CPSC 229, Fall 2008 — Final Exam Info

The final exam for this course will be given in our regular classroom at the time scheduled by the registrar: 1:30 PM on Thursday, December 18. The final counts for 20% of the total grade for the course. The exam will cover material from the entire course, with emphasis on what has been covered since the third test. Not every topic from earlier in the term will be on the test; topics that might be there are listed on this sheet.

You can expect a six-page test. The last page will be an essay question that asks you to discuss what you have learned about the nature of computation in this course. You should be prepared to write a full-page essay on this topic. As for the rest of the test, you can expect about two pages on new material (Section 4.5 and Chapter 5) and one page on material from each of the earlier tests. (This is very approximate.)

Here are some terms and ideas from the new material:

general grammar
derivations using a general grammar
the language generated by a general grammar
examples of non-context-free languages that can be generated by general grammars
constructing grammars to generate specific languages
Turing machine
definition of a Turing machine as a four-tuple $(Q, \Lambda, q_o, \delta)$
tape of a Turing machine
cell on a tape
the blank symbol
start state of a Turing machine
halt state
transition table for a Turing machine
transition diagram for a Turing machine
how a Turing machine computes
determining what a given Turing machine does on some given input
what it means for a Turing machine to accept a language
Turing-acceptable language
what it means for a Turing machine to decide a language
Turing-decidable language
representing a Turing machine as a string of data that can be written on a tape
how a Turing machine can simulate the computation of other Turing machines
recursively enumerable language (synonym for Turing acceptable)
recursive language (synonym for Turing decidable)
a language is Turing acceptable if and only if it is generated by some grammar
a language L is recursive if and only if both L and $\overline{L}$ are recursively enumerable
the list of standardized Turing machines $T_0, T_1, T_2, T_3, \ldots$
the language $K = \{a^n \mid T_n \text{ halts when run with input } a^n\}$

the language K is recursively enumerable but is not recursive
the language $\overline{K}$ is not recursively enumerable
the Halting Problem
computability and uncomputability

<u>Here are some terms and ideas from older material</u>:

translating English into logic and *vice versa*
propositional logic, with operators $\wedge$, $\vee$, $\neg$, $\rightarrow$ and $\leftrightarrow$
truth table
tautology
logical equivalence
converse and contrapositive of a proposition of the form $p \rightarrow q$
DeMorgan's laws: the negations of $p \wedge q$ and of $p \vee q$
the negation of $p \rightarrow q$
predicates and predicate logic
the universal and existential quantifiers, $\forall$ and $\exists$
negation of $\forall x(P(x))$ and $\exists x(P(x))$
basic facts about numbers (primes, divisibility, rational and irrational, the sets $\mathbb{N}$, $\mathbb{Q}$, $\mathbb{R}$)
proofs
logical deduction and formal proofs
proof by contradiction
sets and set notation
$a \in A$, $A \cap B$, $A \cup B$, $A \setminus B$, $\overline{A}$, $\mathcal{P}(A)$, $A \times B$
the empty set $\emptyset$
functions, $f\colon A \rightarrow B$
one-to-one correspondence
finite, countably infinite, and uncountable sets
alphabets, strings, and languages
the empty string ϵ
string operations: xy, x^n, x^R, $|x|$, $n_\sigma(x)$
language operations: LM, L^n, L^*, L^R, and set operations applied to languages
regular expressions and regular languages
finite-state automata; DFAs and NFAs
transition diagrams for DFAs and NFAs
the language accepted by a DFA or NFA
the equivalence of DFAs, NFAs, and regular expressions
the algorithm for converting an NFA to a DFA
finding a regular expression, DFA, or NFA for a given language, and *vice versa*
context-free grammars and context-free languages
production rules, start symbol, terminal symbols, and non-terminal symbols
derivation of a string; the relations $\Longrightarrow$ and $\Longrightarrow^*$
finding a CFG for a given language, and *vice versa*
common examples of languages of all the types we have learned about