Adding Linear Constrains over Real Number to Curry

Wolfgang Lux
Universität Münster
`http://danae.uni-muenster.de/~lux/`

Curry[Han99] is an integrated declarative language which combines features from lazy functional languages, logic languages, and constraint programming. A distinctive feature of Curry is the encapsulated search which allows the programmer to specify other algorithms than the usual depth first search to traverse the search space of a predicate.

The basic kernel of Curry only requires equality constraints to be implemented, but implementations are allowed to provide constraint solvers for other domains. In this paper we integrate a solver for linear constraints over real numbers into our implementation of Curry[Lux99]. The constraint solver is based on the well-known Simplex algorithm. Besides the usual *ask* and *tell* operations, which check for the entailment of a constraint and add new constraints to the constraint store, resp., we also need a *clone* operation which duplicates the current state of the constraint store. This is necessary because different states of the constraint store may exist in parallel in the encapsulated search.

References

[Han99] M. Hanus. Curry: An integrated functional logic language. `http://www-i2.informatik.rwth-aachen.de/~hanus/curry`

[Lux99] W. Lux. Implementing Encapsulated Search for a Lazy Functional Logic Language. *Proc. FLOPS 99*, pp. 100–113, Springer LNCS 1722, 1999.