# Lecture #4: Regular Operations and Regular Expressions
# Proof of Equivalence Claim

This document includes a proof of the following result, which was included in the notes for Lecture #4.

**Theorem 2.** *Let $\Sigma$ be an alphabet that does not include any of the special symbols "$\lambda$", "$\emptyset$", "$\Sigma$", "(", ")", "$\cup$", "$\circ$", or "$^\star$" and let $L \subseteq \Sigma^\star$. Then $L$ is a regular language if and only if $L$ is the language of a regular expression over $\Sigma$.*

## An Easy Direction

One part of this claim is reasonably easy to establish using closure properties that were also stated in the notes for Lecture #4 and proved in another supplemental document for this lecture ("Proofs of Closure Properties").

**Lemma 1.** *Let $\Sigma$ be an alphabet and let $R$ be a regular expression over $\Sigma$. Then the language, $L(R)$, of the regular expression $R$ is a regular language.*

*Proof.* Since $R$ is a regular expression it is a *string* — over an alphabet that includes $\Sigma$ along with a small number of additional symbols. The result will be proved by induction on the length of the string $R$, using the strong form of mathematical induction.

Since the empty string is *not* a regular expression,[1] regular expressions with length $1$ will be considered in the basis.

*Basis:* Let $R$ be a regular expression over $\Sigma$ with length one. Then it follows, by the definition of a "regular expression", that one of the following cases must hold.

- $R$ is the symbol "$\sigma$" for some symbol $\sigma \in \Sigma$.

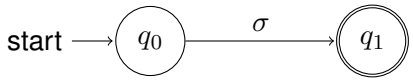
[1] The string "$\lambda$", which represents the empty string, is *not* the empty string, itself: It is a nonempty string with length $1$.

- $R$ is the symbol "$\lambda$" (which represents the empty string).
- $R$ is the symbol "$\emptyset$" (which represents the empty set).
- $R$ is the symbol "$\Sigma$" (which represents the alphabet $\Sigma$).

Each of these cases is considered separately below.

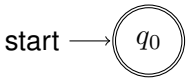
- *Case:* $R$ is the symbol "$\sigma$" for some symbol $\sigma \in \Sigma$.

  As observed in the lecture notes, $L(R) = L(\sigma) = \{\sigma\}$ — and this is a regular language, since it is the language of the nondeterministic finite automaton



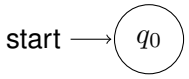

- *Case:* $R$ is the symbol "$\lambda$" (which represents the empty string).

  As observed in the lecture notes, $L(R) = L(\lambda) = \{\lambda\}$ — and this is a regular language, since it is the language of the nondeterministic finite automaton




- *Case:* $R$ is the symbol "$\emptyset$" (which represents the empty set).

  As observed in the lecture notes, $L(R) = L(\emptyset) = \emptyset$ — and this is a regular language, since it is the language of the nondeterministic finite automaton

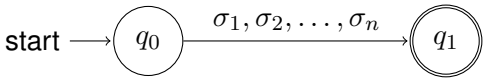



- *Case:* $R$ is the symbol "$\Sigma$" (which represents the alphabet $\Sigma$).

  Since $\Sigma$ is an alphabet,

$$\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$$

  for some positive integer $n$ and (distinct) symbols $\sigma_1, \sigma_2, \ldots, \sigma_n$. Then, as observed in the lecture notes, $L(R) = L(\Sigma) = \Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$ — and this is a regular language, since it is the language of the nondeterministic finite automaton



Thus if $R$ is a regular expression over $\Sigma$ whose length (as a string) is $1$ then $L(R)$ is a regular language, as claimed.

*Inductive Step:* Let $k$ be an integer such that $k \geq 1$. It is necessary and sufficient to use the following

> Inductive Hypothesis: Let $R$ be a regular expression over $\Sigma$ whose length (as a string) is $\ell$, for an integer $\ell$ such that $1 \leq \ell \leq k$. Then the language, $L(R)$, of $R$ is a regular language.

to prove the following

> Inductive Claim: Let $R$ be a regular expression over $\Sigma$ whose length (as a string) is $k+1$. Then the language, $L(R)$, of $R$ is a regular language.

With that noted, let $R$ be a regular expression over $R$ whose length (as a string) is $k+1$. Since $k \geq 1$, $k+1 \geq 2$, and it follows by the definition of a "regular expression" that one of the following cases must hold.

- $R$ is the string "$(R_1 \cup R_2)$" where $R_1$ and $R_2$ are regular expressions over $\Sigma$.
- $R$ is the string "$(R_1 \circ R_2)$" where $R_1$ and $R_2$ are regular expressions over $\Sigma$.
- $R$ is the string "$(R_1)^\star$" where $R_1$ is a regular expression over $\Sigma$.

Each of these cases is considered separately below.

- *Case:* $R$ is the string "$(R_1 \cup R_2)$" where $R_1$ and $R_2$ are regular expressions over $\Sigma$.

  Then the language, $L = L(R)$, of the regular expression $R$, is $L_1 \cup L_2$, where $L_1 \subseteq \Sigma^\star$ is the language of the regular expression $R_1$ and $L_2 \subseteq \Sigma^\star$ is the language of the regular expression $R_2$.

  Since $R_1$ and $R_2$ are regular expressions over $\Sigma$ they are each strings with length at least $1$. On the other hand, $R_1$ is a substring of $R$ which does not include at least four of the symbols in $R$, namely the left and right brackets, symbol "$\cup$" for union, and at least one symbol in $R_2$. Thus the length of $R_1$ is at most $(k+1) - 4 = k - 3$. Switching the roles of $R_1$ and $R_2$ one can argue that the length of $R_2$ is at most $k-3$ as well.

  Thus $R_1$ and $R_2$ are both strings over $\Sigma$ with lengths between $1$ and $k-3$, and it follows by the Inductive Hypothesis that the languages $L_1 = L(R_1) \subseteq \Sigma^\star$ and $L_2 = L(R_2) \subseteq \Sigma^\star$ are both regular languages.

  Theorem #1, part (a), from the lecture notes, now implies that $L = L_1 \cup L_2$ is a regular language too.

- *Case:* $R$ is the string "$(R_1 \circ R_2)$" where $R_1$ and $R_2$ are regular expressions over $\Sigma$.

  Then the language, $L = L(R)$, of the regular expression $R$, is $L_1 \circ L_2$, where $L_1 \subseteq \Sigma^\star$ is the language of the regular expression $R_1$ and $L_2 \subseteq \Sigma^\star$ is the language of the regular expression $R_2$.

  The argument given at the beginning of the previous case can be applied here (with the symbol for union, "$\cup$", replaced with the symbol for concatenation, "$\circ$") to argue that $R_1$ and $R_2$ are each regular expressions over $\Sigma$ with length between $1$ and $k - 3$. Once again, it follows by the Inductive Hypothesis that the languages $L_1 = L(R_1) \subseteq \Sigma^\star$ and $L_2 = L(R_2) \subseteq \Sigma^\star$ are both regular languages.

  Theorem #1, part (b), from the lecture notes, now implies that $L = L_1 \circ L_2$ is also a regular language.

- *Case:* $R$ is the string "$(R_1)^\star$" where $R_1$ is a regular expression over $\Sigma$.

  Then the language, $L = L(R)$, of the regular expression $R$ is $L_1^\star$, where $L_1 \subseteq \Sigma^\star$ is the language of the regular expression $R_1$.

  Since $R_1$ is a regular expression over $\Sigma$ it is a string with length at least $1$. One the other hand, $R_1$ is a substring of $R$ that does not include at least three symbols, namely the left and right brackets and the "star" symbol, "$\star$". Thus the length of $R_1$ is at most $(k + 1) - 3 = k - 2$.

  It now follows by the Inductive Hypothesis that the language $L_1 = L(R_1) \subseteq \Sigma^\star$ is a regular language.

  Theorem #1, part (c), from the lecture notes, now implies that $L = L_1^\star$ is a regular language, as well.

Thus $L = L(R)$ is a regular language in every case. Since $R$ is an arbitrarily chosen regular expression over $\Sigma$ with length $k + 1$, this establishes the Inductive Claim — as needed to complete the Inductive Step.

The claim now follows by induction on the length of the regular expression $R$. □

## A More Challenging Direction

It is somewhat more challenging to prove that every regular language $L \subseteq \Sigma^\star$ is also the language of a regular expression over $\Sigma$. In order to do this, yet another kind of finite state machine, called a ***generalized nondeterministic finite automaton***, will be introduced. Then properties of these machines will be stated, proved, and used to establish this result.

## Generalized Nondeterministic Finite Automata

A ***generalized nondeterministic finite automaton (GNFA)***

$$M = (Q, \Sigma, \delta, q_0, q_{\text{accept}})$$

is yet another kind of "finite state machine:"

- As usual, $\Sigma$ is the machine's ***alphabet***, and the machine processes strings in $\Sigma^\star$.
- As usual, $M$ has a finite set $Q$ of ***states*** — which includes a ***start state*** $q_0$.
- $Q$ also includes a single ***accepting state*** $q_{\text{accept}} \in Q$, which is different from $q_0$.
- For every state $q \in Q \setminus \{q_{\text{accept}}\}$ (that is, for each state except $q_{\text{accept}}$) and for every state $r \in Q \setminus \{q_0\}$ (that is, for every state except $q_0$) there is a transition from $q$ to $r$ that is labelled by some ***regular expression*** $R_{q,r}$ over the alphabet $\Sigma$.

  In other words, the ***transition function*** is a total function

  $$\delta : (Q \setminus \{q_{\text{accept}}\}) \times (Q \setminus \{q_0\}) \to \mathcal{R}_\Sigma$$

  where $\mathcal{R}_\Sigma$ is the set of regular expressions over the alphabet $\Sigma$.

The GNFA $M$ ***accepts*** a string $\omega \in \Sigma$ if and only if there is a sequence

$$r_0, r_1, r_2, \ldots, r_m$$

of the states in $Q$ such that $r_0 = q_0$, $r_1, r_2, \ldots, r_{m-1} \in Q \setminus \{q_0, q_{\text{accept}}\}$, $r_m = q_{\text{accept}}$, and

$$\omega = \omega_0\omega_1 \ldots, \omega_{m-1}$$

where $\omega_i$ is in the language of the regular expression $R_{i,i+1} = \delta(r_i, r_{i+1})$ labelling the transition from $r_i$ to $r_{i+1}$, for $0 \leq i \leq m-1$.

The ***language*** $L(M)$ of a GNFA $M$ is a subset of $\Sigma^\star$, namely, the set of strings $\omega \in \Sigma^\star$ such that $M$ accepts $\omega$ (as defined above).

**Lemma 2.** *Let $L \subseteq \Sigma^\star$, for an alphabet $\Sigma$. If $L$ is a regular language then there exists a generalized nondeterministic finite automaton $M = (Q, \Sigma, \delta, q_0, q_{\text{accept}})$, with alphabet $\Sigma$, such that $L = L(M)$.*

*Sketch of Proof.* Let $L \subseteq \Sigma^\star$, for an alphabet $\Sigma$, such that $L$ is a regular language. Then — as established (as "Lemma 2") in the supplemental document, "Proofs of Closure Properties", $L = L(\widehat{M})$ for some nondeterministic finite automaton

$$\widehat{M} = (Q, \Sigma, \widehat{\delta}, q_0, F)$$

which satisfies the following additional properties.

- There are no transitions into $q_0$ at all. That is, $q_0 \notin \widehat{\delta}(q, \sigma)$ for any state $q \in Q$ or any symbol $\sigma \in \Sigma_\lambda$, so that the only string $\omega \in \Sigma^\star$ such that $q_0 \in \widehat{\delta^\star}(q_0, \omega)$ is the empty string, $\omega = \lambda$.
- $\widehat{M}$ has exactly one accepting state, $q_{\text{accept}}$, and there are no transitions out of this state. That is, $F = \{q_{\text{accept}}\}$ and $\widehat{\delta}(q_{\text{accept}}, \sigma) = \emptyset$ for every symbol $\sigma \in \Sigma_\lambda$.

As the above notation may suggest, let us consider a generalized nondeterministic finite automaton

$$M = (Q, \Sigma, \delta, q_0, q_{\text{accept}})$$

with the same alphabet $\Sigma$ and the same set $Q$ of states as the nondeterministic finite automaton, $M$, has. $M$'s start state is the same state, $q_0$, as for $\widehat{M}$, and $M$'s accepting state is the state $q_{\text{accept}} \in Q$ that belongs to $\widehat{M}$'s set, $F$, of accepting states.

In order to define the transition function $\delta : (Q \setminus \{q_{\text{accept}}\}) \times (Q \setminus \{q_0\}) \to \mathcal{R}_\Sigma$, let $q, r \in Q$ such that $q \neq q_{\text{accept}}$ and $r \neq q_0$, and let

$$S_{q,r} \subseteq \Sigma_\lambda$$

be the set of symbols $\sigma \in \Sigma_\lambda$ such that $r \in \delta(q, \sigma)$. Let $R_{q,r}$ be the regular expression over $\Sigma$ that is defined as follows.

- If $S_{q,r} = \emptyset$ then $R_{q,r}$ is the regular expression "$\emptyset$".
- If $|S_{q,r}| = 1$, so that $S_{q,r} = \{\sigma\}$ for some symbol $\sigma \in \Sigma_\lambda$, then $R_{q,r}$ is the regular expression "$\sigma$".
- Otherwise $|S_{q,r}| = k$ for some integer $k$ such that $k \geq 2$. Suppose that

  $$S_{q,r} = \{\alpha_1, \alpha_2, \ldots, \alpha_k\}$$

  where $\alpha_1, \alpha_2, \ldots, \alpha_k \in \Sigma_\lambda$. Let $r_2$ be the regular expression "$(\alpha_1 \cup \alpha_2)$", so that $L(r_2) = \{\alpha_1, \alpha_2\}$, and, for $2 \leq \ell \leq k-1$, let $r_{\ell+1}$ be the regular expression[2]

  $$(r_\ell \cup \alpha_{\ell+1}).$$

  Then it is easily shown by induction on $\ell$ that, for $2 \leq \ell \leq k$, $r_\ell$ is a regular expression whose language is the set

  $$\{\alpha_1, \alpha_2, \ldots . \alpha_\ell\}.$$

  Set $R_{q,r}$ to be the regular expression $r_k$, so that the language of $R_{q,r}$ is the set $S_{q,r}$.

[2] For example, if $k \geq 3$ then $r_3$ is the regular expression "$((\alpha_1 \cup \alpha_2) \cup \alpha_3)$" and if $k \geq 4$ then $r_4$ is the regular expression "$(((\alpha_1 \cup \alpha_2) \cup \alpha_3) \cup \alpha_4)$".

Now set $\delta(q, r)$ to be the regular expression $R_{q,r}$ for each pair of states $q \in Q \setminus \{q_{\text{accept}}\}$ and $r \in Q \setminus \{q_0\}$. Then $\delta$ is a total function from $(Q \setminus \{q_{\text{accept}}\}) \times (Q \setminus \{q_0\})$ to $\mathcal{R}_\Sigma$, as needed to complete the definition of the generalized nondeterministic finite automaton $M$.

The following is now easily established by induction on $\ell$: For every string $\omega \in \Sigma^\star$, for every positive integer $\ell$, and for every sequence of states

$$r_0, r_1, r_2, \ldots, r_\ell$$

in $Q$, where $r_0 = q_0$, $r_i \in Q \setminus \{q_0, q_{\text{accept}}\}$ for $1 \leq i \leq \ell - 1$, and $r_\ell \in Q \setminus \{q_0\}$, the following conditions are equivalent:

(a) There exists a sequence of strings $\omega_1, \omega_2, \ldots, \omega_\ell \in \Sigma^\star$ such that $\omega_i \in L(R_{r_{i-1}, r_i}) = L(\delta(r_{i-1}, r_i))$ for $1 \leq i \leq \ell$, and such that

$$\omega = \omega_1 \cdot \omega_2 \ldots \omega_\ell.$$

(b) There exists a sequence of strings (with length at most one) $\omega_1, \omega_2, \ldots, \omega_\ell \in \Sigma_\lambda$ such that $r_i \in \widehat{\delta}(r_{i-1}, \omega_i)$ for $1 \leq i \leq \ell$, and such that

$$\omega = \omega_1 \cdot \omega_2 \ldots \omega_\ell.$$

Once this result has been established it can be applied to show that $M$ accepts $\omega$ if and only if $\widehat{M}$ accepts $\omega$. Thus $L = L(\widehat{M}) = L(M)$, so that $L$ is the language of a generalized nondeterministic finite automaton, as needed to establish the claim. □

### State Reduction

Suppose, now, that $\Sigma$ is an alphabet,

$$M = (Q, \Sigma, \delta, q_0, q_{\text{accept}})$$

is a generalized nondeterministic finite automaton such that $|Q| = k \geq 3$, and let $q \in Q$ such that $q \neq q_0$ and $q \neq q_{\text{accept}}$. Let $\widehat{Q} = Q \setminus \{q\}$, so that $\widehat{Q}$ includes $k - 1$ states, and let

$$\widehat{M} = (\widehat{Q}, \Sigma, \widehat{\delta}, q_0, q_{\text{accept}})$$

where $\widehat{\delta}$ is a total function from $(\widehat{Q} \setminus \{q_{\text{accept}}\}) \times (\widehat{Q} \setminus \{q_0\})$ to $\mathcal{R}_\Sigma$ such that, for every pair of states $r, s \in \widehat{Q}$ such that $r \neq q_{\text{accept}}$ and $s \neq q_0$, $\widehat{\delta}(r, s)$ is the regular expression

$$(\delta(r, s) \cup ((\delta(r, q) \circ (\delta(q, q))^\star) \circ \delta(q, s))). \tag{1}$$

Notice that if $\omega \in \Sigma^\star$ such that $\omega$ is in the language of the above regular expression then either $\omega$ is in the language of the regular expression $\delta(r, s)$ — so that one can go directly from state $r$ to state $s$ in $M$ by processing the string $\omega$ — or $\omega$ is in the language of the regular expression

$$((\delta(r, q) \circ (\delta(q, q))^\star) \circ \delta(q, s))$$

— so that one can go indirectly from state $r$ to state $s$ in $M$ by processing a prefix of $\omega$ to apply a transition from state $r$ to state $q$; processing zero or more substrings of $\omega$ to apply transitions from state $q$ to itself; and then processing a suffix of $\omega$ to apply a transition from $q$ to $s$.

**Lemma 3.** *Let $\Sigma$, $M$, and $\widehat{M}$ be as above. The following property is satisfied for every positive integer $k$: For every sequence*

$$q_0 = r_0, r_1, r_2, \ldots, r_k$$

*of states such that $r_i \in Q \setminus \{q_0, q_{\text{accept}}\}$ for $1 \leq i \leq k-1$ and $r_k \in Q \setminus \{q_0, q\}$, and for every string $\omega \in \Sigma^\star$, if there exist strings $\omega_1, \omega_2, \ldots, \omega_k \in \Sigma^\star$ such that $\omega_i \in L(\delta(r_{i-1}, r_i))$ for $1 \leq i \leq k$ and*

$$\omega = \omega_1 \cdot \omega_2 \ldots \omega_k$$

*(so that $r_k$ can be reached from $q_0$ in $M$ by processing $\omega$) then there exists a positive integer $\ell$ such that $\ell \leq k$ and a sequence*

$$q_0 = \widehat{r}_0, \widehat{r}_1, \ldots . \widehat{r}_\ell = r_k$$

*of states such that $r_i \in \widehat{Q} \setminus \{q_0, q_{\text{accept}}\}$ for $1 \leq i \leq \ell - 1$, as well as strings $\widehat{\omega}_1, \widehat{\omega}_2, \ldots, \widehat{\omega}_\ell \in \Sigma^\star$, such that $\widehat{\omega}_i \in L(\widehat{\delta}(\widehat{r}_{i-1}, \widehat{r}_i))$ for $1 \leq i \leq \ell$ and*

$$\omega = \widehat{\omega}_1 \cdot \widehat{\omega}_2 \ldots \widehat{\omega}_\ell$$

*(so that $r_k = \widehat{r}_\ell$ can be reached from $q_0$ in $\widehat{M}$ by processing $\omega$, as well).*

*Sketch of Proof.* This can be proved by induction on $k$, using the strong form of mathematical induction.

*Basis:* Suppose that $k = 1$. Then it is necessary and sufficient to consider a sequence

$$q_0 = r_0, r_1$$

where $r_1 \in Q \setminus \{q_0, q\}$ and a string $\omega \in \Sigma^\star$ such that $\omega \in L(\delta(r_0, r_1))$ — because it must be true that $\omega_1 = \omega$ in case.

Now, since $\widehat{\delta}(r_0, r_1)$ is the regular expression

$$(\delta(r_0, r_1) \cup ((\delta(r_0, q) \circ (\delta(q, q))^\star) \circ \delta(q, r_1))),$$

and $\omega \in L(\delta(r_0, r_1))$, it is certainly the case that $\omega \in L(\widehat{\delta}(r_0, r_1))$ as well. One can therefore set $\ell$ to be $1$ (so that $\ell \leq k$), so that $\widehat{r}_1 = r_1$ and $\widehat{\omega}_1 = \omega$ in order to ensure that required conditions are all satisfied — as needed to complete the basis.

*Inductive Step:* Let $h$ be an integer such that $h \geq 1$.[3] It is necessary and sufficient to use the following

> Inductive Hypothesis: For every integer $m$ such that $1 \leq m \leq h$, for every sequence
>
> $$q_0 = r_0, r_1, \ldots, r_m$$
>
> of states such that $r_i \in Q \setminus \{q_0, q_{\mathsf{accept}}\}$ for $1 \leq i \leq m-1$ and $r_m \in Q \setminus \{q_0, q\}$, and for every string $\omega \in \Sigma^\star$, if there exist strings $\omega_1, \omega_2, \ldots, \omega_m \in \Sigma^\star$ such that $\omega_i \in L(\delta(r_{i-1}, r_i))$ for $1 \leq i \leq m$ and
>
> $$\omega = \omega_1 \cdot \omega_2 \ldots \omega_m$$
>
> (so that $r_m$ can be reached from $q_0$ in $M$ by processing $\omega$) then there exists a positive integer $\ell$ such that $\ell \leq m$ and a sequence
>
> $$q_0 = \widehat{r}_0, \widehat{r}_1, \ldots, \widehat{r}_\ell = r_m$$
>
> of states such that $r_i \in \widehat{Q} \setminus \{q_0, q_{\mathsf{accept}}\}$ for $1 \leq i \leq \ell - 1$, as well as strings $\widehat{\omega}_1, \widehat{\omega}_2, \ldots, \widehat{\omega}_\ell \in \Sigma^\star$, such that $\widehat{\omega}_i \in L(\widehat{\delta}(\widehat{r}_{i-1}, \widehat{r}_i))$ for $1 \leq i \leq \ell$ and
>
> $$\omega = \widehat{\omega}_1 \cdot \widehat{\omega}_2 \ldots \widehat{\omega}_\ell$$
>
> (so that $r_m = \widehat{r}_\ell$ can be reached from $q_0$ in $\widehat{M}$ by processing $\omega$, as well).

to prove the following

> Inductive Claim: For every sequence
>
> $$q_0 = r_0, r_1, \ldots, r_{h+1}$$
>
> of states such that $r_i \in Q \setminus \{q_0, q_{\mathsf{accept}}\}$ for $1 \leq i \leq h$ and $r_{h+1} \in Q \setminus \{q_0, q\}$, and for every string $\omega \in \Sigma^\star$, if there exist strings $\omega_1, \omega_2, \ldots, \omega_{h+1} \in \Sigma^\star$ such that $\omega_i \in L(\delta(r_{i-1}, r_i))$ for $1 \leq i \leq h+1$ and
>
> $$\omega = \omega_1 \cdot \omega_2 \ldots \omega_{h+1}$$
>
> (so that $r_{h+1}$ can be reached from $q_0$ in $M$ by processing $\omega$) then there exists a positive integer $\ell$ such that $\ell \leq h+1$ and a sequence
>
> $$q_0 = \widehat{r}_0, \widehat{r}_1, \ldots, \widehat{r}_\ell = r_{h+1}$$

[3] The name "$k$" is already being used, so the name "$h$" will be used here instead.

of states such that $r_i \in \widehat{Q} \setminus \{q_0, q_{\text{accept}}\}$ for $1 \leq i \leq \ell - 1$, as well as strings $\widehat{\omega}_1, \widehat{\omega}_2, \ldots, \widehat{\omega}_\ell \in \Sigma^\star$, such that $\widehat{\omega}_i \in L(\widehat{\delta}(\widehat{r}_{i-1}, \widehat{r}_i))$ for $1 \leq i \leq \ell$ and

$$\omega = \widehat{\omega}_1 \cdot \widehat{\omega}_2 \ldots \widehat{\omega}_\ell$$

(so that $r_{h+1} = \widehat{r}_\ell$ can be reached from $q_0$ in $\widehat{M}$ by processing $\omega$, as well).

With that noted, consider a sequence

$$q_0 = r_0, r_1, \ldots, r_{h+1}$$

of states such that $r_i \in Q \setminus \{q_0, q_{\text{accept}}\}$ for $1 \leq i \leq h$ and $r_{h+1} \in Q \setminus \{q_0, q\}$, and a string $\omega \in \Sigma^\star$, such that there exist strings $\omega_1, \omega_2, \ldots, \omega_{h+1} \in \Sigma^\star$ such that $\omega_i \in L(\delta(r_{i-1}, r_i))$ for $1 \leq i \leq h+1$ and

$$\omega = \omega_1 \cdot \omega_2 \ldots \omega_{h+1}$$

(so that $r_{h+1}$ can be reached from $q_0$ in $M$ by processing $\omega$).

Let $m$ be the largest integer such that $0 \leq m \leq h$ and $r_m \neq q$. Then either $m = 0$, $1 \leq m \leq h - 1$, or $m = h$. These cases are considered separately below.

- *Case:* $m = 0$. In this case $r_i = q$ for every integer $i$ such that $1 \leq i \leq h$, so that $\omega_1 \in L(\delta(r_0, r_1)) = \delta(q_0, q)$, $\omega_i \in L(\delta(r_{i-1}, r_i)) = L(\delta(q, q))$ for every integer $i$ such that $2 \leq i \leq h$, and $\omega_{h+1} \in L(\delta(r_h, r_{h+1})) = L(\delta(q, r_{h+1}))$. It follows that is in the language of the regular expression

  $$((\delta(q_0, q) \circ (\delta(q, q))^\star) \circ \delta(q, r_{h+1}))$$

  so that $\omega$ is in the language of the regular expression

  $$\widehat{\delta}(q_0, r_{h+1}) = (\delta(q_0, r_{h+1}) \cup ((\delta(q_0, q) \circ (\delta(q, q))^\star) \circ \delta(q, r_{h+1})))$$

  as well. Now let $\ell = 1$ and consider the the sequence

  $$q_0 = \widehat{r}_0, \widehat{r}_1 = \widehat{r}_\ell = r_{h+1},$$

  and the string $\widehat{\omega}_1 = \omega$. It follows from the above that

  $$\widehat{\omega}_1 = \omega \in L(\widehat{\delta}(q_0, r_{h+1})) = L(\widehat{\delta}(\widehat{r}_0, \widehat{r}_1))$$

  and $\widehat{\omega}_1 = \omega$, so that this choice of $\ell$, sequence of states $\widehat{r}_0, \widehat{r}_1$, and string $\widehat{\omega}_1$ satisfy the conditions in the Inductive Claim in this case.

- *Case:* $1 \leq m \leq h-1$. Let

$$\omega_L = \omega_1 \cdot \omega_2 \ldots \omega_m \quad \text{and} \quad \omega_R = \omega_{m+1}\omega_{m+2}\ldots\omega_{h+1}$$

so that $\omega_L, \omega_R \in \Sigma^\star$ and $\omega = \omega_L \cdot \omega_R$. Now $m$ is an integer such that $1 \leq m \leq h$,

$$q_0 = r_0, r_1, \ldots, r_m$$

is a sequence of states such that $r_i \in Q \setminus \{q_0, q_{\text{accept}}\}$ for $1 \leq i \leq m-1$ and $r_m \in Q \setminus \{q_0, q\}$, $\omega_L \in \Sigma^\star$, and $\omega_1, \omega_2, \ldots, \omega_m$ are strings in $\Sigma^\star$ such that $\omega_i \in L(\delta(r_{i-1}, r_i))$ for $1 \leq i \leq m$ and

$$\omega_L = \omega_1 \cdot \omega_2 \ldots \omega_m$$

(so that $r_m$ can be reached from $q_0$ in $M$ by processing $\omega_L$). It now follows by the Inductive Hypothesis that there exists a positive integer $\ell$ such that $\ell \leq m$ and a sequence

$$q_0 = \widehat{r}_0, \widehat{r}_1, \ldots, \widehat{r}_\ell = r_m$$

of states such that $r_i \in \widehat{Q} \setminus \{q_0, q_{\text{accept}}\}$ for $1 \leq i \leq \ell-1$, as well as strings $\widehat{\omega}_1, \widehat{\omega}_2, \ldots, \widehat{\omega}_\ell \in \Sigma^\star$, such that $\widehat{\omega}_i \in L(\widehat{\delta}(\widehat{r}_{i-1}, \widehat{r}_i))$ for $1 \leq i \leq \ell$ and

$$\omega_L = \widehat{\omega}_1 \cdot \widehat{\omega}_2 \ldots \widehat{\omega}_\ell$$

(so that $r_m = \widehat{r}_\ell$ can be reached from $q_0$ in $\widehat{M}$ by processing $\omega_L$, as well).

Since $\ell \leq m \leq h-1$, $\ell+1 \leq h$ and it follows by the definition of $m$ that $r_{m+1} = q$. Indeed, $r_i = q$ for every integer $i$ such that $m+1 \leq i \leq h$, so that $\omega_{m+1} \in L(\delta(r_m, r_{m+1})) = \delta(r_m, q)$, $\omega_i \in L(\delta(r_{i-1}, r_i)) = L(\delta(q, q))$ for every integer $i$ such that $m+2 \leq i \leq h$, and $\omega_{h+1} \in L(\delta(r_h, r_{h+1})) = L(\delta(q, r_{h+1}))$. Since $\omega_R = \omega_{m+1} \cdot \omega_{m+2} \ldots \omega_{h+1}$, it follows that $\omega_R$ is in the language of the regular expression

$$((\delta(r_m, q) \circ (\delta(q, q))^\star) \circ \delta(q, r_{h+1}))$$

so that it is certainly in the language of the regular expression

$$\widehat{\delta}(r_m, r_{h+1}) = (\delta(r_m, r_{h+1}) \cup ((\delta(r_m, q) \circ (\delta(q, q))^\star) \circ \delta(q, r_{h+1}))).$$

Thus, since $1 \leq \ell + 1 \leq m + 1 \leq h$, the integer $\ell + 1$, the sequence of states

$$\widehat{r}_0, \widehat{r}_1, \ldots, \widehat{r}_\ell = r_m, \widetilde{r}_{\ell+1} = r_{h+1}$$

and sequence of strings

$$\widehat{\omega}_1, \widehat{\omega}_2, \ldots, \widehat{\omega}_\ell, \widetilde{\omega}_{\ell+1} = \omega_R$$

satisfy the conditions in the Inductive Claim in this case.

- *Case:* $m = h$. Let

$$\omega_L = \omega_1 \cdot \omega_2 \ldots \omega_h \quad \text{and} \quad \omega_R = \omega_{h+1}$$

so that $\omega_L, \omega_R \in \Sigma^\star$ and $\omega = \omega_L \cdot \omega_R$. Now $h$ is certainly an integer such that $1 \leq m \leq h$, and it follows by the Inductive Hypothesis that there exists a positive integer $\ell$ such that $\ell \leq h$ and a sequence

$$q_0 = \widehat{r}_0, \widehat{r}_1, \ldots, \widehat{r}_\ell = r_h$$

of states such that $r_i \in \widehat{Q} \backslash \{q_0, q_{\text{accept}}\}$ for $1 \leq i \leq \ell - 1$, as well as strings $\widehat{\omega}_1, \widehat{\omega}_2, \ldots, \widehat{\omega}_\ell \in \Sigma^\star$, such that $\widehat{\omega}_i \in L(\widehat{\delta}(\widehat{r}_{i-1}, \widehat{r}_i))$ for $1 \leq i \leq \ell$ and

$$\omega_L = \widehat{\omega}_1 \cdot \widehat{\omega}_2 \ldots \widehat{\omega}_\ell$$

(so that $r_h = \widehat{r}_\ell$ can be reached from $q_0$ in $\widehat{M}$ by processing $\omega_L$, as well).

Since $\omega_R = \omega_{h+1}$, $\omega_R \in L(\delta(r_h, r_{h+1}))$, so that $\omega_R$ is certainly in the language of the regular expression

$$\widehat{\delta}(r_h, r_{h+1}) = (\delta(r_h, r_{h+1}) \cup ((\delta(r_h, q) \circ (\delta(q, q))^\star) \circ \delta(q, r_{h+1}))).$$

Thus the integer $\ell + 1$, the sequence of states

$$\widehat{r}_0, \widehat{r}_1, \ldots, \widehat{r}_\ell = r_m = r_h, \widetilde{r}_{\ell+1} = r_{h+1}$$

and sequence of strings

$$\widehat{\omega}_1, \widehat{\omega}_2, \ldots, \widehat{\omega}_\ell, \widetilde{\omega}_{\ell+1} = \omega_R = \omega_{h+1}$$

satisfy the conditions in the Inductive Claim in this case too.

Since the conditions in the Inductive Claim are established in every case, this establishes the Inductive Claim, as required to complete the Inductive Step. The claim now follows by induction on $k$. □

**Lemma 4.** *Let $\Sigma$, $M$, and $\widehat{M}$ be as above. The following property is satisfied for every positive integer $k$: For every sequence*

$$q_0 = \widehat{r}_0, \widehat{r}_1, \widehat{r}_2, \ldots, \widehat{r}_k$$

*of states such that $\widehat{r}_i \in \widehat{Q} \setminus \{q_0, q_{\text{accept}}\}$ for $1 \leq i \leq k-1$ and $\widehat{r}_k \in \widehat{Q} \setminus \{q_0\}$, and for every string $\omega \in \Sigma^\star$, if there exist strings $\widehat{\omega}_1, \widehat{\omega}_2, \ldots, \widehat{\omega}_k \in \Sigma^\star$ such that $\widehat{\omega}_i \in L(\widehat{\delta}(\widehat{r}_{i-1}, \widehat{r}_i))$ for $1 \leq i \leq k$ and*

$$\omega = \widehat{\omega}_1 \cdot \widehat{\omega}_2 \ldots \widehat{\omega}_k$$

*(so that $\widehat{r}_k$ can be reached from $q_0$ in $\widehat{M}$ by processing $\omega$) then there exists a positive integer $\ell$ such that $\ell \geq k$ and a sequence*

$$q_0 = r_0, r_1, \ldots . r_\ell = \widehat{r}_k$$

*of states such that* $r_i \in Q \setminus \{q_0, q_{\text{accept}}\}$ *for* $1 \leq i \leq \ell - 1$*, as well as strings* $\omega_1, \omega_2, \ldots, \omega_\ell \in \Sigma^\star$*, such that* $\omega_i \in L(\delta(r_{i-1}, r_i))$ *for* $1 \leq i \leq \ell$ *and*

$$\omega = \omega_1 \cdot \omega_2 \ldots \omega_\ell$$

*(so that* $\widehat{r}_k = r_\ell$ *can be reached from* $q_0$ *in* $M$ *by processing* $\omega$*, as well).*

The proof of Lemma 4 is simpler, and shorter, than the proof of Lemma 3 — the result can be proved using induction on $k$, with the *standard* form of mathematical induction. Proving this lemma is left as an ***exercise***.

**Lemma 5.** *Let* $\Sigma$*,* $M$ *and* $\widehat{M}$ *be as above. Then* $\widehat{M}$ *is a generalized nondeterministic finite automaton, with one fewer state than* $M$*, such that* $L(\widehat{M}) = L(M)$*.*

*Proof.* Let $L$, $M$, and $\widehat{M}$ be as above. Then $\widehat{M}$ is a generalized nondeterministic finite automaton with one fewer state than $M$. It is therefore necessary and sufficient to show both that $L(\widehat{M}) \subseteq L(M)$ and $L(M) \subseteq L(\widehat{M})$ in order to establish the claim.

In order to show that $L(\widehat{M}) \subseteq L(M)$, let $\omega \in \Sigma^\star$ such that $L(\widehat{M})$. Then it follows by the definition of acceptance of strings, by GNFAs, that there exists a positive integer $k$, a sequence of states

$$q_0 = \widehat{r}_0, \widehat{r}_1, \ldots, \widehat{r}_k = q_{\text{accept}}$$

such that $\widehat{r}_i \in Q \setminus \{q_0, q_{\text{accept}}\}$ for $1 \leq i \leq k-1$, and a sequence of strings $\widehat{\omega}_1, \widehat{\omega}_2, \ldots, \widehat{\omega}_k \in \Sigma^\star$ such that $\widehat{\omega}_i \in L(\widehat{\delta}(\widehat{r}_{i-1}, \widehat{r}_i))$ for $1 \leq i \leq k$ and

$$\omega = \widehat{\omega}_1 \cdot \widehat{\omega}_2 \ldots \widehat{\omega}_k$$

(so that $q_{\text{accept}}$ can be reached from $q_0$, in $\widehat{M}$, by processing $\omega$).

It now follows by Lemma 4 that $q_{\text{accept}}$ can be reached from $q_0$ in $M$ by processing $\omega$, as well. That is, $\omega \in L(M)$.

Since $\omega$ was arbitrarily chosen from $L(\widehat{M})$ it follows that $L(\widehat{M}) \subseteq L(M)$.

It can be shown that $L(M) \subseteq L(\widehat{M})$ using essentially the same argument, with Lemma 3 used instead of Lemma 4.

Thus $L(\widehat{M}) = L(M)$, as required to establish the claim. □

**Lemma 6.** *Let* $\Sigma$ *be an alphabet and let* $M = (Q, \Sigma, \delta, q_0, q_{\text{accept}})$*. Then there exists a generalized nondeterministic finite automaton* $\widehat{M} = (\widehat{Q}, \Sigma, \widehat{\delta}, \widehat{q}_0, \widehat{q}_{\text{accept}})$ *such that* $|\widehat{Q}| = 2$ *(so that* $\widehat{Q} = \{\widehat{q}_0, \widehat{q}_{\text{accept}}\}$*) and* $L(\widehat{M}) = L(M)$*.*

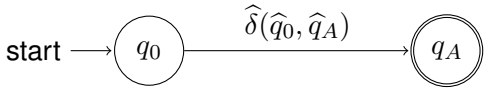
Note that if $\Sigma$ and $M$ are as in the claim then, since $q_0 \in Q$, $q_{\text{accept}} \in Q$ and $q_0 \neq q_{\text{accept}}$, $|Q| \geq 2$. The result can be proved using induction on $|Q|$, using the standard form of mathematical induction — considering the case that $|Q| = 2$ in the basis, and using Lemma 5 to establish what is needed for the inductive step. Writing this proof is left as another ***exercise***.

### Establishing the Desired Result

**Lemma 7.** *Let $\Sigma$ and let $L \subseteq \Sigma^\star$ be a regular language. Then there exists a regular expression $R$ over $\Sigma$ such that $L$ is the language, $L(R)$, of $R$.*

*Proof.* Let $\Sigma$ be an alphabet and let $L \subseteq \Sigma^\star$ such that $L$ is a regular language.

Then it follows by Lemma 2 that there exists a generalized nondeterministic finite automaton $M = (Q, \Sigma, \delta, q_0, q_{\text{accept}})$, with alphabet $\Sigma$ such that $L = L(M)$. Lemma 6 now implies that there is a generalized nondeterministic finite automaton $\widehat{M} = (\widehat{Q}, \Sigma, \widehat{\delta}, \widehat{q}_0, \widehat{q}_{\text{accept}})$ such that $\widehat{Q} = \{\widehat{q}_0, \widehat{q}_{\text{accept}}\}$ and $L = L(\widehat{M})$ as well. The generalized finite automaton $\widehat{M}$ has the form




— where the accepting state, $q_{\text{accept}}$, has been shown as "$q_A$" just to make the picture a little bit simpler. Since only a single transition, from the start state to the accepting state, can be followed when processing a string, a consideration of the definition of "acceptance of a string by a generalized nondeterministic finite automaton" is sufficient to see that one simply needs to set $R$ to be the regular expression $\widehat{\delta}(q_0, q_{\text{accept}})$ in order to ensure that $L = L(R)$, and establish the claim. □

## Finishing the Proof

Theorem 2 now follows from Lemmas 1 and 7.