EXPANDED COURSE DESCRIPTION
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
Lassonde School of Engineering
Electrical Engineering Computer Science
LE / EECS 4111 3.0 SECTION A
AUTOMATA AND COMPUTABILITY
FALL 2018 / WINTER 2019

Last Modified Date: 08/20/2018

COURSE CALENDAR DESCRIPTION

Introduction to more advanced topics in theoretical foundations of computer science, including the study of formal languages and automata, formal models of computation, and computational complexity measures. Integrated with: GS/CSE 5111 3.00. Prerequisites: cumulative GPA of 4.50 or better over all major EECS courses (without second digit "5"); LE/EECS 2030 3.00 or LE/EECS 1030 3.00; LE/EECS 2001 3.00; LE/EECS 3101 3.00. Previously offered as: LE/CSE 4111 3.00.

INSTRUCTOR(S)

<table>
<thead>
<tr>
<th>Name</th>
<th>Section / Format / Term</th>
<th>Contact Email</th>
<th>Contact Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tourlakis, George</td>
<td>Sec. A / LECT / F</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ADDITIONAL INFORMATION

It is the responsibility of students to check the course web page weekly for course related information. Our main aim will be a thorough study of Computability and an introduction to Complexity. We will lay the foundations of the theory using the Unbounded Register Machines of Shepherdson and Sturgis (URMs). Topics will include: A mathematical model of computation (URMs); Primitive Recursion and Loop Programs; Arithmetisation and Kleene's Predicate; Church's Thesis; Unsolvability via Diagonalisation and Reductions; The Recursion Theorem; The connection between Uncomputability and Unprovability: Gödel's First Incompleteness Theorem through the Halting Problem; The complexity of Primitive Recursive functions and relations via the hierarchy approach (Axt, Loop-Program, Grzegorczyk hierarchies) including the Ritchie-Cobham theorem.

WORK-LOAD AND GRADING

The course grade will be based mainly on about 3-4 homework assignments that will be completed by students individually. These will be equally weighted. There will be no programming problems, but each assignment will consist of a number of theoretical problems, for example "prove that such-and-such function is not computable", or "prove that such-and-such function is primitive recursive", etc. Graduate students in the course (registered in CSE5111) will be assigned additional homework and readings, and will be expected to probe the material further and deeper. The concept of "late assignments" does not exist (solutions are posted typically on the due date).

OFFICIAL TEXTBOOK


1. Notes beyond the text are posted on the course web site and are frequently updated.

COURSE LEARNING OUTCOMES

By the end of the course, the students will be expected to be able to:
• Demonstrate knowledge of, via application: the concepts of “computation”, and “computable” and “uncomputable” functions, “decidable”, semi-decidable” (or “recursively enumerable”) and “undecidable”
decision problems in terms of mathematical models of computation (e.g., Unbounded Register Machines, Turing Machines).

- Design programs, in a mathematical model of computation, that compute specified functions or solve specified decision problems.
- Define Church’s Thesis and use it fluently to design “pseudo programs” that compute functions or decide/semi-decide decision problems.
- Prove undecidability and non-semi-decidability results via diagonalisation and/or reduction arguments.
- Understand and reproduce the argument that connects uncomputability in the theory of computation with unprovability in logic: Gödel’s First Incompleteness Theorem through the viewpoint of the halting problem.
- Understand the concept of subrecursive complexity hierarchies (e.g., Axt, Grzegorczyk, Loop-program) and be able to place a given function or decision problem in a complexity class in such a hierarchy.

REFERENCES.
For more on Primitive Recursive and (total) Recursive functions see:

For more on unsolvability of concrete problems of combinatorial or number theoretical nature, see:

(3) M. Davis, *Computability and Unsolvability*, McGraw-Hill (Turing approach, rigorous, fairly difficult).
Matijasevic’s proof of the unsolvability of Hilbert’s 10th problem, using methods initiated by Davis (3) can be found in:

For more on recursion theory, in general, see:

Finally, for more on machines (at an elementary level), see:


ACADEMIC INTEGRITY LINKS
- Senate Policy on Academic Honesty - http://secretariat-policies.info.yorku.ca/policies/academic-honesty-senate-policy-on/
- Academic Integrity - http://lassonde.yorku.ca/academic-integrity

STUDENT LINKS
- Student Rights and Responsibilities - http://oscr.students.uit.yorku.ca/student-conduct
- Religious Observance - https://w2prod.sis.yorku.ca/Apps/WebObjects/cdm.woa/wa/regobs
- Counselling and Disability Services - http://cds.info.yorku.ca/
- York University’s Policies on Gender/LGBTQ*/Positive Space - http://rights.info.yorku.ca/lgbtq/

LAND ACKNOWLEDGEMENT
- We acknowledge our presence on the traditional territory of many Indigenous Nations. The area known as Tkaronto has been care taken by the Anishinabek Nation, the Haudenosaunee Confederacy, the Huron-Wendat, and the Métis. It is now home to many Indigenous Peoples. We acknowledge the current treaty holders, the Mississaugas of the New Credit First Nation. This territory is subject of the Dish With One Spoon Wampum Belt Covenant, an agreement to peaceably share and care for the Great Lakes region.
- The Indigenous Framework for York University: A Guide to Action can be found here: http://indigenous.info.yorku.ca/
- Meaning of a land acknowledgement: http://healthydebate.ca/opinions/indigenous-land-acknowledgements
Many courses utilize Moodle, York University’s course website system. If your course is using Moodle, click here to access it.

Moodle @ York University