Lecture 6

The Subset Construction

Formal Definition of Nondeterministic Finite Automata

A *nondeterministic finite automaton (NFA)* is a five-tuple

$$N = (Q,\ \Sigma,\ \Delta,\ S,\ F),$$

where everything is the same as in a deterministic automaton, except for the following two differences.

- S is a *set* of states, that is, $S \subseteq Q$, instead of a single state. The elements of S are called *start states*.
- Δ is a function

  $$\Delta : Q \times \Sigma \to 2^Q,$$

  where 2^Q denotes the *power set* of Q or the set of all subsets of Q:

  $$2^Q \stackrel{\text{def}}{=} \{A \mid A \subseteq Q\}.$$

Intuitively, $\Delta(p, a)$ gives the set of all states that N is allowed to move to from p in one step under input symbol a. We often write

$$p \xrightarrow{a} q$$

if $q \in \Delta(p, a)$. The set $\Delta(p, a)$ can be the empty set $\varnothing$. The function Δ is called the *transition function.*

Now we define acceptance for NFAs. The function Δ extends in a natural way by induction to a function

$$\widehat{\Delta} : 2^Q \times \Sigma^* \to 2^Q$$

according to the rules

$$\widehat{\Delta}(A, \epsilon) \stackrel{\text{def}}{=} A, \tag{6.1}$$

$$\widehat{\Delta}(A, xa) \stackrel{\text{def}}{=} \bigcup_{q \in \widehat{\Delta}(A,x)} \Delta(q, a). \tag{6.2}$$

Intuitively, for $A \subseteq Q$ and $x \in \Sigma^*$, $\widehat{\Delta}(A, x)$ is the set of all states reachable under input string x from *some* state in A. Note that Δ takes a single state as its first argument and a single symbol as its second argument, whereas $\widehat{\Delta}$ takes a *set* of states as its first argument and a *string* of symbols as its second argument.

Equation (6.1) says that the set of all states reachable from a state in A under the null input is just A. In (6.2), the notation on the right-hand side means the union of all the sets $\Delta(q, a)$ for $q \in \widehat{\Delta}(A, x)$; in other words, $r \in \widehat{\Delta}(A, xa)$ if there exists $q \in \widehat{\Delta}(A, x)$ such that $r \in \Delta(q, a)$.

$$p \xrightarrow{\quad\quad\quad x \quad\quad\quad} q \xrightarrow{a} r$$

Thus $q \in \widehat{\Delta}(A, x)$ if N can move from some state $p \in A$ to state q under input x. This is the nondeterministic analog of the construction of $\widehat{\delta}$ for deterministic automata we have already seen.

Note that for $a \in \Sigma$,

$$\begin{aligned}\widehat{\Delta}(A, a) &= \bigcup_{p \in \widehat{\Delta}(A,\epsilon)} \Delta(p, a) \\ &= \bigcup_{p \in A} \Delta(p, a).\end{aligned}$$

The automaton N is said to *accept* $x \in \Sigma^*$ if

$$\widehat{\Delta}(S, x) \cap F \neq \varnothing.$$

In other words, N accepts x if there exists an accept state q (i.e., $q \in F$) such that q is reachable from a start state under input string x (i.e., $q \in \widehat{\Delta}(S, x)$).

We define $L(N)$ to be the set of all strings accepted by N:

$$L(N) = \{x \in \Sigma^* \mid N \text{ accepts } x\}.$$

Under this definition, every DFA

$$(Q,\ \Sigma,\ \delta,\ s,\ F)$$

is equivalent to an NFA

$$(Q,\ \Sigma,\ \Delta,\ \{s\},\ F),$$

where $\Delta(p,a) \stackrel{\text{def}}{=} \{\delta(p,a)\}$. Below we will show that the converse holds as well: every NFA is equivalent to some DFA.

Here are some basic lemmas that we will find useful when dealing with NFAs. The first corresponds to Exercise 3 of Homework 1 for deterministic automata.

Lemma 6.1 *For any $x, y \in \Sigma^*$ and $A \subseteq Q$,*

$$\widehat{\Delta}(A, xy) = \widehat{\Delta}(\widehat{\Delta}(A, x), y).$$

Proof. The proof is by induction on $|y|$.

Basis

For $y = \epsilon$,

$$\begin{aligned}\widehat{\Delta}(A, x\epsilon) &= \widehat{\Delta}(A, x)\\ &= \widehat{\Delta}(\widehat{\Delta}(A, x), \epsilon) \quad \text{by (6.1).}\end{aligned}$$

Induction step

For any $y \in \Sigma^*$ and $a \in \Sigma$,

$$\begin{aligned}\widehat{\Delta}(A, xya) &= \bigcup_{q \in \widehat{\Delta}(A,xy)} \Delta(q, a) && \text{by (6.2)}\\ &= \bigcup_{q \in \widehat{\Delta}(\widehat{\Delta}(A,x),y)} \Delta(q, a) && \text{induction hypothesis}\\ &= \widehat{\Delta}(\widehat{\Delta}(A, x), ya) && \text{by (6.2).}\end{aligned}$$

□

Lemma 6.2 *The function $\widehat{\Delta}$ commutes with set union: for any indexed family A_i of subsets of Q and $x \in \Sigma^*$,*

$$\widehat{\Delta}(\bigcup_i A_i, x) = \bigcup_i \widehat{\Delta}(A_i, x).$$

Proof. By induction on $|x|$.

Basis

By (6.1),

$$\widehat{\Delta}(\bigcup_i A_i, \epsilon) = \bigcup_i A_i = \bigcup_i \widehat{\Delta}(A_i, \epsilon).$$

Induction step

$$\begin{aligned}
\widehat{\Delta}(\bigcup_i A_i, xa) &= \bigcup_{p \in \widehat{\Delta}(\bigcup_i A_i, x)} \Delta(p, a) && \text{by (6.2)} \\
&= \bigcup_{p \in \bigcup_i \widehat{\Delta}(A_i, x)} \Delta(p, a) && \text{induction hypothesis} \\
&= \bigcup_i \bigcup_{p \in \widehat{\Delta}(A_i, x)} \Delta(p, a) && \text{basic set theory} \\
&= \bigcup_i \widehat{\Delta}(A_i, xa) && \text{by (6.2).}
\end{aligned}$$

□

In particular, expressing a set as the union of its singleton subsets,

$$\widehat{\Delta}(A, x) = \bigcup_{p \in A} \widehat{\Delta}(\{p\}, x). \tag{6.3}$$

The Subset Construction: General Account

The subset construction works in general. Let

$$N = (Q_N,\ \Sigma,\ \Delta_N,\ S_N,\ F_N)$$

be an arbitrary NFA. We will use the subset construction to produce an equivalent DFA. Let M be the DFA

$$M = (Q_M,\ \Sigma,\ \delta_M,\ s_M,\ F_M),$$

where

$$\begin{aligned}
Q_M &\stackrel{\text{def}}{=} 2^{Q_N}, \\
\delta_M(A, a) &\stackrel{\text{def}}{=} \widehat{\Delta}_N(A, a), \\
s_M &\stackrel{\text{def}}{=} S_N, \\
F_M &\stackrel{\text{def}}{=} \{A \subseteq Q_N \mid A \cap F_N \neq \varnothing\}.
\end{aligned}$$

Note that δ_M is a function from states of M and input symbols to states of M, as it should be, because states of M are *sets* of states of N.

Lemma 6.3 *For any $A \subseteq Q_N$ and $x \in \Sigma^*$,*

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

Proof. Induction on $|x|$.

Basis

For $x = \epsilon$, we want to show

$$\widehat{\delta}_M(A, \epsilon) = \widehat{\Delta}_N(A, \epsilon).$$

But both of these are A, by definition of $\widehat{\delta}_M$ and $\widehat{\Delta}_N$.

Induction step

Assume that

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

We want to show the same is true for xa, $a \in \Sigma$.

$$\begin{aligned}
\widehat{\delta}_M(A, xa) &= \delta_M(\widehat{\delta}_M(A, x), a) && \text{definition of } \widehat{\delta}_M \\
&= \delta_M(\widehat{\Delta}_N(A, x), a) && \text{induction hypothesis} \\
&= \widehat{\Delta}_N(\widehat{\Delta}_N(A, x), a) && \text{definition of } \delta_M \\
&= \widehat{\Delta}_N(A, xa) && \text{Lemma 6.1.}
\end{aligned}$$

□

Theorem 6.4 *The automata M and N accept the same set.*

Proof. For any $x \in \Sigma^*$,

$$\begin{aligned}
& x \in L(M) \\
\iff & \widehat{\delta}_M(s_M, x) \in F_M && \text{definition of acceptance for } M \\
\iff & \widehat{\Delta}_N(S_N, x) \cap F_N \neq \varnothing && \text{definition of } s_M \text{ and } F_M\text{, Lemma 6.3} \\
\iff & x \in L(N) && \text{definition of acceptance for } N.
\end{aligned}$$

□

ϵ-Transitions

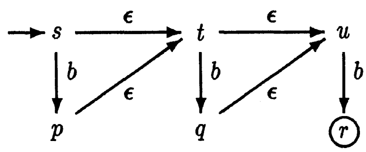
Here is another extension of finite automata that turns out to be quite useful but really adds no more power.

An *ϵ-transition* is a transition with label ϵ, a letter that stands for the null string ϵ:

$$p \xrightarrow{\epsilon} q.$$

The automaton can take such a transition anytime without reading an input symbol.

Example 6.5

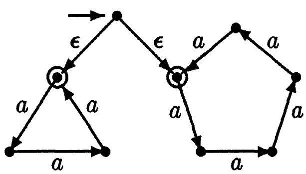


If the machine is in state s and the next input symbol is b, it can nondeterministically decide to do one of three things:

- read the b and move to state p;
- slide to t without reading an input symbol, then read the b and move to state q; or
- slide to t without reading an input symbol, then slide to u without reading an input symbol, then read the b and move to state r.

The set of strings accepted by this automaton is $\{b, bb, bbb\}$. □

Example 6.6 Here is a nondeterministic automaton with ϵ-transitions accepting the set $\{x \in \{a\}^* \mid |x| \text{ is divisible by 3 or 5}\}$:



The automaton chooses at the outset which of the two conditions to check for (divisibility by 3 or 5) and slides to one of the two loops accordingly without reading an input symbol. □

The main benefit of ϵ-transitions is convenience. They do not really add any power: a modified subset construction involving the notion of ***ε-closure*** can be used to show that every NFA with ϵ-transitions can be simulated by a DFA without ϵ-transitions (Miscellaneous Exercise 10); thus all sets accepted by nondeterministic automata with ϵ-transitions are regular. We will also give an alternative treatment in Lecture 10 using homomorphisms.

More Closure Properties

Recall that the concatenation of sets A and B is the set

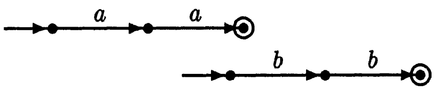
$$AB = \{xy \mid x \in A \text{ and } y \in B\}.$$

For example,

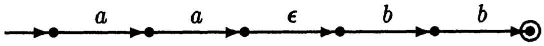
$$\{a, ab\}\{b, ba\} = \{ab, aba, abb, abba\}.$$

If A and B are regular, then so is AB. To see this, let M be an automaton for A and N an automaton for B. Make a new automaton P whose states are the union of the state sets of M and N, and take all the transitions of M and N as transitions of P. Make the start states of M the start states of P and the final states of N the final states of P. Finally, put ϵ-transitions from all the final states of M to all the start states of N. Then $L(P) = AB$.

Example 6.7 Let $A = \{aa\}$, $B = \{bb\}$. Here are automata for A and B:



Here is the automaton you get by the construction above for AB:



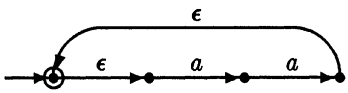
□

If A is regular, then so is its asterate:

$$\begin{aligned} A^* &= \{\epsilon\} \cup A \cup A^2 \cup A^3 \cup \cdots \\ &= \{x_1 x_2 \cdots x_n \mid n \geq 0 \text{ and } x_i \in A,\ 1 \leq i \leq n\}. \end{aligned}$$

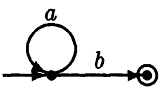
To see this, take an automaton M for A. Build an automaton P for A^* as follows. Start with all the states and transitions of M. Add a new state s. Add ϵ-transitions from s to all the start states of M and from all the final states of M to s. Make s the only start state of P and also the only final state of P (thus the start and final states of M are *not* start and final states of P). Then P accepts exactly the set A^*.

Example 6.8 Let $A = \{aa\}$. Consider the three-state automaton for A in Example 6.7. Here is the automaton you get for A^* by the construction above:



□

In this construction, you must add the new start/final state s. You might think that it suffices to put in ϵ-transitions from the old final states back to the old start states and make the old start states final states, but this doesn't always work. Here's a counterexample:



The set accepted is $\{a^n b \mid n \geq 0\}$. The asterate of this set is

$$\{\epsilon\} \cup \{\text{strings ending with } b\},$$

but if you put in an ϵ-transition from the final state back to the start state and made the start state a final state, then the set accepted would be $\{a, b\}^*$.

Historical Notes

Rabin and Scott [102] introduced nondeterministic finite automata and showed using the subset construction that they were no more powerful than deterministic finite automata.

Closure properties of regular sets were studied by Ginsburg and Rose [46, 48], Ginsburg [43], McNaughton and Yamada [85], and Rabin and Scott [102], among others.