An Introduction to Constraint Programming

Alan M. Frisch
Artificial Intelligence Group
Department of Computer Science
University of York

Updated 7 Oct 2013
Constraint Programming: The Topic
What is Constraint Programming?

Broad Answer

Programming where the use of **constraints** plays a central role.

- alternative to logic programming, functional programming, object-oriented programming

- There are **constraint programming languages** that support this

What is a **constraint**?

- Let \( X_1, X_2, \ldots, X_n \) be a finite sequence of variables, each associated with a domain, \( D_1, D_2, \ldots, D_n \).

- A constraint on \( X_1, X_2, \ldots, X_n \) is a relation \( D_1 \times D_2 \times \cdots \times D_n \).
What is Constraint Programming?

Narrow Answer

A more specific answer is obtained by programming with constraints in a particular manner.

Constraint programming involves solving a problem by:

1. Modelling: Formulate the problem as a finite set of constraints (a Constraint Satisfaction Problem). This is sometimes called reducing the problem (e.g. in COCO).

2. Solving: Solve the CSP, perhaps by using a constraint programming language

3. Mapping: Map the solution to the CSP to a solution to the original problem
Problems vs. Problem Instances

Graph Colouring Problem: Given any graph $G$ and any natural number $k$ determine if $G$ can be coloured with $k$ colours.

Instance of Graph Colouring Problem: a graph and a natural number.

A program to solve the graph colouring problem

- input: graph $G$ and a natural number $k$
- output: True/False indicating whether $G$ can be coloured with $k$ colours.
Constraint Programming Approach (Narrow)

A program to solve the graph colouring problem

- input: graph $G$ and natural number $k$
- step 1: generate a finite set $S$ of constraints such that $S$ is satisfiable iff $G$ can be coloured with $k$ colours
- step 2: determine if $S$ is satisfiable
- output: True if $S$ is satisfiable, False otherwise.
Constraint Programming: The Module
Theme of the Module

Most of the time we will take the “narrow” approach.

Focus on constraint programming with finite domains

- Use MiniZinc (a mid-level constraint modelling language) for solving finite-domain CSPs.
- Focus on solving combinatorial problems, usually ones that are NP-complete
- Emphasise the use of the technology to solve practical problems. Hands on!
- Hence, mostly focus on practical solution techniques and modelling rather than on theory and solution techniques that are rarely used in practice.
Topical Units of the Module

Unit 1: Introduction
• Introduction to the module and overview of the topic.

Unit 2: Building Correct Constraint Models
• How to take a combinatorial problem, formalize it, and build a correct constraint model of the problem.

Unit 3: Translating Constraint Models to CSPs
• How to translate a constraint model and instance data to a CSP instance.

Unit 4: Solving CSPs
• A look at the underlying problem-solving technology.

Unit 5: Building Efficient Constraint Models
• How to choose between alternative correct models and how to transform a correct model to an efficient model.
Organisation of Module

- Class time: 7 scheduled hours per week, for 4 weeks
  - Some time I will lecture
  - Some time on *interactively* solving problems

- Preparation: To learn the necessary skills, it is crucial that you seriously engage in the assigned work and come to class prepared to discuss your work.

- Open exam
  - Involves using the MiniZinc FD constraint modelling system to solve a problem.
  - Hand out: Friday, week 5 of Autumn term, immediately after last COPR class
  - Hand in: Friday, week 1 of Spring term
Module Prerequisites

• Experience with thinking about problems and algorithms to solve them.

• Problem representations, search spaces and basic search algorithms. For example, as presented in
  • The problem solving unit of ARIN module, or
  • Chps. 3 and 4.1 of the Russsell and Norvig book *Artificial Intelligence: A Modern Approach*
Resources

COPR web page: www-course.cs.york.ac.uk/copr

- Announcements
- Lecture notes and other materials for the module
- Past exams
- Online demos: www-course.cs.york.ac.uk/copr/demos
- Page of useful links: www-course.cs.york.ac.uk/copr/links
  - Tutorial papers
  - Information about topics such as MiniZinc

Books

- *Handbook of Constraint Programming.*

Google Group to be set up.