**CS 501: Formal Language Theory** **Spring 2023**

# Final Examination

Released: 5/24/2023, 6:00 pm EST   Time Limit: 120 minutes   Due: 5/24/2023, 8:00 pm EST

**Note**: LaTeX template courtesy of UC Berkeley EECS dept.

**Instructions.** This final contains seven questions on pages 1-9, for a total of 100 points. You have a total of 120 minutes to complete it. There will be a supplemental sheet with some definitions on it.

The final is an **individual effort**. You are required to write your entire attempt yourself, and are forbidden from consulting anyone else. Failure to abide by this will result in immediate failure from the course, among other consequences.

- This is a closed-book exam, with no books, notes, calculators, or collaboration.

**Submissions.** Please write your answers on the test sheet. You may use the backs of pages, but let us know in the indicated place for each question where we can find the rest of your answer. Page 10 and 11 is a supplemental sheet with useful information – do not put answers on it.

---

1. **(10 $\times$ 2 points) Unjustified True/False Questions.** For each of the following questions, indicate simply whether it is TRUE or FALSE. *No justification needed or wanted.*

   (a) Let $\Sigma = \{0, 1\}$. There exists a bijection from all strings over $\Sigma$ to all languages over $\Sigma$.

   (b) If $X$ is a context-free language over the alphabet $\{a, b, c, \ldots, z\}$ that contains all of the strings $a^n b^n c^n$ for all $n$ with $n \geq 0$, then it is not the case that it must contain at least one other string in $a^* b^* c^*$.

   (c) If $Y$ is any infinite TR language, and $n$ is any natural, it is possible that there does not exist a string in $Y$ that is compressible by $n$ bits.

   (d) Let $Z$ be a language in the class $NC^1$. Then there exists a family of boolean *formulas* $\{C_n : n \leq 0\}$ such that each $C_n$ decides membership in $Z$ for strings of length $n$, the depth of $C_n$ is $O(\log n)$, the size of $C_n$ is polynomial in $n$, and its gates are binary AND, binary OR, and unary NOT. Note that a circuit is a **formula** if it is a tree, meaning that a given gate may feed into only one other gate.

   (e) Let $D_7$-SAT be the set of satisfiable formulas with depth at most 7, unbounded fan-in OR and NOT gates, and unary NOT gates. Then, like CIRCUIT-SAT, $D_7$-SAT is NP-complete (assuming $P \neq NP$).

   (f) It is not the case that the class L is a strict subset of the context-free languages.

   (g) It is not the case that the class of context-free languages is closed under union, concatenation, Kleene star, and intersection.

   (h) Let SUBGRAPH-ISO be the set of two undirected graphs $G$ and $H$ where $G$ has a subgraph isomorphic to $H$. Then SUBGRAPH-ISO is NP-complete (assuming $P \neq NP$).

   (i) If $X$ is a language in $NP$, then it is possible that the language $X^*$ is not in $NP$.

   (j) Using the Recursion Theorem, we can build a Turing machine $B$ where we can obtain its own description, find out whether it accepts, and reverse the answer.

2. **(5 × 6 points) Justified True/False Questions.** For each of the following questions, indicate whether it is TRUE or FALSE, *and provide a brief justification (i.e. either a proof or a counterexample).*

   (a) The language $X = \{\langle M \rangle : M$ is a single tape TM and there exists a number $k$ such that $M$ never moves right past the $k$'th cell, on any input$\}$ is not TR.

   (b) The language $X$ from Question 2(a) is TD.

(c) Let $f$ be the function that takes a binary number and outputs its unary representation. Then $f$ is a polynomial-time computable function if and only if $P = NP$.

(d) On Homework #2, we proved the given any directed graph $G$ and any nodes $s$ and $t$, we can construct a grammar whose language is non-empty if and only if there is a path from $s$ to $t$. Then, assuming that this construction can be carried in logspace, and if the language $E_{CFL}$ were in the class $L$, it would still be possible that $L$ is a strict subset of $NL$ (assuming $L \neq NL$).

(e) If every language in the class $P$ can be decided by a polynomial-size log-space uniform family of circuits, then $L = P$ must be true, even assuming $L \neq P$.

3. **(10 points) MAX-2SAT.** The language MAX-2SAT is the set of all pairs $\langle \phi, k \rangle$ where $\phi$ is a 2-CNF formula (a set of clauses, each the OR of one or two literals) and there exists a setting of the variables making $k$ or more of the clauses true. Prove that MAX-2SAT is NP-complete. (**Hint:** Reduce from 3-SAT. For any clause $\psi$ in the 3-CNF formula, with literals $\ell_1$, $\ell_2$, and $\ell_3$, let $c$ be a new variable and consider the ten 2-CNF clauses $\{\ell_1, \ell_2, \ell_3, c, \neg\ell_1 \vee \neg\ell_2, \neg\ell_1 \vee \neg\ell_3, \neg\ell_2 \vee \neg\ell_3, \ell_1 \vee \neg c, \ell_2 \vee \neg c, \ell_3 \vee \neg c\}$. How many of these clauses are satisfied if $\psi$ is satisfied? How many could be satisfied if $\psi$ is not satisfied? Note that the list of clauses above is symmetric in $\ell_1$, $\ell_2$, and $\ell_3$, which might make counting easier.)

4. **(10 points) Bottleneck Sets.** Let BOTTLENECK-SET be the language $\{\langle G, s, t, S\rangle : G$ is a directed graph, $s$ and $t$ are nodes, and for any node $x$ in $G$, (there is a path from $s$ to $t$ without using $x$) if and only if $x \notin S$ $\}$. Hence $S$ is exactly those nodes that can be deleted, individually, to prevent the path from $s$ to $t$. Prove that BOTTLENECK-SET is complete for the class NL under log-space reductions.

5. **(10 points) Equivalence of Regular Expressions.** Prove that the language $EQ_{REG}$ is PSPACE-complete under poly-time reductions.

6. **(10 points) Rotational Closure.** If $U$ is any language, the **rotational closure** of $U$, called $RC(U)$, is that set of all strings $w$ such that there exist strings $x$ and $y$, $w = yx$, and $xy \in U$. For example, if $U = \{abc\}$, $RC(U) = \{abc, bca, cab\}$. A class $C$ of languages is said to be **closed under rotational closure** if $RC(U) \in C$ whenever $U \in C$.

   (a) (6) Prove that the class of regular languages is closed under rotational closure.

   (b) (2) Prove that the class $P$ is closed under rotational closure.

   (c) (2) Prove that the class of TD languages is closed under rotational closure.

7. **(10 points) LINEAR-SAT.** Define a **linear TM** to be a Turing machine $M$, with two tapes, and a constant $c$ such that $M$ always halts within $cn$ steps on any input of length $n$. Define the language LINEAR-SAT to be the set of all linear TM's $M$ such that there exists some string $w$ that $M$ accepts. What is the complexity of the language LINEAR-SAT?

## Supplemental Sheet for COMPSCI 501 Final Exam, Spring 2022

**Language** $A_{TM}$ (similarly $A_{REG}$, etc.) Set of pairs $(M, w)$ such that $M$ accepts $w$

**Language** $ALL_{TM}$**:** (similarly $ALL_{REG}$, etc.) Set of machines that accept all possible strings over their alphabet

**Circuit Complexity:** $PSIZE$ is all languages $X$ for which there is a uniform family of circuits, where circuit $C_n$ decides membership in $X$ for strings of length $n$, and $C_n$ has $n^{O(1)}$ nodes. If the circuit depth is also bounded by $O(\log^i n)$, the language is in the class $NC^i$ (if the circuits have fan-in two) or $AC^i$ (if the circuits have unbounded fan-in).

**Computable Function:** Function $f$ from strings to strings such that some Turing machine, on any input $w$, always halts with $f(w)$ on its tape

**Context-Free Grammar (CFG):** Grammar where rules allow single non-terminals to be replaced by strings

**Context-Free Language (CFL):** Definable by a context-free grammar or a PDA

**co-TR:** A language $A$ is co-TR if and only if its complement $\overline{A}$ is TR (similarly co-$NP$, etc.)

**Language** $E_{TM}$**:** (similarly $E_{REG}$, etc.) Set of machines with empty languages

**Language** $EQ_{TM}$**:** (similarly $EQ_{REG}$, etc.) Set of pairs of machines whose languages are equal

**Log-Space Reduction (**$\leq_L$**):** Like mapping reduction except that the function $f$ is log-space computable

**Mapping Reduction (**$\leq_m$**):** $A \leq_m B$ means that there exists a computable function $f$ such that $\forall w : w \in A \leftrightarrow f(w) \in B$

**NP-Complete Languages:** You may assume without proof that all these are $NP$-complete:

- $SAT$, the set of satisfiable boolean formulas
- 3-$SAT$, the set of 3-CNF boolean formulas
- $CIRCUIT$-$SAT$, the set of satisfiable boolean circuits
- $CLIQUE$, the set of pairs $(G, k)$ with a $k$-clique in the undirected graph $G$
- $VC$, the set of pairs $(G, k)$ where there exists a $k$-vertex cover in the undirected graph $G$
- $HAMPATH$, the set of directed graphs $G$ with a path from one node to another using every node once
- $UHAMPATH$, the set of undirected graphs $G$ with such a path
- $SUBSET$-$SUM$, the set of sequences $(s_1, \ldots, s_k, t)$ where there is a subset of the $s_i$'s summing to $t$

**Poly-Time Reduction (**$\leq_p$**):** Like mapping reduction except the function $f$ is poly-time computable

**Pushdown Automaton (PDA):** Nondeterministic finite-state machine with an added stack

**REACH:** The language $\langle G, s, t \rangle$ such that $G$ is a directed graph in which there is a path from node $s$ to node $t$

**Regular Language:** Can be defined by a DFA, NFA, or regular expression

**Space Complexity:** $DSPACE(f)$ is the set of languages decided by a TM using $O(f)$ steps, $NSPACE$ similarly for NDTM's, $PSPACE = \cup_k DSPACE(n^k)$, $NPSPACE = \cup_k NSPACE(n^k)$, $L = DSPACE(\log n)$, $NL = NSPACE(\log n)$

**Time Complexity:** $DTIME(f)$ is the set of languages decided by a TM using $O(f)$ steps, $NTIME$ similarly for NDTM's, $P = \cup_k DTIME(n^k)$, $NP = \cup_k NTIME(n^k)$

**Turing Decidable (TD):** Is the language of a TM that always halts

**Turing Recognizable (TR):** Is the language of any TM