Chiusura di Linguaggi Regolari

Created	@May 9, 2025 8:37 AM
Class	Fondamenti di informatica: linguaggi formali

Proprietà di chiusura degli ASF

Un linguaggio L si dice **regolare** se esiste un **Automa a Stati Finiti** (ASF), deterministico o non deterministico, che lo accetta. In altre parole:

$$L = L(A)$$

La **classe dei linguaggi regolari** viene indicata con $\mathcal{R}$.

Epsilon-transizioni

Alcuni ASF implementano un tipo speciale di transizione chiamata **epsilon-transizione**, che permette di cambiare stato **senza consumare simboli** dall'input.

Nel caso di un **ASF non deterministico con epsilon-transizioni (ASFND-ε)**, la funzione di transizione è definita come:

$$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow \mathcal{P}(Q)$$

Poiché per ogni configurazione possono esistere **più transizioni possibili**, la computazione dell'automa non è deterministica.

Chiusura della classe $\mathcal{R}$

La classe dei linguaggi regolari $\mathcal{R}$ è **chiusa** rispetto a diverse operazioni, tra cui:

- **Unione**
- **Concatenazione**
- **Stella di Kleene**
- **Complementazione**
- **Intersezione**
- **Differenza**

L'uso degli ASFND con epsilon-transizioni è particolarmente utile per **dimostrare la chiusura rispetto all'unione** (e anche per la concatenazione e la stella di Kleene), perché consente di costruire nuovi automi in modo semplice:

La costruzione di Thompson parte da **espressioni regolari** o **linguaggi regolari semplici** e costruisce, in modo ricorsivo, un **AFN con epsilon-transizioni** che riconosce il linguaggio desiderato, e garantisce che per ogni espressione regolare esiste un **ASFND con epsilon-transizioni,** e che questo automa accetta esattamente il **linguaggio dell'espressione regolare.**

Quindi mostra **costruttivamente** che i linguaggi regolari sono **chiusi rispetto a unione, concatenazione e chiusura riflessiva.**

Chiusura rispetto all'unione

Teorema 3.6 *Dati due linguaggi regolari L_1 e L_2, la loro unione $L_1 \cup L_2$ è un linguaggio regolare.*

Dimostrazione. Siano dati due qualunque automi deterministici $\mathcal{A}_1 = \langle \Sigma_1, Q_1, \delta_{N_1}, q_{0_1}, F_1 \rangle$ e $\mathcal{A}_2 = \langle \Sigma_2, Q_2, \delta_{N_2}, q_{0_2}, F_2 \rangle$, che accettano i linguaggi $L_1 = L(\mathcal{A}_1)$ e $L_2 = L(\mathcal{A}_2)$, rispettivamente. Mostriamo ora come sia possibile, a partire da $\mathcal{A}_1$ e $\mathcal{A}_2$, costruire un automa $\mathcal{A} = \{\Sigma, Q, \delta_N, q_0, F\}$ che riconosce il linguaggio $L = L(\mathcal{A}) = L(\mathcal{A}_1) \cup L(\mathcal{A}_2)$.

La costruzione procede nel modo seguente:

- $\Sigma = \Sigma_1 \cup \Sigma_2$.
- $Q = Q_1 \cup Q_2 \cup \{q_0\}$.
- $F = F_1 \cup F_2$, 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.
- La funzione di transizione δ_N viene definita come segue:

$$\begin{aligned}
\delta_N(q,a) &= \delta_{N_1}(q,a) \text{ se } q \in Q_1, a \in \Sigma_1 \\
\delta_N(q,a) &= \delta_{N_2}(q,a) \text{ se } q \in Q_2, a \in \Sigma_2 \\
\delta_N(q_0,a) &= \delta_{N_1}(q_{0_1},a) \cup \delta_{N_2}(q_{0_2},a), a \in \Sigma.
\end{aligned}$$

Si può verificare immediatamente che l'automa così definito accetta tutte e sole le stringhe in $L(\mathcal{A}_1)$ o in $L(\mathcal{A}_2)$. □

Si noti che anche se gli automi di partenza $\mathcal{A}_1$ e $\mathcal{A}_2$ sono deterministici, l'automa $\mathcal{A}$ costruito secondo la procedura testé descritta potrà risultare non deterministico.

L'automa $\mathcal{A} = \langle \Sigma, Q, \delta_N, q_0', F\}\rangle$ che accetta l'unione dei linguaggi accettati da $\mathcal{A}_1$ e da $\mathcal{A}_2$, si costruisce come segue.

- $\Sigma = \{a, b\}$.
- $Q = \{q_0', q_0, q_1, q_2, q_3\}$, dove q_0' è il un nuovo stato iniziale.
- $F = \{q_1, q_2\} \cup \{q_0'\}$ (perché il secondo automa riconosce anche la stringa vuota).
- La funzione di transizione δ_N è data da

$$\begin{aligned}
\delta_N(q_0',a) &= \big\{\delta_1(q_0,a)\big\} \\
\delta_N(q_0',b) &= \big\{\delta_1(q_0,b), \delta_2(q_2,b)\big\} \\
\delta_N(q_0,a) &= \big\{\delta_1(q_0,a)\big\} \\
\delta_N(q_0,b) &= \big\{\delta_1(q_0,b)\big\} \\
\delta_N(q_1,a) &= \big\{\delta_1(q_1,a)\big\} \\
\delta_N(q_2,b) &= \big\{\delta_2(q_2,b)\big\} \\
\delta_N(q_3,a) &= \big\{\delta_2(q_3,a)\big\} \\
\delta_N(q_3,b) &= \big\{\delta_2(q_3,b)\big\}.
\end{aligned}$$

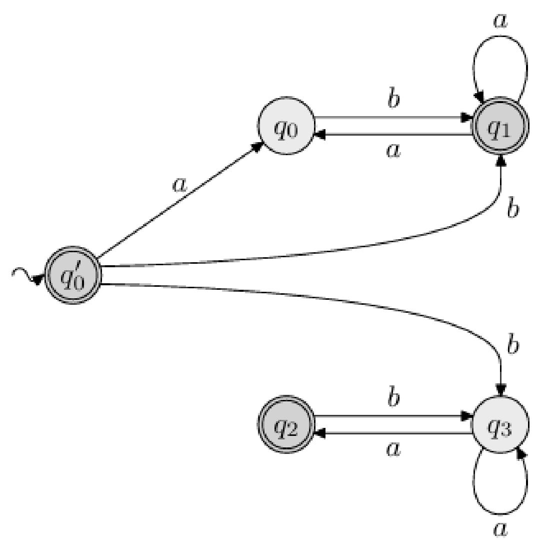

FIGURA 3.18 Automa che riconosce l'unione tra $L(\mathcal{A}_1)$ e $L(\mathcal{A}_2)$.

Chiusura rispetto al complemento

Teorema 3.7 *Dato un linguaggio regolare L, il suo complemento $\overline{L}$ è un linguaggio regolare.*

Dimostrazione. Sia

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

un automa deterministico che riconosce il linguaggio $L = L(\mathcal{A})$: si può allora costruire l'automa

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

che riconosce il linguaggio $\overline{L(\mathcal{A})}$. Infatti, ogni stringa che porta l'automa $\mathcal{A}$ in uno stato finale porta l'automa $\overline{\mathcal{A}}$ in uno stato non finale e, viceversa, ogni stringa che porta $\overline{\mathcal{A}}$ in uno stato finale porta $\mathcal{A}$ in uno stato non finale, per cui

$$L(\overline{\mathcal{A}}) = \Sigma^* - L(\mathcal{A}) = \overline{L(\mathcal{A})}.$$

□

Si noti che, per applicare la costruzione mostrata nel teorema precedente è necessario assumere che la funzione di transizione dell'automa dato sia totale.

quindi, è necessario che l'automa che riconosce il complemento di un linguaggio regolare sia in particolare un ASFD, in particolare:

Prima di tutto, osservando che l'automa dato ha una funzione di transizione parziale δ, è necessario trasformarlo in un automa con funzione di transizione $\overline{\delta}$ totale, aggiungendo il nuovo stato q_3 e le transizioni $\overline{\delta}(q_0, b) = q_3$, $\overline{\delta}(q_1, a) = q_3$, $\overline{\delta}(q_2, b) = q_3$. Per tutte le altre transizioni, $\overline{\delta} = \delta$.

Quindi, si procede come previsto dalla dimostrazione del teorema e si ottiene così l'automa seguente.

- Si introduce il nuovo stato finale d.
- Si pone:

$$\begin{aligned}
\overline{\delta}(q_0, a) &= \delta(q_0, a) \\
\overline{\delta}(q_0, c) &= \delta(q_0, c) \\
\overline{\delta}(q_1, b) &= \delta(q_1, b) \\
\overline{\delta}(q_1, c) &= \delta(q_1, c) \\
\overline{\delta}(q_2, a) &= \delta(q_2, a) \\
\overline{\delta}(q_2, c) &= \delta(q_2, c).
\end{aligned}$$

- Si pone:

$$\begin{aligned}
\overline{\delta}(q_0, b) &= q_3 \\
\overline{\delta}(q_1, a) &= q_3 \\
\overline{\delta}(q_2, b) &= q_3.
\end{aligned}$$

- Si pone

$$\begin{aligned}
\overline{\delta}(d, a) &= q_3 \\
\overline{\delta}(d, b) &= q_3
\end{aligned}$$

Aggiungiamo uno stato di errore q_3, anche detto **stato trappola**, e definiamo le transizioni mancanti verso di esso.

Hai già segnalato queste transizioni mancanti:

- $\delta(q_0, b) = q_3$
- $\delta(q_1, a) = q_3$
- $\delta(q_2, b) = q_3$

Inoltre, una funzione totale richiede che **da q_3 tutte le transizioni puntino a sé stesso** (cioè q_3 è assorbente):

- $\delta(q_3, a) = q_3$
- $\delta(q_3, b) = q_3$
- $\delta(q_3, c) = q_3$

(se l'alfabeto è $\{a, b, c\}$)

Le transizioni già esistenti dell'automa sono mantenute:

- $\delta(q_0, a) = \ldots$
- $\delta(q_0, c) = \ldots$
- $\delta(q_1, b) = \ldots$
- $\delta(q_1, c) = \ldots$
- $\delta(q_2, a) = \ldots$
- $\delta(q_2, c) = \ldots$

> (Qui "..." indica che si mantengono i valori originali dell'automa di partenza.)

3. Invertire gli stati finali:

Ora che l'automa è **completo e deterministico**, possiamo definire il complemento semplicemente **scambiando** gli stati finali:

- Gli **stati finali del nuovo automa** saranno:

$$F' = Q \setminus F$$

  (cioè tutti gli stati che **non** erano finali nel DFA originale)

Adesso che abbiamo costruito l'automa che riconosce il complemento di un linguaggio regolare, dobbiamo effettivamente dimostrare che questo riconosca il linguaggio complementare a quello dato, e quindi abbiamo bisogno di dimostrare che:

🔹 Passo 2: Dimostrazione della doppia inclusione

Abbiamo bisogno di dimostrare che:

$$L(A_2) = \Sigma^* \setminus L(A_1)$$

Inclusione 1: $L(A_2) \subseteq \Sigma^* \setminus L(A_1)$

Se $w \in L(A_2)$, allora w viene accettata dall'automa A_2. Poiché A_2 è stato costruito in modo che gli stati finali di A_2 siano quelli non finali di A_1, questo significa che w non è accettata da A_1. Quindi $w \in \Sigma^* \setminus L(A_1)$.

Inclusione 2: $\Sigma^* \setminus L(A_1) \subseteq L(A_2)$

Se $w \in \Sigma^* \setminus L(A_1)$, allora w non è accettata da A_1. Poiché A_2 accetta le stringhe che non sono accettate da A_1, w deve essere accettata da A_2. Quindi $w \in L(A_2)$.

🟩 Conclusione:

Poiché abbiamo dimostrato entrambe le inclusioni:

$$L(A_2) = \Sigma^* \setminus L(A_1)$$

E quindi, il complemento di un linguaggio regolare è anch'esso regolare.
La classe $\mathcal{R}$ dei linguaggi regolari è **chiusa rispetto al complemento.**

▣

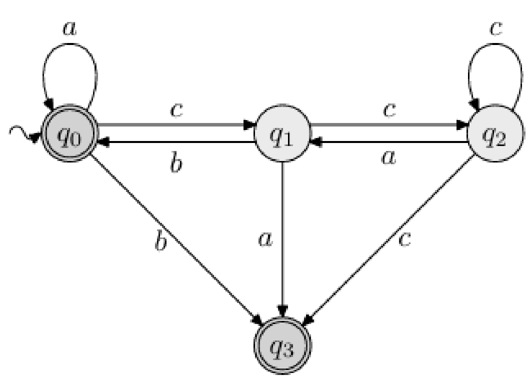

FIGURA 3.20 Automa che accetta il complemento del linguaggio accettato dall'automa di Figura 3.19.

Chiusura rispetto all'intersezione

In conseguenza dei due risultati precedenti, possiamo mostrare immediatamente che i linguaggi regolari sono chiusi anche rispetto all'intersezione.

Teorema 3.8 *Dati due linguaggi regolari L_1 e L_2, la loro intersezione $L = L_1 \cap L_2$ è un linguaggio regolare.*

Dimostrazione. 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 rispetto alla concatenazione

Teorema 3.9 *Dati due linguaggi regolari L_1 e L_2, la loro concatenazione $L = L_1 \circ L_2$ è un linguaggio regolare.*

Dimostrazione. Si considerino gli automi deterministici

$$\begin{aligned}\mathcal{A}_1 &= \langle \Sigma_1, Q_1, \delta_1, q_{01}, F_1 \rangle \\ \mathcal{A}_2 &= \langle \Sigma_2, Q_2, \delta_2, q_{02}, F_2 \rangle,\end{aligned}$$

che riconoscono, rispettivamente, i linguaggi $L_1 = L(\mathcal{A}_1)$ ed $L_2 = L(\mathcal{A}_2)$. Sia $\mathcal{A} = \langle \Sigma, Q, \delta_N, q_0, F \rangle$ un automa non deterministico tale che:

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

$$\begin{aligned}\delta_N(q, a) &= \delta_1(q, a), \forall q \in Q_1 - F_1, a \in \Sigma_1 \\ \delta_N(q, a) &= \delta_1(q, a) \cup \delta_2(q_{02}, a), \forall q \in F_1, a \in \Sigma \\ \delta_N(q, a) &= \delta_2(q, a), \forall q \in Q_2, a \in \Sigma_2.\end{aligned}$$

Per costruzione, l'automa $\mathcal{A}$ accetta tutte e sole le stringhe di $L = L_1 \circ L_2$. $\square$

Esempio 3.17 Si consideri l'automa $\mathcal{A}_1$ definito da

$$\mathcal{A}_1 = \langle \{a, b\}, \{q_0, q_1\}, \delta_1, q_0, \{q_1\} \rangle,$$

ed il cui grafo di transizione è quello di Figura 3.16; si consideri inoltre l'automa $\mathcal{A}_2$ definito da

$$\mathcal{A}_2 = \langle \{a, b\}, \{q_2, q_3\}, \delta_2, q_2, \{q_2\} \rangle,$$

ed il cui grafo di transizione è quello di Figura 3.17. Per costruire l'automa $\mathcal{A}$ che effettua la concatenazione $L(\mathcal{A}_2) \circ L(\mathcal{A}_1)$ si procede come segue.

- Si pone $\Sigma = \{a, b\}$.
- Si pone $Q = \{q_0, q_1, q_2, q_3\}$.
- Si sceglie q_2 come stato iniziale.
- Si pone $F = \{q_1\}$ (perché $\mathcal{A}_1$ non accetta la stringa vuota).
- La funzione di transizione si ricava come segue. Per gli stati appartenenti a $Q_2 - F_2$ si ha:

$$\delta_N(q_3, a) = \delta_2(q_3, a).$$

Per gli stati appartenenti ad F_2 si ha:

$$\begin{aligned}\delta_N(q_2, a) &= \emptyset \cup \{\delta_1(q_0, a)\}\\ \delta_N(q_2, b) &= \{\delta_2(q_2, b)\} \cup \{\delta_1(q_0, b)\}.\end{aligned}$$

Per gli stati appartenenti a Q_1 si ha:

$$\begin{aligned}\delta_N(q_0, a) &= \delta_1(q_0, a)\\ \delta_N(q_0, b) &= \delta_1(q_0, b)\\ \delta_N(q_1, a) &= \delta_1(q_1, a).\end{aligned}$$

Il grafo di transizione dell'automa $\mathcal{A}$ è riportato in Figura 3.21.

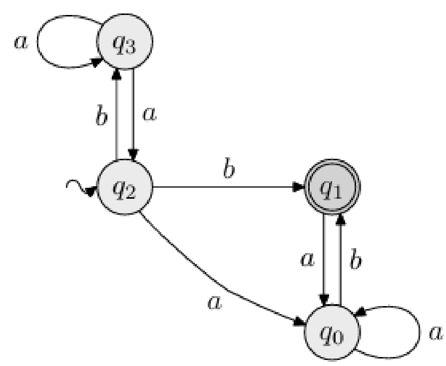

FIGURA 3.21 Automa che accetta la concatenazione dei linguaggi accettati dagli automi di Figura 3.16 e Figura 3.17.

Chiusura rispetto alla star di Kleene

Teorema 3.10 *Dato un linguaggio regolare L, anche L^* è un linguaggio regolare.*

Dimostrazione. Per dimostrare questa proprietà, si consideri l'automa deterministico

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

che riconosce $L = L(\mathcal{A})$. A partire da questo, deriviamo un automa $\mathcal{A}' = \langle \Sigma, Q \cup \{q'_0\}, \delta', q'_0, F \cup \{q'_0\} \rangle$, che riconosce $L^* = (L(\mathcal{A}))^*$, ponendo

$$\begin{aligned}\delta'(q, a) &= \delta(q, a),\ \forall q \in Q - F\\ \delta'(q, a) &= \delta(q, a) \cup \delta(q_0, a),\ \forall q \in F\\ \delta'(q'_0, a) &= \delta(q_0, a).\end{aligned}$$

Esempio 3.18 Si consideri l'automa $\mathcal{A}$ di Figura 3.22.
Per costruire l'automa che riconosce l'iterazione di $L(\mathcal{A})$ si procede come segue.

- Si pone $\Sigma' = \Sigma$.
- Si pone $Q' = Q \cup q_0'$, dove q_0' è il nuovo stato iniziale.
- Si pone $F' = F \cup q_0'$.
- La funzione di transizione è definita al seguente modo. Per gli stati appartenenti a $Q - F$ si ha:

$$\begin{array}{rcl} \delta'(q_0, b) & = & \delta(q_0, b) \\ \delta'(q_1, a) & = & \delta(q_1, a) \\ \delta'(q_1, b) & = & \delta(q_1, b). \end{array}$$

  Per gli stati appartenenti ad F si ha:

$$\begin{array}{rcl} \delta'(q_2, a) & = & \delta(q_2, a) \\ \delta'(q_2, b) & = & \delta(q_0, b) \\ \delta'(q_2, c) & = & \{\delta(q_2, c)\} \cup \{\delta(q_0, c)\}. \end{array}$$

  Inoltre si ha

$$\begin{array}{rcl} \delta'(q_0', b) & = & \delta(q_0, b) \\ \delta'(q_0', c) & = & \delta(q_0, c). \end{array}$$

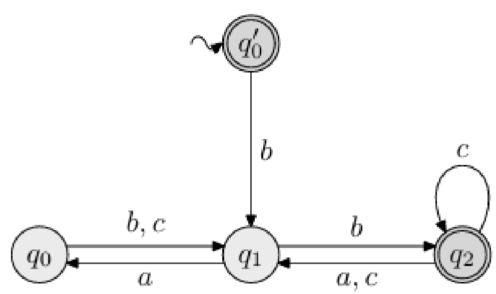

FIGURA 3.23 Iterazione dell'automa di Figura 3.22.