The final exam takes place during the officially scheduled period, Tuesday, December 15, from 1:30 PM to 3:00 PM. The exam is in our regular classroom, Coxe 8.

The test will be five pages long, and should take most people in the class less than one-and-a-half hours to complete. The last page will consist of a summary essay question that asks you to bring together various things that have been covered in the course and to discuss what you have learned about the nature of computation. You should be prepared to write about things like logic and logic circuits, different types of automata and languages, and ideas about computability and uncomputability. There might be some shorter essay-type questions on the first four pages, but most of the questions on the those pages will be similar to questions on previous tests and homework assignments. The exam covers material from the entire course, with some emphasis on material covered since the second test: general grammars, Turing machines, and computability.

The exam will ***not*** *include a pumping lemma proof or a proof by induction. There will, however, be some proofs, possibly including a formal proof of the validity of an argument. The will be no questions about logic circuits or about real computers (Java, BNF, regular expressions on computers), although you might want to bring those things up in your response to the final essay question. You will not be asked to work with pushdown automata.*

Here are some terms and ideas from the last segment of the course:

general grammars
derivations using a general grammar
how a general grammar generates a language
Turing machine
transition diagram for a Turing machine; table of rules for a Turing machine
how a Turing machine operates
how a Turing machine computes a function
evidence that Turing machines are general-purpose computing devices
Turing-acceptable language, also known as a recursively enumerable language
Turing-decidable language, also known as a recursive language
a language L is recursive if and only if both L and $\overline{L}$ are recursively enumerable
a language is recursively enumerable if and only if it is generated by a grammar
the language hierarchy: finite, regular, context-free, recursive, recursively enumerable
the Church-Turing Thesis
Turing machines (up to equivalence) can be assigned numbers: T_o, T_1, T_2, T_3, ...
the set K = $\{n \mid T_n$ halts when run with input $n\}$
the set K is recursively enumerable, but it is not recursive
the set $\overline{K}$ is not recursively enumerable
the Halting Problem and its computational unsolvability
the nature of computation

Here are some things from earlier in the course:

translations from logic to English, and from English to logic
propositional logic; the logical operators "and" ($\wedge$), "or" ($\vee$), and "not" ($\neg$)
truth table
logical equivalence ($\equiv$)
the conditional, or "implies," operator ($\rightarrow$)
the negation of $p \rightarrow q$ is $p \wedge \neg q$

the biconditional operator ($\leftrightarrow$)
tautology
some basic laws of Boolean algebra (double negation, DeMorgan's, commutative, etc.)
converse of an implication
contrapositive of an implication
predicate logic;   quantifiers, "for all" ($\forall$) and "there exists" ($\exists$)
negation of a statement that uses quantifiers (DeMorgan's laws for predicate logic)
the difference between $\forall x\, \exists y$ and $\exists y\, \forall x$
arguments, valid arguments, and deduction
formal proof of the validity of an argument
Modus Ponens and Modus Tollens
direct proof (to prove $P \to Q$, assume P is true, and prove Q)
proof by contradiction
sets
set notations: $\{a, b, c\}$, $\{1, 2, 3, \dots\}$, $\{x \mid P(x)\}$, $\{x \in A \mid P(x)\}$
the empty set, $\varnothing$ or $\{\,\}$
element of a set: $a \in A$
subset: $A \subseteq B$
union, intersection, and set difference: $A \cup B$, $A \cap B$, $A \smallsetminus B$
definition of set operations in terms of logical operators
power set of a set: $\mathscr{P}(A)$
universal set;   complement of a set (in a universal set): $\overline{A}$
ordered pair: (a, b);   cross product of sets: $A \times B$
one-to-one correspondence
infinite set
countably infinite set (in one-to-one correspondence with $\mathbb{N}$)
uncountable set (that is, uncountably infinite)
for any set A, there is no one-to-one correspondence between A and $\mathscr{P}(A)$
alphabet (finite, non-empty set of "symbols")
string over an alphabet Σ
string operations: length ($|w|$), concatenation (xy), reverse (w^R), w^n, $n_\sigma(w)$
language over an alphabet Σ (a subset of Σ^*)
the set of strings over Σ is countable; the set of languages over Σ is uncountable
set operations on languages: union, intersection, set difference, and complement
concatenation (LM), power (L^n), and Kleene star (L^*) of languages
regular expression over an alphabet Σ
regular language; the language $L(r)$ generated by a regular expression r
DFA (Deterministic Finite Automaton)
transition diagram of a DFA
definition of a DFA as a list of five things, $(Q, \Sigma, q_o, \delta, F)$ — and what each thing means
what it means for a DFA to accept a language
nondeterminism;   NFA (Non-deterministic Finite Automaton)
what it means for an NFA to accept a string
the language, $L(M)$, accepted by a DFA or NFA
algorithm for converting an NFA to an equivalent DFA
algorithm for converting a regular expression to an equivalent NFA
DFAs, NFAs, and regular expressions all define the same class of languages
CFGs (Context-Free Grammars); context-free language
definition of a CFG as a 4-tuple $G = (V, \Sigma, P, S)$
derivation (of a string from the start symbol of a CFG)
parse tree for a string in a context-free language
left derivation for a string in a context-free language

Here are some sample questions, taken from old final exams:

*Note: This is **not** meant as a comprehensive review!*

1. Use a *truth table* to show that $(p \rightarrow q) \wedge ((\neg p) \rightarrow q)$ is logically equivalent to q. (State what it is about your table that proves logical equivalence.)

2. Simplify each of the following logical expressions, so that the logical `NOT` operation applies only to simple terms:

a) $\neg(p \vee q \vee (\neg r))$

b) $\neg\big(\forall x\, \exists y\, \big(L(x,y) \wedge \forall z(\neg L(x,z))\big)\big)$

c) $\neg\big(\forall x\, (P(x) \rightarrow (\exists y\, (R(y) \wedge Q(x,y))))\big)$

3. English and logic. . .

a) State the negation of the statement, "If I do well in CS 229, then everyone will think I'm cool."

b) Translate the following sentence in to predicate logic, capturing as much of the meaning as possible: "Every habitable planet is orbited by a moon."

c) The statement "Everyone fears something" is ambiguous since it could mean either $\forall x\, \exists y\, F(x,y)$ or $\exists y\, \forall x\, F(x,y)$, where the meaning of the predicate $F(x,y)$ is "x fears y." Carefully explain the difference.

4. Translate each of the following English sentences into predicate logic, using the following predicates: $M(x)$ means "x is a movie star"; $H(x)$ means "x is a house"; $B(x)$ means "x is big"; and $P(x,y)$ means "x owns y".

a) All houses are big.

b) Someone owns a house.

c) Every movie star owns a big house.

5. Translate the following argument into logic, and either give a formal proof that the argument is valid or explain why it is not valid: "If Jack has a day off from work, it's raining. If the sun is shining, then it's not raining. Jack has a day off from work, so the sun is not shining."

6. Give a formal proof of the validity of the following argument. (Don't forget to give a justification for each step in the proof. For full credit, you should use Modus Ponens and Modus Tollens in the justification, when appropriate.)

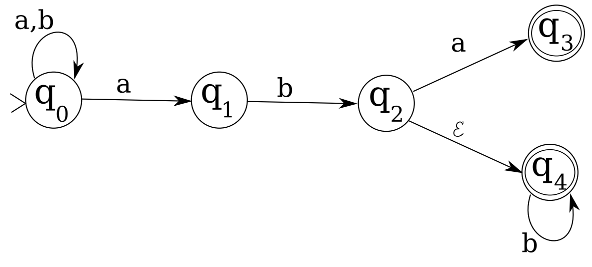
$$\begin{array}{l} t \rightarrow p \\ q \rightarrow s \\ p \rightarrow (q \vee r) \\ t \\ \neg s \\ \hline \therefore r \end{array}$$

7. Suppose that x is an irrational number and r is a rational number. Show that $x + r$ is an irrational number. Use a proof by contradiction.

8. Give an example of a set A such that $A \cap \mathscr{P}(A) \neq \emptyset$. Show both A and $\mathscr{P}(A)$. (Recall that $\mathscr{P}(A)$ is the power set of A.)

9. Draw a DFA that accepts the language $L = \{w \in \{a,b\}^* \mid \text{every } a \text{ in } w \text{ is immediately followed by } bb\}$

10. Use the NFA-to-DFA conversion algorithm, in which states in the DFA correspond to sets of states in the NFA, to construct a DFA that accepts the same language as the following NFA:



11. Identify the language generated by the regular expression $(ab|ba|aa|bb)^*(a|b)$. Explain your reasoning.

12. Find a context-free grammar for the language $\{a^nb^nc^k \,|\, n,k \in \mathbb{N}\}$

13. **Two** of the following languages are context free. Give context-free grammars for the languages that are context free. You do not have to explain your grammars.

$$\{a^nb^ma^nb^m \,|\, n,m \in \mathbb{N}\} \qquad \{a^nb^ma^mb^n \,|\, n,m \in \mathbb{N}\} \qquad \{a^nba^mba^{n+m} \,|\, n,m \in \mathbb{N}\}$$

14. **Prove** the following statement: Suppose that A, B, and C are sets and that $A \subseteq B$ and $A \subseteq C$. Then $A \subseteq B \cap C$.

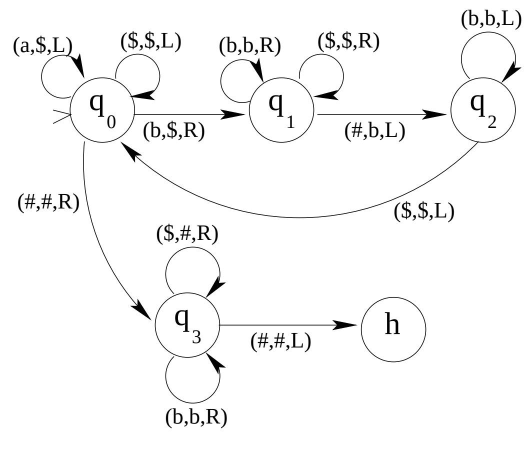
15. Consider the general grammar shown at the right, where S is the start symbol and the non-terminal symbols are a, b, and c.

$$\begin{array}{ll} S \longrightarrow SA & CA \longrightarrow AC \\ S \longrightarrow SAB & BC \longrightarrow CB \\ S \longrightarrow SABC & CB \longrightarrow BC \\ S \longrightarrow \varepsilon & A \longrightarrow a \\ AB \longrightarrow BA & B \longrightarrow b \\ BA \longrightarrow AB & C \longrightarrow c \\ AC \longrightarrow CA & \end{array}$$

a) Find a derivation for the string *baac* using this grammar.

b) Find the language that is generated by this grammar. Explain your answer.

16. The Turing Machine that is shown at the right is designed to process strings over the alphabet $\{a, b\}$. Assume that we always start this Turing Machine on the **rightmost** character of its input string.



a) What does this machine do when it is started on an empty string?

b) What does this machine output when it is started with input *aba*?

c) Describe the computation and output of this machine when it is started with any input string $w \in \{a,b\}^*$.

17. Suppose that L is a language over an alphabet Σ. Explain what it means for L to be *Turing acceptable* and what it means for L to be *Turing decidable*. Explain why every Turing decidable language is also Turing acceptable.

18. We defined the set $K = \{a^n \,|\, T_n \text{ halts on input } a^n\}$, and used K as an example of a set that is recursively enumerable but not recursive. What is T_n in this definition? What does it mean to say that K is recursively enumerable? What does it mean to say that K is not recursive? (Answers can be brief.)