Final Exam Study Guide

CSE 30151 Spring 2023

Exam date: 2023-05-12 10:30am–12:30pm in Hayes-Healy 127

The cover page of the exam will look like this.

Name:

NetID:

- This exam has eight questions, worth 15 points each, for a total of 120 points (20% of your grade).
- You may use your textbook and paper notes, but computers, smartphones, and tablets are **not** allowed.
- You may use the textbook, lectures, and lecture notes for this course without citation. However, you may **not** copy or quote from any other materials in your notes that you are not the author of.
- On this page, please write your name and NetID, but please don't write any solutions. On the remaining pages, front and back, please write your solutions, but please don't write your name.

Problem types

The questions will be of the following types. (Exercise/problem numbers are from Sipser; an $^{\mathsf{I}}$ means 3rd international edition and a $^{\mathsf{U}}$ means 3rd US edition.)

1–2. Two questions will be on topics covered on the midterm exam. See the midterm study guide for examples, but see below for topics specifically not on the final exam.

3–4. Two of the questions will cut across multiple topics covered this semester. They will either be short-answer or at most require one-line proofs.

5. Prove that a language is not context-free (Problems 2.42bc$^{\mathsf{I}}$/2.30bc$^{\mathsf{U}}$).

6. Prove that a TM variant or other formal system is equivalent to a TM (Problems 3.17–18$^{\mathsf{I}}$/3.10–11$^{\mathsf{U}}$).

7. Prove that a language is undecidable. Examples: Exercise 5.1 (hint: use Theorem 5.13), Problems 5.29$^{\mathsf{I}}$/5.13$^{\mathsf{U}}$, 5.25–27$^{\mathsf{I}}$/5.9–11$^{\mathsf{U}}$.

8. Prove that a language is NP-complete. Examples: Problem 7.49–50$^{\mathsf{I}}$/7.22–23$^{\mathsf{U}}$ are on the upper end of what I think is fair on an exam. On the lower end would be:

    - In a graph $G = (V, E)$, an *independent set* is a subset $C \subseteq V$ such that no two nodes in C are adjacent. Prove that it is NP-complete to decide whether a given graph G has an independent set with a given size k.
    - A Boolean circuit with ℓ inputs and 1 output is *satisfiable* iff there is a set of inputs that make the output 1. Prove that it is NP-complete whether a given Boolean circuit using only NAND gates is satisfiable.

Topics *not* on the exam

- Conversion between DFAs, NFAs, and regular expressions
- Conversion between CFGs and PDAs
- Chomsky normal form (108–110)
- Deterministic context-free languages (§2.4)
- Advanced topics in computability theory (§6)
- Examples of decidable languages (§4.1) and polynomial languages (§7.2).
- §8 and beyond

Solutions to selected exercises/problems

Problem 5.25$^{\mathsf{I}}$ This language is undecidable by Rice's Theorem. Alternatively, suppose that this language is decided by a TM R. We can use R to construct a TM S that decides A_{TM} as follows:

$S =$ On input $\langle M, w\rangle$:

1. Construct $M' =$ On input x:
   (a) Simulate M on w.
   (b) If M accepts w and $x = \texttt{01}$, accept.
   (c) If M accepts w and $x \neq \texttt{01}$, reject.
   (d) If M rejects w, reject.
2. Run R on $\langle M'\rangle$.
3. If R accepts $\langle M'\rangle$, reject.
4. If R accepts $\langle M'\rangle$, accept.

If M accepts w, then M' recognizes the language $\{\texttt{01}\}$, so R rejects, so S accepts. However, if M rejects w or loops on w, then M' recognizes the language $\emptyset$, so R accepts, so S rejects. So S decides A_{TM}. But A_{TM} is undecidable, so this is a contradiction.

Problem 5.29$^{\mathsf{I}}$ Suppose that this language is decided by a TM R. We can use R to construct a TM S that decides A_{TM} as follows:

$S =$ On input $\langle M, w\rangle$:

1. Construct $M' =$ On input x:
   (a) Visit every state except for a special state q_{special}. (If desired, one could go into more detail about how this is done.)
   (b) Simulate M on w.
   (c) If M accepts w, visit q_{special} and halt.
   (d) If M rejects w, halt.
2. Run R on $\langle M'\rangle$.
3. If R accepts $\langle M'\rangle$, reject.
4. If R accepts $\langle M'\rangle$, accept.

If M accepts w, then M' (on any input) visits all of its states including q_{special}, so R rejects, so S accepts. However, if M rejects w or loops on w, then M' does not visit q_{special} (on any input), so R accepts, so S rejects. So S decides A_{TM}. But A_{TM} is undecidable, so this is a contradiction.

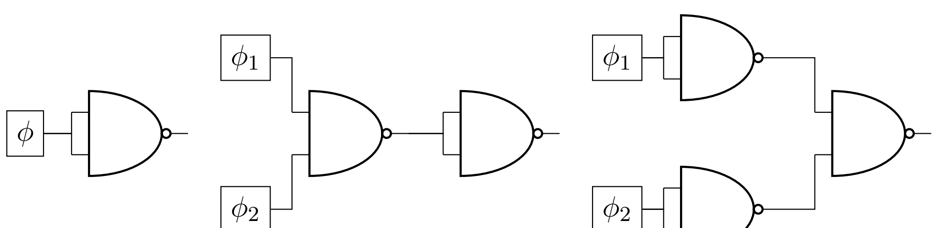
Problem 7.49[I] The certificates for DOUBLE-SAT are pairs of assignments, which can clearly be checked in linear time. We show that DOUBLE-SAT is NP-hard by reduction from SAT. Given a formula ϕ with variables $x_1, \ldots, x_\ell$, let y be a new variable and let

$$f(\phi) = (\neg y \wedge \phi) \vee (y \wedge \neg x_1 \wedge \cdots \wedge \neg x_\ell)$$

which can clearly be constructed in linear time. If ϕ has a satisfying assignment ξ, then we can make two satisfying assignments for $f(\phi)$: one by taking ξ and also setting $y = 0$, and another by setting $x_1 = \cdots = x_\ell = 0$ and $y = 1$. Conversely, if $f(\phi)$ has two satisfying assignments, at most one of them can have $y = 1$, so the other one must satisfy ϕ.

NP-completeness of INDEPENDENT-SET The certificates are subsets $C \subseteq V$ and can clearly be checked in at most quadratic time. We prove this language NP-hard by reduction from CLIQUE. Given $\langle G, k\rangle$, where $G = (V, E)$, let $\bar{G}$ be the graph with vertices V and edges $V \times V \setminus E$, that is, (u, v) is an edge in G iff it is *not* an edge in $\bar{G}$. Then let $f(\langle G, k\rangle) = \langle \bar{G}, k\rangle$, which is clearly computable in linear time. If C is a clique of G, then it is an independent set of $\bar{G}$, and conversely if C is an independent set of $\bar{G}$, then it is a clique of G.

NP-completeness of NAND-SAT The certificates are sets of inputs, which can clearly be checked in linear time. We prove this language NP-hard by reduction from SAT. Given a formula ϕ, we can recursively convert it into a circuit using only NAND gates:



$\neg\phi$ $\qquad$ $\phi_1 \wedge \phi_2$ $\qquad$ $\phi_1 \vee \phi_2$

The circuit has size linear in the size of ϕ, and it computes exactly the same truth value as ϕ, so it is satisfiable if and only if ϕ is.