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

WESTERN UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

CS3331: Foundations of Computer Science – Fall 2021
– Final Exam –

Friday, Dec. 17, 2021, 2:00 - 5:00pm
OWL

Instructor: Prof. Lucian Ilie

The exam will be available Friday at 2:00pm in OWL and you must submit your solutions by 5:10pm as a single pdf file. Students with accommodation for extra time will submit their solutions by email to cs3331@uwo.ca, according to their accommodation details.

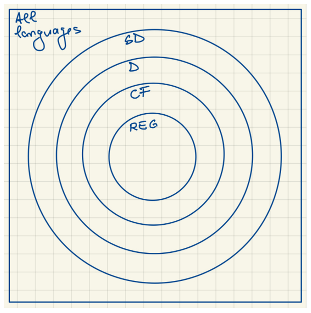
This exam consists of 5 questions (16 pages, including this page), worth a total of 100 marks. All answers are to be written in this booklet. The exam is 180 minutes long and comprises 39% of your final grade.

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

1. (20pt) Consider the context-free language of balanced parentheses of three kinds:

$$L = \{w \in \{\texttt{(},\texttt{)},\texttt{[},\texttt{]},\texttt{\{},\texttt{\}}\}^* \mid \text{all parentheses in } w \text{ are properly balanced}\} \ .$$

Place $\neg L$ correctly in the correct area of the map. Prove your answer.



2. (20pt) Answer the following questions with True or False; prove your answers:
   (a) (2pt) Any regular language is context-free.
   (b) (2pt) Some context-free languages are regular.
   (c) (8pt) Any regular language can be generated by an unambiguous context-free grammar.
   (d) (6pt) Any context-free grammar generating a regular language is unambiguous.
   (e) (2pt) For any non-semidecidable language L, we have that $\varepsilon \in L^*$.

3. (20pt) Consider the following language:

$$L = \{ <G_1, G_2> \mid G_1, G_2 \text{ context-free grammars such that } L(G_1) \cap L(G_2) = \{\varepsilon\}\} \ .$$

Here is an algorithm that decides L:

```
1.  if $\varepsilon \notin L(G_1)$, then reject
2.  if $\varepsilon \notin L(G_2)$, then reject
3.  $k_1 \leftarrow$ the $k$-constant in pumping theorem for $G_1$
4.  $k_2 \leftarrow$ the $k$-constant in pumping theorem for $G_2$
5.  $k \leftarrow \max(k_1, k_2)$
6.  for all $w \in L(G_1) - \{\varepsilon\}$ with $|w| \leq k$ do
7.      if $w \in L(G_2)$ then reject
8.  for all $w \in L(G_2) - \{\varepsilon\}$ with $|w| \leq k$ do
9.      if $w \in L(G_1)$ then reject
10. accept
```

The main idea of the algorithm is that we can check membership for finitely many strings. Therefore, the difficulty arises in the case when both languages are infinite. We know from pumping theorem that, if a language has strings longer than k, then it has also strings shorter than k (we "pump out" to reduce the length, while staying in the language). The algorithm uses this idea with k the larger of the two individual k's of each language, to ensure the strings longer than k are long enough for both languages.

Is the above algorithm corect? Explain your answer.

4. (20pt) Consider an alphabet Σ such that all languages below are over Σ. For each of the languages below, prove, without using Rice's theorem, whether it is in D, SD − D, or ¬SD.

   (a) $L_1 = \{<M>|\ M$ Turing machine, the complement of $L(M)$ is semidecidable$\}$.

   (b) $L_2 = \{<M>|\ M$ nondeterministic finite automaton such that, for any nondeterministic finite automaton M' with $L(M') = L(M)$, M has the same number of states as M' or fewer$\}$.
   (For L_2, any finite automaton M can be encoded as $< M >$ using any of the methods we studied, for instance, the one for Turing Machines. The way the encoding is done is irrelevant for the question.)

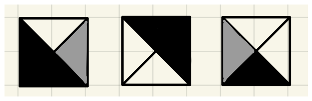
   (c) $L_3 = \{<M>|\ M$ Turing machine, both $L(M)$ and $\neg L(M)$ are infinite$\}$.

   (d) $L_4 = \{<M, w>|$ on input w, M begins by moving right one square onto w, then it never moves outside w, i.e., outside the $|w|$ tape cells occupied by w at the beginning$\}$.

   (All Turing machines are single-tape deterministic.)

5. (20pt) Recall that TILES = $\{< T >|$ any finite surface on the plane can be tiled with the tile set $T\}$? (Recall that tiles cannot be rotated or flipped and adjacent sides of joined tiles must have the same colour.)

   (a) (6pt) Is the set of tiles below in TILES? Explain your answer.

   (b) (6pt) If your answer above is negative, then is it possible to add a single tile that makes the new set (of four tiles) in TILES? Explain your answer.

   (c) (2pt) Which of TILES and ¬TILES is harder? No proof is required.
   For this question, "harder" means (strictly) higher in the hierarchy: $D < SD - D < \neg SD$, that is, $\neg SD$ is harder than $SD - D$, which is harder than D; also, no class is harder than itself.

   (d) (6pt) Given a language L such that $\neg L$ is harder than L, is it possible that L is in: (i) D, (ii) $SD - D$, (iii) $\neg SD$? Explain your answer. You can use examples studied in class without proof.

(scrap work)