Forbidden Patterns in Constraint Satisfaction Problems

Guillaume Escamocher, under the supervision of Martin Cooper and Pierre Régnier

Institut de Recherche en Informatique de Toulouse

12/11/2014
1 Introduction
   - Constraint Satisfaction Problem
   - Classical methods to look for tractable classes
   - A new approach

2 Flat patterns
   - Definition
   - Classical elimination operations
   - The reduction
   - Patterns on two constraints

3 Existential patterns
   - Definitions
   - Variable elimination
   - Tractable classes

4 Conclusion
Data:
- A set $V$ of $n$ variables $\{v_1, \ldots, v_n\}$.
- A set of $n$ domains $\{D_1, \ldots, D_n\}$. Each domain $D_i$ contains the set of possible values for $v_i$. $\langle v_i, a \rangle$ represents the assignment of the value $a \in D_i$ to the variable $v_i$. (assignment $\equiv$ point)
- A set $C$ of constraints $\{C_1, \ldots, C_m\}$. Each constraint $C_j$ is associated to a $k_j$-uple of variables $\langle v_{j_1}, \ldots, v_{j_{k_j}} \rangle$ and describes the $k_j$-uples of assignments allowed for this $k_j$-uple of variables. $k_j$ is the arity of $C_j$.

Question:
- Is there an assignment of the values to the $n$ variables such that all constraints in $C$ are satisfied?
Data:
- A set $V$ of $n$ variables $\{v_1, \ldots, v_n\}$.
- A set of $n$ domains $\{D_1, \ldots, D_n\}$. Each domain $D_i$ contains the set of possible values for $v_i$. $\langle v_i, a \rangle$ represents the assignment of the value $a \in D_i$ to the variable $v_i$. (assignment $\equiv$ point)
- A set $C$ of $n(n+1)/2$ binary constraints $\{C_1, \ldots, C_{n(n+1)/2}\}$. Each constraint $C_j$ is associated to a couple of variables $\langle v_{i_1}, v_{i_2} \rangle$ and describes the couples of assignments allowed for these 2 variables.

Question:
- Is there an assignment of the values to the $n$ variables such that all constraints in $C$ are satisfied?
A CSP instance

A set of 4 variables \{v_1, v_2, v_3, v_4\}

A set of 4 domains \{D_1, D_2, D_3, D_4\}

A set of 6 constraints \{C_1, C_2, C_3, C_4, C_5, C_6\}

Constraint Satisfaction Problem

Classical methods to look for tractable classes

A new approach

Introduction

Flat patterns

Existential patterns

Conclusion

Question: Is there a solution?

Two compatible points.

Two incompatible points.

A trivial constraint.

The constraint graph.
A CSP instance

A set of 4 domains \( \{D_1, D_2, D_3, D_4\} \)

Constraint Satisfaction Problem
Classical methods to look for tractable classes
A new approach
A CSP instance

A set of 4 variables \( \{v_1, v_2, v_3, v_4\} \)

A set of 4 domains \( \{D_1, D_2, D_3, D_4\} \)

A set of 6 constraints \( \{C_1, C_2, C_3, C_4, C_5, C_6\} \)

\[ C_1 = \{\langle v_1, a \rangle, \langle v_3, f \rangle\}, \langle v_1, b \rangle, \langle v_3, g \rangle\} \]

\[ D_1 = \{a, b\} \]

\[ D_2 = \{c, d, e\} \]

\[ D_3 = \{f, g\} \]

\[ D_4 = \{h, i, j\} \]

Question: Is there a solution?

Two compatible points.

Two incompatible points.

A trivial constraint.

The constraint graph.
A CSP instance

\[ D_2 = \{c, d, e\}, \quad D_3 = \{f, g\}, \quad D_4 = \{h, i, j\} \]
A set of 4 variables
\{v_1, v_2, v_3, v_4\}

A set of 4 domains
\{D_1, D_2, D_3, D_4\}

A set of 6 constraints
\{C_1, C_2, C_3, C_4, C_5, C_6\}

Question: Is there a solution?

Answer: Yes! (b, d, g, h)
A CSP instance

A set of 4 variables
\{v_1, v_2, v_3, v_4\}

A set of 4 domains
\{D_1, D_2, D_3, D_4\}

A set of 6 constraints
\{C_1, C_2, C_3, C_4, C_5, C_6\}

\begin{align*}
C_1 &= \{\langle v_1, a \rangle, \langle v_3, f \rangle, \langle v_1, b \rangle, \langle v_3, g \rangle\}
\end{align*}

Question: Is there a solution?

Two compatible points.

Two incompatible points.

A trivial constraint.

The constraint graph.

Answer: Yes! (b, d, g, h)
A CSP instance

\[ C_2 = \{ \langle \langle v_2, d \rangle, \langle v_4, h \rangle \rangle, \langle \langle v_2, d \rangle, \langle v_4, i \rangle \rangle, \langle \langle v_2, d \rangle, \langle v_4, j \rangle \rangle \} \]
A CSP instance

Question: Is there a solution?

A trivial constraint.

Two compatible points.

Two incompatible points.

Forbidden Patterns in Constraint Satisfaction Problems
A CSP instance

A set of 4 variables \{v_1, v_2, v_3, v_4\}

A set of 4 domains \{D_1, D_2, D_3, D_4\}

A set of 6 constraints \{C_1, C_2, C_3, C_4, C_5, C_6\}

\[ C_1 = \{\langle \langle v_1, a \rangle, \langle v_3, f \rangle \rangle, \langle \langle v_1, b \rangle, \langle v_3, g \rangle \rangle\} \]

\[ D_1 = \{a, b\} \]

\[ D_2 = \{c, d, e\} \]

\[ D_3 = \{f, g\} \]

\[ D_4 = \{h, i, j\} \]

Question: Is there a solution?

Answer: Yes! (b,d,g,h)
A CSP instance

A set of 4 variables \( \{v_1, v_2, v_3, v_4\} \)

A set of 4 domains \( \{D_1, D_2, D_3, D_4\} \)

A set of 6 constraints \( \{C_1, C_2, C_3, C_4, C_5, C_6\} \)

\[ C_1 = \{\langle \langle v_1, a \rangle, \langle v_3, f \rangle \rangle, \langle \langle v_1, b \rangle, \langle v_3, g \rangle \rangle\} \]

\[ D_1 = \{a, b\} \]

\[ D_2 = \{c, d, e\} \]

\[ D_3 = \{f, g\} \]

\[ D_4 = \{h, i, j\} \]

Question: Is there a solution?

Two compatible points.

Forbidden Patterns in Constraint Satisfaction Problems

Answer: Yes! \((b,d,g,h)\)
A CSP instance

A set of 4 variables
\{v_1, v_2, v_3, v_4\}

A set of 4 domains
\{D_1, D_2, D_3, D_4\}

A set of 6 constraints
\{C_1, C_2, C_3, C_4, C_5, C_6\}

\[ C_1 = \{\langle\langle v_1, a\rangle, \langle v_3, f\rangle\rangle, \langle\langle v_1, b\rangle, \langle v_3, g\rangle\rangle\} \]

\[ D_1 = \{a, b\} \]

\[ D_2 = \{c, d, e\} \]

\[ D_3 = \{f, g\} \]

\[ D_4 = \{h, i, j\} \]

\[ C_2 = \{\langle\langle v_2, d\rangle, \langle v_4, h\rangle\rangle, \langle\langle v_2, d\rangle, \langle v_4, i\rangle\rangle, \langle\langle v_2, d\rangle, \langle v_4, j\rangle\rangle\} \]

Question: Is there a solution?

Two compatible points.

Two incompatible points.

The constraint graph.

Answer: \textbf{Yes!} (b,d,g,h)
A CSP instance

A trivial constraint.
A CSP instance

The constraint graph.

A set of 4 variables \( \{v_1, v_2, v_3, v_4\} \)

A set of 4 domains \( \{D_1, D_2, D_3, D_4\} \)

A set of 6 constraints \( \{C_1, C_2, C_3, C_4, C_5, C_6\} \)

\[ C_1 = \{ \langle v_1, a \rangle, \langle v_3, f \rangle, \langle v_1, b \rangle, \langle v_3, g \rangle \} \]

\[ D_1 = \{a, b\} \]

\[ D_2 = \{c, d, e\} \]

\[ D_3 = \{f, g\} \]

\[ D_4 = \{h, i, j\} \]

\[ C_2 = \{ \langle v_2, d \rangle, \langle v_4, h \rangle, \langle v_2, d \rangle, \langle v_4, i \rangle, \langle v_2, d \rangle, \langle v_4, j \rangle \} \]
Why the CSP?

- The CSP is NP-Complete: it can represent many problems.
- It is an intuitive problem; reducing other NP-Complete problems to it can be done quickly.
  - Reduction from 3-SAT can be done in linear time.
- It is used in:
  - Planning.
  - Bioinformatics.
  - Computer vision.
  - and many other fields...
The search for tractable classes

**Tractable class**

The CSP is NP-Complete. A polynomial subset of the CSP is a **tractable class**.

**Examples**

1. The set of CSP instances where the size of the domains is smaller or equal than 2.
2. The set of CSP instances where the constraint graph is a tree.
Methods

Two main approaches are used:

- Restrictions on the constraints.
- Restrictions on the constraint graph.
Restrictions on the constraints

- Some constraints are forbidden.
- All constraint graphs are allowed.

**Example: ZOA**

$$\forall v, v' \ \forall a \in A_v, \ a \text{ can be compatible with :}$$

- Zero
- One
- or All
- points in $$A_{v'}$$.
Forbidden Patterns in Constraint Satisfaction Problems
Restrictions on the constraint graphs

- Some constraint graphs are forbidden.
- All constraints are allowed.

**Example: Bounded Tree Width**

The set of instances where the constraint graph has a tree width $\leq k$. 
Forbidden pattern

Pattern = Part of an instance

The pattern $T_4$. 
Forbidden pattern

We associate to a pattern the set of CSP instances in which the pattern does **NOT** appear.
Patterns can define classes we could not get with restrictions on only the constraints or only the graphs.
More refined classes

A lot of patterns (for instance $T_4$) define classes of instances where:

- All possible constraints are represented.
- All possible constraint graphs are represented.
The assets of forbidden patterns

1. Patterns can define classes we could not get with restrictions on only the constraints or only the graphs.

2. Patterns can also define many existing classes based on constraints or on graphs.
Binary Boolean CSP
Forbidden Patterns in Constraint Satisfaction Problems
Forbidden patterns are not perfect...

There is no forbidden pattern equivalent to the set of CSP instances whose constraint graph is a tree.
Forbidden patterns are not perfect...

There is no forbidden pattern equivalent to the set of CSP instances whose constraint graph is a tree.

...but we can improve them.

For example, BTP is a tractable class generalizing the class of CSP instances whose constraint graph is a tree.
There exists an ordering of the variables such that this pattern does not appear.
The assets of forbidden patterns

1. Patterns can define classes we could not get with restrictions on only the constraints or only the graphs.
2. Patterns can also define many existing classes based on constraints or on graphs.
3. Patterns are very malleable, their potential can be easily increased.
The assets of forbidden patterns

1. Patterns can define classes we could not get with restrictions on only the constraints or only the graphs.
2. Patterns can also define many existing classes based on constraints or on graphs.
3. Patterns are very malleable, their potential can be easily increased.
4. Patterns can be detected in polynomial time.
5. A pattern can still appear a limited number of times; one can just isolate the part of the instance where it does.
Introduction
- Constraint Satisfaction Problem
- Classical methods to look for tractable classes
- A new approach

Flat patterns
- Definition
- Classical elimination operations
- The reduction
- Patterns on two constraints

Existential patterns
- Definitions
- Variable elimination
- Tractable classes

Conclusion
A pattern \( \langle V, A, \text{var}, E, \text{cpt} \rangle \) is defined by:

- A set of variables \( V \).
- A set of points \( A \).
- A function \( \text{var} \) which associates a variable with each point.
- A set of edges \( E \).
- A function \( \text{cpt} \) which assigns a compatibility to each edge (either \( T \), or \( F \)).
Example

\[ V = \{ v_0, v_1, v_2 \} \]

\[ A_{v_1} \quad A_{v_2} \quad A_{v_0} \]
Example

Let $A = \{a, b, c, d, e\}$

$\text{var}(a) = v_1$, $\text{var}(b) = v_2$, $\text{var}(c) = \text{var}(d) = \text{var}(e) = v_0$
Example

\[
V = \{ v_0, v_1, v_2 \}
\]

\[
A = \{ a, b, c, d, e \}
\]

\[
\begin{align*}
\text{var}(a) &= v_1, \\
\text{var}(b) &= v_2, \\
\text{var}(c) &= \text{var}(d) = \text{var}(e) = v_0
\end{align*}
\]

\[
E = \{ (a, c), (a, d), (a, e), (b, c), (b, d) \}
\]

\[
cpt(a, c) = T, \quad \text{cpt}(a, d) = T, \quad \text{cpt}(a, e) = F, \quad \text{cpt}(b, c) = F, \quad \text{cpt}(b, d) = T
\]
A CSP instance is a pattern \( \langle V, A, var, E, cpt \rangle \) where \( E \) is the set of couples \((a, b)\) such that \( a \) and \( b \) belong to different domains.
Extension

\[ P \rightarrow P' \]
Merging

\[ P \rightarrow P' \]
Occurrence in a pattern

Definition
We say that a pattern $P$ occurs in a pattern $P'$ if $P'$ can be obtained from $P$ with extension and/or merging.

Definition
We say that a pattern $P$ occurs in an instance $I$ if $I$ can be obtained from $P$ with extension and/or merging.
We note \( \text{CSP}(\overline{P}) \) the set of CSP instances in which the pattern \( P \) does \textbf{NOT} occur.

\[
\begin{align*}
\text{CSP}(\overline{P}) & \text{ is polynomial} \quad \rightarrow \quad P \text{ is tractable.} \\
\text{CSP}(\overline{P}) & \text{ is NP-Complete} \quad \rightarrow \quad P \text{ is NP-Complete.}
\end{align*}
\]
Arc consistency

Forbidden Patterns in Constraint Satisfaction Problems
Arc consistency
Arc consistency

Forbidden Patterns in Constraint Satisfaction Problems
Single valued variable elimination

Forbidden Patterns in Constraint Satisfaction Problems
Single valued variable elimination

Forbidden Patterns in Constraint Satisfaction Problems
Neighborhood substitution

Forbidden Patterns in Constraint Satisfaction Problems
Neighborhood substitution

\[ \begin{align*}
A_{v_1} & \quad b \\
A_{v_3} & \quad f \\
A_{v_4} & \quad i, h \\
A_{v_4} & \quad g
\end{align*} \]
Classical operations properties

- These operations can be applied in polynomial time.
- The order in which we do them does not matter.

Combination with forbidden patterns

Removing a point or a variable cannot introduce a pattern \(\Rightarrow\) We can combine classical operations and forbidden patterns.
Dangling point
Definition
Classical elimination operations
The reduction
Patterns on two constraints

DP elimination

$P \rightarrow P'$
Let $P$ be a pattern and $P'$ the result of a DP elimination on $P$.

$P'$ occurs in an instance $I$

$\Rightarrow$ (arc consistency)

$P$ occurs in $I$
Let $P$ be a pattern and $P'$ the result of a DP elimination on $P$.

$P'$ occurs in an instance $I$

$\Rightarrow$ (arc consistency)

$P$ occurs in $I$

Remark

The same can be said of merging.
We say that a pattern $P$ can be reduced to a pattern $Q$, and that $Q$ is a reduction of $P$, if one of the following conditions is fulfilled:

1. $Q$ is the result of an extension applied to $P$.
2. $Q$ is the result of a merging applied to $P$.
3. $Q$ is the result of a DP elimination applied to $P$.
4. There is a recursive combination of the above conditions.
\( P \) can be reduced to \( P' \)

- \( P \) occurs in \( I \) \( \iff \) \( P' \) occurs in \( I \)
- \( P \) does not occur in \( I \) \( \implies \) \( P' \) does not occur in \( I \)
- \( I \in \text{CSP}(\overline{P}) \) \( \subseteq \) \( I \in \text{CSP}(\overline{P'}) \)
- \( \text{CSP}(\overline{P}) \) \( \subseteq \) \( \text{CSP}(\overline{P'}) \)
- \( P \) is tractable \( \iff \) \( P' \) is tractable
- \( P \) is NP-Complete \( \implies \) \( P' \) is NP-Complete
<table>
<thead>
<tr>
<th>$P$</th>
<th>$\rightarrow$</th>
<th>$P'$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P$ occurs in $I$</td>
<td>$\iff$</td>
<td>$P'$ occurs in $I$</td>
</tr>
<tr>
<td>$P$ does not occur in $I$</td>
<td>$\implies$</td>
<td>$P'$ does not occur in $I$</td>
</tr>
<tr>
<td>$I \in \text{CSP}(\overline{P})$</td>
<td>$\subseteq$</td>
<td>$I \in \text{CSP}(\overline{P'})$</td>
</tr>
<tr>
<td>$\text{CSP}(\overline{P})$</td>
<td>$\subseteq$</td>
<td>$\text{CSP}(\overline{P'})$</td>
</tr>
<tr>
<td>$P$ is tractable</td>
<td>$\iff$</td>
<td>$P'$ is tractable</td>
</tr>
<tr>
<td>$P$ is NP-Complete</td>
<td>$\implies$</td>
<td>$P'$ is NP-Complete</td>
</tr>
</tbody>
</table>

**Usefulness of the reduction**

Small number of results $\rightarrow$ Large number of complexity classes

To establish a dichotomy, we just need to find the patterns located at the “border” between tractability and NP-Completeness.
Let $P$ be a non mergeable flat pattern on two constraints. $P$ is tractable if and only if $P$ can be extended to one of the patterns of $T$. 

$$T_1 \quad T_2 \quad T_3$$

$$T_4 \quad T_5 \quad 2I$$
Proof

$\Leftarrow$

1. We give 6 tractability proofs, one for each pattern in $T$.
2. From the properties of the reduction, we have the result.

$\Rightarrow$

1. We give a set $G$ of gadgets.
2. We show that all gadgets in $G$ are NP-Complete.
3. We show that for every flat pattern on two constraints $P$, either $P$ contains a gadget from $G$, or $P$ can be extended to one of the patterns in $T$.
4. From the properties of the reduction, we have the result.
Examples of NP-Complete gadgets

Forbidden Patterns in Constraint Satisfaction Problems
Definition

Let $I = \langle V, A, \text{var}, E, \text{cpt} \rangle$ be a CSP instance satisfying arc consistency. If the satisfiability of $I$ is the same as the satisfiability of $I \setminus v$, we say that $v$ can be eliminated.
Definition

Let $I = \langle V, A, \text{var}, E, \text{cpt} \rangle$ be a CSP instance satisfying arc consistency. If the satisfiability of $I$ is the same as the satisfiability of $I \setminus v$, we say that $v$ can be \textit{eliminated}.

Question

How to detect such variables?
Introduction

- Constraint Satisfaction Problem
- Classical methods to look for tractable classes
- A new approach

Flat patterns

- Definition
- Classical elimination operations
- The reduction
- Patterns on two constraints

Existential patterns

- Definitions
- Variable elimination
- Tractable classes

Conclusion
Quantified patterns

A quantified pattern $\langle V, A, \text{var}, E, \text{cpt}, v \rangle$ is defined by:

- A set of variables $V$.
- A set of points $A$.
- A function $\text{var}$ which associates a variable with each point.
- A set of edges $E$.
- A function $\text{cpt}$ which assigns a compatibility to each edge (either $T$, or $F$).
- A distinguished variable $v \in V$. 

Forbidden Patterns in Constraint Satisfaction Problems
A quantified pattern $\langle V, A, \text{var}, E, \text{cpt}, v \rangle$. 
Occurrence on a variable

\[ \exists v \]

\[ \forall v \]

BTP

\[ A_{v_0} \]

\[ A_{v_1} \]

\[ A_{v_2} \]

An instance \( I \)
Occurrence on a variable

\[ \exists v \]

\[ \forall v \]

\[ A_v^0 \]

\[ A_v^1 \]

\[ A_v^2 \]

BTP

An instance \( I \)

- **BTP occurs on** \( v_0 \).
- **BTP does not occur** on \( v_1 \).
- **BTP does not occur** on \( v_2 \).
An existential pattern $\langle V, A, var, E, cpt, a \rangle$ is defined by:

- A set of variables $V$.
- A set of points $A$.
- A function $var$ which associates a variable with each point.
- A set of edges $E$.
- A function $cpt$ which assigns a compatibility to each edge (either $T$, or $F$).
- A distinguished point $a \in A$. 
An existential pattern $\langle V, A, \text{var}, E, \text{cpt}, a \rangle$. 
Occurrence on a variable

\[ \exists a \]

\[ X_1 \]

An instance \( I \)

\[ A_{v_1} \]

\[ A_{v_2} \]

\[ A_{v_0} \]

\[ a_0 \]

\[ b_0 \]

\[ c_0 \]
Occurrence on a variable

- $X_1$ occurs on $a_0$.

An instance $I$
Occurrence on a variable

- $X_1$ occurs on $a_0$.
- $X_1$ occurs on $b_0$. 
Occurrence on a variable

- $X_1$ occurs on $a_0$.
- $X_1$ occurs on $b_0$.
- $X_1$ does not occur on $c_0$. 
Occurrence on a variable

- $X_1$ occurs on $a_0$.  
- $X_1$ occurs on $b_0$.  
- $X_1$ does not occur on $c_0$.  

$\Rightarrow X_1$ does not occur on $v_0$.  

Forbidden Patterns in Constraint Satisfaction Problems
VE pattern

Definition

We say that a pattern $P$ is a **Variable Elimination pattern** (or **VE pattern**) if for every instance $I = \langle V, A, \text{var}, E, \text{cpt} \rangle$ satisfying arc consistency and for every variable $v \in V$ we have: $P$ does not occur on $v \Rightarrow v$ can be eliminated
The reduction for VE patterns

\[ P \quad \rightarrow \quad P' \]

- \( P \) occurs on \( v \) \( \iff \) \( P' \) occurs on \( v \)
- \( P \) does not occur on \( v \) \( \implies \) \( P' \) does not occur on \( v \)
- \( P \) is a VE pattern \( \iff \) \( P' \) is a VE pattern
- \( P \) is not a VE pattern \( \implies \) \( P' \) is not a VE pattern
Why the arc consistency requirement?

Because we can: arc consistency can be established in quadratic time.
Why the arc consistency requirement?

1. Because we can: arc consistency can be established in quadratic time.

2. Because otherwise VE patterns would not be interesting: any pattern containing $1C$ would not be a VE pattern.
In what order should we eliminate?

It does not matter:

- Eliminating a variable cannot create an occurrence of a pattern.
- Actually, eliminating a variable can remove occurrences of a pattern, and make other variables eligible for elimination!
Theorem

Let $P$ be a non mergeable pattern. $P$ is a VE pattern if and only if $P$ can be extended to one of the patterns in $B$. 

\[\exists v \quad \exists a \quad \exists v \quad \exists a \quad \exists a \quad \exists a\]

Forbidden Patterns in Constraint Satisfaction Problems
Proof

$\leftarrow$

1. We give 4 proofs to show that the 4 patterns in $B$ allow variable elimination.
2. From the properties of the reduction, we have the result.

$\Rightarrow$

1. We give a set $G$ of gadgets.
2. We show that none of the gadgets from $G$ allows variable elimination.
3. We show that for any non mergeable pattern $P$, either $P$ contains a gadget from $G$, or $P$ can be extended to one of the patterns in $B$.
4. From the properties of the reduction, we have the result.
Examples of gadgets which are not VE

Forbidden Patterns in Constraint Satisfaction Problems
VE patterns define tractable classes.

\[ P \text{ is a VE pattern} \implies P \text{ is a tractable pattern} \]

One can just eliminate all variables from the instance.
Occurrence in an instance

\[ \exists a \]

\[ X_1 \]

\[ A_{v_1} \]

An instance \( I \)

\[ A_{v_0} \]

\[ A_{v_2} \]

\[ a_2 \]

\[ b_2 \]
Occurrence in an instance

\[ \exists a \]

\[ X_1 \]

- \( X_1 \) occurs on \( a_2 \).

An instance \( I \)

\[ A_{v_1} \]

\[ A_{v_2} \]

\[ A_{v_0} \]
Occurrence in an instance

\[ \exists a \]

- \( X_1 \) occurs on \( a_2 \).
- \( X_1 \) occurs on \( b_2 \).
Occurrence in an instance

\( \exists a \)

\[ X_1 \]

- \( X_1 \) occurs on \( a_2 \).
- \( X_1 \) occurs on \( b_2 \).

\[ A_{v_1} \]
\[ A_{v_2} \]
\[ A_{v_0} \]

An instance \( I \)

\[ \Rightarrow X_1 \text{ occurs on } v_2. \]
Occurrence in an instance

\[ \exists a \]

\[ X_1 \text{ occurs on } a_2. \]
\[ X_1 \text{ occurs on } b_2. \]

An instance \( I \)

\[ \Rightarrow X_1 \text{ occurs on } v_2. \]
\[ \Rightarrow X_1 \text{ occurs in } I. \]
**Notation**

We note $\text{CSP}(\overline{P})$ the set of CSP instances in which the existential pattern $P$ does **NOT** occur.

$$\text{CSP}(\overline{P}) \text{ is polynomial } \rightarrow P \text{ is tractable.}$$

$$\text{CSP}(\overline{P}) \text{ is NP-Complete } \rightarrow P \text{ is NP-Complete.}$$
Theorem

Let $P$ be a non mergeable existential pattern on two constraints. $P$ is tractable if and only if $P$ can be extended to one of the patterns in $X$.
Dichotomy on two constraints

**Theorem**

Let $P$ be a non mergeable pattern on two constraints. $P$ is tractable if and only if $P$ is a subpattern of $T_1, T_2, T_3, T_4, X_1, X_2$ or $X_3$. 

Remark $T_5$ and $T_2$ are not mentioned because they are subpatterns of $X_2$ and $X_3$ respectively.
Dichotomy on two constraints

**Theorem**

Let $P$ be a non mergeable pattern on two constraints. $P$ is tractable if and only if $P$ is a subpattern of $T_1$, $T_2$, $T_3$, $T_4$, $X_1$, $X_2$ or $X_3$.

**Remark**

$T_5$ and $2I$ are not mentioned because they are subpatterns of $X_2$ and $X_3$ respectively.
1. Introduction
   - Constraint Satisfaction Problem
   - Classical methods to look for tractable classes
   - A new approach

2. Flat patterns
   - Definition
   - Classical elimination operations
   - The reduction
   - Patterns on two constraints

3. Existential patterns
   - Definitions
   - Variable elimination
   - Tractable classes

4. Conclusion
Some results from my PhD

- Formal definition of forbidden patterns.
  - Flat patterns.
  - Quantified patterns.
  - Existential patterns.
- New tool: the reduction.
- New tractable classes.
  - Tractable forbidden patterns on two constraints.
  - Several tractable forbidden patterns on three variables.
  - Tractable Max-CSP subproblems.
- Instance simplification operations.
  - Forbidden patterns allowing variable elimination.
  - Subdomain fusion.
Recent forbidden patterns related work

- Value elimination.
- Fusion of values.
- Topological minors: related to the notion of minor graph.
Thank you very much for your attention!