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

Martin Cooper, Guillaume Escamocher

Open Archive TOULOUSE Archive Ouverte (OATAO)

Characterising the complexity of constraint satisfaction problems defined by 2-constraint forbidden patterns[☆]

Martin C. Cooper [a], Guillaume Escamocher [b,*]

[a] *IRIT, University of Toulouse, France*
[b] *INSIGHT, University College Cork, Ireland*

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.

1. 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 [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 [4,3,2] or on the constraint (hyper)graph [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 [13,15]. A dichotomy has even been discovered for classes of binary CSP instances defined by forbidding configurations of incompatibilities [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 [13]: a binary CSP instance on variables $X_1, \ldots, X_n$ satisfies the
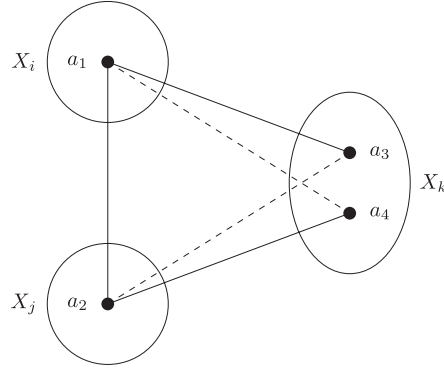
Fig. 1. Pattern forbidden by the broken-triangle property.

broken-triangle property if $\forall i < j < k \in \{1, \ldots, n\}$, whenever the assignments $a_1 = \langle X_i, a \rangle$, $a_2 = \langle X_j, b \rangle$, $a_3 = \langle X_k, c \rangle$, $a_4 = \langle X_k, d \rangle$ 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_4), (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.

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 $a_1 = \langle X_i, a \rangle$, $a_2 = \langle X_j, b \rangle$, $a_3 = \langle X_k, c \rangle$ to three distinct variables, whenever the pairs (a_1, a_2), (a_2, a_3) are both compatible, the third pair (a_1, a_3) is also compatible. The latter class consists of all binary CSP instances in which for all triples of assignments $a_1 = \langle X_i, a \rangle$, $a_2 = \langle X_j, b \rangle$, $a_3 = \langle X_k, c \rangle$ to three distinct variables, whenever the pairs (a_1, a_2), (a_2, a_3) are both incompatible, the third pair (a_1, a_3) is also incompatible. This property is satisfied, for example, by instances consisting of unary constraints and non-overlapping AllDifferent constraints (since $a = b \wedge b = c \Rightarrow a = c$). The class of binary CSP instances satisfying this negative-transitivity property has been generalised to a large tractable class of optimisation problems involving cost functions of arbitrary arity [15,16].

Any class of instances defined by a forbidden pattern is necessarily recognisable in polynomial time by a simple exhaustive search for the pattern.

The present paper provides an essential first step towards the identification of all tractable classes defined by forbidding patterns, namely a dichotomy for the special case of 2-constraint forbidden patterns. This investigation of small forbidden patterns has already allowed us to uncover several novel tractable classes. We expect that this dichotomy will in the future represent an important base case in a more general characterisation of tractable classes of constraint problems defined by local structure. The tractable classes described in this paper may prove to be the inspiration for larger tractable classes of general-arity CSPs or may lead to the development of simplification rules for CSPs.

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
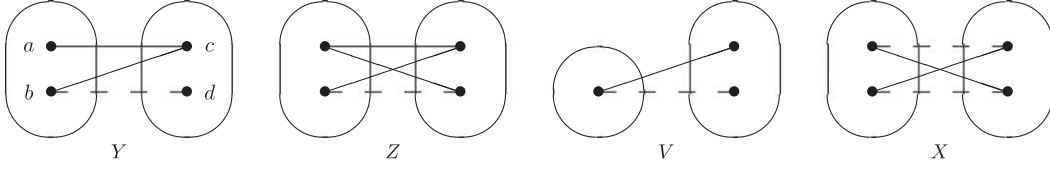
Fig. 2. Four patterns.

of a value to a variable, i.e. a pair $a = \langle v, d \rangle$ 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 $\langle v, d \rangle$ 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 $\langle V, A, var, E, cpt \rangle$ comprising:

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

Definition 2. A *binary CSP instance* is a pattern $\langle V, A, var, E, cpt \rangle$ such that $E = \{\{a, b\} \mid (a, b) \subseteq A \times A, var(a) \neq var(b)\}$ (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 $|\overline{A}| = |V|$, $(\forall a, b \in \overline{A}, var(a) \neq var(b))$ and $(\forall e \in \binom{\overline{A}}{2} \cap E, cpt(e) = T)$?

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 $P = \langle V, A, var, E, cpt \rangle$ and a variable $v \in V$, we use A_v to denote the set of assignments $\{a \in A \mid var(a) = v\}$. The *constraint* on variables $v_1, v_2 \in V$ is the pattern $\langle \{v_1, v_2\}, A_{12}, var|_{A_{12}}, E_{12}, cpt|_{E_{12}} \rangle$ where $A_{12} = A_{v_1} \cup A_{v_2}$ and $E_{12} = \{\{a, b\} \mid a \in A_{v_1}, b \in A_{v_2}\} \cap E$. If $cpt(a, b) = T$ then the two assignments (points) a, b are *compatible* and $\{a, b\}$ is a *compatibility edge*; if $cpt(a, b) = F$ 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 $var(a) \neq var(b)$ 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 $\langle V, A, var, E, cpt \rangle$, we call the set $\{d \mid \langle v, d \rangle \in A\}$ of values that can be assigned to variable v the *domain* of v and the set $\{(d_1, d_2) \mid (\langle v_1, d_1 \rangle, \langle v_2, d_2 \rangle) \in A_{v_1} \times A_{v_2} \wedge cpt(\langle v_1, d_1 \rangle, \langle v_2, d_2 \rangle) = T\}$ 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 $cpt(a, b) = F$. The *constraint graph* of an instance $\langle V, A, var, E, cpt \rangle$ is $\langle V, H \rangle$, 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 $P = \langle V_P, A_P, var_P, E_P, cpt_P \rangle$ and $Q = \langle V_Q, A_Q, var_Q, E_Q, cpt_Q \rangle$, then $V_P \subseteq V_Q, A_P \subseteq A_Q, var_P = var_Q|_{A_P}, E_P \subseteq E_Q, cpt_P = cpt_Q|_{E_P}$.

merging Merging two points in P transforms P into Q: if $P = \langle V_P, A_P, var_P, E_P, cpt_P \rangle$ and $Q = \langle V_Q, A_Q, var_Q, E_Q, cpt_Q \rangle$, then $\exists a, b \in A_P$ such that $var_P(a) = var_P(b)$ and $\forall c \in A_P$ such that $\{a, c\}, \{b, c\} \in E_P, cpt_P(a, c) = cpt_P(b, c)$. Furthermore, $V_P = V_Q, A_Q = A_P \setminus \{b\}, var_Q = var_P|_{A_Q}, E_Q = E_P \cup \{\{a, x\} \mid \{b, x\} \in E_P\}$ and $cpt_Q(a, x) = cpt_Q(b, x)$ if $\{b, x\} \in E_P, cpt_Q(e) = cpt_P(e)$ for all other $e \in E_Q$.

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 $a, b \in A_{v_0}, c, d \in A_{v_1}$ such that $cpt(a, c) = cpt(b, c) = T$ and $cpt(b, d) = F$. 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 a merging followed by an extension.

Notation. Let P be a CSP pattern. We use $\mathrm{CSP}(\overline{P})$ to denote the set of binary CSP instances Q in which P does not occur.

Definition 4. A pattern P is *intractable* if $\mathrm{CSP}(\overline{P})$ is NP-complete. It is *tractable* if there is a polynomial-time algorithm to solve $\mathrm{CSP}(\overline{P})$.

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 $\mathrm{CSP}(\overline{P})$ defined by forbidding a pattern are hereditary: $I \in \mathrm{CSP}(\overline{P})$ and $I' \subseteq I$ (in the sense that I is an extension of I', according to Definition 3) together imply that $I' \in \mathrm{CSP}(\overline{P})$. Furthermore, if $I \in \mathrm{CSP}(\overline{P})$ and I' is isomorphic to I, then $I' \in \mathrm{CSP}(\overline{P})$. 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 $\langle V, A, var, E, cpt \rangle$ 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 $\langle V, A, var, E, cpt \rangle$, *arc consistency* consists in eliminating from A all assignments a for which there is some variable $v \neq var(a)$ in V such that $\forall b \in A_v, cpt(a, b) = F$ [1].

Given a CSP instance $\langle V, A, var, E, cpt \rangle$, if $var(a) = var(b)$ and for all variables $v \neq var(a), \forall c \in A_v, cpt(a, c) = T \Rightarrow cpt(b, c) = T$, 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 $\mathrm{CSP}(\overline{P})$ can introduce the forbidden pattern P.

We now consider two new simplification operations. They are simplification operations that 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 $\langle V, A, var, E, cpt \rangle$ with $v_1, v_2 \in V$. Suppose that there is a *fusion function* $f : A_{v_1} \to A_{v_2}$, such that $\forall u \in A_{v_1}$, 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 $\langle V', A', var', E', cpt' \rangle$ defined by $V' = (V \setminus \{v_1, v_2\}) \cup \{v\}, A' = A \setminus A_{v_2}, var'(u) = var(u)$ for all $u \in A' \setminus A_{v_1}$ and $var'(u) = v$ for all $u \in A_{v_1}, E' = \{\{p, q\} \in \binom{A'}{2} \mid var'(p) \neq var'(q)\}, cpt'(p, q) = cpt(p, q)$ if $p, q \in A' \setminus A_{v_1}, cpt'(u, q) = cpt(u, q) \wedge cpt(f(u), q)$ for all $u \in A_{v_1}$ and all $q \in A' \setminus A_{v_1}$.

Definition 7. Consider a CSP instance $\langle V, A, var, E, cpt \rangle$ with $v_1, v_2 \in V$ and a *hinge value* $a \in A_{v_1}$. Suppose that there is a *fusion function* $f : A_{v_1} \setminus \{a\} \to A_{v_2}$, such that $\forall u \in A_{v_1} \setminus \{a\}$, 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 $\langle V', A', var', E', cpt' \rangle$ defined by $V' = (V \setminus \{v_1, v_2\}) \cup \{v\}, A' = A \setminus \{a\}, var'(u) = var(u)$ for all $u \in A' \setminus (A_{v_1} \cup A_{v_2})$ and $var'(u) = v$ for all $u \in (A_{v_1} \setminus \{a\}) \cup A_{v_2}, E' = \{\{p, q\} \in \binom{A'}{2} \mid var'(p) \neq var'(q)\}, cpt'(p, q) = cpt(p, q)$ if $p, q \in A' \setminus (A_{v_1} \cup A_{v_2}), cpt'(u, q) = cpt(u, q) \wedge cpt(f(u), q)$ for all $u \in A_{v_1} \setminus \{a\}$ and all $q \in A' \setminus (A_{v_1} \cup A_{v_2}), cpt'(p, q) = cpt(a, q) \wedge cpt(p, q)$ for all $p \in A_{v_2}$ 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 iff 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. $\square$
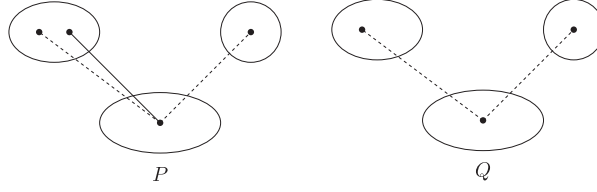
Fig. 3. Example of dp-elimination.

Fusion preserves solvability and the total number of assignments decreases by at least 1 (in fact, by $|A_{v_2}|$ in the case of a simple fusion). However, when solving instances $I \in \text{CSP}(\overline{P})$, 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 $P = \langle V_P, A_P, var_P, E_P, cpt_P \rangle$, 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 $I = \langle V, A, var, E, cpt \rangle$ 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 $P = \langle V_P, A_P, var_P, E_P, cpt_P \rangle$ and $Q = \langle V_Q, A_Q, var_Q, E_Q, cpt_Q \rangle$ be two patterns, such that P can be reduced to a sub-pattern of Q. Let $I = \langle V, A, var, E, cpt \rangle$ 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 (at least one of the two versions of) P 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. $\square$

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 $\text{CSP}(\overline{P})$. 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 $l_i^1 \vee \cdots \vee l_i^{c_i}$. We create the following CSP instance I':

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

- $\forall v_i'$ with $n + 1 \le i \le n + k$, c_{i-n} points $l_{i-n}^1, \ldots, l_{i-n}^{c_{i-n}}$ in $A_{v_i'}$.
- $\forall 1 \le i \le k, \forall 1 \le j \le c_i$, an incompatibility edge between the point $l_i^j \in A_{v_{n+i}'}$ and the point in $A_{v_1'}, \ldots, A_{v_n'}$ corresponding to the literal $\overline{l_i^j}$.

A solution to I' consists of a set of literals assigned true in a solution s to I together with for each clause a literal from this clause which is assigned true in s. Therefore, by construction, I' has a solution if and only if I has a solution. Furthermore, each time an incompatibility edge occurs in a constraint C, this constraint C is between a CSP variable v_i' representing the SAT1 variable v_i and another CSP variable v_{n+j}' representing the SAT1 clause C_j. Since v_i occurs at most once in C_j, there is only one incompatibility edge in C. So I' does not contain the pattern P. So we have reduced SAT1 to CSP($\overline{P}$), as required. □

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 $b \in A$ such that a is compatible with b (respectively such that a is incompatible with b).

Lemma 4. *Let P be a non-mergeable pattern. Then for every variable v in P, there is at most one point in A_v which is not explicitly incompatible.*

Proof. Suppose we have a pattern P such that there are two points a and b with $var(a) = var(b)$ such that neither a nor b is explicitly incompatible. So no point in the pattern is incompatible with either a or b. Hence, we can merge a and b, which is a contradiction. □

Let Z 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($\overline{Z}$), the set of CSP instances in which the pattern Z does not occur.

For $s, t \in \{1, 2, 3\}$, define the relation $R_{s,t} \subseteq \{1, 2, 3\}^2$ by

$$R_{s,t} = \{\langle u, v \rangle | (u = s \wedge v = t) \vee (u \ne s \wedge v \ne t)\}.$$

It is easy to verify that $R_{s,t}$ does not contain the pattern Z. Consider the 5-variable gadget with variables v_i, v_j, u_1, u_2, u_3, each with domain $\{1, 2, 3\}$, and with constraint relations $R_{k,k}$ on variables (v_i, u_k) ($k = 1, 2, 3$) and constraint relations $R_{1+(k \bmod 3),k}$ on variables (u_k, v_j) ($k = 1, 2, 3$). The joint effect of these six constraints is simply to impose the constraint $v_i \ne v_j$. Any instance $\langle V, E \rangle$ of 3-COLOURING, with $V = \{1, \ldots, n\}$, can be reduced to an instance of CSP($\overline{Z}$) 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 $1I$ be the pattern on two variables v and v' with points $a \in A_v$ and $b \in A_{v'}$ such that a and b are incompatible. $1I$ is a trivial tractable pattern, because any CSP instance not containing $1I$ contains only trivial constraints.

Lemma 6. *Let P be a pattern on one constraint. Then either P is reducible to a sub-pattern of $1I$, 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 $1I$. 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 $c \ne a$ in A_v and at most one other point $d \ne b$ in $A_{v'}$. If all three edges $\{a, d\}$, $\{c, b\}$ and $\{c, d\}$ 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 $1I$. 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. 1. P_1 and P_2 are sub-patterns of P. So if one of them is intractable, then P is intractable too, by Corollary 1.
2. Suppose that both P_1 and P_2 are tractable. So there are two polynomial algorithms A_1 and A_2 which solve CSP($\overline{P_1}$) and CSP($\overline{P_2}$), 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_1 occurs on variables u, v in I_1, 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. □
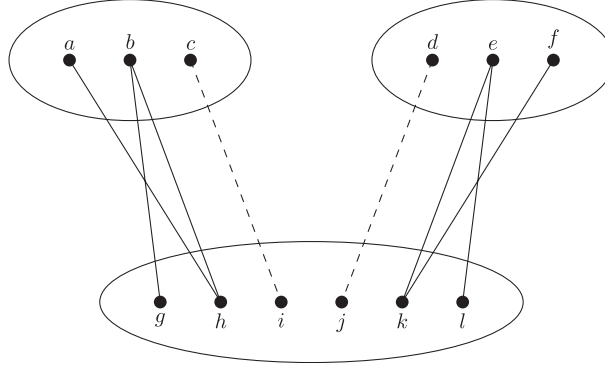
Fig. 4. The pattern $2V$.

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 $a, b, c \in A_{v_1}$, three points $d, e, f \in A_{v_2}$ and six points $g, h, i, j, k, l \in A_{v_0}$, such that a is compatible with h, b is compatible with g and h, c is incompatible with i, d is incompatible with j, e is compatible with k and l, f is compatible with l. The pattern $2V$ also has the associated structure $(a \neq b$ or $g \neq h)$ and $(e \neq f$ 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($\overline{2V}$). 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($\overline{2V}$) shows that $2V$ is intractable. $\quad\square$

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 ($i = 1, \ldots, T_5$) 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 $1I$, 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.

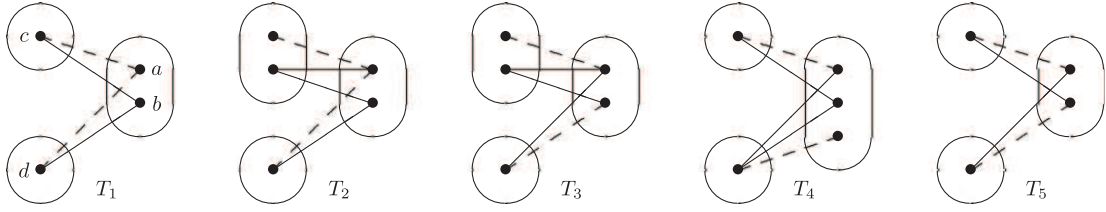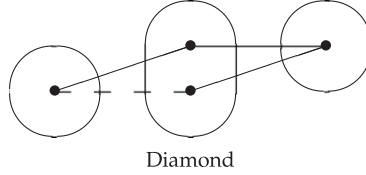**Fig. 5.** The set of tractable patterns T.

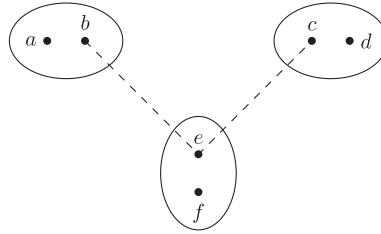Diamond

Fig. 6. The pattern Diamond.

Fig. 7. Incompatibility skeleton of type 1.

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 $T \cup \{2I\}$.*

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 $1I$. 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 incompatibility edge 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.
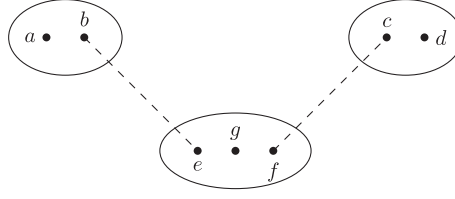
Fig. 8. Incompatibility skeleton of type 2.

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 e, 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_3.

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 $T \cup \{2I\}$.

6.2.2. Sufficient

$\Leftarrow$: We now give the tractability proofs for all patterns in $T \cup \{2I\}$. 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($\overline{T_1}$) 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 I 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.
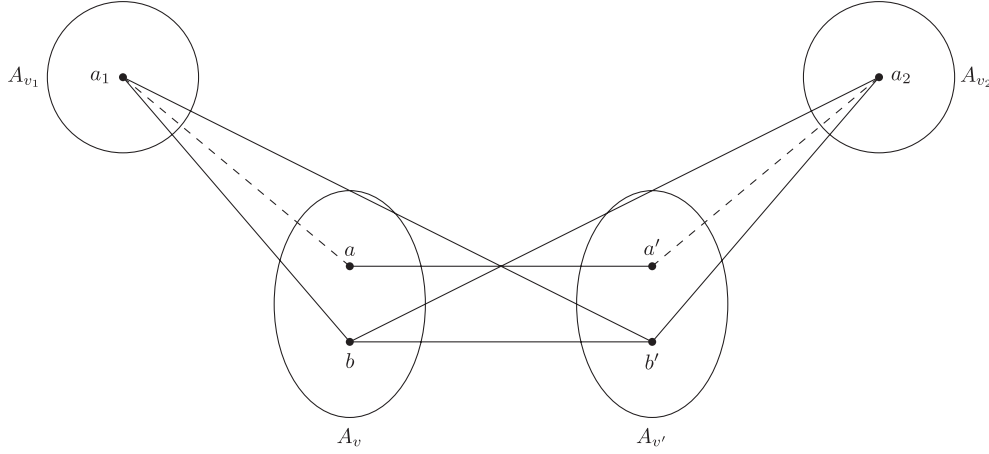
Fig. 9. Introduction of the pattern T_1.

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 $I \in \mathrm{CSP}(\overline{T_1})$ 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 $I \in \mathrm{CSP}(\overline{T_1})$. 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, 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. $\square$

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 $\geq$, $<$ and $>$ 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 $I \in \mathrm{CSP}(\overline{T_1})$*

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. $\square$

Lemma 11. *In a simplified instance $I \in \mathrm{CSP}(\overline{T_1})$, 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 $v_3 \neq v_1, v_2$ 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. $\square$

Lemma 12. *In a simplified instance $I \in \mathrm{CSP}(\overline{T_1})$, $\forall a, b, c, d \in A_{v_0}$, 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. $a < b < c \equiv d$ *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_2' such that $d < c < b$ for v_2'. Since $d < c$ for both v_2 and v_2', $v_2 = v_2'$ 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. $\square$

Corollary 2. *In a simplified instance $I \in \mathrm{CSP}(\overline{T_1})$, if for some (v_0, v_1), we have at least three equivalence classes in the order on A_{v_0} with respect to v_1 then:*

1. *The order on A_{v_0} with respect to v_1 is strict.*
2. *There is v_2 such that the order on A_{v_0} with respect to v_2 is the exact opposite to the order on A_{v_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_{v_0} with respect to v_3.*

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

Lemma 13. *In a simplified instance $I \in \mathrm{CSP}(\overline{T_1})$, $\forall a, b, c, d \in A_{v_0}$, 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 $c \equiv d$ 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_2 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. $\square$

Lemma 14. *In a simplified instance $I \in \mathrm{CSP}(\overline{T_1})$, 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 both 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 $a_1 < a_2 < \cdots < a_d$ for (v, v'). So we have $(a_1', a_2', \ldots, a_{d-1}')$ 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 and a_{i-1}' are compatible. So, by Lemma 10.1 we have $a_1' > a_2' > \cdots > a_{d-1}'$ for v. Moreover, since a_1 is incompatible with a_1', a_1 is incompatible with all a_i' for $1 \leq i < d$. By arc consistency, we have a_0' such that a_1 and a_0' are compatible. So we have $a_0' > a_1' > a_2' > \cdots > a_{d-1}'$. 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. $\square$

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 $I \in \mathrm{CSP}(\overline{T_1})$, 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 $a_1 < a_2 < \cdots < a_d$ for v' and the points in $A_{v'}$ can be denoted $b_1 < b_2 < \cdots < b_d$ for v. We will show that we can perform a simple fusion of v and v' with fusion function f given by $f(a_i) = b_{d+1-i}$ $(i = 1, \ldots, d)$.

Claim: $\forall 1 \leq i \leq d$, $\{b_{d+1-i}, b_{d+1-i+1}, \ldots, b_d\}$ is the exact set of points compatible with a_i.
If we have $a_i < a_j$ 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+1-i} and a_i are in S'. Let b be the point of S in v'. If $b_{d+1-i} = b$, 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 b_{d+1-i}. Otherwise, let $v_c = var(c)$. So $v_c \neq v$. From the above claim we

have $b_{d+1-i} < b$ for v. So $b \leq b_{d+1-i}$ for v_c. 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 $\text{CSP}(\overline{T_1})$, 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 $<$ 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. $\square$

Therefore, from now on, in a fusion-simplified instance $I \in \text{CSP}(\overline{T_1})$, 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 $I \in \text{CSP}(\overline{T_1})$, $\forall v$, if there is v' such that there is only one winner for (v, v'), then v is one-winner. Similarly, if there is 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 $v_1 \neq v_2$ with $a \equiv b < c$ for v_1, $d < e \equiv f$ for v_2, $a \neq b$ and $e \neq f$. If $d \neq c$, then from Lemma 13, we have $a \equiv b \equiv d \equiv c$ for v_1 and $d < e \equiv f \equiv c$ for v_2. So $d < c$ for both v_1 and v_2 with $v_1 \neq v_2$ (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_2. From Lemma 13 we have $c < e \equiv f \equiv a \equiv b$ for v_2. Since we have $a \equiv b$ for both v_1 and v_2, by Lemma 10.2 there is a different variable v_3 such that $a < b$ for v_3. Since $a < c$ for v_1, $c \leq a$ for v_3. So $c < b$ for v_3. So $c < b$ for both v_2 and v_3 with $v_2 \neq v_3$. This is impossible by Lemma 10.3. So we have the lemma. $\square$

Corollary 3. *In a simplified instance $I \in \text{CSP}(\overline{T_1})$, $\forall v$, 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. $\square$

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 $v_a \in E$, there is only one winner a for v_b in v_a. Similarly, there is only one winner b for v_a in v_b. We can perform a complex fusion of v_a and v_b with hinge value a and fusion function the constant function $f = b$.

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 $\text{CSP}(\overline{T_1})$ does not create the forbidden pattern.*

Proof. Suppose that (c, c') and (d, d') are corresponding pairs of points during this fusion, with $c \neq d \in A_{v_a}$ and $c' \neq d' \in A_{v_b}$. 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. $\square$

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

Lemma 18. *In a simplified instance $I \in \text{CSP}(\overline{T_1})$, let $v_a, v_b \in F$ (where F is the set of one-loser variables of I) be such that there is a non-trivial constraint between v_a and v_b. Let $a \in A_{v_a}$ and $b \in A_{v_b}$ be such that a is incompatible with b. If $a' \in A_{v_a}$ 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-loser variable, we know that all points in A_{v_a} other than a 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. $\square$

Lemma 19. *Let v_a, v_b both be one-loser variables in a simplified instance $I \in \text{CSP}(\overline{T_1})$ such that $a \in A_{v_a}$ and $b \in A_{v_b}$ 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 $f = b$ without introducing the forbidden pattern T_1.*
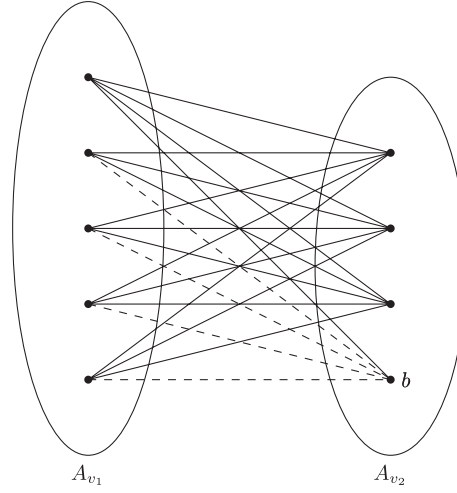
Fig. 10. Constraint between a one-winner variable v_1 and a one-loser variable v_2.

Proof. It follows from Lemma 18 that we only need to consider solutions containing a or b. We can therefore perform a complex fusion of v_a and v_b with hinge value a and fusion function the constant function $f = b$.

In all pairs (c, c') of corresponding points in this fusion, we must have either $c = a$ or $c' = b$. Suppose that (c, c') and (d, d') are corresponding pairs of points during the fusion, with $c \neq d \in A_{v_a}$ and $c' \neq d' \in A_{v_b}$. Without loss of generality, we can assume that $c = a$ and $d' = b$. But we know that a was incompatible with b. From Lemma 9 we can deduce that the fusion does not introduce the pattern T_1. □

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 $I \in \mathrm{CSP}(\overline{T_1})$. 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 incompatible with all points of $A_{v'}$ but one (since $v' \in E$ is a one-winner 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 $C = \{C_1, C_2, \ldots, C_f\}$ such that:
  - Each clause is a disjunction of literals, with a literal being in this case of the form $v_i = a_j$.
  - $\forall i, j, p, q((v_i = a_j) \in C_p) \wedge ((v_i = a_j) \in C_q) \Rightarrow p = q$.

Lemma 21. *$\mathrm{CSP}(\overline{T_1})$ 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 $I \in \mathrm{CSP}(\overline{T_1})$ in polynomial time. Thus suppose we have a fusion-simplified instance in $\mathrm{CSP}(\overline{T_1})$. 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
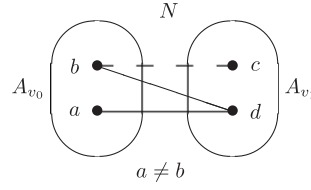
Fig. 11. The gadget N.

[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 $v = b_i \Rightarrow v_i = a_i$, for all values b_i in the domain of v, by the clause $(v_1 = a_1) \vee \ldots \vee (v_d = a_d)$. 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 $\mathrm{CSP}(\overline{T_1})$ can be reduced to NOOSAT. $\square$

The constraints in NOOSAT are convex when viewed as $\{0, \infty\}$-valued cost functions on the assignment-sets $\{\langle v_1, a_1 \rangle, \ldots, \langle d_d, a_d \rangle\}$ (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_1} which are compatible with a are also compatible with b, then we can remove a by neighbourhood substitution. So, assuming that neighbourhood 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 $v_3 \neq v_0$. By arc consistency, there is $h \in A_{v_3}$ such that h is compatible with a. If b and h are incompatible, then we have the forbidden pattern T_2 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}. So, if we have the gadget N, then b is compatible with all the points of the instance outside v_0, v_1.

Definition 12. A constraint C between two variables v and v' is *functional* from v to v' if $\forall a \in A_v$, 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 $\mathrm{CSP}(\overline{T_2})$, 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_6 to v_4.*

Proof. By symmetry, it suffices to prove functionality from v_5 to v_4. We suppose we have the gadget V^-. Let $d \in A_{v_5}$ 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_5} 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_5} and A_{v_6} compatible with f is compatible with only one point of A_{v_4}. So each point in A_{v_5} 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_5} 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. $\square$

Lemma 23. *In a connected component of the constraint graph containing V^- of an instance from $\mathrm{CSP}(\overline{T_2})$, 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.
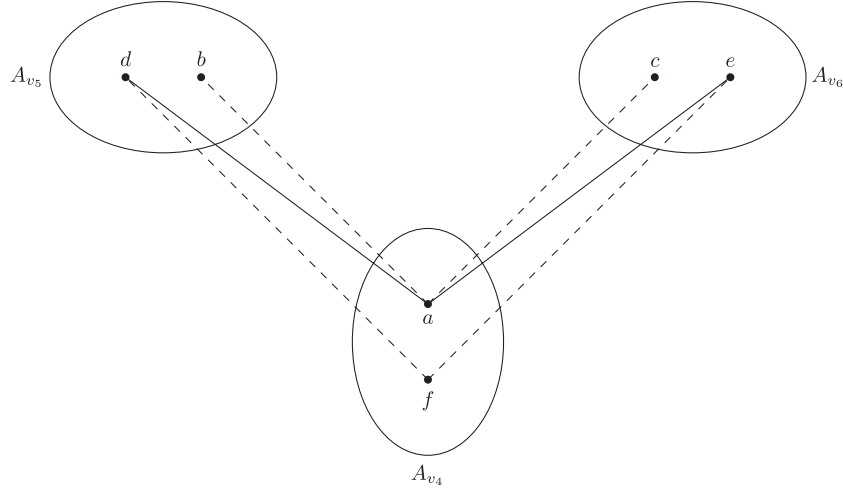
$P(\{v_4, v_5\})$ is true from [Lemma 22](#).

Fig. 12. The three variables v_4, v_5 and v_6.

Let V_{all} be the set of all variables of the connected subgraph of the constraint graph containing V^-. Let V be a maximum (with respect to inclusion) subset of V_{all} for which $P(V)$ is true. Let $V' = V_{\text{all}} \setminus V$. 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 $d \in A_v$ and $e \in A_{v'}$ 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 $f \in A_{v''}$ such that d and f are incompatible. So d, e and f form the gadget V^-. From Lemma 22 we know $C(v, v')$ is functional. So $P(V)$ is true for all subsets of V_{all}. $\square$

Lemma 24. *In an instance from* $\text{CSP}(\overline{T_2})$, $\forall v$ *such that* v *is in a connected component of the constraint graph containing* V^-, *all points in* A_v *are weakly incompatible with the exact same set of variables.*

Proof. Let $a \in A_v$ 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_v can be compatible with only one point in $A_{v'}$. We can assume, by elimination of single-valued variables, that there are at least two points in $A_{v'}$, so every point in A_v 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 $c \in A_{v'}$ 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_v, namely a, so b is incompatible with c. So every point in A_v is weakly incompatible with v'.

So $\forall (v, v')$, $a \in A_v$ weakly incompatible with $v' \Rightarrow \forall b \in A_v$, b weakly incompatible with v'. $\square$

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^- *of an instance from* $\text{CSP}(\overline{T_2})$, $\forall v, v'$, *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 $(v_0, v_1, v_2, \ldots, v_{k-1}, v_k, v_{k+1})$ with $v_{k+1} \neq v_{k-1}$. From Lemma 24 we have a path of incompatibility $(a_0 \in A_{v_0}, a_1 \in A_{v_1}, \ldots, a_k \in A_{v_k}, a_{k+1} \in A_{v_{k+1}})$. So $\forall i \in \{1, \ldots, k\}$, a_{i-1}, a_i and a_{i+1} form the gadget V^-. 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'. $\square$

Variables which are connected to at most one other variable in the constraint graph can be removed from the instance I since, by arc consistency, any solution on the remaining variables can be extended to a solution for I. Once we have removed all such variables, for each connected component of the constraint graph, we only have to set an initial variable v_0 and see if the q chains of implications (with q being the number of points in A_{v_0}) lead to a solution. Since this is clearly polynomial-time, the pattern T_2 is tractable.

Proof of tractability of T_3. Consider an instance from $\text{CSP}(\overline{T_3})$.

Suppose that the gadget N, shown in Fig. 11, is a sub-pattern of the instance and let e be a point in A_{v_2}, with $v_2 \neq v_0, v_1$. If e is compatible with b but not with a, then we have the forbidden pattern T_3. So if b is compatible with a point outside of A_{v_1}, then a is also compatible with the same point.
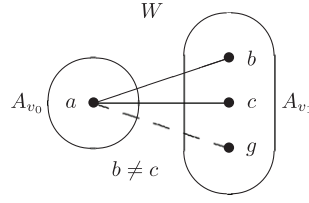
Fig. 13. The gadget W.

Let S be a solution containing b. Let f be the point of S in A_{v_1}. 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 $I \in \mathrm{CSP}(\overline{T_3})$, 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 $\mathrm{CSP}(\overline{T_4})$.

Let W be the gadget shown in Fig. 13: two variables v_0 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_2}, 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_0} 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 e and d 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 $1I$, 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 $e \subseteq A_v$ for some distinguished variable v. 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 $f_x(a) \in A_x$ such that there is no occurrence of P in I in which the existential point a maps to $f_x(a)$.
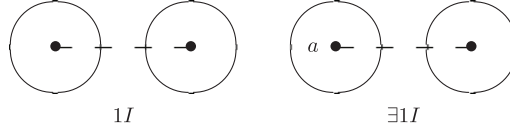
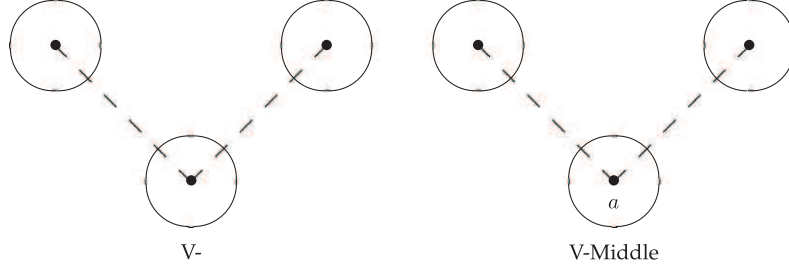**Fig. 14.** A simple pattern $1I$ and an existential version $\exists 1I$ of the same pattern.

Fig. 15. A pattern V- and an existential version V-Middle of the same pattern.
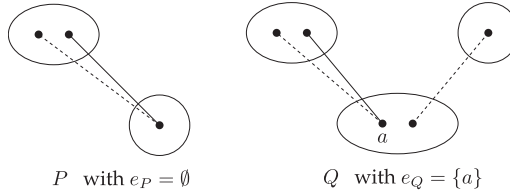

Fig. 16. Example of extension of an existential pattern P to produce the existential pattern Q.

As a simple example, consider the pattern $1I$ and its existential version $\exists 1I$ shown in Fig. 14. Forbidding $1I$ in an instance means that all points are compatible with all other points in the instance, whereas forbidding $\exists 1I$ 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- and its existential version V-Middle shown in Fig. 15. Forbidding V- 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- 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.

When e is not a singleton, forbidding the existential pattern P means that for all variables x in I, there is an injective function $f_x : e \rightarrow A_x$ such that there is no occurrence of P in I in which each $p \in e$ maps to $f_x(p)$. An existential pattern $\langle V, A, var, E, cpt, e \rangle$ is thus a pattern $\langle V, A, var, E, cpt \rangle$ to which we add a set of existential points $e \subseteq A_v$ for some distinguished variable $v \in V$. If $e = \emptyset$, then the existential pattern is equivalent to the (non-existential) pattern $\langle V, A, var, E, cpt \rangle$. 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 $\mathrm{CSP}(\overline{Q})$ than $\mathrm{CSP}(\overline{P})$. 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 $P = \langle V_P, A_P, var_P, E_P, cpt_P, e_P \rangle$ and $Q = \langle V_Q, A_Q, var_Q, E_Q, cpt_Q, e_Q \rangle$, then $V_P \subseteq V_Q, A_P \subseteq A_Q, var_P = var_Q|_{A_P}, E_P \subseteq E_Q, cpt_P = cpt_Q|_{E_P}$, and $e_P \subseteq e_Q$. We give an example in Fig. 16.
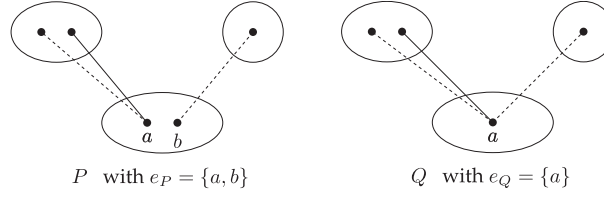
Fig. 17. Example of merging in an existential pattern P to produce the existential patten Q.

merging Merging two points in P transforms P into Q: if $P = \langle V_P, A_P, var_P, E_P, cpt_P, e_P \rangle$ and $Q = \langle V_Q, A_Q, var_Q, E_Q, cpt_Q, e_Q \rangle$, then $\exists a, b \in A_P$ such that $var_P(a) = var_P(b)$, $a \in e_P \Leftrightarrow b \in e_P$ and $\forall c \in A_P$ such that $\{a, c\}, \{b, c\} \in E_P$, $cpt_P(a, c) = cpt_P(b, c)$. Furthermore, $V_P = V_Q$, $A_Q = A_P \setminus \{b\}$, $var_Q = var_P|_{A_Q}$, $E_Q = E_P \cup \{\{a, x\} \mid \{b, x\} \in E_P\}$, $cpt_Q(a, x) = cpt_Q(b, x)$ if $\{b, x\} \in E_P$, $cpt_Q(e) = cpt_P(e)$ for all other $e \in E_Q$, and $e_Q = e_P \setminus \{b\}$. We give an example in Fig. 17.

It follows from Definition 14 that an occurrence of an existential pattern P in an existential pattern Q can also be viewed as the existence of an *occurrence-function* $f : A_P \rightarrow A_Q$ such that

1. $\forall a, b \in A_P$, $var_Q(f(a)) = var_Q(f(b))$ if and only if $var_P(a) = var_P(b)$.
2. $\forall a, b \in A_P$ such that $\{a, b\} \in E_P$, $\{f(a), f(b)\} \in E_Q$ and $cpt_Q(f(a), f(b)) = cpt_P(a, b)$.
3. $\forall a \in A_P$ such that $a \in e_P$, $f(a) \in e_Q$.

Definition 15. Let $I = \langle V, A, var, E, cpt \rangle$ be a CSP instance. Let v be a variable in V. Let $S \subseteq A_v$. Let $P = \langle V_P, A_P, var_P, E_P, cpt_P, e_P \rangle$ be an existential pattern.

If P occurs in the existential pattern $\langle V, A, var, E, cpt, S \rangle$ with occurrence-function $f : A_P \rightarrow A$, such that $f|_{e_P}$ is a bijection, then we say that P *occurs on* S *via* $f|_{e_P}$. (If S is a singleton $\{a\}$, then to simplify notation we simply say that P occurs on a.)

Definition 16. If $I = \langle V, A, var, E, cpt \rangle$ is a CSP instance, then an existential pattern $P = \langle V_P, A_P, var_P, E_P, cpt_P, e_P \rangle$ *appears* in I (and I *contains* P) if $\exists v \in V$ with $|A_v| \geq |e_P|$ such that for all subsets S of A_v 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 $|A_v| \geq |e_P|$, there is an injective mapping $g : e_P \rightarrow A_v$ such that P does not occur on $g(e_P)$ via g.

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 no such pattern 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 $g : e_P \rightarrow A$ 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 $\mathrm{CSP}_{AC}(\overline{P})$ 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 $\mathrm{CSP}_{AC}(\overline{P})$ is NP-complete. It is *tractable* if there is a polynomial-time algorithm to solve $\mathrm{CSP}_{AC}(\overline{P})$.

In an existential pattern $P = \langle V_P, A_P, var_P, E_P, cpt_P, e_P \rangle$, 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 $I = \langle V, A, var, E, cpt \rangle$ 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 $\mathrm{CSP}_{AC}(\overline{P})$ 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 $P = \langle V_P, A_P, var_P, E_P, cpt_P, e_P \rangle$ and $Q = \langle V_Q, A_Q, var_Q, E_Q, cpt_Q, e_Q \rangle$ be two existential patterns, such that P is a sub-pattern of Q. Let $I = \langle V, A, var, E, cpt \rangle$ be an arc-consistent CSP instance. If Q appears in I, then P also appears in I.*
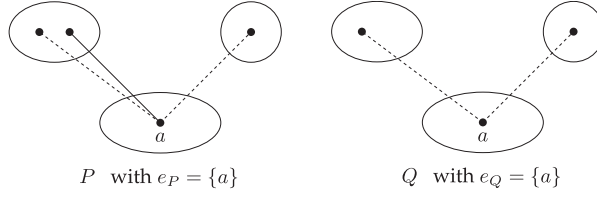
Fig. 18. Example of dp-elimination in an existential pattern P to produce the existential pattern Q.

Proof. Suppose that Q appears in I. So $\exists v \in V$ such that Q occurs on all subsets of A_v of size $|e_Q|$ and for all bijections $g : e_Q \to S$. Let T be any subset of A_v of size $|e_P|$ and let $h : e_P \to T$ 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 $T \subseteq S$ and let $g : e_Q \to S$ be any bijection such that $g|_{e_P} = h$. We know that Q occurs in the existential pattern $\langle V, A, var, E, cpt, S \rangle$ with an occurrence-function f such that $f|_{e_Q} = g$. Since P is a sub-pattern of Q, P occurs in the existential pattern $\langle V, A, var, E, cpt, S \cap f(e_P) \rangle$ with the occurrence-function $f|_{A_P}$. Since $e_P \subseteq e_Q$ 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. $\quad\square$

Lemma 27. *Let $P = \langle V_P, A_P, var_P, E_P, cpt_P, e_P \rangle$ and $Q = \langle V_Q, A_Q, var_Q, E_Q, cpt_Q, e_Q \rangle$ be two existential patterns, such that P can be reduced to a sub-pattern of Q. Let $I = \langle V, A, var, E, cpt \rangle$ be an arc-consistent CSP instance with $|V| \geq |V_P|$. 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 $c \in v_c$, with $v_c \in V_P$, and its corresponding compatibility edge from P transforms P into a sub-pattern Q' of Q, where $Q' = \langle V_{Q'}, A_{Q'}, var_{Q'}, E_{Q'}, cpt_{Q'}, e_{Q'} \rangle$. Since c is a dangling point, from the definition of dp-elimination we know that $c \notin e_P$. 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 $I = \langle V, A, var, E, cpt \rangle$. So $\exists v \in V$ such that for all $S \subseteq A_v$ with $|S| = |e_{Q'}|$ and for all bijections $g : e_{Q'} \to S$, 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 $v_c \in V_{Q'}$, then let $v'_c = var(f(v_c))$ be the variable in I corresponding to v_c in this appearance of Q'. If $v_c \notin V_{Q'}$ (due to being eliminated during dp-elimination), then $|V_{Q'}| < |V_P| \leq |V|$, and so we can set $v'_c \in V$ 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 $c' \in v'_c$ 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 $p \in A_P \setminus \{c\} = A_{Q'}$. Hence P also occurs on S via g, since f and f' are identical on e_P, which completes the proof. $\quad\square$

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 $v' \in V$ is a *copy* in I of $v \in V$ on (A_0, A_1) with $A_0 \subset A_v$ and $A_1 \subset A_{v'}$ if:

- $|A_0| = |A_1|$.
- $\forall a \in A_0, \exists b \in A_1$ such that $cpt(a, b) = T$ and $\forall c \neq b$ in A_1 we have $cpt(a, c) = F$.
- $\forall b \in A_1, \exists a \in A_0$ such that $cpt(a, b) = T$ and $\forall c \neq a$ in A_0 we have $cpt(c, b) = F$.
- $\forall a \in A_0, \forall b \in A_1$ such that $cpt(a, b) = T$, $\forall c \in A \setminus \{A_v, A_{v'}\}$, we have $cpt(b, c) = cpt(a, c)$.

For notational simplicity, we say that v' is a copy of v in I if $A_0 = A_v$ and $A_1 = A_{v'}$.

Lemma 28. *Let $P = \langle V, A, var, E, cpt \rangle$ be a pattern and $P' = \langle V, A, var, E, cpt, a \rangle$ be an existential version of P. Then P' is tractable only if P is tractable.*

Proof. Since $\langle V, A, var, E, cpt \rangle$ is equivalent to $\langle V, A, var, E, cpt, \emptyset \rangle$, the result follows directly from Definition 14 of extension and Corollary 4. $\quad\square$
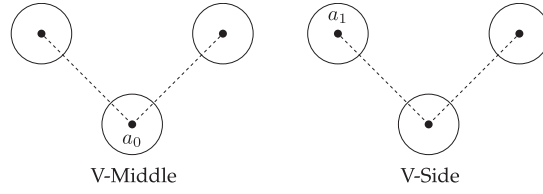
Fig. 19. Two intractable existential patterns on three variables.
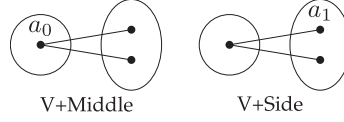

Fig. 20. Two intractable existential patterns on two variables.
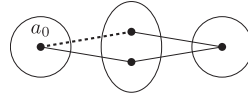

Fig. 21. The existential pattern ExpandedV+.

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 $V_3 = \{v_0, v_1, v_2\}$, $A_3 = \{a_0, a_1, a_2\}$, $var_3(a_i) = v_i$ for $i \in \{0, 1, 2\}$, $E_3 = \{\{a_0, a_1\}, \{a_0, a_2\}\}$ and $cpt_3(a_0, a_1) = cpt_3(a_0, a_2) = F$. Let
V-Middle $= \langle V_3, A_3, var_3, E_3, cpt_3, \{a_0\}\rangle$ be the existential pattern shown on the left of Fig. 19 and V-Side $= \langle V_3, A_3, var_3, E_3, cpt_3, \{a_1\}\rangle$ be the existential pattern shown on the right of Fig. 19.

Lemma 29. *V-Middle and V-Side are NP-Complete.*

Proof. Let $I = \langle V, A, var, E, cpt \rangle$ be an arc-consistent CSP instance. Let $v_1, \ldots, v_k$ be the variables of I. Let $I' = \langle V', A', var', E', cpt' \rangle$ be the CSP instance on variables $v'_1, \ldots, v'_{2k}$ such that:

- $A_{v'_i} = A_{v_i} \cup \{a_i\}$ for all $1 \leq i \leq k$ and $A_{v'_i} = B_{v_{i-k}} \cup \{a_i\}$ for all $k + 1 \leq i \leq 2k$, where $|B_i| = |A_i|$ ($1 \leq i \leq k$). 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 $b \in A_j$, $cpt'(a, b) = cpt(a, b)$. For all $1 \leq i \leq k$, v'_{i+k} is a copy of v'_i in I' on $(A_{v_i}, A_{v'_{i+k}} \setminus \{a_{i+k}\})$.

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'. So for all $1 \leq i \leq k$, neither V-Middle nor V-Side occurs on a_i. So neither V-Middle nor V-side appears in I'. Thus we can reduce any CSP instance I to an arc-consistent CSP instance I' in which neither V-middle nor V-side appears. It follows that V-Middle and V-Side are NP-Complete. □

Let $V_2 = \{v_0, v_1\}$, $A_2 = \{a_0, a_1, a_2\}$, $var_2(a_0) = v_0$, $var_2(a_1) = var_2(a_2) = v_1$, $E_2 = \{\{a_0, a_1\}, \{a_0, a_2\}\}$ and $cpt_2(a_0, a_1) = cpt_2(a_0, a_2) = T$. Let V+Middle be the existential pattern
$\langle V_2, A_2, var_2, E_2, cpt_2, \{a_0\}\rangle$ shown on the left of Fig. 20 and V+Side the existential pattern
$\langle V_2, A_2, var_2, E_2, cpt_2, \{a_1\}\rangle$ shown on the right of Fig. 20.
Let ExpandedV+ $= \langle V, A, var, E, cpt, \{a_0\}\rangle$ be the existential pattern shown in Fig. 21 and given by: $V = \{v_0, v_1, v_2\}$, $A = \{a_0, a_1, a_2, a_3\}$, $var(a_0) = v_0$, $var(a_1) = var(a_2) = v_1$, $var(a_3) = v_2$, $E = \{\{a_0, a_1\}, \{a_0, a_2\}, \{a_3, a_1\}, \{a_3, a_2\}\}$, $cpt(a_0, a_2) = cpt(a_3, a_1) = cpt(a_3, a_2) = T$ and $cpt(a_0, a_1) = F$.

Lemma 30. *V+Middle, V+Side and ExpandedV+ are NP-Complete.*

Proof. Let $I = \langle V, A, var, E, cpt \rangle$ be an arc-consistent CSP instance on variables $v_1, \ldots, v_k$. Let $I' = \langle V', A', var', E', cpt' \rangle$ be the CSP instance on variables $v'_1, \ldots, v'_{3k}$ such that:
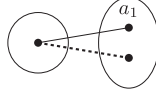
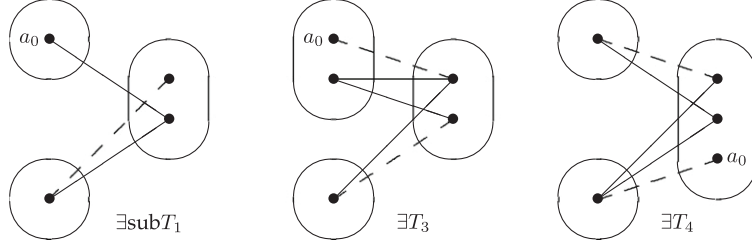**Fig. 22.** The existential pattern V $+$ $-$.

Fig. 23. Three intractable existential patterns.

- $A_{v_i'} = A_{v_i} \cup \{a_i, b_i\}$ for all $1 \le i \le k$, $A_{v_i'} = B_{v_{i-k}} \cup \{a_i, b_i\}$ for all $k + 1 \le i \le 2k$, and $A_{v_i'} = C_{v_{i-2k}} \cup \{a_i, b_i\}$ for $2k + 1 \le i \le 3k$, where $|C_i| = |B_i| = |A_i|$ $(1 \le i \le k)$. We can think of variables v_i', v_{i+k}' and v_{i+2k}' as having the same domain except for the special values corresponding to a_i, b_i.
- For all $1 \le i \le 3k$, for all $1 \le j \le 3k$ such that $i \ne j$, a_i is compatible with b_j and incompatible with all other points of $A_{v_j'}$. For all $1 \le i \le 3k$, for all $1 \le j \le 3k$ such that $i \ne j$, b_i is compatible with a_j and incompatible with all other points of $A_{v_j'}$. For all $1 \le i \le k$, this prevents any three points from $\{a_i, b_i, a_{i+k}, b_{i+k}, a_{i+2k}, b_{i+2k}\}$ to be part of the same solution. The idea here is that for any solution S' to I', for all $1 \le i \le k$, at least one of the points of S' in v_i', v_{i+k}' and v_{i+2k}' will be an original point from I, or a copy of an original point from I.
- For all $1 \le i < j \le k$, for all $a \in A_{v_i}$, for all $b \in A_j$, $cpt'(a, b) = cpt(a, b)$. For all $1 \le i \le k$, v_{i+k} is a copy of v_k in I' on $(A_{v_i}, A_{v_{i+k}'} \setminus \{a_{i+k}, b_{i+k}\})$ and v_{i+2k} is a copy of v_k on $(A_{v_i}, A_{v_{i+2k}'} \setminus \{a_{i+2k}, b_{i+2k}\})$.

By construction, I has a solution if and only I' has a solution. Furthermore, for all $1 \le i \ne j \le k$, a_i is only compatible with b_j in A_{v_j} and b_j is itself only compatible with a_i in A_{v_i}. So for all $1 \le i \le k$, neither V+Middle nor V+Side occurs on a_i. Moreover, for all $1 \le i, j, h \le k$, a_i is only compatible with b_j in A_{v_j}, b_j is only compatible with a_h in A_{v_h} and a_h is only compatible with b_j in A_{v_j}. So for all $1 \le i \le k$, ExpandedV+ does not occur on a_i. So none of V+Middle, V+side or ExpandedV+ appear in I'. Hence V+Middle and V+Side are NP-Complete. $\square$

Let V$+-$ $= \langle V, A, var, E, cpt, \{a_1\} \rangle$ be the existential pattern shown in Fig. 22 and given by $V = \{v_0, v_1\}, A = \{a_0, a_1, a_2\}$, $var(a_0) = v_0$, $var(a_1) = var(a_2) = v_1$, $E = \{\{a_0, a_1\}, \{a_0, a_2\}\}$, $cpt(a_0, a_1) = T$ and $cpt(a_0, a_2) = F$.

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:

- $A_{v_i'} = A_{v_i} \cup \{a_i\}$ for all $1 \le i \le k$.
- For all $1 \le i < j \le k$, a_i is incompatible with a_j. For all $1 \le i < j \le k$, for all $b \in A_{v_j'}$, 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 $1 \le i < j \le k$, for all $a \in A_{v_i}$, for all $b \in A_{v_j}$, $cpt'(a, b) = cpt(a, b)$.

For all $1 \le i \ne j \le k$, we know that a_i is compatible with a point $a' \in A_{v_j}$ if and only if a' is compatible with all points in A_{v_i}. Since at most one point in A_{v_j} is incompatible with a point in A_{v_i}, and since $|A_{v_j}| \ge 2$, there always exists such a point $a' \in A_{v_j}$. So I' is arc consistent. Furthermore, for all $1 \le i \ne j \le k$ we know that if a_i is compatible with a point $a' \in A_{v_j}$, then a' is compatible with all points in A_{v_i}. So for all $1 \le i \le k$, V $+$ $-$ does not occur on a_i and we can 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. $\square$

Let $\exists T_3 = \langle V, A, var, E, cpt, \{a_0\} \rangle$ be the existential pattern shown in the middle of Fig. 23 and defined by $V = \{v_0, v_1, v_2\}$, $A = \{a_0, a_1, a_2, a_3, a_4\}$, $var(a_0) = var(a_1) = v_0$, $var(a_2) = var(a_3) = v_1$, $var(a_4) = v_2$, $E = \{\{a_0, a_2\}, \{a_1, a_2\}, \{a_1, a_3\}, \{a_2, a_4\}, \{a_3, a_4\}\}$, $cpt(a_1, a_2) = cpt(a_1, a_3) = cpt(a_2, a_4) = T$ and $cpt(a_0, a_2) = cpt(a_3, a_4) = F$.

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 $v'_1, \ldots, v'_{2k}$, $u'_1, \ldots, u'_k$ with compatibility function cpt' such that:

- $A_{v'_i} = A_{v_i} \cup \{a_i, b_i\}$ for all $1 \leq i \leq k$, $A_{v'_i} = B_{v_{i-k}} \cup \{a_i, b_i\}$ for all $k+1 \leq i \leq 2k$, where $|B_i| = |A_i|$ ($1 \leq i \leq k$), and $A_{u'_i} = \{c_i, d_i, e_i\}$ 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_i. 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 $A_{v'_j}$. For all $1 \leq i \leq k$: a_i is compatible with a_{i+k} and incompatible with all other points of $A_{v'_{i+k}}$; a_{i+k} is incompatible with all points in $A_{v'_i} \setminus \{a_i\}$; b_i is compatible with b_{i+k} and incompatible with all other points of $A_{v'_{i+k}}$; b_{i+k} is incompatible with all points in $A_{v'_i} \setminus \{b_i\}$.
- 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 $(A_{v'_i} \setminus \{b_i\}) \cup (A_{v'_{i+k}} \setminus \{a_{i+k}\})$ and compatible with all other points in A'; e_i is incompatible with all points in $(A_{v'_i} \setminus \{a_i\}) \cup (A_{v'_{i+k}} \setminus \{b_{i+k}\})$ and compatible with all other points in A'.
- For all $1 \leq i < j \leq k$, for all $a \in A_{v_i}$, for all $b \in A_j$, $cpt'(a, b) = cpt(a, b)$. For all $1 \leq i \leq k$, v_{i+k} is a copy of v_k in I' on $(A_{v_i}, A_{v'_{i+k}} \setminus \{a_{i+k}, b_{i+k}\})$.

The points a_i, b_i, d_i, e_i 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 $\exists T_3$ 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 $\exists T_3$ occurs on a_i. Let v be the variable of a_i. Since $\exists T_3$ occurs on a_i, there is a variable v' and a point $a' \in A_{v'}$ 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-k} if $i > k$) and u'_i (or u'_{i-k} if $i > k$). Since $\exists T_3$ occurs on a_i, there is a point in A_v which is compatible with two different points in $A_{v'}$. However, from the second and fourth bullet points we know that there is no point in A_v compatible with two different points in $A_{v'_i}$ ($A_{v'_{i-k}}$ if $i > k$), and from the third bullet point we also know that there is no point in A_v compatible with two different points in $A_{u'_i}$ ($A_{u'_{i-k}}$ if $i > k$). So $\exists T_3$ cannot occur on a_i. So the existential pattern $\exists T_3$ cannot occur on any a_i, with $1 \leq i \leq 2k$.

Similarly, it is easy to verify that the existential pattern $\exists T_3$ does not occur on c_i (for all $1 \leq i \leq k$). Hence $\exists T_3$ does not appear in I'. It follows that $\exists T_3$ is NP-complete. $\quad\square$

Let $\exists subT_1 = \langle V, A, var, E, cpt, \{a_0\}\rangle$ be the existential pattern shown on the left of Fig. 23 and defined by $V = \{v_0, v_1, v_2\}$, $A = \{a_0, a_1, a_2, a_3\}$, $var(a_0) = v_0$, $var(a_1) = var(a_2) = v_1$, $var(a_3) = v_2$, $E = \{\{a_0, a_1\}, \{a_1, a_3\}, \{a_2, a_3\}\}$, $cpt(a_0, a_1) = cpt(a_1, a_3) = T$ and $cpt(a_2, a_3) = F$.

Lemma 33. $\exists subT_1$ 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 cpt' such that:

- $A_{v'_i} = A_{v_i} \cup \{a_i\} \cup \{b_{ij} \mid j = 1, \ldots, i-1, i+1, \ldots, n\}$ for all $1 \leq i \leq n$.
- For all $1 \leq i, j \leq k$ with $i \neq j$, for all $p \in A_{v'_i}$ and for all $q \in A_{v'_j} \setminus A_{v_j}$, $cpt'(p, q) = T$ if and only if $p = b_{ij}$ or $q = b_{ji}$.
- For all $1 \leq i < j \leq n$, for all $a \in A_{v_i}$, for all $b \in A_{v_j}$, $cpt'(a, b) = cpt(a, b)$.

It is easy to verify that none of the points a_i or b_{ij} belong to a solution to any 4-variable sub-instance of I'. This implies that the solutions to I' are exactly the solutions I.

To complete the proof, it remains to show that for each $i = 1, \ldots, n$, $\exists subT_1$ 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_{ji} in $A_{v'_j}$ which is only compatible with b_{kj} in $A_{v'_k}$. Since b_{kj} is compatible with all points in $A_{v'_j}$, the existential pattern $\exists subT_1$ does not occur on a_i in I'. $\quad\square$

Let $\exists T_4 = \langle V, A, var, E, cpt, \{a_0\}\rangle$ be the existential pattern shown on the right of Fig. 23 and defined by $V = \{v_0, v_1, v_2\}$, $A = \{a_0, a_1, a_2, a_3, a_4\}$, $var(a_0) = var(a_1) = var(a_2) = v_0$, $var(a_3) = v_1$, $var(a_4) = v_2$, $E = \{\{a_0, a_4\}, \{a_1, a_3\}, \{a_1, a_4\}, \{a_2, a_3\}, \{a_2, a_4\}\}$, $cpt(a_1, a_3) = cpt(a_1, a_4) = cpt(a_2, a_4) = T$ and $cpt(a_0, a_4) = cpt(a_2, a_3) = F$.

Lemma 34. $\exists T_4$ is NP-Complete.

Proof. Let $I = \langle V, A, var, E, cpt\rangle$ 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 $\exists T_4$ 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 $I' = \langle V', A', var', E', cpt'\rangle$ 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:
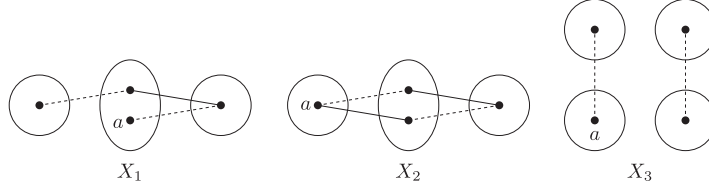
Fig. 24. Three tractable existential patterns.

- $A_{v_i'} = A_{v_i} \cup \{a_i\}$ for all $1 \le i \le n$.
- For all $1 \le i \le n$, $|A_{w_i'}| = |A_{v_i'}|$ and the constraint between v_i' and w_i' is a permutation constraint, i.e. there is a bijection $\pi : A_{v_i'} \to A_{w_i'}$ such that $\forall p \in A_{v_i'}, \forall q \in A_{w_i'}, cpt'(p, q) = T$ if and only if $q = \pi(p)$. For all $1 \le i \le n$, we denote $\pi(a_i)$ by b_i.
- For all $1 \le i, j \le k$ with $i \ne j$: $A_{x_i'} = \{c_i, d_i\}$ and $A_{y_i'} = \{e_i, f_i\}$; the point c_i is compatible with all points in A' except e_i; the point e_i is compatible with all points in A' except c_i; the point d_i is compatible with all points in A' except b_i and f_i; the point f_i is compatible with all points in A' except b_i and d_i. This implies that b_i, and hence a_i cannot be part of any solution on the variables v_i', w_i', x_i', y_i'.
- For all $1 \le i \ne j \le n$: for all $p \in A_{v_i}$, for all $q \in A_{v_j}, cpt'(p, q) = cpt(p, q)$; for all $q \in A_{v_j'}, cpt(a_i, q) = T$.
- For all $1 \le i < j \le n$, for all $p \in A_{w_i}$, for all $q \in A_{w_j}, cpt(p, q) = T$.

For all $1 \le i \le n$, let g_i be any point in $A_{w_i'} \setminus \{b_i\}$. By construction, the existential pattern $\exists T_4$ does not occur on any of the points a_i, g_i, c_i or e_i in I'. Hence $\exists T_4$ does not appear in the instance I'.

For all $1 \le i \le n$, the point a_i cannot be extended to a solution to the sub-instance on variables v_i', w_i', x_i', y_i', whereas all other points in $A_{v_i'}$ can. It follows that the solutions to I' are exactly the solutions to I. Hence $\exists T_4$ is NP-complete. $\square$

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

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

Let $X_3 = \langle V, A, var, E, cpt, \{a\}\rangle$ be the following existential pattern (shown on the right of Fig. 24): $V = \{v_0, v_1, v_2, v_3\}$, $A = \{a, b, c, d\}$, $var(a) = v_0, var(b) = v_1, var(c) = v_2, var(d) = v_3, E = \{\{a, b\}, \{c, d\}\}, cpt(a, b) = cpt(c, d) = F$.

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 $P = \langle V, A, var, E, cpt, \{a_P\}\rangle$ 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 $P = \langle V, A, var, E, cpt, \{a_P\}\rangle$ 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 $P' = \langle V, A, var, E, cpt\rangle$ 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-Side, V-Middle, V+- or $\exists$subT_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-Side, V-Middle, V+-, V+Side, V+Middle, ExpandedV+ or $\exists$subT_1 *or* is a subpattern of X_1. Hence, by Lemmas 29 and 30, Lemmas 31 and 33, P is either intractable or is a subpattern of X_1.

If P' is a sub-pattern of T_3, then the irreducible singleton 3-variable existential pattern P *either* contains one of V+Side, V+Middle, ExpandedV+, V + -, or $\exists T_3$ *or* is a subpattern of X_1 or X_2. Hence, by Lemmas 30–32, P is either intractable or is a subpattern of X_1 or X_2.

If P' is a sub-pattern of T_4, then the irreducible singleton 3-variable existential pattern P *either* contains one of V+Side, V+Middle, ExpandedV+, V + -, $\exists$subT_1 or $\exists T_4$ *or* is a subpattern of X_1 or X_2. Hence, by Lemma 30, Lemma 31, Lemma 33 and Lemma 34, P is either intractable or is a subpattern of X_1 or X_2.

If P' is a sub-pattern of T_5, then the irreducible singleton 3-variable existential pattern P *either* contains $\text{V} + -$ *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.

$\Leftarrow$: We now give the tractability proofs for the patterns X_1, X_2, X_3.

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

Proof of tractability of X_2: Let $I = \langle V, A, var, E, cpt \rangle$ be an arc-consistent CSP instance such that X_2 does not appear in I. So $\forall v_i \in V, \exists a_i \in A_{v_i}$ such that X_2 does not occur on a_i. Suppose that we have a partial solution $S_k = \{s_1 \in A_{v_1}, \ldots, s_k \in A_{v_k}\}$, with $0 \leq k < |V|$. Let Y be the set $i \leq k$ such that s_i and a_{k+1} are compatible and let $\overline{Y}$ be the set of $i \leq k$ such that s_i and a_{k+1} are compatible. By arc consistency, $\forall i \in \overline{Y}, \exists t_i$ such that t_i and a_{k+1} are compatible. Let $S' = \{s'_1 \in A_{v_1}, \ldots, s'_{k+1} \in A_{v_{k+1}}\}$ with

$$
s'_i = \begin{cases} a_i & \text{if } i = k+1 \\ s_i & \text{if } i \in Y \\ t_i & \text{if } i \in \overline{Y}. \end{cases}
$$

Let $i \in Y$ and $j \in \overline{Y}$. Since S is a partial solution, s_i and s_j are compatible. We know that a_{k+1} is compatible with t_j and incompatible with s_j. Since X_2 does not occur on a_{k+1}, s_i and t_j are compatible.

Let $i, j \in \overline{Y}$. From the argument in the previous paragraph, we know that s_i and t_j are compatible. We also know that a_{k+1} is compatible with t_i and incompatible with s_j. Since X_2 does not occur on a_{k+1}, t_i and t_j are compatible.

So all the points in S' are compatible with each other. So S' is a partial solution for variables $(v_1, \ldots, v_{k+1})$. So if we have a partial solution for I on k variables, then we also have a partial solution for I on $k+1$ variables. Hence, assuming $A \neq \emptyset$, there is always a solution for I. So X_2 is tractable.

Proof of tractability of X_3: Let $I = \langle V, A, var, E, cpt \rangle$ be an arc-consistent CSP instance such that X_3 does not appear in I. So $\forall v_i \in V, \exists a_i \in A_{v_i}$ such that X_3 does not occur on a_i. If all a_i are compatible with all points in I, then the set of all a_i is a solution for I. Otherwise, let i and j be such that $\exists b \in A_{v_j}$ such that a_i and b are incompatible. Since X_3 does not occur on a_i, there is no incompatibility edge between two points of $A \setminus (A_{v_i} \cup A_{v_j})$. Thus we can perform the fusion of v_i and v_j into a new variable v_{ij} such that points in $A_{v_{ij}}$ correspond to compatibility edges between v_i and v_j. Since there is no incompatibility edge between two points outside of $A_{v_{ij}}$, applying arc consistency on v_{ij} will determine whether there is a solution for I.

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 $P = \langle V, A, var, E, cpt, e \rangle$ be an irreducible existential pattern on two constraints. Then $\mathrm{CSP}_{AC}(\overline{P})$ 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 $\mathrm{CSP}_{AC}(\overline{P})$ is NP-complete.*

Proof. We first make the observation that for non-existential patterns, i.e. patterns P for which $e = \emptyset$, $\mathrm{CSP}_{AC}(\overline{P})$ is solvable in polynomial time if and only if $\mathrm{CSP}(\overline{P})$ 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_5 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 $T_1, T_2, T_3, T_4, T_5, 2I$. 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 $\text{V} + -$ or V-Side and hence are NP-complete by Lemmas 29 and 31. $\square$

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 [17,11], along with a generalisation of the notion of microstructure [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

[1] C. Bessière, J.-C. Régin, R.H.C. Yap, Y. Zhang, An optimal coarse-grained arc consistency algorithm, Artif. Intell. 165 (2) (2005) 165–185.
[2] A.A. Bulatov, Tractable conservative constraint satisfaction problems, in: LICS 2003: Proceedings of 18th IEEE Symposium on Logic in Computer Science, 2003, pp. 321–330.
[3] A.A. Bulatov, A dichotomy theorem for constraint satisfaction problems on a 3-element set, J. ACM 53 (1) (2006) 66–120.
[4] A. Bulatov, P. Jeavons, A. Krokhin, Classifying the complexity of constraints using finite algebras, SIAM J. Comput. 34 (3) (2005) 720–742.
[5] D.A. Cohen, M.C. Cooper, P. Creed, D. Marx, A.Z. Salamon, The tractability of csp classes defined by forbidden patterns, J. Artif. Intell. Res. (JAIR) 45 (2012) 47–78.
[6] D.A. Cohen, M.C. Cooper, G. Escamocher, S. Živný, Variable elimination in binary csp via forbidden patterns, IJCAI 2013 (2013) 517–523.
[7] S.A. Cook, The complexity of theorem-proving procedures, in: M. A. Harrison, R. B. Banerji, J. D. Ullman (Eds.), STOC, ACM, 1971, pp. 151–158.
[8] M.C. Cooper, Fundamental properties of neighbourhood substitution in constraint satisfaction problems, Artif. Intell. 90 (1–2) (1997) 1–24.
[9] M.C. Cooper, Beyond consistency and substitutability, in: Principles and Practice of Constraint Programming—20th International Conference, CP 2014, Lyon, France, September 8–12, 2014. Proceedings, 2014, pp. 256–271.
[10] M.C. Cooper, D.A. Cohen, P. Jeavons, Characterising tractable constraints, Artif. Intell. 65 (2) (1994) 347–361.
[11] M.C. Cooper, A. El Mouelhi, C. Terrioux, B. Zanuttini, On broken triangles, in: Principles and Practice of Constraint Programming - 20th International Conference, CP 2014, Lyon, France, September 8-12, 2014. Proceedings, 2014 pp. 9–24.
[12] M.C. Cooper, G. Escamocher, A dichotomy for 2-constraint forbidden csp patterns, in: J. Hoffmann, B. Selman (Eds.), AAAI, AAAI Press, 2012.
[13] M.C. Cooper, P.G. Jeavons, A.Z. Salamon, Generalizing constraint satisfaction on trees: Hybrid tractability and variable elimination, Artif. Intell. 174 (9–10) (2010) 570–584.
[14] M.C. Cooper, S. Živný, Hierarchically nested convex VCSP, in: CP 2011, 2011, pp. 187–194.
[15] M.C. Cooper, S. Živný, Hybrid tractability of valued constraint problems, Artif. Intell. 175 (9–10) (2011) 1555–1569.
[16] M.C. Cooper, S. Živný, Tractable triangles and cross-free convexity in discrete optimisation, J. Artif. Intell. Res. (JAIR) 44 (2012) 455–490.
[17] A. El Mouelhi, P. Jégou, C. Terrioux, A hybrid tractable class for non-binary csps, in: 2013 IEEE 25th International Conference on Tools with Artificial Intelligence, Herndon, VA, USA, November 4–6, 2013, 2013, pp. 947–954.
[18] A. El Mouelhi, P. Jégou, C. Terrioux, Microstructures for csps with constraints of arbitrary arity, in: Proceedings of the Tenth Symposium on Abstraction, Reformulation, and Approximation, SARA 2013, 11–12 July 2013, Leavenworth, Washington, USA, 2013.
[19] E.C. Freuder, Eliminating interchangeable values in constraint satisfaction problems, in: T. L. Dean, K. McKeown (Eds.), AAAI, AAAI Press/The MIT Press, 1991, pp. 227–233.
[20] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman, San Francisco, CA, 1979.
[21] M. Grohe, The complexity of homomorphism and constraint satisfaction problems seen from the other side, J. ACM 54 (1) (2007) 1–24.
[22] D. Marx, Can you beat treewidth? Theory Comput. 6 (1) (2010) 85–112.
[23] D. Marx, Tractable hypergraph properties for constraint satisfaction and conjunctive queries, in: STOC'10: Proceedings of the 42nd ACM Symposium on Theory of Computing, ACM, 2010, pp. 735–744.
[24] F. Rossi, P. van Beek, T. Walsh, Handbook of Constraint Programming, in: Foundations of Artificial Intelligence, Elsevier, 2006.