F_1, a \in \Sigma$$

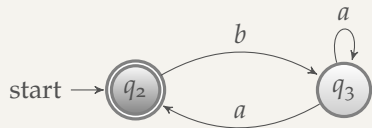
$$\delta_N(q, a) = \delta_2(q, a), \forall q \in Q_2, a \in \Sigma_2.$$

Esempio

\mathcal{A}_1

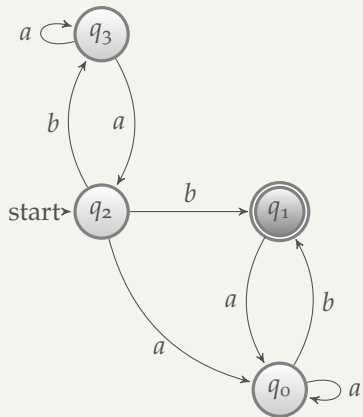


\mathcal{A}_2



Esempio

\mathcal{A}



Chiusura dei linguaggi regolari: iterazione

Dato un linguaggio regolare L , la sua iterazione L^* è un linguaggio regolare.

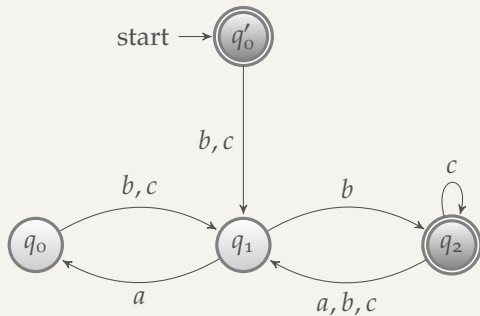
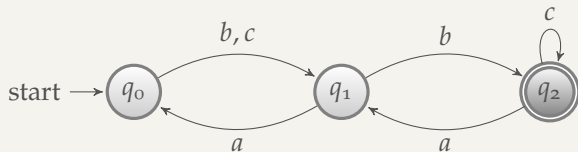
Infatti, sia $\mathcal{A} = \langle \Sigma, Q, \delta_N, q_0, F \rangle$ un ASFD che accetta L . Costruiamo da \mathcal{A} un automa $\mathcal{A}' = \langle \Sigma, Q \cup \{q'_0\}, \delta', q'_0, F \cup \{q'_0\} \rangle$ che riconosce il linguaggio L^* ponendo

$$\delta'(q, a) = \delta(q, a), \quad \forall q \in Q - F$$

$$\delta'(q, a) = \delta(q, a) \cup \delta(q_0, a), \quad \forall q \in F$$

$$\delta'(q'_0, a) = \delta(q_0, a).$$

Esempio



Decidibilità di predicati

Teorema

È possibile decidere se il linguaggio accettato da un dato automa a stati finiti è vuoto.

Osservazione: se un automa con n stati accetta qualche stringa, allora accetta una stringa di lunghezza inferiore ad n .

Sia z la stringa più breve accettata dall'automa, se fosse $|z| \geq n$ allora, in base al pumping lemma, z potrebbe essere scritta come uvw e anche la stringa uw , più breve di z , sarebbe accettata dall'automa.

Decidibilità di predicati su LR

Dato un automa \mathcal{A} con n stati, per decidere se $L(\mathcal{A}) = \Lambda$ basta verificare se \mathcal{A} accetta almeno una delle $\sum_{i=0}^{n-1} |\Sigma|^i = O(|\Sigma|^{n-1})$ stringhe di lunghezza inferiore ad n .

Il tempo necessario per tale operazione sarà:

$$\sum_{i=0}^{n-1} |\Sigma|^i = \frac{n}{(n-1)^2} ((|\Sigma| - 1)n|\Sigma|^n - |\Sigma|^n + 1) = O(|\Sigma|^n)$$

L'algoritmo ha quindi complessità esponenziale.

Decidibilità di predicati su LR

Osservazione: dato il grafo di transizione di \mathcal{A} , $L(\mathcal{A})$ è non vuoto se e solo se esiste almeno uno stato finale raggiungibile da q_0 .

Algoritmo polinomiale: visita del grafo di transizione, avente n nodi e $m \leq n|\Sigma|$ archi (se \mathcal{A} deterministico, altrimenti $m \leq n^2$).

La visita richiede tempo $O(n + m)$ polinomiale.

Teorema

È possibile decidere se il linguaggio accettato da un dato automa a stati finiti è finito.

Osservazione 1: se \mathcal{A} accetta una z lunga almeno n allora, per il pumping lemma, accetta infinite stringhe.

Osservazione 2: se $L(\mathcal{A})$ è infinito, allora esiste $z \in L(\mathcal{A})$ tale che $n \leq |z| \leq 2n$.

- chiaramente $|z| > n$, altrimenti L sarebbe finito
- per la parte $|z| < 2n$, assumiamo che ciò non sia vero (e quindi che $|z| = k \geq 2n$): allora, dato che il pumping lemma vale, possiamo scrivere $z = uvw$, con $1 \leq |v| \leq n$, $uw \in L(\mathcal{A})$ e $2n - n = n \leq k - n \leq |uw| \leq k - 1$.
Se $n \leq |uw| \leq 2n$, allora uw è la stringa cercata. Altrimenti, l'argomento si applica su uw , ottenendo una stringa di lunghezza inferiore, ma maggiore o uguale a n .

In conclusione:

$L(\mathcal{A})$ è infinito se e solo se esiste $z \in L(\mathcal{A})$ tale che $n \leq |z| \leq 2n$.

Dato un automa \mathcal{A} con n stati, per decidere se $L(\mathcal{A})$ è infinito basta verificare se \mathcal{A} accetta almeno una delle $\sum_{i=n}^{2n} |\Sigma|^i = O(|\Sigma|^{2n})$ stringhe di lunghezza compresa tra n e $2n$.

Il tempo necessario per tale operazione sarà evidentemente $O(|\Sigma|^{2n})$: l'algoritmo ha quindi complessità esponenziale.

Osservazione: dato il grafo di transizione di \mathcal{A} , $L(\mathcal{A})$ è infinito se e solo se esiste almeno un ciclo a partire da uno stato su un cammino da q_0 a uno stato finale.

Algoritmo polinomiale: visita del grafo di transizione, avente n nodi e $m \leq n|\Sigma|$ archi (se \mathcal{A} deterministico, altrimenti $m \leq n^2$).

La visita richiede tempo $O(n + m)$, polinomiale.

Dalla possibilità di decidere se un linguaggio regolare è vuoto, ne deriva immediatamente la decidibilità di altri predicati sui linguaggi regolari.

Teorema

Il problema dell'equivalenza di due linguaggi regolari è decidibile.

Decidibilità di predicati su LR

Per dimostrare l'equivalenza di due linguaggi, basta dimostrare che la loro intersezione coincide con la loro unione, cioè che la loro differenza simmetrica (l'unione dell'intersezione del primo con il complemento del secondo e dell'intersezione del secondo con il complemento del primo) è il linguaggio vuoto.