# Computer Science 351

## DFA Design and Verification — Part Two

Instructor: Wayne Eberly

Department of Computer Science
University of Calgary

Lecture #4

# Goals for Today

***Goals for Today:***

- *Conclusion* of a presentation of a process that can be used to ***design*** a deterministic finite automaton that has a given language.
- Description of a process to ***prove*** that a given deterministic finite automaton has a given language

# Ongoing Example

***Goal:*** Design a deterministic finite automata for the following language over the alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$:

$$L = \{\omega \in \Sigma^\star \mid \text{either } \omega \text{ does not include an “a”} \text{ or } \omega \text{ does not include a “b”}\}.$$

# Following a Design Process — So Far

We started by considering ***what a DFA must remember about the string that has been processed, so far***.

- *Our Answer, So Far:* We *guessed* that the DFA must remember whether both an “a” and a “b” have already been seen: The string *is* in the desired the language if this is not true yet — but it it is *not* in the language if this *is* true.

# Following a Design Process — So Far

This was used to participation the set, $\Sigma^\star$, of all strings over $\Sigma$, in to a pair of subsets, namely

$$S_0 = L = \{\omega \in \Sigma^\star \mid \text{either } \omega \text{ does not include an “a” or } \omega \text{ does not include a “b”}\}$$

and

$$S_1 = \Sigma^\star \setminus L = \{\omega \in \Sigma^\star \mid \omega \text{ includes both an “a” and a “b”}\}.$$

These would correspond to states $q_0$ and $q_1$, respectively, in a DFA to be designed.

# Following a Design Process — So Far

Several ***necessary conditions*** — or "sanity checks" — were considered.

1. We checked that only ***finitely many*** subsets of $\Sigma^\star$ had been identified.
2. We checked that ***every string belongs to exactly one of these subsets***.

   Since $\lambda \in S_0$ we can set the start state to be the state, $q_0$, corresponding to $S_0$.

# Following a Design Process — So Far

3. For every set $S_i$ that has been identified (with corresponding state $q_i$) either $S_i \subseteq L$ — in which case $q_i \in F$ — or $S_i \cap L = \emptyset$ — in which case $q_i \notin F$.

   Since $S_0 \subseteq L$ and $S_1 \cap L = \emptyset$, for the example, this test has also been passed — and $F = \{q_0\}$.

# Following a Design Process — So Far

Suppose subsets $S_0, S_1, \ldots, S_{n-1}$ have been identified (as is true, for the example, with $n = 2$).

4. For every integer $i$ such that $0 \leq i \leq n-1$ and symbol $\sigma \in \Sigma$ there must exist an integer $j$ such that $0 \leq j \leq n-1$ and $\delta(q_i, \sigma) = q_j$. Then it must also be true that

$$\{\omega \cdot \sigma \mid \omega \in S_i\} \subseteq S_j.$$

*Good News:* Since $\omega \cdot \sigma \in S_1$, whenever $\omega \in S_1$ — so that

$$\{\omega \cdot \sigma \mid \omega \in S_1\} \subseteq S_1$$

for all $\sigma \in \Sigma$, *the transitions out of* $q_1$ *were well-defined*.

# Following a Design Process — So Far

*Not-So-Good News:* While

$$\{\omega \cdot \mathtt{c} \mid \omega \in S_0\} \subseteq S_0,$$

so that the transition for $q_0$ and "c" is well-defined (and $\delta(q_0, \mathtt{c}) = q_0$), it was also discovered that

$$\{\omega \cdot \mathtt{a} \mid \omega \in S_0\} \not\subseteq S_0 \quad \text{and} \quad \{\omega \cdot \mathtt{a} \mid \omega \in S_0\} \not\subseteq S_1$$

— so that the transition for $q_0$ and "a" is not well-defined — and that

$$\{\omega \cdot \mathtt{b} \mid \omega \in S_0\} \not\subseteq S_0 \quad \text{and} \quad \{\omega \cdot \mathtt{b} \mid \omega \in S_0\} \not\subseteq S_1$$

— so that the transition for $q_0$ and "b" is not well-defined, either.

# Following a Design Process — So Far

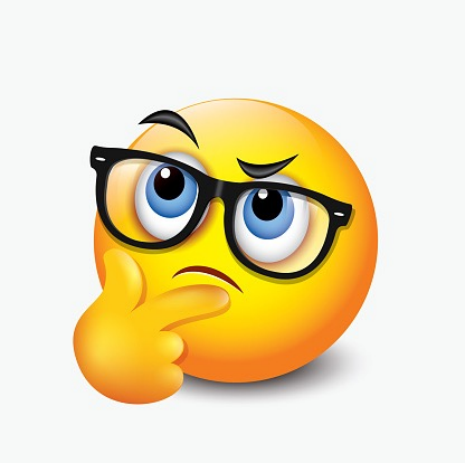
***What We Concluded:***

The fourth “sanity check” has failed.

# Following a Design Process — So Far

- ***Further Conclusion:*** The automaton must remember ***different*** information — or, possibly, ***additional*** information — in order to recognize the language $L$.

# All is Not Lost!



However, what we have done so far is useful — because it can help us to discover the “different information — or, possibly additional information” that is needed.

# Application to the Example — Starting a Second Attempt

***Hypothesis:*** In order to recognize $L$ you must remember the following information.

1. You must remember whether an "a" has already been seen.

   *Explanation:* This is needed to decide what to do if the string seen so far is in $L$, but a "b" is the next symbol that is seen.

2. You must *also* remember whether a "b" has already been seen.

   *Explanation:* This is needed to decide what to do if the string seen so far is in $L$ but an "a" is the next symbol that is seen.

# Application to the Example — Starting a Second Attempt

***Four cases*** concerning the string that has been seen so far — corresponding to four subsets of $\Sigma^\star$ — will now be used:

- $\widehat{S}_0 = \{\omega \in \Sigma^\star \mid \omega$ does not include an "a" and $\omega$ does not include a "b"$\}$

  ***Note:*** $\widehat{S}_0 \subseteq S_0$.

- $\widehat{S}_1 = \{\omega \in \Sigma^\star \mid \omega$ includes at least one "a" and $\omega$ does not include a "b"$\}$

  ***Note:*** $\widehat{S}_1 \subseteq S_0$.

# Application to the Example — Starting a Second Attempt

- $\widehat{S}_2 = \{\omega \in \Sigma^\star \mid \omega$ includes at least one "b" and $\omega$ does not include an "a"$\}$

  ***Note:*** $\widehat{S}_2 \subseteq S_0$.

- $\widehat{S}_3 = \{\omega \in \Sigma^\star \mid \omega$ includes at least one "a" and $\omega$ includes at least one "b"$\}$

  ***Note:*** $\widehat{S}_3 = S_1$.

# Wait a Minute! What Just Happened Here?

- Notice that we are ***rolling back***, and ***starting this design process all over again***.
- In particular: The above information is a new answer for the "question" that was considered at the *beginning* of this design process— and a new collection of subsets of $\Sigma^\star$ corresponding to this.
- ***We have learned something from the first attempt*** — and we are making use of new information that has been discovered — so that answers for questions will be different, and there is a chance that the process will complete, successfully, this time!
- This is one form of a process that is called ***refinement***.

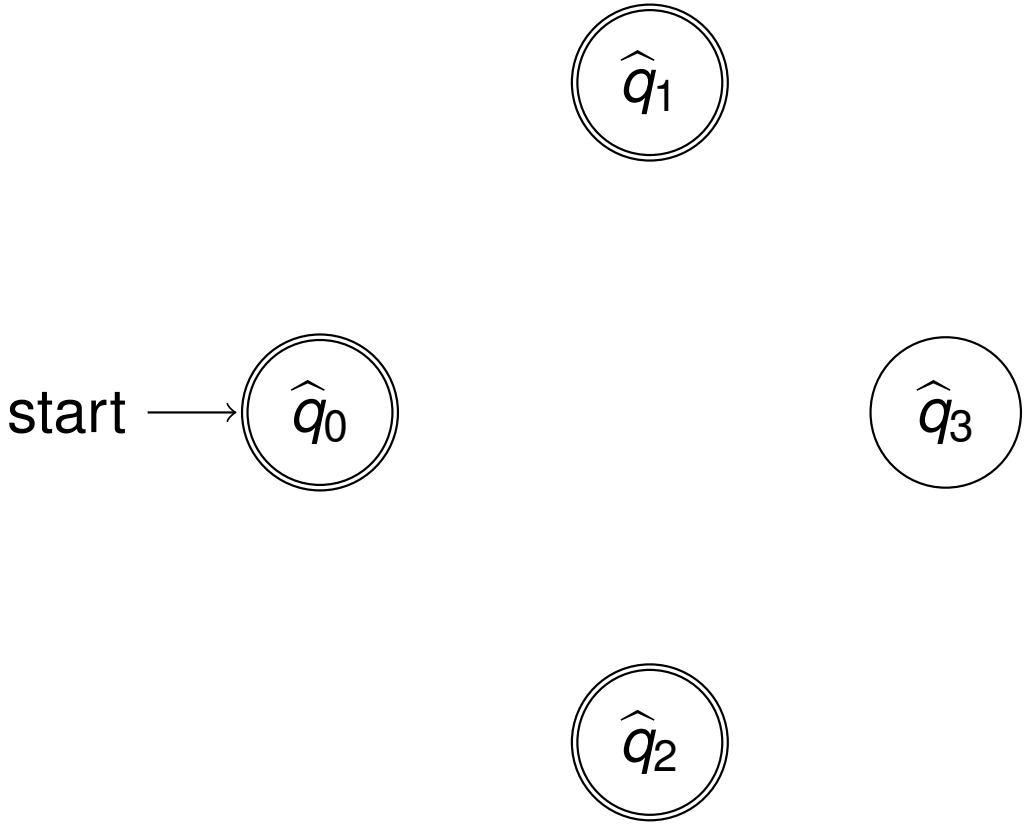
# Proceeding with the "Rolled Back" Process

- The first "sanity check" is passed, once again: Only a finite number of subsets of $\Sigma^\star$ (and corresponding states) have been identified.
- The second "sanity check" is passed as well: It follows from the descriptions of $\widehat{S}_0$, $\widehat{S}_1$, $\widehat{S}_2$ and $\widehat{S}_3$ that every string in $\Sigma^\star$ belongs to *exactly one* of these sets.

  Let $\widehat{q}_0$, $\widehat{q}_1$, $\widehat{q}_2$ and $\widehat{q}_3$ be states corresponding to the subsets $\widehat{S}_0$, $\widehat{S}_1$, $\widehat{S}_2$ and $\widehat{S}_3$, respectively. Then, since $\lambda \in \widehat{S}_0$, $\widehat{q}_0$ is the start state for the automaton being designed.
- The third "sanity check" is also passed because $\widehat{S}_0 \subseteq L$, $\widehat{S}_1 \subseteq L$, $\widehat{S}_2 \subseteq L$, and $\widehat{S}_3 \cap L = \emptyset$. It follows from this that $\widehat{q}_0$, $\widehat{q}_1$ and $\widehat{q}_2$ are all accepting states, and $\widehat{q}_3$ is not.

# The Example, So Far

Here is what we have, so far.




***Note:*** We are now back at the step where the first attempt failed.

# Continuing the Example: Discovering Transactions

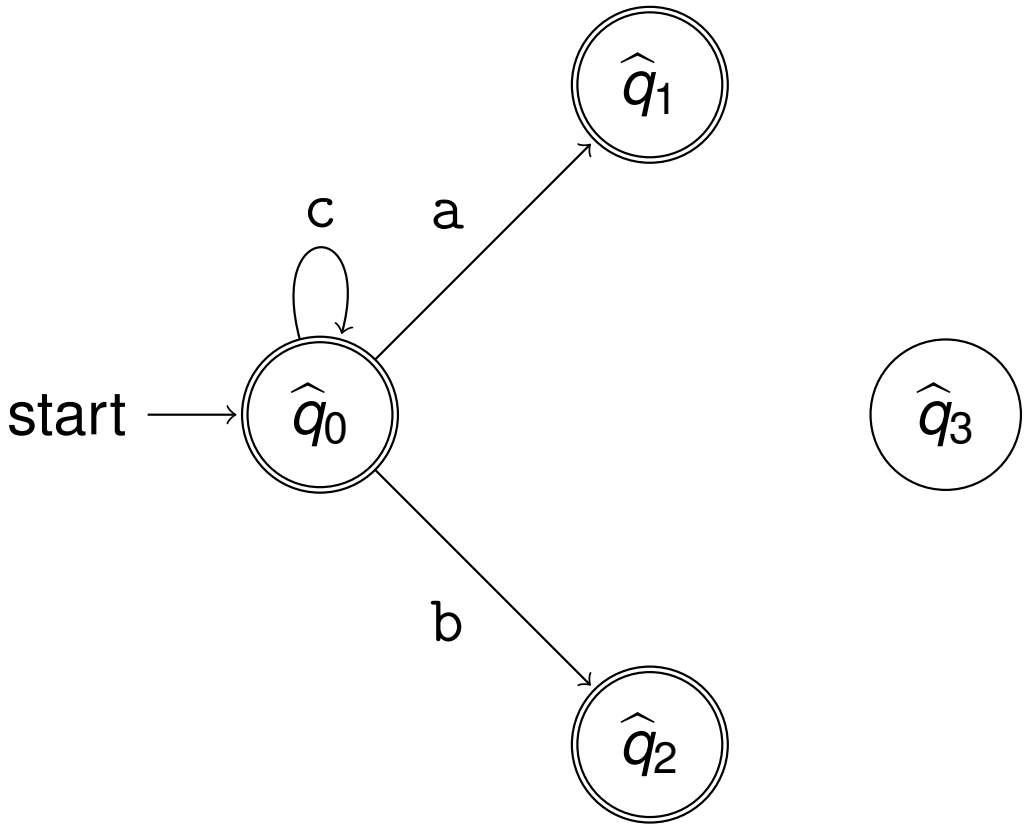
Consider transactions out of the state $\widehat{q}_0$, which corresponds to the set

$$\widehat{S}_0 = \{\omega \in \Sigma^\star \mid \omega \text{ does not include an “a” and } \omega \text{ does not include a “b”}\}.$$

- If $\omega \in \widehat{S}_0$ then $\omega \cdot \mathtt{a}$ includes at least one “a”, but not a “b”, so that $\omega \cdot \mathtt{a} \in \widehat{S}_1$. Thus $\{\omega \cdot \mathtt{a} \mid \omega \in \widehat{S}_0\} \subseteq \widehat{S}_1$, and
$$\delta(\widehat{q}_0, \mathtt{a}) = \widehat{q}_1.$$
- If $\omega \in \widehat{S}_0$ then $\omega \cdot \mathtt{b}$ includes at least one “b”, but not an “a”, so that $\omega \cdot \mathtt{b} \in \widehat{S}_2$. Thus $\{\omega \cdot \mathtt{b} \mid \omega \in \widehat{S}_0\} \subseteq \widehat{S}_2$, and
$$\delta(\widehat{q}_0, \mathtt{b}) = \widehat{q}_2.$$
- If $\omega \in \widehat{S}_0$ then $\omega \cdot \mathtt{c}$ does not include an “a” or a “b”, so that $\omega \cdot \mathtt{c} \in \widehat{S}_0$. Thus $\{\omega \cdot \mathtt{c} \mid \omega \in \widehat{S}_0\} \subseteq \widehat{S}_0$, and
$$\delta(\widehat{q}_0, \mathtt{c}) = \widehat{q}_0.$$

# The Example, So Far

Here is what we have, so far.



# Continuing the Example: Discovering Transactions

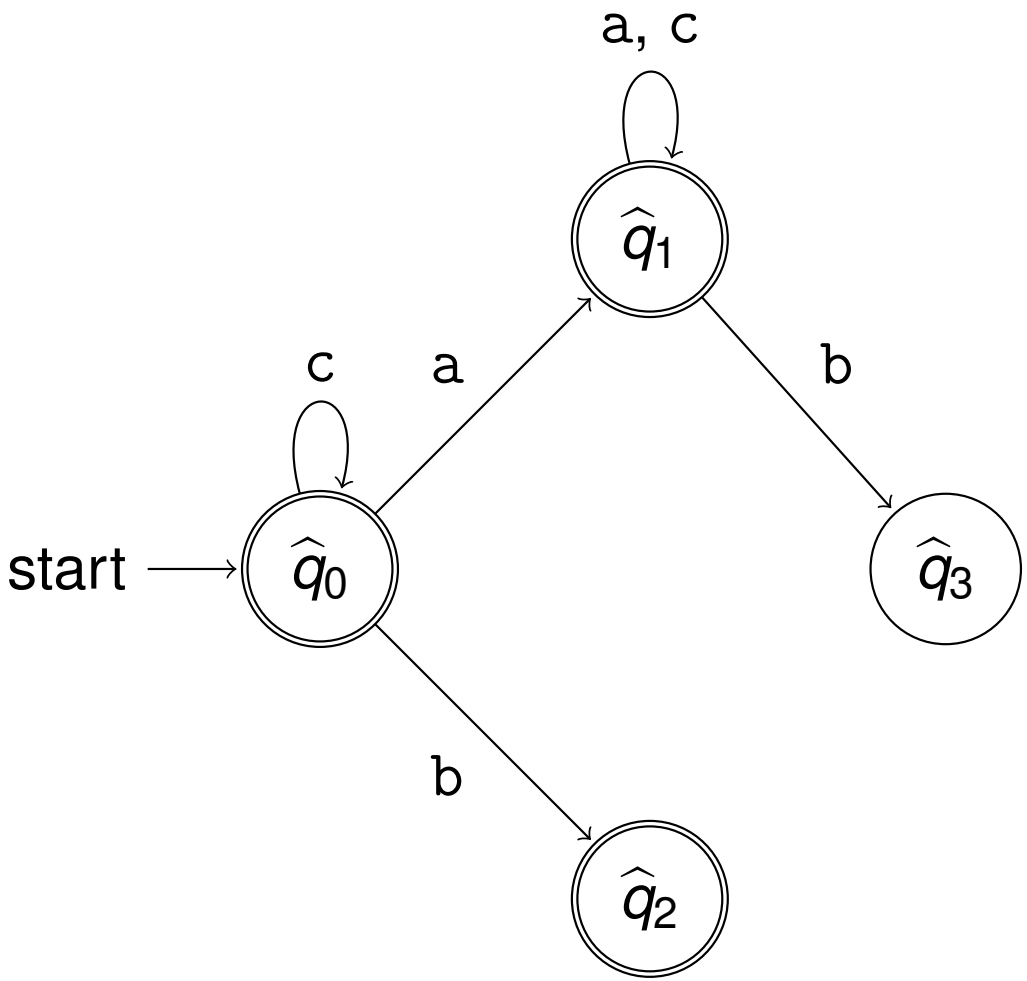
Consider transactions out of the state $\widehat{q}_1$, which corresponds to the set

$$\widehat{S}_1 = \{\omega \in \Sigma^\star \mid \omega \text{ contains at least one “a” and } \omega \text{ does not include a “b”}\}.$$

- If $\omega \in \widehat{S}_1$ then $\omega \cdot \mathtt{a}$ includes at least one “a”, but not a “b”, so that $\omega \cdot \mathtt{a} \in \widehat{S}_1$. Thus $\{\omega \cdot \mathtt{a} \mid \omega \in \widehat{S}_1\} \subseteq \widehat{S}_1$, and
$$\delta(\widehat{q}_1, \mathtt{a}) = \widehat{q}_1.$$
- If $\omega \in \widehat{S}_1$ then $\omega \cdot \mathtt{b}$ includes both an “a” and a “b”, so that $\omega \cdot \mathtt{b} \in \widehat{S}_3$. Thus $\{\omega \cdot \mathtt{b} \mid \omega \in \widehat{S}_1\} \subseteq \widehat{S}_3$, and
$$\delta(\widehat{q}_1, \mathtt{b}) = \widehat{q}_3.$$
- If $\omega \in \widehat{S}_1$ then $\omega \cdot \mathtt{c}$ includes at least one “a”, but not a “b”, so that $\omega \cdot \mathtt{c} \in \widehat{S}_1$. Thus $\{\omega \cdot \mathtt{c} \mid \omega \in \widehat{S}_1\} \subseteq \widehat{S}_1$, and
$$\delta(\widehat{q}_1, \mathtt{c}) = \widehat{q}_1.$$

# The Example, So Far

Here is what we have, so far.



# Continuing the Example: Discovering Transactions

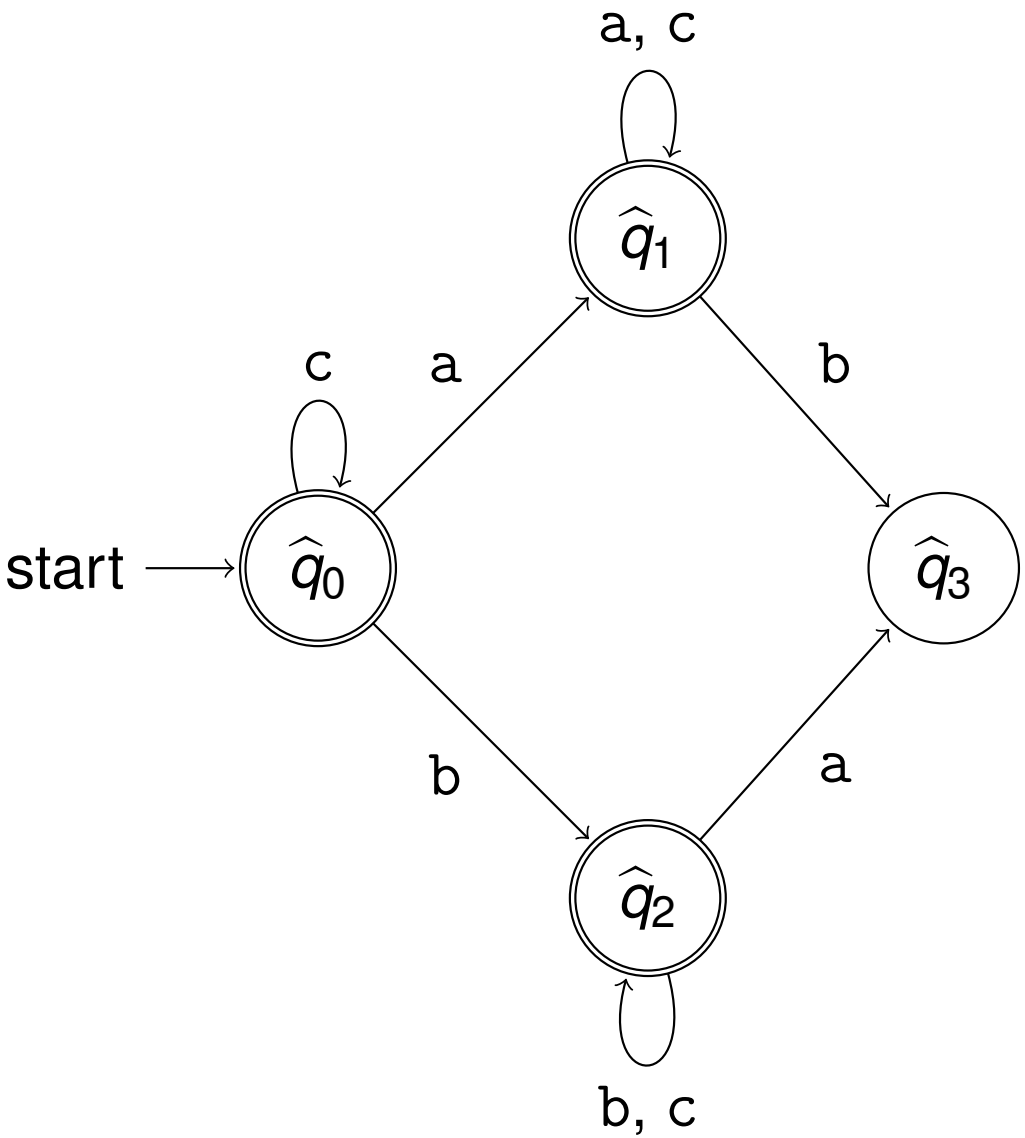
Consider transactions out of the state $\widehat{q}_2$, which corresponds to the set

$$\widehat{S}_2 = \{\omega \in \Sigma^\star \mid \omega \text{ contains at least one “b” and } \omega \text{ does not include an “a”}\}.$$

- If $\omega \in \widehat{S}_2$ then $\omega \cdot \mathtt{a}$ includes both an “a” and a “b”, so that $\omega \cdot \mathtt{a} \in \widehat{S}_3$. Thus $\{\omega \cdot \mathtt{b} \mid \omega \in \widehat{S}_2\} \subseteq \widehat{S}_3$, and
$$\delta(\widehat{q}_2, \mathtt{a}) = \widehat{q}_3.$$
- If $\omega \in \widehat{S}_2$ then $\omega \cdot \mathtt{b}$ includes at least one “b”, but not an “a”, so that $\omega \cdot \mathtt{b} \in \widehat{S}_2$. Thus $\{\omega \cdot \mathtt{b} \mid \omega \in \widehat{S}_2\} \subseteq \widehat{S}_2$, and
$$\delta(\widehat{q}_2, \mathtt{b}) = \widehat{q}_2.$$
- If $\omega \in \widehat{S}_2$ then $\omega \cdot \mathtt{c}$ includes at least one “b”, but not an “a”, so that $\omega \cdot \mathtt{c} \in \widehat{S}_2$. Thus $\{\omega \cdot \mathtt{c} \mid \omega \in \widehat{S}_2\} \subseteq \widehat{S}_2$, and
$$\delta(\widehat{q}_2, \mathtt{c}) = \widehat{q}_2.$$

# The Example, So Far

Here is what we have, so far.



# Continuing the Example: Discovering Transactions

Consider transactions out of the state $\widehat{q}_3$, which corresponds to the set

$$\widehat{S}_3 = \{\omega \in \Sigma^\star \mid \omega \text{ includes an “a” and } \omega \text{ includes a “b”}\}.$$

- If $\omega \in \widehat{S}_3$ then $\omega$ includes both an “a” and a “b”, so that $\omega \cdot \mathtt{a}$, $\omega \cdot \mathtt{b}$, and $\omega \cdot \mathtt{c}$ each contain both an “a” and a “b” as well. Thus $\omega \cdot \mathtt{a}, \omega \cdot \mathtt{b}, \omega \cdot \mathtt{c} \in \widehat{S}_3$, implying that

$$\{\omega \cdot \sigma \mid \omega \in \widehat{S}_3\} \subseteq \widehat{S}_3$$

for $\sigma = \mathtt{a}$, for $\sigma = \mathtt{b}$, and for $\sigma = \mathtt{c}$, so that

$$\delta(\widehat{q}_3, \mathtt{a}) = \delta(\widehat{q}_3, \mathtt{b}) = \delta(\widehat{q}_3, \mathtt{c}) = \widehat{q}_3.$$

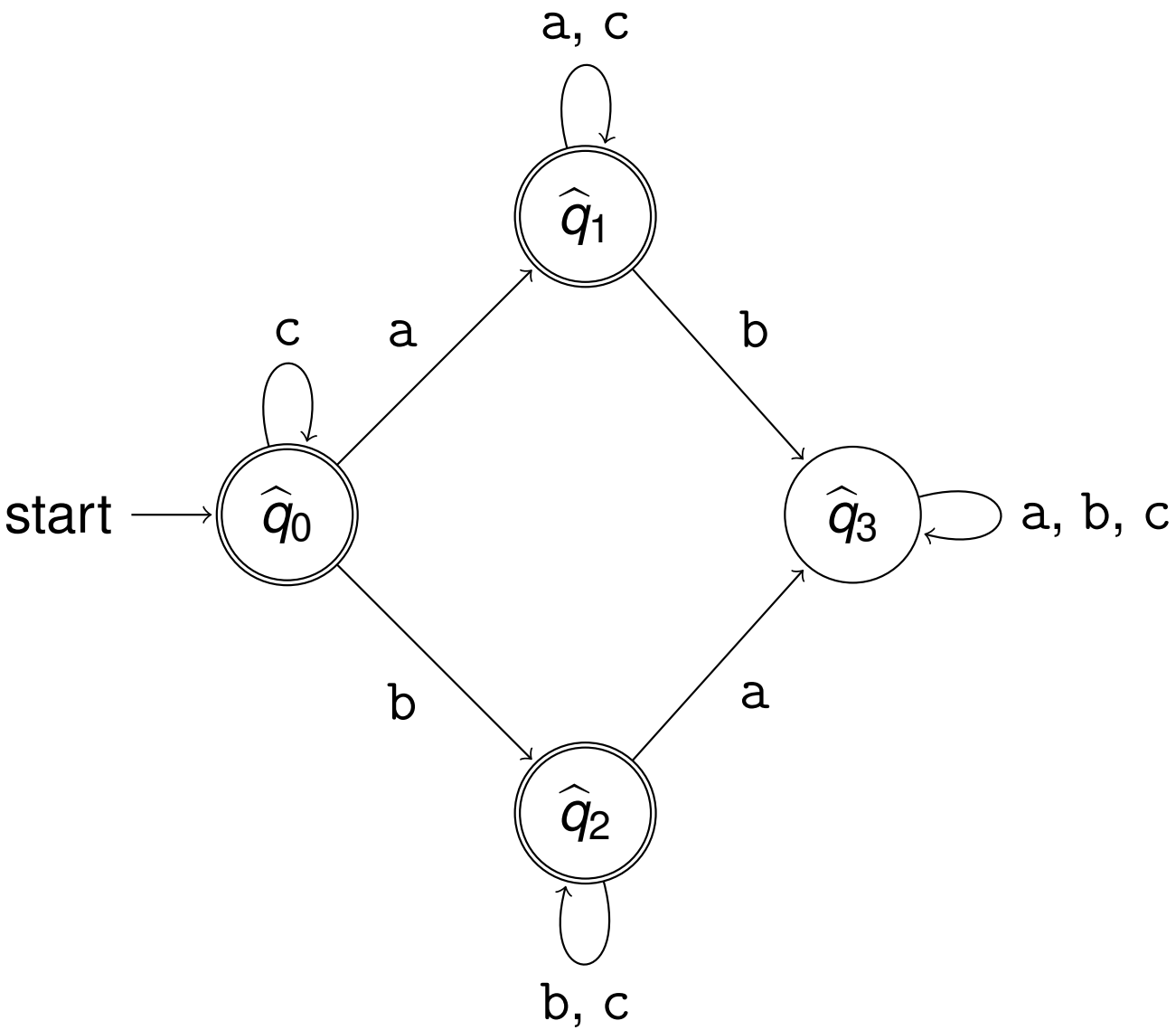
# Continuing the Example: Discovering Transitions

Hurray!

The fourth “sanity check” has now been passed as well. A DFA for the language $L$ — which includes the transitions that have been identified — is as follows.

# The Example, Concluded

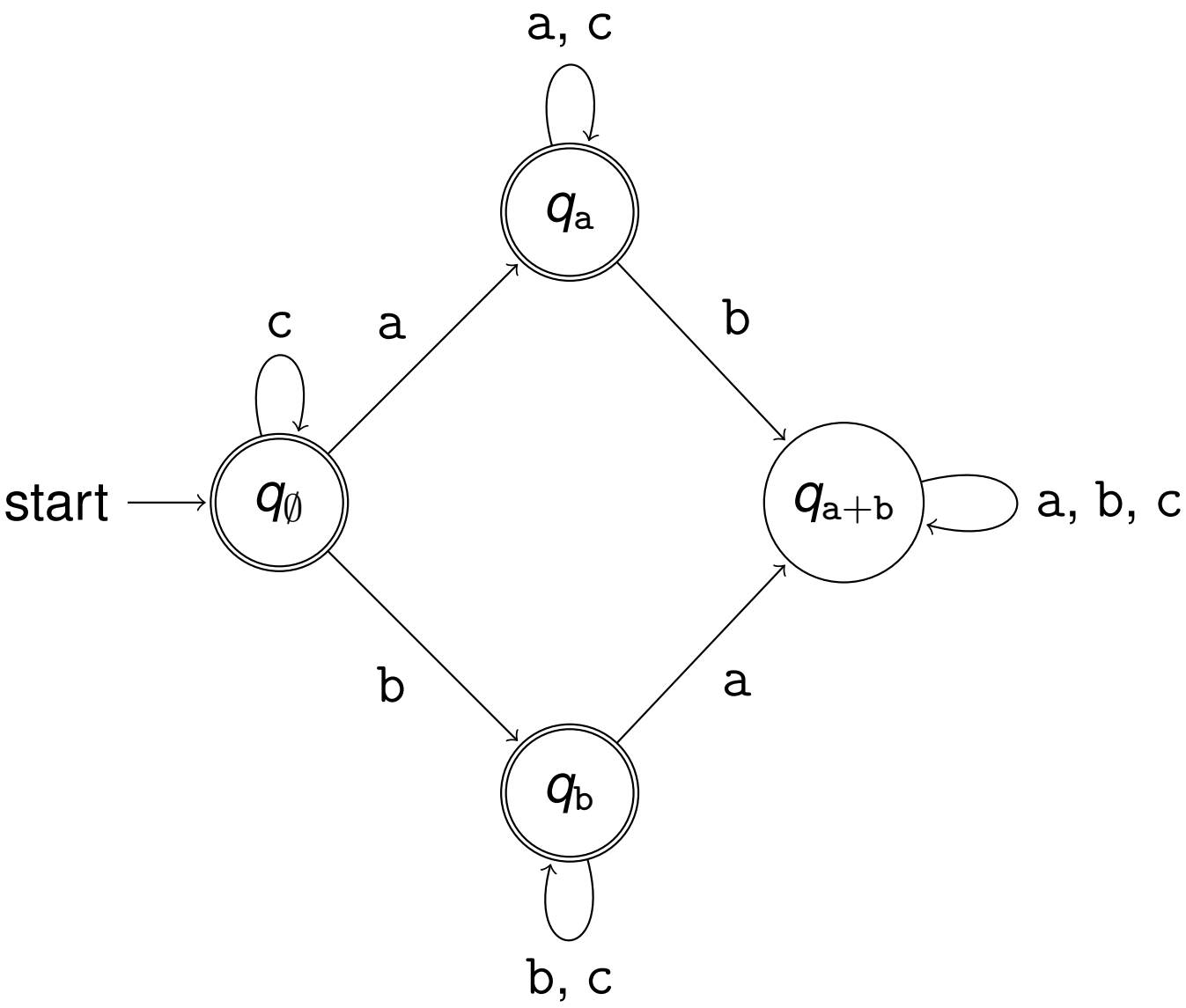
Here is what we now have.



# The Example, Concluded

- Names of subsets, and states, where chosen here just to make the general description and example match.
- We could also have used more descriptive names, like $S_{\emptyset}$ instead of $\widehat{S}_0$, $S_{\mathtt{a}}$ instead of $\widehat{S}_1$, $S_{\mathtt{b}}$ instead of $\widehat{S}_2$, and $S_{\mathtt{a+b}}$ instead of $\widehat{S}_3$.
- If corresponding names of states ($q_{\emptyset}$, $q_{\mathtt{a}}$, $q_{\mathtt{b}}$ and $q_{\mathtt{a+b}}$) are used then the DFA, obtained, would be as shown on the following slide.

# The Example, Concluded



# Suggestions

***Suggestions about Figuring out "What to Remember"***

- Consider the language to be recognized very carefully. This may be enough to figure out what to remember, or it may provide a good start.
- Learn from unsuccessful attempts instead of discarding them. This helped us to design a DFA for $L$!
- Study Examples — and ***Practice!*** As you consider more DFA's for languages you will begin to recognize common situations and patterns.

# Verification

## We're not done yet!

- Why should you — or anyone else — believe that ***this process is correct***, that is, that this process is guaranteed to provide what is claimed (a DFA for the given language) whenever you completed the process, without making mistakes?
- How could you *convince someone else* that a given DFA has a given language?

# A Useful Result

## Theorem (Correctness of a DFA)

*Let $L \subseteq \Sigma^\star$, for an alphabet $\Sigma$, and let*

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

*be a deterministic finite automaton with the same alphabet $\Sigma$. Suppose that (after renaming states, if needed)*

$$Q = \{q_0, q_1, \ldots, q_{n-1}\}$$

*where $n = |Q| \geq 1$. Suppose, as well, that*

$$S_0, S_1, \ldots, S_{n-1}$$

*are subsets of $\Sigma^\star$ such that the following properties are satisfied.*

# A Useful Result

(a) *Every string in* $\Sigma^\star$ *belongs to* ***exactly one*** *of*

$$S_0, S_1, \ldots, S_{n-1}.$$

(b) $\lambda \in S_0$.

(c) $S_i \subseteq L$ *for every integer* $i$ *such that* $0 \leq i \leq n-1$ *and* $q_i \in F$.

(d) $S_i \cap L = \emptyset$ *for every integer* $i$ *such that* $0 \leq i \leq n-1$ *and* $q_i \notin F$.

(e) *The following property is satisfied, for every integer* $i$ *such that* $0 \leq i \leq n-1$ *and for every symbol* $\sigma \in \Sigma$*:*

*"Suppose that* $q_j = \delta(q_i, \sigma)$ *(so that* $0 \leq j \leq n-1$*). Then*

$$\{\omega \cdot \sigma \mid \omega \in S_i\} \subseteq S_j."$$

*Then* $L(M) = L$.

# How This Result is Proved

- Properties (b) and (e) are used to prove that, for every string $\omega \in \Sigma^\star$ and for every integer $i$ such that $0 \leq i \leq n-1$,

$$\delta^\star(q_0, \omega) = q_i \quad \text{if and only if} \quad \omega \in S_i$$

  — by induction on the length of $\omega$.
- Properties (a), (c) and (d) are then used, along with this result, to prove that $L(M) = L$, as claimed.
- A complete proof of this result is given in a supplement for this lecture.

# Using This Result

- You do not need to know how to prove that this result is correct — but you can ***use*** it to prove that a given deterministic finite automaton has a given language.
- ***Note:*** This is, generally, easier to use if you produced a DFA using this design process than it if you got it another way, because most of the work needed to apply the result has already been done.