Department of Computer Science

HOFSTRA UNIVERSITY

Fall 2008 Colloquium Series

A Unified Sequent Calculus for Focused Proofs*

Speaker: Prof. Chuck Liang

Date: December 2, 2008

Time: 2:30 pm – 3:30 pm

Venue: Adams Hall, Room 018

Abstract: Computer science traces its roots to important advancements in the study of mathematical logic. In the nineteenth and early twentieth century thinkers such as Frege, Brouwer, Russell, Whitehead, and Godel elevated the study of logic into a mathematical science. In the 1930s, the German logician Gehard Gentzen invented the *sequent calculus*, which formulates logical reasoning as symbolic computation. That is to say, logical reasoning was consolidated into the mechanical application of a certain set of rules. Today the sequent calculus is used in computer science to provide logical foundations for programming languages, database systems, as well as expert systems in artificial intelligence.

Logic as studied in computer science comes in a variety of flavors. In *classical* logic, which is the most commonly known, "not false" is equated with "true". In *intuitionistic* and *linear* logics, stronger criteria are required to establish the truth of statements. For example, in classical logic the formula "A implies B" is merely another way to say "not-A or B". But, in intuitionistic (and linear) logic, proving such a formula means finding a mathematical function that maps domain A to co-domain B. These stronger logics are called "constructive" because more meaningful information can be extracted from proofs. These varieties of logics are typically studied separately from each other.

Prof. Chuck Liang will present a new system called *LKU*, which unifies classical, intuitionistic and linear logic in to a common, *universal sequent calculus*. The different varieties of logic are represented by adding restrictions to the common system. Furthermore, proofs in these logics may communicate through a process called cut-elimination. For example, an intuitionistic proof may use a linear lemma and vice-versa. Central to the system is a phenomenon called "polarization and focalization", which divides a formal proof into alternating *positive* and *negative* stages. Studying logic in this setting sheds light to the question of how precisely proofs correspond to computation. We see, for example, that classical proofs can also be constructive under the right circumstances. Many common computer science concepts, such as the difference between synchronous and asynchronous computation, and between forward and backward-reasoning systems, can now be seen as differences in how formal proofs are constructed.

Speaker's Biography: Prof. Chuck Liang wrote his first program on an Apple][computer in 1980. He graduated from college with a double-major in computer science and mathematics from the University of Oregon in 1989. He then acquired a Ph.D. in computer and information science from the University of Pennsylvania in 1995. He has taught at the University of Delaware, Frostburg State University and Trinity College before coming to Hofstra in 2000. He is currently associate professor of computer science. Prof. Liang endeavors to be a well-rounded computer scientist. In addition to the core areas of logic and programming language theory, he is also interested in networking and computer architecture among other subjects.

* This work has been done in collaboration with Prof. Dale Miller, Laboratoire d'Informatique, INRIA / Ecole Polytechnique, Palaiseau, France.

Refreshments will be served in Adams Hall, Room 017!

For more information about this colloquium, please contact Habib M. Ammari at cschma@hofstra.edu.