CS 3350 Automata, Computability, and Formal Languages
Fall 2018, Test 1

Last 4 digits of your UTEP ID number: _______ _______ 100

General comments:

- you are allowed up to 5 pages of handwritten notes;
- if you need extra pages, place the last 4 digits of ID number on each extra page;
- the main goal of most questions is to show that you know the corresponding algorithms; so, if you are running out of time, just follow the few first steps of the corresponding algorithm;
- each question will be graded on its own merit; so, for example, if when answering to the first part of the question, you got a wrong automaton, but on the second part, you correctly traced the new automaton, you will get full credit for the second part.

Good luck!

1-4. In class, we studied an automaton for recognizing valid Java identifiers. This automaton has 3 states: start (s), identifier (i), and error (e). Start is the starting state, identifier is the only final state. The transitions are as follows:

- from s, any letter (a, ..., z, A, ..., Z) leads to i, any other symbol leads to e;
- from i, any letter, any digit (0, ..., 9), or an underscore symbol _ lead back to i, while any other symbol leads to e;
- from e, every symbol leads to e.

1. Is any of the three states a sink state? Explain your answer.

2-3. Trace, step-by-step, how this finite automaton will check whether the following two words (sequences of symbols) represent a valid Java identifier:

- the word number2 (which this automaton should accept) and
- the word 2ndNumber (which this automaton should reject).

4. Write down the tuple <Q, Σ, δ, q₀, F> 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;
- δ: Q x Σ → Q is the function that describes, for each state q and for each symbol s, the state δ(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);
- q₀ is the staring state, and
- F is the set of all final states.
1. Error is a sink state because it is a black hole, once you are in that state, you cannot get out, it only produces an error. The automaton rejects the input once it reaches the error state.

2. \texttt{number 2} \quad \text{accepted}

3. \texttt{2nd Number} \quad \text{rejected}
4.) \( Q = \{ \text{start}, \text{identifier}, \text{error} \} \)
\( \Sigma = \{ a, \ldots, z, A, \ldots, Z, 0, \ldots, 9, \_, \text{every other symbol} \} \)
\( \delta(\text{start}, b) = \text{identifier} \)
\( \delta(\text{identifier}, +) = \text{error} \)
\( q_0 = \text{start} \)
\( F = \{ \text{identifier} \} \)
5. Draw an automaton for recognizing all possible binary signed integers. Trace this automaton on the example of numbers +10 (that it should accept), -10 (also accepted), and 10 (should be rejected).
6-8. Let A be an automaton described in Problem 1. Let B be the following automaton that accepts all the strings that contain only letters but not any other symbols. This automaton has two states: the start state which is also a final state, and the sink state. The transitions are as follows:

- from the start state, any letter leads back to the start state, every other symbol leads to sink;
- from the sink state, any symbol leads back to sink.

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

- the automaton that recognizes the union \( A \cup B \) of the two corresponding languages, and
- the automaton that recognizes the intersection of the languages A and B.

7-8. Test these two new automata step-by-step on the following words:

- test the union automaton on the example of the words Var (that it should accept) and 2words (that it should reject);
- test the intersection automaton on the example of the words Var (that it should accept) and Var2 (that it should reject).
7.) union

- var
  - s, st
  - i, st
  - s

accepted

- var
  - s, st
  - e, si
  - s, si
  - s, si
  - s, si
  - s, si

(accepted)

8.) intersection

- var
  - s, st
  - i, st
  - i

accepted

- var
  - s, st
  - i, st
  - i, st
  - s, st
  - i, si

(s, si)

rejected
9-10. Use the general algorithm that we learned in class to design a non-deterministic finite automaton that recognizes the language \((0 \cup 1)(1 \cup 2)\):

- first, describe the automata for recognizing 0, 1, and 2;
- then, combine them into the automata for recognizing the unions 0 \cup 1 and 1 \cup 2;
- finally, combine the two union automata into an automaton for recognizing the composition \((0 \cup 1)(1 \cup 2)\) of the two union languages.