Introduction to constraint programming

Per Kreuger piak@sics.se &
Markus Bohlin bola@sics.se

November 2002

SWEDISH INSTITUTE OF COMPUTER SCIENCE SICS

Introduction & Outline

- Constraints, Constraint satisfaction (CSP) & Constraint programming (CP)

- Outline
  - Introductory examples
  - Definitions & Terminology
  - A global constraint example
  - Historical Remarks

Introductory example

Find distinct digits for all letters, such that $S \neq 0, M \neq 0$ and the following equation holds.

$$\begin{array}{cccccc} & S & E & N & D \\ + & M & O & R & E \\ \hline M & O & N & E & Y \end{array}$$

SWEDISH INSTITUTE OF COMPUTER SCIENCE SICS

A CLP(FD)-program modelling this problem

Algorithm 1 Send More Money in CLP.

```
smm(L):-
    L = [S,E,N,D,M,O,R,Y],
    domain(L, 0, 9),
    S #> 0,
    M #> 0,
    all_different(L),
                 1000*S + 100*E + 10*N + D
    +            1000*M + 100*O + 10*R + E
    #= 10000*M + 1000*O + 100*N + 10*E + Y.
```

SWEDISH INSTITUTE OF COMPUTER SCIENCE SICS

Solving

```
| ?- smm([S,E,N,D,M,O,R,Y]).

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

yes

| ?- smm([S,E,N,D,M,O,R,Y]),
labeling([],[S,E,N,D,M,O,R,Y]).

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

yes
```

Constraint Propagation using *interval reasoning*

Constraint propagation is an *inference rule* for finite domain problems that reduces the domains of variables.

- Given the following constraints

$$X \in \{0, \ldots, 9\}$$
$$Y \in \{0, \ldots, 9\}$$

$$X + Y = 9 \quad (1)$$
$$2X + 4Y = 24 \quad (2)$$

- we can conclude:

$$\text{Eq. 2} \Rightarrow X \in \{0,...,8\}, Y \in \{2,...,6\}$$

$$\text{Eq. 1} \Rightarrow X \in \{3,...,7\}, Y \in \{2,...,6\}$$

$$\text{Eq. 2} \Rightarrow X \in \{4,...,6\}, Y \in \{3,4\}$$

$$\text{Eq. 1} \Rightarrow X \in \{5,6\}, Y \in \{3,4\}$$

$$\text{Eq. 2} \Rightarrow X \in \{6\}, Y \in \{3\}$$

Solving in a CLP system

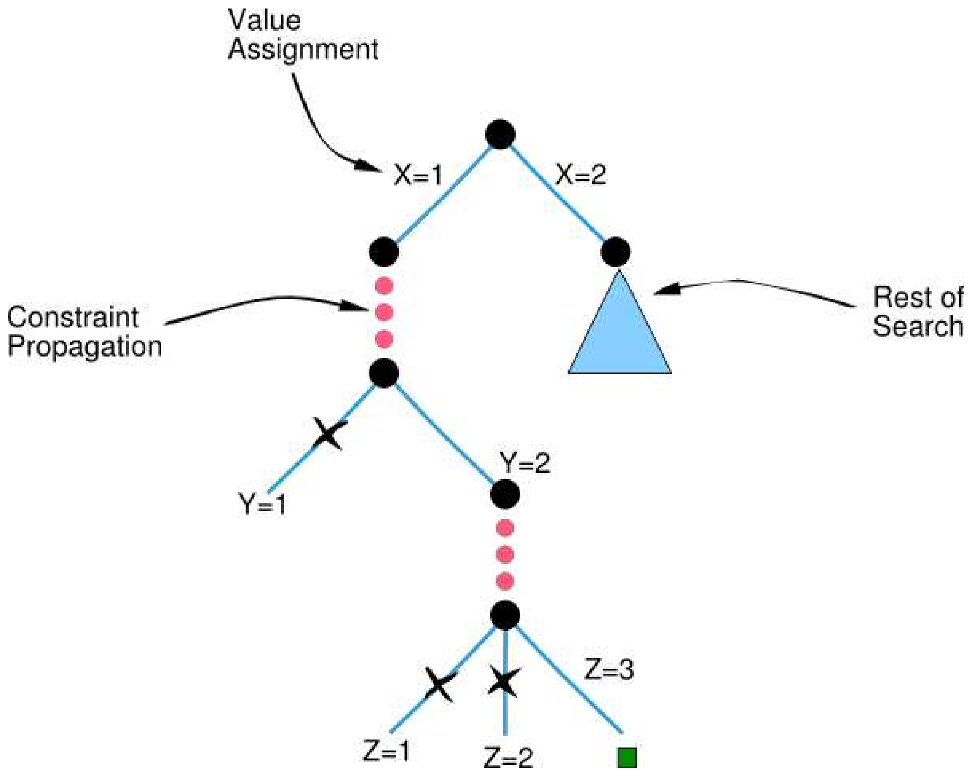
```
| ?- X in 0..9, Y in 0..9, X+Y #= 9, 2*X+4*Y #= 24.

X = 6,
Y = 3 ?

yes
```

Constrained search

Solving in CP interleaves propagation and speculative assignment (labeling)



Constraints

What are constraints?

Formally A relation between objects (problem variables)

Informally A statement of how the value of one or more variables restrict the possible values of others

E.g.

- Expressed as a relation in mathematical language

$$a, b \in \{1, 2, 3\}, a \neq b$$

- Expressed as tuples

$$\langle a,b \rangle \in \{\langle 1,2 \rangle, \langle 1,3 \rangle, \langle 2,1 \rangle, \langle 2,3 \rangle, \langle 3,1 \rangle, \langle 3,2 \rangle\}$$

Variables domains

- Discrete/Continuous
- Finite/Infinite

Constraint systems

A theory for a given variable domain and a class or type of constraints. E.g.:

H Syntactical equality between Herbrand (Prolog) terms

Domain: $a, f(a, g(b))$ etc.
Constraints: $f(X, a, U) = f(b, Y, V)$
Solutions: $X = b, Y = a, U = V$

B (In)Equality between boolean expressions

Domain: $\{0, 1\}$
Constraints: $X \vee Y = 1, X \wedge Y \leq X \vee Y$
Solutions: $(\exists U)\left(X = \overline{Y \wedge U}\right)$ or $\langle X, Y\rangle \in \{\langle 0, 1\rangle, \langle 1, 0\rangle, \langle 1, 1\rangle\}$

SWEDISH INSTITUTE OF COMPUTER SCIENCE SICS

Q Equality and inequality between linear expressions over rational numbers

Domain: $2, 15, \frac{3}{5}$ etc.
Constraints: $X \geq Y - 2, X \leq 6 - y, X = Y + \frac{2}{3}$
Solutions: $Y = X - 2, X \leq 4$

FD Relations between finite domain (usually integers)

Domain: $3, 8, 12$ etc.
Constraints: $X, Y, Z \in 1..4, X < Y, 2Y = Z$
Solutions: $X = 1, Y = 2, Z = 4$

SWEDISH INSTITUTE OF COMPUTER SCIENCE SICS

Properties and relations of constraints

Let Γ be a constraint theory, C a set of constraints and c a contraint.

Satisfiability $\Gamma \models (\exists) C$

Consistency $\Gamma \models (\exists)(C \wedge c)$

Tautology $\Gamma \models (\forall) C$

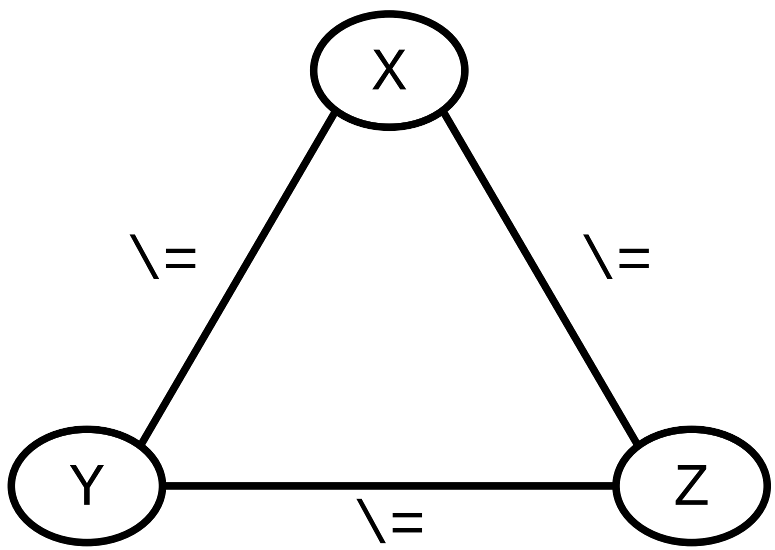
Entailment $\Gamma \models (\forall)(C \rightarrow c)$

Generalization Find the "tightest" C for a set of constraints $\{C_1, \ldots, C_n\}$ such that

$$\Gamma \models (\forall)((C_1 \rightarrow C) \wedge \cdots \wedge (C_n \rightarrow C))$$

Example 2 — A global constraint

The program given in the send-more-money example uses a global constraint called `all_different`. This constraint imposes pairwise distinctness between a set of variables.



Naive binarization is incomplete!

$$all\_different(X,Y,Z) \Leftrightarrow X \neq Y \wedge Y \neq Z \wedge X \neq Z$$

- However, if

  $$X,Y,Z \in \{1,2\}$$

  no *local* reasoning (using information only about variable domains and a single binary constraint) can conclude that this set of constraints is inconsistent

- I.e. more "global" reasoning is necessary for completeness in this case; for `all_different` complete algorithms are known that works in quadratic time

- Global reasoning can in general provide strong propagation even if complete algorithms are not known or are too computationally demanding for practical use

Applications

- Design
- Diagnosis
- Planning
- Scheduling & Timetabling
- Packing & Placement
- Logistics
- Validation

Historical remarks

- Early use of constraints in graphics software SKETCHPAD c.a. 1963
- CSP first systematically studied in AI (vision) 1971-1975
- First steps in CLP by Jaffar & Lassez (CLP(R))c.a 1987
- Finite domain CSP studied by Mackworth c.a. 1992
- First practical implementations by Colmerauer et. al. (Prolog-III) reported c.a. 1990
- Pioneering effort by team at ECRC on the CHIP system set stage for large scale applications from 1988 and onward

- CHIP commercialized by COSYTEC 1990
- ILOG SOLVER introduced 1994
- First global constraint (all-different) c.a. 1994
- *Global constraints* in CHIP c.a. 1993 — New level of
- ILOG SOLVER commercialized by ILOG1987 — Very successful
- CLP(FD) introduced in SICStus Prolog 1997 — State of the art system

SWEDISH INSTITUTE OF COMPUTER SCIENCE SICS