Automata theory

An algorithmic approach

Member(x, X)	:	returns **true** if $x \in X$, **false** otherwise.
Complement(X)	:	returns $U \setminus X$.
Intersection(X, Y)	:	returns $X \cap Y$.
Union(X, Y)	:	returns $X \cup Y$.
Empty(X)	:	returns **true** if $X = \emptyset$, **false** otherwise.
Universal(X)	:	returns **true** if $X = U$, **false** otherwise.
Included(X, Y)	:	returns **true** if $X \subseteq Y$, **false** otherwise.
Equal(X, Y)	:	returns **true** if $X = Y$, **false** otherwise.
Projection_1(R)	:	returns the set $\pi_1(R) = \{x \mid \exists y\ (x, y) \in R\}$.
Projection_2(R)	:	returns the set $\pi_2(R) = \{y \mid \exists y\ (x, y) \in R\}$.
Join(R, S)	:	returns $R \circ S = \{(x, z) \mid \exists y \in X\ (x, y) \in R \wedge (y, z) \in S\}$
Post(X, R)	:	returns $post_R(X) = \{y \in U \mid \exists x \in X : (x, y) \in R\}$.
Pre(X, R)	:	returns $pre_R(X) = \{y \in U \mid \exists x \in X : (y, x) \in R\}$.

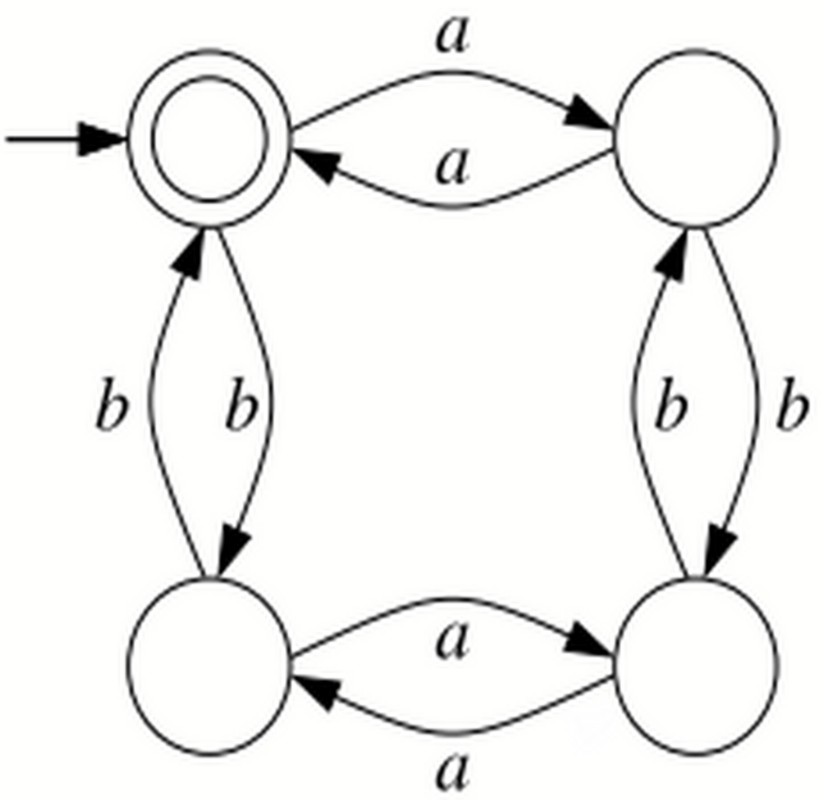
Automata classes and conversions

Regular expressions

$$r ::= \emptyset \mid a \mid r_1 r_2 \mid r_1 + r_2 \mid r^* \qquad \textit{where } a \in \Sigma \cup \{\varepsilon\}$$

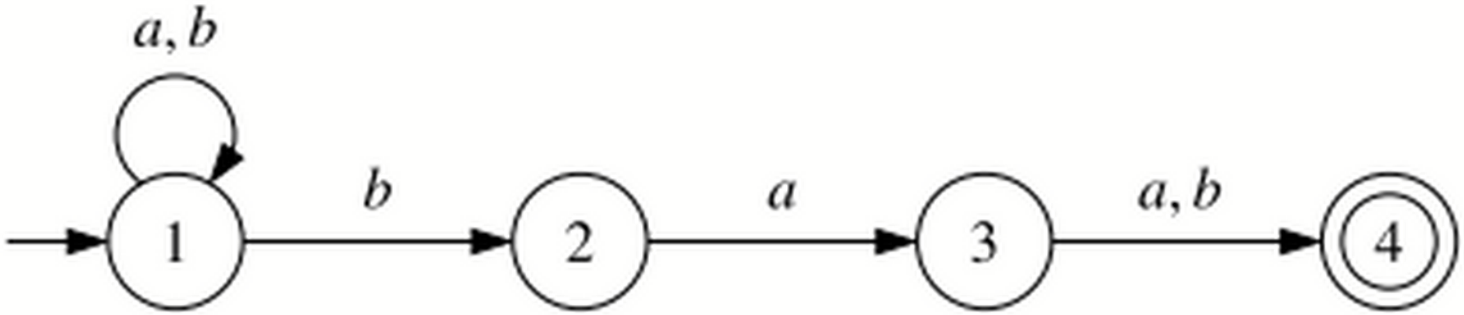
- $\mathcal{L}(\emptyset) = \emptyset$,
- $\mathcal{L}(a) = \{a\}$,
- $\mathcal{L}(r_1 r_2) = \mathcal{L}(r_1) \cdot \mathcal{L}(r_2)$,
- $\mathcal{L}(r_1 + r_2) = \mathcal{L}(r_1) \cup \mathcal{L}(r_2)$, *and*
- $\mathcal{L}(r^*) = \{w_1 \ldots w_k \mid k \geq 0, w_i \in \mathcal{L}(r)\}$ *for all* $1 \leq i \leq k\}$.

Deterministic finite automata (DFA)



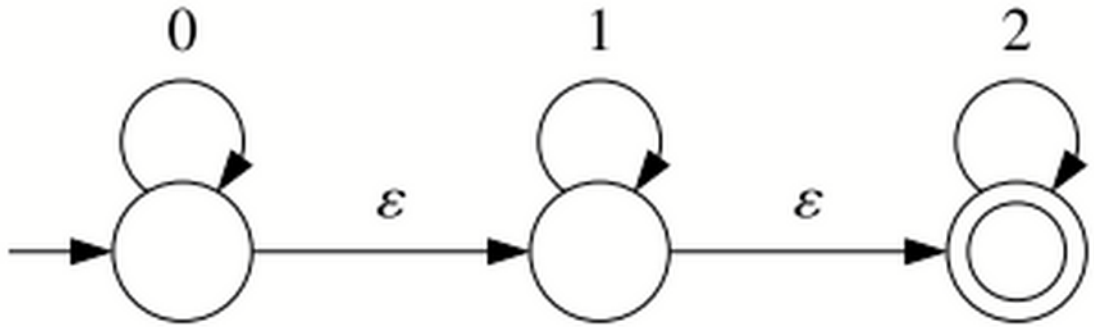
- Q *is a set of* states,
- Σ *is an alphabet,*
- $\delta\colon Q \times \Sigma \rightarrow Q$ *is a* transition function,
- $q_0 \in Q$ *is the* initial state, *and*
- $F \subseteq Q$ *is the set of* final states.

Nondeterministic finite automata (NFA)



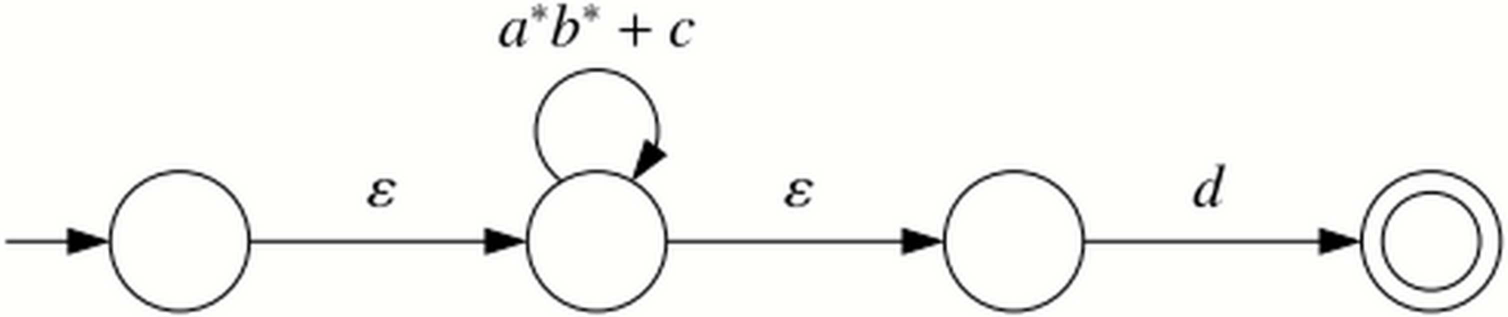
- $\delta\colon Q \times \Sigma \to \mathcal{P}(Q)$ *is a* transition relation.

Nondeterministic finite automata with epsilon-transitions (NFA-e)



- $\delta: Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q)$ *is a* transition relation.

Nondeterministic finite automata with regular-expression transitions (NFA-reg)



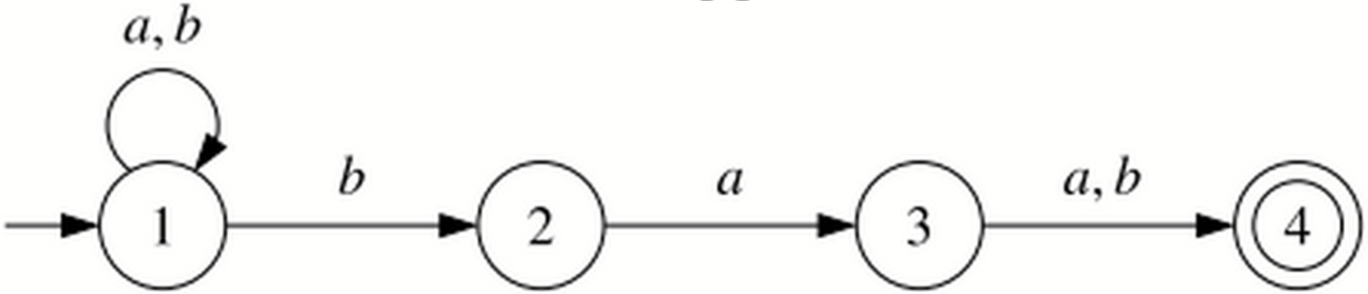
- $\delta\colon Q \times \mathcal{RE}(\Sigma) \to \mathcal{P}(Q)$ *is a relation such that* $\delta(q, r) = \emptyset$ *for all but a finite number of pairs* $(q, r) \in Q \times \mathcal{RE}(\Sigma)$.

Normal form

Definition 2.5 *Let* $A = (Q, \Sigma, \delta, q_0, F)$ *be an automaton. A state* $q \in Q$ *is* reachable from $q' \in Q$ *if* $q = q'$ *or if there exists a run* $q' \xrightarrow{a_1} \ldots \xrightarrow{a_n} q$ *on some input* $a_1 \ldots a_n \in \Sigma^*$. *A is in* normal form *if every state is reachable from the initial state.*

Conversions

NFA to DFA

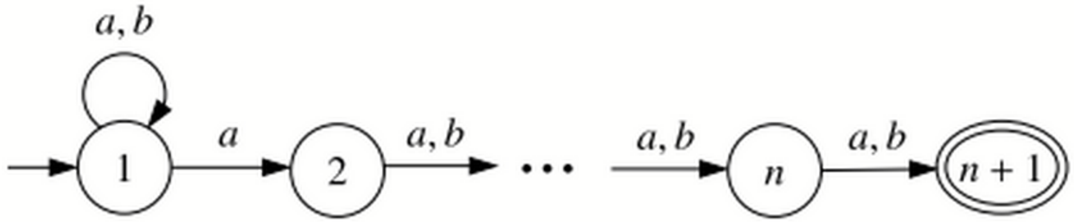


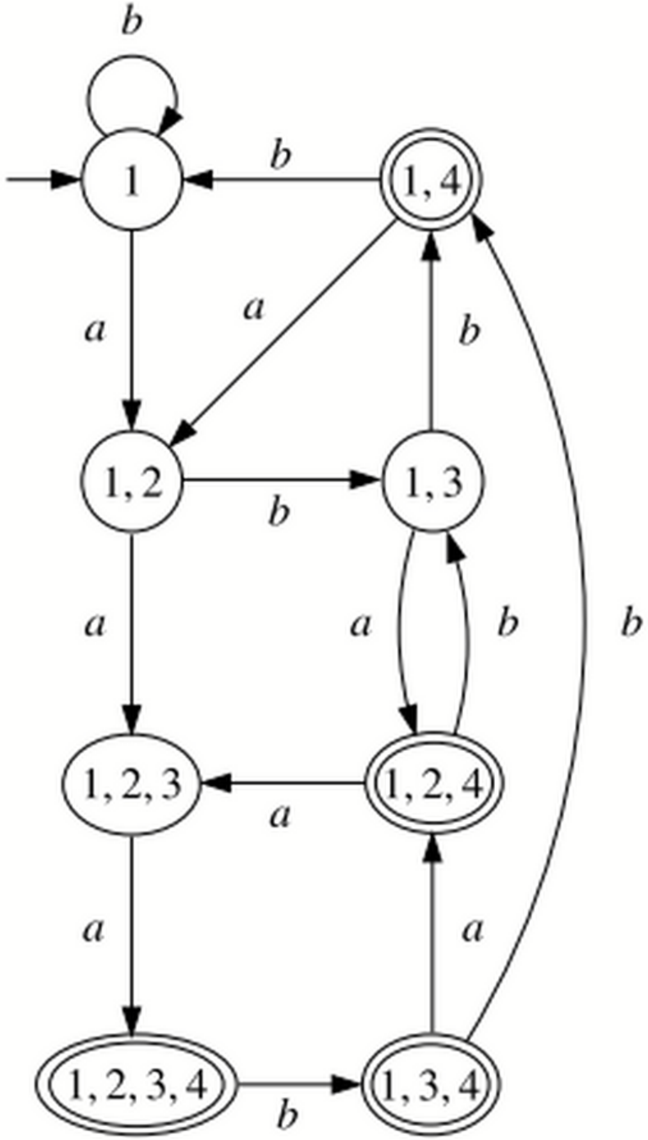
NFAtoDFA(A)

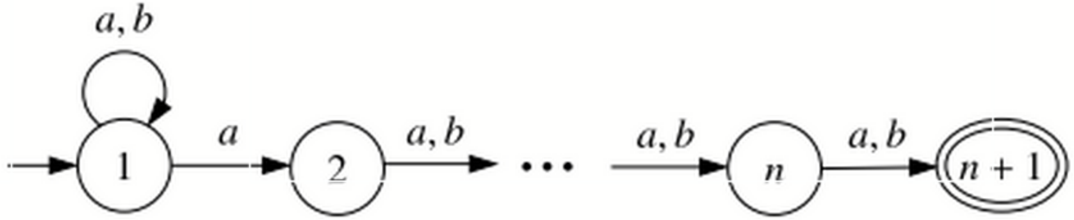
Input: NFA $A = (Q, \Sigma, \delta, q_0, F)$

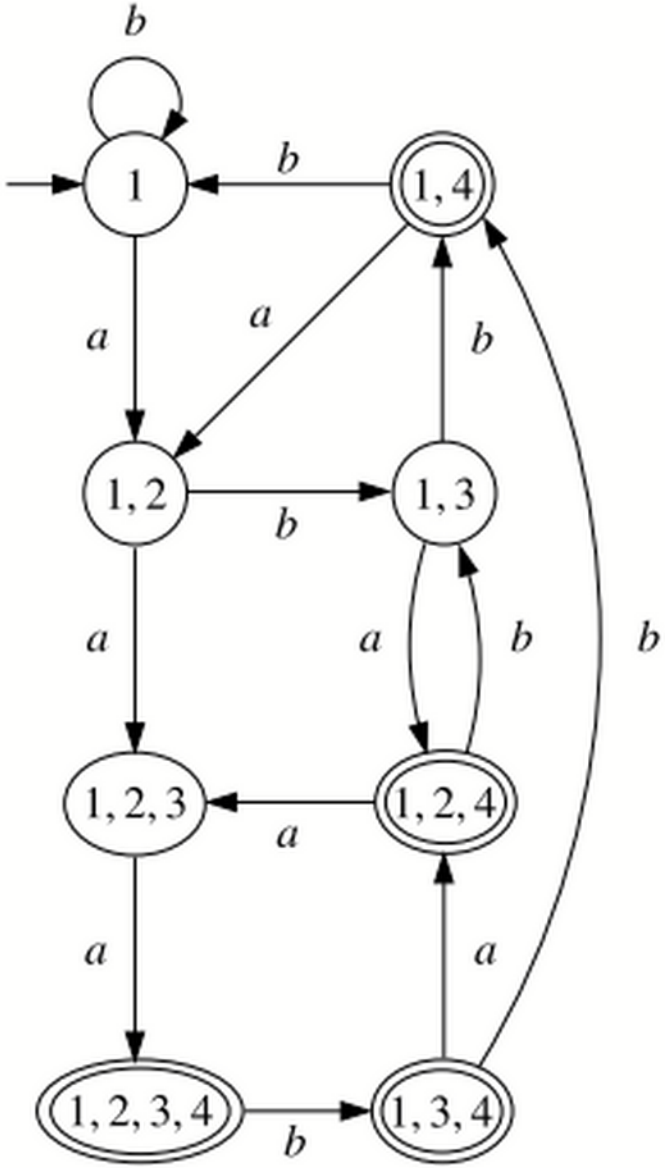
Output: DFA $B = (\mathcal{Q}, \Sigma, \Delta, Q_0, \mathcal{F})$ with $\mathcal{L}(B) = \mathcal{L}(A)$

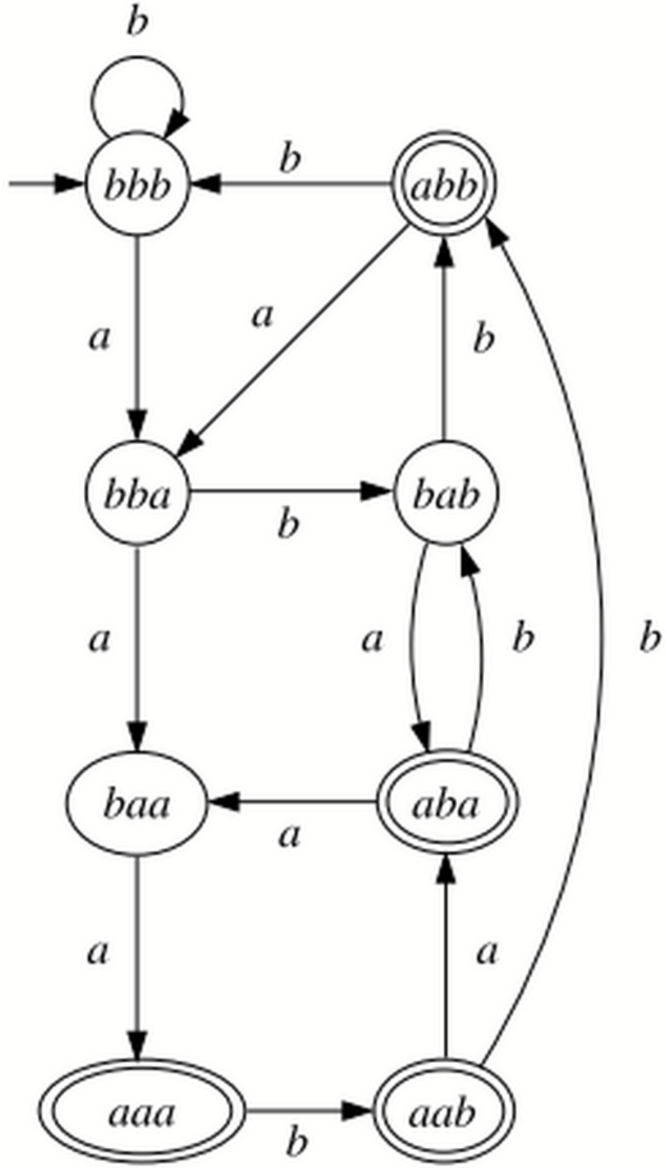
$\mathcal{Q}, \Delta, \mathcal{F} \leftarrow \emptyset$; $Q_0 \leftarrow \{q_0\}$
$\mathcal{W} = \{Q_0\}$
while $\mathcal{W} \neq \emptyset$ **do**
    pick Q' **from** $\mathcal{W}$
    add Q' **to** $\mathcal{Q}$
    if $Q' \cap F \neq \emptyset$ **then add** Q' **to** $\mathcal{F}$
    for all $a \in \Sigma$ **do**
        $Q'' \leftarrow \delta(Q', a)$
        if $Q'' \notin \mathcal{Q}$ **then add** Q'' **to** $\mathcal{W}$
        add (Q', a, Q'') **to** Δ


a, b
1
a
2
a, b
a, b
n
a, b
n + 1

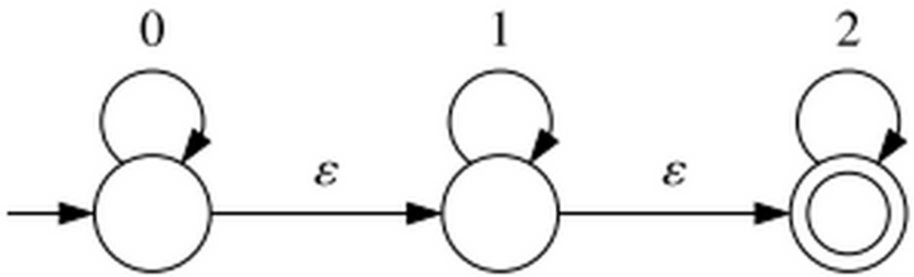

b
1
b
1, 4
a
a
b
1, 2
b
1, 3
a
a
b
b
1, 2, 3
a
1, 2, 4
a
a
1, 2, 3, 4
b
1, 3, 4


a, b
1
a
2
a, b
a, b
n
a, b
n + 1

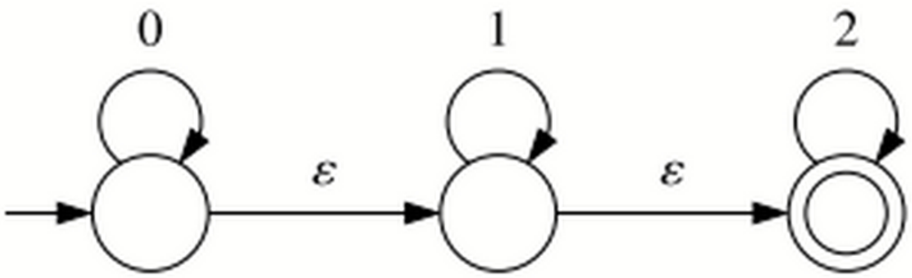

b
1
b
1, 4
a
a
b
1, 2
b
1, 3
a
a
b
b
1, 2, 3
a
1, 2, 4
a
a
1, 2, 3, 4
b
1, 3, 4

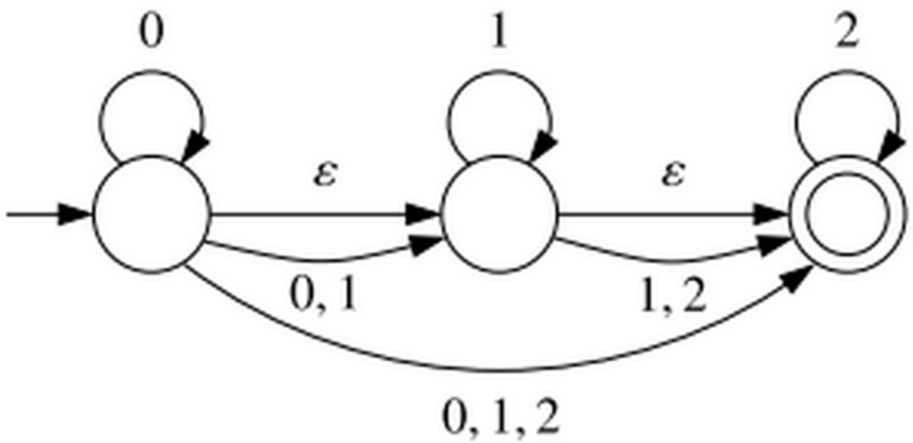

b
bbb
b
abb
a
a
b
bba
b
bab
a
a
b
b
baa
a
aba
a
a
aaa
b
aab

NFA-e to

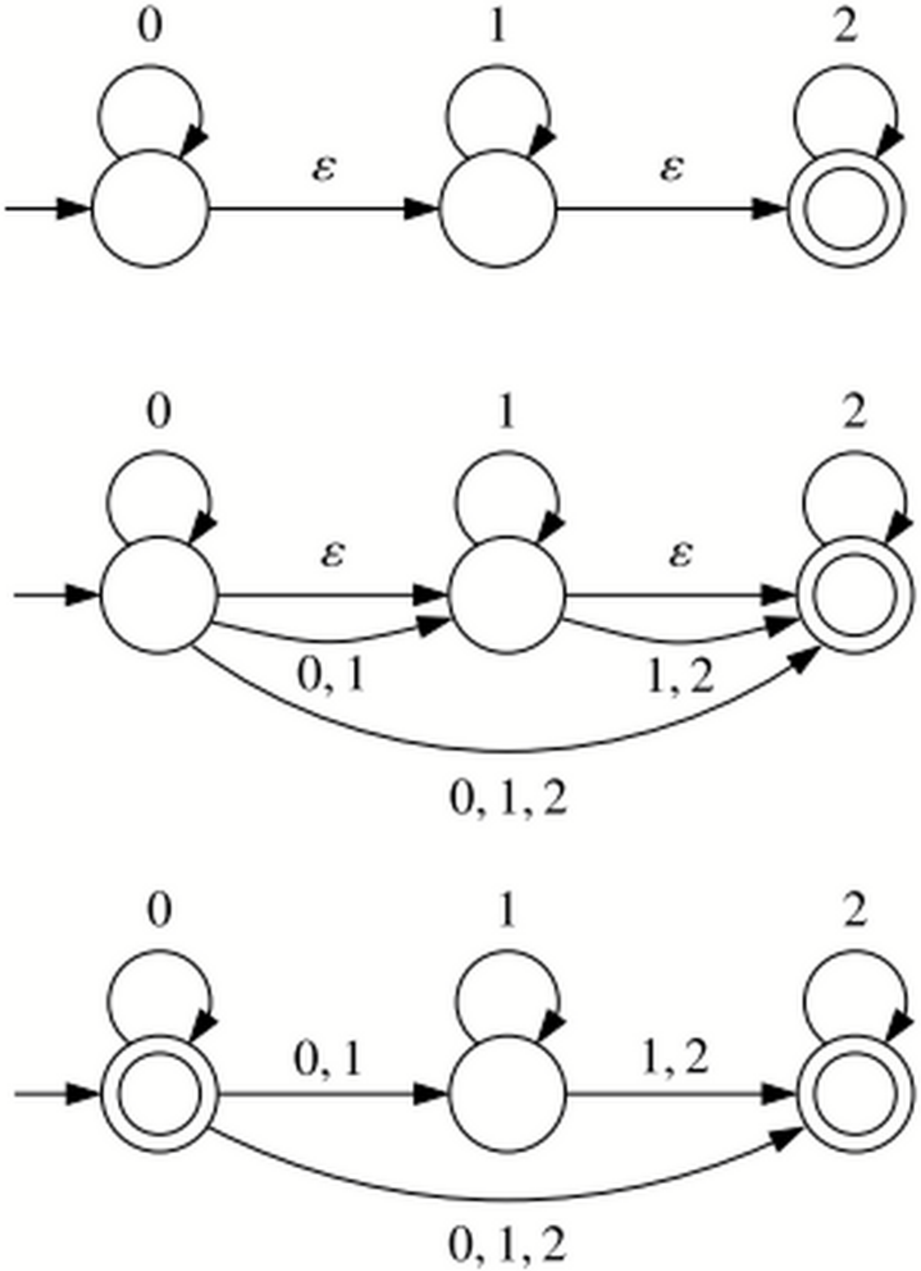


NFA-e to





NFA-e to



NFAεtoNFA(A)

Input: NFA-ε $A = (Q, \Sigma, \delta, q_0, F)$

Output: NFA $B = (Q', \Sigma, \delta', q'_0, F')$ with $\mathcal{L}(B) = \mathcal{L}(A)$

$q'_0 \leftarrow q_0$
$Q' \leftarrow \{q_0\}$; $\delta' \leftarrow \emptyset$; $F' \leftarrow F \cap \{q_0\}$
$W \leftarrow \{(q, \alpha, q') \in \delta \mid q = q_0\}$
while $W \neq \emptyset$ **do**
    pick (q_1, α, q_2) **from** W
    if $\alpha \neq \varepsilon$ **then**
        add q_2 **to** Q'; **add** (q_1, α, q_2) **to** δ'; **if** $q_2 \in F$ **then add** q_2 **to** F'
        for all $(q_2, \varepsilon, q_3) \in \delta$ **do**
            if $(q_1, \alpha, q_3) \notin \delta'$ **then add** (q_1, α, q_3) **to** W
        for all $(q_2, a, q_3) \in \delta$ **do**
            if $(q_1, a, q_3) \notin \delta'$ **then add** (q_1, a, q_3) **to** W
    else $/ * \alpha = \varepsilon * /$
        if $q_0 \notin F'$ and $q_2 \in F$ **then add** q_0 **to** F'
        for all $(q_2, \beta, q_3) \in \delta$ **do**
            if $(q_1, \beta, q_3) \notin \delta'$ **then add** (q_1, β, q_3) **to** W

Proposition 2.7 *Let A be a NFA-ε, and let B = NFAεtoNFA(A). Then B is a NFA and* $\mathcal{L}(A) = \mathcal{L}(B)$.

Proof: Notice first that every transition is added to W at most once, and so the algorithm terminates. It follows that every non-ε transition added to W is eventually added to δ' (line 7). This fact is implicitly used all along the proof.

To show that B is a NFA we have to prove that it only has non-ε transitions, and that it is in normal form, i.e., that every state of Q' is reachable from q_0 in B.

invariant, which can be easily proved by inspection: for every transition (q_1, α, q_2) added to W, if $\alpha = \epsilon$ then $q_1 = q_0$, and if $\alpha \neq \epsilon$, then q_2 is reachable in B

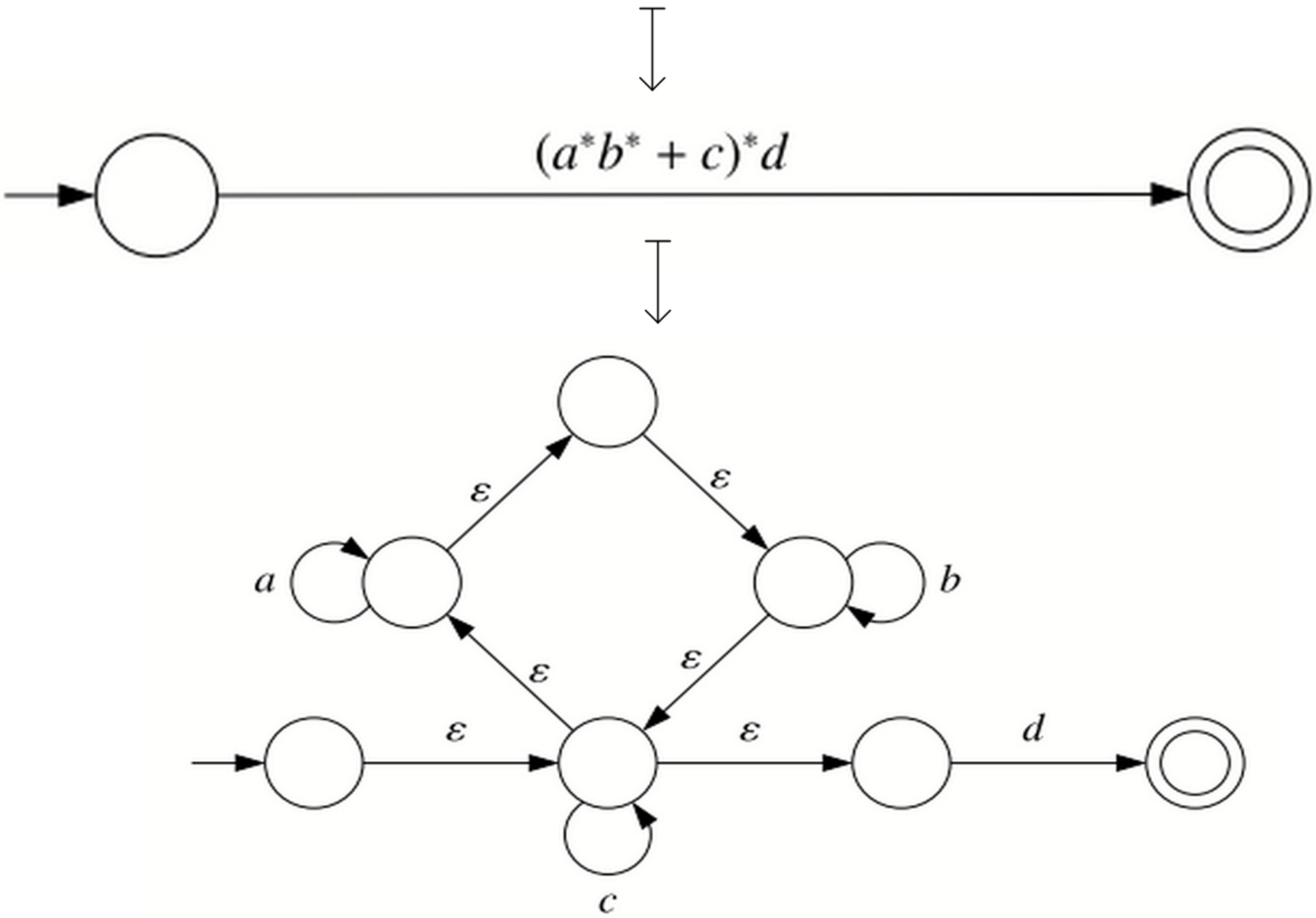
$\mathcal{L}(A) = \mathcal{L}(B)$. The inclusion $\mathcal{L}(A) \supseteq \mathcal{L}(B)$ follows from the fact that every transition added to δ' is a shortcut, which can be proved by inspection. For the inclusion $\mathcal{L}(A) \subseteq \mathcal{L}(B)$, we first prove that $\varepsilon \in \mathcal{L}(A)$ implies $\varepsilon \in \mathcal{L}(B)$. Let $q_0 \xrightarrow{\varepsilon} q_1 \ldots q_{n-1} \xrightarrow{\varepsilon} q_n$ be a run of A such that $q_n \in F$. If $n = 0$ (i.e., $q_n = q_0$), then we are done. If $n > 0$, then we prove by induction on n that a transition (q_0, ε, q_n) is eventually added to δ', and so that

We now prove that for every $w \in \Sigma^+$, if $w \in \mathcal{L}(A)$ then $w \in \mathcal{L}(B)$. Let $w = a_1 a_2 \ldots a_n$ with $n \geq 1$. Then A has a run

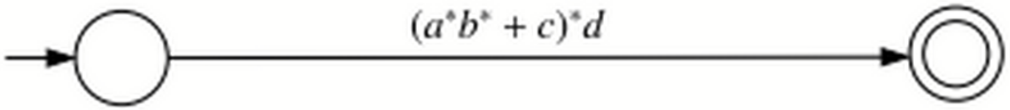
$$q_0 \xrightarrow{\varepsilon} \ldots \xrightarrow{\varepsilon} q_{m_1} \xrightarrow{a_1} q_{m_1+1} \xrightarrow{\varepsilon} \ldots \xrightarrow{\varepsilon} q_{m_n} \xrightarrow{a_n} q_{m_n+1} \xrightarrow{\varepsilon} \ldots \xrightarrow{\varepsilon} q_m$$

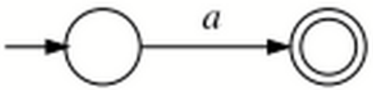
Regular expressions to NFA-e

$$(a^*b^* + c)^*d$$



$$
\begin{array}{rcl@{\qquad}rcl}
\emptyset \cdot r & \rightsquigarrow & \emptyset & r \cdot \emptyset & \rightsquigarrow & \emptyset \\
r + \emptyset & \rightsquigarrow & r & \emptyset + r & \rightsquigarrow & r \\
\emptyset^* & \rightsquigarrow & \epsilon & & &
\end{array}
$$

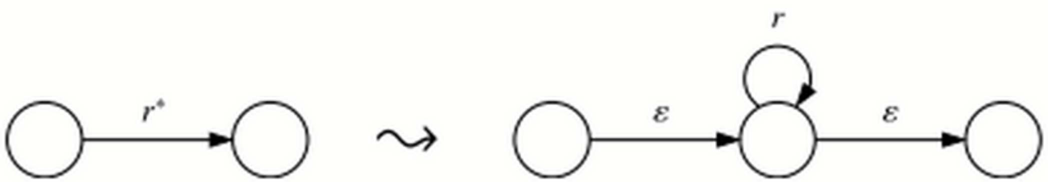




Automaton for the regular expression a, where $a \in \Sigma \cup \{\epsilon\}$

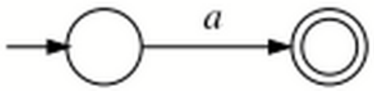
Rule for concatenation

Rule for choice



Rule for Kleene iteration

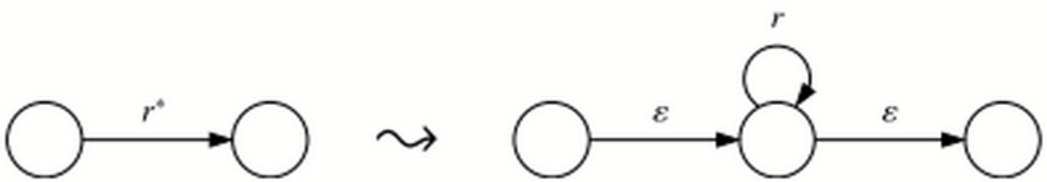
$$
\begin{array}{rclcrcl}
\emptyset \cdot r & \leadsto & \emptyset & \quad & r \cdot \emptyset & \leadsto & \emptyset \\
r + \emptyset & \leadsto & r & & \emptyset + r & \leadsto & r \\
\emptyset^* & \leadsto & \epsilon & & & &
\end{array}
$$



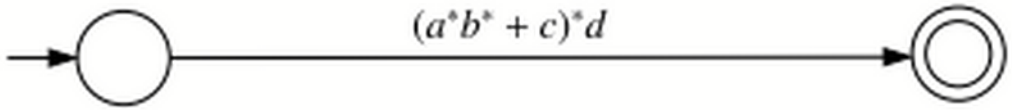
Automaton for the regular expression a, where $a \in \Sigma \cup \{\epsilon\}$

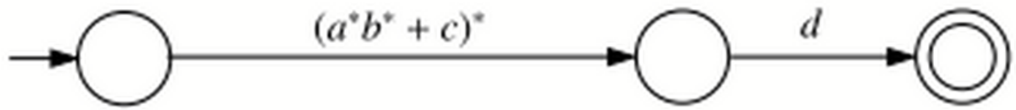
Rule for concatenation

Rule for choice

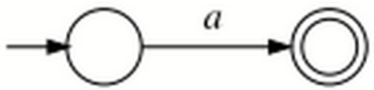


Rule for Kleene iteration





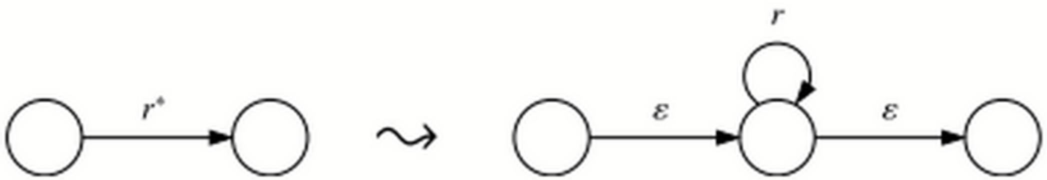
$$
\begin{array}{rclrcl}
\emptyset \cdot r & \rightsquigarrow & \emptyset & \quad r \cdot \emptyset & \rightsquigarrow & \emptyset \\
r + \emptyset & \rightsquigarrow & r & \quad \emptyset + r & \rightsquigarrow & r \\
\emptyset^* & \rightsquigarrow & \epsilon & & &
\end{array}
$$



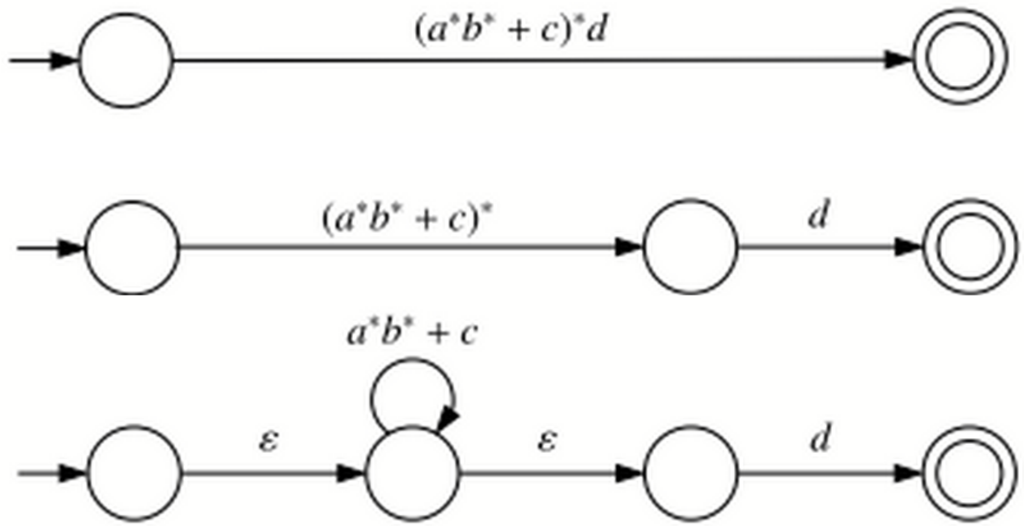
Automaton for the regular expression a, where $a \in \Sigma \cup \{\epsilon\}$

Rule for concatenation

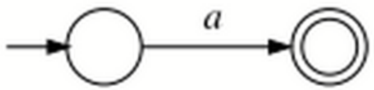
Rule for choice



Rule for Kleene iteration



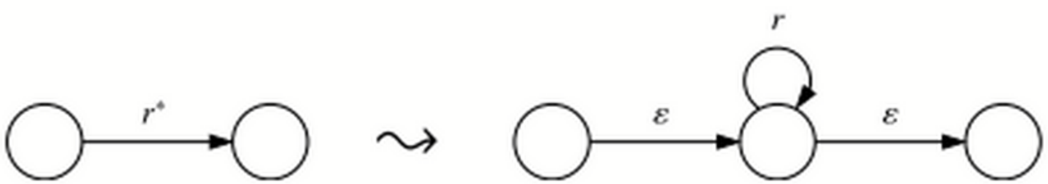
$$
\begin{array}{rclcrcl}
\emptyset \cdot r & \rightsquigarrow & \emptyset & \quad & r \cdot \emptyset & \rightsquigarrow & \emptyset \\
r + \emptyset & \rightsquigarrow & r & & \emptyset + r & \rightsquigarrow & r \\
\emptyset^* & \rightsquigarrow & \epsilon & & & &
\end{array}
$$



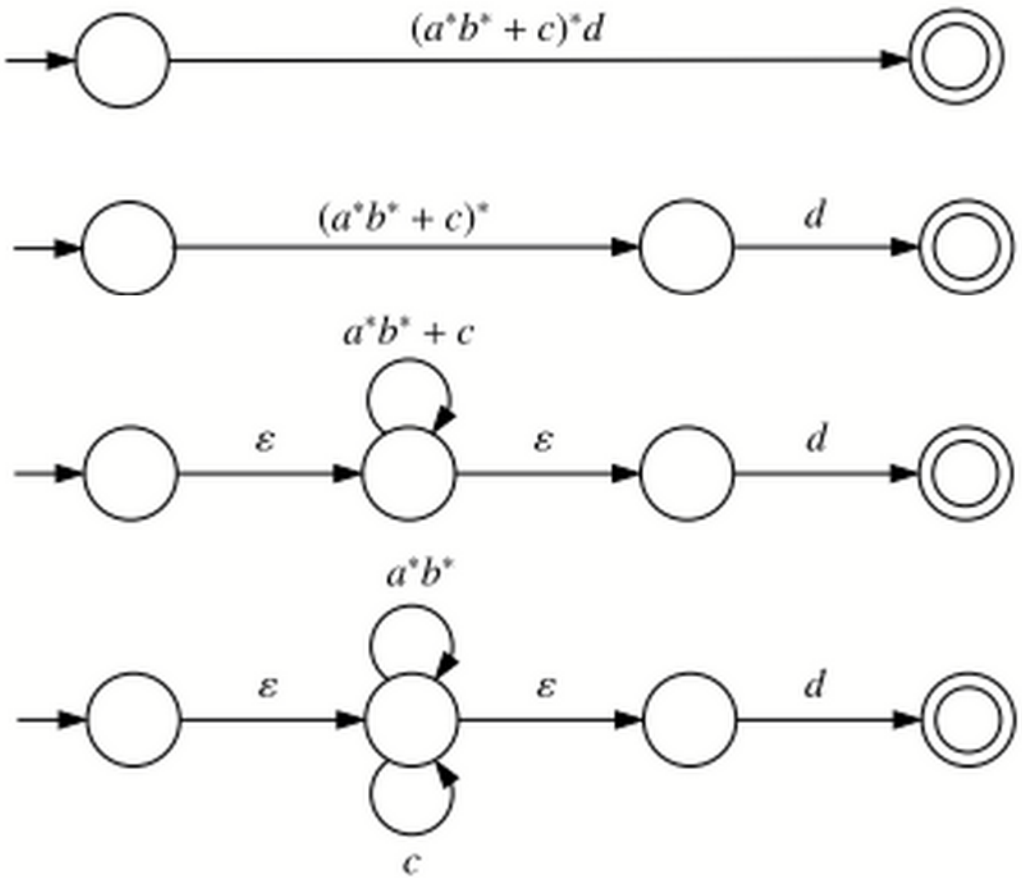
Automaton for the regular expression a, where $a \in \Sigma \cup \{\epsilon\}$

Rule for concatenation

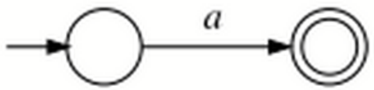
Rule for choice



Rule for Kleene iteration



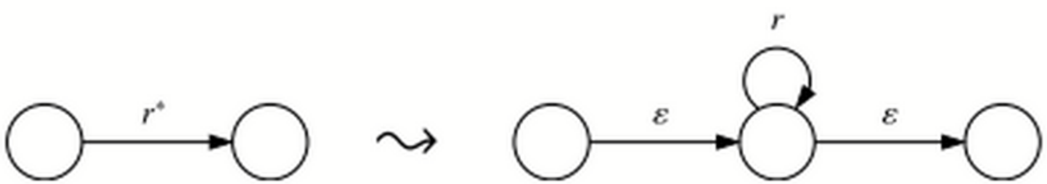
$$
\begin{array}{rcl@{\qquad}rcl}
\emptyset \cdot r & \rightsquigarrow & \emptyset & r \cdot \emptyset & \rightsquigarrow & \emptyset \\
r + \emptyset & \rightsquigarrow & r & \emptyset + r & \rightsquigarrow & r \\
\emptyset^* & \rightsquigarrow & \epsilon & & &
\end{array}
$$



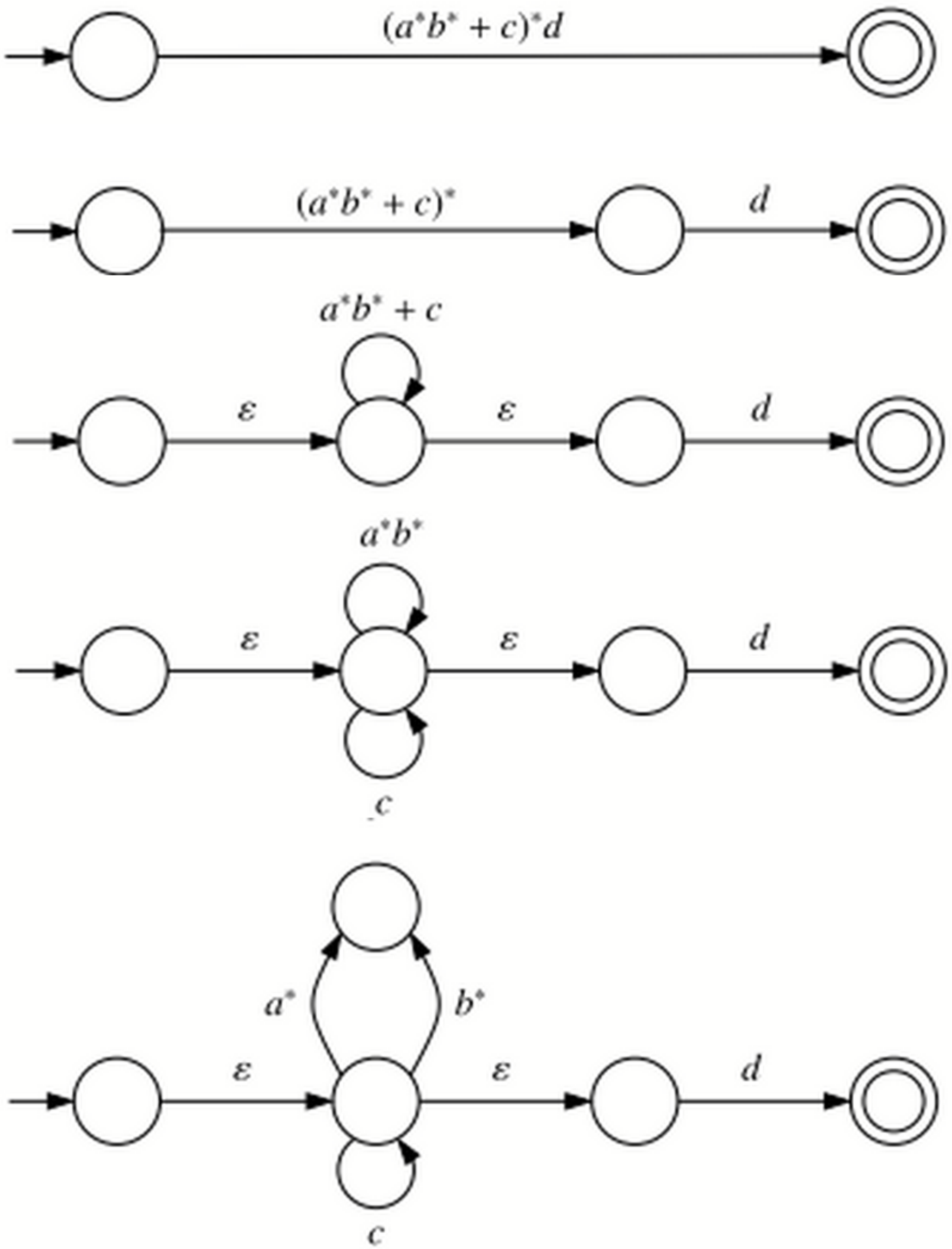
Automaton for the regular expression a, where $a \in \Sigma \cup \{\epsilon\}$

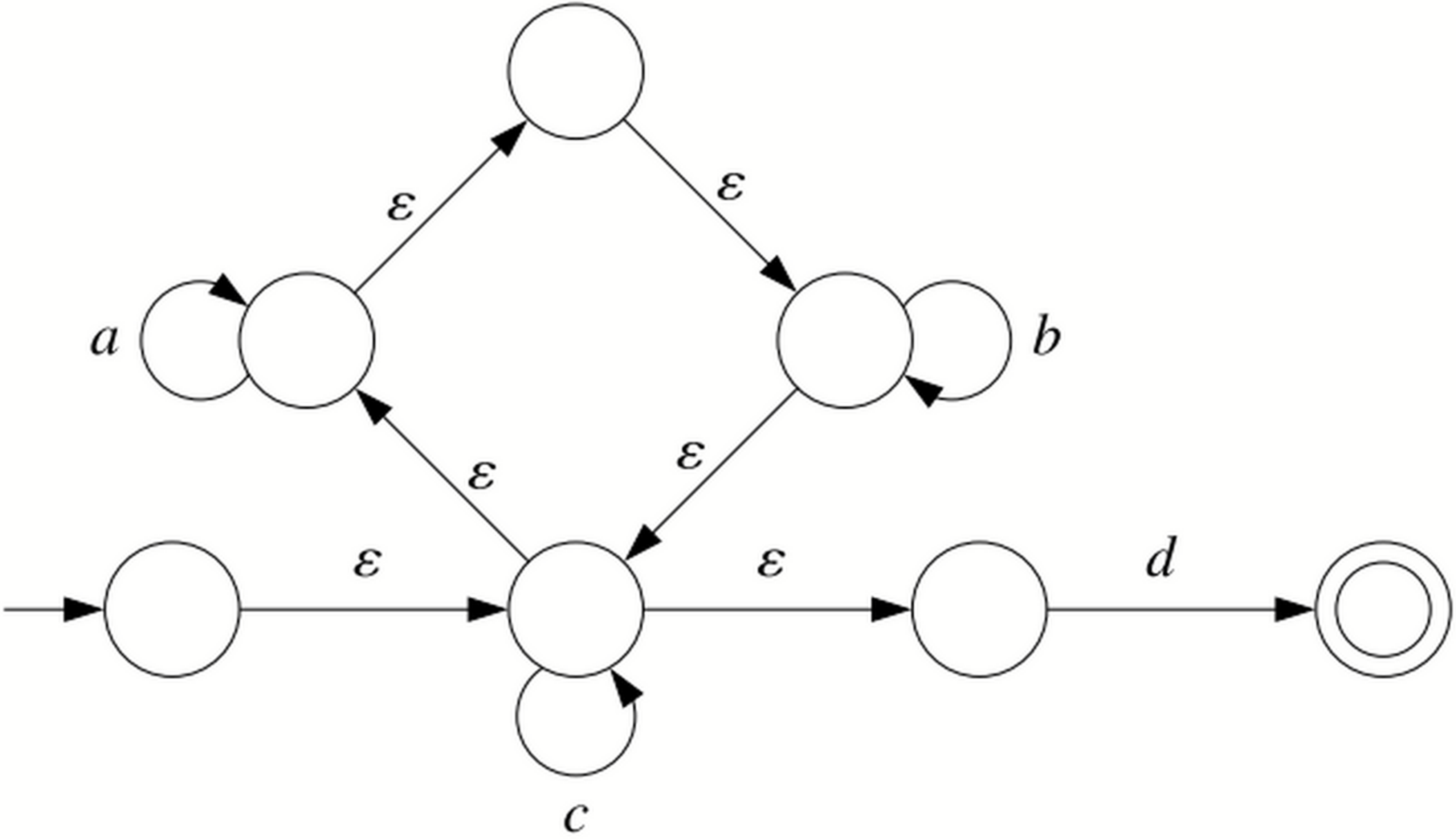
Rule for concatenation

Rule for choice



Rule for Kleene iteration

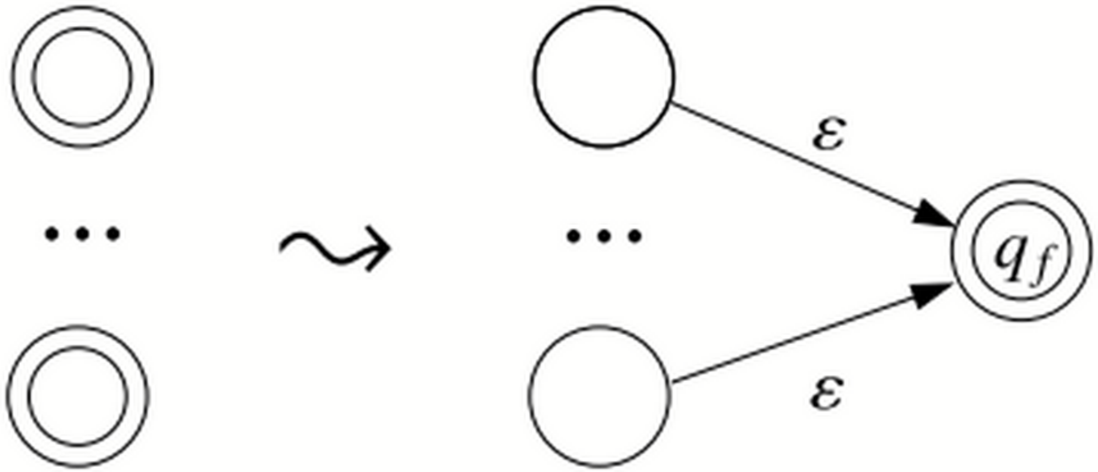




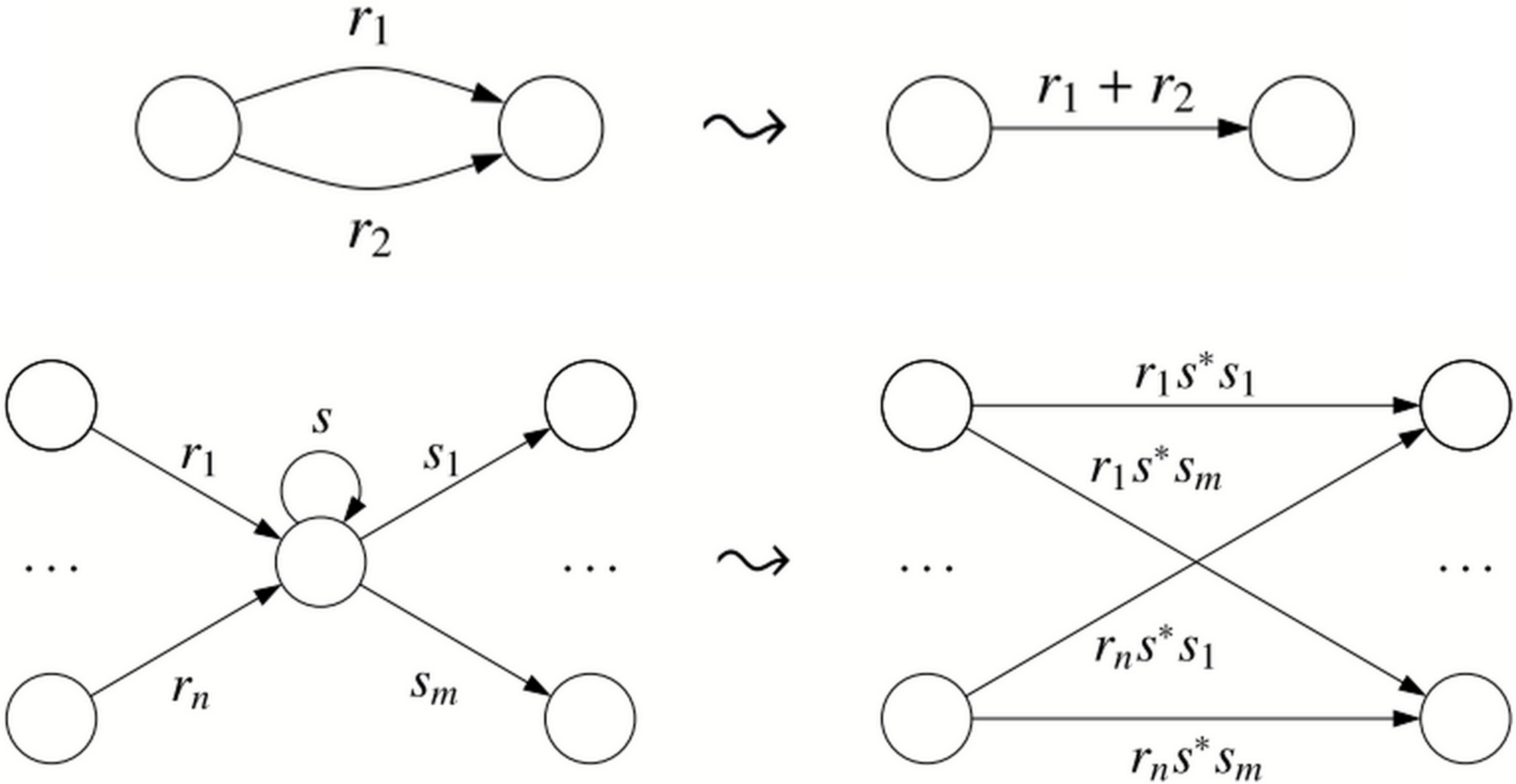
ε
ε
a
b
ε
ε
ε
ε
d
c

NFA-e to regular expressions

Preprocessing:

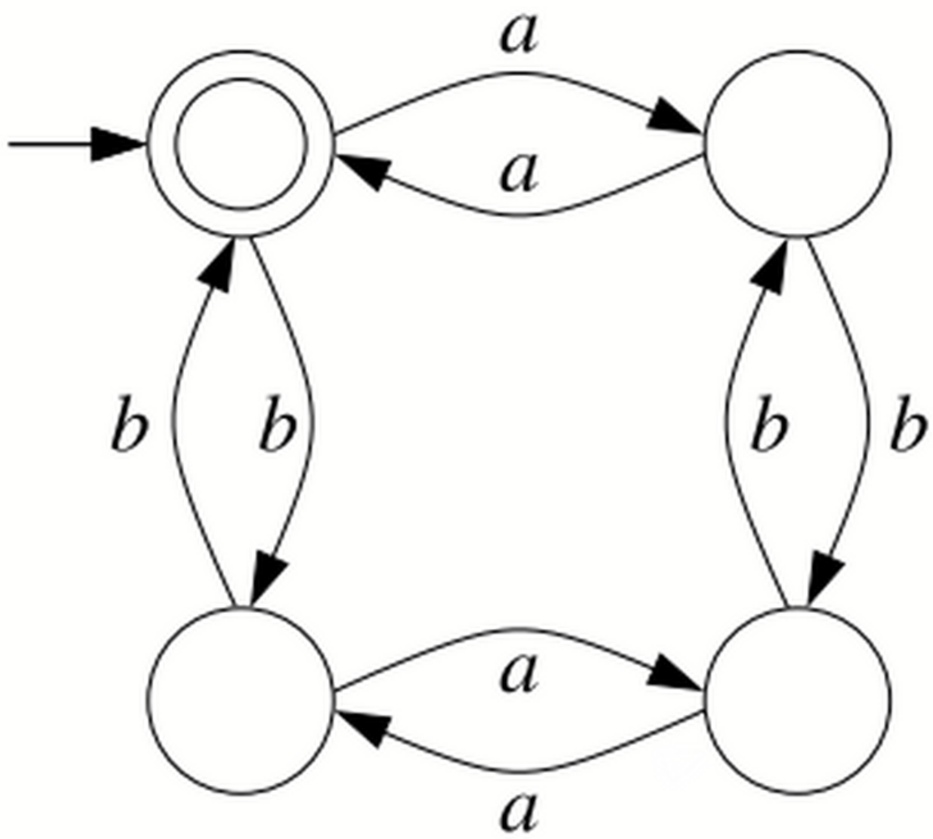


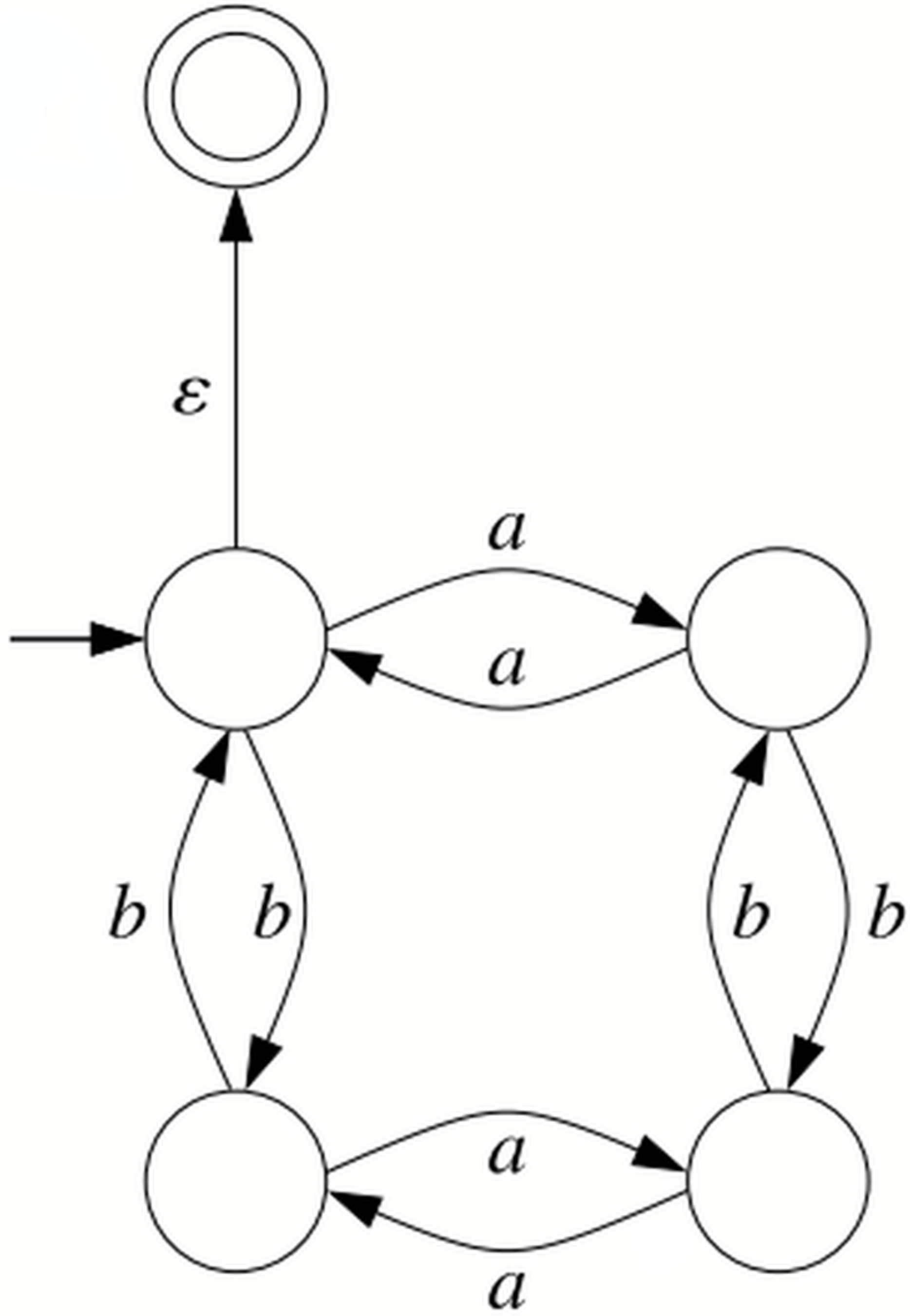
Processing:

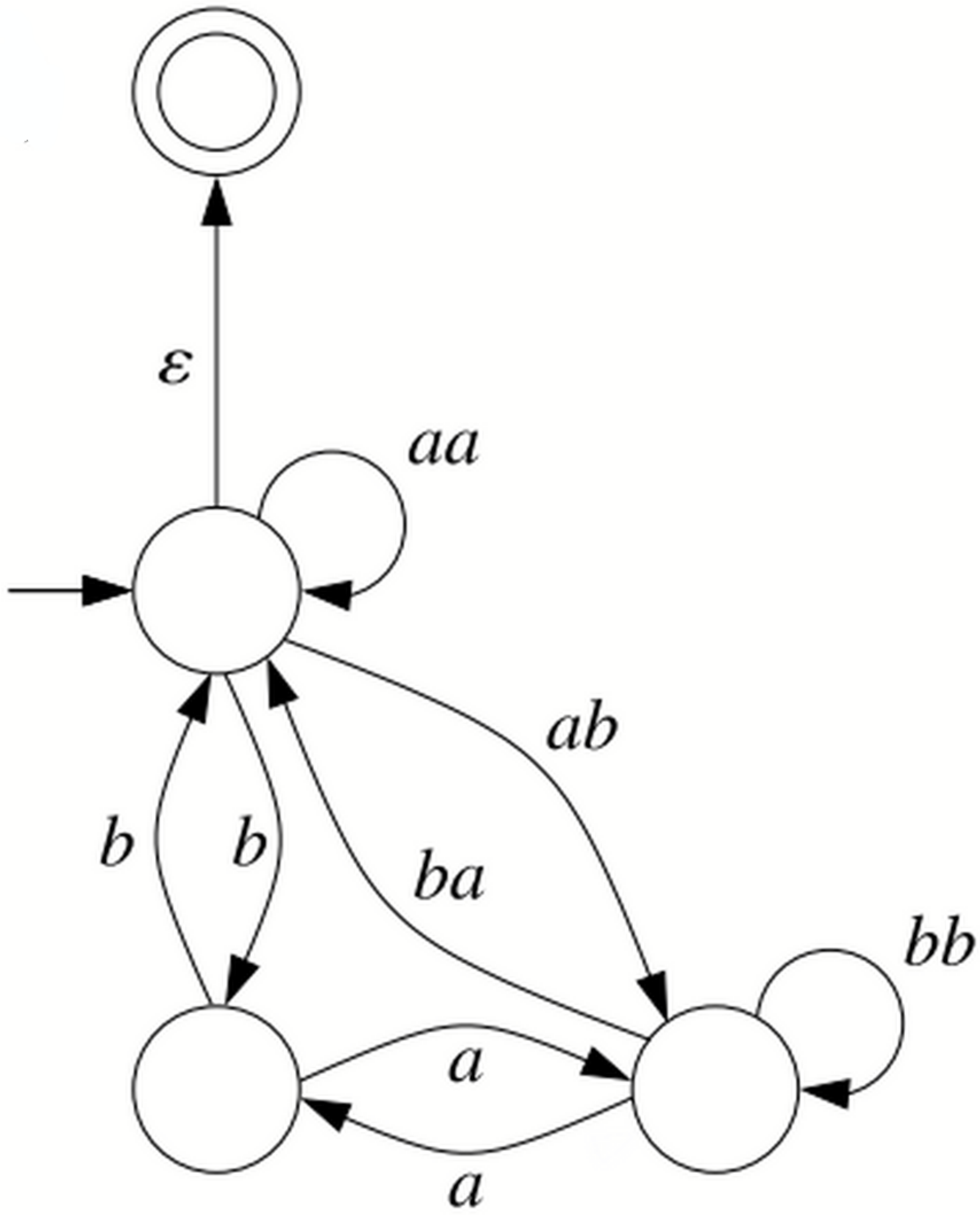


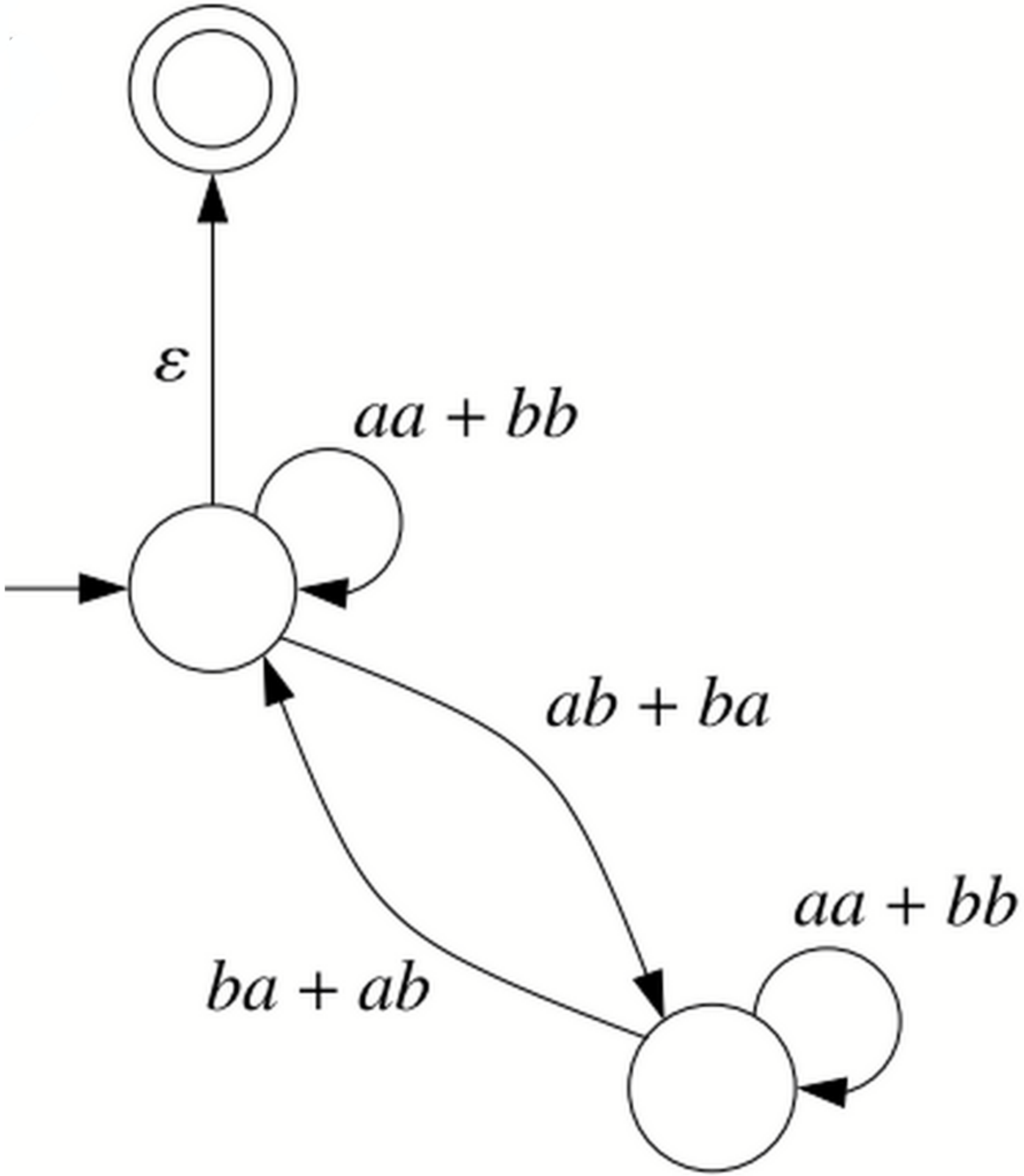
r_1
r_2
$\rightsquigarrow$
$r_1 + r_2$
s
r_1
s_1
r_n
s_m
$\cdots$
$\cdots$
$\rightsquigarrow$
$r_1 s^* s_1$
$r_1 s^* s_m$
$r_n s^* s_1$
$r_n s^* s_m$
$\cdots$
$\cdots$

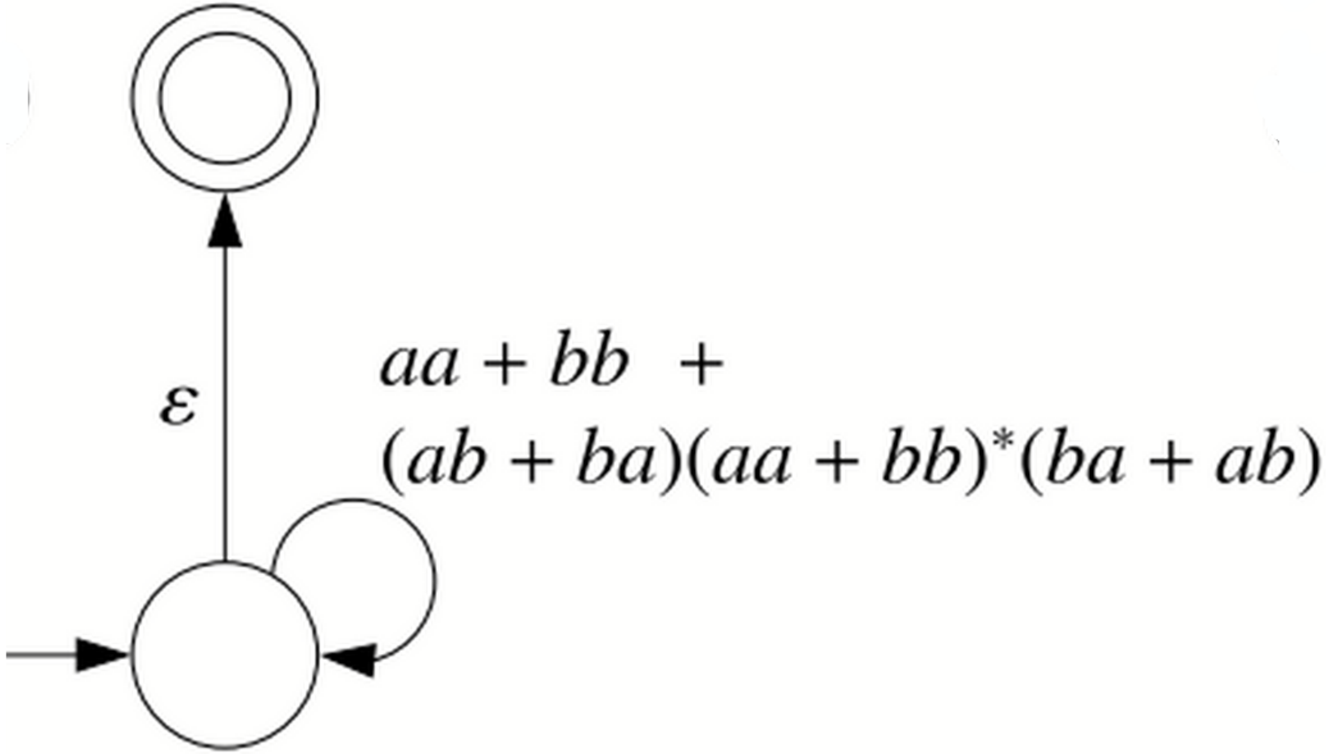
Postprocessing (if necessary):


a
a
b
b
b
b
a
a


ε
a
a
b
b
b
b
a
a


ε
aa
ab
b
b
ba
bb
a
a


ε
aa + bb
ab + ba
aa + bb
ba + ab


ε
aa + bb +
(ab + ba)(aa + bb)*(ba + ab)