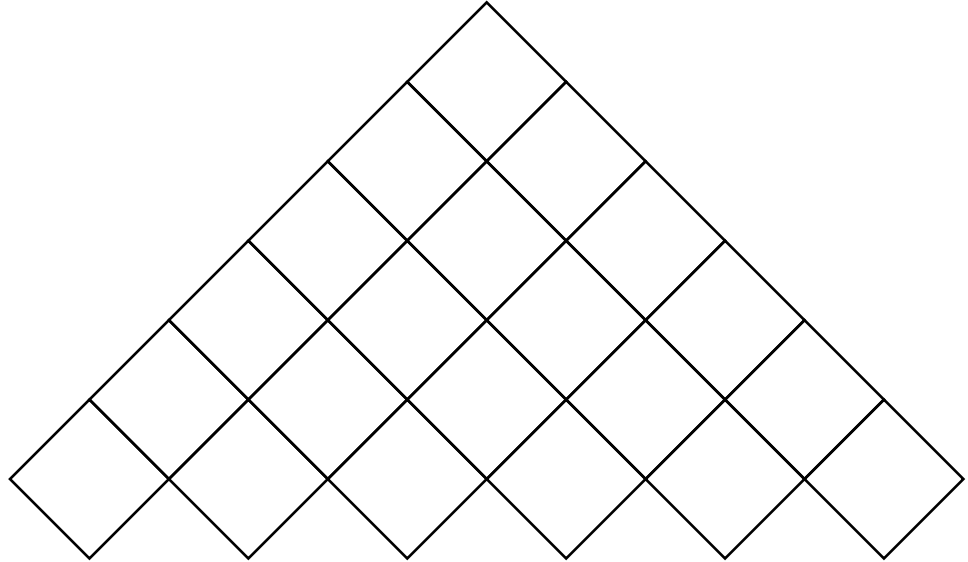


A *binary function* is defined to be a function  $F$  on binary strings such that, for each binary string  $w$ ,  $F(w)$  is a binary string. (Of course, the strings could be numerals.)

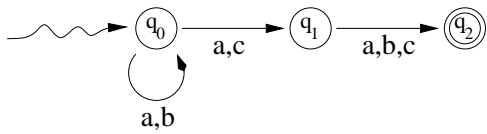
1. True or False. T = true, F = false, and O = open, meaning that the answer is not known to science at this time.
  - (i) Every subset of a regular language is regular.
  - (ii) ----- The complement of a CFL is always a CFL.
  - (iii) ----- The class of context-free languages is closed under union.
  - (iv) ----- The class of context-free languages is closed under intersection.
  - (v) ----- The language of binary numerals for multiples of 23 is regular.
  - (vi) ----- The set of binary numerals for prime numbers is in  $\mathcal{P}$ -TIME.
  - (vii) ----- Every language is countable.
  - (viii) ----- The set of languages over the binary alphabet is uncountable.
  - (ix) ----- The complement of any  $\mathcal{NP}$  language is  $\mathcal{NP}$ .
  - (x) ----- The complement of any decidable language is decidable.
  - (xi) ----- The complement of any undecidable language is undecidable.
  - (xii) -----  $\mathcal{P} = \mathcal{NP}$ .
  - (xiii) ----- The complement of any  $\mathcal{P}$ -SPACE language is  $\mathcal{P}$ -SPACE.
  - (xiv) ----- The complement of every recursive language is recursive.
  - (xv) ----- The complement of every recursively enumerable language is recursively enumerable.
  - (xvi) ----- If a language  $L$  is accepted by an NFA  $M$  with  $p$  states, then  $L$  has pumping length  $p$ .
  - (xvii) ----- Given any unambiguous context-free grammar  $G$  and any string  $w \in L(G)$ , there is always a unique leftmost derivation of  $w$  using  $G$ .
  - (xviii) ----- For any deterministic finite automaton, there is always a unique minimal non-deterministic finite automaton equivalent to it.
  - (xix) ----- Let  $\pi$  be the ratio of the circumference of a circle to its diameter. Then  $\pi$  is recursive.
  - (xx) ----- The Kleene closure of any recursive language is recursive.
  - (xxi) ----- If  $\mathcal{P} = \mathcal{NP}$ , then all one-way encoding systems are breakable in polynomial time.
  - (xxii) ----- A language  $L$  is in  $\mathcal{NP}$  if and only if there is a polynomial time reduction of  $L$  to SAT.
  - (xxiii) ----- The intersection of any context-free language with any regular language is context-free.

- (xxiv) ----- If  $L_1$  reduces to  $L_2$  in polynomial time, and if  $L_2$  is  $\mathcal{NP}$  and  $L_1$  is  $\mathcal{NP}$ -complete, then  $L_2$  is  $\mathcal{NP}$ -complete.
- (xxv) ----- Every language which is accepted by some non-deterministic machine is accepted by some deterministic machine.
- (xxvi) ----- The language of all regular expressions over  $\{a, b\}$  is a regular language.
- (xxvii) ----- The equivalence problem for C++ programs is decidable.
- (xxviii) ----- Every function that can be mathematically defined is recursive.
- (xxix) ----- If there is a recursive reduction of the halting problem to a language  $L$ , then  $L$  is undecidable.
- (xxx) ----- The set of rational numbers is countable.
- (xxxi) ----- The set of real numbers is countable.
- (xxxii) ----- The set of recursive real numbers is countable.
- (xxxiii) ----- There are countably many functions from integers to integers.
- (xxxiv) ----- There are countably many recursive functions from integers to integers.
- (xxxv) ----- The context-free grammar equivalence problem is  $\text{co-}\mathcal{RE}$ .
- (xxxvi) ----- Let  $L = \{(G_1, G_2)\} : G_1 \text{ and } G_2 \text{ are not equivalent}$ . Then  $L$  is recursively enumerable.
- (xxxvii) ----- If  $L$  is a recursively enumerable language, there must be a machine which enumerates  $L$  in canonical order.
- (xxxviii) ----- The set of all positive real numbers is countable.
- (xxxix) ----- The union of any two undecidable languages is undecidable.
- (xl) -----  $\text{co-}\mathcal{P}\text{-TIME} = \mathcal{P}\text{-TIME}$
2. Give an unambiguous CFG which generates the language of all palindromes over the alphabet  $\Sigma = \{a, b\}$ .
  3. Suppose  $L$  is a problem such that you can check any suggested solution in polynomial time. Which one of these statements is certainly true?
    - (a)  $L$  is  $\mathcal{P}$ .
    - (b)  $L$  is  $\mathcal{NP}$ .
    - (c)  $L$  is  $\mathcal{NP}$ -complete.
  4. Use the CYK algorithm to prove that  $iiwaea$  is generated by the following grammar.

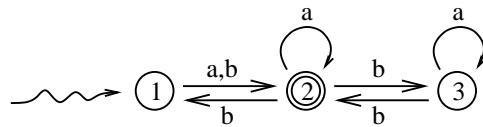
1.  $S \rightarrow a$
2.  $S \rightarrow WS$
3.  $W \rightarrow w$
4.  $S \rightarrow IS$
5.  $S \rightarrow AB$
6.  $A \rightarrow IS$
7.  $B \rightarrow ES$
8.  $E \rightarrow e$
9.  $I \rightarrow i$



5. Give a grammar, with at most 3 variables, for the language accepted by the following NFA.



6. Give a regular expression for the language accepted by the following NFA



7. Let  $L$  be the language consisting of all strings over  $\{a, b\}$  which have equal numbers of each symbol. Give a CFG for  $L$ .
8. Design a DPDA which accepts the language described in Problem 7.
9. Give a context-free language whose complement is not context-free.



16. Every language falls into exactly one of these classes.
- (a) Known to be  $\mathcal{P}$ -complete.
  - (b) Known to be  $\mathcal{NC}$ .
  - (c) Known to be  $\mathcal{NP}$ , but not known to be  $\mathcal{P}$ -TIME, but not known to be  $\mathcal{NP}$ -complete.
  - (d) Known to be  $\mathcal{P}$ -SPACE, but not known to be  $\mathcal{NP}$ .
  - (e) Known to be undecidable and  $\mathcal{RE}$  (recursively enumerable).
  - (f) Known to be decidable, but not known to be  $\mathcal{P}$ -SPACE.
  - (g) Known to be undecidable and co- $\mathcal{RE}$ .
  - (h) Not known to be either  $\mathcal{RE}$  or co- $\mathcal{RE}$ .

In which of the above list of classes is each of these languages or problems?

- (i) ----- The Dyck language.
- (ii) ----- The language of binary numerals for multiples of 3, where leading zeros are not allowed.
- (iii) ----- The Boolean circuit problem.
- (iv) ----- The set of all configurations of the game RUSH HOUR from which it is possible to win.
- (v) ----- The context-free grammar equivalence problem.
- (vi) ----- The halting problem.

17. Prove, by induction, that  $\sum_{i=0}^n 2^i = 2^{n+1} - 1$ . (For example,  $1 + 2 + 4 + 8 = 16 - 1$ .)

18. Review **tf1ans.pdf**, **tf2ans.pdf** and **tf3ans.pdf**.

19. Use the pumping lemma for regular languages to prove that  $L = \{a^n b^n\}$  is not regular.

20. Use the pumping lemma for context-free languages to prove that  $L = \{a^n b^n c^n\}$  is not context-free.

21. Prove that the complement of  $\{a^n b^n c^n\}$  is context-free. (Hint: A CF grammar for that language has lots of productions.)

22. Prove that the set of recursive real numbers is countable.
23. Prove the theorems on the handout CanonEnum.pdf.
24. Review the proof that the halting problem is undecidable.
25. What is the Church-Turing thesis?
26. Give a definition of the class  $\mathcal{NC}$ .
27. Give a definition of the class  $\mathcal{P}$ -complete.
28. Name a language which is known to be  $\mathcal{P}$ -complete.
29. Give a definition of the class  $\mathcal{P}$ -SPACE-complete.
30. Name a language which is known to be  $\mathcal{P}$ -SPACE-complete.
31. Let  $L$  be the Dyck language, but where each left parenthesis is written as  $a$  and every right parenthesis as  $b$ . (This makes grading easier, since if you write parentheses carelessly, they look alike.)

Here is an unambiguous CFG for  $L$ .

1.  $S \rightarrow a_2 S_3 b_4 S_5$
2.  $S \rightarrow \lambda$

(a) Fill in the action and goto tables for the grammar given above. I have started the tables by writing row 0 and row 4.

(b) Show the computation of the parser for the input string  $aabbab$ .

	$a$	$b$	$\$$	$S$
0	$s2$		$r2$	1
1				
2				
3				
4	$s2$	$r2$	$r2$	5
5				

32. The following CF grammar models an assignment statement. We allow just two identifiers,  $x$  and  $y$ , and two operators  $+$  and  $*$ . We have three grammar variables,  $S$  for assignment statement,  $I$  for identifier, and  $E$  for expression. We have the equal sign as a symbol. The start symbol is  $S$ .

1.  $S \rightarrow I_2 =_3 E_4$
2.  $I \rightarrow x_5$
3.  $I \rightarrow y_6$
4.  $E \rightarrow I_7$
5.  $E \rightarrow E +_8 E_9$
6.  $E \rightarrow E *_10 E_{11}$

(a) Sketch the parse tree of the string

$$x = y + x * y$$

	$x$	$y$	$+$	$*$	$=$	$\$$	$S$	$I$	$E$
0	$s5$	$s6$					1	2	
1						<b>halt</b>			
2					$s3$				
3	$s5$	$s6$						7	4
4			$s8$	$s10$		$r1$			
5			$r2$	$r2$	$r2$	$r2$			
6			$r3$	$r3$	$r3$	$r3$			
7			$r4$	$r4$	$r4$	$r4$			
8	$s5$	$s6$						7	9
9			$r5$	$s10$		$r5$			
10	$s5$	$s6$						7	11
11			$r6$	$r6$		$r6$			

(b) Identify the entries of the Action table which ensure that addition and multiplication are left associative and that multiplication has precedence over addition.

33. (a) Give a CNF grammar for the language  $L$  of problem 32.  
 (b) Use that grammar and the CYK algorithm to prove that  $aababb \in L$ .
34. Prove that  $\sum_{i=1}^n i^3 = \frac{n^2(n+1)^2}{4}$  for every positive integer  $n$ . (Hint: The binomial theorem states that  $(a+b)^2 = a^2 + 2ab + b^2$ , and  $(a+b)^3 = a^3 + 3a^2b + 3ab^2 + b^3$ .)
35. State the pumping lemma for context-free languages.
36. Give an example of a CFL that is not regular.
37. Give an example of a CFL that is not a DCFL.
38. Sketch the diagram of a DPDA which accepts the language of all strings over  $\{a, b\}$  which have twice as many  $a$ 's as  $b$ 's.
39. Read and understand [Handout/complexityIII.pdf](#)
40. Prove that every language which is enumerated in canonical order by some machine is decidable.
41. Prove that every decidable language is enumerated in canonical order by some machine.
42. Prove that every recursively enumerable language is accepted by some machine.

43. Prove that every language accepted by a machine is recursively enumerable.
44. Write a (pseudo-code) program which accepts HALT. (Hint: You can write it in no more the four lines.)
45. Prove that HALT is undecidable.
46.
  - (a) Read Handout/lalrhandout1.
  - (b) That handout contains 8 questions. Answers to questions 1, 2, 3, and 6. are given in the handout. Understand those questions and answers.
  - (c) Work questions 4, 5, 7, and 8.
  - (d) Using the grammar on page 1 of the handout, give two *different* parse trees of the string  $a*a+a$ , showing that the grammar is ambiguous. Which one of those parse trees is “correct,” *i.e.*, respects the standard precedence of operators?