



4. Prove that any language which is enumerated by some machine is accepted by some other machine.

5. I have repeatedly stated in class that no language that has parentheses can be regular. For that to be true, there must be parenthetical strings of arbitrary nesting depth. (If you don't know what nesting depth is, look it up.)

Some programming languages have limitations on nesting depth. For example, I have read that ABAP has maximum nesting depth of 256. (Who would ever want to go that far!)

The Dyck language is generated by the following context-free grammar. (As usual, to make grading easier, I use  $a$  and  $b$  for left and right parentheses.)

1.  $S \rightarrow aSbS$

2.  $S \rightarrow \lambda$

(a) Use the pumping lemma to prove that the Dyck language is not regular.

(b) Let  $D$  be any positive integer. Let  $L$  be the language consisting of all members of the Dyck language whose nesting depth does not exceed  $D$ . Prove that  $L$  is regular.

6. We know that context-free languages are exactly those which are accepted by push-down automata. We now define a new class of machines, which we call “limited push-down automata.” An LPDA is exactly the same as a PDA, but with the restriction that the stack is never allowed to be larger than some given constant. What is the class of languages accepted by limited push-down automata? Think!
7. Prove that every context-sensitive language is decidable. The way to do this is to start with an arbitrary context-sensitive grammar, using the definition I gave in class (that’s not the only definition) namely that the right side of any production must be at least as long as the left side, and then design a program which decides whether any given string is generated by that grammar. (If the string has length  $n$ , the running time of your program could be very long, maybe an exponentially bounded function of  $n$ ?)

1. True/False/Open

- (i) ----- A minimal DFA equivalent to a given NFA with 3 states can have no more than 7 states which are not dead. ----- Every subset of a regular language is regular.
- (ii) ----- Let  $L$  be the language over  $\{a, b, c\}$  consisting of all strings which have more  $a$ ’s than  $b$ ’s and more  $b$ ’s than  $c$ ’s. There is some PDA that accepts  $L$ .
- (iii) ----- The intersection of any regular language with any context-free language is context-free.
- (iv) ----- The intersection of any two context-free languages is context-free.
- (v) ----- Every language accepted by a non-deterministic machine is accepted by some deterministic machine.
- (vi) ----- The language  $\{a^n b^n c^n : n \text{ is prime}\}$  is decidable.

- (vii) ----- If a language  $L$  is EXP-SPACE complete,  $L$  must be decidable.
- (viii) -----  $\mathcal{NC} = \mathcal{P}$ .
- (ix) -----  $\mathcal{P} = \mathcal{NP}$ .
- (x) ----- EXP-TIME =  $\mathcal{P}$ -TIME.
- (xi) ----- The Boolean Circuit Problem is in  $\mathcal{NC}$ .
- (xii) ----- 3-SAT is  $\mathcal{P}$ -TIME.
- (xiii) ----- Addition of binary numerals is in  $\mathcal{NC}$ .
- (xiv) ----- Every language generated by a general grammar is recursive.
- (xv) ----- Every language generated by a general grammar is recursively enumerable.
- (xvi) ----- The problem of whether two given context-free grammars are equivalent is co- $\mathcal{RE}$ .
- (xvii) ----- The language of all fractions (using base 10 numeration) whose values are less than  $\pi$  is decidable. (Recall that a fraction is a string consisting of two numerals separated by  $"/$ ).
- (xviii) ----- If  $L$  is  $\mathcal{RE}$  and co- $\mathcal{RE}$ , then  $L$  must be decidable.
- (xix) ----- For every real number  $x$ , there exists a machine that runs forever and outputs the string of decimal digits of  $x$ .
- (xx) ----- There exists a mathematical proposition that can be neither proved nor disproved.
- (xxi) ----- If a Boolean expression is satisfiable, there is a  $\mathcal{P}$ -TIME proof that it's satisfiable.
- (xxii) ----- Every subset of any enumerable set is enumerable.
- (xxiii) ----- Every subset of any recursively enumerable set is recursively enumerable.
- (xxiv) ----- The binary numeral factorization problem is co- $\mathcal{NP}$ .

2. Every language, or problem, falls into exactly one of these categories. For each of the languages, write a letter indicating the correct category.

- A** Known to be  $\mathcal{NC}$ .
- B** Known to be  $\mathcal{P}$ -TIME, but not known to be  $\mathcal{NC}$ .
- C** Known to be  $\mathcal{NP}$ , but not known to be  $\mathcal{P}$ -TIME and not known to be  $\mathcal{NP}$ -complete.
- D** Known to be  $\mathcal{NP}$ -complete.
- E** Known to be  $\mathcal{P}$ -SPACE but not known to be  $\mathcal{NP}$
- F** Known to be EXP-TIME but not known to be  $\mathcal{P}$ -SPACE.
- G** Known to be EXP-SPACE but not known to be EXP-TIME.
- H** Known to be decidable, but not known to be EXP-SPACE.
- K**  $\mathcal{RE}$  but not decidable.
- L** co- $\mathcal{RE}$  but not decidable.

M Neither  $\mathcal{RE}$  nor  $\text{co-}\mathcal{RE}$ .

- (a) ----- All C++ programs which halt with no input.
- (b) ----- All base 10 numerals for perfect squares.
- (c) ----- All context-free grammars that generate the Dyck language.
- (d) ----- All configurations of RUSH HOUR from which it's possible to win.
- (e) ----- All satisfiable Boolean expressions.
- (f) ----- All binary numerals for composite integers. (Composite means not prime.)

3. Give a definition of each term.

- (a) Accept. (That is, what does it mean for a machine to accept a language.)
  
- (b) Decide. (That is, what does it mean for a machine to decide a language.)
  
- (c) Canonical order of a language  $L$ .

4. Find a minimal DFA equivalent to the NFA shown in Figure 1.

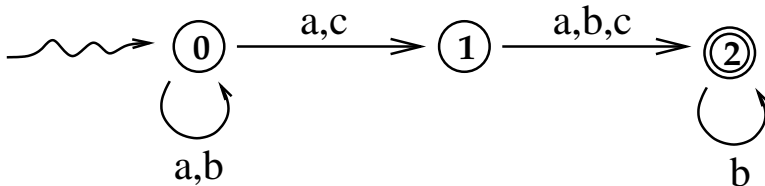


Figure 1: NFA for problems 4 and 5

5. Give a regular grammar with **no more than three variables** for the language accepted by the machine in Figure 1.

6. Give a regular expression for the language accepted by the machine in Figure 2

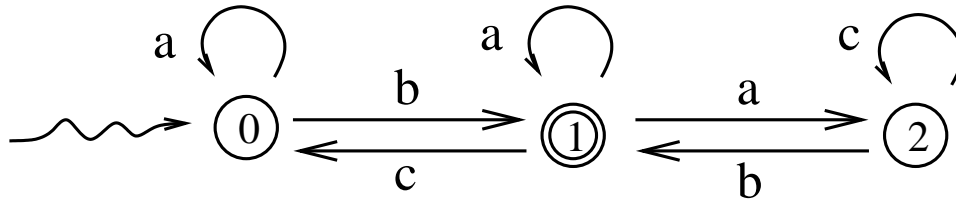


Figure 2: NFA for problem 6.

7. Which class of languages does each of these machine classes accept?

- (a) Deterministic finite automata. ....
- (b) Non-deterministic finite automata. ....
- (c) Push-down automata. ....
- (d) Turing Machines. ....

8. Let  $L = \{w \in \{a, b\}^* : \#_a(w) = \#_b(w)\}$ , which is generated by the following context-free grammar.

1.  $S \rightarrow aSbS$
2.  $S \rightarrow bSaS$
3.  $S \rightarrow \lambda$

Draw a PDA which accepts  $L$ .

9. The grammar below is an ambiguous CF grammar and is parsed by the LALR parser whose ACTION and GOTO tables are shown here. Write a computation of the parser for the input string  $iiwaea$ .

1.  $S \rightarrow i_2 S_3$
2.  $S \rightarrow i_2 S_3 e_4 S_5$
3.  $S \rightarrow w_6 S_7$
4.  $S \rightarrow a_8$

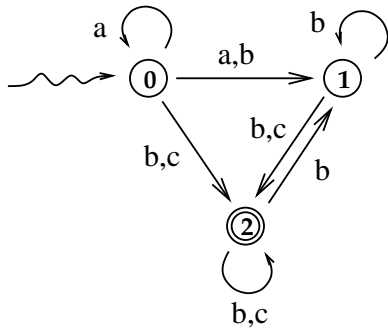
	$a$	$i$	$e$	$w$	$\$$	$S$
0	s8	s2		s6		1
1					halt	
2	s8	s2		s6		3
3			s4		r1	
4	s8	s2		s6		5
5			r2		r2	
6	s8	s2		s6		7
7			r3		r3	
8			r4		r4	

1. True or False. T = true, F = false, and O = open, meaning that the answer is not known science at this time. In the questions below,  $\mathcal{P}$  and  $\mathcal{NP}$  denote  $\mathcal{P}$ -TIME and  $\mathcal{NP}$ -TIME, respectively.
  - (i) There is some PDA that accepts  $L$ , where  $L$  is the language over  $\{a, b, c\}$  consisting of all strings which have more  $a$ 's than  $b$ 's and more  $b$ 's than  $c$ 's.
  - (ii) \_\_\_\_\_ The language  $\{a^n b^n \mid n \geq 0\}$  is context-free.
  - (iii) \_\_\_\_\_ The language  $\{a^n b^n c^n \mid n \geq 0\}$  is context-free.
  - (iv) \_\_\_\_\_ The language  $\{a^i b^j c^k \mid j = i + k\}$  is context-free.  
 \_\_\_\_\_ The intersection of any regular language with any context-free language is context-free.
  - (v) \_\_\_\_\_ The intersection of any two context-free languages is context-free.
  - (vi) \_\_\_\_\_ If  $L$  is a context-free language over an alphabet with just one symbol, then  $L$  is regular.
  - (vii) \_\_\_\_\_ The set of strings that your high school algebra teacher would accept as legitimate expressions is a context-free language.
  - (viii) \_\_\_\_\_ Every language accepted by a non-deterministic machine is accepted by some deterministic machine.
  - (ix) \_\_\_\_\_ The problem of whether a given string is generated by a given context-free grammar is decidable.
  - (x) \_\_\_\_\_ Every language generated by an unambiguous context-free grammar is accepted by some DPDA.
  - (xi) \_\_\_\_\_ The language  $\{a^n b^n c^n \mid n \geq 0\}$  is in the class  $\mathcal{P}$ -TIME.
  - (xii) \_\_\_\_\_ There exists a polynomial time algorithm which finds the factors of any positive integer, where the input is given as a binary numeral.

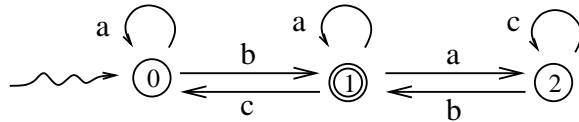
- (xiii) — The intersection of any two undecidable languages is undecidable.
- (xiv) — Every  $\mathcal{NP}$  language is decidable.
- (xv) — The intersection of two  $\mathcal{NP}$  languages must be  $\mathcal{NP}$ .
- (xvi) — There exists a  $\mathcal{P}$ -TIME algorithm which finds a maximum independent set in any graph  $G$ .
- (xvii) —  $\mathcal{NC} = \mathcal{P}$ .
- (xviii) —  $\mathcal{P} = \mathcal{NP}$ .
- (xix) —  $\mathcal{NP} = \mathcal{P}$ -SPACE
- (xx) —  $\text{EXP-TIME} = \mathcal{P}$ -TIME.
- (xxi) —  $\text{EXP-SPACE} = \mathcal{P}$ -SPACE.
- (xxii) — The traveling salesman problem (TSP) is  $\mathcal{NP}$ -complete.
- (xxiii) — The knapsack problem is  $\mathcal{NP}$ -complete.
- (xxiv) — The language consisting of all satisfiable Boolean expressions is  $\mathcal{NP}$ -complete.
- (xxv) — The Boolean Circuit Problem is in  $\mathcal{P}$ .
- (xxvi) — The Boolean Circuit Problem is in  $\mathcal{NC}$ .
- (xxvii) — 2-SAT is  $\mathcal{P}$ -TIME.
- (xxviii) — 3-SAT is  $\mathcal{P}$ -TIME.
- (xxix) — Primality, using binary numerals, is  $\mathcal{P}$ -TIME.
- (xxx) — There is a  $\mathcal{P}$ -TIME reduction of the halting problem to 3-SAT.
- (xxxi) — Every context-free language is in  $\mathcal{P}$ .
- (xxxii) — Every context-free language is in  $\mathcal{NC}$ .
- (xxxiii) — Every language generated by an unrestricted grammar is recursive.
- (xxxiv) — For any two languages  $L_1$  and  $L_2$ , if  $L_1$  is undecidable and there is a recursive reduction of  $L_1$  to  $L_2$ , then  $L_2$  must be undecidable.
- (xxxv) — If  $P$  is a mathematical proposition that can be written using a string of length  $n$ , and  $P$  has a proof, then  $P$  must have a proof whose length is  $O(2^{2^n})$ .
- (xxxvi) — If  $L$  is any  $\mathcal{NP}$  language, there must be a  $\mathcal{P}$ -TIME reduction of  $L$  to the partition problem.
- (xxxvii) — Every bounded function is recursive.
- (xxxviii) — If  $L$  is  $\mathcal{NP}$  and also  $\text{co-}\mathcal{NP}$ , then  $L$  must be  $\mathcal{P}$ .

- (xxxix) \_\_\_\_\_ Recall that if  $\mathcal{L}$  is a class of languages,  $\text{co-}\mathcal{L}$  is defined to be the class of all languages that are not in  $\mathcal{L}$ . Let  $\mathcal{RE}$  be the class of all recursively enumerable languages. If  $L$  is in  $\mathcal{RE}$  and also  $L$  is in  $\text{co-}\mathcal{RE}$ , then  $L$  must be decidable.
- (xl) \_\_\_\_\_ If a language  $L$  is undecidable, then there can be no machine that enumerates  $L$ .
- (xli) \_\_\_\_\_ There is a non-recursive function which grows faster than any recursive function.
- (xlii) \_\_\_\_\_ For every real number  $x$ , there exists a machine that runs forever and outputs the string of decimal digits of  $x$ .
- (xliii) \_\_\_\_\_ Every subset of a regular language is regular.
- (xliv) \_\_\_\_\_ The computer language C++ has Turing power.
- (xlv) \_\_\_\_\_ If  $L$  is any  $\mathcal{P}$ -TIME language, there is an  $\mathcal{NC}$  reduction of  $L$  to the Boolean circuit problem.
- (xlvi) \_\_\_\_\_ The binary integer factorization problem is  $\text{co-}\mathcal{NP}$ .
- (i) True/False/Open
- i. \_\_\_\_\_ Every subset of a regular language is decidable.
  - ii. \_\_\_\_\_ The intersection of any two  $\mathcal{NP}$  languages is  $\mathcal{NP}$ . **Think!**
  - iii. \_\_\_\_\_ Every language accepted by a non-deterministic machine is accepted by some deterministic machine.
  - iv. \_\_\_\_\_  $\mathcal{NC} = \mathcal{P}$ .
  - v. \_\_\_\_\_  $\mathcal{P} = \mathcal{NP}$ .
  - vi. \_\_\_\_\_ The Boolean Circuit Problem (CVP) is in  $\mathcal{NC}$ .
  - vii. \_\_\_\_\_ The independent set problem is  $\mathcal{P}$ -TIME.
  - viii. \_\_\_\_\_ If  $S$  is a recursive set of positive integers, then  $\sum_{n \in S} 2^{-n}$  must be a recursive real number.
  - ix. \_\_\_\_\_ Multiplication of matrices with binary numeral entries is  $\mathcal{NC}$ .
  - x. \_\_\_\_\_ Equivalence of regular expressions is decidable.
  - xi. \_\_\_\_\_ Every recursively enumerable language is generated by a general grammar.
  - xii. \_\_\_\_\_ Equivalence of context-free grammars is  $\text{co-}\mathcal{RE}$ .
  - xiii. \_\_\_\_\_ The language consisting of all fractions whose values are less than the natural logarithm of 5.0 is recursive.
  - xiv. \_\_\_\_\_ If  $L$  is in  $\mathcal{RE}$  and also  $\text{co-}\mathcal{RE}$ , then  $L$  must be decidable.
  - xv. \_\_\_\_\_ For every real number  $x$ , there exists a machine that runs forever and outputs the string of decimal digits of  $x$ .
  - xvi. \_\_\_\_\_ Every sliding block problem is  $\mathcal{P}$ -SPACE.

- xvii. ----- There are uncountably many  $\text{co-}\mathcal{RE}$  languages.
  - xviii. ----- If  $L$  is any  $\mathcal{P}$ -TIME language, there is an  $\mathcal{NC}$  reduction of  $L$  to CVP, the Boolean circuit problem.
  - xix. ----- There is a polynomial time algorithm for checking whether an integer is prime.
  - xx. ----- Every finite language is regular.
  - xxi. ----- If  $L$  is a  $\mathcal{P}$ -TIME language, there is a Turing Machine which decides  $L$  in polynomial time.
  - xxii. ----- If anyone ever finds a polynomial time algorithm for any  $\mathcal{NP}$ -complete language, then  $\mathcal{P} = \mathcal{NP}$ .
  - xxiii. ----- RSA encryption is believed to be secure because it is believed that the factorization problem for integers is very hard.
- (ii) Find an NFA with at most 4 states which accepts the language of binary strings which contain the substring 111.
- (iii) Construct a minimal DFA equivalent to the NFA shown below.



- (iv) Find an NFA which accepts the language generated by this grammar.
- $$\begin{aligned}
 S &\rightarrow aA|cS|cC \\
 A &\rightarrow aA|bS|cB|\lambda \\
 B &\rightarrow aA|cB|bC|\lambda \\
 C &\rightarrow aB
 \end{aligned}$$
- (v) Give a regular expression which describes the language accepted by this NFA.



- (vi) Use the CYK algorithm to decide whether  $abcab$  is generated by the CNF grammar:

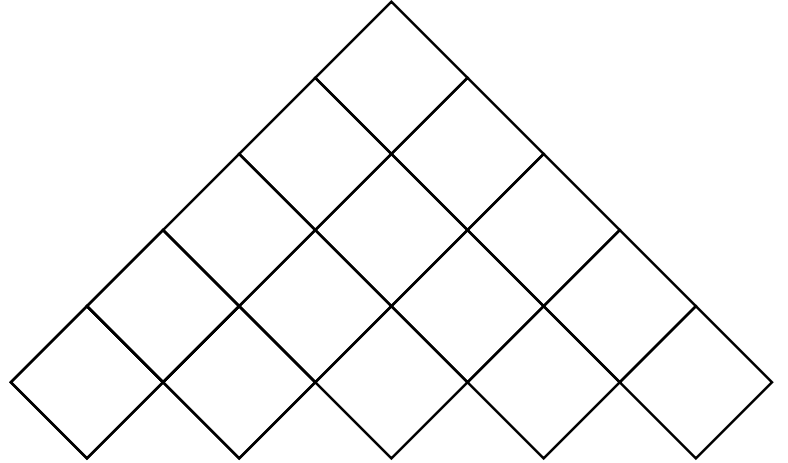
$$S \rightarrow AB \mid BC \mid CA$$

$$A \rightarrow a$$

$$B \rightarrow SA \mid SS \mid b$$

$$C \rightarrow c$$

by filling in the matrix.



- (vii) The LALR parser given for this grammar:

$$1. E \rightarrow E -_2 E_3$$

$$2. E \rightarrow E *_4 E_5$$

$$3. E \rightarrow x_6$$

contains errors, meaning that it might parse a string in a manner that would be considered incorrect by your programming instructor. Find those errors and correct them.

	$x$	$-$	$*$	$\$$	$E$
0	$s6$				1
1		$s2$	$s4$	$halt$	
2	$s6$				3
3		$r1$	$r1$	$r1$	
4	$s6$				5
5		$s2$	$s4$	$r2$	
6		$r3$	$r3$	$r3$	

- (viii) Prove that the grammar given in Problem 1g is ambiguous by giving two different leftmost derivations for some string. (If you simply give two different parse trees, you have not answered the question.)

- (ix) State the pumping lemma for regular languages.

- (x) Give the verifier-certificate definition of the class  $\mathcal{NP}$ .

(xi) What is the importance nowadays of  $\mathcal{NC}$ ?

(xii) What complexity class contains sliding block problems?

(xiii) Give a polynomial time reduction of the subset sum problem to the partition problem.

(xiv) Label each of the following sets as countable or uncountable.

- i. ----- The set of integers.
- ii. ----- The set of rational numbers.
- iii. ----- The set of real numbers.
- iv. ----- The set of binary languages.
- v. ----- The set of co- $\mathcal{RE}$  binary languages.
- vi. ----- The set of undecidable binary languages.
- vii. ----- The set of functions from integers to integers.
- viii. ----- The set of recursive real numbers.

2. Every language, or problem, falls into exactly one of these categories. For each of the languages, write a letter indicating the correct category.

**A** Known to be  $\mathcal{NC}$ .

**B** Known to be  $\mathcal{P}$ -TIME, but not known to be  $\mathcal{NC}$ .

**C** Known to be  $\mathcal{NP}$ , but not known to be  $\mathcal{P}$ -TIME and not known to be  $\mathcal{NP}$ -complete.

**D** Known to be  $\mathcal{NP}$ -complete.

**E** Known to be  $\mathcal{P}$ -SPACE but not known to be  $\mathcal{NP}$

**F** Known to be decidable, but not known to be  $\mathcal{P}$ -SPACE.

**G**  $\mathcal{RE}$  but not decidable.

**H** co- $\mathcal{RE}$  but not decidable.

**I** Neither  $\mathcal{RE}$  nor co- $\mathcal{RE}$ .

- (i) ----- All C++ programs which do not halt if given themselves as input.
  - (ii) ----- All base 10 numerals for perfect squares.
  - (iii) ----- The Dyck language.
  - (iv) -----  $\{\langle G \rangle : L(G) \text{ is the Dyck language.}\}$
  - (v) ----- The Jigsaw problem. (That is, given a finite set of two-dimensional pieces, can they be assembled into a rectangle, with no overlap and no spaces.)
  - (vi) ----- Factorization of binary numerals.
3. Find a DFA equivalent to the NFA shown in Figure 1.
  4. Give a regular grammar for the language accepted by the machine in Figure 1.
  5. Which class of languages does each of these machine classes accept?
    - (i) Deterministic finite automata. -----
    - (ii) Non-deterministic finite automata. -----
    - (iii) Push-down automata. -----
    - (iv) Turing Machines. -----
  6. Let  $L = \{w \in \{a, b\}^* : \#_a(w) = \#_b(w)\}$ , that is, each string of  $L$  has equal numbers of each symbol. Draw a PDA which accepts  $L$ .
  7. The grammar below is an ambiguous CF grammar with start symbol  $E$ , and is parsed by the LALR parser whose ACTION and GOTO tables are shown here. The ACTION table is missing actions for the second column, when the next input symbol is the “minus” sign. Fill it in. Remember the C++ precedence of operators. (Hint: the column has seven different actions: s2, s4, r1, r2, r3, r4, and r5, some more than once, and has no blank spaces.)

1.  $E \rightarrow E -_2 E_3$
2.  $E \rightarrow -_4 E_5$
3.  $E \rightarrow E *_6 E_7$
4.  $E \rightarrow ({}_8 E_9)_{10}$
5.  $E \rightarrow x_{11}$

	$x$	$-$	$*$	$($	$)$	$\$$	$S$
0	s11			s8			1
1			s6			halt	
2	s11			s8			3
3			s6		r1	r1	
4	s11			s8			5
5			r2		r2	r2	
6	s11			s8			7
7			r3		r3	r3	
8	s11			s8			9
9			s6		s6		
10			r4	r4	r4	r4	
11			r5		r5	r5	

8. Prove that any decidable language can be enumerated in canonical order by some machine.
9. Give a polynomial time reduction of 3-SAT to to the independent set problem.
1. True or False. T = true, F = false, and O = open, meaning that the answer is not known science at this time. In the questions below,  $\mathcal{P}$  and  $\mathcal{NP}$  denote  $\mathcal{P}$ -TIME and  $\mathcal{NP}$ -TIME, respectively.
  - (i) \_\_\_\_\_ There is some PDA that accepts  $\{w \in \{a, b, c\}^* : \#_a(w) > \#_b(w) > \#_c(w)\}$ , that is, the language over  $\{a, b, c\}$  consisting of all strings which have more  $a$ 's than  $b$ 's and more  $b$ 's than  $c$ 's.
  - (ii) \_\_\_\_\_ The intersection of any two context-free languages is context-free.
  - (iii) \_\_\_\_\_ Every language accepted by a non-deterministic machine is accepted by some deterministic machine.
  - (iv) \_\_\_\_\_ The set of palindromes over  $\{a, b\}$  is accepted by some DPDA.
  - (v) \_\_\_\_\_ The language  $\{a^n b^n c^n \mid n \geq 0\}$  is in the class  $\mathcal{NC}$ .
  - (vi) \_\_\_\_\_ There exists a polynomial time algorithm which finds the factors of any positive integer, where the input is given as a binary numeral.
  - (vii) \_\_\_\_\_ Every problem that can be mathematically defined has an algorithmic solution.
  - (viii) \_\_\_\_\_ The complement of any undecidable language is undecidable.
  - (ix) \_\_\_\_\_  $\mathcal{NC} = \mathcal{P}$ .
  - (x) \_\_\_\_\_  $\mathcal{P} = \mathcal{NP}$ .
  - (xi) \_\_\_\_\_ The Boolean Circuit Problem (also known as the CVP problem) is in  $\mathcal{P}$ .

- (xii) \_\_\_\_\_ The Boolean Circuit Problem (CVP) is  $\mathcal{NC}$ .
- (xiii) \_\_\_\_\_  $\text{co-}\mathcal{P} = \mathcal{P}$ .
- (xiv) \_\_\_\_\_ 2-SAT is  $\mathcal{P}$ -TIME.
- (xv) \_\_\_\_\_ 3-SAT is  $\mathcal{P}$ -TIME.
- (xvi) \_\_\_\_\_ The set of binary numerals for prime numbers is  $\mathcal{P}$ -TIME.
- (xvii) \_\_\_\_\_ Every language generated by an unrestricted (general) grammar is recursive.
- (xviii) \_\_\_\_\_ If  $L$  is  $\mathcal{NP}$  and also  $\text{co-}\mathcal{NP}$ , then  $L$  must be  $\mathcal{P}$ -TIME.
- (xix) \_\_\_\_\_ If  $L$  is in  $\mathcal{RE}$  and also  $\text{co-}\mathcal{RE}$ , then  $L$  must be recursive (decidable).
- (xx) \_\_\_\_\_ There is a mathematical proposition that is true but cannot be proved true.
- (xxi) \_\_\_\_\_ There is a non-recursive function which grows faster than any recursive function.
- (xxii) \_\_\_\_\_ There is a Turing machine which, when turned on, runs forever, writing the decimal expansion of  $\pi$  (the well-known ratio of the circumference of a circle to its diameter).
- (xxiii) \_\_\_\_\_ The binary integer factorization problem is  $\text{co-}\mathcal{NP}$ .
- (xxiv) \_\_\_\_\_ If  $L$  is  $\mathcal{NP}$ , there is a polynomial time reduction of  $L$  to the subset sum problem.
- (xxv) \_\_\_\_\_ The intersection of any two  $\mathcal{NP}$  languages is  $\mathcal{NP}$ .
- (xxvi) \_\_\_\_\_ The intersection of any two  $\text{co-}\mathcal{NP}$  languages is  $\text{co-}\mathcal{NP}$ .
- (xxvii) \_\_\_\_\_ The intersection of any two  $\text{co-}\mathcal{RE}$  languages is  $\text{co-}\mathcal{RE}$ .
- (xxviii) \_\_\_\_\_ Multiplication of matrices with binary numeral entries is  $\mathcal{NC}$ .
- (xxix) \_\_\_\_\_ Every recursively enumerable language is generated by an unrestricted (general) grammar.
- (xxx) \_\_\_\_\_ Equivalence of context-free grammars is  $\text{co-}\mathcal{RE}$ .
- (xxx1) \_\_\_\_\_ The language of all true mathematical statements is recursively enumerable.
- (xxx2) \_\_\_\_\_ The language of all **provably** true mathematical statements is recursively enumerable.
- (xxx3) \_\_\_\_\_ There are uncountably many undecidable languages over the binary alphabet.
- (xxx4) \_\_\_\_\_ If there exists a polynomial time algorithm for any  $\mathcal{NP}$ -complete problem, then  $\mathcal{P} = \mathcal{NP}$ .
- (xxx5) \_\_\_\_\_ RSA encryption is accepted as secure by most experts, because they believe that the factorization problem for binary numerals is very hard.
- (xxx6) \_\_\_\_\_ The language of all  $\langle G_1 \rangle \langle G_2 \rangle$  such that  $G_1$  and  $G_2$  are CF grammars which are **not** equivalent is  $\text{co-}\mathcal{RE}$ .

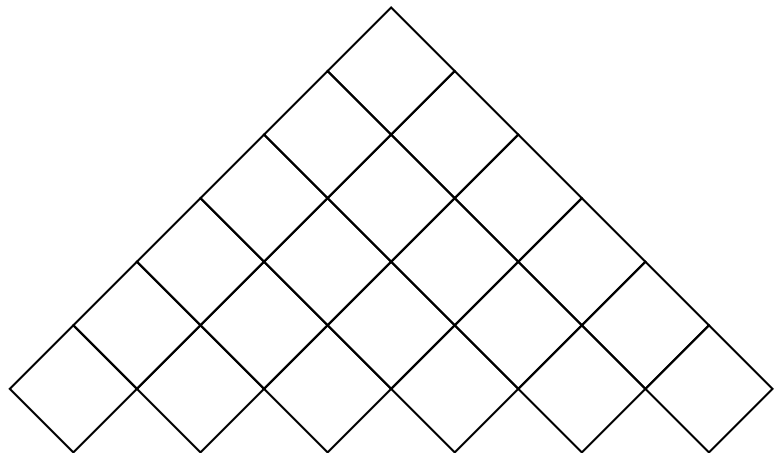
- (xxxvii) ----- A real number  $x$  is recursive if and only if the set of fractions whose values are greater than  $x$  is recursive (decidable).
- (xxxviii) ----- For any real number  $x$ , there exists a machine that runs forever and outputs the string of decimal digits of  $x$ .
- (xxxix) ----- If a Boolean expression is satisfiable, there is a  $\mathcal{P}$ -TIME proof that it's satisfiable.
  - (xl) ----- If there is a solution to a given instance of any sliding block problem, there must be a solution of polynomial length.
  - (xli) ----- If the Boolean circuit problem (CVP) is  $\mathcal{NC}$ , then  $\mathcal{P} = \mathcal{NC}$ .

- 2. (a) Give a context-sensitive grammar for  $\{a^n b^n c^n : n > 0\}$
- (b) Using that grammar, give a derivation of the string  $aaabbbccc$ .

- 3. Illustrate an NFA which accepts the language generated by this grammar.
  - $S \rightarrow aA|cS|cC$
  - $A \rightarrow aA|bS|cB|\lambda$
  - $B \rightarrow aA|cB|bC|\lambda$
  - $C \rightarrow aB$

- 4. Use the CYK algorithm to decide whether  $x - x - -x$  is generated by the CNF grammar below, by filling in the matrix.

- $E \rightarrow ME$
- $A \rightarrow EM$
- $E \rightarrow AE$
- $M \rightarrow -$
- $E \rightarrow x$



- 5. What is the importance nowadays of  $\mathcal{NC}$ ?
- 6. Label each of the following sets as countable or uncountable.
  - (a) ----- The set of integers.
  - (b) ----- The set of rational numbers.
  - (c) ----- The set of real numbers.
  - (d) ----- The set of binary languages.
  - (e) ----- The set of co-RE binary languages.
  - (f) ----- The set of undecidable binary languages.

- (g) ----- The set of functions from integers to integers.
- (h) ----- The set of recursive real numbers.
- (i) ----- The set of  $\mathcal{P}$ -SPACE languages over the binary alphabet.
- (j) The set of functions from the integers to the binary alphabet  $\{0, 1\}$ .

7. Let  $L = \{w \in \{a, b\}^* : \#_a(w) = \#_b(w)\}$ , that is, each string of  $L$  has equal numbers of each symbol. Draw a DPDA which accepts  $L$ . (Recall that the input to that DPDA must be of the form  $w\$,$  where  $w \in L$  and  $\$$  is the end-of-file symbol.)

8. Consider the CF grammar below. The ACTION and GOTO tables are given below, except that six actions are missing, indicated by question marks. Fill in the missing actions (below the question marks). The actions of your table must be consistent with the precedence of operators in C++.

	$x$	$-$	$*$	$\$$	$E$
1. $E \rightarrow E -_2 E_3$	0	$s8$	$s4$		1
2. $E \rightarrow -_4 E_5$	1		$s2$	$s6$	HALT
3. $E \rightarrow E *_6 E_7$	2	$s8$	$s4$		3
4. $E \rightarrow x_8$	3		?	?	$r1$
	4	$s8$	$s4$		5
	5		?	?	$r2$
	6	$s8$	$s4$		7
	7		?	?	$r3$
	8	$s8$	$r4$	$r4$	$r4$

Prove, by induction, that  $\sum_{i=1}^n n^2 = \frac{n(n+1)(2n+1)}{6}$

9. Prove, by contradiction, that  $\sqrt{3}$  is irrational.

10. Prove that the halting problem is undecidable.