

Sample Midterm Exam

February 2012

This is a sample based upon an old hour exam. The material covered and notation may be slightly different; the purpose of this handout is just to give you some idea of the sort of thing you might see.

1. (20 points) Using the construction from class or the text, construct a deterministic finite automaton (DFA) equivalent to the three-state NFA shown below.

PICTURE of 3-state NFA HERE.

Solution: Run the algorithm from the book. The NFA N_4 on page 57 and the DFA D on page 58 form a sample question and answer.

2. (a) (25 points) Prove that the language

$$L = \{\text{all strings over } \{0, 1\} \text{ containing more 0's than 1's}\}$$

is *not* regular.

Solution: Assume for contradiction that L is regular. Then the Pumping Lemma applies with pumping length p . Consider the string $s = 1^p 0^{p+1}$. By the Pumping Lemma, we can write $s = xyz$, with $|xy| \leq p$ and y nonempty. It must be that y is a nonempty string of 1's. $xyyyz$ must have at least $(p+2)$ 1's (it has $p+2|y|$ 1's), so $xyyyz \notin L$, which is a contradiction. Therefore L is not regular.

- (b) (15 points) Prove that L is context free.

Solution: PDA idea for L : Use the stack to keep track of the next 0 vs. 1 balance seen so far.

Push a bottom of stack marker onto the stack, let's call it Z , and transition to the main working state.

In the main working state, the following transitions keep us in the main working state:

- if the top of stack is Z: If input 0 is read, push a A onto the stack; if input 1 is read, push a B onto the stack. (Formally the first transition would be 0; Z; AZ, and the second would be 1; Z; BZ.)
- If the top of stack is A: If 0 is read, push another A; if 1 is read, pop that A.
- if the top of stack is B: If 0 is read, pop that B; if 1 is read, push another B.

One more transition from the main working state to the unique accepting state: Ignore the input, and transition only if the top of stack is a A. (I.e., we can accept if we have a net surplus of character 0.)

3. (15 points) Consider the following variation of a deterministic finite automaton (DFA). Everything is the same, except that we allow more than one “start” state. A string is accepted by this machine if at least one of the start states sends it to an accepting (final) state. Argue that any language accepted by such a machine can also be accepted by an ordinary DFA.

Solution: Here is the algorithm to convert such a variant to a DFA for the same language:

- Convert the variant machine to an NFA by adding a new start state, and an ϵ transition from that new start state to each of the former start states.
- Convert that NFA to a DFA for the same language using the standard algorithm.

4. (15 points) If L is a regular language, show that

$$\text{Reverse}(L) = \{\text{all strings in } L \text{ written backwards}\}$$

is also a regular language. (Hint: Use the result from the previous problem.)

Solution: Let M be the DFA for L . Change M to a variant DFA as in the previous problem by reversing all the arrows of M 's transition diagram, making M 's start state an accepting state, and making each of M 's accepting states a start state. This variant DFA accepts the reverse of L .

5. (10 points) We know that regular languages are closed under finite union. Show that regular languages are *not* closed under infinite union. (Hint: Find a language that is not regular but is the infinite union of regular languages.)

Solution: Put $L_i = \{0^i 1^i\}$ for each nonnegative integer i . Each L_i contains only one string, and so is regular.

But $\bigcup_{i=0}^{\infty} L_i = \{0^n 1^n : n \geq 0\}$, which is not regular.