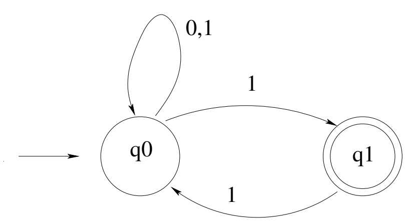
CS 3719 (Theory of Computation and Algorithms) – Lecture 5

Antonina Kolokolova*

January 14, 2011

1 More examples of NFAs

Example 1. Consider the following NFA. It accepts strings ending with a 1 (you can easily think of variations of it, as well as a DFA, also accepting strings ending with 1: for example, removing the 1-arrow from q_1 to q_0 does not change the language).



Consider a string 10111. Possible computations of this automaton on 10111 are:

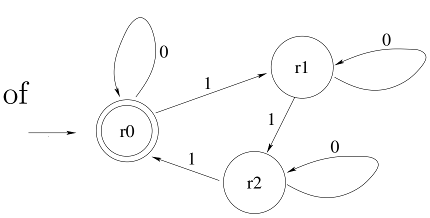
1) q_0, q_1 There is no continuation on 0 from q_1.

2) $q_0, q_0, q_0, q_1, q_0, q_1$. This is an accepting computation.

3) $q_0, q_0, q_0, q_0, q_1, q_0$. This computation does not end in an accept state.

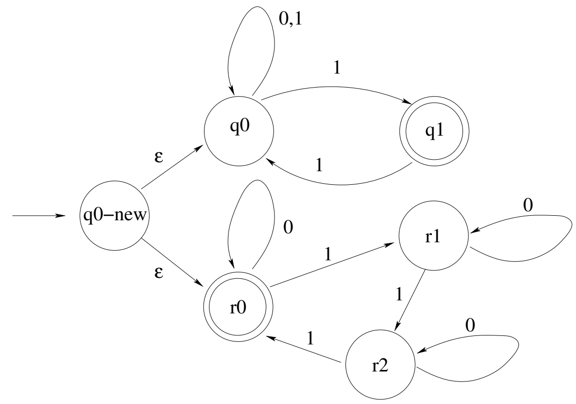
4) $q_0, q_0, q_0, q_0, q_0, q_1$. This is another accepting computation.

Example 2. The following DFA accepts all strings with number of 1s divisible by 3.



*The material in this set of notes came from many sources, in particular "Introduction to Theory of Computation" by Sipser and course notes of U. of Toronto CS 364.

Example 3. Now, the following NFA accepts strings that either end with 1 or have number of 1s divisible by 3.



Lemma 2. *The class of regular languages is closed under the Union operation.*

Proof. Let A_1 and A_2 be two regular languages. Since they are regular, there exist two automata (say, non-deterministic) N_1 and N_2 such that $\mathcal{L}(N_1) = A_1$ and $\mathcal{L}(N_2) = A_2$. We will show that the language $A_1 \cup A_2$ is regular by constructing an automaton N such that $\mathcal{L}(N) = A_1 \cup A_2$. For simplicity, let's assume that both A_1 and A_2 are languages over the same alphabet Σ; otherwise, take the new alphabet to be the union of the alphabets of A_1 and A_2. We will use $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ and $N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ to construct $N = (Q, \Sigma, \delta, q_0, F)$.

You have seen already the intuition for combining automata to obtain a union of their languages: add a new start state and connect it with ϵ-arrows to the start states of the original automata. Thus, we construct N as follows:

1) $Q = Q_1 \cup Q_2 \cup \{q_0\}$. Here, we are assuming that Q_1 and Q_2 are disjoint; if some states in different automata happen to have the same name, just rename them. Also, we add a new state q_0 which will be the start state of N.

2) Σ is the same as the Σ in N_1 and N_2 if they are the same, or the union $\Sigma = \Sigma_1 \cup Sigma_2$ of them if they are different.

3) The new start state is q_0.

4) To accept a string, the automaton should finish either in an accept state of N_1 or an accept state of N_2. Thus, $F = F_1 \cup F_2$.

5) Finally, we describe the transition function as follows:

$$\delta(q,a) = \begin{cases} \delta_1(q,a) & q \in Q_1 \\ \delta_2(q,a) & q \in Q_2 \\ \{q_1, q_2\} & q = q_0 \text{ and } a = \epsilon \\ \emptyset & \text{otherwise} \end{cases}$$

You can check yourself that if a string is accepted by N, then it is accepted by either N_1 or N_2 and, for the other direction, if a string is accepted by N it is accepted by either N_1 or N_2. $\square$

Example 4. Consider languages from examples 1 and 2. Let the NFA $N_1 = (\{q_0, q_1\}, \{0,1\}, q_0, \delta_1, \{q_1\})$ be the NFA from example 1 and $N_2 = (\{r_0, r_1, r_2\}, \{0,1\}, r_0, \delta_2, \{r_0\})$ be the NFA from example 2. Now, $N = \{q_0\text{-new}, q_0, q_1, r_0, r_1, r_2\}, \{0,1\}, q_0\text{-new}, \delta, \{q_1, r_0\}$ where $\delta_1, \delta_2, \delta$ are as follows.

δ_1	0	1	ϵ
q_0	$\{q_0\}$	$\{q_0, q_1\}$	$\emptyset$
q_1	$\emptyset$	$\{q_0\}$	$\emptyset$

δ_2	0	1	ϵ
r_0	$\{r_0\}$	$\{r_1\}$	$\emptyset$
r_1	$\{r_1\}$	$\{r_2\}$	$\emptyset$
r_2	$\{r_2\}$	$\{r_0\}$	$\emptyset$

δ	0	1	ϵ
q_0-new	$\emptyset$	$\emptyset$	$\{q_0, r_0\}$
q_0	$\{q_0\}$	$\{q_0, q_1\}$	$\emptyset$
q_1	$\emptyset$	$\{q_0\}$	$\emptyset$
r_0	$\{r_0\}$	$\{r_1\}$	$\emptyset$
r_1	$\{r_1\}$	$\{r_2\}$	$\emptyset$
r_2	$\{r_2\}$	$\{r_0\}$	$\emptyset$

Notice that you can use this construction to take unions of more than two languages. Just connect the start state of the union NFA by ϵ-arrows with all start states of the automata that you are unioning together.