CISC 462 Assignment 1 Postmortem

Taylor J. Smith

October 4, 2018

1. **Whole question: 10 marks.**

   Most students understood that the Turing machine could only accept the language `a`*. Very few students thought that the Turing machine could accept a string with `b`'s in it. The example input string should make it clear why that isn't the case.

   It is important to note that, if a Turing machine doesn't accept a string, then it also doesn't necessarily reject that string. A Turing machine can enter into an infinite loop and neither accept nor reject. No marks were deducted if you mistakenly said that the Turing machine "rejected" the string `abba`.

   In terms of notation, many students wrote the language accepted by the Turing machine as $\{a^i \mid i \geq 0\}$. A much more concise way of writing this is by using the Kleene star notation; that is, by writing `a`*. Some students wrote $\{a\}^*$ or $\{a\}^i$. The brackets surrounding the symbol are unnecessary; we are operating on the symbol itself, not the set containing the symbol.

2. **Part (a): 2.5 marks. Part (b): 2.5 marks. Part (c): 2.5 marks. Part (d): 2.5 marks.**

   This question was generally done well.

   Be sure that you read questions carefully! Some students thought the input string for part (d) was `0000`, when it was actually `000000`.

3. **Part (a): 5 marks. Part (b): 5 marks.**

   When we simulate a DFA using a Turing machine, we don't accept solely based on the fact that we reached the end of the input (otherwise we could accept all strings just by reading them). We must also consider whether the DFA would be in an accepting state by that point.

   A few students assumed that the DFA would only accept an input string after reading exactly three symbols (instead of reading three or more symbols). Other students neglected to consider what the DFA/Turing machine would do if it were given an empty input string.

   There were a few small details that some students neglected to include when defining their Turing machine: Turing machine transitions require a "write symbol", even if we don't care what is being written on the tape, and we must always define a reject state for a Turing machine. All non-accepting DFA computations would then go to the reject state.

4. **"Getting the idea": 5 marks. Providing simulation details: 5 marks.**

   This question was generally done well.

   A number of students used "bottom of stack" symbols to determine when a stack was empty. Strictly speaking, these symbols are not necessary, though using them doesn't affect the solution.

5. **Turing machine: 8 marks. Sequence of configurations: 2 marks.**

   Although every Turing machine should include a reject state, such a state was not needed/used in this question, so no marks were deducted if you didn't include one.

   The Turing machine in this question should work on blank tapes (by transitioning immediately to the accept state), though no marks were deducted if it didn't.

6. **Whole question: 10 marks.**

Many students were testing the condition $2 \cdot |w|_{\mathtt{a}} = |w|_{\mathtt{b}}$, instead of the actual condition $|w|_{\mathtt{a}} = 2 \cdot |w|_{\mathtt{b}}$. If you did this, then no marks were deducted because it's essentially the same procedure. But again, be sure that you read questions carefully!

Some students shifted the input string by one space to the right so that they could determine when the input head reached the first symbol of the string. This is not necessary; you can use a special "marked" symbol to accomplish the same task (e.g., if the first symbol is `a`, overwrite the symbol with the "marked" symbol $\dot{\mathtt{a}}$).

Questions/comments? Feel free to stop by my office hours or send me an email at `tsmith [at] cs [dot] queensu [dot] ca`.