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) **F** Every subset of a regular language is regular.

   (ii) **F** The complement of a CFL is always a CFL.

   (iii) **T** The class of context-free languages is closed under union.

   (iv) **F** The class of context-free languages is closed under intersection.

   (v) **T** The language of binary numerals for multiples of 23 is regular.

   (vi) **T** The set of binary numerals for prime numbers is in $\mathcal{P}$–TIME.

   (vii) **T** Every language is countable.

   (viii) **T** The set of languages over the binary alphabet is uncountable.

   (ix) **O** The complement of any $\mathcal{NP}$ language is $\mathcal{NP}$.

   (x) **T** The complement of any decidable language is decidable.

   (xi) **T** The complement of any undecidable language is undecidable.

   (xii) **O** $\mathcal{P} = \mathcal{NP}$.

   (xiii) **T** The complement of any $\mathcal{P}$–SPACE language is $\mathcal{P}$–SPACE.

   (xiv) **T** The complement of every recursive language is recursive.

   (xv) **F** The complement of every recursively enumerable language is recursively enumerable.

   (xvi) **T** If a language L is accepted by an NFA M with p states, then L has (regular) pumping length p.

   (xvii) **T** 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) **F** For any deterministic finite automaton, there is always a unique minimal non-deterministic finite automaton equivalent to it.

   (xix) **T** Let π be the ratio of the circumference of a circle to its diameter. Then π is recursive.

   (xx) **T** The Kleene closure of any recursive language is recursive.

   (xxi) **T** If $\mathcal{P} = \mathcal{NP}$, then all one-way encoding systems are breakable in polynomial time.

   (xxii) **T** A language L is in $\mathcal{NP}$ if and only if there is a polynomial time reduction of L to SAT.

   (xxiii) **T** The intersection of any context-free language with any regular language is context-free.

(xxiv) **T** 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) bf T Every language which is accepted by some non-deterministic machine is accepted by some deterministic machine.

(xxvi) **F** The language of all regular expressions over $\{a, b\}$ is a regular language.

(xxvii) **F** The equivalence problem for C++ programs is decidable.

(xxviii) **F** Every function that can be mathematically defined is recursive.

(xxix) **T** If there is a recursive reduction of the halting problem to a language L, then L is undecidable.

(xxx) **T** The set of rational numbers is countable.

(xxxi) **F** The set of real numbers is countable.

(xxxii) **T** The set of recursive real numbers is countable.

(xxxiii) **F** There are countably many functions from integers to integers.

(xxxiv) **T** There are countably many recursive functions from integers to integers.

(xxxv) **F** The context-free grammar equivalence problem is co-$\mathcal{RE}$.

(xxxvi) **T** Let $L = \{(G_1, G_2)\} : G_1$ and G_2 are not equivalent. Then L is recursively enumerable.

(xxxvii) **T** If L is a recursively enumerable language, there must be a machine which enumerates L in canonical order.

(xxxviii) **F** The set of all positive real numbers is countable.

(xxxix) **F** The union of any two undecidable languages is undecidable.

(xl) **T** co-$\mathcal{P}$–TIME = $\mathcal{P}$–TIME

2. Give an unambiguous CFG which generates the language of all palindromes over the alphabet $\Sigma = \{a, b\}$.

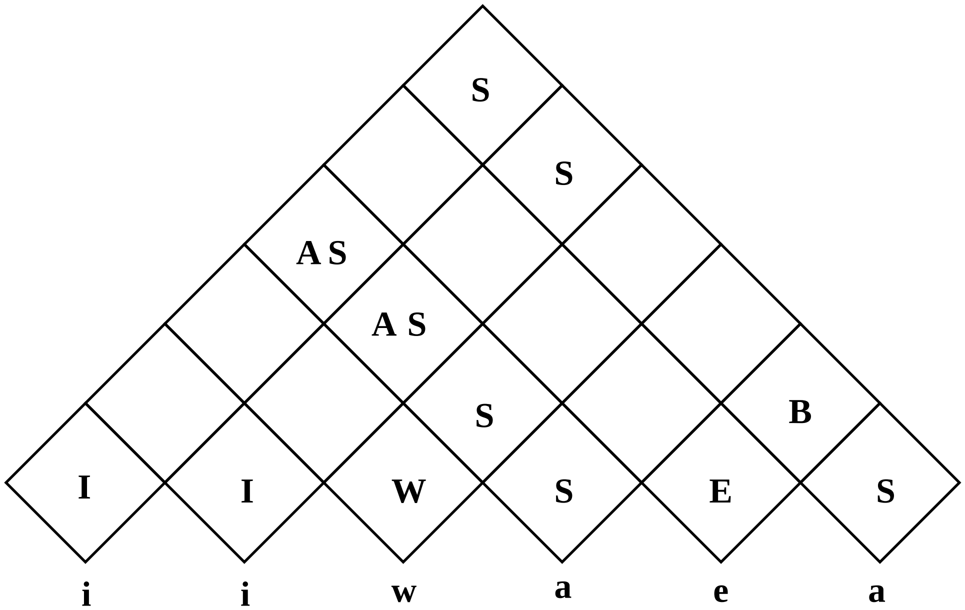
$S \to aSa$
$S \to bSb$
$S \to a$
$S \to b$
$S \to \lambda$

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?

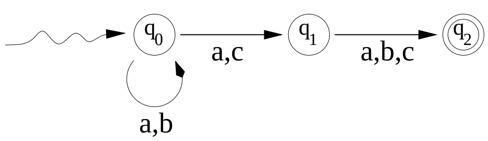
   (a) L is $\mathcal{P}$.
   (b) L is $\mathcal{NP}$.
   (c) L is $\mathcal{NP}$-complete. (b)

4. Use the CYK algorithm to prove that *iiwaea* is generated by the following grammar.

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

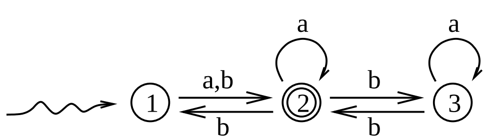


5. Give a grammar, with at most 3 variables, for the language accepted by the follwing NFA.



$S \to aS$
$S \to bS$
$S \to aA$
$S \to cA$
$B \to a$
$B \to b$
$B \to c$

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

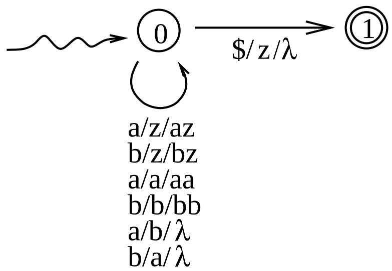


$(a + b)(b(a + b) + a + ba^*b)^*$

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.

$S \to aSb$
$S \to bSa$
$S \to SS$
$S \to \lambda$

8. Design a DPDA which accepts the language described in the previous Problem.



9. Give a context-free language whose complement is not context-free.

$L = \{a^n b^n c^n : n \geq 9\}$ is not context free. Let L' be the complement of L. Then L' is context-free, but its complement is L, which is not context-free.

10. State the pumping lemma for regular languages.

This definition is in a handout.

11. State the Church-Turing thesis. Why is it important?

Any computation that can be done by any machine can be done by some TM (Turing machine).

This is important because, in order to prove that something is uncomputable, it suffices to prove that no TM can compute it. Turing machines are simple, making such a proof easier.

12. Prove that a language is decidable if and only of it is enumerated by some machine in canonical order. (This is actually two theorems.)

These proofs are in a handout.

13. Prove that a language is $\mathcal{RE}$ if and only if it is enumerated by some machine. (This is actually two theorems.)

These proofs are in a handout.

14. Give a definition of a recursive real number. (There is more than one correct definition.)

Here are two definitions.

1. A real number x is recursive if and only if the set of fractions whose values are less than x is decidable.
2. A real number x is recursive if and only if there is a machine which computes the decimal expansion of x.

15. Which of these languages (problems) are **known** to be $\mathcal{NP}$-complete? If a language, or problem, is known to be $\mathcal{NP}$-complete, fill in the first circle. If it is either known not be be $\mathcal{NP}$-complete, or if whether it is $\mathcal{NP}$-complete is not known at this time, fill in the second circle.

⊗ ◯ Boolean satisfiability.
◯ ⊗ 2–SAT.
⊗ ◯ 3–SAT.
⊗ ◯ Subset sum problem.
⊗ ◯ Traveling salesman problem.
⊗ ◯ Dominating set problem.
◯ ⊗ C++ program equivalence.
⊗ ◯ Partition.
◯ ⊗ Regular language membership problem.
⊗ ◯ Block sorting.

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) **(b)** The Dyck language.

(ii) **(b)** The language of binary numerals for multiples of 3, where leading zeros are not allowed.

(iii) **(a)** The Boolean circuit problem.

(iv) **(d)** The set of all configurations of the game RUSH HOUR from which it is possible to win.

(v) **(g)** The context-free grammar equivalence problem.

(vi) **(e)** The halting problem.

17. State the pumping lemma for context-free languages. For any context-free language L,
there exists an integer p,‘
such that for any $w \in L$ of length at least p,
there exist strings u, v, x, y, z such that the following statements hold:

1. $uvxyz = w$
2. $|vxy| \leq p$
3. $|v| + |y| > 0$
4. for any integer $i \geq 0$, $uv^ixy^iz \in L$.

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

Proof: The statement is true for $n = 0$, since $2^0 = 2^2 - 1$.

Now, assume the inductive hypothesis, namely that the statement holds for n. Then, for $n+1$, we have

$$\begin{aligned}\sum_{i=0}^{n+1} 2^i &= \sum_{i=0}^{n} 2^i + 2^{n+1} \\ &= 2^{n+1} - 1 + 2^{n+1} \text{by the inductive hypothesis} \\ &= 2^{n+2} - 1\end{aligned}$$

∎

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

20. Use the pumping lemma for regular languages to prove that $L = \{a^n b^n\}$ is not regular. *Proof:* Suppose L is regular. Let p be the pumping length of L. Let $w = a^p b^p$, which is in L and has length at least p. Then the four statements of the pumping lemma hold. Since $xyz = w$ and $|xy| \leq n$, we know that xy has no symbol b. Thus, $y = a^k$ for some k. Since y is not the empty string, $k > 0$. By the fourth statement, letting $i = 0$, we have $xz \in L$ but $xz = a^{p-k} b^p \notin L$, contradiction. Thus L is not regular. ∎

21. Use the pumping lemma for context-free languages to prove that $L = \{a^n b^n c^n\}$ is not context-free. *Proof:* Suppose L is context-free. let p be the pumping length of L. Let $w = a^p b^p c^p \in L$, which has length at least p. Then there exist strings u, v, x, y, z such that the four concluding statements of the pumping lemma hold.

    1. $w = uvxyz$
    2. $|vxy| \leq p$
    3. v and y are not both empty
    4. For any $i \geq 0$ $uv^i xy^i z \in L$.

    By statement 2., either $\#_a(vxy) = 0$ or $\#_c(vxy) = 0$. Without loss of generality, $\#_c(vxy) = 0$. By statement 4, letting $i = 0$, we have $w' = vxz \in L$. Then, by statment 3, $\#_a(w') + \#_b(w') < 2p$, while $\#_c(w') = p$. Thus $w' \not{in} L$, contradiction. ∎

22. Prove that the complement of $L = \{a^n b^n c^n\}$ is context-free. (Hint: A CF grammar for that language has lots of productions.) Let L' be the complement of L. Each string w of L' fails to be in L for at least one of the following reasons:

    1. w contains the substring ba
    2. w contains the substring ca
    3. w contains the substring cb
    4. $w = a^i b^j c^k$ for $i \neq j$
    5. $w = a^i b^j c^k$ for $j \neq k$

    For $i = 1, 2, 3, 4, 5$, let L_i be the set of strings which satisfy statement i above. L_1 is regular since it has the regular expression $(a + b + c)ba(a + b + c)$, hence is context-free. Similarly, L_2 and L_3 are regular. L_4 is generated by the CF grammar

    $S \to A$
    $S \to B$

$A \to aAb$
$A \to aA$
$A \to a$
$A \to Ac$
$B \to aBb$
$B \to Bb$
$B \to b$
$B \to Bc$

Similarly, L_5 is context-free. The union of context-free languages is context free, hence $L' = L_1 + L_2 + L_3 + L_4 + L_5$ is context free.

23. Prove that the set of recursive real numbers is countable. *Proof:* The decimal expansion of any recursive real number is written by some C++ program. There are only countably many C++ programs, hence only countably many recursive real numbers. $\blacksquare$

24. Prove the theorems on the handout CanonEnum.pdf.

25. Review the proof that the halting problem is undecidable.

26. What is the Church-Turing thesis?

27. Give a definition of the class $\mathcal{NC}$. A language L is $\mathcal{NC}$ if there is a number k such that the membership problem for L is decided in $O(\log^k)$ time by $O(n^k)$ parallel processors, where n is the length of the input string.

28. Give a definition of the class $\mathcal{P}$–complete. A language L is $\mathcal{P}$ complete if L is $\mathcal{P}$–TIME, and every $\mathcal{NC}$ language is reduced to L by an $\mathcal{NC}$ reduction.

29. Name a language which is known to be $\mathcal{P}$–complete. CVP, the circuit value problem.

30. Give a definition of the class $\mathcal{P}$–SPACE–complete. L is $\mathcal{P}$–SPACE complete if L is $\mathcal{P}$–SPACE and every $\mathcal{P}$–SPACE language is reduced to L in polynomial time.

31. Name a language which is known to be $\mathcal{P}$–SPACE–complete. All configurations of the game RUSH HOUR from which it's possible to win.

32. 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. $S \to a_2 S_3 b_4 S_5$
    2. $S \to \lambda$

    (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			*halt*	
2	$s2$	$r2$		
3	$r2$	$s4$		
4	$s2$	$r2$	$r2$	5
5		$r1$	$r1$	

$\$_0$	$aabbab\$$		
$\$_0a_2$	$abbab\$$		$s2$
$\$_0a_2a_2$	$bbab\$$		$s2$
$\$_0a_2a_2S_3$	$bbab\$$	2	$r2$
$\$_0a_2a_2S_3b_4$	$bab\$$	2	$s4$
$\$_0a_2a_2S_3b_4S_5$	$bab\$$	22	$r2$
$\$_0a_2S_3$	$bab\$$	221	$r1$
$\$_0a_2S_3b_4$	$ab\$$	221	$s4$
$\$_0a_2S_3b_4a_2$	$b\$$	221	$s2$
$\$_0a_2S_3b_4a_2S_3$	$b\$$	2212	$r2$
$\$_0a_2S_3b_4a_2S_3b_4$	\$	2212	$b4$
$\$_0a_2S_3b_4a_2S_3b_4S_5$	\$	22122	$r2$
$\$_0a_2S_3b_4S_5$	\$	221221	$r1$
$\$_0S_1$	\$	2212211	$r1$

halt

33. 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. $S \to I_2 =_3 E_4$
2. $I \to x_5$
3. $I \to y_6$
4. $E \to I_7$
5. $E \to E +_8 E_9$
6. $E \to E *_{10} E_{11}$

(a) Sketch the parse tree of the string

$$x = y + x * y$$

	x	y	$+$	$*$	$=$	\$	S	I	E
0	$s5$	$s6$					1	2	
1						**halt**			
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 multiplicaion are left associative and that multiplication has precedence over addition.

34. (a) Give a CNF grammar for the language L of problem. (Dyck)

(b) Use that grammar and the CYK algorithm to prove that $aababb \in L$.

35. Prove that $\sum_{i=1}^{n} i^3 = \dfrac{n^2(n+1)^2}{4}$ for every positive integer n. (Hint: The binomial theorem states that $(a+b)^2 = a^2 + 2ab + b^2$, and $(a+b)^3 = a^3 + 3a^2b + 3ab^2 + b^3$.)

36. State the pumping lemma for context-free languages.

    See above.

37. Give an example of a CFL that is not regular.

    $\{a^n b^n\}$

38. Give an example of a CFL that is not a DCFL.

    Palindromes over $\{a, b\}$.

39. 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. **This is hard. I have it in some file, but I certainly won't give it in the third exam.**

    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.

    That is in a handout.

42. Prove that every recursively enumerable language is accepted by some machine.

    That is in a handout.

43. Prove that every language accepted by a machine is recursively enumerable.

    That is in a handout.

44. Write a (pseudo-code) program which accepts HALT. (Hint: You can write it in no more the four lines.)

    $P(M, w)$

    Run M with input w. If it ever halts, accept (M, w).

45. Prove that HALT is undecidable.

    That is in a handout.

    46. tem Read Handout/lalrhandout1.

        That handout contains 8 questions. Answers to questions 1, 2, 3, and 6. are given in the handout. Understand those questions and answers.

    (b) Work questions 4, 5, 7, and 8.

(c) 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?