8 Non-Deterministic Finite-State Automata (NFAs)

26. Give two distinct accepting computations of the NFA of the previous example on the string '$abab$'.

    (Solution)

$$\langle\langle 0, abab\rangle\,, \langle 1, bab\rangle\,, \langle 3, ab\rangle\,, \langle 3, b\rangle\,, \langle 2, \varepsilon\rangle\rangle$$

$$\langle\langle 0, abab\rangle\,, \langle 3, bab\rangle\,, \langle 0, ab\rangle\,, \langle 1, b\rangle\,, \langle 0, \varepsilon\rangle\rangle$$

27. Give two distinct non-accepting computations of the NFA of the previous example on the string '$abab$'.

    (Solution)

$$\langle\langle 0, abab\rangle\,, \langle 1, bab\rangle\,, \langle 1, ab\rangle\,, \langle 2, b\rangle\rangle$$

$$\langle\langle 0, abab\rangle\,, \langle 1, bab\rangle\,, \langle 0, ab\rangle\,, \langle 1, b\rangle\,, \langle 1, \varepsilon\rangle\rangle$$

28. Prove the claim: for all $w \in \Sigma^+$ and $q \in Q$, $\hat{\delta}'(q, w) = \hat{\delta}(q, w)$.

    (Solution)

    First note that

$$\begin{aligned}
\varepsilon\text{-Closure}(A \cup B) &= \bigcup_{q \in A \cup B} [\varepsilon\text{-Closure}(q)] \\
&= \bigcup_{q \in A} [\varepsilon\text{-Closure}(q)] \cup \bigcup_{q \in B} [\varepsilon\text{-Closure}(q)] \\
&= \varepsilon\text{-Closure}(A) \cup \varepsilon\text{-Closure}(B).
\end{aligned}$$

From which it follows

$$\varepsilon\text{-Closure}(\hat{\delta}(q, w)) = \hat{\delta}(q, w)$$

since

$$\varepsilon\text{-Closure}(\hat{\delta}(q, \varepsilon)) = \varepsilon\text{-Closure}(\varepsilon\text{-Closure}(q)) = \varepsilon\text{-Closure}(q) = \hat{\delta}(q, \varepsilon).$$

and

$$\begin{aligned}
\varepsilon\text{-Closure}(\hat{\delta}(q,w\sigma)) &= \varepsilon\text{-Closure}(\bigcup_{q'\in\hat{\delta}(q,w)}[\varepsilon\text{-Closure}(\delta(q',\sigma))]) \\
&= \bigcup_{q'\in\hat{\delta}(q,w)}[\varepsilon\text{-Closure}(\varepsilon\text{-Closure}(\delta(q',\sigma)))] \\
&= \bigcup_{q'\in\hat{\delta}(q,w)}[\varepsilon\text{-Closure}(\delta(q',\sigma))] \\
&= \hat{\delta}(q,w\sigma).
\end{aligned}$$

Proceeding by induction on w.

(Basis:)

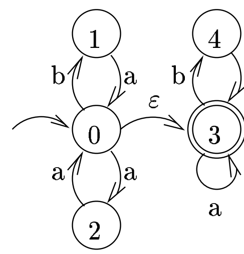
$$\begin{aligned}
\hat{\delta}'(q,\sigma) &= \delta'(\hat{\delta}'(q,\varepsilon),\sigma) \\
&= \delta'(q,\sigma) \\
&= \hat{\delta}(q,\sigma).
\end{aligned}$$

(Induction:)

$$\begin{aligned}
\hat{\delta}'(q,w\sigma) &= \bigcup_{q'\in\hat{\delta}'(q,w)}[\delta'(q',\sigma)] \quad \text{by def of } \hat{\delta} \\
&= \bigcup_{q'\in\hat{\delta}(q,w)}[\delta'(q',\sigma)] \quad \text{by IH} \\
&= \bigcup_{q'\in\hat{\delta}(q,w)}[\hat{\delta}(q',\sigma)] \quad \text{by def of } \delta' \\
&= \bigcup_{q'\in\hat{\delta}(q,w)}[\bigcup_{q''\in\hat{\delta}(q',\varepsilon)}[\varepsilon\text{-Closure}(\delta(q'',\sigma))]] \quad \text{by def of } \hat{\delta} \\
&= \bigcup_{q'\in\hat{\delta}(q,w)}[\bigcup_{q''\in\varepsilon\text{-Closure}(q')}[\varepsilon\text{-Closure}(\delta(q'',\sigma))]] \\
&= \bigcup_{q''\in\varepsilon\text{-Closure}(\hat{\delta}(q,w))}[\varepsilon\text{-Closure}(\delta(q'',\sigma))]]
\end{aligned}$$

$$= \bigcup_{q'' \in \hat{\delta}(q,w)} [\varepsilon\text{-Closure}(\delta(q'', \sigma))]$$
$$= \hat{\delta}(q, w\sigma).$$

29. Convert the following NFA to a DFA.



(Solution)

q	$\delta(q,a)$	$\delta(q,b)$	$\delta(q,\varepsilon)$
0	$\{2\}$	$\{1\}$	$\{3\}$
1	$\{0\}$	$\emptyset$	$\emptyset$
2	$\{0\}$	$\emptyset$	$\emptyset$
3	$\{3\}$	$\{4\}$	$\emptyset$
4	$\{3\}$	$\emptyset$	$\emptyset$

q	$\varepsilon -$ Closure
0	$\{0,3\}$
1	$\{1\}$
2	$\{2\}$
3	$\{3\}$
4	$\{4\}$

q	$\delta'(q,a)$	$\delta'(q,b)$
0	$\{2,3\}$	$\{1,4\}$
1	$\{0,3\}$	$\emptyset$
2	$\{0,3\}$	$\emptyset$
3	$\{3\}$	$\{4\}$
4	$\{3\}$	$\emptyset$

$F = \{0,3\}$

S	$\delta''(S,a)$	$\delta''(S,b)$
$\{0\}$	$\{2,3\}$	$\{1,4\}$
$\{2,3\}$	$\{0,3\}$	$\{4\}$
$\{1,4\}$	$\{0,3\}$	$\emptyset$
$\{0,3\}$	$\{2,3\}$	$\{1,4\}$
$\{4\}$	$\{3\}$	$\emptyset$
$\emptyset$	$\emptyset$	$\emptyset$
$\{3\}$	$\{3\}$	$\{4\}$

$F = \{\{0\},\{2,3\},\{0,3\}\{3\}\}$

9 The Equivalence of DFAs and Regular Expressions

30. Give the formal definition of $\mathcal{A}$ if $\mathcal{A}_1$ and $\mathcal{A}_2$ are as in the previous case.

    (Solution)

    $$\mathcal{A} = \langle Q_1 \cup Q_2 \cup \{q_0, q_f\}, \Sigma, q_0, \delta, \{q_f\}\rangle$$

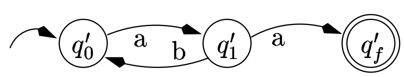
    where for all $q \in Q_1 \cup Q_2 \cup \{q_0, q_f\}$ and $x \in \Sigma \cup \{\varepsilon\}$:

    $$\delta(q,x) = \begin{cases} \delta_1(q,x) & \text{if } q \in Q_1 \setminus q'_f, \\ \delta_2(q,x) & \text{if } q \in Q_2 \setminus q''_f, \\ \delta_1(q,x) & \text{if } q = q'_f \text{ and } x \neq \varepsilon, \\ \delta_2(q,x) & \text{if } q = q''_f \text{ and } x \neq \varepsilon, \\ \delta_1(q,\varepsilon) \cup \{q_f\} & \text{if } q = q'_f \text{ and } x = \varepsilon, \\ \delta_2(q,\varepsilon) \cup \{q_f\} & \text{if } q = q''_f \text{ and } x = \varepsilon, \\ \{q'_0, q''_0\} & \text{if } q = q_0 \text{ and } x = \varepsilon. \end{cases}$$

31. Why does this construction not simply use the initial and final states of $\mathcal{A}_1$ as the initial and final states of $\mathcal{A}$, simply adding ε-transitions from q'_0 to q'_f and from q'_f to q'_0? Give an example of an automaton $\mathcal{A}_1$ for which the simpler construction fails. This example will *not* be an automaton which would be constructed from a regular expression. The simpler construction *will* work in the context of the proof, but the induction hypothesis needs to be strengthened slightly. How? Explain why this suffices.

    (Solution)

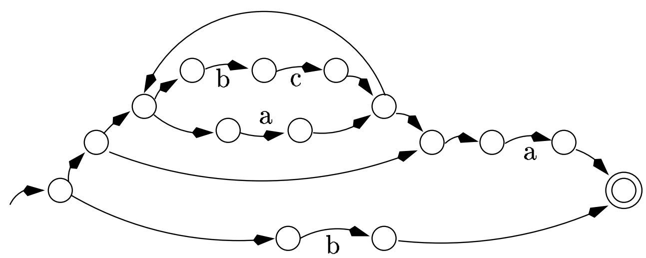
    With the simpler construction we cannot (in general) guarantee that every path from the initial state of the modified automaton to its final state will be the concatenation of zero or more such paths through the original automaton. If there is any w such that $q'_0 \in \hat{\delta}(q'_0, w)$ but $q'_f \notin \hat{\delta}(q'_0, w)$, i.e., a path that re-enters the start state which is labeled with a string that is not in $L(\mathcal{A}_1)$ then adding an ε-edge from q'_0 to q'_f will, effectively, add that string to language. For example, let $\mathcal{A}_1$ be:



Here $L(\mathcal{A}_1) = L((ab)^*aa)$. But the simpler construction, in adding an ε-transition from q_0' to q_f' effectively adds $L((ab)^*)$ to this. The result is that, rather than $L(((ab)^*aa)^*)$, $L(\mathcal{A})$ will be $L(((ab)^*(\varepsilon + aa))^*)$. Note that this would also fail if the edge from q_1' to q_0' were an ε-transition (we would get $L(a^*)$ rather than $L((aa)^*)$). The reason this would actually be harmless, given the rest of the construction, is that we never have any transitions into the initial state except for (in the modified construction) possibly an ε-transition from the final state. Thus an ε-transition from the initial to the final state will not add any strings other than ε to the language. To account for this, the IH could be strengthened to say something like: "has a single final state and for all strings $w \in \Sigma^+$ if $q_0 \in \hat{\delta}(q_0, w)$ then $w \in L(\mathcal{A})$."

32. Construct an NFA accepting $L((a + bc)^*a + b)$.

(Solution)



(All unlabeled transitions are ε-transitions.)

33. Show that if $P = \{\varepsilon\}$ then every set that contains Q as a subset is a solution to $R = Q + RP$.

(Solution)

Suppose $Q \subseteq R$. Then, $L(Q + R) = L(Q) \cup L(R) = L(R)$ and

$$R = Q + R\{\varepsilon\} = R$$

34. Give an example of an R, P and Q such that $R = Q + RP$ but $R \neq QP^*$.

(Solution)

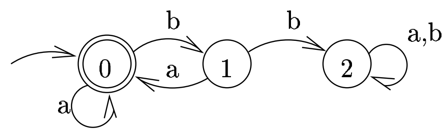
Let $R = \{a, b\}$, $Q = \{a\}$, and $P = \{\varepsilon\}$. Then

$$\{a, b\} = \{a\} + \{a, b\}\{\varepsilon\} = \{a, b\}$$

but

$$\{a\}\{\varepsilon\}^* = \{a\} \neq \{a, b\}.$$

35. Construct a regular expression denoting the language accepted by the following DFA.



(Solution)

$$\begin{aligned}
R_0 &= \varepsilon + R_0 a + R_1 a \\
R_1 &= R_0 b \\
R_2 &= R_1 b + R_2(a+b)
\end{aligned}$$

$$\begin{aligned}
R_0 &= \varepsilon + R_0 a + R_0 ba \\
&= \varepsilon + R_0(a + ba) \\
R_0 &= (a+ba)^* \\
R_1 &= (a+ba)^* b \\
R_2 &= (a+ba)^* bb(a+b)^*
\end{aligned}$$

Since $F = \{0\}$, $L(\mathcal{A}) = L((a+ba)^*)$. Note that there is actually no need to solve for either R_1 or R_2 in this case.