CSC 456/656 Spring 2023 Answers to Third Examination April 12, 2023

1. True or False. 5 points each. T = true, F = false, and O = open, meaning that the answer is not known science at this time.

    (i) **T** Every finite language is decidable.

    (ii) **T** Every context-free language is in Nick's class.

    (iii) **F** 2SAT is known to be $\mathcal{NP}$-complete.

    (iv) **T** The complement of any $\mathcal{P}$-TIME language is $\mathcal{P}$-TIME.

    (v) **T** The complement of any $\mathcal{P}$–SPACE language is $\mathcal{P}$–SPACE.

    The *jigsaw puzzle problem* is, given a set of various polygons, and given a rectangular table, is it possibe to assemble those polygons to exactly cover the table?

    The *furniture mover's problem* is, given a room with a door, and given a set of objects outside the room, it is possible to move all the objects into the room through the door?

    (vi) **T** The jigsaw puzzle problem is known to be $\mathcal{NP}$ complete.

    (vii) **F** The jigsaw puzzle problem is known to be $\mathcal{P}$–SPACE complete.

    (viii) **F** The furniture mover's problem is known to be $\mathcal{NP}$ complete.

    (ix) **T** The furniture mover's problem is known to be $\mathcal{P}$–SPACE complete.

    (x) **T** The complement of any recursive language is recursive.

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

    (xii) **F** Every undecidable language is either $\mathcal{RE}$ or co-$\mathcal{RE}$.

    (xiii) **T** For any infinite countable sets A and B, there is a 1-1 correspondence between A and B.

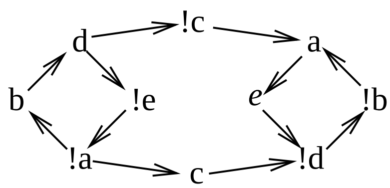
    (xiv) **T** A language L is recursively enumerable if and only if there is a machine which accepts L.

    (xv) **T** Every $\mathcal{NP}$ language is reducible to the independent set problem in polynomial time.

    (xvi) **T** If a Boolean expression is satisfiable, there is a polynomial time proof that it is satisfiable.

    (xix) **T** The Post correspondence problem is undecidable.

2. [20 points] Determine whether the following Boolean expression is satisfiable. If so, give a satisfying assignment.
$(a + b) * (a + c) * (!a + e) * (!b + d) * (!c+!d) * (!d+!e)$



The graph G contains the path $!a \to b \to d \to !c \to a$, and thus any satisfying assignment must assign $a = 1$. The cycle $a \to !b \to !d \to e \to a$ shows that a, $!b$, $!d$, and e must be assigned the same value. We thus can assume $a = e = 1$ and $b = d = 0$.

Substituting those values, the expression becomes $(1 + 0) * (1 + c) * (0 + 1) * (1 + 0) * (!c + 1) * (1 + 0)$ which reduces to the empty expression. The expression is satisfiable, and c can be assigned either 0 or 1.

3. [20 points] Using the fact that 3SAT is $\mathcal{NP}$-complete, prove that the independent set problem is $\mathcal{NP}$-complete.

The proof is given in the handout *satInd.pdf.* However, to get full credit on the test, you only need to describe the reduction and to state the final sentence of this paragraph. Suppose $E = C_1 * \ldots * C_k$ is a 3CNF Boolean expression. The reduction takes E to a graph whose vertices are in 1-1 correspondence with the terms of E. There are two kinds of edges of G: short edges, which connect vertices corresponding to pairs of terms in the same clause of E, and long edges connecting any two vertices whose corresponding terms contradict each other. Then G has an independent set of size k if and only if E is satisfiable.

4. [20 points] State the Church Turing thesis. Why is it important?

Almost everyone correctly stated that the thesis is that any computation that can be done by any machine can be done by some Turing machine. But many students got the answer to the second question backwards. It is important because, to prove that no machine can do a computation, you only need show that no Turing machine can do that computation.

5. [20 points] Prove that every language which can be enumerated in canonical order by some machine is recursive.

Suppose L is a language which can be enumerated in canonical order by some machine. If L is finite, then it's automatically recursive. Suppose L is infinite, and some machine write the strings of L in canonical order, $w_1, w_2, \ldots$. Then a machine that executes the following program decides whether a given string w is a member of L.

Read w For $i = 1$ to ∞
  If $(w_i = w)$
    Accept w and halt.
  Else if $(w_i > w)$ in the canonical order
    Reject w and halt.

The program will check all w_i until it finds one that equals or exceeds w. In the latter case, it will never see w, and thus rejects.

6. [20 points] Why is the question of whether $\mathcal{NC} = \mathcal{P}$–TIME so important nowadays?

It has nothing to do with the $\mathcal{P} = \mathcal{NP}$ problem or the problem of decryption. Since parallel computers are increasingly being used, the question of whether a program can be efficiently parallelized becomes of greater importance. If $\mathcal{P} = \mathcal{NC}$, then any polynomial time algorithm can be efficiently parallelized.

8. [20 points] Prove that the halting problem is undecidable. Do not quote any theorem or lemma from the handouts.

I was pleased to note that most students did this correctly.

Proof by contradiction. Assume the halting problem is decidable. Let D be a machine which executes the following program.

```
Read a machine description ⟨M⟩.
If (M halts with input ⟨M⟩ ) // This can be evaluated since the halting problem is decidable
   Enter an infinite loop
Else
   Halt
```

Note that D halts, that is, accepts $\langle M \rangle$, if and only if M does not halt with input $\langle M \rangle$.

Question: Does D accept $\langle D \rangle$?

If the answer to the question is YES, then D must halt with input $\langle D \rangle$. But if the answer is YES, the if condition is true, and D runs forever, contradiction.

If the answer to the question is NO, then D must not halt with input $\langle D \rangle$. But if the answer is NO, the if condition is false, and D halts, contradiction.

In either case, there is a contradiction. We conclude that the halting problem is undecidable.