## Computer Science 456/656 Spring 2020

## Answers to Second Examination April 30, 2020

The entire examination is 205 points.

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

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

   ii **T** EXP-TIME $\subseteq$ EXP-SPACE.

   iii **T** There exists a context-sensitive language which is $\mathcal{P}$-SPACE complete.

   iv **T** Every finite language is regular.

   v **T** The language $\left\{a^i b^j a^j b^i \ : \ i, j \geq 0\right\}$ is context-free.

   vi **T** Any languge generated by an unrestricted grammar is recursively enumerable.

   vii **F** Every polynomial time language is context-free.

   viii **T** If $L$ is in $\mathcal{P}$-SPACE, there is a reduction of $L$ to the regular expression equivalence problem.

   ix **F** The union of two undecidable languages is always undecidable.

   x **T** The union of two recursively enumerable languages is always recursively enumerable.

   xi **T** The union of two co-RE languages is always co-RE. Hint: THINK!

   xii **T** $\mathcal{NC} = co - \mathcal{NC}$. Hint: THINK!

   xiii **T** The set of all regular expressions for regular languages over the alphabet $\{a, b\}$ is a context-free language.

   xiv **O** Various websites, such as https://www.youtube.com/watch?v=bQnjbDHefgc give solutions to various instances of RUSH HOUR. If there is a solution to a particular instance of RUSH HOUR, that solution can always be explained in polynomial time.

   xv **T** The factoring problem for an integer written in binary is both $\mathcal{NP}$ and co-$\mathcal{NP}$.

   xvi **T** If someone somewhere on the Earth publishes a correct proof that the partition problem is in $\mathcal{P}$-TIME, then it will be known that $\mathcal{P} = \mathcal{NP}$.

   xvii **F** If someone somewhere on the Earth publishes a correct proof that the factoring problem for binary numerals is in $\mathcal{P}$-TIME, then it will be known that $\mathcal{P} = \mathcal{NP}$.

   xviii **F** $\mathcal{NC} = \mathcal{P}$-SPACE

   xix **T** co-$\mathcal{NP} \subseteq \mathcal{P}$-SPACE.

2. Fill in the blanks. [10 points each blank.]

   (a) If $L \subseteq \Sigma^*$ is $\mathcal{NP}$ time, there is a constant $k$ and a deterministic machine $V$ such that, for string $w \in \Sigma^*$, we have $w \in L$ if and only if there is a string $c$, called a **certificate** or **witness** for $w$, such that $|c| \leq |w|^k$ and $V$ accepts the string $cw$ within $|w|^k$ steps.

   (b) The practicality of the RSA one-way encryption system depends on the assumption (which has not been verified) that the **factoring** problem cannot be solved in polynomial time.

   (c) $\mathcal{NC}$ is the class of languages which can be decided in **polylogarithmic** time using polynomially many processors.

3. [20 points] Every context-free language has a minimum pumping length. For example, the minimum pumping length of $\{a^n b^n : n \geq 0\}$ is 2.

   The language $L = \{a^n b^m cde^n : n, m \geq 0\}$ is context-free.

   (a) Find the minimum pumping length of $L$.

   The minimum pumpting length, $p$, is 4. Some people gave 5 as the answer, some 3.

   (b) For every string $w \in L$ of length at least $p$, there are strings $u, v, x, y, z$ such that $w = uvwxy$ and three other conditions are satisfied. Find the strings $u, v, x, y, z$ if $w = aabcdee$.

   $u = aa$, $v = b$, $x = cd$, $y = \lambda$, $z = ee$. By pumping up or down, we obtain the strings *aacdee*, *aabbcdee*, *aabbbcdee*, *etc.*

   Many people thought that $p = 3$ because $|vxy| = 3$, However, We need 4. If $w = abce$ we must choose $u = \lambda$, $v = a$, $x = bc$, $y = e$, and $z = \lambda$. By pumping $v$ and $y$, we get the strings *cd*, *aacdee*, *aaacdeee*, *etc.*

4. [20 points] Give a polynomial time reduction of 3-SAT to the independent set problem.

   Let $E$ be a Boolean expression in CNF normal form, where each clause has three terms.

   Let $K$ be the number of clauses of $E$. Let $C_i$ be the $i^{\text{th}}$ clause of $E$. Let $T_{i,j}$ be the $j^{\text{th}}$ term of $C_i$, for $j = 1, 2, 3$.

   Let $G = (V, E)$ be the graph where $V = \{v_{i,j}\}$ for $1 \leq i \leq K$ and $1 \leq j \leq 3$. Let $E$ consists of the following edges:

   (a) $(v_{i,1}, v_{i,2}), (v_{i,1}, v_{i,3}), and (v_{i,2}, v_{i,3})$ for each $1 \leq i \leq k$ We call these the internal edges.

   (b) $(v_i, k), (v_j, \ell)$ for all $i, j, k, \ell$ for which $T_{i,k} \wedge T_{j,\ell}$ is a contradiction. We call these the external edges.

   If $E$ is satisfiable, $G$ has an independent set of $K$ vertices.

   *Proof:* Pick a satisfying assignment. For each $i$. Pick a term of $C_i$ which is true under that assignments. Without loss of generality, that term is $T_{i,1}$, since we can permute the terms of $C_i$. Let $S = \{V_{i,1}\}$. Note that $S$ has size $K$. We claim $S$ is independent. If $i \neq j$, there is no external edge from $v_{i,1}$ to $v_{j,1}$ because the terms $T_{i,1}$ and $T_{j,1}$ are both true under the satisfying assingment, and hence cannot contradict each other. Thus $S$ is independent. ∎

   Conversely, if $G$ has an independent set of size $K$, then $E$ is satisfiable.

   *Proof:* Pick an independent set $S \subseteq V$ of size $K$. For each $i$, there can be at most one $j$ for which $v_{i,j} \in S$. since otherwise two members of $S$ would have an internal edge in common. Thus, for each

$i$, there is some $j$ for which $v_{i,j} \in S$. Without loss of generality, $j = 1$. Choose an assignment of $E$ for which $T_{i,1}$ is true for all $i$. This assignment is possible because, $T_{i,1}$ cannot contradict $T_{k,1}$ because otherwise they would be connected by an external edge and hence $S$ would not be independent. Under that assignment, $C_i$ is true because $T_{i,1}$ is true. ∎

5. [20 points] Prove that the halting problem is undecidable.

   We write $\langle M \rangle$ for the binary encoding of a Turing machine $M$. Define the *diagonal language* $L_{diag} = \{\langle M \rangle \langle M \rangle : \langle M \rangle \notin L(M)\}$

   Claim: $L_{diag}$ is not accepted by any Turing machine.

   *Proof:* Assume that there is a Turing machine $M_{diag}$ which accepts $L_diag$. For any encoding $\langle M \rangle$ of a Turing machine $M$, we have the following two statements:

   (a) $\langle M \rangle \in L_{diag} \Leftrightarrow \langle M \rangle \in L(M_{diag})$ by the definition of $M_{diag}$.

   (b) $\langle M \rangle \in L_{diag} \Leftrightarrow \langle M \rangle \notin L(M)$ by the definition of $L_{diag}$.

   By universal instantiation, we obtain the two statements:

   (a) $\langle M_{diag} \rangle \in L_{diag} \Leftrightarrow \langle M_{diag} \rangle \in L(M_{diag})$

   (b) $\langle M_{diag} \rangle \in L_{diag} \Leftrightarrow \langle M_{diag} \rangle \notin L(M_{diag})$

   Contradiction, hence $M_{diag}$ cannot exist. ∎

   $L_{diag}$ is not R.E., and hence undecidable. We now give a reduction $R.$ of $L_{diag}$ to the complement of HALT. For any string $w$, let $R(w) = ww$. If $w \in L_{diag}$ then $w = \langle M \rangle$ $M$ does not accept $\langle M \rangle$. Thus $R(w) = \langle M \rangle \langle M \rangle$ is not in HALT. It follows that the complement of HALT is not decidable, hence HALT is not decidable.

   Some students gave the following alternative proof, which is shorter, but harder to explain. Thus, I'll stick with my proof.

   Suppose HALT is decidable. Let $M_L$ be a turing machine which decides $L$, and let $M_{diag}$ be a TM equivalent to the following program, which takes input any $\langle M \rangle w$.

   | if ($M_L$ halts with input $\langle M \rangle w$) run forever else halt |
   |---|

   If $M_{diag}$ halts with input $\langle M_{diag} \rangle \langle M_{diag} \rangle$, we have a contradiction, if not, we also have a contradiction. Thus HALT is not decidable.

6. [20 points] Prove that the language $L = \{a^n b^m c^m d^n : n, m \geq 0\}$ is context-free by giving a context-free grammar for $L$.

   $S \rightarrow aSd|T$
   $T \rightarrow bTc|\lambda$