# Complexity of Clausal Constraints Over Chains

**Miki Hermann**[1]

joint work with

**Nadia Creignou**[2] **Andrei Krokhin**[3] **Gernot Salzer**[4]

[1]LIX, École Polytechnique, France

[2]Université de la Méditerranée, Marseille, France

[3]University of Durham, Great Britain

[4]Technical University of Vienna, Austria

Schloß Dagstuhl, October 2006

# Outline

**Purpose of our work:**

1. use the generalisation of propositional logic to regular multi-valued logic,
2. where the domain is totally ordered, i.e., it forms a chain,
3. to define a new concept of patterns,
4. which allows us to use a new kind of constraint language,
5. for which we have constraint satisfaction problems,
6. that have a nice complexity characterization,
7. for every domain cardinality.

# Propositional Logic

**Definitions**

A Boolean domain is a set $D = \{0, 1\}$ with implicit ordering $0 < 1$.

A literal is a variable $x$ or its negation $\neg x$.

# Propositional Logic

**Definitions**

A Boolean domain is a set $D = \{0, 1\}$ with implicit ordering $0 < 1$.

A literal is a variable $x$ or its negation $\neg x$.

**Alternative Definition**

A literal is an expression $x \geq 1$ or $x \leq 0$.

# Regular Multi-Valued Logic

### Definitions

A finite domain is a set $D = \{0, \ldots, n-1\}$ with an explicit ordering $0 < \cdots < n-1$.

Literals are expressions $x \geq d$ (positive) and $x \leq d$ (negative) for some value $d \in D$.

An interval is $[a, b] = \{x \in D \mid a \leq x \leq b\}$.

# Regular Multi-Valued Logic

## Definitions

A finite domain is a set $D = \{0, \ldots, n-1\}$ with an explicit ordering $0 < \cdots < n-1$.

Literals are expressions $x \geq d$ (positive) and $x \leq d$ (negative) for some value $d \in D$.

An interval is $[a, b] = \{x \in D \mid a \leq x \leq b\}$.

## Example

A logic with ordered values

$$\text{DON'T\_KNOW} < \text{NO} < \text{YES} < \text{DON'T\_CARE}$$

represents $D = \{0, 1, 2, 3\}$.

# Syntax

## Basics

**Clause:** disjunction of literals

**CNF formula:** conjunction of clauses

**Relation:** $R \subseteq D^k$ for some arity $k$ on a domain $D$

**Constraint:** $R(x_1, \ldots, x_k)$ for a relation $R$ of arity $k$

**Satisfaction:** An assignment $I\colon V \to D$ satisfies the constraint $R(x_1, \ldots, x_k)$ if $(I(x_1), \ldots, I(x_k)) \in R$.

# Syntax

## Definition

A **pattern** is an abstraction of a clause away from variables

$$(x_1 \geq a_1 \vee \cdots \vee x_p \geq a_p \vee x_{p+1} \leq b_1 \vee \cdots \vee x_{p+q} \leq b_q)$$
$$\downarrow$$
$$(+a_1, \ldots, +a_p, -b_1, \ldots, -b_q)$$

# Syntax

**Definition**

A **pattern** is an abstraction of a clause away from variables

$$(x_1 \geq a_1 \vee \cdots \vee x_p \geq a_p \vee x_{p+1} \leq b_1 \vee \cdots \vee x_{p+q} \leq b_q)$$
$$\downarrow$$
$$(+a_1, \ldots, +a_p, -b_1, \ldots, -b_q)$$

**Clausal constraint**

$P(x_1, \ldots, x_k)$ for a pattern $P$ of length $|P| = k$
If $P = (+a_1, \ldots, +a_p, -b_1, \ldots, -b_q)$ then
$P(x_1, \ldots, x_k) = (x_1 \geq a_1 \vee \cdots \vee x_p \geq a_p \vee x_{p+1} \leq b_1 \vee \cdots \vee x_{p+q} \leq b_q)$.

# Syntax

## Definition

A **pattern** is an abstraction of a clause away from variables

$$(x_1 \geq a_1 \vee \cdots \vee x_p \geq a_p \vee x_{p+1} \leq b_1 \vee \cdots \vee x_{p+q} \leq b_q)$$
$$\downarrow$$
$$(+a_1, \ldots, +a_p, -b_1, \ldots, -b_q)$$

## Clausal constraint

$P(x_1, \ldots, x_k)$ for a pattern $P$ of length $|P| = k$
If $P = (+a_1, \ldots, +a_p, -b_1, \ldots, -b_q)$ then
$P(x_1, \ldots, x_k) = (x_1 \geq a_1 \vee \cdots \vee x_p \geq a_p \vee x_{p+1} \leq b_1 \vee \cdots \vee x_{p+q} \leq b_q)$.

## Example

Let $P = (+1, -0, -0)$ be a pattern on the Boolean domain $D = \{0, 1\}$.
Then the clausal constraint $P(x, y, z) = (x \geq 1 \vee y \leq 0 \vee z \leq 0)$ is logically equivalent to the clause $(x \vee \neg y \vee \neg z)$.

# Syntax

## Definitions

A clausal language is a finite set $L = \{P_1, \ldots, P_m\}$ of patterns.

An $L$-formula is a finite conjunction of clausal constraints constructed from the clausal language $L$.

# Syntax

## Definitions

A clausal language is a finite set $L = \{P_1, \ldots, P_m\}$ of patterns.

An $L$-formula is a finite conjunction of clausal constraints constructed from the clausal language $L$.

## Example

Let $L = \{P_1, P_2\}$ be a clausal language consisting of two patterns $P_1 = (-0, -0)$ and $P_2 = (+1, +1)$. The $L$-formula

$$\begin{aligned} \varphi(x, y) &= P_1(x, y) \wedge P_2(x, y) \\ &= (x \leq 0 \vee y \leq 0) \wedge (x \geq 1 \vee y \geq 1) \end{aligned}$$

is logically equivalent to $(\neg x \vee \neg y) \wedge (x \vee y) = (x \not\equiv y)$.

# Constraint Satisfaction Problems

## Relational Constraint Satisfaction Problems

Parameter: a finite set $S$ of relations over a *finite* domain $D$

**Problem**: CSP($S$)
*Input*: An $S$-formula $\varphi$ over a domain $D$.
*Question*: Is $\varphi$ satisfiable?

# Constraint Satisfaction Problems

## Relational Constraint Satisfaction Problems

Parameter: a finite set $S$ of relations over a *finite* domain $D$

**Problem**: CSP($S$)
*Input*: An $S$-formula $\varphi$ over a domain $D$.
*Question*: Is $\varphi$ satisfiable?

## Boolean Constraint Satisfaction Problems

Parameter: a finite set $S$ of Boolean relations

**Problem**: CSP($S$)
*Input*: An $S$-formula $\varphi$.
*Question*: Is $\varphi$ satisfiable?

# Constraint Satisfaction Problems

**Clausal Constraint Satisfaction Problems**

Parameter: a finite set $L$ of patterns

**Problem**: CSP$(L)$
*Input*: An $L$-formula $\varphi$ over a finite totally ordered domain $D$.
*Question*: Is $\varphi$ satisfiable?

# Complexity Classification for Boolean CSP

**Theorem (Schaefer's Dichotomy Theorem, 1978)**

*Let $S$ be a finite set of Boolean relations. If*

- *$S$ is Horn,*
- *$S$ is dual Horn,*
- *$S$ is bijunctive,*
- *$S$ is affine,*
- *$S$ is 0- or 1-valid*

*then $\text{CSP}(S)$ is decidable in polynomial time, otherwise it is $\mathrm{NP}$-complete.*

# Complexity Classification for Relational CSP

**Conjecture (Feder and Vardi 1993-98)**

There exists a dichotomy theorem for the CSP problems for every finite domain $D$.

# Complexity Classification for Relational CSP

**Conjecture (Feder and Vardi 1993-98)**

There exists a dichotomy theorem for the CSP problems for every finite domain $D$.

**Theorem (Bulatov 2002)**

CSP *is dichotomic for* $D = \{0, 1, 2\}$.

**Classification of Clausal CSP**

*The goal of this talk*

# SU-Closed Clausal Languages

**Definition**

A clausal language $L$ is called **SU-closed**, if it is closed under taking subpatterns and it contains all non-trivial unary patterns, i.e., if it satisfies the following conditions:

1. If $P \in L$ and $P'$ is a subpattern of $P$ then $P' \in L$.
2. $(+d) \in L$ for all $d \in \{1, \ldots, n-1\}$.
3. $(-d) \in L$ for all $d \in \{0, \ldots, n-2\}$.

# SU-Closed Clausal Languages

## Definition

A clausal language $L$ is called **SU-closed**, if it is closed under taking subpatterns and it contains all non-trivial unary patterns, i.e., if it satisfies the following conditions:

1. If $P \in L$ and $P'$ is a subpattern of $P$ then $P' \in L$.
2. $(+d) \in L$ for all $d \in \{1, \ldots, n-1\}$.
3. $(-d) \in L$ for all $d \in \{0, \ldots, n-2\}$.

## Trivial patterns

Trivial unary patterns are $+0$ and $-(n-1)$ since $x \geq 0$ and $x \leq n-1$ are always satisfied, hence their correspond to TRUE.

# SU-Closed Clausal Languages

**Example**

The clausal language

$$L = \{(-0),\ (-1),\ (+1),\ (+2),\ (+1,-0),\ (+1,+1),\ (-1,-1),\ (+2,+2),\ (+1,+1,+1)\}$$

over the domain $D = \{0, 1, 2\}$ is SU-closed.

# Complexity of SU-Closed CSP

**Proposition (Language Equivalence)**

*For any $L$ there exists an SU-closed clausal language $L'$, such that*

1. $\texttt{CSP}(L)$ *and* $\texttt{CSP}(L')$ *are polynomial-time equivalent,*
2. $L'$ *can be constructed from* $L$ *in polynomial time*

# Complexity of SU-Closed CSP

**Proposition (Language Equivalence)**

*For any $L$ there exists an SU-closed clausal language $L'$, such that*

1. $\texttt{CSP}(L)$ *and* $\texttt{CSP}(L')$ *are polynomial-time equivalent,*
2. $L'$ *can be constructed from* $L$ *in polynomial time*

**Consequence**

We consider only SU-closed clausal languages $L$ in the sequel.

# Some More Notation

## Definitions

For a pattern $P = (+a_1, \ldots, +a_p, -b_1, \ldots, -b_q)$

- $P^+ = (+a_1, \ldots, +a_p)$ is the positive part
- $P^- = (-b_1, \ldots, -b_q)$ is the negative part
- $P$ is Horn, dual Horn, bijunctive, or binary if the corresponding clause $P(x_1, \ldots, x_k)$ has this property.
- $P$ is $d$-valid for some $d \in D$ if $(d, \ldots, d) \in [P(x_1, \ldots, x_k)]$

# Some More Notation

## Definitions

For a pattern $P = (+a_1, \ldots, +a_p, -b_1, \ldots, -b_q)$

- $P^+ = (+a_1, \ldots, +a_p)$ is the positive part
- $P^- = (-b_1, \ldots, -b_q)$ is the negative part
- $P$ is Horn, dual Horn, bijunctive, or binary if the corresponding clause $P(x_1, \ldots, x_k)$ has this property.
- $P$ is $d$-valid for some $d \in D$ if $(d, \ldots, d) \in [P(x_1, \ldots, x_k)]$

## Definition

A clausal language $L = \{P_1, \ldots, P_k\}$ is Horn, dual Horn, bijunctive, or $d$-valid if every pattern $P_i$ is Horn, dual Horn, bijunctive, or $d$-valid, respectively.

# Known Results About Clausal CSP

**Proposition (Hähnle *et al.*; Cohen, Gyssens, Jeavons)**

$\texttt{CSP}(L)$ *for a Horn, dual Horn, or bijunctive clausal language $L$ is decidable in polynomial time.*

# Known Results About Clausal CSP

**Proposition (Hähnle *et al.*; Cohen, Gyssens, Jeavons)**

$\texttt{CSP}(L)$ *for a Horn, dual Horn, or bijunctive clausal language* $L$ *is decidable in polynomial time.*

**Remark**

With very high probability, Hähnle-Cohen-Gyssens-Jeavons Proposition does not cover all polynomial cases.

# Resolution

## Definitions

Let

$$P = (v_1, \ldots, v_p, +a) \qquad \text{and} \qquad Q = (-b, v'_1, \ldots, v'_q)$$

be two patterns with $b < a$. Then the pattern

$$R = (v_1, \ldots, v_p, v'_1, \ldots, v'_q)$$

is called a resolvent of $P$ and $Q$.

We say that $R$ is obtained from $P$ and $Q$ by resolution.

# 3-Saturation

## Definition

Let $L$ be an SU-closed clausal language. The **3-saturation** of $L$, denoted by $\widehat{L}$, contains all patterns that can be constructed inductively from $L$ by the following rules:

1. If $P \in L$ and $|P| \leq 3$, then $P \in \widehat{L}$ (introduction).
2. If $P$ and $Q$ are patterns in $\widehat{L}$ such that $|P| + |Q| \leq 5$, then all resolvents of $P$ and $Q$ are in $\widehat{L}$ (restricted resolution).

# 3-Saturation

## Definition

Let $L$ be an SU-closed clausal language. The **3-saturation** of $L$, denoted by $\widehat{L}$, contains all patterns that can be constructed inductively from $L$ by the following rules:

1. If $P \in L$ and $|P| \leq 3$, then $P \in \widehat{L}$ (introduction).
2. If $P$ and $Q$ are patterns in $\widehat{L}$ such that $|P| + |Q| \leq 5$, then all resolvents of $P$ and $Q$ are in $\widehat{L}$ (restricted resolution).

## Every 3-saturation $\widehat{L}$

- contains only patterns $P$ of length $|P| \leq 3$,
- is an SU-closed language,
- can be computed from $L$ in finite time.

# 3-Saturation

## Definition

Let $L$ be an SU-closed clausal language. The **3-saturation** of $L$, denoted by $\widehat{L}$, contains all patterns that can be constructed inductively from $L$ by the following rules:

1. If $P \in L$ and $|P| \leq 3$, then $P \in \widehat{L}$ (introduction).
2. If $P$ and $Q$ are patterns in $\widehat{L}$ such that $|P| + |Q| \leq 5$, then all resolvents of $P$ and $Q$ are in $\widehat{L}$ (restricted resolution).

## Every 3-saturation $\widehat{L}$

- contains only patterns $P$ of length $|P| \leq 3$,
- is an SU-closed language,
- can be computed from $L$ in finite time.

## Proposition

*CSP($\widehat{L}$) is reducible to CSP($L$) in polynomial time.*

# PN-Pair

## Notation

For a pattern $P = (+a_1, \ldots, +a_p, -b_1, \ldots, -b_q)$

- $\min(P^+) = \min\{a_1, \ldots, a_p\}$
- $\max(P^-) = \max\{b_1, \ldots, b_q\}$

be the minimum and maximum values of $P^+$ and $P^-$, respectively.

# PN-Pair

## Notation

For a pattern $P = (+a_1, \ldots, +a_p, -b_1, \ldots, -b_q)$

- $\min(P^+) = \min\{a_1, \ldots, a_p\}$
- $\max(P^-) = \max\{b_1, \ldots, b_q\}$

be the minimum and maximum values of $P^+$ and $P^-$, respectively.

## Definitions

Two patterns $P$ and $N$ form a **PN-pair** if

1. $P$ is positive ($P = P^+$),
2. $N$ is negative ($N = N^-$),
3. one of them has length 2,
4. the other has length 3.

A PN-pair $(P, N)$ is called **disjoint** when $\max(N) < \min(P)$ holds.

# What Does an PN-Pair Encode?

## Encoding

Each disjoint PN-pair encodes the Boolean $\mathrm{CSP}(or_3^+, or_2^-)$, where

$$\begin{array}{lcl cl l} or_3^+ & = & \{0,1\}^3 \setminus \{000\} & = & [x \vee y \vee z] & \text{(by } P\text{)} \\ or_2^- & = & \{0,1\}^2 \setminus \{11\} & = & [\neg x \vee \neg y] & \text{(by } N\text{)} \end{array}$$

# NP-Complete Case

**Proposition (NP-completeness)**

*If the 3-saturation $\widehat{L}$ contains a disjoint PN-pair of patterns then* $\texttt{CSP}(L)$ *is* NP*-complete.*

# $\mathbf{NP}$-Complete Case

**Proposition ($\mathrm{NP}$-completeness)**

*If the 3-saturation $\widehat{L}$ contains a disjoint PN-pair of patterns then $\mathtt{CSP}(L)$ is $\mathrm{NP}$-complete.*

**Proof.**

$\mathtt{CSP}(or_3^+, or_2^-)$ is $\mathrm{NP}$-complete by Schaefer's Theorem. □

# Polynomial Cases

Case analysis for the polynomial cases

### Case 1

All positive (negative) patterns in $\widehat{L}$ have length 1.

# Polynomial Cases

Case analysis for the polynomial cases

## Case 1

All positive (negative) patterns in $\widehat{L}$ have length 1.

## Proposition

*If all positive (negative) patterns in $\widehat{L}$ have length 1 then both $L$ and $\widehat{L}$ are Horn (dual Horn) and $\mathrm{CSP}(L)$ is decidable in polynomial time.*

# Polynomial Cases

### Case 2

$\widehat{L}$ contains at least one positive and one negative pattern of length greater or equal to 2.

# Polynomial Cases

**Case 2**

$\widehat{L}$ contains at least one positive and one negative pattern of length greater or equal to 2.

Case 2 splits into two sub-cases.

**Case 2.1**

All patterns in $\widehat{L}$ have length smaller or equal to 2.

# Polynomial Cases

**Case 2**

$\widehat{L}$ contains at least one positive and one negative pattern of length greater or equal to 2.

Case 2 splits into two sub-cases.

**Case 2.1**

All patterns in $\widehat{L}$ have length smaller or equal to 2.

**Proposition**

*If all patterns in $\widehat{L}$ have length smaller or equal to 2 then $L$ is bijunctive and $\mathrm{CSP}(L)$ is decidable in polynomial time.*

# Polynomial Cases

**Case 2**

$\widehat{L}$ contains at least one positive and one negative pattern of length greater or equal to 2.

Case 2 splits into two sub-cases.

**Case 2.1**

All patterns in $\widehat{L}$ have length smaller or equal to 2.

**Proposition**

*If all patterns in $\widehat{L}$ have length smaller or equal to 2 then $L$ is bijunctive and $\text{CSP}(L)$ is decidable in polynomial time.*

**Case 2.2**

There exists at least one pattern $M \in \widehat{L}$ of length $|M| = 3$.

# Markers

### Definition

Let $L$ be a clausal language, such that $\widehat{L}$ contains a positive binary and a negative binary pattern. We call the values

$$
\begin{aligned}
p_{\max} &= \max\{\min(P) \mid P \in \widehat{L}, P = P^{+}, |P| = 2\} \quad \text{and} \\
q_{\min} &= \min\{\max(N) \mid N \in \widehat{L}, N = N^{-}, |N| = 2\}
\end{aligned}
$$

the **markers** of the binary positive and negative patterns in the saturation $\widehat{L}$, respectively.

# Polynomial Cases

Case 2.2 splits into two subcases

**Case 2.2.1**

$p_{\max} \leq q_{\min}$

# Polynomial Cases

Case 2.2 splits into two subcases

**Case 2.2.1**

$p_{\max} \leq q_{\min}$

**Proposition ($p_{\max} \leq q_{\min}$)**

*If the markers satisfy $p_{\max} \leq q_{\min}$, then $\mathtt{CSP}(L)$ is decidable in polynomial time.*

# Polynomial Cases

Case 2.2 splits into two subcases

**Case 2.2.1**

$p_{\max} \leq q_{\min}$

**Proposition ($p_{\max} \leq q_{\min}$)**

*If the markers satisfy $p_{\max} \leq q_{\min}$, then $\mathtt{CSP}(L)$ is decidable in polynomial time.*

**Proof.**

(Hint) Every $L$-formula $\varphi$ is $d$-valid for every $d \in [p_{\max}, q_{\min}]$. □

# Satisfiability on an Interval

**Recall**

An interval from $a$ to $b$ is the set $[a, b] = \{x \in D \mid a \leq x \leq b\}$

# Satisfiability on an Interval

**Recall**

An interval from $a$ to $b$ is the set $[a, b] = \{x \in D \mid a \leq x \leq b\}$

**Definition**

A pattern $M$ of length $k$ is called

- **$[a, b]$-valid** if every $I \colon V \to [a, b]$ satisfies $M(x_1, \ldots, x_k)$,
- **$[a, b]$-satisfiable** if $M(x_1, \ldots, x_k)$ is satisfiable by an $I \colon V \to [a, b]$,
- **$[a, b]$-unsatisfiable** when no $I \colon V \to [a, b]$ satisfies $M(x_1, \ldots, x_k)$.

# Polynomial Cases

## Case 2.2.2

$q_{\min} < p_{\max}$

# Polynomial Cases

**Case 2.2.2**

$q_{\min} < p_{\max}$

We need two lemmas

**Lemma (Ternary Patterns)**

*If $\widehat{L}$ does not contain a disjoint PN-pair and the markers satisfy the condition $q_{\min} < p_{\max}$, then every pattern $M \in L$ of length $|M| \geq 3$ is $[q_{\min}, p_{\max}]$-valid.*

# Polynomial Cases

## Case 2.2.2

$q_{\min} < p_{\max}$

We need two lemmas

## Lemma (Ternary Patterns)

*If $\widehat{L}$ does not contain a disjoint PN-pair and the markers satisfy the condition $q_{\min} < p_{\max}$, then every pattern $M \in L$ of length $|M| \geq 3$ is $[q_{\min}, p_{\max}]$-valid.*

## Conclusion

If we restrict ourselves to the interval $[q_{\min}, p_{\max}]$ then the patterns of arity 3 and more can be neglected.

# Second Lemma

**Lemma (Binary Patterns)**

*If $\widehat{L}$ does not contain a disjoint PN-pair and the markers satisfy the condition $q_{\min} < p_{\max}$, then every binary pattern in $L$ is $[q_{\min}, p_{\max}]$-satisfiable.*

# Second Lemma

**Lemma (Binary Patterns)**

*If $\widehat{L}$ does not contain a disjoint PN-pair and the markers satisfy the condition $q_{\min} < p_{\max}$, then every binary pattern in $L$ is $[q_{\min}, p_{\max}]$-satisfiable.*

**Conclusion**

There always exists a solution to the binary restriction of an $L$-formula on the interval $[q_{\min}, p_{\max}]$.

# Polynomial Cases

**Proposition ($q_{\min} < p_{\max}$)**

*If $\widehat{L}$ does not contain a disjoint PN-pair and the markers satisfy the condition $q_{\min} < p_{\max}$, then $\mathtt{CSP}(L)$ is decidable in polynomial time.*

# Polynomial Cases

**Proposition ($q_{\min} < p_{\max}$)**

*If $\widehat{L}$ does not contain a disjoint PN-pair and the markers satisfy the condition $q_{\min} < p_{\max}$, then $\mathtt{CSP}(L)$ is decidable in polynomial time.*

**Proof.**

(Hint) The algorithm computing an assignment $I$ works as follows:

# Polynomial Cases

**Proposition ($q_{\min} < p_{\max}$)**

*If $\widehat{L}$ does not contain a disjoint PN-pair and the markers satisfy the condition $q_{\min} < p_{\max}$, then $\mathtt{CSP}(L)$ is decidable in polynomial time.*

**Proof.**

(Hint) The algorithm computing an assignment $I$ works as follows:

1. Solve the restriction $\varphi_2$ to binary clauses/patterns of the $L$-formula $\varphi$.

# Polynomial Cases

**Proposition ($q_{\min} < p_{\max}$)**

*If $\widehat{L}$ does not contain a disjoint PN-pair and the markers satisfy the condition $q_{\min} < p_{\max}$, then $\mathtt{CSP}(L)$ is decidable in polynomial time.*

**Proof.**

(Hint) The algorithm computing an assignment $I$ works as follows:

1. Solve the restriction $\varphi_2$ to binary clauses/patterns of the $L$-formula $\varphi$.
2. Move "a little bit" the solution (if necessary) to satisfy all unary clauses in $\varphi$: if a unit clause $(x \leq v)$ or $(x \geq v)$ in $\varphi$ is not satisfied, set $I(x) = v$.

# Polynomial Cases

**Proposition ($q_{\min} < p_{\max}$)**

*If $\widehat{L}$ does not contain a disjoint PN-pair and the markers satisfy the condition $q_{\min} < p_{\max}$, then $\mathrm{CSP}(L)$ is decidable in polynomial time.*

**Proof.**

(Hint) The algorithm computing an assignment $I$ works as follows:

1. Solve the restriction $\varphi_2$ to binary clauses/patterns of the $L$-formula $\varphi$.
2. Move "a little bit" the solution (if necessary) to satisfy all unary clauses in $\varphi$: if a unit clause $(x \leq v)$ or $(x \geq v)$ in $\varphi$ is not satisfied, set $I(x) = v$.
3. The computed solution satisfies the $L$-formula $\varphi$. □

# Dichotomy Theorem

**Theorem (Dichotomy of Clausal CSP)**

*Let $L$ be an SU-closed clausal language. If the saturation $\widehat{L}$ does not contain a disjoint PN-pair then $\mathrm{CSP}(L)$ is decidable in polynomial time, otherwise it is $\mathrm{NP}$-complete.*

# Dichotomy Theorem

**Theorem (Dichotomy of Clausal CSP)**

*Let $L$ be an SU-closed clausal language. If the saturation $\widehat{L}$ does not contain a disjoint PN-pair then $\text{CSP}(L)$ is decidable in polynomial time, otherwise it is $\text{NP}$-complete.*

**Remark**

To the best of our knowledge, the tractable cases identified by Proposition $q_{\min} < p_{\max}$ are new.

# Conclusion

**Good news**

- We have a Dichotomy Theorem for each domain cardinality for a certain non-trivial language.
- We completely characterized the complexity of regular multi-valued logic.

# Conclusion

## Good news

- We have a Dichotomy Theorem for each domain cardinality for a certain non-trivial language.
- We completely characterized the complexity of regular multi-valued logic.

## Bad news

The clausal CSP are coarser than the relational ones.

# Conclusion

## Good news

- We have a Dichotomy Theorem for each domain cardinality for a certain non-trivial language.
- We completely characterized the complexity of regular multi-valued logic.

## Bad news

The clausal CSP are coarser than the relational ones.

## Open questions

- Does the result generalize to partial orderings on the domain?
- What can be implied from our results to (non-clausal) relational CSP?

# Thank You For Your Attention

Questions, please?

**Availability of the paper**

1. My web page:
   `http://www.lix.polytechnique.fr/~hermann/`
   and click on `Publications` then `ResearchReports`
2. Send email to `hermann@lix.polytechnique.fr`
3. Accepted for publication in *Theory of Computing Systems*