

# Our LIPS Are Sealed: Interfacing Logic and Functional Programming Systems

Gary Lindstrom<sup>1</sup>  
Jan Małuszyński<sup>2</sup>  
Takeshi Ogi<sup>3</sup>

UUCS-92-009

Department of Computer Science  
University of Utah  
Salt Lake City, UT 84112 USA

March 23, 1992

## Abstract

We report on a technique for interfacing an untyped logic language to a statically polymorphically typed functional language. Our key insight is that polymorphic types can be interpreted as “need to know” specifications on function arguments. This leads to a criterion for liberally yet safely invoking the functional language to reduce application terms as required during unification in the logic language. This method, called P-unification, enriches the capabilities of each language while retaining the integrity of their individual semantics and implementation technologies. An experimental test has been successfully performed, whereby a Horn clause logic programming (HCLP) interpreter written in COMMON LISP was interfaced to the STANDARD ML OF NEW JERSEY system. The latter implementation was employed (i) on untyped or dynamically typed data, even though it is statically typed; (ii) lazily, even though it is strict, and (iii) on alien HCLP terms such as unbound variables — without the slightest modification!

---

<sup>1</sup>This material is based upon work supported by the National Science Foundation under Grants CCR-8920971 and ASC-9016131.

<sup>2</sup>Department of Computer and Information Science, Linköping University, S-581 83 Linköping, Sweden.

<sup>3</sup>Kyocera Corp., 2-14-9 Tamagawa-dai, Setagaya-ku, Tokyo, Japan.