

Why are Boolean CSP interesting ?

Complexity of Abduction

Bases for Boolean co-clones

Non-exhaustive list of references (only those mentioned in the talk)

# Boolean CSP

Nadia CREIGNOU

LIF, Université de la Méditerranée

Maths CSP 2006

## Outline

- 1 Why are Boolean CSP interesting ?
  - Preliminaries
  - When does the Galois connection apply to complexity ?
- 2 Complexity of Abduction
  - Definition of the problem
  - Known results
  - New tractable cases
  - A complete classification
- 3 Bases for Boolean co-clones
  - Bases and plain bases for Boolean co-clones
  - The infinite part of Post's lattice
  - Preferred representations
  - Algorithmic applications
- 4 Non-exhaustive list of references (only those mentioned in the talk)

# Outline

- 1 Why are Boolean CSP interesting ?
  - Preliminaries
  - When does the Galois connection apply to complexity ?
- 2 Complexity of Abduction
  - Definition of the problem
  - Known results
  - New tractable cases
  - A complete classification
- 3 Bases for Boolean co-clones
  - Bases and plain bases for Boolean co-clones
  - The infinite part of Post's lattice
  - Preferred representations
  - Algorithmic applications
- 4 Non-exhaustive list of references (only those mentioned in the talk)

# Boolean CSP are interesting

- non trivial
- many complete classifications
- Post's lattice

We can learn a lot from them :

- guess when the Galois connection applies
- a better description of Post relational lattice
- there are still some open questions !

# S-formulas

- Let  $S$  be a set of Boolean relations.
- $C = R(x_1, \dots, x_n)$  with  $R \in S$  is an  $S$ -constraint
- $F = \bigwedge C_i$  where each  $C_i$  is an  $S$ -constraint is a CNF( $S$ )-formula

SAT( $S$ )

Input :  $F$  an  $S$ -formula

Question : Is  $F$  satisfiable ?

# expressive power of $S$

$\langle S \rangle$  is the co-clone generated by  $S$ , that is the set of all relations  $R$  which can be implemented as :

$$R(x_1, \dots, x_n) \equiv \exists x_{n+1} \dots \exists x_m F(\vec{x}),$$

where  $F$  is an  $S$ -formula

# Galois correspondence

- $\text{Pol}(S)$  is the set of Boolean functions  $f$  that are *polymorphisms of* (which *preserve*) every relation in  $S$ .  $\text{Pol}(S)$  is a **clone**.
- $\text{Inv}(B)$  is the set of relations that are preserved by every function in  $B$ , it is a **co-clone**

$\text{Pol}$  and  $\text{Inv}$  form a Galois correspondence between closed sets of relations and closed sets of Boolean functions

$$\langle S \rangle = \text{Inv}(\text{Pol}(S)).$$

The expressive power of  $S$  depends on the (co-)clone generated by  $S$

# Galois correspondence

- $\text{Pol}(S)$  is the set of Boolean functions  $f$  that are *polymorphisms of* (which *preserve*) every relation in  $S$ .  
 $\text{Pol}(S)$  is a **clone**.
- $\text{Inv}(B)$  is the set of relations that are preserved by every function in  $B$ , it is a co-clone

$\text{Pol}$  and  $\text{Inv}$  form a Galois correspondence between closed sets of relations and closed sets of Boolean functions

$$\langle S \rangle = \text{Inv}(\text{Pol}(S)).$$

The expressive power of  $S$  depends on the (co-)clone generated by  $S$

# Galois correspondence applies to complexity

$$\text{SAT}(\mathcal{S}) \equiv_m^{\log} \text{SAT}(\langle \mathcal{S} \rangle)$$

The complexity of  $\text{SAT}(\mathcal{S})$  depends on the clone  $\text{Pol}(\mathcal{S})$ .

If  $\mathcal{S}_1$  and  $\mathcal{S}_2$  are two sets of relations such that  $\mathcal{S}_1$  is finite and  $\text{Pol}(\mathcal{S}_2) \subseteq \text{Pol}(\mathcal{S}_1)$ , then

$$\text{SAT}(\mathcal{S}_1) \leq \text{SAT}(\mathcal{S}_2).$$

⇒ In the Boolean case all clones are identified (Post 1941)

# Galois correspondence applies to complexity

$$\text{SAT}(\mathcal{S}) \equiv_m^{\log} \text{SAT}(\langle \mathcal{S} \rangle)$$

The complexity of  $\text{SAT}(\mathcal{S})$  depends on the clone  $\text{Pol}(\mathcal{S})$ .

If  $\mathcal{S}_1$  and  $\mathcal{S}_2$  are two sets of relations such that  $\mathcal{S}_1$  is finite and  $\text{Pol}(\mathcal{S}_2) \subseteq \text{Pol}(\mathcal{S}_1)$ , then

$$\text{SAT}(\mathcal{S}_1) \leq \text{SAT}(\mathcal{S}_2).$$

⇒ In the Boolean case all clones are identified (Post 1941)



# Decision problem : SAT(S)

SAT(S)

**Input :** F an S-formula

**Question :** Is F satisfiable ?

**Theorem** (Schaefer, 1978)

- if S is 0-valid (1-valid), bijunctive, Horn(dual Horn) or affine, then SAT(S) is in P,
- otherwise SAT(S) is NP-complete.

Why are Boolean CSP interesting ?

Complexity of Abduction

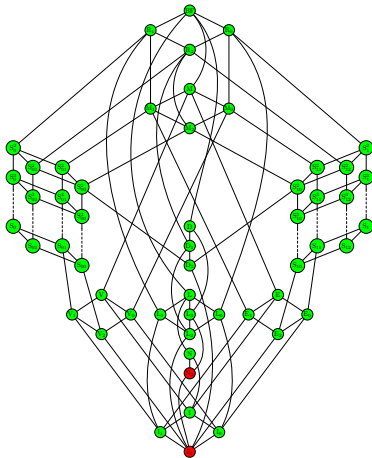
Bases for Boolean co-clones

Non-exhaustive list of references (only those mentioned in the talk)

Preliminaries

When does the Galois connection apply to complexity ?

# Decision problem : SAT(S)



# Counting problem : $\#SAT(S)$

$\#SAT(S)$

**Input :**  $F$  an  $S$ -formula

**Question :** How many satisfying assignments for  $F$  ?

## Theorem

- if  $S$  is affine, then  $\#SAT(S)$  is in FP,
- otherwise  $\#SAT(S)$  is  $\#P$ -complete.

Why are Boolean CSP interesting ?

Complexity of Abduction

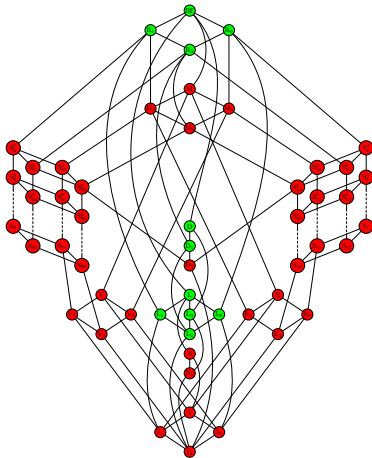
Bases for Boolean co-clones

Non-exhaustive list of references (only those mentioned in the talk)

Preliminaries

When does the Galois connection apply to complexity ?

# Counting problem : $\#SAT(S)$



# Enumeration problem

Enumerate SAT(S)

**Input :** F an S-formula

**Question :** Enumerate all the satisfying assignments.

## Theorem

- if S is bijunctive, Horn (dual Horn) or affine, then one can enumerate all the solutions with polynomial delay
- otherwise such an algorithm does not exist unless  $P=NP$ .

Why are Boolean CSP interesting ?

Complexity of Abduction

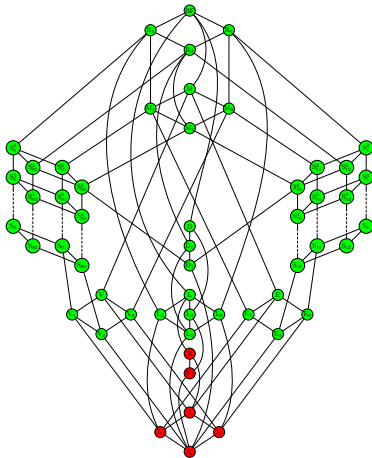
Bases for Boolean co-clones

Non-exhaustive list of references (only those mentioned in the talk)

Preliminaries

When does the Galois connection apply to complexity ?

# Enumeration problem



# Optimization problem : Max-SAT(S)

## Max-SAT(S)

**Input :** F an S-formula

**Question** Find an assignment that satisfies a maximum number of S-clauses

### Definition

A relation is 2-monotone if it can be expressed as a DNF-formula either of the form  $(x_1 \wedge \dots \wedge x_p)$  or  $(\neg y_1 \wedge \dots \wedge \neg y_q)$  or  $(x_1 \wedge \dots \wedge x_p) \vee (\neg y_1 \wedge \dots \wedge \neg y_q)$ .

### Theorem

- if S is 0-valid (1-valid) or 2-monotone, then Max-SAT(S) is in PO,
- otherwise Max-SAT(S) is APX-complete.

Why are Boolean CSP interesting ?

Complexity of Abduction

Bases for Boolean co-clones

Non-exhaustive list of references (only those mentioned in the talk)

Preliminaries

When does the Galois connection apply to complexity ?

## Optimization problem : Max-SAT(S)

No picture !

2-monotone relations do not constitute a co-clone

Why are Boolean CSP interesting ?

Complexity of Abduction

Bases for Boolean co-clones

Non-exhaustive list of references (only those mentioned in the talk)

Preliminaries

When does the Galois connection apply to complexity ?

## Optimization problem : Max-SAT(S)

No picture !

2-monotone relations do not constitute a co-clone

Why are Boolean CSP interesting ?

Complexity of Abduction

Bases for Boolean co-clones

Non-exhaustive list of references (only those mentioned in the talk)

Preliminaries

When does the Galois connection apply to complexity ?

## Optimization problem : Max-SAT(S)

No picture !

2-monotone relations do not constitute a co-clone

# Summary

Pb	Class. in lattice	Does Galois apply ?
SAT(S)	yes	yes
#SAT(S)	yes	yes
Enumeration	yes	no
Max-SAT(S)	no	no

A. Bulatov, V. Dalmau

H. Schnoor, I. Schnoor

# Summary

Pb	Class. in lattice	Does Galois apply ?
SAT(S)	yes	yes
#SAT(S)	yes	yes
Enumeration	yes	no
Max-SAT(S)	no	no

A. Bulatov, V. Dalmau

H. Schnoor, I. Schnoor

# Summary

Pb	Class. in lattice	Does Galois apply ?
SAT(S)	yes	yes
#SAT(S)	yes	yes
Enumeration	yes	no
Max-SAT(S)	no	no

A. Bulatov, V. Dalmau

H. Schnoor, I. Schnoor

# Summary

Pb	Class. in lattice	Does Galois apply ?
SAT(S)	yes	yes
#SAT(S)	yes	yes
Enumeration	yes	no
Max-SAT(S)	no	no

A. Bulatov, V. Dalmau

H. Schnoor, I. Schnoor

# Summary

Pb	Class. in lattice	Does Galois apply ?
SAT(S)	yes	yes
#SAT(S)	yes	yes
Enumeration	yes	no
Max-SAT(S)	no	no

A. Bulatov, V. Dalmau

H. Schnoor, I. Schnoor

# Summary

Pb	Class. in lattice	Does Galois apply ?
SAT(S)	yes	yes
#SAT(S)	yes	yes
Enumeration	yes	no
Max-SAT(S)	no	no

A. Bulatov, V. Dalmau

H. Schnoor, I. Schnoor

# Summary

Pb	Class. in lattice	Does Galois apply ?
SAT(S)	yes	yes
#SAT(S)	yes	yes
Enumeration	yes	no
Max-SAT(S)	no	no

A. Bulatov, V. Dalmau

H. Schnoor, I. Schnoor

# Summary

Pb	Class. in lattice	Does Galois apply ?
SAT(S)	yes	yes
#SAT(S)	yes	yes
Enumeration	yes	no
Max-SAT(S)	no	no

A. Bulatov, V. Dalmau

H. Schnoor, I. Schnoor

# Summary

Pb	Class. in lattice	Does Galois apply ?
SAT(S)	yes	yes
#SAT(S)	yes	yes
Enumeration	yes	no
Max-SAT(S)	no	no

A. Bulatov, V. Dalmau

H. Schnoor, I. Schnoor

# Summary

Pb	Class. in lattice	Does Galois apply ?
SAT(S)	yes	yes
#SAT(S)	yes	yes
Enumeration	yes	no
Max-SAT(S)	no	no

A. Bulatov, V. Dalmau

H. Schnoor, I. Schnoor

# Summary

Pb	Class. in lattice	Does Galois apply ?
SAT(S)	yes	yes
#SAT(S)	yes	yes
Enumeration	yes	no
Max-SAT(S)	no	no

A. Bulatov, V. Dalmau

H. Schnoor, I. Schnoor

# Summary

Pb	Class. in lattice	Does Galois apply ?
SAT(S)	yes	yes
#SAT(S)	yes	yes
Enumeration	yes	no
Max-SAT(S)	no	no

A. Bulatov, V. Dalmau

H. Schnoor, I. Schnoor

Why are Boolean CSP interesting ?

Complexity of Abduction

Bases for Boolean co-clones

Non-exhaustive list of references (only those mentioned in the talk)

Preliminaries

When does the Galois connection apply to complexity ?

## Open Question 1

Can we identify the computational goals for which the Galois connection applies ?

# Outline

- 1 Why are Boolean CSP interesting ?
  - Preliminaries
  - When does the Galois connection apply to complexity ?
- 2 Complexity of Abduction
  - Definition of the problem
  - Known results
  - New tractable cases
  - A complete classification
- 3 Bases for Boolean co-clones
  - Bases and plain bases for Boolean co-clones
  - The infinite part of Post's lattice
  - Preferred representations
  - Algorithmic applications
- 4 Non-exhaustive list of references (only those mentioned in the talk)

# Abduction

$L$  denotes a finite set of Boolean relations

PQ-Abduction( $L$ )

**Input :** An  $L$ -formula  $\varphi$ , a set of variables  $A \subseteq \text{Vars}(\varphi)$  and a variable  $q \in \text{Vars}(\varphi) \setminus A$

**Question :** Is there a set  $E \subseteq \text{Lits}(A)$  such that  $\varphi \wedge \bigwedge E$  is satisfiable but  $\varphi \wedge \bigwedge E \wedge \neg q$  is not ?

If one exists, such a set  $E$  is called a *solution* of the abduction problem.

## Known hard problems

- General problem (Eiter, Gottlob 1995)  
If  $\mathcal{C}$  is the class of all propositional CNF formulas, then  $\text{PQ-ABDUCTION}(\mathcal{C})$  is  $\Sigma_2\text{P}$ -complete.
- (dual) Horn abduction (Selman, Levesque 1990)  
If  $\mathcal{C}$  is the class of all propositional Horn formulas, then  $\text{PQ-ABDUCTION}(\mathcal{C})$  is NP-complete. The same holds if  $\mathcal{C}$  is the class of all propositional dual Horn formulas.

## Known hard problems

- General problem (Eiter, Gottlob 1995)  
If  $\mathcal{C}$  is the class of all propositional CNF formulas, then  $\text{PQ-ABDUCTION}(\mathcal{C})$  is  $\Sigma_2\text{P}$ -complete.
- (dual) Horn abduction (Selman, Levesque 1990)  
If  $\mathcal{C}$  is the class of all propositional Horn formulas, then  $\text{PQ-ABDUCTION}(\mathcal{C})$  is NP-complete. The same holds if  $\mathcal{C}$  is the class of all propositional dual Horn formulas.

## Known tractable problems

- Affine (Zanuttini 2003)  
If  $L$  is an affine language, then the problem  $PQ\text{-ABDUCTION}(L)$  is in  $P$ .
- Bijunctive (Marquis 2000)  
If  $L$  is a bijunctive language, then the problem  $PQ\text{-ABDUCTION}(L)$  is in  $P$ .
- Definite Horn (Eiter, Gottlob 1995)  
If  $L$  is a definite Horn language, then the problem  $PQ\text{-ABDUCTION}(L)$  is in  $P$ .

## Tools for easy cases

### Definition

A clause  $C = \bigvee_{i \in I} l_i$  is said to be a *prime implicate* of  $\varphi$  if  $\varphi \wedge \bigwedge \{\neg l_i \mid i \in I\}$  is unsatisfiable but for all  $i_0 \in I$ , the formula  $\varphi \wedge \bigwedge \{\neg l_i \mid i \in I, i \neq i_0\}$  is satisfiable.

### Lemma

- An abduction problem  $(\varphi, A, q)$  has a solution if and only if there is a prime implicate of  $\varphi$  of the form  $(l_1 \vee \dots \vee l_k \vee q)$  where for all  $i$ ,  $l_i$  is a literal built upon a variable in  $A$ .
- a solution is then  $E = (\neg l_1 \wedge \dots \wedge \neg l_k)$

$\Rightarrow$  all the prime implicates of a given formula  $\varphi$  in CNF can be generated by repeatedly applying *resolution*.

## Tools for easy cases

### Definition

A clause  $C = \bigvee_{i \in I} l_i$  is said to be a *prime implicate* of  $\varphi$  if  $\varphi \wedge \bigwedge \{\neg l_i \mid i \in I\}$  is unsatisfiable but for all  $i_0 \in I$ , the formula  $\varphi \wedge \bigwedge \{\neg l_i \mid i \in I, i \neq i_0\}$  is satisfiable.

### Lemma

- An abduction problem  $(\varphi, A, q)$  has a solution if and only if there is a prime implicate of  $\varphi$  of the form  $(l_1 \vee \dots \vee l_k \vee q)$  where for all  $i$ ,  $l_i$  is a literal built upon a variable in  $A$ .
- a solution is then  $E = (\neg l_1 \wedge \dots \wedge \neg l_k)$

$\Rightarrow$  all the prime implicates of a given formula  $\varphi$  in CNF can be generated by repeatedly applying *resolution*.

## Tools for easy cases

### Definition

A clause  $C = \bigvee_{i \in I} l_i$  is said to be a *prime implicate* of  $\varphi$  if  $\varphi \wedge \bigwedge \{\neg l_i \mid i \in I\}$  is unsatisfiable but for all  $i_0 \in I$ , the formula  $\varphi \wedge \bigwedge \{\neg l_i \mid i \in I, i \neq i_0\}$  is satisfiable.

### Lemma

- An abduction problem  $(\varphi, A, q)$  has a solution if and only if there is a prime implicate of  $\varphi$  of the form  $(l_1 \vee \dots \vee l_k \vee q)$  where for all  $i$ ,  $l_i$  is a literal built upon a variable in  $A$ .
- a solution is then  $E = (\neg l_1 \wedge \dots \wedge \neg l_k)$

$\Rightarrow$  all the prime implicates of a given formula  $\varphi$  in CNF can be generated by repeatedly applying *resolution*.

# Implicative Hitting Set-Bounded IHS-B

## Definition

A relation is

- *IHS-B*– if it can be described by a formula in CNF whose clauses are all of one of the following types :  $(x_i)$ , or  $(\neg x_{i_1} \vee x_{i_2})$ , or  $(\neg x_{i_1} \vee \dots \vee \neg x_{i_k})$
  - *IHS-B*+ if it can be described by a formula in CNF whose clauses are all of one of the following types :  $(\neg x_i)$ , or  $(\neg x_{i_1} \vee x_{i_2})$ , or  $(x_{i_1} \vee \dots \vee x_{i_k})$
- *IHS-B*– : any solution for the abduction pb contains 0 or 1 literal.
  - *IHS-B*+ : there are  $O(n^k)$  prime implicates

# Implicative Hitting Set-Bounded IHS-B

## Definition

A relation is

- *IHS-B*<sup>-</sup> if it can be described by a formula in CNF whose clauses are all of one of the following types :  $(x_i)$ , or  $(\neg x_{i_1} \vee x_{i_2})$ , or  $(\neg x_{i_1} \vee \dots \vee \neg x_{i_k})$
- *IHS-B*<sup>+</sup> if it can be described by a formula in CNF whose clauses are all of one of the following types :  $(\neg x_i)$ , or  $(\neg x_{i_1} \vee x_{i_2})$ , or  $(x_{i_1} \vee \dots \vee x_{i_k})$
- *IHS-B*<sup>-</sup> : any solution for the abduction pb contains 0 or 1 literal.
- *IHS-B*<sup>+</sup> : there are  $O(n^k)$  prime implicates

# Implicative Hitting Set-Bounded IHS-B

- If  $L$  is an IHS- $B^-$  language, then the problem PQ-ABDUCTION( $L$ ) is in P.
- If  $L$  is an IHS- $B^+$  language, then the problem PQ-ABDUCTION( $L$ ) is in P.

⇒ the polynomial complexity obtained here strongly relies on the fact that  $L$  is finite.

# Implicative Hitting Set-Bounded IHS-B

- If  $L$  is an IHS- $B_-$  language, then the problem PQ-ABDUCTION( $L$ ) is in P.
- If  $L$  is an IHS- $B_+$  language, then the problem PQ-ABDUCTION( $L$ ) is in P.

⇒ the polynomial complexity obtained here strongly relies on the fact that  $L$  is finite.

## Tools for hardness results

### Lemma

*If  $L$  can be implemented by  $L'$ , i.e., every relation in  $L$  can be expressed as a conjunctive query over  $L'$ , then*

$$\text{PQ-ABDUCTION}(L) \leq \text{PQ-ABDUCTION}(L')$$

⇒ Hardness results are obtained as Schaefer's original ones

## Tools for hardness results

### Lemma

*If  $L$  can be implemented by  $L'$ , i.e., every relation in  $L$  can be expressed as a conjunctive query over  $L'$ , then*

$$\text{PQ-ABDUCTION}(L) \leq \text{PQ-ABDUCTION}(L')$$

$\Rightarrow$  Hardness results are obtained as Schaefer's original ones

# Main theorem

Let  $L$  be a finite constraint language. [Creignou, Zanuttini, 2005]

- if  $L$  is bijunctive, affine, definite Horn, IHS- $B_+$  or IHS- $B_-$ , the problem PQ-ABDUCTION( $L$ ) is polynomial,
- otherwise, if  $L$  is Horn or dual Horn, the problem PQ-ABDUCTION( $L$ ) is NP-complete,
- in all other cases, the problem PQ-ABDUCTION( $L$ ) is  $\Sigma_2$ P-complete.

Each of these conditions can be checked in polynomial time given a language written in extension.

**Non trivial at this stage**

Why are Boolean CSP interesting ?

## Complexity of Abduction

Bases for Boolean co-clones

Non-exhaustive list of references (only those mentioned in the talk)

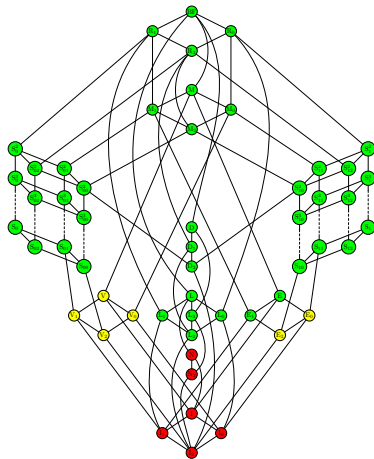
Definition of the problem

Known results

New tractable cases

A complete classification

# Complexity of abduction



# Global tractability versus tractability

## Definition

A constraint language  $S$  is called *globally tractable* for a problem  $\mathcal{A}$ , if  $\mathcal{A}(S)$  is tractable, and it is called *tractable* if for every finite  $L \subseteq S$ ,  $\mathcal{A}(L)$  is tractable

⇒ These two notions

- coincide for most of the computational problems
- **do not coincide** for the abduction problem (for IHS- $B+$  languages)

# Global tractability versus tractability

## Definition

A constraint language  $S$  is called *globally tractable* for a problem  $\mathcal{A}$ , if  $\mathcal{A}(S)$  is tractable, and it is called *tractable* if for every finite  $L \subseteq S$ ,  $\mathcal{A}(L)$  is tractable

⇒ These two notions

- coincide for most of the computational problems
- **do not coincide** for the abduction problem (for IHS- $B^+$  languages)

Why are Boolean CSP interesting ?

Complexity of Abduction

Bases for Boolean co-clones

Non-exhaustive list of references (only those mentioned in the talk)

Definition of the problem

Known results

New tractable cases

A complete classification

## Open Question 2

Can we identify/recognize the computational goals for which the notions of tractability and global tractability coincide ?

## Open Question 3

(basic) Propositional circumscription

**Input :**  $\varphi$  an S-formula and  $c$  a clause

**Question :** Is  $c$  satisfied in every minimal model of  $\varphi$  ?

Conjecture (Kirousis, Kolaitis) :

a trichotomy P, coNP-complete or  $\Pi_2^P$ -complete holds

# Outline

- 1 Why are Boolean CSP interesting ?
  - Preliminaries
  - When does the Galois connection apply to complexity ?
- 2 Complexity of Abduction
  - Definition of the problem
  - Known results
  - New tractable cases
  - A complete classification
- 3 Bases for Boolean co-clones
  - Bases and plain bases for Boolean co-clones
  - The infinite part of Post's lattice
  - Preferred representations
  - Algorithmic applications
- 4 Non-exhaustive list of references (only those mentioned in the talk)

## Bases and plain bases

Given a co-clone  $ICI$ , a set of relations  $B \subseteq ICI$  is called

- a *basis* for  $ICI$  if for every  $R \in ICI$

$$R(x_1, \dots, x_n) \equiv \exists \{y_1, \dots, y_m\} \mathcal{C}(\vec{x}, \vec{y})$$

where  $\mathcal{C}$  is some conjunction of constraints using only relations in  $B$ .

- a *plain basis* for  $ICI$  if for every  $R \in ICI$

$$R(x_1, \dots, x_n) \equiv \mathcal{C}(\vec{x})$$

where  $\mathcal{C}$  is some conjunction of constraints using only relations in  $B$ , and the scope of every constraint in  $\mathcal{C}$  is a sequence of distinct variables.

## Bases and plain bases

Given a co-clone  $ICI$ , a set of relations  $B \subseteq ICI$  is called

- a **basis** for  $ICI$  if for every  $R \in ICI$

$$R(x_1, \dots, x_n) \equiv \exists \{y_1, \dots, y_m\} \mathcal{C}(\vec{x}, \vec{y})$$

where  $\mathcal{C}$  is some conjunction of constraints using only relations in  $B$ .

- a **plain basis** for  $ICI$  if for every  $R \in ICI$

$$R(x_1, \dots, x_n) \equiv \mathcal{C}(\vec{x})$$

where  $\mathcal{C}$  is some conjunction of constraints using only relations in  $B$ , and the scope of every constraint in  $\mathcal{C}$  is a sequence of distinct variables.

## Bases and plain bases

Given a co-clone  $ICI$ , a set of relations  $B \subseteq ICI$  is called

- a **basis** for  $ICI$  if for every  $R \in ICI$

$$R(x_1, \dots, x_n) \equiv \exists \{y_1, \dots, y_m\} \mathcal{C}(\vec{x}, \vec{y})$$

where  $\mathcal{C}$  is some conjunction of constraints using only relations in  $B$ .

- a **plain basis** for  $ICI$  if for every  $R \in ICI$

$$R(x_1, \dots, x_n) \equiv \mathcal{C}(\vec{x})$$

where  $\mathcal{C}$  is some conjunction of constraints using only relations in  $B$ , and the scope of every constraint in  $\mathcal{C}$  is a sequence of distinct variables.

## Bases and plain bases differ

In the definition of plain basis :

- there is **no existential variables**
- **no replication of variables** in the scope of each constraint.

## Bases and plain bases differ

In the definition of plain basis :

- there is **no existential variables**
- **no replication of variables** in the scope of each constraint.

# Bases and plain bases

co-clone	(plain) basis	property
$IS_{00}^n$	$\{(\neg x), (\neg x \vee y)\} \cup \{(x_1 \vee \dots \vee x_k) \mid k \leq n\}$	IHSB $^+_n$
$IS_{00}$	$\{(\neg x), (\neg x \vee y)\} \cup \{(x_1 \vee \dots \vee x_k) \mid k \in \mathbb{N}\}$	IHSB $^+$
$IS_{10}^n$	$\{(x), (\neg x \vee y)\} \cup \{(\neg x_1 \vee \dots \vee \neg x_k) \mid k \leq n\}$	IHSB $^-_n$
$IS_{10}$	$\{(x), (\neg x \vee y)\} \cup \{(\neg x_1 \vee \dots \vee \neg x_k) \mid k \in \mathbb{N}\}$	IHSB $^-$

Boehler, Reith, Schnoor, Vollmer, 2005. Bases of minimal order  
 Creignou, Kolaitis, Zanuttini, 2005. Plain bases minimal w.r.t  
 inclusion

$\Rightarrow$  There is no finite constraint language  $L$  such that  
 $\text{Pol}(L) = S_{10}$  or  $S_{00}$ .

# Bases and plain bases

co-clone	(plain) basis	property
$IS_{00}^n$	$\{(\neg x), (\neg x \vee y)\} \cup \{(x_1 \vee \dots \vee x_k) \mid k \leq n\}$	IHSB $^+_n$
$IS_{00}$	$\{(\neg x), (\neg x \vee y)\} \cup \{(x_1 \vee \dots \vee x_k) \mid k \in \mathbb{N}\}$	IHSB $^+$
$IS_{10}^n$	$\{(x), (\neg x \vee y)\} \cup \{(\neg x_1 \vee \dots \vee \neg x_k) \mid k \leq n\}$	IHSB $^-_n$
$IS_{10}$	$\{(x), (\neg x \vee y)\} \cup \{(\neg x_1 \vee \dots \vee \neg x_k) \mid k \in \mathbb{N}\}$	IHSB $^-$

Boehler, Reith, Schnoor, Vollmer, 2005. Bases of minimal order  
 Creignou, Kolaitis, Zanuttini, 2005. Plain bases minimal w.r.t  
 inclusion

$\Rightarrow$  There is no finite constraint language  $L$  such that  
 $\text{Pol}(L) = S_{10}$  or  $S_{00}$ .

# When basis and plain basis differ

$ C $	basis	plain basis
$IE$	$(\bar{x} \vee \bar{y} \vee z)$	$\{(\bar{x}_1 \vee \dots \vee \bar{x}_k \vee y) \mid k \geq 1\}$
$IE_0$	$\{(\bar{x} \vee \bar{y} \vee z), (\bar{x})\}$	$\{N_k \mid k \geq 0\} \cup \{(\bar{x}_1 \vee \dots \vee \bar{x}_k \vee y) \mid k \geq 1\}$
$IE_1$	$\{(\bar{x} \vee \bar{y} \vee z), (x)\}$	$\{(\bar{x}_1 \vee \dots \vee \bar{x}_k \vee y) \mid k \geq 0\}$
$IE_2$	$\{(\bar{x} \vee \bar{y} \vee z), (x), (\bar{x})\}$	$\{N_k \mid k \geq 0\} \cup \{(\bar{x}_1 \vee \dots \vee \bar{x}_k \vee y) \mid k \geq 0\}$

## Advantages / Disadvantages

- infinite plain bases are necessary for co-clones  $IL_{(c)}$ ,  $IV_{(c)}$ ,  $IE_{(c)}$ ,  $IN_{(c)}$  and  $II_{(c)}$ , while they have finite (classical) bases
- for each co-clone there is a canonical plain basis

# Minimal plain bases

## Proposition

Every co-clone has a unique plain basis which is minimal with respect to inclusion.

one can identify a preferred representation of a relation in a co-clone with respect to this minimal plain basis

## Minimal plain bases

### Proposition

Every co-clone has a unique plain basis which is minimal with respect to inclusion.

one can identify a preferred representation of a relation in a co-clone with respect to this minimal plain basis

# Preferred representation

## Definition (preferred representation)

Let  $ICl$  be a co-clone and let  $R$  be a relation in  $ICl$ . Then a conjunction of constraints  $\varphi$  is called a *preferred representation for  $R$  with respect to  $ICl$*  if  $\varphi$  represents  $R$  and every constraint in  $\varphi$  is built upon a relation out of the (unique) minimal plain basis of  $ICl$ .

# Preferred representation and prime CNF

## Proposition

Given a relation  $R$  of arity  $n$  and containing  $m$  vectors and a co-clone  $ICl$  to which  $R$  belongs, a preferred representation of  $R$  with respect to  $ICl$  can be found in time  $O(m^2 n^2)$ .

**Proof :** Compute a prime CNF  $\varphi$  representing  $R$ .

$\varphi$  contains  $O(mn)$  clauses and can be computed in time  $O(m^2 n^2)$  (Zanuttini, Hébrard 2002).

# Preferred representation and prime CNF

## Proposition

Given a relation  $R$  of arity  $n$  and containing  $m$  vectors and a co-clone  $ICl$  to which  $R$  belongs, a preferred representation of  $R$  with respect to  $ICl$  can be found in time  $O(m^2 n^2)$ .

**Proof :** Compute a prime CNF  $\varphi$  representing  $R$ .

$\varphi$  contains  $O(mn)$  clauses and can be computed in time  $O(m^2 n^2)$  (Zanuttini, Hébrard 2002).

# Computing the minimal co-clone

## Proposition

Given a relation  $R$  of arity  $n$  and containing  $m$  vectors, the minimal co-clone containing  $R$  can be found in time  $O(m^2 n^2)$ .

**Proof :** Compute a prime CNF representing  $R$  and use the list of plain bases.

# Computing the minimal co-clone

## Proposition

Given a relation  $R$  of arity  $n$  and containing  $m$  vectors, the minimal co-clone containing  $R$  can be found in time  $O(m^2 n^2)$ .

**Proof :** Compute a prime CNF representing  $R$  and use the list of plain bases.

# Expressibility

**Input :** Given a finite set of relations  $S$  and a relation  $R$

**Question :** is  $R$  expressible by a conjunctive query over  $S$  (i.e., whether  $R$  is in the minimal co-clone containing every relation in  $S$ ) ?

## Proposition

The expressibility problem can be solved in polynomial time.

**Proof :** Compute  $C_S$  ( $C_R$ ) the minimal co-clone containing all the relations in  $S$  (resp.  $R$ ). the relation  $R$  is expressible by  $S$  if and only if  $C_R \subseteq C_S$ .

# Expressibility

**Input :** Given a finite set of relations  $S$  and a relation  $R$

**Question :** is  $R$  expressible by a conjunctive query over  $S$  (i.e., whether  $R$  is in the minimal co-clone containing every relation in  $S$ ) ?

## Proposition

The expressibility problem can be solved in polynomial time.

**Proof :** Compute  $C_S$  ( $C_R$ ) the minimal co-clone containing all the relations in  $S$  (resp.  $R$ ). the relation  $R$  is expressible by  $S$  if and only if  $C_R \subseteq C_S$ .

## Open question 4

### INVERSE SATISFIABILITY

**Input :** Given a finite set of relations  $S$  and a relation  $R$

**Question :** is  $R$  expressible by a CNF( $S$ )-formula ? (with no existential variables, i.e., no projection)

A dichotomy theorem was obtained by Kavvadias and Sideri for the complexity of problem with constants. Does a dichotomy hold without the constants ? Are the Schaefer cases still tractable ?

## Open question 4

### INVERSE SATISFIABILITY

**Input :** Given a finite set of relations  $S$  and a relation  $R$

**Question :** is  $R$  expressible by a CNF( $S$ )-formula ? (with no existential variables, i.e., no projection)





A dichotomy theorem was obtained by Kavvadias and Sideri for the complexity of problem with constants. Does a dichotomy hold without the constants ? Are the Schaefer cases still tractable ?

## Conclusion

- There is still something to learn from the Boolean case
  - from complexity to the lattice of (co)-clones
  - from the lattice to complexity
- There remains some open questions

# Outline

- 1 Why are Boolean CSP interesting ?
  - Preliminaries
  - When does the Galois connection apply to complexity ?
- 2 Complexity of Abduction
  - Definition of the problem
  - Known results
  - New tractable cases
  - A complete classification
- 3 Bases for Boolean co-clones
  - Bases and plain bases for Boolean co-clones
  - The infinite part of Post's lattice
  - Preferred representations
  - Algorithmic applications
- 4 Non-exhaustive list of references (only those mentioned in the talk)

-  Böhler, E., Creignou, N., Reith, S. and Vollmer, H.  
Playing with Boolean Blocks, Part II : Post's Lattice with  
Applications to Complexity Theory.  
*ACM-SIGACT News* 35(1), Complexity Theory Column 43,  
pages 22–35, 2004.
-  Böhler, E., Reith S., Schnoor, H. and Vollmer, H. .  
Bases for Boolean co-clones.  
*Information Processing Letters*, 96 : 59–66, 2005.
-  Bulatov, A. and Dalmau, V.  
Towards a dichotomy theorem for the counting constraints  
satisfaction problem.  
In *Proc. FOC'2003*, pages 562-. 2003.
-  Creignou, N.

A dichotomy theorem for maximum generalized satisfiability.

*Journal of Computer and System Sciences*,  
51(3) :511-522, 1995.



Creignou, N. and Hermann, M.

Complexity of generalized satisfiability counting problems.

*Information and Computation*, 125(1) :1-12, 1996.



Creignou, N. and Hébrard, J.-J.

On generating all satisfying truth assignments of a generalized CNF-formula.

*Theoretical Informatics and Applications* 31(6) : 499-511,  
1997.



Creignou, N., Khanna, S. and Sudan, M.

*Complexity classifications of Boolean constraint satisfaction problems.*

SIAM Monographs on discrete mathematics and applications, 2001.



Creignou, N., Kolaitis, P. and Zanuttini, B.

Preferred representations of Boolean relations.

*Electronic Colloquium on Computational Complexity (ECCC)*, Report TR05-119, 2005.



Creignou, N. and Zanuttini, B.

A complete classification of the complexity of propositional abduction.

To appear in *SIAM Journal on computing*, 2006.



Eiter, T. and Gottlob, G.

The complexity of logic-based abduction.

*Journal of the ACM*, 42(1) :3–42, 1995.



Jeavons, P., Cohen, D. and Gyssens, M.

Closure properties of constraints.

*Journal of the ACM*, 44 :527–548, 1997.



D. Kavvadias and M. Sideri.

The inverse satisfiability problem.

*SIAM Journal on Computing*, 28(1) :152–163, 1998.



Kirousis, L. and Kolaitis, Ph.

A dichotomy in the complexity of propositional circumscription.

*Theory of Computing Systems*, 37(6) :695–716, 2004.



Nordh, G. and Zanuttini, B.

Propositional abduction is almost always hard.

In *Proc. 19th International Joint Conference on Artificial Intelligence (IJCAI'05)*, pages 534–539. IJCAI ,2005.



Post, E.L.

The two-valued iterative systems of mathematical logic.  
*Annals of Mathematical Studies*, 5 :1–122, 1941.



Schaefer, T.

The complexity of satisfiability problems.

In *Proc. 10th STOC, San Diego (CA, USA)*, pages  
216–226. Association for Computing Machinery, 1978.



Schnoor, H. and Schnoor, I.

Enumerating all Solutions for Constraint Satisfaction  
Problems.

Submitted for publication, 2006.