CS 4510: Automata and Complexity 09/06/2022

Lecture 5: NFA To DFA

Lecturer: Zvi Galil *Author: Austin Peng*

Contents

1 Equivalence Of NFAs and DFAs (cont.)

Theorem 1. Every NFA has an equivalent DFA.

Proof. Let $N = (Q, \Sigma, \delta, q_0, F)$ be the NFA recognizing some language A. We construct DFA $M = (Q', \Sigma, \delta', q'_0, F')$ recognizing A. Before doing the full construction, first consider the easier case when N has no ε arrows. We will take ε into account later.

1. $Q' = \mathcal{P}(Q)$ (the set of subsets of Q)

   Every state of M is a set of states of N.

2. Σ (the alphabet) doesn't change

3. For $R \in Q'$, and $a \in \Sigma$, let $\delta'(R, a) = \{q \in Q \mid q \in \delta(r, a) \text{ for some } r \in R\}$

   If R is a state of M, it is also a set of states of N. When M reads a symbol a in state R, it shows where A takes each state in R. Because each state may go to a set of states, we take the union of all these sets.

   $$\delta'(R, a) = \bigcup_{r \in R} \delta(r, a)$$

4. $q'_0 = \{q_0\}$

   M starts in the state corresponding to the collection containing just the start state of N.

5. $F' = \{R \in Q' \mid R \text{ contains an accepting state of } N\}$

   The machine M accepts if one of the possible states that N could be in at this point is an accepting state.

Now consider the ε arrows. For any state R of M, we define $E(R)$ to be the collection of states that can be reached from members of R by going only along ε arrows, including members of R themselves. Formally, for $R \subseteq Q$, let

$$E(R) = \{q \mid q \text{ can be reached from } R \text{ by traveling along 0 or more } \varepsilon \text{ arrows}\}$$

Then we modify the transition function of M to place additional fingers on all states that can be reached by going along ε arrows after every step. Replacing $\delta(r, a)$ by $E(\delta(r, a))$ achieves this. Finally, we need to modify the start state of M to move the fingers initially to all possible states that can be reached from the start state of N along the ε arrows.

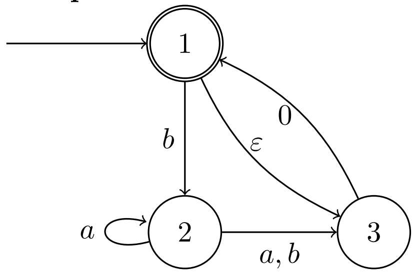
The changes mentioned above to account for ε arrows are shown below:

3. $\delta'(R, a) = \{q \in Q \mid q \in E(\delta(r, a)) \text{ for some } r \in R\}$

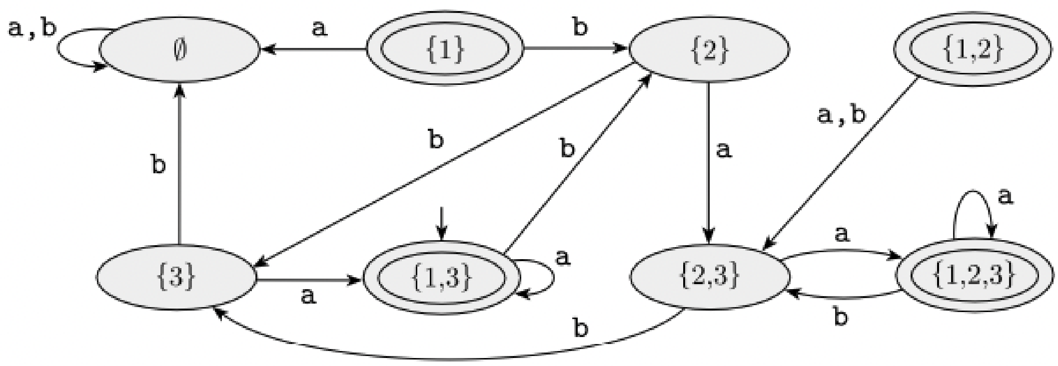
4. $q'_0 = E(\{q_0\})$

$\square$

Example 1. Consider the following NFA N:



Note that the DFA will have 8 states, one for each subset of the states of N. The DFA and its transitions are shown below:



The NFA's start state is 1, so the DFA's start state is $E(\{1\}) = \{1, 3\}$ (the set of states reachable from 1 by travelling along ε arrows and 1 itself). The NFA's accepting state is 1, so the DFA's accepting states are all sets of states that include 1: $\{\{1\}, \{1, 2\}, \{1, 3\}, \{1, 2, 3\}\}$

As for D's transition function, each of D's states goes to one place on input a and one place on input b (by definition of DFA). We will illustrate a few.

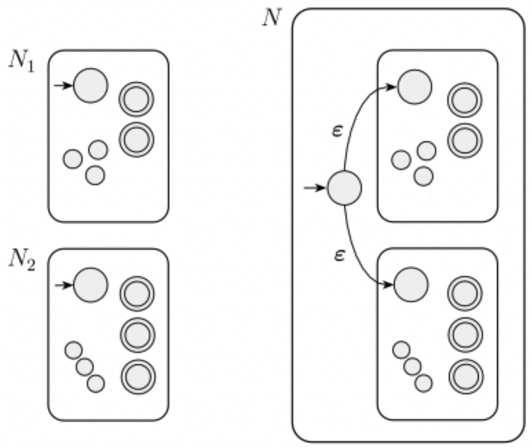
- in D, state $\{2\}$ goes to $\{2, 3\}$ on input a because in N, state 2 goes to both 2 and 3 on input a.
- in D, state $\{1\}$ goes to $\emptyset$ on input a because no a arrows exit it.
- in D, state $\{1, 2\}$ goes to $\{2, 3\}$ on input a because in N, state 1 goes nowhere on input a and state 2 goes to both 2 and 3 on input a

NFA with n states $\rightarrow$ DFA with 2^n states.

2 Closure Under Regular Operations

Theorem 2. The class of regular languages is closed under the union operation.

Proof Idea. We have regular languages A_1 and A_2 and want to prove that $A_1 \cup A_2$ is regular. The idea is to take 2 NFAs, N_1 and N_2, that accept A_1 and A_2 respectively, and combine them into a single new NFA N.



Proof. Let $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ recognize A_1 and $N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ recognize A_2.

Construct $N = (Q, \Sigma, \delta, q_0, F)$ to recognize $A_1 \cup A_2$.

1. $Q = \{q_0\} \cup Q_1 \cup Q_2$

   The states of N are all the states of N_1 and N_2 with the addition of a new start state q_0.

2. The state q_0 is the start state of N.

3. The set of accepting states $F = F_1 \cup F_2$.

   The accepting states of N are all the accepting states of N_1 and N_2. That way, N accepts if either N_1 or N_2 accepts.

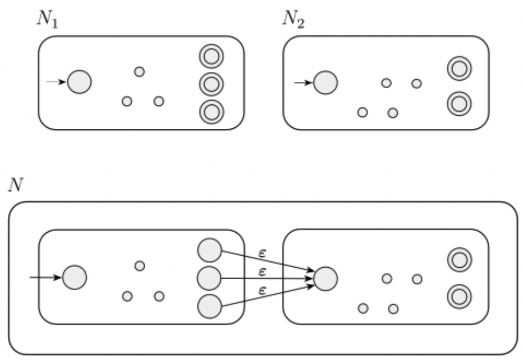
4. Define δ so that for any $q \in Q$ and any $a \in \Sigma_\varepsilon$,

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

□

Theorem 3. The class of regular languages is closed under the concatenation operation.

Proof Idea. We have regular languages A_1 and A_2 and want to prove that $A_1 \circ A_2$ is regular. The idea is to take 2 NFAs, N_1 and N_2, and combine them into a single new NFA N (like how we did in the union closure proof, but with a few changes).



Proof. Let $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ recognize A_1 and $N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ recognize A_2.

Construct $N = (Q, \Sigma, \delta, q_0, F)$ to recognize $A_1 \circ A_2$.

1. $Q = Q_1 \cup Q_2$

   The states of N are all the states of N_1 and N_2.

2. The state q_1 is the start state of N_1.

3. The set of accepting states F_2 are the same as the accepting states of N_2.

4. Define δ so that for any $q \in Q$ and any $a \in \Sigma_\varepsilon$,

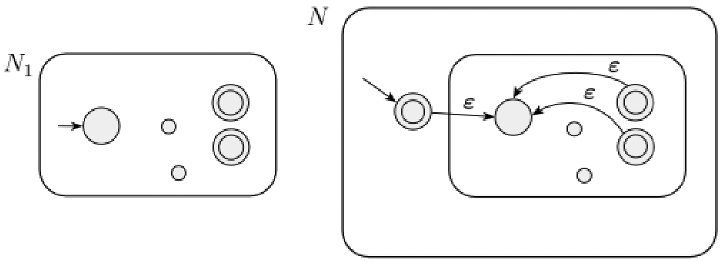
$$\delta(q,a) = \begin{cases} \delta_1(q,a) & q \in Q_1 \text{ and } q \notin F_1 \\ \delta_1(q,a) & q \in F_1 \text{ and } a \neq \varepsilon \\ \delta_1(q,a) \cup \{q_2\} & q \in F_1 \text{ and } a \in \varepsilon \\ \delta_2(q,a) & q \in Q_2 \end{cases}$$

□

Theorem 4. The class of regular languages is closed under the Kleene star operation.

Proof Idea. We have a regular language A and want to prove that A_1^* also is regular. We take an NFA N_1 for A_1 and modify it to recognize A_1^* as shown. The resulting NFA N will accept its input whenever it can be broken into several pieces and N accepts each piece.

We can construct N like N_1 with additional ε arrows returning to the start state from the accepting states. This way when the processing gets to the end of a piece that N_1 accepts, the machine N has the option of jumping back to the start state to try to read another piece that N_1 accepts.



Proof. Let $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ recognize A_1. Construct $N = (Q, \Sigma, \delta, q_0, F)$ to recognize A_1^*.

1. $Q = \{q_0\} \cup Q_1$

   The states of N are the states of N_1 plus a new start state.

2. The state q_0 is the new start state.

3. $F = \{q_0\} \cup F_1$

   The accepting states are the old accepting states plus the new start state.

4. Define δ so that for any $q \in Q$ and any $a \in \Sigma_\varepsilon$,

$$\delta(q,a) = \begin{cases} \delta_1(q,a) & q \in Q_1 \text{ and } q \notin F_1 \\ \delta_1(q,a) & q \in F_1 \text{ and } a \notin \varepsilon \\ \delta_1(q,a) \cup \{q_1\} & q \in F_1 \text{ and } a \in \varepsilon \\ \{q_1\} & q = q_0 \text{ and } a \in \varepsilon \\ \emptyset & q = q_0 \text{ and } a \notin \varepsilon \end{cases}$$

□