Regular Expressions

Regular expressions describe regular languages. The grammar of regular expressions is:

$$
\begin{aligned}
S &\rightarrow \emptyset \\
S &\rightarrow \lambda \\
S &\rightarrow a \quad \text{for each } a \in \Sigma \\
S &\rightarrow S + S \\
S &\rightarrow S \cdot S \mid SS \\
S &\rightarrow S^* \\
S &\rightarrow (S)
\end{aligned}
$$

The language $\mathcal{L}(r)$ of an r.e. r is:

$$
\begin{aligned}
\mathcal{L}(\emptyset) &= \emptyset \\
\mathcal{L}(\lambda) &= \{\lambda\} \\
\mathcal{L}(a) &= \{a\} \quad \text{for each } a \in \Sigma \\
\mathcal{L}(r_1 + r_2) &= \mathcal{L}(r_1) + \mathcal{L}(r_2) \\
\mathcal{L}(r_1 r_2) &= \mathcal{L}(r_1)\mathcal{L}(r_2) \\
\mathcal{L}((r_1)) &= \mathcal{L}(r_1) \\
\mathcal{L}(r_1^*) &= \mathcal{L}(r_1)^*
\end{aligned}
$$

Examples

$a^*b^*a^*$

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

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

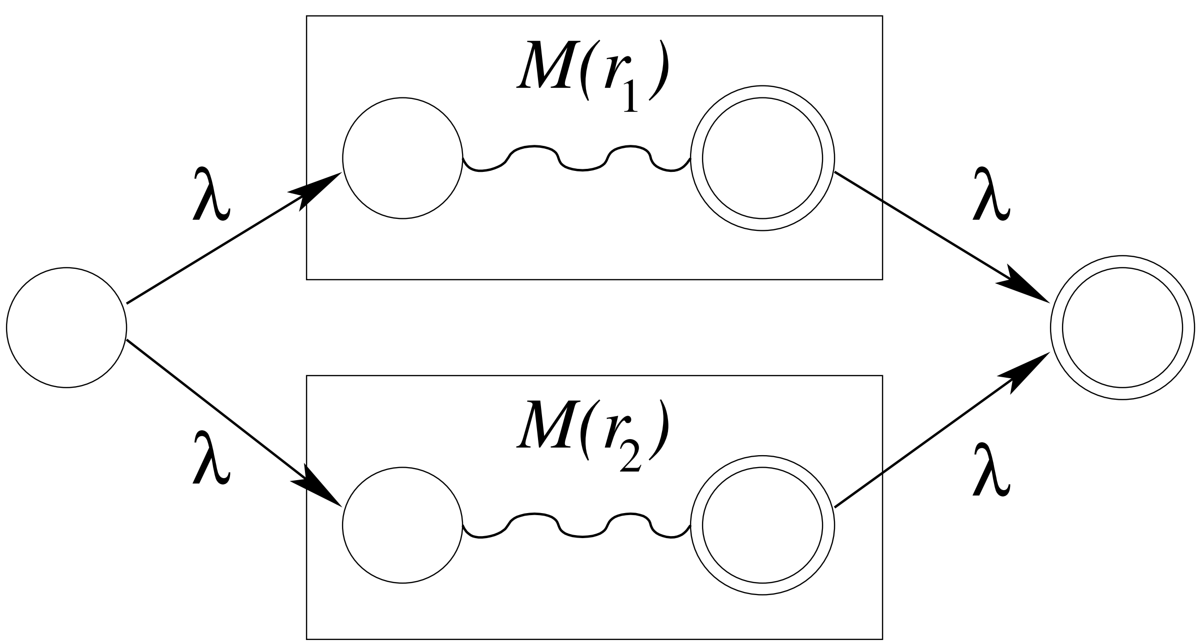
$(ab+abc)^*$

$(a+ba+bba)^*(\lambda+b+bb)$

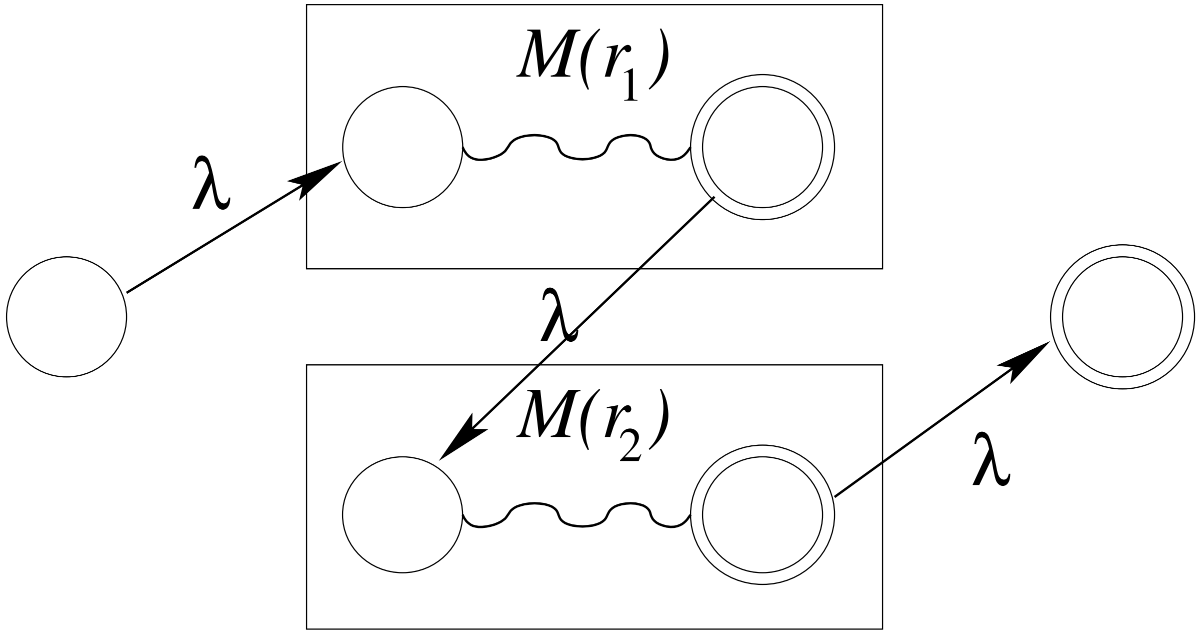
$(\lambda+a+aa)((b+bb)(a+aa))^*(\lambda+b+bb)$

Mapping Regular Expressions to NFAs

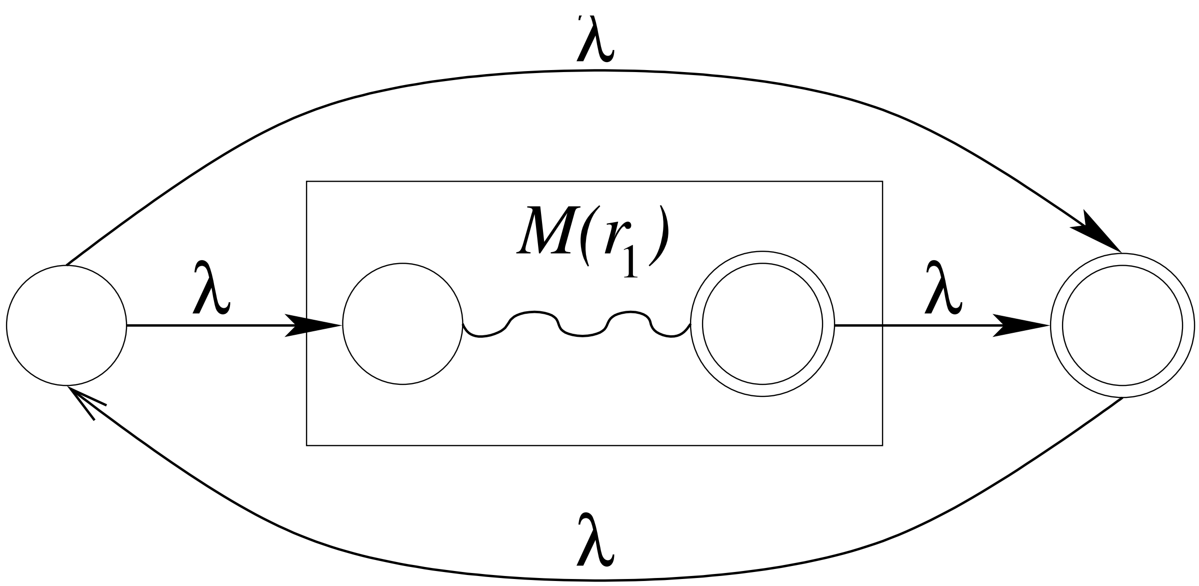
This is the automaton for $r_1 + r_2$.



This is the automaton for $r_1 r_2$.

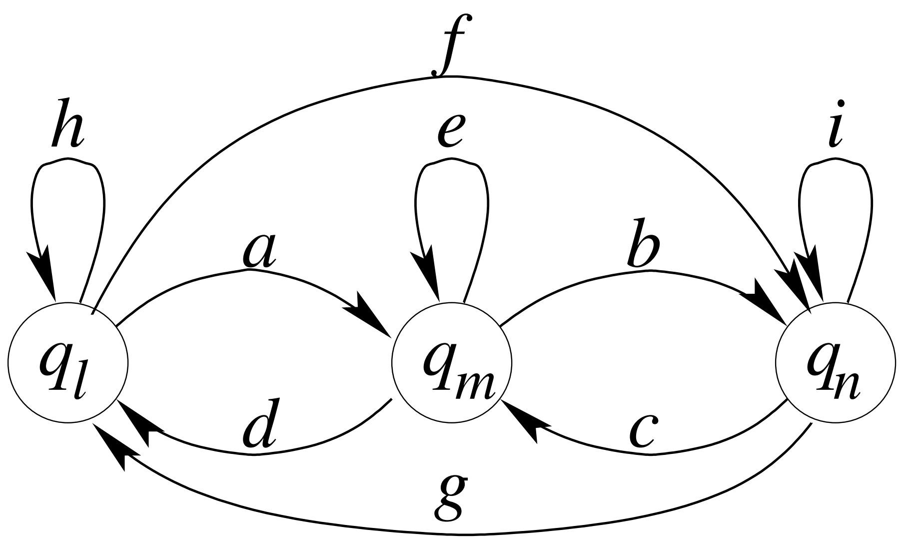


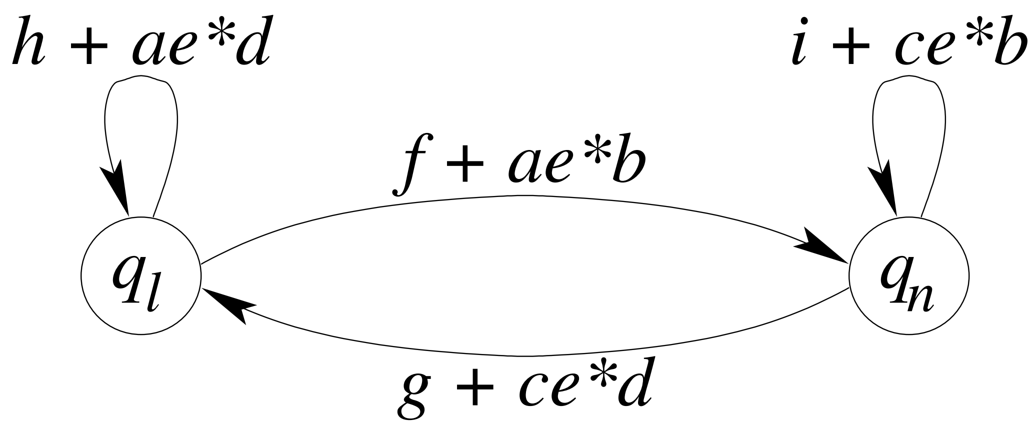
This is the automaton for $(r_1)^*$.

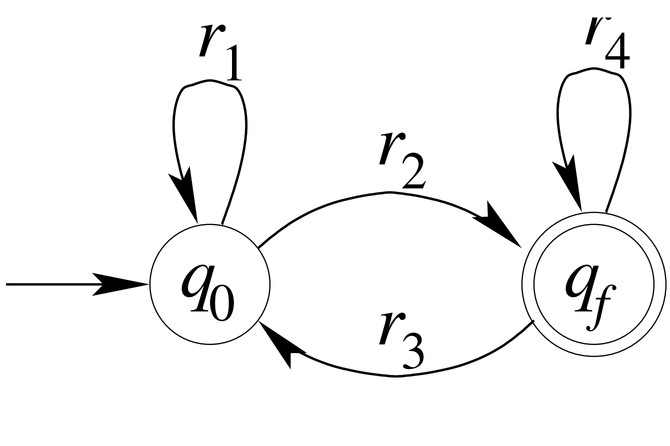


Mapping NFAs to Regular Expressions

A *generalized transition graph* allows labels to be regular expressions. The idea is to eliminate one state at a time.







$= r_1^* r_2 (r_4 + r_3 r_1^* r_2)^*$