Preparation for the Final Exam

Reading: The exam will be based on the reading, lectures and supplementary handouts from the entire semester.

- **Partee, Chapter 1.** (Don’t worry about Russell’s paradox).
- **Sipser, Chapter 0:** Sections 0.1 and 0.2 except for 0.2.4 (Graphs) and 0.2.6 (Boolean Logic).
- **Sipser, Chapter 1:** Everything except the following constructions. You are still responsible for knowing that these conversions are always possible, and for understanding the consequences of this.
  1. The construction used in Theorem 1.12: Intersection and union of regular languages using DFAs. (“Cartesian product” construction).
  2. The construction used in Theorem 1.19: Converting an NFA to a DFA (“Power-set construction”).
  3. Anything to do with GNFA (pp. 70–76), used in Lemma 1.32 to convert an NFA to a regular expression. (“GNFA construction”).
  4. The case-by-case analysis of Pumping Lemma example 1.38. (See example 1.39 instead).

Note that you still need to know about the Cartesian product and power set of sets! (Chapter 0).

- **Sipser, Chapter 2.1:** All of it.

- **Pinker, Chapters 2 and 4,** up to page 105 only. You should be familiar enough with them to answer content questions.

- The following handouts explain or add to the material in Sipser:
  1. Mathematical Variables.
  2. Designing DFAs (page 3 in the Midterm Preparation handout).
  3. Pumping Lemma Roadmap.
  4. Gold’s Theorem.

- Anything we did in class but is not really explained in the book, in particular, how to convert regular expressions into CFGs.
Cheat sheet: You may bring with you any handwritten notes, diagrams etc. you can fit on one full notebook page (written on one side only). No photocopies are allowed. If you prepare your sheet carefully, the preparation process will be an effective study aid.

Content: Pretty much what we have been doing in class and in the homeworks. The exam will cover material from the entire semester, with more emphasis on the second half. (There’s no way to understand the material in the second half without command of the early stuff, anyway). In general, the amount of time we spent on something in class is a good predictor of its importance in the exam.

You should have an understanding of the relevance and implications of automata theory for linguistics. There will be a few general questions, which will require you to understand how the specific things we learned fit together. Then there will be some set theory problems, some questions related to formal definitions and concepts, some constructions of automata, regular expressions and grammars, and at least one pumping-lemma question. You will need to be very clear on the meaning of the terminology and the notation (your “cheat sheet” will help you cut down on memorizing, but it is no substitute for understanding). You will especially need to know how to design simple automata and grammars, and how to combine them using the various constructions we learned.

Preparation: Your best bet is to study your notes and the reading carefully, especially anything that you got wrong in the homework, or that didn’t make much sense the first time through. Test your understanding by re-working any past homeworks that you got wrong. If you have questions, call or email to ask me to explain, or set up an appointment.

Key Concepts and Topics

1 The big picture

Equivalence results about NFAs, DFAs, regular expressions. How CFGs can generate some non-regular languages.

What the various results we have seen tell us about language. Properties of English that make it non-regular (Pinker). Learnability and implications for the nature of language. (Gold’s theorem).

2 Set Theory

Sets, finite and infinite, and the notation for specifying them. (E.g., what is \{x \mid 5 \leq x \leq 10\}?). Set operations: union, intersection, complement, power set, Cartesian product. Element vs. subset, sets of sets, etc. Ordered pairs and tuples. The standard sets \mathcal{N}, \mathcal{Z}. 
**Functions:** Understand how to write and use function tables for one- and two-argument functions; domain and range; using functions.

**Strings:** $\varepsilon$; language, alphabet, length of a string.

3 **Finite State Automata**

**Languages:** Understand the meaning of languages given in set notation or in English; the difference between $\phi$ (or \{}\r{}\) vs. $\varepsilon$ vs. \{}\r{} vs. \{0\} vs. 0.

**FSA Basics:** start state, accept and reject states; formal description (as a 5-tuple); the transition function; requirements for a well-formed FSA (no missing or multiple arrows, labels, etc).

**Understanding FSAs:** Given an automaton, you should be able to tell: how it works, and exactly how it will work if it is fed some string of my choice; what kinds of strings it accepts and what it rejects. Designing FSAs to order (see Addendum).

**Negating a DFA:** How to make an automaton that recognizes the complement of some language for which you have (or can write) a DFA.

**NFA Basics:** formal description; differences between NFAs and DFAs; $\varepsilon$-transitions. Designing simple NFAs.

**NFA Constructions:** building NFAs from smaller NFAs, using the regular operations.

**Regular languages:** the regular operations: $\cup$, $\ast$, $\circ$; closure.

4 **More on Regular Languages**

**Regular Expressions.** Reading and constructing. Conversion of REs to NFAs.

**Regular Languages.** Equivalence of DFAs, NFAs, and REs.

**Non-Regular Languages.** The infamous pumping lemma: What it says, why it’s true, how to use it.

5 **Context-Free Grammars**

**Basics:** how they work, how to use them, how to write simple ones.

**Constructions:** How to make the union, concatenation and star of CFGs (the regular operations); how to convert a regular expression into a CFG. *(Not in the book: Refer to your notes).*

**Applications:** Using CFGs for linguistic grammars.