Syllabus for Midterm 1

The first midterm will test material covered in lectures from week 1 through 4; see lecture schedule on the course web page. This roughly corresponds to the following parts of Jeff's book: Appendix I (induction), Chapter 1 (strings), Chapter 2 (regular languages and expressions), Chapter 3 (DFAs) except section 3.9 (Myhill Nerode theorem) and 3.10 (Minimal DFAs), Chapter 4 (NFAs), and Chapter 5 (Context-free Languages) except Sections 5.6–5.8. Turing machines are not covered in midterm 1. Any results proved in discussion labs and homework assignments and guided problem sets are also part of material that can be tested on the midterm.

Specific skills that may be test include (the following list may not be exhaustive):

Basic mathematics

- Comfort with set notation, especially set operations like cross product and power set. Should know how to read and understand formally described sets, and should be able to describe new sets precisely.
- Familiarity with alphabets, strings, and languages.
- Ability to critically evaluate proofs and write proofs, especially induction proofs.
- Ability to comprehend inductive definitions.

Formal models of computation (regular expressions, DFAs, NFAs, CFGs)

- Understand formal definitions of machines, grammars and expressions. Be able to execute machines on simple examples, and infer if strings belong to sets defined by expressions/grammars. Understand what it means for a language to be described/accepted by a computational model.
- Ability to design machines/grammar/expressions to describe/accept languages. Ability to formally describe them.

Transformations between computational models

- Familiarity with proofs transforming NFAs to DFAs, and regular expressions to NFAs. Ability to carry out these constructions on examples.
- Familiarity with the cross product construction to run multiple machines simultaneously.
- Know asymptotic bounds of the resulting automata constructed by these transformations.
- Ability to perform new transformations on automata to prove regularity or construct automata/expressions with special properties.

Closure properties

- Know standard closure properties (concatenation, union, intersection, complementation, set difference, Kleene star, reverse) for regular languages covered in lectures, labs and homework. Understand the proofs for these properties.
- Know how to prove new closure properties either through automata transformations or using previously established closure properties.

Non-regularity

- Ability to distinguish regular and non-regular languages
- Ability to prove languages to be non regular using the fooling set argument. Know how to prove lower bounds on the number of DFA states using the fooling set argument as well.