Constraint Programming: Introduction

Eric MONFROY

IRIN, Université de Nantes

Introduction

Constraint programming

- an alternative approach to programming
- interleaves reasoning and computing
- constraint : a relation on variables and their domains
- Constraint Satisfaction Problem (CSP) : a set of constraints together with a set of variable domains

Constraint programming process

- modelling : formulate the problem with constraints as a CSP
- resolution : solve the CSP using constraint solvers
  - domain specific methods (Simplexe, Gröbner bases, . . .),
  - generic methods (constraint propagation),
  - or a combination of methods (solver cooperation)

programming=modelling
focus on WHAT and not on HOW

Programming (modelling)

Different programming paradigms(host languages) :

- imperative programming
  Pascal, C,. . .
- functional programming
  Lisp, Caml,. . .
- object-oriented programming
  C++, Java,. . .
- logic programming
  Prolog, Mercury,. . .

constraint programming

$\Rightarrow$ orthogonal to the programming paradigm

Solving

Solving CSPs can be :

- compute whether the CSP has a solution (is the CSP consistent ?)
- find A solution
- find ALL solutions
- find THE optimal solution (global optimum)
- find A good solution (local optimum)

Constraint solvers : domain specific methods

Algorithms devoted to specific variables and specific constraints

- program for solving systems of linear equations,
- package for linear programming,
- implementation of the unification algorithm,
- Gröbner bases computation,
- Simplexe like techniques,
- local search methods (borderline)
- Gaussian elimination, . . .

Constraint solvers : generic methods

general methods that can be adapted to several types of constraints and several types of variables

- constraint propagation algorithms
  to repeatedly remove inconsistent values from domains of variables

- search methods
  to explore the search space

Constraint programming : characteristics

- declarative programming : modelling
- flexible representation :
  constraints can be added, removed, or modified
- a two phase programming process :
  - generation of a problem representation as a CSP
  - solution of it
- support in the form of built-in mechanisms :
  - features to declare/define objects
  - features to set/generate constraints over these objects
  - constraint solvers
  - constraint propagation algorithms
  - search techniques

Constraint programming : summary

“Constraint programming represents one of the closest approaches computer science has yet made to the Holy Grail of programming: the user states the problem, the computer solves it.”

Eugene C. Freuder, CONSTRAINTS, Avril 1997

Constraint programming $=$ 2 level architecture :

- a language component
  (independant of the paradigm)
  $\Rightarrow$ add/remove constraints to the constraint store
- a constraint solver component
  solution of the problem by considering constraints in the store

Constraint programming : applications (1/2)

- operation research problems (optimization problems such as : scheduling, sequencing, ressource allocation, timetabling, job-shop, traveling salesman, . . .)
- electrical engineering (location of faults in circuits, circuit layout, verification of circuit design, . . .)
- business applications (option trading)
- graphic systems (to maintain consistency of moving objects, computer-aided design, . . .)
- molecular biology (search for patterns, 3D models of proteins, . . .)

Constraint programming : applications (2/2)

- natural language (efficient parsers, speach recognition with semantics, . . .)
- numerical computation (computation with guaranteed precision for chemistry, engineering, design, . . .)
- internet (constrained web queries)