Characterising the Complexity of Constraint Satisfaction Problems Defined by 2-Constraint Forbidden Patterns

Martin C. Cooper, Guillaume Escamocher

To cite this version:

HAL Id: hal-01283864
https://hal.archives-ouvertes.fr/hal-01283864
Submitted on 9 Mar 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Open Archive TOULOUSE Archive Ouverte (OATAO)

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible.

This is an author-deposited version published in: http://oatao.univ-toulouse.fr/
Eprints ID: 15210

To link to this article: DOI: 10.1016/j.dam.2014.10.035
URL: http://dx.doi.org/10.1016/j.dam.2014.10.035

To cite this version: Cooper, Martin C. and Escamocher, Guillaume

Any correspondence concerning this service should be sent to the repository administrator: staff-oatao@listes-diff.inp-toulouse.fr
Characterising the complexity of constraint satisfaction problems defined by 2-constraint forbidden patterns✩

Martin C. Cooper\textsuperscript{a}, Guillaume Escamocher\textsuperscript{b,∗}

\textsuperscript{a} IRIT, University of Toulouse, France
\textsuperscript{b} INSIGHT, University College Cork, Ireland

\section*{Abstract}

Although the CSP (constraint satisfaction problem) is NP-complete, even in the case when all constraints are binary, certain classes of instances are tractable. We study classes of binary CSP instances defined by excluding subproblems. This approach has recently led to the discovery of novel tractable classes. The complete characterisation of all tractable classes defined by forbidding patterns (where a pattern is simply a compact representation of a set of subproblems) is a challenging problem. We demonstrate a dichotomy in the case of forbidden patterns consisting of either one or two constraints. This has allowed us to discover several new tractable classes including, for example, a novel generalisation of 2SAT. We then extend this dichotomy to existential patterns which are only forbidden on specific domain values.

\section{Introduction}

In this paper we study the generic combinatorial problem known as the binary constraint satisfaction problem (CSP) in which the aim is to determine the existence of an assignment of values to \( n \) variables such that a set of constraints on pairs of variables are simultaneously satisfied. The generic nature of the CSP has led to diverse applications, notably in the fields of Artificial Intelligence and Operations Research \cite{24}.

A fundamental research question in complexity theory is the identification of tractable subproblems of NP-complete problems. Classical approaches have consisted in identifying types of constraints which imply the existence of a polynomial-time algorithm. Among the most well-known examples, we can cite linear constraints and Horn clauses. In an orthogonal approach, restrictions are placed solely on the (hyper)graph of constraint scopes, known as the constraint (hyper)graph. In some cases, dichotomies have even been proved characterising all tractable classes definable by placing restrictions either on the constraint relations \cite{4,3,2} or on the constraint (hyper)graph \cite{21–23}.

Recently, a new avenue of research has been investigated: the identification of tractable classes of CSP instances defined by forbidding a specific (set of) subproblems. Novel tractable classes have been discovered by forbidding simple 3-variable subproblems \cite{13,15}. A dichotomy has even been discovered for classes of binary CSP instances defined by forbidding configurations of incompatibilities \cite{5}.

One concrete example of a tractable class defined by forbidding a generic subproblem (known as a pattern) is the set of binary CSP instances satisfying the broken-triangle property \cite{13}: a binary CSP instance on variables \( X_1, \ldots, X_n \) satisfies the

\footnotetext[1]{Supported by ANR Project ANR-10-BLAN-0210.}

\footnotetext[2]{Corresponding author.}

E-mail addresses: cooper@irit.fr (M.C. Cooper), guillaume.escamocher@insight-centre.org (G. Escamocher).
tractable patterns permit variable elimination. It was possible to characterise all patterns
of the CSP instance. Note that any pattern permitting this kind of elimination also defines a tractable class when it does not
appear anywhere in the instance), [6] show that even with only local properties of the same kind (a given pattern does not appear
on any variable, but not all tractable patterns permit variable elimination. While in the present paper we infer tractability from a globally-held property (that is, a given pattern does not appear
in the constraint graph), whose constraint graph is a tree satisfies the broken-triangle property for some ordering of its variables; furthermore such an ordering can be determined in polynomial time. However, tractability is not due to a property of the constraint graph, since instances satisfying the broken-triangle property exist for arbitrary constraint graphs.

Recently the broken-triangle property has also inspired the development of simplification operations based on the
absence of patterns of compatibilities and incompatibilities on particular variables or values (known as existential patterns). While in the present paper we infer tractability from a globally-held property (that is, a given pattern does not appear anywhere in the instance), [6] show that even with only local properties of the same kind (a given pattern does not appear on a given variable X_i) it is possible to deduce information about the relationship between the variable X_i and the rest of the CSP instance. Depending on that information, it may then be possible to remove the variable without modifying the satisfiability of the CSP instance. Note that any pattern permitting this kind of elimination also defines a tractable class when it does not occur on any variable, but not all tractable patterns permit variable elimination. It was possible to characterise all patterns permitting variable elimination [6], but for the more challenging problem of characterising all forbidden patterns defining tractable classes, we restrict ourselves in the present paper to 2-constraint patterns as an important first step towards a complete characterisation.

Two other examples of forbidden patterns which define tractable classes of binary CSP instances are based on the
transitivity of compatibilities or incompatibilities [16]. The former class consists of all binary CSP instances in
which for all triples of assignments \( a_1 = (X_i, a), a_2 = (X_j, b), a_3 = (X_k, c) \), \( a_4 = (X_d, d) \) are such that the pairs of assignments \( (a_1, a_2), (a_1, a_3), (a_2, a_4) \) are compatible, then at least one of the pairs of assignments \( (a_1, a_3), (a_2, a_3) \) is also compatible. The forbidden subproblem is shown in Fig. 1. It has three constraints, because it has at least one edge between each of the three different pairs of variables. For example, any binary CSP instance whose constraint graph is a tree satisfies the broken-triangle property for some ordering of its variables; furthermore such an ordering can be determined in polynomial time. However, tractability is not due to a property of the constraint graph, since instances satisfying the broken-triangle property exist for arbitrary constraint graphs.

The paper is structured as follows. In Sections 2–4 we give the necessary definitions concerning patterns, tractability and reductions between patterns, together with some preprocessing operations which can always be assumed to have been applied. In Section 5 we give a dichotomy for one-constraint patterns. Section 6 is devoted to the main dichotomy result for two-constraint patterns. This result first appeared as a conference paper without the proof of the most difficult case [12]. Finally, we extend this dichotomy to include existential patterns in Section 7.

2. Definitions

We first define the notion of a CSP pattern. A pattern can be seen as a generalisation of a binary CSP instance: it represents a set of subproblems by leaving the consistency of some tuples undefined. We use the term point to denote an assignment
of a value to a variable, i.e. a pair \(a = (v, d)\) where \(d\) is in the domain of variable \(v\). A pattern is a graph in which vertices correspond to points and both vertices and edges are labelled. The label of a vertex corresponding to an assignment \((v, d)\) is simply the variable \(v\) and the label of an edge between two vertices describes the compatibility of the pair of assignments corresponding to the pair of vertices.

**Definition 1.** A **pattern** is a quintuplet \((V, A, \text{var}, E, \text{cpt})\) comprising:

- a set \(V\) of variables,
- a set \(A\) of points (assignments),
- a variable function \(\text{var} : A \to V\),
- a set \(E \subseteq \binom{A}{2}\) of edges (unordered pairs of elements of \(A\)) such that \([a, b] \in E \Rightarrow \text{var}(a) \neq \text{var}(b)\),
- a Boolean-valued compatibility function \(\text{cpt} : E \to \{F, T\}\), where for notational simplicity, we write \(\text{cpt}(a, b)\) instead of \(\text{cpt}((a, b))\).

**Definition 2.** A **binary CSP instance** is a pattern \((V, A, \text{var}, E, \text{cpt})\) such that \(E = \{(a, b) \mid (a, b) \subseteq A \times A, \text{var}(a) \neq \text{var}(b)\}\) (i.e. the compatibility of each pair of assignments to distinct variables is specified by the compatibility function). The question corresponding to the instance is: does there exist a consistent set of assignments to all the variables, that is a **solution** \(\overline{A} \subseteq A\) such that \(|\overline{A}| = |V|\), \((\forall a, b \in \overline{A}, \text{var}(a) \neq \text{var}(b))\) and \((\forall e \in \binom{\overline{A}}{2}) \cap E, \text{cpt}(e) = T)\)?

A CSP instance can be viewed as a “total” pattern, that is a pattern where every two points not in the same domain are either compatible or incompatible with each other.

For a pattern \(P = (V, A, \text{var}, E, \text{cpt})\) and a variable \(v \in V\), we use \(A_v\) to denote the set of assignments \([a \in A \mid \text{var}(a) = v]\). The **constraint on variables** \(v_1, v_2 \in V\) is the pattern \([\{(v_1, v_2), A_{v_1} \cdot \text{var} | A_{v_1}, E_{v_1}, \text{cpt} | E_{v_1}\}]\) where \(A_{v_1} = A_{v_1} \cup A_{v_2}\) and \(E_{v_1} = \{(a, b) \mid a \in A_{v_1}, b \in A_{v_2}\} \cap E\). If \(\text{cpt}(a, b) = T\) then the two assignments (points) \(a, b\) are compatible and \((a, b)\) is a compatibility edge; if \(\text{cpt}(a, b) = F\) then the two assignments \(a, b\) are incompatible and \((a, b)\) is an incompatibility edge. In a pattern, the compatibility of a pair of points \(a, b\) such that \(\text{var}(a) \neq \text{var}(b)\) and \((a, b) \notin E\) is undefined. A pattern can be viewed as a compact means of representing the set of all instances obtained by arbitrarily specifying the compatibility of such pairs. Two patterns \(P\) and \(Q\) are isomorphic if they are identical except for a possible renaming of variables and assignments.

In a CSP instance \((V, A, \text{var}, E, \text{cpt})\), we call the set \([d \mid (v, d) \in A]\) of values that can be assigned to variable \(v\) the **domain** of \(v\) and the set \([\{(d_1, d_2) \mid ((v_1, d_1), (v_2, d_2)) \in A_{v_1} \times A_{v_2} \land \text{cpt}((v_1, d_1), (v_2, d_2)) = T\}]\) of compatible pairs of values that can be assigned to two variables \(v_1, v_2 \in V\) the **constraint relation** on \(v_1, v_2\). The constraint between variables \(v_1\) and \(v_2\) in a CSP instance is **non-trivial** if there is at least one incompatible pair of assignments, i.e. \(a \in A_{v_1}\) and \(b \in A_{v_2}\) such that \(\text{cpt}(a, b) = F\). The **constraint graph** of an instance \((V, A, \text{var}, E, \text{cpt})\) is \((V, H)\), where \(H\) is the set of pairs of variables \(v_1, v_2 \in V\) such that the constraint on \(v_1, v_2\) is non-trivial.

**Definition 3.** We say that a pattern \(P\) occurs in a pattern \(P'\) (or that \(P'\) **contains** \(P\)) if \(P'\) is isomorphic to a pattern \(Q\) in the transitive closure of the following two operations (extension and merging) applied to \(P\):

**extension** \(P\) is a sub-pattern of \(Q\) (and \(Q\) an extension of \(P\)) if \(P = (V_F, A_F, \text{var}_F, E_F, \text{cpt}_F)\) and \(Q = (V_Q, A_Q, \text{var}_Q, E_Q, \text{cpt}_Q)\), then \(V_F \subseteq V_Q, A_F \subseteq A_Q, \text{var}_F = \text{var}_Q | A_F, E_F \subseteq E_Q, \text{cpt}_F = \text{cpt}_Q | E_F\).

**merging** Merging two points \(P\) transforms \(P\) into \(Q\) if \(P = (V_F, A_F, \text{var}_F, E_F, \text{cpt}_F)\) and \(Q = (V_Q, A_Q, \text{var}_Q, E_Q, \text{cpt}_Q)\), then \(\exists a, b \in A_F\) such that \(\text{var}_F(a) = \text{var}_F(b)\) and \(\forall c \in A_F\) such that \([a, c], [b, c] \in E_F, \text{cpt}_F(a, c) = \text{cpt}_F(b, c)\).

Furthermore, \(V_F = V_Q, A_F = A_Q \setminus [b], \text{var}_Q = \text{var}_A | A_Q, E_Q = E_F \cup \{[a, x] | [b, x] \in E_F\} \cup \text{cpt}_Q(a, x) = \text{cpt}_Q(b, x)\).

Consider the four patterns shown in **Fig. 2**. Assignments (points) are represented by bullets, and assignments to the same variable \(v\) are grouped together within an oval representing \(A_v\). Solid lines represent compatibility edges and dashed lines incompatibility edges. For example, \(Y\) consists of 4 points \(a, b \in A_v, c, d \in A_v\) such that \(\text{cpt}(a, c) = \text{cpt}(b, c) = T\) and \(\text{cpt}(b, d) = F\). \(Y\) occurs in \(Z\) since \(Z\) is an extension of \(Y\). \(Y\) occurs in \(V\) since \(V\) can be obtained from \(Y\) by merging points \(a, b\). \(Y\) also occurs in \(X\) by an extension followed by an extension.

**Notation.** Let \(P\) be a CSP pattern. We use \(\text{CSP}(P)\) to denote the set of binary CSP instances \(Q\) in which \(P\) does not occur.
Definition 4. A pattern $P$ is intractable if $CSP(\overline{P})$ is NP-complete. It is tractable if there is a polynomial-time algorithm to solve $CSP(\overline{P})$.

As we will show, all forbidden patterns we study are either polynomial (tractable) or NP-Complete (intractable).

Definition 5. A pattern $P$ is mergeable (non-mergeable) if $P$ can (cannot) be transformed into a pattern $Q \neq P$ by merging.

Forbidding a mergeable pattern is equivalent to forbidding more than one pattern. Since known tractable classes defined by forbidding patterns [13,15] are defined by forbidding a single non-mergeable pattern, we concentrate on this case in this paper. We characterise all tractable non-mergeable two-constraint patterns.

It is worth observing that, in a class of CSP instances defined by forbidding a pattern, there is no bound on the size of domains. Recall, however, that CSP instances have finite domains since the set of all possible assignments is assumed to be given in extension as part of the input.

Clearly, all classes of CSP instances $CSP(\overline{P})$ defined by forbidding a pattern are hereditary: $I \subseteq CSP(\overline{P})$ and $I' \subseteq I$ (in the sense that $I$ is an extension of $I'$, according to Definition 3) together imply that $I' \subseteq CSP(\overline{P})$. Furthermore, if $I \subseteq CSP(\overline{P})$ and $I'$ is isomorphic to $I$, then $I' \subseteq CSP(\overline{P})$. Forbidding a pattern therefore only allows us to define hereditary classes closed under arbitrary permutations of variable domains.

3. Preprocessing operations on CSP instances

This section describes polynomial-time simplification operations on CSP instances. Assuming that these operations have been applied facilitates the proof of tractability of many patterns.

Let $(V, A, var, E, cpt)$ be a CSP instance. If for some variable $v, A_v$ is a singleton $\{a\}$, then the elimination of a single-valued variable corresponds to making the assignment $a$ and consists of eliminating $v$ from $V$ and eliminating $a$ from $A$ as well as all assignments $b$ which are incompatible with $a$.

Given a CSP instance $(V, A, var, E, cpt)$, arc consistency consists in eliminating from $A$ all assignments $a$ for which there is some variable $v \neq var(a)$ in $V$ such that $v_b \in A_v, cpt(a, b) = f^{-1}(1)$.

Given a CSP instance $(V, A, var, E, cpt)$, if $var(a) = var(b)$ and for all variables $v \neq var(a), vc \in A_v, cpt(a, c) = T \Rightarrow cpt(b, c) = T$, then we can eliminate $a$ from $A$ by neighbourhood substitution, since in any solution in which $a$ appears, we can replace $a$ by $b$ [19]. Establishing arc consistency and eliminating single-valued variables until convergence produces a unique result, and the result of applying neighbourhood substitution operations until convergence is unique modulo isomorphism [8]. Since removing points or variables from a CSP instance does not introduce any pattern, none of these three operations when applied to an instance in $CSP(\overline{P})$ can introduce the forbidden pattern $P$.

We now consider two new simplification operations. They are simple operations which can be applied to certain CSP instances. We can always perform the fusion of two variables $v_1, v_2$ in a CSP instance into a single variable $v$ whose set of assignments is the cartesian product of the sets of assignments to $v_1$ and to $v_2$. Under certain conditions, we do not need to keep all elements of this cartesian product and, indeed, the total number of assignments actually decreases. The semantics of the two fusion operations defined below will become clear with the explanations given in the proof of Lemma 1.

Definition 6. Consider a CSP instance $(V, A, var, E, cpt)$ with $v_1, v_2 \in V$. Suppose that there is a fusion function $f : A_{v_1} \rightarrow A_{v_2}$ such that for all $u \in A_{v_1}$, whenever $u$ is in a solution $S$, there is a solution $S'$ containing both $u$ and $f(u)$. Then we can perform the simple fusion of $v_2$ and $v_1$ to create a new fused variable $v$. The resulting instance is $(V', A', var', E', cpt')$ defined by $V' = (V \setminus \{v_1, v_2\}) \cup \{v\}$, $A' = A \setminus \{v_1, v_2\}$, $var'(v) = var(u)$ for all $u \in A' \setminus A_{v_1}$, and $var'(u) = v \forall u \in A_{v_1}$, $E' = \{p, q \in A' \mid var'(p) \neq var'(q), \text{ and } f(p, q) = \text{ cpt}(p, q) \text{ if } p, q \in V' \setminus A_{v_2}, \text{ and } \text{ cpt}(u, q) = \text{ cpt}(u, q) \land \text{ cpt}(f(u), q) \text{ for all } u \in A_{v_1} \}$.

Definition 7. Consider a CSP instance $(V, A, var, E, cpt)$ with $v_1, v_2 \in V$ and a hinge value $a \in A_{v_1}$. Suppose that there is a fusion function $f : A_{v_1} \setminus \{a\} \rightarrow A_{v_2}$ such that for all $u \in A_{v_1} \setminus \{a\}$, whenever $u$ is in a solution $S$, there is a solution $S'$ containing both $u$ and $f(u)$. Then we can perform the complex fusion of $v_2$ and $v_1$ to create a new fused variable $v$. The resulting instance is $(V', A', var', E', cpt')$ defined by $V' = (V \setminus \{v_1, v_2\}) \cup \{v\}$, $A' = A \setminus \{a\}$, $var'(v) = var(u)$ for all $u \in A' \setminus (A_{v_1} \cup A_{v_2})$ and $var'(u) = v \forall u \in (A_{v_1} \setminus \{a\}) \cup A_{v_2}$, $E' = \{p, q \in A' \mid \text{ var}(p) \neq \text{ var}(q), \text{ and } f(p, q) = \text{ \text{ cpt}(p, q) \text{ if } p, q \in A' \setminus (A_{v_1} \cup A_{v_2}), \text{ and } \text{ cpt}(u, q) = \text{ cpt}(u, q) \land \text{ \text{ cpt}(f(u), q) \text{ for all } u \in A_{v_1} \setminus \{a\} \text{ and all } q \in A' \setminus (A_{v_1} \cup A_{v_2})} \}$.

Lemma 1. If $I$ is a CSP instance and $I'$ the result of a (simple or complex) fusion of two variables in $I$, then $I'$ is solvable if $I$ is solvable.

Proof. We give the proof only for the case of a complex fusion, since a simple fusion can be considered as a special case. Among the assignments in the cartesian product of $A_{v_1}$ and $A_{v_2}$, it is sufficient, in order to preserve solvability, to keep only those of the form $(a, p)$ where $p \in A_{v_2}$ or of the form $(u, f(u))$ where $u \in A_{v_1} \setminus \{a\}$. So if $I$ is solvable, then $I'$ is solvable. To complete the proof, it suffices to observe that in $A'$ we use $p \in A_{v_2}$ to represent the pair of assignments $(a, p)$ and $u \in A_{v_1} \setminus \{a\}$ to represent $(u, f(u))$. So if $I'$ is solvable, then $I$ is solvable. □
Corollary 1-constraint patterns that are essential for the proof of the 2-constraint dichotomy given in Section 6.

5. One-constraint patterns

and that do not have as a sub-pattern a reduction of a known intractable pattern.

Lemma 3. Let $P$ be a pattern such that a constraint in $P$ contains two non-mergeable incompatibility edges that cannot be merged. Then $P$ is intractable.

Proof. Let $P$ be a pattern such that a constraint in $P$ contains two non-mergeable incompatibility edges. Let SAT1 be the set of SAT instances with at most one occurrence of each variable in each clause. SAT1 is trivially equivalent to SAT which is well known to be NP-complete [7]. To prove the lemma it suffices to give a polynomial reduction from SAT1 to CSP($P$). We suppose that we have a SAT1 instance $I = \{V, S\}$ with $V$ a set of variables $\{v_1, v_2, \ldots, v_n\}$ and $S$ a set of clauses $\{C_1, C_2, \ldots, C_k\}$ such that each clause $C_i$ is a disjunction of literals $l_i^1 \lor \cdots \lor l_i^{m_i}$. We create the following CSP instance $I'$:

- $n + k$ variables $v'_1, \ldots, v'_n, \tau_1, \ldots, \tau_k$,
- $\forall v'_i$ with $1 \leq i \leq n$, two points $v_i$ and $\tau_i$ in $A_{v'_i}$.

Fusion preserves solvability and the total number of assignments decreases by at least 1 (in fact, by $|A_{v_i}|$ in the case of a simple fusion). However, when solving instances $I \in \text{CSP}(P)$, for some pattern $P$, a fusion operation will only be useful if it does not introduce the forbidden pattern $P$.

4. Reduction

In a pattern $P = (V_p, A_p, \text{var}_p, E_p, \text{cpt}_p)$, a point $a$ which is linked by a single compatibility edge to the rest of $P$ is known as a dangling point. If an arc consistent instance $I = (V, A, \text{var}, E, \text{cpt})$ with $|V| \geq |V_p|$ does not contain the pattern $P$ then it does not contain the pattern $P'$ which is equivalent to $P$ in which the dangling point $a$ and the corresponding compatibility edge have been deleted. Thus, since arc consistency is a polynomial-time operation which cannot introduce a forbidden pattern, to decide tractability we only need to consider patterns without dangling points.

Definition 8. We say that a pattern $P$ can be reduced to a pattern $Q$, and that $Q$ is a reduction of $P$, if $Q = P$ or if $Q$ is in the transitive closure of the two operations merging and dp-elimination applied to $P$, where dp-elimination is the following operation:

dp-elimination Eliminating a dangling point, its corresponding compatibility edge and its corresponding variable $v$ (if $A_v$ becomes empty) from $P$ transforms $P$ into $Q$. We give an example in Fig. 3.

Lemma 2. Let $P = (V_p, A_p, \text{var}_p, E_p, \text{cpt}_p)$ and $Q = (V_Q, A_Q, \text{var}_Q, E_Q, \text{cpt}_Q)$ be two patterns, such that $P$ can be reduced to a sub-pattern of $Q$. Let $I = (V, A, \text{var}, E, \text{cpt})$ be a CSP instance satisfying arc consistency, with $|V| \geq |V_p|$. If $Q$ occurs in $I$, then $P$ also occurs in $I$.

Proof. By definition, reduction is a transitive relation. Therefore, by induction, it suffices to prove the result for each of the individual operations: merging and dp-elimination. We suppose $Q$ occurs in $I$. If merging two points $a$ and $b$ in $P$ transforms it into a sub-pattern $Q'$ of $Q$, then $P$ actually covers two different patterns: the one where $a$ and $b$ are different points, and the one where $a$ and $b$ are the same point. The latter pattern is $Q'$. So the set of instances containing $Q$ is a subset of the set of instances containing $Q'$ and we have the result. If adding a dangling point and its corresponding compatibility edge to a sub-pattern $Q'$ of $Q$ transforms $Q'$ into $P$, then since $I$ satisfies arc consistency $P$ also occurs in $I$.

The following corollary follows immediately from the fact that arc consistency can be established in polynomial time.

Corollary 1. Let $P$ and $Q$ be two patterns, such that $P$ can be reduced to a sub-pattern of $Q$. Then
- If $Q$ is tractable, then $P$ is tractable.
- If $P$ is intractable, then $Q$ is intractable.

It follows that we only need to study those patterns that cannot be reduced to a sub-pattern of a known tractable pattern and that do not have as a sub-pattern a reduction of a known intractable pattern.

5. One-constraint patterns

In this section we prove a dichotomy for patterns composed of a single constraint. We also prove some results concerning 1-constraint patterns that are essential for the proof of the 2-constraint dichotomy given in Section 6.

Lemma 3. Let $P$ be a pattern such that a constraint in $P$ contains two distinct incompatibility edges that cannot be merged. Then $P$ is intractable.

Proof. Let $P$ be a pattern such that a constraint in $P$ contains two non-mergeable incompatibility edges. Let SAT1 be the set of SAT instances with at most one occurrence of each variable in each clause. SAT1 is trivially equivalent to SAT which is well known to be NP-complete [7]. To prove the lemma it suffices to give a polynomial reduction from SAT1 to CSP($P$). We suppose that we have a SAT1 instance $I = \{V, S\}$ with $V$ a set of variables $\{v_1, v_2, \ldots, v_n\}$ and $S$ a set of clauses $\{C_1, C_2, \ldots, C_k\}$ such that each clause $C_i$ is a disjunction of $C_i$ literals $l_i^1 \lor \cdots \lor l_i^{m_i}$. We create the following CSP instance $I'$:

- $n + k$ variables $v'_1, \ldots, v'_n, \tau_1, \ldots, \tau_k$,
- $\forall v'_i$ with $1 \leq i \leq n$, two points $v_i$ and $\tau_i$ in $A_{v'_i}$.
Suppose we have a pattern $P$. Let $\langle V, A, var, E, cpt \rangle$ be a CSP instance such that there is a point $a \in A$ and points $b, c \in A'$ such that $a$ is compatible with $b$ and $c$ and incompatible with $d$. Hence, we can merge $a$ and $b$, which is a contradiction. 

Let $I'$ be the pattern on two variables $v$ and $v'$, shown in Fig. 2, with points $a, b \in A_v$ and points $c, d \in A_{v'}$ such that $a$ is compatible with both $c$ and $d$, $b$ is compatible with $c$ and incompatible with $d$.

**Lemma 5.** $Z$ is intractable.

**Proof.** Since 3-colouring is NP-complete [20], it suffices to give a polynomial reduction from 3-colouring to CSP($Z$), the set of CSP instances in which the pattern $Z$ does not occur. For $s, t \in \{1, 2, 3\}$, define the relation $R_{s,t} \subseteq \{1, 2, 3\}$ by

$$R_{s,t} = \{(u, v) | (u = s \land v = t) \lor (u = s \land v = t)\}.$$ 

It is easy to verify that $R_{s,t}$ does not contain the pattern $Z$. Consider the 5-variable gadget with variables $v_1, v_2, u_1, u_2, u_3$, each with domain $\{1, 2, 3\}$, and with constraint relations $R_{k, k}$ on variables $(v_i, u_i)$ ($k = 1, 2, 3$) and constraint relations $R_{1+i \mod 3, k}$ on variables $(u_i, v_j)$ ($k = 1, 2, 3$). Any instance $(V, E)$ of 3-colouring, with $V = \{1, \ldots, n\}$, can be reduced to an instance of CSP($Z$) with variables $v_1, \ldots, v_n$ by placing a copy of this gadget between every pair of variables $(v_i, v_j)$ such that $(i, j) \in E$. This reduction is clearly polynomial. 

Let $I$ be the pattern on two variables $v$ and $v'$ with points $a, b \in A_v$ and $c, d \in A_{v'}$ such that $a$ and $b$ are incompatible. $I$ is a trivial tractable pattern, because any CSP instance not containing $I$ contains only trivial constraints.

**Lemma 6.** Let $P$ be a pattern on one constraint. Then either $P$ is reducible to a sub-pattern of $I$, and thus is tractable, or $P$ is intractable.

**Proof.** Let $P$ be a pattern on one constraint between two variables $v$ and $v'$. From Lemma 3, we know that if $P$ has two non mergeable incompatibility edges, then $P$ is intractable. If there is no incompatibility edge at all in $P$, then $P$ is reducible by merging and/or dp-elimination to the empty pattern, which is a sub-pattern of $I$. We therefore suppose that there is exactly one incompatibility edge in $P$, or that $P$ can be reduced by merging to a pattern with only one incompatibility edge. Let $a \in A_v$, and $b \in A_{v'}$ be the points defining this edge. From Lemma 4, we know that we only need to consider at most one other point $c \neq a$ in $A_v$, and at most one other point $d \neq b$ in $A_{v'}$. If all three edges $[a, d], [c, b]$ and $[a, d]$ are compatibility edges, then $P$ is intractable from Lemma 5. If only two or less of these edges are compatibility edges, then $P$ is reducible by merging and/or dp-elimination to $I$. So we have the lemma. 

**Lemma 7.** Let $P$ be a pattern composed of two separate one-constraint patterns: $P_1$ on variables $v_0, v_1$ and $P_2$ on variables $v_2, v_3$, where all four variables are distinct. Then

1. If either $P_1$ or $P_2$ is intractable, then $P$ is intractable too.
2. If both $P_1$ and $P_2$ are tractable, then $P$ is tractable.

**Proof.** Let $P_1$ and $P_2$ be sub-patterns of $P$. If one of them is intractable, then $P$ is intractable too, by Corollary 1. Suppose that both $P_1$ and $P_2$ are tractable. So there are two polynomial algorithms $A_1$ and $A_2$, which solve CSP($F_1$) and CSP($P_2$), respectively. Let $I$ be a CSP instance such that $P$ does not occur in $I$. If $P_1$ does not occur in $I$ then this can be detected in polynomial time and $I$ can be solved by $A_1$. If $P_2$ occurs on variables $u, v$ in $I$, then for each assignment of values to the pair of variables $u, v$, the resulting instance $I'$ cannot contain $P_2$ and hence can be solved by $A_2$. 

**Definition 9.** Given a pattern $P = \langle V, A, var, E, cpt \rangle$, a variable $v \in V$, and a point $a \in A_v$, we say that $a$ is explicitly compatible (respectively explicitly incompatible) if there is a point $b \in A_v$ such that $a$ is compatible with $b$ (respectively such that $a$ is incompatible with $b$).
The following lemma concerns a pattern in which some structure is imposed on domain elements. It is essential for our two-constraint dichotomy.

Let $2V$ be the pattern on three variables $v_0, v_1$ and $v_2$ with three points $a, b, c \in A_{v_0}$, three points $d, e, f \in A_{v_2}$ and six points $g, h, i, j, k, l \in A_{v_1}$, such that $a$ is compatible with $h$, $b$ is compatible with $g$ and $c$ is incompatible with $i, d$ is incompatible with $j$, $e$ is compatible with $k$ and $f$ is compatible with $l$. The pattern $2V$ also has the associated structure $(a \neq b \text{ or } g \neq h)$ and $(e \neq f \text{ or } k \neq l)$. When a pattern has an associated structure given by a property $\mathcal{P}$, the property $\mathcal{P}$ must be preserved by extension and reduction operations. For example, if $\mathcal{P}$ is $a \neq b$ then the points $a$ and $b$ cannot be merged during a reduction. It is worth pointing out that in a CSP instance, all points are assumed to be distinct and hence a property such as $a \neq b$ is necessarily satisfied. The pattern $2V$ is represented in Fig. 4.

Lemma 8. $2V$ is intractable.

Proof. Let the gadget $V^+$ be the pattern on two variables $v_0, v_1$ with points $a \in A_{v_0}$ and $b, c \in A_{v_1}$ such that $a$ is compatible with both $b$ and $c$, together with the structure $b \neq c$. In the pattern $2V$, either $b$ is compatible with two different points $g$ and $h$, or $h$ is compatible with two different points $a$ and $b$. So, if $2V$ occurs in a CSP instance on variables $v_0, v_1, v_2$, then the gadget $V^+$ necessarily occurs in the constraint between $v_0$ and $v_1$. By an identical argument, the gadget $V^+$ must also occur in the constraint between $v_0$ and $v_2$.

We define an equality constraint between two variables $v$ and $v'$ as the constraint consisting of compatibility edges between identical values in the domains of $v$ and $v'$ and incompatibility edges between all pairs of distinct values. Thus, by definition, a point in an equality constraint is compatible with only one point. Since the gadget $V^+$ contains a point $a$ compatible with two different points, $V^+$ does not occur in an equality constraint.

We will reduce CSP to CSP($2V$). Let $I$ be a CSP instance. For each pair of variables $v, w$ in $I$ such that there is a non-trivial constraint between $v$ and $w$, we introduce two new variables $v'$ and $w'$ such that the domain of $v'$ is the same as the domain of $v$, the domain of $w'$ is the same as the domain of $w$. We add equality constraints between $v$ and $v'$, and between $w$ and $w'$, and we add between $v'$ and $w'$ the same constraint as there was between $v$ and $w$. All other constraints involving $v'$ or $w'$ are trivial. We also replace the constraint between $v$ and $w$ by a trivial constraint. After this transformation, $v$ and $w$ are the only variables which share a non trivial constraint with $v'$.

Let $I'$ be the instance obtained after all such transformations are simultaneously performed on $I$. By construction, $I'$ has a solution if and only if $I$ has a solution.

We now suppose that we have three variables $v_0, v_1$ and $v_2$ in $I'$ such that there are non-trivial constraints between $v_0$ and $v_1$ and between $v_0$ and $v_2$. By construction, at least one of these constraints is an equality constraint. Hence, the gadget $V^+$ cannot occur in both of these constraints. It follows that $2V$ cannot occur in $I'$. So we have reduced $I$ to an instance without any occurrence of the pattern $2V$. This polynomial reduction from CSP to CSP($2V$) shows that $2V$ is intractable.

6. Two-constraint patterns

6.1. A dichotomy for two-constraint patterns

Let $T$ be the set $\{T_1, T_2, T_3, T_4, T_5\}$ of patterns shown in Fig. 5.

No pattern in $T$ can be reduced to a sub-pattern of a different pattern in $T$. As we will show, each $T_i (i = 1, \ldots, 5)$ defines a tractable class of binary CSP instances. For example, $T_4$ defines a class of instances which includes as a proper subset all instances with zero–one–all constraints [10]. Zero–one–all constraints can be seen as a generalisation of 2SAT clauses to multi-valued logics.

Let $2I$ represent the pattern composed of two separate copies of $I$, i.e. $2I$ consists of four points $a, b, c, d$ such that $var(a), var(b), var(c), var(d)$ are all distinct and both $a, b$ and $c, d$ are pairs of incompatible points.
Definition 10. We say that a pattern $P$ is irreducible if we cannot apply merging or $dp$-elimination on $P$.

Theorem 1. Let $P$ be an irreducible pattern on two constraints. Then $P$ is tractable if and only if $P$ is a sub-pattern of one of the patterns in $T \cup \{1^V\}$.

6.2. Proof of Theorem 1

6.2.1. Necessary

$\Rightarrow$: A two-constraint pattern involves either three or four distinct variables. Consider first the latter case, in which $P$ is composed of two separate irreducible one-constraint patterns $P_1$ and $P_2$ on four distinct variables. By Lemma 7, $P$ is tractable if and only if both $P_1$ and $P_2$ are tractable. Furthermore, by Lemma 6, all tractable one-constraint irreducible patterns are sub-patterns of $1^I$. Thus, if $P$ is tractable, then it is a sub-pattern of $2I$, by a combination of $P_1$ and $P_2$ being sub-patterns of $1I$. It only remains to study two-constraint patterns on three variables.

From Lemmas 3, 5 and Corollary 1, we know that we only have to study patterns $P$ with at most one explicitly incompatible point in each constraint such that $P$ does not contain the pattern $Z$. If one of the constraints does not contain any incompatibility edge at all, then the pattern is reducible by merging and/or $dp$-elimination to a pattern with only one constraint or to the pattern $Diamond$, shown in Fig. 6, which is a sub-pattern of $T_2$, $T_3$ and $T_4$. So we can assume from now on that there is exactly one incompatibility edge $(p \in A_{v_0}, b \in A_{v_1})$ between $v_0$ and $v_1$, and also exactly one incompatibility edge $(p' \in A_{v_0}, c \in A_{v_2})$ between $v_0$ and $v_2$. The “skeleton” of incompatibility edges of an irreducible tractable pattern can thus take two forms according to whether $p = p'$ (skeleton of type 1) or $p \neq p'$ (skeleton of type 2).

From Lemma 4 we know that $|A_v| \leq 2$ for each variable $v$ with only one explicitly incompatible point, and that $|A_v| \leq 3$ for each variable $v$ with two explicitly incompatible points. We know from Lemmas 5 and 8 that both $Z$ and $2V$ are intractable, so by Corollary 1 we must look for patterns in which neither one occurs. We know that we have two possible incompatibility skeletons to study, each one implying a maximum number of points appearing in the pattern.

We first consider the incompatibility skeleton of type 1, shown in Fig. 7.

Suppose that $a$ is a point in the pattern. Then there must be a compatibility edge between $a$ and $e$, otherwise we could merge $a$ and $b$. There also must be a compatibility edge between $a$ and $f$, otherwise $a$ would be a dangling point. Similarly, if $d$ is a point in the pattern, then there must be compatibility edges between $d$ and $e$, and between $d$ and $f$. So if both $a$ and $d$ are points in the pattern, then the pattern $2V$ occurs. So, by Lemma 8 and Corollary 1, $a$ and $d$ cannot be both points of the pattern. Since they play symmetric roles, we only have two cases to consider: either $a$ is a point in the pattern and not $d$, or neither $a$ nor $d$ is a point in the pattern.

Fig. 5. The set of tractable patterns $T$.

Fig. 6. The pattern Diamond.

Fig. 7. Incompatibility skeleton of type 1.
If $a$ is a point in the pattern and not $d$, then the only remaining edges to consider are $\{f, b\}$ and $\{f, c\}$. $\{f, b\}$ cannot be a compatibility edge, because otherwise the pattern $Z$ would occur. $\{f, c\}$ must be a compatibility edge, otherwise we could merge $f$ and $e$. Thus the pattern is $T_2$.

On the other hand, if neither $a$ nor $d$ is a point in the pattern, then the only remaining edges to consider are $\{f, b\}$ and $\{f, c\}$. If one of them is a compatibility edge but not the other, then $f$ would be a dangling point. So either both $\{f, b\}$ and $\{f, c\}$ are compatibility edges, or neither of them is. However, the latter case is a sub-pattern of the former one which is $T_1$. So the only possible irreducible tractable patterns with this incompatibility skeleton are sub-patterns of $T_1$ or $T_2$.

We now consider the incompatibility skeleton of type 2, shown in Fig. 8.

If $g$ is a point in the pattern, then there must be a compatibility edge between $g$ and $b$, otherwise we could merge $g$ and $e$. There also must be a compatibility edge between $g$ and $c$, otherwise we could merge $g$ and $f$. We suppose, for a contradiction, that $a$ is a point in the pattern. Then there is a compatibility edge between $a$ and $g$, otherwise we could merge $a$ and $b$. There is also a compatibility edge either between $a$ and $f$ or between $a$ and $g$, otherwise $a$ would be a dangling point. We cannot have a compatibility edge between $a$ and $g$, otherwise the pattern $Z$ would occur. So there is a compatibility edge between $a$ and $f$. There is a compatibility edge either between $b$ and $f$ or between $c$ and $e$, otherwise we could merge $e$ and $f$. We cannot have a compatibility edge between $b$ and $f$, otherwise the pattern $Z$ would occur. We cannot have a compatibility edge between $c$ and $e$, otherwise the pattern $2V$ would occur. So $a$ cannot be a point in the pattern. Since $a$ and $d$ play symmetric roles, we can also deduce that $d$ cannot be a point in the pattern. So the only remaining edges are $\{b, f\}$ and $\{c, e\}$. At least one of them is a compatibility edge, otherwise we could merge $e$ and $f$. If both of them are compatibility edges, the pattern $2V$ occurs. So exactly one of them is a compatibility edge. Since they play symmetric roles, we can assume for instance that $\{b, f\}$ is a compatibility edge while $\{c, e\}$ is an unknown edge which means that the pattern is $T_4$.

We now consider the case in which $g$ is not a point in the pattern. Suppose that $a$ is a point in the pattern. There is a compatibility edge between $a$ and $e$, otherwise we could merge $a$ and $b$. There is also a compatibility edge between $a$ and $f$, otherwise $a$ would be a dangling point. Similarly, if $d$ is a point in the pattern, then there must be compatibility edges between $d$ and $e$, and between $d$ and $f$. At least one of the edges $\{b, f\}$ and $\{c, e\}$ must be a compatibility edge, otherwise we could merge $e$ and $f$. In either case, $Z$ occurs in the pattern. So $a$ and $d$ cannot both be points of the pattern. Since they play symmetric roles, we only have two cases to consider: either $a$ is a point in the pattern and not $d$, or neither $a$ nor $d$ is a point in the pattern.

If $a$ is a point in the pattern, then the only remaining edges to consider are $\{b, f\}$ and $\{c, e\}$. At least one of them is a compatibility edge, otherwise we could merge $e$ and $f$. There is no compatibility edge between $b$ and $f$, otherwise the pattern $Z$ would occur. So there is a compatibility edge between $c$ and $e$. Hence the pattern is $T_4$.

If neither $a$ nor $d$ is a point in the pattern, then the only remaining edges are $\{b, f\}$ and $\{c, e\}$. At least one of them is a compatibility edge, otherwise we could merge $e$ and $f$. So either exactly one of them is a compatibility edge, or they both are. However, the former case is a sub-pattern of the latter which corresponds to pattern $T_5$. So the only possible irreducible tractable patterns with this incompatibility skeleton are sub-patterns of $T_3$, $T_4$ or $T_5$.

So if $P$ is a tractable irreducible pattern on two constraints, then $P$ is reducible to a sub-pattern of one of the patterns in $T \cup \{T_1\}$.

6.2.2. Sufficient
$
\Leftarrow$: We now give the tractability proofs for all patterns in $T \cup \{T_1\}$. We assume throughout that we have applied until convergence the preprocessing operations: arc consistency, neighbourhood substitution and single-valued variable elimination. The proof of tractability of $T_1$ is by far the longest of these proofs and will require a dozen lemmas showing that many simplification operations can be applied to instances in CSP($T_1$) without introducing the pattern $T_1$ and describing the structure of the simplified instance. The final step consists in observing that the simplified instance belongs to a known tractable class [14]. The proofs of tractability of the other patterns are based on the same principle: simplification operations can be applied which do not introduce the pattern and the resulting simplified instance belongs to a known, sometimes trivial, tractable class.

Proof of tractability of $T_1$. Let $l$ be an instance in CSP($\overline{T_1}$). Let the gadget $X$ be the pattern on two variables $v_0, v_1$, shown in Fig. 2, with points $a, b \in A_{v_0}$ and $c, d \in A_{v_1}$ such that $a$ is incompatible with $c$ and compatible with $d$, and $b$ is compatible with $c$ and incompatible with $d$.
Suppose that the gadget $X$ is a sub-pattern of the instance $I$. Suppose $a$ is in a solution $S$. Let $e \in A_{v_2}$ be such that $v_2 \neq v_0$, $v_2 \neq v_1$ and $e \in S$. Let $f$ be the point of $S$ in $v_1$.

If $b$ is incompatible with $e$ then $a$, $b$, $d$ and $e$ form the forbidden pattern. So $b$ is compatible with $e$. Similarly, if $c$ is incompatible with $e$, then $a$, $c$, $f$ and $e$ form the forbidden pattern. So $c$ is compatible with $e$. So if we replace $a$ by $b$ and $f$ by $c$ in $S$, then we have another solution. So if $a$ is in a solution, then $b$ is also in a solution. So we can remove $a$ while preserving the solvability of the instance.

So we can assume from now on that the gadget $X$ is not a sub-pattern of the instance. We say that an instance $I \in \text{CSP}(T_1)$ is simplified if we have applied neighbourhood substitution operations until convergence and all gadgets $X$ have been eliminated from $I$. We say that $I$ is fusion-simplified if it is simplified and all (simple or complex) fusion operations have been performed that do not introduce $T_1$. The following lemma indicates when we can perform fusion operations.

**Lemma 9.** Consider a (simple or complex) fusion of two variables $v, v'$ in an instance $I \in \text{CSP}(T_1)$. Suppose that whenever $(a, a')$ and $(b, b')$ are pairs of fused points during this fusion, such that $a \neq b \in A_v$ and $a' \neq b' \in A_{v'}$, either $a$ and $b'$ were incompatible in $I$ or $b$ and $a'$ were incompatible in $I$. Then the pattern $T_1$ cannot be introduced by this fusion.

**Proof.** By the definition of (simple or complex) fusion, the only way that $T_1$ could be introduced is when the two points in the central variable of $T_1$ are created by the fusion of pairs of points $(a, a')$ and $(b, b')$ such that the compatibilities of the points $a, b \in A_v$ and $a', b' \in A_{v'}$ with the two other points $a_1, a_2$ of $T_1$ are as shown in Fig. 9.

Now, if $a$ and $b'$ were incompatible, then $T_1$ was already present on points $a_1, a, b'$ in the original instance, and hence cannot be introduced by the fusion. Similarly, if $b$ and $a'$ were incompatible, then $T_1$ was already present on points $b, a', b'$, $a_2$ in the original instance. □

**Definition 11.** $\forall v, v', \forall a, b \in A_v$, we say that $b$ is better than $a$ with respect to $v'$, which we denote by $a \leq b$ for $(v, v')$ (or for $v'$ if the variable $v$ is obvious from the context), if every point in $A_v$ compatible with $a$ is also compatible with $b$.

It is easy to see that $\leq$ is a partial order. We also have the relations $\geq, <, >$ derived in the obvious way from $\leq$. We write $a \equiv b$ if $a \leq b$ and $b \leq a$.

**Lemma 10.** In a simplified instance $I \in \text{CSP}(T_1)$

1. $\forall (v, v')$, the order $\leq$ on $A_v$ with respect to $v'$ is total.
2. $\forall v, \forall a, b \in A_v$, there is $v'$ such that $a < b$ for $v'$.
3. $\forall v, \forall a, b \in A_v$, there is only one $v'$ such that $a < b$ for $v'$.

**Proof.** 1. Because the gadget $X$ cannot occur.
2. Otherwise $b$ is dominated by $a$ and we can remove it by neighbourhood substitution.
3. Because of the initial forbidden pattern. □

**Lemma 11.** In a simplified instance $I \in \text{CSP}(T_1)$, if $a < b < c$ for $(v_0, v_1)$, then there exists $v_2 \neq v_1$ such that $c < b < a$ for $(v_0, v_2)$.

**Proof.** Since we have $a < b$ for $(v_0, v_1)$, from Lemma 10.2 there is some $v_2$ such that $b = a$ for $(v_0, v_2)$. Since $b < c$ for $(v_0, v_1)$, $c \leq b$ for $(v_0, v_2)$ by Lemma 10.3. If $c < b$ for $v_2$, then we have the lemma. Otherwise, we have $c \equiv b < a$ for $v_2$. Since $b < c$ for $v_1$, there exists $v_3 \neq v_2$ such that $c < b$ for $v_3$. Since $a < b$ for $v_1$, $b \leq a$ for $v_3$. So $c < b \leq a$ for $v_3$. So we have $c < a$ for both $v_2$ and $v_3$, which is not possible. So we must have $c < b < a$ in $v_2$. □
Lemma 12. In a simplified instance $I \in \text{CSP}(T_3)$, $\forall a, b, c, d \in A_0$, for all $v_1 \neq v_0$ none of the following is true:

1. $a \equiv b < c < d$ for $v_1$.
2. $a < b \equiv c < d$ for $v_1$.
3. $a < b \equiv c \equiv d$ for $v_1$.

Proof. We give the proof only for the case 1, since the proofs of cases 2 and 3 are almost identical. Since we have $a < c < d$ for $v_1$, from Lemma 11 there exists $v_2$ such that $d < c < a$ for $v_2$. Likewise, since $b < c < d$ for $v_1$, there exists $v_3$ such that $d < c < b$ for $v_3$. Since $d < c$ for both $v_2$ and $v_3$, $v_2 \equiv v_3$ by Lemma 10.3. This leaves three possibilities:

1. $d < c < b < a$ for $v_2$; from Lemma 11 we know there is $v_3$ such that $a < b < c$ for $v_3$. So we have $a < c$ for both $v_1$ and $v_3$ with $v_1 \neq v_3$ (since $a \equiv b$ for $v_1$), which is not possible by Lemma 10.3. So we cannot have this possibility.
2. $d < c < b \equiv a$ for $v_2$; since $a \equiv b$ for both $v_1$ and $v_2$, by Lemma 10.2 there is a different $v_3$ such that $a < b$ for $v_3$. Since $c < b$ for $v_2$ and $v_3 \neq v_2$, $b \leq c$ for $v_3$. So $a < c$ for $v_3$. But we also have $a < c$ for $v_1$ and $v_1 \neq v_3$. So by Lemma 10.3 we cannot have this possibility.
3. $d < c < a < b$ for $v_2$; equivalent to the case $d < c < b < a$ after interchanging $a$ and $b$. $\Box$

Corollary 2. In a simplified instance $I \in \text{CSP}(T_3)$, if for some $(v_0, v_1)$, we have at least three equivalence classes in the order on $A_0$ with respect to $v_1$, then:

1. The order on $A_0$ with respect to $v_1$ is strict.
2. There is $v_2$ such that the order on $A_0$ with respect to $v_2$ is the exact opposite to the order on $A_0$ with respect to $v_1$.
3. $\forall v_3$ such that $v_3 \neq v_0, v_1, v_2$, there is only one equivalence class in the order on $A_0$ with respect to $v_3$.

Proof. Points 1, 2 and 3 follow respectively from Lemmas 12, 11 and 10. $\Box$

Lemma 13. In a simplified instance $I \in \text{CSP}(T_3)$, $\forall a, b, c, d \in A_0$, there is no $v_1$ such that $a \equiv b < c \equiv d$ for $v_1$.

Proof. By Lemma 10.2, we know there is some $v_2$ such that $a < b$ for $v_2$. Since we have $a < c$ and $a < d$ for $v_1$, by Lemma 10.3, we have $c \leq a$ and $d \leq a$ for $v_2$. From Corollary 2, we cannot have $c < a < b$ or $d < a < b$ for $v_2$, so we have $d \equiv c \equiv a < b$ for $v_2$. Since we have $d \equiv c \equiv a < b$ for both $v_1$ and $v_2$, we have a different variable $v_3$ such that $c < d$ for $v_3$. Since $c < b$ for $v_3$ and $v_3 \neq v_2$, $b \leq c$ for $v_3$. So $b < d$ for $v_3$. But we also have $b < d$ for $v_1$ and $v_1 \neq v_3$. So, by Lemma 10.3, we cannot have this possibility. $\Box$

Lemma 14. In a simplified instance $I \in \text{CSP}(T_3)$, if for some $(v, v')$ there are at least three equivalence classes in the order on $A_v$ with respect to $v'$, then there are the same number of points in both $A_v$ and $A_{v'}$ and the order on $A_v$ with respect to $v'$ and the order on $A_{v'}$ with respect to $v$ are strict.

Proof. Let $d$ be the number of points in $A_v$ and $d'$ the number of points in $A_{v'}$. From Lemma 12 we know that the order on $A_v$ with respect to $v'$ is strict. So we have $a_1 \prec a_2 \prec \cdots \prec a_d$ for $(v, v')$. So we have $(a_1', a_2', \ldots, a_{d'}')$ such that $\forall i \in [1, \ldots, d]$, $a_i$ and $a'_i$ are incompatible but $a_{i+1}$ and $a''_i$ are compatible. So $\forall i \in [2, \ldots, d]$ we have $a_i$ and $a'_i$ which are incompatible but $a_{i+1}$ and $a''_{i+1}$ are compatible. So, by Lemma 10.1 we have $a'_i \prec a''_i \prec \cdots \prec a'_{d'}$. Moreover, since $a_i$ is incompatible with $a'_{i+1}$, $a_i$ is compatible with all $a'_i$ for $1 \leq i < d$. By arc consistency, we have $a'_0$ such that $a_i$ and $a'_0$ are compatible. So we have $a'_0 \prec a''_1 \prec a''_2 \cdots \prec a''_{d'}$. So we have $d \leq d'$ and at least three equivalence classes in the order on $A_{v'}$ with respect to $v$. By switching $v$ and $v'$ in the proof, we can prove the remaining claims of the lemma. $\Box$

We say that the pair of variables $(v, v')$ is a 3-tiers pair if there are at least 3 classes of equivalence in the order on $A_v$ with respect to $v'$; we say that it is a 2-tiers pair otherwise.

Lemma 15. In a simplified instance $I \in \text{CSP}(T_3)$, suppose we have $v$ and $v'$ such that $(v, v')$ is a 3-tiers pair. Then we can perform the simple fusion of $v$, $v'$ without introducing $T_1$.

Proof. Let $d$ be the number of points in $A_v$. From Lemma 14 we know that the points in $A_v$ can be denoted $a_1 \prec a_2 \prec \cdots \prec a_d$ for $v'$ and the points in $A_{v'}$ can be denoted $b_1 \prec b_2 \prec \cdots \prec b_d$ for $v$. We will show that we can perform a simple fusion of $v$ and $v'$ with fusion function $f$ given by $f(a_i) = b_{d+i-1}(i = 1, \ldots, d)$.

Claim: $\forall 1 \leq i \leq d$, $(b_{d+i-1}, b_{d+i-2}, \ldots, b_d)$ is the exact set of points compatible with $a_i$.

If we have $a_i \prec a_j$ for $v'$, it means $a_i$ is compatible with strictly less points in $A_{v'}$ than $a_j$. By arc consistency, every point in $A_v$ is compatible with a point in $A_{v'}$. So $\forall 1 \leq i \leq d$, we have $d$ possibilities $(1, 2, \ldots, d)$ for the number of points compatible with $a_i$. Since we have $d$ points in $A_v$, it means that $\forall 1 \leq i \leq d$, $a_i$ is compatible with $i$ points in $A_{v'}$. By definition of the order on a variable with respect to another variable, the points in $A_{v'}$ compatible with a point $a_i \in A_v$ are the greatest points for $v$. So we have the claim.

We now show that $\forall 1 \leq i \leq d$, if $a_i$ is in a solution $S$, then there is a solution $S'$ such that both $b_{d+i-1}$ and $a_i$ are in $S'$. Let $b$ be the point of $S$ in $v'$. If $b_{d+i-1} = b$, then we have the result. Otherwise, let $c \neq b$ be a point of $S$. If $c = a_i$, then from the above claim we know that $c$ is compatible with $d_{d+i-1}/v'$. Otherwise, let $v_c = \text{var}(c)$. So $v_c \neq v$. From the above claim we
have $b_{d+1-i} < b$ for $v_i$. So $b_i \leq b_{d+1-i}$ for $v_i$. So $b_{d+1-i}$ is compatible with $c$. So $b_{d+1-i}$ is compatible with all the points in $S$. So we have a solution $S'$ obtained by replacing $b$ by $b_{d+1-i}$ in $S$ which contains both $a_i$ and $b_{d+1-i}$.

We now perform the simple fusion of $v$ and $v'$ with fusion function $f(a_i) = b_{d+1-i}$ for $1 \leq i \leq d$; we have just shown that this is a valid simple fusion. It only remains to show that the resulting instance is in CSP($T_1$), since by Lemma 1 it is solvable if and only if the original instance was solvable. Let $a, b$ be two distinct points in $A_v$. Without loss of generality, suppose that $a < b$ for $v'$. By choice of the fusion function $f$, $b$ is the smallest (according to the order $c$ for $v'$) of the points in $A_v$ compatible with $f(b)$. Therefore, $a$ and $f(b)$ are incompatible. The result then follows from Lemma 9.

Therefore, from now on, in a fusion-simplified instance $I \in$ CSP($T_1$), we can assume that each pair $(v, v')$ is a 2-tiers pair. We call winner for $(v, v')$ the points in the greater equivalence class in the order for $(v, v')$. The other points are called losers for this order. A same point can (and actually will) be a winner for a given order and a loser for another order. If for a given order there is only one equivalence class, then all the points are considered winners.

The winners for $(v, v')$ are compatible with all the points in $A_v$. The losers for $(v, v')$ are only compatible with the winners for $(v', v)$.

We say that a variable $v$ is one-winner if $\forall v' \neq v$, either only one point of $A_v$ is a winner for $(v, v')$ or all the points in $A_v$ are. Similarly, we say that a variable $v$ is one-loser if $\forall v' \neq v$, either only one point of $A_v$ is a loser for $(v, v')$ or all the points of $A_v$ are winners for $(v', v)$.

**Lemma 16.** In a simplified instance $I \in$ CSP($T_1$), $\forall v$, if there is $v'$ such that there is only one winner for $(v, v')$, then $v$ is one-winner. Similarly, if $\forall v'$ such that there is only one loser for $(v', v)$, then $v$ is one-loser.

**Proof.** Let $a, b, c, d, e, f \in A_v$ be such that there are $v_i \neq v_j$ with $a_i = b < c$ for $v_i$, $d < e \equiv f$ for $v_j$, $a \neq b$ and $e \neq f$. If $d \neq c$, then from Lemma 13, we have $a \equiv b \equiv d \equiv c$ for $v_i$ and $d < e \equiv f$ for $v_j$. So $d < c$ for both $v_i$ and $v_j$ with $v_i \neq v_j$ (which is a contradiction by Lemma 10.3). So we cannot have $d \neq c$. So $d = c$. So we have $c < e \equiv f$ for $v_j$. From Lemma 13 we have $c < e \equiv f \equiv a \equiv b$ for $v_i$, So we have $a \equiv b$ for both $v_i$ and $v_j$, by Lemma 10.2 there is a different variable $v_i$ such that $a < b$ for $v_i$, Since $a < c$ for $v_i$, $c < a$ for $v_j$, So $c < b$ for $v_j$, So $c < b$ for both $v_i$ and $v_j$ with $v_i \neq v_j$. This is impossible by Lemma 10.3. So we have the lemma.

**Corollary 3.** In a simplified instance $I \in$ CSP($T_1$), $\forall v$, either $v$ is one-winner or $v$ is one-loser.

**Proof.** Lemma 10.2 tells us that there exists $v'$ and $a, b \in A_v$ such that $a < b$ for $v'$. By Lemma 13, either there is only one winner for $(v, v')$ or only one loser. The result follows directly from Lemma 16.

Let $E$ be the set of one-winner variables and $F = V \setminus E$ with $V$ being the set of all variables. From Corollary 3, the variables in $F$ are one-loser. Let $v_a$, $v_b \in E$ be such that there is a non-trivial constraint between $v_a$ and $v_b$. Since $v_b \in E$, there is only one winner $a$ for $v_b$ in $v_a$. Similarly, there is only one winner $b$ for $v_b$ in $v_a$. We can perform a complex fusion of $v_a$ and $v_b$ with hinge value $a$ and fusion function the constant function $f \equiv b$.

By Lemma 1, the instance resulting from this fusion is solvable if and only if the original instance was solvable.

**Lemma 17.** The complex fusion of two one-winner variables $v_a$ and $v_b$ in a simplified instance of CSP($T_1$) does not create the forbidden pattern.

**Proof.** Suppose that $(c, c')$ and $(d, d')$ are corresponding pairs of points during this fusion, with $c \neq d \in A_v$, and $c' \neq d' \in A_v$. Since $v_a$ only has one winner for $v_b$, we know that either $c$ or $d$ is a loser for $v_b$. Without loss of generality, suppose $d$ is a loser for $v_b$. Since $v_b$ only has one winner for $v_a$, and losers are only compatible with winners, we know that $d$ is incompatible with $c'$ (since it is necessarily compatible with $d'$ for the fusion to take place). The result now follows directly from Lemma 9.

We have shown that we can fusion any pair of variables in $F$ between which there is a non-trivial constraint. We now do the same for $F$, the set of one-winner variables.

**Lemma 18.** In a simplified instance $I \in$ CSP($T_1$), let $v_a, v_b \in F$ (where $F$ is the set of one-winner variables of $I$) be such that there is a non-trivial constraint between $v_a$ and $v_b$. Let $a \in A_v$ and $b \in A_v$ be such that $a$ is incompatible with $b$. If $a' \in A_v$ is in a solution $S$ and $a' \neq a$, then $b$ is in a solution $S'$ containing $a'$.

**Proof.** Let $b'$ be the point of $S$ in $v_b$. If $b' = b$, then we have the result. Since $v_a$ is a one-winner variable, we know that all points in $A_v$, other than $b$ are winners. Thus $a'$ is compatible with $b$. By a symmetric argument, $b'$ is compatible with $a$. If we have $c \in S$ such that $b$ is incompatible with $c$, then $a, b', c$ and $b$ form the forbidden pattern. So $b$ is compatible with all the points in $S$. So if we replace $b'$ by $b$ in $S$ we get a solution $S'$ containing both $a'$ and $b$.

**Lemma 19.** Let $v_a, v_b$ both be one-winner variables in a simplified instance $I \in$ CSP($T_1$) such that $a \in A_v$ and $b \in A_v$ are incompatible. Then we can perform the complex fusion of $v_a$ and $v_b$ with hinge value $a$ and fusion function the constant function $f \equiv b$ without introducing the forbidden pattern $T_1$. 


From the definition of a one-loser variable, we know that there is some variable $v$ with one and only one variable. $v$ is incompatible with one and only one variable, namely $a$. Let $v$ be a complex fusion of $v$. We can assume that $v$ is compatible with all points of $A_v$. This is illustrated in Fig. 10. It is easily seen that this constraint can be written $v \equiv (v_1 = a)$.

We say a point $a$ is weakly incompatible with a variable $v$ if there exists some $b \in A_v$ such that $a$ is incompatible with $b$.

**Lemma 20.** Let $v$ be a one-loser variable in a simplified instance $I \in \text{CSP}(T_1)$. Let $f$ be a point in $A_v$. Then $f$ is weakly incompatible with one and only one variable.

**Proof.** From the definition of a one-loser variable, we know that there is some variable $v'$ such that $f$ is a loser for $(v, v')$. So $f$ is weakly incompatible with $v'$. From Lemma 10.3, we know that $f$ is a loser only for $(v, v')$. Furthermore, by arc consistency we know that $f$ is compatible with all points of $A_u$, for all variables $u$ such that $f$ is not a loser for $(v, u)$. So $f$ is weakly incompatible with one and only one variable, namely $v'$, and we have the lemma.

We have shown that after all possible fusions of pairs of variables, we have two sets of variables $E$ (the set of one-winner variables) and $F = V \setminus E$ (the set of one-loser variables) such that:

- $\forall v, v' \in E$, there is no non-trivial constraint between $v$ and $v'$.
- $\forall v, v' \in F$, there is no non-trivial constraint between $v$ and $v'$.
- $\forall v \in F, \forall f \in A_v, f$ is weakly incompatible with one and only one variable $v' \in E$. This is from Lemma 20. Furthermore, $f$ is compatible with all points of $A_v$ but one (since $v' \in E$ is a one-loser variable).
- The only possible non-trivial constraint between a variable $v_1 \in E$ and another variable $v_2 \in F$ is the following with $d_1$ being the size of the domain of $v_1$:
  - There is a point $b \in A_{v_2}$, incompatible with exactly $d_1 - 1$ points in $A_{v_1}$.
  - $\forall b' \in A_{v_2}$ with $b' \neq b$, $b'$ is compatible with all points in $A_{v_1}$.

This is illustrated in Fig. 10. It is easily seen that this constraint can be written $(v_2 = b) \Rightarrow (v_1 = a)$.

We call NOOSAT (for Non-binary Only Once Sat) the following problem:

- A set of variables $V = \{v_1, v_2, \ldots, v_e\}$.
- A set of values $A = \{a_1, a_2, \ldots, a_n\}$.
- A set of clauses $C = \{C_1, C_2, \ldots, C_e\}$ such that:
  - Each clause is a disjunction of literals, with a literal being in this case of the form $v_i = a_j$.
  - $\forall i, j, p, q((v_i = a_j) \in C_p) \land ((v_i = a_j) \in C_q) \Rightarrow p = q$.

**Lemma 21.** $\text{CSP}(T_1)$ can be reduced to NOOSAT in polynomial time.

**Proof.** The total number of assignments decreases when we fuse variables, so the total number of (simple or complex) fusions that can be performed is linear in the size of the original instance. Hence we can produce a fusion-simplified version of an instance $I \in \text{CSP}(T_1)$ in polynomial time. Thus suppose we have a fusion-simplified instance in $\text{CSP}(T_1)$. We have shown that the non-trivial constraints between variables $v \in F$ and $v' \in E$ are all of the form $v = b \Rightarrow v' = a$. Furthermore, from
Lemma 20 and the third bullet point in the description of a post-fusions instance, each variable-value assignment \( v = b \) occurs in exactly one such constraint. For any \( v \in F \), we can replace the set of such constraints \( v = b \Rightarrow v = a_i \), for all values \( b_i \) in the domain of \( v \), by the clause \( (v_1 = a_1) \lor \ldots \lor (v_d = a_d) \). It only remains to prove that no literal appears in two distinct clauses. Suppose that we have a literal \( v_1 = a \) which occurs in two distinct clauses. Then there must have been two constraints \( v_2 = b \Rightarrow v_1 = a \) and \( v_3 = c \Rightarrow v_1 = a \) and with \( v_1 \in E, v_2 \neq v_3 \in F \). Let \( a' \neq a \) be a point in \( A_{v_1} \). Then \( b \) and \( c \) are both incompatible with \( a' \) but compatible with \( a \). But this is precisely the forbidden pattern. This contradiction shows that \( \text{CSP}(T_1) \) can be reduced to NOOSAT.

The constraints in NOOSAT are convex when viewed as \( [0, \infty] \)-valued cost functions on the assignment-sets \([\{v_1, a_1\}, \ldots, \{d_i, a_d\}]\) (the cost being infinite if and only if the number of assignments in this set is 0) and these assignment-sets (corresponding to clauses) are non overlapping. So, from [14], it is solvable in polynomial time. Hence the forbidden pattern \( T_1 \) is tractable.

**Proof of tractability of \( T_2 \).** Let \( N \) be the gadget shown in Fig. 11: two variables \( v_0, v_1 \) with points \( a, b \in A_{v_0} \) and \( c, d \in A_{v_1} \), such that \( a, b \) are both compatible with \( d \), \( b \) is incompatible with \( c \), and with the structure \( a \neq b \).

Suppose we are given a CSP instance containing the gadget \( N \). Let \( v_2 \) be a variable with \( v_2 \neq v_0, v_2 \neq v_1 \) and let \( e \) be a point in \( A_{v_2} \) such that \( a \) and \( e \) are compatible. If \( b \) is incompatible with \( e \), then we have the forbidden pattern \( T_2 \) on \( d, c, b, a, e \). So \( b \) is compatible with \( e \). If all the points in \( A_{v_2} \) which are compatible with \( a \) are also compatible with \( b \), then we can remove \( a \) by neighbourhood substitution. So, assuming that neighborhood substitution operations have been applied until convergence, if we have the gadget \( N \), then there is a point \( g \in A_{v_1} \) compatible with \( a \) and incompatible with \( b \).

Let \( v_3 \neq v_1 \) such that \( v_1 \neq v_0 \). By arc consistency, there is \( h \in A_{v_1} \) such that \( h \) is compatible with \( a \). If \( b \) and \( h \) are incompatible, then we have the forbidden pattern \( T_3 \) on \( d, g, b, a, h \). So \( b \) and \( h \) are compatible. If there is \( i \in A_{v_3} \) such that \( b \) and \( i \) are incompatible, then we have the forbidden pattern on \( h, i, b, a, g \). So \( b \) is compatible with all the points in \( A_{v_3} \).

**Definition 12.** A constraint \( C \) between two variables \( v \) and \( v' \) is functional from \( v \) to \( v' \) if for all \( a \in A_{v_0} \), there is one and only one point in \( A_{v'} \) compatible with \( a \).

Let the gadget \( V^- \) be the pattern comprising three variables \( v_4, v_5, v_6 \) and points \( a \in A_{v_4}, b \in A_{v_5}, c \in A_{v_6} \) such that \( a \) is incompatible with both \( b \) and \( c \).

From now on, since \( V^- \) is a tractable pattern [16], we only need to consider the connected components of the constraint graph which contain \( V^- \).

**Lemma 22.** If in an instance from \( \text{CSP}(T_2) \), we have the gadget \( V^- \), then the constraint between \( v_5 \) and \( v_4 \) is functional from \( v_5 \) to \( v_4 \) and the constraint between \( v_4 \) and \( v_6 \) is functional from \( v_4 \) to \( v_6 \).

**Proof.** By symmetry, it suffices to prove functionality from \( v_5 \) to \( v_4 \). We suppose we have the gadget \( V^- \). Let \( d \in A_{v_4} \) be compatible with \( a \). Since \( a \) is weakly incompatible with two different variables, \( a, b \) and \( d \) cannot be part of the gadget \( N \), because otherwise \( T_2 \) would be present. So the only point in \( A_{v_4} \) compatible with \( d \) is \( a \). So if a point in \( A_{v_4} \) is compatible with \( a \), then it is only compatible with \( a \). Likewise, if a point in \( A_{v_6} \) is compatible with \( a \), then it is only compatible with \( a \).

Let \( f \neq a \) be a point in \( A_{v_4} \). By arc consistency, we have \( d \in A_{v_5} \) and \( e \in A_{v_6} \) such that \( a \) is compatible with \( d \) and with \( e \). From the previous paragraph, we know that both \( d \) and \( e \) are incompatible with \( f \). So we have the situation illustrated in Fig. 12.

So \( d, e \) and \( f \) form the gadget \( V^- \). So each point in \( A_{v_4} \) and \( A_{v_6} \) compatible with \( f \) is compatible with only one point of \( A_{v_4} \). So each point in \( A_{v_4} \) and \( A_{v_6} \) compatible with a point in \( A_{v_4} \) is compatible with only one point of \( A_{v_4} \). By arc consistency, each point of \( A_{v_4} \) and \( A_{v_6} \) is compatible with exactly one point of \( A_{v_4} \). So the constraint between \( v_4 \) and \( v_5 \) is functional from \( v_5 \) to \( v_4 \).

**Lemma 23.** In a connected component of the constraint graph containing \( V^- \) of an instance from \( \text{CSP}(T_2) \), all constraints are either functional or trivial.

**Proof.** Let \( P(V) \) be the following property: \( V \) is a connected subgraph of size at least two of the constraint graph and all constraints in \( V \) are either functional or trivial. \( P(\{v_4, v_5\}) \) is true from Lemma 22.
Let $V_{all}$ be the set of all variables of the connected subgraph of the constraint graph containing $V^\ast$. Let $V$ be a maximum (with respect to inclusion) subset of $V_{all}$ for which $P(V)$ is true. Let $V' = V_{all} \setminus V$. Let $v' \in V'$. Let $v \in V$ be such that $C(v, v')$ (the constraint on $v$, $v'$) is non-trivial. So there is $d \in A_i$ and $e \in A_{i'}$ such that $d$ and $e$ are incompatible. Since $V$ is connected and of cardinality at least two, there is $v'' \in V$ such that $C(v, v'')$ is functional. By arc consistency and elimination of single-valued variables, there is necessarily a point $f \in A_i$ such that $d$ and $f$ are incompatible. So $d$, $e$ and $f$ form the gadget $V^\ast$.

From Lemma 22 we know $C(v, v')$ is functional. So $P(V)$ is true for all subsets of $V_{all}$. □

**Lemma 24.** In an instance from CSP($T_q^2$), $\forall v$ such that $v$ is in a connected component of the constraint graph containing $V^\ast$, all points in $A_i$ are weakly incompatible with the exact same set of variables.

**Proof.** Let $a, b \in A_i$ be weakly incompatible with $v'$. So $C(v, v')$ is non-trivial. So $C(v, v')$ is functional.

If $C(v, v')$ is functional from $v$ to $v'$, then a point in $A_i$ can be compatible with only one point in $A_{i'}$. We can assume, by elimination of single-valued variables, that there are at least two points in $A_{i'}$, so every point in $A_i$ is weakly incompatible with $v'$.

If $C(v, v')$ is functional from $v'$ to $v$, then let $b \neq a$ in $v$. By arc consistency, we know there is $c \in A_i$ such that $a$ and $c$ are compatible. Since $C(v, v')$ is functional from $v'$ to $v$, $c$ is compatible with only one point in $A_i$, namely $a$, so $b$ is incompatible with $c$. So every point in $A_i$ is weakly incompatible with $v'$.

So $\forall (v, v')$, $a \in A_i$, weakly incompatible with $v' \Rightarrow \forall b \in A_{i'}, b$ weakly incompatible with $v'$. □

**Definition 13.** A sequence of variables $(v_0, v_1, \ldots, v_k)$ is a path of functionality if $\forall i \in [0, \ldots, k-1], C(v_i, v_{i+1})$ is functional from $v_i$ to $v_{i+1}$.

**Lemma 25.** In a connected component of the constraint graph containing $V^\ast$ of an instance from CSP($T_q^2$), $\forall v, v'$, either $v'$ is connected to only one other variable in the constraint graph, or there is a path of functionality from $v$ to $v'$.

**Proof.** Since we are in a connected component, there is a path of incompatibility $(v_0 = v, v_1, v_2, \ldots, v_k = v')$ with all $v_i$ different and at least one incompatibility edge between $v_i$ and $v_{i+1}$ for $0 \leq i \leq k-1$. If $v'$ is connected to at least two other variables in the constraint graph, then we have a path of incompatibility $(v_0, v_1, v_2, \ldots, v_k)$ with $v_k \neq v_{k-1}$. From Lemma 24 we have a path of incompatibility $(a_0 \in A_{i_{a_0}}, a_1 \in A_{i_{a_1}}, \ldots, a_k \in A_{i_{a_k}}, a_{k+1} \in A_{i_{a_{k+1}}})$. So $\forall i \in [1, \ldots, k], a_{i-1}$, $a_i$, and $a_{i+1}$ form the gadget $V^\ast$. So from Lemma 22, $\forall i \in [1, \ldots, k], C(v_{i-1}, v_i)$ is functional from $v_{i-1}$ to $v_i$. So we have a path of functionality from $v$ to $v'$. □
Let $S$ be a solution containing $b$. Let $f$ be the point of $S$ in $A_{v_0}$. If $f$ is compatible with $a$, then we can replace $b$ by $a$ in $S$ while maintaining the correctness of the solution, since all the points in the instance outside of $A_{v_1}$ which are compatible with $b$ are also compatible with $a$.

If $f$ is not compatible with $a$, then edges $\{a, f\}$, $\{f, b\}$ and $\{b, d\}$ form the gadget $N$. So, by our previous argument, if $f$ is compatible with a point outside of $A_{v_1}$, then $d$ is also compatible with the same point. We can then replace $b$ by $a$ and $f$ by $d$ in $S$ while maintaining the correctness of the solution, since all the points in the instance outside of $A_{v_1}$ which are compatible with $b$ are also compatible with $a$ and all the points in the instance outside of $A_{v_0}$ which are compatible with $f$ are also compatible with $d$. So if a solution contains $b$, then there is another solution containing $a$. Thus we can remove $b$ while preserving solvability.

So each time the gadget $N$ is present in an instance $I \in \text{CSP}(\overline{T}_5)$, we can remove one of its points and hence eliminate $N$. Absence of the gadget $N$ in $I$ is equivalent to saying that all constraints are either trivial or bijections and hence (a subclass of) zero–one-all constraints [10]. Since all gadgets $N$ can be removed in polynomial time and CSP instances with zero–one-all constraints can be solved in polynomial time, it follows that the pattern $T_3$ is tractable.

**Proof of tractability of $T_4$.** Consider an instance from $\text{CSP}(\overline{T}_4)$. Let $W$ be the gadget shown in Fig. 13: two variables $v_2$ and $v_1$ such that we have $a$ in $A_{v_0}$, $b$, $c$, $g$ in $A_{v_1}$, with $b \neq c$, $a$ compatible with both $b$ and $c$, and $a$ incompatible with $g$. Suppose we have $W$ in the instance.

Let $f$ be a point in $A_{v_1}$, with $v_2 \neq v_0$, $v_1$. If $f$ is compatible with $b$ but not with $c$, then we have the forbidden pattern $T_4$. Likewise, if $f$ is compatible with $c$ but not with $b$, then we have the forbidden pattern $T_4$. So all the points of the instance not in $A_{v_0}$ or $A_{v_1}$ have the same compatibility towards $b$ and $c$.

If all points in $A_{v_0}$ compatible with $b$ are also compatible with $c$, then all the points in the instance compatible with $b$ are also compatible with $c$ and by neighbourhood substitution we can remove $b$. Thus we can assume there is $d$ in $A_{v_1}$ such that $d$ is compatible with $b$ but not with $c$.

Let $S$ be a solution containing $c$. Let $e$ be the point of $S$ in $A_{v_0}$. If $e$ is compatible with $b$, then we can replace $c$ by $b$ in $S$ while maintaining the correctness of the solution, since $b$ and $c$ have the same compatibility towards all the points in the instance outside of $A_{v_0}$ and $A_{v_1}$. If $e$ is not compatible with $b$, then edges $\{b, e\}$, $\{b, a\}$ and $\{b, d\}$ form the gadget $W$. So, by our argument above, $a$ and $d$ have the same compatibility towards all the points in the instance outside of $A_{v_0}$ and $A_{v_1}$. Similarly, edges $\{c, d\}$, $\{c, a\}$ and $\{c, e\}$ form the gadget $W$. So $a$ and $e$ have the same compatibility towards all the points in the instance outside of $A_{v_0}$ and $A_{v_1}$. So $d$ and $e$ have the same compatibility towards all the points in the instance outside of $A_{v_0}$ and $A_{v_1}$. Thus we can replace $c$ by $b$ and $e$ by $d$ in $S$ while maintaining the correctness of the solution, since $b$ and $c$ have the same compatibility towards all the points in the instance outside of $A_{v_0}$ and $A_{v_1}$, and $d$ and $e$ have the same compatibility towards all the points in the instance outside of $A_{v_0}$ and $A_{v_1}$. So if a solution contains $c$, then there is another solution containing $b$. Thus we can remove $c$.

Therefore, each time the gadget $W$ is present, we can remove one of its points. The gadget $W$ is a known tractable pattern since forbidding $W$ is equivalent to saying that all constraints are zero–one-all [10]. So if it is not present, the instance is tractable. Hence pattern $T_4$ is tractable.

**Proof of tractability of $T_5$.** The pattern $T_5$ is a sub-pattern of the broken-triangle pattern $BTP$, a known tractable pattern [13] on three constraints. So the pattern $T_5$ is tractable by Corollary 1.

**Proof of tractability of $2I$.** Since $2I$ is the disjoint union of two copies of the trivially tractable pattern $I$, the tractability of $2I$ follows directly from Lemma 7.

We have proved that all patterns in $T$ are tractable. This concludes the proof of the theorem.

### 7. Two-constraint existential patterns

#### 7.1. Definitions, reduction and properties

In this section we consider a different way of defining a class of CSP instances by forbidding patterns. An existential pattern is a pattern $P$ with a set of points $e \subseteq A$, for some distinguished variable $x$. We call the points in $e$ existential points. Often $e$ will be a singleton $\{a\}$. In this case, forbidding the existential pattern $P$ means that for all variables $x$ in the instance $I$, there is some point $f_e(a) \in A_x$ such that there is no occurrence of $P$ in $I$ in which the existential point $a$ maps to $f_e(a)$.
We say that an existential pattern

**Definition 14.**

existential patterns (under the very reasonable assumption that all instances are arc consistent).

tractable classes can be defined by existential patterns. In this section we give a complete dichotomy for 2-constraint patterns

distinguished variable \(v\) of \(f\) point.

Forbidding \(\exists\) \(x\) variable, whereas forbidding \(V\)-Middle imposes the less restrictive assumption that for each variable \(x\) there exists some point \(f_x(a) \in A_x\) which is incompatible with points in at most one other variable. From Theorem 1, we know that the set of CSP instances in which we forbid the pattern \(V\)-Middle is NP-Complete. Actually, if we only consider arc-consistent instances, there even exists a linear time algorithm which can find a solution in any such instance. However, as we show later in Lemma 29, the set of instances in which we forbid the pattern \(V\)-Middle is NP-Complete, even when only considering arc-consistent instances.

When \(e\) is not a singleton, forbidding the existential pattern \(P\) means that for all variables \(x\) in \(I\), there is an injective function \(f_e : e \rightarrow A_x\) such that there is no occurrence of \(P\) in \(I\) in which each \(p \in e\) maps to \(f_e(p)\). An existential pattern \((V, A, \text{var}, E, \text{cpt}, e)\) is thus a pattern \((V, A, \text{var}, E, \text{cpt})\) to which we add a set of existential points \(e \subseteq A_x\) for some distinguished variable \(v \in V\). If \(e = \emptyset\), then the existential pattern is equivalent to the (non-existential) pattern \((V, A, \text{var}, E, \text{cpt})\).

Existential patterns have been previously studied in order to characterise under which conditions a variable can be eliminated from a binary CSP instance without the need to add any constraints [6].

Forbidding an existential version \(Q\) of a pattern \(P\) defines a much larger class \(\text{CSP}(Q)\) than \(\text{CSP}(P)\). Although existential patterns were first introduced in order to define variable elimination rules, an interesting question is whether any new tractable classes can be defined by existential patterns. In this section we give a complete dichotomy for 2-constraint existential patterns (under the very reasonable assumption that all instances are arc consistent).

We now give versions of the definitions of extension, merging, dp-elimination, occurrence and tractability generalised to existential patterns.

**Definition 14.** We say that an existential pattern \(P\) occurs in an existential pattern \(P'\) (or that \(P'\) contains \(P\)) if \(P'\) is isomorphic to an existential pattern \(Q\) in the transitive closure of the following two operations (extension and merging) applied to \(P\):

- **extension** \(P\) is a sub-pattern of \(Q\) (and \(Q\) an extension of \(P\)): if \(P = (V_P, \text{var}_P, E_P, \text{cpt}_P, e_P)\) and \(Q = (V_Q, \text{var}_Q, E_Q, \text{cpt}_Q, e_Q)\), then \(V_P \subseteq V_Q, A_P \subseteq A_Q, \text{var}_P = \text{var}_Q|_{A_P}, E_P \subseteq E_Q, \text{cpt}_P = \text{cpt}_Q|_{E_P}\), and \(e_P \subseteq e_Q\). We give an example in Fig. 16.

As a simple example, consider the pattern \(1\) and its existential version \(\exists 1\) shown in Fig. 14. Forbidding \(1\) in an instance means that all points are compatible with all other points in the instance, whereas forbidding \(\exists 1\) imposes the less restrictive assumption that for each variable \(x\) there exists some point \(f_x(a) \in A_x\) which is compatible with all other points of the instance.

As a slightly more elaborate example, consider the pattern \(V\)-Middle and its existential version \(V-Middle\) shown in Fig. 15. Forbidding \(V\)-Middle in an instance means that all points in the instance are incompatible with points in at most one other variable, whereas forbidding \(V-Middle\) imposes the less restrictive assumption that for each variable \(x\) there exists some point \(f_x(a) \in A_x\) which is incompatible with points in at most one other variable. From Theorem 1, we know that the set of CSP instances in which we forbid the pattern \(V\)-Middle is tractable. Actually, if we only consider arc-consistent instances, there even exists a linear time algorithm which can find a solution in any such instance. However, as we show later in Lemma 29, the set of instances in which we forbid the pattern \(V-Middle\) is NP-Complete, even when only considering arc-consistent instances.

![Fig. 14. A simple pattern 1 and an existential version \(\exists 1\) of the same pattern.](image1)

![Fig. 15. A pattern \(V\)- and an existential version \(V-Middle\) of the same pattern.](image2)

![Fig. 16. Example of extension of an existential pattern \(P\) to produce the existential pattern \(Q\).](image3)
as the existence of an occurrence-function $f : A_P \rightarrow A_Q$ such that
\begin{enumerate}
  \item $\forall a, b \in A_P, \forall x \in V$ with $|A_a| \leq |e_p|$ such that for all subsets $S$ of $A_a$ with $|S| = |e_p|$ and all bijections $g : e_p \rightarrow S$, $P$ occurs on $S$ via $g$. Conversely, $P$ does not appear in $I$ if $\forall v \in V$ with $|A_a| \geq |e_p|$, there is an injective mapping $g : e_p \rightarrow A_a$ such that $P$ does not occur on $g(e_p)$ via $g$.
\end{enumerate}

It is worth pointing out that we will show later that when $e_p \geq 2$ for some non-trivial existential pattern $P$, the set of CSP instances not containing $P$ is NP-Complete. This explains why such patterns appears on the tractability side of our main result and why for most of the following we only need to consider existential patterns $P$ in which $e_p$ is a singleton.

Suppose that an existential pattern $P$ does not appear in an instance $I$ since for all variables $v$, there is a subset $S_v$ of $A_v$ and a bijection $g : e_p \rightarrow S_v$ such that $P$ does not occur on $S_v$ via $g$. Establishing arc consistency in $I$ may eliminate some of the assignments in the sets $S_v$ with the consequence that $P$ may now appear in the arc-consistent version of $I$. Since arc consistency is a basic filtering operation applied by all constraint solvers to reduce the size of variable domains, we choose to study only arc consistent CSP instances.

**Notation.** Let $P$ be an existential pattern. We use $\text{CSP}_{AC}(P)$ to denote the set of arc-consistent binary CSP instances $I$ in which $P$ does not appear.

**Definition 17.** An existential pattern $P$ is intractable if $\text{CSP}_{AC}(P)$ is NP-complete. It is tractable if there is a polynomial-time algorithm to solve $\text{CSP}_{AC}(P)$.

In an existential pattern $P = (V_P, A_P, \forall x \in V_P, e_p, cpt_P, e_P)$, a point $p \notin e_p$ which is linked by a single compatibility edge to the rest of $P$ is known as a dangling point. If an arc consistent instance $I = (V, A, \forall x \in V, e_p, cpt_P)$ with $|V| \geq |V_P|$ does not contain the existential pattern $P$ then it does not contain the pattern $P'$ which is equivalent to $P$ in which the dangling point $p$ and the corresponding compatibility edge have been deleted. Thus, to decide the tractability of $\text{CSP}_{AC}(P)$ we only need to consider patterns $P$ without dangling points.

**Definition 18.** We say that an existential pattern $P$ can be reduced to an existential pattern $Q$, and that $Q$ is a reduction of $P$, if $Q = P$ or if $Q$ is in the transitive closure of the two operations merging and dp-elimination applied to $P$, where dp-elimination is the following operation:

**dp-elimination** Eliminating a dangling point, its corresponding compatibility edge and its corresponding variable $v$ (if $A_v$ becomes empty) from $P$ transforms $P$ into $Q$. We give an example in Fig. 18.

**Lemma 26.** Let $P = (V_P, A_P, \forall x \in V_P, e_p, cpt_P)$ and $Q = (V_Q, A_Q, \forall x \in V_Q, e_Q, cpt_Q, e_Q)$ be two existential patterns, such that $P$ is a sub-pattern of $Q$. Let $I = (V, A, \forall x \in V, e_P, cpt_P)$ be an arc-consistent CSP instance. If $Q$ appears in $I$, then $P$ also appears in $I$. 

![Fig. 17. Example of merging in an existential pattern P to produce the existential pattern Q.](image-url)

Merging two points in $P$ transforms $P$ into $Q$: if $P = (V_P, A_P, \forall x \in V_P, e_p, cpt_P, e_P)$ and $Q = (V_Q, A_Q, \forall x \in V_Q, e_Q, cpt_Q, e_Q)$, then $\exists a, b \in A_P$ such that $\forall x \in V_P$ with $|A_a| \leq |e_p|$ and $\forall x \in V_Q$ with $|A_b| \geq |e_Q|$ such that for all subsets $S$ of $A_a$ with $|S| = |e_p|$ and all bijections $g : e_p \rightarrow S$, $P$ occurs on $S$ via $g$. Conversely, $P$ does not occur on $S$ via $g$. Establishing arc consistency in $I$ may eliminate some of the assignments in the sets $S_v$ with the consequence that $P$ may now appear in the arc-consistent version of $I$. Since arc consistency is a basic filtering operation applied by all constraint solvers to reduce the size of variable domains, we choose to study only arc consistent CSP instances.
Proof. Suppose that Q appears in I. So ∃v ∈ V such that Q occurs on all subsets of A_v of size |e_Q| and for all bijections g : e_Q → S. Let T be any subset of A_v of size |e_P| and let h : e_P → T be any bijection. We have to show that P appears in I on T via h.

Let S be any subset of A_v of size |e_Q| such that T ⊆ S and let g : e_Q → S be any bijection such that g|e_P = h. We know that Q occurs in the existential pattern (V, A, var, E, cpt, S) with an occurrence-function f such that f|e_P = g. Since P is a sub-pattern of Q, P occurs in the existential pattern (V, A, var, E, cpt, S ∩ f(e_P)) with the occurrence-function f|e_P. Since e_P ≤ e_Q by the definition of a sub-pattern, we have f|e_P = g|e_P = h. Thus P appears in I on T = h(e_P) via h and we are done. □

Lemma 27. Let P = (V_P, A_P, var_P, E_P, cpt_P, e_P) and Q = (V_Q, A_Q, var_Q, E_Q, cpt_Q, e_Q) be two existential patterns, such that P can be reduced to a sub-pattern of Q. Let I = (V, A, var, E, cpt) be an arc-consistent CSP instance with |V| ≥ |V_P|. If Q appears in I, then P also appears in I.

Proof. By definition, reduction is a transitive relation. Therefore, by induction, it suffices to prove the result for each of the individual operations: merging and dp-elimination.

If merging two points a and b in P transforms it into a sub-pattern Q’ of Q, then P actually covers two different patterns: the one where a and b are different points, and the one where a and b are the same point. The latter pattern is Q’ which appears in I, by Lemma 26, since it is a sub-pattern of Q. So the set of instances containing Q is a subset of the set of instances containing (at least one of the two versions of) P and we have the result.

We now suppose that eliminating a dangling point c ∈ v_c with v_c ∈ V_P, and its corresponding compatibility edge from P transforms P into a sub-pattern Q’ of Q, where Q’ = (V_Q, A_Q’, var_Q’, E_Q’, cpt_Q’, e_Q’). Since c is a dangling point, from the definition of dp-elimination we know that c ∉ e_P. So e_Q’ = e_P. Let d be the point such that [c, d] is the compatibility edge eliminated from P to produce Q’. Since Q’ is a sub-pattern of Q, by Lemma 26, we know that Q’ appears in I = (V, A, var, E, cpt). So ∃v ∈ V such that for all S ⊆ A_v with |S| = |e_Q’| and for all bijections g : e_Q’ → S, Q’ occurs on S via g. Let f be the corresponding occurrence-function. Since e_P = e_Q’, it suffices to show that P also occurs on S via g. If v_c ∈ V_Q, then let v_c = var(f(v_c)) be the variable in I corresponding to v_c in this appearance of Q’. If v_c ∉ V_Q (due to being eliminated during dp-elimination), then |V_Q’| < |V’| ≤ |V|, and so we can set v_c′ ∈ V to be a variable of I not corresponding to any variable in V_Q’ in this appearance of Q’. In both cases, since I satisfies arc consistency, there is a point c’ ∈ v_c′ compatible with f(d). We can thus extend f to an occurrence-function f’ of P in I by setting: f’(c) = c’, and f’(p) = f(p) for all p ∈ A_P \ {c} = A_Q’. Hence P also occurs on S via g, since f and f’ are identical on e_P, which completes the proof. □

The following corollary follows immediately from the fact that arc consistency can be established in polynomial time.

Corollary 4. Let P and Q be two existential patterns, such that P can be reduced to a sub-pattern of Q. Then
• If Q is tractable, then P is tractable.
• If P is intractable, then Q is intractable.

It follows that we only need to study those existential patterns that cannot be reduced to a sub-pattern of a known tractable existential pattern and that do not have as a sub-pattern a reduction of a known intractable existential pattern.

Let I be a CSP instance. We say that v = V is a copy in I of v ∈ V on (A_0, A_1) with A_0 ⊆ A_v and A_1 ⊆ A_v if:

• |A_0| = |A_1|.
• ∀a ∈ A_0, ∀b ∈ A_1 such that cpt(a, b) = T and ∀c ≠ b in A_1 we have cpt(a, c) = F.
• ∀b ∈ A_1, ∀a ∈ A_0 such that cpt(a, b) = T and ∀c ≠ a in A_0 we have cpt(c, b) = F.
• ∀a ∈ A_0, ∀b ∈ A_1 such that cpt(a, b) = T, ∀c ∈ A \ {A_1, A_v}, we have cpt(b, c) = cpt(a, c).

For notational simplicity, we say that v = V is a copy of v in I if A_0 = A_v and A_1 = A_v.

Lemma 28. Let P = (V, A, var, E, cpt) be a pattern and P’ = (V, A, var, E, cpt, a) be an existential version of P. Then P’ is tractable only if P is tractable.

Proof. Since (V, A, var, E, cpt) is equivalent to (V, A, var, E, cpt, ∅), the result follows directly from Definition 14 of extension and Corollary 4. □
7.2. Some NP-complete existential patterns

In order to identify all tractable existential patterns, we begin by showing that many simple existential patterns are NP-complete. Let $V_2 = \{v_0, v_1, v_2\}$, $A_3 = \{a_0, a_1, a_2\}$, $\text{var}(a_i) = v_i$ for $i \in \{0, 1, 2\}$, $E_3 = \{(a_0, a_1), (a_0, a_2)\}$ and $\text{cpt}_3(a_0, a_1) = \text{cpt}_3(a_0, a_2) = F$. Let $V\text{-Middle} = (V_3, A_3, \text{var}_3, E_3, \text{cpt}_3, \{a_0\})$ be the existential pattern shown on the left of Fig. 19 and $V\text{-Side} = (V_4, A_4, \text{var}_4, E_3, \text{cpt}_3, \{a_1\})$ be the existential pattern shown on the right of Fig. 19.

**Lemma 29.** $V\text{-Middle}$ and $V\text{-Side}$ are NP-Complete.

**Proof.** Let $I = (V, A, \text{var}, E, \text{cpt})$ be an arc-consistent CSP instance. Let $v_1, \ldots, v_k$ be the variables of $I$. Let $I' = (V', A', \text{var}', E', \text{cpt}')$ be the CSP instance on variables $v'_1, \ldots, v'_k$ such that:

- $A'_i = A_i \cup \{a_i\}$ for all $1 \leq i \leq k$ and $A'_i = B_{i-k} \cup \{a_i\}$ for all $k + 1 \leq i \leq 2k$, where $|B| = |A| (1 \leq i \leq k)$. We can think of variables $v'_i$ and $v'_{i-k}$ as having the same domain except for the special value corresponding to $a_i$.
- For all $1 \leq i \leq k$, $a_i$ is incompatible with $a_{i-k}$ and compatible with all other points of $I'$. For all $k + 1 \leq i \leq 2k$, $a_i$ is incompatible with $a_{i-k}$ and compatible with all other points of $I'$. For all $1 \leq i \leq k$, this prevents both $a_i$ and $a_{i-k}$ to be part of the same solution. The idea here is that for any solution $S'$ to $I'$, for all $1 \leq i \leq k$, either the point of $S'$ in $v'_i$ or the point of $S'$ in $v'_{i-k}$ will be an original point from $I$, or a copy of an original point from $I$.
- For all $1 \leq i < j \leq k$, for all $a \in A_i$ for all $b \in A_j$ $\text{cpt}_i(a, b) = \text{cpt}(a, b)$. For all $1 \leq i \leq k$, $v'_i$ is a copy of $v'_i$ in $I'$ on $(A_i, A_{i-k} \setminus \{a_i\})$.

By construction, $I'$ has a solution if and only if $I$ has a solution, since (1) a solution to $I$ can be duplicated to produce a solution to $I'$, and (2) a solution to $I'$ without the assignments $a_i$ and after elimination of duplicates is a solution to $I$.

Furthermore, for all $1 \leq i \leq k$, $a_i$ is incompatible with only one other point in $I'$, either the point of $S'$ in $v'_i$ or the point of $S'$ in $v'_{i-k}$. So neither $V\text{-Middle}$ nor $V\text{-Side}$ occurs on $a_i$. So neither $V\text{-Middle}$ nor $V\text{-Side}$ appears in $I'$. Thus we can reduce any CSP instance $I$ to an arc-consistent CSP instance $I'$ in which neither $V\text{-Middle}$ nor $V\text{-Side}$ appears. It follows that $V\text{-Middle}$ and $V\text{-Side}$ are NP-Complete. \[\square\]

Let $V_2 = \{v_0, v_1\}$, $A_2 = \{a_0, a_1, a_2\}$, $\text{var}_2(a_0) = v_0$, $\text{var}_2(a_1) = \text{var}_2(a_2) = v_1$, $E_2 = \{(a_0, a_1), (a_0, a_2)\}$ and $\text{cpt}_2(a_0, a_1) = \text{cpt}_2(a_0, a_2) = T$. Let $V\text{-Middle}$ be the existential pattern $(V_2, A_2, \text{var}_2, E_2, \text{cpt}_2, \{a_0\})$ shown on the left of Fig. 20 and $V\text{-Side}$ be the existential pattern $(V_2, A_2, \text{var}_2, E_2, \text{cpt}_2, \{a_1\})$ shown on the right of Fig. 20.

Let $\text{ExpandedV}^+ = (V, A, \text{var}, E, \text{cpt}, \{a_0\})$ be the existential pattern shown in Fig. 21 and given by: $V = \{v_0, v_1, v_2\}$, $A = \{a_0, a_1, a_2, a_3\}$, $\text{var}(a_0) = v_0$, $\text{var}(a_1) = \text{var}(a_2) = v_1$, $\text{var}(a_3) = v_2$, $E = \{(a_0, a_1), (a_0, a_2), (a_0, a_3), (a_3, a_1)\}$, $\text{cpt}(a_0, a_2) = T$ and $\text{cpt}(a_0, a_1) = F$.

**Lemma 30.** $V\text{-Middle}$, $V\text{-Side}$ and $\text{ExpandedV}^+$ are NP-Complete.

**Proof.** Let $I = (V, A, \text{var}, E, \text{cpt})$ be an arc-consistent CSP instance on variables $v_1, \ldots, v_k$. Let $I' = (V', A', \text{var}', E', \text{cpt}')$ be the CSP instance on variables $v'_1, \ldots, v'_k$ such that:
Lemma 31. $V+-$ is NP-Complete.

Proof. Let $I$ be an arc-consistent CSP instance on variables $v_1, \ldots, v_k$ with at most one incompatibility edge in each constraint. Let $I'$ be the CSP instance on variables $v_1', \ldots, v_k'$ such that:

- $A_{v_i}' = A_{v_i} \cup \{a, b\}$ for all $1 \leq i \leq k$.
- For all $1 \leq i < j \leq k$, $a_i$ is incompatible with $a_j$. For all $1 \leq i < j \leq k$, for all $a \in A_{v_i}$, $a_i$ is incompatible with $b$ if $b$ is incompatible with a point $c \in A_{v_i}$ and $a_i$ is compatible with $b$ otherwise.
- For all $1 \leq i < j \leq k$, for all $a \in A_{v_i}$, for all $b \in A_{v_j}$, $a$ is compatible with $b$.

For all $1 \leq i \neq j \leq k$, we know that $a_i$ is compatible with a point $a' \in A_i$ if and only if $a'$ is compatible with all points in $A_i$. Since at most one point in $A_i$ is compatible with a point in $A_{v_i}$, and since $|A_{v_i}| \geq 2$, there always exists such a point $a' \in A_{v_i}$. So $I'$ is arc consistent. Furthermore, for all $1 \leq i \neq j \leq k$ we know that if $a_i$ is compatible with a point $a' \in A_i$, then $a'$ is compatible with all points in $A_i$. So for all $1 \leq i \leq k$, $V+-$ does not occur on $a_i$, and we cannot remove $a_i$ by neighbourhood substitution. So $V+-$ does not appear in $I'$ and the solvability of $I'$ is the same as that of $I$. So we can reduce any CSP instance with at most one incompatibility edge in each constraint $I$ to a CSP instance $I'$ in which $V+-$ does not appear. From Lemma 3, the set of CSP instances with at most one incompatibility edge in each constraint is NP-Complete. Thus $V+-$ is NP-Complete.

Lemma 32. $\exists T_3$ is NP-Complete.
Proof. Let $I$ be an arc-consistent CSP instance on variables $v_1, \ldots, v_k$. Let $I'$ be the CSP instance on variables $v'_1, \ldots, v'_{2k}$, $u'_1, \ldots, u'_n$ with compatibility function $\text{cpt}'$ such that:

- $A'_i = A_i \cup \{a, b\}$ for all $1 \leq i \leq k$, $A'_i = B_{n+k} \cup \{a, b\}$ for all $k + 1 \leq i \leq 2k$, where $|B_i| = |A_i| (1 \leq i \leq k)$, and
- $A'_i = \{c, d, e\}$ for $1 \leq i \leq k$. We can think of variables $v'_i$ and $v'_{i+k}$ as having the same domain except for the special values corresponding to $a_i, b$. The role of the variables $u'_i$ is to ensure that $a_i, b_i$ cannot be part of any solution to $I'$.

- For all $1 \leq i \leq 2k$, for all $1 \leq j \leq 2k$ such that $i$ and $j$ are not equal modulo $k$, $a_i$ and $b_i$ are compatible with all points of $A'_i$. For all $1 \leq i \leq k$: $a_i$ is compatible with $a_{i+k}$ and incompatible with all other points of $A'_{i+k}'; a_{i+k}$ is incompatible with all points in $A_i \setminus \{a_i\}$; $b_i$ is compatible with $b_{i+k}$ and incompatible with all other points of $A'_{i+k}'; b_{i+k}$ is incompatible with all points in $A_i \setminus \{b_i\}$.

- For all $1 \leq i \leq k$, $c_i$ is incompatible with $a_i, b_i, a_{i+k}, b_{i+k}$ and compatible with all other points of $A'$; $d_i$ is incompatible with all points in $(A'_{i+k}(\{b_i\}) \cup (A'_{i+k} \setminus \{b_{i+k}\}))$ and compatible with all other points in $A'$; and $e_i$ is compatible with all points in $(A'_{i} \setminus \{a_i\}) \cup (A'_{i+k} \setminus \{b_{i+k}\})$ and compatible with all other points in $A'$.

- For all $1 \leq i < j \leq k$, for all $a \in A_i$, for all $b \in A_j$, $\text{cpt}'(a, b) = \text{cpt}(a, b)$. For all $1 \leq i \leq k$, $v_{i+k}$ is a copy of $v_k$ in $I'$ on $(A_i, A'_{i+k} \setminus \{a_{i+k}, b_{i+k}\})$.

The points $a_i, b_i, a_{i+k}, b_{i+k}$ do not belong to any solution to the sub-instance of $I'$ on variables $v'_i, v'_{i+k}, u'_i$, whereas $c_i$ is compatible with all points in the original instance $I$. Furthermore, apart from these special points, variables $v'_i, v'_{i+k}$ are just copies of variable $v_i$. Thus, by construction, $I$ has a solution if and only $I'$ has a solution.

We will now show that the existential pattern $\exists I_1$ cannot occur on any $a_i$, with $1 \leq i \leq 2k$. Suppose that there is some $i$, with $1 \leq i \leq 2k$, such that the existential pattern $\exists I_1$ occurs on $a_i$. Let $v$ be the variable of $a_i$. Since $\exists I_1$ occurs on $a_i$, there is a variable $v'$ and a point $a'$ in $A_i$ such that $a_i$ and $a'$ are incompatible. By construction, $v'$ can only be one of the following variables: $v_{i+k}$ (or $v'_i$ if $i > k$) and $u'_i$ (or $u'_{i+k}$ if $i < k$). Since $\exists I_1$ occurs on $a_i$, there is a point in $A_i$ which is compatible with two different points in $A_i$. However, from the second and fourth bullet points we know that there is no point in $A_i$ compatible with two different points in $A_i' (A_i' \setminus \{a_i\})$, and from the third bullet point we also know that there is no point in $A_i$ compatible with two different points in $A_i'' (A''_i \setminus \{a_i\})$. So $\exists I_3$ cannot occur on $a_i$. So the existential pattern $\exists I_1$ cannot occur on any $a_i$, with $1 \leq i \leq 2k$.

Similarly, it is easy to verify that the existential pattern $\exists I_2$ does not occur on $c_i$ (for all $1 \leq i \leq k$). Hence $\exists I_2$ does not appear in $I'$. It follows that $\exists I_3$ is NP-complete. \hfill $\Box$

Let $\exists \text{Sub}_1 = (V, A, \text{var}, E, \text{cpt}, \{a_0\})$ be the existential pattern shown on the left of Fig. 23 and defined by $V = \{v_0, v_1, v_2\}, A = \{a_0, a_1, a_2, a_3\}, \text{var}(a_0) = v_0, \text{var}(a_1) = \text{var}(a_2) = v_1, \text{var}(a_3) = v_2, E = \{(a_0, a_1), (a_1, a_2)\}, \text{cpt}(a_0, a_1) = \text{cpt}(a_1, a_2) = T$ and $\text{cpt}(a_2, a_3) = F$.

**Lemma 33.** $\exists \text{Sub}_1$ is NP-Complete.

Proof. Let $I$ be an arc-consistent binary CSP instance on variables $v_1, \ldots, v_n$, where $n > 3$. Let $I'$ be the CSP instance on variables $v'_1, \ldots, v'_n$ with compatibility function $\text{cpt}'$ such that:

- $A'_i = A_i \cup \{a, b\}$ for all $j \in \{1, \ldots, i-1, i+1, \ldots, n\}$ for all $1 \leq i \leq n$.
- For all $1 \leq i < j \leq n$, for all $a \in A_i$, for all $b \in A_j$, $\text{cpt}'(a, b) = \text{cpt}(a, b)$.

It is easy to verify that none of the points $a_i, b_i$ belong to a solution to any 4-variable sub-instance of $I'$. This implies that the solutions to $I'$ are exactly the solutions to $I$.

To complete the proof, it remains to show that for each $1 = 1, \ldots, n, \exists \text{Sub}_1$ does not occur on $a_i$ in $I'$. Let $v'_i, v'_j, v'_k$ be any three distinct variables in $I'$. The point $a_i$ is only compatible with $b_j$ in $A_i'$ which is only compatible with $b_k$ in $A_i$. Since $b_j$ is compatible with all points in $A_i$, the existential pattern $\exists \text{Sub}_1$ does not occur on $a_i$ in $I'$. \hfill $\Box$

Let $\exists I_3 = (V, A, \text{var}, E, \text{cpt}, \{a_0\})$ be the existential pattern shown on the right of Fig. 23 and defined by $V = \{v_0, v_1, v_2\}, A = \{a_0, a_1, a_2, a_3\}, \text{var}(a_0) = \text{var}(a_1) = \text{var}(a_2) = v_0, \text{var}(a_3) = v_1, \text{var}(a_4) = v_2, E = \{(a_0, a_4), (a_1, a_3), (a_1, a_4), (a_2, a_1), (a_2, a_4), \text{cpt}(a_1, a_3) = \text{cpt}(a_1, a_4) = \text{cpt}(a_2, a_4) = \text{cpt}(a_2, a_3) = T$ and $\text{cpt}(a_0, a_4) = \text{cpt}(a_2, a_3) = F$.

**Lemma 34.** $\exists I_3$ is NP-Complete.

Proof. Let $I = (V, A, \text{var}, E, \text{cpt})$ be an arc-consistent binary CSP instance on variables $v_1, \ldots, v_n$. We will construct an equivalent instance $I'$ in which we add an assignment $a_i$ for each variable so that $\exists I_3$ does not occur on $a_i$. For each such point $a_i$, we will also add a 3-variable gadget to prevent $a_i$ from being part of a solution. Let $I' = (V', A', \text{var}', E', \text{cpt}')$ be the CSP instance on variables $v'_1, \ldots, v'_n, w'_1, \ldots, w'_n, x'_1, \ldots, x'_n, y'_1, \ldots, y'_n$ such that:

- $v'_i, v'_j, v'_k$ are}\)
Let \( P \) be a singleton existential pattern on four variables. Then \( P \) is either intractable or is a subpattern of one of the existential patterns \( X \) can. It follows that the solutions to \( I' \) are exactly the solutions to \( I \). Hence \( \exists T_4 \) is NP-complete.

7.3. A dichotomy for two-constraint existential patterns

**Definition 19.** We say that an existential pattern \( P \) is irreducible if we cannot apply merging or dp-elimination to \( P \).

Let \( X_1 = (V, A, \text{var}, E, \text{cpt}, \{a\}) \) be the following existential pattern (shown on the left of Fig. 24): \( V = \{v_0, v_1, v_2\}, A = \{a, b, c, d\}, \text{var}(\alpha) = \text{var}(\beta) = v_0, \text{var}(c) = v_1, \text{var}(d) = v_2, E = \{(a, c), (b, c), (b, d)\}, \text{cpt}(a, c) = \text{cpt}(b, d) = F \) and \( \text{cpt}(b, c) = T \).

Let \( X_2 = (V, A, \text{var}, E, \text{cpt}, \{a\}) \) be the following existential pattern (shown in the middle of Fig. 24): \( V = \{v_0, v_1, v_2\}, A = \{a, b, c, d\}, \text{var}(\alpha) = \text{var}(\beta) = v_0, \text{var}(c) = \text{var}(d) = v_1, E = \{(a, b), (a, c), (b, d), (c, d)\}, \text{cpt}(a, b) = \text{cpt}(c, d) = F \) and \( \text{cpt}(a, c) = \text{cpt}(b, d) = T \).

Let \( X_3 = (V, A, \text{var}, E, \text{cpt}, \{a\}) \) be the following existential pattern (shown on the right of Fig. 24): \( V = \{v_0, v_1, v_2\}, A = \{a, b, c, d\}, \text{var}(\alpha) = \text{var}(\beta) = v_0, \text{var}(c) = v_1, \text{var}(d) = v_2, E = \{(a, b), (b, c), (b, d)\}, \text{cpt}(a, b) = \text{cpt}(c, d) = F \).

We say that an existential pattern is a singleton existential pattern if its set of existential points is a singleton. We first characterise the tractability of irreducible singleton 2-constraint existential patterns. This will then directly lead to a dichotomy for general existential patterns.

**Proposition 1.** Let \( P = (V, A, \text{var}, E, \text{cpt}, \{a\}) \) be an irreducible singleton existential pattern on two constraints. Then \( P \) is tractable if and only if \( P \) is a sub-pattern of one of the existential patterns \( X_1, X_2, X_3 \).

**Proof.** \( \Rightarrow \): Let \( P = (V, A, \text{var}, E, \text{cpt}, \{a\}) \) be a tractable irreducible existential pattern on two constraints. A two-constraint existential pattern involves either three or four variables. From Lemma 28 and Theorem 1, all potentially-tractable irreducible singleton existential patterns on four variables are sub-patterns of \( X_3 \). Therefore we only need to consider two-constraint existential patterns on three variables.

By Lemma 28 and Theorem 1, we only need to consider patterns \( P \) such that the corresponding non-existential pattern \( P' = (V, A, \text{var}, E, \text{cpt}) \) is a sub-pattern of one of \( T_1, T_2, T_3, T_4, T_5 \).

If \( P' \) is a sub-pattern of \( T_1 \), then the irreducible singleton 3-variable existential pattern \( P \) either contains one of \( V \text{-Side}, V \text{-Middle}, V^+ \) or \( 3 \text{sub} T_1 \) or is a subpattern of \( X_1 \). Hence, by Lemmas 29, 31 and 33, \( P \) is either intractable or is a subpattern of \( X_1 \).

If \( P' \) is a sub-pattern of \( T_2 \), then the irreducible singleton 3-variable existential pattern \( P \) either contains one of \( V \text{-Side}, V \text{-Middle}, \text{Expanded} V^+ \) or \( 3 \text{sub} T_1 \) or is a subpattern of \( X_1 \). Hence, by Lemmas 29 and 30, Lemmas 31 and 33 and Lemma 34, \( P \) is either intractable or is a subpattern of \( X_1 \) or \( X_3 \).

If \( P' \) is a sub-pattern of \( T_3 \), then the irreducible singleton 3-variable existential pattern \( P \) either contains one of \( V \text{-Side}, V \text{-Middle}, \text{Expanded} V^+ \) or \( 3 \text{sub} T_1 \) or is a subpattern of \( X_1 \) or \( X_3 \). Hence, by Lemma 30, Lemma 31, Lemma 33 and Lemma 34, \( P \) is either intractable or is a subpattern of \( X_1 \) or \( X_3 \).
If \( P \) is a sub-pattern of \( T_3 \), then the irreducible singleton 3-variable existential pattern \( P \) either contains \( V+\) or is a subpattern of \( X_1 \) or \( X_2 \). Hence, by Lemma 31, \( P \) is either intractable or is a subpattern of \( X_1 \) or \( X_2 \).

\[\begin{align*}
\text{Proof of tractability of } X_1: & \text{ Let } I = (V, A, \text{ var, } E, \text{ cpt}) \text{ be an arc-consistent CSP instance such that } X_1 \text{ does not appear in } I. \text{ So } \\
& \forall v_i \in V, \exists a_i \in A_i \text{ such that } X_1 \text{ does not occur on } a_i. \text{ Suppose that we have a partial solution } S_k = \{s_1 \in A_{i_1}, \ldots, s_k \in A_{i_k}\}, \text{ with } 0 \leq k < |V|. \text{ If } a_{k+1} \text{ is compatible with all } s_i \text{ for } 1 \leq i \leq k, \text{ then } S_k \cup a_{k+1} \text{ is a partial solution for variables } (v_1, \ldots, v_{k+1}). \text{ Suppose that for some } 1 \leq i \leq k, \text{ we have that } s_i \text{ and } a_{k+1} \text{ are incompatible. By arc-consistency, we know there is } b \in A_{k+1} \text{ such that } s_i \text{ and } b \text{ are compatible. Since } X_1 \text{ does not occur on } a_{k+1}, \text{ } b \text{ is compatible with all points in } A \setminus (A_{i_1} \cup A_{k+1}), \text{ in particular all } s_j \text{ for } j \neq i. \text{ So } b \text{ is compatible with all points in } S_k. \text{ So } S_k \cup b \text{ is a partial solution for variables } (v_1, \ldots, v_{k+1}). \text{ So if we have a partial solution for } I \text{ on } k \text{ variables, then we also have a partial solution for } I \text{ on } k+1 \text{ variables. Hence, assuming } A \neq \emptyset, \text{ there is always a solution for } I. \text{ So } X_2 \text{ is tractable.}
\end{align*}\]

\[\begin{align*}
\text{Proof of tractability of } X_2: & \text{ Let } I = (V, A, \text{ var, } E, \text{ cpt}) \text{ be an arc-consistent CSP instance such that } X_2 \text{ does not appear in } I. \text{ So } \\
& \forall v_i \in V, \exists a_i \in A_i \text{ such that } X_2 \text{ does not occur on } a_i. \text{ Suppose that we have a partial solution } S_k = \{s_1 \in A_{i_1}, \ldots, s_k \in A_{i_k}\}, \text{ with } 0 \leq k < |V|. \text{ Let } \overline{Y} \text{ be the set } i \leq k \text{ such that } s_i \text{ and } a_{k+1} \text{ are compatible and let } \overline{Y} \text{ be the set } i \leq k \text{ such that } s_i \text{ and } a_{k+1} \text{ are compatible. By arc consistency, we know that } a_{k+1} \text{ is compatible with } t_j \text{ and } s_i \text{ is compatible with } s_j. \text{ Hence, assuming } A \neq \emptyset, \text{ there is always a solution for } I. \text{ So } X_2 \text{ is tractable.}
\end{align*}\]

7.4. The main dichotomy

We can now combine Proposition 1 with Theorem 1 to obtain a complete dichotomy for irreducible 2-constraint existential patterns.

**Theorem 2.** Let \( P = (V, A, \text{ var, } E, \text{ cpt, } e) \) be an irreducible existential pattern on two constraints. Then \( \text{CSP}_{AC}(P) \) is solvable in polynomial-time if \( P \) is a sub-pattern of one of the patterns \( T_1, T_2, T_3, T_4, X_1, X_2, X_3; \) if not \( \text{CSP}_{AC}(P) \) is NP-complete.

**Proof.** We first make the observation that for non-existential patterns, i.e. patterns \( P \) for which \( e = \emptyset \), \( \text{CSP}_{AC}(P) \) is solvable in polynomial time if and only if \( \text{CSP}(P) \) is solvable in polynomial time, since non-existential patterns cannot be introduced by establishing arc consistency. Thus the case \( e = \emptyset \) corresponds exactly to the dichotomy for non-existential patterns given in Theorem 1. Note that the patterns \( T_1 \) and \( 2I \) are equivalent to sub-patterns, respectively, of \( X_2 \) and \( X_3 \) which is why we do not explicitly mention them in the statement of the theorem.

The case \( |e| = 1 \) corresponds exactly to Proposition 1. For the case \( |e| > 1 \), by Lemma 28, we only need to consider existential versions of sub-patterns of \( T_1, T_2, T_4, T_3, 2I \). But all existential patterns \( P \) with \( |e| > 1 \) which are sub-patterns of one of \( T_1, T_2, T_3, T_4, T_5, 2I \) must contain either \( V+ \) or \( V \)-Side and hence are NP-complete by Lemmas 29 and 31.

8. Conclusion

We have investigated the computational complexity of classes of binary CSP instances defined by forbidding 2-constraint patterns. We have given a dichotomy for irreducible 2-constraint patterns which has brought to light several novel tractable classes.

One avenue for future research is to investigate the possible generalisations of the seven tractable classes defined by forbidding patterns \( T_1, T_2, T_3, T_4, X_1, X_2 \) or \( X_3 \). Possible generalisations include the addition of costs, adding extra constraints to the patterns and replacing binary constraints by constraints of arbitrary arity. Concerning general-arity constraints, two distinct generalisations of BTP have been proposed \cite{17,11}, along with a generalisation of the notion of microstructure \cite{18}. 

New and interesting tractable classes may exist which can only be defined by forbidding more than one irreducible pattern. It is an interesting open question whether such classes exist. In this paper we have limited our study to single irreducible patterns, due to the sheer number of cases when multiple patterns are simultaneously forbidden.

Another interesting avenue for research is to enrich the language of forbidden patterns, as we have done for existential patterns. The fact that forbidding patterns only on one point per variable has allowed us to find a strict generalisation of the tractable class defined by the pattern $T_5$ indicates the validity of this approach. Recently it has been shown that further enriching patterns with a sequence of arbitrary quantifiers on all variables and values can lead to the discovery of new variable elimination rules [9].

References