LAST NAME (please print)	
First name (please print)	
Student Number	

WESTERN UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

CS3331: Foundations of Computer Science – Fall 2020
– Final Exam –

Saturday, Dec. 12, 2020, 2:00 - 5:00pm
Location: OWL

Instructor: Prof. Lucian Ilie

Upload your solutions in OWL by 5:30pm.
Approved accommodation: email to cs3331@uwo.ca by 2:00pm + your approved time + 30min.
In either case, failure to do so will result in your exam being discarded; no exceptions.

This exam consists of 5 questions (6 pages, including this page), worth a total of 100 marks.
The exam is 180 minutes long and comprises 39% of your final grade.

For each questions, solve only part $V_i, 0 \leq i \leq 3$, where $i = \mathit{student\_number} \mod 4$.
Failure to answer the correct version will result in your answer being discarded.

$V_i =$	

(1) 20pt	
(2) 20pt	
(3) 20pt	
(4) 30pt	
(5) 10pt	
Grade	

1. (20pt) Remember to solve only your version V_i; calculate correctly your $i = student\_number \mod 4$.
Construct a deterministic Turing machine M that performs the action indicated at your V_i below. M starts with the initial configuration $(s, \underline{\square}w)$ and halts with the configuration $(h, \underline{\square}w)$. Describe M using the macro language (the one that looks like this: $R_{\square,\mathtt{a}} \xrightarrow{\square} \mathtt{a}R\mathtt{b}L_{\neg\square}$).

$V_0 :$ $\Sigma = \{a, b\}$; M replaces any occurrence of aba in the input with aca.

$V_1 :$ $\Sigma = \{0, 1\}$; M adds 2 to its input, seen as a binary number.

$V_2 :$ $\Sigma = \{a, b\}$; M moves the leftmost a (if any) to the end of the input, then closes the hole where a was.

$V_3 :$ $\Sigma = \{a, b, c\}$; M replaces the input with the number of a's in the input, written in unary (using 1's).

2. (20pt) Describe in clear English a Turing machine that semidecides the language L given below:

   V_0 : $L = \{<M> \mid M$ rejects at least two strings$\}$

   V_1 : $L = \{<M> \mid M$ accepts at least one string starting with $a\}$

   V_2 : $L = \{<M> \mid M$ accepts the empty string and at least one string of odd length$\}$

   V_3 : $L = \{<M> \mid M$ accepts at least two strings of different lengths$\}$

3. (20pt) Prove your answers to the questions below ((a) – 6pt, (b) – 5pt, (c) – 9pt: 3pt for each (i)-(iii)):

V_0 : (a) Is it possible that $L \in$ D and $L \cap \neg L \notin$ SD $-$ D?

(b) If we modify an FSM to allow infinitely many states, then we can easily accept any language, e.g., we can build a path to accept every string in the language. That means, we can accept also non-SD languages. Does this contradict Church's thesis?

(c) Can the union $L_1 \cup L_2$, for $L_1 \in$ SD, $L_2 \in \neg$SD be in: (i) D, (ii) SD $-$ D, (iii) $\neg$SD?

V_1 : (a) Is it possible that L is regular and $\neg(L \cap \neg L) \notin$ SD $-$ D?

(b) Is it possible to design a new mechanism (that would have a finite description) such that the languages it accepts are precisely the non-SD languages, thus contradicting Church's thesis?

(c) Can the intersection $L_1 \cap L_2$, for $L_1 \in$ SD, $L_2 \in \neg$SD be in: (i) D, (ii) SD $-$ D, (iii) $\neg$SD?

V_2 : (a) Is it possible that L is context-free and $L - \neg\neg L \notin$ SD $-$ D?

(b) Is it possible to define a new mechanism that uses a Turing Machine but accepts exactly when the TM does not; such a mechanism would then accept languages such as $\neg H$, which is not in SD, thus contradicting Church's thesis?

(c) Can the difference $L_1 - L_2$, for $L_1 \in$ SD, $L_2 \in \neg$SD be in: (i) D, (ii) SD $-$ D, (iii) $\neg$SD?

V_3 : (a) Is it possible that $L \notin$ SD and $L \cup \neg L \in$ D?

(b) Let's define a new class of languages, which is obtained as the closure of SD under complement. This new class would strictly include SD, as it would include, for instance, $\neg H$. Does this contradict Church's thesis?

(c) Can the intersection $L_1 \cap L_2$, for $L_1 \in \neg$SD, $L_2 \in$ SD be in: (i) D, (ii) SD $-$ D, (iii) $\neg$SD?

4. (30pt) Consider an alphabet Σ such that all languages below are over Σ.

   (i) (18pt) For each of the languages below, prove, without using Rice's theorem, whether it is in D, SD − D, or ¬SD. Explain first intuitively why you think it is in D, SD − D, or ¬SD, then prove your assertion rigorously.

   (ii) (12pt) Explain whether Rice's Theorem applies or not to each of these languages.

V_0 : (a) $L_1 = \{<M_1, M_2> \mid M_1$ accepts at least two strings and M_2 rejects at least one string$\}$
(b) $L_2 = \{<M> \mid M$ accepts only the string $aba\}$
(c) $L_3 = a^*$

V_1 : (a) $L_1 = \{<M_1, M_2> \mid M_1$ accepts at least one string in a^* and M_2 rejects at least one string in $b^*\}$
(b) $L_2 = \{<M> \mid |L(M)| \leq 10\}$
(c) $L_3 = \{\varepsilon\}$

V_2 : (a) $L_1 = \{<M_1, M_2> \mid M_1$ accepts a and $L(M_2)$ is not empty$\}$
(b) $L_2 = \{<M> \mid M$ accepts finitely many even-length strings$\}$
(c) $L_3 = \emptyset$

V_3 : (a) $L_1 = \{<M_1, M_2> \mid M_1$ accepts at least one string and M_2 rejects at least one string$\}$
(b) $L_2 = \{<M> \mid M$ accepts two palindromes and nothing else$\}$
(c) $L_3 = \{ab\}$

5. (10pt) Answer your version of the PCP question below:

V_0 : PCP over one letter (that is, all strings are from 1^*) is decidable, because we work with numbers instead of strings. If PCP over one letter is decidable, and we can always encode any number of letters into a single letter (e.g., $a = 1, b = 11, c = 111, d = 1111$, etc.), explain how is it possible that PCP over an arbitrary alphabet is undecidable?

V_1 : Explain what is wrong with the following proof that PCP is decidable. Denote the top and bottom strings by $(x_1, x_2, \ldots, x_n)$ and $(y_1, y_2, \ldots, y_n)$, resp. Considering all possible subsets of all possible permutations of these blocks, we obtain $2^{n!}$ possibilities. If we denote the maximum length of a string by $m = \max_{i=1..n}(\max(|x_i|, |y_i|))$, this means the shortest solution, if any, must be of length at most $m2^{n!}$. The PCP is then decided by checking all potential solutions up to this length.

V_2 : Show that the following restricted version of PCP is decidable: $\mathrm{PCP}_r = \{((x_1, x_2, \ldots, x_n), (y_1, y_2, \ldots, y_n)) \mid n \geq 1, x_i, y_i \in \{a, b\}^+, \max(|x_i|, |y_i|) \leq 4, \text{ for all } 1 \leq i \leq n\}$.

V_3 : Given a positive number n, construct a PCP problem over $\{a, b\}$ whose shortest solution has n blocks. Explain why your answer is correct.