Review of material covered since midterm

Sections covered: Section 4.9, Chapters 9 and 10, 5.2–5.6, 6.1–6.4, 7.1–7.4

1 Section 4.9

unary primitive recursive functions, the total function $\phi(t, x)$ that gives the value of the tth unary primitive recursive function on input x, $\phi(x, x) + 1$ as an example of a computable function that is not primitive recursive

2 Chapter 9

finite automaton (dfa), transition function, regular language, state transition diagrams, nondeterministic finite automaton (ndfa), language accepted by an automaton, examples of automata and regular languages, closure under $\cup, \cap$, complement in A^*, $\cdot,^*$, Kleene's theorem, regular expressions, second version of Kleene's theorem, pumping lemma, Myhill-Nerode theorem

3 Chapter 10

context-free productions, context-free grammars, context-free languages, derivation of strings, kernel, positive context-free grammar, derivation trees, branching grammars, regular grammars, right-linear grammars, Chomsky normal form, Bar-Hillel's pumping lemma, closure under $\cup$, complement in A^*, but not $\cap$, when $R \cap L$ is context-free, erasure (Er_P), testing whether a context-free language contains a particular string, is empty, finite or infinite, unsolvability of determining whether the intersection of two context-free languages is empty, ambiguous, unambiguous grammars, unsolvability of determining whether a grammar is ambiguous, bracket languages, $PAR_n(A)$, Δ, the regular grammar constructed from a context-free grammar Γ by introducing brackets, Chomsky-Schützenberger representation theorem ($L = Er_P(R \cap PAR_n(T))$), Dyck languages, pushdown automata, $\gamma_i(u)$ function, balanced words, tape alphabet A, pushdown alphabet Ω,quintuple representation of transitions of a pushdown automaton, deterministic pushdown automaton, u-configurations, u-computations, language accepted by a pushdown automaton, atomic pushdown automaton, relationship between context-free languages and the languages of pushdown automata, compilers

4 5.2–5.6

string computation programming language $\mathcal{L}_n$, relation to $\mathcal{L}$, $n = 1$ is sufficient, Post-Turing programming language $\mathcal{T}$, simulation of $\mathcal{L}_n$ in $\mathcal{T}$, simulation of $\mathcal{T}$ in $\mathcal{L}$ (I did not go through the proof details)

5 6.1–6.4

deterministic and nondeterministic Turing machines, quadruple representation of transitions, strict computation, functions computable with a Turing machine, quintuple Turing machines, a quintuple Turing machine that simulates a given Turing machine, Post-Turning program to simulate a quintuple Turing machine, existence of a universal Turing machine, language of a Turing machine, halting problem for Turing machines

6 7.1–7.4

semi-Thue productions, rewrite rules, semi-Thue process, rewrite system, derivation of words, simulation of nondeterministic Turning machines by semi-Thue processes, the word problem, Post-Markov theorem (reduction of certain Turing machine halting problems to word problems), unsolvability of word problems, Post correspondence systems, Post correspondence problem, solution to a Post correspondence problem (the word that can be broken up in two ways), unsolvability of Post correspondence problems