NSERC Undergraduate Student Research Award
2019

For further information:
http://www.nserc-crsng.gc.ca/Students-Etudiants/UG-PC/USRA-BRPC_eng.asp

Scholarships are valued at $7840 (includes $4500 from NSERC and minimum $3340 from the faculty supervisor)

The Department of Computing and Software is accepting applications for the following projects. Applications must be received in the CAS Departmental Office (ITB/202) no later than **Monday, February 11, 2019**. Applications must include Part 1 of Form 202 (Application for an Undergraduate Student Research Award), available on the NSERC website, and official transcripts. The form must be completed electronically by logging into the NSERC website, then printed and signed prior to submission to the departmental office. The departmental submission should include a separate note indicating which project(s) you are interested in.

****Project # 1 (A. Deza): Computational, Combinatorial, and Geometric Aspects of Linear Optimization**

Rational decision-making through quantitative modelling and analysis is the guiding principle behind optimization, a field with several far-reaching applications across engineering, sciences, and industry. In many cases, the problems can be formulated, or approximated, as linear optimization problems that involve maximizing or minimizing a linear function over a domain defined by a set of linear inequalities. The algorithmic issues are closely related to the combinatorial and geometric structure of the feasible region. The project focuses on the analysis of worst-case constructions leading to computationally highly challenging instances. The tasks of the student involve combining theoretical and computational approaches.

Required qualifications:
Strong computational and mathematical skills

****Project # 2 (W. Farmer): HOL Light QE**

Philosophers, logicians, and computer scientists have long been interested in metareasoning, that is, reasoning about reasoning itself. Reasoning is performed in a formal logic by manipulating symbols. Hence metareasoning in a formal logic involves reasoning about the manipulation of symbols. Traditional logics like first-order logic and simple type theory (a classical form of higher-order logic) do not provide much support for reasoning about symbol manipulation. To overcome this deficiency, we have developed a version of simple type theory called STT_qe [2] that includes quotation and evaluation operators inspired by the quote and eval operators in the Lisp programming language. With these operators it is possible to reason about the interplay of the manipulation of symbols and what the manipulations mean mathematically.

HOL Light QE [1] is an implementation of STT_qe produced by modifying HOL Light, a software system developed by John Harrison at the Intel Corporation that assists users in proving mathematical conjectures in a version of simple type theory. HOL Light is open source software written in OCaml hosted on GitHub at https://github.com/jrh13/hol-light/. Although it is a relatively small system, it has

been used to formalize many kinds of mathematics and to proof-check many proofs including the lion's share of Tom Hale's proof of the Kepler conjecture.

Equipped with quotation and evaluation operators, HOL Light QE is intended to be an experimental system for defining syntax-based mathematical algorithms and for stating, proving, and applying formulas that specify the mathematical meaning of these algorithms. Examples of syntax-based mathematical algorithms include algorithms that compute arithmetic operations by manipulating numerals, linear transformations by manipulating matrices, and derivatives by manipulating functional expressions. The HOL Light QE system is available at https://github.com/JacquesCarette/hol-light-qe

The purpose of this project is to (1) continue the development of HOL Light QE and (2) demonstrate its expressive power by defining and reasoning about several syntax-based mathematical algorithms in it. The project will involve functional programming using the OCaml programming language, formalizing symbolic algorithms using quotation and evaluation, and proving theorems in higher-order logic.

References:

[1] J. Carette, W. M. Farmer, and P. Laskowski, "HOL Light QE", in: J. Avigad and A. Mahboubi, eds., Interactive Theorem Proving, LNCS 10895:215-234, 2018. http://imps.mcmaster.ca/doc/hol-light-qe.pdf

[2] W. M. Farmer, "Incorporating Quotation and Evaluation into Church's Type Theory", Information and Computation, 260:9-50, 2018. http://imps.mcmaster.ca/doc/qe-in-church.pdf

Qualifications:

A solid background in computer science or software engineering is required for this project as well as a strong interest in mathematics, logic, and formal proofs. Knowledge of functional programming and higher-order logic would be considered a significant plus.

**Project # 3 (S. Smith & J. Carette): Generate All Things

Interested in novel approaches to software design? If so, then you're the person we're looking for! Our team is testing a new approach to software development centered on using 'recipes' to assemble knowledge from a central repository to form all required software artifacts, from requirements documents to tests and, of course, code.

We have several examples, but they need to be rebuilt using our approach. And we need help to do this.

Main requirement: a willingness to learn new ideas and new programming languages. Our main infrastructure is in Haskell, we generate code in Python, C++, Java, Lua (and more, and trying to add more), as well as documentation in LaTeX and HTML. We also have several embedded DSLs.

All our work is publicly available:
https://github.com/JacquesCarette/Drasil

Your contributions would then become part of your portfolio for future employers or graduate supervisors.

**Project # 4 (D. Down & G. Badawy): Studying Pedestrian Dynamics

Pedestrian behavior has been persistently studied for approximately a century. For a very long time using mounted cameras was the only possible way to study pedestrian behavior in different environments until recently where tracking through Wi-Fi seems to be gaining more popularity. This popularity comes from the ubiquity of modern smartphones, of which it is known that most have their Wi-Fi enabled all the time. However, no studies have been conducted that compare the differences between data gathered by cameras and those gathered using Wi-Fi. In this project we want to investigate this correlation. As a first step, a system comprised of both hardware and software will be developed, that can correlates the data gathered by Wi-Fi (i.e. the number of independent MAC addresses scanned by the wireless scanner) to the number of individuals passing through the area using image acquisition and processing, and generate heuristic models of such correlation. Then we will use the collected data to define predictive and decision support capacities, e.g., the ability to predict foot-traffic, or perform what-if analyses, by learning from large datasets in identifying features using machine learning.

**Project #5 (R. Zheng & G. Badawy): Data Centre Monitoring Framework

Data Centres (DCs)consume up to 3% of the energy produced worldwide, 80% of which is wasted, and the consumption doubles every five years. This wastage stems from the lack of real-time visibility of fine-grained thermal distribution and power consumption of IT equipment in DC. Cooling systems in many data centers are often overcompensated by operating at a low temperature setting and the highest fan speed, to reduce the creation hotspots. In this project we will build an autonomous monitoring system that will adaptively streamline DCs performance, and perform/trigger predictive maintenance efforts when the need arises is necessary to improve its energy efficiency. The monitoring application will collect environmental information from a wireless sensor network, IT load and power consumption information from power distribution systems and cooling information from cooling units. The data that this system will collect will help DC operators efficiently manage the data center, visualize and implement system changes that reduce operating and capital costs, increase capacity and reduce failures. The data can be extremely useful as data centers start to implement more sophisticated cooling control to accommodate environmental and workload changes.

**Project # 6 (R. Zheng): Quantifying Me

The aim of the project is to develop a software solution to collect, visualize, process, analyze (using machine learning) data from multiple wearable sensors (mbientlab) attached to different positions of a person’s body and from a motion capture system (optitrack) during physical exercises and activities. Students will work closely with graduate students and have the opportunities to interact with medical professionals.

Language: Familiarity with Python, C#/Java

**Project # 7 (F. Chiang): The Data Science Lab

The Data Science Lab (http://db.cas.mcmaster.ca) seeks students to build the next generation of data profiling and data preparation tools to improve data quality over enterprise and time series data. The project will involve: (i) working closely with graduate students on algorithm design involving probabilistic graphical models and transfer learning; (ii) implementing and integrating the algorithms with existing systems; and (iii) extensive system testing for correctness and efficiency.

Requirements: undergraduate databases course equivalent to CS/SE 3DB3, strong algorithms and data structures background, and programming proficiency in Python and Java.

Questions can be directed to Fei Chiang (fchiang@mcmaster.ca)

**Project # 8 (M. Lawford): Centralized Automotive Powertrain E/E Architectures

The project strives to evaluate various aspects of centralized automotive E/E (Electric/Electronic) architectures, including hardware support, communication protocols, functional safety, and software quality.

The work will involve porting existing powertrain controllers of the industrial partner's hybrid electric vehicle to new hardware platforms and bringing up stacks for communication between them. The resulting prototype of the powertrain E/E architecture will then be evaluated for different communication protocols and various schemes of allocation of software components to available hardware units.

Experience with C programming of embedded systems, Matlab/Simulink, and programming in C# is an asset.

**Project # 9 (M. Lawford): C Code Refactoring

The project involves tool-supported analysis of the C implementation of a module of a motor controller for the industrial partner's hybrid electric vehicles. Based on the results of the analysis, the C code will be refactored. The refactored code will then be translated to Simulink/Stateflow. Test harnesses will be developed to verify that the refactored C code and resulting Simulink/Stateflow models are functionally equivalent to the original C implementation.

Experience with C is required. Experience with Simulink/Stateflow is an asset.

**Project # 10 (M. Lawford): Development of Tools for Simulink

The project entails development of tools to help model based design in Simulink.

The work on the project involves developing and maintaining tools that analyze and refactor Simulink models. The tools are largely developed in Matlab, with some functionalities developed in Python and in future, potentially, in other languages. The tools are tested on large production-scale models and integrated into the industrial partner's tool chain.

Experience with Matlab/Simulink is an asset.

Project # 11 (M. Lawford): Model Based Design of an Autonomous Driving Testbed

This project involves the creation of a scale model autonomous driving testbed using multiple RC cars on a test track at the McMaster Automotive Resource Centre (MARC).

The work will involve developing autonomous vehicle controls software in Matlab/Simulink and C/C++ code to create a testbed for easy experimentation with autonmous vehcile controls algorithms. The work will also involve the development of teaching material to allow middle school and high school students to experiment with their own algorithms dealing with, e.g. collision free intersection navigation. They goal is to get students interested in careers in STEM in general and autonomous driving in particular.

Experience with C/C++ programming of embedded systems, Matlab/Simulink and image processing is an asset.

Project # 12 (E. Sekerinski): Software Development for Water Quality Sensor Networks

Two summer positions are available for a collaborative project that involves a local industrial partner as well as several communities in Canada that are affected by poor water quality. This project uses small low-power wireless sensor nodes (Arduino, Raspberry Pi) to which various water quality sensors (e.g. conductivity, turbidity, dissolved oxygen) are attached (some sensors are commercial and some are developed by other research groups at McMaster). The research aspects of the project are code generation from models (as well as correctness and quantitative analysis from the models) and using machine learning for analyzing the data (e.g. detecting deterioration of the sensors, abnormal events). The tasks involve developing software for visualizing and transmitting the data on nodes, interfacing of sensors, interfacing to a database and web server. While the task will be focused, this summer job allows to get rather broad experience and interact with other summer and graduate students and well as other researchers and users.

Required qualifications include a strong programming background and the willingness to catch up with various technologies used. Experience with hardware interfacing, Arduino, databases, web sites is welcome.

Project # 13 (W. Kahl): Extending the Teaching Tool CalcCheck

CalcCheck (http://CalcCheck.McMaster.ca/) is a proof checker for calculational proofs in the logics of the popular textbook ``A Logical Approach to Discrete Math'' (LADM) by David Gries and Fred Schneider currently used in COMPSCI&SFWRENG 2DM3.

CalcCheck is implemented in the functional programming language Haskell, with the web application aspects implemented in Haste. Although the system is already useful in its current state, there is ample room for improving the capabilities and the user experience, which is what this project plans to address. Since CalcCheck is used by students throughout their learning process, for preacticing on exercises as well as for completing and submitting assignments and examinations, there is considerable potential for collecting data that can be used in SoTL (scholarship of teaching and learning) projects; such data collection however demands more systematic and onfigurable logging capabilities than the system currently has.

This project requires strength and interest in logics and discrete mathematics.
Some previous knowledge of Haskell is necessary.

The student's role will be to add functionality to CalcCheck, initially including at least some of server logging and access control, web interface improvements, automated exercise and test problem generation. In each case, this involves all of design, implementation, and documentation. After familiarisation with the system via these smaller projects, larger projects currently envisaged include support for checking program correctness proofs, and improved integration with Avenue conforming to the LTI standards.

CalcCheck: http://CalcCheck.McMaster.ca/

Haskell: http://haskell.org/
Haste: https://haste-lang.org/

****Project # 14 (W. Kahl): Utility Libraries for Certified Programming in Agda**

The dependently-typed programming language and proof checker Agda2 allows us to write functional programs that include their correctness proofs as part of the source code.

As part of an ongoing effort aiming to produce verified Agda implementations of general graph transformation mechanisms, this project will produce self-contained auxiliary libraries, including input and output facilities, file handling and data interchange, graph layout and display.

This project requires strength and interest in logics and discrete mathematics.
Some previous knowledge of Agda or Haskell is necessary.

The student's role will be to produce libraries satisfying specific project needs.
In each case, this involves all of design, implementation, and documentation.

Project home: http://relmics.mcmaster.ca/RATH-Agda/

Agda: http://wiki.portal.chalmers.se/agda/