Converting NFAs to DFAs

Alyssa Lytle

Fall 2025

1 Defining Δ and $\hat{\Delta}$

We're going to start today's lesson by defining Δ and $\hat{\Delta}$ in a more formal manner.

Recall 1

For an automaton $M = (Q, \Sigma, \delta, s, F)$,

$$\delta : Q \times \Sigma \rightarrow Q$$

and $\hat{\delta}$ is defined such that:

$$\hat{\delta} : Q \times \Sigma^* \rightarrow Q$$

and for $a \in \Sigma, x \in \Sigma^*$

$$\begin{aligned}
\hat{\delta}(q, \epsilon) &\stackrel{\text{def}}{=} q \text{ (base case)} \\
\hat{\delta}(q, a) &\stackrel{\text{def}}{=} \delta(q, a) \text{ (optional second base case)} \\
\hat{\delta}(q, xa) &\stackrel{\text{def}}{=} \delta(\hat{\delta}(q, x), a)
\end{aligned}$$

Formally, x is *accepted* by M if $\hat{\delta}(s, x) \in F$. [Koz07]

Definition 1

For an nondeterministic automaton $N = (Q, \Sigma, \Delta, S, F)$,

$$\Delta : Q \times \Sigma \rightarrow 2^Q$$

and $\hat{\Delta}$ is defined such that,

$$\hat{\Delta} : Q \times \Sigma^* \rightarrow 2^Q$$

and for $A \subseteq Q, a \in \Sigma, x \in \Sigma^*$

$$\begin{aligned}
\hat{\Delta}(A, \epsilon) &\stackrel{\text{def}}{=} A \text{ (base case)} \\
\hat{\Delta}(A, a) &\stackrel{\text{def}}{=} \bigcup_{q \in A} \Delta(q, a) \text{ (optional second base case)} \\
\hat{\Delta}(A, xa) &\stackrel{\text{def}}{=} \bigcup_{q \in \hat{\Delta}(A, x)} \Delta(q, a)
\end{aligned}$$

A string x is *accepted* by N if $\hat{\Delta}(S, x) \cap F \neq \emptyset$

2 Proof by Construction

We want to prove that any NFA can be converted to an equivalent DFA.

For this proof by construction, we must first formalize the transition from NFA to DFA, then prove that they are equivalent.

2.1 Formalizing the Transition from NFA to DFA

Now, let's get to formalizing the transition of an NFA to a DFA!

(We did an example last class, but now let's generalize it.)

Let $N = (Q_N, \Sigma, \Delta_N, S_N, F_N)$ be an NFA.

We will construct a DFA $M = (Q_M, \Sigma, \delta_M, s_M, F_M)$.

With:

- $Q_M \stackrel{\text{def}}{=} 2^{Q_N}$
- $\delta_M(A, a) \stackrel{\text{def}}{=} \hat{\Delta}_N(A, a)$
- $s_M \stackrel{\text{def}}{=} S_N$
- $F_M \stackrel{\text{def}}{=} \{A \subseteq Q_N | A \cap F_N \neq \emptyset\}$

2.2 Adding some lemmas to the toolbox

To prove equivalence, we need some lemmas.

We are going use a lemma a prove a lemma:

Lemma 1

For any $x, y \in \Sigma^*$ and $A \subseteq Q$, $\hat{\Delta}(A, xy) = \hat{\Delta}(\hat{\Delta}(A, x), y)$.

Lemma 2

For any $A \subseteq Q_N$ and $x \in \Sigma^*$,

$$\hat{\delta}_M(A, x) = \hat{\Delta}_N(A, x)$$

2.3 Proof of Lemma 2:

This will be a proof by induction on $|x|$ (the length of x).

Base case: Let $x = \epsilon$

$\hat{\delta}_M(A, \epsilon) = A$ by the definition of $\hat{\delta}$. And $\hat{\Delta}_N(A, \epsilon) = A$ by the definition of $\hat{\Delta}$.

So,

$$\begin{aligned} \hat{\delta}_M(A, \epsilon) &= \hat{\Delta}_N(A, \epsilon) \\ A &= A \ \square \end{aligned}$$

Inductive Hypothesis: For an x of arbitrary length, $\hat{\delta}_M(A, x) = \hat{\Delta}_N(A, x)$.

Want to prove: For $a \in \Sigma$, $\hat{\delta}_M(A, xa) = \hat{\Delta}_N(A, xa)$.

$\hat{\delta}_M(A, x) = \hat{\Delta}_N(A, x)$	Inductive Hypothesis
$\delta_M(\hat{\delta}_M(A, x), a) = \delta_M(\hat{\Delta}_N(A, x), a)$	Applied $\delta_M(\cdot, a)$ to both sides.
$\delta_M(\hat{\delta}_M(A, x), a) = \hat{\Delta}_N(\hat{\Delta}_N(A, x), a)$	Applied construction definition of $\hat{\delta}_M$ to right side.
$\delta_M(\hat{\delta}_M(A, x), a) = \hat{\Delta}_N(A, xa)$	Applied Lemma 1 to right side.
$\hat{\delta}_M(A, xa) = \hat{\Delta}_N(A, xa)$	Applied $\hat{\delta}$ definition to left side.

Table 1: Inductive Step of Proof

2.4 Proof of Equivalence (Theorem 1):

Now we can prove the N amd M are equivalent! In other words, we want to prove

Theorem 1

The automata M and N accept the same set.
For all $x \in \Sigma$, $x \in L(M) \iff x \in L(N)$

This is a direct proof.

1. $x \in L(M)$	Given.
2. $\hat{\delta}_M(s_M, x) \in F_M$	Definition of acceptance for a DFA.
3.$\hat{\Delta}_N(S_N, x) \bigcap F_N \neq \emptyset$	Application of Lemma 2 and construction definition of $s_M = S_N$ and $F_M = F_N$
4. $x \in L(N)$	Definition of acceptance for an NFA.

Table 2: Proof of Theorem 1

Note 1: This proof technically needs to be done in both "directions". In other words, we proved $x \in L(M) \implies x \in L(N)$, but not $x \in L(N) \implies x \in L(M)$. However, since we simply applied equivalences/definitions for each step of the proof, you can see that just reversing the lines 1-4 would prove this. Therefore, just proving it in one "direction" is sufficient.

Note 2: On line 3 of the proof, $\hat{\delta}$ evaluates to a state, where $\hat{\Delta}$ evaluates to a *set* of states. That's why we switch from saying "it's an element of the set of accept states" ($\in F_M$) to saying "it shares at least one state with the set of accept states" ($\bigcap F_N \neq \emptyset$).

References

[Koz07] Dexter C Kozen. *Automata and computability*. Springer Science & Business Media, 2007.