Espressioni regolari

Arturo Carpi

Dipartimento di Matematica e Informatica
Università di Perugia

Corso di Linguaggi Formali e Compilatori - a.a. 2021/22

Operazioni sui linguaggi

1. unione, intersezione, complemento,
2. concatenazione: $L_1 L_2 = \{uv \mid u \in L_1,\ v \in L_2\}$,
3. potenza: $L^n = \underbrace{L\,L \cdots L}_{n\ volte}$,
4. chiusura di Kleene:

$$L^* = \bigcup_{n \geq 0} L^n = \{u_1 u_2 \cdots u_n \mid n \geq 0,\ u_1, \ldots, u_n \in L\}$$

Osservazione

Unione, concatenazione e chiusura di Kleene sono dette operazioni regolari.

Esempi

Se $L_1 = \{ab, cb\}$, $L_2 = \{aa, c\}$, allora

$$L_1 L_2 = \{abaa, abc, cbaa, cbc\}.$$

Se $L = \{a, ab\}$, allora

$$L^0 = \{\varepsilon\}, \quad L^1 = L, \quad L^2 = \{aa, aab, aba, abab\},$$
$$L^3 = \{aaa, aaab, aaba, aabab, abaa, abaab, ababa, ababab\}.$$

$$\begin{aligned} L^* &= \{\varepsilon\} \cup L \cup L^2 \cup \cdots \cup L^n \cup \cdots \\ &= \{\varepsilon, a, ab, aa, aab, aba, abab, aaaaaa, aaab, aaba, \ldots\} \\ &= \{\varepsilon\} \cup \{a^{n_1} ba^{n_2} b \cdots a^{n_{k-1}} ba^{n_k} \mid k, n_1, \ldots, n_{k-1} \geq 1,\ n_k \geq 0\} \end{aligned}$$

(parole che iniziano con a e non contengono due b consecutive)

Definizione

Sia $\widehat{\Sigma}$ l'alfabeto ottenuto aggiungendo a Σ le lettere $\emptyset$, $+$, $*$, $($, $)$.
Si dicono espressioni regolari sull'alfabeto Σ le parole sull'alfabeto $\widehat{\Sigma}$ che si ottengono applicando un numero finito di volte le regole seguenti:

- (i) Ogni lettera $a \in \Sigma$ è un'espressione regolare,
  $\emptyset$ è un'espressione regolare,
- (ii) Se E e F sono espressioni regolari, allora $(E+F)$, (EF) e E^* sono espressioni regolari.

A ogni espressione regolare è associato un linguaggio, detto linguaggio denotato dall'espressione regolare e definito dalle regole seguenti:

- (i) per ogni $a \in \Sigma$, l'espressione regolare a denota il linguaggio $\{a\}$;
  l'espressione regolare $\emptyset$ denota il linguaggio vuoto.
- (ii) Detti L_E e L_F i linguaggi denotati dalle espressioni regolari E ed F, i linguaggi denotati dalle espressioni regolari $(E+F)$, (EF), E^* sono, rispettivamente, $L_E \cup L_F$, $L_E L_F$, ${L_E}^*$.

Definizione

I linguaggi denotati da espressioni regolari si dicono linguaggi regolari.

Osservazione

La classe dei linguaggi regolari è la più piccola famiglia di linguaggi che

- contiene i linguaggi finiti,
- è chiusa per le operazioni regolari.

Osservazione

Possiamo omettere qualche parentesi, rispettando la priorità:

1. chiusura di Kleene,
2. concatenazione,
3. somma.

Esempio

$bab + ab^*$ equivale a $(((ba)b) + (ab^*))$.

Esempi

1. $a + b$
2. $(a + b)^*$
3. $\emptyset^*$
4. $((a + b)(a + b)(a + b))^*$
5. $(a + ab)^*$
6. $(a + b)^* abb$
7. $(b^* ab^* ab^*)^* b^* ab^*$
8. $(a + b + c)^* abac(a + b + c)^*$
9. $((a + b + c),)^*(a + b + c)\&(a + b + c) + a + b + c$

Teorema (Kleene) Un linguaggio è regolare se e soltanto se è riconosciuto da un automa a stati finiti.

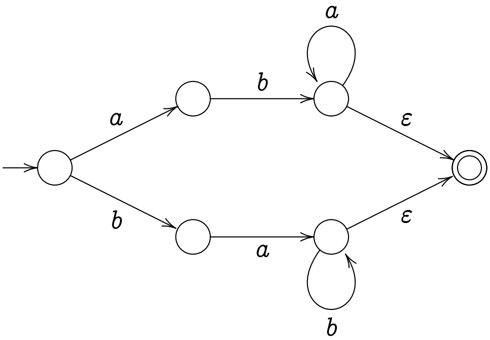
Esite una procedura effettiva che

- Data un'espressione regolare, produce un automa a stati finiti che accetta il linguaggio denotato da tale espressione (sintesi),
- Dato un automa a stati finiti, produce un'espressione regolare che denota il linguaggio accettato da tale automa (analisi).

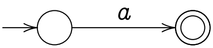
Dall'espressione regolare all'automa

Utilizzeremo esclusivamente automi non deterministici con ε-transizioni con un unico stato finale (non restrittivo)

Esempio

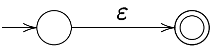


Base



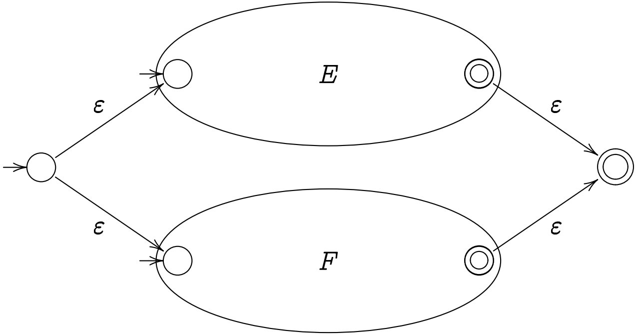
$E = a$

$E = \emptyset$



$E = \varepsilon \ (= \emptyset^*)$

Unione



$E + F$

Unione

Siano $\mathcal{A}_1 = \langle Q_1, \Sigma, \delta_1, i_1, \{f_1\}\rangle$ e $\mathcal{A}_2 = \langle Q_2, \Sigma, \delta_2, i_2, \{f_2\}\rangle$ gli automi che riconoscono i linguaggi denotati dalle espressioni regolari E e F.

Supponiamo $Q_1 \cap Q_2 = \emptyset$.

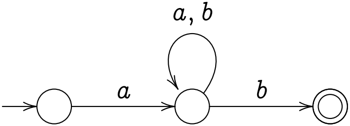
Costruiamo $\mathcal{A} = \langle Q, \Sigma, \delta, i, \{f\}\rangle$ come segue:

- $Q = Q_1 \cup Q_2 \cup \{i, f\}$, ove i, f sono due nuovi stati,
- i e f sono, rispettivamente lo stato iniziale e l'unico stato finale,
- gli archi del grafo di $\mathcal{A}$ sono gli archi dei grafi di $\mathcal{A}_1$ e $\mathcal{A}_2$ e, inoltre, (i, ε, i_1), (i, ε, i_2), (f_1, ε, f), (f_2, ε, f).

Allora $\mathcal{A}$ accetta il linguaggio denotato da $(E + F)$.

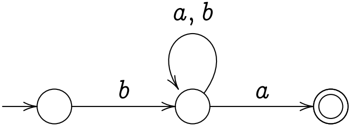
Esempio

$\mathcal{A}_1$

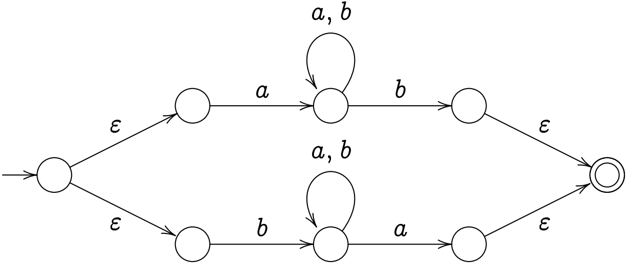


$a(a+b)^*b$

$\mathcal{A}_2$

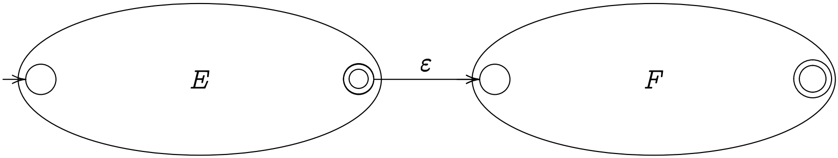


$b(a+b)^*a$



$a(a+b)^*b + b(a+b)^*a$

Concatenazione



EF

Concatenazione

Siano $\mathcal{A}_1 = \langle Q_1, \Sigma, \delta_1, i_1, \{f_1\} \rangle$ e $\mathcal{A}_2 = \langle Q_2, \Sigma, \delta_2, i_2, \{f_2\} \rangle$ gli automi che riconoscono i linguaggi denotati dalle espressioni regolari E e F.

Supponiamo $Q_1 \cap Q_2 = \emptyset$.

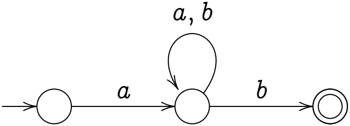
Costruiamo $\mathcal{A} = \langle Q, \Sigma, \delta, i, \{f\} \rangle$ come segue:

- $Q = Q_1 \cup Q_2$,
- lo stato iniziale è quello di $\mathcal{A}_1$ e lo stato finale è quello di $\mathcal{A}_2$,
- gli archi del grafo di $\mathcal{A}$ sono gli archi dei grafi di $\mathcal{A}_1$ e $\mathcal{A}_2$ con l'aggiunta dell'arco (f_1, ε, i_2).

Allora $\mathcal{A}$ accetta il linguaggio denotato da (EF).

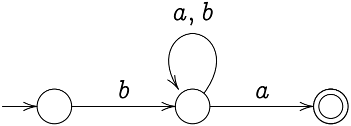
Esempio

$\mathcal{A}_1$

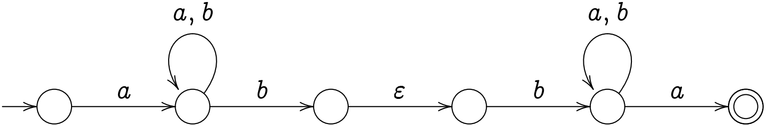


$a(a+b)^*b$

$\mathcal{A}_2$

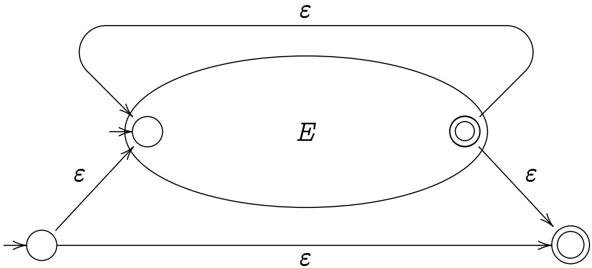


$b(a+b)^*a$



$a(a+b)^*bb(a+b)^*a$

Chiusura di Kleene



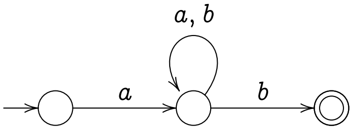
E^*

Sia $\mathcal{A}_1 = \langle Q_1, \Sigma, \delta_1, i_1, \{f_1\} \rangle$ l'automa che riconosce il linguaggio denotato dall'espressione regolare E. Costruiamo $\mathcal{A} = \langle Q, \Sigma, \delta, i, \{f\} \rangle$ come segue:

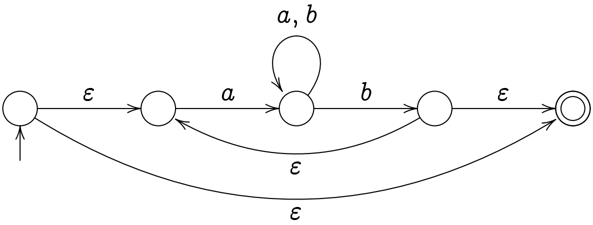
- $Q = Q_1 \cup \{i, f\}$, ove i, f sono due nuovi stati,
- i e f sono, rispettivamente lo stato iniziale e l'unico stato finale,
- gli archi del grafo di $\mathcal{A}$ sono gli archi del grafo di $\mathcal{A}_1$ e, inoltre, $(f_1, \varepsilon, i_1), (i, \varepsilon, i_1),\ (f_1, \varepsilon, f),\ (i, \varepsilon, f)$.

Allora $\mathcal{A}$ accetta il linguaggio denotato da E^*.

Esempio



$a(a+b)^*b$



$a(a+b)^*b^*$

L'algoritmo di sintesi

Proposizione Data un'espressione regolare G, si può effettivamente costruire un automa a stati finiti che riconosce il linguaggio denotato da G.

1. se G è un'espressione regolare di base (lettera o insieme vuoto), allora
   restituisco l'automa corrispondente;
2. se $G = (E + F)$, allora
   calcolo gli automi $\mathcal{A}_1$ e $\mathcal{A}_2$ corrispondenti a E e F;
   costruisco l'automa dell'espressione $(E + F)$;
3. se $G = (EF)$, allora
   calcolo gli automi $\mathcal{A}_1$ e $\mathcal{A}_2$ corrispondenti a E e F;
   costruisco l'automa dell'espressione (EF);
4. se $G = E^*$, allora
   calcolo l'automa $\mathcal{A}_1$ corrispondente a E;
   costruisco l'automa dell'espressione E^*;

Esempio

$G = (a + b)^* abb = ((((a + b)^* a) b) b)$