The final exam for this course is on Thursday, December 16, from 7:00 to 10:00 PM. It counts for 20% of your overall grade for the course. It will be held in our regular classroom. You can expect the exam to be about six pages long. Most people will probably be able to complete it in less than two hours.

The exam will cover the entire course, with some extra emphasis on material that was covered since the third test. Since the third test, we have covered parts of Sections 4.3 through 4.6 and of Chapter 5. There is quite a lot of material in those sections, and we didn't cover it all. You are only responsible for the parts that were covered in class.

Most of the questions on the exam will be similar to those on the tests, but you can expect some sort of summary essay questions. You should review the information sheets for the three in-class tests. You can find copies of them on the course web page: http://math.hws.edu/eck/cs229

I will have office hours from 11:00 to 12:00 and 1:30 to 3:00 on Monday, December 13, during Reading Period. I will also have office hours from 11:00 to 3:00 on Wednesday, December 15, the day before the exam, and from 1:00 to 4:00 on the day of the exam.

---

**Some terms and ideas covered since the third test:**

parse tree for a string generated by a context-free grammar

left derivations, right derivations, and their relationship to parse trees

ambiguous grammar

the idea of parsing (but **not** LL(1) or LR(1) parsing)

pushdown automata; the language $L(M)$ accepted by a pushdown automaton $M$

interpreting (but **not** creating) transition diagrams for pushdown automata

a language $L$ is context-free if and only if there is a pushdown automaton $M$ such that $L = L(M)$

general grammars; how they differ from context-free grammars

recursively enumerable language (one generated by a general grammar)

creating a general grammar for a given language

examples of languages that are recursively enumerable but not context free

Turing machine

tape of Turing machine

halt state

rule table for a Turing machine

transition diagram for a Turing machine

how a Turing machine computes

evidence that Turing machines are general-purpose computing devices

Turing-decidable language; also called a recursive language

Turing-acceptable language

a language is Turing-acceptable if and only if it is recursively enumerable

a language is recursive if and only if both it and its complement are recursively enumerable the standardized Turing machines $T_0$, $T_1$, $T_2$, ...

the language $K = \{n \in \mathbb{N} \mid T_n \text{ halts when run with input } n\}$

$K$ is recursively enumerable (Turing acceptable) but not recursive (Turing decidable)

$\overline{K}$ is not recursively enumerable

the Halting problem and its unsolvability

---

**Some of the salient terms and ideas from earlier in the course:**

propositional logic; the logical operators $\wedge$, $\vee$, $\neg$, $\rightarrow$, and $\leftrightarrow$

truth tables

tautology

logical equivalence

converse and contrapositive of $p \rightarrow q$

the definition of $p \rightarrow q$ as $(\neg p) \vee q$ and the negation rule $\neg(p \rightarrow q) \equiv p \wedge \neg q$ !!

Boolean algebra; DeMorgan's Laws and other basic laws of Boolean algebra

predicates and predicate logic; the quantifiers $\forall$ and $\exists$

domain of discourse of a predicate

translations from logic to English and *vice versa*

valid arguments and formal proof

Modus Ponens and Modus Tollens

proof; techniques for writing mathematical proofs

proof by contradiction

sets; elements of sets; set notation; the empty set

set operations: union, intersection, set difference

universal set; complement of a set (in the universal set)

subsets and the powers set $\mathcal{P}(A)$ of a set

infinite sets; countably vs. uncountably infinite; examples of countable and uncountable sets

the proof that there is no one-to-one correspondence from $A$ to $\mathcal{P}(A)$ !!

functions; the notation $f\colon A \rightarrow B$

alphabets, strings, and languages

any language is countable; the set of all possible languages over an alphabet $\Sigma$ is uncountable

operations on languages, including set operations plus concatenation and Kleene star

regular expressions and regular languages; the language generated by a regular expression

finding a regular expression for a langauge, and *vice versa*

finite state automata; DFAs and NFAs

transition diagrams

DFAs, NFAs, and regular expressions all define the same set of languages

the NFA-to-DFA conversion algorithm

nondeterminism

context-free grammars and context-free languages

finding a context-free grammar for a langauge, and *vice versa*

examples of various types of languages that are context-free but not regular