#### Constraint Programming with CHR

#### Thom Frühwirth

Faculty of Computer Science University of Ulm, Germany

#### Uni Dortmund November 2005

Images are subject to copyright of the respective owners

Citations may be incomplete for space reasons

イロト イボト イヨト イヨト

San

Constraint Reasoning and Programming

# Part I

# **Constraint Programming**

1 Constraint Reasoning

2 Constraint Programming

マロト イラト イラト

# The Holy Grail



**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, Inaugural issue of the *Constraints Journal*, 1997.

- E - E

## Constraint Reasoning

#### The Idea



- Combination Lock Example 0 1 2 3 4 5 6 7 8 9 Greater or equal 5. Prime number.
- Declarative problem representation by variables and constraints:
   x ∈ {0,1,...,9} ∧ x ≥ 5 ∧ prime(x)
- Constraint propagation and simplification reduce search space:

 $x \in \{0, 1, \dots, 9\} \land x \ge 5 \ \to \ x \in \{5, 6, 7, 8, 9\}$ 

- 4 回下 - 4 回下 - 4 回下

## Constraint Reasoning

#### The Idea



- Combination Lock Example
   Ø 1 2 3 4 5 6 7 8 9
   Greater or equal 5.
   Prime number.
- Declarative problem representation by variables and constraints:
   x ∈ {0,1,...,9} ∧ x ≥ 5 ∧ prime(x)
- Constraint propagation and simplification reduce search space:

 $x \in \{0, 1, \dots, 9\} \land x \ge 5 \ \to \ x \in \{5, 6, 7, 8, 9\}$ 

## Constraint Reasoning

#### The Idea



- Combination Lock Example
   Ø 1 2 3 4 5 6 7 8 9
   Greater or equal 5.
   Prime number.
- Declarative problem representation by variables and constraints:
   x ∈ {0,1,...,9} ∧ x ≥ 5 ∧ prime(x)
- Constraint propagation and simplification reduce search space:

 $x \in \{0, 1, \dots, 9\} \land x \ge 5 \ \to \ x \in \{5, 6, 7, 8, 9\}$ 

## Constraint Reasoning Everywhere



Combination



#### Simplification



Contradiction



Redundancy

-

Sac

▲ □ ► ▲ □ ►

### Sudoku

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

c wikipedia.com

```
sudoko(Matrix) <=>
gendomain(Matrix,1..9),
Matrix=[A1,A2,...,A9,B1,...,I9],
alldiff(A1,A2,A3,A4,A5,A6,A7,A8,A9),
...,
alldiff(A1,B1,C1,D1,E1,F1,G1,H1,I1),
...,
alldiff(A1,B1,C1,A2,B2,C2,A3,B3,C3),
...,
labeling(Matrix).
```

・ 同下 ・ ヨト ・ ヨト

-

San

### Sudoku

5				7					
Ľ	Ŭ			•					
6			1	9	5-				
	9	8					6		
8				6					3
4			8		3			•	
7				2				(	;
	6					2	8		
			4	1	9				5
				8			7	Ş	Э

c wikipedia.com

```
sudoko(Matrix) <=>
gendomain(Matrix,1..9),
Matrix=[A1,A2,...,A9,B1,...,I9],
alldiff(A1,A2,A3,A4,A5,A6,A7,A8,A9),
...,
alldiff(A1,B1,C1,D1,E1,F1,G1,H1,I1),
...,
alldiff(A1,B1,C1,A2,B2,C2,A3,B3,C3),
...,
labeling(Matrix).
```

・ 同下 ・ ヨト ・ ヨト

-

San

### Mortgage

- D: Amount of Loan, Debt, Principal
- T: Duration of loan in months
- I: Interest rate per month
- R: Rate of payments per month
- S: Balance of debt after T months

```
mortgage(D, T, I, R, S) <=>
    T = 0,
    D = S
    ;
    T > 0,
    T1 = T - 1,
    D1 = D + D*I - R,
    mortgage(D1, T1, I, R, S).
```

 ∃ >

### Mortgage II

```
mortgage(D, T, I, R, S) <=>
    T = 0, D = S
    ;
    T > 0, T1 = T - 1, D1 = D + D*I - R,
    mortgage(D1, T1, I, R, S).
```

- mortgage(100000,360,0.01,1025,S) yields S=12625.90.
- mortgage(D,360,0.01,1025,0) yields D=99648.79.
- mortgage(100000,T,0.01,1025,S), S=<0 yields T=374, S=-807.96.
- mortgage(D,360,0.01,R,0) yields R=0.0102861198\*D.

- 4 目 1 4 日 1 4 H

# Advantages of Constraint Logic Programming

#### Theoretical

Logical Foundation – First-Order Logic

Conceptual Sound Modeling

#### Practical

Efficient Algorithms/Implementations Combination of different Solvers

(4月) (4日) (4日)

San

# Early Commercial Applications (in the 90s)

- Lufthansa: Short-term staff planning.
- Hongkong Container Harbor: Resource planning.
- Renault: Short-term production planning.
- Nokia: Software configuration for mobile phones.
- Airbus: Cabin layout.
- Siemens: Circuit verification.
- Caisse d'epargne: Portfolio management.

In Decision Support Systems for Planning and Configuration, for Design and Analysis.

・ 何ト ・ ヨト ・ ヨト

San

# Part II

# CHR Constraint Handling Rules

- 3 Constraint Handling Rules (CHR)
- Constraint Solvers
- **5** Program Analysis

- 4 周 ト 4 戸 ト 4 戸 ト

Example Partial Order Operational Semantics Operational Properties

#### Example Partial Order Constraint

 $\begin{array}{rcl} X \leq X & \Leftrightarrow & true & (reflexivity) \\ X \leq Y \wedge Y \leq X & \Leftrightarrow & X = Y & (antisymmetry) \\ X \leq Y \wedge Y \leq Z & \Rightarrow & X \leq Z & (transitivity) \end{array}$ 

 $\underline{A \leq B} \land \underline{B \leq C} \land C \leq A$   $\downarrow \qquad (transitivity)$   $A \leq B \land B \leq C \land \underline{C \leq A} \land \underline{A \leq C}$   $\downarrow \qquad (antisymmetry)$   $A \leq B \land B \leq C \land \underline{A = C}$   $\downarrow \qquad (built-in solven)$   $\underline{A \leq B} \land \underline{B \leq A} \land A = C$   $\downarrow \qquad (antisymmetry)$   $A = B \land A = C$   $\downarrow \qquad (antisymmetry)$ 

Example Partial Order Operational Semantics Operational Properties

#### Example Partial Order Constraint

 $\begin{array}{rcl} X \leq X & \Leftrightarrow & true & (reflexivity) \\ X \leq Y \wedge Y \leq X & \Leftrightarrow & X = Y & (antisymmetry) \\ X \leq Y \wedge Y \leq Z & \Rightarrow & X \leq Z & (transitivity) \end{array}$ 

 $\underline{A \leq B} \land \underline{B \leq C} \land C \leq A$   $\downarrow \qquad (transitivity)$   $A \leq B \land B \leq C \land \underline{C \leq A} \land \underline{A \leq C}$   $\downarrow \qquad (antisymmetry)$   $A \leq B \land B \leq C \land \underline{A = C}$   $\downarrow \qquad (built-in solver)$   $\underline{A \leq B} \land \underline{B \leq A} \land A = C$   $\downarrow \qquad (antisymmetry)$   $A = B \land A = C$   $\downarrow \qquad (antisymmetry)$ 

Example Partial Order Operational Semantics Operational Properties

#### Example Partial Order Constraint

 $\begin{array}{rcl} X \leq X & \Leftrightarrow & true & (reflexivity) \\ X \leq Y \wedge Y \leq X & \Leftrightarrow & X = Y & (antisymmetry) \\ X \leq Y \wedge Y \leq Z & \Rightarrow & X \leq Z & (transitivity) \end{array}$ 

 $\underline{A \leq B} \land \underline{B \leq C} \land C \leq A$   $\downarrow \qquad (transitivity)$   $A \leq B \land B \leq C \land \underline{C \leq A} \land \underline{A \leq C}$   $\downarrow \qquad (antisymmetr)$   $A \leq B \land B \leq C \land \underline{A = C}$   $\downarrow \qquad (built-in solve)$   $\underline{A \leq B} \land \underline{B \leq A} \land A = C$   $\downarrow \qquad (antisymmetr)$   $A = B \land A = C$ 

Example Partial Order Operational Semantics Operational Properties

#### Example Partial Order Constraint

 $\begin{array}{rcl} X \leq X & \Leftrightarrow & true & (reflexivity) \\ X \leq Y \wedge Y \leq X & \Leftrightarrow & X = Y & (antisymmetry) \\ X \leq Y \wedge Y \leq Z & \Rightarrow & X \leq Z & (transitivity) \end{array}$ 

 $\underbrace{A \leq B} \land \underline{B \leq C} \land C \leq A \qquad (transitivity) \\
A \leq B \land B \leq C \land \underline{C \leq A} \land \underline{A \leq C} \qquad \downarrow \qquad (antisymmetry) \\
A \leq B \land B \leq C \land \underline{A = C} \qquad \downarrow \qquad (built-in solver) \\
\underbrace{A \leq B} \land \underline{B \leq A} \land A = C \qquad \downarrow \qquad (antisymmetry) \\
A = B \land A = C$ 

Example Partial Order Operational Semantics Operational Properties

#### Example Partial Order Constraint

 $\begin{array}{rcl} X \leq X & \Leftrightarrow & true & (reflexivity) \\ X \leq Y \wedge Y \leq X & \Leftrightarrow & X = Y & (antisymmetry) \\ X \leq Y \wedge Y \leq Z & \Rightarrow & X \leq Z & (transitivity) \end{array}$ 

 $\underbrace{A \leq B} \land \underline{B \leq C} \land C \leq A \qquad (transitivity)$   $A \leq B \land B \leq C \land \underline{C \leq A} \land \underline{A \leq C} \qquad (antisymmetry)$   $\underbrace{A \leq B \land B \leq C \land \underline{A = C}}_{\downarrow} \qquad (built-in solver)$   $\underbrace{A \leq B} \land \underline{B \leq A} \land A = C \qquad (antisymmetry)$   $\underbrace{A = B \land A = C}_{A = C} \qquad (antisymmetry)$ 

Example Partial Order Operational Semantics Operational Properties

#### Example Partial Order Constraint

 $\begin{array}{rcl} X \leq X & \Leftrightarrow & true & (reflexivity) \\ X \leq Y \wedge Y \leq X & \Leftrightarrow & X = Y & (antisymmetry) \\ X \leq Y \wedge Y \leq Z & \Rightarrow & X \leq Z & (transitivity) \end{array}$ 

 $\begin{array}{cccc}
\underline{A \leq B} \land \underline{B \leq C} \land C \leq A \\
\downarrow & (transitivity) \\
\underline{A \leq B} \land \underline{B \leq C} \land \underline{C \leq A} \land \underline{A \leq C} \\
\downarrow & (antisymmetry) \\
\underline{A \leq B} \land \underline{B \leq C} \land \underline{A = C} \\
\downarrow & (built-in solver) \\
\underline{A \leq B} \land \underline{B \leq A} \land A = C \\
\downarrow & (antisymmetry) \\
\underline{A = B} \land A = C
\end{array}$ 

# **Operational Semantics**

#### Apply rules until exhaustion in any order (fixpoint computation).

#### Simplify

If  $(H \Leftrightarrow C \mid B)$  rule with renamed fresh variables  $\bar{x}$ and  $CT \models G_{builtin} \rightarrow \exists \bar{x}(H=H' \land C)$ then  $H' \land G \mapsto G \land H=H' \land B$ 

#### Propagate

If  $(H \Rightarrow C \mid B)$  rule with renamed fresh variables  $\bar{x}$ and  $CT \models G_{builtin} \rightarrow \exists \bar{x}(H=H' \land C)$ then  $H' \land G \mapsto H' \land G \land H=H' \land B$ 

イロト 不得 トイヨト イヨト

# **Operational Semantics**

Apply rules until exhaustion in any order (fixpoint computation).

#### Simplify

If  $(H \Leftrightarrow C \mid B)$  rule with renamed fresh variables  $\bar{x}$ and  $CT \models G_{builtin} \rightarrow \exists \bar{x}(H=H' \land C)$ then  $H' \land G \mapsto G \land H=H' \land B$ 

#### Propagate

If 
$$(H \Rightarrow C \mid B)$$
 rule with renamed fresh variables  $\bar{x}$   
and  $CT \models G_{builtin} \rightarrow \exists \bar{x}(H=H' \land C)$   
then  $H' \land G \mapsto H' \land G \land H=H' \land B$ 

イロト イ押ト イヨト イヨト

San

## **Operational Semantics**

Apply rules until exhaustion in any order (fixpoint computation).

#### Simplify

If  $(H \Leftrightarrow C \mid B)$  rule with renamed fresh variables  $\bar{x}$ and  $CT \models G_{builtin} \rightarrow \exists \bar{x}(H=H' \land C)$ then  $H' \land G \mapsto G \land H=H' \land B$ 

#### Propagate

If  $(H \Rightarrow C \mid B)$  rule with renamed fresh variables  $\bar{x}$ and  $CT \models G_{builtin} \rightarrow \exists \bar{x}(H=H' \land C)$ then  $H' \land G \mapsto H' \land G \land H=H' \land B$ 

・ロト ・ 同ト ・ 王ト ・ 王ト

San

#### Anytime Algorithm

Computation can be interrupted and restarted at any time. Intermediate results approximate final result.

 $\underline{A \leq B} \land \underline{B \leq C} \land C \leq A$   $\downarrow$   $A \leq B \land B \leq C \land \underline{C \leq A} \land \underline{A \leq C}$   $\downarrow$   $A \leq B \land B \leq C \land \underline{A = C}$   $\downarrow$   $\underline{A \leq B} \land \underline{B \leq A} \land A = C$   $\downarrow$   $A = B \land A = C$ 

(transitivity)

antisymmetry)

(built-in solver)

(antisymmetry)

### Anytime Algorithm

Computation can be interrupted and restarted at any time. Intermediate results approximate final result.

 $\underline{A \leq B} \land \underline{B \leq C} \land C \leq A$   $\downarrow$   $A \leq B \land B \leq C \land \underline{C \leq A} \land \underline{A \leq C}$   $\downarrow$   $A \leq B \land B \leq C \land \underline{A = C}$   $\downarrow$   $\underline{A \leq B} \land \underline{B \leq A} \land A = C$   $\downarrow$   $A = B \land A = C$ 

(transitivity)

(antisymmetry)

(built-in solver)

(antisymmetry)

### Anytime Algorithm

Computation can be interrupted and restarted at any time. Intermediate results approximate final result.

 $\begin{array}{cccc}
\underline{A \leq B} \land \underline{B \leq C} \land C \leq A \\
\downarrow & (transitivity) \\
\underline{A \leq B} \land \underline{B \leq C} \land \underline{C \leq A} \land \underline{A \leq C} \\
\downarrow & (antisymmetry) \\
\underline{A \leq B} \land \underline{B \leq C} \land \underline{A = C} \\
\downarrow & (built-in solver) \\
\underline{A \leq B} \land \underline{B \leq A} \land A = C \\
\downarrow & (antisymmetry) \\
\underline{A = B} \land A = C
\end{array}$ 

### **Online Algorithm**

The complete input is initially unknown.

The input data arrives incrementally during computation.

No recomputation from scratch necessary.

Monotonicity and Incrementality If  $G \longrightarrow G'$ then  $G \wedge C \longmapsto G' \wedge C$  $\underline{A \leq B} \land \underline{B \leq C} \land C \leq A$  $A \leq B \land B \leq C \land \underline{A \leq C} \land \underline{C \leq A}$ 

(transitivity)

イロト イヨト イヨト

### **Online Algorithm**

The complete input is initially unknown.

The input data arrives incrementally during computation.

No recomputation from scratch necessary.

 $\begin{array}{cccc} & \text{Monotonicity and Incrementality} \\ & \text{If} & G & \longmapsto & G' \\ & \text{then} & G \land C & \longmapsto & G' \land C \\ \\ & \underbrace{A \leq B} \land \underline{B \leq C} \land C \leq A \\ & \downarrow & & (\text{transitivity}) \\ & A \leq B \land B \leq C \land \underline{A \leq C} \land \underline{C \leq A} \\ & \downarrow & & (\text{antisymmet}) \\ & A \leq B \land B \leq C \land \underline{A = C} \\ & \downarrow & & \\ \end{array}$ 

### **Online Algorithm**

The complete input is initially unknown.

The input data arrives incrementally during computation.

No recomputation from scratch necessary.

 $\begin{array}{cccc} & \text{Monotonicity and Incrementality} \\ & \text{If} & G & \longmapsto & G' \\ & \text{then} & G \land C & \longmapsto & G' \land C \\ \\ & \underbrace{A \leq B} \land \underline{B \leq C} \land C \leq A \\ & \downarrow & (\text{transitivity}) \\ & A \leq B \land B \leq C \land \underline{A \leq C} \land C \leq A \\ & \downarrow & (\text{antisymmet}) \\ & A \leq B \land B \leq C \land \underline{A = C} \\ & \downarrow & \end{array}$ 

### **Online Algorithm**

The complete input is initially unknown.

The input data arrives incrementally during computation.

No recomputation from scratch necessary.

 $\begin{array}{ccc} \text{Monotonicity and Incrementality} \\ \text{If} & G & \longmapsto & G' \\ \text{then} & G \wedge C & \longmapsto & G' \wedge C \end{array}$ 

$$\underline{A \leq B} \land \underline{B \leq C} \land \underline{C \leq A}$$

$$A \leq B \land B \leq C \land \underline{A \leq C} \land \underline{C \leq A}$$

$$\downarrow$$

$$A \leq B \land B \leq C \land \underline{A \leq C}$$

$$\downarrow$$

$$A \leq B \land B \leq C \land \underline{A = C}$$

$$\downarrow$$

. . .

(transitivity)

(antisymmetry)

#### Concurrency

Rules can be applied in parallel to different parts of the problem.

$$\begin{array}{cccc} \text{If} & A & \longmapsto & B \\ \text{and} & C & \longmapsto & D \\ \text{then} & A \land C & \longmapsto & B \land D \end{array}$$



#### Concurrency

Rules can be applied in parallel to different parts of the problem.

$$\begin{array}{ccccc} \text{If} & A & \longmapsto & B \\ \text{and} & C & \longmapsto & D \\ \text{then} & A \wedge C & \longmapsto & B \wedge D \end{array}$$



#### Concurrency

Rules can be applied in parallel to different parts of the problem.

$$\begin{array}{ccccc} \text{If} & A & \longmapsto & B \\ \text{and} & C & \longmapsto & D \\ \text{then} & A \wedge C & \longmapsto & B \wedge D \end{array}$$



#### Concurrency

Rules can be applied in parallel to different parts of the problem.

$$\begin{array}{cccc} \text{If} & A & \longmapsto & B \\ \text{and} & C & \longmapsto & D \\ \text{then} & A \land C & \longmapsto & B \land D \end{array}$$



**Equations** of the form  $a_1x_1 + \ldots + a_nx_n + b = 0$ . **Solved form:** leftmost variable occurs only once. Reach solved normal form by **variable elimination**.

```
A1*X+P1=0 ∧ XP=0 ⇔
find(A2*X,XP,P2) |
compute(P2-(P1/A1)*A2,P3) ∧
A1*X+P1=0 ∧ P3=0.
```

 $B=0 \Leftrightarrow number(B) \mid zero(B).$ 

1\*X+3\*Y+5=0 ∧ 3\*X+2\*Y+8=0 compute((2\*Y+8) - ((3\*Y+5)/1)\*3,P3) % P3=-7\*Y+ -7 1\*X+3\*Y+5=0 ∧ -7\*Y+ -7=0 % Y=-1 compute((1\*X+5) - ((-7)/-7)\*3,P3') % P3'=1\*X+2 1\*X+2=0 ∧ -7\*Y+ -7=0 % X=-2

▲ロト ▲冊 ト ▲ ヨ ト ▲ ヨ ト ● の へ つ

**Equations** of the form  $a_1x_1 + ... + a_nx_n + b = 0$ . **Solved form:** leftmost variable occurs only once. Reach solved normal form by **variable elimination**.

```
A1*X+P1=0 ∧ XP=0 ⇔
find(A2*X,XP,P2) |
compute(P2-(P1/A1)*A2,P3) ∧
A1*X+P1=0 ∧ P3=0.
```

```
B=0 \Leftrightarrow number(B) \mid zero(B).
```

#### 1\*X+3\*Y+5=0 ∧ 3\*X+2\*Y+8=0

compute((2\*Y+8) - ((3\*Y+5)/1)\*3,P3) % P3=-7\*Y+ -7 1\*X+3\*Y+5=0 ∧ -7\*Y+ -7=0 % Y=-1 compute((1\*X+5) - ((-7)/-7)\*3,P3') % P3'=1\*X+2 1\*X+2=0 ∧ -7\*Y+ -7=0 % X=-2

◆□▶ ◆□▶ ◆□▶ ◆□▶ ●□ ● ●

**Equations** of the form  $a_1x_1 + ... + a_nx_n + b = 0$ . **Solved form:** leftmost variable occurs only once. Reach solved normal form by **variable elimination**.

```
A1*X+P1=0 ∧ XP=0 ⇔
find(A2*X,XP,P2) |
compute(P2-(P1/A1)*A2,P3) ∧
A1*X+P1=0 ∧ P3=0.
```

```
B=0 \Leftrightarrow number(B) \mid zero(B).
```

1\*X+3\*Y+5=0 ∧ 3\*X+2\*Y+8=0 compute((2\*Y+8) - ((3\*Y+5)/1)\*3,P3) % P3=-7\*Y+ -7 1\*X+3\*Y+5=0 ∧ -7\*Y+ -7=0 % Y=-1 compute((1\*X+5) - ((-7)/-7)\*3,P3') % P3'=1\*X+2 1\*X+2=0 ∧ -7\*Y+ -7=0 % X=-2

◆□▶ ◆□▶ ◆□▶ ◆□▶ ●□ ● ●

**Equations** of the form  $a_1x_1 + ... + a_nx_n + b = 0$ . **Solved form:** leftmost variable occurs only once. Reach solved normal form by **variable elimination**.

```
A1*X+P1=0 ∧ XP=0 ⇔
find(A2*X,XP,P2) |
compute(P2-(P1/A1)*A2,P3) ∧
A1*X+P1=0 ∧ P3=0.
```

 $B=0 \Leftrightarrow number(B) \mid zero(B).$ 

1\*X+3\*Y+5=0 ∧ 3\*X+2\*Y+8=0 compute((2\*Y+8) - ((3\*Y+5)/1)\*3,P3) % P3=-7\*Y+ -7 1\*X+3\*Y+5=0 ∧ -7\*Y+ -7=0 % Y=-1 compute((1\*X+5) - ((-7)/-7)\*3,P3') % P3'=1\*X+2 1\*X+2=0 ∧ -7\*Y+ -7=0 % X=-2

▲ロト ▲掃 ト ▲ ヨ ト ▲ ヨ ト 一 ヨ … の Q ()

# Confluence

Given a goal, every computation leads to the same result no matter what rules are applied.

A decidable, sufficient and necessary condition for confluence of terminating CHR programs through joinability of critical pairs.

 $\begin{array}{rcl} X \leq X & \Leftrightarrow & true & (reflexivity) \\ X \leq Y \wedge Y \leq X & \Leftrightarrow & X = Y & (antisymmetry) \end{array}$ 

Start from overlapping minimal states



# Confluence

Given a goal, every computation leads to the same result no matter what rules are applied.

A decidable, sufficient and necessary condition for confluence of terminating CHR programs through joinability of critical pairs.

 $\begin{array}{rcl} X \leq X & \Leftrightarrow & true & (reflexivity) \\ X \leq Y \wedge Y \leq X & \Leftrightarrow & X = Y & (antisymmetry) \end{array}$ 

Start from overlapping minimal states



# Confluence

Given a goal, every computation leads to the same result no matter what rules are applied.

A decidable, sufficient and necessary condition for confluence of terminating CHR programs through joinability of critical pairs.

 $\begin{array}{rcl} X \leq X & \Leftrightarrow & true & (reflexivity) \\ X \leq Y \wedge Y \leq X & \Leftrightarrow & X = Y & (antisymmetry) \end{array}$ 

Start from overlapping minimal states



## **CHR** Machine



c Jon Sneyers, K.U. Leuven

Analogous to RAM and Turing machines. CHR is turing-complete. Every algorithm can be implemented in CHR with best known time and space complexity. [Sneyers,Schrijvers,Demoen, CHR'05]

< 🗇 🕨

# Part III

### Applications



- Trends in Applications
- 8 Application Projects

イロト 不得下 イヨト イヨト

Sac

-

#### **POPULAR - Planning Cordless Communication**



T. Frühwirth, P. Brisset Optimal Placement of Base Stations in Wireless Indoor Communication Networks, IEEE Intelligent Systems Magazine 15(1), 2000.

Voted Among Most Innovative Telecom Applications of the Year by IEEE Expert Magazine, Winner of CP98 Telecom Application Award.

### MRA - The Munich Rent Advisor

	Nativegraphic Results Ner	refe Head Adverse	
£.,	10 2 <del>2</del> 2 2 2 2 2	2	N
	The Calculation Derived the l	Following Result:	
	Pype	Republic DM	
	Bent	between \$77.73 unit 1956.15	
	Rout of Front 'Kelcolovacos'	https://www.s81.46.and?68.05	
	"Nebon-setzn"	letween 296226 and 315466	
	tron if you any exampled all continue, it here the statistical workersed. We used the following inform	icie vili still besche in zociona nation you gave us:	
	traci Concerner assessed allocavina, d here due statistical worker and. We used the following inform Rada Information	inne välntäll leannar an seerivan Nation yoti gave us: Yeur Inpur	
	tern L'processer converse all'occasione, d incrès die matarical acodet scol. We used the following inform Bode Information Storetto die inspacemente	ine with till to scar an assistant istion you gave us: Your laper generation of 10	
	ten Contains converted discosting, d ine to de attacht i hardel hard. We used the following inform Role Information Size (Thar fist in upsemerter Yeur, in 40th the horse was ball.	inne will still fers care an acceleran nation your gave us: Your layer creates 651 and 18 between 1501 and 1500	
	tern Concurre converted discussion, di ne to die attachtal handel hand. We used the following inform Rade Information Size (True fist in oppermeren View, in «bich the house was built Narwärt de room.	inn vältsäll teistan an sociaan nation you gave teis: Veen legen somme Afrika Tei Seisens 1808 and 1900 Konens 2 and 2	

T. Frühwirth, S. Abdennadher The Munich Rent Advisor, Journal of Theory and Practice of Logic Programming, 2000. *Most Popular* 

Constraint-Based Internet Application.

< 🗇 🕨

- 4 E b

→ Ξ →

# University Course Timetabling



S. Abdennadher, M. Saft, S. Will Classroom Assignment using Constraint Logic Programming, PACLP 2000.

*Operational at University of Munich. Room-Allocation for 1000 Lectures a Week.* 

・ 同ト ・ ヨト・・

∃ >

Sac

Spatio-Temporal Reasoning Testing and Verification Agents and Actions Semantic Web

# Spatio-Temporal Reasoning



M. T. Escrig, F. Toledo, Universidad Jaume I, Castellun, Spain. Qualitative Spatial Reasoning: Theory and Practice, Application to Robot Navigation, IOS Press, 1998. Qualitative Spatial Reasoning on 3D Orientation Point Objects, QR2002.

Integrates orientation, distance, cardinal directions over points as well as extended objects.

・ロト ・ 同ト ・ ヨト ・ ヨト

- Spatio-Temporal Annotated CLP A. Raffaeta, Univ. Venice.
- Diagrammatic Reasoning B. Meyer, Monash Melbourne.
- RCC Reasoning B. Bennet, A.G. Cohn, Leeds UK.
- PMON logic for dynamical temporal systems E. Sandewall, Linkoeping Univ.
- GRF Temporal Reasoning G. Dondossola, E. Ratto, CISE Milano.

Spatio-Temporal Reasoning Testing and Verification Agents and Actions Semantic Web

#### Testing and Verification



Model Based Testing for Real: The Inhouse Card Case Study, A. Pretschner, O. Slotosch, E. Aiglstorfer, S. Kriebel, TU Munich, Journal on Software Tools for Technology Transfer (STTT) 5:2-3, Springer 2004.

- Automatic Generation of Test Data J. Harm, University Rostock, Germany.
- Executable Z-Specifications P. Stuckey, Ph. Dart, University Melbourne.

Spatio-Temporal Reasoning Testing and Verification Agents and Actions Semantic Web

#### Agents and Actions

FLUX: A Logic Programming Method for Reasoning Agents, Michael Thielscher, TPLP CHR Special Issue 2005. *Fluent Calculus, Reasoning about Actions, Robotics.* 

Specification and Verification of Agent Interaction... Alberti, Chesani, Gavanelli, Lamma, Mello, Torroni, ACM SAC 2004. Social integrity constraints on agent behaviour.



• Multi Agent Systems Using Constrains Handling Rules, IC-AI 2002 - B. Bauer, M. Berger, Siemens Munich, Germany - S. Hainzer, Uni Linz, Austria.

• PMON logic for dynamical temporal systems with actions and change - M. Bjgareland, E. Sandewall, Linkoeping University, Sweden.

・ロト ・ 同ト ・ ヨト ・ ヨト

Spatio-Temporal Reasoning Testing and Verification Agents and Actions Semantic Web

### Semantic Web



COIN Context Interchange Project, Stuart E. Madnick, MIT Cambridge. Reasoning About Temporal Context Using Ontology and Abductive CLP, PPSWR 2004 LNCS 3208.

Semantic Web Reasoning for Ontology-Based Integration of Resources, Liviu Badea, Doina Tilivea and Anca Hotaran, PPSWR 2004 LNCS 3208.

• S. Bressan, C.H. Goh, S. Madnick, M. Siegel et. al. Context Knowledge Representation and Reasoning in the Context Interchange System, Applied Intelligence, Vol 13:2, 2000; Context Interchange...for the intelligent integration of information, ACM Transactions on Information Systems, 1999.

イロト イヨト イヨト

JMMSolve Java Memory Machine Lung Cancer Diagnosis Cuypers Multimedia Web Presentation Manifico Business Rules for Optimization

#### Java Memory Machine

#### JMM by Vijay Saraswat, IBM TJ Watson Research and Penn State Univ. Implementation JMMSolve by Tom Schrijvers, K.U. Leuven, Belgium

Thread A y = 1 Thread B lock M Everything x = 1hefore the unlock on M. unlock M lock M ...visible to i = x everything after the lock on M unlock M i=v

Conditional Read Xr = (Cond)?Xw1:Xi ite(true,Xr,Xw1,Xi) <=> Xr = Xw1.

A E > 4

-

JMMSolve Java Memory Machine Lung Cancer Diagnosis Cuypers Multimedia Web Presentation Manifico Business Rules for Optimization

### Lung Cancer Diagnosis

Veronica Dahl, Simon Fraser University, Vancouver, Canada. Lung cancer is leading cause of cancer death, very low survival rate. Use bio-markers indicating gene mutations to diagnose lung cancer.



Lung Cancer and Metastasis

Concept Formation Rules (CFR) in CHR. Retractable constraints.

イロト イヨト イヨト イ

age(X,A),history(X,smoker), serum\_data(X,marker\_type) <=> marker(X,marker\_type,P,B), probability(P,X,B) | possible\_lung\_cancer(yes,X).

JMMSolve Java Memory Machine Lung Cancer Diagnosis Cuypers Multimedia Web Presentation Manifico Business Rules for Optimization

Multimedia Transformation Engine for Web Presentations

Joost Geurts, University of Amsterdam.

Automatic generation of interactive, time-based and media centric WWW presentations from semi-structured multimedia databases.





JMMSolve Java Memory Machine Lung Cancer Diagnosis Cuypers Multimedia Web Presentation Manifico Business Rules for Optimization

# Business Rules for Optimization



Business Rule: defines or constrains behavior or structure of business. "A car must be available to be assigned to a rental agreement".



DERBY EU Car Rent Case in CHR, O. Bouissou.

reservation(Renter,Group,From,To), available(car(Id,Group,...),From) <=>... rentagreement(Renter,Id,From,To).

JMMSolve Java Memory Machine Lung Cancer Diagnosis Cuypers Multimedia Web Presentation Manifico Business Rules for Optimization

### References

#### Google "constraint handling rules"



Essentials of Constraint Programming Thom Frühwirth, Slim Abdennadher Springer, 2003.

Constraint-Programmierung Lehrbuch Thom Frühwirth, Slim Abdennadher Springer, 1997.



I D > I D > I D > I D >