Computer Science 351

Equivalence of Deterministic Finite Automata and Nondeterministic Finite Automata

Instructor: Wayne Eberly

Department of Computer Science
University of Calgary

Lecture #6

Goal for Today

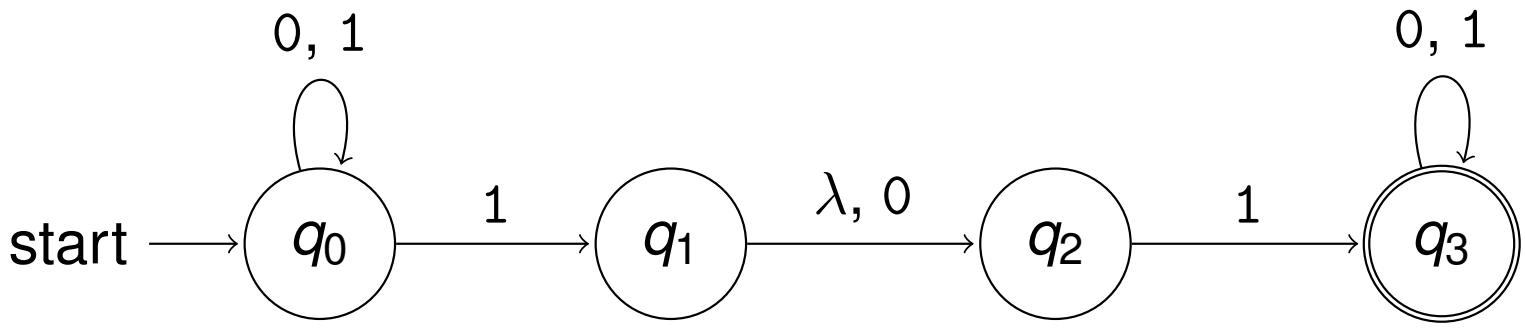
Goal for Today:

- *Prove* that every language $L \subseteq \Sigma^\star$ is the language of an NFA *if and only if* it is the language of a DFA.
- A construction to *convert* an NFA into a DFA with the same language will be provided in order to complete the "hard part" of this proof.

  This introduces an important technique — the ***simulation*** of one computational device by another.

Ongoing Example

A DFA will be designed that has the same language (a language in $\Sigma^\star$, for $\Sigma = \{0, 1\}$) as the first NFA used as an example in the previous lecture:



Ongoing Example

Note: For every string $\omega = \Sigma^\star$...

- $q_0 \in \delta^\star(q_0, \omega)$.
- $q_1 \in \delta^\star(q_0, \omega)$ if and only if ω ends with a `1`.
- $q_2 \in \delta^\star(q_0, \omega)$ if and only if ω ends with either `1` or `10`.
- $q_3 \in \delta^\star(q_0, \omega)$ if and only if either `11` or `101` is a substring of ω — that is, if and only if either $\omega = \mu\texttt{11}\nu$ or $\omega = \mu\texttt{101}\nu$ for strings $\mu, \nu \in \Sigma^\star$ (or both).

Since $F = \{q_3\}$, it follows that the language of this NFA is the set of strings $\omega \in \Sigma^\star$ such that either `11` or `101` (or both) is a substring of ω.

The Easy Part of the Proof

Suppose you are given a DFA $M = (Q, \Sigma, \delta, q_0, F)$ for a language $L \subseteq \Sigma^\star$.

Goal: Describe an NFA $\widehat{M} = (Q, \Sigma, \widehat{\delta}, q_0, F)$ so that transitions are used to remember which state M would be in after processing a string — that is, find a transition function $\widehat{\delta}$ for $\widehat{M}$ such that — for every state $q \in Q$ and for every string $\omega \in \Sigma^\star$ —

$$\widehat{\delta}^\star(q, \omega) = \{\delta^\star(q, \omega)\}. \tag{1}$$

Note: If the condition at line (1) is satisfied then it follows that

$$L(\widehat{M}) = L(M).$$

Since M is an arbitrarily chosen DFA it would follow that ***every regular language is the language of an NFA.***

The Easy Part of the Proof

Now suppose that we define the transition function

$$\widehat{\delta} : Q \times \Sigma_\lambda \to \mathcal{P}(Q)$$

as follows: For every state $q \in Q$ and for every state $\sigma \in \Sigma$,

$$\widehat{\delta}(q, \sigma) = \{\delta(q, \sigma)\}$$

and (again, for every state $q \in Q$)

$$\widehat{\delta}(q, \lambda) = \emptyset.$$

The Easy Part of the Proof

- Then $\widehat{\delta}$ is a well-defined total function from $Q \times \Sigma_\lambda$ to $\mathcal{P}(Q)$ (and an NFA with alphabet Σ has now been defined).
- It can be proved, by induction on the length of a string ω, that

$$\widehat{\delta^\star}(q, \omega) = \{\delta^\star(q, \omega)\}$$

for every state $q \in Q$ and string $\omega \in \Sigma^\star$ — so the condition at line (1) is satisfied (completing this part of a proof).

The Easy Part of the Proof

One is essentially just thinking of a DFA, here, as a restricted kind of NFA such that there is always exactly one choice of what to do next.

- The modifications described here, to produce an NFA $\widehat{M}$ from a DFA M, are only needed to make sure that M really is a "nondeterministic finite automaton" (with the expected type of transition diagram).
- The transition diagrams for M and $\widehat{M}$ will, generally, be identical — and the changes needed are only evident if the construction of $\widehat{M}$, from M, is being described in detail.

The Hard Part — the "Subset Construction"

Suppose now that $L \subseteq \Sigma^\star$ (for some alphabet Σ) and that $L = L(M)$ for some ***nondeterministic*** finite automaton

$$M = (Q, \Sigma, \delta, q_0, F).$$

Suppose, now, that we would like to ***design*** a ***deterministic*** finite automaton

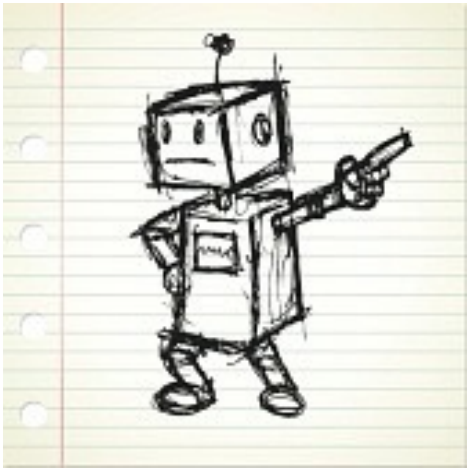
$$\widehat{M} = (\widehat{Q}, \Sigma, \widehat{\delta}, \widehat{q}_0, \widehat{F})$$

such that $L(\widehat{M}) = L = L(M)$.

Note: It is *not* necessary for $\widehat{M}$ to have the same set of states as M (and it will often be necessary for these sets to be different).

The Hard Part — the “Subset Construction”

Recall that in order to do this we should try to...



Be the DFA!

The Hard Part — the "Subset Construction"

In order to this we have to try to answer the following:

- ***Question:*** What do you need to remember about the part μ of the input string you have seen so far?
- ***Answer (Which Might Be Surprising):*** $\delta^\star(q_0, \mu)$ — that is, the subset of Q including states that can be reached from q_0 by processing μ!

The Hard Part — the "Subset Construction"

Sipser's text, *Introduction to the Theory of Computation*, describes one version of the construction that will be used.[1]

- This starts by including a state, in the DFA being constructed, for *every possible answer to this question* — that is, this includes a state in our DFA $\widehat{M}$ for each subset of Q.
- ***Good News:*** This results in a construction that is easy to describe and to analyze

[1] This is found on pages 54–54 of the third edition of this book.

The Hard Part — the "Subset Construction"

- ***Not-So-Good News:*** The construction is not very useful in practice:

- Since our example NFA, has four states in Q, we would now include $2^4 = 16$ states in our DFA. That seems like an awful lot — especially if you are being asked to find a DFA that is equivalent to a given NFA on a test!

The Hard Part — the "Subset Construction"

- More generally, if your NFA had n states then you would include 2^n states in your DFA. Here are a few examples:

n	2^n
10	1,024
20	1,048,576
30	1,073,741,824
40	1,099,511,627,776

The Hard Part — the "Subset Construction"

- The version of the "Subset Construction" described next is — admittedly — not quite a simple as the version in *Introduction to the Theory of Computation.*

The Hard Part — the "Subset Construction"

- However, it avoids this exponential increase in the number of states you must work with... at least, *some* of the time.

- In particular, the DFA we will design will include states corresponding to only *some* of the subsets of Q — namely, only those that are actually "reachable" from the start state.

The Hard Part — the "Subset Construction"

Once again, let

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

be a given nondeterministic finite automaton. In order to construct a deterministic finite automaton

$$\widehat{M} = (\widehat{Q}, \Sigma, \widehat{\delta}, \widehat{q}_0, \widehat{F}),$$

such that $L(\widehat{M}) = L(M)$, we will also construct a function

$$\varphi : \widehat{Q} \to \mathcal{P}(Q)$$

such that, for each state $\widehat{q}$ of $\widehat{Q}$, $\varphi(\widehat{q})$ is a *subset* of Q corresponding to $\widehat{q}$.

The Hard Part — and the "Subset Construction"

Getting Started:

- Recall that $\delta^\star(q_0, \lambda) = Cl_\lambda(q_0)$ — so that

  $$Cl_\lambda(q_0) = \delta^\star(q_0, \omega) \text{ for } \textit{some} \text{ string } \omega \in \Sigma^\star\text{"},$$

  and $Cl_\lambda(q_0)$ is one of the subsets of Q that *must* be represented by a state in the DFA being constructed.
- Let us give the name $\widehat{q}_0$ to the state, in our DFA, that corresponds to the subset $Cl_\lambda(q_0)$ of states in the given NFA — so that $\varphi(\widehat{q}_0) = Cl_\lambda(q_0)$.
- To start out, $Cl_\lambda(q_0)$ is the *only* subset of Q that we are sure must be included so that $\widehat{q}_0$ is the only state in the DFA that we know about.
- We don't know what any transitions out of = $\widehat{q}_0$ should be, at this point.

The Hard Part — and the "Subset Construction"

Continuing:

- Suppose V is a subset of Q such that $V = \delta^\star(q_0, \mu)$ for some string $\mu \in \Sigma^\star$ (so, it must be included) — and suppose that $V = \varphi(\widehat{q}_k)$ for some state $\widehat{q}_k \in \widehat{Q}$.
- Let $\sigma \in \Sigma$. Then — as noted in the previous lecture — the state that should be reached, from the one corresponding to V, by processing σ, corresponds to the set

$$W = \delta^\star(q_0, \mu \cdot \sigma) = \bigcup_{r \in \delta^\star(q_0,\mu)} \left(\bigcup_{s \in \delta(r,\sigma)} Cl_\lambda(s) \right)$$

$$= \bigcup_{r \in V} \left(\bigcup_{s \in \delta(r,\sigma)} Cl_\lambda(s) \right).$$

The Hard Part — and the "Subset Construction"

The algorithm to be described next will generate the subsets of Q that should correspond to states in the DFA being constructed by keeping track of *two* sets:

- $\widehat{Q}$ — subsets of Q that will correspond to states, such that the transitions out of these states (in the DFA) *have* been defined. This will initially be the empty set.
- $\widehat{R}$ — subsets of Q that will correspond to states, such that the transitions out of these states in the DFA *have not* yet been defined. This will initially be $\{\widehat{q}_0\}$.

The Hard Part — and the "Subset Construction"

- Thus — while this algorithm is progress — $\varphi(\widehat{q})$ might be a state $\widehat{r} \in \widehat{Q} \cup \widehat{R}$, for each state $\widehat{q} \in \widehat{Q}$.
- New elements will be added to $\widehat{R}$ as the transition function of the DFA is being defined. Elements will move from $\widehat{R}$ to $\widehat{Q}$ as transitions for them are defined.

The Hard Part — and the "Subset Construction"

Pseudocode to compute a set $\widehat{Q} \subseteq \mathcal{P}(Q)$ of states for our DFA is as shown on this and the following slide.

1. Set $\widehat{q}_0$ a state such that $\varphi(\widehat{q}_0) = Cl_\lambda(q_0)$
2. `integer i` $:= 1$
3. $\widehat{R} := \{\widehat{q}_0\}$
4. $\widehat{Q} := \emptyset$

These initial steps simply carry out the initialization described above. The integer `i` is used to count the number of states, in the DFA, that have been recognized as necessary, so far.

The Hard Part — and the "Subset Construction"

5. `while` $(\widehat{R} \neq \emptyset)$ $\{$
6. Choose a state $\widehat{q}$ from $\widehat{R}$; let V be the set of states, in the given NFA, such that $\varphi(\widehat{q}\,) = V$
7. `for each symbol` $\sigma \in \Sigma$ `do`
8. $$W := \bigcup_{r \in V} \left(\bigcup_{s \in \delta(r,\sigma)} Cl_\lambda(s) \right)$$
9. Set $\widetilde{q}$ to be a (possibly new) state such that $\varphi(\widetilde{q}\,) = W$ — reusing an existing state, if one already exists; $\widehat{\delta}(\widehat{q}, \sigma)$ will be set to be $\widetilde{q}$

The Hard Part — and the "Subset Construction"

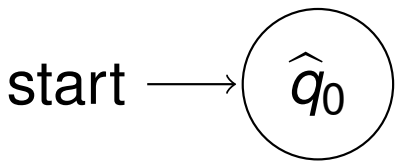
10. `if` ($\widetilde{q}$ is a new state) { // Update `i` and $\widehat{R}$
11. $\widehat{q}_{\mathtt{i}} := \widetilde{q}$
12. $\mathtt{i} := \mathtt{i} + 1$
13. $\widehat{R} := \widehat{R} \cup \{\widetilde{q}\}$
    }
    } // End of `for` loop
14. $\widehat{R} := \widehat{R} \setminus \{\widehat{q}\}$
15. $\widehat{Q} := \widehat{Q} \cup \{\widehat{q}\}$
    } // End of `while` loop
16. `return` $\widehat{Q}$

Subset Construction: Application to the Example

- The initial state of the NFA M is the state q_0, so the initial state of the corresponding DFA (which we call $\widehat{q}_0$) corresponds to the set

$$\widehat{q}_0 = Cl_\lambda(q_0) = \{q_0\} \in \mathcal{P}(Q).$$

- Consequently, the algorithm sets $\widehat{R}$ to be $\{\widehat{q}_0\}$, for $\widehat{q}_0$ as above, and sets $\widehat{Q}$ to be $\emptyset$.
- Our DFA so far:



Subset Construction: Application to the Example

- In the first execution of the body of the `while` loop $\widehat{q}$ must be set to be $\widehat{q}_0$ because this is the only member of $\widehat{R}$ — so V must be the corresponding set, $\varphi(\widehat{q}_0) = Cl_\lambda(q_0) = \{q_0\}$.
- Since $\Sigma = \{0, 1\}$, σ must be set to each of `0` and `1` during the execution of the inner `for` loop.
- Suppose that during this execution — and all other executions of the `for` loop — we set σ to be `0` during the first execution of the body of the `for` loop, and we set σ to be `1` during the second execution of the body of the `for` loop.

Subset Construction: Application to the Example

- Then, during the first execution of the body of the `for` loop (included in the first execution of the body of the outer `while` loop) $V = \varphi(\widehat{q}_0) = \{q_0\}$ and $\sigma = 0$, so that

$$\begin{aligned} W &= \bigcup_{r \in \{q_0\}} \left(\bigcup_{s \in \delta(r,0)} Cl_\lambda(s) \right) \\ &= \bigcup_{s \in \delta(q_0,0)} Cl_\lambda(s) \\ &= Cl_\lambda(q_0) \\ &= \{q_0\} = \varphi(\widehat{q}_0). \end{aligned}$$

- Since $\widehat{q}_0$ already belongs to $\widehat{Q} \cup \widehat{R}$, $\widehat{R}$ is not changed by this execution of the body of the `for` loop.

Subset Construction: Application to the Example

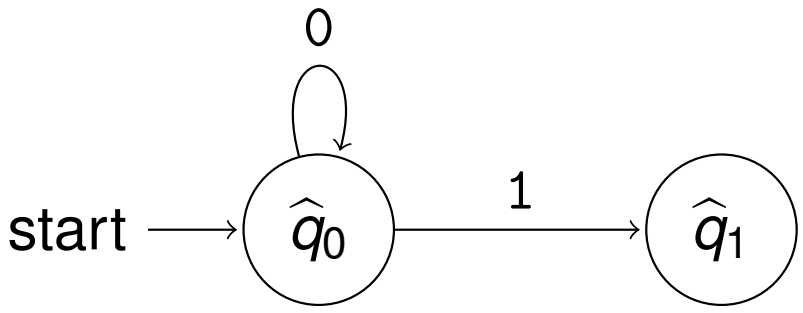
- During the second execution of the body of the `for` loop (included in the first execution of the body of the outer `while` loop) $V = \varphi(\widehat{q}_0) = \{q_0\}$ and $\sigma = 1$, so that

$$\begin{aligned} W &= \bigcup_{r \in \{q_0\}} \left(\bigcup_{s \in \delta(r,1)} Cl_\lambda(s) \right) \\ &= \bigcup_{s \in \delta(q_0,1)} Cl_\lambda(s) \\ &= Cl_\lambda(q_0) \cup Cl_\lambda(q_1) \\ &= \{q_0\} \cup \{q_1, q_2\} \\ &= \{q_0, q_1, q_2\}. \end{aligned}$$

- This set does not correspond to an existing state, so a new state $\widehat{q}_1$, such that $\varphi(\widehat{q}_1) = W = \{q_0, q_1, q_2\}$, is created and added to $\widehat{R}$.

Subset Construction: Application to the Example

- At the end of the first execution of body of the `while` loop $\widehat{q}_0$ is moved from $\widehat{R}$ to $\widehat{Q}$, so that $\widehat{Q} = \{\widehat{q}_0\}$ and $\widehat{R} = \{\widehat{q}_1\}$.The DFA, so far, is as follows.



Subset Construction: Application to the Example

- In the second execution of the body of the `while` loop $\widehat{q}$ must be set to be $\widehat{q}_1$, because this is the only element of $\widehat{R}$ — so that V must be the corresponding set,

$$\varphi(\widehat{q}_1) = \{q_0, q_1, q_2\}.$$

Subset Construction: Application to the Example

- Since $V = \{q_0, q_1, q_2\}$ and $\sigma = 0$ at the beginning of the first execution of the body of the `for` loop,

$$
\begin{aligned}
W &= \bigcup_{r \in \{q_0, q_1, q_2\}} \left(\bigcup_{s \in \delta(r,0)} Cl_\lambda(s) \right) \\
&= \bigcup_{s \in \delta(q_0,0)} Cl_\lambda(s) \cup \bigcup_{s \in \delta(q_1,0)} Cl_\lambda(s) \cup \bigcup_{s \in \delta(q_2,0)} Cl_\lambda(s) \\
&= Cl_\lambda(q_0) \cup Cl_\lambda(q_2) \cup \emptyset \\
&= \{q_0\} \cup \{q_2\} \cup \emptyset \\
&= \{q_0, q_2\}.
\end{aligned}
$$

- This set does not correspond to an existing state, so a new state $\widehat{q}_2$, such that $\varphi(\widehat{q}_2) = W = \{q_0, q_2\}$, is created and added to $\widehat{R}$.

Subset Construction: Application to the Example

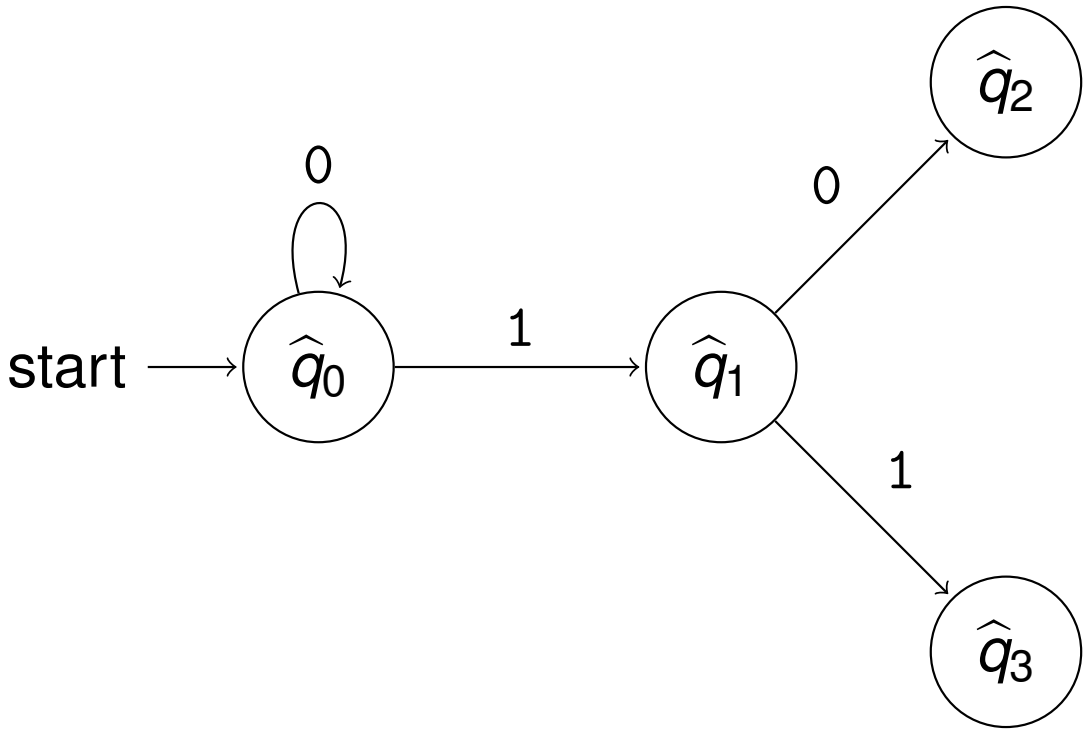
- Since $V = \{q_0, q_1, q_2\}$ and $\sigma = 1$ at the beginning of the second execution of the body of the `for` loop,

$$
\begin{aligned}
W &= \bigcup_{r \in \{q_0, q_1, q_2\}} \left(\bigcup_{s \in \delta(r,1)} Cl_\lambda(s) \right) \\
&= \bigcup_{s \in \delta(q_0,1)} Cl_\lambda(s) \cup \bigcup_{s \in \delta(q_1,1)} Cl_\lambda(s) \cup \bigcup_{s \in \delta(q_2,1)} Cl_\lambda(s) \\
&= (Cl_\lambda(q_0) \cup Cl_\lambda(q_1)) \cup \emptyset \cup Cl_\lambda(q_3) \\
&= Cl_\lambda(q_0) \cup Cl_\lambda(q_1) \cup Cl_\lambda(q_3) \\
&= \{q_0\} \cup \{q_1, q_2\} \cup \{q_3\} \\
&= \{q_0, q_1, q_2, q_3\}.
\end{aligned}
$$

- This set does not correspond to an existing state, so a new state $\widehat{q}_3$, such that $\varphi(\widehat{q}_3) = W = \{q_0, q_1, q_2, q_3\}$, is created and added to $\widehat{R}$.

Subset Construction: Application to the Example

- At the end of the second execution of body of the `while` loop $\widehat{q}_1$ is moved from $\widehat{R}$ to $\widehat{Q}$, so that $\widehat{Q} = \{\widehat{q}_0, \widehat{q}_1\}$ and $\widehat{R} = \{\widehat{q}_2, \widehat{q}_3\}$.The DFA, so far, is as follows.



Subset Construction: Application to the Example

- Since $\widehat{R} = \{q_2, q_3\}$ we might continue by setting $\widehat{q}$ (during the third execution of the body of the `while` loop to be *either* q_2 or q_3. States will be created in a different order for each choice (but it will turn out that the same set of states is ultimately created).
- We will continue, for this example, by setting $\widehat{q}$ to be $\widehat{q}_2$, so that V is the corresponding set

$$V = \varphi(\widehat{q}_2) = \{q_0, q_2\}.$$

Subset Construction: Application to the Example

- Since $V = \{q_0, q_2\}$ and $\sigma = 0$ at the beginning of the first execution of the body of the `for` loop,

$$\begin{aligned} W &= \bigcup_{r \in \{q_0, q_2\}} \left(\bigcup_{s \in \delta(r,0)} Cl_\lambda(s) \right) \\ &= \bigcup_{s \in \delta(q_0,0)} Cl_\lambda(s) \quad \cup \quad \bigcup_{s \in \delta(q_2,0)} Cl_\lambda(s) \\ &= Cl_\lambda(q_0) \cup \emptyset \\ &= \{q_0\} \cup \emptyset \\ &= \{q_0\} = \varphi(\widehat{q}_0). \end{aligned}$$

- Since $\widehat{q}_0$ already belongs to $\widehat{Q} \cup \widehat{R}$, neither $\widehat{Q}$ nor $\widehat{R}$ is changed.

Subset Construction: Application to the Example

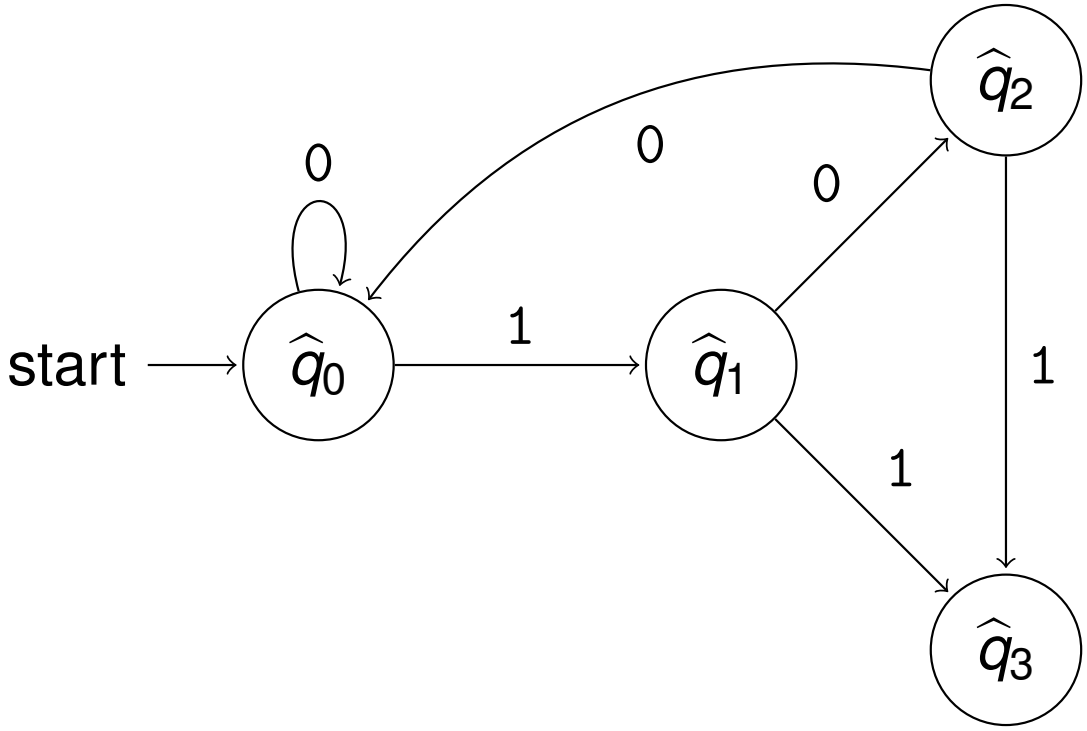
- During the second execution of the body of the `for` loop $V = \{q_0, q_2\}$ and $\sigma = 1$, so that

$$\begin{aligned} W &= \bigcup_{r \in \{q_0, q_2\}} \left(\bigcup_{s \in \delta(r,1)} Cl_\lambda(s) \right) \\ &= \bigcup_{s \in \delta(q_0,1)} Cl_\lambda(s) \quad \cup \quad \bigcup_{s \in \delta(q_2,1)} Cl_\lambda(s) \\ &= (Cl_\lambda(q_0) \cup Cl_\lambda(q_1)) \cup Cl_\lambda(q_3) \\ &= Cl_\lambda(q_0) \cup Cl_\lambda(q_1) \cup Cl_\lambda(q_3) \\ &= \{q_0\} \cup \{q_1, q_2\} \cup \{q_3\} \\ &= \{q_0, q_1, q_2, q_3\} = \varphi(\widehat{q}_3). \end{aligned}$$

- Since $\widehat{q}_3$ already belongs to $\widehat{Q} \cup \widehat{R}$, neither $\widehat{Q}$ nor $\widehat{R}$ is changed.

Subset Construction: Application to the Example

- At the end of the third execution of body of the `while` loop $\widehat{q}_2$ is moved from $\widehat{R}$ to $\widehat{Q}$, so that $\widehat{Q} = \{\widehat{q}_0, \widehat{q}_1, \widehat{q}_2\}$ and $\widehat{R} = \{\widehat{q}_3\}$. The DFA, so far, is as follows.



Subset Construction: Application to the Example

- In the fourth execution of the body of the `while` loop $\widehat{q}$ is set to be $\widehat{q}_3$, since this is the only element of $\widehat{R}$ — so that V must be the corresponding set

$$\varphi(\widehat{q}_3) = \{q_0, q_1, q_2, q_3\}.$$

Subset Construction: Application to the Example

- Since $V = \{q_0, q_1, q_2, q_3\}$ and $\sigma = 0$ at the beginning of the first execution of the body of the `for` loop,

$$
\begin{aligned}
W &= \bigcup_{r \in \{q_0, q_1, q_2, q_3\}} \left(\bigcup_{s \in \delta(r,0)} Cl_\lambda(s) \right) \\
&= \bigcup_{s \in \delta(q_0,0)} Cl_\lambda(s) \cup \bigcup_{s \in \delta(q_1,0)} Cl_\lambda(s) \cup \bigcup_{s \in \delta(q_2,0)} Cl_\lambda(s) \cup \bigcup_{s \in \delta(q_3,0)} Cl_\lambda(s) \\
&= Cl_\lambda(q_0) \cup Cl_\lambda(q_2) \cup \emptyset \cup Cl_\lambda(q_3) \\
&= \{q_0\} \cup \{q_2\} \cup \emptyset \cup \{q_3\} \\
&= \{q_0, q_2, q_3\}.
\end{aligned}
$$

- This set does not correspond to an existing state, so a new state $\widehat{q}_4$, such that $\varphi(\widehat{q}_4) = W = \{q_0, q_2, q_3\}$, is created and added to $\widehat{R}$.

Subset Construction: Application to the Example

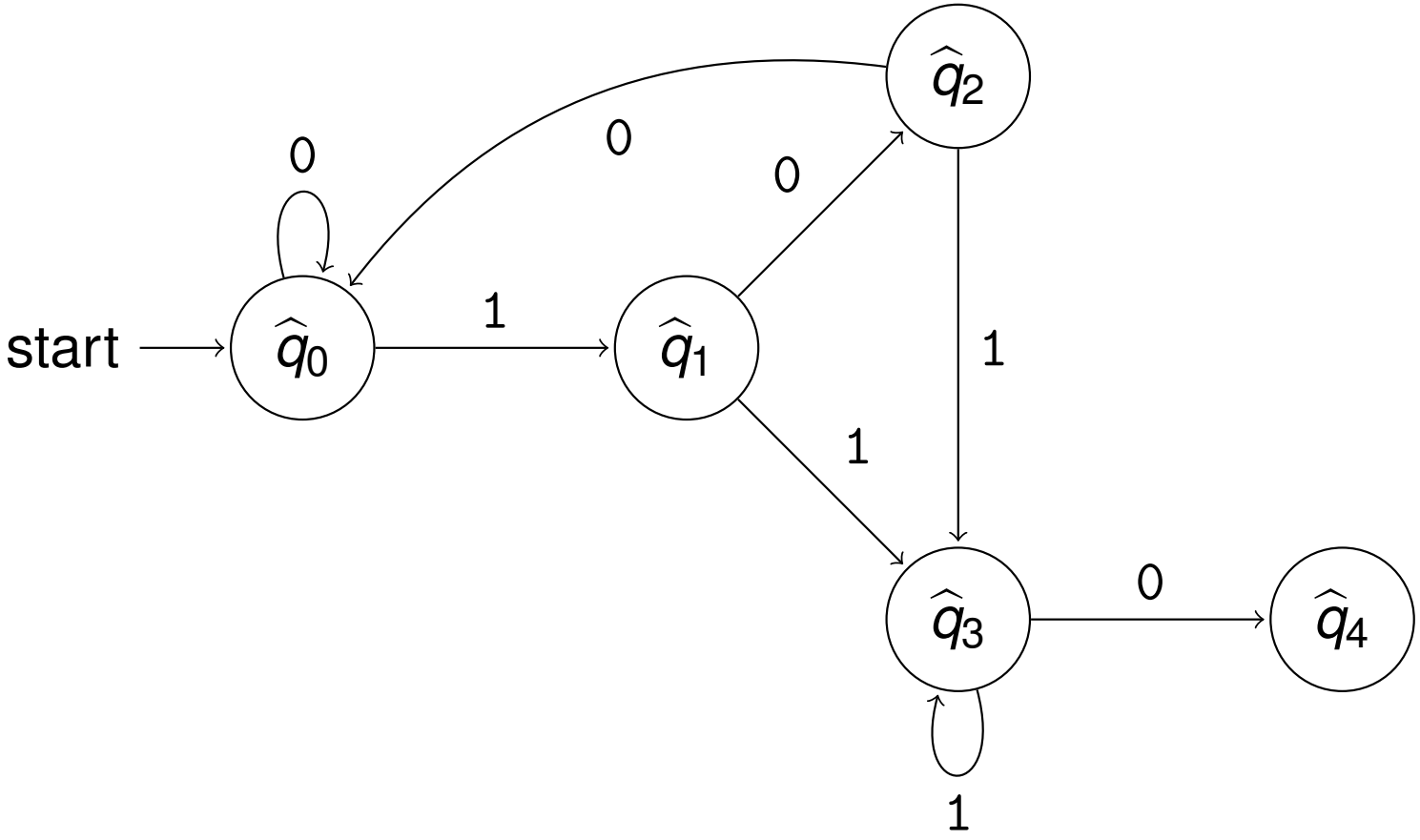
- Since $V = \{q_0, q_1, q_2, q_3\}$ and $\sigma = \mathtt{1}$ at the beginning of the second execution of the body of the `for` loop,

$$
\begin{aligned}
W &= \bigcup_{r \in \{q_0,q_1,q_2,q_3\}} \left(\bigcup_{s \in \delta(r,1)} Cl_\lambda(s) \right) \\
&= \bigcup_{s \in \delta(q_0,1)} Cl_\lambda(s) \cup \bigcup_{s \in \delta(q_1,1)} Cl_\lambda(s) \cup \bigcup_{s \in \delta(q_2,1)} Cl_\lambda(s) \\
&\qquad \cup \bigcup_{s \in \delta(q_3,1)} Cl_\lambda(s) \\
&= (Cl_\lambda(q_0) \cup Cl_\lambda(q_1)) \cup \emptyset \cup Cl_\lambda(q_3) \cup Cl_\lambda(q_3) \\
&= (\{q_0\} \cup \{q_1, q_2\}) \cup \emptyset \cup \{q_3\} \cup \{q_3\} \\
&= \{q_0, q_1, q_2, q_3\} = \varphi(\widehat{q}_3).
\end{aligned}
$$

- Since $\widehat{q}_3$ already belongs to $\widehat{Q} \cup \widehat{R}$, neither $\widehat{Q}$ nor $\widehat{R}$ is changed.

Subset Construction: Application to the Example

- At the end of the fourth execution of body of the `while` loop $\widehat{q}_3$ is moved from $\widehat{R}$ to $\widehat{Q}$, so that $\widehat{Q} = \{\widehat{q}_0, \widehat{q}_1, \widehat{q}_2, \widehat{q}_3\}$ and $\widehat{R} = \{\widehat{q}_4\}$.The DFA, so far, is as follows.



Subset Construction: Application to the Example

- In the fifth execution of the body of the `while` loop $\widehat{q}$ is set to be $\widehat{q}_4$, since this is the only element of $\widehat{R}$ — so that V must be the corresponding set

$$\varphi(\widehat{q}_4) = \{q_0, q_2, q_3\}.$$

Subset Construction: Application to the Example

- Since $V = \{q_0, q_2, q_3\}$ and $\sigma = 0$ at the beginning of the first execution of the body of the `for` loop,

$$
\begin{aligned}
W &= \bigcup_{r \in \{q_0, q_2, q_3\}} \left(\bigcup_{s \in \delta(r,0)} Cl_\lambda(s) \right) \\
&= \bigcup_{s \in \delta(q_0,0)} Cl_\lambda(s) \quad \cup \quad \bigcup_{s \in \delta(q_2,0)} Cl_\lambda(s) \quad \cup \quad \bigcup_{s \in \delta(q_3,0)} Cl_\lambda(s) \\
&= Cl_\lambda(q_0) \cup \emptyset \cup Cl_\lambda(q_3) \\
&= \{q_0\} \cup \emptyset \cup \{q_3\} \\
&= \{q_0, q_3\}.
\end{aligned}
$$

- This set does not correspond to an existing state, so a new state $\widehat{q}_5$, such that $\varphi(\widehat{q}_5) = W = \{q_0, q_3\}$, is created and added to $\widehat{R}$.

Subset Construction: Application to the Example

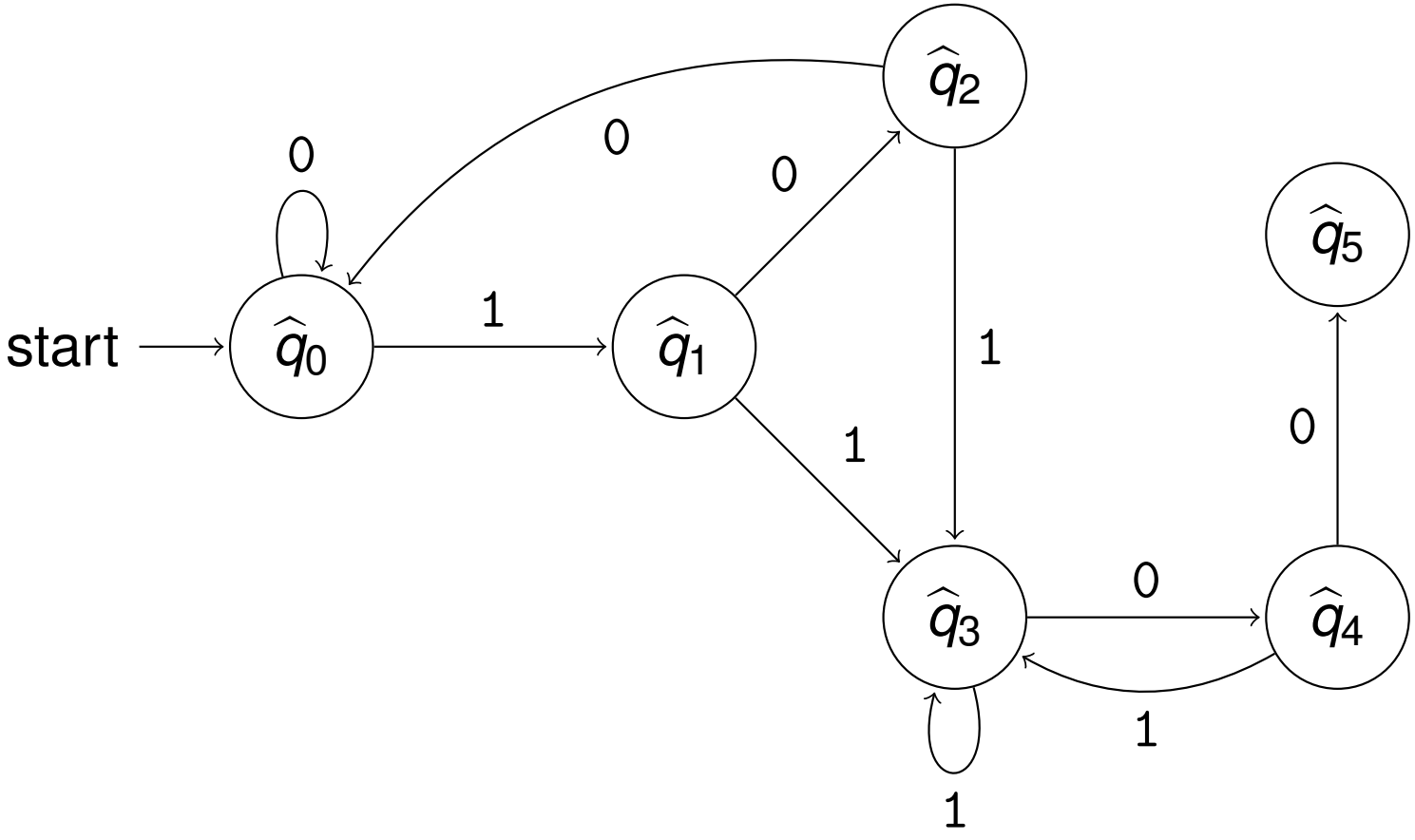
- Since $V = \{q_0, q_2, q_3\}$ and $\sigma = \mathtt{1}$ at the beginning of the second execution of the body of the `for` loop,

$$
\begin{aligned}
W &= \bigcup_{r \in \{q_0, q_2, q_3\}} \left(\bigcup_{s \in \delta(r,1)} Cl_\lambda(s) \right) \\
&= \bigcup_{s \in \delta(q_0,1)} Cl_\lambda(s) \quad \cup \quad \bigcup_{s \in \delta(q_2,1)} Cl_\lambda(s) \quad \cup \quad \bigcup_{s \in \delta(q_3,1)} Cl_\lambda(s) \\
&= (Cl_\lambda(q_0) \cup Cl_\lambda(q_1)) \cup Cl_\lambda(q_3) \cup Cl_\lambda(q_3) \\
&= (\{q_0\} \cup \{q_1, q_2\}) \cup \{q_3\} \cup \{q_3\} \\
&= \{q_0, q_1, q_2, q_3\} = \varphi(\widehat{q}_3).
\end{aligned}
$$

- Since $\widehat{q}_3$ already belongs to $\widehat{Q} \cup \widehat{R}$, neither $\widehat{Q}$ nor $\widehat{R}$ is changed.

Subset Construction: Application to the Example

- At the end of the fifth execution of body of the `while` loop $\widehat{q}_4$ is moved from $\widehat{R}$ to $\widehat{Q}$, so that $\widehat{Q} = \{\widehat{q}_0, \widehat{q}_1, \widehat{q}_2, \widehat{q}_3, \widehat{q}_4\}$ and $\widehat{R} = \{\widehat{q}_5\}$.The DFA, so far, is as follows.



Subset Construction: Application to the Example

- In the sixth execution of the body of the `while` loop $\widehat{q}$ is set to be $\widehat{q}_5$, since this is the only element of $\widehat{R}$ — so that V must be the corresponding set

$$\varphi(\widehat{q}_5) = \{q_0, q_3\}.$$

Subset Construction: Application to the Example

- Since $V = \{q_0, q_3\}$ and $\sigma = 0$ at the beginning of the first execution of the body of the `for` loop,

$$\begin{aligned} W &= \bigcup_{r \in \{q_0, q_3\}} \left(\bigcup_{s \in \delta(r,0)} Cl_\lambda(s) \right) \\ &= \bigcup_{s \in \delta(q_0,0)} Cl_\lambda(s) \quad \cup \quad \bigcup_{s \in \delta(q_3,0)} Cl_\lambda(s) \\ &= Cl_\lambda(q_0) \cup Cl_\lambda(q_3) \\ &= \{q_0\} \cup \{q_3\} \\ &= \{q_0, q_3\} = \varphi(\widehat{q}_5). \end{aligned}$$

- Since $\widehat{q}_5$ already belongs to $\widehat{Q} \cup \widehat{R}$, neither $\widehat{Q}$ nor $\widehat{R}$ is changed.

Subset Construction: Application to the Example

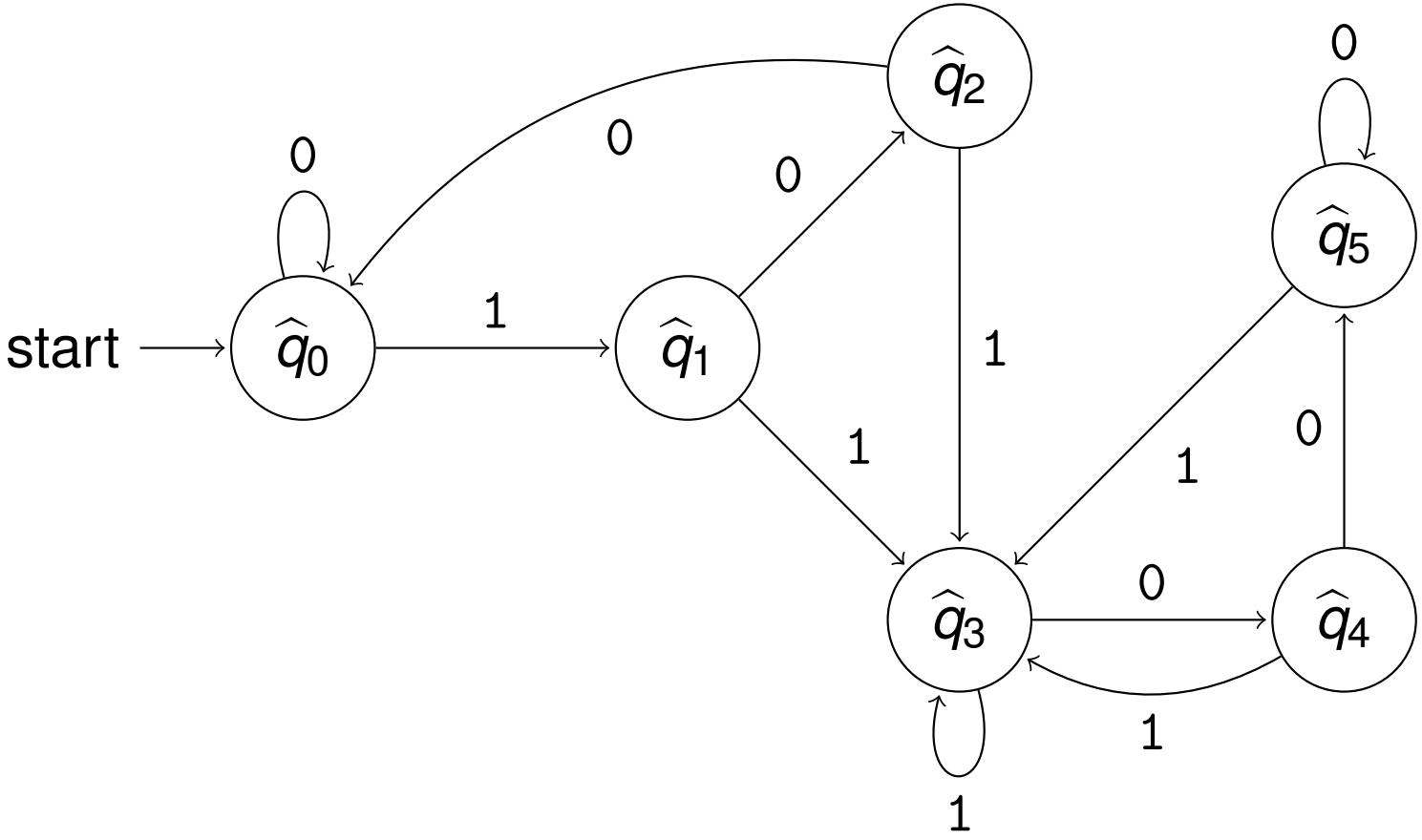
- Since $V = \{q_0, q_3\}$ and $\sigma = \texttt{1}$ at the beginning of the second execution of the body of the `for` loop,

$$\begin{aligned} W &= \bigcup_{r \in \{q_0, q_3\}} \left(\bigcup_{s \in \delta(r,1)} Cl_\lambda(s) \right) \\ &= \bigcup_{s \in \delta(q_0,1)} Cl_\lambda(s) \quad \cup \quad \bigcup_{s \in \delta(q_3,1)} Cl_\lambda(s) \\ &= (Cl_\lambda(q_0) \cup Cl_\lambda(q_1)) \cup Cl_\lambda(q_3) \\ &= \{q_0\} \cup \{q_1, q_2\} \cup \{q_3\} \\ &= \{q_0, q_1, q_2, q_3\} = \varphi(\widehat{q}_3). \end{aligned}$$

- Since $\widehat{q}_3$ already belongs to $\widehat{Q} \cup \widehat{R}$, neither $\widehat{Q}$ nor $\widehat{R}$ is changed.

Subset Construction: Application to the Example

- At the end of the sixth execution of body of the `while` loop $\widehat{q}_5$ is moved from $\widehat{R}$ to $\widehat{Q}$, so that $\widehat{Q} = \{\widehat{q}_0, \widehat{q}_1, \widehat{q}_2, \widehat{q}_3, \widehat{q}_4, \widehat{q}_5\}$ and $\widehat{R} = \emptyset$.The DFA, so far, is as follows.



The "Subset Construction:" Why This Works

It can be proved (using induction on the number of executions of the body of the `while` loop) that the following properties are satisfied at the beginning and end of ***every*** execution of the body of this loop:

1. $\delta^\star(q_0, \lambda) = Cl_\lambda(q_0) = \varphi(\widehat{q}_0)$, so that $\delta^\star(q_0, \lambda)$ corresponds to a state that belongs to either $\widehat{Q}$ or $\widehat{R}$.
2. For every subset V of Q corresponding to at state in $\widehat{Q} \cup \widehat{R}$, there is a string $\mu \in \Sigma^\star$ such that $V = \delta^\star(q_0, \mu)$.
3. $\widehat{Q} \cap \widehat{R} = \emptyset$.
4. For every subset V of Q that corresponds to a state in $\widehat{Q}$, and for every symbol $\sigma \in \Sigma$, the set

$$W = \bigcup_{r \in V} \left(\bigcup_{s \in \delta(r,\sigma)} Cl_\lambda(s) \right)$$

corresponds to a state in $\widehat{Q} \cup \widehat{R}$.

The "Subset Construction:" Why This Works

This implies that if $\widehat{R} = \emptyset$ then

5. $\delta^\star(q_0, \lambda) = Cl_\lambda(q_0) = \varphi(\widehat{q}_0)$, so that $\delta^\star(q_0, \lambda)$ corresponds to a state that belongs to $\widehat{Q}$;
6. For every subset V of Q corresponding to a state in $\widehat{Q}$, there is a string $\mu \in \Sigma^\star$ such that $V = \delta^\star(q_0, \mu)$.
7. For every subset V of Q that corresponds to a state in $\widehat{Q}$, and for every symbol $\sigma \in \Sigma$, the set

$$W = \bigcup_{r \in V} \left(\bigcup_{s \in \delta(r,\sigma)} Cl_\lambda(s) \right)$$

also corresponds to a state in $\widehat{Q}$ (so the "transition function" can actually be defined).

The "Subset Construction:" Why This Works

- Notice that $\widehat{Q}$ is initially empty and its size is increased by one during each execution of the body of the `while` loop.
- ***Conclusion:*** The execution of the loop must end (with $\widehat{R} = \emptyset$) after at most 2^k executions of the loop body if $k = |Q|$ — because $\widehat{Q}$ would have to include every set in $\mathcal{P}(Q)$ after that. Then, since $\widehat{R} \subseteq \mathcal{P}(Q)$ and $\widehat{Q} \cap \widehat{R} = \emptyset$, it would follow that $\widehat{R}$ would *have* to be empty, as claimed.
- The above properties 5–7 imply that the the transition function has been completely defined (and is a well-defined total function $\widehat{\delta} : \widehat{Q} \times \Sigma \to \widehat{Q}$ at that point, as required).

The "Subset Construction:" Choosing Final Sets

- Finally, for each state $\widehat{q} \in \widehat{Q}$, include $\widehat{q}$ in the set $\widehat{F}$ ***if and only if*** $\varphi(\widehat{q}) \cap F \neq \emptyset$.
- *Application to our Example:* $\widehat{Q} = \{\widehat{q}_0, \widehat{q}_1, \widehat{q}_2, \widehat{q}_3, \widehat{q}_4, \widehat{q}_5\}$
  - $\varphi(\widehat{q}_0) \cap F = \{q_0\} \cap \{q_3\} = \emptyset$, so $\widehat{q}_0 \notin \widehat{F}$.
  - $\varphi(\widehat{q}_1) \cap F = \{q_0, q_1, q_2\} \cap \{q_3\} = \emptyset$, so $\widehat{q}_1 \notin \widehat{F}$.
  - $\varphi(\widehat{q}_2) \cap F = \{q_0, q_2\} \cap \{q_3\} = \emptyset$, so $\widehat{q}_2 \notin \widehat{F}$.
  - $\varphi(\widehat{q}_3) \cap F = \{q_0, q_1, q_2, q_3\} \cap \{q_3\} = \{q_3\}$, so $\widehat{q}_3 \in \widehat{F}$.
  - $\varphi(\widehat{q}_4) \cap F = \{q_0, q_2, q_3\} \cap \{q_3\} = \{q_3\}$, so $\widehat{q}_4 \in \widehat{F}$.
  - $\varphi(\widehat{q}_5) \cap F = \{q_0, q_3\} \cap \{q_3\} = \{q_3\}$, so $\widehat{q}_5 \in \widehat{F}$.

  Thus $\widehat{F} = \{\widehat{q}_3, \widehat{q}_4, \widehat{q}_5\}$.

The "Subset Construction:" Why This Works

- Recall that if $\omega \in \Sigma^\star$ then both
  - $\delta^\star(q_0, \omega)$ — the value of the extended transition function for the NFA M for the start state, q_0, and the string ω, and
  - $\varphi(\widehat{\delta}^\star(\widehat{q}_0, \omega))$ — the subset of Q corresponding to the value of the extended tradition function for the DFA $\widehat{M}$ for the start state, $\widehat{q}_0 = Cl_\lambda(q_0)$ and string ω

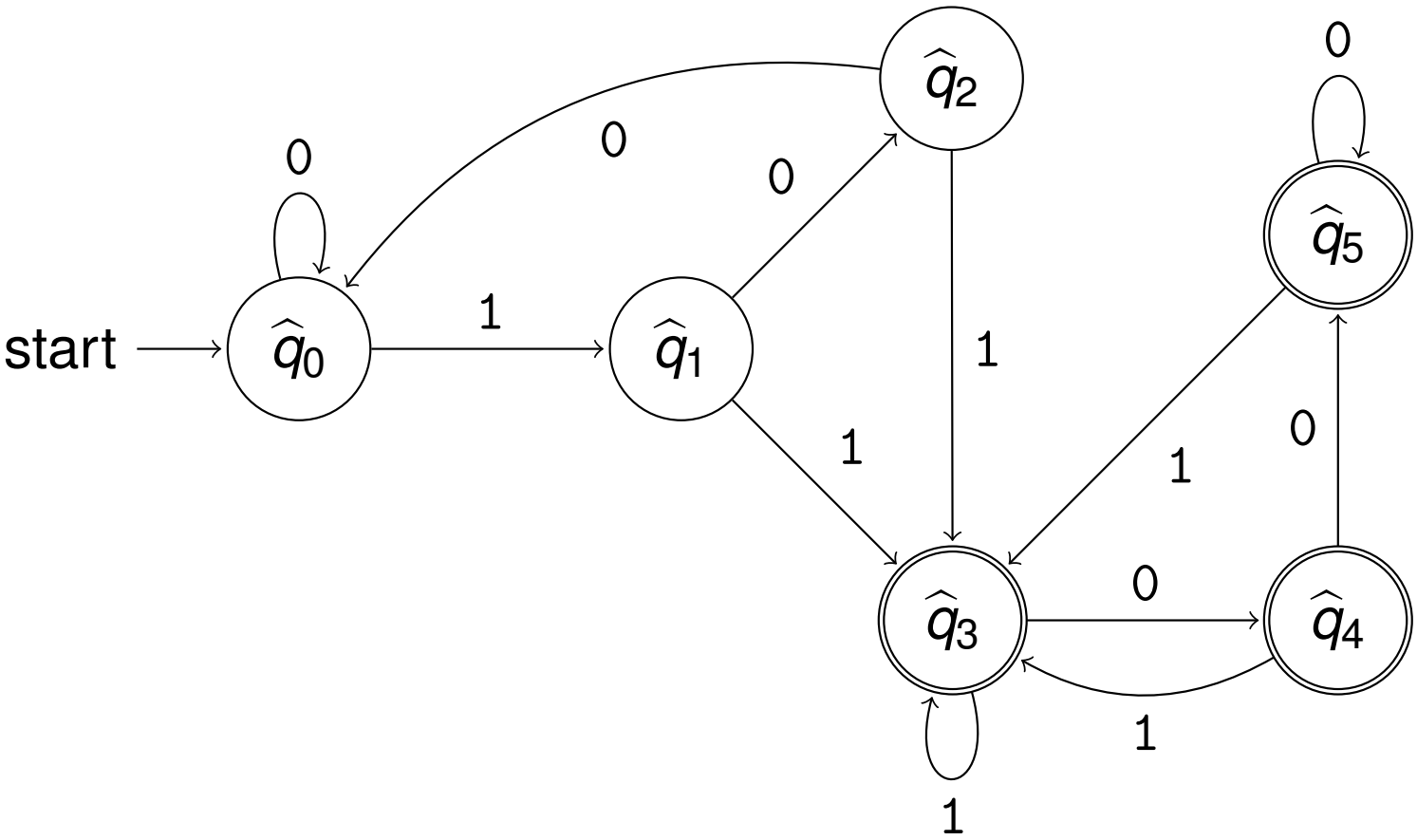
  are *sets* of states in Q — that is, elements of $\mathcal{P}(Q)$.
- It is possible to prove the following result (by induction on the length of the string ω):

$$\delta^\star(q_0, \omega) = \varphi(\widehat{\delta}(\widehat{q}_0, \omega))$$

  for ***every*** string $\omega \in \Sigma^\star$
- If $\widehat{F}$ is as defined above then it follows that M accepts ω if and only if $\widehat{M}$ accepts ω, for all $\omega \in \Sigma^\star$. Thus $L(\widehat{M}) = L(M)$, as required.

Application to the Example

The resulting DFA is as follows.



Simulations

A ***simulation*** is something that can be presented to relate the power of two models of computation. In order to show that the machines described by a ***second*** model of computation are (in some sense) at least as powerful or efficient as the machines described by a ***first*** model of computation, we generally do the following:

(a) Consider an arbitrary machine M, of the type described by the ***first*** model of computation.

(b) Use M to define another machine $\widehat{M}$, of the type described by the ***second*** model of computation.

(c) Prove that $\widehat{M}$ solves the same problem as M.

Simulations

The notes have sketched *two* simulations:

- In the first simulation, the first model of computation was “the set of deterministic Turing machines” and the second model of computation was “the set of nondeterministic Turing machines”.
- In the second simulation, the first model of computation was “the set of nondeterministic Turing machines” and the first model of computation was “the set of deterministic Turing machines”.

Simulations

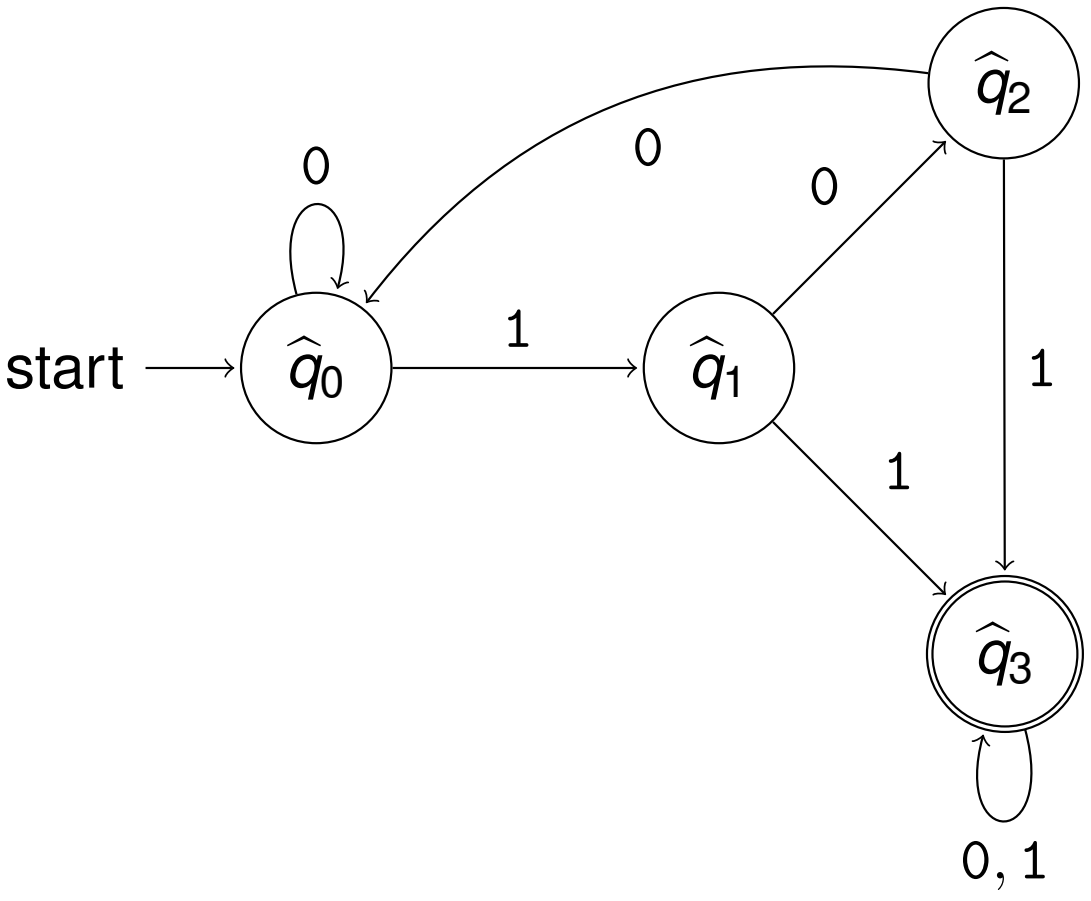
- It follows by the results, established using these simulations, that a language $L \subseteq \Sigma^\star$ is a regular language ***if and only if*** it is the language of some nondeterministic finite automaton.
- Somewhat more complicated simulations will be used, later in this course, to relate the computational power of other machine models.

How Many States Do We Need?

Consider the DFA that was produced, in our ongoing example.

- Notice that, once this DFA enters an accepting state, it simply moves from one accepting state to another accepting state — it never returns to a state that is *not* in $\widehat{F}$.
- Thus all the accepting states can be replaced by a single one without changing the language of the DFA. The resulting, simpler DFA (for the same language) is as follows.

How Many States Do We Need?



How Many States Do We Need?

- The second simulation, in these notes, establishes that if $L \subseteq \Sigma^\star$ and there is an NFA

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

with language L such that M has k states (i.e., $|Q| = k$), then there is also a DFA

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

which also has language L, and which has *at most* 2^k states.

A Bad Case for the "Subset Construction"

- Much of the time, the DFA generated using the "subset construction" will have a *much* smaller set of states than this suggest.
- Indeed, after eliminating unnecessary states in the DFA for our example, we produced a DFA that had the same number of states as the NFA we started with.

A Bad Case for the "Subset Construction"

- ***Question:*** Is an exponential increase in the number of states ever really necessary?
- ***Answer:*** Yes! (We sometimes need *almost* as many states as the above result might suggest.) In particular, there exists an infinite sequence of languages

$$L_1, L_2, \cdots \subseteq \Sigma^\star,$$

for the alphabet $\Sigma = \{0, 1\}$, such that L_k has an NFA with $k + 1$ states, but *every* DFA for L_k must include at least 2^k states, for every positive integer k.

- A supplemental document, including a proof of this result, is available.

Expectations for Students

- Students *will not* be asked to explain why the "subset construction" is correct (even though a proof of this is sketched in these online notes).
- Students *might* be given an NFA M for a language in $\Sigma^\star$ and asked to generate a DFA for the same language and explain why the languages of the two automata are the same. Explaining why the NFA and DFA have the same language is easy to do (because the explanation is "the subset construction is correct") if the subset construction is used to solve this problem.
- Students will be expected to understand the relationship between the number of states in an NFA for a given language, and the number of states that might be needed in a DFA for the same language — but they will not be expected to know (or be able to explain) how to prove this.