# Lecture 7: NFAs are equivalent to DFAs

9 February 2010

## 1 From NFAs to DFAs

### 1.1 NFA handling an input word

For the NFA $N = (Q, \Sigma, \delta, q_0, F)$ that has no $\epsilon$-transitions, let us define $\Delta_N(X, c)$ to be the set of states that $N$ might be in, if it was in a state of $X \subseteq Q$, and it handled the input $c$. Formally, we have that

$$\Delta_N(X, c) = \bigcup_{x \in X} \delta(x, c).$$

We also define $\Delta_N(X, \epsilon) = X$. Given a word $w = w_1, w_2, \ldots, w_n$, we define

$$\Delta_N(X, w) = \Delta_N\Big(\Delta_N(X, w_1 \ldots w_{n-1}), w_n\Big) = \Delta_N(\Delta_N(\ldots \Delta_N(\Delta_N(X, w_1), w_2) \ldots), w_n).$$

That is, $\Delta_N(X, w)$ is the set of all the states $N$ might be in, if it starts from a state of $X$, and it handles the input $w$.

The proof of the following lemma is by an easy induction on the length of $w$.

**Lemma 1.1** *Let $N = (Q, \Sigma, \delta, q_0, F)$ be a given NFA with no $\epsilon$-transitions. For any word $w \in \Sigma^*$, we have that $q \in \Delta_N(\{q_0\}, w)$, if and only if, there is a way for $N$ to be in $q$ after reading $w$ (when starting from the start state $q_0$).*

**More details.** We include the proof for the sake of completeness, but the reader should by now be able to fill in such a proof on their own.

*Proof:* The proof is by induction on the length of $w = w_1 w_2 \ldots w_k$.

If $k = 0$ then $w$ is the empty word, and then $N$ stays in $q_0$. Also, by definition, we have $\Delta_N(\{q_0\}, w) = \{q_0\}$, and the claim holds in this case.

Assume that the claim holds for all word of length at most $n$, and let $k = n + 1$ be the length of $w$. Consider a state $q_{n+1}$ that $N$ reaches after reading $w_1 w_2 \ldots w_n w_{n+1}$, and let $q_n$ be the state $N$ was before handling the character $w_{n+1}$ and reaching $q_{n+1}$. By induction, we know that $q_n \in \Delta_N(\{q_0\}, w_1 w_2 \ldots w_n)$. Furthermore, we know that $q_{n+1} \in \delta(q_n, w_{n+1})$. As such, we have that

$$\begin{aligned} q_{n+1} \in \delta(q_n, w_{n+1}) \subseteq & \bigcup_{q \in \Delta_N(\{q_0\}, w_1 w_2 \ldots w_n)} \delta(q, w_{n+1}) \\ &= \Delta_N(\Delta_N(\{q_0\}, w_1 w_2 \ldots w_n), w_{n+1}) = \Delta_N(\{q_0\}, w_1 w_{@} \ldots w_{n+1}) \\ &= \Delta_N(\{q_0\}, w). \end{aligned}$$

Thus, $q_{n+1} \in \Delta_N(\{q_0\}, w)$.

As for the other direction, if $p_{n+1} \in \Delta_N(\{q_0\}, w)$, then there must exist a state $p_n \in \Delta_N(\{q_0\}, w_1 \ldots w_n)$, such that $p_{n+1} \in \delta(p_n, w_{n+1})$. By induction, this implies that there is execution trace for $N$ starting at $q_0$ and ending at $p_n$, such that $N$ reads $w_1 \ldots w_n$ to reach $p_n$. As such, appending the transition from $p_n$ to $p_{n+1}$ (that read the character $w_{n+1}$ to this trace, results in a trace for $N$ that starts at $q_0$, reads $w$, and end up in the state $p_{n+1}$.

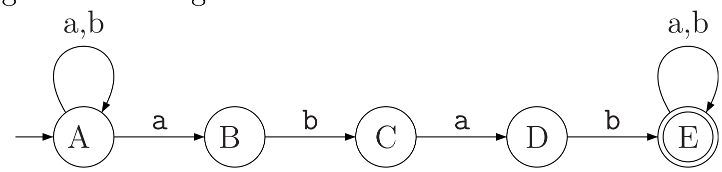
Putting these two arguments together, imply the claim. ■

## 1.2 Simulating NFAs with DFAs

One possible way of thinking about simulating NFAs is to consider each state to be a "light" that can be either on or off. In the beginning, only the initial state is on. At any point in time, all the states that the NFA might be in are turned on. As a new input character arrives, we need to update the states that are on.

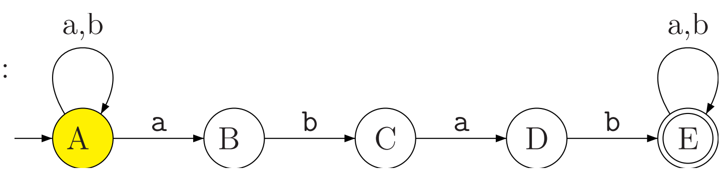
As a concrete examples, consider the automata below (which you had seen before), that accepts strings containing the substring `abab`.

(N1)



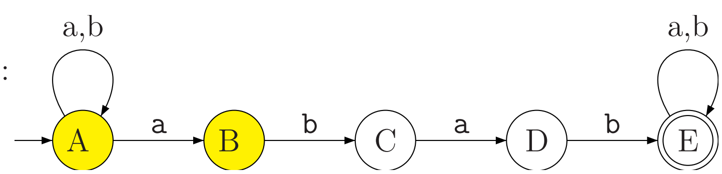

Let us run an explicit search for the above NFA (N1) on the input string `ababa`.

$t = 0$:



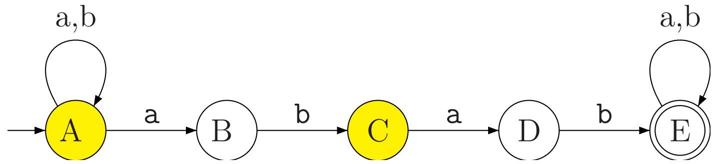

Remaining input: `ababa`.

$t = 1$:



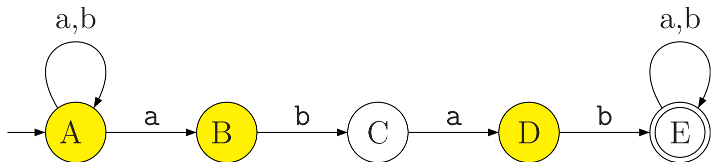

Remaining input: `baba`.

$t = 2$:



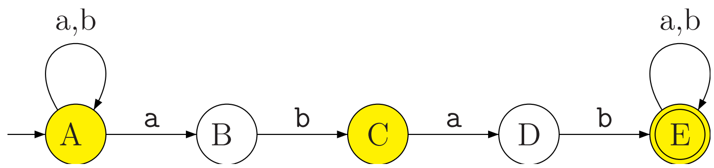

Remaining input: `aba`.

$t = 3$:



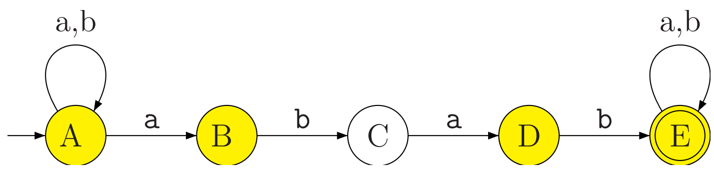

Remaining input: `ba`.

$t = 4$:




Remaining input: `a`.

$t = 5$:




Remaining input: $\epsilon$.

Note, that (N1) accepted `ababa` because when its done reading the input, the accepting state is on.

This provide us with a scheme to simulate this NFA with a DFA: (i) Generate all possible configurations of states that might be turned on, and (ii) decide for each configuration what is the next configuration, what is the next configuration. In our case, in all configurations the first state is turned on. The initial configuration is when only state $A$ is turned on. If this sounds familiar, it should, because what you get is just a big nasty, hairy DFA, as shown on the last page of this class notes. The same DFA with the unreachable states removed is shown in Figure 1.

Every state in the DFA of Figure 1 can be identified by the subset of the original states that is turned on (namely, the original automata might be any of these states).

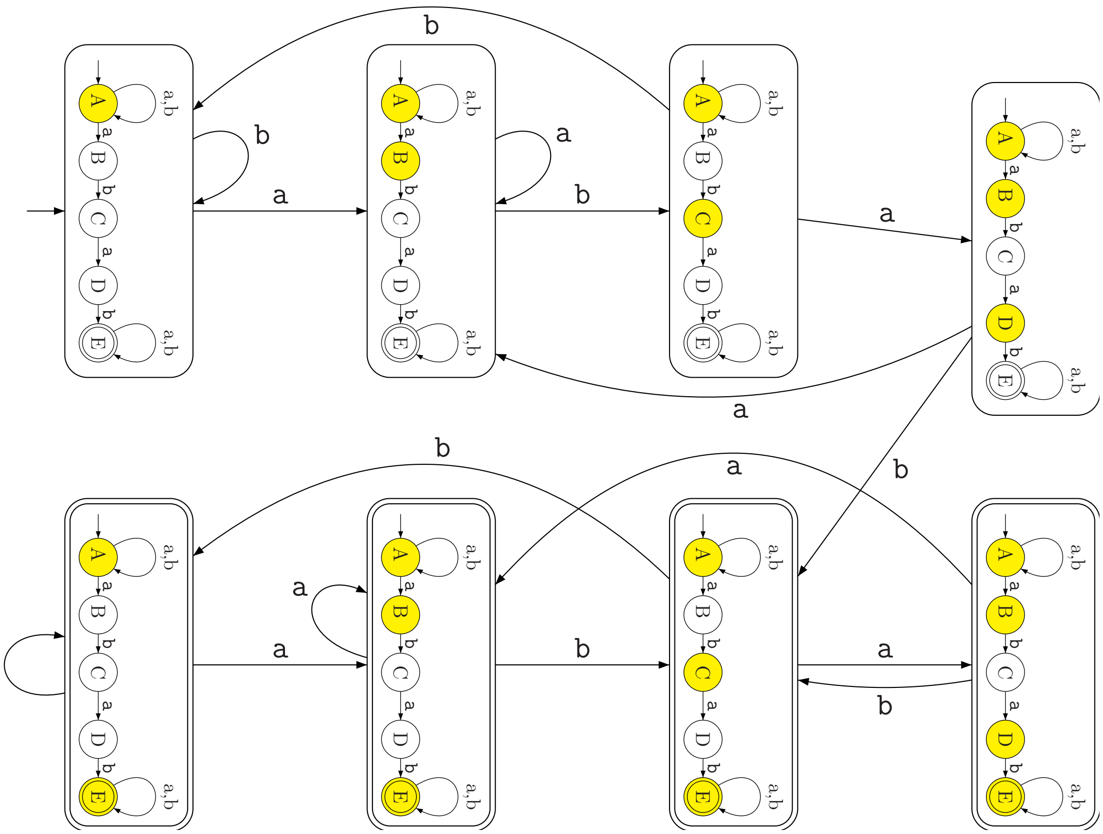
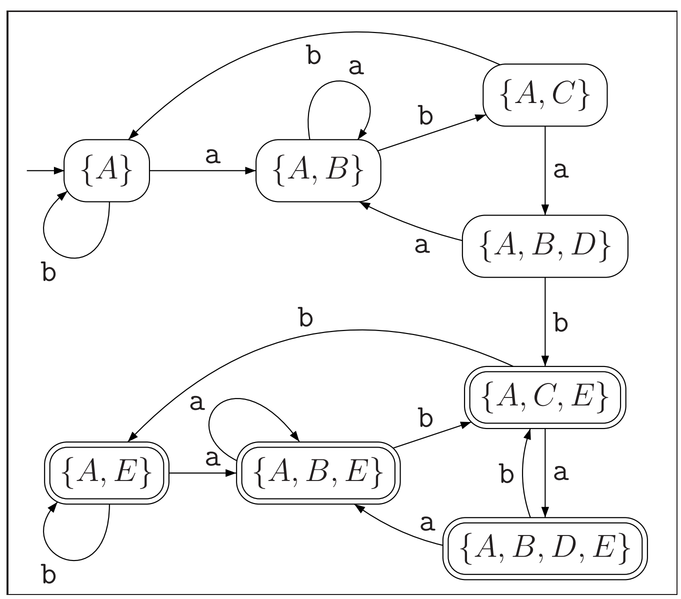



Figure 1: The resulting DFA

Thus, a more conventional drawing of this automata is shown on the right.




Thus, to convert an NFA $N$ with a set of states $Q$ into a DFA, we consider all the subsets of $Q$ that $N$ might be realized as. Namely, every subset of $Q$ (i.e., a member of $\mathbb{P}(Q)$ – the power set of $Q$) is going to be a state in the new automata. Now, consider a subset $X \subseteq Q$, and for every input character $c \in \Sigma$, let us figure out in what states the original NFA $N$ might be in if it is in one of the states of $X$, and it handles the characters $c$. Let $Y$ be the resulting set of such states.

Clearly, we had just computed the transition function of the new (equivalent) DFA, showing that if the NFA is in one of the states of $X$, and we receive $c$, then the NFA now might be in one of the states of $Y$.

Now, if the initial state of the NFA $N$ is $q_0$, then the new DFA $M_{\mathsf{DFA}}$ would start with the state (i.e., configuration) $\{q_0\}$ (since the original NFA might be only in $q_0$ at this point in time).

Its important that our simulation is *faithful*: At any point in time, if we are in state $X$ in $M_{\mathsf{DFA}}$ then there is a path in the original NFA $N$, with the given input, to reach each state of $Q$ that is in $X$ (and similarly, $X$ includes all the states that are reachable with such an input).

When does $M_{\mathsf{DFA}}$ accepts? Well, if it is in state $X$ (here $X \subseteq Q$), then it accepts only if $X$ includes one of the accepting states of the original NFA $N$.

Clearly, the resulting DFA $M_{\mathsf{DFA}}$ is equivalent to the original NFA.

## 1.3 The construction of a DFA from an NFA

Let $N = (Q, \Sigma, \delta, q_0, F)$ be the given NFA that does not have any $\epsilon$-transitions. The new DFA is going to be

$$M_{\mathsf{DFA}} = \left(\mathbb{P}(Q), \Sigma, \widehat{\delta}, \widehat{q_0}, \widehat{F}\right),$$

where $\mathbb{P}(Q)$ is the power set of $Q$, and $\widehat{\delta}$ (the transition function), $\widehat{q_0}$ the initial state, and the set of accepting states $\widehat{F}$ are to be specified shortly. Note that the states of $M_{\mathsf{DFA}}$ are subsets of $Q$ (which is slightly confusing), and as such the starting state of $M_{\mathsf{DFA}}$, is $\widehat{q_0} = \{q_0\}$ (and not just $q_0$).

We need to specify the transition function, so consider $X \in \mathbb{P}(Q)$ (i.e., $X \subseteq Q$), and a character $c$. For a state $s \in X$, the NFA might go into any state in $\delta(s, c)$ after reading $q$. As such, the set of all possible states the NFA might be in, if it started from a state in $X$, and received $c$, is the set

$$Y = \bigcup_{s \in X} \delta(s, c).$$

As such, the transition of $M_{\mathsf{DFA}}$ from $X$ receiving $c$ is the state of $M_{\mathsf{DFA}}$ defined by $Y$. Formally,

$$\widehat{\delta}(X, c) = Y = \bigcup_{s \in X} \delta(s, c). \tag{1}$$

As for the accepting states, consider a state $X \in \mathbb{P}(Q)$ of $M_{\mathsf{DFA}}$. Clearly, if there is a state of $F$ in $X$, then $X$ is an accepting state; namely, $F \cap X \neq \emptyset$. Thus,

$$\widehat{F} = \left\{ X \;\middle|\; X \in \mathbb{P}(Q), X \cap F \neq \emptyset \right\}.$$

### 1.3.1 Proof of correctness

**Claim 1.2** *For any $w \in \Sigma^*$, the set of states reached by the NFA $N$ on $w$ is precisely the state reached by $M_{\mathsf{DFA}}$ on $w$. That is $\Delta_N(\{q_0\}, w) = \widehat{\delta}(\{q_0\}, w)$.*

*Proof:* The proof is by induction on the length of $w$.

If $w$ is the empty word, then $N$ is at $q_0$ after reading $\epsilon$ (i.e., $\Delta_N(\{q_0\}, \epsilon) = \{q_0\}$), and the $M_{\mathsf{DFA}}$ is still in its initial state which is $\{q_0\}$.

So assume that the claim holds for all words of length at most $k$.

Let $w = w_1 w_2 \ldots w_{k+1}$. Let $X$ be the set of states that $N$ might reach from $q_0$ after reading $w' = w_1 \ldots w_n$; that is $X = \Delta_N(\{q_0\}, w')$. By the induction hypothesis, we have that $M_{\mathsf{DFA}}$ is in the state $X$ after reading $w'$ (formally, we have that $\widehat{\delta}(\{q_0\}, w') = X$).

Now, the NFA $N$, when reading the last character $w_{k+1}$, can start from any state of $X$, and use any transition from such a state that reads the character $w_{k+1}$. Formally, the NFA $N$ is in one of the states of

$$Z = \Delta_N(X, w_{k+1}) = \bigcup_{s \in X} \delta(s, w_{k+1}).$$

Similarly, by the definition of $M_{\mathsf{DFA}}$, we have that from the state $X$, after reading $w_{k+1}$, the DFA $M_{\mathsf{DFA}}$ is in the state

$$Y = \widehat{\delta}(X, w_{k+1}) = \bigcup_{s \in X} \delta(s, w_{k+1}),$$

see Eq. (1). But clearly, $Z = Y$, which establishes the claim. ■

**Lemma 1.3** *Any NFA $N$, without $\epsilon$-transitions, can be converted into a DFA $M_{\mathsf{DFA}}$, such that $M_{\mathsf{DFA}}$ accepts the same language as $N$.*

*Proof:* The construction is described above.

So consider a word $w \in \Sigma^*$, and observe that $w \in L(N)$ if and only if, the set of states $N$ might be in after reading $w$ (that is $\Delta_N(\{q_0\}, w)$), contains an accepting state of $N$. Formally, $w \in L(N)$ if and only if

$$\Delta_N\Big(\{q_0\}, w\Big) \cap F \neq \emptyset.$$

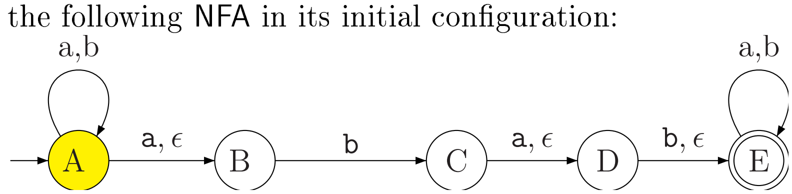
The DFA $M_{\mathsf{DFA}}$ is in the state $\widehat{\delta}(\{q_0\}, w)$ after reading $w$. Claim 1.2, implies that $Y = \widehat{\delta}(\{q_0\}, w) = \Delta_N(\{q_0\}, w)$. By construction, the $M_{\mathsf{DFA}}$ accepts at this state, if and only if, $Y \in \widehat{F}$, which equivalent to that $Y$ contains a final state of $N$. That is $Y \cap F \neq \emptyset$. Namely, $M_{\mathsf{DFA}}$ accepts $w$ if

$$\widehat{\delta}\Big(\{q_0\}, w\Big) \cap F \neq \emptyset \iff \Delta_N\Big(\{q_0\}, w\Big) \cap F \neq \emptyset.$$

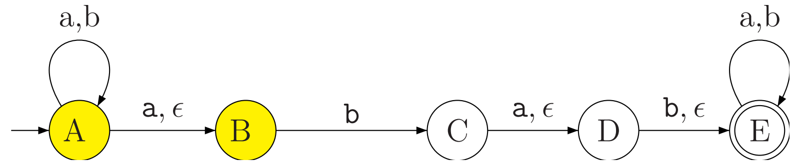
Implying that $M_{\mathsf{DFA}}$ accepts $w$ if and only if $N$ accepts $w$. ■

### 1.3.2 Handling $\epsilon$-transitions

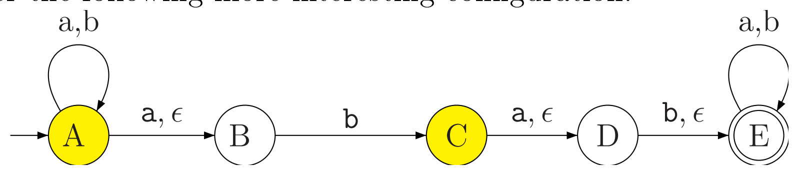
Now, we would like to handle a general NFA that might have $\epsilon$-transitions. The problem is demonstrated in the following NFA in its initial configuration:



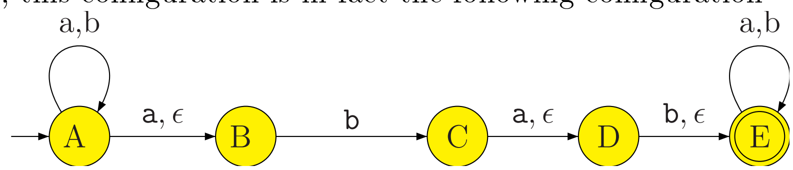

Clearly, the initial configuration here is $\{A, B\}$ (and not the one drawn above), since the automata can immediately jump to $B$ if the NFA is already in $A$. So, the configuration $\{A\}$ should not be considered at all. As such, the true initial configuration for this automata is

(N2) 


Next, consider the following more interesting configuration.




But here, not only we can jump from $A$ to $B$, but we can also jump from $C$ to $D$, and from $D$ to $E$. As such, this configuration is in fact the following configuration

(N3) 


In fact, this automata can only be in these two configurations because of the $\epsilon$-transitions.

So, let us formalize the above idea: Whenever the NFA $N$ might be in a state $s$, we need to extend the configuration to all the states of the NFA reachable by $\epsilon$-transitions from $s$. Let $R_\epsilon(s)$ denote the set of all states of $N$ that are reachable by a sequence of $\epsilon$-transitions from $s$ ($s$ is also in $R_\epsilon(s)$ naturally, since we can reach $s$ without moving anywhere).

Thus, if $N$ might be any state of $X \subseteq Q$, then it might be in any state of

$$\mathcal{E}(X) = \bigcup_{s \in X} R_\epsilon(s)\,.$$

As such, whenever we consider the set of states $X$ for $Q$, in fact, we need to consider the *extended set of states* $\mathcal{E}(X)$. As such, for the above automata, we have

$$\mathcal{E}(\{A\}) = \{A, B\} \qquad \text{and} \qquad \mathcal{E}(\{A, C\}) = \{A, B, C, D, E\}\,.$$

Now, we can essentially repeat the above proof.

**Theorem 1.4** *Any NFA $N$ (with or without $\epsilon$-transitions) can be converted into a DFA $M_{DFA}$, such that $M_{DFA}$ accepts the same language as $N$.*

*Proof:* Let $N = (Q, \Sigma, \delta, q_0, F)$. The new DFA is going to be

$$M_{\mathsf{DFA}} = \left(\mathbb{P}(Q)\,, \Sigma, \delta_M, q_S, \widehat{F}\right).$$

Here, $\mathbb{P}(Q)$, $\Sigma$ and $\widehat{F}$ are the same as above.

Now, for $X \in \mathbb{P}(Q)$ and $c \in \Sigma$, let

$$\delta_M(X) = \mathcal{E}\left(\widehat{\delta}(X, c)\right),$$

where $\widehat{\delta}$ is the old transition function from the proof of Lemma 1.3; namely, we always extend the new set of states to include all the states we can reach by $\epsilon$-transitions. Similarly, the initial state is now

$$q_S = \mathcal{E}(\{q_0\}) .$$

It is now straightforward to verify that the new DFA is indeed equivalent to the original NFA, using the argumentation of Lemma 1.3. ∎