University of Nevada, Las Vegas Computer Science 456/656 Spring 2025

Practice Problems for the Final Examination on May 14, 2025

Throughout, $\mathcal{P}$ means $\mathcal{P}$–TIME.

Despite my efforts, this list of problems may contain duplicates. Sorry about that!

1. Consult the lists of true false questions that are posted in Handouts.

2. 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) ________ The union of any two regular languages is regular.

   (ii) ________ The intersection of two regular languages is regular.

   (iii) ________ The concatenation of two regular languages is regular.

   (iv) ________ The complement of any regular language is regular.

   (v) ________ The Kleene closure of any regular language is regular.

   (vi) ________ The union of two context-free languages is context-free.

   (vii) ________ The intersection of two context-free languages is context-free.

   (viii) ________ The concatenation of two context-free languages is context-free.

   (ix) ________ The complement of any context-free language is context-free.

   (x) ________ The Kleene closure of any context-free language is context-free.

   (xi) ________ The union of two $\mathcal{P}$ languages is $\mathcal{P}$.

   (xii) ________ The intersection of two $\mathcal{P}$ languages is $\mathcal{P}$.

   (xiii) ________ The concatenation of two $\mathcal{P}$ languages is $\mathcal{P}$.

   (xiv) ________ The complement of any $\mathcal{P}$ language is $\mathcal{P}$.

   (xv) ________ The Kleene closure of any $\mathcal{P}$ language is $\mathcal{P}$.

   (xvi) ________ The union of two $\mathcal{NP}$ languages is $\mathcal{NP}$.

   (xvii) ________ The intersection of two $\mathcal{NP}$ languages is $\mathcal{NP}$.

   (xviii) ________ The concatenation of two $\mathcal{NP}$ languages is $\mathcal{NP}$.

   (xix) ________ The complement of any $\mathcal{NP}$ language is $\mathcal{NP}$.

   (xx) ________ The Kleene closure of any $\mathcal{NP}$ language is $\mathcal{NP}$.

   (xxi) ________ The union of two decidable languages is decidable.

(xxii) ________ The intersection of two decidable languages is decidable.

(xxiii) ________ The concatenation of two decidable languages is decidable.

(xxiv) ________ The complement of any decidable language is decidable.

(xxv) ________ The Kleene closure of any decidable language is decidable.

(xxvi) ________ The union of two $\mathcal{RE}$ languages is $\mathcal{RE}$.

(xxvii) ________ The intersection of two $\mathcal{RE}$ languages is $\mathcal{RE}$.

(xxviii) ________ The concatenation of two $\mathcal{RE}$ languages is $\mathcal{RE}$.

(xxix) ________ The complement of any $\mathcal{RE}$ language is $\mathcal{RE}$.

(xxx) ________ The Kleene closure of any $\mathcal{RE}$ language is $\mathcal{RE}$.

(xxxi) ________ The union of two undecidable languages is undecidable.

(xxxii) ________ The intersection of two undecidable languages is undecidable.

(xxxiii) ________ The complement of any undecidable language is undecidable.

(xxxiv) ________ The Kleene closure of any undecidable language is undecidable.

(xxxv) ________ The union of any $\mathcal{NC}$ languages is $\mathcal{NC}$.

(xxxvi) ________ The intersection of two $\mathcal{NC}$ languages is $\mathcal{NC}$.

(xxxvii) ________ The concatenation of $\mathcal{NC}$ languages is $\mathcal{NC}$.

(xxxviii) ________ The complement of any $\mathcal{NC}$ language is $\mathcal{NC}$.

(xxxix) ________ The Kleene closure of any $\mathcal{NC}$ language is $\mathcal{NC}$.

(xl) ________ $\mathcal{NC}. = \mathcal{P}$.

(xli) ________ $\mathcal{NC}. = \mathcal{P}$–SPACE.

(xlii) ________ $\mathcal{P} = \mathcal{NP}$.

(xliii) ________ $\mathcal{P} = \mathcal{P}$–SPACE.

(xliv) ________ $\mathcal{P} =$ EXP–TIME.

(xlv) ________ $\mathcal{P}$–SPACE = EXP–SPACE.

(xlvi) ________ There is a PDA that accepts the language of all palindromes over $\{a, b\}$.

(xlvii) ________ If every $w \in L$ can be proved to be in L, then L must be decidable.

(xlviii) ________ 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.

(xlix) ______ The language $\{a^n b^n \mid n \geq 0\}$ is context-free.

(l) ______ The language $\{a^n b^n c^n \mid n \geq 0\}$ is context-free.

(li) ______ The language $\left\{a^i b^j c^k \mid j = i + k\right\}$ is context-free.

(lii) ______ The intersection of any regular language with any context-free language is context-free.

(liii) ______ If L is a context-free language over an alphabet with just one symbol, then L is regular.

(liv) ______ The set of strings that your high school algebra teacher would accept as legitimate expressions is a context-free language.

(lv) ______ Every language accepted by a non-deterministic machine is accepted by some deterministic machine.

(lvi) ______ The problem of whether a given string is generated by a given context-free grammar is decidable.

(lvii) ______ Every language generated by an unambiguous context-free grammar is accepted by some DPDA.

(lviii) ______ The language $\{a^n b^n c^n \mid n \geq 0\}$ is in the class $\mathcal{P}$-TIME.

(lix) ______ There exists a polynomial time algorithm which finds the factors of any positive integer, where the input is given as a binary numeral.

(lx) ______ Every $\mathcal{NP}$ language is decidable.

(lxi) ______ $\mathcal{NP} = \mathcal{P}$-SPACE

(lxii) ______ EXP-TIME = EXP-SPACE

(lxiii) ______ The traveling salesman problem (TSP) is $\mathcal{NP}$–complete.

(lxiv) ______ The knapsack problem is $\mathcal{NP}$–complete.

(lxv) ______ The language consisting of all satisfiable Boolean expressions is $\mathcal{NP}$-complete.

(lxvi) ______ The Boolean Circuit Problem is in $\mathcal{P}$.

(lxvii) ______ The Boolean Circuit Problem is in $\mathcal{NC}$.

(lxviii) ______ 2-SAT is $\mathcal{P}$-TIME.

(lxix) ______ 3-SAT is $\mathcal{P}$-TIME.

(lxx) ______ Primality, using binary numerals, is $\mathcal{P}$-TIME.

(lxxi) ______ Every context-free language is in $\mathcal{P}$.

(lxxii) ______ Every context-free language is in $\mathcal{NC}$.

(lxxiii) ______ Every language generated by an unrestricted grammar is recursive.

(lxxiv) ______ The problem of whether two given context-free grammars generate the same language is decidable.

(lxxv) ______ The language of all fractions (using base 10 numeration) whose values are less than π is decidable.

(lxxvi) ______ There exists a polynomial time algorithm which finds the factors of any positive integer, where the input is given as a unary ("caveman") numeral.

(lxxvii) ______ 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.

(lxxviii) ______ 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 $O(2^{2^n})$.

(lxxix) ______ If L is any $\mathcal{NP}$ language, there must be a $\mathcal{P}$–TIME reduction of L to the partition problem.

(lxxx) ______ Every bounded function is recursive.

(lxxxi) ______ If L is $\mathcal{NP}$ and also co-$\mathcal{NP}$, then L must be $\mathcal{P}$.

(lxxxii) ______ If a language L is in $\mathcal{RE}$ and also in co-$\mathcal{RE}$, then L must be decidable.

(lxxxiii) ______ Every language is enumerable.

(lxxxiv) ______ If a language L is undecidable, then there can be no machine that enumerates L.

(lxxxv) ______ There is a non-recursive function which grows faster than any recursive function.

(lxxxvi) ______ There exists a machine that runs forever and outputs the string of decimal digits of π (the well-known ratio of the circumference of a circle to its diameter).

(lxxxvii) ______ **Rush Hour**, the puzzle sold in game stores everywhere, generalized to a board of arbitrary size, is $\mathcal{NP}$–complete.

(lxxxviii) ______ There is a polynomial time algorithm which determines whether any two regular expressions are equivalent.

(lxxxix) ______ If two regular expressions are equivalent, there is a polynomial time proof that they are equivalent.

(xc) ______ Every subset of a regular language is regular.

(xci) ______ 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.

(xcii) ______ Every subset of any enumerable set is enumerable.

(xciii) ______ If L is a context-free language which contains the empty string, then $L\backslash\{\lambda\}$ must be context-free.

(xciv) ________ The computer language C++ has Turing power.

(xcv) ________ Let Σ be the binary alphabet. Every $w \in \Sigma^*$ which starts with 1 is a binary numeral for a positive integer. Let $Sq : \Sigma^* \to \Sigma^*$ be a function which maps the binary numeral for any integer n to the binary numeral for n^2. Then Sq is an $\mathcal{NC}$ function.

(xcvi) ________ If L is any $\mathcal{P}$-TIME language, there is an $\mathcal{NC}$ reduction of L to the Boolean circuit problem.

(xcvii) ________ If an abstract Pascal machine can perform a computation in polynomial time, there must be some Turing machine that can perform the same computation in polynomial time.

(xcviii) ________ The binary integer factorization problem is co-$\mathcal{NP}$.

(xcix) ________ Every subset of a regular language is decidable.

(c) ________ Every language accepted by a non-deterministic machine is accepted by some deterministic machine.

(ci) ________ The independent set problem is $\mathcal{P}$-TIME.

(cii) ________ If S is a set of positive integers whose set of binary numerals decidable, then $\sum_{n\in S} 2^{-n}$ must be a recursive real number.

(ciii) ________ Multiplication of matrices with binary numeral entries is $\mathcal{NC}$.

(civ) ________ Equivalence of regular expressions is decidable.

(cv) ________ Equivalence of context-free grammars is co–$\mathcal{RE}$.

(cvi) ________ The language consisting of all fractions whose values are less than the natural logarithm of 5.0 is recursive.

(cvii) ________ Every sliding block problem is $\mathcal{P}$–SPACE.

(cviii) ________ There are uncountably many co–$\mathcal{RE}$ binary languages.

(cix) ________ If L is any $\mathcal{P}$–TIME language, there is an $\mathcal{NC}$ reduction of L to CVP, the Boolean circuit problem.

(cx) ________ Every finite language is regular.

(cxi) ________ If L is a $\mathcal{P}$–TIME language, there is a Turing Machine which decides L in polynomial time.

(cxii) ________ If anyone ever finds a polynomial time algorithm for any $\mathcal{NP}$–complete language, then we will know that $\mathcal{P} = \mathcal{NP}$.

(cxiii) ________ RSA encryption is believed to be secure because it is believed that the factorization problem for integers is very hard.

(cxiv) ________ If S is a set of positive integers whose set of binary numerals is recursively enumerable, then $\sum_{n\in S} 2^{-n}$ must be a recursive real number.
(Hint: This problem is very hard.)

(cxv) ______ There is some PDA that accepts $\left\{w \in \{a,b,c\}^* : \#_a(w) > \#_b(w) > \#_c(w)\right\}$, 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.

(cxvi) ______ If L is $\mathcal{RE}$ and $w \in L$, there is a proof that $w \in L$.

(cxvii) ______ Every language accepted by a non-deterministic machine is accepted by some deterministic machine.

(cxviii) ______ The set of palindromes over $\{a,b\}$ is accepted by some DPDA.

(cxix) ______ The language $\{a^n b^n c^n \mid n \geq 0\}$ is in the class $\mathcal{NC}$.

(cxx) ______ Every problem that can be mathematically defined has an algorithmic solution.

(cxxi) ______ $\mathcal{NC} = \mathcal{P}$.

(cxxii) ______ $\mathcal{P} = \mathcal{NP}$.

(cxxiii) ______ The set of binary numerals for prime numbers is $\mathcal{P}$-TIME.

(cxxiv) ______ There is a gathematical proposition that is true but cannot be proved true.

(cxxv) ______ The binary integer factorization problem is co-$\mathcal{NP}$.

(cxxvi) ______ If L is $\mathcal{NP}$, there is a polynomial time reduction of L to the subset sum problem.

(cxxvii) ______ The intersection of any two $\mathcal{NP}$ languages is $\mathcal{NP}$.

(cxxviii) ______ The intersection of any two co-$\mathcal{NP}$ languages is co-$\mathcal{NP}$.

(cxxix) ______ The intersection of any two co-$\mathcal{RE}$ languages is co-$\mathcal{RE}$.

(cxxx) ______ Multiplication of matrices with binary numeral entries is $\mathcal{NC}$.

(cxxxi) ______ Every recursively enumerable language is generated by an unrestricted (general) grammar.

(cxxxii) ______ Equivalence of context-free grammars is co–$\mathcal{RE}$.

(cxxxiii) ______ The language of all true mathematical statements is recursively enumerable.

(cxxxiv) ______ The language of all **provably** true mathematical statements is recursively enumerable.

(cxxxv) ______ There are uncountably many undecidable languages over the binary alphabet.

(cxxxvi) ______ RSA encryption is accepted as secure by most experts, because they believe that the factorization problem for binary numerals is very hard.

(cxxxvii) ______ 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 $\mathcal{RE}$.

(cxxxviii) ______ A real number x is recursive if and only if the set of fractions whose values are greater than x is recursive (decidable).

(cxxxix) ________ For any real number x, there exists a machine that runs forever and outputs the string of decimal digits of x.

(cxl) ________ If a Boolean expression is satisfiable, there is a $\mathcal{P}$–TIME proof that it's satisfiable.

(cxli) ________ If there is a solution to a given instance of any sliding block problem, there must be a solution of polynomial length.

(cxlii) ________ If the Boolean circut problem (CVP) is $\mathcal{NC}$, then $\mathcal{P} = \mathcal{NC}$.

3. Give a definition of each term.

   (i) Accept. (That is, what does it mean for a machine to accept a language.)

   (ii) Decide. (That is, what does it mean for a machine to decide a language.)

   (iii) Canonical order of a language L.

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

(v) State the pumping lemma for regular languages.

(vi) State the pumping lemma for context-free languages.

(vii) What is the importance nowdays of $\mathcal{NC}$?

(viii) State the Church-Turing thesis.

4. 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. ________________________________

5. The LALR parser given for this grammar:
1. $E \to E -_2 E_3$
2. $E \to E *_4 E_5$
3. $E \to x_6$
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		$r3$	$r3$	$r3$	

6. 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 nown to be $\mathcal{P}$–SPACE.
G Known to be EXP–SPACE but not nown to be EXP–TIME.
H Known to be decidable, but not nown to be EXP–SPACE.
I $\mathcal{RE}$ but not decidable.
K co-$\mathcal{RE}$ but not decidable.
L Neither $\mathcal{RE}$ nor co-$\mathcal{RE}$.

(i) ________ All C++ programs which halt with no input.
(ii) ________ All base 10 numerals for perfect squares.
(iii) ________ All configurations of RUSH HOUR from which it's possible to win.
(iv) ________ All satisfiable Boolean expressions.
(v) ________ All binary numerals for composite integers. (Composite means not prime.)
(vi) ________ The furniture mover's problem.
(vii) ________ The set of all positions of Chinese GO, on a board of any size, from which white can win.

(viii) ________ All C++ programs which do not halt if given themselves as input.

(ix) ________ The Dyck language.

(x) ________ 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.)

(xi) ________ Factorization of binary numerals.

(xii) ________ SAT.

(xiii) ________ 3-SAT.

(xiv) ________ 2-SAT.

(xv) ________ The Independent Set problem.

(xvi) ________ The Subset Sum Problem.

(xvii) ________ The block sorting problem.

(xviii) ________ The sliding block problem.

(xix) ________ The Hamiltonian cyle problem.

(xx) ________ The traveling salesman problem.

(xxi) ________ The graph isomorphism problem.

(xxii) ________ The 3-coloring problem.

(xxiii) ________ The 2-coloring problem.

(xxiv) ________ The set of binary numerals for Busy Beaver numbers.

7. Prove that every context-sensitive language is decidable. The way to do this is to start with an arbitrary non-contracting grammar, 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.) You do not have to actually write the program, not even pseudocode; just explain how you would do it.

8. Prove that every decidable language is enumerated in canonical order by some machine.

9. Prove that every language that is enumerated in canonical order by some machine is decided by some other machine..

10. Prove that eny language accepted by any machine can be enumerated by some other machine.

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

12. Prove that the halting problem is undecidable.

13. Prove that the grammar given in Problem 5 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.)

14. Use the pumping lemma to prove that the Dyck language is not regular.

15. Give a polynomial time reduction of 3-SAT to to the independent set problem.

16. Give a polynomial time reduction of the subset sum problem to the partition problem.

17. 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.)

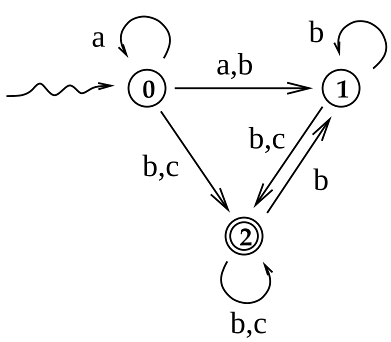
    1. $S \to aSbS$
    2. $S \to \lambda$

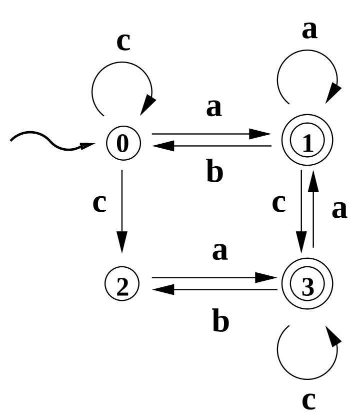
    Prove that the subset of the Dyck language consisting of strings whose nesting depth is no more than 50 is regular.

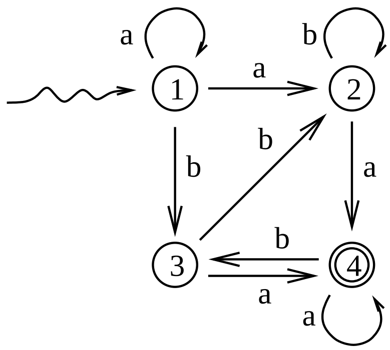
18. Find an NFA with at most 4 states which accepts the language of binary strings which contain the substring 111.

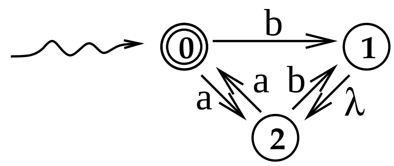
19. Let L be the language of all binary numerals for positive multiples of either 3 or 4, where leading zeros are not permitted. That is, $L = \{11, 100, 110, 1000, 1001, 1100, \ldots\}$. Find an NFA with 8 states which accepts L. (There is also a DFA with 12 states which accepts L.)

20. Construct a minimal DFA equivalent to each of the NFA shown below.

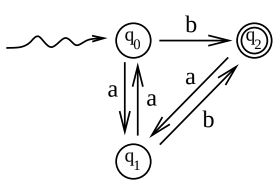


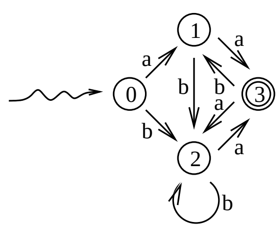


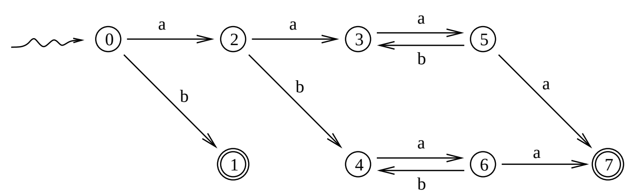




21. Minimize each of the following DFA.



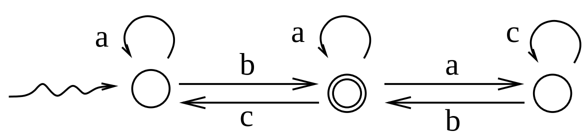




22. Find an NFA which accepts the language generated by this grammar.

$S \to aA|cS|cC$
$A \to aA|bS|cB|\lambda$
$B \to aA|cB|bC|\lambda$
$C \to aB$

23. Give a regular expression which describes the language accepted by this NFA.



25. Give a regular grammar with **no more than three variables** for the language accepted by the machine in Figure 2.

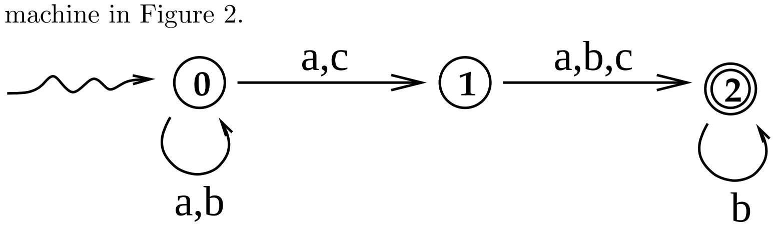


Figure 2: NFA for problems 25, 26 and 28

26. Find a minimal DFA equivalent to the NFA shown in Figure 2.

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

29. Illustrate an NFA which accepts the language generated by this grammar.

$S \to aA|cS|cC$
$A \to aA|bS|cB|\lambda$
$B \to aA|cB|bC|\lambda$
$C \to aB$

30. Let $L = \{w \in \{a,b\}^* : \#_a(w) = \#_b(w)\}$, 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.) L is generated by the context-free grammar below.
   1. $S \to aSbS$
   2. $S \to bSaS$
   3. $S \to \lambda$

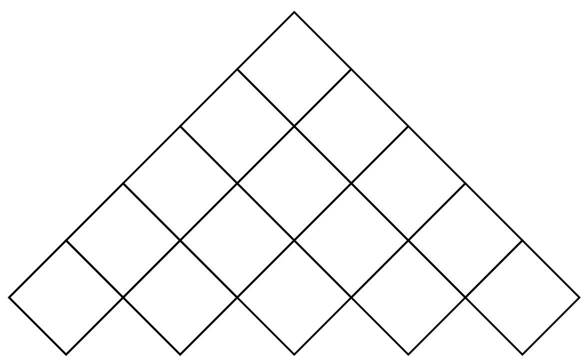
31. We know that context-free languages are exactly those which are accepted by push-down automata. We now define a new class of machines, which we call "limited push-down automata." An LPDA is exactly the same as a PDA, but with the restriction that the stack is never allowed to be larger than some given constant. What is the class of languages accepted by limited push-down automata? Think!

32. What is the class of languages decided by 2-PDA? A 2-PDA is the same as a PDA, except that it has two statcks instead of just one.

33. Use the CYK algorithm to decide whether *abcab* is generated by the CNF grammar. The start symbol is S.

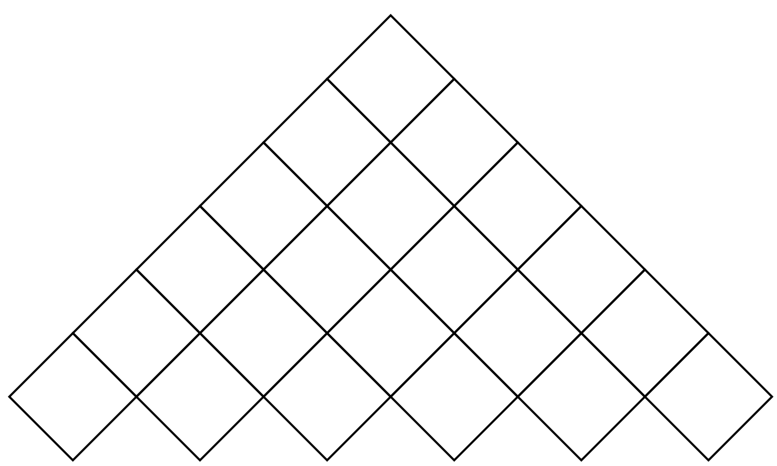
    $S \to AB \,|\, BC \,|\, CA$
    $A \to a$
    $B \to SA \,|\, SS \,|\, b$
    $C \to c$

    by filling in the matrix.



34. Use the CYK algorithm to decide whether $x - x - -x$ is generated by the CNF grammar below, by filling in the matrix. The start symbol is E.

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



35. What complexity class contains all sliding block problems?

36. Label each of the following sets as countable or uncountable. Write **C** or **U**.

(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.

(ix) ________ The set of $\mathcal{P}$–SPACE languages over the binary alphabet.

(x) ________ The set of functions from the integers to the binary alphabet $\{0, 1\}$.

(xi) ________ The set of functions from the binary alphabet $\{0, 1\}$ to the integers.

37. 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. ____

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

	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			$r5$		$r5$	$r5$	

40. Consider the CF grammar below. The ACTION and GOTO tables are given below, except that six actions are mising, 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++.

1. $E \to E -_2 E_3$
2. $E \to -_4 E_5$
3. $E \to E *_6 E_7$
4. $E \to x_8$

	x	$-$	$*$	\$	E
0	$s8$	$s4$			1
1		$s2$	$s6$	HALT	
2	$s8$	$s4$			3
3		?	?	$r1$	
4	$s8$	$s4$			5
5		?	?	$r2$	
6	$s8$	$s4$			7
7		?	?	$r3$	
8	$s8$	$r4$	$r4$	$r4$	

41. When there is an "else" after two "if"s, which "if" does the "else" pair with? Here is CF grammar, G_3, which isolates this problem. The start symbol S is the only variable. The symbol i represents "if(condition)", e represents "else," w represents "while(condition)" and a represents any other statement, such as an assignment statement.

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

	ACTION					GOTO
	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	

Which entry, or entries, solve the dangling else problem?

Walk through the actions of the LALR parser for the input string *iwiaewa*.

stack	input	output	action
$\$_0$	$iwiaewa\$$		

42. (i) Give a context-sensitive grammar for $\{a^n b^n c^n : n > 0\}$

(ii) Using that grammar, give a derivation of the string *aaabbbccc*.

43. Work Exercise 1 on the handout reglrNC.pdf.