Constraint Programming

Master Parisien de Recherche en Informatique

2008/2009

François Fages et Sylvain Soliman
Project team Contraintes
INRIA Paris-Rocquencourt

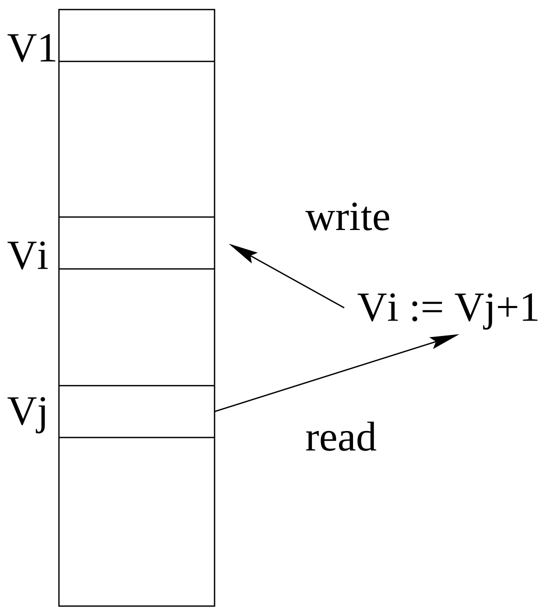
`http://contraintes.inria.fr/ fages`

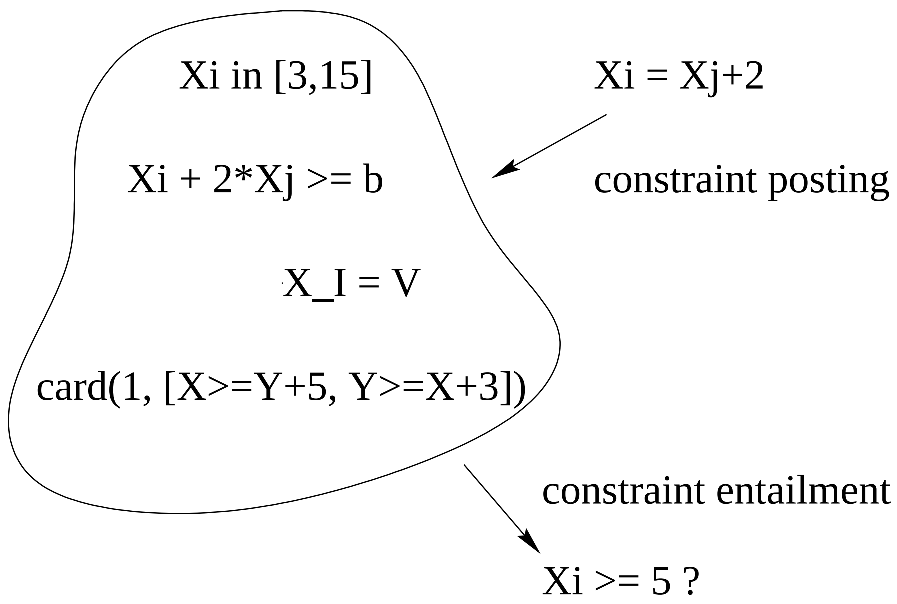
`Francois.Fages@inria.fr`

Constraint Programming Machine

Computation on partial information structures.

Von Neuman machine	Constraints machine
memory of values	memory of constraints
assignment variables	mathematical variables





The Paradigm of Constraint Logic Programming

Program = Logical Formula

Axiomatization:

"Domain of discourse" $\mathcal{X}$,

Model of the problem P

Execution = Proof search

Constraint satisfiability,

Logical resolution principle

Class of languages CLP($\mathcal{X}$) parametrized by $\mathcal{X}$:

- Primitive Constraints over $\mathcal{X}$

$$U = R * I$$

- Relations defined by logical formulae

$$\forall x, y\ path(x, y) \Leftrightarrow edge(x, y) \vee \exists z(edge(x, z) \wedge path(z, y))$$

Languages for defining new relations

- First-order logic predicate calculus

$$\forall x, y \; path(x,y) \Leftrightarrow edge(x,y) \vee \exists z (edge(x,z) \wedge path(z,y))$$

- Prolog/CLP($\mathcal{X}$) clauses

```
path(X,Y):- edge(X,Y).
path(X,Y):- edge(X,Z), path(Z,Y).
```

- Concurrent constraint process languages CC($\mathcal{X}$)

  Process $A = c \; | p(x) \mid (A \parallel A) \mid A + A \mid ask(c) \rightarrow A \mid \exists x A$

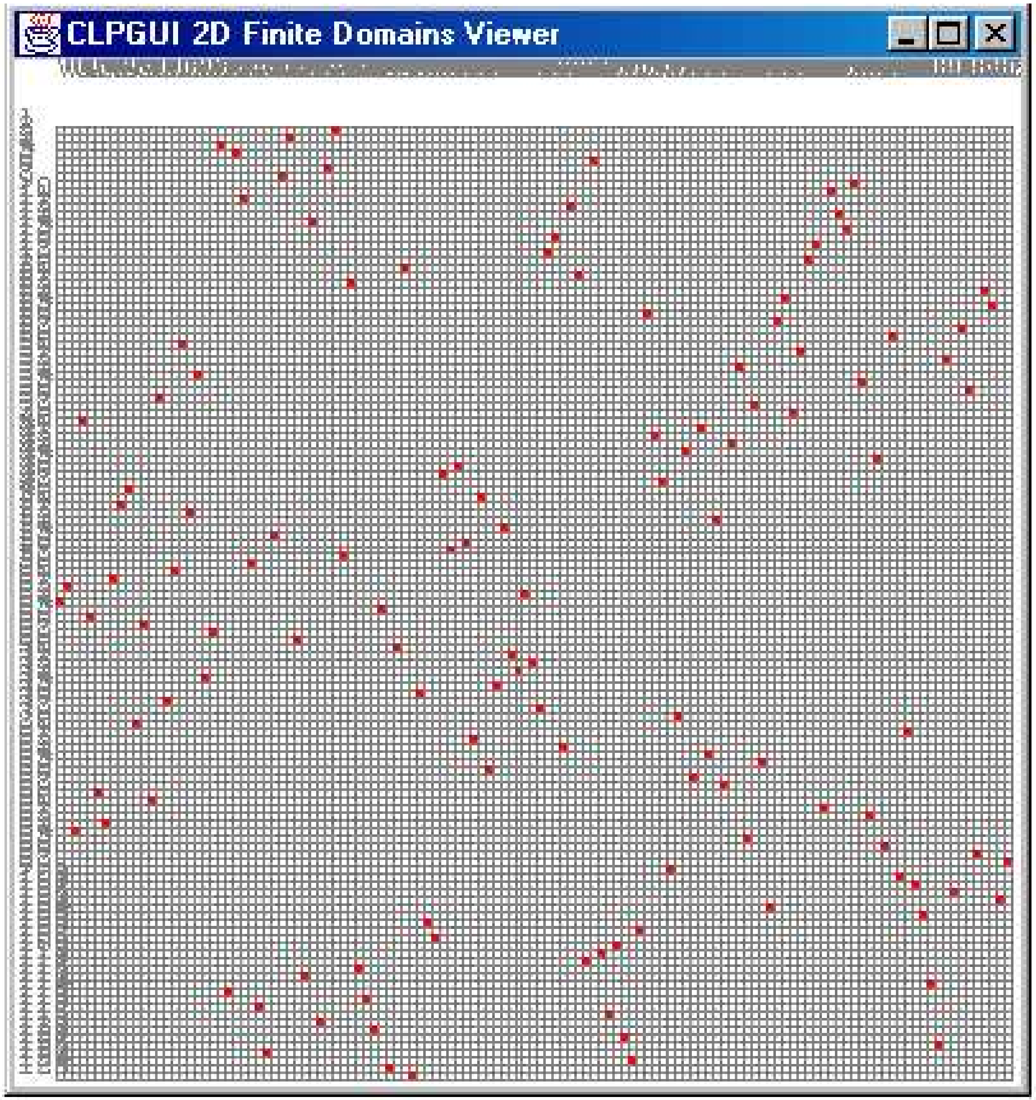
$$path(X,Y) :: edge(X,Y) + \exists Z (edge(X,Z) || path(Z,Y))$$

- Constraint libraries in object-oriented/functional/imperative languages.

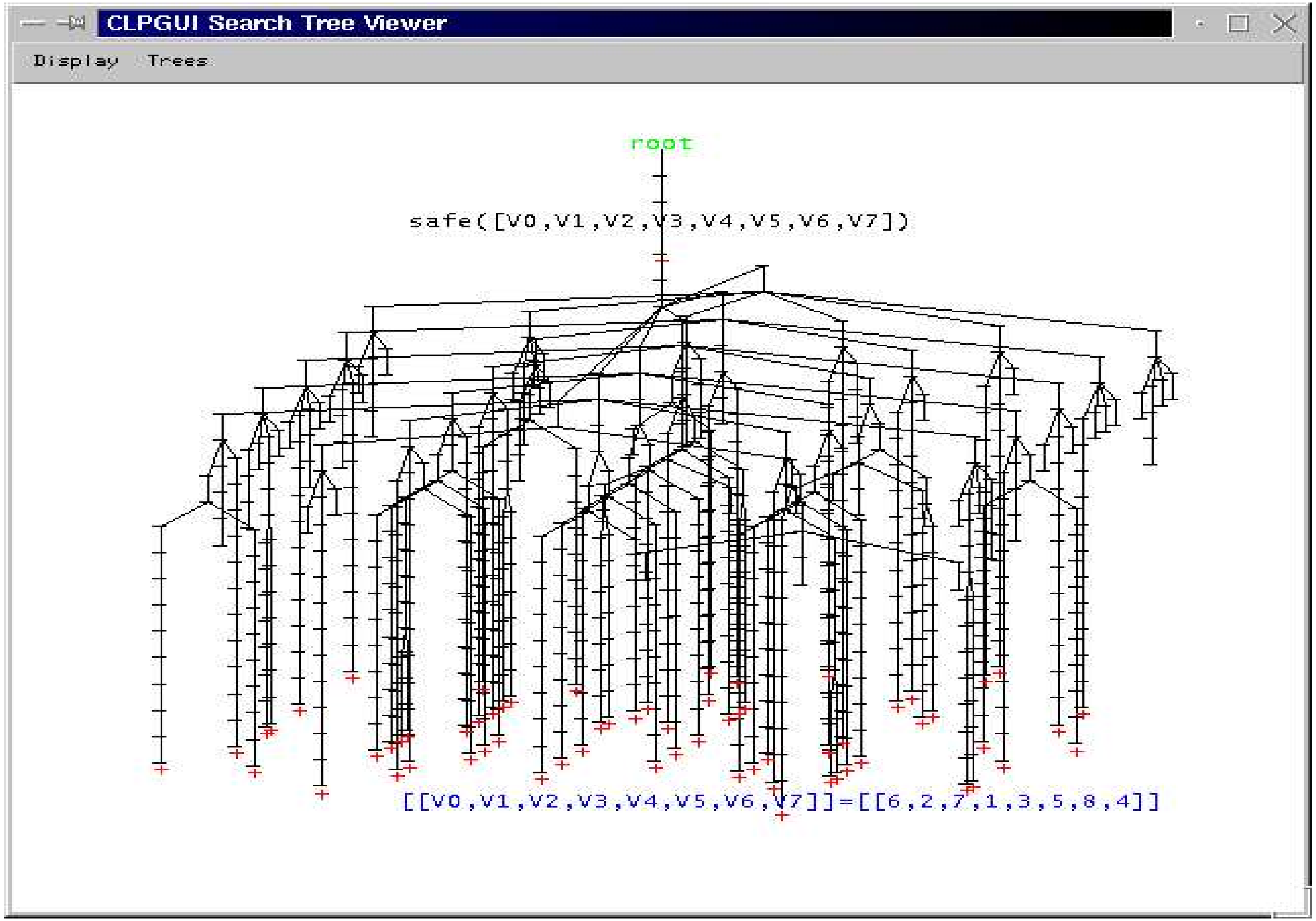
CLP(FD) N-queens Problem

GNU-Prolog program:

```
queens(N,L):-
    length(L,N),
    fd_domain(L,1,N),
    safe(L),
    fd_labeling(L,first_fail).
safe([]).
safe([X|L]):-
    noattack(L,X,1),
    safe(L).
noattack([],_,_).
noattack([Y|L],X,I):-
    X#\=Y,
    X#\=Y+I,
    X+I#\=Y,
    I1 is I+1,
    noattack(L,X,I1).
```



Search space of all solutions



Successes in combinatorial search problems

Job shop scheduling, resource allocation, graph coloring,...

- Decision Problems: existence of a solution (of given cost)

  in P: if algorithm of polynomial time complexity

  in NP: if *non-deterministic* algorithm of polynomial complexity.

  NP-complete if polynomial encoding of any other NP problem

- Optimisation Problems: computation of a solution of optimal cost

  NP-hard if the decision problem is NP-complete

- The size of the search space does not tell the complexity of the problem

  Sorting n elements in $O(n \log n)$, search space in $!n$...

  SAT over n Boolean in $O(2^n)$, search space in 2^n.

Hot Research Topics in Constraint Programming

- Combinatorial optimization problems

  Ressource allocation, scheduling, frequency allocation, geometrical placement problems, ...

  Benchmarks (open shop 6x6, graph colouring, time tabling,...)

  New applications in Bioinformatics, Web, multimedia,...

- Algorithms

  Global constraints, hybridization of algorithms CP, MILP, local search

  Symmetry detection and breaking

  Search procedures, randomization, adaptive solving strategies

- Languages

  Modelling languages

  Extensible CP languages

  Automatic synthesis of constraint solvers

Panorama of CP Systems

- Prolog-based systems

  open source: *CLP(H,FD)* GNU-Prolog (Univ. Paris 5, INRIA)

  CC(H,FD) SWI-Prolog (CWI), Cia-Prolog (UPM) ,...

  LCC(H,FD) SiLCC (INRIA)

  commercial: *CC(H,FD,Q,R)* Sicstus-Prolog (SICS), Cosytec CHIP,...

- Java-based systems

  open source: Choco (EMN), CHR JCK toolkit (Univ. Ulm)

  commercial: JSolver, JConfigurator (ILOG), Koalog, ...

- C++ based systems

  Ilog-Solver, Ilog-Scheduler, Cosytec Chip++

- CAML-based system

  FaCiLe (Univ. Toulouse)

Workplan of the Lecture (1/2)

1. Introduction
   - Constraint Programming
   - Logical background
2. Constraint Solving
   - Decidability of constraint languages and complete theories
   - Constraint solving algorithms on $\mathcal{H}$ and $\mathcal{R}$ by rewriting
   - Contraint propagation algorithms on $\mathcal{FD}$
   - Symmetry breaking
3. Constraint Logic Programs
   - Operational semantics and examples $CLP(\mathcal{FD})$
   - Fixed semantics and abstract interpretation
   - Logical semantics, theorem proving and negation

Workplan of the Lecture (2/2)

4 Concurrent Constraint Languages

- Operational semantics and examples $CC(\mathcal{FD})$
- Denotational semantics of determinate and non-determinate CC programs
- Logical semantics in Linear Logic
- Phase model checking
- LL constraint systems and imperative features

–– 17/11/2009 Small constraint programming project in GNU-Prolog

–– 09/02/2010 Examination

–– 16/02/2009 Start internships

Lecture notes

- `http://contraintes.inria.fr/~fages/Teaching`
  - Article “Des contraintes au programme”. Revue littraire Actes de Savoirs, PUF. 2007.
  - Chapter “Programmation logique et contraintes” Encyclopdie Vuibert d’Informatique.
  - Slides of lectures 1-4
  - Lecture notes on Constraint Logic Programming
- `http://contraintes.inria.fr/~soliman`
  - Slides of lectures 5-16
- Krzysztof Apt, “Principles of Constraint Programming”, Cambridge University Press, 2003.