

# Constraint Logic Programming

Sylvain Soliman

Sylvain.Soliman@inria.fr



Project-Team CONTRAINTES

MPRI 2.35.1 Course – December 2011 - February 2012

# Part I: CLP - Introduction and Logical Background

- 1 The Constraint Programming paradigm
- 2 Examples and Applications
- 3 First Order Logic
- 4 Models
- 5 Logical Theories

# Part II: Constraint Logic Programs

6 Constraint Languages

7  $\text{CLP}(\mathcal{X})$

8  $\text{CLP}(\mathcal{H})$

9  $\text{CLP}(\mathcal{R}, \mathcal{FD}, \mathcal{B})$

# Part III: CLP - Operational and Fixpoint Semantics

10 Operational Semantics

11 Fixpoint Semantics

12 Program Analysis

# Part IV: Logical Semantics

13 Logical Semantics of CLP( $\mathcal{X}$ )

14 Automated Deduction

15 CLP( $\lambda$ )

16 Negation as Failure

# Soundness of CSLD Resolution

## Theorem 1 ([JL87popl])

*If  $c$  is a computed answer for the goal  $G$  then  $M_P^{\mathcal{X}} \models c \supset G$ ,  
 $P \models_{\mathcal{X}} c \supset G$  and  $P, \mathcal{T} \models c \supset G$ .*

If  $G = (d | A_1, \dots, A_n)$ , we deduce from the  $\wedge$ -compositionality lemma, that there exist computed answers  $c_1, \dots, c_n$  for the goals  $A_1, \dots, A_n$  such that  $c = d \wedge \bigwedge_{i=1}^n c_i$  is satisfiable. For every  $1 \leq i \leq n$

$c_i | A_i \in S_P^{\mathcal{X}} \uparrow \omega$ ,

$[c_i | A_i]_{\mathcal{X}} \subset M_P^{\mathcal{X}}$ , hence  $M_P^{\mathcal{X}} \models \forall (c_i \supset A_i)$ ,

$P \models_{\mathcal{X}} \forall (c_i \supset A_i)$  as  $M_P^{\mathcal{X}}$  is the least  $\mathcal{X}$ -model of  $P$ ,

$P \models_{\mathcal{X}} \forall (c \supset A_i)$  as  $\mathcal{X} \models \forall (c \supset c_i)$  for all  $i$ ,  $1 \leq i \leq n$ .

Therefore we have  $P \models_{\mathcal{X}} \forall (c \supset (d \wedge A_1 \wedge \dots \wedge A_n))$ ,

and as the same reasoning applies to any model  $\mathcal{X}$  of  $\mathcal{T}$ ,

$P, \mathcal{T} \models \forall (c \supset (d \wedge A_1 \wedge \dots \wedge A_n))$

# Completeness of CSLD resolution

## Theorem 2 ([Maher87iclp])

If  $M_\rho^{\mathcal{X}} \models c \supset G$  then there exists a set  $\{c_i\}_{i \geq 0}$  of computed answers for  $G$ , such that:  $\mathcal{X} \models \forall(c \supset \bigvee_{i \geq 0} \exists Y_i c_i)$ .

## Proof.

For every solution  $\rho$  of  $c$ , for every atom  $A_j$  in  $G$ ,

$M_\rho^{\mathcal{X}} \models A_j \rho$  iff  $A_j \rho \in T_\rho^{\mathcal{X}} \uparrow \omega$ , iff  $A_j \rho \in [S_\rho^{\mathcal{X}} \uparrow \omega]_{\mathcal{X}}$

iff  $c_{j,\rho} | A_j \in S_\rho^{\mathcal{X}} \uparrow \omega$ , for some constraint  $c_{j,\rho}$  s.t.  $\rho$  is solution of  $\exists Y_{j,\rho} c_{j,\rho}$ ,  
where  $Y_{j,\rho} = V(c_{j,\rho}) \setminus V(A_j)$ ,

iff  $c_{j,\rho}$  is a computed answer for  $A_j$  and  $\mathcal{X} \models \exists Y_{j,\rho} c_{j,\rho}$ .

Let  $c_\rho$  be the conjunction of  $c_{j,\rho}$  for all  $j$ .  $c_\rho$  is a computed answer for  $G$ .

By taking the collection of  $c_\rho$  for all  $\rho$  we get  $\mathcal{X} \models \forall(c \supset \bigvee_{c_\rho} \exists Y_\rho c_\rho)$   $\square$

# Completeness w.r.t. the theory of the structure

## Theorem 3 ([Maher87iclp])

If  $P, \mathcal{T} \models c \supset G$  then there exists a finite set  $\{c_1, \dots, c_n\}$  of computed answers to  $G$ , such that:

$$\mathcal{T} \models \forall (c \supset \exists Y_1 c_1 \vee \dots \vee \exists Y_n c_n).$$

### Proof.

If  $P, \mathcal{T} \models c \supset G$  then for every model  $\mathcal{X}$  of  $\mathcal{T}$ , for every  $\mathcal{X}$ -solution  $\rho$  of  $c$ , there exists a computed constraint  $c_{\mathcal{X}, \rho}$  for  $G$  s.t.  $\mathcal{X} \models c_{\mathcal{X}, \rho}$ . Let  $\{c_i\}_{i \geq 1}$  be the set of these computed answers. Then for every model  $\mathcal{X}$  and for every  $\mathcal{X}$ -valuation  $\rho$ ,  $\mathcal{X} \models c \supset \bigvee_{i \geq 1} \exists Y_i c_i$ ,

therefore  $\mathcal{T} \models c \supset \bigvee_{i \geq 1} \exists Y_i c_i$ ,

As  $\mathcal{T} \cup \{\exists (c \wedge \neg \exists Y_i c_i)\}_i$  is unsatisfiable, by applying the compactness theorem of first-order logic there exists a finite part  $\{c_i\}_{1 \leq i \leq n}$ ,

s.t.  $\mathcal{T} \models c \supset \bigvee_{i=1}^n \exists Y_i c_i$ . □

# Soundness of Negation as Finite Failure

## Theorem 4

If  $G$  is finitely failed then  $P^*, \mathcal{T} \models \neg G$ .

## Proof.

By induction on the height  $h$  of the tree in finite failure for  $G = c|A, \alpha$  where  $A$  is the selected atom at the root of the tree.

In the base case  $h = 1$ , the constrained atom  $c|A$  has no CSLD transition, we can deduce that  $P^*, \mathcal{T} \models \neg(c \wedge A)$  hence that  $P^*, \mathcal{T} \models \neg G$ .

For the induction step, let us suppose  $h > 1$ . Let  $G_1, \dots, G_n$  be the sons of the root and  $Y_1, \dots, Y_n$  be the respective sets of introduced variables. We have  $P^*, \mathcal{T} \models G \leftrightarrow \exists Y_1 G_1 \vee \dots \vee \exists Y_n G_n$ . By induction hypothesis,  $P^*, \mathcal{T} \models \neg G_i$  for every  $1 \leq i \leq n$ , therefore  $P^*, \mathcal{T} \models \neg G$ .  $\square$

# Completeness of Negation as Failure

## Theorem 5 ([JL87popl])

*If  $P^*, \mathcal{T} \models \neg G$  then  $G$  is finitely failed.*

We show that if  $G$  is not finitely failed then  $P^*, \mathcal{T}, \exists(G)$  is satisfiable. If  $G$  has a success then by the soundness of CSLD resolution 1,  $P^*, \mathcal{T} \models \exists G$ . Else  $G$  has a fair infinite derivation  $G = c_0 | G_0 \rightarrow c_1 | G_1 \rightarrow \dots$

For every  $i \geq 0$ ,  $c_i$  is  $\mathcal{T}$ -satisfiable, thus by the **compactness theorem**,  $c_\omega = \bigwedge_{i \geq 0} c_i$  is  $\mathcal{T}$ -satisfiable. Let  $\mathcal{X}$  be a model of  $\mathcal{T}$  s.t.  $\mathcal{X} \models \exists(c_\omega)$ . Let  $I_0 = \{A\rho \mid A \in G_i \text{ for some } i \geq 0 \text{ and } \mathcal{X} \models c_\omega\rho\}$ . As the derivation is fair, every atom  $A$  in  $I_0$  is selected, thus  $c_\omega | A \rightarrow c_\omega | A_1, \dots, A_n$  with  $[c_\omega | A] \cup \dots \cup [c_\omega | A_n] \subset I_0$ . We deduce that  $I_0 \subset T_P^\mathcal{X}(I_0)$ . By Knaster-Tarski's theorem, the iterated application up to ordinal  $\omega$  of the operator  $T_P^\mathcal{X}$  from  $I_0$  leads to a fixed point  $I$  s.t.  $I_0 \subset I$ , thus  $[c_\omega | G_0] \subset I$ . Hence  $P^*, \exists(G)$  is  $\mathcal{X}$ -satisfiable, and  $P^*, \mathcal{T}, \exists(G)$  is satisfiable.

# Part V: Constraint Solving

17 Solving by Rewriting

18 Solving by Domain Reduction

## Part VI

# Practical CLP Programming

## Part VI: Practical CLP Programming

- 19 CLP implementation, the WAM
- 20 Optimizing CLP
- 21 Symmetries
- 22 Symmetry Breaking During Search
- 23 Detecting Symmetries

# The Warren Abstract Machine

First Prolog implementation in the early 70's (by Colmerauer et al.).

In 1983, David H. Warren creates the [Warren Abstract Machine](#).

Remains the state of the art (for term representation, basic instructions, ...)

Slightly extended for CLP

(C)SLD resolution seen as a call stack (with marks for choice points)

# The Warren Abstract Machine

First Prolog implementation in the early 70's (by Colmerauer et al.).

In 1983, David H. Warren creates the [Warren Abstract Machine](#).

Remains the state of the art (for term representation, basic instructions, ...)

Slightly extended for CLP ([constraints instead of substitutions](#))

(C)SLD resolution seen as a call stack (with marks for choice points)

## Optimizations from the WAM

Search for predicates should be almost in constant time

Use a hash table - **indexing** - for the predicate name/arity,

Each call normally adds a frame to the call stack (removed on backtracking)

As for other programming paradigms, not always necessary

## Optimizations from the WAM

Search for predicates should be almost in constant time

Use a hash table - **indexing** - for the predicate name/arity, and the functor of the first argument

Each call normally adds a frame to the call stack (removed on backtracking)

As for other programming paradigms, not always necessary

## Optimizations from the WAM

Search for predicates should be almost in constant time

Use a hash table - **indexing** - for the predicate name/arity, and the functor of the first argument

Each call normally adds a frame to the call stack (removed on backtracking)

As for other programming paradigms, not always necessary

**Tail recursion** can be optimized,

## Optimizations from the WAM

Search for predicates should be almost in constant time

Use a hash table - **indexing** - for the predicate name/arity, and the functor of the first argument

Each call normally adds a frame to the call stack (removed on backtracking)

As for other programming paradigms, not always necessary

**Tail recursion** can be optimized, when calling and called contexts are **deterministic**.

# Putting it all together

## Naive sum

```
sum(0, []).  
sum(S, [H | T]) :-  
    sum(S1, T),  
    S is S1 + H.
```

# Putting it all together

## Naive sum

```
sum(0, []).  
sum(S, [H | T]) :-  
    sum(S1, T),  
    S is S1 + H.
```

## Much better

```
sum(L, S) :-  
    sum_aux(L, 0, S).  
  
sum_aux([], S, S).  
sum_aux([H | T], S0, S) :-  
    S1 is S0 + H,  
    sum_aux(T, S1, S).
```

## Putting it all together

If numbers are coded as the `fact` `number(X)`?

```
sum(S) :- findall(X, number(X), L), sum(L, S).
```

## Putting it all together

If numbers are coded as the fact `number(X)`?

```
sum(S) :- findall(X, number(X), L), sum(L, S).
```

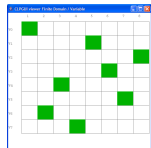
```
sum(S) :-  
    g_assign(sum, 0),  
    (  
        number(N),  
        g_read(sum, S1),  
        S2 is S1 + N,  
        g_assign(sum, S2),  
        fail  
    );  
    g_read(sum, S)  
).
```

# Cutting choice-points

```
try(S) :-
    stream_property(S,
                    input),
    (
        repeat,
        read_term(S, G),
        call(G),
        ground(G),
        !,
        write(G)
    ).
try(S) :-
    ...
```

```
try(S) :-
    stream_property(S,
                    input),
    (
        repeat,
        read_term(S, G),
        call(G),
        ground(G)
        ->
        write(G)
    ).
try(S) :-
    ...
```

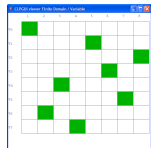
# Symmetries in the N-queens problem



$\text{queens}(N, [X_1, \dots, X_N])$

iff

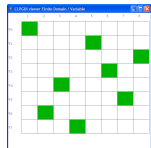
# Symmetries in the N-queens problem



$\text{queens}(N, [X_1, \dots, X_N])$

iff  $\text{queens}(N, [X_N, \dots, X_1])$  *vertical axis symmetry*

# Symmetries in the N-queens problem

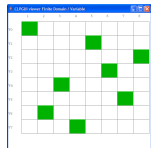


$\text{queens}(N, [X_1, \dots, X_N])$

iff  $\text{queens}(N, [X_N, \dots, X_1])$  *vertical axis symmetry*

*variable symmetry*

# Symmetries in the N-queens problem



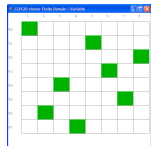
$\text{queens}(N, [X_1, \dots, X_N])$

iff  $\text{queens}(N, [X_N, \dots, X_1])$  *vertical axis symmetry*  
*variable symmetry*

iff  $\text{queens}(N, [N+1-X_1, \dots, N+1-X_N])$  *horizontal axis symmetry*



# Symmetries in the N-queens problem



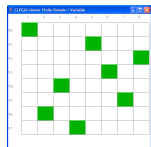
$\text{queens}(N, [X_1, \dots, X_N])$

iff  $\text{queens}(N, [X_N, \dots, X_1])$  *vertical axis symmetry*  
variable symmetry

iff  $\text{queens}(N, [N+1-X_1, \dots, N+1-X_N])$  *horizontal axis symmetry*  
value symmetry

iff  $\text{queens}(N, [Y_1, \dots, Y_N])$  where  $X_i = j$  iff  $Y_j = N+1-i$  *rotation symmetry*  
variable-value symmetry

# Symmetries in the N-queens problem



$\text{queens}(N, [X_1, \dots, X_N])$

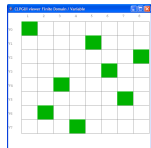
iff  $\text{queens}(N, [X_N, \dots, X_1])$  *vertical axis symmetry*  
*variable symmetry*

iff  $\text{queens}(N, [N+1-X_1, \dots, N+1-X_N])$  *horizontal axis symmetry*  
*value symmetry*

iff  $\text{queens}(N, [Y_1, \dots, Y_N])$  where  $X_i = j$  iff  $Y_j = N+1-i$  *rotation symmetry*  
*variable-value symmetry*

iff  $\text{queens}(N, [Y_1, \dots, Y_N])$  where  $X_i = j$  iff  $Y_j = i$  *rotation symmetry*  
*variable-value symmetry*

# Symmetries in the N-queens problem



$\text{queens}(N, [X_1, \dots, X_N])$

iff  $\text{queens}(N, [X_N, \dots, X_1])$  *vertical axis symmetry*

*variable symmetry* broken by  $X_1 < X_N$

iff  $\text{queens}(N, [N+1-X_1, \dots, N+1-X_N])$  *horizontal axis symmetry*

*value symmetry*

iff  $\text{queens}(N, [Y_1, \dots, Y_N])$  where  $X_i = j$  iff  $Y_j = N+1-i$  *rotation*

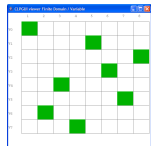
*symmetry*

*variable-value symmetry*

iff  $\text{queens}(N, [Y_1, \dots, Y_N])$  where  $X_i = j$  iff  $Y_j = i$  *rotation symmetry*

*variable-value symmetry*

# Symmetries in the N-queens problem



$\text{queens}(N, [X_1, \dots, X_N])$

iff  $\text{queens}(N, [X_N, \dots, X_1])$  *vertical axis symmetry*

*variable symmetry broken by  $X_1 < X_N$*

iff  $\text{queens}(N, [N+1-X_1, \dots, N+1-X_N])$  *horizontal axis symmetry*

*value symmetry broken by  $X_1 < 5$*

iff  $\text{queens}(N, [Y_1, \dots, Y_N])$  where  $X_i = j$  iff  $Y_j = N+1-i$  *rotation symmetry*

*variable-value symmetry*

iff  $\text{queens}(N, [Y_1, \dots, Y_N])$  where  $X_i = j$  iff  $Y_j = i$  *rotation symmetry*

*variable-value symmetry*

## Variable Symmetries

Given a Constraint Satisfaction Problem  $c(x_1, \dots, x_n)$  over  $\mathcal{X}$  a **variable symmetry**  $\sigma$  is a bijection on variables that preserves solutions:

$$\mathcal{X} \models c(x_1, \dots, x_n) \text{ iff } \mathcal{X} \models c(x_{\sigma(1)}, \dots, x_{\sigma(n)})$$

### Proposition 6 ([Crawford96kr])

*If  $(\mathcal{X}, \leq)$  is an order, all variable symmetries can be broken by the global constraint*

$$\bigwedge_{\sigma \in \Sigma} [x_1, \dots, x_n] \leq_{lex} [x_{\sigma(1)}, \dots, x_{\sigma(n)}]$$

### Proof.

This is one way to choose a unique member in each equivalence class of symmetric assignments. □

# Breaking Several Variable Symmetries

## Proposition 7 ([Puget05cp,Walsh06cp])

$AC(\bigwedge_{\sigma \in \Sigma} [x_1, \dots, x_n] \leq_{lex} [x_{\sigma(1)}, \dots, x_{\sigma(n)}])$  is strictly stronger than  $\bigwedge_{\sigma \in \Sigma} AC([x_1, \dots, x_n] \leq_{lex} [x_{\sigma(1)}, \dots, x_{\sigma(n)}])$ .

## Proof.

Let  $x_1, x_2, x_4 \in \{0, 1\}$  and  $x_3 = 1$ . Consider two symmetries (1243) and (1423), we have  $AC([x_1, x_2, x_3, x_4] \leq_{lex} [x_2, x_4, x_1, x_3])$  and  $AC([x_1, x_2, x_3, x_4] \leq_{lex} [x_4, x_3, x_1, x_2])$ .

cases  $x_1 = 0$   $[x_1 \ x_2 \ x_3 \ x_4] \leq_{lex} [x_2 \ x_4 \ x_1 \ x_3]$   $[x_1 \ x_2 \ x_3 \ x_4] \leq_{lex} [x_4 \ x_3 \ x_1 \ x_2]$



# Breaking Several Variable Symmetries

## Proposition 7 ([Puget05cp,Walsh06cp])

$AC(\bigwedge_{\sigma \in \Sigma} [x_1, \dots, x_n] \leq_{lex} [x_{\sigma(1)}, \dots, x_{\sigma(n)}])$  is strictly stronger than  $\bigwedge_{\sigma \in \Sigma} AC([x_1, \dots, x_n] \leq_{lex} [x_{\sigma(1)}, \dots, x_{\sigma(n)}])$ .

## Proof.

Let  $x_1, x_2, x_4 \in \{0, 1\}$  and  $x_3 = 1$ . Consider two symmetries (1243) and (1423), we have  $AC([x_1, x_2, x_3, x_4] \leq_{lex} [x_2, x_4, x_1, x_3])$  and  $AC([x_1, x_2, x_3, x_4] \leq_{lex} [x_4, x_3, x_1, x_2])$ .

cases  $[x_1 \ x_2 \ x_3 \ x_4] \leq_{lex} [x_2 \ x_4 \ x_1 \ x_3] \quad [x_1 \ x_2 \ x_3 \ x_4] \leq_{lex} [x_4 \ x_3 \ x_1 \ x_2]$   
 $x_1 = 0 \quad \begin{matrix} x_1 & x_2 & x_3 & x_4 \\ 0 & 0 & 1 & 1 \end{matrix} \leq_{lex} \begin{matrix} x_2 & x_4 & x_1 & x_3 \\ 0 & 1 & 0 & 1 \end{matrix} \quad \begin{matrix} x_1 & x_2 & x_3 & x_4 \\ 0 & 0 & 1 & 1 \end{matrix} \leq_{lex} \begin{matrix} x_4 & x_3 & x_1 & x_2 \\ 1 & 1 & 0 & 0 \end{matrix}$   
 $x_1 = 1$

# Breaking Several Variable Symmetries

## Proposition 7 ([Puget05cp,Walsh06cp])

$AC(\bigwedge_{\sigma \in \Sigma} [x_1, \dots, x_n] \leq_{lex} [x_{\sigma(1)}, \dots, x_{\sigma(n)}])$  is strictly stronger than  $\bigwedge_{\sigma \in \Sigma} AC([x_1, \dots, x_n] \leq_{lex} [x_{\sigma(1)}, \dots, x_{\sigma(n)}])$ .

## Proof.

Let  $x_1, x_2, x_4 \in \{0, 1\}$  and  $x_3 = 1$ . Consider two symmetries (1243) and (1423), we have  $AC([x_1, x_2, x_3, x_4] \leq_{lex} [x_2, x_4, x_1, x_3])$  and  $AC([x_1, x_2, x_3, x_4] \leq_{lex} [x_4, x_3, x_1, x_2])$ .

cases	$[x_1$	$x_2$	$x_3$	$x_4]$	$\leq_{lex}$	$[x_2$	$x_4$	$x_1$	$x_3]$	$[x_1$	$x_2$	$x_3$	$x_4]$	$\leq_{lex}$	$[x_4$	$x_3$	$x_1$	$x_2]$
$x_1 = 0$	0	0				0	1			0	0				0	1		
$x_1 = 1$	1	1	1	1		1	1	1	1									

# Breaking Several Variable Symmetries

## Proposition 7 ([Puget05cp,Walsh06cp])

$AC(\bigwedge_{\sigma \in \Sigma} [x_1, \dots, x_n] \leq_{lex} [x_{\sigma(1)}, \dots, x_{\sigma(n)}])$  is strictly stronger than  $\bigwedge_{\sigma \in \Sigma} AC([x_1, \dots, x_n] \leq_{lex} [x_{\sigma(1)}, \dots, x_{\sigma(n)}])$ .

## Proof.

Let  $x_1, x_2, x_4 \in \{0, 1\}$  and  $x_3 = 1$ . Consider two symmetries (1243) and (1423), we have  $AC([x_1, x_2, x_3, x_4] \leq_{lex} [x_2, x_4, x_1, x_3])$  and  $AC([x_1, x_2, x_3, x_4] \leq_{lex} [x_4, x_3, x_1, x_2])$ .

cases	$[x_1$	$x_2$	$x_3$	$x_4]$	$\leq_{lex}$	$[x_2$	$x_4$	$x_1$	$x_3]$	$[x_1$	$x_2$	$x_3$	$x_4]$	$\leq_{lex}$	$[x_4$	$x_3$	$x_1$	$x_2]$
$x_1 = 0$	0	0				0	1			0	0				0	1		
$x_1 = 1$	1	1	1	1		1	1	1	1	1	0				1	1		
$x_2 = 0$																		

# Breaking Several Variable Symmetries

## Proposition 7 ([Puget05cp,Walsh06cp])

$AC(\bigwedge_{\sigma \in \Sigma} [x_1, \dots, x_n] \leq_{lex} [x_{\sigma(1)}, \dots, x_{\sigma(n)}])$  is strictly stronger than  $\bigwedge_{\sigma \in \Sigma} AC([x_1, \dots, x_n] \leq_{lex} [x_{\sigma(1)}, \dots, x_{\sigma(n)}])$ .

## Proof.

Let  $x_1, x_2, x_4 \in \{0, 1\}$  and  $x_3 = 1$ . Consider two symmetries (1243) and (1423), we have  $AC([x_1, x_2, x_3, x_4] \leq_{lex} [x_2, x_4, x_1, x_3])$  and  $AC([x_1, x_2, x_3, x_4] \leq_{lex} [x_4, x_3, x_1, x_2])$ .

cases	$[x_1$	$x_2$	$x_3$	$x_4]$	$\leq_{lex}$	$[x_2$	$x_4$	$x_1$	$x_3]$	$[x_1$	$x_2$	$x_3$	$x_4]$	$\leq_{lex}$	$[x_4$	$x_3$	$x_1$	$x_2]$
$x_1 = 0$	0	0				0	1			0	0				0	1		
$x_1 = 1$	1	1	1	1		1	1	1	1	1	0				1	1		
$x_2 = 0$	0	0				0	1											

# Breaking Several Variable Symmetries

## Proposition 7 ([Puget05cp,Walsh06cp])

$AC(\bigwedge_{\sigma \in \Sigma} [x_1, \dots, x_n] \leq_{lex} [x_{\sigma(1)}, \dots, x_{\sigma(n)}])$  is strictly stronger than  $\bigwedge_{\sigma \in \Sigma} AC([x_1, \dots, x_n] \leq_{lex} [x_{\sigma(1)}, \dots, x_{\sigma(n)}])$ .

## Proof.

Let  $x_1, x_2, x_4 \in \{0, 1\}$  and  $x_3 = 1$ . Consider two symmetries (1243) and (1423), we have  $AC([x_1, x_2, x_3, x_4] \leq_{lex} [x_2, x_4, x_1, x_3])$  and  $AC([x_1, x_2, x_3, x_4] \leq_{lex} [x_4, x_3, x_1, x_2])$ .

cases	$[x_1$	$x_2$	$x_3$	$x_4]$	$\leq_{lex}$	$[x_2$	$x_4$	$x_1$	$x_3]$	$[x_1$	$x_2$	$x_3$	$x_4]$	$\leq_{lex}$	$[x_4$	$x_3$	$x_1$	$x_2]$
$x_1 = 0$	0	0				0	1			0	0				0	1		
$x_1 = 1$	1	1	1	1		1	1	1	1	1	0				1	1		
$x_2 = 0$	0	0				0	1			0	0				1			
$x_2 = 1$																		

# Breaking Several Variable Symmetries

## Proposition 7 ([Puget05cp,Walsh06cp])

$AC(\bigwedge_{\sigma \in \Sigma} [x_1, \dots, x_n] \leq_{lex} [x_{\sigma(1)}, \dots, x_{\sigma(n)}])$  is strictly stronger than  $\bigwedge_{\sigma \in \Sigma} AC([x_1, \dots, x_n] \leq_{lex} [x_{\sigma(1)}, \dots, x_{\sigma(n)}])$ .

## Proof.

Let  $x_1, x_2, x_4 \in \{0, 1\}$  and  $x_3 = 1$ . Consider two symmetries (1243) and (1423), we have  $AC([x_1, x_2, x_3, x_4] \leq_{lex} [x_2, x_4, x_1, x_3])$  and  $AC([x_1, x_2, x_3, x_4] \leq_{lex} [x_4, x_3, x_1, x_2])$ .

cases	$[x_1$	$x_2$	$x_3$	$x_4]$	$\leq_{lex}$	$[x_2$	$x_4$	$x_1$	$x_3]$	$[x_1$	$x_2$	$x_3$	$x_4]$	$\leq_{lex}$	$[x_4$	$x_3$	$x_1$	$x_2]$
$x_1 = 0$	0	0				0	1			0	0				0	1		
$x_1 = 1$	1	1	1	1		1	1	1	1	1	0				1	1		
$x_2 = 0$	0	0				0	1			0	0				1			
$x_2 = 1$	0	1				1												

# Breaking Several Variable Symmetries

## Proposition 7 ([Puget05cp,Walsh06cp])

$AC(\bigwedge_{\sigma \in \Sigma} [x_1, \dots, x_n] \leq_{lex} [x_{\sigma(1)}, \dots, x_{\sigma(n)}])$  is strictly stronger than  $\bigwedge_{\sigma \in \Sigma} AC([x_1, \dots, x_n] \leq_{lex} [x_{\sigma(1)}, \dots, x_{\sigma(n)}])$ .

## Proof.

Let  $x_1, x_2, x_4 \in \{0, 1\}$  and  $x_3 = 1$ . Consider two symmetries (1243) and (1423), we have  $AC([x_1, x_2, x_3, x_4] \leq_{lex} [x_2, x_4, x_1, x_3])$  and  $AC([x_1, x_2, x_3, x_4] \leq_{lex} [x_4, x_3, x_1, x_2])$ .

cases	$[x_1$	$x_2$	$x_3$	$x_4]$	$\leq_{lex}$	$[x_2$	$x_4$	$x_1$	$x_3]$	$[x_1$	$x_2$	$x_3$	$x_4]$	$\leq_{lex}$	$[x_4$	$x_3$	$x_1$	$x_2]$
$x_1 = 0$	0	0			0	1			0	0			0	0	1			
$x_1 = 1$	1	1	1	1	1	1	1	1	1	1	0			1	1			
$x_2 = 0$	0	0			0	1			0	0				1				
$x_2 = 1$	0	1			1				0	1				1				
$x_4 = 0$																		

# Breaking Several Variable Symmetries

## Proposition 7 ([Puget05cp,Walsh06cp])

$AC(\bigwedge_{\sigma \in \Sigma} [x_1, \dots, x_n] \leq_{lex} [x_{\sigma(1)}, \dots, x_{\sigma(n)}])$  is strictly stronger than  $\bigwedge_{\sigma \in \Sigma} AC([x_1, \dots, x_n] \leq_{lex} [x_{\sigma(1)}, \dots, x_{\sigma(n)}])$ .

## Proof.

Let  $x_1, x_2, x_4 \in \{0, 1\}$  and  $x_3 = 1$ . Consider two symmetries (1243) and (1423), we have  $AC([x_1, x_2, x_3, x_4] \leq_{lex} [x_2, x_4, x_1, x_3])$  and  $AC([x_1, x_2, x_3, x_4] \leq_{lex} [x_4, x_3, x_1, x_2])$ .

cases	$[x_1$	$x_2$	$x_3$	$x_4]$	$\leq_{lex}$	$[x_2$	$x_4$	$x_1$	$x_3]$	$[x_1$	$x_2$	$x_3$	$x_4]$	$\leq_{lex}$	$[x_4$	$x_3$	$x_1$	$x_2]$
$x_1 = 0$	0	0			0	1			0	0			0	0	1			
$x_1 = 1$	1	1	1	1	1	1	1	1	1	1	0			1	1			
$x_2 = 0$	0	0			0	1			0	0				1				
$x_2 = 1$	0	1			1				0	1				1				
$x_4 = 0$	0				1													

# Breaking Several Variable Symmetries

## Proposition 7 ([Puget05cp,Walsh06cp])

$AC(\bigwedge_{\sigma \in \Sigma} [x_1, \dots, x_n] \leq_{lex} [x_{\sigma(1)}, \dots, x_{\sigma(n)}])$  is strictly stronger than  $\bigwedge_{\sigma \in \Sigma} AC([x_1, \dots, x_n] \leq_{lex} [x_{\sigma(1)}, \dots, x_{\sigma(n)}])$ .

## Proof.

Let  $x_1, x_2, x_4 \in \{0, 1\}$  and  $x_3 = 1$ . Consider two symmetries (1243) and (1423), we have  $AC([x_1, x_2, x_3, x_4] \leq_{lex} [x_2, x_4, x_1, x_3])$  and  $AC([x_1, x_2, x_3, x_4] \leq_{lex} [x_4, x_3, x_1, x_2])$ .

cases	$[x_1$	$x_2$	$x_3$	$x_4]$	$\leq_{lex}$	$[x_2$	$x_4$	$x_1$	$x_3]$	$[x_1$	$x_2$	$x_3$	$x_4]$	$\leq_{lex}$	$[x_4$	$x_3$	$x_1$	$x_2]$
$x_1 = 0$	0	0			0	1			0	0			0	0	1			
$x_1 = 1$	1	1	1	1	1	1	1	1	1	1	0			1	1			
$x_2 = 0$	0	0			0	1			0	0				1				
$x_2 = 1$	0	1			1				0	1				1				
$x_4 = 0$	0				1				0	0				0	1			
$x_4 = 1$																		

# Breaking Several Variable Symmetries

## Proposition 7 ([Puget05cp,Walsh06cp])

$AC(\bigwedge_{\sigma \in \Sigma} [x_1, \dots, x_n] \leq_{lex} [x_{\sigma(1)}, \dots, x_{\sigma(n)}])$  is strictly stronger than  $\bigwedge_{\sigma \in \Sigma} AC([x_1, \dots, x_n] \leq_{lex} [x_{\sigma(1)}, \dots, x_{\sigma(n)}])$ .

## Proof.

Let  $x_1, x_2, x_4 \in \{0, 1\}$  and  $x_3 = 1$ . Consider two symmetries (1243) and (1423), we have  $AC([x_1, x_2, x_3, x_4] \leq_{lex} [x_2, x_4, x_1, x_3])$  and  $AC([x_1, x_2, x_3, x_4] \leq_{lex} [x_4, x_3, x_1, x_2])$ .

cases	$[x_1$	$x_2$	$x_3$	$x_4]$	$\leq_{lex}$	$[x_2$	$x_4$	$x_1$	$x_3]$	$[x_1$	$x_2$	$x_3$	$x_4]$	$\leq_{lex}$	$[x_4$	$x_3$	$x_1$	$x_2]$
$x_1 = 0$	0	0			0	1				0	0			0	1			
$x_1 = 1$	1	1	1	1	1	1	1	1	1	1	0			1	1			
$x_2 = 0$	0	0			0	1				0	0			1				
$x_2 = 1$	0	1			1					0	1			1				
$x_4 = 0$	0				1					0	0			0	1			
$x_4 = 1$	0				1													

# Breaking Several Variable Symmetries

## Proposition 7 ([Puget05cp,Walsh06cp])

$AC(\bigwedge_{\sigma \in \Sigma} [x_1, \dots, x_n] \leq_{lex} [x_{\sigma(1)}, \dots, x_{\sigma(n)}])$  is strictly stronger than  $\bigwedge_{\sigma \in \Sigma} AC([x_1, \dots, x_n] \leq_{lex} [x_{\sigma(1)}, \dots, x_{\sigma(n)}])$ .

### Proof.

Let  $x_1, x_2, x_4 \in \{0, 1\}$  and  $x_3 = 1$ . Consider two symmetries (1243) and (1423), we have  $AC([x_1, x_2, x_3, x_4] \leq_{lex} [x_2, x_4, x_1, x_3])$  and  $AC([x_1, x_2, x_3, x_4] \leq_{lex} [x_4, x_3, x_1, x_2])$ .

cases	$[x_1$	$x_2$	$x_3$	$x_4]$	$\leq_{lex}$	$[x_2$	$x_4$	$x_1$	$x_3]$	$[x_1$	$x_2$	$x_3$	$x_4]$	$\leq_{lex}$	$[x_4$	$x_3$	$x_1$	$x_2]$
$x_1 = 0$	0	0			0	1			0	0			0	0	1			
$x_1 = 1$	1	1	1	1	1	1	1	1	1	1	0			1	1			
$x_2 = 0$	0	0			0	1			0	0				1				
$x_2 = 1$	0	1			1				0	1				1				
$x_4 = 0$	0				1				0	0				0	1			
$x_4 = 1$	0				1				0					1				

□

However, their conjunction is not AC. Indeed, suppose that  $x_4 = 0$ ,

# Breaking Several Variable Symmetries

## Proposition 7 ([Puget05cp,Walsh06cp])

$AC(\bigwedge_{\sigma \in \Sigma} [x_1, \dots, x_n] \leq_{lex} [x_{\sigma(1)}, \dots, x_{\sigma(n)}])$  is strictly stronger than  $\bigwedge_{\sigma \in \Sigma} AC([x_1, \dots, x_n] \leq_{lex} [x_{\sigma(1)}, \dots, x_{\sigma(n)}])$ .

### Proof.

Let  $x_1, x_2, x_4 \in \{0, 1\}$  and  $x_3 = 1$ . Consider two symmetries (1243) and (1423), we have  $AC([x_1, x_2, x_3, x_4] \leq_{lex} [x_2, x_4, x_1, x_3])$  and  $AC([x_1, x_2, x_3, x_4] \leq_{lex} [x_4, x_3, x_1, x_2])$ .

cases	$[x_1$	$x_2$	$x_3$	$x_4]$	$\leq_{lex}$	$[x_2$	$x_4$	$x_1$	$x_3]$	$[x_1$	$x_2$	$x_3$	$x_4]$	$\leq_{lex}$	$[x_4$	$x_3$	$x_1$	$x_2]$
$x_1 = 0$	0	0			0	1			0	0			0	0	1			
$x_1 = 1$	1	1	1	1	1	1	1	1	1	1	0			1	1			
$x_2 = 0$	0	0			0	1			0	0				1				
$x_2 = 1$	0	1			1				0	1				1				
$x_4 = 0$	0				1				0	0				0	1			
$x_4 = 1$	0				1				0					1				

□

However, their conjunction is not AC. Indeed, suppose that  $x_4 = 0$ , we have  $x_1 = x_2 = 0$  and  $x_3 = 0$ , which is not possible.

## Value Symmetry Breaking

A **value symmetry** is a bijection  $\sigma$  on values that preserves solutions.

$\{x_i = v_i | 1 \leq i \leq n\}$  is a solution iff  $\{x_i = \sigma(v_i) | 1 \leq i \leq n\}$  is a solution

All value symmetries can be broken by posting for each value symmetry  $\sigma$

$[x_1, \dots, x_n] \leq_{lex} [\sigma(x_1), \dots, \sigma(x_n)]$  [PS03cp]

### Example 8 ( $\sigma(i) = n + 1 - i$ )

The symmetry breaking constraint implies  $x_1 \leq n + 1 - x_1$

If  $n$  is even, the constraint is thus equivalent to  $x_1 \leq \frac{n}{2}$

If  $n$  is odd, it is equivalent to  $x_1 \leq \frac{n+1}{2} \wedge x_1 = \frac{n+1}{2} \Rightarrow x_2 \leq \frac{n+1}{2} \wedge \dots$

## Breaking Variable and Value Symmetries

### Theorem 9 ([Puget05cp,Walsh06cp])

*The constraints  $[x_1, \dots, x_n] \leq_{lex} [x_{\sigma(1)}, \dots, x_{\sigma(n)}]$  for each variable symmetry  $\sigma \in \Sigma$*

*and  $[x_1, \dots, x_m] \leq_{lex} [\sigma'(x_1), \dots, \sigma'(x_n)]$  for each value symmetry  $\sigma' \in \Sigma'$*

*leave at least one assignment in each equivalence class of solutions.*

### Proof.

For any assignment  $\nu$ ,

## Breaking Variable and Value Symmetries

### Theorem 9 ([Puget05cp,Walsh06cp])

*The constraints  $[x_1, \dots, x_n] \leq_{lex} [x_{\sigma(1)}, \dots, x_{\sigma(n)}]$  for each variable symmetry  $\sigma \in \Sigma$*

*and  $[x_1, \dots, x_m] \leq_{lex} [\sigma'(x_1), \dots, \sigma'(x_n)]$  for each value symmetry  $\sigma' \in \Sigma'$*

*leave at least one assignment in each equivalence class of solutions.*

### Proof.

For any assignment  $\nu$ , one can pick the lex leader  $\nu_1$  of  $\nu$  under  $\Sigma$

## Breaking Variable and Value Symmetries

### Theorem 9 ([Puget05cp,Walsh06cp])

*The constraints  $[x_1, \dots, x_n] \leq_{lex} [x_{\sigma(1)}, \dots, x_{\sigma(n)}]$  for each variable symmetry  $\sigma \in \Sigma$*

*and  $[x_1, \dots, x_m] \leq_{lex} [\sigma'(x_1), \dots, \sigma'(x_n)]$  for each value symmetry  $\sigma' \in \Sigma'$*

*leave at least one assignment in each equivalence class of solutions.*

### Proof.

For any assignment  $\nu$ , one can pick the lex leader  $\nu_1$  of  $\nu$  under  $\Sigma$  and then the lex leader  $\nu_2$  of  $\nu_1$  under  $\Sigma'$

## Breaking Variable and Value Symmetries

### Theorem 9 ([Puget05cp,Walsh06cp])

*The constraints  $[x_1, \dots, x_n] \leq_{lex} [x_{\sigma(1)}, \dots, x_{\sigma(n)}]$  for each variable symmetry  $\sigma \in \Sigma$*

*and  $[x_1, \dots, x_m] \leq_{lex} [\sigma'(x_1), \dots, \sigma'(x_n)]$  for each value symmetry  $\sigma' \in \Sigma'$*

*leave at least one assignment in each equivalence class of solutions.*

### Proof.

For any assignment  $\nu$ , one can pick the lex leader  $\nu_1$  of  $\nu$  under  $\Sigma$  and then the lex leader  $\nu_2$  of  $\nu_1$  under  $\Sigma'$

If  $\nu_2$  does not satisfy the lex leader constraint under  $\Sigma$ , iterate.

As the lexicographic orders are well-founded, the process terminates, with an assignment that satisfies all lex leader constraints. □

## Breaking Several Variable and Value Symmetries

The iterated lex leader may leave several symmetric assignments.

### Example 10

Consider the composition of the reflection symmetries on both variables and boolean values.

The solutions  $[0, 1, 1]$  and  $[0, 0, 1]$  are symmetric but satisfy the lex constraints

$$[x_1, x_2, x_3] \leq [x_3, x_2, x_1]$$

$$[x_1, x_2, x_3] \leq [\neg x_1, \neg x_2, \neg x_3]$$

Indeed  $[0, 1, 1] \leq [1, 1, 0]$  and  $[0, 1, 1] \leq [1, 0, 0]$

$$[0, 0, 1] \leq [1, 0, 0] \text{ and } [0, 0, 1] \leq [1, 1, 0]$$

hence both symmetric solutions  $[0, 1, 1]$  and  $[0, 0, 1]$  are lex leaders.

## Variable-Value Symmetries

**Definition** A variable-value symmetry (or *general symmetry*) is a bijection  $\sigma$  on pairs (variable, value) that preserves solutions.

**Definition** A valuation  $[x_1, \dots, x_n]$  is admissible for  $\sigma$  iff  $|\{k \mid x_i = j, \sigma(i, j) = (k, l)\}| = n$ .

E.g. In the 3-queens, the assignment  $[2, 3, 1]$  is admissible for r90 but not  $[2, 3, 3]$ .

If  $[x_1, \dots, x_n]$  is **admissible** for  $\sigma$ , let  $\sigma[x_1, \dots, x_n]$  be its image under  $\sigma$ ,  $\sigma[x_1, \dots, x_n] = [y_1, \dots, y_n]$  where  $y_k = l$  whenever  $x_i = j$  and  $\sigma(i, j) = (k, l)$

# Variable-Value Symmetry Breaking

## Proposition 11

*All variable-value symmetries can be broken by posting the constraints*

$$\bigwedge_{\sigma \in \Sigma} \text{admissible}(\sigma, [x_1, \dots, x_n]) \wedge [x_1, \dots, x_n] \leq_{\text{lex}} \sigma[x_1, \dots, x_n]$$

## Example 12

In the 3-queens, let  $x_1 = 2$ ,  $x_2 \in \{1, 3\}$ ,  $x_3 \in \{1, 2, 3\}$   
 $\text{r90}[x_1, \dots, x_3]$  prunes  $x_3 \neq 2$  for admissibility, and  $x_3 \neq 1$ ,  $x_2 \neq 3$  for lex.

# SBDS

In 99, [BW99cp] proposed a completely different symmetry breaking technique, **Symmetry Breaking During Search** (SBDS).

It overcomes the main drawback of static symmetry breaking:

# SBDS

In 99, [BW99cp] proposed a completely different symmetry breaking technique, **Symmetry Breaking During Search** (SBDS).

It overcomes the main drawback of static symmetry breaking: the choice of the representative element in each class of solutions

# SBDS

In 99, [BW99cp] proposed a completely different symmetry breaking technique, **Symmetry Breaking During Search** (SBDS).

It overcomes the main drawback of static symmetry breaking: the choice of the representative element in each class of solutions is **forced**

breaking all trials at improving performance by

# SBDS

In 99, [BW99cp] proposed a completely different symmetry breaking technique, **Symmetry Breaking During Search** (SBDS).

It overcomes the main drawback of static symmetry breaking: the choice of the representative element in each class of solutions is **forced**

breaking all trials at improving performance by clever **search heuristics**

# Symmetric Constraints

Consider a set  $\Sigma$  of symmetries, such that for any constraint  $c$  and all  $\sigma \in \Sigma$  one can find a constraint  $\sigma(c)$  corresponding to the symmetric of  $c$

$$\mathcal{X} \models \sigma(c)\rho \Leftrightarrow c\sigma(\rho)$$

For example, if  $\sigma$  is the value symmetry that turns  $v$  into  $N - v$  we have  $\sigma(x = v)$  is

# Symmetric Constraints

Consider a set  $\Sigma$  of symmetries, such that for any constraint  $c$  and all  $\sigma \in \Sigma$  one can find a constraint  $\sigma(c)$  corresponding to the symmetric of  $c$

$$\mathcal{X} \models \sigma(c)\rho \Leftrightarrow c\sigma(\rho)$$

For example, if  $\sigma$  is the value symmetry that turns  $v$  into  $N - v$  we have  $\sigma(X = v)$  is  $X = (N - v)$

We can now define a technique for removing symmetries adding constraints when choice-points are explored, *à la* branch and bound.

# Enumerating Solutions

The general method of enumeration of solutions is, at each choice-point, to add

- on one branch the constraint  $c$  assigning a value to a variable;
- on the other branch the negation of this constraint  $\neg c$

SBDS adds supplementary constraints on the second branch:

supposing a partial assignment  $\mathcal{A}$  at the choice-point, for all  $\sigma \in \Sigma$  such that  $\sigma(\mathcal{A}) = \mathcal{A}$  one adds  $\sigma(\neg c)$ .

## Example

Consider the 4-queens problem over  $X_1, X_2, X_3, X_4 \in \{1, 2, 3, 4\}$

with a single (value-)symmetry:  $v \mapsto 5 - v$

suppose that at the top of the search tree the leftmost branch corresponds to  $X_1 = 1$

when backtracking at the top, the next branch to explore will correspond to the constraint:

$$X_1 \neq 1$$

## Example

Consider the 4-queens problem over  $X_1, X_2, X_3, X_4 \in \{1, 2, 3, 4\}$

with a single (value-)symmetry:  $v \mapsto 5 - v$

suppose that at the top of the search tree the leftmost branch corresponds to  $X_1 = 1$

when backtracking at the top, the next branch to explore will correspond to the constraint:

$$X_1 \neq 1 \wedge X_1 \neq 4$$

# Unicity

## Theorem 13 (Non-symmetric Solutions)

*If  $\rho_1$  and  $\rho_2$  are two solutions obtained by SBDS, then*

$$\forall \sigma \in \Sigma \quad \sigma(\rho_1) \neq \rho_2$$

# Unicity

## Theorem 13 (Non-symmetric Solutions)

*If  $\rho_1$  and  $\rho_2$  are two solutions obtained by SBDS, then*

$$\forall \sigma \in \Sigma \quad \sigma(\rho_1) \neq \rho_2$$

### Proof.

Suppose that  $\sigma_0(\rho_1) = \rho_2$  for some  $\sigma_0$

# Unicity

## Theorem 13 (Non-symmetric Solutions)

If  $\rho_1$  and  $\rho_2$  are two solutions obtained by SBDS, then

$$\forall \sigma \in \Sigma \quad \sigma(\rho_1) \neq \rho_2$$

### Proof.

Suppose that  $\sigma_0(\rho_1) = \rho_2$  for some  $\sigma_0$   
let  $\mathcal{A}$  be the partial assignment at the choice-point that differentiates the  $\rho_1$  and  $\rho_2$  branches, and  $c$  the constraint added on the  $\rho_1$  branch there.

# Unicity

## Theorem 13 (Non-symmetric Solutions)

If  $\rho_1$  and  $\rho_2$  are two solutions obtained by SBDS, then

$$\forall \sigma \in \Sigma \quad \sigma(\rho_1) \neq \rho_2$$

### Proof.

Suppose that  $\sigma_0(\rho_1) = \rho_2$  for some  $\sigma_0$   
let  $\mathcal{A}$  be the partial assignment at the choice-point that differentiates the  $\rho_1$  and  $\rho_2$  branches, and  $c$  the constraint added on the  $\rho_1$  branch there.

We have  $\sigma_0(\mathcal{A}) = \mathcal{A}$

# Unicity

## Theorem 13 (Non-symmetric Solutions)

If  $\rho_1$  and  $\rho_2$  are two solutions obtained by SBDS, then

$$\forall \sigma \in \Sigma \quad \sigma(\rho_1) \neq \rho_2$$

### Proof.

Suppose that  $\sigma_0(\rho_1) = \rho_2$  for some  $\sigma_0$   
let  $\mathcal{A}$  be the partial assignment at the choice-point that differentiates the  $\rho_1$  and  $\rho_2$  branches, and  $c$  the constraint added on the  $\rho_1$  branch there.

We have  $\sigma_0(\mathcal{A}) = \mathcal{A}$

since both are solutions, we get that  $c$  is true in  $\rho_1$

# Unicity

## Theorem 13 (Non-symmetric Solutions)

If  $\rho_1$  and  $\rho_2$  are two solutions obtained by SBDS, then

$$\forall \sigma \in \Sigma \quad \sigma(\rho_1) \neq \rho_2$$

### Proof.

Suppose that  $\sigma_0(\rho_1) = \rho_2$  for some  $\sigma_0$   
let  $\mathcal{A}$  be the partial assignment at the choice-point that differentiates the  $\rho_1$  and  $\rho_2$  branches, and  $c$  the constraint added on the  $\rho_1$  branch there.

We have  $\sigma_0(\mathcal{A}) = \mathcal{A}$

since both are solutions, we get that  $c$  is true in  $\rho_1$   
and that  $\sigma_0(\neg c)$  is true in  $\rho_2$

# Unicity

## Theorem 13 (Non-symmetric Solutions)

If  $\rho_1$  and  $\rho_2$  are two solutions obtained by SBDS, then

$$\forall \sigma \in \Sigma \quad \sigma(\rho_1) \neq \rho_2$$

### Proof.

Suppose that  $\sigma_0(\rho_1) = \rho_2$  for some  $\sigma_0$   
let  $\mathcal{A}$  be the partial assignment at the choice-point that differentiates the  $\rho_1$  and  $\rho_2$  branches, and  $c$  the constraint added on the  $\rho_1$  branch there.

We have  $\sigma_0(\mathcal{A}) = \mathcal{A}$

since both are solutions, we get that  $c$  is true in  $\rho_1$   
and that  $\sigma_0(\neg c)$  is true in  $\rho_2$  i.e.,  $\neg c$  is true in  $\rho_1$

$\Rightarrow$  contradiction □

## About Partial Assignments

If one adds systematically  $\sigma(\neg c)$  even when  $\sigma(\mathcal{A}) \neq \mathcal{A}$

## About Partial Assignments

If one adds systematically  $\sigma(\neg c)$  even when  $\sigma(\mathcal{A}) \neq \mathcal{A}$   
one **loses solutions!**

### Example 14

Consider again the 4 queens problem,  
at some point we explore the branch  $X_1 = 2$  and then  $X_2 = 1$

## About Partial Assignments

If one adds systematically  $\sigma(\neg c)$  even when  $\sigma(\mathcal{A}) \neq \mathcal{A}$   
one **loses solutions!**

### Example 14

Consider again the 4 queens problem,  
at some point we explore the branch  $X_1 = 2$  and then  $X_2 = 1$   
 $\Rightarrow$  failure

## About Partial Assignments

If one adds systematically  $\sigma(\neg c)$  even when  $\sigma(\mathcal{A}) \neq \mathcal{A}$   
one **loses solutions!**

### Example 14

Consider again the 4 queens problem,  
at some point we explore the branch  $X_1 = 2$  and then  $X_2 = 1$   
 $\Rightarrow$  failure  $\Rightarrow X_2 \neq 4$   
we never find any solution. . .

Conversely, new **local** symmetries might appear in some partial assignments (the overhead of handling those is usually not worth it).

## Detecting Symmetries

[GHK02cp] show that constraint symmetries such as those considered for SDBS form a group

they link CSPs (in ECLiPSe) with the GAP computational abstract algebra system

many symmetries (even local ones) can be detected automatically

remains costly and not much used...