Azzolini Riccardo 2020-11-20

Chiusura dei linguaggi regolari rispetto alle operazioni booleane

1 Chiusura rispetto all'unione

Teorema: Siano L_1 e L_2 linguaggi regolari. Allora, $L_1 \cup L_2$ è un linguaggio regolare.

Questa proprietà, così come le altre proprietà di questo tipo, può essere dimostrata usando uno qualunque dei formalismi che caratterizzano la classe dei linguaggi regolari: i DFA, gli NFA, gli ϵ-NFA o le espressioni regolari. In questo caso, il formalismo più comodo è quello delle espressioni regolari.

Siccome L_1 è regolare, esiste un'espressione regolare E_1 tale che $L(E_1) = L_1$. Analogamente, poiché L_2 è regolare, esiste un'espressione regolare E_2 tale che $L(E_2) = L_2$. Si può quindi costruire l'espressione regolare $E_1 + E_2$, che per la semantica dell'operatore $+$ genera il linguaggio regolare

$$L(E_1 + E_2) = L(E_1) \cup L(E_2) = L_1 \cup L_2$$

ovvero $L_1 \cup L_2$ è un linguaggio regolare.

2 Chiusura rispetto al complemento

Dato un linguaggio $L \subseteq \Sigma^*$, il **complemento** di L è il linguaggio $\overline{L} = \Sigma^* \setminus L$.

Teorema: Se L è un linguaggio regolare, allora $\overline{L}$ è un linguaggio regolare.

Questa volta, la dimostrazione viene fatta utilizzando la caratterizzazione dei linguaggi regolari tramite i DFA. Per definizione, se L è regolare esiste un DFA $A = \langle Q, \Sigma, \delta, q_0, F \rangle$ tale che $L(A) = L$. Si costruisce poi un altro automa $\overline{A} = \langle Q, \Sigma, \delta, q_0, Q \setminus F \rangle$, che ha tutti gli stessi elementi di A, ad eccezione l'insieme degli stati finali, che in $\overline{A}$ contiene tutti e soli gli stati non finali di A.

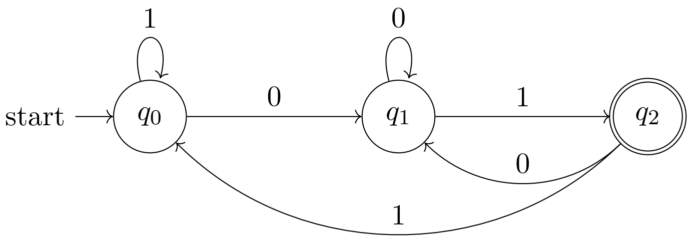
Data una stringa $w \in L$, si deduce che

$$\begin{aligned}
w \in L &\iff w \in L(A) && \text{[per definizione di } A\text{]} \\
&\iff \hat{\delta}(q_0, w) \in F && \text{[per definizione di accettazione su un DFA]} \\
&\iff \hat{\delta}(q_0, w) \notin (Q \setminus F) && \text{[per le proprietà di } \setminus\text{]} \\
&\iff w \notin L(\overline{A}) && \text{[per definizione di accettazione su un DFA]}
\end{aligned}$$

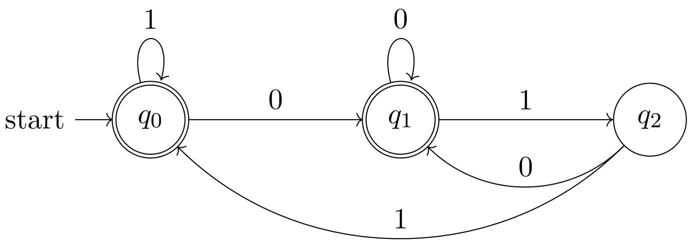
cioè che una stringa appartiene a L se e solo se non appartiene a $L(\overline{A})$: questo vuol dire che $L(\overline{A}) = \overline{L}$. Così, si conclude che $\overline{L}$ è un linguaggio regolare, in quanto riconosciuto da un DFA.

2.1 Esempio

Sia L il linguaggio delle stringhe su $\{0,1\}$ che terminano per 01, cioè il linguaggio generato dall'espressione regolare $(0+1)^*01$. Intuitivamente, L è riconosciuto dal seguente DFA A:



$\overline{L}$ è il linguaggio delle stringhe su $\{0,1\}$ che *non* terminano per 01, generato dall'espressione regolare $\epsilon + 1 + (0+1)^*(0+11)$. Per la costruzione vista prima, l'automa $\overline{A}$ che riconosce $\overline{L}$ è ottenuto a partire dall'automa A che riconosce L, semplicemente scambiando i ruoli di stati finali e non finali:



2.2 Uso per dimostrare che un linguaggio non è regolare

Le proprietà dei linguaggi regolari appena viste, così come quelle che verranno presentate successivamente, forniscono ulteriori strumenti per dimostrare che un linguaggio non è regolare.

Ad esempio, il linguaggio

$$L_{eq} = \{w \in \{0,1\}^* \mid w \text{ contiene lo stesso numero di 0 e 1}\}$$

non è regolare, e lo si può dimostrare con la tecnica basata sul pumping lemma, ragionando sulla stringa $0^m 1^m$, dove $m \geq N$ e N è la costante di pumping (supponendo per assurdo che L_{eq} sia regolare). Il ragionamento è uguale a quello svolto nella dimostrazione del fatto che il linguaggio $\{0^n 1^n \mid n \geq 1\}$ non è regolare.

Si consideri ora il linguaggio

$$L_{neq} = \{w \in \{0,1\}^* \mid w \text{ contiene un numero diverso di 0 e 1}\}$$

I due linguaggi L_{eq} e L_{neq} sono l'uno il complemento dell'altro, quindi in particolare si ha che $\overline{L_{neq}} = L_{eq}$. Allora, se L_{neq} fosse regolare, per la chiusura dei linguaggi regolari rispetto al complemento dovrebbe essere regolare anche L_{eq}, contrariamente a quanto appena dimostrato. Di conseguenza, L_{neq} non è regolare.

3 Chiusura rispetto all'intersezione

Teorema: Siano L e M due linguaggi regolari. Allora, il linguaggio $L \cap M$ è regolare.

Questo teorema può essere dimostrato semplicemente sfruttando una delle leggi di De Morgan, che permette di definire l'intersezione a partire dalle operazioni di unione e complemento,

$$L \cap M = \overline{\overline{L} \cup \overline{M}}$$

e i risultati precedenti, secondo i quali $\overline{\overline{L} \cup \overline{M}}$ è regolare.

In alternativa, si potrebbe dare una dimostrazione costruttiva sugli automi, mostrando come costruire un automa che riconosca $L \cap M$ a partire dagli automi che riconoscono L e M.

4 Chiusura rispetto alla differenza insiemistica

Teorema: Siano L e M due linguaggi regolari. Allora, il linguaggio $L \setminus M$ è regolare.

Anche in questo caso, la dimostrazione si basa sul fatto che l'operatore di differenza insiemistica può essere definito a partire dall'intersezione e dal complemento,

$$L \setminus M = L \cap \overline{M}$$

dove $L \cap \overline{M}$ è un linguaggio che si è già dimostrato essere regolare.