Extended transition function of a DFA

The next two pages describe the extended transition function of a DFA in a more detailed way than Handout 3.

Formal approach to accepted strings

We define the **extended transition function** $\hat{\delta}$. It takes a state q and an input **string** w to the resulting state. The definition proceeds by **induction** over the length of w.

- Induction basis (w has length 0): in this case, w is the **empty string**, i.e. the string of length 0, for which we write ϵ. We define

$$\hat{\delta}(q, \epsilon) = q.$$

Formal approach to accepted strings

- Induction step (from length l to length $l+1$): in this case, w, which has length $l+1$, is of the form va, where v is a string of length l and a is a symbol. We define

$$\hat{\delta}(q, va) = \delta(\hat{\delta}(q, v), a).$$

This works because, by induction hypothesis, $\hat{\delta}(q, v)$ is already defined.

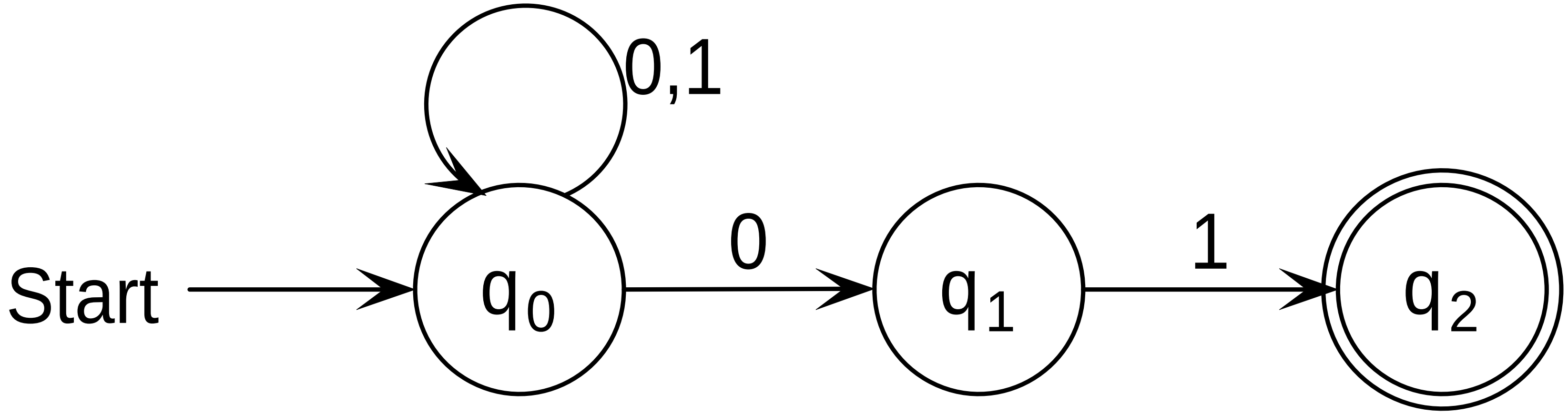
Non-deterministic finite automata (NFA's)

Non-deterministic FA (NFA)

- An NFA is like a DFA, except that it can be in several states at once.
- This can be seen as the ability to guess something about the input.
- Useful for searching texts.

NFA: example

An NFA accepting all strings that end in 01:

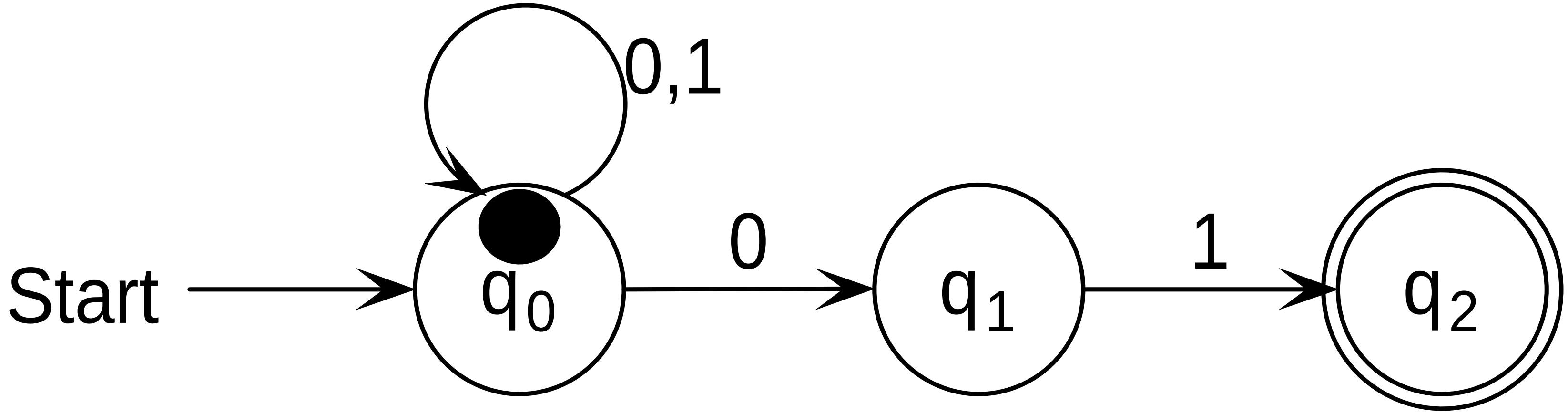


It is non-deterministic because input 0 in state q_0 can lead to both q_0 and q_1.

NFA example - correction

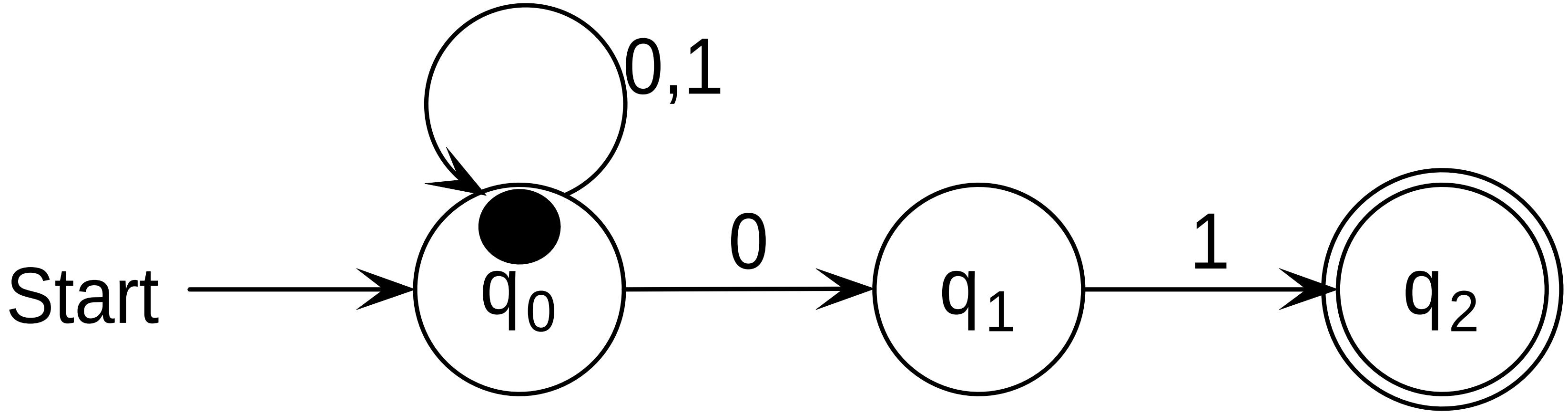
The example on the next few pages is a corrected version of the **wrong** example in Handout 3!

Using the NFA



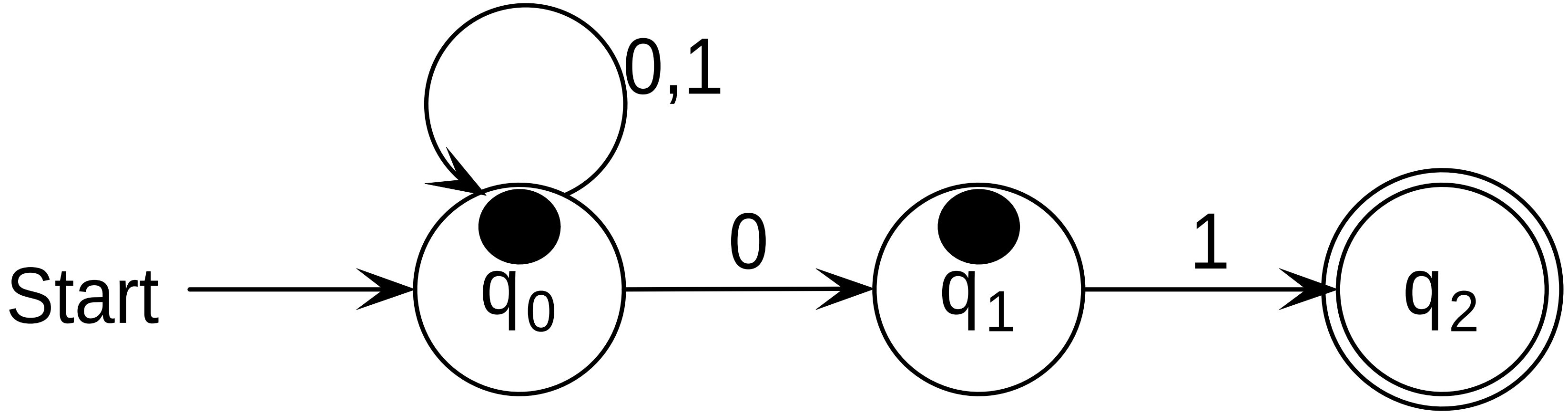
Suppose the input string is 100101. The NFA starts in state q_0, as indicated by the token.

Using the NFA



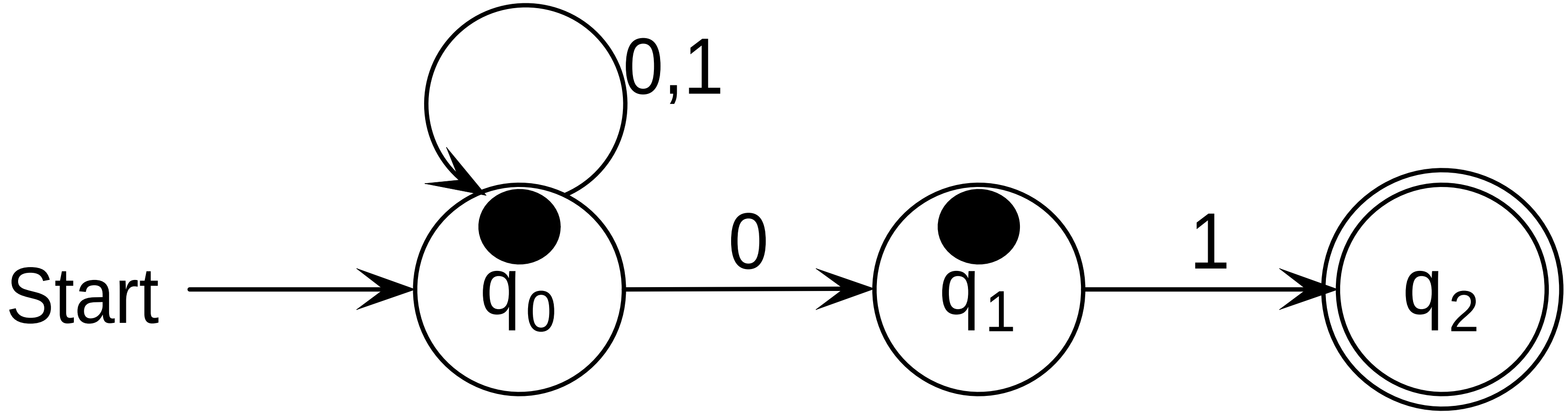
The remaining input string is 100101. The NFA reads the first symbol, 1. It remains in state q_0.

Using the NFA



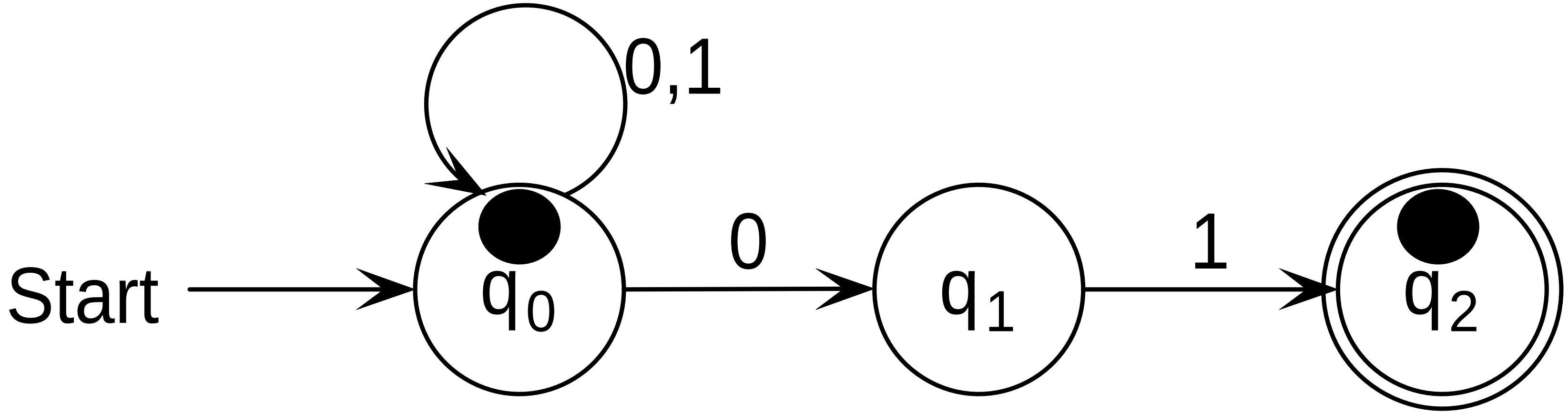
The remaining input string is 00101. The NFA reads the next symbol, 0. The resulting possible states are q_0 or q_1.

Using the NFA



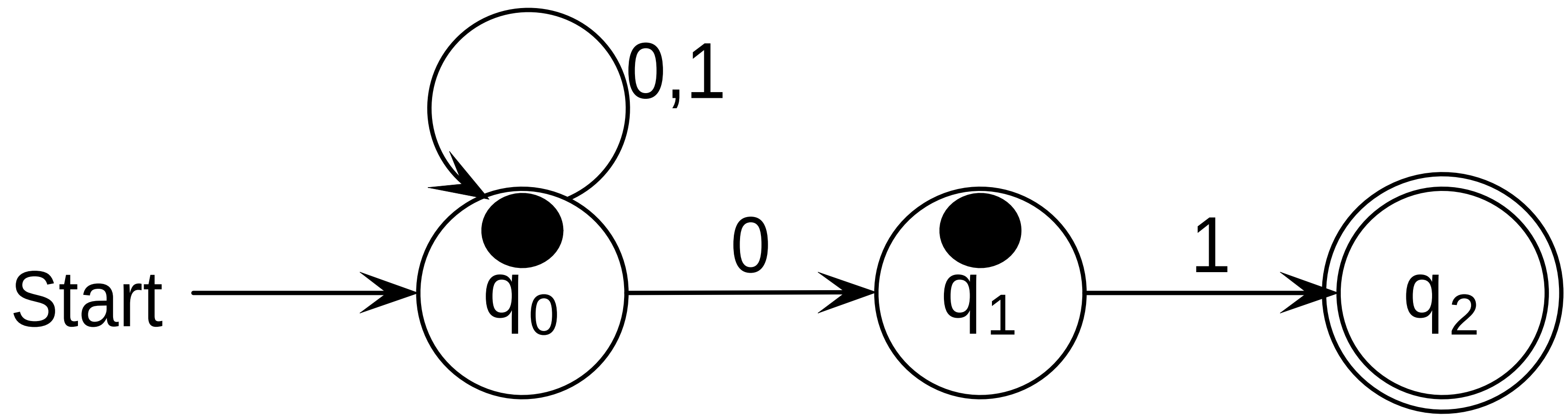
The remaining input string is 0101. The NFA reads the next symbol, 0. The resulting possible states are still q_0 or q_1.

Using the NFA



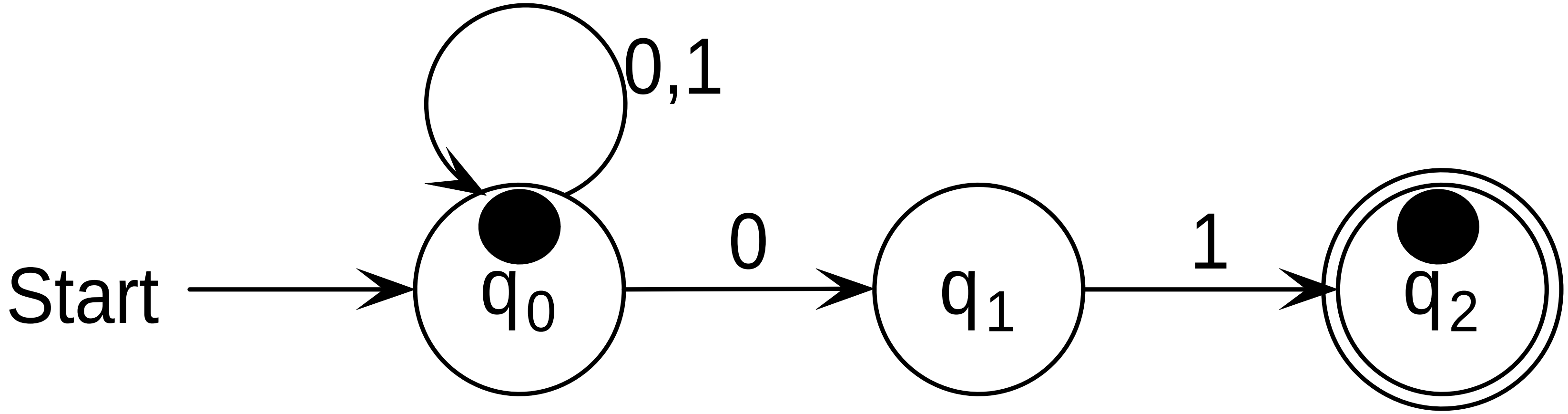
The remaining input string is 101. The NFA reads the next symbol, 1. The resulting possible states are q_0 and q_2. (Because q_2 is a final states, this means that the word so far, 1001, would be accepted.)

Using the NFA



The remaining input string is 01. The NFA reads the next symbol, 0. There is no transition for 0 from q_2, so the token on q_2 **dies**. The resulting possible states are q_0 or q_1.

Using the NFA



The remaining input string is 1. The NFA reads the next symbol, 1. The possible states are q_0 and q_2. Because q_2 is final, the NFA accepts the word, 100101.

Formal definition of NFA

Definition. A **non-deterministic finite automaton (NFA)** consists of

- a finite set of **states**, often denoted Q,
- a finite set Σ of **input symbols**,
- a **transition function** $\delta : Q \times \Sigma \to P(Q)$,
- a **start state** $q_0 \in Q$, and
- a set $F \subseteq Q$ of **final** or **accepting states**.

Difference between NFA and DFA

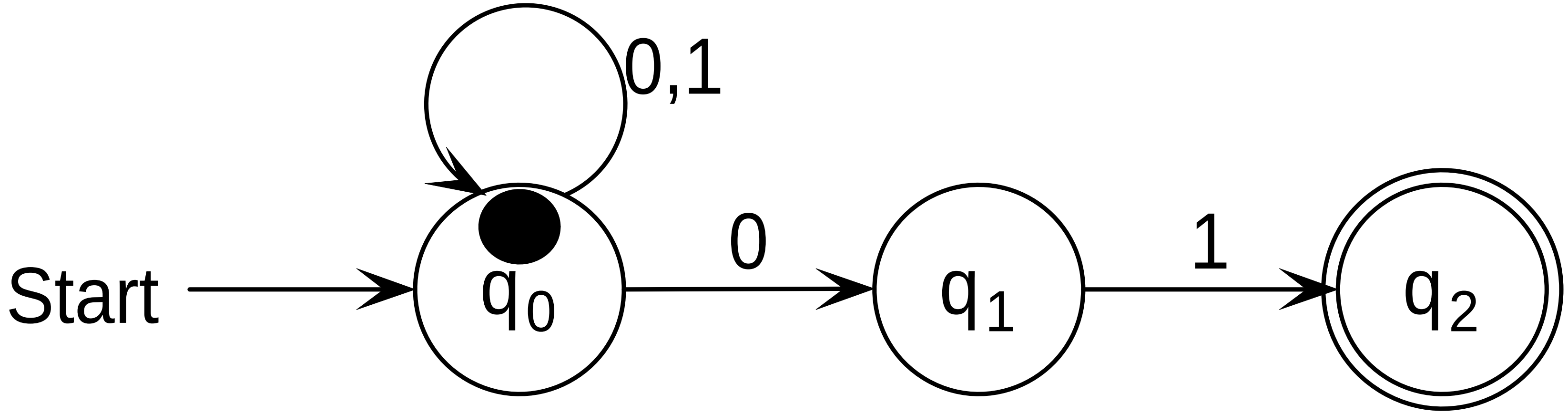
Suppose that q is a state and a is an input symbol.

- In a DFA, we have $\delta(q, a) \in Q$, that is, $\delta(q, a)$ is a state.
- In a NFA, we have $\delta(q, a) \in P(Q)$, that is, $\delta(q, a)$ is a **set of states**; it can be seen as the possible states that can result from input a in state q.

Formal approach to accepted strings

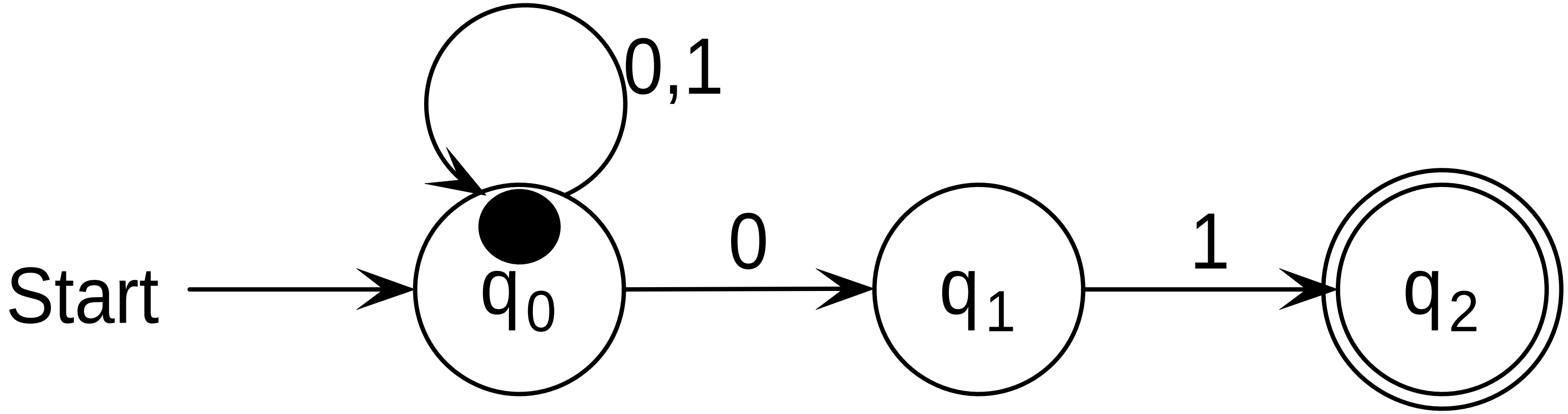
- We are aiming to describe the language $L(A)$ accepted by a NFA A.
- This description is similar to the DFA case, but a bit more sophisticated.
- As in the DFA case, we first define the **extended transition function**: $\hat{\delta} : Q \times \Sigma \to P(Q)$.
- That function $\hat{\delta}$ will be used to define $L(A)$.

Example of $\hat{\delta}$ (input string 100101)



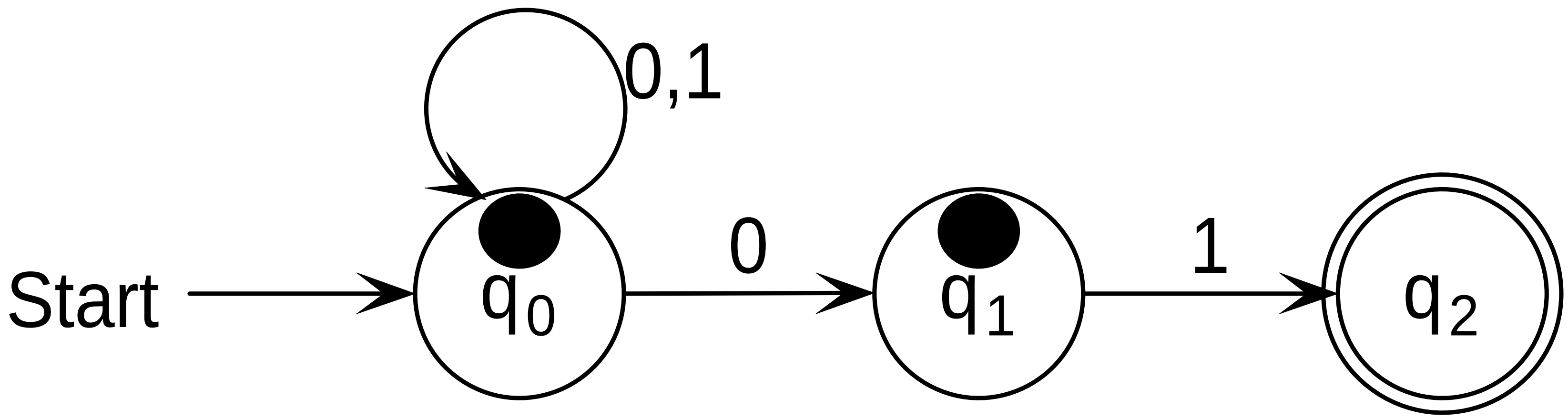
Before reading any symbols, the set of possible states is $\hat{\delta}(q_0, \epsilon) = \{q_0\}$.

Example of $\hat{\delta}$ (input string: 100101)



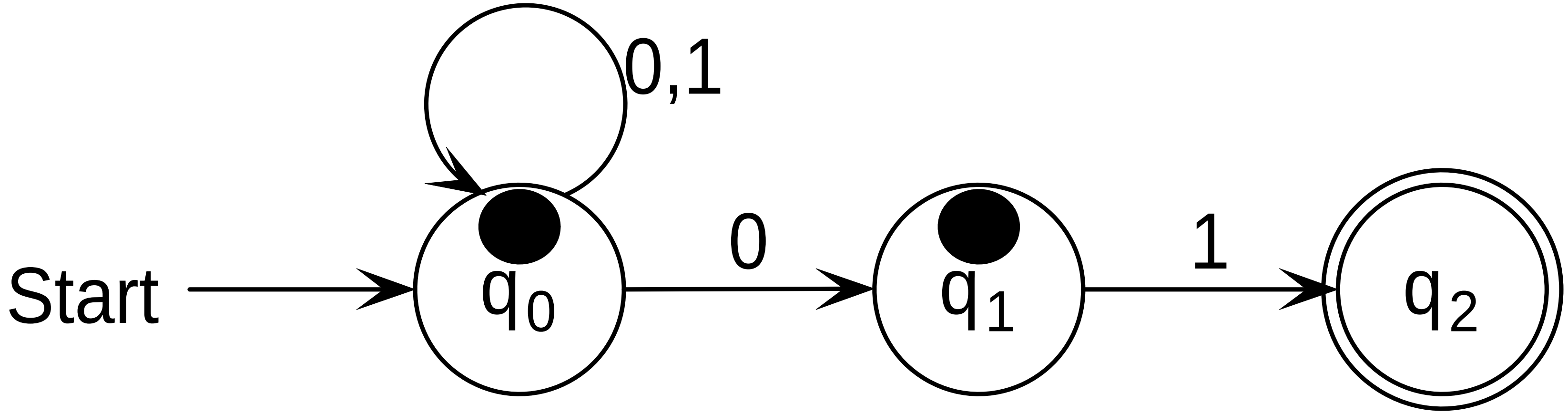
We have $\hat{\delta}(q_0, 1) = \{q_0\}$.

Example of $\hat{\delta}$ (input string: 100101)



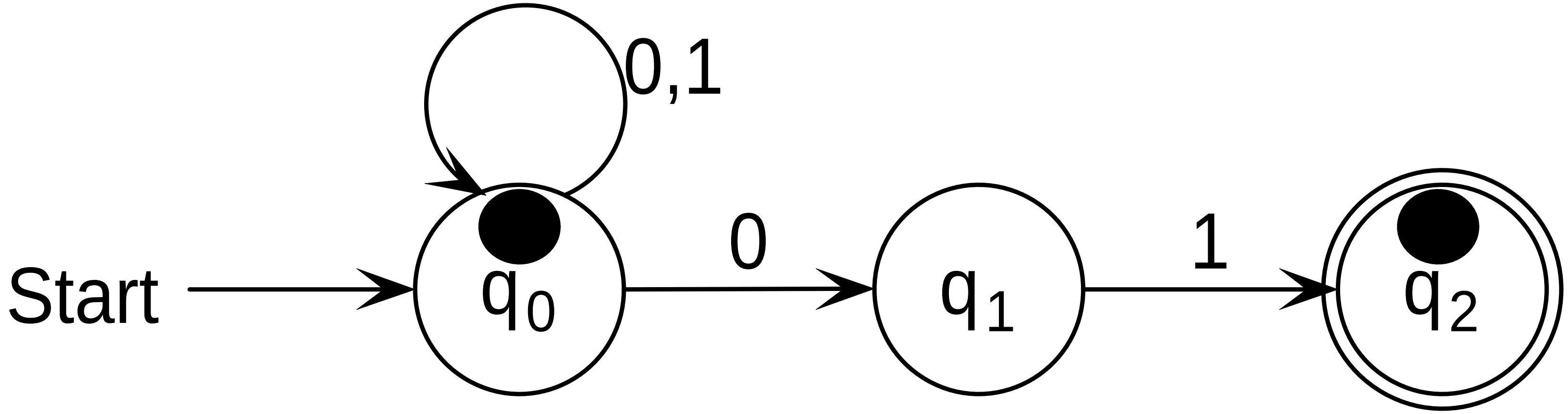
We have $\hat{\delta}(q_0, 10) = \{q_0, q_1\}$.

Example of $\hat{\delta}$ (input string: 100101)



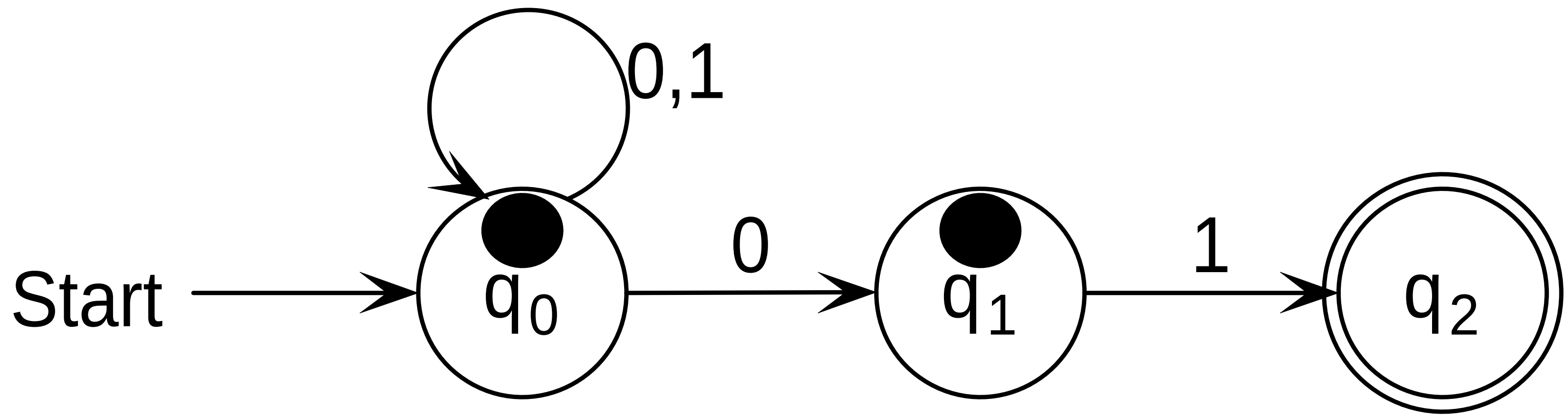
We have $\hat{\delta}(q_0, 100) = \{q_0, q_1\}$.

Example of $\hat{\delta}$ (input string: 100101)



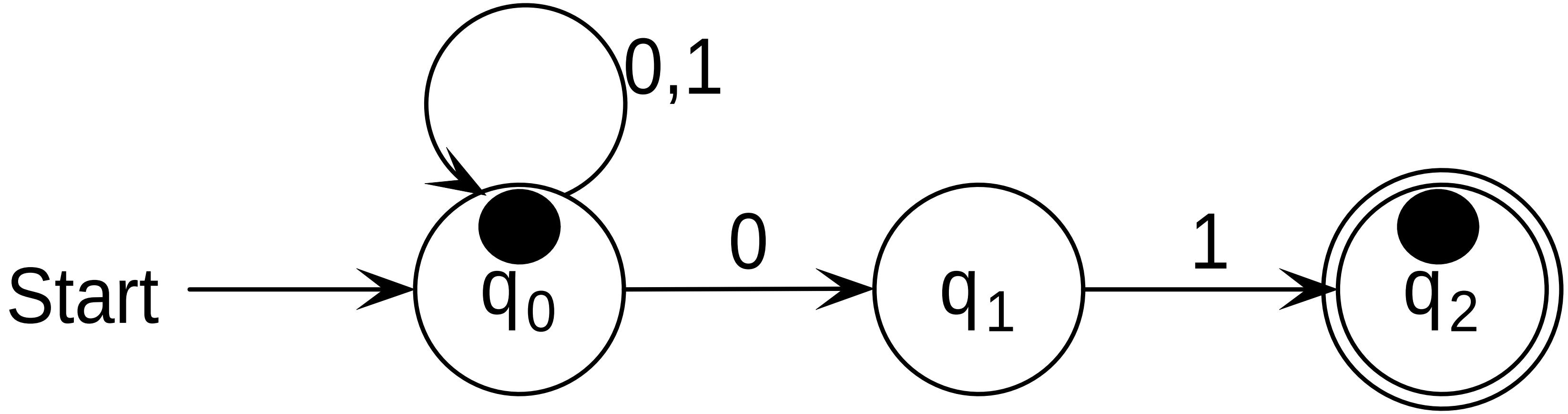
We have $\hat{\delta}(q_0, 1001) = \{q_0, q_2\}$.

Example of $\hat{\delta}$ (input string: 100101)



We have $\hat{\delta}(q_0, 10010) = \{q_0, q_1\}$.

Example of $\hat{\delta}$ (input string: 100101)



We have $\hat{\delta}(q_0, 100101) = \{q_0, q_2\}$. Because $\{q_0, q_2\} \cap F = \{q_0, q_2\} \cap \{q_2\} = \{q_2\} \neq \emptyset$, the NFA accepts.

Formal definition of $\hat{\delta}$

Definition. The **extended transition function** $\hat{\delta} : Q \times \Sigma \to P(Q)$ of an NFA is defined inductively as follows:

- Induction basis (length 0):

$$\hat{\delta}(q, \epsilon) = \{q\}$$

- Induction step (from length l to length $l+1$):

$$\hat{\delta}(q, va) = \bigcup_{q' \in \hat{\delta}(q,v)} \delta(q', a).$$

The language of an NFA

- Intuitively, the language of a DFA A is the set of strings w that lead from the start state to an accepting possible state.
- Formally, the language $L(A)$ accepted by the FA A is defined as follows:

$$L(A) = \{w \mid \hat{\delta}(q_0, w) \cap F \neq \emptyset\}.$$

Exercise

Give NFA to accept the following languages.

1. The set of strings over an alphabet $\{0, 1, \ldots, 9\}$ such that the final digit has appeared before.
2. The set of strings over an alphabet $\{0, 1, \ldots, 9\}$ such that the final digit has **not** appeared before.
3. The set of strings of 0's and 1's such that there are two 0's separated by a number of positions that is a multiple of 4.

DFA's and NFA's

- Evidently, DFA's are precisely those NFA's for which the set of states $\delta(q, a)$ has exactly one element for all q and a.
- So, trivially, every language accepted by a DFA is also accepted by some NFA.
- Is every language accepted by an NFA also accepted by some DFA?
- Surprisingly, the answer is "yes"!

Simulation of an NFA by a DFA

Let $N = (Q_N, \Sigma, \delta_N, q_0^N, F_N)$ be a NFA. The equivalent DFA D is obtained from the so-called **powerset construction** (also called “subset construction”.) We define

$$D = (Q_D, \Sigma, \delta_D, q_0^D, F_D),$$

where…

Simulation of an NFA by a DFA

- The alphabet of D is that of N.
- The states of D are sets of states of N:

$$Q_D = P(Q_N)$$

- The initial state q_0^D of D is $\{q_0^N\}$.

Simulation of an NFA by a DFA

- The final states of D are those sets that contain the final state of N:

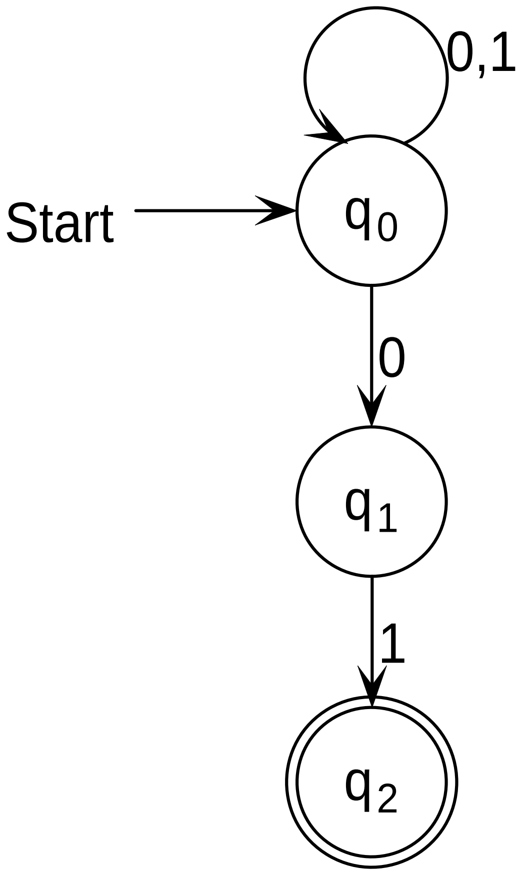
$$F_D = \{S \in P(Q_N) \mid S \cap F_N \neq \emptyset\}$$

- The transition function of D arises from the transition function of N as follows:

$$\delta_D(S, a) = \bigcup_{q' \in S} \delta_N(q', a)$$

Example of powerset construction: table

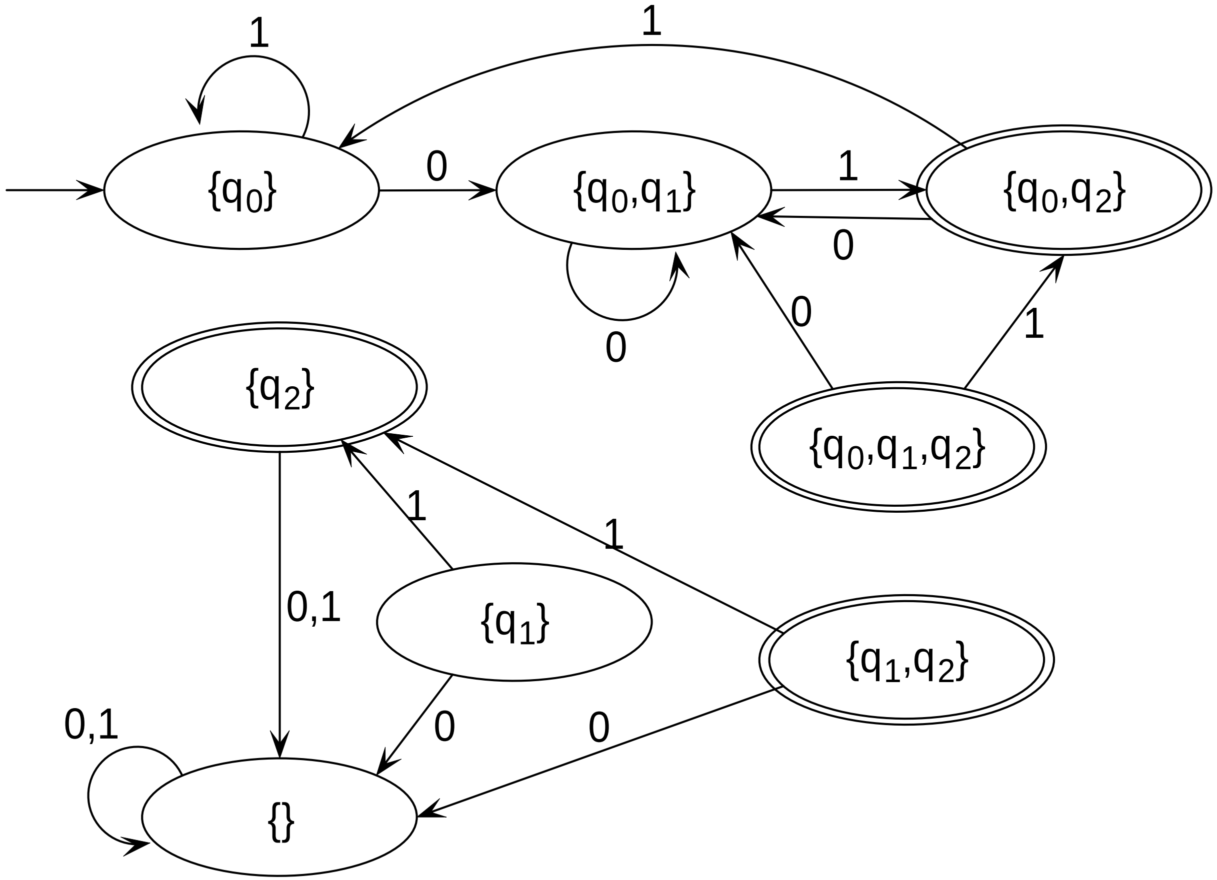
NFA



DFA

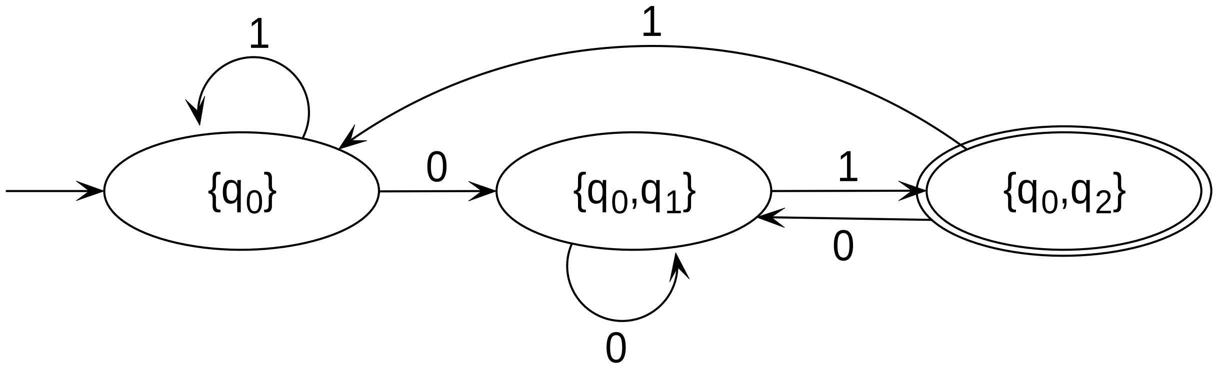
	0	1
$\emptyset$	$\emptyset$	$\emptyset$
$\rightarrow \{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_1\}$	$\emptyset$	$\{q_2\}$
$*\{q_2\}$	$\emptyset$	$\emptyset$
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$
$*\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0\}$
$*\{q_1, q_2\}$	$\emptyset$	$\{q_2\}$
$*\{q_0, q_1, q_2\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$

Example of powerset construction: graph



Transition graph of the resulting DFA.

Example of powerset construction: graph



Optionally, we can remove the unreachable states of the DFA.

Proposition about the simulation

Proposition. For every NFA N, there is a DFA D such that $L(D) = L(N)$.

Proof of the proposition (part 1/3)

First, we show that for every string w we have

$$\widehat{\delta_D}(\{q_0\}, w) = \widehat{\delta_N}(q_0, w) \tag{1}$$

We proceed by induction on the length l of w.

- Base case ($l = 0$): in this case, w is the empty string, ϵ. We have

$$\begin{aligned} \widehat{\delta_D}(\{q_0\}, \epsilon) &= \{q_0\} && \text{(by defn. of } \widehat{\delta_D}\text{)} \\ &= \widehat{\delta_N}(q_0, \epsilon) && \text{(by defn. of } \widehat{\delta_N}\text{).} \end{aligned}$$

Proof of the proposition (part 2/3)

- Induction step (from l to $l+1$): in this case, w, which is of length $l+1$, is of the form va, where v is a string of length l and a is a symbol. We have

$$\begin{aligned}
\widehat{\delta_D}(\{q_0\}, va) &= \delta_D(\widehat{\delta_D}(\{q_0\}, v), a) && \text{(by defn. of } \widehat{\delta_D}\text{)} \\
&= \delta_D(\widehat{\delta_N}(q_0, v), a) && \text{(by indn. hypoth.)} \\
&= \bigcup_{q' \in \widehat{\delta_N}(q_0, v)} \delta_N(q', a) && \text{(by defn. of } \delta_D\text{)} \\
&= \widehat{\delta_N}(q_0, va) && \text{(by defn. of } \widehat{\delta_N}\text{).}
\end{aligned}$$

Proof of the proposition (part 3/3)

Finally, we use Equation **??**, which we just proved, to prove that the languages of D and N are equal:

$$
\begin{aligned}
w \in L(D) &\iff \widehat{\delta_D}(\{q_0\}, w) \in F_D && \text{(by defn. of } L(D)\text{)} \\
&\iff \widehat{\delta_N}(q_0, w) \in F_D && \text{(by Equation ??)} \\
&\iff \widehat{\delta_N}(q_0, w) \cap F_N \neq \emptyset && \text{(by defn. of } F_D\text{)} \\
&\iff w \in L(N) && \text{(by defn. of } L(N)\text{).}
\end{aligned}
$$

Languages accepted by DFAs and NFAs

The proposition implies:

Corollary. A language L is accepted by some DFA if and only if L is accepted by some NFA.

Proof. $\Rightarrow$: this is the powerset construction we have just seen.

$\Leftarrow$: this is true because every DFA is a special case of an NFA, as observed earlier.

Warning

- Let N be an NFA, and let D be the DFA that arises from the powerset construction.
- As we have seen, we have $Q_D = P(Q_N)$.
- So, if Q_N has size k, then the size of Q_D is 2^k.
- This exponential growth of the number of states makes the powerset construction unusable in practice.
- It can be shown that removing unreachable states does not prevent this exponential growth.

Exercise

Convert the following NFA to a DFA:

	0	1
$\rightarrow p$	$\{p, q\}$	$\{p\}$
q	$\{r\}$	$\{r\}$
r	$\{s\}$	$\{\}$
$*s$	$\{s\}$	$\{s\}$.

Exercise

Convert the following NFA to a DFA:

	0	1
$\rightarrow p$	$\{q, s\}$	$\{q\}$
$*q$	$\{r\}$	$\{q, r\}$
r	$\{s\}$	$\{p\}$
$*s$	$\{\}$	$\{p\}.$

Exercise

Convert the following NFA to a DFA:

	0	1
$\to p$	$\{p, q\}$	$\{p\}$
q	$\{r, s\}$	$\{t\}$
r	$\{p, r\}$	$\{t\}$
$*s$	$\{\}$	$\{\}$
$*t$	$\{\}$	$\{\}$

Describe informally the language accepted by this NFA accept? (Don't worry if you need tutor's help for this.)