Math 385 Final Exam review

Refer to the review sheets for previous exams for topics that were on previous exams. These topics will also be covered by the final. This sheet covers topics since the second mid-term.

You should be able to give formal definitions for the following, and, given a particular example, explain how it matches up with the formal definition.

- Turing machine
- Instantaneous description of the state of a Turing machine, and taking one step in the execution of a Turing machine affects it.
- Language accepted by a Turing machine
- Variants of Turing machines, including machines with multiples, multiple heads, or both, machines with stay options or other movement options. This may include variants you have not seen before.
- Universal Turing machines, including a scheme for encoding its input.
- Turing-decidable, Turing-acceptable, and Turing-enumerable languages.

You should be able to do the following:

- Use the pumping lemma for context-free languages to prove that specific languages are not context-free.
- Use closure properties of the family of context-free languages to prove that specific languages are not context-free.
- Construct a Turing machine (with a description at the requested level of detail) that accepts a particular language.
- Construct a Turing machine (with a description at the requested level of detail) that performs a certain computation on the input.
- Give the sequence of instantaneous descriptions as a Turing machine is run on a given input.
- Explain how a variant Turing machine of a certain specific type can be simulated by a standard Turing machine.
- Sketch how the universal Turing machine works.
- Explain why Turing-decidable languages are Turing-acceptable.
- Explain why Turing-acceptable languages are Turing enumerable, and vice versa.
- Prove statements about other relationships between Turing-decidable, Turing-acceptable, and Turing-enumerable languages.
- Explain our construction of a language that is not Turing-acceptable.
- Explain our construction of a language that is Turing-acceptable but not Turing-decidable.
- Give in outline the proof that the halting problem is not Turing-decidable.
- Use the fact that halting problem is not Turing-decidable to explain why some other language is not Turing-decidable.
- Explain why the language generated by an arbitrary grammar is Turing-acceptable, and why every Turing-acceptable language is generated by some grammar.

Relevant problems in the book with solutions in the back:

- 8.1: 3, 7, 8, 10
- 9.1: 2, 7ab, 10, 12, 19
- 9.2: 3a, 5, 8
- 10.1: 4ab, 6, 9, 11
- 10.2: 2, 5, 9
- 10.3: 3, 6, 7
- 10.4: 3, 8
- 11.1: 2, 6, 11, 14, 18
- 11.2: 1, 3, 7
- 12.1: 3, 7, 10, 13, 16
- 12.2: 3, 6