Language accepted by an NFA$^{\varepsilon}$

$M = (Q, \Sigma, \Delta, s, F, T)$

- Look at paths in the transition graph (including ε-transitions) from start state to *some* accepting state.
- Each such path gives a string in Σ^*, namely the string of non-ε labels that occur along the path.
- The set of all such strings is by definition **the language accepted by** M, written $L(M)$.

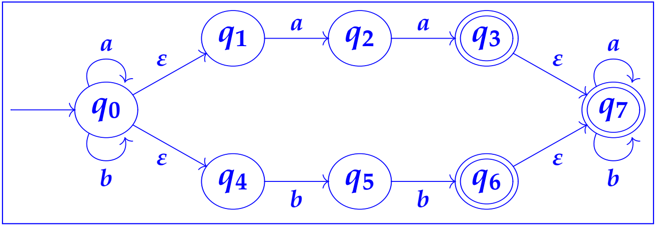
Notation: write $q \stackrel{u}{\Rightarrow} q'$ to mean that there is a path in M from state q to state q' whose non-ε labels form the string $u \in \Sigma^*$.

An **NFA with ε-transitions** (NFA$^\varepsilon$)

$$M = (Q, \Sigma, \Delta, s, F, T)$$

is an NFA $(Q, \Sigma, \Delta, s, F)$ together with a subset $T \subseteq Q \times Q$, called the **ε-transition relation**.

Example:



For this NFA$^\varepsilon$ we have, e.g.: $q_0 \stackrel{aa}{\Rightarrow} q_2$, $q_0 \stackrel{aa}{\Rightarrow} q_3$ and $q_0 \stackrel{aa}{\Rightarrow} q_7$.

In fact the language of accepted strings is equal to the set of strings matching the regular expression $(a|b)^*(aa|bb)(a|b)^*$.

Sets of Languages Accepted by Finite Automata

- every DFA is an NFA (with transition mapping Δ being a next-state function δ)

Sets of Languages Accepted by Finite Automata

- every DFA is an NFA (with transition mapping Δ being a next-state function δ)
- every NFA is an NFA$^\varepsilon$ (with empty ε-transition relation)

Sets of Languages Accepted by Finite Automata

- every DFA is an NFA (with transition mapping Δ being a next-state function δ)
- every NFA is an NFA$^{\varepsilon}$ (with empty ε-transition relation)

Sets of Languages Accepted By Finite Automata

- every DFA is an NFA (with transition mapping Δ being a next-state function δ)
- every NFA is an NFA$^{\varepsilon}$ (with empty ε-transition relation)

clearly

$$L(\mathbb{DFA}) \subseteq L(\mathbb{NFA}) \subseteq L(\mathbb{NFA}^{\epsilon})$$

Sets of Languages Accepted By Finite Automata

- every DFA is an NFA (with transition mapping Δ being a next-state function δ)
- every NFA is an NFA$^{\varepsilon}$ (with empty ε-transition relation)

clearly

$$L(\mathbb{DFA}) \subseteq L(\mathbb{NFA}) \subseteq L(\mathbb{NFA}^{\epsilon})$$

But

$$L(\mathbb{DFA}) \subset L(\mathbb{NFA}) \subset L(\mathbb{NFA}^{\epsilon}) ???$$

NFA^ε accepts if there exists a path...

DFA: path is determined one symbol at a time

Let Q be the states of some NFA^ε. What if we thought, one symbol at a time, about the states we could be in, or more precisely the subset of Q containing the states we could be in

NFA^{ε} accepts if there exists a path...

DFA: path is determined one symbol at a time

Let Q be the states of some NFA^{ε}. What if we thought, one symbol at a time, about the states we could be in, or more precisely the subset of Q containing the states we could be in

Then we could construct a new DFA whose states were taken from the powerset of Q from the NFA^{ε}

Subset Construction

Given an NFA$^\varepsilon$ M with states Q construct a DFA PM whose states are subsets of the states of M

Subset Construction

Given an NFA$^{\varepsilon}$ M with states Q construct a DFA PM whose states are subsets of the states of M

the start state in PM would be a set containing the start state of M together with any states that can be reached by ε-transitions from that state.

Subset Construction

Given an NFA$^\varepsilon$ M with states Q construct a DFA PM whose states are subsets of the states of M

the start state in PM would be a set containing the start state of M together with any states that can be reached by ε-transitions from that state.

accepting states in PM would be any subset containing an accepting state of M

Subset Construction

Given an NFA$^{\varepsilon}$ M with states Q construct a DFA PM whose states are subsets of the states of M

the start state in PM would be a set containing the start state of M together with any states that can be reached by ε-transitions from that state.

accepting states in PM would be any subset containing an accepting state of M

alphabet is the same as the alphabet of M

Subset Construction

Given an NFA^{ε} M with states Q construct a DFA PM whose states are subsets of the states of M

the start state in PM would be a set containing the start state of M together with any states that can be reached by ε-transitions from that state.

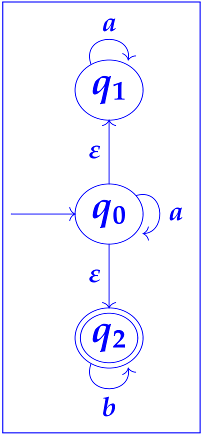
accepting states in PM would be any subset containing an accepting state of M

alphabet is the same as the alphabet of M

That just leaves δ

Example of the subset construction

M



next-state function for PM

	a	b
$\emptyset$	$\emptyset$	$\emptyset$
$\{q_0\}$	$\{q_0, q_1, q_2\}$	$\{q_2\}$
$\{q_1\}$	$\{q_1\}$	$\emptyset$
$\{q_2\}$	$\emptyset$	$\{q_2\}$
$\{q_0, q_1\}$	$\{q_0, q_1, q_2\}$	$\{q_2\}$
$\{q_0, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_2\}$
$\{q_1, q_2\}$	$\{q_1\}$	$\{q_2\}$
$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_2\}$

A word about $\emptyset$ in the subset construction

Potential for confusion

- The DFA has a state which corresponds to the empty set of states in the NFA$^{\varepsilon}$ which we have designated as $\emptyset$.
- Once you enter this state we get stuck in it. Why?
- Could rewrite (next slide)

DFA State	subset of NFA$^\varepsilon$	a	b
S_1	$\emptyset$	S_1	S_1
S_2	$\{q_0\}$	S_8	S_4
S_3	$\{q_1\}$	S_3	S_1
S_4	$\{q_2\}$	S_2	S_4
S_5	$\{q_0, q_1\}$	S_8	S_4
S_6	$\{q_0, q_2\}$	S_8	S_4
S_7	$\{q_1, q_2\}$	S_3	S_4
S_8	$\{q_0, q_1, q_2\}$	S_8	S_4

Noting that S_8 is the start state (why?) we could eliminate states that can't be reached (i.e. S_2, S_5, S_6 and S_7; and thence S_3) if we cared. Here we don't. (Care that is).

Theorem. For each NFA$^\varepsilon$ $M = (Q, \Sigma, \Delta, s, F, T)$ there is a DFA $PM = (\mathcal{P}(Q), \Sigma, \delta, s', F')$ accepting exactly the same strings as M, i.e. with $L(PM) = L(M)$.

Definition of PM:

- set of states is the powerset $\mathcal{P}(Q) = \{S \mid S \subseteq Q\}$ of the set Q of states of M
- same input alphabet Σ as for M
- next-state function maps each $(S, a) \in \mathcal{P}(Q) \times \Sigma$ to $\delta(S, a) \triangleq \{q' \in Q \mid \exists q \in S.\, q \stackrel{a}{\Rightarrow} q' \text{ in } M\}$
- start state is $s' \triangleq \{q' \in Q \mid s \stackrel{\varepsilon}{\Rightarrow} q'\}$
- subset of accepting sates is $F' \triangleq \{S \in \mathcal{P}(Q) \mid S \cap F \neq \emptyset\}$

To prove the theorem we show that $L(M) \subseteq L(PM)$ and $L(PM) \subseteq L(M)$.

Consider a string $a_1 a_2 \ldots a_n \in L(M)$, i.e. is accepted by our NFA$^\varepsilon$ M

Then we have

$$s \overset{a_1}{\Rightarrow} q_1 \overset{a_2}{\Rightarrow} \ldots \overset{a_n}{\Rightarrow} q_n \quad \in F \text{ in } M$$

Consider a string $a_1 a_2 \ldots a_n \in L(M)$, i.e. is accepted by our NFA$^\varepsilon$ M

Then we have

$$\begin{array}{ccccccc}
s & \stackrel{a_1}{\Rightarrow} & q_1 & \stackrel{a_2}{\Rightarrow} & \ldots & \stackrel{a_n}{\Rightarrow} & q_n \quad \in F \text{ in } M \\
\cap & & \cap & & & & \\
S' & \stackrel{a_1}{\rightarrow} & S_1 & & & & \\
& & \| & & & & \\
& & \delta(S', a_1) & & & &
\end{array}$$

Consider a string $a_1 a_2 \ldots a_n \in L(M)$, i.e. is accepted by our NFA$^\varepsilon$ M

Then we have

$$\begin{array}{ccccccccl} s & \stackrel{a_1}{\Rightarrow} & q_1 & \stackrel{a_2}{\Rightarrow} & \ldots & \stackrel{a_n}{\Rightarrow} & q_n & & \in F \text{ in } M \\ \Cap & & \Cap & & & & \Cap & & \\ S' & \stackrel{a_1}{\longrightarrow} & S_1 & \stackrel{a_2}{\longrightarrow} & \ldots & \stackrel{a_n}{\longrightarrow} & S_n & & \text{in } PM \end{array}$$

Consider a string $a_1 a_2 \ldots a_n \in L(M)$, i.e. is accepted by our NFA$^\varepsilon$ M

Then we have

$$\begin{array}{ccccccccc} s & \stackrel{a_1}{\Rightarrow} & q_1 & \stackrel{a_2}{\Rightarrow} & \ldots & \stackrel{a_n}{\Rightarrow} & q_n & & \in F \text{ in } M \\ \cap & & \cap & & & & \cap & & \\ S' & \stackrel{a_1}{\rightarrow} & S_1 & \stackrel{a_2}{\rightarrow} & \ldots & \stackrel{a_n}{\rightarrow} & S_n & & \text{in } PM \end{array}$$

Consider a string $a_1 a_2 \ldots a_n \in L(M)$, i.e. is accepted by our NFA$^{\varepsilon}$ M

Then we have

$$
\begin{array}{ccccccccc}
s & \overset{a_1}{\Rightarrow} & q_1 & \overset{a_2}{\Rightarrow} & \ldots & \overset{a_n}{\Rightarrow} & q_n & & \in F \text{ in } M \\
\rotatebox{90}{$\in$} & & \rotatebox{90}{$\in$} & & & & \rotatebox{90}{$\in$} & & \\
S' & \overset{a_1}{\rightarrow} & S_1 & \overset{a_2}{\rightarrow} & \ldots & \overset{a_n}{\rightarrow} & S_n \in F' & & \text{in } PM
\end{array}
$$

Consider a string $a_1 a_2 \ldots a_n \in L(M)$, i.e. is accepted by our NFA$^\varepsilon$ M

Then we have

$$\begin{array}{ccccccccl} s & \stackrel{a_1}{\Rightarrow} & q_1 & \stackrel{a_2}{\Rightarrow} & \ldots & \stackrel{a_n}{\Rightarrow} & q_n & & \in F \text{ in } M \\ \Cap & & \Cap & & & & \Cap & & \\ S' & \stackrel{a_1}{\rightarrow} & S_1 & \stackrel{a_2}{\rightarrow} & \ldots & \stackrel{a_n}{\rightarrow} & S_n \in F' & & \text{in } PM \end{array}$$

so $a_1 a_2 \ldots a_n \in L(PM)$

Consider a string $a_1 a_2 \ldots a_n \in L(M)$, i.e. is accepted by our NFA$^\varepsilon$ M

Then we have

$$
\begin{array}{ccccccccl}
s & \overset{a_1}{\Rightarrow} & q_1 & \overset{a_2}{\Rightarrow} & \ldots & \overset{a_n}{\Rightarrow} & q_n & & \in F \text{ in } M \\
\cap & & \cap & & & & \cap & & \\
S' & \overset{a_1}{\longrightarrow} & S_1 & \overset{a_2}{\longrightarrow} & \ldots & \overset{a_n}{\longrightarrow} & S_n \in F' & & \text{in } PM
\end{array}
$$

so $a_1 a_2 \ldots a_n \in L(PM)$

so $L(M) \subseteq L(PM)$

Consider a string $a_1 a_2 \ldots a_n \in L(PM)$, i.e. is accepted by our DFA PM

Then we have

$$S' \xrightarrow{a_1} S_1 \xrightarrow{a_2} \ldots \; S_{n-1} \xrightarrow{a_n} S_n \in F' \quad \text{in } PM$$

Consider a string $a_1 a_2 \ldots a_n \in L(PM)$, i.e. is accepted by our DFA PM

Then we have

$$S' \xrightarrow{a_1} S_1 \xrightarrow{a_2} \ldots \; S_{n-1} \xrightarrow{a_n} S_n \in F' \quad \text{in } PM$$

$$\begin{array}{c} \uplus \\ q_n \in F \quad \text{in } M \end{array}$$

Consider a string $a_1 a_2 \ldots a_n \in L(PM)$, i.e. is accepted by our DFA PM

Then we have

$$\begin{array}{ccccccccl} S' & \xrightarrow{a_1} & S_1 & \xrightarrow{a_2} & \ldots & S_{n-1} & \xrightarrow{a_n} & S_n \in F' & \text{in } PM \\ \Cup & & \Cup & & & & & \Cup & \\ q_0 & \overset{a_1}{\Rightarrow} & q_1 & \overset{a_2}{\Rightarrow} & \ldots & q_{n-1} & \overset{a_n}{\Rightarrow} & q_n \in F & \text{in } M \end{array}$$

Consider a string $a_1 a_2 \ldots a_n \in L(PM)$, i.e. is accepted by our DFA PM

Then we have

$$
\begin{array}{ccccccccc}
S' & \xrightarrow{a_1} & S_1 & \xrightarrow{a_2} & \ldots & S_{n-1} & \xrightarrow{a_n} & S_n \in F' & \text{in } PM \\
\rotatebox{90}{$\in$} & & \rotatebox{90}{$\in$} & & & & & \rotatebox{90}{$\in$} & \\
q_0 & \overset{a_1}{\Longrightarrow} & q_1 & \overset{a_2}{\Longrightarrow} & \ldots & q_{n-1} & \overset{a_n}{\Longrightarrow} & q_n \in F & \text{in } M \\
\Uparrow \varepsilon & & & & & & & & \\
s & & & & & & & &
\end{array}
$$

Consider a string $a_1 a_2 \ldots a_n \in L(PM)$, i.e. is accepted by our DFA PM

Then we have

$$
\begin{array}{ccccccccl}
S' & \xrightarrow{a_1} & S_1 & \xrightarrow{a_2} & \ldots & S_{n-1} & \xrightarrow{a_n} & S_n \in F' & \text{in } PM \\
\rotatebox{90}{$\in$} & & \rotatebox{90}{$\in$} & & & & & \rotatebox{90}{$\in$} & \\
q_0 & \stackrel{a_1}{\Rightarrow} & q_1 & \stackrel{a_2}{\Rightarrow} & \ldots & q_{n-1} & \stackrel{a_n}{\Rightarrow} & q_n \in F & \text{in } M \\
\Uparrow \varepsilon & & & & & & & & \\
s & & & & & & & &
\end{array}
$$

Consider a string $a_1 a_2 \ldots a_n \in L(PM)$, i.e. is accepted by our DFA PM

Then we have

$$
\begin{array}{ccccccccc}
S' & \xrightarrow{a_1} & S_1 & \xrightarrow{a_2} & \ldots & S_{n-1} & \xrightarrow{a_n} & S_n \in F' & \text{in } PM \\
\rotatebox{90}{$\in$} & & \rotatebox{90}{$\in$} & & & & & \rotatebox{90}{$\in$} & \\
q_0 & \overset{a_1}{\Rightarrow} & q_1 & \overset{a_2}{\Rightarrow} & \ldots & q_{n-1} & \overset{a_n}{\Rightarrow} & q_n \in F & \text{in } M \\
\Uparrow \varepsilon & & & & & & & & \\
s & & & & & & & &
\end{array}
$$

so $a_1 a_2 \ldots a_n \in L(M)$

Consider a string $a_1 a_2 \ldots a_n \in L(PM)$, i.e. is accepted by our DFA PM

Then we have

$$\begin{array}{ccccccccl}
S' & \xrightarrow{a_1} & S_1 & \xrightarrow{a_2} & \ldots & S_{n-1} & \xrightarrow{a_n} & S_n \in F' & \text{in } PM \\
\Cup & & \Cup & & & & & \Cup & \\
q_0 & \overset{a_1}{\Rrightarrow} & q_1 & \overset{a_2}{\Rrightarrow} & \ldots & q_{n-1} & \overset{a_n}{\Rrightarrow} & q_n \in F & \text{in } M \\
\Uparrow \varepsilon & & & & & & & & \\
s & & & & & & & &
\end{array}$$

so $a_1 a_2 \ldots a_n \in L(M)$

so $L(PM) \subseteq L(M)$

So we have shown

$$L(M) \subseteq L(PM) \text{ and } L(PM) \subseteq L(M)$$

so that

$$L(M) = L(PM)$$

where PM is specified by M through subset construction.

Thus for every NFA$^\varepsilon$ there is an equivalent DFA

Theorem. For each NFA$^\varepsilon$ $M = (Q, \Sigma, \Delta, s, F, T)$ there is a DFA $PM = (\mathcal{P}(Q), \Sigma, \delta, s', F')$ accepting exactly the same strings as M, i.e. with $L(PM) = L(M)$.

Definition of PM:

- set of states is the powerset $\mathcal{P}(Q) = \{S \mid S \subseteq Q\}$ of the set Q of states of M
- same input alphabet Σ as for M
- next-state function maps each $(S, a) \in \mathcal{P}(Q) \times \Sigma$ to $\delta(S, a) \triangleq \{q' \in Q \mid \exists q \in S.\ q \stackrel{a}{\Rightarrow} q' \text{ in } M\}$
- start state is $s' \triangleq \{q' \in Q \mid s \stackrel{\varepsilon}{\Rightarrow} q'\}$
- subset of accepting sates is $F' \triangleq \{S \in \mathcal{P}(Q) \mid S \cap F \neq \emptyset\}$

To prove the theorem we show that $L(M) \subseteq L(PM)$ and $L(PM) \subseteq L(M)$.

At this point we should think of

- the set of all language $\{L(r)\}$ defined by a some regular expression r, each language being the set of strings which match the regular expression r

At this point we should think of

- the set of all language $\{L(r)\}$ defined by a some regular expression r, each language being the set of strings which match the regular expression r
- the set of all languages $\{L(M)\}$ accepted by some determinisitic finite automaton M

We are about to show that these are equivalent

- we define one of these sets of languages to be the set of regular languages

We are about to show that these are equivalent

- we define one of these sets of languages to be the set of regular languages
- we prove that other set is also the set of regular languages

We are about to show that these are equivalent

- we define one of these sets of languages to be the set of regular languages
- we prove that other set is also the set of regular languages
- in real life we never remember which way round

We are about to show that these are equivalent

- we define one of these sets of languages to be the set of regular languages
- we prove that other set is also the set of regular languages
- in real life we never remember which way round
- here we will define a language to be regular on the basis of recognition by a DFA

Kleene's Theorem

Definition. A language is **regular** iff it is equal to $L(M)$, the set of strings accepted by some deterministic finite automaton M.

Theorem.

(a) For any regular expression r, the set $L(r)$ of strings matching r is a regular language.

(b) Conversely, every regular language is the form $L(r)$ for some regular expression r.

The first part requires us to demonstrate that for any regular expression r, we can construct a DFA, M with $L(M) = L(r)$

We will do this by demonstrating that for any r we can construct a NFA$^{\varepsilon}$ M' with $L(M') = L(r)$ and rely on the subset construction theorem to give us the DFA M.

We consider each axiom and rule that define regular expressions

Axioms

$U = (\Sigma \cup \Sigma')^*$

axioms: $\dfrac{}{a}$ $\quad$ $\dfrac{}{\epsilon}$ $\quad$ $\dfrac{}{\emptyset}$

(where $a \in \Sigma$ and $r, s \in U$)

with straightforward matching rules

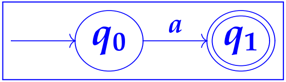
Axioms

$$U = (\Sigma \cup \Sigma')^*$$

axioms: $\dfrac{}{a}$ $\dfrac{}{\epsilon}$ $\dfrac{}{\emptyset}$

(where $a \in \Sigma$ and $r, s \in U$)

with straightforward matching rules



just accepts the one-symbol string a

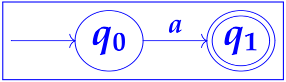
Axioms

$U = (\Sigma \cup \Sigma')^*$

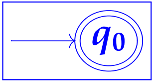
axioms: $\dfrac{}{a}$ $\qquad$ $\dfrac{}{\epsilon}$ $\qquad$ $\dfrac{}{\emptyset}$

(where $a \in \Sigma$ and $r, s \in U$)

with straightforward matching rules



just accepts the one-symbol string a



just accepts the null string, ε

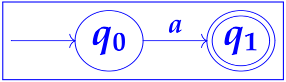
Axioms

$U = (\Sigma \cup \Sigma')^*$

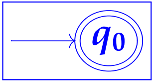
axioms: $\dfrac{}{a}$ $\dfrac{}{\epsilon}$ $\dfrac{}{\emptyset}$

(where $a \in \Sigma$ and $r, s \in U$)

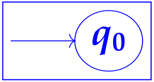
with straightforward matching rules



just accepts the one-symbol string a



just accepts the null string, ε



accepts no strings