Automata Test 1, Fall 2023

Problem 1. Why do we need to study automata? Provide two main reasons.

Problem 2–4. Let us consider the automaton that has two states: n (the student is in normal state) and p (the student is on probation); n is the starting state and it is also a final state. The symbols are a, b, c, and f that describe letter grades. From each state, grades a, b, and c lead to n, while grade f leads to p.

Problem 2. Trace, step-by-step, how this finite automaton will check that the word afa belongs to this language. Use the tracing to find the parts x, y, and z of the word afa corresponding to the Pumping Lemma. Check that the "pumped" word $xyyz$ will also be accepted by this automaton.

Problem 3. Write down the tuple $\langle Q, \Sigma, \delta, q_0, F\rangle$ corresponding to this automaton:

- Q is the set of all the states,
- Σ is the alphabet, i.e., the set of all the symbols that this automaton can encounter;
- $\delta : Q \times \Sigma \to Q$ is the function that describes, for each state q and for each symbol s, the state $\delta(q, s)$ to which the automaton that was originally in the state q moves when it sees the symbol s (you do not need to describe all possible transitions this way, just describe two of them);
- q_0 is the staring state, and
- F is the set of all final states.

Problem 4. Use a general algorithm that we had in class to generate a context-free grammar corresponding to this automaton. Show how this grammar will generate the word afa.

Problem 5. Let A_1 be the automaton described in Problem 2. Let A_2 be an automaton that accepts only straight-A students. This automaton has two states: the starting state s which is also final, and the error state e. The transitions are: as follows:

- from the start state, a lead back to the start state, while every other grade leads to the state e;

- from the state e, any symbol leads back to this state.

Use the algorithm that we had in class to describe the following two new automata:

- the automaton that recognizes the union $A_1 \cup A_2$ of the two corresponding languages, and
- the automaton that recognizes the intersection of the languages A_1 and A_2.

Problem 6. Use the general algorithm that we learned in class to design a non-deterministic finite automaton that recognizes the language $f(a \cup b)^*$ – that corresponds to the case when a student first got an F but after that gets only As and Bs:

- first, describe the automata for recognizing a, b, and f;
- then, combine them into the automata for recognizing the union $a \cup b$, and the Kleene star $(a \cup b)^*$;
- finally, combine the automata for f and $(a \cup b)^*$ into an automaton for recognizing the desired composition of the two languages.

Problem 7. Use the general algorithm to transform the resulting non-deterministic finite automaton into a deterministic one.

Problem 8–9. Use a general algorithm to transform the finite automaton from Problem 2 into the corresponding regular expression. Start with eliminating the state n.

Problem 10. To make changes to Texas Constitution, we need to have at least 2/3 votes, i.e., the number of those who vote For should be at least twice larger than the number of those who vote Against. If we denote For by f, and Against by a, then the sequences faf and $ffaf$ lead to acceptance while the sequence $afaf$ does not. Prove that the language L of all the sequence that lead to acceptance – i.e., that have at least twice more f's than a's – is not regular.