Equivalenza tra ASF, RG e RE

a.a. 2020-2021

Corso di Fondamenti di Informatica - 1 modulo

Corso di Laurea in Informatica

Università di Roma “Tor Vergata”

Prof. Giorgio Gambosi

Grammatiche regolari

ASF e grammatiche di tipo 3

Per ogni grammatica regolare $\mathscr{G} = \langle V_T, V_N, P, S \rangle$, esiste un ASFND $\mathscr{A}_N = \left\langle \Sigma, Q, \delta_N, q_0, F \right\rangle$ che riconosce il linguaggio che essa genera.

Viceversa, per ogni ASFND $\mathscr{A}_N$ esiste una grammatica regolare che genera il linguaggio che esso riconosce.

ASF e grammatiche di tipo 3

Sia $\mathcal{G} = \langle V_T, V_N, P, S \rangle$ una grammatica di tipo 3, con al più la sola ε-produzione $S \longrightarrow \varepsilon$.

Definiamo una procedura che partire da $\mathcal{G}$ produca un ASFND $\mathcal{A}_N = \left\langle \Sigma, Q, \delta_N, q_0, F \right\rangle$ equivalente (che accetta tutte e sole stringhe prodotte da $\mathcal{G}$).

Da $\mathscr{G}$ a $\mathscr{A}_N$

$$\Sigma = V_T$$

$$Q = \{q_I \mid I \in V_N\} \cup \{q_F\}$$

$$q_0 = q_S$$

$$F = \begin{cases} \{q_0, q_F\} & \text{se } S \longrightarrow \varepsilon \in P \\ \{q_F\} & \text{altrimenti} \end{cases}$$

Da $\mathscr{G}$ a $\mathscr{A}_N$

Per ogni coppia $a \in V_T$ e $B \in V_N$,

$$\delta_N(q_B, a) = \begin{cases} \{q_C \mid B \longrightarrow aC \in P\} \cup \{q_F\} & \text{se } B \longrightarrow a \in P \\ \{q_C \mid B \longrightarrow aC \in P\} & \text{altrimenti.} \end{cases}$$

L'automa è, in generale, non deterministico.

Da $\mathscr{G}$ a $\mathscr{A}_N$. Equivalenza di $\mathscr{G}$ e $\mathscr{A}_N$

Per dimostrare l'equivalenza tra $\mathscr{G}$ e $\mathscr{A}_N$, dobbiamo mostrare che per ogni $x \in \Sigma^*$ si ha che

$$S \overset{*}{\underset{\mathscr{G}}{\Longrightarrow}} x \quad \text{se e solo se} \quad \overline{\delta}_N(q_S, x) \cap F \neq \emptyset$$

Questo è chiaramente vero se $x = \varepsilon$, in quanto $\overline{\delta}_N(q_0, \varepsilon) = q_0 \in F$, se e solo se $S \longrightarrow \varepsilon \in P$, per costruzione.

Nel caso $x \in \Sigma^+$ mostriamo, per induzione sulla lunghezza di x, la proprietà più generale

$$S \overset{*}{\Longrightarrow} xZ \quad \text{se e solo se} \quad q_Z \in \overline{\delta}_N(q_S, x)$$

Da $\mathscr{G}$ a $\mathscr{A}_N$. Equivalenza di $\mathscr{G}$ e $\mathscr{A}_N$

Iniziamo da

$$S \overset{*}{\Longrightarrow} xZ \quad \text{implica} \quad q_Z \in \overline{\delta}_N(q_S, x)$$

Passo base: $|\, x \,| = 1$, per cui $x = a$, con $a \in \Sigma$. Allora abbiamo che $S \Longrightarrow aZ$ se e solo se $S \longrightarrow aZ \in P$ e quindi se e solo se, per costruzione dell'automa, $q_Z \in \delta_N(q_S, a)$.

Da $\mathcal{G}$ a $\mathcal{A}_N$. Equivalenza di $\mathcal{G}$ e $\mathcal{A}_N$

Passo induttivo: $|x| > 1$, per cui $x = ya$, con $|y| = n \geq 1$ e $a \in \Sigma$.

Per l'ipotesi induttiva il risultato si assume valido per y, quindi

$$S \stackrel{*}{\Longrightarrow} yZ \quad \text{se e solo se} \quad q_Z \in \overline{\delta}_N(q_S, y)$$

Osserviamo che $S \stackrel{*}{\Longrightarrow} xZ'$ se e solo se esiste $Z \in V_N$ tale che $S \stackrel{*}{\Longrightarrow} yZ \Longrightarrow yaZ' = xZ'$. Ne deriva che

- $q_Z \in \overline{\delta}_N(q_S, y)$ per induzione
- $Z \longrightarrow aZ' \in P$, e quindi $q_{Z'} \in \delta_N(a, Z)$ per costruzione

Quindi, $q_{Z'} \in \overline{\delta}_N(q_S, ya) = \overline{\delta}_N(q_S, x)$

Da $\mathscr{G}$ a $\mathscr{A}_N$. Equivalenza di $\mathscr{G}$ e $\mathscr{A}_N$

Abbiamo verificato che $S \stackrel{*}{\Longrightarrow} xZ$ se e solo se $q_Z \in \overline{\delta}_N(q_S, x)$.

Osserviamo ora che $S \stackrel{*}{\Longrightarrow} x$ se e solo se esistono $Z \in V_N$, $y \in \Sigma^*$ e $Z \longrightarrow a \in P$ tali che $x = ya$ e $S \stackrel{*}{\Longrightarrow} yZ \stackrel{*}{\Longrightarrow} ya = x$.

Da quanto visto sopra, ciò è vero se e solo se $q_Z \in \overline{\delta}_N(q_S, y)$ e $q_F \in \delta_N(q_Z, a)$, e quindi se e solo se $q_F \in \overline{\delta}_N(q_S, ya) = \overline{\delta}_N(q_S, x)$.

In conclusione, per ogni linguaggio regolare (generato da una grammatica di tipo 3) esiste un ASFND che lo accetta (e quindi anche un ASFD che lo decide).

ASF e grammatiche di tipo 3

Sia $\mathscr{A} = \left\langle \Sigma, Q, \delta, q_0, F \right\rangle$ un ASFD.

Definiamo una procedura che partire da $\mathscr{A}$ produca una grammatica di tipo 3 $\mathscr{G} = \langle V_T, V_N, P, S \rangle$ equivalente (che genera tutte e sole stringhe accettate da $\mathscr{A}$).

Da $\mathscr{A}$ a $\mathscr{G}$

Se $q_0 \notin F$:

$V_T = \Sigma$

$V_N = \{A_i \mid \text{ per ogni } q_i \in Q\}$

$S = A_0$

per ogni regola di transizione $\delta(q_i, a) = q_j$ esiste $A_i \longrightarrow aA_j \in P$, e se $q_j \in F$ esiste anche $A_i \longrightarrow a \in P$

Da $\mathscr{A}$ a $\mathscr{G}$

Se $q_0 \in F$:

$V_T = \Sigma$

$V_N = \{A_i \mid \text{ per ogni } q_i \in Q\} \cup \{A'_0\}$

$S = A'_0$

per ogni regola di transizione $\delta(q_i, a) = q_j$ esiste $A_i \longrightarrow aA_j \in P$, e se $q_j \in F$ esiste anche $A_i \longrightarrow a \in P$ (tutte le precedenti). Inoltre, per ogni $\delta(q_0, a) = q_j$ esiste $A'_0 \longrightarrow aA_j \in P$, e se $q_j \in F$ esiste anche $A'_0 \longrightarrow a \in P$ (A'_0 ha tutte le produzioni di A_0), infine, esiste $A'_0 \longrightarrow \varepsilon \in P$.

Da $\mathscr{A}$ a $\mathscr{G}$. Equivalenza di $\mathscr{G}$ e $\mathscr{A}$

Come prima, per dimostrare l'equivalenza tra $\mathscr{G}$ e $\mathscr{A}_N$, dobbiamo mostrare che per ogni $x \in \Sigma^*$ si ha che

$$\overline{\delta}(q_0, x) \in F \qquad \text{se e solo se} \qquad S \overset{*}{\underset{\mathscr{G}}{\Longrightarrow}} x$$

Questo è chiaramente vero se $x = \varepsilon$, in quanto in tal caso necessariamente $q_0 \in F$ e, per costruzione, l'assioma di $\mathscr{G}$ è A_0' e $A_0' \longrightarrow \varepsilon \in P$.

Nel caso $x \in \Sigma^+$ mostriamo, per induzione sulla lunghezza di x, entrambe le proprietàă

$$A_i \overset{*}{\Longrightarrow} xA_j \qquad \text{se e solo se} \qquad \overline{\delta}(q_i, x) = q_j$$

$$A_i \overset{*}{\Longrightarrow} x \qquad \text{se e solo se} \qquad \overline{\delta}(q_i, x) \in F$$

Da $\mathscr{A}$ a $\mathscr{G}$. Equivalenza di $\mathscr{G}$ e $\mathscr{A}$

Passo base: $\mid x \mid = 1$, ad esempio $x = a$. Abbiamo allora che

Per costruzione, $A_i \longrightarrow aA_j \in P$ (e quindi $A_i \Longrightarrow aA_j$) se e solo se $\delta(q_i, a) = q_j$ (e quindi $\overline{\delta}(q_i, a) = q_j$)

e inoltre che, per costruzione,

$A_i \longrightarrow a \in P$ (e quindi $A_i \Longrightarrow a$) se e solo se $q_j \in F$

Da $\mathscr{A}$ a $\mathscr{G}$. Equivalenza di $\mathscr{G}$ e $\mathscr{A}$

Passo induttivo: $\mid x \mid= n > 1$.

Sia $x = ya$, con $|y| = n - 1$: per l'ipotesi induttiva, la proprietà è valida per y, e quindi

$$A_i \stackrel{*}{\Longrightarrow} yA_k \quad \text{se e solo se} \quad \overline{\delta}(q_i, y) = q_k$$

Supponiamo $A_i \stackrel{*}{\Longrightarrow} xA_j = yaA_j$: ciò è possibile se e solo se esiste A_k tale che $A_i \stackrel{*}{\Longrightarrow} yA_k$ e $A_k \longrightarrow aA_j \in P$

Da $\mathscr{A}$ a $\mathscr{G}$. Equivalenza di $\mathscr{G}$ e $\mathscr{A}$

Per l'ipotesi induttiva, $A_i \overset{*}{\Longrightarrow} yA_k$ se e solo se $\overline{\delta}(q_i, y) = q_k$.

Per costruzione, $A_k \longrightarrow aA_j \in P$ se e solo se $\delta(q_k, a) = q_j$.

Ne consegue che

$$A_i \overset{*}{\Longrightarrow} yA_k \Longrightarrow yaA_j = xA_j$$

se e solo se

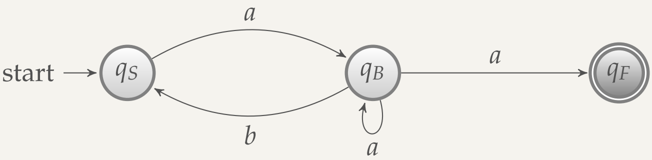
$$q_j = \delta(q_k, a) = \delta(\overline{\delta}(q_i, y), a) = \overline{\delta}(q_i, ya) = \overline{\delta}(q_i, x)$$

Esempio

Il linguaggio rappresentato da $a(a + ba)^*a$ è generato dalla grammatica

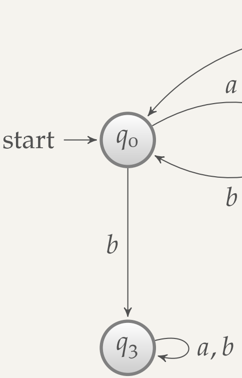
$$
\begin{aligned}
S &\longrightarrow aB \\
B &\longrightarrow aB \mid bS \mid a.
\end{aligned}
$$

ed è riconosciuto dall'ASFND



Esempio

A partire dall'ASFND è possibile derivare un ASFD equivalente



Esempio

E da questo una grammatica di tipo 3 equivalente a quella iniziale, dove $S = A_0$

$$\begin{array}{rcl} A_0 & \longrightarrow & aA_1 \\ A_1 & \longrightarrow & bA_0 \mid aA_2 \mid a \\ A_2 & \longrightarrow & aA_2 \mid bA_0 \mid a \end{array}$$

Per costruzione, questa grammatica ha, per ogni coppia $X \in V_N$ e $c \in V_T$, al più un $Y \in V_N$ tale che $X \longrightarrow cY \in P$.

Esercizio

Si consideri la grammatica regolare avente le seguenti produzioni:

$$\begin{aligned} S &\longrightarrow 0A \mid 1B \mid 0S \\ A &\longrightarrow aB \mid bA \mid a \\ B &\longrightarrow bA \mid aB \mid b. \end{aligned}$$

Si derivino un ASFND e un ASFD che riconoscono il linguaggio generato da tale grammatica. A partire dall'automa deterministico, derivare poi una grammatica di tipo 3 equivalente.

Espressioni regolari

Espressioni regolari

Teorema

Tutti i linguaggi definiti da espressioni regolari sono regolari.

Espressioni regolari e grammatiche di tipo 3

Teorema

Data una grammatica $\mathcal{G}$ di tipo 3, esiste una espressione regolare r tale che $L(\mathcal{G}) = \mathcal{L}(r)$, che descrive cioè il linguaggio generato da $\mathcal{G}$.

Espressioni regolari e grammatiche di tipo 3

Consideriamo una grammatica $\mathcal{G}$ di tipo 3 ed il linguaggio L da essa generato, che per semplicità assumiamo non contenga la stringa vuota ε.

Se così non fosse, applichiamo le considerazioni seguenti al linguaggio $L - \{\varepsilon\}$, anch'esso regolare: una volta derivata un'espressione regolare r che lo definisce, l'espressione regolare che definisce L sarà chiaramente $r + \varepsilon$.

Espressioni regolari e grammatiche di tipo 3

Alla grammatica $\mathscr{G}$ possiamo far corrispondere un sistema di equazioni su espressioni regolari.

Estensione del linguaggio delle espressioni regolari con variabili $A, \ldots, Z$, associando una variabile ad ogni non terminale in $\mathscr{G}$.

Tali variabili potranno assumere valori nell'insieme delle espressioni regolari.

Espressioni regolari e grammatiche di tipo 3

Raggruppamento di tutte le produzioni che presentano a sinistra lo stesso non terminale. Per ogni produzione del tipo

$$A \longrightarrow a_1 B_1 \mid a_2 B_2 \mid \ldots \mid a_n B_n \mid b_1 \mid \ldots \mid b_m$$

equazione del tipo

$$A = a_1 B_1 + a_2 B_2 + \ldots + a_n B_n + b_1 + \ldots + b_m.$$

Espressioni regolari e grammatiche di tipo 3

Da una grammatica regolare si ottiene un sistema di *equazioni lineari destre*, in cui ogni monomio contiene una variabile a destra di simboli terminali.

Espressioni regolari e grammatiche di tipo 3

Risoluzione del sistema di equazioni su espressioni regolari estese:

individuazione dei valori (espressioni regolari normali, prive delle variabili che definiscono a loro volta espressioni regolari) che, una volta sostituiti alle variabili, soddisfano il sistema di equazioni.

Espressioni regolari e grammatiche di tipo 3

$$\begin{aligned} A &\longrightarrow aA \mid bB \\ B &\longrightarrow bB \mid c \end{aligned}$$

corrisponde al sistema di equazioni

$$\begin{cases} A = aA + bB \\ B = bB + c. \end{cases}$$

Espressioni regolari e grammatiche di tipo 3

Per risolvere il sistema è possibile utilizzare, le trasformazioni algebriche applicabili sulle operazioni di unione e concatenazione (distributività, fattorizzazione, ecc.), oltre alle seguenti due regole.

Espressioni regolari e grammatiche di tipo 3

Sostituzione di una variabile con un'espressione regolare estesa.

Con riferimento all'esempio precedente abbiamo

$$\begin{cases} A & = & aA + b(bB + c) = aA + bbB + bc \\ B & = & bB + c. \end{cases}$$

Espressioni regolari e grammatiche di tipo 3

Eliminazione della ricursione.

L'equazione $B = bB + c$ si risolve in $B = b^*c$. Infatti, sostituendo a destra e sinistra abbiamo

$$b^*c = b(b^*c) + c = b^+c + c = (b^+ + \varepsilon)c = b^*c.$$

Più in generale abbiamo che un'equazione del tipo

$$A = \alpha_1 A + \alpha_2 A + \ldots + \alpha_n A + \beta_1 + \beta_2 + \ldots + \beta_m$$

si risolve in

$$A = (\alpha_1 + \alpha_2 + \ldots + \alpha_n)^*(\beta_1 + \beta_2 + \ldots + \beta_m),$$

dove $\alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_m$ sono espressioni regolari estese.

Espressioni regolari e grammatiche di tipo 3

Grammatica regolare

$$
\begin{aligned}
A_0 &\longrightarrow aA_1 \mid a \\
A_1 &\longrightarrow bA_3 \mid bA_2 \\
A_2 &\longrightarrow aA_2 \mid bA_0 \mid b \\
A_3 &\longrightarrow bA_3 \mid aA_2.
\end{aligned}
$$

da cui si ottiene il seguente sistema lineare.

$$
\begin{cases}
A_0 = aA_1 + a \\
A_1 = bA_3 + bA_2 \\
A_2 = aA_2 + bA_0 + b \\
A_3 = bA_3 + aA_2
\end{cases}
$$

Espressioni regolari e grammatiche di tipo 3

per eliminazione della ricursione su A_3:

$$\begin{cases} A_0 & = & aA_1 + a \\ A_1 & = & bA_3 + bA_2 \\ A_2 & = & aA_2 + bA_0 + b \\ A_3 & = & b^*aA_2 \end{cases}$$

per eliminazione della ricursione su A_2:

$$\begin{cases} A_0 & = & aA_1 + a \\ A_1 & = & bA_3 + bA_2 \\ A_2 & = & a^*(bA_0 + b) \\ A_3 & = & b^*aA_2 \end{cases}$$

Espressioni regolari e grammatiche di tipo 3

per sostituzione di A_2 nell'equazione relativa ad A_3

$$\begin{cases} A_0 & = & aA_1 + a \\ A_1 & = & bA_3 + bA_2 \\ A_2 & = & a^*(bA_0 + b) \\ A_3 & = & b^*aa^*(bA_0 + b) \end{cases}$$

per sostituzione di A_2 e A_3 nell'equazione relativa ad A_1

$$\begin{cases} A_0 & = & aA_1 + a \\ A_1 & = & b(b^*aa^*(bA_0 + b)) + b(a^*(bA_0 + b)) \\ A_2 & = & a^*(bA_0 + b) \\ A_3 & = & b^*aa^*(bA_0 + b) \end{cases}$$

per sostituzione di A e A nell'equazione relativa ad A

Espressioni regolari e grammatiche di tipo 3

per fattorizzazione nell'equazione relativa ad A_1:

$$\begin{cases} A_0 & = & aA_1 + a \\ A_1 & = & b(b^*aa^* + a^*)(bA_0 + b) \\ A_2 & = & a^*(bA_0 + b) \\ A_3 & = & b^*aa^*(bA_0 + b) \end{cases}$$

per sostituzione di A_1 nell'equazione relativa ad A_0:

$$\begin{cases} A_0 & = & a(b(b^*aa^* + a^*)(bA_0 + b)) + a \\ A_1 & = & b(b^*aa^* + a^*)(bA_0 + b) \\ A_2 & = & a^*(bA_0 + b) \\ A_3 & = & b^*aa^*(bA_0 + b) \end{cases}$$

Espressioni regolari e grammatiche di tipo 3

per fattorizzazione nell'equazione relativa ad A_0:

$$
\begin{cases}
A_0 & = & ab(b^*aa^* + a^*)bA_0 + ab(b^*aa^* + a^*)b + a \\
A_1 & = & b(b^*aa^* + a^*)(bA_0 + b) \\
A_2 & = & a^*(bA_0 + b) \\
A_3 & = & b^*aa^*(bA_0 + b)
\end{cases}
$$

Espressioni regolari e grammatiche di tipo 3

$$
\begin{aligned}
S &\longrightarrow aS \mid bA \mid \varepsilon \\
A &\longrightarrow aA \mid bS \mid \varepsilon
\end{aligned}
$$

Espressioni regolari e grammatiche di tipo 3

Eliminazione della produzione $A \longrightarrow \varepsilon$:

$$\begin{array}{rcl} S & \longrightarrow & aS \mid bA \mid \varepsilon \mid b \\ A & \longrightarrow & aA \mid bS \mid a. \end{array}$$

Espressioni regolari e grammatiche di tipo 3

$$
\begin{aligned}
S &= aS + bA + b + \varepsilon \\
A &= aA + bS + a
\end{aligned}
$$

Espressioni regolari e grammatiche di tipo 3

$$
\begin{aligned}
S &= aS + bA + b + \varepsilon \\
A &= a^*(bS + a)
\end{aligned}
$$

Espressioni regolari e grammatiche di tipo 3

$$\begin{aligned} S &= aS + ba^*(bS + a) + b + \varepsilon \\ A &= a^*(bS + a) \end{aligned}$$

Espressioni regolari e grammatiche di tipo 3

$$
\begin{aligned}
S &= (a + ba^*b)S + ba^*a + b + \varepsilon \\
A &= a^*(bS + a)
\end{aligned}
$$

Espressioni regolari e grammatiche di tipo 3

$$
\begin{aligned}
S &= (a + ba^*b)^*(ba^*a + b + \varepsilon) \\
A &= a^*(bS + a)
\end{aligned}
$$

Esercizio

Si consideri la seguente grammatica:

$$\begin{array}{rcl} A & \longrightarrow & aB \mid bC \mid a \\ B & \longrightarrow & aA \mid bD \mid b \\ C & \longrightarrow & ab \mid aD \mid a \\ D & \longrightarrow & aC \mid bB \mid b \end{array}$$

che genera le stringhe contenenti un numero dispari di a o un numero dispari di b.

Si costruisca l'espressione regolare corrispondente.

Espressioni regolari e ASF

Dato un ASFD $\mathscr{A}$, esiste una espressione regolare r tale che $L(\mathscr{A}) = \mathscr{L}(r)$, che descrive cioè il linguaggio riconosciuto da $\mathscr{A}$.

Espressioni regolari e ASF

Sia $\mathscr{A} = \langle \Sigma, Q, \delta, q_0, F \rangle$ un ASFD e sia L il linguaggio da esso riconosciuto. Assumiamo $F = \{q_F\}$.

Sia $n = |Q|$ e sia $\langle q_0, \ldots, q_{n-1} \rangle$ un qualunque ordinamento degli stati tale che $q_{n-1} = q_F$.

Espressioni regolari e ASF

Definiamo ora come

$$R_{ij}^{k} \qquad 0 \leq i, j \leq n-1; k \geq \max(i, j)$$

l'insieme delle stringhe tali da portare $\mathcal{A}$ da q_i a q_j senza transitare per nessuno stato q_h con $h \geq k$.

Abbiamo cioè che $x = a_1, \ldots, a_m \in R_{ij}^{k}$ se e solo se:

1. $\overline{\delta}(q_i, x) = q_j$;
2. se $\overline{\delta}(q_i, a_1 \ldots a_l) = q_{i_l}$ allora $i_l < k$, per $1 \leq l \leq m-1$.

Per $k = 0$ si ha:

$$R_{ij}^{0} = \begin{cases} \bigcup\{a\} & \text{tali che } \delta(q_i, a) = q_j, \text{ se ne esiste almeno uno;} \\ \emptyset & \text{altrimenti.} \end{cases}$$

Espressioni regolari e ASF

Per $k > 0$, se $x \in R_{ij}^{k+1}$ è una stringa che conduce da q_i a q_j senza transitare per nessuno stato q_h con $h \geq k+1$, possono verificarsi due casi:

1. x conduce da q_i a q_j senza transitare per q_k, dal che deriva che $x \in R_{ij}^{k}$.
2. x conduce da q_i a q_j transitando per q_k

Espressioni regolari e ASF

Nel secondo caso la sequenza degli stati attraversati può essere divisa in varie sottosequenze:

1. una prima sequenza, da q_i a q_k senza transitare per nessuno stato q_h con $h > k$, la corrispondente sottostringa di x appartiene quindi a R^k_{ik};
2. $r \geq 0$ sequenze, ognuna delle quali inizia e termina in q_k senza transitare per nessuno stato q_h con $h \geq k$, le corrispondenti sottostringhe di x appartengono quindi ciascuna a R^k_{kk};
3. una sequenza finale, da q_k a q_j senza transitare per nessuno stato q_h con $h \geq k$, la corrispondente sottostringa di x appartiene quindi a R^k_{kj}.

In conseguenza, ne deriva la relazione

$$R^{k+1}_{ij} = R^k_{ij} \cup R^k_{ik} \circ (R^k_{kk})^* \circ R^k_{kj}$$

Espressioni regolari e ASF

Dalle osservazioni precedenti deriva che è possibile costruire tutti gli insiemi R_{ij}^{k} a partire da $k = 0$ e derivando poi man mano i successivi.

Osserviamo anche che $L = R_{0(n-1)}^{n}$

Ogni insieme di stringhe R_{ij}^k può essere descritto per mezzo di una opportuna espressione regolare r_{ij}^k, infatti abbiamo che, per $k = 0$,

$$r_{ij}^0 = \begin{cases} a_{i_1} + \ldots + a_{i_l} & \text{dove } \delta(q_i, a_{i_k}) = q_j, k = 1, \ldots, l; \\ \emptyset & \text{se } l = 0. \end{cases}$$

Espressioni regolari e ASF

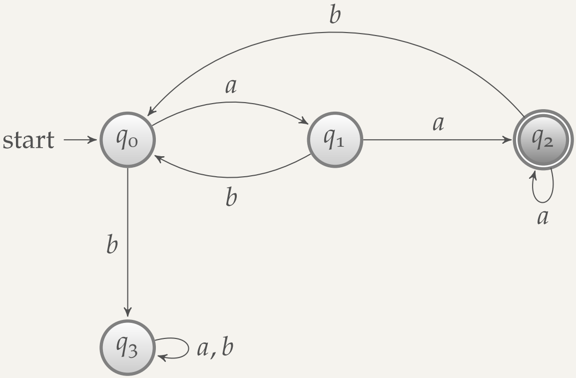
Per $k \geq 1$, abbiamo che, dalla relazione tra R_{ij}^{k+1}, R_{ik}^{k}, R_{kk}^{k} e R_{kj}^{k}, deriva che

$$r_{ij}^{k+1} = r_{ij}^{k} + r_{ik}^{k}(r_{kk}^{k})^{*}r_{kj}^{k}$$

Quindi, il linguaggio L sarà descritto dall'espressione regolare

$$r_{0(n-1)}^{n}$$

Espressioni regolari e ASF



Assumiamo l'ordinamento $q_1 = q_0, q_2 = q_1, q_3 = q_3, q_4 = q_2$ tra gli stati. Allora:

$$r^0_{00} = \emptyset;\ r^0_{01} = a;\ r^0_{02} = b;\ r^0_{03} = \emptyset;$$

$$r^0_{10} = b;\ r^0_{11} = \emptyset;\ r^0_{12} = \emptyset;\ r^0_{13} = a;$$

$$r^0_{20} = \emptyset;\ r^0_{21} = \emptyset;\ r^0_{22} = a + b;\ r^0_{23} = \emptyset;$$

$$r^0_{30} = b;\ r^0_{31} = \emptyset;\ r^0_{32} = \emptyset;\ r^0_{33} = a;$$

Espressioni regolari e ASF

$$r^1_{00} = r^0_{00} + r^0_{00}(r^0_{00})^* r^0_{00} = \emptyset + \emptyset(\emptyset)^*\emptyset = \emptyset$$
$$r^1_{01} = r^0_{01} + r^0_{00}(r^0_{00})^* r^0_{01} = a + \emptyset(\emptyset)^* a = a$$
$$r^1_{02} = r^0_{02} + r^0_{00}(r^0_{00})^* r^0_{02} = b + \emptyset(\emptyset)^* b = b$$
$$r^1_{03} = r^0_{03} + r^0_{00}(r^0_{00})^* r^0_{03} = \emptyset + \emptyset(\emptyset)^*\emptyset = \emptyset$$
$$r^1_{10} = r^0_{10} + r^0_{10}(r^0_{00})^* r^0_{00} = b + b(\emptyset)^*\emptyset = b$$
$$r^1_{11} = r^0_{11} + r^0_{10}(r^0_{00})^* r^0_{01} = \emptyset + b(\emptyset)^* a = ba$$
$$r^1_{12} = r^0_{12} + r^0_{10}(r^0_{00})^* r^0_{02} = \emptyset + b(\emptyset)^* b = bb$$
$$r^1_{13} = r^0_{13} + r^0_{10}(r^0_{00})^* r^0_{03} = a + b(\emptyset)^*\emptyset = a$$
$$\ldots$$

Espressioni regolari e ASF

$$r^1_{00} = \emptyset;\ r^1_{01} = a;\ r^1_{02} = b;\ r^1_{03} = \emptyset;$$

$$r^1_{10} = b;\ r^1_{11} = ba;\ r^1_{12} = bb;\ r^1_{13} = a;$$

$$r^1_{20} = \emptyset;\ r^1_{21} = \emptyset;\ r^1_{22} = a + b;\ r^1_{23} = \emptyset;$$

$$r^1_{30} = b;\ r^1_{31} = ba;\ r^1_{32} = bb;\ r^1_{33} = a;$$

Espressioni regolari e ASF

$$r^2_{00} = a(ba)^*b;\ r^2_{01} = a + a(ba)^*ba;\ r^2_{02} = b + a(ba)^*bb;\ r^2_{03} = a(ba)^*a;$$

$$r^2_{10} = b + ba(ba)^*b;\ r^2_{11} = ba + ba(ba)^*ba;\ r^2_{12} = bb + ba(ba)^*bb; r^2_{13} = a + ba(ba)^*a;$$

$$r^2_{20} = \emptyset;\ r^2_{21} = \emptyset;\ r^2_{22} = a + b;\ r^2_{23} = \emptyset;$$

$$r^2_{30} = b + ba(ba)^*b;\ r^2_{31} = ba + ba(ba)^*ba;\ r^2_{32} = bb + ba(ba)^*bb;\ r^2_{33} = a + ba(ba)^*a;$$

Il linguaggio accettato dall'automa sarà descritto dall'espressione regolare

$$r^4_{03}$$

State elimination

Procedura iterativa di eliminazione degli stati su un automa non deterministico generalizzato equivalente, in cui:

1. la funzione di transizione è definita su $Q \times E$, dove E è l'insieme delle espressioni regolari su Σ , per cui gli archi sono etichettati con e.r.
2. lo stato iniziale non ha archi entranti, per cui $\nexists q \in Q, e \in E : q_0 \in \delta_N(q, e)$
3. esiste un solo stato finale q_F senza archi uscenti, per cui $\nexists e \in E : \delta_N(q_F, e) \neq \emptyset$

State elimination

Dato un qualunque automa $\mathcal{A}$ non deterministico, un automa generalizzato $\mathcal{A}'$ equivalente può essere immediatamente ottenuto:

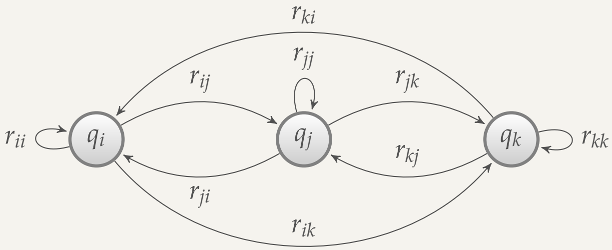
1. mantenendo gli stati di $\mathcal{A}$
2. introducendo, per ogni arco del grafo di transizione di $\mathcal{A}$ etichettato con l'insieme $a_1, \ldots, a_k$, un arco nel grafo di transizione di $\mathcal{A}'$ etichettato $a_1 + \ldots + a_k$
3. se lo stato iniziale q_0 di $\mathcal{A}$ ha archi entranti, introducendo in $\mathcal{A}'$ un nuovo stato iniziale $\overline{q}_0$ senza archi entranti, e la ε-transizione $\delta'_N(\overline{q}_0, \varepsilon) = \{q_0\}$
4. se esistono più stati finali in F, o se il solo stato finale ha archi uscenti, introducendo un ulteriore stato q_F, ponendo $F' = q_F$ e introducendo la ε-transizione $\delta'_N(q, \varepsilon) = \{q_F\}$ per ogni $q \in F$

State elimination

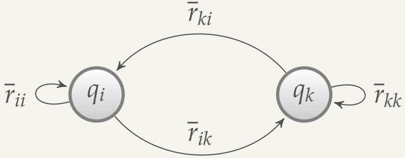
- Dato un automa nondeterministico (con ε-transizioni) $\mathscr{A}$ con insieme di stati Q, e dato uno stato q non iniziale né finale, è possibile ottenere un automa generalizzato equivalente $\mathscr{A}'$ con stati $Q - \{q\}$ effettuando una opportuna operazione di eliminazione dello stato
- L'eliminazione dello stato viene effettuata considerando tutti i possibili cammini di lunghezza 3 passanti per q (sequenze q_i, q, q_j per le quali esistono archi da q_i a q e da q a q_j)
- Per ogni cammino, le etichette degli archi interessati vengono modificate come mostrato di seguito
- Al termine, rimangono lo stato iniziale e quello finale, collegati da un arco, la cui etichetta fornisce l'espressione regolare cercata

State elimination

Da

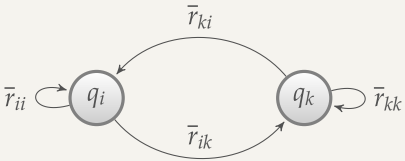


a



State elimination

Le espressioni regolari risultanti possono comunque essere complesse



$$\bar{r}_{ii} = r_{ij}(r_{jj} + r_{jk}r_{kk}^*r_{kj})^*r_{ji} + r_{ij}(r_{jj} + r_{jk}r_{kk}^*r_{kj})^*r_{jk}r_{kk}^*r_{ki} + r_{ik}(r_{kj}r_{jj}^*r_{jk} + r_{kk})^*r_{kj}r_{jj}^*r_{ji} + r_{ik}r_{ki}$$

$$\bar{r}_{kk} = r_{kj}(r_{jj} + r_{kj}r_{ii}^*r_{ij})^*r_{jk} + r_{kj}(r_{jj} + r_{ji}r_{ii}^*r_{ij})^*r_{ji}r_{ii}^*r_{ik} + r_{ki}(r_{ij}r_{jj}^*r_{ji} + r_{ii})^*r_{ij}r_{jj}^*r_{jk} + r_{ki}r_{ik}$$

$$\bar{r}_{ik} = r_{ik} + r_{ij}r_{jj}^*r_{jk}$$

$$\bar{r}_{ki} = r_{ki} + r_{kj}r_{jj}^*r_{jk}$$

In effetti, se esistono n cammini $q_iq_jq_h$ ($h = k_1, \ldots, k_n$), allora si ha che

$$\bar{r}_{ik} = r_{ik_1} + r_{ij}r_{jj}^*r_{jk_1} + r_{ik_2} + r_{ij}r_{jj}^*r_{jk_2} + \ldots + r_{ik_n} + r_{ij}r_{jj}^*r_{jk_n}$$

lo stesso, evidentemente, vale per $\bar{r}_{ki}$