

Esempio

Dato un alfabeto $\Sigma = \{e, B\}$, l'insieme $L_3 = \{e^n B^n \mid n \geq 1\}$ è un linguaggio e' composto da tutte le stringhe contenute nelle concatenazioni di un certo numero di e , seguito dalla concatenazione dello stesso numero di B .

$$e^0 B^0 = \epsilon \notin L_3 \quad e^2 B^2 = eBBe \rightarrow e^2 B^2 = eeBB$$

$$e^3 B^3 = eee BBB$$

Intersezione: $L_1 \cap L_2 = \{x \in \Sigma^* \mid x \in L_1 \wedge x \in L_2\}$

Unione: $L_1 \cup L_2 = \{x \in \Sigma^* \mid x \in L_1 \vee x \in L_2\}$

Complemento: $\overline{L_1} = \{x \in \Sigma^* \mid x \notin L_1\}$

Definizione

Le potenze L^h di un linguaggi sono definite come $L^h = L^1 \circ L^{h-1} \dots \circ L^1$ con la convenzione che $L^0 = \{\epsilon\}$

$$L \subseteq \Sigma \text{ con } h \geq 0$$

Osservazione

L'insieme delle stringhe di lunghezza h sull'alfabeto Σ è lo insieme con Σ^h

Definizione \rightarrow stelle di Kleene

Il linguaggio L^* viene definito come $L^* = \bigcup_{h=0}^{\infty} L^h$

$$L^* = L^1 \cup L^2 \cup \dots \cup L^{\infty}$$

L^* prende il nome di chiusura relativa del linguaggio L rispetto all'operazione di concatenazione.

notiamo che:

$$L, \epsilon \in L^*$$

$$\downarrow \\ \text{perché } L^0 = \{\epsilon\}$$

$$\Delta^1 = \{\epsilon\}$$

$$\Delta^0 = \{\epsilon\}$$

Esempio

$$L = \{ ee \} \Rightarrow L^* = \{ e^{2m} \mid m \geq 0 \}$$

Poiché

$$L = \bigcup_{h=0}^{\infty} L^h = \bigcup_{h=0}^{\infty} \{ ee \}^h = \bigcup_{h=0}^{\infty} \{ e^2 \}^h = \bigcup_{h=0}^{\infty} \{ e^{2h} \}$$

Esempio

$$L_1 = \{ BiS \} \quad L_2 = \{ NonNo \}$$

$$\text{ottenere } L_1^* L_2 = \bigcup_{h=0}^{\infty} L_1^h \cdot L_2 = \{ NonNo, BiSNonNo, BiSBiSNonNo, \dots \}$$

il simbolo di

compostezione

più veree non scritte

Definizione

Si dice con L^+ le chiusure (non riflessive) definite da

$$L^+ = \bigcup_{h=1}^{\infty} L^h$$

Dato queste definizione siamo dire che: $L^* = L^+ \cup \{ \epsilon \}$

Osservazioni

- $\epsilon \in L^*$ per definizione
- $\epsilon \in L^+$ solo se $\epsilon \in L$
- $\perp^+ = \perp$
- $L^* = L^+ \cup \{ \epsilon \}$ ma $L^+ \neq L^* \setminus \{ \epsilon \}$ poiché se $\epsilon \in L^+$ se ci fesse l'equivalenza allora anche $L^* \setminus \{ \epsilon \}$ dovrebbe contenere ϵ ma per costruzione $\epsilon \notin L^* \setminus \{ \epsilon \}$

$$\Sigma^* = \Sigma \cup \{ \epsilon \}$$

Esempio:

$$\text{Se } L = \{\text{ab, ba}\}$$

$$L^2 = L^1 \circ L^{2-1} = L^1 \circ L^1 = \{\text{aabb, babb, bbab, ab ba}\}$$

$$L^1 = L^{1-1} \circ L = \{\epsilon\} L^1 = L^1$$

$$\text{FORMULA } L^h = L^{h-1} \circ L^1$$

Proverete con L^3

$$L^3 = L^{3-1} \cdot L^1 =$$

$$L^* = \{\epsilon\}^* = \{\epsilon\}^0 \cup \{\epsilon\}^1 \cup \{\epsilon\}^2 \dots = \{\epsilon\}$$

↑
caso finito di L^*

Linguaggi regolari

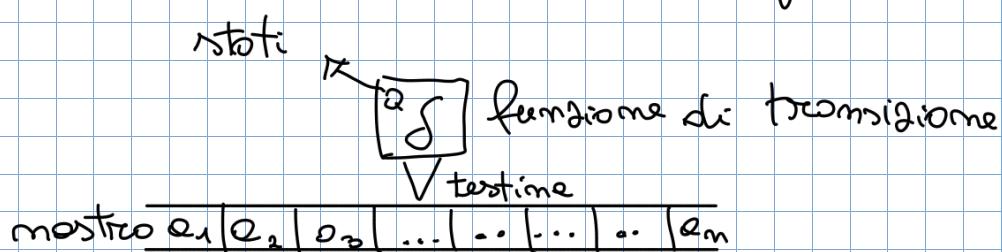
I metodi per rappresentare in modo finito i linguaggi:

- **Riconoscitivi**: metodi che prendono in input una stringa e ci dicono se è appartenente al linguaggio oppure no.
- **Generativi**: metodi che partono da un simbolo iniziale e un insieme finito di regole e da lì si generano tutte le stringhe del linguaggio dette GRAMATICHE

Riconoscitore: strumento che riconosce insiemi di stringhe.
Dato avere un mostro su cui prendere in input una stringa e ci può riconoscere se è. Esso è diviso in celle con un simbolo dell'alfabeto Σ di partenza. Il simbolo di BLANK indica la mancanza di un elemento nelle celle \emptyset . Il mostro è infinito in entrambe le direzioni, c'è la testina di lettura e/o scrittura che lavora su una cella del mostro alle volte e si muove avanti e indietro una cella alle volte. Il riconoscitore può essere definito tramite gli stati che cambiamo nel tempo. Un riconoscitore ha una serie di regole che lo guidano nei movimenti e negli stati. tramite le funzione di transizione del riconoscitore. Ad ogni momento delle computazioni soltanto un numero finito di celle contengono simboli diversi dal simbolo BLANK. In questo caso potremmo di mostro infinito perché le celle vuote ci potrebbe servire durante la computazione e non sono occupate le celle del mostro perché non deve

potere sostenere strimpe di qualsiasi lunghezza
 Un punto è uno spostamento delle testine, un eventuale cambio di stato, una lettura o un'eventuale scrittura.
 Una configurazione è una "fotografia" del riconoscitore
 e determina un'istanza della computazione. Queste configurazioni dipende da una successione di configurazioni che porta da una configurazione "particolare" ad eventualmente termini in un'oltre

le configurazioni termine ↗ di accettazione se le strimpe è al linguaggio
di rifiuto se le strimpe è al linguaggio



TipoLOGIE di dispositivo per il riconoscimento di linguaggi
Automi e stati finiti (ASF)

Definizione: un Autome o stato finiti deterministico (ASD) è una quintupla $A = \langle \Sigma, Q, \delta, q_0, F \rangle$ dove
 Σ è l'alfabeto di input

$Q = \{q_0, \dots, q_m\}$ è un insieme finito di stati

$F \subseteq Q$ è un insieme di stati finali,

$q_0 \in Q$ è lo stato iniziale

$\delta: Q \times \Sigma \rightarrow Q$ è una funzione di transizione che ad ogni coppia di (stato, carattere in input) associa lo stato successivo.

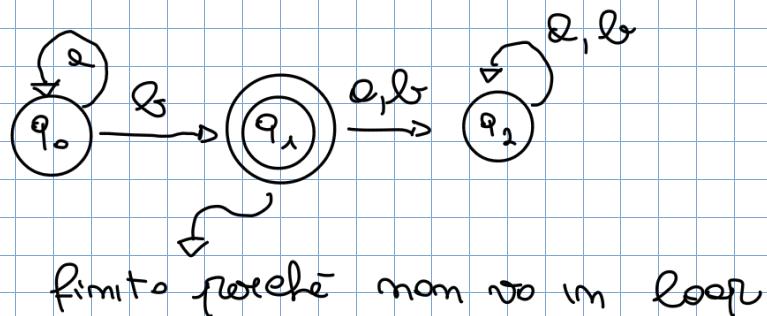
$\delta: Q \cdot \Sigma \rightarrow Q$ tabella di transizione che cui risponde
ogni uno gli stati, alle colonne i caratteri un input e gli
elementi rappresentano il risultato dell'applicazione
delle δ allo stato identificato dalle righe e del carattere
associato alle colonne delle tabelle

S	a	b
q_0	q_0	q_1
q_1	q_2	q_2
q_2	q_2	q_2

$$\begin{aligned}\delta: Q \cdot \Sigma &\rightarrow Q \\ (q_0, a) &\rightarrow q_0 \\ (q_1, b) &\rightarrow q_2\end{aligned}$$

Altre rappresentazione

è il
prolo di transizione in cui l'outcome è rappresentato
mediante un prolo orientato in cui i nodi rappresentano
gli stati, mentre gli archi rappresentano le
transizioni e sono etichettati con il carattere le cui
lettere determinano le transizioni. Gli stati finali
sono rappresentati da nodi con doppio cerchio mentre
quelli iniziali sono individuati tramite une frecce



l'outcome ad ogni passo offre il carattere successivo
delle strisce e applica la funzione di transizione

terminata le lettura delle strisce, essa viene accettata se lo stato finale è uno stato finale (che appartiene all'insieme F) altrimenti viene rifiutata.

N.B.

Si nota che un AFN termina sempre le sue computazioni.

Definizione (Pag 43)

Dato un automa e stati finiti $A = \langle \Sigma, Q, \delta, q_0, F \rangle$ una configurazione di A è una coppia $\langle q, x \rangle$ con $q \in Q$ e $x \in \Sigma^*$.

Definizione: Una configurazione $\langle q, x \rangle$ con $q \in Q$ e $x \in \Sigma^*$

di A è detta:

- iniziale se $q = q_0$
- finale se $x = \epsilon$
- accettante se $x = \epsilon$ e $q \in F$

La funzione di transizione permette di definire le relazioni di transizione che indichiamo con \vdash
tra configurazioni.

Queste relazioni sono od una configurazione, le configurazioni successive, nel seguente modo:

Definizione: Dato un AFN $A = \langle \Sigma, Q, \delta, q_0, F \rangle$ e due configurazioni (q, x) e (q', y) di A avremo che $(q, x) \vdash (q', y) \Leftrightarrow$ rispettano le due condizioni:

1) $\exists e \in \Sigma$ tale che $x = ey$

2) $\delta(q, e) = q'$

$\xrightarrow{\text{di base}}$

A è associata tramite la funzione di transizione dell'AFN A .

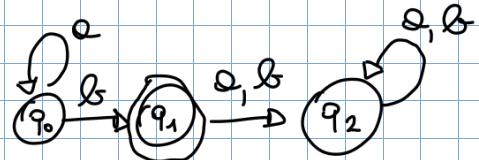
ASFD $A = \langle \Sigma, Q, \delta, q_0, F \rangle$ $x \in \Sigma^*$ è accettato da $A \Leftrightarrow (q_0, x) \vdash (q, \epsilon)$ con $q \in F$

e possiamo definire il linguaggio riconosciuto

$$L(A) = \{x \in \Sigma^* \mid (q_0, x) \stackrel{*}{\vdash}_A (q, \epsilon), q \in F\}$$

Esempio

stringhe eeb è accettata dall'ASFD \rightarrow



configurazione iniziale (q_0, eeb)

l'automa raffigura la configurazione di accettazione $^m(q_1, \epsilon)$

$$(q_0, eeb) \vdash (q_0, eb) \vdash (q_0, b) \vdash (q_1, \epsilon)$$

leggiamo e si rimane in q_0

leggiamo e si rimane in q_0

leggiamo b si andiamo in $q_1 \rightarrow$ finiamo