Assignment 7

15-816: Linear Logic
Frank Pfenning

Due
Wednesday, May 2, 2012

This assignment consists of several, somewhat open-ended problems.

> **You should pick one of them, or any of the problems from Assignment 5 or Assignment 6 that you have not done yet.**

If you have proposed a project in Assignments 5 or 6, you should submit a final report.

You may do these assignments **by yourself or in pairs**. They are somewhat open-ended, so you have to use your judgment as to when you consider the homework completed. Feel free to contact the instructor when you have questions about the extent of a problem.

As usual, you are allowed and encouraged to use *all* resources (papers, lecture notes, technical reports) that you can find, but you must properly cite and acknowledge any resources you use.

Please submit this assignment as a PDF by email. LaTeX templates and macros that may be helpful are available on the course web pages, but you are not required to use them.

Exercise 1 (Reasoning with Resource Constraints) In the resource semantics for linear logic (either directly, or via its embedding into intutionistic logic), we have to solve equations over resource expressions

$$\text{Resources} \quad p \quad ::= \quad \epsilon \mid p_1 * p_2 \mid \alpha$$

that are subject to associativity and commutativity. Moreover, ϵ is the unit of resource combination.

$p * (q * r) = (p * q) * r$	Associativity
$p * q = q * p$	Commutativity
$p * \epsilon = \epsilon * p = p$	Unit

We have the additional invariant that no resource parameter α appears more than once in a resource expression.

(i) Define an algorithm for solving systems of equations between resources with free resource variables Q. Like the resource management system, this algorithm should not enumerate solutions, but determine whether solutions exist or not, and possibly allow computation of substitutions when and where they are uniquely determined.

(ii) Due to the order in which resource variables Q and resource parameters α are introduced in a proof, some dependencies must be respected. This can be captured by working in a constraint logic with the formulas

$$\text{Constraint Formulas} \quad F \quad ::= \quad p \doteq q \mid F_1 \wedge F_2 \mid \top \mid \forall \alpha.\, F \mid \exists Q.\, F$$

Extend your algorithm to this class.

(iii) It appears that for embedding the positive connectives of linear logic into intuitionistic logic, we needed constraint entailment rather than just satisfiability. Consider how to extend (i) or (ii) to handle constraint entailment. Feel free to take advantage of any special properties of the constraints arising from the embedding of linear logic.

Exercise 2 (Affine Resource Semantics) Modify the embedding of linear logic in intuitionistic logic to handle affine resources, in addition to linear ones. You may restrict yourself to the negative fragment, but you should give proofs of soundness and completeness.

Exercise 3 (Strict Resource Semantics) Modify the embedding of linear logic in intuitionistic logic to handle strict resources, instead of linear ones. You may restrict yourself to the negative fragment, but you should give proofs of soundness and completeness.

Exercise 4 (Double-Negation Translation) The relationship between classical and intuitionistic linear logic explored by Chang et al. [CCP03] almost completely characterizes a whole family of double-negation translations. However, one case (Section 4.3.4) is left open. Can you independently characterize this translation, analogous to the other ones?

References

[CCP03] Bor-Yuh Evan Chang, Kaustuv Chaudhuri, and Frank Pfenning. A judgmental analysis of linear logic. Technical Report CMU-CS-03-131R, Carnegie Mellon University, Department of Computer Science, December 2003.