Constraint Programming
Sudoku, Schach und Stundenplanung

Thom Frühwirth

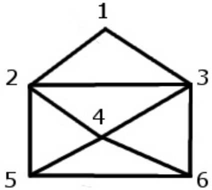
Faculty of Engineering and Computer Science
University of Ulm, Germany

Studium Generale
Ulm, November 2007

Part I

Constraints

Map-Coloring



Color neighbouring countries with different colors.

Map-Coloring

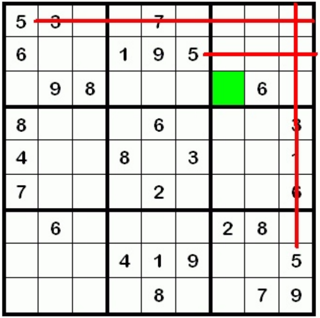

```
color(Map) <=>
  Map=[V1,V2,V3,...V6],
  domain(Map,[r,g,b]),
  alldiff(V1,V2,V3),
  alldiff(V2,V3,V4),
  ...,
  alldiff(V4,V5,V6),
  labeling(Map).

V1=r, V2=b, V3=g,...
```

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

Place numbers $1, \ldots, 9$ in each row, column and square.

Sudoku




```
sudoko(Matrix) <=>
  Matrix=[A1,A2,...,A9,B1,...,I9],
  domain(Matrix,1..9),
  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).

C7=5,...
```

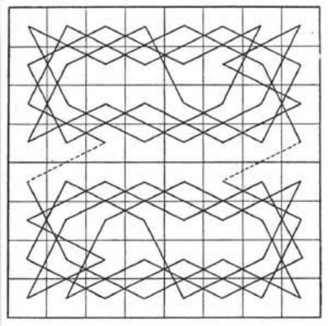
n-Queens Problem

Place n queens on an $n \times n$ chess board, such that they do not attack each other.

```
nqueens(List,N) <=>
  List=[A,B,C,D,E,...],
  domain(List,1..N),
  alldiff(A,B,C,D,E,...),
  alldiff(A,B+1,C+2,D+3,E+4,...),
  alldiff(A,B-1,C-2,D-3,E-4,...),
  labeling(List).
```

```
A=7, B=5, C=3,...
```

Knight's Tour Problem



Edouard Lucas: Récréations mathématiques
Bd. 4, Paris, 1894

- 9th century, Kashmiri poet Rudrata
- 1758, mathematician Euler
- 1823, Warnsdorff's algorithm

  Move to square with least possibilities

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:
  $x \in \{0, 1, \ldots, 9\} \;\wedge\; x \geq 5 \;\wedge\; \text{prime}(x)$
- *Constraint propagation and simplification* reduce search space:
  $x \in \{0, 1, \ldots, 9\} \wedge x \geq 5 \;\rightarrow\; x \in \{5, 6, 7, 8, 9\}$

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:
  $x \in \{0, 1, \ldots, 9\} \;\wedge\; x \geq 5 \;\wedge\; \text{prime}(x)$
- *Constraint propagation and simplification* reduce search space:
  $x \in \{0, 1, \ldots, 9\} \wedge x \geq 5 \;\rightarrow\; x \in \{5, 6, 7, 8, 9\}$

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:
  $x \in \{0, 1, \ldots, 9\} \ \wedge \ x \geq 5 \ \wedge \ \text{prime}(x)$
- *Constraint propagation and simplification* reduce search space:
  $x \in \{0, 1, \ldots, 9\} \wedge x \geq 5 \ \rightarrow \ x \in \{5, 6, 7, 8, 9\}$

Constraint Reasoning Everywhere

Combination

Simplification


Contradiction

Redundancy

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.

Part II

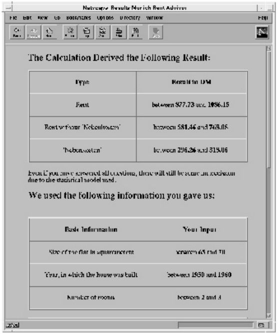
Applications

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**.

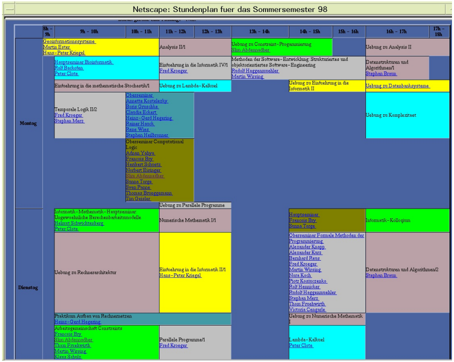
MRA - The Munich Rent Advisor



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

Most Popular Constraint-Based Internet Application.

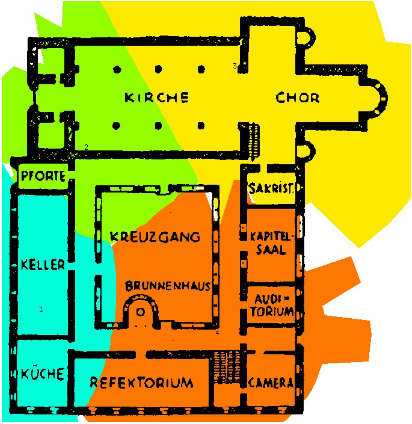
University Course Timetabling



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.

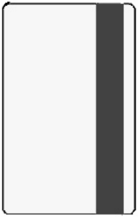
POPULAR - Planning Cordless Communication



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.

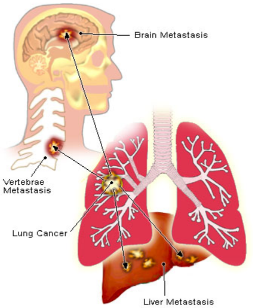
Testing and Verification



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.

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.

```
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).
```

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.

Powered by Cuypers

Finally...

驰

Finally...

驰

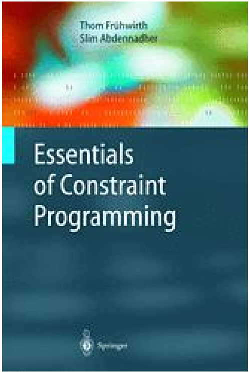
Transcribed as **CHR**, means

Finally...

驰

Transcribed as **CHR**, means
to speed, to propagate, to be famous

References



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

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

Early History of Constraint Programming

60s Constraint networks in artificial intelligence.
70s Logic programming (Prolog).
80s Constraint logic programming.
80s Concurrent logic programming.
90s Concurrent constraint programming.
90s Commercial applications.

Constraint Reasoning Algorithms

Adaption and combination of existing efficient algorithms from

- Mathematics
  - Operations research
  - Graph theory
  - Algebra
- Computer Science
  - Finite automata
  - Theorem proving
- Economics
- Linguistics

Application Domains

- Modeling
- Executable Specifications
- Solving Combinatorial Problems

Scheduling, Planning, Timetabling
Configuration, Layout, Placement, Design
Analysis: Simulation, Verification, Diagnosis
of **software, hardware and industrial processes**.

Applications in Research

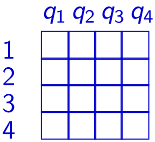
- Artificial Intelligence
  - Machine Vision
  - Natural Language Understanding
  - Temporal and Spatial Reasoning
  - Theorem Proving
  - Qualitative Reasoning
  - Robotics
  - Agents

Applications in Research II

- Computer Science: Program Analysis, Robotics, Agents
- Molecular Biology, Biochemestry, Bioinformatics:
  Protein Folding, Genomic Sequencing
- Economics: Scheduling
- Linguistics: Parsing
- Medicine: Decision Support
- Physics: System Modeling
- Geography: Geo-Information-Systems

n-Queens Problem

Place n queens $q_1, \ldots, q_n$ on an $n \times n$ chess board,
such that they do not attack each other.



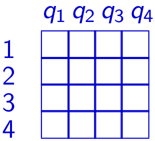
$$q_1, \ldots, q_n \in \{1, \ldots, n\}$$

$$\forall\ i \neq j.\ q_i \neq q_j \wedge |q_i - q_j| \neq |i - j|$$

- no two queens on same row, column or diagonal
  - each row and each column with exactly one queen
  - each diagonal at most one queen
- q_i: row position of the queen in the i-th column

n-Queens Problem II

Place n queens $q_1, \ldots, q_n$ on an $n \times n$ chess board,
such that they do not attack each other.



$$q_1, \ldots, q_n \in \{1, \ldots, n\}$$

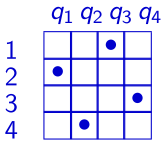
$$\forall\ i \neq j.\ q_i \neq q_j \wedge |q_i - q_j| \neq |i - j|$$

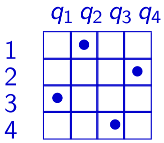
```
solve(N,Qs)         <=> makedomains(N,Qs), queens(Qs), enum(Qs).
queens([Q|Qs])      <=> safe(Q,Qs,1), queens(Qs).
safe(X,[Y|Qs],N)    <=> noattack(X,Y,N), safe(X,Qs,N+1).
noattack(X,Y,N)     <=> X ne Y, X+N ne Y, Y+N ne X.
```

n-Queens Problem III

`solve(4,[Q1,Q2,Q3,Q4])`

- `makedomains` produces possible positions for queens
  `Q1 in [1,2,3,4], Q2 in [1,2,3,4]`
  `Q3 in [1,2,3,4], Q4 in [1,2,3,4]`
- `safe` adds `noattack` for each ordered pair of queens
- `noattack` produces `ne` constraints between queens
- `enum` called for labeling using the domains of queens
- `[Q1,Q2,Q3,Q4] = [2,4,1,3]`, `[Q1,Q2,Q3,Q4] = [3,1,4,2]`





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.

Crypto-Arithmetic Problem

		S	E	N	D
+		M	O	R	E
=	M	O	N	E	Y

```
solve(S,E,N,D,M,O,R,Y) :-
      [S,E,N,D,M,O,R,Y] in 0..9,
     S≠0, M ≠0,
     alldifferent([S,E,N,D,M,O,R,Y]),
                 1000*S + 100*E + 10*N + D
     +           1000*M + 100*O + 10*R + E
     = 10000*M + 1000*O + 100*N + 10*E + Y,
     labeling([S,E,N,D,M,O,R,Y]).
```

S=9, E in 4..7, N in 5..8, M=1, O=0, [D,R,Y] in 2..8

With Search: S=9, E=5, N=6, D=7, M=1, O=0, R=8, Y=2

Crypto-Arithmetic Problem

		S	E	N	D
+		M	O	R	E
=	M	O	N	E	Y

```
solve(S,E,N,D,M,O,R,Y) :-
      [S,E,N,D,M,O,R,Y] in 0..9,
     S≠0, M ≠0,
     alldifferent([S,E,N,D,M,O,R,Y]),
                 1000*S + 100*E + 10*N + D
     +           1000*M + 100*O + 10*R + E
     = 10000*M + 1000*O + 100*N + 10*E + Y,
     labeling([S,E,N,D,M,O,R,Y]).
```

S=9, E in 4..7, N in 5..8, M=1, O=0, [D,R,Y] in 2..8

With Search: S=9, E=5, N=6, D=7, M=1, O=0, R=8, Y=2

Crypto-Arithmetic Problem

		S	E	N	D
+		M	O	R	E
=	M	O	N	E	Y

```
solve(S,E,N,D,M,O,R,Y) :-
        [S,E,N,D,M,O,R,Y] in 0..9,
       S≠0, M ≠0,
       alldifferent([S,E,N,D,M,O,R,Y]),
                   1000*S + 100*E + 10*N + D
       +           1000*M + 100*O + 10*R + E
       = 10000*M + 1000*O + 100*N + 10*E + Y,
       labeling([S,E,N,D,M,O,R,Y]).
```

S=9, E in 4..7, N in 5..8, M=1, O=0, [D,R,Y] in 2..8

With Search: S=9, E=5, N=6, D=7, M=1, O=0, R=8, Y=2

Crypto-Arithmetic Problem

		S	E	N	D
+		M	O	R	E
=	M	O	N	E	Y

```
solve(S,E,N,D,M,O,R,Y) :-
        [S,E,N,D,M,O,R,Y] in 0..9,
       S≠0, M ≠0,
       alldifferent([S,E,N,D,M,O,R,Y]),
                   1000*S + 100*E + 10*N + D
       +           1000*M + 100*O + 10*R + E
       = 10000*M + 1000*O + 100*N + 10*E + Y,
       labeling([S,E,N,D,M,O,R,Y]).
```

S=9, E in 4..7, N in 5..8, M=1, O=0, [D,R,Y] in 2..8
With Search: S=9, E=5, N=6, D=7, M=1, O=0, R=8, Y=2

Crypto-Arithmetic Problem

		S	E	N	D
+		M	O	R	E
=	M	O	N	E	Y

```
solve(S,E,N,D,M,O,R,Y) :-
         [S,E,N,D,M,O,R,Y] in 0..9,
        S≠0, M ≠0,
        alldifferent([S,E,N,D,M,O,R,Y]),
                    1000*S + 100*E + 10*N + D
        +           1000*M + 100*O + 10*R + E
        = 10000*M + 1000*O + 100*N + 10*E + Y,
        labeling([S,E,N,D,M,O,R,Y]).
```

S=9, E in 4..7, N in 5..8, M=1, O=0, [D,R,Y] in 2..8
With Search: S=9, E=5, N=6, D=7, M=1, O=0, R=8, Y=2

Crypto-Arithmetic Problem

		S	E	N	D
+		M	O	R	E
=	M	O	N	E	Y

```
solve(S,E,N,D,M,O,R,Y) :-
         [S,E,N,D,M,O,R,Y] in 0..9,
        S≠0, M ≠0,
        alldifferent([S,E,N,D,M,O,R,Y]),
                    1000*S + 100*E + 10*N + D
        +           1000*M + 100*O + 10*R + E
        = 10000*M + 1000*O + 100*N + 10*E + Y,
        labeling([S,E,N,D,M,O,R,Y]).
```

S=9, E in 4..7, N in 5..8, M=1, O=0, [D,R,Y] in 2..8

With Search: S=9, E=5, N=6, D=7, M=1, O=0, R=8, Y=2

Crypto-Arithmetic Problem

		S	E	N	D
+		M	O	R	E
=	M	O	N	E	Y

```
solve(S,E,N,D,M,O,R,Y) :-
         [S,E,N,D,M,O,R,Y] in 0..9,
        S≠0, M ≠0,
        alldifferent([S,E,N,D,M,O,R,Y]),
                    1000*S + 100*E + 10*N + D
        +           1000*M + 100*O + 10*R + E
        = 10000*M + 1000*O + 100*N + 10*E + Y,
        labeling([S,E,N,D,M,O,R,Y]).
```

S=9, E in 4..7, N in 5..8, M=1, O=0, [D,R,Y] in 2..8

With Search: S=9, E=5, N=6, D=7, M=1, O=0, R=8, Y=2

Crypto-Arithmetic Problem

		S	E	N	D
+		M	O	R	E
=	M	O	N	E	Y

```
solve(S,E,N,D,M,O,R,Y) :-
         [S,E,N,D,M,O,R,Y] in 0..9,
        S≠0, M ≠0,
        alldifferent([S,E,N,D,M,O,R,Y]),
                    1000*S + 100*E + 10*N + D
        +           1000*M + 100*O + 10*R + E
        = 10000*M + 1000*O + 100*N + 10*E + Y,
        labeling([S,E,N,D,M,O,R,Y]).
```

S=9, E in 4..7, N in 5..8, M=1, O=0, [D,R,Y] in 2..8

With Search: S=9, E=5, N=6, D=7, M=1, O=0, R=8, Y=2

Crypto-Arithmetic Problem

		S	E	N	D
+		M	O	R	E
=	M	O	N	E	Y

```
solve(S,E,N,D,M,O,R,Y) :-
         [S,E,N,D,M,O,R,Y] in 0..9,
        S≠0, M ≠0,
        alldifferent([S,E,N,D,M,O,R,Y]),
                    1000*S + 100*E + 10*N + D
        +           1000*M + 100*O + 10*R + E
        = 10000*M + 1000*O + 100*N + 10*E + Y,
        labeling([S,E,N,D,M,O,R,Y]).
```

S=9, E in 4..7, N in 5..8, M=1, O=0, [D,R,Y] in 2..8

With Search: S=9, E=5, N=6, D=7, M=1, O=0, R=8, Y=2

Semantic Web

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

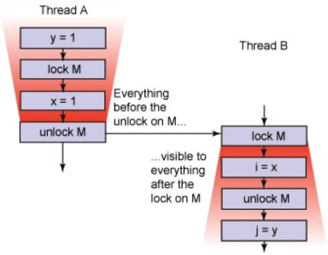
Semantic Web Reasoning for Ontology-Based Integration of Resources,
Liviu Badea, Doina Tilivea and Anca Hotaran, PPSWR 2004 LNCS 3208.

- 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.

Java Memory Machine

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



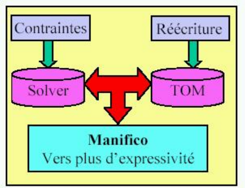
Conditional Read

```
Xr = (Cond)?Xw1:Xi

ite(true,Xr,Xw1,Xi)  <=> Xr = Xw1.
ite(false,Xr,Xw1,Xi) <=> Xr = Xi.
ite(Cond,Xr,X,X)     <=> Xr = X.
```

Business Rules for Optimization

MANIFICO - Francois Fages, Claude Kirchner, Hassan Ait-Kaci,...France

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).
```