University of Nevada, Las Vegas Computer Science 456/656 Fall 2024

Answers to Practice Problems for the Final Examination on December 11, 2024

`Mon Dec 9 02:53:07 PM PST 2024`

1. State the pumping lemma for regular languages.

   For any regular language L
   There exists a positive number p such that
   For any $w \in L$ of length at least p
   There exist strings x, y, z such that the following hold:
   1. $w = xyz$
   2. $|xy| \leq p$
   3. y is not empty
   4. For any $i \geq 0$, $xy^iz \in L$

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

   For any context-free language L
   There exists a positive number 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 hold:
   1. $w = uvxyz$
   2. $|vxy| \leq p$
   3. v and y are not both empty
   4. For any $i \geq 0$, $uv^ixy^iz \in L$

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

   Let Σ be the alphabet of L. Let $w_1, w_2, \ldots$ be the canonical order enumeration of Σ^*. The following program enumerates L in canonical order.

   For i from 1 to ∞
   if $w_i \in L$ write w_i

4. Prove that every language L which is enumerated in canonical order by some machine is decidable.

   The following program decides L. Let $w_1, w_2, \ldots$ be the canonical order of L, which we can use as an input to our program.

   Read w
   For all w_i in canonical order
   If $w = w_i$ halt and accept
   If $w < w_i$ (in the canonical order) halt and reject

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

   Let Σ be the alphabet of L. Let $w_1, w_2, \ldots$ be the canonical order enumeration of Σ^*. The following program enumerates L.

   For integers t from 1 to ∞
   For integers i from 1 to t
   If M accepts w_i within t steps, write w_i.

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

   Let $w_1, w_2 \ldots$ be an enumeration of L written by some machine. The following program accepts L.

   Read w
   For i from 1 to ∞
   If $w = w_i$ halt and accept

7. 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. $S \to \lambda$

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

   Assume the Dyck language L is regular. Let p be the pumping length of L. Let $w = a^p b^p \in L$. Then $|w| = 2p \geq p$. Thus there must exist strings x, y, z, such that all four conclusions of the pumping lemma hold. By 1. $xyz = w$. By 2., $|xy| \leq p$ Since xy is a prefix of w, it can contain no b. By 3., y is a non-empty string of b's. Write $y = b^k$ for some $k > 0$. By 4., choosing $i = 9$, we have $xz = a^{p-k}b^p \in L$, contradiction. Thus the Dyck language is not regular.

8. 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.
   K $\mathcal{RE}$ but not decidable.
   L co-$\mathcal{RE}$ but not decidable.
   M Neither $\mathcal{RE}$ nor co-$\mathcal{RE}$.

(i) D SAT.

(ii) D 3-SAT.

(iii) B 2-SAT.

(iv) D The Independent Set problem.

(v) D The Subset Sum Problem.

(vi) E All positions of RUSH HOUR from which it is possible to win.

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

(viii) A All base 10 numerals for perfect squares.

(ix) A The Dyck language.

(x) D 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) C Factorization of binary numerals.

(xii) K All C++ programs which halt with no input.

(xiii) B The Boolean Circuit Problem.

(xiv) E All configurations of RUSH HOUR from which it's possible to win.

(xv) K All pairs $(\langle M \rangle, w)$ such that M is a machine which halts with input w.

(xvi) All positions of generalized checkers (any size board) from which Black can force a win.

(xvii) A Does a square matrix with integer entries have determinan zero?

(xviii) D All satisfiable Boolean expressions.

(xix) B All binary numerals for composite integers. (Composite means not prime.)

(xx) A All binary numerals for multiples of 3

(xxi) A All binary numerals for square integers, that is $0, 1, 100, 1001, \ldots$

(xxii) C All pairs of binary numerals $(\langle p \rangle, \langle q \rangle)$ such that p has a divisor greater than q.

(xxiii) L All pairs of context-free grammars $(\langle G_1 \rangle, \langle G_2 \rangle)$ such that $L(G_1) = L(G_2)$.

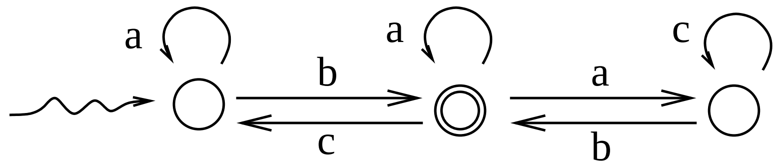
9. Give a definition of each term.

(a) Accept. (That is, what does it mean for a machine to accept a language.) A machine M accepts a language L if, given any string w as input, M will halt in an accepting state if and only if $w \in L$.

(b) Decide. (That is, what does it mean for a machine to decide a language.) A machine M decides a language L if, given any string w as input, M will halt in an accepting state if and only if $w \in L$, and will halt in a rejrecting state if and only if $w \notin L$.

(c) Canonical order of a language L. For any $u, v \in L$, we define $u < v$ if one of the following two conditions holds:
$|u| < |v|$
$|u| = |v|$ and u comes before v in alphabetic order.

10. Give a regular expression for the language accepted by the machine in the figure below.



There are many correct answers, but I believe the simplest is $a^*b(a + ca^*b + ac^*b)^*$.

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

    (a) Deterministic finite automata. Regular languages.

    (b) Non-deterministic finite automata. Regular languages.

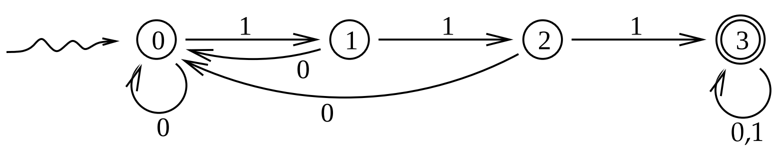
    (c) Push-down automata. Context-free languages.

    (d) Limited push-down automata. (LPDA) Regular languages.

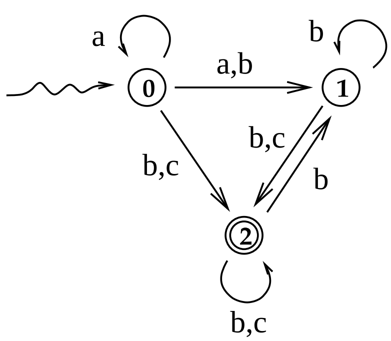
    An LPDA is a PDA with a limit on its stack size. That is, there is some integer D such that the stack can hold no more than D symbols.

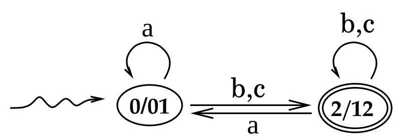
    (e) Turing Machines. i Recursively enumerable languages.

12. Let L be the binary language which consists of all strings which contain the substring 111. Construct a DFA which accepts L.



13. Construct a minimal DFA equivalent to the NFA shown below.





14. The grammar below is an ambiguous CF grammar and is parsed by the LALR parser whose Action and Goto tables are shown. Write a computation of the parser for the input string *iiwaea*.

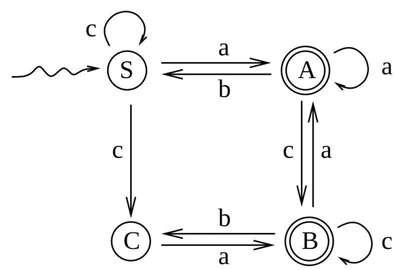
1. $S \to i_2 S_3$
2. $S \to i_2 S_3 e_4 S_5$
3. $S \to w_6 S_7$
4. $S \to a_8$

	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			$r4$		$r4$	

$\$_0$	$iiwaea\$$		
$\$_0 i_2$	$iwaea\$$		$s2$
$\$_0 i_2 i_2$	$waea\$$		$s2$
$\$_0 i_2 i_2 w_6$	$aea\$$		$s6$
$\$_0 i_2 i_2 w_6 a_8$	$ea\$$		$s8$
$\$_0 i_2 i_2 w_6 S_7$	$ea\$$	4	$r4$
$\$_0 i_2 i_2 S_3$	$ea\$$	43	$r3$
$\$_0 i_2 i_2 S_3 e_4$	$a\$$	43	$s4$
$\$_0 i_2 i_2 S_3 e_4 a_8$	\$	43	$s8$
$\$_0 i_2 i_2 S_3 e_4 S_5$	\$	434	$r4$
$\$_0 i_2 S_3$	\$	4342	$r2$
$\$_0 S_1$	\$	43421	$r1$
HALT			

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

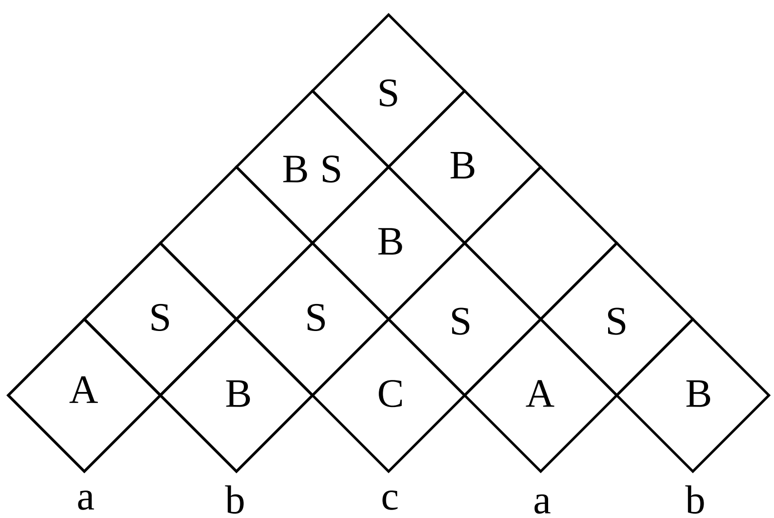
$S \to aA|cS|cC$
$A \to aA|bS|cB|\lambda$
$B \to aA|cB|bC|\lambda$
$C \to aB$



16. Use the CYK algorithm to decide whether *abcab* is generated by the CNF grammar below by filling in the matrix.

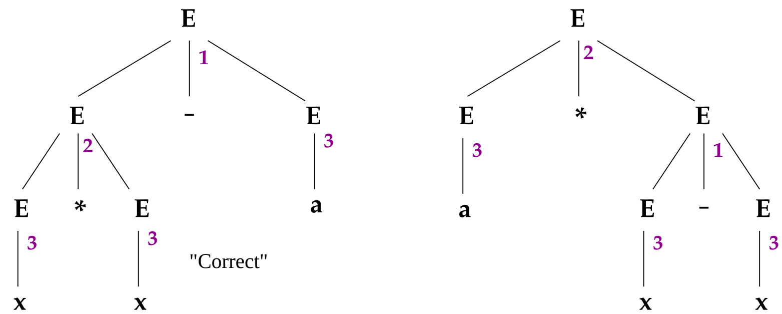
$S \to AB \,|\, BC \,|\, CA$
$A \to a$
$B \to SA \,|\, SS \,|\, b$
$C \to c$

The start symbol is in the top cell, hence *abcab* is generated by the grammar.



17. Consider the grammar G below.
    1. $E \to E -_2 E_3$
    2. $E \to E *_4 E_5$
    3. $E \to x_6$

    Prove that G is ambiguous by giving two different parse trees for $x - x * x$.
    One of those parse trees is the "right" one, meaning that it respects the standard precedence of operators.
    The LALR parser given below will not give the "right" parse tree. Indicate the changes you need to make to correct this problem.



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

Replace the entry s2 in row 5, column "$-$" by r2.
Replace the entry r1 in row 4, column "$*$" by s4.

18. Give the verifier-certificate definition of the class $\mathcal{NP}$.

    A language L is in $\mathcal{NP}$ if there exists a machine V and a constant k such that, for any string $w \in L$ of length n, there exists a string c, the certificate, such that with input w and c, V will halt and accept in time which is polynomial in n. Furthermore, if $w \notin L$, no certificate exists.

19. What is the importance nowadays of $\mathcal{NC}$?

    Many new machines have many processers working in parallel. If a problem is $\mathcal{NC}$, the equired work can be efficiently assigned to different processors, enabling the problem to be worked faster.

20. What complexity class contains sliding block problems?

    $\mathcal{P}$–SPACE

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

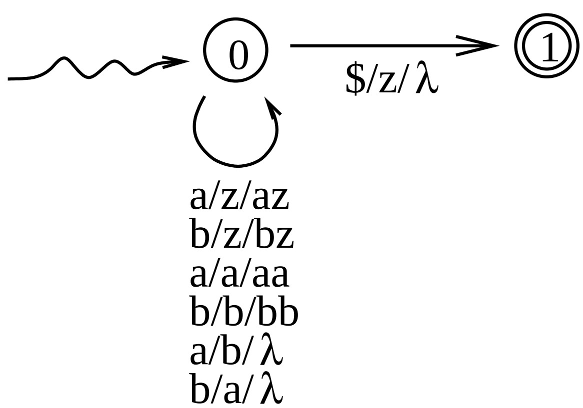
    Let $E = C_1 * C_2 * \cdots C_k$ be Boolean expression in 3–CNF form For any i, let $C_i = t_{i,1} + t_{i,2} + t_{i,3}$ where each $t_{i,p}$ is either a variable or the negation of a variagle. Let G be the graph with $3k$ vertices $\{v_{i,j}\}$ each labeled with one term of E. Let there be an edge from $v_{i,p}$ to $v_{j,q}$ if either $i = j$ or $t_{i,p} * t_{j,q}$ is a contradiction. Then E is satisfiable if and only if G has an independent set of order k.

22. 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. $E \to x_{11}$

	ACTION						GOTO
	x	$-$	$*$	(	)	\$	S
0	$s11$	$s4$		$s8$			1
1		$s2$	$s6$			halt	
2	$s11$	$s4$	$s4$	$s8$			3
3		$r1$	$s6$		$r1$	$r1$	
4	$s11$	$s4$		$s8$			5
5		$r2$	$r2$		$r2$	$r2$	
6	$s11$	$s4$		$s8$			7
7		$r3$	$r3$		$r3$	$r3$	
8	$s11$	$s4$		$s8$			9
9		$s2$	$s6$		$s6$		
10		$r4$	$r4$	$r4$	$r4$	$r4$	
11		$r5$	$r5$		$r5$	$r5$	

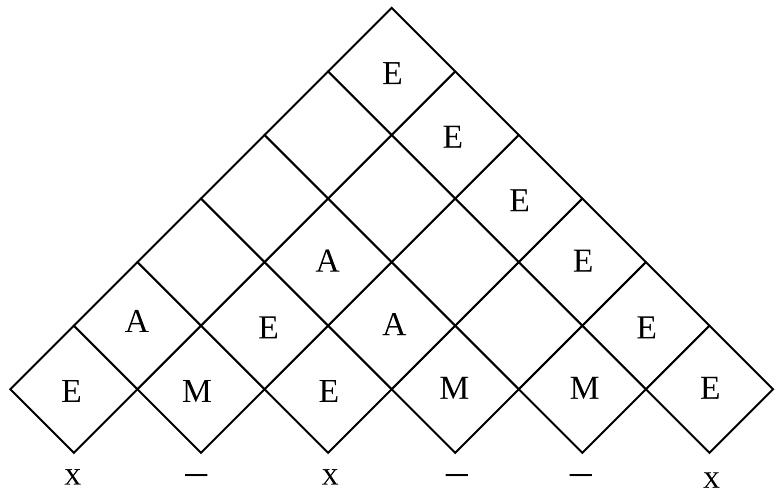
23. Let $L = \{w \in \{a,b\}^* : \#_a(w) = \#_b(w)\}$, that is, each string of L has equal numbers of each symbol. Draw a DPDA which accepts L.



24. 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.

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

The string $x - x - -x$ is generated by the grammar,since the top cornej of the figure contains the start symbol.



25. 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. $E \to x_8$

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

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

If $X = (x_1, x_2, \ldots x_n, K)$ is an instance of the subset sum problem, let $S = \sum_{i=1}^{n} x_i$. Then $Y = (x_1, x_2, \ldots x_n, K + 1, S - K + 1)$ is an instance of the partition problem which has a solution if and only if X has a solution.

27. Prove that the halting problem is undecidable.

By constradiction. Assume HALT is decidable. Let P be the following program

Read a machine description $\langle M \rangle$.
If M halts with input $\langle M \rangle$, enter an infinite loop.
Else halt.

The input to P could be $\langle P \rangle$. If so, does it halt?

If P halts with input $\langle P \rangle$, then, with input $\langle P \rangle$, the code will cause P to enter an infinite loop, that is, not halt, contradiction.

Alternatively, if P does not halt with input $\langle P \rangle$, then, with intput $\langle P \rangle$, the code will cause P to halt, contradiction.

In either case, we have a contradiction. We conclude that HALT is not decidable.

29. Label each of the following sets as countable or uncountable.

    (a) **countable** The set of integers.
    (b) **countable** The set of rational numbers.
    (c) **uncountable** The set of real numbers.
    (d) **uncountable** The set of binary languages.
    (e) **countable** The set of co-$\mathcal{RE}$ binary languages.
    (f) **uncountable** The set of undecidable binary languages.
    (g) **uncountable** The set of functions from integers to integers.
    (h) **countable** The set of recursive real numbers.

30. What is the Church-Turing Thesis? Why is it important?

    The thesis states that any computation that is done by any machine is also done by some Turing machine. Thus, a proof that a particular computation is impossible requires only a proof that no Turing machine can do that computation.