AINT503: Constraint Programming (1)

Dr Torsten Anders
Prof E R Miranda
Interdisciplinary Centre for Computer Music Research (ICCMR)
University of Plymouth
http://cmr.soc.plymouth.ac.uk/

25 November 2008

Goal of this Workshop (4 Sessions)

We will study a specific programming paradigm: Constraint Programming

- Discuss underlying theory
- Program in a small programming language for constraint programming
- Learn about application examples

Outline of this Lecture

Why Exist So Many Programming Languages? I

Question

What programming languages have you used before?
What languages you have heard of?

Why Exist So Many Programming Languages? II

Question

Why do we have so many programming languages?
How do programming languages differ?
What can be advantage of one language over the other?

Programming Paradigms I

Imperative Programming Computations execute commands and change memory values; paradigm similar to how computer hardware works

Object-Oriented Programming Computations model interaction of objects

Concurrent Programming Multiple computations run in parallel

Distributed Programming Multiple computations run in parallel on different processors or computers

Functional Programming Computations model mathematical functions

Logic Programming Computations model mathematical logic: relations between variables

Programming Paradigms II

Constraint Programming Computations model relations between variables (logic, numeric, . . .)

We study one paradigm: constraint programming

What is Constraint Programming?

CSP example (first-order logic notation)

$$X + Y = 7$$
$$\wedge\ X < Y$$
$$\text{where } X \in \{1, \ldots, 10\} \wedge Y \in \{1, \ldots, 10\}$$

possible solution: $X = 3, Y = 4$

Definitions

A *constraint satisfaction problem* (CSP) states *constrains* (mathematical relations) between *variables* (unknowns) with a specific *domain* (a set of possible variable values). Constrains specify properties of a solution, which is found by search.

What is Constraint Programming?

CSP example (first-order logic notation)

$$X + Y = 7$$
$$\wedge\ X < Y$$
$$\text{where } X \in \{1, \ldots, 10\} \wedge Y \in \{1, \ldots, 10\}$$

possible solution: $X = 3, Y = 4$

Definitions

A *constraint satisfaction problem* (CSP) states *constrains* (mathematical relations) between *variables* (unknowns) with a specific *domain* (a set of possible variable values). Constrains specify properties of a solution, which is found by search.

What is Constraint Programming?

CSP example (first-order logic notation)

$$X + Y = 7$$
$$\wedge\ X < Y$$
$$\text{where } X \in \{1, \ldots, 10\} \wedge Y \in \{1, \ldots, 10\}$$

possible solution: $X = 3, Y = 4$

Definitions

A *constraint satisfaction problem* (CSP) states *constrains* (mathematical relations) between *variables* (unknowns) with a specific *domain* (a set of possible variable values). Constrains specify properties of a solution, which is found by search.

Advantages of Constraint Programming

Advantages

- Complex problems simple to model: model states only *what*, not *how*
- There exist efficient solvers to search for solution(s)

Application Examples of Constraint Programming I

Constraint programming successfully applied to many different problem areas: many real-live problems naturally modelled by constraints

- Planning and Scheduling COSYTEC (`http://www.cosytec.com/`): Software company for constraint-based products solving highly complex problems in resources management & optimization
- Natural language processing D. Duchier (2000). Constraint Programming for Natural Language Processing[1]

[1] `http://www.ps.uni-sb.de/Papers/abstracts/duchier-esslli2000.html`

Application Examples of Constraint Programming II

- Graphics reactive systems[2]
- Graphical User interfaces Cassowary constraint solving toolkit[3]
- Games R. Collet: Minesweeper with Assistant :) [4]

[2] W. Hower and W. H. Graf. A bibliographical survey of constraint-based approaches to CAD, graphics, layout, visualization, and related topics. Knowledge-Based Systems, 9(7), 1996.

[3] `http://en.wikipedia.org/wiki/Cassowary_constraint_solver`

[4] `http://www.info.ucl.ac.be/~raph/minesweeper/`

Music Composition with Constraint Programming

My own research: constraint programming applied for music composition
`http://strasheela.sourceforge.net/strasheela/doc/StrasheelaExamples.html`

Introduction to Mini-Oz

- Oz is multi-paradigm programming language: supports all paradigms listed before
- Implementation of Oz is called Mozart
- We study a small subset of the Oz programming language, suitable for constraint programming: Mini-Oz (created for this workshop)

Variable Concept

Question

What is a variable

- in mathematics?
- in programming (most programming paradigms)?

What are variables good for?

Some variable purposes

- Make programs more general: replace actual value with a placeholder
- Make a program more easy to understand: give values a name

Variable Concept

Question

What is a variable

- in mathematics?
- in programming (most programming paradigms)?

What are variables good for?

Some variable purposes

- Make programs more general: replace actual value with a placeholder
- Make a program more easy to understand: give values a name

Variable Concept II

Variables in Oz share similarities with variables in mathematics

- The variable value can be unknown
- Variables cannot change their value: they are bound only once
- Constrained variables have a domain (set of possible values)

Variables in Oz I

Variable notation

Variables starts with capital letter. Examples:

```
X
MyVariable
```

Variables in Oz II

Global variables introduced with `declare`

```
declare
X = 1
```

Variables in Oz III

Local variables introduced with `local`, scope can be nested

```
local
   X = 1
in
   local
      Y = 2
   in
      X + Y
   end
end
```

Variables in Oz IV

Constrained variables

```
declare X
X :: 1#10
```

Procedure Concept

Question

What is a procedure (or subprogram, subroutine, function, method)?
What is it good for?

Procedures

- encapsulate computations: abstracts computations which belong together
- make a program more easy to understand: give complex computations a name
- reduce the size of a program: avoids code repetition
- make programs to data: programs (procedures) can create and process programs

Procedure Concept

Question

What is a procedure (or subprogram, subroutine, function, method)?
What is it good for?

Procedures

- encapsulate computations: abstracts computations which belong together
- make a program more easy to understand: give complex computations a name
- reduce the size of a program: avoids code repetition
- make programs to data: programs (procedures) can create and process programs

Procedures in Oz

Use procedure `Square`: $3^2 = X$

```
{Square 3 X}
```

Define procedure `Square`

```
proc {Square X Y}
   X * X = Y
end
```

Procedures in Oz

Use procedure `Square`: $3^2 = X$

```
{Square 3 X}
```

Define procedure `Square`

```
proc {Square X Y}
   X * X = Y
end
```

Constraints

Constraints are procedures, some support infix notation

Procedure call: constrain $max(X, Y) = Z$

```
{FD.max X Y Z}
```

Infix notation: constrain $X + Y = Z$

```
X + Y =: Z
```

Homework (Prepares Assessment)

- For 26th November (tomorrow): read Oz tutorial `MiniOz.html` (in Sharepoint folder)[5]
- For 2nd December: search for an application example of constraint programming, for example, from your subject area. You could use, e.g.,
  - `http://scholar.google.com`
  - `http://citeseer.ist.psu.edu`
  - `http://www.google.com`

[5] If you prefer trying out the examples: Oz is installed on computers in Smeaton 204 (program Mozart, under M). For installation at home read `MozartInstallationInstructions.html` (in Sharepoint folder)

Summary

- Different programming languages and paradigms are suitable for different problems
- Constraint programming good for solving combinatorial problems which require searching
- Programming language Mini-Oz supports constraint programming