TOY: A CFLP Language and System∗

I. Castiñeiras, J. Correas, S. Estévez-Martín, and F. Sáenz-Pérez

June 21, 2012

TOY is a Constraint Functional Logic Programming (CFLP) language and system [10, 11] that solves goals by means of a demand-driven lazy-narrowing strategy combined with constraint solving. Constraints are integrated as first class citizens in a language that intermixes features of Logic and Functional Programming: Logical variables, relational notation, non-determinism, backtracking, domain variables, functional notation, curried expressions, higher-order functions, patterns, partial applications, lazy evaluation, types, polymorphism and constraint composition. This language relies on strong mathematical foundations [9]. As a system, it is multiplatform (as long as supported by SICStus), open-source, distributed under GPL, has been interfaced to the ACIDE GUI, and enjoys declarative debugging [4] including totality constraints.

TOY core system supports several constraint domains which are built with SICStus technology: $H$ for symbolic equations and disequations [2], $R$ for arithmetic constraints over the real numbers [3], and $FD$ for finite domain constraints [8]. Furthermore, TOY enjoys a cooperation mechanism between solvers [7, 6]. The most related CFLP language is Curry and its implementations, as PAKCS [1] which also runs on SICStus. However, this system does not support constraint cooperation.

TOY has also been interfaced to external constraint solvers: (1) $FD$ and $FS$ (finite sets) from ECL’PS*, (2) $FD$ from IBM ILOG Solver, and (3) $FD$ from Gecode. Each of these interfaces has an external implementation (cf. [11]). There are two communication mechanisms with external solvers: For the first interface [6], ECL’PS* runs as a separate process and a simple communication protocol is built based on standard pipes. For the last two interfaces [5], the C++ interface is provided by each external solver to build a single application. In these systems, both batch (for pure CP applications) and online (for model reasoning) constraint solving are allowed.

Current trends for future work include improving performance for tackling real-life problems which are amenable to be modeled with constraints and their combinations in different domains.

∗This work has been partially supported by the Spanish projects STAMP (TIN2008-06622-C03-01) (first, second and fourth authors), DOVES (TIN-2008-05624) (third author), Prometidos-CM (S2009TIC-1465), and GPD (UCM-BSCH-GR35/10-A-910502).
References


