Galois Connections for Patterns: An Algebra of Labelled Graphs

David A Cohen, Martin Cooper, Peter G Jeavons, Stanislav Živný

Galois Connections for Patterns: An Algebra of Labelled Graphs

David A. Cohen[1], Martin C. Cooper[2(✉)], Peter G. Jeavons[3], and Stanislav Živný[3]

[1] Royal Holloway, University of London, Egham, UK
dave@cs.rhul.ac.uk
[2] IRIT, University of Toulouse, Toulouse, France
cooper@irit.fr
[3] University of Oxford, Oxford, UK
{peter.jeavons,standa.zivny}@cs.ox.ac.uk

Abstract. A pattern is a generic instance of a binary constraint satisfaction problem (CSP) in which the compatibility of certain pairs of variable-value assignments may be unspecified. The notion of forbidden pattern has led to the discovery of several novel tractable classes for the CSP. However, for this field to come of age it is time for a theoretical study of the algebra of patterns. We present a Galois connection between lattices composed of sets of forbidden patterns and sets of generic instances, and investigate its consequences. We then extend patterns to augmented patterns and exhibit a similar Galois connection. Augmented patterns are a more powerful language than flat (i.e. non-augmented) patterns, as we demonstrate by showing that, for any $k \geq 1$, instances with tree-width bounded by k cannot be specified by forbidding a finite set of flat patterns but can be specified by a finite set of augmented patterns. A single finite set of augmented patterns can also describe the class of instances such that each instance has a weak near-unanimity polymorphism of arity k (thus covering all tractable language classes).We investigate the power of forbidding augmented patterns and discuss their potential for describing new tractable classes.

Keywords: Constraint satisfaction · Tractability · Forbidden patterns · Galois connection · Lattice

The authors were supported by EPSRC grant EP/L021226/1. Martin Cooper was supported by the grants ANR-18-CE40-0011 and ANR-19-PI3A-000. Stanislav Živný was supported by a Royal Society University Research Fellowship. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 714532). The paper reflects only the authors' views and not the views of the ERC or the European Commission. The European Union is not liable for any use that may be made of the information contained therein.

M. Cochez et al. (Eds.): GKR 2020, LNAI 12640, pp. 125–150, 2021.
https://doi.org/10.1007/978-3-030-72308-8_9

1 Introduction

The CSP (Constraint Satisfaction Problem) is a classical abstract framework for the modelling of finite-domain constrained assignment problems [8,32]. Although first inspired by applications in computer vision and artificial intelligence, it's generic nature has allowed it to become a programming paradigm in its own right used in, for example, scheduling, product configuration, planning and bioinformatics. It is well known that the CSP is NP-complete and remains so even when restricted to binary constraints since all instances have an equivalent dual instance which is binary [22,40].

An interesting avenue of theoretical research on CSPs consists in the characterisation of tractable subproblems defined by placing a restriction on the type of constraints that can occur (the constraint language) and again it is known that it is possible to limit attention to languages of binary relations [5,10]. A major advance towards the recent characterisation of tractable constraint languages [3,41] was the algebraic approach based on the study of pointwise closure operations of constraint relations, known as polymorphisms, and the identities satisfied by these polymorphisms [1,4]. Of particular interest is the Galois connection between (sets of) polymorphisms and (sets of) relations [27]. In parallel, tractable subproblems of the CSP based on restrictions on the (hyper)-graph of constraint scopes (the constraint (hyper)graph) were also characterised [26].

In order to define new classes, we need to go beyond placing restrictions on constraint languages or on the structure of the constraint (hyper)-graph. A natural way of defining sets of instances is to consider properties of the microstructure of binary CSP instances [30]. A *pattern* can be seen as a partial microstructure (i.e. a binary CSP instance in which the compatibility of some assignments may be left undefined) or, more abstractly, as a graph with vertices labelled by names of variables and edges which may be positive or negative. Defining sets of binary CSP instances by forbidding patterns has led to the discovery of novel tractable classes [9,18]. For example, in each of the following cases, forbidding a simple 3-variable pattern defines a tractable class of binary CSP instances which strictly generalises a known tractable class:

- The Broken-Triangle Property (BTP) [16] includes all instances whose constraint graph is a tree. It has also led to the discovery of interesting reduction operations [14] and has been extended in different ways to define larger tractable classes [12,35–37].
- The Joint-Winner Property (JWP) [17] includes all CSP instances defined by a single All-Different constraint [38] together with arbitrary unary constraints.
- The Extended Max-Closed (EMC) class [19] includes all binary max-closed instances [29]. The stable marriage problem [31] is just one example of a class of problems that can be expressed as binary max-closed CSPs [25].
- The T_4 pattern [15] generalises the ZOA language class [13] which is itself a generalisation of 2SAT. Three other patterns have also recently been shown to define tractable classes that generalise 2SAT [7].

In this paper we initiate the study of the underlying theory of forbidden (sets of) patterns, an essential foundation on which to build a characterisation of all tractable classes defined by forbidden (sets of) patterns. We begin by studying what we call flat patterns before studying augmented patterns with extra structure, such as partial orders on variables or domain values. Adding such structure is not only essential to define certain hybrid classes such as BTP [16] and EMC [19], but, as we will show in Sect. 6, also allows us to define (families of) polymorphisms [28] and bounded tree-width [20] within the same framework.

For both flat and augmented patterns, we exhibit a Galois connection between sets of patterns and sets of instances. In each case, we investigate the tractability consequences of the Galois connection, including the possibility of defining new tractable classes by combination of known tractable classes via the lattice operations. We notably show that tractable classes form a sublattice.

2 Definitions and Notation

We assume that there is a countable collection of variables $\mathcal{X}$ and a countable domain $\mathcal{D}$ of values. A variable-value pair (x, a), representing the assignment of value $a \in \mathcal{D}$ to variable $x \in \mathcal{X}$, is known as a *point*. A *flat pattern* (or simply a *pattern*) $P = \langle A_P, \rho_P \rangle$ is a subset A_P of $\mathcal{X} \times \mathcal{D}$ equipped with a (partial) function ρ_P from the pairs of points $(x, a), (y, b)$ of P such that $x \neq y$ to {negative, positive}. Thus P consists of a set of variable-value assignments (x, a) together with a set of negative and positive edges representing the compatibility of pairs of assignments. In figures we represent negative edges by dashed lines, positive edges by solid lines and points corresponding to assignments to the same variable are grouped into ovals. Three patterns $P1$, $P2$, $P3$ are shown in Fig. 1.

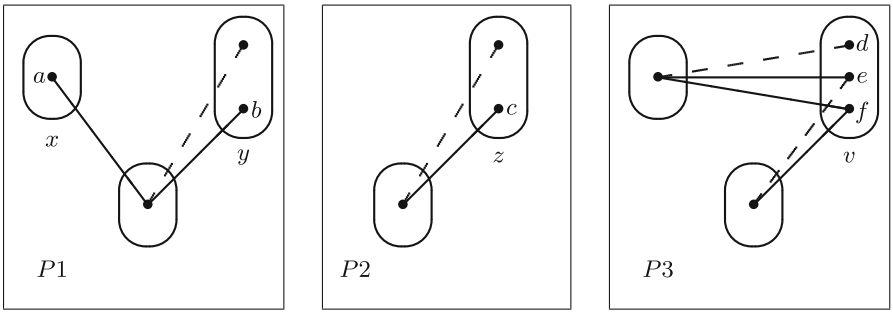

Fig. 1. Examples of the occurrence of a pattern in another pattern: $P1 \to P2$, $P2 \to P1$, $P1 \to P3$, $P2 \to P3$.

We give a recursive definition of connectedness. Two points $(x, a), (y, b)$ in a pattern P are *connected* if $x = y$ or $\rho_P((x, a), (y, b)) \in \{\text{negative, positive}\}$ or if $(x, a), (y, b)$ are both connected to some point (z, c) of P. Clearly, each pattern has a decomposition into connected components according to this definition of connectedness.

A *completely specified binary CSP instance* (or simply an *instance*) is a pattern $I = \langle A_I, \rho_I \rangle$ in which the function ρ_I is total, i.e. the compatibility of each pair of variable-value assignments (to distinct variables) is specified. Given an instance I on n variables, a *solution* to I is a clique of positive edges of size n, which corresponds to a pairwise-compatible assignment of values to variables. The question associated with an instance is the existence of a solution. An instance I is *arc consistent* if for all points (x, a) of I and all variables $y \neq x$ of I, (x, a) has a support at y, i.e. $\exists b \in \mathcal{D}$ such that $\{(x, a), (y, b)\}$ is a positive edge in I.

A pattern $P = \langle A_P, \rho_P \rangle$ *occurs* in pattern $Q = \langle A_Q, \rho_Q \rangle$ if there is a mapping f from A_P to A_Q which respects variables, maps negative edges to negative edges and positive edges to positive edges, i.e.

1. $f(x, a) = (u, c)$ and $f(x, b) = (v, d)$ implies that $u = v$.
2. $f(x, a) = (u, c)$, $f(y, b) = (v, d)$ and $\rho_P((x, a), (y, b)) \in \{\text{negative, positive}\}$ implies that $u \neq v$ and $\rho_P((x, a), (y, b)) = \rho_Q((u, c), (v, d))$.

We use the notation $P \to Q$ to denote that P *occurs* in pattern Q (and $P \not\to Q$ if it does not). It is easy to see from its definition that occurrence is transitive: $P \to Q$ and $Q \to R$ implies $P \to R$. We consider two patterns P, Q to be equivalent if $P \to Q$ and $Q \to P$: we write $P \approx Q$. For example, patterns $P1$ and $P2$ in Fig. 1 are equivalent; we notably have $P1 \to P2$ since (x, a), (y, b) can both map to (z, c). Clearly, we have $P2 \to P3$, and then, by transitivity, $P1 \to P3$. For simplicity of presentation, throughout this paper, we will talk about patterns rather than equivalence classes of patterns.

Each pattern P defines a corresponding set of instances in which P does not occur. For example, the pattern $P3$ of Fig. 1 defines a set of instances which includes all binary CSP instances with Boolean domains, since if $P3 \to I$ then the points (v, d), (v, e), (v, f) must map to three distinct values for the same variable in I, due to the positive and negative edges in $P3$.

Note that in previous work, it has sometimes been convenient to assume that when P occurs in Q, distinct variables of P map to distinct variables of Q [11,15,19]. However, to establish a Galois connection for flat patterns, we require a looser definition of occurrence in which two or more variables of P may map to the same variable in Q. To impose the stricter definition of occurrence (inducing an injective mapping of variables of P), it suffices, for each pair of distinct variables x, y, to add two new points (x, a), (y, b) to A_P and an extra dummy positive edge between points (x, a), (y, b) in P; this prevents x, y mapping to the same variable in Q (and only changes the semantics of P in a trivial way). A more elegant solution (in order to impose an injective mapping of variables) is to augment the patterns with a not-equal-to relation between variables which is possible in the framework of augmented patterns discussed in Sect. 6.

We consider sets S of patterns. These sets will usually be finite, indeed, often a singleton. When forbidden, a set S of patterns defines a set of instances (those sets of instances in which none of the patterns in S occurs). Such sets T of instances are hereditary in the sense that $(I \in T) \wedge (I' \subseteq I) \Longrightarrow (I' \in T)$, where $I' \subseteq I$ means $(A_{I'} \subseteq A_I) \wedge (\rho_{I'} = \rho_I|_{A_{I'}})$. Many, but not all, classes of interest are hereditary. For example, for any k, the set of instances whose tree-width is bounded by k is hereditary. On the other hand, the set of instances which is arc-consistent is not hereditary, since a value which has a support at another variable in an instance I will not necessarily have a support in $I' \subset I$. Thus forbidden flat patterns alone cannot express any class of instances which requires arc consistency (or a higher level of consistency) [36]. Nevertheless, we will see in Sect. 6 how a combination of augmented patterns and filters on instances provides a very expressive language in which to define classes on instances, allowing us to express such classes of instances.

In order to obtain a Galois connection we consider sets of generic instances, where a generic instance can be viewed as a partially-specified instance and is, in fact, again just a pattern. However, the lattice structure on sets of patterns is different depending on whether we view these patterns as partially-specified instances or as forbidden sub-instances. When defining tractability of sets of generic instances we filter instances keeping only those that are completely specified.

Definition 1. *A set T of generic instances is* tractable *if there is a polynomial-time algorithm which decides all completely-specified instances in T. A set S of forbidden patterns is tractable if the corresponding set of instances in which none of the patterns in S occur is tractable.*

To define lattices of (sets of) instances and (sets of) patterns, we also require the following operation on patterns: if P, Q are patterns, then $P + Q$ is a single pattern consisting of (copies of) the two patterns P and Q (without any common points and without any edges between P and Q). We call this the *juxtaposition* of the two patterns P and Q. Observe that $P + P \approx P$ (since $P + P \to P$ follows from the definition of occurrence which allows us to map the two copies of P to P). If S_1, S_2 are sets of patterns, then $S_1 + S_2$ is the set of patterns $\{P + Q \mid P \in S_1 \wedge Q \in S_2\}$.

We also require another operation on pairs of patterns, which can be seen as the greatest lower bound of the two patterns. If P, Q are patterns, then $P \times Q$ is a single pattern built by forming the juxtaposition of all patterns R such that $(R \to P) \wedge (R \to Q)$. We say that such patterns R are *common factors* of P and Q. We only include patterns R which are maximal in the sense that there is no $R' \not\approx R$ such that $R \to R'$ and $(R' \to P) \wedge (R' \to Q)$. Observe that including only maximal R, ensures that we have $P \times P \approx P$. The operation $\times$ is illustrated in Fig. 2. In this example, the patterns P and Q have only two maximal common

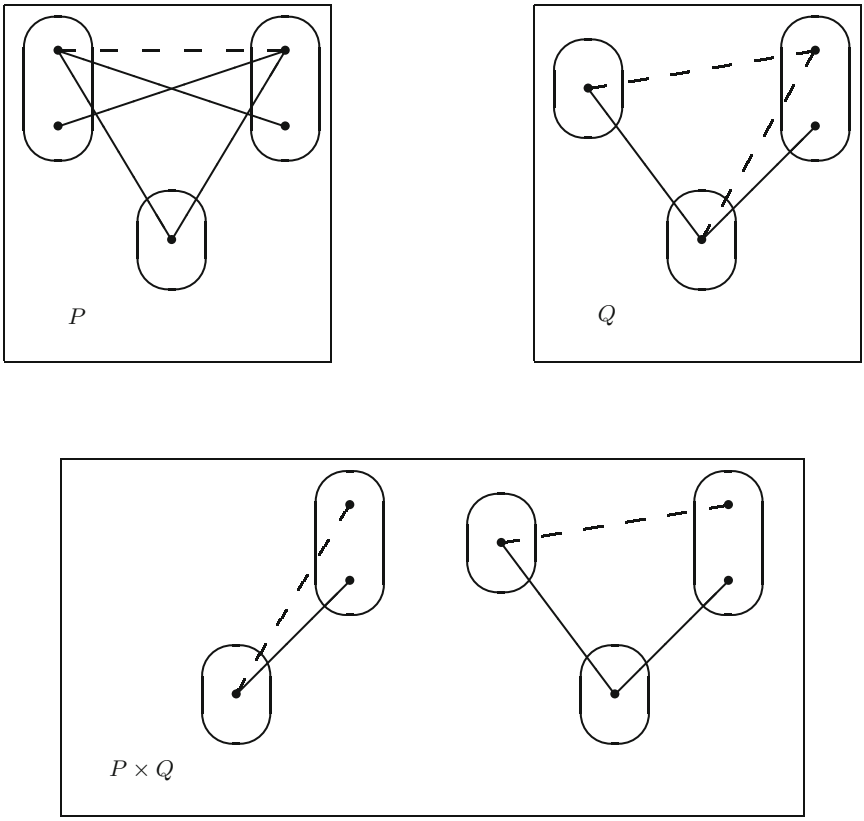

Fig. 2. The operation $P \times Q$.

factors (modulo the equivalence relation $\approx$) and $P \times Q$ is the juxtaposition of these two common factors. Note that $P1$ and $P2$ (shown in Fig. 1) are both common factors of P and Q, but since $P1 \approx P2$ we only need to include one of these patterns in $P \times Q$. If S_1, S_2 are sets of patterns, then $S_1 \times S_2$ is the set of patterns $\{P \times Q \mid P \in S_1 \wedge Q \in S_2\}$.

The following lemmas provide a logical interpretation of the $+$ and $\times$ operations on patterns.

Lemma 1. *For all patterns P_1, P_2, I, we have $P_1 + P_2 \nrightarrow I$ if and only if $(P_1 \nrightarrow I \vee P_2 \nrightarrow I)$*

Proof. For all patterns P_1, P_2, I, $P_1 + P_2 \rightarrow I$ if and only if $(P_1 \rightarrow I \wedge P_2 \rightarrow I)$ by the definition of $P_1 + P_2$. By contraposition, for all patterns P_1, P_2, I, $P_1 + P_2 \nrightarrow I$ if and only if $(P_1 \nrightarrow I \vee P_2 \nrightarrow I)$.

Lemma 2. *For all patterns P, I_1, I_2, $P \nrightarrow I_1 \times I_2$ if and only if $(P \nrightarrow I_1 \vee P \nrightarrow I_2)$.*

Proof. By contraposition, it suffices to show that $P \rightarrow I_1 \times I_2$ if and only if $P \rightarrow I_1 \wedge P \rightarrow I_2$. If $P \rightarrow I_1 \wedge P \rightarrow I_2$, then P is a common factor of I_1 and I_2 and hence $P \rightarrow I_1 \times I_2$. On the other hand, if $P \rightarrow I_1 \times I_2$, then, due to the lack of edges between the connected components of $I_1 \times I_2$, P must be the juxtaposition of patterns $P_1, \ldots, P_r$ where for each $i = 1, \ldots, r$, $P_i \rightarrow R_i$ for some R_i which is one of the connected components of $I_1 \times I_2$. Each connected component R_i of $I_1 \times I_2$ satisfies $R_i \rightarrow R_i'$ for some common factor R_i' of I_1 and I_2. By transitivity of the occurrence relation and by definition of $I_1 \times I_2$, we have $P_i \rightarrow I_1$ and $P_i \rightarrow I_2$ (for $i = 1, \ldots, r$) and hence $P \rightarrow I_1$ and $P \rightarrow I_2$.

3 The Two Lattices

Let $\mathcal{P}$ be the set of all patterns and $\mathcal{I}$ be the set of all generic instances. Let $\mathcal{T}$ be the set of all subsets of $\mathcal{I}$. Let $\mathcal{S}$ be the set of all subsets of $\mathcal{P}$. In this section we show that $\mathcal{S}$ and $\mathcal{T}$ have lattice structures with partial orders based on notions of occurrence. Although $\mathcal{P} = \mathcal{I}$, $\mathcal{S}$ and $\mathcal{T}$ are distinct since they do not have the same partial order.

We require two different definitions of occurrence of one set of patterns in another, depending on whether the sets of patterns are considered as forbidden patterns or sets of generic instances. For $S_1, S_2 \in \mathcal{S}$, we write $S_1 \twoheadrightarrow S_2$ to mean that $\forall Q \in S_2$, $\exists P \in S_1$ such that $P \rightarrow Q$. We write $S_1 \twoheadleftarrow\!\!\!\twoheadrightarrow S_2$ if $S_1 \twoheadrightarrow S_2$ and $S_2 \twoheadrightarrow S_1$. For $T_1, T_2 \in \mathcal{T}$, we write $T_1 \mapsto T_2$ to mean $\forall P \in T_1$, $\exists Q \in T_2$ such that $P \rightarrow Q$. We write $T_1 \leftrightarrow T_2$ if $T_1 \mapsto T_2$ and $T_2 \mapsto T_1$. It follows directly from their definitions that $\twoheadleftarrow\!\!\!\twoheadrightarrow$ and $\leftrightarrow$ are equivalence relations.

Let $\overline{\mathcal{T}}$ be the set of all equivalence classes (according to $\leftrightarrow$) of sets of generic instances. Let $\overline{\mathcal{S}}$ be the set of all equivalence classes (according to $\twoheadleftarrow\!\!\!\twoheadrightarrow$) of sets of forbidden patterns.

It is not difficult to see that $\mapsto$ is a partial order on $\overline{\mathcal{T}}$ and that $\twoheadrightarrow$ is a partial order on $\overline{\mathcal{S}}$. It follows that $\overline{\mathcal{T}}$ and $\overline{\mathcal{S}}$ both have a lattice structure [2,21]. The following proposition shows that the set $\overline{\mathcal{T}}$ has a lattice structure with meet and join operations $\times$ and $\cup$, whereas the set $\overline{\mathcal{S}}$ has a lattice structure with meet and join operations $+$ and $\cup$.

Proposition 1. *For all $S_1, S_2 \in \mathcal{S}$, (1) $S_2 \twoheadrightarrow S_1 \Leftrightarrow S_1 + S_2 \twoheadleftarrow\!\!\!\twoheadrightarrow S_1$ and (2) $S_2 \twoheadrightarrow S_1 \Leftrightarrow S_1 \cup S_2 \twoheadleftarrow\!\!\!\twoheadrightarrow S_2$. For all $T_1, T_2 \in \mathcal{T}$, (3) $T_1 \mapsto T_2 \Leftrightarrow T_1 \times T_2 \leftrightarrow T_1$ and (4) $T_1 \mapsto T_2 \Leftrightarrow T_1 \cup T_2 \leftrightarrow T_2$.*

Proof. (1) $\Rightarrow$: $S_2 \twoheadrightarrow S_1$ means $\forall P \in S_1$, $\exists Q \in S_2$ such that $Q \rightarrow P$ and hence $P + Q \rightarrow P$. Thus $S_1 + S_2 \twoheadrightarrow S_1$. Clearly $S_1 \twoheadrightarrow S_1 + S_2$.
(1) $\Leftarrow$: $S_1 + S_2 \twoheadrightarrow S_1$ means $\forall P \in S_1$, $\exists R + Q \in S_1 + S_2$ such that $R + Q \rightarrow P$ which implies $Q \rightarrow P$. Hence $S_2 \twoheadrightarrow S_1$.
(2) $\Rightarrow$: $S_2 \twoheadrightarrow S_1$ means $\forall P \in S_1$, $\exists Q \in S_2$ such that $Q \rightarrow P$. Now, since $\forall Q$, $Q \rightarrow Q$, we have $\forall R \in S_1 \cup S_2$, $\exists Q \in S_2$ such that $Q \rightarrow R$. Hence $S_2 \twoheadrightarrow S_1 \cup S_2$. Clearly $S_1 \cup S_2 \twoheadrightarrow S_2$.
(2) $\Leftarrow$: $S_2 \twoheadrightarrow S_1 \cup S_2$ implies that $\forall P \in S_1$, $\exists Q \in S_2$ such that $Q \rightarrow P$ and so $S_2 \twoheadrightarrow S_1$.

(3) $T_1 \mapsto T_2$ means that $\forall I \in T_1$, $\exists J \in T_2$ such that $I \to J$, so I is a common factor of I and J and hence $I \to I \times J$. Thus $T_1 \mapsto T_1 \times T_2$. Thus, by definition of $\times$, $T_1 \times T_2 \mapsto T_1$.
(3) $\Leftarrow$: $T_1 \mapsto T_1 \times T_2$ means that $\forall I \in T_1$, $\exists I \times J \in T_1 \times T_2$ such that each connected component of I occurs in a common factor of I and J, and hence each connected component of I occurs in J and so $I \to J$. Thus $T_1 \mapsto T_2$.
(4) $\Rightarrow$: $T_1 \mapsto T_2$ means $\forall I \in T_1$, $\exists J \in T_2$ such that $I \to J$. Thus $T_1 \cup T_2 \mapsto T_2$. Clearly $T_2 \mapsto T_1 \cup T_2$.
(4) $\Leftarrow$: $T_1 \cup T_2 \to T_2$ implies $\forall I \in T_1$, $\exists J \in T_2$ such that $I \to J$ which is exactly $T_1 \mapsto T_2$.

The following lemmas are not essential for the lattice structure of $\overline{\mathcal{S}}$ and $\overline{\mathcal{T}}$, but will be useful later.

Lemma 3. *If $S_1 \supseteq S_2$ then $S_1 \twoheadrightarrow S_2$. If $T_1 \subseteq T_2$ then $T_1 \mapsto T_2$.*

Proof. If $S_1 \supseteq S_2$ then $\forall Q \in S_2$, $\exists P = Q \in S_1$ such that $P \to Q$. If $T_1 \subseteq T_2$ then $\forall P \in T_1$, $\exists Q = P \in T_2$ such that $P \to Q$.

Lemma 4. *For all sets of patterns S_1, S_2, $S_1 + S_2 \twoheadrightarrow S_1 \cap S_2$ and $S_1 \cap S_2 \mapsto S_1 \times S_2$.*

Proof. We have $\forall P \in S_1 \cap S_2$, $P \twoheadleftarrow\!\!\twoheadrightarrow P + P \in S_1 + S_2$. Hence $S_1 + S_2 \twoheadrightarrow S_1 \cap S_2$. Also $\forall I \in S_1 \cap S_2$, $I \leftrightarrow I \times \in S_1 \times S_2$. Hence $S_1 \cap S_2 \mapsto S_1 \times S_2$.

If we consider that $S_1 \leq S_2$ if $S_2 \twoheadrightarrow S_1$, then the minimal element in the lattice $\overline{\mathcal{S}}$ is the empty set of patterns and the maximal element is $\{P_\emptyset\}$ where $P_\emptyset$ is the pattern containing no points or edges. If we consider that $T_1 \leq T_2$ if $T_1 \mapsto T_2$ then the minimal element of $\overline{\mathcal{T}}$ is the empty set of patterns and the maximal element is the set of all patterns.

The two lattices $\overline{\mathcal{S}}$ and $\overline{\mathcal{T}}$ are both distributive, as shown by the following proposition.

Proposition 2. *For all $S_1, S_2, S_3 \in \mathcal{S}$, we have $S_1 + (S_2 \cup S_3) \twoheadleftarrow\!\!\twoheadrightarrow (S_1 + S_2) \cup (S_1 + S_3)$ and for all $T_1, T_2, T_3 \in \mathcal{T}$, we have $T_1 \cup (T_2 \times T_3) \leftrightarrow (T_1 \times T_2) \cup (T_1 \times T_3)$.*

Proof. These follow immediately from the definitions.

4 The Galois Connection

The Galois connection is based on two functions $f : \mathcal{S} \to \mathcal{T}$ and $g : \mathcal{T} \to \mathcal{S}$, defined as follows.

$$f(S) = \{I \in \mathcal{I} \mid \forall P \in S, P \twoheadrightarrow I\}$$
$$g(T) = \{P \in \mathcal{P} \mid \forall I \in T, P \twoheadrightarrow I\}$$

Theorem 1. *There is an antitone Galois connection between $\overline{\mathcal{S}}$ and $\overline{\mathcal{T}}$.*

Proof. The functions f, g, applied to equivalence classes of $\mathcal{S}$ and $\mathcal{T}$ define a Galois connection between $\overline{\mathcal{S}}$ and $\overline{\mathcal{T}}$ if $\forall S \in \mathcal{S}$, $\forall T \in \mathcal{T}$, $T \leq f(S) \Leftrightarrow S \leq g(T)$. This corresponds to $(T \mapsto f(S)) \Leftrightarrow (g(T) \twoheadrightarrow S)$, which holds because $(T \mapsto f(S))$ and $(g(T) \twoheadrightarrow S)$ are both equivalent to $\forall P \in S$, $\forall I \in T$, $P \nrightarrow I$. We therefore have a Galois connection between $\overline{\mathcal{S}}$ and $\overline{\mathcal{T}}$.

We now study this Galois connection in more detail.

Proposition 3. *For all $S_1, S_2 \in \mathcal{S}$, if $S_1 \twoheadrightarrow S_2$ then $f(S_1) \subseteq f(S_2)$. For all $T_1, T_2 \in \mathcal{T}$, if $T_1 \mapsto T_2$ then $g(T_2) \subseteq g(T_1)$.*

Proof. Suppose $S_1 \twoheadrightarrow S_2$. Then $\forall P_2 \in S_2$, $\exists P_1 \in S_1$ such that $P_1 \twoheadrightarrow P_2$. Consider $I \in f(S_1)$. By definition of f, $\forall P_1 \in S_1$, $P_1 \nrightarrow I$. It follows that $I \in f(S_2)$ since otherwise we would have some $P_2 \in S_2$ such that $P_2 \rightarrow I$ and some $P_1 \in S_1$ with $P_1 \rightarrow P_2 \rightarrow I$ which contradicts $P_1 \nrightarrow I$.

Suppose $T_1 \mapsto T_2$. Then $\forall I_1 \in T_1$, $\exists I_2 \in T_2$ such that $I_1 \rightarrow I_2$. Consider $P \in f(T_2)$. By definition of g, $\forall I_2 \in T_2$, $P \nrightarrow I_2$. It follows that $P \in g(T_1)$ since otherwise we would have some $I_1 \in T_1$ such that $P \rightarrow I_1$ and some $I_2 \in T_2$ such that $P \rightarrow I_1 \rightarrow I_2$ which contradicts $P \nrightarrow I_2$.

We immediately have the following corollary.

Corollary 1. *For all $S_1, S_2 \in \mathcal{S}$, $S_1 \twoheadrightarrow S_2 \Rightarrow f(S_1) \mapsto f(S_2)$. For all $T_1, T_2 \in \mathcal{T}$, $T_1 \mapsto T_2 \Rightarrow g(T_1) \twoheadrightarrow g(T_2)$.*

Proposition 4. *For any patterns S_1, S_2, $f(S_1) = f(S_2)$ if and only if $S_1 \twoheadleftarrow\!\!\twoheadrightarrow S_2$.*

Proof. Suppose $f(S_1) = f(S_2)$. Then $\forall I$, $(\forall P \in S_1, P \nrightarrow I) \Leftrightarrow (\forall P \in S_2, P \nrightarrow I)$. This is equivalent to $\forall I$, $(\exists P \in S_1, P \rightarrow I) \Leftrightarrow (\exists P \in S_2, P \rightarrow I)$. It follows, by setting $I = P \in S_2$, that $\forall P \in S_2$, $\exists P' \in S_1$ such that $P' \rightarrow P$, and hence $S_1 \twoheadrightarrow S_2$. By setting $I = P \in S_1$, by a symmetrical argument, we obtain $S_2 \twoheadrightarrow S_1$, and hence $S_1 \twoheadleftarrow\!\!\twoheadrightarrow S_2$.

Now suppose that $S_1 \twoheadleftarrow\!\!\twoheadrightarrow S_2$. Then, by Proposition 3, we can deduce that $f(S_1) = f(S_2)$.

It is important to observe that $\mathcal{T}$ includes sets of partially-specified instances. If we considered only sets of completely-specified instances in $\mathcal{T}$, then Proposition 4 would not hold. For example, consider S_1 and S_2 shown in Fig. 3. It is easy to see that we do not have $S_1 \twoheadrightarrow S_2$, even though S_1 and S_2 define the same set of completely-specified instances when forbidden, namely those instances which have only positive edges or only negative edges. They do not define the same set of *generic instances*, since, for example, the single pattern $Q \in S_2$ is in $f(S_1)$ but not $f(S_2)$.

Proposition 5. *For any patterns T_1, T_2, $g(T_1) = g(T_2)$ if and only if $T_1 \leftrightarrow T_2$.*

Proof. Suppose $g(T_1) = g(T_2)$. Then $\forall P$, $(\forall I \in T_1, P \nrightarrow I) \Leftrightarrow (\forall I \in T_2, P \nrightarrow I)$. This is equivalent to $\forall P$, $(\exists I \in T_1, P \rightarrow I) \Leftrightarrow (\exists I \in T_2, P \rightarrow I))$. Setting $P = I \in T_1$, we obtain $\forall I \in T_1$, $\exists I' \in T_2$ such that $I \rightarrow I'$, and hence $T_1 \mapsto T_2$.

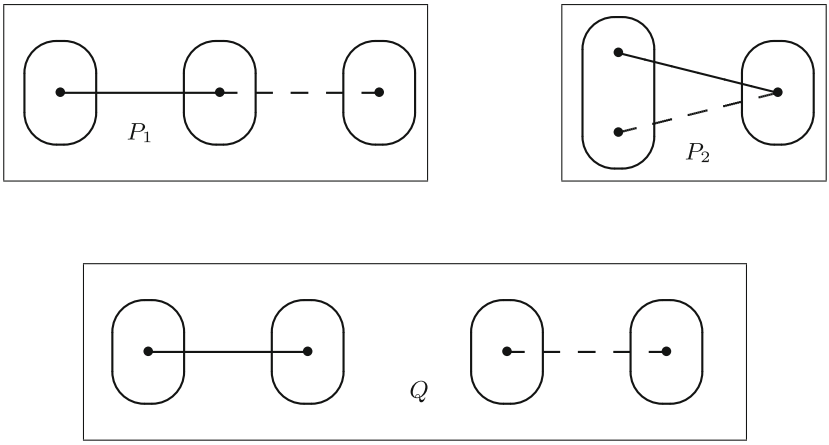

Fig. 3. The sets of patterns $S_1 = \{P_1, P_2\}$ and $S_2 = \{Q\}$ define the same set of completely specified instances when forbidden, but $f(S_1) \neq f(S_2)$.

Setting $P = I \in T_2$, by a symmetrical argument, we obtain $T_2 \mapsto T_1$, and hence $T_1 \leftrightarrow T_2$.

Now suppose that $T_1 \leftrightarrow T_2$. By Proposition 3, we can deduce that $g(T_1) = g(T_2)$.

We now show to what extent the lattice structure of $\mathcal{S}$ and $\mathcal{T}$ is preserved via the mappings f and g.

Theorem 2. $\forall S_1, S_2 \in \mathcal{S},\ f(S_1) \cup f(S_2) = f(S_1 + S_2)$.

Proof. For $i = 1, 2$, $f(S_i) = \{I \mid \forall P \in S_i, P \nrightarrow I\}$. So $f(S_1) \cup f(S_2) = \{I \mid (\forall P \in S_1, P \nrightarrow I) \vee (\forall P \in S_2, P \nrightarrow I)\} = \{I \mid \forall P_1 \in S_1, \forall P_2 \in S_2 (P_1 \nrightarrow I \vee P_2 \nrightarrow I)\}$. Thus, by Lemma 1, $f(S_1) \cup f(S_2) = \{I \mid (\forall P_1 \in S_1, \forall P_2 \in S_2 (P_1 + P_2 \nrightarrow I)\} = \{I \mid \forall P_1 + P_2 \in S_1 + S_2 (P_1 + P_2 \nrightarrow I)\} = f(S_1 + S_2)$.

Theorem 3. $\forall S_1, S_2 \in \mathcal{S},\ f(S_1) \cap f(S_2) = f(S_1 \cup S_2)$.

Proof. $f(S_1 \cup S_2) = \{I \mid \forall P \in S_1 \cup S_2, P \nrightarrow I\} = \{I \mid \forall P_1 \in S_1, P \nrightarrow I\} \cap \{I \mid \forall P_2 \in S_2, P \nrightarrow I\} = f(S_1) \cap f(S_2)$.

The lattice structure and Theorems 2 and 3 are illustrated in Fig. 4.

Theorem 4. $\forall T_1, T_2 \in \mathcal{T},\ g(T_1) \cap g(T_2) = g(T_1 \cup T_2)$.

Proof. $g(T_1 \cup T_2) = \{P \mid \forall I \in T_1 \cup T_2, P \nrightarrow I\} = \{P_1 \mid \forall I \in T_1, P_1 \nrightarrow I\} \cap \{P_2 \mid \forall I \in T_2, P_2 \nrightarrow I\} = g(T_1) \cap g(T_2)$

Theorem 5. $\forall T_1, T_2 \in \mathcal{T},\ g(T_1) \cup g(T_2) = g(T_1 \times T_2)$.

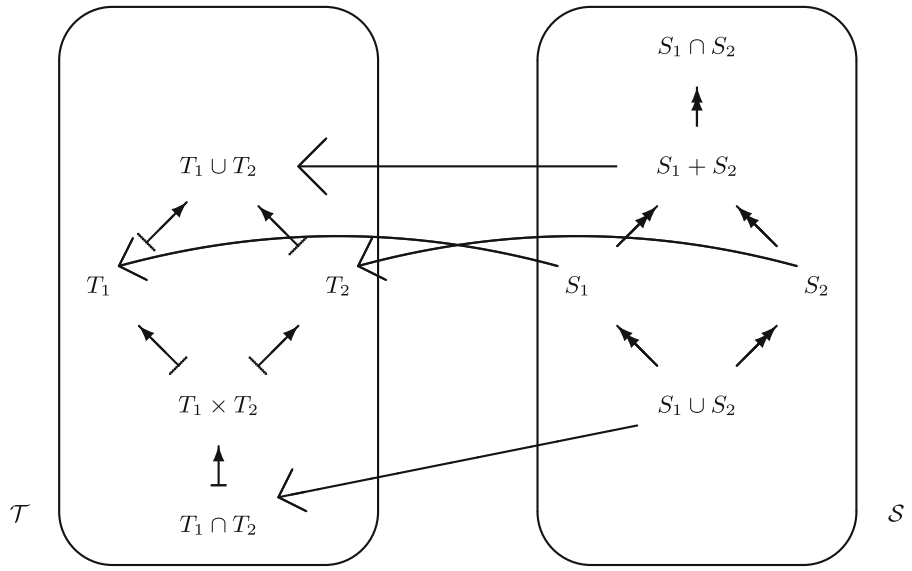

Fig. 4. The function f from $\mathcal{S}$ to $\mathcal{T}$

Proof. $g(T_1 \times T_2) = \{P \mid \forall I \in T_1 \times T_2, P \nrightarrow I\} = \{P \mid \forall I_1 \in T_1, \forall I_2 \in T_2, P \nrightarrow I_1 \times I_2\}$. By Lemma 2, this is equal to $\{P \mid \forall I_1 \in T_1, \forall I_2 \in T_2, (P \nrightarrow I_1 \vee P \nrightarrow I_2)\}$ $= \{P \mid \forall I_1 \in T_1, P \nrightarrow I_1\} \cup \{P \mid \forall I_2 \in T_2, P \nrightarrow I_2\} = g(T_1) \cup g(T_2)$.

Theorems 4 and 5 are illustrated in Fig. 5.

Definition 2. *A set T of patterns is* downward-closed *if for all patterns P, Q, $(P \rightarrow Q) \wedge (Q \in T) \Rightarrow (P \in T)$. A set of patterns S is* upward-closed *if for all patterns P, Q, $(P \rightarrow Q) \wedge (P \in S) \Rightarrow (Q \in S)$.*

In the case of upward-closed sets of forbidden patterns and/or downward-closed sets of generic instances, the lattices, and the corresponding Galois connection, become simpler as the following proposition shows. In this case the two lattices become lattices of sets with meet and join operations $\cap$ and $\cup$. In practice, however, we are generally interested in small sets of forbidden patterns which cannot be upward-closed (otherwise they would be infinite).

Proposition 6. *If S_1, S_2 are upward-closed, then $S_1 + S_2 \twoheadleftarrow\!\!\!\rightarrow S_1 \cap S_2$. If T_1, T_2 are downward-closed, then $T_1 \cap T_2 \leftrightarrow T_1 \times T_2$.*

Proof. $\forall P + Q \in S_1 + S_2$, we have $P \rightarrow P + Q$ and $Q \rightarrow P + Q$. By the upward closedness of both S_1 and S_2, it follows that $P + Q \in S_1 \cap S_2$. Thus $S_1 \cap S_2 \twoheadrightarrow S_1 + S_2$. By Lemma 4, we have $S_1 + S_2 \twoheadleftarrow\!\!\!\rightarrow S_1 \cap S_2$.

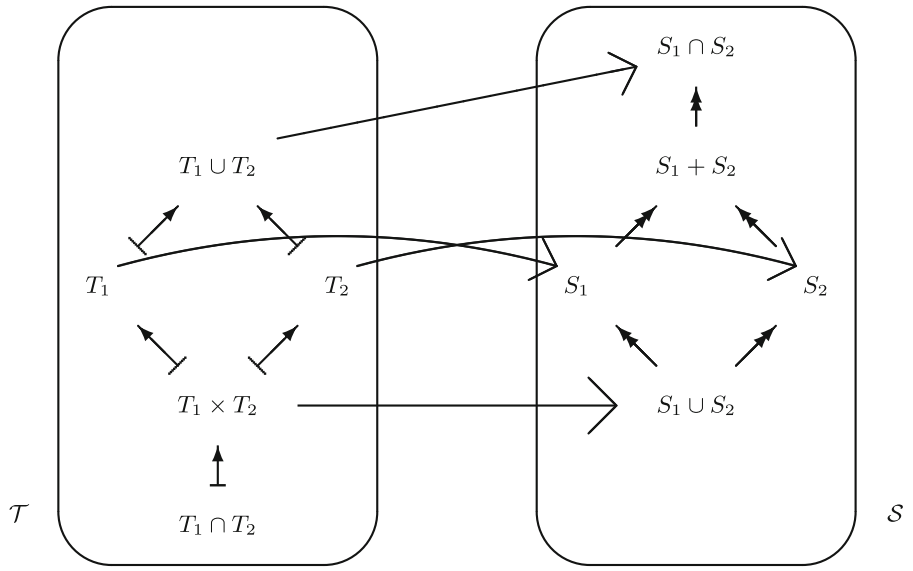

Fig. 5. The function g from $\mathcal{T}$ to $\mathcal{S}$

$\forall P \times Q \in T_1 \times T_2$, $P \times Q \to P$ and $P \times Q \to Q$. If T_1, T_2 are downward-closed, then $P \times Q \in T_1 \cap T_2$. Thus $T_1 \times T_2 \mapsto T_1 \cap T_2$. By Lemma 4, we have $T_1 \cap T_2 \leftrightarrow T_1 \times T_2$.

5 Tractability Consequences of the Galois Connection

In this section we show that tractable sets of patterns form a sublattice of $\overline{\mathcal{S}}$.

Recall that we say that $T \in \mathcal{T}$ is tractable if there is a polynomial-time algorithm to decide all completely-specified instances in T. We consider that incompletely-specified instances (i.e. generic instances with at least one pair of points not joined by a (positive or negative) edge) can be recognised as such in polynomial time and hence do not affect the tractability of T. A consequence of this is that it is not true that $T_1 \mapsto T_2 \wedge (T_2 \text{ tractable}) \Rightarrow T_1$ tractable. For example, T_2 could be trivially tractable because it contains no completely-specified instance even when T_1 is the set of all binary CSP instances. However, we have the following important result.

Proposition 7. *If* $T_1 = f(S_1)$ *and* $T_2 = f(S_2)$, *then* $(T_1 \mapsto T_2 \wedge (T_2$ *tractable*$)) \Rightarrow T_1$ *tractable.*

Proof. Let $T_1 = f(S_1)$ and $T_2 = f(S_2)$, where $T_1 \mapsto T_2$. By Proposition 3, we have $g(T_2) \subseteq g(T_1)$ and so by Lemma 3, $g(T_1) \twoheadrightarrow g(T_2)$. By definition of the

functions f and g, we have $f(g(f(S))) = f(S)$ for all S, and so $f(g(T_1)) = f(S_1)$ and $f(g(T_2)) = f(S_2)$. It follows from Proposition 4 that $S_1 \twoheadleftarrow\!\!\!\rightarrow g(T_1)$ and $S_2 \twoheadleftarrow\!\!\!\rightarrow g(T_2)$. Thus $S_1 \twoheadleftarrow\!\!\!\rightarrow g(T_1) \twoheadrightarrow g(T_2) \twoheadleftarrow\!\!\!\rightarrow S_2$. By transitivity of $\twoheadrightarrow$, we have $S_1 \twoheadrightarrow S_2$ and, by Proposition 3, $T_1 = f(S_1) \subseteq f(S_2) = T_2$. It follows that if T_2 is tractable, then so is T_1.

This means that it may be possible to classify the complexity of all classes $f(S)$ for all finite sets $S \in \mathcal{S}$. Indeed we conjecture that there is a P/NP-complete dichotomy. This has already been proved for sets of patterns containing only negative edges [9].

The following proposition tells us that the tractable sets of patterns form a sub-lattice of $\overline{\mathcal{S}}$.

Proposition 8. *If S_1, S_2 are tractable sets of patterns, then so are $S_1 \cup S_2$ and $S_1 + S_2$.*

Proof. $f(S_1 + S_2) = f(S_1) \cup f(S_2)$ and hence can be solved in polynomial time if $f(S_1)$ and $f(S_2)$ can be. A similar remark holds for $f(S_1 \cup S_2) = f(S_1) \cap f(S_2)$.

We can observe that the finite sets of $\mathcal{S}$ form a sublattice of $\mathcal{S}$ since $S_1 + S_2$ and $S_1 \cup S_2$ are finite if S_1, S_2 are finite. It follows that the finite tractable sets of $\mathcal{S}$ form a sublattice. We are particularly interested in finite sets of patterns, since detecting the absence of finite sets of patterns can be achieved in polynomial time, whereas testing the absence of an infinite set of patterns may not even by computable. We can observe that there are infinite sets of patterns S such that $f(S)$ is tractable but for no finite subset S' of S is $f(S')$ tractable, e.g. acyclic instances that can be defined by forbidding cycles of all lengths but by no finite set of flat patterns [11].

6 Augmented Patterns: Motivation

We can make the language of patterns much richer by adding relations to patterns (and possibly quantifying over these relations). A *flat* pattern (the kind of pattern we have studied up to now in this paper) has only the binary relations of compatibility between points (positive edges), incompatibility between points (negative edges) and the equivalence relation between points corresponding to assignments to the same variable (represented in figures by ovals representing its equivalence classes). Suppose that we add a new relation, such as an ordering or a colouring of the points of the pattern. We call this an *augmented pattern*. In this section, we motivate the study of augmented patterns by showing that they can be used to define interesting tractable classes that cannot be defined using flat patterns. Examples of such augmented patterns are a pattern in which we add an ordering between points (the new relation is binary) or a colouring of points (in which case the new relation is unary). For these new relations to be meaningful, they must satisfy the basic properties of, for example, orderings or colourings. To impose this we can replace a single pattern P by a set of patterns,

one being the augmented pattern P and the others designed in such a way as to impose the required properties of the new relation.

Consider a binary relation $R_<$. Each of the following three statements can be seen as an augmented pattern involving only the relation $R_<$:

$$R_<(a, a) \tag{1}$$
$$R_<(a, b) \wedge R_<(b, a) \tag{2}$$
$$R_<(a, b) \wedge R_<(b, c) \wedge R_<(c, a) \tag{3}$$

By forbidding these three patterns, we impose that $R_<$ is an irreflexive, anti-symmetric relation with no length-3 cycles. In the following we only consider instances in which $R_<$ is total in the sense that for all distinct a, b, we have $R_<(a, b)$ or $R_<(b, a)$. It is easy to see that this implies that $R_<$ is a strict total order (since, in particular, forbidding pattern (3) corresponds to transitivity). From now on, for notational convenience, we use the operator $<$ instead of the relation $R_<$, i.e. we write $a < b$ instead of $R_<(a, b)$. If we also forbid the augmented pattern shown in Fig. 6(a), then we not only impose an order on the points of an instance, but we also impose that there is a corresponding order on the variables which is consistent with this order on the points.

If we also forbid the augmented pattern in Fig. 6(b), then we are saying that there is a total ordering of the variables of the instance such that each variable is constrained by at most one previous variable in this order. The set of completely-specified instances with a total ordering on its points in which none of these five augmented patterns occurs corresponds exactly to the set of instances whose constraint graph is acyclic. It is well known that this class of binary CSP instances is tractable since it is solved by arc consistency [22]. Recall that no finite set of forbidden *flat* patterns defines the set of acyclic instances [11]. This example demonstrates the power of augmented patterns compared to flat patterns, since acyclicity can be defined by forbidding a set of just five augmented patterns.

In fact, for any fixed $k \geq 1$, we can define the class of instances with tree-width bounded by k using a finite set of augmented patterns. We saw above that the patterns (1), (2), (3) together with the pattern shown in Fig. 6(a) effectively allows us to impose an order on variables. Apart from this variable-order relation, we also introduce another binary relation IE (for Induced Edge between two variables in the constraint graph) which, using the same idea as in Fig. 6(a), is also effectively a relation on variables. For simplicity of presentation, in the following, we apply $<$ and IE to variables rather than points. We also require the relation $\overline{IE}$ and we will consider only those instances in which IE and $\overline{IE}$ cover all pairs of variables. To ensure that $\overline{IE}$ is the complement of IE we forbid the augmented pattern

$$IE(x, y) \ \wedge \ \overline{IE}(x, y)$$

The semantics of the induced-edge relation IE is given by the following rules:

1. IE is symmetric.
2. If there is a negative edge between variables x and y, then $IE(x, y)$.

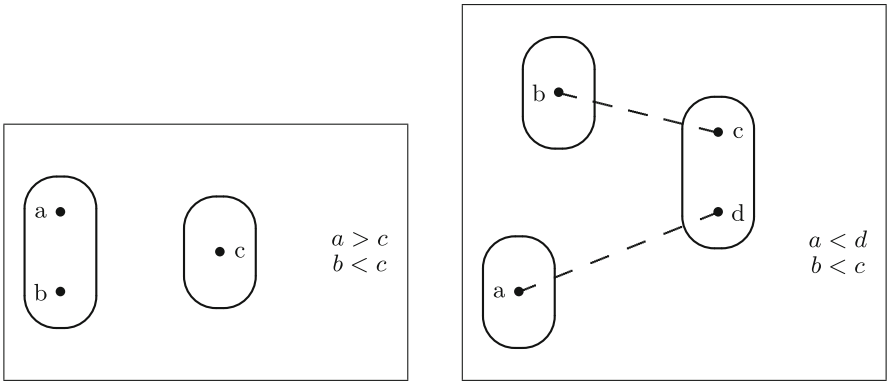

Fig. 6. Examples of augmented patterns.

3. If $x < z$, $y < z$, $IE(x, z)$ and $IE(y, z)$, then $IE(x, y)$.

These rules can easily be coded using forbidden augmented patterns involving $<$, IE and $\overline{IE}$. Symmetry is coded by the forbidden pattern

$$IE(x, y) \ \wedge \ \overline{IE}(y, x)$$

Rule 2, above, can be imposed by forbidding the augmented pattern shown in Fig. 7. Rule 3 can be coded by the forbidden pattern:

$$(x < z) \ \wedge \ (y < z) \ \wedge \ IE(x, z) \ \wedge \ IE(y, z) \ \wedge \ \overline{IE}(x, y)$$

In order to impose a bound of k on the tree-width of the constraint graph, there must exist a total variable order and relations IE, $\overline{IE}(x, y)$ (that cover all pairs of variables) such that the following augmented pattern does not occur:

$$(x_1 < z) \wedge \ldots \wedge (x_{k+1} < z) \wedge IE(x_1, z) \wedge \ldots \wedge IE(x_{k+1}, z)$$

This corresponds to a well-known characterisation of graphs with bounded tree-width as subgraphs of k-trees [22,24]. This example illustrates the fact that we need to apply a filter to the set of instances I defined by forbidding a set of augmented patterns. In this case, the filter is that I is completely specified, $<$ is a total order on variables and IE, $\overline{IE}$ form a cover. When defining tractability of augmented patterns, we are only concerned in deciding instances satisfying the filter.

Another example which motivates the use of augmented patterns is the study of tractable languages. All known tractable constraint languages are defined by the existence of a polymorphism (a pointwise closure operation) which guarantees tractability [27]. Indeed, tractability is guaranteed by the identities satisfied by the polymorphism [4]. The existence of a polymorphism satisfying any given

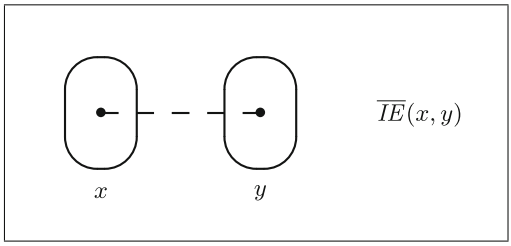

Fig. 7. An augmented pattern.

set of identities can be stated in terms of a forbidden augmented pattern. Indeed, an augmented pattern can enforce the fact that the constraints of the instance must all have a polymorphism f and other patterns can enforce the identities that f must satisfy. By existentially quantifying over f we can then define the class of all instances whose constraints all have some majority polymorphism f, for example, or all of whose constraints have a Siggers polymorphism [39].

We illustrate this for weak near-unanimity polymorphisms, given their importance in the characterisation of tractable languages [3,41]. A binary CSP instance I has the k-ary polymorphism $f : \mathcal{D}^k \to \mathcal{D}$ if for all binary relations R of I we have $\forall (a_1, b_1), \ldots, (a_k, b_k) \in R$, $(f(a_1, \ldots, a_k), f(b_1, \ldots, b_k)) \in R$. The first step to expressing the fact that a binary CSP instance has the k-ary polymorphism f is to forbid the augmented pattern $\text{POLY}_k(f)$ shown in Fig. 8 for the case $k = 4$. A weak near-unanimity operation is a function $f : \mathcal{D}^k \to \mathcal{D}$ satisfying the identities $f(b, a, \ldots, a) = f(a, b, a, \ldots, a) = \ldots = f(a, \ldots, a, b)$. These identities are equivalent to forbidding each of the following augmented patterns

$$\begin{gathered}
(f(b, a, \ldots, a) = c) \wedge (f(a, b, a, \ldots, a) = d) \wedge (c \neq d) \\
(f(b, a, \ldots, a) = c) \wedge (f(a, a, b, a, \ldots, a) = d) \wedge (c \neq d) \\
\vdots \\
(f(b, a, \ldots, a) = c) \wedge (f(a, \ldots, a, b) = d) \wedge (c \neq d)
\end{gathered}$$

For some fixed k, after forbidding these augmented patterns (the polymorphism pattern $\text{POLY}_k(f)$ as illustrated in Fig. 8 together with the above patterns corresponding to the identities of a weak near-unanimity polymorphism of arity k), we obtain a set of instances. We then have to apply a filter so that we only keep those instances $I = \langle A_I, \rho_I \rangle$ in which f is a total function and such that all domains are closed under f, i.e. for all $x \in \mathcal{X}$ and for all $a_1, \ldots, a_k \in \mathcal{D}$ such that $(x, a_i) \in A_I$ $(i = 1, \ldots, k)$, we have $(x, f(a_1, \ldots, a_k)) \in A_I$. This example again illustrates the fact that tractability of augmented patterns depends on the existence of a polynomial-time algorithm to decide instances satisfying the corresponding filter.

Another motivating example involves a colouring of points. Suppose that both S_1 and S_2 are tractable sets of flat patterns. Then we know that $S_1 + S_2$ defines the tractable class of instances in which either S_1 does not occur or S_2

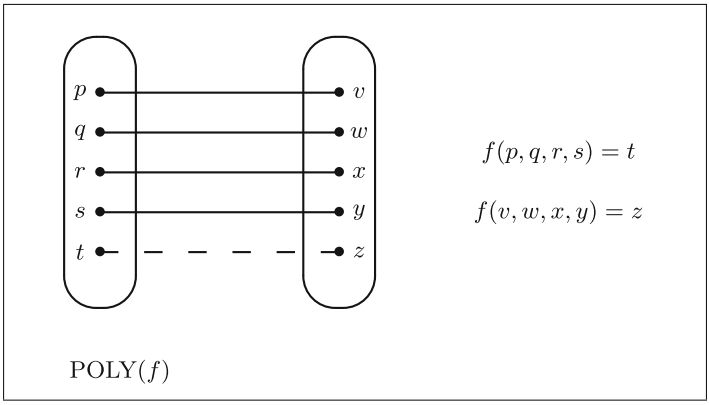

Fig. 8. Polymorphisms can be defined by forbidding augmented patterns, as illustrated for this arity-4 polymorphism f.

does not occur. The number of patterns in $S_1 + S_2$ is (in the worst case) quadratic in the size of S_1 and S_2. We can give a set of augmented patterns which is linear in the size of S_1 and S_2 as follows. We augment each pattern in S_1 by colouring all its points red and each pattern in S_2 by colouring all its points green. We then add a pattern consisting of two points, one red and the other green. The set of instances for which there is a 2-colouring of its points in which none of these augmented patterns occurs is exactly the set of instances in $f(S_1) \cup f(S_2)$.

7 Augmented Patterns: Definitions

An *augmented pattern* is simply a flat pattern together with a conjunction of atomic formulas such as $R_i(p_1, \ldots, p_{a_i})$ where each R_i is a relation (of arity a_i) and $p_1, \ldots, p_{a_i}$ are points. An augmented pattern P occurs in another augmented pattern Q if there is a mapping from P to Q which corresponds to the occurrence of the flat version of P in the flat version of Q and which also preserves the new relation(s) R_i. The new relation(s) R_i may, for example, correspond to an order. As an example, the augmented pattern in Fig. 9(a) does not occur in the augmented pattern in Fig. 9(b) since the variable order is not preserved. On the other hand, the pattern $P1$ in Fig. 1 does occur in Fig. 9(b) since there is no variable order in $P1$ to preserve.

As a starting point, we can consider instances augmented with one or more new relation(s). In other words we consider structured instances (e.g. instances with an order on the variables). As usual, in order to establish a Galois connection, we have to consider the lattice of all generic instances including partially-specified instances (partial in the sense that certain pairs of points are joined by neither a negative nor a positive edge *or* the new relations do not form a cover, e.g. the variable order is only partial). The operations $\times$ and $+$ and the functions f and g are defined as for sets of flat patterns. In particular, in $P + Q$

there is no relation (e.g. no variable ordering) between the copies of P and Q in $P + Q$. The two lattice structures and the Galois connection between them follow from exactly the same arguments as for flat patterns.

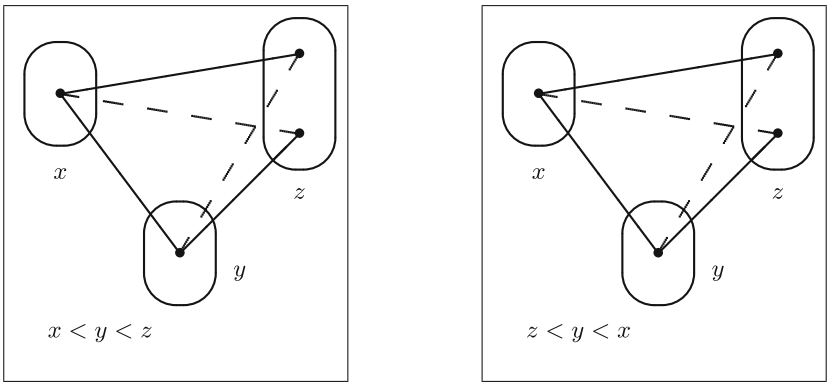

Fig. 9. (a) The broken-triangle pattern (BTP). (b) An alternative pattern which defines the same class.

However, our aim is to consider the existential quantification of the relations (variable ordering, polymorphism, colouring) associated with a (set of) augmented pattern(s). As an example of an augmented pattern, consider the broken-triangle pattern (BTP) [16] shown in Fig. 9(a). We associate with this pattern all instances for which there is some variable ordering for which BTP does not occur. It turns out that, in the case of BTP, it is decidable in polynomial time whether such a variable ordering exists [16]. In general, each structured instance (e.g. an instance with new relations such as a variable ordering) has a corresponding flat version in which the new relations are forgotten, and our aim is to establish a Galois connection between sets of *flat* instances and augmented patterns.

We would like to establish a Galois connection between the set of sets of flat generic instances $\mathcal{T}$ and the set of sets of augmented patterns which we denote by $\mathcal{S}_\mathcal{A}$. However, this does not seem possible. Instead we present in Sect. 8 a Galois connection between $\mathcal{T}$ and Σ_A the set of sets of sets of augmented patterns. Each $\sigma \in \Sigma_A$ is a set of the form $\{S_1, S_2, \ldots\}$ where each $S_i \in \mathcal{S}_\mathcal{A}$ is a set of patterns. Observe that since every element S of $\mathcal{S}_\mathcal{A}$ has a corresponding singleton element $\{S\}$ in Σ_A, we can consider Σ_A as an extension of $\mathcal{S}_\mathcal{A}$. We extend our definition of $\twoheadrightarrow$ from $\mathcal{S}_\mathcal{A}$ to Σ_A as follows: $\sigma_1 \twoheadrightarrow \sigma_2$ if $\forall S_2 \in \sigma_2$, $\exists S_1 \in \sigma_1$ such that $S_1 \twoheadrightarrow S_2$. We define $\overline{\Sigma}_A$ to be the set of equivalence classes with respect to the equivalence relation $\twoheadleftarrow\!\!\!\!\twoheadrightarrow$ in Σ_A.

We first have to understand the lattice structure of $\langle \overline{\Sigma}_A, \leq \rangle$, where $\sigma_1 \leq \sigma_2$ if and only if $\sigma_2 \twoheadrightarrow \sigma_1$. The meet and join operations of this lattice are the operations $+$ and $\cup$. This follows from the following lemmas.

Lemma 5. *For $\sigma, \sigma_1, \sigma_2 \in \overline{\Sigma}_A$, if $\sigma_1 \twoheadrightarrow \sigma$ and $\sigma_2 \twoheadrightarrow \sigma$ then $(\sigma_1 + \sigma_2) \twoheadrightarrow \sigma$.*

Proof. Suppose that $\sigma_1 \twoheadrightarrow \sigma$ and $\sigma_2 \twoheadrightarrow \sigma$ and consider any $S \in \sigma$. We have $\exists S_i \in \sigma_i$ such that $S_i \twoheadrightarrow S$ $(i = 1, 2)$. So $\forall P \in S$, $\exists P_i \in S_i$ such that $P_i \to P$ $(i = 1, 2)$. Thus $P_1 + P_2 \to P$ and hence $S_1 + S_2 \twoheadrightarrow S$. It follows that $(\sigma_1 + \sigma_2) \twoheadrightarrow \sigma$.

Lemma 6. *For $\sigma, \sigma_1, \sigma_2 \in \overline{\Sigma}_A$, if $\sigma \twoheadrightarrow \sigma_1$ and $\sigma \twoheadrightarrow \sigma_2$ then $\sigma \twoheadrightarrow (\sigma_1 \cup \sigma_2)$.*

Proof. If $\sigma \twoheadrightarrow \sigma_1$ and $\sigma \twoheadrightarrow \sigma_2$, then $\forall S_i \in \sigma_i$, $\exists S \in \sigma$ such that $S \twoheadrightarrow S_i$ $(i = 1, 2)$. Hence, $\sigma \twoheadrightarrow (\sigma_1 \cup \sigma_2)$.

We fix a relational signature. Indeed, for simplicity of presentation, in the following we assume that there is a single new relation *Rel* of a fixed arity a (which could be the cartesian product of several relations). We denote by $\mathcal{REL}$ the set of all possible functions from the set of (flat) instances to the set of relations of arity a. Thus, given a flat instance $I \in \mathcal{I}$ and a function $Rel \in \mathcal{REL}$, $\langle I, Rel(I) \rangle$ is an augmented version of I (e.g. the instance I with an ordering on its variables). We can now define occurrence of a set $S \in \mathcal{S}_{\mathcal{A}}$ of augmented patterns in an instance $I \in \mathcal{I}$ as $\forall Rel \in \mathcal{REL}$, $\exists P_A \in S$ such that $P_A \to \langle I, Rel(I) \rangle$. Hence, S does not occur in I if

$$\exists Rel \in \mathcal{REL} \text{ such that } \forall P_A \in S,\ P_A \nrightarrow \langle I, Rel(I) \rangle.$$

Thus occurrence of a *set* S of augmented patterns depends on a single quantification over $\mathcal{REL}$. This is the reason why we need to consider sets of sets of augmented patterns to obtain a Galois connection.

8 A Galois Connection for Augmented Patterns

In order to establish a Galois connection between $\overline{\Sigma}_A$ and $\overline{\mathcal{T}}$, we require the following functions $F : \Sigma_A \to \mathcal{T}$ and $G : \mathcal{T} \to \Sigma_A$.

$$F(\sigma) = \{I \in \mathcal{I} \mid \forall S \in \sigma, \exists Rel \in \mathcal{REL} \text{ such that } \forall P \in S, P \nrightarrow \langle I, Rel(I) \rangle\}$$
$$G(T) = \{S \in \mathcal{S}_{\mathcal{A}} \mid \forall I \in T, \exists Rel \in \mathcal{REL} \text{ such that } \forall P \in S, P \nrightarrow \langle I, Rel(I) \rangle\}$$

To give a concrete example to illustrate the definition of F, if S contains patterns which when forbidden impose that *Rel* is a partial order on the variables, then $F(\{S\})$ only contains instances equipped with a partial order on their variables. As in the case of BTP, we may want to impose a total order on the variables. $F(\{S\})$ contains many instances which are either incompletely specified or for which *Rel* is not total; such instances can be recognised (and filtered out) in polynomial time and thus are irrelevant for deciding whether S is tractable or not, but are essential for the Galois connection. This is analogous to the Galois connection for flat pattern where $f(S)$ included incompletely-specified instances.

Given a set of instances T, there may be more than one way of describing T using forbidden augmented patterns. For example, let S_1 be the set of augmented patterns imposing a partial order on variables (as described in Sect. 6) together with the pattern BTP shown in Fig. 9(a), and let S_2 be identical to S_1 except that BTP is replaced by the pattern in Fig. 9(b). It is easy to see that $F(\{S_1\}) = F(\{S_2\})$. Hence, if $T = F(\{S_1\})$, then $S_1, S_2 \in G(T)$.

Theorem 6. *The functions F and G define an antitone Galois connection between $\overline{\Sigma}_A$ and $\overline{\mathcal{T}}$.*

Proof. To show that we have an antitone Galois connection between $\overline{\Sigma}_A$ and $\overline{\mathcal{T}}$, it suffices to show that $\forall \sigma \in \Sigma_A$, $\forall T \in \mathcal{T}$, $T \leq F(\sigma) \Leftrightarrow \sigma \leq G(T)$. This corresponds to $(T \mapsto F(\sigma)) \Leftrightarrow (G(T) \twoheadrightarrow \sigma)$.

By definition, $T \mapsto F(\sigma)$ if and only if $\forall I_T \in T$, $\exists I \in \mathcal{I}$ with $I_T \rightarrow I$ and such that $\forall S \in \sigma$, $\exists Rel \in \mathcal{REL}$ such that $\forall P \in S$, $P \nrightarrow \langle I, Rel(I)\rangle$. Thus $T \mapsto F(S)$ if and only if $\forall I_T \in T$, $\forall S \in \sigma$, $\exists Rel \in \mathcal{REL}$ such that $\forall P \in S$, $P \nrightarrow \langle I, Rel(I)\rangle$.

On the other hand, $G(T) \twoheadrightarrow \sigma$ if and only if $\forall S \in \sigma$, $\exists S' \in \mathcal{S}_{\mathcal{A}}$ with $S' \twoheadrightarrow S$ and such that $\forall I \in T$, $\exists Rel \in \mathcal{REL}$ such that $\forall P' \in S'$, $P' \nrightarrow \langle I, Rel(I)\rangle$. Thus $G(T) \twoheadrightarrow \sigma$ if and only if $\forall S \in \sigma$, $\forall I \in T$, $\exists Rel \in \mathcal{REL}$ such that $\forall P \in S$, $P \nrightarrow \langle I, Rel(I)\rangle$.

We therefore have $(T \mapsto F(\sigma)) \Leftrightarrow (G(T) \twoheadrightarrow \sigma)$ which completes the proof.

The Galois connection is similar to the Galois connection between $\overline{\mathcal{T}}$ and $\overline{\mathcal{S}}$, as demonstrated by the following results.

Theorem 7. *For all $\sigma_1, \sigma_2 \in \overline{\Sigma}_A$, $F(\sigma_1 + \sigma_2) = F(\sigma_1) \cup F(\sigma_2)$.*

Proof. $F(\sigma_1 + \sigma_2) = \{I \in \mathcal{I} \mid \forall S \in \sigma_1 + \sigma_2, \exists Rel \in \mathcal{REL} \text{ such that } \forall P \in S, P \nrightarrow \langle I, Rel(I)\rangle\} = \{I \in \mathcal{I} \mid \forall S_1 \in \sigma_1, \forall S_2 \in \sigma_2, \exists Rel \in \mathcal{REL} \text{ such that } \forall P_1 \in S_1, \forall P_2 \in S_2, P_1 + P_2 \nrightarrow \langle I, Rel(I)\rangle\}$. But $P_1 + P_2 \nrightarrow \langle I, Rel(I)\rangle$ if and only if $P_1 \nrightarrow \langle I, Rel(I)\rangle$ or $P_2 \nrightarrow \langle I, Rel(I)\rangle$ (by an immediate generalisation of Lemma 1 to augmented patterns). Furthermore, $\forall P_1 \in S_1$, $\forall P_2 \in S_2$, $P_1 \nrightarrow \langle I, Rel(I)\rangle$ or $P_2 \nrightarrow \langle I, Rel(I)\rangle$ if and only if $\forall P_1 \in S_1$, $P_1 \nrightarrow \langle I, Rel(I)\rangle$ or $\forall P_2 \in S_2$, $P_2 \nrightarrow \langle I, Rel(I)\rangle$. From all this, it follows that $F(\sigma_1 + \sigma_2) = \{I \in \mathcal{I} \mid \forall S_1 \in \sigma_1, \exists Rel \in \mathcal{REL} \text{ such that } \forall P \in S_1, P \nrightarrow \langle I, Rel(I)\rangle\} \cup \{I \in \mathcal{I} \mid \forall S_2 \in \sigma_2, \exists Rel \in \mathcal{REL} \text{ such that } \forall P \in S_2, P \nrightarrow \langle I, Rel(I)\rangle\} = F(\sigma_1) \cup F(\sigma_2)$.

Theorem 8. *For all $\sigma_1, \sigma_2 \in \overline{\Sigma}_A$, $F(\sigma_1 \cup \sigma_2) = F(\sigma_1) \cap F(\sigma_2)$.*

Proof. $F(\sigma_1 \cup \sigma_2) = \{I \in \mathcal{I} \mid \forall S \in \sigma_1 \cup \sigma_2, \exists Rel \in \mathcal{REL} \text{ such that } \forall P \in S, P \nrightarrow \langle I, Rel(I)\rangle\} = \{I \in \mathcal{I} \mid \forall S \in \sigma_1, \exists Rel \in \mathcal{REL} \text{ such that } \forall P \in S, P \nrightarrow \langle I, Rel(I)\rangle\} \cap \{I \in \mathcal{I} \mid \forall S \in \sigma_2, \exists Rel \in \mathcal{REL} \text{ such that } \forall P \in S, P \nrightarrow \langle I, Rel(I)\rangle\} = F(\sigma_1) \cap F(\sigma_2)$.

The lattice structure of Σ_A and Theorems 7 and 8 are illustrated in Fig. 10.

Theorem 9. *For all $T_1, T_2 \in \overline{\mathcal{T}}$, $G(T_1 \cup T_2) = G(T_1) \cap G(T_2)$.*

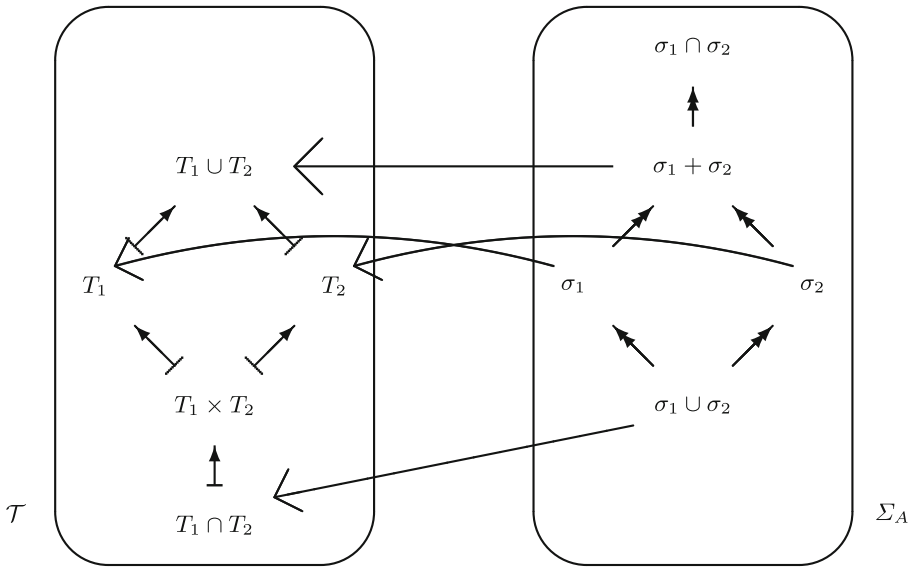

Fig. 10. The function F from Σ_A to $\mathcal{T}$

Proof. $G(T_1 \cup T_2) = \{S \in \mathcal{S}_\mathcal{A} \mid \forall I \in T_1 \cup T_2, \exists Rel \in \mathcal{REL}$ such that $\forall P \in S$, $P \nrightarrow \langle I, Rel(I)\rangle\} = \{S \in \mathcal{S}_\mathcal{A} \mid \forall I \in T_1, \exists Rel \in \mathcal{REL}$ such that $\forall P \in S$, $P \nrightarrow \langle I, Rel(I)\rangle\} \cap \{S \in \mathcal{S}_\mathcal{A} \mid \forall I \in T_2, \exists Rel \in \mathcal{REL}$ such that $\forall P \in S$, $P \nrightarrow \langle I, Rel(I)\rangle\} = G(T_1) \cap G(T_2)$.

Theorem 10. *For all* $T_1, T_2 \in \overline{\mathcal{T}}$, $G(T_1 \times T_2) = G(T_1) \cup G(T_2)$.

Proof. $G(T_1 \times T_2) = \{S \in \mathcal{S}_\mathcal{A} \mid \forall I \in T_1 \times T_2, \exists Rel \in \mathcal{REL}$ such that $\forall P \in S$, $P \nrightarrow \langle I, Rel(I)\rangle\} = \{S \in \mathcal{S}_\mathcal{A} \mid \forall I_1 \in T_1, \forall I_2 \in T_2, \exists Rel \in \mathcal{REL}$ such that $\forall P \in S$, $P \nrightarrow \langle I_1 \times I_2, Rel(I_1 \times I_2)\rangle\}$. Now, for any $Rel \in \mathcal{REL}$, $\langle I_1, Rel(I_1)\rangle \times \langle I_2, Rel(I_2)\rangle \rightarrow \langle I_1 \times I_2, Rel(I_1 \times I_2)\rangle$. Thus $P \nrightarrow \langle I_1 \times I_2, Rel(I_1 \times I_2)\rangle$ implies $P \nrightarrow \langle I_1, Rel(I_1)\rangle \times \langle I_2, Rel(I_2)\rangle$ which (by an immediate extension of Lemma 2 to augmented patterns) is equivalent to $(P \nrightarrow \langle I_1, Rel(I_1)\rangle) \vee (P \nrightarrow \langle I_2, Rel(I_2)\rangle)$. It follows from the above that $G(T_1 \times T_2) \subseteq \{S \in \mathcal{S}_\mathcal{A} \mid \forall I_1 \in T_1, \forall I_2 \in T_2, \exists Rel \in \mathcal{REL}$ such that $(P \nrightarrow \langle I_1, Rel(I_1)\rangle) \vee (P \nrightarrow \langle I_2, Rel(I_2)\rangle)\}$. But, the latter is equal to $\{S \in \mathcal{S}_\mathcal{A} \mid (\forall I_1 \in T_1, \exists Rel \in \mathcal{REL}$ such that $(P \nrightarrow \langle I_1, Rel(I_1)\rangle)) \vee (\forall I_1 \in T_2, \exists Rel \in \mathcal{REL}$ such that $P \nrightarrow \langle I_2, Rel(I_2)\rangle)\} = G(T_1) \cup G(T_2)$. Thus $G(T_1 \times T_2) \subseteq G(T_1) \cup G(T_2)$.

In order to show $G(T_1) \cup G(T_2) \subseteq G(T_1 \times T_2)$, and hence to complete the proof, without loss of generality, we only need to show $G(T_1) \subseteq G(T_1 \times T_2)$. Consider $S \in G(T_1)$. We have $\forall I_1 \in T_1, \exists Rel_1 \in \mathcal{REL}$ such that $\forall P \in S$, $P \nrightarrow \langle I_1, Rel_1(I_1)\rangle$. Therefore, for all common factors I of I_1 and I_2, $\exists Rel \in \mathcal{REL}$ such that $\forall P \in S$, $P \nrightarrow \langle I, Rel(I)\rangle$. Indeed, we can clearly choose $Rel = Rel_1$ for each such I. Now $I_1 \times I_2$ is the juxtaposition of copies of such common factors I. These copies are comprised of disjoint sets of points. For each such copy of a

common factor I composing $I_1 \times I_2$, there is a corresponding version of $Rel_1(I)$ which we denote by $Rel_I(I)$. The relations $Rel_I(I)$ are disjoint (since within $I_1 \times I_2$ each common factor I is comprised of disjoint sets of points). Let R be the relation which is the union of all these $Rel_I(I)$. Then $\exists Rel \in \mathcal{REL}$ such that $Rel(I_1 \times I_2) = R$. Now $\forall P \in S$, $P \twoheadrightarrow \langle I_1 \times I_2, Rel(I_1 \times I_2)\rangle$. Therefore $S \in G(T_1 \times T_2)$ which completes the proof.

Theorems 9 and 10 are illustrated in Fig. 11.

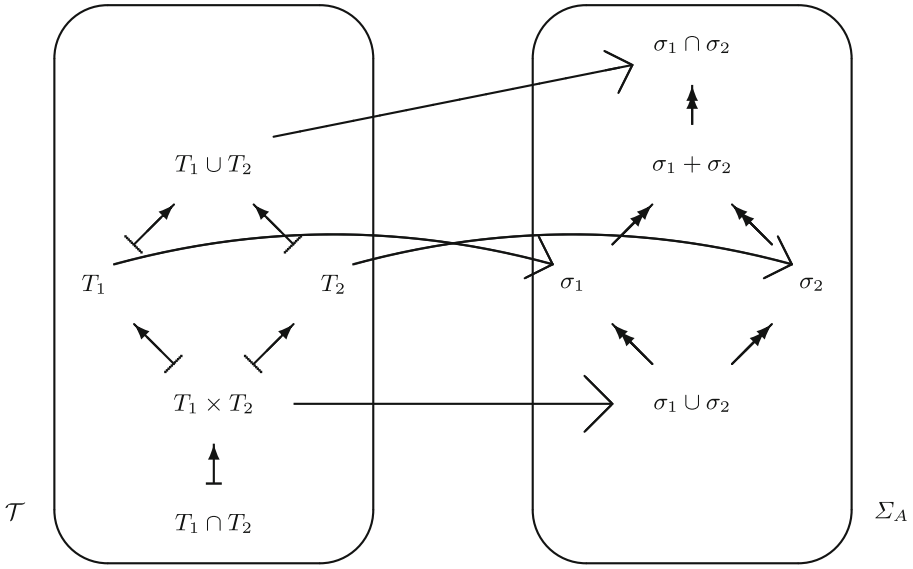

Fig. 11. The function G from $\mathcal{T}$ to Σ_A

In order to define tractability of sets of augmented patterns we must apply a filter to instances so that we only consider completely-specified instances with a certain property. Examples of filters include the property that an ordering relation is total or that two relations (such as the relations IE and $\overline{IE}$ that we introduced in Sect. 6) form a cover of all pairs of assignments to distinct variables. For example, in the case of BTP, we are only interested in instances equipped with a total ordering on the variables, since the pattern shown in Fig. 9(a) trivially does not occur on variables which are not ordered. This leads to the following definition of tractability.

Definition 3. *Let $\mathcal{F}$ be a property of instances $I \in \mathcal{I}$ that can be verified in polynomial time. We say that $\sigma \in \Sigma_A$ is* tractable *(with respect to the filter $\mathcal{F}$) if there is a polynomial-time algorithm to decide the set of completely-specified instances in $F(\sigma)$ (which satisfy the filter $\mathcal{F}$). In particular, we say that $S \in \mathcal{S}_A$ is tractable (w.r.t. $\mathcal{F}$) if $\{S\}$ is tractable (w.r.t. $\mathcal{F}$).*

Proposition 9. *The tractable elements of $\overline{\Sigma}_A$ form a sublattice. Furthermore, the tractable sets of augmented patterns form a join semi-lattice of $\mathcal{S}_\mathcal{A}$.*

Proof. If $\sigma_1, \sigma_2 \in \overline{\Sigma}_A$ are tractable, then so are $\sigma_1 + \sigma_2$ and $\sigma_1 \cup \sigma_2$. This follows immediately from the fact that $F(\sigma_1 + \sigma_2) = F(\sigma_2) \cup F(\sigma_2)$ and $F(\sigma_1 \cup \sigma_2) = F(\sigma_2) \cap F(\sigma_2)$. The tractable sets of augmented patterns form a join semi-lattice, since S_1, S_2 tractable implies that $S_1 + S_2$ is tractable.

9 Discussion and Conclusion

In this paper we have initiated the study of the Galois connection between lattices of sets of forbidden patterns and sets of instances. The consequences of this Galois connection for expressibility and tractability questions remains largely unexplored. However, we have shown that the tractable sets of patterns form a sub-lattice.

Augmented patterns provide a rich language in which we can define many interesting classes of instances in a concise form, notably by adding an order on the variables or the values. We have seen that both bounded treewidth and the existence of a polymorphism satisfying a set of identities can be expressed using augmented patterns (together with a filter on the set of instances). This leads to an orthogonal question of the tractability of the recognition of classes defined by augmented patterns. For example, given a binary CSP instance, it is NP-hard to determine whether there exists an ordering of the values under which all relations are max-closed [25]. On the other hand, it is tractable to decide whether the relations have a conservative Mal'tsev polymorphism [6]. Determining the tractability frontier of this meta-problem is an open question for augmented patterns. As we have pointed out, the recognition problem is always tractable for finite sets of flat patterns.

It is natural to ask whether the Feder-Vardi dichotomy [23] (for classes of CSP instances defined by finite languages of constraint relations) generalises to classes of CSP instances defined by augmented patterns. However, we know that no such P/NP-hard dichotomy can exist by the work on lifted patterns by Kun and Nešetřil [33] and by Ladner's theorem [34]. It is an open question whether classes of CSP instances defined by forbidding *flat* patterns exhibit a dichotomy in the following sense: all finite sets of patterns are either tractable or NP-complete. We conjecture that this is true.

References

1. Barto, L., Krokhin, A.A., Willard, R.: Polymorphisms, and how to use them. In: Krokhin, A., Živný, S. (eds.) [32], pp. 1–44. https://doi.org/10.4230/DFU.Vol7.15301.1
2. Birkhoff, G.: Lattice Theory, vol. 25, 3rd edn. AMS Colloquium Publications, American Mathematical Society, New York (1967)

3. Bulatov, A.A.: A dichotomy theorem for nonuniform CSPs. In: Umans, C. (ed.) 58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, 15–17 October 2017, pp. 319–330. IEEE Computer Society (2017). https://doi.org/10.1109/FOCS.2017.37
4. Bulatov, A.A., Jeavons, P., Krokhin, A.A.: Classifying the complexity of constraints using finite algebras. SIAM J. Comput. **34**(3), 720–742 (2005). https://doi.org/10.1137/S0097539700376676
5. Bulin, J., Delic, D., Jackson, M., Niven, T.: A finer reduction of constraint problems to digraphs. Log. Methods Comput. Sci. **11**(4), 1–33 (2015). https://doi.org/10.2168/LMCS-11(4:18)2015
6. Carbonnel, C.: The dichotomy for conservative constraint satisfaction is polynomially decidable. In: Rueher, M. (ed.) CP 2016. LNCS, vol. 9892, pp. 130–146. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-44953-1_9
7. Carbonnel, C., Cohen, D.A., Cooper, M.C., Živný, S.: On singleton arc consistency for CSPs defined by monotone patterns. Algorithmica **81**(4), 1699–1727 (2019). https://doi.org/10.1007/s00453-018-0498-2
8. Carbonnel, C., Cooper, M.C.: Tractability in constraint satisfaction problems: a survey. Constraints **21**(2), 115–144 (2016). https://doi.org/10.1007/s10601-015-9198-6
9. Cohen, D.A., Cooper, M.C., Creed, P., Marx, D., Salamon, A.Z.: The tractability of CSP classes defined by forbidden patterns. J. Artif. Intell. Res. (JAIR) **45**, 47–78 (2012). https://doi.org/10.1613/jair.3651
10. Cohen, D.A., Cooper, M.C., Jeavons, P.G., Krokhin, A.A., Powell, R., Živný, S.: Binarisation for valued constraint satisfaction problems. SIAM J. Discret. Math. **31**(4), 2279–2300 (2017). https://doi.org/10.1137/16M1088107
11. Cohen, D.A., Cooper, M.C., Jeavons, P.G., Živný, S.: Binary constraint satisfaction problems defined by excluded topological minors. Inf. Comput. **264**, 12–31 (2019). https://doi.org/10.1016/j.ic.2018.09.013
12. Cooper, M.C.: Beyond consistency and substitutability. In: O'Sullivan, B. (ed.) CP 2014. LNCS, vol. 8656, pp. 256–271. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10428-7_20
13. Cooper, M.C., Cohen, D.A., Jeavons, P.: Characterising tractable constraints. Artif. Intell. **65**(2), 347–361 (1994). https://doi.org/10.1016/0004-3702(94)90021-3
14. Cooper, M.C., Duchein, A., Mouelhi, A.E., Escamocher, G., Terrioux, C., Zanuttini, B.: Broken triangles: from value merging to a tractable class of general-arity constraint satisfaction problems. Artif. Intell. **234**, 196–218 (2016). https://doi.org/10.1016/j.artint.2016.02.001
15. Cooper, M.C., Escamocher, G.: Characterising the complexity of constraint satisfaction problems defined by 2-constraint forbidden patterns. Discrete Appl. Math. **184**, 89–113 (2015). https://doi.org/10.1016/j.dam.2014.10.035
16. Cooper, M.C., Jeavons, P.G., Salamon, A.Z.: Generalizing constraint satisfaction on trees: hybrid tractability and variable elimination. Artif. Intell. **174**(9–10), 570–584 (2010). https://doi.org/10.1016/j.artint.2010.03.002
17. Cooper, M.C., Živný, S.: Hybrid tractability of valued constraint problems. Artif. Intell. **175**(9–10), 1555–1569 (2011). https://doi.org/10.1016/j.artint.2011.02.003
18. Cooper, M.C., Živný, S.: Hybrid tractable classes of constraint problems. In: Krokhin, A., Živný, S. (eds.) [32], pp. 113–135. https://doi.org/10.4230/DFU.Vol7.15301.4
19. Cooper, M.C., Živný, S.: The power of arc consistency for CSPs defined by partially-ordered forbidden patterns. Log. Methods Comput. Sci. **13**(4) (2017). https://doi.org/10.23638/LMCS-13(4:26)2017

20. Courcelle, B., Mosbah, M.: Monadic second-order evaluations on tree-decomposable graphs. Theor. Comput. Sci. **109**(1&2), 49–82 (1993). https://doi.org/10.1016/0304-3975(93)90064-Z
21. Davey, B.A., Priestley, H.A.: Introduction to Lattices and Order, 2nd edn. Cambridge University Press, Cambridge (2002)
22. Dechter, R., Pearl, J.: Tree clustering for constraint networks. Artif. Intell. **38**(3), 353–366 (1989). https://doi.org/10.1016/0004-3702(89)90037-4
23. Feder, T., Vardi, M.Y.: The computational structure of monotone monadic SNP and constraint satisfaction: a study through datalog and group theory. SIAM J. Comput. **28**(1), 57–104 (1998). https://doi.org/10.1137/S0097539794266766
24. Freuder, E.C.: A sufficient condition for backtrack-bounded search. J. ACM **32**(4), 755–761 (1985). https://doi.org/10.1145/4221.4225
25. Green, M.J., Cohen, D.A.: Domain permutation reduction for constraint satisfaction problems. Artif. Intell. **172**(8–9), 1094–1118 (2008). https://doi.org/10.1016/j.artint.2007.12.001
26. Grohe, M.: The complexity of homomorphism and constraint satisfaction problems seen from the other side. J. ACM **54**(1), 1:1–1:24 (2007). https://doi.org/10.1145/1206035.1206036
27. Jeavons, P.: On the algebraic structure of combinatorial problems. Theor. Comput. Sci. **200**(1–2), 185–204 (1998). https://doi.org/10.1016/S0304-3975(97)00230-2
28. Jeavons, P., Cohen, D.A., Gyssens, M.: Closure properties of constraints. J. ACM **44**(4), 527–548 (1997). https://doi.org/10.1145/263867.263489
29. Jeavons, P., Cooper, M.C.: Tractable constraints on ordered domains. Artif. Intell. **79**(2), 327–339 (1995). https://doi.org/10.1016/0004-3702(95)00107-7
30. Jégou, P.: Decomposition of domains based on the micro-structure of finite constraint-satisfaction problems. In: Fikes, R., Lehnert, W.G. (eds.) Proceedings of the 11th National Conference on Artificial Intelligence, Washington, DC, USA, pp. 731–736. AAAI Press/The MIT Press (1993). http://www.aaai.org/Library/AAAI/1993/aaai93-109.php
31. Knuth, D.E.: Stable Marriage and Its Relation to Other Combinatorial Problems, CRM Proceedings & Lecture Notes, vol. 10. American Mathematical Society, Providence (1996)
32. Krokhin, A.A., Živný, S. (eds.): The Constraint Satisfaction Problem: Complexity and Approximability, Dagstuhl Follow-Ups, vol. 7. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2017). http://www.dagstuhl.de/dagpub/978-3-95977-003-3
33. Kun, G., Nešetřil, J.: Forbidden lifts (NP and CSP for combinatorialists). Eur. J. Comb. **29**(4), 930–945 (2008). https://doi.org/10.1016/j.ejc.2007.11.027
34. Ladner, R.E.: On the structure of polynomial time reducibility. J. ACM **22**(1), 155–171 (1975). https://doi.org/10.1145/321864.321877
35. Mouelhi, A.E.: Tractable classes for CSPs of arbitrary arity: from theory to practice. Constraints **22**(1), 97–98 (2017). https://doi.org/10.1007/s10601-016-9262-x
36. Naanaa, W.: Unifying and extending hybrid tractable classes of CSPs. J. Exp. Theor. Artif. Intell. **25**(4), 407–424 (2013). https://doi.org/10.1080/0952813X.2012.721138
37. Naanaa, W.: Extending the broken triangle property tractable class of binary CSPs. In: Bassiliades, N., Bikakis, A., Vrakas, D., Vlahavas, I.P., Vouros, G.A. (eds.) Proceedings of the 9th Hellenic Conference on Artificial Intelligence, SETN 2016, Thessaloniki, Greece, pp. 3:1–3:6. ACM (2016). https://doi.org/10.1145/2903220.2903230

38. Régin, J.: A filtering algorithm for constraints of difference in CSPs. In: Hayes-Roth, B., Korf, R.E. (eds.) Proceedings of the 12th National Conference on Artificial Intelligence, vol. 1, pp. 362–367. AAAI Press/The MIT Press (1994). http://www.aaai.org/Library/AAAI/1994/aaai94-055.php
39. Siggers, M.H.: A strong Mal'cev condition for locally finite varieties omitting the unary type. Algebra Univers. **64**(1–2), 15–20 (2010)
40. Stergiou, K., Samaras, N.: Binary encodings of non-binary constraint satisfaction problems: algorithms and experimental results. J. Artif. Intell. Res. (JAIR) **24**, 641–684 (2005). https://doi.org/10.1613/jair.1776
41. Zhuk, D.: A proof of the CSP dichotomy conjecture. J. ACM **67**(5), 30:1–30:78 (2020). https://doi.org/10.1145/3402029