

CS 360: Theory of Computation

Winter, 2011

Proving regularity of languages, an example: $\text{half}(L)$

Various operations can be performed on regular languages to yield other languages. The languages obtained thus may or may not be regular. Assignment #3 will ask you to example several such operations.

As an illustration of some of the techniques one uses to solve such problems, we will solve one of the exercises from the textbook.

Problem 1. Let L be a language. Define $\text{half}(L)$ to be

$$\{x \mid \text{for some } y \text{ such that } |x| = |y|, xy \text{ is in } L\} .$$

That is, $\text{half}(L)$ is the set of first halves of strings in L . Prove that for each regular language L , $\text{half}(L)$ is regular.

Suppose we are told that L is regular. What useful information does this give us? Well, we know that there is a DFA that accepts L . Also, there is a regular expression that represents L . We could use any of these facts in proving that $\text{half}(L)$ is regular.

Let M be the DFA accepting L . Let q_0 be the start state and F the set of final states of M . Now consider a string x . How do we check if x belongs to $\text{half}(L)$? We have to check if there is a string y of the same length as x which when concatenated with x gives a string in L , i.e., the automaton moves to a final state on input xy . Suppose M goes to state q_i on seeing input x , i.e., $\delta(q_0, x) = q_i$. We must check if there is a string y , $|y| = |x|$, such that y takes the DFA from q_i to a final state, i.e., $\delta(q_i, y) \in F$.

Suppose we have seen n symbols so far, i.e., $|x| = n$. Let S_n be the set of all states that lead to a final state on *some* input of length n . If $q_i \in S_n$, then $x \in \text{half}(L)$ and vice versa. (Prove this!) Thus if we can keep track of S_n and q_i , we can determine if $x \in \text{half}(L)$. Note that S_0 is just F . We can obtain S_{n+1} from S_n and δ . Let T be the set of all states that have *some* transition (on a single symbol) to a state in S_n . We claim that $S_{n+1} = T$. Indeed, if for each state $q \in S_n$ there is some input of length n that takes M from q to a final state, then for each state $q' \in T$ there is some input of length $n + 1$ that takes M from q' to a final state. Further, if S_n contains *all* states from where a final state can be reached in n transitions, T contains *all* states from where a final state can be reached in $n + 1$ transitions. Thus $T = S_{n+1}$.

How to we keep track of and update S_n ? We construct a DFA M' that stores this information in its state.

Let Q be the set of states of M . The states of M' are elements of $Q \times 2^Q$. Each state of M' is a pair consisting of a single state of M , and a set of states of M . Let δ' be the transition function for M' . We construct M' such that if an input x of length n takes M to state q_i , x takes M' to the state (q_i, S_n) where S_n is the set of states defined above.

Formally we define

$$\forall S \in S^Q, \text{prev}(S) = \{q \in Q \mid \exists a \in \Sigma, q' \in S, \delta(q, a) = q'\}$$

Note that $S_{n+1} = \text{prev}(S_n)$. We define the transition function δ' as follows:

$$\delta'((q, S), a) = (\delta(q, a), \text{prev}(S))$$

The start state is $q'_0 = (q_0, S_0)$, i.e., (q_0, F) . The set of final states is $F' = \{(q, S) \mid q \in Q, S \in 2^Q, q \in S\}$.

Now we can prove by induction that $\delta'(q'_0, x) = (\delta(q_0, x), S_n)$, where $|x| = n$. By definition of the set of final states, x is accepted iff $\delta(q_0, x) \in S_n$. Thus M' accepts the language $\text{half}(L)$.

Thus we have constructed a DFA that accepts $\text{half}(L)$. This proves that $\text{half}(L)$ is regular. As an exercise, complete the inductive proofs we need to prove the correctness of this construction.

Note: In general, to show that two languages L_1 and L_2 are equal, we must prove that every string in L_1 belongs to L_2 and every string in L_2 belongs to L_1 . This establishes that $L_1 \subseteq L_2$ and $L_2 \subseteq L_1$, proving that $L_1 = L_2$. In proving the correctness of the above construction, we have to prove something similar to show that $L(M') = \text{half}(L)$. Consider $x \in \Sigma^*$ such that $\delta'(q'_0, x) = (\delta(q_0, x), S_n)$ where $|x| = n$. Then we must prove that $x \in \text{half}(L) \Rightarrow q'_0 \in S_n$ and $q'_0 \in S_n \Rightarrow x \in \text{half}(L)$. We mentioned these facts in the above discussion, but did not prove them. They can be proved from the definition of $\text{half}(L)$ and S_n .

This technique for proving equality of languages also applies to proving equivalence of regular expressions or proving that a regular expression corresponds exactly to the set defined by an informal English language description. For example, given an English description of a language, to prove the *correctness* of a regular expression, we must prove that every string generated by the regular expression fits the English language description, and further, every string that fits the English language description can be generated by the regular expression.

We can also construct an NFA M'' for $\text{half}(L)$, the states of which are elements of $Q \times Q' \cup \{q''_0\}$. The idea behind this construction is similar to the idea that the DFA construction was based upon. Here, if an input x , $|x| = n$, takes M to state q_i , x takes M'' to all states of the form (q_i, q) , where q is a state from which a final state can be reached in n transitions, i.e., $q \in S_n$. Note that the first component q_i is the same as the first component in the state for the DFA M' . However for the second component, in place of the set of states S_n , the NFA *guesses* an element of S_n . The start state of M'' is a new state q''_0 . q''_0 has ϵ transitions to all states of the form (q_0, q) where $q \in F$. The final states are the states of the form (q, q) . To define the transition function of M'' , we first define the function prev on states of M :

$$\forall p \in Q, \text{prev}(p) = \{q \in Q \mid \exists a \in \Sigma, \delta(q, a) = p\} .$$

The transition function δ'' of M'' defined as follows:

$$\delta''((q, p), a) = \{(\delta(q, a), p') \mid p' \in \text{prev}(p)\} .$$

Do you see the connection between this construction and the one given previously? As an exercise, prove that $L(M'') = \text{half}(L)$. Your proof should be similar to the proof of the correctness of the DFA construction.