More About Nondeterministic Finite Automata with Epsilon-Transitions

1 Introduction

“Epsilon-NFAs” have recently been discussed in lectures. During this discussion, the “epsilon-closure” of each state was defined in order to give a formal description of the computation of these devices on strings.

This document presents a different description of the epsilon-closure of a state than the one that was presented in class. This can be used to describe the notion of acceptance of a string by an $\epsilon$-NFA in a different way.

The document also includes a sketch of a proof that this new description agrees with the definition of an epsilon-closure that was presented in class, so that either description of an epsilon-closure be used without changing the notion of acceptance of strings by $\epsilon$-NFAs.

The new description is useful if you wish to write a computer program that decides whether a given $\epsilon$-NFA accepts a given string. It might also be helpful when reading proofs in the textbook.

2 More about Epsilon-Closures

Consider an $\epsilon$-NFA $N = (Q, \Sigma, \delta, q_0, F)$ like the one shown in Figure 1 on page 2. This $\epsilon$-NFA processes strings over the alphabet $\Sigma = \{a, b, c\}$.

Suppose you wanted to write a computer program that could be used to decide whether an $\epsilon$-NFA like this one accepts a given string or, possibly, to use this to generate a DFA for the same language. If you follow the process that has been described in class, then you will need to compute the epsilon-closures ECLOSE($q$) of states $q \in Q$. It is is probably easy enough to do this by hand for small examples like this one — but how would you write a program that could be used to do this for any given $\epsilon$-NFA?

We will consider another collection of sets of states, in order to see how to do this.

Definition 1. Suppose that $q$ is a state in a given $\epsilon$-NFA $N = (Q, \Sigma, \delta, q_0, F)$, and that $k$ is an integer that is greater than or equal to zero. Then

$$\text{EPATH}(q, k) \in \mathcal{P}(Q)$$

is the set of states that can be reached from $q$ by following a sequence of at most $k$ $\epsilon$-transitions.
It is reasonably easy to compute this set: It follows by Definition 1 that
\[ \text{EPATH}(q, 0) = \{q\} \]  
for every state \( q \), because the only state that you can reach from \( q \) without following any transitions at all is the state \( q \) itself. If \( k \geq 0 \), then
\[ \text{EPATH}(q, k + 1) = \text{EPATH}(q, k) \cup \bigcup_{s \in \text{EPATH}(q, k)} \delta(s, \epsilon). \]  
After all, in order to reach a state \( s \) by following \( k + 1 \) \( \epsilon \)-transitions, one must first reach a state \( r \) by following a sequence of \( k \) \( \epsilon \)-transitions, and then follow one more \( \epsilon \)-transition to get the rest of the way.

**Exercise:** Compute \( \text{EPATH}(q, k) \) for every state \( q \) in the given \( \epsilon \)-NFA and for \( 0 \leq k \leq 3 \).

You should be able to write a computer program that can be used to compute \( \text{EPATH}(q, k) \) for any given state and for any given integer \( k \geq 0 \) — all that you need to do is apply the equations (1) and (2) that are given above.

Now consider the definition of an epsilon-closure that is presented in the textbook: \( \text{ECLOSE}(q) \) should include all the states that can be reached from \( q \) by following a sequence of \( \epsilon \)-transitions. Since the sequence of \( \epsilon \)-transitions used to go from \( q \) to any state in \( \text{ECLOSE}(q) \) must be finite, the definitions that have presented for these sets suggest the following alternative description of an epsilon-closure.

**Claim 2.** The epsilon-closure of a state \( q \in Q \) is the union of its epsilon-paths, that is,
\[ \text{ECLOSE}(q) = \bigcup_{i \geq 0} \text{EPATH}(q, i). \]
This does not really help us to see how to compute ECLOSE(q). However, we will be able to discover an additional property of these sets and then use this property to produce a more useful relationship from the above equation.

**Claim 3.** If \( q \in Q \), \( k \geq 0 \), and \( \text{EPATH}(q, k) = \text{EPATH}(q, k+1) \), then \( \text{EPATH}(q, k) = \text{EPATH}(q, j) \) as well, for every integer \( j \geq k \).

**Sketch of Proof.** This can be proved using induction on \( j - k \): Notice that if \( j \geq k + 1 \) and

\[ s \in \text{EPATH}(q, j) \]

then there is a sequence of states

\[ q = r_0, r_1, r_2, \ldots, r_j = s \]

such that there is an \( \varepsilon \)-transition from \( r_{i-1} \) to \( r_i \) for every integer between 1 and \( j \), as shown in Figure 2, above.

Notice that this implies that

\[ r_{k+1} \in \text{EPATH}(q, k+1), \]

since it is possible to reach state \( r_{k+1} \) from \( q \) by following a sequence of \( k + 1 \) \( \varepsilon \)-transitions.

Since it is given that \( \text{EPATH}(q, k+1) = \text{EPATH}(q, k) \), it follows that

\[ r_{k+1} \in \text{EPATH}(q, k) \]

as well. It is therefore possible to reach \( r_{k+1} \) from \( q \) by following a shorter sequence of at most \( k \) \( \varepsilon \)-transitions too.

Now, if we extend this **shorter** sequence by following \( \varepsilon \)-transitions from \( r_{k+1} \) to reach the sequence of states \( r_{k+2}, r_{k+3}, \ldots, r_j = s \), then we end up with a sequence of at most \( j - 1 \) \( \varepsilon \)-transitions that can be used to \( s \) from \( q \). Since \( s \) was arbitrarily chosen from \( \text{EPATH}(q, j) \) it follows that

\[ \text{EPATH}(q, j) = \text{EPATH}(q, j - 1). \]

Since \( j \) was an arbitrarily chosen as an integer that is greater than or equal to \( k + 1 \), the above equality holds for every integer \( j \geq k + 1 \).

As noted above, one can include this in a proof by induction (that is only a few lines longer) to establish that

\[ \text{EPATH}(q, j) = \text{EPATH}(q, k) \]

for every integer \( j \geq k \), as claimed.
Now, we are making progress: After all, since $\text{EPATH}(q, j)$ is defined to include all states that can be reached from $q$ by following at most $k$ $\epsilon$-transitions (rather than “exactly” $k$ such transitions), it is clear that

$$\text{EPATH}(q, j) \subseteq \text{EPATH}(q, j + 1) \quad \text{for every integer } j \geq 0.$$  

This implies that $\text{EPATH}(q, j) \subseteq \text{EPATH}(q, k)$ for all integers $j$ and $k$ such that $0 \leq j \leq k$. If it is also true that $\text{EPATH}(q, j) = \text{EPATH}(q, k)$ for every integer $j \geq k$, as suggested above, then it follows by equation (3) that

$$\text{ECLOSE}(q) = \bigcup_{i \geq 0} \text{EPATH}(q, i) = \bigcup_{0 \leq i \leq k} \text{EPATH}(q, k) = \text{EPATH}(q, k).$$  

Unfortunately, this does not tell us how large $k$ might be, or even whether these conditions might be satisfied at all. However, the next result is easy to prove, and it provides a useful bound.

**Claim 4.** If $i \geq 1$ and $\text{EPATH}(q, i) \neq \text{EPATH}(q, i - 1)$ then

$$|\text{EPATH}(q, i)| \geq i + 1.$$  

*Sketch of Proof.* Once again, it is easy to prove this using induction. Notice that $\text{EPATH}(q, 0)$ has size one, and that the size of $\text{EPATH}(q, i)$ must be at least one more than the size of $\text{EPATH}(q, i - 1)$ if the sets $\text{EPATH}(q, i - 1)$ and $\text{EPATH}(q, i)$ are different. □

Since $\text{EPATH}(q, i)$ is a subset of $Q$, its size cannot be greater than $|Q|$. Therefore, the above claim and equation (4) imply the following.

**Claim 5.** Let $N = (Q, \Sigma, \delta, q_0, F)$ be an $\epsilon$-NFA and let $q \in Q$. Then

$$\text{EPATH}(q, |Q| - 1) = \text{EPATH}(q, |Q|) = \text{ECLOSE}(q).$$  

Therefore we will not have to compute very many sets, at all, before we succeed in computing the \(\epsilon\)-closure $\text{ECLOSE}(q)$.

The following programming exercise should now be reasonably straightforward (provided that you have found a way to represent an $\epsilon$-NFA so that you can specify one as input):

**Exercise:** Write a computer program that takes an $\epsilon$-NFA as input and that computes the $\epsilon$-closure of each of the given automaton’s states.

### 3 Another Way to Think about Reachability

The above ideas can also be used to find an alternative (but equivalent) description of the extended transition function

$$\delta : Q \times \Sigma^* \to \mathcal{P}(Q)$$

that might be easier to use, or that might make it easier to understand some of the proofs that will be presented later on.

Recall the definition that we have now:

$$\tilde{\delta}(q, \epsilon) = \text{ECLOSE}(q)$$
\[ \tau_i \in \Sigma \cup \{\epsilon\} \text{ for } 1 \leq i \leq \ell \text{ and } w = \tau_1 \tau_2 \tau_3 \ldots \tau_{\ell} \]

Figure 3: \( r \in \widehat{\delta}(q, w) \)

and

\[ \widehat{\delta}(q, y\sigma) = \bigcup_{s \in \delta(q, y)} \bigcup_{t \in \delta(s, \sigma)} \text{ECLOSE}(t) \]

whenever \( y \in \Sigma^* \) and \( \sigma \in \Sigma \).

This implies that, if \( w = \sigma_1 \sigma_2 \ldots \sigma_n \in \Sigma^* \) is a string of length \( n \), then

\[ r \in \widehat{\delta}(q, w) \]

if and only if there is a sequence of states

\[ q = t_0, u_0, t_1, u_1, t_2, u_2, \ldots, t_{n-1}, u_{n-1}, t_n, u_n = r \]

such that \( u_i \in \text{ECLOSE}(t_i) \) for \( 0 \leq i \leq n \) and such that \( u_i \in \delta(t_{i-1}, \sigma_i) \) for \( 1 \leq i \leq n \). (Once again, this complicated statement can be proved — using induction on the length \( n \) of the string \( w \).)

Notice that, by what has been established above, it must be possible to reach \( u_i \) from \( t_i \) by following a sequence of at most \( |Q| - 1 \) \( \epsilon \)-transitions. This can be used to establish the following.

**Claim 6.** Suppose that \( q \) and \( r \) are states in an \( \epsilon \)-NFA \( N = (Q, \Sigma, \delta, q_0, F) \) and that \( w \in \Sigma^* \). Then

\[ r \in \widehat{\delta}(q, w) \]

if and only if there is a sequence of \( \ell \) symbols or copies of the empty string

\[ \tau_1, \tau_2, \ldots, \tau_{\ell} \in \Sigma \cup \{\epsilon\} \]

for some integer \( \ell \geq 0 \), and a sequence of states (with length \( \ell + 1 \))

\[ q = s_0, s_1, s_2, \ldots, s_{\ell} = r \]

such that

\[ s_i \in \delta(s_{i-1}, \tau_i) \]

for \( 1 \leq i \leq \ell \) and such that \( w \) is the concatenation

\[ w = \tau_1 \tau_2 \ldots \tau_{\ell}, \]

as shown in Figure 3 above.
The integer $\ell$ is not necessarily the same as the length of the string $w$: It is the sum of the length of $w$ and the number of $\epsilon$-transitions that are followed along the above path.

Some claims discussed in this course (including the correctness of a construction for the generation of finite state machines for languages from “regular expressions” for the same languages) might be easier to understand and prove if this characterization is used. It is therefore useful to keep this characterization of reachability in mind. Furthermore, the above analysis proves a bound on $\ell$: It can be established yet again, by induction on the length of the string, that if $r \in \tilde{\delta}(q, w)$ then sequences like the above exist and, furthermore,

$$ \ell \leq |Q| \cdot (|w| + 1). $$

**Exercise:** Complete a proof that a string $w \in \Sigma^*$ is accepted by a given $\epsilon$-NFA, using the description given above, if and only if the string is accepted by the $\epsilon$-NFA, using the definition given in class.