Equivalence for DFA and NFA

We can often avoid **exponential growth** of states in Q_D using a technique called **lazy evaluation** (or deferred evaluation)

State q of DFA A is **accessible** if there is at least one string w such that $\hat{\delta}_A(q_0, w) = q$

We build the transition table of D **only** for the accessible states of D

Equivalence for DFA and NFA

Construction of DFA D through lazy evaluation

Base $S = \{q_0\}$ is accessible in D

Induction If state S is accessible in D, then state $\delta_D(S, a)$ is also accessible in D, for every $a \in \Sigma$

Equivalence for DFA and NFA

Theorem Let D be the DFA obtained from an NFA N using the subset construction. Then $L(D) = L(N)$

Proof We first prove that, for every string $w \in \Sigma^*$, we have

$$\hat{\delta}_D(\{q_0\}, w) = \hat{\delta}_N(q_0, w)$$

Check that both sides in the above equation are sets!

We use induction on $|w|$

Base $w = \epsilon$. The claim follows from the definition

Equivalence for DFA and NFA

Induction

$$\begin{array}{rcll}
\hat{\delta}_D(\{q_0\}, xa) & = & \delta_D(\hat{\delta}_D(\{q_0\}, x), a) & \text{definition of } \hat{\delta}_D \\
 & = & \delta_D(\hat{\delta}_N(q_0, x), a) & \text{induction} \\
 & = & \bigcup_{p \in \hat{\delta}_N(q_0, x)} \delta_N(p, a) & \text{definition of } \delta_D \\
 & = & \hat{\delta}_N(q_0, xa) & \text{definition of } \hat{\delta}_N
\end{array}$$

$L(D) = L(N)$ now follows from the definition of F_D $\square$

Equivalence for DFA and NFA

Theorem A language L is accepted by a DFA if and only if L is accepted by an NFA

Proof (If) Previous theorem

(Only if) Any DFA can be converted into an equivalent NFA by modifying δ_D into δ_N according to the following rule

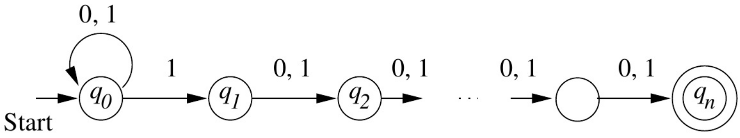
If $\delta_D(q, a) = p$, *then* $\delta_N(q, a) = \{p\}$

By induction on $|w|$ one can show that $\hat{\delta}_D(q_0, w) = p$ if and only if $\hat{\delta}_N(q_0, w) = \{p\}$ $\square$

Exponential growth of the state set

Theorem There exists an NFA N with $n+1$ states that has no equivalent DFA with less than 2^n states

Proof Let N be the NFA



$$L(N) = \{x1c_2c_3 \cdots c_n \mid x \in \{0,1\}^*, c_i \in \{0,1\}\}$$

Intuitively, an equivalent DFA must "remember" the last n symbols it has read

Those symbols might all be relevant for the final decision

Exponential growth of the state set

Suppose there exists a DFA D equivalent to N with fewer than 2^n states

There are 2^n binary strings of length n. Since D has fewer that 2^n states, there must be

- a state q,
- binary strings $a_1 a_2 \cdots a_n \neq b_1 b_2 \cdots b_n$,

such that

pidgeon hole principle

$$\hat{\delta}_D(q_0, a_1 a_2 \cdots a_n) = \hat{\delta}_D(q_0, b_1 b_2 \cdots b_n) = q$$

The above reasoning uses the so-called pigeonhole principle

Exponential growth of the state set

Since $a_1 a_2 \cdots a_n \neq b_1 b_2 \cdots b_n$, there exists i with $1 \leqslant i \leqslant n$ such that $a_i \neq b_i$; we assume $a_i = 1$ and $b_i = 0$ (the other case being symmetrical)

Case 1: $i = 1$; we have

$$\hat{\delta}_D(q_0, 1a_2 \cdots a_n) \in F$$
$$\hat{\delta}_D(q_0, 0b_2 \cdots b_n) \notin F$$

which is a contradiction

Exponential growth of the state set

Case 2: $i > 1$; since $\hat{\delta}_D(q_0, a_1 a_2 \cdots a_n) = \hat{\delta}_D(q_0, b_1 b_2 \cdots b_n)$ and D is deterministic, we have

$$\hat{\delta}_D(q_0, a_1 \cdots a_{i-1} 1 a_{i+1} \cdots a_n 0^{i-1}) =$$
$$\hat{\delta}_D(q_0, b_1 \cdots b_{i-1} 0 b_{i+1} \cdots b_n 0^{i-1})$$

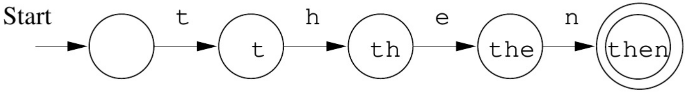
→ add i-1 zeros so that the n-last symbol is 1 for a and 0 for b

From the definition of L, we must have

$$\hat{\delta}_D(q_0, a_1 \cdots a_{i-1} 1 a_{i+1} \cdots a_n 0^{i-1}) \in F$$
$$\hat{\delta}_D(q_0, b_1 \cdots b_{i-1} 0 b_{i+1} \cdots b_n 0^{i-1}) \notin F$$

which is a contradiction □

Partial DFA



This is not a DFA, since for some symbols in Σ transitions are not specified

→ no branching (not NFA)

A **partial** DFA has **at most** one outgoing transition for each state in Q and for each symbol in Σ

A partial DFA can be completed to a DFA if we add one non-accepting state having the status of a **trap state**, from which you cannot escape

Exercise with solution

Consider the NFA

$$N = (\{q_0, q_1\}, \{0, 1\}, \delta_N, q_0, \{q_1\}),$$

where $\delta_N(q_0, 0) = \{q_0, q_1\}$, $\delta_N(q_0, 1) = \{q_1\}$, $\delta_N(q_1, 0) = \emptyset$, $\delta_N(q_1, 1) = \{q_0, q_1\}$

- check whether strings $w_1 = 101$ and $w_2 = 0010$ are in $L(N)$, showing all steps in the computations
- construct the transition diagram of the DFA equivalent to N
- using a set-former, define the language accepted by the automaton; **suggestion**: this is easier if you look at the DFA

Exercise with solution

$w_1 = 101 \in L(A)$?

- $\hat{\delta}(q_0, \epsilon) = \{q_0\}$
- $\hat{\delta}(q_0, 1) = \delta(q_0, 1) = \{q_1\}$
- $\hat{\delta}(q_0, 10) = \delta(q_1, 0) = \emptyset$, then $w_1 = 101 \notin L(A)$

$w_2 = 0010 \in L(A)$?

- $\hat{\delta}(q_0, \epsilon) = \{q_0\}$
- $\hat{\delta}(q_0, 0) = \delta(q_0, 0) = \{q_0, q_1\}$
- $\hat{\delta}(q_0, 00) = \delta(q_0, 0) \cup \delta(q_1, 0) = \{q_0, q_1\} \cup \emptyset = \{q_0, q_1\}$
- $\hat{\delta}(q_0, 001) = \delta(q_0, 1) \cup \delta(q_1, 1) = \{q_1\} \cup \{q_0, q_1\} = \{q_0, q_1\}$
- $\hat{\delta}(q_0, 0010) = \delta(q_0, 0) \cup \delta(q_1, 0) = \{q_0, q_1\} \cup \emptyset = \{q_0, q_1\}$
- since $\{q_0, q_1\} \cap \{q_1\} \neq \emptyset$, $w_2 = 0010 \in L(A)$

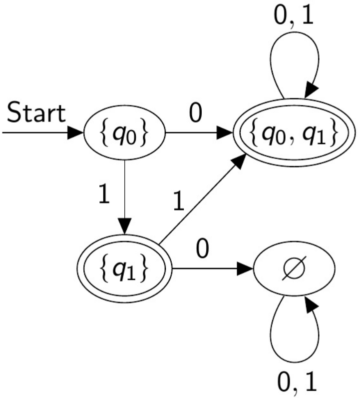
Exercise with solution

We construct the transition diagram of the equivalent DFA using the subset construction and lazy evaluation

- $\delta_D(\{q_0\}, 0) = \delta_N(q_0, 0) = \{q_0, q_1\}$
- $\delta_D(\{q_0\}, 1) = \delta_N(q_0, 1) = \{q_1\}$
- $\delta_D(\{q_0, q_1\}, 0) = \delta_N(q_0, 0) \cup \delta_N(q_1, 0)$
  $= \{q_0, q_1\} \cup \emptyset = \{q_0, q_1\}$
- $\delta_D(\{q_0, q_1\}, 1) = \delta_N(q_0, 1) \cup \delta_N(q_1, 1)$
  $= \{q_1\} \cup \{q_0, q_1\} = \{q_0, q_1\}$
- $\delta_D(\{q_1\}, 0) = \delta_N(q_1, 0) = \emptyset$
- $\delta_D(\{q_1\}, 1) = \delta_N(q_1, 1) = \{q_0, q_1\}$
- $\delta_D(\emptyset, 0) = \delta_D(\emptyset, 1) = \emptyset$
- $\{q_0\}$ initial state, $\{q_0, q_1\}$ e $\{q_1\}$ final states

Exercise with solution

Graphical representation of the transition diagram



Exercise with solution

Using a set-former, define the language accepted by the automaton

$$\begin{aligned} L(A) \quad = \quad & \{w \in \{0,1\}^{+} \mid w = 1 \text{ or } w = 0x \\ & \text{or } w = 11y,\ x, y \in \{0,1\}^{*}\} \end{aligned}$$

Exercises

Specify an NFA A for each of the following languages defined on the alphabet $\{0, 1\}$

- set of strings with two consecutive 0 or two consecutive 1
- set of strings such that at least one of the last three symbols is 1