CSC 456/656 Spring 2023 Answers to Final Examination May 10

Name:_____

The entire test is 475 points.

No books, notes, scratch paper, or calculators. Use pen or pencil, any color. Use the rest of this page and the backs of the pages for scratch paper. If you need more scratch paper, it will be provided.

1. True or False. T = true, F = false, and O = open, meaning that the answer is not known science at this time. [5 points each] In the questions below, $\mathcal{P}$ and $\mathcal{NP}$ denote $\mathcal{P}$-TIME and $\mathcal{NP}$-TIME, respectively.

   (i) **F** 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) **F** The intersection of any two context-free languages is context-free.

   (iii) **T** Every language accepted by a non-deterministic machine is accepted by some deterministic machine.

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

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

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

   (vii) **F** Every problem that can be mathematically defined has an algorithmic solution.

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

   (ix) **O** $\mathcal{NC} = \mathcal{P}$.

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

   (xi) **T** The Boolean Circuit Problem (also known as the CVP problem) is in $\mathcal{P}$.

   (xii) **O** The Boolean Circuit Problem (CVP) is $\mathcal{NC}$.

   (xiii) **T** co-$\mathcal{P} = \mathcal{P}$.

   (xiv) **T** 2-SAT is $\mathcal{P}$-TIME.

   (xv) **O** 3-SAT is $\mathcal{P}$-TIME.

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

   (xvii) **T** Every context-free language is $\mathcal{NC}$.

   (xviii) **T** Addition of binary numerals is $\mathcal{NC}$.

   (xix) **F** Every language generated by an unrestricted (general) grammar is recursive.

(xx) **T** 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, L_2 must be undecidable.

(xxi) **O** If L is $\mathcal{NP}$ and also co-$\mathcal{NP}$, then L must be $\mathcal{P}$–TIME.

(xxii) **T** If L is in $\mathcal{RE}$ and also co-$\mathcal{RE}$, then L must be recursive (decidable).

(xxiii) **T** There is a mathematical proposition that is true but cannot be proved true.

(xxiv) **T** There is a non-recursive function which grows faster than any recursive function.

(xxv) **T** There is a Turing machine which, when turned on, runs forever, writing the decimal expansion of π (the well-known ratio of the circumference of a circle to its diameter).

(xxvi) **T** The binary integer factorization problem is co-$\mathcal{NP}$.

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

(xxviii) **T** The intersection of any two $\mathcal{NP}$ languages is $\mathcal{NP}$.

(xxix) **T** The intersection of any two co-$\mathcal{NP}$ languages is co-$\mathcal{NP}$.

(xxx) **T** Every language accepted by a non-deterministic machine is accepted by some deterministic machine.

(xxxi) **T** If S is a set of positive integers and if $\{1^n : n \in S\}$ is a recursive language, then $\sum_{n \in S} 2^{-n}$ must be a recursive real number.

(xxxii) **T** Multiplication of matrices with binary numeral entries is $\mathcal{NC}$.

(xxxiii) **T** Every recursively enumerable language is generated by an unrestricted (general) grammar.

(xxxiv) **T** Equivalence of context-free grammars is co–$\mathcal{RE}$.

(xxxv) **F** The language of all true mathematical statements is recursively enumerable.

(xxxvi) **T** The language of all **provably** true mathematical statements is recursively enumerable.

(xxxvii) **T** There are uncountably many undecidable languages over the binary alphabet.

(xxxviii) **T** If anyone ever finds a polynomial time algorithm for any $\mathcal{NP}$–complete problem, then $\mathcal{P} = \mathcal{NP}$.

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

(xl) **F** 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 co-$\mathcal{RE}$.

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

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

(xliii) **T** If a Boolean expression is satisfiable, there is a $\mathcal{P}$–TIME proof that it's satisfiable.

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

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

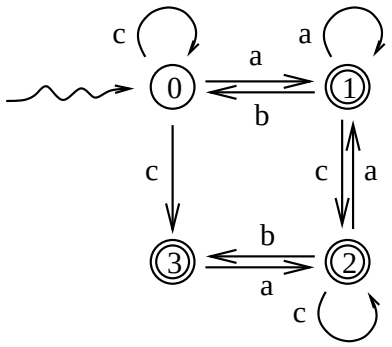
2. We did not cover CS grammars this semester.

   (i) [10 points] Give a context-sensitive grammar for $\{a^n b^n c^n : n > 0\}$

   (ii) [5 points] Using that grammar, give a derivation of the string *aaabbbccc*.

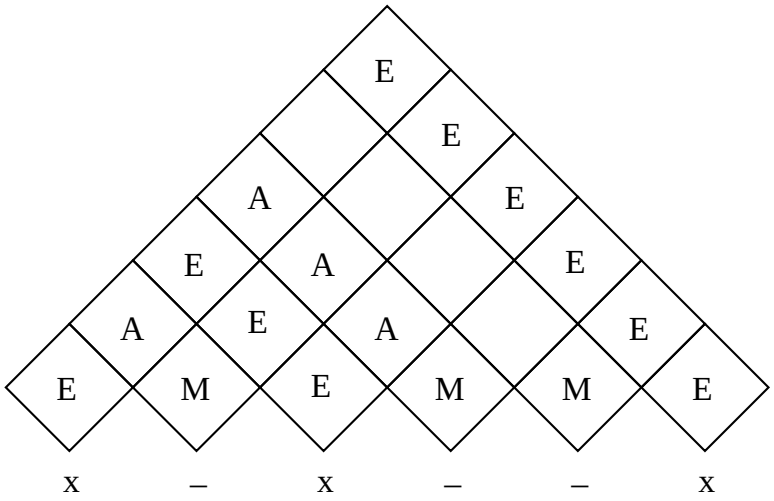
3. [10 points] Illustrate an NFA which accepts the language generated by this grammar.

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



4. [20 points] Use the CYK algorithm to decide whether $x - x - -x$ is generated by the CNF grammar below, by filling in the matrix.

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



5. [10 points] What is the importance nowadays of $\mathcal{NC}$?

   Machines with many processors have become more common, and it is important to consider which problems can be solved with programs that make efficient use of that parallel structure.

6. [20 points] Give a polynomial time reduction of the subset sum problem to the partition problem. You may assume all the numbers are positive.

   Let $\alpha = (x_1, x_2, \ldots x_n, K)$ be an instance of the subset sum problem. We reduce α to an instance β of the partition problem which has a solution if and only if α has a solution.

Let $S = \sum_{i=1}^{n}$. If $S < K$, α has no solution, so we let $\beta = (1)$, an instance of the partion problem that, trivially, has no solution. Otherwise, we let $\beta = (x_1, x_2, \ldots x_n, K+1, S-K+1)$.

7. Label each of the following sets as countable or uncountable. [5 points each]

   (i) **countable** The set of integers.

   (ii) **countable** The set of rational numbers.

   (iii) **uncoutable** The set of real numbers.

   (iv) **uncountable** The set of binary languages.

   (v) **countable** The set of co-RE binary languages.

   (vi) **uncountable** The set of undecidable binary languages.

   (vii) **uncountable** The set of functions from integers to integers.

   (viii) **countable** The set of recursive real numbers.

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

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

8. [20 points] Give a regular expression for the language accepted by the machine in Figure 1

   $a^*b(ca^*b + a + ac^*b)^*$

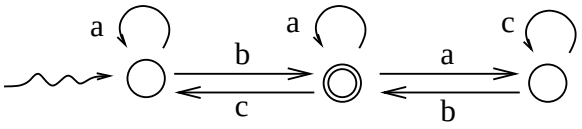
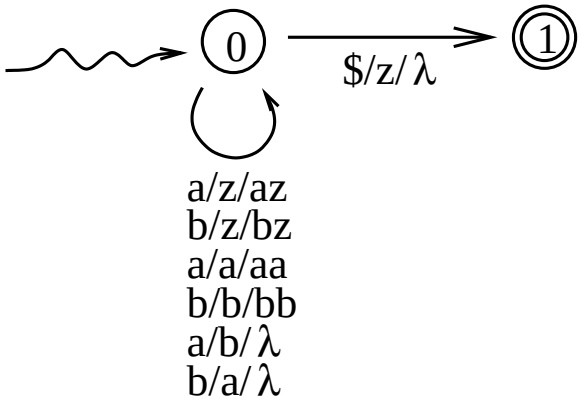


Figure 1: NFA for problem 8.

9. [20 points]

   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 is a string, and \$ is the end-of-file symbol.) Draw a PDA which accepts L.



10. [20 points] 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		? $r1$	? $s6$	$r1$	
4	$s8$	$s4$			5
5		? $r2$	? $r2$	$r2$	
6	$s8$	$s4$			7
7		? $r3$	? $r3$	$r3$	
8	$s8$	$r4$	$r4$	$r4$	

11. Each of the languages listed below falls into one of these categories. Indicate which for each language. [5 points each]

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.
K 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) **A** The Dyck language.

(vii) **H** The set of all (descriptions of) grammars which generate the Dyck language. That is, all $\langle G \rangle$ such that $L(G)$ is the Dyck language.

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

(ix) **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.)

(x) **C** Factorization of binary numerals.

12. [20 points] Prove that the halting problem is undecidable.

By contradiction. Suppose the halting problem is decidable. Let L_{diag} be the set of programs which do not halt if given themselves as input. The following program, which we call P_{diag}, accepts L_{diag}:

```
read a program P
if P halts with input P
    Enter an infinite loop
else HALT
```

Note that the condition in the second line of P_{diag} can always be evaluated because the halting problem is decidable.

We now ask, is P_{diag} a member of L_{diag}? The answer cannot be either yes or no. Thus, the halting problem is undecidable.