

Recommended problems 11 — CS3512 — Fall '11

1 Use the Distinguishability Theorem to prove that any DFA that recognizes the language $L = \{0, 1\}^* \{00\} \{0, 1\}$ has at least 5 states.

Proof. If we can produce 5 strings that are pairwise distinguishable wrt L , then it follows by the Distinguishability Theorem that any DFA recognizing L must have at least 5 states.

5 such strings are: 111,110,100,000,001.

Now we need to check the claim, by finding a distinguishing string for each pair. 00 distinguishes 111 and 110 wrt L . 0 distinguishes 111 and 100. Λ distinguishes 111 and 000, and also distinguishes 111 and 001. 0 distinguishes 110 and 100. Λ distinguishes 110 and 000, as well as 110 and 001, and 100 and 000, and 100 and 001. Finally, 0 distinguishes 000 and 001.

2 Use the Distinguishability Corollary to prove that no DFA recognizes the language of binary palindromes.

Proof. If we can produce an infinite set of strings that are pairwise distinguishable wrt this language, then it follows by the Distinguishability Corollary that no DFA recognizes the language. Consider the strings $0^n 1$ for all $n \in \mathcal{N}$. Take any two such strings $0^m 1$ and $0^n 1$ with $m \neq n$. They are distinguished by the string 0^n , since $0^n 1 0^n$ is a palindrome while $0^m 1 0^n$ is not.

(Note: The lecture notes give a more complex argument showing that *all* strings are distinguishable from one another wrt this language.)

3 Show that $L = \{0^{2^n} \mid n \in \mathcal{N}\}$ is not regular. Use the Distinguishability Corollary.

What we'll observe is that L itself is an infinite set of strings that are pairwise distinguishable wrt L . From this it follows by the Distinguishability Corollary that L is not accepted by any DFA, and so by Kleene's Theorem that L is not regular.

Take any strings $0^{2^m}, 0^{2^n}$ with $m < n$. These strings are distinguished wrt L by 0^{2^n} , since $2^n < 2^m + 2^n < 2^n + 2^n = 2^{n+1}$. Thus, any two strings in L are distinguishable wrt L .

4 If $L = 0(0+1)^*0(0+1)$, how many equivalence classes of I_L ? (Equivalently, how many states in a minimal DFA recognizing L ?) Also, give regular expressions for the equivalence classes $[0]$, $[00]$, $[000]$ and $[001]$?

There are six equivalence classes.

$$[0] = 0 + 01 + 0(0+1)^*11$$

$$[00] = 00 + 0(0+1)^*10$$

$$[000] = 0(0+1)^*00$$

$$[001] = 0(0+1)^*01$$

How to find these answers? Draw the minimal DFA, as given by the Minimal DFA Theorem in the lecture notes. (As in class, draw the DFA starting from the initial state, and use the Distinguishability Lemma to decide when a new state is needed. Notice that you can verify that your DFA is minimal using the Distinguishability Theorem.) The minimal DFA for L has 6 states, and, as we discussed in class, each of those states corresponds to one of the equivalence classes — which is exactly the set of all strings that “take the DFA to that state”.

Then you can find a regular expression for each of the six equivalence classes by inspecting the DFA. (I'm not saying this is easy in most cases, but it works ok here. And we did things like this in class. There are algorithms that can be used to accomplish the various tasks here, but we are not learning them.)

For completeness in checking your DFA, note that $[\Lambda] = \Lambda$ and $[1] = 1(0+1)^*$.

For help in thinking about how to find the regular expressions above, here are

some informal remarks. Notice that for the other 4 equivalence classes, all strings begin with 0. Beyond that, you can notice that any string reaching the state for [000] not only begins with 0 but also ends with 00, and all such strings reach that state. Similar remarks apply for [001]. For [00] you might notice that 00 is unusual among the strings that take us to the state for [00]; all the others end with 10. Moreover, every string that begins with 0 and ends with 10 goes to the state for [00]. And about [0], you might notice that all strings that both start with 0 and end with 11 take us to the state for [0]. In addition, we can get there on strings 0 and 01.

You may also notice that this DFA is partly similar to the DFA for the language L_1 from the family $L_n = (0 + 1)^*1(0 + 1)^n$ of languages that were used to demonstrate the Distinguishability Theorem (and can also be used to illustrate the relationship between DFA's and NFA's).