University of Nevada, Las Vegas
Computer Science 456/656 Spring 2006
Review for Final Exam

This version `Mon May 8 12:18:23 PDT 2006` **I may add just a few more questions if I realize that I omitted a topic.**

Disclaimer: This "practice final" is much longer than the actual exam will be. To save paper, I did not leave enough room to work the problems, as I will on the actual examination.

1. True or False. If the question is currently open, write "O" or "Open."

   (a) _____ Every subset of a regular language is regular.

   (b) _____ The intersection of any context-free language with any context-free language is context-free.

   (c) _____ The complement of every recursive language is recursive.

   (d) _____ The complement of every recursively enumerable language is recursively enumerable.

   (e) _____ Every language which is generated by an unrestricted grammar is recursively enumerable.

   (f) _____ The question of whether two context-free grammars generate the same language is undecidable.

   (g) _____ There exists some proposition which is true but which has no proof.

   (h) _____ The set of all binary numerals for prime numbers is in the class $\mathcal{P}$.

   (i) _____ If L_1 reduces to L_2 in polynomial time, and if L_2 is $\mathcal{NP}$, and if L_1 is $\mathcal{NP}$-complete, then L_2 must be $\mathcal{NP}$-complete.

   (j) _____ Given any context-free grammar G and any string $w \in L(G)$, there is always a unique leftmost derivation of w using G.

   (k) _____ For any deterministic finite automaton, there is always a unique minimal non-deterministic finite automaton equivalent to it.

   (l) _____ Using multi-processors and other advanced technology, it is possible to design a machine which decides the halting problem.

   (m) _____ The question of whether two regular expressions are equivalent is decidable.

   (n) _____ The intersection of any context-free language with any regular language is context-free.

   (o) _____ Let $L = \{\langle M \rangle \mid M \text{ halts with no input}\}$. Then L is recursively enumerable.

   (p) _____ The complement of every context-free language is context-free.

   (q) _____ No language which has an ambiguous context-free grammar can be accepted by a DPDA.

(r) _____ The union of any two context-free languages is context-free.

(s) _____ The question of whether a given Turing Machine halts with empty input is decidable.

(t) _____ The class of languages accepted by non-deterministic finite automata is the same as the class of languages accepted by deterministic finite automata.

(u) _____ The intersection of any two regular languages is regular.

(v) _____ The intersection of any two context-free languages is context-free.

(w) _____ If L_1 reduces to L_2 in polynomial time, and if L_2 is $\mathcal{NP}$, then L_1 must be $\mathcal{NP}$.

(x) _____ Let $F(0) = 1$, and let $F(n) = 2^{F(n-1)}$ for $n > 0$. Then F is Turing-computable.

(y) _____ Every language which is accepted by some non-deterministic machine is accepted by some deterministic machine.

(a) _____ The language of all regular expressions over the binary alphabet is a regular language.

(b) _____ Let π be the ratio of the circumference of a circle to its diameter. (That's the usual meaning of π you learned in second grade.) The problem of whether the n^{th} digit of π, for a given n, is equal to a given digit is decidable.

(c) _____ There cannot exist any computer program that can decide whether any two C++ programs are equivalent.

(d) _____ An undecidable language is necessarily $\mathcal{NP}$-complete.

(e) _____ Every context-free language is in the class $\mathcal{P}$-TIME.

(f) _____ Every function that can be mathematically defined is Turing computable.

(g) _____ The language of all binary strings which are the binary numerals for multiples of 23 is regular.

(h) _____ The language of all binary strings which are the binary numerals for prime numbers is context-free.

(i) _____ Every bounded function from integers to integers is Turing-computable. (We say that f is *bounded* if there is some B such that $|f(n)| \leq B$ for all n.)

(j) _____ The language of all palindromes over $\{0, 1\}$ is inherently ambiguous.

(k) _____ Every context-free grammar can be parsed by some deterministic top-down parser.

(l) _____ Every context-free grammar can be parsed by some non-deterministic top-down parser.

(m) _____ Every context-free grammar can be parsed by some deterministic bottom-up parser.

(n) _____ Every context-free grammar can be parsed by some non-deterministic bottom-up parser.

(o) _____ Commercially available parsers cannot use the LALR technique, since most modern programming languages are not context-free.

(p) _____ The boolean satisfiability problem is undecidable.

(q) _____ There is a parallel processor machine which can solve the boolean circuit problem in poly-logarithmic time.

(r) _____ The set of all binary numerals for prime numbers is in the class $\mathcal{P}$-TIME.

(s) _____ If RSA coding is insecure, then $\mathcal{P} = \mathcal{NP}$.

2. Every context-free language is accepted by some _____.

3. If there is an easy reduction from L_1 to L_2, then _____ is at least as hard as _____.

4. For each language given, write "R" if the language is recursive, write "RE not R" if the language is recursively enumerable but not recursive, and write "not RE" if the language is not recursively enumerable.

   (a) _____ The language consisting of all Pascal programs p such that p halts if given p as its input file.

   (b) _____ The language of all encodings of Turing Machines which fail to halt for at least one possible input string.

   (c) _____ The 0-1 Traveling Salesman Problem.

   (d) _____ The diagonal language.

   (e) _____ The universal language.

   (f) _____ L_{sat}, the set of satisfiable boolean expressions.

5. For each question, give one example of a language, either by using the standard name of the language, or describing it in very few words. No proofs are required.

   (a) Give an example of an infinite language that is regular.

   (b) Give an example of a context-free language that is not regular.

   (c) Give an example of a language in the class $\mathcal{P}$ that is not context-free.

   (d) Give an example of a recursive language that is $\mathcal{NP}$-complete.

   (e) Give an example of a recursively enumerable language that is not recursive.

   (f) Give an example of a language that is not recursively enumerable.

6. Draw the state diagram for a minimal DFA that accepts the language desribed by the regular expression `(a+b)*ab`

7. Write a regular expression for the language accepted by the NFA shown if Figure 1.

8. Let L be the language of all binary numerals for positive integers equivalent to 2 modulo 3. Thus, for example, the binary numerals for 2, 5, 8, 11, 14, 17 ... are in L. We allow a binary numeral to have leading zeros; thus (for example) $001110 \in L$, since it is a binary numeral for 14. Draw a minimal DFA which accepts L.

9. Consider the context-free grammar G_1 with start symbol S and productions as follows:

   $S \to SS$

   $S \to aSb$

   $S \to \epsilon$

   (a) Show that G_1 is ambiguous.

   (b) Find an unambiguous context-free grammar G_2 that is equivalent to G_1.

10. Consider the context-free grammar with start symbol S and productions as follows:

    $S \to s$

    $S \to bLn$

    $S \to wS$

    $L \to \epsilon$

    $L \to SL$

    Write a leftmost derivation of the string $bswsbwsnn$

11. (a) What class of machines accepts the class of context free languages?

    (b) What class of machines accepts the class of regular languages?

    (c) What class of machines accepts the class of recursively enumerable languages?

    (d) What is the Church-Turing Thesis, and why is it important?

    (e) What is an unrestricted (same as general) grammar?

    (f) What does it mean to say that a language can be generated in *canonical order*? What is the class of languages that can be so generated?

    (g) What does it mean to say that machines M_1 and M_2 are *equivalent*?

    (h) What does it mean to say that a context free grammar is *ambiguous*?

12. Give definitions for each of the following. Assume that you are writing the definitions for a top mathematics graduate student who has never taken a course in automata theory, but who did sit in a lecture once where "alphabet," "string," and "language" were defined. The student also once read an article in which Turing machines were defined.

    (a) Give a definition of the language class $\mathcal{NP}$-TIME.

    (b) Give a definition of *$\mathcal{NP}$-complete language.*

(c) What does it mean to say that a language L is *decidable*?

(d) Give the definition of a *polynomial time reduction* of a language L_1 to another language L_2.

13. Let $\Sigma = \{0, 1\}$, the binary alphabet. We say a string w over Σ is *mostly positive* if w has more 1's than 0's. Let L be the set of mostly positive strings over Σ.

    Give a context-free grammar for L.

14. Let $\Sigma = \{0, 1\}$, the binary alphabet. Let L be the set of all strings w over Σ of the form $1^n 0^n 1^n$, where $n \geq 0$. Use the pumping lemma to prove that L is not a context-free language.

15. Draw a minimal DFA which accepts the language L over the binary alphabet $\Sigma = \{0, 1\}$ consisting of all strings in which every '001' is followed by '1'.

16. Construct a minimal DFA equivalent equivalent to the NFA shown in Figure 1.

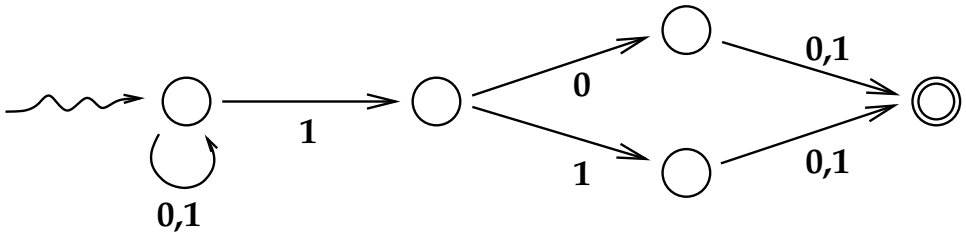


Figure 1: The NFA for Problems 7 and 16.

17. Consider the context-free grammar G, with start symbol S and productions as follows:

    $S \to s$

    $S \to bLn$

    $S \to iS$

    $S \to iSeS$

    $L \to \epsilon$

    $L \to LS$

    Prove that G is ambiguous by giving two different leftmost derivations for some string.

18. (a) What does it mean to say that a language L_1 reduces to a language L_2 in polynomial time?

    (b) What does it mean to say that a language L is decidable?

19. Let $\Sigma = \{0, 1\}$, the binary alphabet. Let $L \subseteq \Sigma^\star$ be recursively enumerable, but not recursive, and let M be a Turing machine that accepts L. If $w \in L$, let $T_M(w)$ be the number of steps of M in the valid computation with input w. For any string $w \in \Sigma^\star$, define

$$f(w) \quad = \quad \begin{cases} T_M(w) \text{ if } w \in L \\ 0 \text{ if } w \notin L \end{cases}$$

    Prove that f is not recursive.

    Classify each of the following problems or languages as just one of the following three:

    Known to be polynomial, known to be NP but not whether NP-complete, or known to be NP-complete.

(a) _____ Boolean satisfiability.

(b) _____ The 0-1 traveling salesman problem.

(c) _____ The knapsack problem where, for each instance, the size of each item is a positive integer that does not exceed the square of the number of items, and all the numbers are written in binary notation.

(d) _____ The clique problem.

(e) _____ The language generated by a given context-free grammar.

(f) _____ The set of all pairs of integers (n, m) such that n has a divisor which is at least m. (The input for an instance of this problem is the string consisting of the binary numeral for n, followed by a comma, followed by the binary numeral for m.)

20. Consider the context-free grammar, and the GOTO/ACTION table for an LALR parser for that grammar, given below. (Note that the grammar is ambiguous.) Show the sequence of configurations of the parser, the output sequence, and the parse tree determined by the parser, for each of the given input strings.

GRAMMAR

1. `S` → `S-S`
2. `S` → `S/S`
3. `S` → `(S)`
4. `S` → `x`
5. `S` → `y`

ACTION-GOTO TABLES

	x	y	-	/	(	)	**eof**	S
0.	s9	s10			s3			4
1.	s9	s10			s3			5
2.	s9	s10			s3			6
3.	s9	s10			s3			7
4.			s1	s2			**halt**	
5.			r1	s2		r1	r1	
6.			r2	r2		r2	r2	
7.			s1	s2		s8		
8.			r3	r3		r3	r3	
9.			r4	r4		r4	r4	
10.			r5	r5		r5	r5	

INPUT STRINGS

(a) `x-y-x`

(b) `x-(y-x)`

(c) `x/y-x`

(d) `x/(y-x)`

(e) `(x-y)/x`

(f) `x-y/x`

21. Consider the grammar given below. Design an LALR parser for that grammar.

For your convenience, stack states are given on the right hand side of the production.

(a) $S \rightarrow \epsilon$

(b) $S \rightarrow a_2 S_3 b_4 S_5$

22. For each language given, write "REG" if the language is regular, write "CF not "REG" if the language is context-free but not regular, write "R not CF" if the language is recursive but not context-free, write "RE not R" if the language is recursively enumerable but not recursive, and write "not RE" if the language is not recursively enumerable.

(a) _____ The set of all strings over the alphabet $\{a, b, c\}$ where are **not** of the form $a^n b^n c^n$.

(b) _____ The language of all encodings of Turing Machines which halt for at least one possible input string.

(c) _____ The 0-1 Traveling Salesman Problem.

(d) _____ The diagonal language.

23. Draw a minimal DFA which accepts the language L over the binary alphabet $\Sigma = \{a, b, c\}$ consisting of all strings which contain either aba or caa as a substring.

24. Consider the language L of all strings which would be acceptable as algebraic expressions involving variables and constants, where:

    - Every variable name is either x, y, or z.
    - Every constant is a natural number between 0 and 9
    - The only operators are addition, subtraction, and multiplication.
    - The symbol '$-$' is used only for subtraction. There is no negation.
    - There is no multiplication symbol. Multiplication is indicated by concatenating strings.
    - In multiplication of a constant by anything else, the constant must come first, and there can be at most one constant factor in any term.
    - Parentheses can be used.

    Here are some strings in the language $x(y + 2z)$, $x - 1 - z$, $4(zx - 2y)(x + z(x - 1))$.

    Give a grammar for L which is consistent with the usual semantics (as you learned in school) of such expressions.

25. Draw a minimal DFA which accepts the language of all strings over $\{a, b, c\}$ which do not contain the substring $abaa$.

26. Give an unambiguous context-free grammar for the language of all regular expressions over the alphabet $\{a, b\}$. Your grammar should be consistent with the semantics of regular expressions.

27. Let L be the set of all strings over $\{a, b\}$ which have equal numbers of a's and b's. Prove, by contradiction, that L is not regular, as follows. First, you know that $\{a^n b^n\}$ is not regular (don't prove it; you can just take it as given). Second, you know that $\{a^i b^j\}$ is regular. (How do you know this?) Third, there is a theorem about intersections of regular language. Finally, obtain a contradiction.

28. Indicate, using words and diagrams, how to prove that any 2-tape Turing machine can be emulated by a 1-tape Turing machine with a multiple track tape.

29. Draw a transition-diagram (called "state diagram" in our textbook) for a Turing Machine that accepts $\{a^n b^n | n \geq 0\}$. (Do **not** draw a transition diagram for a PDA.)

30. Which of the problems below are known to be $\mathcal{NP}$-complete?

    (a) _____ The traveling salesman problem.

    (b) _____ Boolean satisfiability.

    (c) _____ The halting problem.

    (d) _____ Primality.

    (e) _____ The context-free grammar equivalence problem.

(f) _____ The independent set problem.

31. State the pumping lemma for regular languages accurately. If you have all the right words bucontext-free in the wrong order, that means you truly do not understand the lemma, and you might get no partial credit at all.

32. State the pumping lemma for context-free languages accurately. If you have all the right words but in the wrong order, that means you truly do not understand the lemma, and you might get no partial credit at all.

33. What does the Turing machine in figure 2 do?

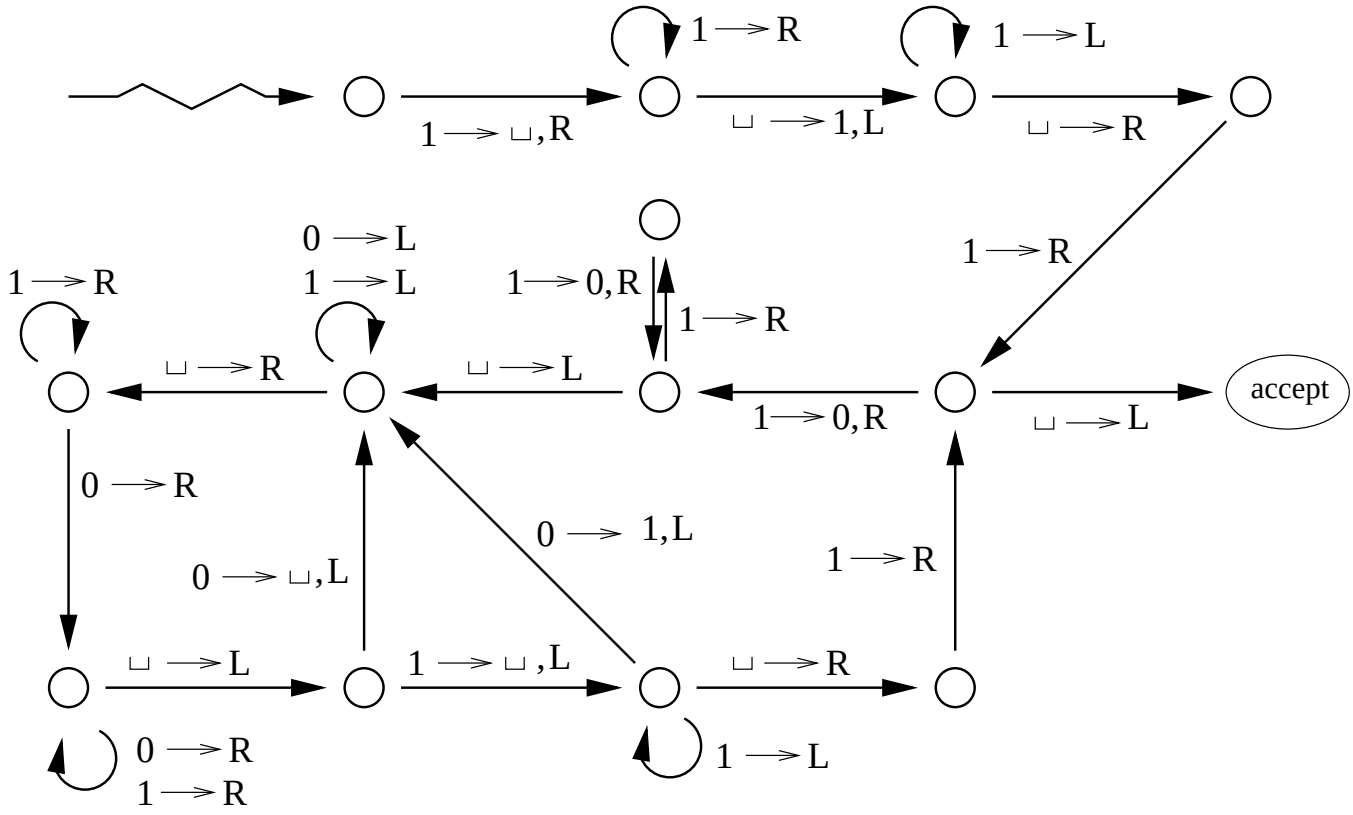


Figure 2: Turing machine for problem 33.

34. Let H be a function whose prototype is:

```
int H(int n,bool x);
```

such that

(a) $H(n, x) < n/2$,

(b) It takes $O(\log n)$ time to compute $H(n, x)$.

Let $\mathcal{B}$ be the set of unary numerals for "blue" numbers, where

(c) 1 is blue.

(d) If $n > 1$, then n is blue if and only if $H(n, 0)$ is positive and blue, or $H(n, 1)$ is positive and not blue

Prove that $\mathcal{B}$ is in $\mathcal{P}$.

Explain why there is no easy proof that $\mathcal{B}$ is in Nick's class, using only the above information.

35. We define *Collatz* numbers as follows.

(a) 1 is a Collatz number.

(b) If n is a Collatz number, then $2n$ is a Collatz number.

(c) If $n = 3m + 1$ is a Collatz number and m is an integer, then m is a Collatz number.

(d) There are no Collatz numbers except for those that can be proved to be Collatz by using the above rules.

Let C be the set of binary numerals for Collatz numbers. Prove that C is recursively enumerable.

36. Let $\Sigma = \{0, 1\}$, the binary alphabet. Let G be the unrestricted grammar whose terminal alphabet is Σ, whose start symbol is S, and whose productions are listed below:

$S \to LR$

$L \to L0Y$

$L \to LX$

$X1 \to 1X$

$X0 \to 0X$

$X0 \to 1Y$

$Y1 \to 0Y$

$YR \to R$

$L \to \epsilon$

$R \to \epsilon$

Describe $L(G)$ in English using no more than four words.

37. **Warning: this problem is very hard.** Every sliding block puzzle

`http://www.maa.org/editorial/mathgames/mathgames_12_13_04.html`

can be encoded as a string of bits. All sliding block puzzles that you read about are solvable, or else they wouldn't be proposed, but you could obviously make up a sliding block puzzle that has no solution.

Do you believe that the set of encodings of solvable sliding block puzzles is $\mathcal{NP}$? Explain your opinion.