Computer Science 351

Equivalence of Deterministic Finite Automata and Nondeterministic Finite Automata

Instructor: Wayne Eberly

Department of Computer Science
University of Calgary

Lecture #6

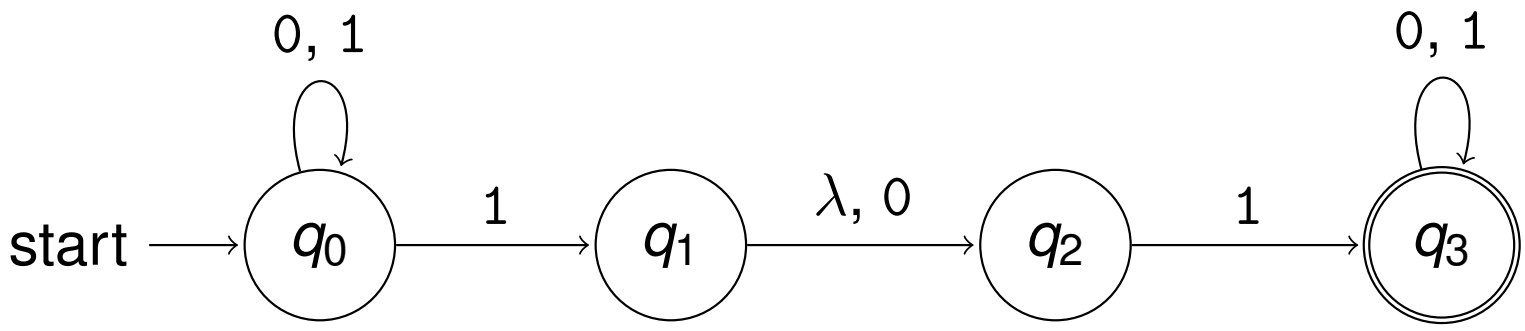
Goal for Today

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:



The Easy Part of the Proof: The Idea

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 $\omega \in \Sigma^\star$ —

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

To do this, describe a nondeterministic finite automaton with the same *picture* as the given deterministic finite automaton.

The Easy Part of the Proof: The Details

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: The Details

- 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 the proof).
- Since $\widehat{M}$ has the same set of accepting states as M this can be used to prove that $L(\widehat{M}) = L(M)$.

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

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

We would like to ***design*** a ***deterministic*** finite automaton

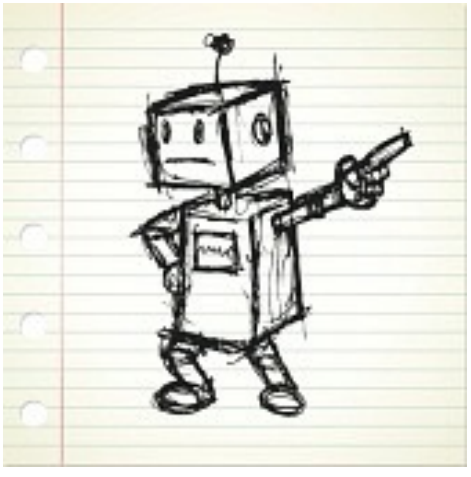
$$\widehat{M} = (\widehat{Q}, \Sigma, \widehat{\delta}, \widehat{q}_0, \widehat{F})$$

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.*

It *only* includes states corresponding to subsets V of Q such that $\delta^\star(q_0, \omega) = V$ for some string $\omega \in \Sigma^\star$.

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.

- The DFA we will design will include states that are actually "reachable" from the start state.

The Hard Part — the "Subset Construction"

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 $Cl_\lambda(q_0)$ is one of the subsets of Q that *must* be represented by a state n the DFA being constructed.
- Let us give the name $\widehat{q}_0$ to the state, in our DFA, that corresponds to $Cl_\lambda(q_0)$ — 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 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 construction will generate the subsets of Q that should correspond to states in the DFA being constructed by keeping track of *two* sets:

- $\widehat{Q}$ — states, corresponding to subsets of Q, such that the transitions out of these states (in the DFA) *have* all been defined.
  Initially $\widehat{Q} = \emptyset$.
- $\widehat{R}$ — states, corresponding to subsets of Q, such that the transitions out of these states in the DFA *have not* yet been defined.
  Initially $\widehat{R} = \{\widehat{q}_0\}$ — where $\widehat{q}_0$ corresponds to $Cl_\lambda(q_0)$.

The Hard Part — and the "Subset Construction"

- The construction proceeds by choosing a state $\widehat{q} \in \widehat{R}$ and discovering $\widehat{\delta}(\widehat{q}, \sigma)$ for symbols $\sigma \in \Sigma$.
- New elements are added to $\widehat{Q}$ whenever it is discovered that $\widehat{\delta}(\widehat{q}, \sigma)$ is a "new state" $\widehat{r}$ — because the corresponding subset $\varphi(\widehat{r})$ of Q has not been included yet.
- Elements will move from $\widehat{R}$ to $\widehat{Q}$ when all their transitions have been defined.
- Eventually $\widehat{R} = \emptyset$ — and both $\widehat{Q}$ and the function $\widehat{\delta} : \widehat{Q} \times \Sigma \to \widehat{Q}$ have been completed.
- For all $\widehat{q} \in \widehat{Q}$, $\widehat{q}$ should be included in $\widehat{F}$ if and only if

$$\varphi(\widehat{q}) \cap F \neq \emptyset,$$

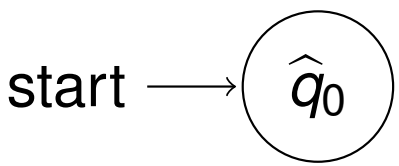
that is, the set of state in Q, corresponding to $\widehat{q}$, includes at least one accepting state in M.

Subset Construction: Application to the Example

- To begin, $\widehat{Q} = \emptyset$ and $\widehat{R} = \{\widehat{q}_0\}$ where

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

- $\widehat{q}_0$ will be the start state for the DFA being designed.
- Our DFA so far:



Subset Construction: Application to the Example

- Since $R = \{\widehat{q}_0\}$ we must choose $\widehat{q}$ to be $\widehat{q}_0$.
- Since $\varphi(\widehat{q}) = \varphi(\widehat{q}_0) = \{q_0\}$, $\widehat{\delta}(\widehat{q}, 0)$ must correspond to the set

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

Thus $\widehat{\delta}(\widehat{q}_0, 0) = \widehat{q}_0$.

Subset Construction: Application to the Example

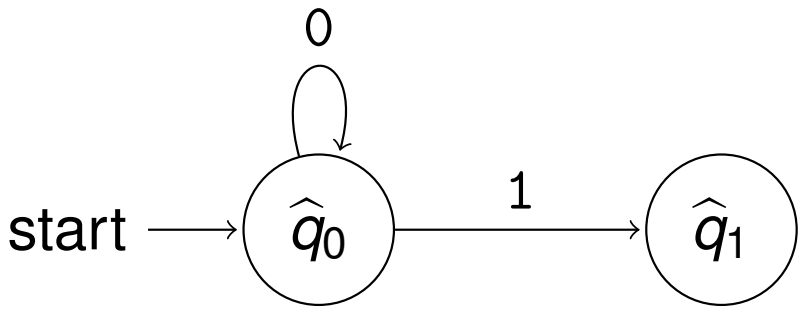
- Since $\varphi(\widehat{q}) = \varphi(\widehat{q}_0) = \{q_0\}$, $\widehat{\delta}(\widehat{q}, 1)$ must correspond to the set

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

- 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 $\widehat{R}$. Now $\widehat{\delta}(\widehat{q}_0, 1) = \widehat{q}_1$ and $\widehat{R} = \{\widehat{q}_0, \widehat{q}_1\}$.

Subset Construction: Application to the Example

- Since all transitions out of $\widehat{q}_0$ have been identified, $\widehat{q}_0$ is moved to $\widehat{Q}$. Now $\widehat{Q} = \{\widehat{q}_0\}$ and $\widehat{R} = \{\widehat{q}_1\}$.
- Our DFA, so far:



Subset Construction: Application to the Example

- Since $R = \{\widehat{q}_1\}$ we must choose $\widehat{q}$ to be $\widehat{q}_1$.
- Since $\varphi(\widehat{q}) = \varphi(\widehat{q}_1) = \{q_0, q_1, q_2\}$, $\widehat{\delta}(\widehat{q}_1, 0)$ must correspond to the set

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

- 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 $\widehat{R}$. Now $\widehat{R} = \{\widehat{q}_1, \widehat{q}_2\}$.

Subset Construction: Application to the Example

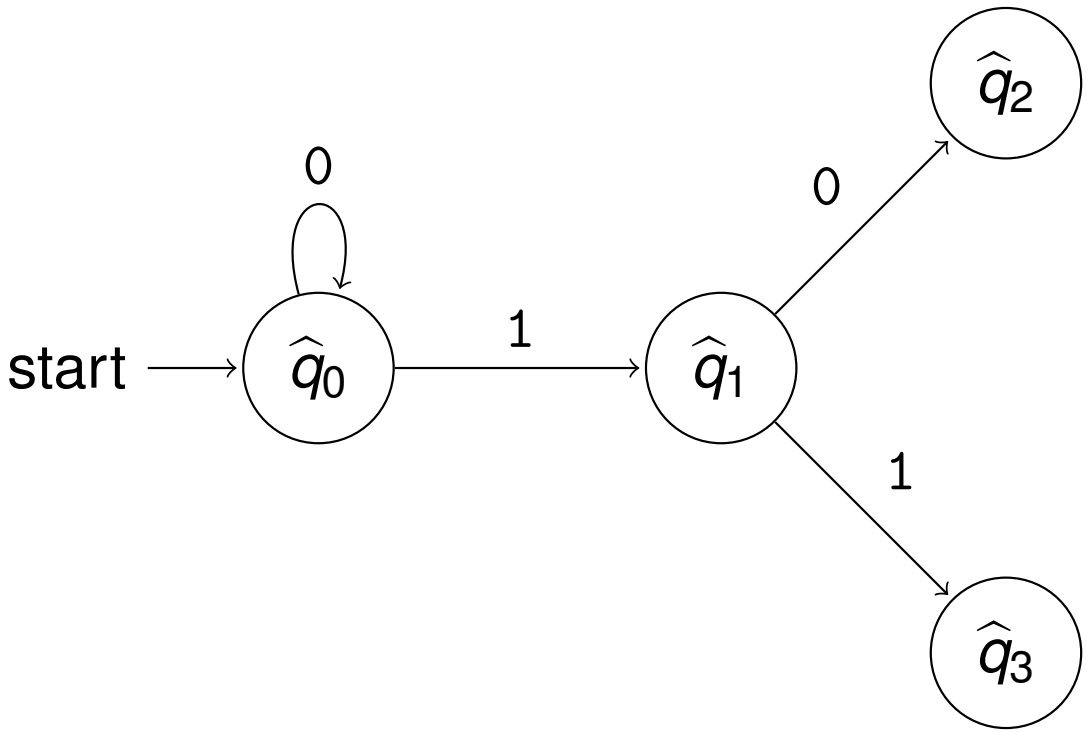
- Since $\varphi(\widehat{q}) = \varphi(\widehat{q}_1) = \{q_0, q_1, q_2\}$, $\widehat{\delta}(\widehat{q}_1, 1)$ must correspond to the set

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

- 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 $\widehat{R}$. Now $\widehat{R} = \{\widehat{q}_1, \widehat{q}_2, \widehat{q}_3\}$.

Subset Construction: Application to the Example

- Since all transitions out of $\widehat{q}_1$ have been identified, $\widehat{q}_1$ is moved to $\widehat{Q}$. Now $\widehat{Q} = \{\widehat{q}_0, \widehat{q}_1\}$ and $\widehat{R} = \{\widehat{q}_2, \widehat{q}_3\}$.
- Our DFA, so far:



Subset Construction: Application to the Example

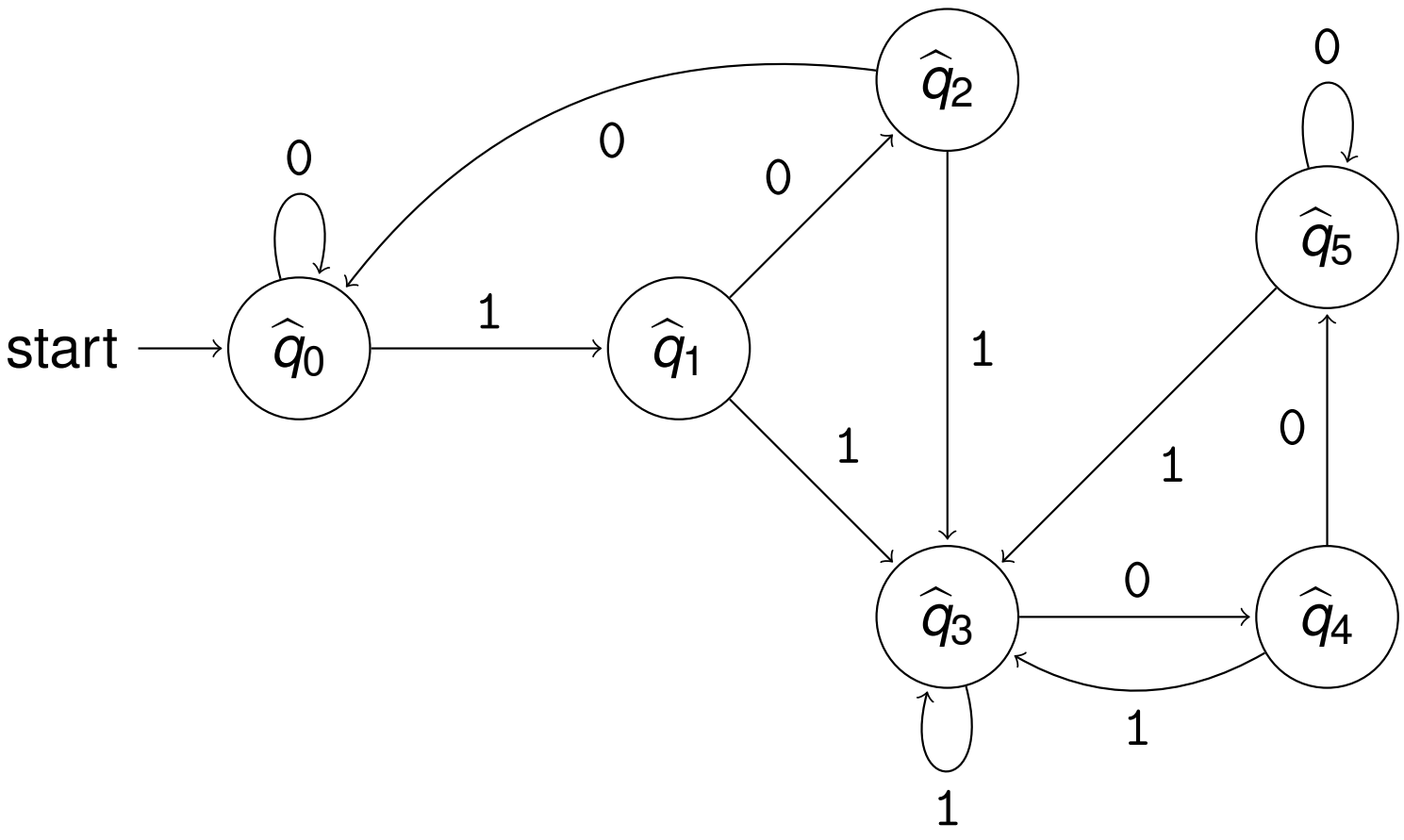
- This goes on for quite a while — and the completion is given in a supplemental document. Eventually, the following is true:

$$\widehat{Q} = \{\widehat{q}_0, \widehat{q}_1, \widehat{q}_2, \widehat{q}_3, \widehat{q}_4, \widehat{q}_5\} \quad \text{and} \quad \widehat{R} = \emptyset.$$

- States in $\widehat{Q}$ correspond to subsets of Q as follows.
  - $\varphi(\widehat{q}_0) = \{q_0\}$.
  - $\varphi(\widehat{q}_1) = \{q_0, q_1, q_2\}$.
  - $\varphi(\widehat{q}_2) = \{q_0, q_2\}$.
  - $\varphi(\widehat{q}_3) = \{q_0, q_1, q_2, q_3\}$.
  - $\varphi(\widehat{q}_4) = \{q_0, q_2, q_3\}$.
  - $\varphi(\widehat{q}_5) = \{q_0, q_3\}$.

Subset Construction: Application to the Example

- Our DFA, so far:



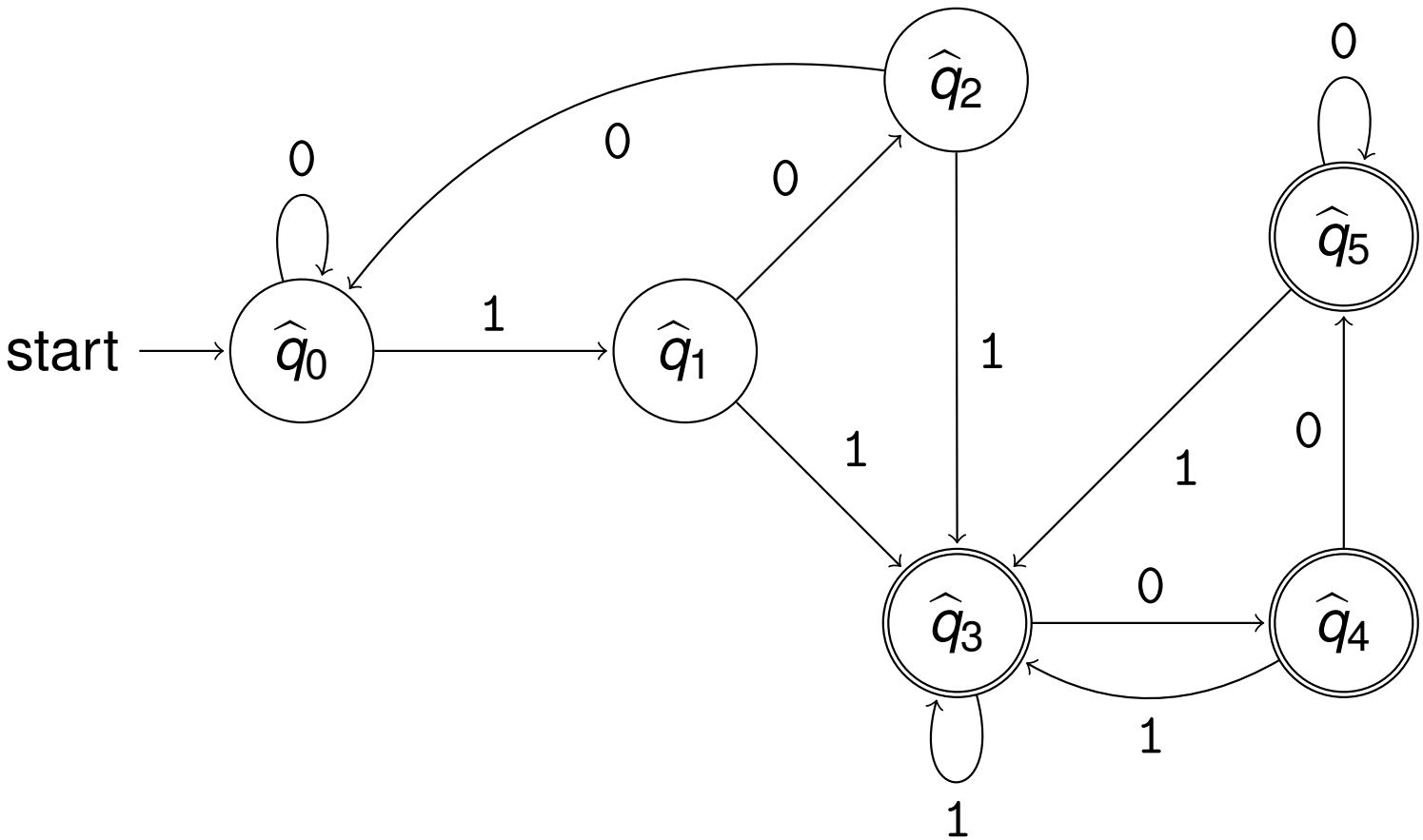
Subset Construction: Application to the Example

- Recall that, for each state $\widehat{q} \in \widehat{Q}$, include $\widehat{q}$ in the set $\widehat{F}$ ***if and only if*** $\varphi(\widehat{q}) \cap F \neq \emptyset$.
- Checking $\varphi(\widehat{q})$ for all $\widehat{q} \in \widehat{Q}$:
  - $\varphi(\widehat{q}_0) \cap F = \{q_0\} \cap \{q_3\} = \emptyset$, 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 $\widehat{F} = \{\widehat{q}_3, \widehat{q}_4, \widehat{q}_5\}$.

Application to the Example

The resulting DFA is as follows.



Correctness of This Construction

- The construction can be described, a bit more formally, as an ***algorithm*** with a doubly nested `while` loop — which receives an NFA as input and generates a DFA as output.
- Techniques for proving the correctness of algorithms (seen in CPSC 251, 331 and 413) can be used to prove that the algorithm is correct — so the construction is reliable too.
- See *another* supplemental document, for this lecture, for details.

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

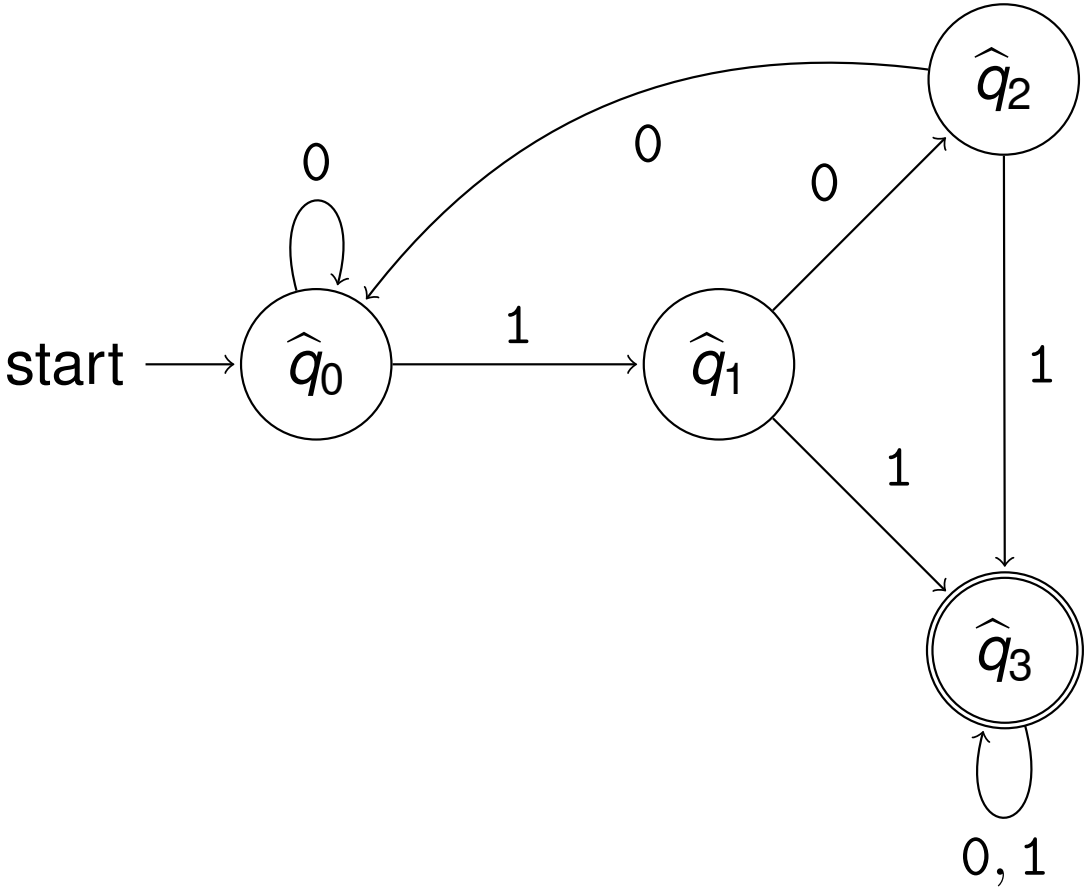
- It follows by the results, established using these simulations, that a language $L \subseteq \Sigma^\star$ is a regular language ***if and only if*** it is the language of some nondeterministic finite automaton.
- Somewhat more complicated simulations will be used, later in this course, to relate the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