CS-Prolog:
A Generalized Unification Based Constraint Solver

Toshio KAWAMURA, Hayato OHWADA and Fumio MIZOGUCHI
Dept. of Industrial Administration, Science Univ. of Tokyo
Noda, Chiba 278, Japan

1. Introduction

Most of recent researches about logic programming present new logic programming language system as extension to PROLOG. Constraint Logic Programming is an attractive new approach. There are several approaches to it. For example, the integration of functional programming and logic programming, and introducing data types etc. However, there are same in the sense of Constraint Logic Programming. Jaffar [Jaffar 86] shows the common framework which he calls "CLP Schema" and gives us formal foundations for these approaches.

Unification can be viewed as equality constraint solver and we can use PROLOG as constraint solver. But it is not sufficient for using the unification of standard PROLOG (DEC10-PROLOG, C-PROLOG etc.) as constraint solver. For example, PROLOG can not handle equality between arithmetic terms (unify 1+2 with 3, obtain substitution x = -1 by unifying 3+x with 2). Generalized unification is a basic approach to constraint logic programming. For example PROLOG-II [Colmerauer 84] generalizes the unification to solve equations over the structure of infinite trees and PROLOG-Q [Jaffar 86b] generalizes the unification to solve linear arithmetic equations using Gaussian method.
We aim to develop a powerful constraint solver in logic programming framework. And we have been developing constraint logic programming language. Our design goal is as follows.

(i) The language must be a natural extension to PROLOG.
(ii) The language is based on the generalized unification.
(iii) The language is small, portable and easy to develop.
(iv) The language is efficient.

To achieve this design goal, we choose PROLOG for the target language for following reasons:

(i) Meta programming gives easy way to develop a new logic programming language.
(ii) Partial evaluation gives efficient program execution.
(iii) There is no effort to develop a basic logic programming feature such as syntactic equality, inference engine and memory management etc.
(iv) PROLOG (Quintus Prolog on Sun-3) has efficient compiler.

This paper describes a new constraint logic programming language CS-Prolog (Constraint Solver Prolog) and is organized as follows. Chapter 2 discusses the delay mechanism in standard logic programming language. The constraint is regarded as relationships over variables, and delay mechanism manipulate the constraint by its variable instantiation. Chapter 3 shows how to solve linear arithmetic equations. CS-Prolog solves it using 4 rules over PROLOG data structure (i.e. arithmetic terms). Domain concept is presented in [Dincbas 86], chapter 4 firstly gives this and we naturally extends it to real arithmetic. In real arithmetic "Domain Concept" can be viewed as inequation. It is difficult to solve inequations. Thus CS-Prolog has 2 versions. One is an ordering relation between two