Computing Science Course Outlines 2018 Fall

CMPT 417 - D100 Intelligent Systems

Instructor(s): David Mitchell

Calendar Objective/Description:
Intelligent Systems using modern constraint programming and heuristic search methods. A survey of this rapidly advancing technology as applied to scheduling, planning, design and configuration. An introduction to constraint programming, heuristic search, constructive (backtrack) search, iterative improvement (local) search, mixed-initiative systems and combinatorial optimization.

Instructor’s Objectives:
Many real computing applications involve computational problems for which no generally efficient algorithms are known. These problems often involve relatively small inputs, but astronomically-sized search spaces in which solutions may be well hidden. Developing practical software for solving these problems is challenging and costly. This course examines methods for automatically solving these problems based on high-level specifications, much as we use a query language rather than low-level algorithms to answer complex questions about a large data set. The course will pursue two tracks in parallel. In one, we will examine the design and use of real systems that are products of major companies or widely used research tools, including hands-on experience. In the other, we will develop formal foundations to support principled design and analysis of such systems and their use in critical applications, and to understand their strengths and limitations. We will begin with lectures on fundamentals, and toward the end shift to considering current issues and research directions. The course is suitable for students with an interest in research or in tackling problems that are not well-solved. A strong discrete math background is recommended. Ideally students should have taken CMPT-307, or at least have MACM-201 plus some exposure to complexity theory.

Prerequisites:
CMPT 225.

Topics:
- Intractable search and optimization problems; Search algorithms.
- Constraint modelling languages; Model-and-solve systems.
- Basic model theory and first order logic.
- SAT, SAT algorithms, reductions, SAT solvers;
- Logic, databases, query languages, and problem specifications.
- Treewidth, proof complexity, fixed-parameter tractability.
- Applications in science (e.g. bio-informatics), hardware and software, business, etc.

Grading:
(Preliminary) Assignments (40%), Midterm Tests (40%), Project (20%).

Academic Honesty Statement::
Academic honesty plays a key role in our efforts to maintain a high standard of academic excellence and integrity. Students are advised that ALL acts of intellectual dishonesty will be handled in accordance with the SFU Academic Honesty and Student Conduct Policies (http://www.sfu.ca/policies/gazette/student.html).