Tutorial on Constraint Reasoning and Programming

Thom Frühwirth
Ludwig-Maximilians-University Munich
Department of Computer Science
Oettingenstrasse 67, D-80538 Munich Germany
mailto:fruehwir@informatik.uni-muenchen.de
http://www.pst.informatik.uni-muenchen.de/~fruehwir/

The advent of constraints in logic programming (LP) is one of the rare cases where theoretical, practical and commercial aspects of a programming language have been improved simultaneously. Constraint logic programming (CLP) combines the advantages of logic programming and constraint solving. In logic programming, problems are stated in a declarative way using rules to define relations (predicates). Problems are solved by the built-in logic programming engine using chronological backtrack search to explore choices. In constraint solving, efficient special-purpose algorithms are employed to solve sub-problems involving distinguished relations referred to as constraints. CLP has proven its merits in a variety of application areas including decision support systems, especially in scheduling and resource planning.

We first introduce the basic ideas behind the family of (concurrent) constraint logic programming languages in a calculus-based framework. Constraint handling rules will be used to describe the constraint systems and their solver in a single high-level executable notation.

We will present shortly some of the most common constraint domains, their solvers and applications such as Boolean constraints for circuit design, linear polynomial equations for financial and engineering applications and finite domains for scheduling.

Literature