CMPT 827 - G100 Intelligent Systems

Instructor(s): David Mitchell  

Calendar Objective/Description:
Intelligent systems are knowledge-based computer programs which emulate the reasoning abilities of human experts. This introductory course will analyze the underlying artificial intelligence methodology and survey advances in rule-based systems, constraint solving, incremental reasoning, intelligent backtracking and heuristic local search methods. We will look specifically at research applications in intelligent scheduling, configuration and planning. The course is intended for graduate students with a reasonable background in symbolic programming.

Instructor's Objectives:
The course will examine modern methods for satisfiability-based (aka constraint-based) problem solving. Emphasis is on problems which are apparently intractable (i.e., NP-hard), and are computationally challenging in practice. Examples include AI planning, synthesizing or checking correctness of a hardware and software systems, constructing timetables and schedules, etc. The challenge of finding good algorithms for these problems, along with the wide range of applicability, has motivated a large body of research, theoretical and applied, and led to a number of software solver competitions. We will study representation of a problem or domain as a set of constraints, methods for constructing effective solvers, and aspects of the power and limitations of these methods. The course will begin with lectures on fundamentals, and evolve to study of original research papers reflecting recent key research directions and interests or projects of students. Students may choose projects involving theory (e.g., algorithms and complexity), applications (e.g., problems from AI, computational biology, engineering, etc.), or system development (e.g., use of multi-core or GPU, new heuristics).

Prerequisites:
None

Topics:
- Combinatorial search problems, constraint modelling languages, search algorithms.
- Propositional logic, problem representation, SAT algorithms.
- Hard and easy instances, resolution, complexity, tree-width.
- Solver design, implementation, and evaluation; solver competitions.
- First-order logic, specifications, grounding.
- Related technologies: CLP, ASP, integer programming.
- Applications: planning, configuration, temporal reasoning, phylogeny, etc.

Grading:
Assignments (30%), Midterm Test (20%), Project (50%).

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).