

# Lecture #6: Equivalence of Deterministic Finite Automata and Nondeterministic Finite Automata

## A Bad Case for the Subset Construction

Near the end of the lecture notes, it was claimed that there exists an infinite sequence of languages

$$L_1, L_2, L_3, \dots \subseteq \Sigma^*$$

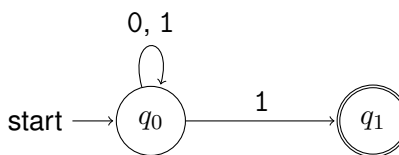
over the alphabet  $\Sigma = \{0, 1\}$ , such that — for every positive integer  $k$  —  $L_k$  is the language of a nondeterministic finite automaton with  $k + 1$  states. but such that every *deterministic* finite automaton with language  $L_k$  must include at least  $2^k$  states.

This document — which is for interest only (and is certainly not required reading) — presents a proof of this claim. It is based on material found in Section 2.3 of the text of Hopcroft, Motwani and Ullman [1].

As above, let  $\Sigma = \{0, 1\}$ , and let

$$L_1 = \{\omega \in \Sigma^* \mid \omega \text{ ends with a } 1\}$$

Then the following nondeterministic finite automaton has language  $L_1$ :

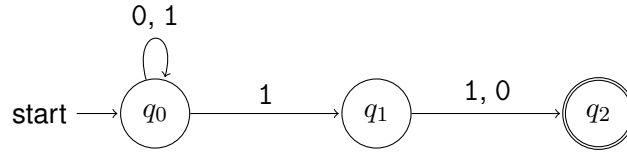


Languages  $L_2, L_3, L_4, \dots \subseteq \Sigma^*$  can be “inductively defined” by setting

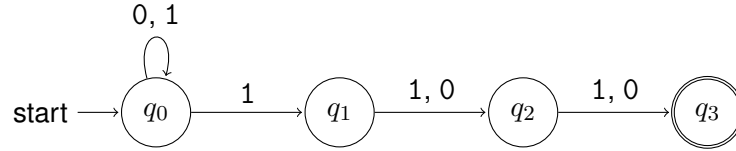
$$L_{k+1} = \{\omega \cdot \sigma \mid \omega \in L_k \text{ and } \sigma \in \Sigma\}.$$

Then  $L_2$  includes all strings in  $\Sigma^*$  whose *second-to-last* symbol is 1,  $L_3$  includes all strings in  $\Sigma^*$  whose *third-to-last* symbol is 1, and so on. and so on.

Now, the following NFA has language  $L_2$ :



Similarly, the following NFA has language  $L_3$ :



For  $k \geq 1$  consider an NFA  $M_k = (Q_k, \Sigma, \delta_k, F_k)$  where

- $Q_k = \{q_0, q_1, q_2, \dots, q_k\}$ , so that  $M_k$  has  $k + 1$  states.
- $\delta_k(q_0, 0) = \{q_0\}$ ,  $\delta_k(q_0, 1) = \{q_0, q_1\}$ , and  $\delta_k(q_0, \lambda) = \emptyset$ ;
- For every integer  $j$  such that  $1 \leq j \leq k - 1$ ,  $\delta_k(q_j, 0) = \delta_k(q_j, 1) = \{q_{j+1}\}$  and  $\delta_k(q_j, \lambda) = \emptyset$ ;
- $\delta_k(q_k, 0) = \delta_k(q_k, 1) = \delta_k(q_k, \lambda) = \emptyset$ .
- $F_k = \{q_k\}$

Note that the nondeterministic finite automata, shown above, are the NFA's  $M_2$  and  $M_3$ , respectively.

It is possible to prove the following (about  $M_k$ ) by induction on  $i$ : For every integer  $i$  such that  $1 \leq i \leq k$ , and for every string  $\omega \in \Sigma^*$ ,

$$q_i \in \delta^*(q_0, \omega) \text{ if and only if } \omega \in L_i.$$

Thus  $L(M_k) = L_k$  — so that  $L_k$  has an NFA with only  $k + 1$  states.

**Claim 1.** Let  $\widehat{M} = (\widehat{Q}, \Sigma, \widehat{\delta}, \widehat{q}_0, \widehat{F})$  be a DFA such that  $L(\widehat{M}) = L_k$ . Then  $|\widehat{Q}| \geq 2^k$ , that is,  $\widehat{M}$  has at least  $2^k$  states.

*Proof.* This will be proved **by contradiction**. Let  $k$  be a positive integer and suppose — to obtain a contradiction — that there exists a deterministic finite automaton

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

with alphabet  $\Sigma$ , whose language is  $M_k$ , such that  $|Q| < 2^k$  (that is,  $M$  has strictly fewer than  $2^k$  states).

$\Sigma^*$  has *exactly*  $2^k$  strings with length  $k$  so it follows by the “Pigeonhole Principle” that there exist strings

$$\omega_1 = \sigma_1 \sigma_2 \dots \sigma_k \text{ and } \omega_2 = \tau_1 \tau_2 \dots \tau_k$$

in  $\Sigma^*$ , both with length  $k$ , such that  $\omega_1 \neq \omega_2$  but  $\hat{\delta}^*(\hat{q}_0, \omega_1) = \hat{\delta}^*(\hat{q}_0, \omega_2)$ .

Since  $\omega_1 \neq \omega_2$  there is an integer  $i$  such that  $1 \leq i \leq k$  and  $\sigma_i \neq \tau_i$ . Without loss of generality we may assume that  $\sigma_i = 1$  and  $\tau_i = 0$  (we can just switch  $\omega_1$  and  $\omega_2$  otherwise). Then  $\omega_1 \in L_{k-i+1}$  and  $\omega_2 \notin L_{k-i+1}$

For  $\ell \geq 0$  let  $1^\ell$  denote a string of  $\ell$  1’s — so that  $1^0 = \lambda$ ,  $1^1 = 1$ ,  $1^2 = 11$ , and so on.

Each of the following things can now be proved by induction on  $\ell$ : For every integer  $\ell \geq 0$ ,

- a)  $\omega_1 \cdot 1^\ell \in L_{k+\ell-i+1}$  and  $\omega_2 \cdot 1^\ell \notin L_{k+\ell-i+1}$  — so that (in particular, with  $\ell = i - 1$ )  $\omega_1 \cdot 1^{i-1} \in L_k$  and  $\omega_2 \cdot 1^{i-1} \notin L_k$ .
- b)  $\hat{\delta}(\hat{q}_0, \omega_1 \cdot 1^\ell) = \hat{\delta}(\hat{q}_0, \omega_2 \cdot 1^\ell)$  — so that (in particular, with  $\ell = i - 1$ )  $\hat{\delta}(\hat{q}_0, \omega_1 \cdot 1^{i-1})$  and  $\hat{\delta}(\hat{q}_0, \omega_2 \cdot 1^{i-1})$  are both equal to the same state  $\hat{q} \in \hat{Q}$ .

Now, since  $\omega_1 \cdot 1^{i-1} \in L_k$ ,  $\hat{\delta}(\hat{q}_0, \omega_1 \cdot 1^{i-1}) = \hat{q}$ , and  $L(\hat{M}) = L_k$ , it must be true that  $\hat{q} \in \hat{F}$ .

Since  $\hat{\delta}(\hat{q}_0, \omega_2 \cdot 1^{i-1}) = \hat{q}$  it now follows that  $\omega_2 \cdot 1^{i-1} \in L(\hat{M}) = L_k$  as well.

We have a **contradiction** — because we already know that  $\omega_2 \cdot 1^{i-1} \notin L_k$ .

So, an assumption that we made, along the way, must be incorrect. We only made one assumption, so *that* one must be false: “The DFA for  $L_k$  being considered has fewer than  $2^k$  states.”

Since this was an arbitrarily chosen DFA whose language is  $L_k$ , it now follows that **every** DFA whose language is  $L_k$  must have at least  $2^k$  states, as claimed.  $\square$

## References

- [1] John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Pearson Education, third edition, 2007.