



# UMass Lowell Computer Science Colloquium Announcement

**Speaker:** Professor Ray Gumb  
Dept of Computer Science, UMass Lowell

**Date:** Feb. 20, 2002

**Time:** 3:00pm--4:00pm

**Place:** Olsen 311 (The Media Lab)  
Refreshments are served at 2:30pm

---

## LOGIC FOR SPECIFYING AND VERIFYING PROGRAMS, PART II

The Logic of Partial Terms LPT was the subject of Part I of this talk. LPT is a strict negative free logic that provides an economical framework for developing many traditional mathematical theories having partial functions. In these traditional theories, all functions and predicates are strict. For example, if a unary function (predicate) is applied to an undefined argument, the result is undefined (respectively, false). On the other hand, every practical programming language incorporates at least one nonstrict or lazy construct, such as the if-then-else, but nonstrict functions cannot be either primitive or introduced in definitional extensions in LPT. Consequently, lazy programming language constructs do not fit the traditional mathematical mold inherent in LPT. A nonstrict (positive free) logic is required to handle nonstrict functions and predicates.

Previously developed nonstrict logics are not fully satisfactory because they are verbose in describing strict functions (which predominate in many programming languages), and some logicians find their semantics philosophically unpalatable. The newly developed Lazy Logic of Partial Terms LL is as concise as LPT in describing strict functions and predicates, and strict and nonstrict functions and predicates can be introduced in definitional extensions of traditional mathematical theories. LL is "built on top of" LPT, and, like LPT, admits only one domain in the semantics. In the semantics, for the case of a nonstrict unary function  $h$  in an LL theory  $T$ , we have  $T \models h(\_)=y \leftrightarrow (Ax)(h(x)=y)$ , where  $\_$  is a canonical undefined term. Correspondingly, in the axiomatization, the "indifference" (to the value of the argument) axiom  $h(\_)=y \leftrightarrow (Ax)(h(x)=y)$  guarantees a proper fit with the semantics. The price paid for LL's naturalness is that it is tailored for describing a specific area of computer science, program specification and verification, possibly limiting its role in explicating classical mathematical and philosophical subjects.

---

Colloquium Coordinator: Jie Wang, [wang@cs.uml.edu](mailto:wang@cs.uml.edu)  
Website: <http://www.cs.uml.edu/~wang/colloquia/>