

Homework 2 Solutions

Note: These solutions are not necessarily model answers. Rather, they are designed to be tutorial in nature, and sometimes contain a little more explanation than an ideal solution. Also, bear in mind that there may be more than one correct solution.

- 1 First, we observe that any DFA is an all-paths NFA, so all-paths NFAs accept all regular languages. For the other direction, we need to show that any language accepted by an all-paths NFA is regular. Let $N = (Q, \Sigma, \delta, q_0, F)$ be an arbitrary all-paths NFA. We will construct a standard NFA $M = (Q', \Sigma, \delta', q'_0, F')$ that recognizes the same language as N . The construction will be similar to the conversion from NFAs to DFAs discussed in class (and in Theorem 1.19 of Sipser). The resulting NFA M will be almost a DFA, in that there will be at most one possible path in each computation. However, unlike in a DFA, every time we see a “dying” computation in N , the corresponding computation in M will die as well. (Note that this is where our construction differs from the NFA-to-DFA construction!)

$$\begin{aligned}
 Q' &= \mathcal{P}(Q) \\
 q'_0 &= \{q_0\} \\
 \delta'(R, a) &= \begin{cases} \emptyset & \text{if for some } r \in R, \delta(r, a) = \emptyset; \\ \{q \in Q \mid q \in E(\delta(r, a)) \text{ for some } r \in R\} & \text{otherwise.} \end{cases}
 \end{aligned}$$

(Recall that $E(R)$ is the set of states reachable from R using zero or more ϵ transitions.) Finally, we want M to be accepting if its final state contains *only* accepting states of N , so we define $F' = \mathcal{P}(F)$.

An alternative argument is the following: First, augment the all-paths NFA N into an all-paths NFA N_1 such that N_1 accepts the same language as N , but no computation of N_1 dies. (One way to do this is to add a rejecting state q_{die} to N ; then for all states $q \in Q \cup \{q_{\text{die}}\}$, add a transition from q to q_{die} labeled by all symbols in Σ unused by other outgoing arrows at q .) Now, reverse the accepting and rejecting states of N_1 ; call the resulting machine N_2 . Think of N_2 as a standard NFA: The language accepted by N_2 is then the complement of the language L accepted by the all-paths NFA N_1 . It follows that the complement of L is regular. Since regular languages are closed under complement (see Problem 3 of Homework 1), L must be regular as well.

3 (Coming)

4 For each of these, there are many possible valid regular expressions.

(b) $((a \cup e \cup i \cup o \cup u) \Sigma^* \cup \epsilon) (\text{ing})$

(d) $0^* (10^* 10^* 10^*)^*$

(f) $(10 \cup 01)^*(1 \cup 0 \cup \epsilon)$

To see the above, first notice that every even length string that belongs in the language is such that *every* prefix of it has equally many zeros and ones, because if not, then there are either two more zeros than ones or two more ones than zeros. It is easy to check, by induction on the length of the string, that the set of all even length strings where every prefix has the same number of zeros and ones is given by $(10 \cup 01)^*$. To then get our language, we simply concatenate either a zero or a one or nothing to the end of every even length string with the above property.

- 5 (a) False. E.g., take $R = 0$ and $S = 1$. Then the string 010 belongs to $(R \cup S)^*$ but not to $R^* \cup S^*$.
[Note that, when the answer is 'False', the only convincing way to justify it is by giving a concrete counterexample (like the one above). More general, woolly arguments don't work. The same applies to part (c) below.]
- (b) True. The fact that $L(R^*) \subseteq L((R^*)^*)$ is immediate because the language on the right contains all words that consist of finite sequences of words from $L(R^*)$, so in particular it contains all words in $L(R^*)$. We also have to show that $L((R^*)^*) \subseteq L(R^*)$. To see this, note that any word in $L((R^*)^*)$ can be written in the form $w_1 w_2 \dots w_n$ for some $n \geq 0$, where each w_i is a word in $L(R^*)$. But each w_i can in turn be written in the form $x_{i1} x_{i2} \dots x_{im_i}$ for some $m_i \geq 0$, where each x_{ij} is a word in $L(R)$. So any word in $L((R^*)^*)$ can be written as a sequence of words from $L(R)$, and hence belongs to $L(R^*)$. Thus $L((R^*)^*) \subseteq L(R^*)$, as claimed.
[You could try for some kind of informal argument for this part, e.g., based on conversion to NFAs. The above scheme is really the only convincing way to go here: note that we're just trying to show that two sets are equal, so we have to show each is contained in the other.]
- (c) False. E.g., take $R = 0$ and $S = 0 \cup \epsilon$. Then $L(R^*) = L(S^*) = \{0\}^*$, but $L(R) = \{0\}$ and $L(S) = \{0, \epsilon\}$.