Logic Programming in a Fragment of Intuitionistic Linear Logic

by

Joshua Hodas, PhD student at UPenn
[now an attorney in Los Angeles]

and

Dale Miller, Edinburgh (sabbatical leave from UPenn)
[now INRIA]

Remembering the early 1990’s

Two new, exciting innovations:
- linear logic [1987]
- $\pi$-calculus [1989]

Many areas of computational logic, concurrency theory, and programming language semantics have been influence by them.

... but there was a steep learning curve.

Linear logic was strange: proof nets, slices, phase semantics, additive/multiplicativel/exponential connectives, etc.

The LICS 91 paper showed that
- logic programming became more expressive using linear logic, and
- linear logic programming had applications.
In the ’91 and ’94 papers

\[ \text{Lolli} = \{ T, \&, \forall, \supset \} \cup \{ \neg \circ \} \]

- Linear logic without exponentials: \( \text{LL} = \text{Lolli} \cup \{ \bot \} \)
- Completeness of “goal directed search”
- A polarized embedding of intuitionistic logic into linear logic (needs half as many exponentials).
- A canonical model given as a resource indexed Kripke model
- Lazy splitting of contexts
- Several applications.
The linear logic extension changed the minuses to pluses.
A word about the future (paraphrasing *The Graduate*)

**Mr. McGuire:** I just want to say one word to you. Just one word.

**Ben:** Yes, sir.

**Mr. McGuire:** Are you listening?

**Ben:** Yes, I am.

**Mr. McGuire:** *Focused proof systems*

**Ben:** But isn’t that three words?

Focused proof systems provide control of the structural rules without a direct appeal to linear logic. They provide remarkably flexible normal forms.

Completeness of a focusing proof systems is the *second* most important result about a sequent proof system for CS applications.

**Mr. McGuire:** But what is the most important result?

**Ben:** Cut-elimination, of course.

— Thank you —