University of Nevada, Las Vegas
Computer Science 456/656 Fall 2020

Practice for Final Examination December 8, 2021

The entire practice examination is 475 points. The real examination will be shorter.

1. True/False/Open See `tfI.pdf` and `tfII.pdf` for more T/F questions.

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

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

   (iii) **F** $\mathcal{P}$–SPACE = EXP–SPACE.

   (iv) **F** Given a regular expression, an equivalent minimal DFA can always be constructed in polynomial time.

   (v) **T** $L = \{0^n 1^n 0^n 1^n \,:\, n \geq 1\}$ is context-sensitive.

   (vi) **F** The intersection of two context-free lanuages must be context-free.

   (vii) **T** The concatenation of two context-free languages must be context-free.

   (viii) **T** The concatenation of two context-sensitive languages is context-sensitive.

   (ix) **F** The intersection of two co-RE languages is co-RE.

   (x) **T** Suppose a language L has matching delimeters. That is, its alphabet contains symbols ℓ and r, such that, in each $w \in L$, any instance of ℓ must be uniquely matched with an instance or r to its right. Then it is impossible for L to be regular.

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

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

   (xiii) **T** Every language which is generated by a general grammar is recursively enumerable.

   (xiv) **O** The set of all binary numerals for prime numbers is in the class $\mathcal{P}$.

   (xv) **F** If L_1 reduces to L_2 in polynomial time, and if L_1 is $\mathcal{NP}$, and if L_2 is $\mathcal{NP}$-complete, then L_1 must be $\mathcal{NP}$-complete.

   (xvi) **T** The union of any two context-free languages is context-free.

   (xvii) **T** The class of languages accepted by non-deterministic Turing machines is the same as the class of languages accepted by deterministic Turing machines.

   (xviii) **F** The class of languages accepted by non-deterministic push-down automata is the same as the class of languages accepted by deterministic push-down automata.

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

   (xx) **T** If L_1 reduces to L_2 in polynomial time, and if L_2 is $\mathcal{NP}$, then L_1 must be $\mathcal{NP}$.

   (xxi) **F** The language of all regular expressions over the binary alphabet is a regular language.

(xxii) **T** Let $e = \sum_{i=0}^{\infty} \frac{1}{i!} = 2.71828\ldots$, the base of the natural logarithm. The problem of whether the n^{th} digit of e, for a given n, is equal to a given digit is decidable.

(xxiii) **T** Every regular language is in the class $\mathcal{NC}$

(xxiv) **T** (Recall that $\langle x \rangle$ is the binary numeral for an integer x.) Let $x_1, x_2, \ldots$ be an arithmetic sequence of integers. The language $\{\langle x_i \rangle\}$ is regular.

(xxv) **F** (Recall that $\langle x \rangle$ is the binary numeral for an integer x.) Let $y_1, y_2, \ldots$ be a geometric sequence of integers. The language $\{\langle y_i \rangle\}$ is regular.

(xxvi) **O** The language of all binary strings which are the binary numerals for prime numbers is in the class $\mathcal{P}$–TIME.

(xxvii) **T** Every context-free grammar can be parsed by some non-deterministic top-down parser.

(xxviii) **F** If anyone ever proves that the integer factorization problem is $\mathcal{P}$–TIME, all public key/private key encryption systems will be known to be insecure.

(xxix) **T** If a string w is generated by a context-free grammer G, then w has a unique leftmost derivation if and only if it has a unique rightmost derivation.

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

(xxxi) **T** If there is an $\mathcal{NC}$ reduction from L_1 to L_2, and if L_2 is in Nick's class, then L_1 must be in Nick's class.

2. Every language, or problem, falls into exactly one of these categories. For each of the languages, write a letter indicating the correct category. [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 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-$\mathcal{RE}$.

(a) **D** 3-CNF-SAT (usually known simply as 3-SAT)

(b) ________ 2-CNF-SAT (usually known simply as 2-SAT)

(c) **K** Halting problem.

(d) **B** Boolean circuit problem.

(e) ________ Regular grammar equivalence.

(f) **L** Context-free grammar equivalence.

(g) **E** Regular expression equivalence.

(h) **E** General sliding block problem.

(i) **F** Generalized checkers (any size rectangular board).

(j) **A** DFA equivalence.

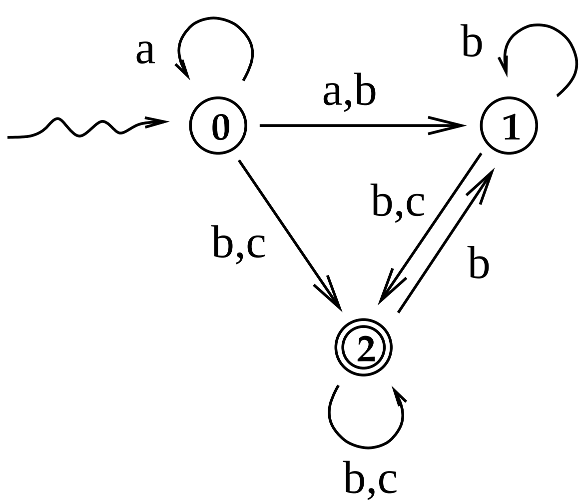
(k) **A** All fractions whose values are less than $\sqrt{6}$.

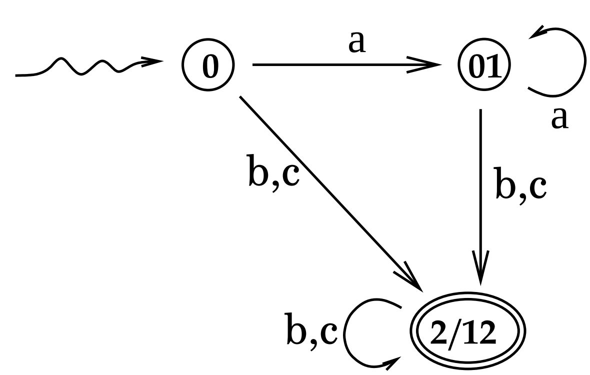
(l) **A** $\{a^n b^n c^n d^n : n \geq 1\}$

(m) **D** You have a number of containers, each of a given size and shape. You also have a set of objects of various sizes and shapes. Can you fit all the objects into the containers?

(n) **B** Dynamic programming, where each subproblem works in constant time and has Boolean output.

3. [20 points] Find a minimal DFA equivalent to the NFA shown below.





4. [20 points] Give a regular expression for the language accepted by the machine in Figure 1

   $a^*bb^*a(a + (b + c)b^*a)^*$

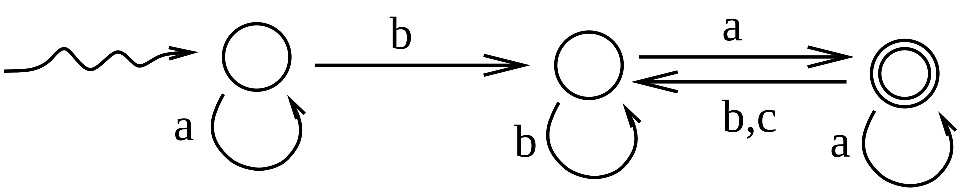


Figure 1: DFA for problem 4.

5. Which class of languages does each of these machine classes accept? [5 points each]

   (a) Deterministic finite automata.

   Regular languages

   (b) Non-deterministic finite automata.

   Regular languages

   (c) Push-down automata.

   Context-free languages

   (d) Turing Machines.

   Recursively enumerable languages

6. [20 points] The output of an LALR parser corresponds to the (pick one)

   (a) preorder

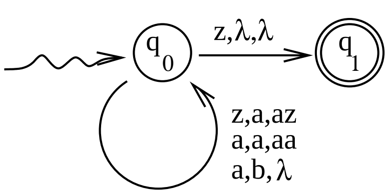
   (b) postorder

   (c) reverse preorder

   (d) reverse postorder

   postorder visitation of the internal nodes of the parse tree.

7. [20 points] Design a PDA that accepts the Dyck language, whose grammar is given in problem 8.



8. [20 points]
The grammar below is an unambiguous CF grammar for the Dyck language, 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 $aabb$.

1. $S \to S_{1,3}\, a_2 S_3 b_4$
2. $S \to \lambda$

	a	b	\$	S
0	$r2$		$r2$	1
1	$s2$		halt	
2	$r2$	$r2$		3
3	$s2$	$s4$		
4	$r1$	$r1$	$r1$	

$\$_0$	$aabb\$$	
$\$_0 S_1$	$aabb\$$	2
$\$_0 S_1 a_2$	$abb\$$	2
$\$_0 S_1 a_2 S_3$	$abb\$$	22
$\$_0 S_1 a_2 S_3 a_2$	$bb\$$	22
$\$_0 S_1 a_2 S_3 a_2 S_3$	$bb\$$	222
$\$_0 S_1 a_2 S_3 a_2 S_3 b_4$	$b\$$	222
$\$_0 S_1 a_2 S_3$	$b\$$	2221
$\$_0 S_1 a_2 S_3 b_4$	\$	2221
$\$_0 S_1$	\$	22211
HALT		

I will definitely give one of the following four proofs on the final exam.

9. [20 points] Prove that a recursively enumerable language is accepted by some machine.

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

Read w.
For all i from 1 to ∞
  if($w_i = w$) accept and halt

10. [20 points] Prove that any language accepted by a machine is recursively enumerable.

The following program enumrates L. Let $w_1, w_2, \ldots$ be the enumeration of Σ^* in canonical order. Let M be a machine which accepts L. The following program enumerates L.

For integers t from 1 to ∞
  For integers i from 1 to t
    If M accepts w_i is at most t steps
      Write w_i

11. [20 points] Prove that any decidable language is enumerated in canonical order by some machine.

Let $w_1, w_2, \ldots$ be the enumeration of Σ^* in canonical order. Let M be a machine which decides L. The following program enumerates L in canonical order.

For integers i from 1 to ∞
  If M accepts w_i Write w_i

12. [20 points] Prove that any language which can be enumerated in canonical order by some machine is decidable.

There are two cases.

Case 1. L is finite. Let $\{w_1, w_2, \ldots w_n\}$ be the members of L The following program decides L.

Read w.
For all i from 1 to n

if($w_i = w$) accept and halt
Reject.

Case 2. L is infinite. Let $w_1, w_2, \ldots$ be a canonical order enumeration of L computed by some machine. The following program decides L.

Read w.
For all i from 1 to ∞
if($w_i = w$) accept and halt
else if($w_i > w$) in the canonical order of Σ^* reject and halt

13. [20 points] Prove that the halting problem is undecidable. This question will definitely be on the final exam.

I will accept any correct proof, provided it's written correctly. Here is my proof, which is by contradiction.

Assume the halting problem is decidable, that is $HALT = \{\langle M\rangle w : M \text{ accepts } w\}$ is decided by some machine M_{HALT}.

Let DIAG $= \{\langle M\rangle : \langle M\rangle\langle M\rangle \notin \text{HALT}\}$ Let M_{DIAG} be a machine equivalent to the following program:

Read w
If w is the encoding of a machine and $ww \notin$ HALT accept
Else reject

M_{DIAG} decides DIAG.

Statement 1: $\langle M_{\text{DIAG}}\rangle \in$ DIAG if and only if M_{DIAG} does not accept $\langle M_{\text{DIAG}}\rangle$, by the definition of DIAG.

Statement 2: $\langle M_{\text{DIAG}}\rangle \in$ DIAG if and only if M_{DIAG} accepts $\langle M\rangle$, by the definition of M_{DIAG}

Contradiction. Therefore HALT is not decidable.

I will definitely give one of the following two reductions on the final exam.

14. [20 points] Give a polynomial time reduction of 3-SAT to the independent set problem.

Let E be a Boolean expression in 3-CNF form. That is, $E = C_1 * C_2 * \cdots * C_k$, where for each $1 \leq i \leq k$, $C_i = t_{i,1} + t_{i,2} + t_{i,3}$ where each term $t_{i,j}$ is either a Boolean variable or the negation of a Boolean variable.

Let $G = (V, E)$ be a graph where V has $3k$ vertices, named $v_{i,j}$ for all $1 \leq i \leq k$ and $j \in 1, 2, 3$, E consists of pairs $\{v_{i,1}, v_{i,2}\}$, $\{v_{i,1}, v_{i,3}\}$, $\{v_{i,2}, v_{i,3}\}$, for all i, and $\{v_{i,j}, v_{i',j'}\}$ for all i, j, i', j' such that $t_{i,j} + t_{i',j'}$ is a contradiction. Then E has an independent set of order k if and only if E is satisfiable.

15. [20 points] Give a polynomial time reduction of the subset sum problem to the partition problem.

$\mathcal{I} = (K, x_1, x_2, \ldots x_n)$, an instance of the subset sum problem and define

$$R(\mathcal{I}) = (x_1, x_2, \ldots x_n, K + 1, S - K + 1)$$

an instance of the partition problem, where $S = x_1 + x_2 + \cdots + x_n$. The sum of the sequence $R(\mathcal{I})$ is $2S + 2$, and $R(\mathcal{I})$ has a subsequence which totals $S + 1$ if and only if there is a subsequence of $x_1, \ldots x_n$ whose total is K, and thus R is a reduction of the subset sum problem to the partition problem.