The complexity of conservative valued CSPs

VLADIMIR KOLMOGOROV, Institute of Science and Technology (IST), Austria
STANISLAV ŽIVNÝ, University of Warwick, United Kingdom

We study the complexity of valued constraint satisfaction problems (VCSPs) parametrised by a *constraint language*, a fixed set of cost functions over a finite domain. An instance of the problem is specified by a sum of cost functions from the language and the goal is to minimise the sum. Under the unique games conjecture, the approximability of finite-valued VCSPs is well-understood, see Raghavendra [STOC'08]. However, there is no characterisation of finite-valued VCSPs, let alone general-valued VCSPs, that can be solved exactly in polynomial time, thus giving insights from a combinatorial optimisation perspective.

We consider the case of languages containing all possible unary cost functions. In the case of languages consisting of only $\{0, \infty\}$-valued cost functions (i.e. relations), such languages have been called *conservative* and studied by Bulatov [LICS'03, ACM TOCL'11] and recently by Barto [LICS'11]. Since we study valued languages, we call a language *conservative* if it contains all finite-valued unary cost functions. The computational complexity of conservative valued languages has been studied by Cohen *et al.* [AIJ'06] for languages over Boolean domains, by Deineko *et al.* [JACM'08] for $\{0, 1\}$-valued languages (a.k.a Max-CSP), and by Takhanov [STACS'10] for $\{0, \infty\}$-valued languages containing all finite-valued unary cost functions (a.k.a. Min-Cost-Hom).

We prove a Schaefer-like dichotomy theorem for conservative valued languages: if all cost functions in the language satisfy a certain condition (specified by a complementary combination of *STP and MJN multimorphisms*), then any instance can be solved in polynomial time (via a new algorithm developed in this paper), otherwise the language is NP-hard. This is the *first* complete complexity classification of *general-valued constraint languages* over non-Boolean domains. It is a common phenomenon that complexity classifications of problems over non-Boolean domains are significantly harder than the Boolean cases. The polynomial-time algorithm we present for the tractable cases is a generalisation of the submodular minimisation problem and a result of Cohen *et al.* [TCS'08].

Our results generalise previous results by Takhanov [STACS'10] and (a subset of results) by Cohen *et al.* [AIJ'06] and Deineko *et al.* [JACM'08]. Moreover, our results do not rely on any computer-assisted search as in Deineko *et al.* [JACM'08], and provide a powerful tool for proving hardness of finite-valued and general-valued languages.

Categories and Subject Descriptors: F.2.0 [**Analysis of Algorithms and Problem Complexity**]: General

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Complexity, dichotomy, valued constraint satisfaction problems, multimorphisms, discrete optimisation, submodularity

---

An extended abstract of this work appeared in the *Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2012.
Vladimir Kolmogorov was supported by the Royal Academy of Engineering/EPSRC, UK. Stanislav Živný was supported by a Junior Research Fellowship at Oxford's University College. Part of this work was done while the second author was visiting Microsoft Research Cambridge.
Author's addresses: V. Kolmogorov, Institute of Science and Technology Austria (IST Austria), Am Campus 1, 3400 Klosterneuburg, Austria; S. Živný, Department of Computer Science, University of Warwick, Coventry CV4 7AL, United Kingdom.

DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

ACM Reference Format:
Kolmogovor, V. and Živný, S. 2011. The complexity of conservative valued CSPs. J. ACM 1, 1, Article 1 (October 2011), 39 pages.
DOI = 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

1. INTRODUCTION

The constraint satisfaction problem is a central generic problem in computer science. It provides a common framework for many theoretical problems as well as for many real-life applications, see [Hell and Nešetřil 2008] for a nice survey. An instance of the *constraint satisfaction problem* (CSP) consists of a collection of variables which must be assigned values subject to specified constraints [Montanari 1974]. CSP is equivalent to the problem of evaluating conjunctive queries on databases [Kolaitis and Vardi 2000], and to the homomorphism problem for relational structures [Feder and Vardi 1998].

An important line of research on the CSP is to identify all tractable cases; that is, cases that are recognisable and solvable in polynomial time. Most of this work has been focused on one of the two general approaches: either identifying structural properties of the way constraints interact which ensure tractability no matter what forms of constraints are imposed [Dechter and Pearl 1988], or else identifying forms of constraints which are sufficiently restrictive to ensure tractability no matter how they are combined [Bulatov et al. 2005; Feder and Vardi 1998].

The first approach has been used to characterise all tractable cases of bounded-arity CSPs: the *only* class of structures which ensures tractability (subject to a certain complexity theory assumption, namely FPT $\neq$ W[1]) are structures of bounded tree-width modulo homomorphic equivalence [Dalmau et al. 2002; Grohe 2007]; and recently also for unbounded-arity CSPs [Marx 2010b]; see also [Grohe and Marx 2006; Marx 2010a]. The second approach has led to identifying certain algebraic properties known as polymorphisms [Jeavons 1998] which are necessary for a set of constraint types to ensure tractability. A set of constraint types which ensures tractability is called a *tractable constraint language*.

Schaefer in his seminal work [Schaefer 1978] gave a complete complexity classification of Boolean constraint languages. The algebraic approach based on polymorphisms [Jeavons et al. 1997] has been so far the most successful tool in generalising Schaefer's result to languages over a 3-element domain [Bulatov 2006], languages with all unary relations [Bulatov 2003; 2011; Barto 2011], languages comprised of a single binary relation without sources and sinks [Barto et al. 2009b] (see also [Barto and Kozik 2010]), and languages comprised of a single binary relation that is a special triad [Barto et al. 2009a]. The algebraic approach has also been essential in characterising the power of local consistency [Barto and Kozik 2009] and the "few subpowers property" [Berman et al. 2010; Idziak et al. 2010], the two main tools known for solving tractable CSPs. A major open question in this line of research is the *Dichotomy Conjecture* of Feder and Vardi, which states that every constraint language is either tractable or NP-hard [Feder and Vardi 1998]. We remark that there are other approaches to the dichotomy conjecture; see, for instance, [Hell and Nešetřil 2008] for a nice survey, and [Kun and Szegedy 2009] for a connection between the Dichotomy Conjecture and probabilistically checkable proofs.

Since in practice many constraint satisfaction problems are over-constrained, and hence have no solution, or are under-constrained, and hence have many solutions, *soft* constraint satisfaction problems have been studied [Dechter 2003]. In an instance of the soft CSP, every constraint is associated with a cost function (rather than a relation as in the CSP) which represents preferences among different partial assignments, and the goal is to find the best assignment. Several very general soft CSP frameworks have been proposed in the literature [Schiex et al. 1995; Bistarelli et al. 1997]. In this paper

we focus on one of the very general frameworks, the *valued* constraint satisfaction problem (VCSP) [Schiex et al. 1995]. Throughout the paper, we use the term *constraint language* (or just *language*) for a set of cost functions over a finite domain. If all cost functions from a given language Γ are $\{0, \infty\}$-valued (i.e. relations), we call Γ a *crisp* language. (If necessary, to stress the fact that Γ is a language, but not a crisp language, we call Γ a *general-valued* language.)

Similarly to the CSP, an important line of research on the VCSP is to identify tractable cases which are recognisable in polynomial time. Is is well known that structural reasons for tractability generalise to the VCSP [Bertelé and Brioshi 1972; Gottlob et al. 2009]. In the case of language restrictions, only a few conditions are known to guarantee tractability of a given language [Cohen et al. 2006; Cohen et al. 2008]. Recently, the power of linear programming relaxations for VCSPs has been characterised [Thapper and Živný 2012a; Kolmogorov 2012] and also the complexity of all finite-valued languages has been established [Thapper and Živný 2013]. Apart from structural and language restrictions on VCSPs, hybrid restrictions have also recently been studied [Cooper and Živný 2011; 2012].

Related work. The problem of characterising the complexity of different languages has received significant attention in the literature. For some classes researchers have established a Schaefer-like dichotomy theorem of the following form: if a language Γ admits certain *polymorphisms* or *multimorphisms* then it is tractable, otherwise it is NP-hard. Some of these classes are as follows: Boolean languages, i.e. languages with a 2-element domain [Cohen et al. 2006]; crisp languages including all unary relations - [Bulatov 2003; 2011] and recently [Barto 2011]; crisp languages with a 3-element domain [Bulatov 2006]; $\{0, 1\}$-valued languages including all unary cost functions [Deineko et al. 2008]; crisp languages including additionally all finite-valued unary cost functions [Takhanov 2010a]; crisp languages including additionally a certain subset of finite-valued unary cost functions [Takhanov 2010b].

Our proof exploits the results of Takhanov [Takhanov 2010a], who showed the existence of a majority polymorphism as a necessary condition for tractability of crisp languages including additionally all finite-valued unary cost functions. Other related work includes the work of Creignou *et al.* who studied various generalisations of the CSP to optimisation problems over Boolean domains [Creignou 1995], see also [Creignou et al. 2001; Khanna et al. 2001]. [Raghavendra 2008] and [Raghavendra and Steurer 2009] have shown how to optimally approximate any finite-valued VCSP.

Contributions. This paper focuses on valued languages containing all finite-valued unary cost functions; we call such languages *conservative*. Our main result is a dichotomy theorem for all conservative languages: if a conservative language Γ admits a complementary combination of *STP (symmetric tournament pair) and MJN (majority-majority-minority) multimorphisms*, then it is tractable, otherwise Γ is NP-hard. This is the first complete complexity classification of general-valued languages over non-Boolean domains, generalising previously obtained results in [Cohen et al. 2006; Deineko et al. 2008; Takhanov 2010a] as follows:

— Cohen *et al.* proved a dichotomy for arbitrary Boolean languages ($|D| = 2$). We generalise it to arbitrary domains ($|D| \geq 2$), although only for conservative languages.
— Deineko *et al.* [Deineko et al. 2008] and Takhanov [Takhanov 2010a] proved a dichotomy for the following languages, respectively:
  – $\{0, 1\}$-valued languages containing additionally all unary cost functions;
  – $\{0, \infty\}$-valued languages containing additionally all unary cost functions.

  In both of these case the languages are conservative, so these classifications are special cases of our result. Note, however, that Deineko *et al.* additionally give a di-

chotomy with respect to approximability (PO vs. APX-hard), even when the number of occurrences of variables in instances is bounded; this part of [Deineko et al. 2008] does not follow from our classification.

Moreover, our results provide a new powerful tool and do not rely on a computer-assisted search as in [Deineko et al. 2008]. Building on techniques from this paper, Jonsson *et al.* [Jonsson et al. 2011] have recently shown that the same approach can be also used for certain non-conservative languages, and Chen *et al.* [Chen et al. 2012] have recently shown that the same approach can be also used for approximate counting.

Since the complexity of Boolean conservative languages is known, we start, similarly to Bulatov and Takhanov [Bulatov 2003; Takhanov 2010a], by exploring the interactions between different 2-element subdomains. Given a conservative language Γ, we will investigate properties of a certain graph G_Γ associated with the language and cost functions expressible over Γ. We link the complexity of Γ to certain properties of the graph G_Γ.

First, we show that if G_Γ does not satisfy certain properties, then Γ is intractable. Second, using G_Γ, we construct a (partial) *STP multimorphism* and a (partial) *MJN multimorphism*. Finally, we show that any language which admits a complementary combination of *STP and MJN multimorphisms* is tractable, thus generalising a tractable class of Cohen *et al.* [Cohen et al. 2008], which in turn is a generalisation of the submodular minimisation problem. Thus we obtain a dichotomy theorem. The tractable criterion in the finite-valued case turns out to be equivalent to the condition of submodularity. The general-valued case is much more involved than the finite-valued case, and requires different techniques compared to previous results.

Given a finite language Γ, the graph G_Γ is finite as well, but depends on the expressive power of Γ (see Section 2 for precise definitions), which is infinite. In order to test whether Γ is tractable, we do not need to construct the graph G_Γ as it follows from our result that we just need to test for the existence of a complementary combination of two multimorphisms, which can be established in polynomial time.

Our results are formulated using the terminology of valued constraint satisfaction problems, but they apply to various other optimisation frameworks that are equivalent to valued constraint satisfaction problems such as Gibbs energy minimisation, Markov Random Fields, Min-Sum problems, and other models [Lauritzen 1996; Wainwright and Jordan 2008].

Organisation of the paper. The rest of the paper is organised as follows. In Section 2, we define valued constraint satisfaction problems (VCSPs), conservative languages, multimorphisms and other necessary definitions needed throughout the paper. We state our results in Section 3, and then give their proofs in Sections 4-7.

2. BACKGROUND AND NOTATION

We denote by $\mathbb{Q}_+$ the set of all non-negative rational numbers. We define $\overline{\mathbb{Q}}_+ = \mathbb{Q}_+ \cup \{\infty\}$ with the standard addition operation extended so that for all $a \in \mathbb{Q}_+$, $a + \infty = \infty$. Members of $\overline{\mathbb{Q}}_+$ are called *costs*. Throughout the paper, we denote by D any fixed finite set, called a *domain*. Elements of D are called *domain values* or *labels*.

A function f from D^m to $\overline{\mathbb{Q}}_+$ will be called a *cost function* on D of *arity* m. If the range of f lies entirely within $\mathbb{Q}_+$, then f is called a *finite-valued* cost function. If the range of f is $\{0, \infty\}$, then f is called a *crisp* cost function. If the range of a cost function f includes non-zero finite costs and infinity, we emphasise this fact by calling f a *general-valued* cost function. Let $f : D^m \to \overline{\mathbb{Q}}_+$ be an m-ary cost function f. We

denote by $\mathtt{dom} f = \{\boldsymbol{x} \in D^m \mid f(\boldsymbol{x}) < \infty\}$ the effective domain of f. The argument of f is called an *assignment* or a *labelling*. Functions f of arity $m = 2$ are called *binary*.

A *language* is a set of cost functions with the same domain D. A language Γ is called finite-valued (crisp, general-valued respectively) if all cost functions in Γ are finite-valued (crisp, general-valued respectively). A language Γ is *Boolean* if $|D| = 2$.

Definition 2.1. An instance $\mathcal{I}$ of the *valued constraint satisfaction problem* (VCSP) is a function $D^V \to \overline{\mathbb{Q}}_+$ given by

$$\mathrm{Cost}_{\mathcal{I}}(\boldsymbol{x}) = \sum_{t \in T} f_t\left(x_{i(t,1)}, \ldots, x_{i(t,m_t)}\right)$$

It is specified by a finite set of nodes V, a finite set of terms (also known as constraints) T, cost functions $f_t : D^{m_t} \to \overline{\mathbb{Q}}_+$ or arity m_t and indices $i(t,k) \in V$ for $t \in T$, $k = 1, \ldots, m_t$. A *solution* to $\mathcal{I}$ is an assignment $\boldsymbol{x} \in D^V$ with minimum cost.

We denote by $\mathsf{VCSP}(\Gamma)$ the class of all VCSP instances whose terms f_t belong to Γ. A finite language Γ is called *tractable* if $\mathsf{VCSP}(\Gamma)$ can be solved in polynomial time, and *intractable* if $\mathsf{VCSP}(\Gamma)$ is NP-hard. An infinite language Γ is tractable if every finite subset $\Gamma' \subseteq \Gamma$ is tractable, and intractable if there is a finite subset $\Gamma' \subseteq \Gamma$ that is intractable.

The idea behind conservative languages is to contain all possible unary cost functions: Bulatov has called a crisp language Γ conservative if Γ contains all unary relations [Bulatov 2003]. We are interested in valued languages containing all possible unary cost functions and hence define conservative languages as follows:

Definition 2.2. A language Γ is called *conservative* if Γ contains all $\{0,1\}$-valued unary cost functions $u : D \to \{0,1\}$.

Such languages have been studied by Deineko *et al.* [Deineko et al. 2008] and Takhanov [Takhanov 2010a]. Note, we could have defined Γ to be conservative if it contains all possible general-valued unary cost functions $u : D \to \overline{\mathbb{Q}}_+$. However, the weaker definition 2.2 will be sufficient for our purposes: it is shown in Section 4 that adding all possible unary cost functions $u : D \to \overline{\mathbb{Q}}_+$ to a conservative language Γ does not change the complexity of Γ.

We now define polymorphisms, which have played a crucial role in the complexity analysis of crisp languages [Jeavons et al. 1997; Bulatov et al. 2005].

Definition 2.3. A mapping $F : D^k \to D$, $k \geq 1$ is called a *polymorphism* of a cost function $f : D^m \to \overline{\mathbb{Q}}_+$ if

$$F(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k) \in \mathtt{dom} f \qquad \forall \boldsymbol{x}_1, \ldots, \boldsymbol{x}_k \in \mathtt{dom} f$$

where F is applied component-wise. F is a polymorphism of a language Γ if F is a polymorphism of every cost function in Γ.

Multimorphisms [Cohen et al. 2006] are generalisations of polymorphisms. To make the paper easier to read, we only define binary and ternary multimorphisms as we will not need multimorphisms of higher arities.

Definition 2.4. Let $\langle \sqcap, \sqcup \rangle$ be a pair of operations, where $\sqcap, \sqcup : D \times D \to D$, and let $\langle F_1, F_2, F_3 \rangle$ be a triple of operations, where $F_i : D \times D \times D \to D$, $1 \leq i \leq 3$.

— The pair $\langle \sqcap, \sqcup \rangle$ is called a (binary) *multimorphism* of a cost function $f : D^m \to \overline{\mathbb{Q}}_+$ if

$$f(\boldsymbol{x} \sqcap \boldsymbol{y}) + f(\boldsymbol{x} \sqcup \boldsymbol{y}) \leq f(\boldsymbol{x}) + f(\boldsymbol{y}) \qquad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathtt{dom} f \tag{1}$$

where operations $\sqcap, \sqcup$ are applied component-wise. $\langle \sqcap, \sqcup \rangle$ is a multimorphism of a language Γ if $\langle \sqcap, \sqcup \rangle$ is a multimorphism of every f from Γ.

— The triple $\langle F_1, F_2, F_3 \rangle$ is called a (ternary) *multimorphism* of a cost function $f : D^m \to \overline{\mathbb{Q}}_+$ if

$$f(F_1(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}))+f(F_2(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}))+f(F_3(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})) \leq f(\boldsymbol{x})+f(\boldsymbol{y})+f(\boldsymbol{z}) \quad \forall \boldsymbol{x},\boldsymbol{y},\boldsymbol{z} \in \mathtt{dom} f \quad (2)$$

where operations F_1, F_2, F_3 are applied component-wise. $\langle F_1, F_2, F_3 \rangle$ is a multimorphism of a language Γ if $\langle F_1, F_2, F_3 \rangle$ is a multimorphism of every f from Γ.

— Operation $F : D^k \to D$ is called *conservative* if $F(x_1, \ldots, k_k) \in \{x_1, \ldots, x_k\}$ for all $x_1, \ldots, x_k \in D$.

— The pair $\langle \sqcap, \sqcup \rangle$ is called *conservative* if $\{\{a \sqcap b, a \sqcup b\}\} = \{\{a, b\}\}$ for all $a, b \in D$, where $\{\{\ldots\}\}$ denotes a *multiset*, i.e. in the case of repetitions elements' multiplicities are taken into account. Similarly, triple $\langle F_1, F_2, F_3 \rangle$ is called *conservative* if $\{\{F_1(a,b,c), F_2(a,b,c), F_3(a,b,c)\}\} = \{\{a,b,c\}\}$ for all $a, b, c \in D$. In other words, applying $\langle F_1, F_2, F_3 \rangle$ to (a, b, c) should give a permutation of (a, b, c).

— The pair $\langle \sqcap, \sqcup \rangle$ is called a *symmetric tournament pair (STP)* if it is conservative and both operations $\sqcap, \sqcup$ are commutative, i.e. $a \sqcap b = b \sqcap a$ and $a \sqcup b = b \sqcup a$ for all $a, b \in D$.

— An operation $\mathtt{Mj} : D^3 \to D$ is called a *majority operation* if for every tuple $(a, b, c) \in D^3$ with $|\{a, b, c\}| = 2$ the operation $\mathtt{Mj}$ returns the unique majority element among a, b, c (that occurs twice). An operation $\mathtt{Mn} : D^3 \to D$ is called a *minority operation* if for every tuple $(a, b, c) \in D^3$ with $|\{a, b, c\}| = 2$ the operation $\mathtt{Mn}$ returns the unique minority element among a, b, c (that occurs once).

— The triple $\langle \mathtt{Mj}_1, \mathtt{Mj}_2, \mathtt{Mn}_3 \rangle$ is called an *MJN* if it is conservative, $\mathtt{Mj}_1, \mathtt{Mj}_2$ are (possibly different) majority operations, and $\mathtt{Mn}_3$ is a minority operation.

We say that $\langle \sqcap, \sqcup \rangle$ is a multimorphism of a language Γ, or Γ admits $\langle \sqcap, \sqcup \rangle$, if all cost functions $f \in \Gamma$ satisfy (1). Using a polynomial-time algorithm for minimising submodular functions [Schrijver 2000; Iwata et al. 2001], Cohen *et al.* have obtained the following result:

THEOREM 2.5 ([COHEN ET AL. 2008]). *If a language Γ admits an STP, then Γ is tractable.*

The existence of an MJN multimorphism also leads to tractability. This was shown for a specific choice of an MJN by Cohen *et al.* [Cohen et al. 2006].

Our tractability result, presented in the next section, will include both above-mentioned tractable classes as special cases.

Expressibility. Finally, we define the important notion of expressibility, which captures the idea of introducing auxiliary variables in a VCSP instance and the possibility of minimising over these auxiliary variables. (For crisp languages, this is equivalent to implementation [Creignou et al. 2001], pp-definability [Chen 2006], existential inverse satisfiability [Creignou et al. 2008], structure identification [Dechter and Pearl 1992], and join and projection operations in relational databases [Ullman 1989].

Definition 2.6. A cost function $f : D^m \to \overline{\mathbb{Q}}_+$ is *expressible* over a language Γ if there exists an instance $\mathcal{I} \in \mathsf{VCSP}(\Gamma)$ with the set of nodes $V = \{1, \ldots, m, m+1, \ldots, m+k\}$ where $k \geq 0$ such that

$$f(\boldsymbol{x}) = \min_{\boldsymbol{y} \in D^k} \mathrm{Cost}_{\mathcal{I}}(\boldsymbol{x}, \boldsymbol{y}) \qquad \forall \boldsymbol{x} \in D^m$$

We define Γ^* to be the *expressive power* of Γ; that is, the set of all cost functions f such that f is expressible over Γ.

The importance of expressibility is in the following result:

THEOREM 2.7 ([COHEN ET AL. 2006]).
For any language Γ, Γ *is tractable iff* Γ^* *is tractable.*

It is easy to observe and well known that any polymorphism (multimorphism) of Γ is also a polymorphism (multimorphism) of Γ^* [Cohen et al. 2006].

3. OUR RESULTS

In this section, we relate the complexity of a conservative language Γ to certain properties of a carefully chosen graph G_Γ associated with Γ.

Given a conservative language Γ, let $G_\Gamma = (P, E)$ be the graph with the set of nodes $P = \{(a, b) \mid a, b \in D, a \neq b\}$ and the set of edges E defined as follows: there is an edge between $(a, b) \in P$ and $(a', b') \in P$ iff there exists a binary cost function $f \in \Gamma^*$ such that

$$f(a, a') + f(b, b') > f(a, b') + f(b, a') \ , \quad (a, b'), (b, a') \in \mathtt{dom} f \tag{3}$$

Note that G_Γ may have self-loops. For a node $p \in P$ we denote the self-loop by $\{p, p\}$. We say that an edge $\{(a, b), (a', b')\} \in E$ is *soft* if there exists a binary cost function $f \in \Gamma^*$ satisfying (3) such that at least one of the assignments (a, a'), (b, b') is in $\mathtt{dom} f$. Edges in E that are not soft are called *hard*. For a node $p = (a, b) \in P$ we denote $\bar{p} = (b, a) \in P$. Note, a somewhat similar graph (but not the same) was used by Takhanov [Takhanov 2010a] for languages Γ containing crisp functions and finite unary cost functions.[1]

We denote by $M \subseteq P$ the set of vertices $(a, b) \in P$ without self-loops, and by $\overline{M} = P - M$ the complement of M. It follows from the definition that the set M is *symmetric*, i.e. $(a, b) \in M$ iff $(b, a) \in M$. We will write $\{a, b\} \in M$ to indicate that $(a, b) \in M$; this is consistent due to the symmetry of M. Similarly, we will write $\{a, b\} \in \overline{M}$ if $(a, b) \in \overline{M}$, and $\{a, b\} \in P$ if $(a, b) \in P$, i.e. $a, b \in D$ and $a \neq b$.

Definition 3.1. Let $\langle \sqcap, \sqcup \rangle$ be a pair of binary operations and $\langle \mathtt{Mj}_1, \mathtt{Mj}_2, \mathtt{Mn}_3 \rangle$ be a triple of ternary operations.

— The pair $\langle \sqcap, \sqcup \rangle$ is an ***STP on*** M if $\langle \sqcap, \sqcup \rangle$ is conservative and $\langle \sqcap, \sqcup \rangle$ is commutative on M; that is, for any $\{a, b\} \in M$, $a \sqcap b = b \sqcap a$ and $a \sqcup b = b \sqcup a$.
— The triple $\langle \mathtt{Mj}_1, \mathtt{Mj}_2, \mathtt{Mn}_3 \rangle$ is an ***MJN on*** $\overline{M}$ if it is conservative and for each triple $(a, b, c) \in D^3$ with $\{a, b, c\} = \{x, y\} \in \overline{M}$ operations $\mathtt{Mj}_1(a, b, c)$, $\mathtt{Mj}_2(a, b, c)$ return the unique majority element among a, b, c (that occurs twice) and $\mathtt{Mn}_3(a, b, c)$ returns the remaining minority element.

Our main results are given by the following three theorems.

THEOREM 3.2. *Let* Γ *be a conservative language.*

(a) If G_Γ *has a soft self-loop then* Γ *is NP-hard.*
(b) If G_Γ *does not have soft self-loops then* Γ *admits a pair* $\langle \sqcup, \sqcap \rangle$ *which is an STP on* M *and satisfies additionally* $a \sqcap b = a$, $a \sqcup b = b$ *for* $\{a, b\} \in \overline{M}$.

THEOREM 3.3. *Let* Γ *be a conservative language. If* Γ *does not admit an MJN on* $\overline{M}$ *then it is NP-hard.*

[1] Roughly speaking, the graph structure in [Takhanov 2010a] was defined via a "min" polymorphism rather than a $\langle \min, \max \rangle$ multimorphism, so the property $\{p, q\} \in E \Rightarrow \{\bar{p}, \bar{q}\} \in E$ (that we prove for our graph in the next section) might not hold in Takhanov's case. Also, in [Takhanov 2010a] edges were not classified as being soft or hard.

THEOREM 3.4. *Suppose language Γ admits an STP on M' and an MJN on $P - M'$, for some choice of symmetric $M' \subseteq P$. Then Γ is tractable.*

Theorems 3.2-3.4 give the dichotomy result for conservative languages:

THEOREM 3.5. *Let Γ be a conservative language and P the set of nodes of G_Γ. If there is a symmetric set $M' \subseteq P$ such that Γ admits an STP on M' and an MJN on $P - M'$ then Γ is tractable. Otherwise Γ is NP-hard.*

PROOF. The first part follows from Theorem 3.4; let us show the second part. Suppose that the precondition of the theorem does not hold, then one of the following cases must be true (we assume below that M is the set of nodes without self-loops in G_Γ):

— G_Γ has a soft self-loop. Then Γ is NP-hard by Theorem 3.2(a).
— G_Γ does not have soft self-loops and Γ does not admit an STP on M. This is a contradiction by Theorem 3.2(b).
— G_Γ does not have soft self-loops and Γ does not admit an MJN on $\overline{M}$. Then Γ is NP-hard by Theorem 3.3.

□

In the finite-valued case, we get a simpler tractability criterion, namely an STP multimorphism, which turns out to be equivalent to the condition of submodularity [Schrijver 2000; Iwata et al. 2001].

THEOREM 3.6. *Let Γ be a conservative finite-valued language. If Γ is submodular on some chain on D then Γ is tractable. Otherwise Γ is NP-hard.*

PROOF. Consider the graph G_Γ associated with Γ. If G_Γ contains a soft self-loop, then, by Theorem 3.2(a), Γ is NP-hard. Suppose that G_Γ does not contain soft self-loops. As Γ is finite-valued, G_Γ cannot have hard self-loops. Therefore, $\overline{M}$ is empty and $M = P$. By Theorem 3.2(b), Γ admits an STP and the tractability then follows from Theorem 3.4.

If a finite-valued language admits an STP multimorphism, it also admits a submodularity multimorphism. This result is implicitly contained in [Cohen et al. 2008]. In particular, the STP might contain cycles, but [Cohen et al. 2008, Lemma 7.15] tells us that on cycles we have, in the finite-valued case, only unary cost functions. Since an acyclic tournament is equivalent to a total order on the domain and unary cost functions are submodular with respect to any total order, it follows that the cost functions admitting the STP must be submodular with respect to some total order.

A formal proof of this statement (based on a different argument) is given in [Kolmogorov 2012, Section 4]. □

Given a finite language Γ, the *meta-problem* [Creignou et al. 2001] consists in deciding whether Γ is tractable. For languages defined on a fixed domain, the meta-problem is solvable in polynomial time. This follows from the fact that for any fixed domain, there is only a fixed number of possible sets M and a fixed number of possible binary multimorphisms that behave as an STP on M and a fixed number of possible ternary multimorphisms that behave as an MJN on $P - M$; each such candidate can be tested whether it is indeed a multimorphism of Γ.

Our tractability result holds true even in the so-called *uniform* model [Kolaitis and Vardi 2000], in which the language is treated as part of the input (and thus the domain is finite, but not fixed); however, we need to assume that an STP operation on M is also a part of the input. We do not know what the complexity of the problem without this assumption is. The complexity of the meta-problem in the uniform case also remains open.

4. PROOF PRELIMINARIES: STRENGTHENING THE DEFINITION OF CONSERVATIVITY

First, we show that we can strengthen the definition of conservative languages without loss of generality. More precisely, we prove in this section that it suffices to establish Theorems 3.2 and 3.3 under the following simplifying assumption:

Assumption 1. Γ contains all general-valued unary cost functions $u : D \to \overline{\mathbb{Q}}_+$.

For a language Γ, let $\bar{\Gamma}$ be the language obtained from Γ by adding all possible general-valued unary cost functions $u : D \to \overline{\mathbb{Q}}_+$. Note, $\bar{\Gamma}$ may be different from Γ since Γ is only guaranteed to have all possible $\{0,1\}$-valued unary cost functions.

PROPOSITION 4.1. *(a) Graphs G_Γ and $G_{\bar{\Gamma}}$ are the same: if $\{(a,b),(a',b')\}$ is a soft (hard) edge in G_Γ then it is also a soft (hard) edge in $G_{\bar{\Gamma}}$, and vice versa. (b) If $\bar{\Gamma}$ is NP-hard then so is Γ.*

PROOF. Let $\mathbb{Z}_+$ be the set of non-negative integers, and let $\overline{\mathbb{Z}}_+ = \mathbb{Z}_+ \cup \{\infty\}$. It is easy to see that any unary cost function $u : D \to \mathbb{Z}_+$ can be represented as a sum of at most $\max_{a \in D} u(a)$ $\{0,1\}$-valued unary cost functions from Γ, and so $u \in \Gamma^*$; we will use this fact below.

Part (a) One direction is trivial: if $\{(a,b),(a',b')\} \in G_\Gamma$ then $\{(a,b),(a',b')\} \in G_{\bar{\Gamma}}$, and if $\{(a,b),(a',b')\}$ is soft in G_Γ then it is also soft in $G_{\bar{\Gamma}}$. For the other direction we need to show the following: (i) if $\{(a,b),(a',b')\}$ is an edge in $G_{\bar{\Gamma}}$ then it is also an edge in G_Γ, and (ii) if $\{(a,b),(a',b')\}$ is a soft edge in $G_{\bar{\Gamma}}$ then it is also soft in G_Γ.

Suppose that $\{(a,b),(a',b')\} \in G_{\bar{\Gamma}}$. Let $f \in (\bar{\Gamma})^*$ be the corresponding binary cost function. If the edge $\{(a,b),(a',b')\}$ is soft in $G_{\bar{\Gamma}}$, then we choose f according to the definition of a soft edge. We have

$$f(x,y) = \min_{\boldsymbol{z} \in D^{m-2}} g(x,y,\boldsymbol{z}) \qquad \forall x,y \in D$$

where $g : D^m \to \overline{\mathbb{Q}}_+$ is a sum of cost functions from $\bar{\Gamma}$. We can assume without loss of generality that all unary terms present in this sum are $\overline{\mathbb{Z}}_+$-valued. Indeed, this can be ensured by multiplying g by an appropriate integer R. (More precisely, unary terms $u : D \to \overline{\mathbb{Q}}_+$ in the sum are replaced with terms $R \cdot u \in \bar{\Gamma}$, and other terms h in the sum are replaced by R copies of h.)

Let C be a sufficiently large finite integer constant (namely, $C > 2 \cdot \max\{g(\boldsymbol{z}) \mid \boldsymbol{z} \in \mathtt{dom} g\}$), and let g^C be the function obtained from g as follows: we take every unary cost function $u : D \to \overline{\mathbb{Q}}_+$ present in g and replace it with function $u^C(z) = \min\{u(z), C\}$. Clearly, $g^C \in \Gamma^*$. Define

$$f^C(x,y) = \min_{\boldsymbol{z} \in D^{m-2}} g^C(x,y,\boldsymbol{z}) \qquad \forall x,y \in D$$

then $f^C \in \Gamma^*$. It is easy to see that f and f^C have the following relationship: (i) if $f(x,y) < \infty$ then $f^C(x,y) = f(x,y) < C$; (ii) if $f(x,y) = \infty$ then $f^C(x,y) \geq C$. We have $f(a,a') + f(b,b') > f(a,b') + f(b,a')$ and $(a,b'),(b,a') \in \mathtt{dom} f$; this implies that $f^C(a,a') + f^C(b,b') > f^C(a,b') + f^C(b,a')$, and thus $\{(a,b),(a',b')\} \in G_\Gamma$. If the edge $\{(a,b),(a',b')\}$ is soft in $G_{\bar{\Gamma}}$ then at least one of the assignments $(a,a'),(b,b')$ is in $\mathtt{dom} f$ (and thus in $\mathtt{dom} f^C$), and so $\{(a,b),(a',b')\}$ is soft in G_Γ.

Part (b) Suppose that $\bar{\Gamma}$ is NP-hard, i.e. there exists a finite language $\bar{\Gamma}' \subseteq \bar{\Gamma}$ which is NP-hard. Let Γ' be the language obtained from $\bar{\Gamma}'$ by first removing unary cost function $u : D \to \overline{\mathbb{Q}}_+$ present in $\bar{\Gamma}'$, and then adding all possible $\{0,1\}$-valued unary cost functions $u : D \to \{0,1\}$. Clearly, $\Gamma' \subseteq \Gamma$. We prove below that Γ' is NP-hard using a reduction from $\bar{\Gamma}'$.

Let R be a constant integer number such that multiplying unary cost functions from $\bar{\Gamma}'$ by R gives $\overline{\mathbb{Z}}_+$-valued functions. Also let $C_\circ$ be a sufficiently large finite integer constant, namely $C_\circ > \max\{R \cdot f(\boldsymbol{x}) \mid f \in \bar{\Gamma}', \boldsymbol{x} \in \mathtt{dom} f\}$. Now consider an instance $\bar{\mathcal{I}}$ from $\mathsf{VCSP}(\bar{\Gamma}')$ with the cost function

$$f(\boldsymbol{x}) = \sum_{t \in T_1} u_t\left(x_{i(t,1)}\right) + \sum_{t \in T_*} f_t\left(x_{i(t,1)}, \ldots, x_{i(t,m_t)}\right)$$

where T_1 is the index set of unary cost functions and T_* is the index set of cost functions of higher arities. Thus, $u_t \in \bar{\Gamma}'$ for $t \in T_1$ and $f_t \in \bar{\Gamma}'$ for $t \in T_*$. For each $t \in T_1$ we define the unary cost function $u_t^C(z) = \min\{R \cdot u_t(z), C\}$ where $C = C_\circ \cdot (|T_1| + |T_*|)$. Note, we have $C > \max\{R \cdot f(\boldsymbol{x}) \mid \boldsymbol{x} \in \mathtt{dom} f\}$.

Let us define instance $\mathcal{I}$ with the cost function

$$f^C(\boldsymbol{x}) = \sum_{t \in T_1} u_t^C\left(x_{i(t,1)}\right) + \sum_{t \in T_*} R \cdot f_t\left(x_{i(t,1)}, \ldots, x_{i(t,m_t)}\right)$$

It can be viewed as an instance from Γ'. Indeed, u_t^C can be represented as a sum of at most C $\{0,1\}$-valued unary cost functions from Γ', and the multiplication of R and f_t can be simulated by repeating the latter term R times. Then f_C contains at most $C|T_1| + R|T_*| = C_\circ(|T_1| + |T_*|)|T_1| + R|T_*|$ terms, so the size of instance $\mathcal{I}$ is bounded by a polynomial function of the size of $\bar{\mathcal{I}}$.

It is easy to see that f and f^C have the following relationship: (i) if $f(\boldsymbol{x}) < \infty$ then $f^C(\boldsymbol{x}) = R \cdot f(\boldsymbol{x}) < C$; (ii) if $f(\boldsymbol{x}) = \infty$ then $f^C(\boldsymbol{x}) \geq C$. Thus, solving $\mathcal{I}$ will also solve $\bar{\mathcal{I}}$. □

Proposition 4.1 shows that it suffices to prove Theorems 3.2 and 3.3 for $\bar{\Gamma}$. Indeed, consider Theorem 3.2 for a conservative language Γ. If G_Γ has a soft self-loop then by Proposition 4.1(a) so does $G_{\bar{\Gamma}}$. Theorem 3.2(a) for $\bar{\Gamma}$ would imply that $\bar{\Gamma}$ is NP-hard, and therefore Γ is also NP-hard by Proposition 4.1(b). If G_Γ does not have soft self-loops then neither does $G_{\bar{\Gamma}}$. Theorem 3.2(b) for $\bar{\Gamma}$ would imply that $\bar{\Gamma}$ admits the appropriate multimorphism $\langle \sqcup, \sqcap \rangle$ that is an STP on M. (Note, the definition of M is the same for both Γ and $\bar{\Gamma}$ by proposition 4.1(a).) Since $\Gamma \subseteq \bar{\Gamma}$, $\langle \sqcup, \sqcap \rangle$ is also a multimorphism of Γ.

A similar argument holds for Theorem 3.3. If $\bar{\Gamma}$ admits an MJN on $\overline{M}$ then so does Γ. If $\bar{\Gamma}$ does not admit an MJN on $\overline{M}$ then Theorem 3.3 for $\bar{\Gamma}$ and Proposition 4.1(b) would imply that Γ is NP-hard.

In conclusion, from now on we will assume that Γ satisfies Assumption 1 when proving Theorems 3.2 and 3.3.

5. PROOF OF THEOREM 3.2

In Section 5.1 we will first prove part (a). Then in Section 5.2 we will prove certain properties of G_Γ assuming that G_Γ does not have self-loops. Using these properties, we will construct an STP on M in Section 5.3.

5.1. NP-hard case

In this section we prove Theorem 3.2(a). From the assumption, there is a binary cost function $f \in \Gamma^*$ such that $f(a,a) + f(b,b) > f(a,b) + f(b,a)$, and at least one of the assignments $(a,a), (b,b)$ is in $\mathtt{dom} f$. First, let us assume that both (a,a) and (b,b) are in $\mathtt{dom} f$. Define a binary cost function g as follows: $g(x,y) = f(x,y) + f(y,x)$. Clearly, $g \in \Gamma^*$ and g has the following properties: $g(a,b) = g(b,a)$ and at least one of $\{g(a,a), g(b,b)\}$ is strictly bigger than $g(a,b)$. Let $\alpha = g(a,a)$ and $\beta = g(b,b)$. If $\alpha \neq \beta$, let $\alpha < \beta$ (the other case is analogous). Define a binary cost function h as follows: $h(x,y) = g(x,y) + u(x) + u(y)$, where $u(x) = (\beta - \alpha)/2$ if $x = a$, $u(x) = 0$ if $x = b$, and $u(x) = \infty$ otherwise. Clearly, $h \in \Gamma^*$ and h satisfies $h(a,a) = h(b,b) > h(a,b) = h(b,a)$. Now if

$h(a,a) = h(b,b) = 1$ and $h(a,b) = h(b,a) = 0$, this would correspond to the Max-Cut problem, which is NP-hard [Garey and Johnson 1979]. Since adding a constant to all cost functions and scaling all costs by a constant factor do not affect the difficulty of solving a VCSP instance, and Γ is conservative, we can conclude that Γ is intractable.

Without loss of generality, let us now assume that $(a,a) \in \mathtt{dom} f$ and $(b,b) \notin \mathtt{dom} f$. As above, define $g(x,y) = f(x,y) + f(y,x)$. Clearly, $g \in \Gamma^*$ and g has the following properties: $g(b,b) = \infty$ and $g(a,b) = g(b,a)$. Let $\alpha = g(a,a)$ and $\beta = g(a,b)$. If $\alpha \neq \beta$, let $\alpha < \beta$ (the other case is analogous). Define a binary cost function $h(x,y) = g(x,y) + u(x) + u(y)$, where $u(x) = (\beta - \alpha)/2$ if $x = a$, $u(x) = 0$ if $x = b$, and $u(x) = \infty$ otherwise. Clearly, $h \in \Gamma^*$ and h satisfies $h(a,a) = h(a,b) = h(b,a) = \alpha'$ and $h(b,b) = \infty$, where α' is some finite constant. Since adding a constant to h does not affect the difficulty of solving a VCSP instance, we can assume without loss of generality that $\alpha' = 0$. Using h and unary cost functions, we now reduce from the maximum independent set problem in graphs, a well-known NP-hard problem [Garey and Johnson 1979]. Given a graph (V,E), we define a $\mathsf{VCSP}(\Gamma)$ instance $\mathcal{I}$ with the set of nodes V, the set of vertices in G, and with the cost function $\sum_{\{i,j\}\in E} h(i,j) + \sum_{i\in V} u_{\{a,b\}}(i) + \sum_{i\in V} u'(i)$, where $u_{\{a,b\}}(x) = 0$ if $x \in \{a,b\}$ and $u_{\{a,b\}}(x) = \infty$ otherwise, and $u'(x) = 1$ if $x = a$, $u'(x) = 0$ if $x = b$, and $u'(x) = \infty$ otherwise. Intuitively, the domain value a represents "not being in the independent set" and the domain value b represents "being in the independent set". The binary cost functions h ensure that no adjacent vertices are both included in the independent set. The unary cost functions u enforce the effective domain of every node to be $\{a,b\}$. Finally, the unary cost functions u' count the number of nodes assigned the value a. Since minimising the number of variables assigned a is the same as maximising the number of variables assigned b, a solution to $\mathcal{I}$ corresponds to a maximum independent set in G. □

5.2. Properties of graph G_Γ

From now on we assume that E does not have soft self-loops. Our goal is to show that Γ admits an STP on M.

In the lemma below, a *path* of length k is a sequence of edges $\{p_0,p_1\},\{p_1,p_2\},\ldots,\{p_{k-1},p_k\}$, where $\{p_{i-1},p_i\} \in E$. Note that we allow edge repetitions. A path is *even* iff its length is even. A path is a *cycle* if $p_0 = p_k$. If $X \subseteq P$ then $(X, E[X])$ denotes the subgraph of (P,E) induced by X.

LEMMA 5.1. *Graph $G_\Gamma = (P,E)$ satisfies the following properties:*

(a) $\{p,q\} \in E$ implies $\{\bar{p},\bar{q}\} \in E$ and vice versa. The two edges are either both soft or both hard.
(b) Suppose that $\{p,q\} \in E$ and $\{q,r\} \in E$, then $\{p,\bar{r}\} \in E$. If at least one of the first two edges is soft then the third edge is also soft.
(c) For each $p \in P$, nodes p and $\bar{p}$ are either both in M or both in $\overline{M}$.
(d) There are no edges from M to $\overline{M}$.
(e) Graph $(M, E[M])$ does not have odd cycles.
(f) If node p is not isolated (i.e. it has at least one incident edge $\{p,q\} \in E$) then $\{p,\bar{p}\} \in E$.
(g) Nodes $p \in \overline{M}$ do not have incident soft edges.

PROOF.

(a) Follows from the definition.

(b) Let $p = (a_1,b_1)$, $q = (a_2,b_2)$ and $r = (a_3,b_3)$. From the definition of the graph, let $f,g \in \Gamma^*$ be the binary cost functions such that $(*)$ $f(a_1,a_2) + f(b_1,b_2) > f(a_1,b_2) + f(b_1,a_2)$ and $g(a_2,a_3) + g(b_2,b_3) > g(a_2,b_3) + g(b_2,a_3)$. Without loss of generality, we can

assume that

$$\begin{array}{lll} f(a_1,a_2)=\alpha, & f(a_1,b_2)=f(b_1,a_2)=\gamma, & f(b_1,b_2)=\alpha' \\ g(a_2,a_3)=\beta, & g(a_2,b_3)=g(b_2,a_3)=\gamma, & g(b_2,b_3)=\beta' \end{array} \tag{4}$$

This can be achieved as follows. First we show that we can assume that $f(a_1,b_2) = f(b_1,a_2)$. Without loss of generality, let $f(a_1,b_2) < f(b_1,a_2)$. Define $f'(x,y) = f(x,y) + u(x)$, where $u(x) = 0$ if $x = b_1$, $u(x) = f(b_1,a_2) - f(a_1,b_2)$ if $x = a_1$, and $u(x) = \infty$ if $x \notin \{a_1,b_1\}$. Clearly, $f' \in \Gamma^*$, $f'(a_1,a_2)+f'(b_1,b_2) > f'(a_1,b_2)+f'(b_1,a_2)$, and $f'(a_1,b_2) = f'(b_1,a_2)$. Similarly, we can assume that $g(a_2,b_3) = g(b_2,a_3)$. Let $\gamma = f(a_1,b_2)$ and $\gamma' = g(a_2,b_3)$. If $\gamma = \gamma'$ we are done. Without loss of generality, let $\gamma < \gamma'$. Define $f'(x,y) = f(x,y)+u(x)$, where $u(x) = \gamma' - \gamma$ if $x \in \{a_1,b_1\}$ and $u(x) = \infty$ otherwise. The cost functions f' and g now satisfy $(*)$ and (4).

From $(*)$ we get $\alpha + \alpha' > 2\gamma$; thus, by adding unary terms to f we can ensure that $\alpha > \gamma$ and $\alpha' > \gamma$. Similarly, we can assume that $\beta > \gamma$ and $\beta' > \gamma$. (Note that γ must be finite.)

Let $h(x,z) = \min_{y\in D}\{f(x,y)+u_{\{a_2,b_2\}}(y)+g(y,z)\}$, where $u_{\{a_2,b_2\}}(y) = 0$ if $y \in \{a_2,b_2\}$, and $u_{\{a_2,b_2\}}(y) = \infty$ otherwise. From the definition of h and (4) we get $h(a_1,a_3) = h(b_1,b_3) = 2\gamma$ and $h(a_1,b_3) = \gamma + \min\{\alpha,\beta'\} > 2\gamma$, $h(b_1,a_3) = \gamma + \min\{\alpha',\beta\} > 2\gamma$. Therefore, $h(a_1,b_3) + h(b_1,a_3) > h(a_1,a_3) + h(b_1,b_3)$, and so $\{p,\bar{r}\} \in E$.

Now suppose that at least one of the edges $\{p,q\}$, $\{q,r\}$ is soft, then we can assume that at least one of α, α', β, β' is finite, and so at least one of $h(a_1,b_3)$, $h(b_1,a_3)$ is finite, and thus $\{p,\bar{r}\}$ is soft.

(c) Follows from (a).

(d) Suppose $\{p,q\} \in E$ and $q \in \overline{M}$. The latter fact implies $\{q,q\} \in E$, so by (b) we have $\{p,\bar{q}\} \in E$. From (a) we also get $\{q,\bar{p}\} \in E$. Applying (b) again gives $\{p,p\} \in E$. Thus $p \in \overline{M}$.

(e) We prove by induction on k that $(M,E[M])$ does not have cycles of length $2k+1$. For $k = 0$ the claim is by assumption (nodes of M do not have self-loops). Suppose it holds for $k \geq 0$, and suppose that $(M,E[M])$ has a cycle $\mathcal{P},\{p,q\},\{q,r\},\{r,s\}$ of length $2k+3$ where $\mathcal{P}$ is a path from $s \in M$ to $p \in M$ of length $2k$. Properties (b) and (a) give respectively $\{p,\bar{r}\} \in E$ and $\{\bar{r},\bar{s}\} \in E$. Applying (b) again gives $\{p,s\} \in E$, therefore $(M,E[M])$ has a cycle $\mathcal{P},\{p,s\}$ of length $2k+1$. This contradicts the induction hypothesis.

(f) Follows from (b).

(g) Suppose $p \in \overline{M}$ (implying E has a hard self-loop $\{p,p\}$) and $\{p,q\}$ is a soft edge in E. Properties (b) and (a) give respectively $\{p,\bar{q}\} \in E$ and $\{\bar{q},\bar{p}\} \in E$, and furthermore both edges are soft. Applying (b) again gives that $\{p,p\} \in E$ and this edge is soft. This contradicts the assumption that (P,E) does not have soft self-loops. □

5.3. Constructing $\langle \sqcap, \sqcup \rangle$

In this section we complete the proof of Theorem 3.2 by constructing a pair of operations $\langle \sqcap, \sqcup \rangle$ for Γ that behaves as an STP on M and as a multi-projection (returning its two arguments in the same order) on $\overline{M}$.

LEMMA 5.2. *There exists an assignment $\sigma : M \to \{-1,+1\}$ such that (i) $\sigma(p) = -\sigma(q)$ for all edges $\{p,q\} \in E$, and (ii) $\sigma(p) = -\sigma(\bar{p})$ for all $p \in M$.*

PROOF. By Lemma 5.1(e) graph $(M,E[M])$ does not have odd cycles. Therefore, graph $(M,E[M])$ is bipartite and there exists an assignment $\sigma : M \to \{-1,+1\}$ that satisfies property (i). Let us modify this assignment as follows: for each isolated node $p \in M$ (i.e. a node without incident edges) set $\sigma(p)$, $\sigma(\bar{p})$ so that $\sigma(p) = -\sigma(\bar{p}) \in$

$\{-1,+1\}$. (Note, if p is isolated then by Lemma 5.1(a) so is $\bar{p}$). Clearly, property (i) still holds. Property (ii) holds for each node $p \in M$ as well: if p is isolated then (ii) holds by construction, otherwise by Lemma 5.1(f) there exists an edge $\{p,\bar{p}\} \in E$, and so (ii) follows from property (i). □

Given the assignment σ constructed in Lemma 5.2, we now define operations $\sqcap, \sqcup : D^2 \to D$ as follows:

— $a \sqcap a = a \sqcup a = a$ for $a \in D$.
— If $(a,b) \in M$ then $a \sqcap b$ and $a \sqcup b$ are the unique elements of D satisfying $\{a \sqcap b, a \sqcup b\} = \{a,b\}$ and $\sigma(a \sqcap b, a \sqcup b) = +1$.
— If $(a,b) \in \overline{M}$ then $a \sqcap b = a$ and $a \sqcup b = b$.

LEMMA 5.3. *For any binary cost function $f \in \Gamma^*$ and any $\boldsymbol{x}, \boldsymbol{y} \in \mathtt{dom} f$ there holds*

$$f(\boldsymbol{x} \sqcap \boldsymbol{y}) + f(\boldsymbol{x} \sqcup \boldsymbol{y}) \le f(\boldsymbol{x}) + f(\boldsymbol{y}) \tag{5}$$

PROOF. Denote $(a,a') = \boldsymbol{x} \sqcap \boldsymbol{y}$ and $(b,b') = \boldsymbol{x} \sqcup \boldsymbol{y}$. We can assume without loss of generality that $\{\boldsymbol{x},\boldsymbol{y}\} \neq \{(a,a'),(b,b')\}$, otherwise the claim is straightforward. It is easy to check that the assumption has two implications: (i) $a \neq b$ and $a' \neq b'$; (ii) $\{\boldsymbol{x},\boldsymbol{y}\} = \{(a,b'),(b,a')\}$.

If $f(a,a') + f(b,b') = f(a,b') + f(b,a')$, then (5) holds trivially. If $f(a,a') + f(b,b') \neq f(a,b') + f(b,a')$, then E contains at least one of the edges $\{(a,b),(a',b')\}$, $\{(a,b),(b',a')\}$. By Lemma 5.1(c) and Lemma 5.1(d), the pairs (a,b) and (a',b') must either be both in $\overline{M}$ or both in M. In the former case, (5) contradicts the above assumptions, so we assume the latter case.

The definition of $\sqcap, \sqcup$ and the fact that $(a,a') = \boldsymbol{x} \sqcap \boldsymbol{y}$ and $(b,b') = \boldsymbol{x} \sqcup \boldsymbol{y}$ imply that $\sigma(a,b) = \sigma(a',b') = +1$. Thus, the edge set E does contain $\{(a,b),(a',b')\}$, and therefore

$$f(a,a') + f(b,b') \le f(a,b') + f(b,a')$$

which is equivalent to (5). □

In order to proceed, we introduce the following notation. Given a cost function f of arity m, we denote by V the set of variables corresponding to the arguments of f, with $|V| = m$. For two assignments $\boldsymbol{x}, \boldsymbol{y} \in D^m$ we denote by $\Delta(\boldsymbol{x},\boldsymbol{y}) = \{i \in V \mid x_i \neq y_i\}$ the set of variables on which $\boldsymbol{x}$ and $\boldsymbol{y}$ differ.

LEMMA 5.4. *Condition* (5) *holds for any cost function $f \in \Gamma^*$ and assignments $\boldsymbol{x}, \boldsymbol{y} \in \mathtt{dom} f$ with $|\Delta(\boldsymbol{x},\boldsymbol{y})| \le 2$.*

PROOF. If $|\Delta(\boldsymbol{x},\boldsymbol{y})| \le 1$ then $\{\boldsymbol{x} \sqcap \boldsymbol{y}, \boldsymbol{x} \sqcup \boldsymbol{y}\} = \{\boldsymbol{x},\boldsymbol{y}\}$, so the claim is trivial. We now prove it in the case $|\Delta(\boldsymbol{x},\boldsymbol{y})| = 2$ using induction on $|V|$. The base case $|V| = 2$ follows from Lemma 5.3; suppose that $|V| \ge 3$. Choose $k \in V - \Delta(\boldsymbol{x},\boldsymbol{y})$. For simplicity of notation, let us assume that k corresponds to the first argument of f. Define a cost function of $|V|-1$ variables by

$$g(\boldsymbol{z}) = \min_{a \in D} \{u(a) + f(a,\boldsymbol{z})\} \qquad \forall \boldsymbol{z} \in D^{V-\{k\}} \tag{6}$$

where u is the following unary cost function: $u(a) = 0$ if $a = x_k = y_k$, and $u(a) = \infty$ otherwise.

Let $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{y}}$ be the restrictions of respectively $\boldsymbol{x}$ and $\boldsymbol{y}$ to $V - \{k\}$. Clearly, $g \in \Gamma^*$, $g(\hat{\boldsymbol{x}}) = f(\boldsymbol{x}) < \infty$ and $g(\hat{\boldsymbol{y}}) = f(\boldsymbol{y}) < \infty$. By the induction hypothesis

$$g(\hat{\boldsymbol{x}} \sqcap \hat{\boldsymbol{y}}) + g(\hat{\boldsymbol{x}} \sqcup \hat{\boldsymbol{y}}) \le g(\hat{\boldsymbol{x}}) + g(\hat{\boldsymbol{y}}) = f(\boldsymbol{x}) + f(\boldsymbol{y}) \tag{7}$$

This implies that $g(\hat{\boldsymbol{x}} \sqcap \hat{\boldsymbol{y}}) < \infty$, which is possible only if $g(\hat{\boldsymbol{x}} \sqcap \hat{\boldsymbol{y}}) = f(a, \hat{\boldsymbol{x}} \sqcap \hat{\boldsymbol{y}}) = f(\boldsymbol{x} \sqcap \boldsymbol{y})$ where $a = x_k = y_k$. Similarly, $g(\hat{\boldsymbol{x}} \sqcup \hat{\boldsymbol{y}}) = f(a, \hat{\boldsymbol{x}} \sqcup \hat{\boldsymbol{y}}) = f(\boldsymbol{x} \sqcup \boldsymbol{y})$. Thus, (7) is equivalent to (5). □

LEMMA 5.5. *Condition* (5) *holds for any cost function* $f \in \Gamma^*$ *and any* $\boldsymbol{x}, \boldsymbol{y} \in \mathtt{dom} f$.

PROOF. We use induction on $|\Delta(\boldsymbol{x}, \boldsymbol{y})|$. The base case $|\Delta(\boldsymbol{x}, \boldsymbol{y})| \leq 2$ follows from Lemma 5.4; suppose that $|\Delta(\boldsymbol{x}, \boldsymbol{y})| \geq 3$. Let us partition $\Delta(\boldsymbol{x}, \boldsymbol{y})$ into three sets A, B, C as follows:

$$\begin{aligned} A &= \{i \in \Delta(\boldsymbol{x}, \boldsymbol{y}) \mid (x_i, y_i) \in M, \ \ (x_i \sqcap y_i, x_i \sqcup y_i) = (x_i, y_i)\} \\ B &= \{i \in \Delta(\boldsymbol{x}, \boldsymbol{y}) \mid (x_i, y_i) \in M, \ \ (x_i \sqcap y_i, x_i \sqcup y_i) = (y_i, x_i)\} \\ C &= \{i \in \Delta(\boldsymbol{x}, \boldsymbol{y}) \mid (x_i, y_i) \in \overline{M}\} \end{aligned}$$

Two cases are possible.

Case 1 $|A \cup C| \geq 2$. Let us choose variable $k \in A \cup C$, and define assignments $\boldsymbol{x}', \boldsymbol{y}'$ as follows: $x'_i = y'_i = x_i = y_i$ if $x_i = y_i$, and for other variables

$$(x'_i, y'_i) = \begin{cases} (x_i, x_i) & \text{if } i = k \\ (y_i, y_i) & \text{if } i \in (A \cup C) - \{k\} \\ (x_i, y_i) & \text{if } i \in B \end{cases}$$

It can be checked that

$$\boldsymbol{x} \sqcap \boldsymbol{y}' = \boldsymbol{x} \sqcap \boldsymbol{y} \qquad \boldsymbol{x} \sqcup \boldsymbol{y}' = \boldsymbol{x}' \qquad \boldsymbol{x}' \sqcap \boldsymbol{y} = \boldsymbol{y}' \qquad \boldsymbol{x}' \sqcup \boldsymbol{y} = \boldsymbol{x} \sqcup \boldsymbol{y}$$

Furthermore, $\Delta(\boldsymbol{x}, \boldsymbol{y}') = \Delta(\boldsymbol{x}, \boldsymbol{y}) - \{k\}$ and $\Delta(\boldsymbol{x}', \boldsymbol{y}) = \Delta(\boldsymbol{x}, \boldsymbol{y}) - ((A \cup C) - \{k\})$ so by the induction hypothesis

$$f(\boldsymbol{x} \sqcap \boldsymbol{y}) + f(\boldsymbol{x}') \leq f(\boldsymbol{x}) + f(\boldsymbol{y}') \tag{8}$$

assuming that $\boldsymbol{y}' \in \mathtt{dom} f$, and

$$f(\boldsymbol{y}') + f(\boldsymbol{x} \sqcup \boldsymbol{y}) \leq f(\boldsymbol{x}') + f(\boldsymbol{y}) \tag{9}$$

assuming that $\boldsymbol{x}' \in \mathtt{dom} f$. Two cases are possible:

— $\boldsymbol{y}' \in \mathtt{dom} f$. Inequality (8) implies that $\boldsymbol{x}' \in \mathtt{dom} f$. The claim then follows from summing (8) and (9).

— $\boldsymbol{y}' \notin \mathtt{dom} f$. Inequality (9) implies that $\boldsymbol{x}' \notin \mathtt{dom} f$. Assume for simplicity of notation that k corresponds to the first argument of f. Define cost function of $|V| - 1$ variables

$$g(\boldsymbol{z}) = \min_{a \in D} \{u(a) + f(a, \boldsymbol{z})\} \qquad \forall \boldsymbol{z} \in D^{V - \{k\}}$$

where $u(a)$ is the following unary cost function: $u(x_k) = 0$, $u(y_k) = C$ and $u(a) = \infty$ for $a \in D - \{x_k, y_k\}$. Here C is a sufficiently large finite constant, namely $C > f(\boldsymbol{x}) + f(\boldsymbol{y})$. Let $\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{x}}', \hat{\boldsymbol{y}}'$ be restrictions of respectively $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{x}', \boldsymbol{y}'$ to $V - \{k\}$. Clearly, $g \in \Gamma^*$ and

$$\begin{aligned} g(\hat{\boldsymbol{y}}) = g(\hat{\boldsymbol{y}}') &= u(y_k) + f(y_k, \hat{\boldsymbol{y}}) = f(\boldsymbol{y}) + C \qquad \text{(since } (x_k, \hat{\boldsymbol{y}}) = \boldsymbol{y}' \notin \mathtt{dom} f\text{)} \\ g(\hat{\boldsymbol{x}}) &= f(x_k, \hat{\boldsymbol{x}}) = f(\boldsymbol{x}) \end{aligned}$$

By the induction hypothesis

$$g(\hat{\boldsymbol{x}} \sqcap \hat{\boldsymbol{y}}) + g(\hat{\boldsymbol{x}} \sqcup \hat{\boldsymbol{y}}) \leq g(\hat{\boldsymbol{x}}) + g(\hat{\boldsymbol{y}}) = f(\boldsymbol{x}) + f(\boldsymbol{y}) + C \tag{10}$$

We have $g(\hat{\boldsymbol{x}} \sqcup \hat{\boldsymbol{y}}) < \infty$, so we must have either $g(\hat{\boldsymbol{x}} \sqcup \hat{\boldsymbol{y}}) = f(x_k, \hat{\boldsymbol{x}} \sqcup \hat{\boldsymbol{y}})$ or $g(\hat{\boldsymbol{x}} \sqcup \hat{\boldsymbol{y}}) = f(y_k, \hat{\boldsymbol{x}} \sqcup \hat{\boldsymbol{y}}) + C = f(\boldsymbol{x} \sqcup \boldsymbol{y}) + C$. The former case is impossible since $(x_k, \hat{\boldsymbol{x}} \sqcup \hat{\boldsymbol{y}}) = \boldsymbol{x}' \notin$

$\mathtt{dom}f$, so $g(\hat{\boldsymbol{x}} \sqcup \hat{\boldsymbol{y}}) = f(\boldsymbol{x} \sqcup \boldsymbol{y}) + C$. Combining it with (10) gives

$$g(\hat{\boldsymbol{x}} \sqcap \hat{\boldsymbol{y}}) + f(\boldsymbol{x} \sqcup \boldsymbol{y}) \leq f(\boldsymbol{x}) + f(\boldsymbol{y}) \tag{11}$$

This implies that $g(\hat{\boldsymbol{x}} \sqcap \hat{\boldsymbol{y}}) < C$, so we must have $g(\hat{\boldsymbol{x}} \sqcap \hat{\boldsymbol{y}}) = f(x_k, \hat{\boldsymbol{x}} \sqcap \hat{\boldsymbol{y}}) = f(\boldsymbol{x} \sqcap \boldsymbol{y})$. Thus, (11) is equivalent to (5).

Case 2 $|B| \geq 2$. Let us choose variable $k \in B$, and define assignments $\boldsymbol{x}', \boldsymbol{y}'$ as follows: $x'_i = y'_i = x_i = y_i$ if $x_i = y_i$, and for other variables

$$(x'_i, y'_i) = \begin{cases} (y_i, y_i) & \text{if } i = k \\ (x_i, y_i) & \text{if } i \in A \cup C \\ (x_i, x_i) & \text{if } i \in B - \{k\} \end{cases}$$

It can be checked that

$$\boldsymbol{x}' \sqcap \boldsymbol{y} = \boldsymbol{x} \sqcap \boldsymbol{y} \qquad \boldsymbol{x}' \sqcup \boldsymbol{y} = \boldsymbol{y}' \qquad \boldsymbol{x} \sqcap \boldsymbol{y}' = \boldsymbol{x}' \qquad \boldsymbol{x} \sqcup \boldsymbol{y}' = \boldsymbol{x} \sqcup \boldsymbol{y}$$

Furthermore, $\Delta(\boldsymbol{x}', \boldsymbol{y}) = \Delta(\boldsymbol{x}, \boldsymbol{y}) - \{k\}$ and $\Delta(\boldsymbol{x}, \boldsymbol{y}') = \Delta(\boldsymbol{x}, \boldsymbol{y}) - (B - \{k\})$ so by the induction hypothesis

$$f(\boldsymbol{x} \sqcap \boldsymbol{y}) + f(\boldsymbol{y}') \leq f(\boldsymbol{x}') + f(\boldsymbol{y}) \tag{12}$$

assuming that $\boldsymbol{x}' \in \mathtt{dom}f$, and

$$f(\boldsymbol{x}') + f(\boldsymbol{x} \sqcup \boldsymbol{y}) \leq f(\boldsymbol{x}) + f(\boldsymbol{y}') \tag{13}$$

assuming that $\boldsymbol{y}' \in \mathtt{dom}f$. Using Inequalities 12 and 13, the same argument as in **Case 1**, distinguishing whether or not $\boldsymbol{x}' \in \mathtt{dom}f$, finishes the proof. □

6. PROOF OF THEOREM 3.3

For a language Γ let $\mathsf{Feas}(\Gamma)$ be the language obtained from Γ by converting all finite values of f to 0 for all $f \in \Gamma$, and let $\mathsf{MH}(\Gamma)$ be the language obtained from $\mathsf{Feas}(\Gamma)$ by adding all possible integer-valued unary cost functions $u : D \to \mathbb{Z}_+$. Note, $\mathsf{MH}(\Gamma)$ corresponds to the *minimum-cost homomorphism* problem introduced in [Gutin et al. 2006] and recently studied in [Takhanov 2010a]. We will need the following fact which is a simple corollary of results of Takhanov [Takhanov 2010a].

THEOREM 6.1. *(a) If* $\mathsf{MH}(\Gamma)$ *does not admit a majority polymorphism then* $\mathsf{MH}(\Gamma)$ *is NP-hard. (b) If* $\mathsf{MH}(\Gamma)$ *is NP-hard then so is* Γ.

PROOF.
Part (a) Takhanov has studied crisp languages including additionally all integer-valued unary cost functions [Takhanov 2010a]. For such a language Γ, he considers the functional clone of all polymorphisms of Γ, denoted by F, and a certain graph denoted by T_F. Takhanov's Theorem 3.3, Theorem 3.4, and Theorem 5.5 give the following:

— If F does not satisfy the necessary local conditions or T_F is not bipartite then Γ is NP-hard.
— If F satisfies the necessary local conditions and T_F is bipartite then F contains a majority operation.

This implies part (a).

Part (b) Let $\mathsf{MH}(\Gamma)' \subseteq \mathsf{MH}(\Gamma)$ be a finite language with costs in $\overline{\mathbb{Z}}_+ = \mathbb{Z}_+ \cup \{\infty\}$ which is NP-hard. Denote by $\mathsf{MH}(\Gamma)'_1$ and $\mathsf{MH}(\Gamma)'_*$ the subsets of $\mathsf{MH}(\Gamma)'$ of arity $m = 1$ and $m \geq 2$ respectively. The definition of $\mathsf{MH}(\Gamma)$ implies that for every $f \in \mathsf{MH}(\Gamma)'_*$ there exists a cost function $f^\circ \in \Gamma$ such that $f(\boldsymbol{x}) = 0$ if $f^\circ(\boldsymbol{x}) < \infty$, and $f(\boldsymbol{x}) = \infty$ if $f^\circ(\boldsymbol{x}) = \infty$. Denote $C = \max\{f^\circ(\boldsymbol{x}) \mid f \in MH(\Gamma)'_*, \boldsymbol{x} \in \mathtt{dom}f^\circ\} + 1$. Construct a language

Γ' as follows:

$$\Gamma' = \left\{u^C \mid u \in \mathsf{MH}(\Gamma)'_1\right\} \quad \cup \quad \left\{f^\circ \mid f \in \mathsf{MH}(\Gamma)'_*\right\}$$

where function u^C is defined by $u^C(z) = C \cdot u(z)$. Clearly, $\Gamma' \subseteq \Gamma$. We prove below that Γ' is NP-hard using a reduction from $\mathsf{MH}(\Gamma)'$.

Let $\hat{\mathcal{I}}$ be an instance from $\mathsf{MH}(\Gamma)'$ with the cost function

$$f(\boldsymbol{x}) = \sum_{t \in T_1} u_t\left(x_{i(t,1)}\right) + \sum_{t \in T_*} f_t\left(x_{i(t,1)}, \ldots, x_{i(t,m_t)}\right)$$

where T_1 is the index set of unary cost functions and T_* is the index set of cost functions of higher arities. Note, $u_t \in \mathsf{MH}(\Gamma)'_1$ for $t \in T_1$ and $f_t \in \mathsf{MH}(\Gamma)'_*$ for $t \in T_*$. Now define instance $\mathcal{I}$ with the cost function

$$f^C(\boldsymbol{x}) = \sum_{t \in T_1} N \cdot u_t^C\left(x_{i(t,1)}\right) + \sum_{t \in T_*} f_t^\circ\left(x_{i(t,1)}, \ldots, x_{i(t,m_t)}\right)$$

where $N = |T_*|$. It can be viewed as an instance from Γ', if we simulate multiplication of N and u_t^C by repeating the latter term N times; the size of the expression grows only polynomially. For any $\boldsymbol{x} \in \mathtt{dom} f$ we have

$$\begin{aligned} f^C(\boldsymbol{x}) \;&\geq\; \sum_{t \in T_1} N \cdot u_t^C\left(x_{i(t,1)}\right) = N \cdot C \cdot f(\boldsymbol{x}) \\ f^C(\boldsymbol{x}) \;&<\; \sum_{t \in T_1} N \cdot u_t^C\left(x_{i(t,1)}\right) + \sum_{t \in T_*} C = N \cdot C \cdot (f(\boldsymbol{x}) + 1) \end{aligned}$$

Furthermore, $f(\boldsymbol{x}) = \infty$ iff $f^C(\boldsymbol{x}) = \infty$. Function f have values in $\overline{\mathbb{Z}}_+$, therefore solving $\mathcal{I}$ will also solve $\hat{\mathcal{I}}$. $\square$

Suppose that Γ does not admit a majority polymorphism. Clearly, this implies that $\mathsf{MH}(\Gamma)$ also does not admit a majority polymorphism. By Theorem 6.1, Γ is NP-hard, and so Theorem 3.3 holds in this case. Hence without loss of generality we can assume:

Assumption 2. Γ admits a majority polymorphism.

By Theorem 3.2(a), if G_Γ has a soft self-loop then Γ is NP-hard. Hence without loss of generality we can assume:

Assumption 3. G_Γ does not have soft self-loops.

Note, a recent paper [Chen et al. 2012] states that our Assumptions 1 and 3 actually imply Assumption 2; their proof also uses results from [Takhanov 2010a]. This would be an alternative way to justify Assumption 2.

We will prove Theorem 3.3 by showing the existence of an MJN multimorphism on $\overline{M}$ under Assumptions 1-3. We denote by $\langle \sqcap, \sqcup \rangle$ an STP multimorphism on M with the properties given in Theorem 3.2(b).

6.1. Constructing $\langle \mathtt{Mj}_1, \mathtt{Mj}_2, \mathtt{Mn}_3 \rangle$

Let us introduce function μ which maps every set $B \subseteq D$ with $|B| \leq 3$ to a subset of B. If $|B| \leq 2$ then we define $\mu(B) = \varnothing$. If $|B| = 3$ then $\mu(B)$ is the set of labels $c \in B$ that satisfy the following condition: if $B - \{c\} = \{a, b\}$ then there exists a binary cost function $f \in \Gamma^*$ and a pair $(a', b') \in \overline{M}$ such that

$$\mathtt{dom} f = \{(a, a'), (b, a'), (c, b')\}$$

If $c \in \mu(B)$ where $B = \{a, b, c\}$ then we will illustrate this fact using the following diagram:

$$\begin{array}{l} a \\ \diagup\!\diagdown\, b \\ c \end{array}$$

LEMMA 6.2. *Consider set $B = \{a, b, c\} \subseteq D$. (a) Set $\mu(B)$ contains at most one label. (b) If $\mu(B) = \{c\}$ then $(a, c) \in \overline{M}$ and $(b, c) \in \overline{M}$.*

PROOF. **Part (a)** Suppose that $a, c \in \mu(\{a, b, c\})$ where $a \neq c$, then there exist binary functions $f, g \in \Gamma^*$ and pairs $(a', b'), (a'', b'') \in \overline{M}$ such that

$$\mathtt{dom} f = \{(a', a), (b', b), (b', c)\} \qquad \mathtt{dom} g = \{(a, a''), (b, a''), (c, b'')\}$$

Consider function

$$h(x', x'') = \min_{x \in D} \{f(x', x) + g(x, x'')\} \tag{14}$$

Clearly, $\mathtt{dom} h = \{(a', a''), (b', a''), (b', b'')\}$, so $(a', b') \in \overline{M}$ has an incident soft edge $\{(a', b'), (b'', a'')\}$ in G_Γ - a contradiction with Lemma 5.1(g).

Part (b) Suppose that $\mu(\{a, b, c\}) = \{c\}$. There exists a binary function $f \in \Gamma^*$ with $\mathtt{dom} f = \{(a, a'), (b, a'), (c, b')\}$ and $(b', a') \in \overline{M}$ (by Lemma 5.1(c)). Therefore, graph G_Γ contains edges $\{(a, c), (b', a')\}$ and $\{(b, c), (b', a')\}$. Lemma 5.1(d) now implies that $(a, c) \in \overline{M}$ and $(b, c) \in \overline{M}$. □

We are now ready to construct operation $\mathtt{MJN} = \langle \mathtt{Mj}_1, \mathtt{Mj}_2, \mathtt{Mn}_3 \rangle$. Given a tuple $(a, b, c) \in D^3$, we define

$$\mathtt{MJN}(a, b, c) \;=\; \begin{cases} (x, x, y) & \text{if } \{\{a, b, c\}\} = \{\{x, x, y\}\}, \{x, y\} \in \overline{M} \quad (15a) \\ (b \sqcap c, b \sqcup c, a) & \text{if } \mu(\{a, b, c\}) = \{a\} \quad (15b) \\ (a \sqcap c, a \sqcup c, b) & \text{if } \mu(\{a, b, c\}) = \{b\} \quad (15c) \\ (a \sqcap b, a \sqcup b, c) & \text{in any other case} \quad (15d) \end{cases}$$

where $\{\{\ldots\}\}$ denotes a *multiset*, i.e. elements' multiplicities are taken into account. It is straightforward to check that the triple $\langle \mathtt{Mj}_1, \mathtt{Mj}_2, \mathtt{Mn}_3 \rangle$ is conservative.

THEOREM 6.3. *If $f \in \Gamma^*$ and $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathtt{dom} f$ then*

$$f(\mathtt{Mj}_1(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})) + f(\mathtt{Mj}_2(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})) + f(\mathtt{Mn}_3(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})) \leq f(\boldsymbol{x}) + f(\boldsymbol{y}) + f(\boldsymbol{z}) \tag{16}$$

The remainder of Section 6 is devoted to the proof of this statement.

6.2. Proof of Theorem 6.3: preliminaries

We say that an instance $(f, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ is *valid* if $f \in \Gamma^*$ and $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathtt{dom} f$. It is *non-violating* if (16) holds, and *violating* otherwise. For a triple $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in D^V$ denote $\delta(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = \sum_{i \in V} |\{x_i, y_i, z_i\}|$, $\Delta(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = \{i \in V \mid x_i \neq y_i\}$ and $\Delta^M(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = \{i \in \Delta(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \mid \{x_i, y_i, z_i\} = \{a, b\} \in M\}$.

Suppose that a violating instance exists. From now on we assume that $(f, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ is a lowest violating instance with respect to the partial order $\preceq$ defined as the lexico-

graphical order with components

$$(\delta(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}), \quad |\Delta(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})|, \quad |\Delta^M(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})|, \quad |\{i \in V \mid \mu(\{x_i,y_i,z_i\}) = \{x_i\}\}|) \tag{17}$$

(the first component is more significant). We denote $\delta_{\min} = \delta(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$. Thus, we have

Assumption 4. All valid instances $(f,\boldsymbol{x}',\boldsymbol{y}',\boldsymbol{z}')$ with $(\boldsymbol{x}',\boldsymbol{y}',\boldsymbol{z}') \prec (\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$ (and in particular with $\delta(\boldsymbol{x}',\boldsymbol{y}',\boldsymbol{z}') < \delta_{\min}$) are non-violating, while the instance $(f,\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$ is violating.

We will assume without loss of generality that for any $\boldsymbol{u} \in \mathtt{dom} f$ there holds $u_i \in \{x_i,y_i,z_i\}$ for all $i \in V$. Indeed, this can be achieved by adding unary cost functions $g_i(u_i)$ to f with $\mathtt{dom} g_i = \{x_i,y_i,z_i\}$; this does not affect the "violatedness" of $(f,\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$.

The following cases can be easily eliminated:

PROPOSITION 6.4. *The following cases are impossible: (a)* $|V| = 1$*; (b)* $|\{x_i,y_i,z_i\}| = 1$ *for some* $i \in V$.

PROOF. If $|V| = 1$ then (16) is a trivial equality contradicting the choice of $(f,\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$. Suppose that $x_i = y_i = z_i = a$, $i \in V$. Consider function

$$g(\boldsymbol{u}) = \min_{d \in D} f(d,\boldsymbol{u}) \qquad \forall \boldsymbol{u} \in D^{\hat{V}}$$

where $\hat{V} = V - \{i\}$ and we assumed for simplicity of notation that i corresponds to the first argument of f. For an assignment $\boldsymbol{w} \in V$ we denote by $\hat{\boldsymbol{w}}$ the restriction of $\boldsymbol{w}$ to $\hat{V}$. Clearly, $g \in \Gamma^*$, $g(\hat{\boldsymbol{x}}) = f(\boldsymbol{x})$, $g(\hat{\boldsymbol{y}}) = f(\boldsymbol{y})$, $g(\hat{\boldsymbol{y}}) = f(\boldsymbol{y})$ and $(\hat{\boldsymbol{x}},\hat{\boldsymbol{y}},\hat{\boldsymbol{z}}) \prec (\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$, so Assumption 4 gives

$$g(\mathtt{Mj}_1(\hat{\boldsymbol{x}},\hat{\boldsymbol{y}},\hat{\boldsymbol{z}})) + g(\mathtt{Mj}_2(\hat{\boldsymbol{x}},\hat{\boldsymbol{y}},\hat{\boldsymbol{z}})) + g(\mathtt{Mn}_3(\hat{\boldsymbol{x}},\hat{\boldsymbol{y}},\hat{\boldsymbol{z}})) \le g(\hat{\boldsymbol{x}}) + g(\hat{\boldsymbol{y}}) + g(\hat{\boldsymbol{z}}) = f(\boldsymbol{x}) + f(\boldsymbol{y}) + f(\boldsymbol{z})$$

This implies that $\mathtt{Mj}_1(\hat{\boldsymbol{x}},\hat{\boldsymbol{y}},\hat{\boldsymbol{z}}) \in \mathtt{dom} g$ and thus $g(\mathtt{Mj}_1(\hat{\boldsymbol{x}},\hat{\boldsymbol{y}},\hat{\boldsymbol{z}})) = f(a,\mathtt{Mj}_1(\hat{\boldsymbol{x}},\hat{\boldsymbol{y}},\hat{\boldsymbol{z}})) = f(\mathtt{Mj}_1(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}))$. Similarly, $g(\mathtt{Mj}_2(\hat{\boldsymbol{x}},\hat{\boldsymbol{y}},\hat{\boldsymbol{z}})) = f(\mathtt{Mj}_2(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}))$ and $g(\mathtt{Mn}_3(\hat{\boldsymbol{x}},\hat{\boldsymbol{y}},\hat{\boldsymbol{z}})) = f(\mathtt{Mn}_3(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}))$, so the inequality above is equivalent to (16). □

It is also easy to show the following fact.

PROPOSITION 6.5. *There exists node* $i \in V$ *for which operation* $\mathtt{MJN}(x_i,y_i,z_i)$ *is defined by equation* (15a)*,* (15b) *or* (15c)*, i.e. either* $\{x_i,y_i,z_i\} = \{a,b\} \in \overline{M}$*,* $\mu(\{x_i,y_i,z_i\}) = \{x_i\}$*, or* $\mu(\{x_i,y_i,z_i\}) = \{y_i\}$.

PROOF. If such a node does not exist then $\mathtt{MJN}(x_i,y_i,z_i)$ is defined by equation (15d) for all nodes $i \in V$, i.e. $\mathtt{MJN}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) = (\boldsymbol{x} \sqcap \boldsymbol{y}, \boldsymbol{x} \sqcup \boldsymbol{y}, \boldsymbol{z})$. The fact that $\langle \sqcap, \sqcup \rangle$ is a multimorphism of f then implies inequality (16), contradicting the choice of $(f,\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$. □

In the next section we show that case (15a) is impossible, while the remaining two cases (15b), (15c) are analysed in section 6.4.

The following equalities are easy to verify; they will be useful for verifying various identities:

$$\alpha \sqcap (\alpha \sqcup \beta) = \alpha \sqcap (\beta \sqcup \alpha) = (\alpha \sqcap \beta) \sqcup \alpha = (\beta \sqcap \alpha) \sqcup \alpha = \alpha \qquad \forall \alpha,\beta \in D \tag{18a}$$

$$\mathtt{MJN}(\alpha,\alpha,\beta) = (\alpha,\alpha,\beta) \qquad \forall \alpha,\beta \in D \tag{18b}$$

$$\{\{\mathtt{Mj}_1(\alpha,\beta,\gamma), \mathtt{Mj}_2(\alpha,\beta,\gamma), \mathtt{Mn}_3(\alpha,\beta,\gamma)\}\} = \{\{\alpha,\beta,\gamma\}\} \qquad \forall \alpha,\beta,\gamma \in D \tag{18c}$$

6.3. Eliminating case (15a)

We will need the following result.

LEMMA 6.6. *Suppose that $i \in V$ is a node with $\{\{x_i, y_i, z_i\}\} = \{\{a, b, b\}\}$ where $\{a, b\} \in \overline{M}$. Let $\boldsymbol{w} \in \{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}\}$ be the labelling with $w_i = a$, and let $\boldsymbol{w}'$ be the labelling obtained from $\boldsymbol{w}$ by setting $w'_i = b$. Then $\boldsymbol{w}' \in \mathtt{dom} f$.*

PROOF. Assume that $\boldsymbol{w} = \boldsymbol{x}$ (the cases $\boldsymbol{w} = \boldsymbol{y}$ and $\boldsymbol{w} = \boldsymbol{z}$ will be entirely analogous). Accordingly, we denote $\boldsymbol{x}' = \boldsymbol{w}'$. By Assumption 2, f admits a majority polymorphism. This implies [Baker and Pixley 1975] that $\mathtt{dom} f$ is *decomposable into unary and binary relations*, i.e. there holds

$$\boldsymbol{u} \in \mathtt{dom} f \qquad \Leftrightarrow \qquad [u_i \in \rho_i \ \ \forall i \in V \ \text{ and } \ (u_i, u_j) \in \rho_{ij} \ \ \forall i, j \in V, i \neq j]$$

where unary relations $\rho_i \subseteq D$ for $i \in V$ and binary relations $\rho_{ij} \subseteq D \times D$ for distinct $i, j \in V$ are defined as

$$\rho_i = \{u_i \mid \boldsymbol{u} \in \mathtt{dom} f\} \qquad \rho_{ij} = \{(u_i, u_j) \mid \boldsymbol{u} \in \mathtt{dom} f\}$$

Suppose that $\boldsymbol{x}' \notin \mathtt{dom} f$, then there exists a node $j \in V - \{i\}$ such that $(x'_i, x'_j) = (b, x_j) \notin \rho_{ij}$. We must have $(a, x_j), (b, y_j), (b, z_j) \in \rho_{ij}$ since $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathtt{dom} f$. This implies, in particular, that $y_j \neq x_j$ and $z_j \neq x_j$. Furthermore, $(a, y_j), (a, z_j) \notin \rho_{ij}$, otherwise pair $(a, b) \in \overline{M}$ would have an incident soft edge in G_Γ. Two cases are possible:

— $y_j = z_j$. The edge $\{(a, b), (y_j, x_j)\}$ belongs to G_Γ, therefore $(x_j, y_j) \in \overline{M}$.
— $y_j \neq z_j$. We have $\rho_{ij} = \{(a, x_j), (b, y_j), (b, z_j)\}$, therefore $\mu(\{x_j, y_j, z_j\}) = \{x_j\}$.

In each case $\mathtt{Mj}_1(x_j, y_j, z_j) \neq x_j$, $\mathtt{Mj}_2(x_j, y_j, z_j) \neq x_j$ and $\mathtt{Mn}_3(x_j, y_j, z_j) = x_j$. Now let us "minimise out" variable x_i, i.e. define function

$$g(\boldsymbol{u}) = \min_{d \in D} f(d, \boldsymbol{u}) \qquad \forall \boldsymbol{u} \in D^{\hat{V}} \tag{19}$$

where $\hat{V} = V - \{i\}$ and we assumed that i corresponds to the first argument of f. For an assignment $\boldsymbol{u} \in V$ we denote by $\hat{\boldsymbol{u}}$ the restriction of $\boldsymbol{u}$ to $\hat{V}$. Due to the presence of relation ρ_{ij} we have

$$\begin{array}{lll} g(\hat{\boldsymbol{x}}) = f(\boldsymbol{x}) & & g(\mathtt{Mj}_1(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})) = f(\mathtt{Mj}_1(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})) \\ g(\hat{\boldsymbol{y}}) = f(\boldsymbol{y}) & & g(\mathtt{Mj}_2(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})) = f(\mathtt{Mj}_2(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})) \\ g(\hat{\boldsymbol{z}}) = f(\boldsymbol{z}) & & g(\mathtt{Mn}_3(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})) = f(\mathtt{Mn}_3(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})) \end{array}$$

Indeed, let us show, for example, that $g(\hat{\boldsymbol{x}}) = f(\boldsymbol{x})$. From (19) we get $g(\hat{\boldsymbol{x}}) = \min\{f(a, \hat{\boldsymbol{x}}), f(b, \hat{\boldsymbol{x}})\}$. We have $(b, \hat{\boldsymbol{x}}) \neq \mathtt{dom} f$ (since $(b, x_j) \notin \rho_{ij}$), and therefore $g(\hat{\boldsymbol{x}}) = f(a, \hat{\boldsymbol{x}}) = f(\boldsymbol{x})$. The 5 other equalities above are proved in a similar way.

Since $\delta(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}}) < \delta(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$, Assumption 4 gives

$$g(\mathtt{Mj}_1(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})) + g(\mathtt{Mj}_2(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})) + g(\mathtt{Mn}_3(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})) \leq f(\boldsymbol{x}) + f(\boldsymbol{y}) + f(\boldsymbol{z})$$

which is equivalent to (16). Thus, the instance $(f, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ is non-violating; this contradicts Assumption 4. □

Let us denote

$$\begin{aligned} V^M &= \{i \in V \mid \{x_i, y_i, z_i\} = \{a, b\} \in M\} \\ V^{\overline{M}} &= \{i \in V \mid \{x_i, y_i, z_i\} = \{a, b\} \in \overline{M}\} \\ V_1^{\overline{M}} &= \{i \in V^{\overline{M}} \mid (x_i, y_i, z_i) = (a, b, b)\} \subseteq \Delta(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \\ V_2^{\overline{M}} &= \{i \in V^{\overline{M}} \mid (x_i, y_i, z_i) = (b, a, b)\} \subseteq \Delta(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \\ V_3^{\overline{M}} &= \{i \in V^{\overline{M}} \mid (x_i, y_i, z_i) = (b, b, a)\} \end{aligned}$$

The goal of this subsection is to prove that set $V^{\overline{M}} = V_1^{\overline{M}} \cup V_2^{\overline{M}} \cup V_3^{\overline{M}}$ is empty (and so the case (15a) is never used).

PROPOSITION 6.7. *Suppose that* $i \in V^{\overline{M}}$.

(a) If $(x_i, y_i, z_i) = (a, b, b)$ *then* $\Delta(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = \{i\}$ *and consequently* $V_1^{\overline{M}} = \{i\}$, $\Delta^M(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = \varnothing$.
(b) If $(x_i, y_i, z_i) = (b, a, b)$ *then* $\Delta(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = \{i\}$ *and consequently* $V_2^{\overline{M}} = \{i\}$, $\Delta^M(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = \varnothing$.
(c) If $(x_i, y_i, z_i) = (b, b, a)$ *then* $V_3^{\overline{M}} = \{i\}$, $|\{x_j, y_j, z_j\}| \le 2$ *for all* $j \in V$ *and* $\Delta^M(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = \varnothing$.

PROOF. Note, in all three cases $\{a, b\} \in \overline{M}$.

Part (a) Suppose that $(x_i, y_i, z_i) = (a, b, b)$ and $\Delta(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ is a strict superset of $\{i\}$. Let us define $\boldsymbol{u} = \mathtt{Mn}_3(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$. It can be checked that $\mathtt{Mj}_1(\boldsymbol{x}, \boldsymbol{x}, \boldsymbol{u}) = \mathtt{Mj}_2(\boldsymbol{x}, \boldsymbol{x}, \boldsymbol{u}) = \boldsymbol{x}$ and $\mathtt{Mn}_3(\boldsymbol{x}, \boldsymbol{x}, \boldsymbol{u}) = \boldsymbol{u}$. Therefore, if we define $\boldsymbol{x}' = \boldsymbol{x}$ and $\boldsymbol{u}' = \boldsymbol{u}$ then the following identities will hold:

$$\begin{array}{lcllcl} \mathtt{Mj}_1(\boldsymbol{x}', \boldsymbol{y}, \boldsymbol{z}) & = & \mathtt{Mj}_1(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) & \mathtt{Mj}_1(\boldsymbol{x}, \boldsymbol{x}', \boldsymbol{u}') & = & \boldsymbol{x}' \\ \mathtt{Mj}_2(\boldsymbol{x}', \boldsymbol{y}, \boldsymbol{z}) & = & \mathtt{Mj}_2(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) & \mathtt{Mj}_2(\boldsymbol{x}, \boldsymbol{x}', \boldsymbol{u}') & = & \boldsymbol{x}' \\ \mathtt{Mn}_3(\boldsymbol{x}', \boldsymbol{y}, \boldsymbol{z}) & = & \boldsymbol{u}' & \mathtt{Mn}_3(\boldsymbol{x}, \boldsymbol{x}', \boldsymbol{u}') & = & \mathtt{Mn}_3(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \end{array}$$

Let us modify $\boldsymbol{x}'$ and $\boldsymbol{u}'$ by setting $x'_i = u'_i = b$. Using the definition of the MJN operation, it can be checked that the identities above still hold. By Lemma 6.6, $\boldsymbol{x}' \in \mathtt{dom} f$. We also have $\delta(\boldsymbol{x}', \boldsymbol{y}, \boldsymbol{z}) < \delta(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$, so Assumption 4 gives

$$f(\mathtt{Mj}_1(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})) + f(\mathtt{Mj}_2(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})) + f(\boldsymbol{u}') \;\le\; f(\boldsymbol{x}') + f(\boldsymbol{y}) + f(\boldsymbol{z}) \tag{20}$$

This implies, in particular, that $\boldsymbol{u}' \in \mathtt{dom} f$. We have $(\boldsymbol{x}, \boldsymbol{x}', \boldsymbol{u}') \prec (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ since $\delta(\boldsymbol{x}, \boldsymbol{x}', \boldsymbol{u}') \le \delta(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$, $\Delta(\boldsymbol{x}, \boldsymbol{x}', \boldsymbol{u}') = \{i\}$ and we assumed that $\Delta(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ is a strict superset of $\{i\}$. Therefore, Assumption 4 gives

$$f(\boldsymbol{x}') + f(\boldsymbol{x}') + f(\mathtt{Mn}_3(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})) \;\le\; f(\boldsymbol{x}) + f(\boldsymbol{x}') + f(\boldsymbol{u}') \tag{21}$$

Summing (20) and (21) gives (16).

Part (b) Suppose that $(x_i, y_i, z_i) = (b, a, b)$ and $\Delta(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ is a strict superset of $\{i\}$. Let $\boldsymbol{u} = \mathtt{Mn}_3(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$. If we define $\boldsymbol{y}' = \boldsymbol{y}$ and $\boldsymbol{u}' = \boldsymbol{u}$ then the following identities will hold:

$$\begin{array}{lcllcl} \mathtt{Mj}_1(\boldsymbol{x}, \boldsymbol{y}', \boldsymbol{z}) & = & \mathtt{Mj}_1(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) & \mathtt{Mj}_1(\boldsymbol{y}, \boldsymbol{y}', \boldsymbol{u}') & = & \boldsymbol{y}' \\ \mathtt{Mj}_2(\boldsymbol{x}, \boldsymbol{y}', \boldsymbol{z}) & = & \mathtt{Mj}_2(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) & \mathtt{Mj}_2(\boldsymbol{y}, \boldsymbol{y}', \boldsymbol{u}') & = & \boldsymbol{y}' \\ \mathtt{Mn}_3(\boldsymbol{x}, \boldsymbol{y}', \boldsymbol{z}) & = & \boldsymbol{u}' & \mathtt{Mn}_3(\boldsymbol{y}, \boldsymbol{y}', \boldsymbol{u}') & = & \mathtt{Mn}_3(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \end{array}$$

Let us modify $\boldsymbol{y}'$ and $\boldsymbol{u}'$ by setting $y'_i = u'_i = b$. It can be checked that the identities above still hold. The rest of the proof is analogous to the proof for part (a).

Part (c) Suppose that $(x_i, y_i, z_i) = (b, b, a)$ and (c) does not hold. Let $\boldsymbol{u} = \mathtt{Mn}_3(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$. If we define $\boldsymbol{z}' = \boldsymbol{z}$ and $\boldsymbol{u}' = \boldsymbol{u}$ then the following identities will hold:

$$\begin{array}{lcllcl} \mathtt{Mj}_1(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}') & = & \mathtt{Mj}_1(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) & \mathtt{Mj}_1(\boldsymbol{z}, \boldsymbol{z}', \boldsymbol{u}') & = & \boldsymbol{z}' \\ \mathtt{Mj}_2(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}') & = & \mathtt{Mj}_2(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) & \mathtt{Mj}_2(\boldsymbol{z}, \boldsymbol{z}', \boldsymbol{u}') & = & \boldsymbol{z}' \\ \mathtt{Mn}_3(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}') & = & \boldsymbol{u}' & \mathtt{Mn}_3(\boldsymbol{z}, \boldsymbol{z}', \boldsymbol{u}') & = & \mathtt{Mn}_3(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \end{array}$$

Let us modify $\boldsymbol{z}'$ and $\boldsymbol{u}'$ by setting $z'_i = u'_i = b$. It can be checked that the identities above still hold.

We claim that $(*)$ $(\boldsymbol{z}, \boldsymbol{z}', \boldsymbol{u}') \prec (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$. Indeed, since (c) does not hold we must have one of the following:

— $V_3^{\overline{M}}$ contains another node j besides i. Then $(*)$ holds since $|\{z_j, z'_j, u'_j\}| = 1 < |\{x_j, y_j, z_j\}| = 2$.

— $|\{x_j, y_j, z_j)| = 3$ for some $j \in V$. Then $(*)$ holds since $|\{z_j, z'_j, u'_j\}| \leq 2$.
— $|\Delta^M(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})| \geq 1$. Then $(*)$ holds since $|\Delta(\boldsymbol{z}, \boldsymbol{z}', \boldsymbol{u}')| = 1 \leq |\Delta^M(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})| \leq |\Delta(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})|$ and $|\Delta^M(\boldsymbol{z}, \boldsymbol{z}', \boldsymbol{u}')| = 0$.

The rest of the proof is analogous to the proof for part (a). □

Next, we show that if $V^{\overline{M}}$ is non-empty then V^M is empty. To prove this, assume that $V^{\overline{M}} \neq \varnothing$, then by Proposition 6.7 we know that $\Delta^M(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ is empty. Thus, if $i \in V^M$ then we must have $(x_i, y_i, z_i) = (b, b, a)$. This case is eliminated by the following proposition.

PROPOSITION 6.8. *For a node $i \in V$ the following situations are impossible:*

S1 $(x_i, y_i, z_i) = (b, b, a)$, $(a, b) \in M$, $a \sqcup b = b$.
S2 $(x_i, y_i, z_i) = (b, b, a)$, $(a, b) \in M$, $a \sqcap b = b$.

PROOF.

Case S1 Let us define $\boldsymbol{u} = \mathtt{Mn}_3(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$. By inspecting each case (15a)-(15d) and using equations (18) one can check that $\boldsymbol{u} \sqcup \boldsymbol{z} = \boldsymbol{z}$ and consequently $\boldsymbol{u} \sqcap \boldsymbol{z} = \boldsymbol{u}$. Therefore, if we define $\boldsymbol{z}' = \boldsymbol{z}$ and $\boldsymbol{u}' = \boldsymbol{u}$ then the following identities will hold:

$$\begin{array}{lcllcl} \mathtt{Mj}_1(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}') & = & \mathtt{Mj}_1(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) & \boldsymbol{u}' \sqcap \boldsymbol{z} & = & \mathtt{Mn}_3(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \\ \mathtt{Mj}_2(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}') & = & \mathtt{Mj}_2(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) & \boldsymbol{u}' \sqcup \boldsymbol{z} & = & \boldsymbol{z}' \\ \mathtt{Mn}_3(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}') & = & \boldsymbol{u}' & & & \end{array}$$

Let us modify $\boldsymbol{z}'$ and $\boldsymbol{u}'$ by setting $z'_i = u'_i = b$, so that we have

— $a = z_i = u_i \;\; = \mathtt{Mn}_3(x_i, y_i, z_i)$
— $b = z'_i = u'_i \;\; = \mathtt{Mj}_{1,2}(x_i, y_i, z_i) \quad (= x_i = y_i)$

$\{a, b\} \in M$, $a \sqcup b = b$

It can be checked that the identities above still hold. We have $\delta(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}') < \delta(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$, so Assumption 4 gives

$$f(\mathtt{Mj}_1(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})) + f(\mathtt{Mj}_2(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})) + f(\boldsymbol{u}') \;\leq\; f(\boldsymbol{x}) + f(\boldsymbol{y}) + f(\boldsymbol{z}') \tag{22}$$

assuming that $\boldsymbol{z}' \in \mathtt{dom} f$, and the fact that $\langle \sqcap, \sqcup \rangle$ is a multimorphism of f gives

$$f(\mathtt{Mn}_3(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})) + f(\boldsymbol{z}') \;\leq\; f(\boldsymbol{u}') + f(\boldsymbol{z}) \tag{23}$$

assuming that $\boldsymbol{u}' \in \mathtt{dom} f$. If $\boldsymbol{z}' \in \mathtt{dom} f$ then (22) implies that $\boldsymbol{u}' \in \mathtt{dom} f$; summing (22) and (23) gives (16). We thus assume that $\boldsymbol{z}' \notin \mathtt{dom} f$, then (23) implies that $\boldsymbol{u}' \notin \mathtt{dom} f$.

Let C be a sufficiently large constant, namely $C > f(\boldsymbol{x}) + f(\boldsymbol{y}) + f(\boldsymbol{z})$. Consider function

$$g(\boldsymbol{u}) = \min_{d \in D} \{[d = a] \cdot C + f(d, \boldsymbol{u})\} \qquad \forall \boldsymbol{u} \in D^{\hat{V}} \tag{24}$$

where $\hat{V} = V - \{i\}$, $[\cdot]$ is the Iverson bracket (it is 1 if its argument is true, and 0 otherwise) and we assumed for simplicity of notation that i corresponds to the first argument of f. For an assignment $\boldsymbol{w} \in V$ we denote by $\hat{\boldsymbol{w}}$ the restriction of $\boldsymbol{w}$ to $\hat{V}$. We can write

$$g(\hat{\boldsymbol{z}}) = f(\boldsymbol{z}) + C \qquad g(\hat{\boldsymbol{x}}) = f(\boldsymbol{x}) \qquad g(\hat{\boldsymbol{y}}) = f(\boldsymbol{y}) \qquad g(\hat{\boldsymbol{u}}) = f(\boldsymbol{u}) + C$$

where the first equation holds since $(b, \hat{\boldsymbol{z}}) = \boldsymbol{z}' \notin \mathtt{dom} f$ and the last equation holds since $(b, \hat{\boldsymbol{u}}) = \boldsymbol{u}' \notin \mathtt{dom} f$. Assumption 4 gives

$$\begin{array}{r} g(\mathtt{Mj}_1(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})) + g(\mathtt{Mj}_2(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})) + g(\mathtt{Mn}_3(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})) \;\leq\; g(\hat{\boldsymbol{x}}) + g(\hat{\boldsymbol{y}}) + g(\hat{\boldsymbol{z}}) \\ g(\mathtt{Mj}_1(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})) + g(\mathtt{Mn}_3(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})) + [(f(\boldsymbol{u}) + C] \;\leq\; f(\boldsymbol{x}) + f(\boldsymbol{y}) + [f(\boldsymbol{z}) + C] \end{array}$$

Therefore, $g(\mathtt{Mj}_1(\hat{\boldsymbol{x}},\hat{\boldsymbol{y}},\hat{\boldsymbol{z}})) < C$, and thus $g(\mathtt{Mj}_1(\hat{\boldsymbol{x}},\hat{\boldsymbol{y}},\hat{\boldsymbol{z}})) = f(b,\mathtt{Mj}_1(\hat{\boldsymbol{x}},\hat{\boldsymbol{y}},\hat{\boldsymbol{z}})) = f(\mathtt{Mj}_1(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}))$. Similarly, $g(\mathtt{Mj}_2(\hat{\boldsymbol{x}},\hat{\boldsymbol{y}},\hat{\boldsymbol{z}})) = f(b,\mathtt{Mj}_2(\hat{\boldsymbol{x}},\hat{\boldsymbol{y}},\hat{\boldsymbol{z}})) = f(\mathtt{Mj}_2(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}))$, and hence the inequality above is equivalent to (16).

Case S2 Let us define $\boldsymbol{u} = \mathtt{Mn}_3(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$. It can be checked that $\boldsymbol{z} \sqcap \boldsymbol{u} = \boldsymbol{z}$ and consequently $\boldsymbol{z} \sqcup \boldsymbol{u} = \boldsymbol{u}$. Therefore, if we define $\boldsymbol{z}' = \boldsymbol{z}$ and $\boldsymbol{u}' = \boldsymbol{u}$ then the following identities will hold:

$$\begin{array}{lcllcl} \mathtt{Mj}_1(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}') & = & \mathtt{Mj}_1(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) & \boldsymbol{z} \sqcap \boldsymbol{u}' & = & \boldsymbol{z}' \\ \mathtt{Mj}_2(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}') & = & \mathtt{Mj}_2(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) & \boldsymbol{z} \sqcup \boldsymbol{u}' & = & \mathtt{Mn}_3(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) \\ \mathtt{Mn}_3(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}') & = & \boldsymbol{u}' & & & \end{array}$$

Let us modify $\boldsymbol{z}'$ and $\boldsymbol{u}'$ by setting $z'_i = u'_i = b$, so that we have

— $a = z_i = u_i \;\; = \mathtt{Mn}_3(x_i,y_i,z_i)$
— $b = z'_i = u'_i \;\; = \mathtt{Mj}_{1,2}(x_i,y_i,z_i) \quad (= x_i = y_i)$

$\{a,b\} \in M$, $a \sqcap b = b$

It can be checked that the identities above still hold. The rest of the proof proceeds analogously to the proof for the case S1. □

We are now ready to prove the main result of this subsection.

PROPOSITION 6.9. *Set $V^{\overline{M}}$ is empty.*

PROOF. Suppose that $V^{\overline{M}} \neq \varnothing$. As we just showed, we must have $V^M = \varnothing$. For each $i \in V$ we also have $|\{x_i,y_i,z_i\}| \neq 1$ by Proposition 6.4 and $|\{x_i,y_i,z_i\}| \neq 3$ by Proposition 6.7. Therefore, $V = V^{\overline{M}}$. Proposition 6.7 implies that each of the sets $V_1^{\overline{M}}$, $V_2^{\overline{M}}$, $V_3^{\overline{M}}$ contains at most one node, and furthermore $|V_1^{\overline{M}} \cup V_2^{\overline{M}}| \leq 1$. Since $|V| \geq 2$ by Proposition 6.4, we conclude that $V = \{i,j\}$ where $i \in V_3^{\overline{M}}$ and $j \in V_1^{\overline{M}} \cup V_2^{\overline{M}}$.

Suppose that $j \in V_1^{\overline{M}}$, then we have $\boldsymbol{x} = (b,a')$, $\boldsymbol{y} = (b,b')$, $\boldsymbol{z} = (a,b')$ where $\{a,b\},\{a',b'\} \in \overline{M}$. Inequality (16) reduces to

$$f(b,b') + f(b,b') + f(a,a') \leq f(b,a') + f(b,b') + f(a,b') \tag{25}$$

We claim that $f(a,a')+f(b,b') = f(a,b')+f(b,a')$. Indeed, if $f(a,a')+f(b,b') > f(a,b')+f(b,a')$ then $\{(a,b),(a',b')\}$ is a soft edge in G_Γ, and if $f(a,a')+f(b,b') < f(a,b')+f(b,a')$ then $\{(a,b),(b',a')\}$ is a soft edge in G_Γ; both case cases contradict Lemma 5.1(g). We thus showed that (25) is an equality, and so instance $(f,\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$ is non-violating; this contradicts Assumption 4. The case $j \in V_2^{\overline{M}}$ is completely analogous. Proposition 6.9 is proved. □

6.4. Eliminating cases (15b) and (15c)

Propositions 6.5 and 6.9 show that there must exist node $i \in V$ with $\mu(\{x_i,y_i,z_i\}) = \{x_i\}$ or $\mu(\{x_i,y_i,z_i\}) = \{y_i\}$. In this section we show that this leads to a contradiction, thus proving Theorem 6.3.

Consider a variable $i \in V$ with $\mu(\{x_i,y_i,z_i\}) \neq \varnothing$. Let us define a transformation that produces a new instance $(\bar{f},\bar{\boldsymbol{x}},\bar{\boldsymbol{y}},\bar{\boldsymbol{z}})$ from $(f,\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$; this transformation will be called the *i-expansion of* $(f,\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$. The set of variables of the new instance will be $\bar{V} = V \cup \{j\}$ where $j \notin V$. The cost function will be

$$\bar{f}(\boldsymbol{u}) = f(\hat{\boldsymbol{u}}) + g(u_i,u_j) \qquad \forall \boldsymbol{u} \in D^{\bar{V}}$$

where g is a binary function taken from the definition of the set $\mu(\{x_i,y_i,z_i\})$ and $\hat{\boldsymbol{u}}$ is the restriction of $\boldsymbol{u}$ to V. Finally, labellings $\bar{\boldsymbol{x}},\bar{\boldsymbol{y}},\bar{\boldsymbol{z}}$ are obtained by extending $\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}$ to $\bar{V}$ in the unique way so that $(\bar{f},\bar{\boldsymbol{x}},\bar{\boldsymbol{y}},\bar{\boldsymbol{z}})$ is a valid instance.

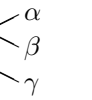
It is easy to check that if $(f, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ is non-violating then so is $(\bar{f}, \bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}, \bar{\boldsymbol{z}})$. In the proofs below we will use the following approach: after constructing the i-expansion, we will add some unary function for node i and then "minimise out" variable x_i, obtaining another instance $(g, \hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})$ with $\delta(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}}) = (\delta(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) + 2) - 3 = \delta_{\min} - 1$. We will then invoke Assumption 4 for the new instance obtaining a contradiction.

For simplicity, we will not change the notation when applying the i-expansion operation to $(f, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$, i.e. the new instance $(\bar{f}, \bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}, \bar{\boldsymbol{z}})$ will be denoted as $(f, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$. Variable j introduced by the i-expansion will be called the *control variable* for i. It is easy to check the following.

PROPOSITION 6.10. *Let j be a control variable for $i \in V$ with $\mu(\{x_i, y_i, z_i\}) \neq \varnothing$ in an instance $(f, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$. Let $\boldsymbol{u}$, $\boldsymbol{v}$, $\boldsymbol{w}$ be a permutation of $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$ such that $\mu(\{x_i, y_i, z_i\}) = \{u_i\}$. Then*

—*Any labelling obtained from one of the labellings in $\{\boldsymbol{u}, \mathtt{Mn}_3(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})\}$ by changing the label of i from u_i to v_i or w_i does not belong to $\mathtt{dom} f$.*
—*Any labelling obtained from one of the labellings in $\{\boldsymbol{v}, \boldsymbol{w}, \mathtt{Mj}_1(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}), \mathtt{Mj}_2(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})\}$ by changing the label of i from $\{v_i, w_i\}$ to u_i does not belong to $\mathtt{dom} f$.*

Recall that the following diagram illustrates the fact that $\mu(\{\alpha, \beta, \gamma\}) = \{\gamma\}$:



PROPOSITION 6.11. *For a node $i \in V$ the following situations are impossible:*

T1 $\mu(\{x_i, y_i, z_i\}) = \{y_i\}$, $(x_i, z_i) \in M$, $x_i \sqcap z_i = z_i$.
T2 $\mu(\{x_i, y_i, z_i\}) = \{y_i\}$, $(x_i, z_i) \in M$, $x_i \sqcup z_i = z_i$.
T3 $\mu(\{x_i, y_i, z_i\}) = \{x_i\}$, $(y_i, z_i) \in M$, $y_i \sqcup z_i = z_i$.
T4 $\mu(\{x_i, y_i, z_i\}) = \{x_i\}$, $(y_i, z_i) \in M$, $y_i \sqcap z_i = z_i$.

PROOF. We will analyse cases T1-T4 separately, and will derive a contradiction in each case.

Case T1 Let us define $\boldsymbol{u} = \mathtt{Mj}_2(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$. It can be checked that $\boldsymbol{x} \sqcap \boldsymbol{u} = \boldsymbol{x}$ and consequently $\boldsymbol{x} \sqcup \boldsymbol{u} = \boldsymbol{u}$. Therefore, if we define $\boldsymbol{x}' = \boldsymbol{x}$ and $\boldsymbol{u}' = \boldsymbol{u}$ then the following identities will hold:

$$\begin{array}{lcllcll} \mathtt{Mj}_1(\boldsymbol{x}', \boldsymbol{y}, \boldsymbol{z}) & = & \mathtt{Mj}_1(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) & \boldsymbol{x} \sqcap \boldsymbol{u}' & = & \boldsymbol{x}' & \\ \mathtt{Mj}_2(\boldsymbol{x}', \boldsymbol{y}, \boldsymbol{z}) & = & \boldsymbol{u}' & \boldsymbol{x} \sqcup \boldsymbol{u}' & = & \mathtt{Mj}_2(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) & = \boldsymbol{u} \\ \mathtt{Mn}_3(\boldsymbol{x}', \boldsymbol{y}, \boldsymbol{z}) & = & \mathtt{Mn}_3(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) & & & & \end{array} \tag{26}$$

Let us modify $\boldsymbol{x}', \boldsymbol{u}'$ by setting $x'_i = u'_i = \mathtt{Mj}_1(x_i, y_i, z_i)$ so that we have

$$\begin{array}{ll} a = x_i = u_i & = \mathtt{Mj}_2(x_i, y_i, z_i) \\ c = x'_i = u'_i & = \mathtt{Mj}_1(x_i, y_i, z_i) \ \ (= z_i) \\ b & = \mathtt{Mn}_3(x_i, y_i, z_i) \ \ (= y_i) \end{array} \qquad \{a, c\} \in M,\ a \sqcap c = c$$

where we denoted $(a, b, c) = (x_i, y_i, z_i)$. It can be checked that identities (26) still hold, and furthermore $\delta(\boldsymbol{x}', \boldsymbol{y}, \boldsymbol{z}) < \delta(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$. Assumption 4 gives

$$f(\mathtt{Mj}_1(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})) + f(\boldsymbol{u}') + f(\mathtt{Mn}_3(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})) \ \leq \ f(\boldsymbol{x}') + f(\boldsymbol{y}) + f(\boldsymbol{z}) \tag{27}$$

assuming that $\boldsymbol{x}' \in \mathtt{dom} f$, and the fact that $\langle \sqcap, \sqcup \rangle$ is a multimorphism of f gives

$$f(\boldsymbol{x}') + f(\mathtt{Mj}_2(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})) \ \leq \ f(\boldsymbol{x}) + f(\boldsymbol{u}') \tag{28}$$

assuming that $\boldsymbol{u}' \in \mathtt{dom} f$. If $\boldsymbol{x}' \in \mathtt{dom} f$ then (27) implies that $\boldsymbol{u}' \in \mathtt{dom} f$; summing (27) and (28) gives (16). We thus assume that $\boldsymbol{x}' \notin \mathtt{dom} f$, then (28) implies that $\boldsymbol{u}' \notin \mathtt{dom} f$.

Let us apply the i-expansion transformation to instance $(f, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$. For simplicity, we do not change the notation, so we assume that V now contains a control variable for i and $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{u}, \boldsymbol{x}', \boldsymbol{u}'$ have been extended to the new set accordingly. We have $\delta(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = \delta_{\min} + 2$.

Let C be a sufficiently large constant, namely $C > f(\boldsymbol{x}) + f(\boldsymbol{y}) + f(\boldsymbol{z})$. Consider function

$$g(\boldsymbol{w}) = \min_{d \in D} \{[d = a] \cdot C + f(d, \boldsymbol{w})\} \qquad \forall \boldsymbol{w} \in D^{\hat{V}} \tag{29}$$

where $\hat{V} = V - \{i\}$, $[\cdot]$ is the Iverson bracket (it returns 1 if its argument is true and 0 otherwise) and we assumed for simplicity of notation that i corresponds to the first argument of f. For an assignment $\boldsymbol{w} \in V$ we denote by $\hat{\boldsymbol{w}}$ the restriction of $\boldsymbol{w}$ to $\hat{V}$. We can write

$$g(\hat{\boldsymbol{x}}) = f(\boldsymbol{x}) + C \qquad g(\hat{\boldsymbol{y}}) = f(\boldsymbol{y}) \qquad g(\hat{\boldsymbol{z}}) = f(\boldsymbol{z}) \qquad g(\hat{\boldsymbol{u}}) = f(\boldsymbol{u}) + C \tag{30}$$

To show the first equation, observe that the minimum in (29) cannot be achieved at $d = c$ since $(c, \hat{\boldsymbol{x}}) = \boldsymbol{x}' \notin \mathtt{dom} f$, and also the minimum cannot be achieved at $d = b$ by Proposition 6.10. Therefore, $g(\hat{\boldsymbol{x}}) = g(a, \hat{\boldsymbol{x}}) = f(\boldsymbol{x}) + C$. Other equations can be derived similarly.

Clearly, $(g, \hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})$ is a valid instance and $\delta(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}}) = \delta_{\min} - 1$, so Assumption 4 gives

$$\begin{aligned} g(\mathtt{Mj}_1(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})) + g(\mathtt{Mj}_2(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})) + g(\mathtt{Mn}_3(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})) &\leq g(\hat{\boldsymbol{x}}) + g(\hat{\boldsymbol{y}}) + g(\hat{\boldsymbol{z}}) \\ g(\mathtt{Mj}_1(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})) + [f(\boldsymbol{u}) + C] + g(\mathtt{Mn}_3(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})) &\leq [f(\boldsymbol{x}) + C] + f(\boldsymbol{y}) + f(\boldsymbol{z}) \end{aligned}$$

Therefore, $g(\mathtt{Mj}_1(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})) < C$, and thus $g(\mathtt{Mj}_1(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})) = f(c, \mathtt{Mj}_1(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})) = f(\mathtt{Mj}_1(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}))$. (Note, labelling $(b, \mathtt{Mj}_1(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}}))$ is not in $\mathtt{dom} f$ by Proposition 6.10.) Similarly, $g(\mathtt{Mn}_3(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})) = f(b, \mathtt{Mn}_3(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})) = f(\mathtt{Mn}_3(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}))$, and hence the inequality above is equivalent to (16).

Case T2 Let us define $\boldsymbol{u} = \mathtt{Mj}_1(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$. It can be checked that $\boldsymbol{u} \sqcup \boldsymbol{x} = \boldsymbol{x}$ and consequently $\boldsymbol{u} \sqcap \boldsymbol{x} = \boldsymbol{u}$. Therefore, if we define $\boldsymbol{x}' = \boldsymbol{x}$ and $\boldsymbol{u}' = \boldsymbol{u}$ then the following identities will hold:

$$\begin{array}{lcllcll} \mathtt{Mj}_1(\boldsymbol{x}', \boldsymbol{y}, \boldsymbol{z}) & = & \boldsymbol{u}' & & \boldsymbol{u}' \sqcap \boldsymbol{x} & = & \mathtt{Mj}_1(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = \boldsymbol{u} \\ \mathtt{Mj}_2(\boldsymbol{x}', \boldsymbol{y}, \boldsymbol{z}) & = & \mathtt{Mj}_2(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) & & \boldsymbol{u}' \sqcup \boldsymbol{x} & = & \boldsymbol{x}' \\ \mathtt{Mn}_3(\boldsymbol{x}', \boldsymbol{y}, \boldsymbol{z}) & = & \mathtt{Mn}_3(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) & & & & \end{array}$$

Let us modify $\boldsymbol{x}', \boldsymbol{u}'$ by setting $x'_i = u'_i = \mathtt{Mj}_2(x_i, y_i, z_i)$ so that we have

$$\begin{array}{ll} a = x_i = u_i & = \mathtt{Mj}_1(x_i, y_i, z_i) \\ c = x'_i = u'_i & = \mathtt{Mj}_2(x_i, y_i, z_i) \ \ (= z_i) \\ b & = \mathtt{Mn}_3(x_i, y_i, z_i) \ \ (= y_i) \end{array} \qquad \{a, c\} \in M,\ a \sqcup c = c$$

It can be checked that the identities above still hold. The rest of the proof proceeds analogously to the proof for the case T1.

Case T3 Let us define $\boldsymbol{u} = \mathtt{Mj}_1(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$. It can be checked that $\boldsymbol{u} \sqcup \boldsymbol{y} = \boldsymbol{y}$ and consequently $\boldsymbol{u} \sqcap \boldsymbol{y} = \boldsymbol{u}$. Therefore, if we define $\boldsymbol{y}' = \boldsymbol{y}$ and $\boldsymbol{u}' = \boldsymbol{u}$ then the following identities will hold:

$$\begin{array}{lcllcll} \mathtt{Mj}_1(\boldsymbol{x}, \boldsymbol{y}', \boldsymbol{z}) & = & \boldsymbol{u}' & & \boldsymbol{u}' \sqcap \boldsymbol{y} & = & \mathtt{Mj}_1(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = \boldsymbol{u} \\ \mathtt{Mj}_2(\boldsymbol{x}, \boldsymbol{y}', \boldsymbol{z}) & = & \mathtt{Mj}_2(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) & & \boldsymbol{u}' \sqcup \boldsymbol{y} & = & \boldsymbol{y}' \\ \mathtt{Mn}_3(\boldsymbol{x}, \boldsymbol{y}', \boldsymbol{z}) & = & \mathtt{Mn}_3(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) & & & & \end{array}$$

Let us modify $\boldsymbol{y}', \boldsymbol{u}'$ by setting $y'_i = u'_i = \mathtt{Mj}_2(x_i, y_i, z_i)$ so that we have

$$\begin{array}{ll} b = y_i = u_i & = \mathtt{Mj}_1(x_i, y_i, z_i) \\ c = y'_i = u'_i & = \mathtt{Mj}_2(x_i, y_i, z_i) \ (= z_i) \\ a & = \mathtt{Mn}_3(x_i, y_i, z_i) \ (= x_i) \end{array} \qquad \{b, c\} \in M, b \sqcup c = c$$

It can be checked that the identities above still hold. The rest of the proof proceeds analogously to the proof for the case T1.

Case T4 Let us define $\boldsymbol{u} = \mathtt{Mj}_2(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$. It can be checked that $\boldsymbol{y} \sqcap \boldsymbol{u} = \boldsymbol{y}$ and consequently $\boldsymbol{y} \sqcup \boldsymbol{u} = \boldsymbol{u}$. Therefore, if we define $\boldsymbol{y}' = \boldsymbol{y}$ and $\boldsymbol{u}' = \boldsymbol{u}$ then the following identities will hold:

$$\begin{array}{lcl} \mathtt{Mj}_1(\boldsymbol{x}, \boldsymbol{y}', \boldsymbol{z}) & = & \mathtt{Mj}_1(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \\ \mathtt{Mj}_2(\boldsymbol{x}, \boldsymbol{y}', \boldsymbol{z}) & = & \boldsymbol{u}' \\ \mathtt{Mn}_3(\boldsymbol{x}, \boldsymbol{y}', \boldsymbol{z}) & = & \mathtt{Mn}_3(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \end{array} \qquad \begin{array}{lcl} \boldsymbol{y} \sqcap \boldsymbol{u}' & = & \boldsymbol{y}' \\ \boldsymbol{y} \sqcup \boldsymbol{u}' & = & \mathtt{Mj}_2(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \ = \boldsymbol{u} \end{array}$$

Let us modify $\boldsymbol{y}', \boldsymbol{u}'$ by setting $y'_i = u'_i = \mathtt{Mj}_1(x_i, y_i, z_i)$ so that we have

$$\begin{array}{ll} b = y_i = u_i & = \mathtt{Mj}_2(x_i, y_i, z_i) \\ c = y'_i = u'_i & = \mathtt{Mj}_1(x_i, y_i, z_i) \ (= z_i) \\ a & = \mathtt{Mn}_3(x_i, y_i, z_i) \ (= x_i) \end{array} \qquad \{b, c\} \in M, b \sqcap c = c$$

It can be checked that the identities above still hold. The rest of the proof proceeds analogously to the proof for the case T1. □

There are two possible cases remaining: $\mu(\{x_i, y_i, z_i\}) = \{y_i\}$, $\{x_i, z_i\} \in \overline{M}$ or $\mu(\{x_i, y_i, z_i\}) = \{x_i\}$, $\{y_i, z_i\} \in \overline{M}$. They are eliminated by the next two propositions; we use a slightly different argument.

PROPOSITION 6.12. *For a node $i \in V$ the following situation is impossible:*

T5 $\mu(\{x_i, y_i, z_i\}) = \{y_i\}$, $\{x_i, z_i\} \in \overline{M}$.

PROOF. For a labelling $\boldsymbol{w} \in D^V$ let $\hat{\boldsymbol{w}}$ be the restriction of $\boldsymbol{w}$ to $V - \{i\}$. Two cases are possible.

Case 1 $(\mathtt{Mj}_2(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}}), \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}}) \prec (\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})$. Let us define $\boldsymbol{u} = \mathtt{Mj}_2(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ and $\boldsymbol{v} = \mathtt{Mj}_2(\boldsymbol{u}, \boldsymbol{y}, \boldsymbol{z})$. It can be checked that $\mathtt{MJN}(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{z}) = (\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{z})$. [2] Therefore, if we define $\boldsymbol{z}' = \boldsymbol{z}$ and $\boldsymbol{u}' = \boldsymbol{u}$ then the following identities will hold:

$$\begin{array}{lcl} \mathtt{Mj}_1(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}') & = & \mathtt{Mj}_1(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \\ \mathtt{Mj}_2(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}') & = & \boldsymbol{u}' \\ \mathtt{Mn}_3(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}') & = & \mathtt{Mn}_3(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \end{array} \qquad \begin{array}{lcl} \mathtt{Mj}_1(\boldsymbol{u}', \boldsymbol{v}, \boldsymbol{z}) & = & \mathtt{Mj}_2(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = \boldsymbol{u} \\ \mathtt{Mj}_2(\boldsymbol{u}', \boldsymbol{v}, \boldsymbol{z}) & = & \boldsymbol{v} \\ \mathtt{Mn}_3(\boldsymbol{u}', \boldsymbol{v}, \boldsymbol{z}) & = & \boldsymbol{z}' \end{array} \qquad \boldsymbol{v} = \mathtt{Mj}_2(\boldsymbol{u}', \boldsymbol{y}, \boldsymbol{z})$$

Let us modify $\boldsymbol{z}'$ and $\boldsymbol{u}'$ according to the following diagram:

$$\begin{array}{ll} c = z_i = u_i & = \mathtt{Mj}_2(x_i, y_i, z_i) \ (= v_i) \\ a = z'_i = u'_i & = \mathtt{Mj}_1(x_i, y_i, z_i) \ (= x_i) \\ b & = \mathtt{Mn}_3(x_i, y_i, z_i) \ (= y_i) \end{array} \qquad \{a, c\} \in \overline{M}$$

It can be checked that the identities above still hold. The assumption of Case 1 gives $(\boldsymbol{u}', \boldsymbol{y}, \boldsymbol{z}) \prec (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ (note that $u'_i = x_i$). Therefore, the fact that $\boldsymbol{v} = \mathtt{Mj}_2(\boldsymbol{u}', \boldsymbol{y}, \boldsymbol{z})$ and Assumption 4 give the following relationship: $(*)$ if $\boldsymbol{u}' \in \mathtt{dom} f$ then $\boldsymbol{v} \in \mathtt{dom} f$.

[2] If $u_j = v_j$ then obviously $\mathtt{MJN}(u_j, v_j, z_j) = (u_j, v_j, z_j)$; suppose that $u_j \neq v_j$. This implies $u_j \neq x_j$ and $u_j \neq y_j$ (if $u_j = y_j$ then we would have $v_j = \mathtt{Mj}_2(u_j, u_j, z_j) = u_j$). Therefore, $u_j = z_j$. We must have $v_j = \mathtt{Mj}_2(z_j, y_j, z_j) = y_j$ since $v_j \neq u_j = z_j$. Thus, $\mathtt{MJN}(u_j, v_j, z_j) = \mathtt{MJN}(z_j, y_j, z_j) = (\alpha, y_j, \beta)$. We have $\{\{z_j, y_j, z_j\}\} = \{\{\alpha, y_j, \beta\}\}$, and so $\alpha = \beta = z_j$.

We have $\delta(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}') < \delta(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$ (since $x_i = z'_i$) and $\delta(\boldsymbol{u}',\boldsymbol{v},\boldsymbol{z}) < \delta(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$ (since $v_i = z_i$), so Assumption 4 gives

$$f(\mathtt{Mj}_1(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})) + f(\boldsymbol{u}') + f(\mathtt{Mn}_3(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})) \;\leq\; f(\boldsymbol{x}) + f(\boldsymbol{y}) + f(\boldsymbol{z}') \tag{31}$$

assuming that $\boldsymbol{z}' \in \mathtt{dom} f$, and

$$f(\mathtt{Mj}_2(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})) + f(\boldsymbol{v}) + f(\boldsymbol{z}') \;\leq\; f(\boldsymbol{u}') + f(\boldsymbol{v}) + f(\boldsymbol{z}) \tag{32}$$

assuming that $\boldsymbol{u}', \boldsymbol{v} \in \mathtt{dom} f$. If $\boldsymbol{z}' \in \mathtt{dom} f$ then (31) implies that $\boldsymbol{u}' \in \mathtt{dom} f$, and so $(*)$ implies that $\boldsymbol{v} \in \mathtt{dom} f$. Summing (31) and (32) gives (16). We thus assume that $\boldsymbol{z}' \notin \mathtt{dom} f$, then we have $\boldsymbol{u}' \notin \mathtt{dom} f$. (If $\boldsymbol{u}' \in \mathtt{dom} f$ then $(*)$ gives $\boldsymbol{v} \in \mathtt{dom} f$, and equation (32) then gives $\boldsymbol{z}' \in \mathtt{dom} f$ - a contradiction.)

The rest of the argument proceeds similarly to that for the case T1. Let us apply the i-expansion transformation to $(f, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ (again, without changing the notation). Consider function

$$g(\boldsymbol{w}) = \min_{d \in D} \{[d = c] \cdot C + f(d, \boldsymbol{w})\} \qquad \forall \boldsymbol{w} \in D^{\hat{V}}$$

where $\hat{V} = V - \{i\}$ and $C > f(\boldsymbol{x}) + f(\boldsymbol{y}) + f(\boldsymbol{z})$ is a sufficiently large constant. We can write

$$g(\hat{\boldsymbol{z}}) = f(\boldsymbol{z}) + C \qquad g(\hat{\boldsymbol{x}}) = f(\boldsymbol{x}) \qquad g(\hat{\boldsymbol{y}}) = f(\boldsymbol{y}) \qquad g(\hat{\boldsymbol{u}}) = f(\boldsymbol{u}) + C$$

Clearly, $(g, \hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})$ is a valid instance and $\delta(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}}) = \delta_{\min} - 1$, so Assumption 4 gives

$$\begin{gathered} g(\mathtt{Mj}_1(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})) + g(\mathtt{Mj}_2(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})) + g(\mathtt{Mn}_3(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})) \;\leq\; g(\hat{\boldsymbol{x}}) + g(\hat{\boldsymbol{y}}) + g(\hat{\boldsymbol{z}}) \\ g(\mathtt{Mj}_1(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})) + [f(\boldsymbol{u}) + C] + g(\mathtt{Mn}_3(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})) \;\leq\; f(\boldsymbol{x}) + f(\boldsymbol{y}) + [f(\boldsymbol{z}) + C] \end{gathered}$$

Therefore, $g(\mathtt{Mj}_1(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})) < C$, and thus $g(\mathtt{Mj}_1(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})) = f(a, \mathtt{Mj}_1(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})) = f(\mathtt{Mj}_1(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}))$. Similarly, $g(\mathtt{Mn}_3(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})) = f(b, \mathtt{Mn}_3(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})) = f(\mathtt{Mn}_3(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}))$, and hence the inequality above is equivalent to (16).

Case 2 $(\mathtt{Mj}_2(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}}), \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}}) \not\prec (\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})$. This implies, in particular, the following condition:

$(*)$ if $|\{x_j, y_j, z_j\}| = 3$ for $j \in V - \{i\}$ then $\mathtt{Mj}_2(x_j, y_j, z_j) = x_j$.

It is easy to check that $\Delta(\mathtt{Mj}_2(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}}), \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}}) \subseteq \Delta(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})$. Indeed, consider a node $j \in V - \{i\}$ with $\mathtt{Mj}_2(x_j, y_j, z_j) \neq y_j$; we need to show that $x_j \neq y_j$. If $|\{x_j, y_j, z_j\}| = 3$ then the claim is trivial, so it remains to consider the case when $\mathtt{MJN}(x_j, y_j, z_j)$ is defined via (15d) (case (15a) was eliminated by Proposition 6.9). We then have $\mathtt{Mj}_2(x_j, y_j, z_j) = x_j \sqcup y_j$, and so $x_j \sqcup y_j \neq y_j$ clearly implies $x_j \neq y_j$.

We thus must have $\Delta(\mathtt{Mj}_2(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}}), \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}}) = \Delta(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})$, otherwise the assumption of Case 2 would not hold. This implies the following:

$(**)$ if $x_j \neq y_j$ for $j \in V - \{i\}$ then $\mathtt{Mj}_2(x_j, y_j, z_j) \neq y_j$.

Let us define $\boldsymbol{u} = \mathtt{Mj}_1(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$, and let $\boldsymbol{x}', \boldsymbol{u}'$ be the labellings obtained from $\boldsymbol{x}, \boldsymbol{u}$ by setting $x'_i = u'_i = z_i$, so that we have

$$\begin{array}{ll} a = x_i = u_i & = \mathtt{Mj}_1(x_i, y_i, z_i) \\ c = x'_i = u'_i & = \mathtt{Mj}_2(x_i, y_i, z_i) \;\; (= z_i) \\ b & = \mathtt{Mn}_3(x_i, y_i, z_i) \;\; (= y_i) \end{array} \qquad \{a, c\} \in \overline{M}$$

We claim that the following identities hold:

$$\begin{array}{lcl} \mathtt{Mj}_1(\boldsymbol{x}', \boldsymbol{y}, \boldsymbol{z}) & = & \boldsymbol{u}' \\ \mathtt{Mj}_2(\boldsymbol{x}', \boldsymbol{y}, \boldsymbol{z}) & = & \mathtt{Mj}_2(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \\ \mathtt{Mn}_3(\boldsymbol{x}', \boldsymbol{y}, \boldsymbol{z}) & = & \mathtt{Mn}_3(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \end{array} \qquad \begin{array}{lcll} \boldsymbol{x} \sqcap \boldsymbol{u}' & = & \mathtt{Mj}_1(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) & = \boldsymbol{u} \\ \boldsymbol{x} \sqcup \boldsymbol{u}' & = & \boldsymbol{x}' & \end{array}$$

Indeed, we need to show that $x_j \sqcup u_j = x_j$ for $j \in V - \{i\}$. If $\mathtt{MJN}(x_j, y_j, z_j)$ was defined via (15b) then $\mathtt{Mj}_2(x_j, y_j, z_j) = y_j \sqcup z_j \neq x_j$ contradicting condition $(*)$. Similarly, if it was defined via (15c) then $\mathtt{Mj}_2(x_j, y_j, z_j) = x_j \sqcup z_j = z_j \neq x_j$ again contradicting condition $(*)$. (Note, in the latter case $x_j \sqcup z_j = z_j$ since by Proposition 6.11 we cannot have $\{x_j, z_j\} \in M$.) We showed that $\mathtt{MJN}(x_j, y_j, z_j)$ must be determined via (15d), so $u_j = \mathtt{Mj}_1(x_j, y_j, z_j) = x_j \sqcap y_j$ and $\mathtt{Mj}_2(x_j, y_j, z_j) = x_j \sqcup y_j$. If $x_j = y_j$ then the claim $x_j \sqcup u_j = x_j$ is trivial. If $x_j \neq y_j$ then condition $(**)$ implies $x_j \sqcup y_j \neq y_j$, and consequently $x_j \sqcup y_j = x_j$, $u_j = x_j \sqcap y_j = y_j$ and $x_j \sqcup u_j = x_j \sqcup y_j = x_j$, as claimed.

The rest of the proof proceeds analogously to the proof for the case T1. □

PROPOSITION 6.13. *For a node $i \in V$ the following situation is impossible:*

T6 $\mu(\{x_i, y_i, z_i\}) = \{x_i\}$, $\{y_i, z_i\} \in \overline{M}$.

PROOF. Let us define $\boldsymbol{u} = \mathtt{Mj}_2(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ and $\boldsymbol{v} = \mathtt{Mj}_2(\boldsymbol{u}, \boldsymbol{x}, \boldsymbol{z})$. It can be checked that $\mathtt{MJN}(\boldsymbol{v}, \boldsymbol{u}, \boldsymbol{z}) = (\boldsymbol{v}, \boldsymbol{u}, \boldsymbol{z})$. [3] Therefore, if we define $\boldsymbol{z}' = \boldsymbol{z}$ and $\boldsymbol{u}' = \boldsymbol{u}$ then the following identities will hold:

$$\begin{array}{lll} \mathtt{Mj}_1(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}') = \mathtt{Mj}_1(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) & \mathtt{Mj}_1(\boldsymbol{v}, \boldsymbol{u}', \boldsymbol{z}) = \boldsymbol{v} & \boldsymbol{v} = \mathtt{Mj}_2(\boldsymbol{u}', \boldsymbol{x}, \boldsymbol{z}) \\ \mathtt{Mj}_2(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}') = \boldsymbol{u}' & \mathtt{Mj}_2(\boldsymbol{v}, \boldsymbol{u}', \boldsymbol{z}) = \mathtt{Mj}_2(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = \boldsymbol{u} & \\ \mathtt{Mn}_3(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}') = \mathtt{Mn}_3(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) & \mathtt{Mn}_3(\boldsymbol{v}, \boldsymbol{u}', \boldsymbol{z}) = \boldsymbol{z}' & \end{array}$$

Let us modify $\boldsymbol{z}'$ and $\boldsymbol{u}'$ according to the following diagram:

$$\begin{array}{lll} c = z_i = u_i & = \mathtt{Mj}_2(x_i, y_i, z_i) \ (= v_i) & \\ b = z'_i = u'_i & = \mathtt{Mj}_1(x_i, y_i, z_i) \ (= y_i) & \{b, c\} \in \overline{M} \\ a & = \mathtt{Mn}_3(x_i, y_i, z_i) \ (= x_i) & \end{array}$$

It can be checked that the identities above still hold. Two cases are possible.

Case 1 $(\boldsymbol{u}', \boldsymbol{x}, \boldsymbol{z}) \prec (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$. This case is analogous to the Case 1 of T5. Namely, the fact that $\boldsymbol{v} = \mathtt{Mj}_2(\boldsymbol{u}', \boldsymbol{x}, \boldsymbol{z})$ and Assumption 4 give the following relationship: $(*)$ if $\boldsymbol{u}' \in \mathtt{dom} f$ then $\boldsymbol{v} \in \mathtt{dom} f$. We have $\delta(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}') < \delta(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ (since $y_i = z'_i$) and $\delta(\boldsymbol{v}, \boldsymbol{u}', \boldsymbol{z}) < \delta(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ (since $v_i = z_i$), so Assumption 4 gives

$$f(\mathtt{Mj}_1(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})) + f(\boldsymbol{u}') + f(\mathtt{Mn}_3(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})) \ \leq \ f(\boldsymbol{x}) + f(\boldsymbol{y}) + f(\boldsymbol{z}') \tag{33}$$

assuming that $\boldsymbol{z}' \in \mathtt{dom} f$, and

$$f(\boldsymbol{v}) + f(\mathtt{Mj}_2(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})) + f(\boldsymbol{z}') \ \leq \ f(\boldsymbol{v}) + f(\boldsymbol{u}') + f(\boldsymbol{z}) \tag{34}$$

assuming that $\boldsymbol{u}', \boldsymbol{v} \in \mathtt{dom} f$. If $\boldsymbol{z}' \in \mathtt{dom} f$ then (33) implies that $\boldsymbol{u}' \in \mathtt{dom} f$, and so $(*)$ implies that $\boldsymbol{v} \in \mathtt{dom} f$. Summing (33) and (34) gives (16). We thus assume that $\boldsymbol{z}' \notin \mathtt{dom} f$, then we have $\boldsymbol{u}' \notin \mathtt{dom} f$. (If $\boldsymbol{u}' \in \mathtt{dom} f$ then $(*)$ gives $\boldsymbol{v} \in \mathtt{dom} f$, and equation (34) then gives $\boldsymbol{z}' \in \mathtt{dom} f$ - a contradiction.)

The rest of the argument proceeds similarly to that for the case T1.

Case 2 $(\boldsymbol{u}', \boldsymbol{x}, \boldsymbol{z}) \not\prec (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$. Let us show that each node $j \in V - \{i\}$ satisfies the following:

(a) If $j \in \Delta(\boldsymbol{u}', \boldsymbol{x}, \boldsymbol{z})$ then $j \in \Delta(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$. In other words, if $u'_j \neq x_j$ then $y_j \neq x_j$.

[3] If $u_j = v_j$ then obviously $\mathtt{MJN}(v_j, u_j, z_j) = (v_j, u_j, z_j)$; suppose that $u_j \neq v_j$. This implies $u_j \neq x_j$ (otherwise we would have $v_j = \mathtt{Mj}_2(u_j, u_j, z_j) = u_j$). If $\mathtt{MJN}(x_j, y_j, z_j)$ is determined via (15b) then $\{y_j, z_j\} \in \overline{M}$ by Proposition 6.11 and so $u_j = z_j$ and $v_j = z_j$. It remains to consider the case when it is determined via (15d) (cases (15a) and (15c) have been eliminated).

We have $u_j = x_j \sqcup y_j = y_j$ since $u_j \neq x_j$, and so $v_j = \mathtt{Mj}_2(y_j, x_j, z_j) = y_j \sqcup x_j = x_j$ since $v_j \neq u_j = y_j$ (clearly, $\mathtt{Mj}_2(y_j, x_j, z_j)$ is also determined via (15d)). We thus have $\mathtt{MJN}(v_j, u_j, z_j) = \mathtt{MJN}(x_j, y_j, z_j) = (\alpha, u_j, z_j)$. Condition $\{\{v_j, u_j, z_j\}\} = \{\{\alpha, u_j, z_j\}\}$ implies that $\alpha = v_j$.

(b) If $j \in \Delta^M(\boldsymbol{u}', \boldsymbol{x}, \boldsymbol{z})$ then $j \in \Delta^M(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$. Namely, if $(u'_j, x_j, z_j) = (\alpha, \beta, \beta)$ or $(u'_j, x_j, z_j) = (\beta, \alpha, \beta)$ where $\{\alpha, \beta\} \in M$ then $u'_j = y_j$ and thus $(x_j, y_j, z_j) = (\beta, \alpha, \beta)$ or $(x_j, y_j, z_j) = (\alpha, \beta, \beta)$ respectively.
(c) $\mu(\{u'_j, x_j, z_j\}) \neq \{u'_j\}$.

If $\mathtt{MJN}(x_j, y_j, z_j)$ is determined via (15b) then we must have $\{y_j, z_j\} \in \overline{M}$ by Proposition 6.11, and so $u'_j = \mathtt{Mj}_2(x_j, y_j, z_j) = z_j$. Checking (a-c) is then straightforward.

It remains to consider the case when $\mathtt{MJN}(x_j, y_j, z_j)$ is determined via (15d) - all other cases have been eliminated. Condition (c) then clearly holds, and $u'_j = \mathtt{Mj}_2(x_j, y_j, z_j) = x_j \sqcup y_j$. If $u'_j = x_j$ then (a,b) are trivial since their preconditions do not hold. It is also straightforward to check that (a,b) hold if $u'_j = y_j \neq x_j$.

We proved that (a-c) hold for each $j \in V - \{i\}$. This implies that $(\boldsymbol{u}', \boldsymbol{x}, \boldsymbol{z}) \prec (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ (contradicting the assumption of Case 2) due to the fourth component in (17) which is zero for the triple $(\boldsymbol{u}', \boldsymbol{x}, \boldsymbol{z})$. (Note that at node i we have $(u'_i, x_i, z_i) = (y_i, x_i, z_i)$ and so $\mu(\{u'_i, x_i, z_i\} = \{x_i\} \neq \{u'_i\})$. □

7. PROOF OF THEOREM 3.4

In this section we present an algorithm for minimising instances from VCSP(Γ). The idea for the algorithm and some of the proof techniques have been influenced by the techniques used by Takhanov [Takhanov 2010a] for proving the absence of *arithmetical deadlocks* in certain instances. However, the algorithm itself is very different from Takhanov's approach. (The latter does not rely on submodular minimization algorithms; instead, it performs a reduction to an optimization problem in a perfect graph).

Let $f : \mathcal{D} \to \overline{\mathbb{Q}}_+$ be the function to be minimised that admits an STP on M and an MJN on $P - M$, for some symmetric $M \subseteq P$. (We no longer assume that M is determined by the language; instead, it is an arbitrary symmetric set.) Let V be the set of variables of function f (which we will also call nodes), and D_i be the domain of variable $i \in V$ with $\mathcal{D} = \times_{i \in V} D_i$. In the beginning all domains are the same ($D_i = D$), but as the algorithm progresses we will allow D_i to become different for different $i \in V$. Similarly, operations $\sqcap, \sqcup$ may act differently on different components of vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{D}$. We denote by $\sqcap_i, \sqcup_i : D_i \times D_i \to D_i$ the i-th operations of $\langle \sqcap, \sqcup \rangle$. We also denote by $\mathtt{Mj}_{1i}, \mathtt{Mj}_{2i}, \mathtt{Mn}_{3i} : D_i \times D_i \times D_i \to D_i$ the i-th operations of $\langle \mathtt{Mj}_1, \mathtt{Mj}_2, \mathtt{Mn}_3 \rangle$. The definition of a binary multimorphism (Definition 2.4) naturally extends to our setting where operations $\sqcap, \sqcup$ may act differently on different components of vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{D}$. In particular, a collection $\langle \sqcap, \sqcup \rangle$ of pairs of operations is called a (binary *multi-sorted*) multimorphism of f if

$$f(\boldsymbol{x} \sqcap \boldsymbol{y}) + f(\boldsymbol{x} \sqcup \boldsymbol{y}) \leq f(\boldsymbol{x}) + f(\boldsymbol{y}) \qquad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathtt{dom} f \tag{35}$$

where $\boldsymbol{x} \sqcap \boldsymbol{y} = (x_1 \sqcap_1 y_1, \ldots, x_n \sqcap_n y_n)$ and $\boldsymbol{x} \sqcup \boldsymbol{y} = (x_1 \sqcup_1 y_1, \ldots, x_n \sqcup_n y_n)$ with $\boldsymbol{x} = (x_1, \ldots, x_n)$, $\boldsymbol{y} = (y_1, \ldots, y_n)$, and n is the arity of f. (The term multi-sorted comes from the study of crisp VCSPs with operations acting differently on different components [Bulatov and Jeavons 2003].)

We denote by P the collection of sets $P = (P_i)_{i \in V}$ where $P_i = \{\{a, b\} \,|\, a, b \in D_i, a \neq b\}$. We denote by M a collection of subsets $M = (M_i)_{i \in V}$, $M_i \subseteq P_i$, and $\overline{M} = (\overline{M}_i)_{i \in V}$, $\overline{M}_i = P_i - M_i$. We now extend Definition 3.1 as follows.

Definition 7.1. Let $\langle \sqcap, \sqcup \rangle$ be a collection of pairs of binary operations and $\langle \mathtt{Mj}_1, \mathtt{Mj}_2, \mathtt{Mn}_3 \rangle$ be a collection of triples of ternary operations.

— $\langle \sqcap, \sqcup \rangle$ is an ***STP on*** M if for all $i \in V$ the pair $\langle \sqcap_i, \sqcup_i \rangle$ is an STP on M_i.

— $\langle \mathtt{Mj}_1, \mathtt{Mj}_2, \mathtt{Mn}_3 \rangle$ is an ***MJN on*** $\overline{M}$ if for all $i \in V$ the triple $\langle \mathtt{Mj}_{1i}, \mathtt{Mj}_{2i}, \mathtt{Mn}_{3i} \rangle$ is an MJN on $\overline{M}_i$.

We assume that multimorphisms $\langle \sqcap_i, \sqcup_i \rangle$ for all $i \in V$ are given as a part of the input. (We can make this assumption since in Theorem 3.4 the language is fixed. To repeat, in the beginning multimorphisms $\langle \sqcap_i, \sqcup_i \rangle$ are same for all nodes i, but they will change as the algorithm progresses.)

We are now ready to present the algorithm; it will consist of three stages.

Stage 1: Decomposition into binary relations

We will need the following fact.

PROPOSITION 7.2. *Function f admits a majority polymorphism.*

PROOF. We use an argument from [Takhanov 2010a]. Define

$$\begin{aligned} \bar{\mu}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) &= [(\boldsymbol{y} \sqcup \boldsymbol{x}) \sqcap (\boldsymbol{y} \sqcup \boldsymbol{z})] \sqcap (\boldsymbol{x} \sqcup \boldsymbol{z}) \\ \mu(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) &= \mathtt{Mj}_1(\bar{\mu}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}), \bar{\mu}(\boldsymbol{y}, \boldsymbol{z}, \boldsymbol{x}), \bar{\mu}(\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{y})) \end{aligned}$$

Suppose that $\{x, y, z\} = \{a, b\} \in P$. It can be checked that $\bar{\mu}(x, y, z)$ acts as the majority operation if $\langle \sqcap, \sqcup \rangle$ is commutative on $\{a, b\}$, and $\bar{\mu}(x, y, z) = x$ otherwise. This implies that μ acts as the majority operation on P. □

Since the instance admits a majority polymorphism, $\mathtt{dom} f$ can be decomposed [Baker and Pixley 1975] into unary relations $\rho_i \subseteq D_i, i \in D_i$ and binary relations $\rho_{ij} \subseteq D_i \times D_j$, $i, j \in V$, $i \neq j$ such that

$$\boldsymbol{x} \in \mathtt{dom} f \quad \Leftrightarrow \quad [x_i \in \rho_i \ \forall i \in V] \quad \text{and} \quad [(x_i, x_j) \in \rho_{ij} \ \forall i, j \in V, i \neq j] \tag{36}$$

We will always assume that $(x, y) \in \rho_{ij} \Leftrightarrow (y, x) \in \rho_{ji}$. We use the following notation for relations:

— If $\rho_{ij} \in D_i \times D_j$, $X \subseteq D_i$ and $Y \subseteq D_j$ then

$$\rho_{ij}(X, \cdot) = \{y \mid \exists x \in X \text{ s.t. } (x, y) \in \rho_{ij}\} \qquad \rho_{ij}(\cdot, Y) = \{x \mid \exists y \in Y \text{ s.t. } (x, y) \in \rho_{ij}\}$$

If $X = \{x\}$ and $Y = \{y\}$ then these two sets will be denoted as $\rho_{ij}(x, \cdot)$ and $\rho_{ij}(\cdot, y)$ respectively.

— If $\rho \in D_1 \times D_2$ and $\rho' \in D_2 \times D_3$ then we define their composition as

$$\rho \circ \rho' = \{(x, z) \in D_1 \times D_3 \mid \exists y \in D_2 \text{ s.t. } (x, y) \in \rho, (y, z) \in \rho'\}$$

In the first stage we establish arc- and path-consistency using the standard constraint-processing techniques [Cooper 1989] so that the resulting relations satisfy

$$\begin{aligned} &\text{(arc-consistency)} && \{x \mid (\exists y)(x, y) \in \rho_{ij}\} = \rho_i && \forall \text{ distinct } i, j \in V \\ &\text{(path-consistency)} && \rho_{ik}(x, \cdot) \cap \rho_{jk}(y, \cdot) \neq \varnothing && \forall \text{ distinct } i, j, k \in V, (x, y) \in \rho_{ij} \end{aligned}$$

It is well known that for instances with unary and binary relations establishing arc- and path-consistency is equivalent to establishing *strong 3-consistency*, which is defined as the following property: for any three variables $i, j, k \in V$ and any solution to the subproblem on variables i and j, defined as the projection of the problem onto i and j, can be extended to a solution to the subproblem on variables i, j, and k [Cooper 1989]. It is known that in the presence of a majority polymorphism strong 3-consistency is equivalent to global consistency [Jeavons et al. 1998], that is, $\mathtt{dom} f$ is empty iff all ρ_i and ρ_{ij} are empty. Moreover, the relations ρ_i, ρ_{ij} are uniquely determined by f via

$$\rho_i = \{x_i \mid \boldsymbol{x} \in \mathtt{dom} f\} \qquad \rho_{ij} = \{(x_i, x_j) \mid \boldsymbol{x} \in \mathtt{dom} f\}$$

The second equation implies that any polymorphism of f is also a polymorphism of ρ_{ij}.

From now on we will assume that $D_i = \rho_i$ for all $i \in V$. This can be achieved by reducing sets D_i if necessary. We will also assume that all sets D_i are non-empty.

Stage 2: Modifying M and $\langle \sqcap, \sqcup \rangle$

At this point we have the following data: function f on the set of variables V, relations ρ_i, ρ_{ij} (with $\rho_i = D_i$), and a multi-sorted multimorphism $\langle \sqcap, \sqcup \rangle$. For each $i \in V$ let us set M_i to be the set of pairs $\{a, b\} \in P_i$ on which $\langle \sqcap_i, \sqcup_i \rangle$ is commutative.

The second stage of the algorithm works by iteratively growing sets M_i and simultaneously modifying operations $\langle \sqcap_i, \sqcup_i \rangle$ so that (i) $\langle \sqcap_i, \sqcup_i \rangle$ is still a conservative pair which is commutative on M_i and non-commutative on $\overline{M}_i$, and (ii) $\langle \sqcap, \sqcup \rangle$ is a multi-sorted multimorphism of f. It stops when we get $M_i = P_i$ for all $i \in V$. Thus, the output of Stage 2 is a new multi-sorted operation $\langle \sqcap, \sqcup \rangle$ which is an STP multimorphism of f. Note that the function f is never modified.

We now describe one iteration. First, we identify subset $U \subseteq V$ and subsets $A_i, B_i \subseteq D_i$ for each $i \in U$ using the following algorithm:

1: pick node $k \in V$ and pair $\{a, b\} \in \overline{M}_k$.
   (If they do not exist, terminate and go to Stage 3.)
2: set $U = \{k\}$, $A_k = \{a\}$, $B_k = \{b\}$
3: **while** there exists $i \in V - U$ such that $\rho_{ki}(A_k, \cdot) \cap \rho_{ki}(B_k, \cdot) = \varnothing$ **do**
4: add i to U, set $A_i = \rho_{ki}(A_k, \cdot)$, $B_i = \rho_{ki}(B_k, \cdot)$
   // *compute "closure" of sets* A_i *for* $i \in U$
5: **while** there exists $a \in D_k - A_k$ s.t. $a \in \rho_{ki}(\cdot, A_i)$ for some $i \in U - \{k\}$ **do**
6: add a to A_k, set $A_j = \rho_{kj}(A_k, \cdot)$ for all $j \in U - \{k\}$
7: **end while**
   // *compute "closure" of sets* B_i *for* $i \in U$
8: **while** there exists $b \in D_k - B_k$ s.t. $b \in \rho_{ki}(\cdot, B_i)$ for some $i \in U - \{k\}$ **do**
9: add b to B_k, set $B_j = \rho_{kj}(B_k, \cdot)$ for all $j \in U - \{k\}$
10: **end while**
   // *done*
11: **end while**
12: **return** set $U \subseteq V$ and sets $A_i, B_i \subseteq D_i$ for $i \in U$

LEMMA 7.3. *Sets U and A_i, B_i for $i \in U$ produced by the algorithm have the following properties:*

(a) Sets A_i and B_i for $i \in U$ are disjoint.
(b) $\{a, b\} \in \overline{M}_i$ for all $i \in U$, $a \in A_i$, $b \in B_i$.
(c) $\rho_{ki}(A_k, \cdot) = A_i$, $\rho_{ki}(B_k, \cdot) = B_i$, $\rho_{ki}(\cdot, A_i) = A_k$, $\rho_{ki}(\cdot, B_i) = B_k$ for all $i \in U - \{k\}$ where k is the node chosen in line 1.
(d) Suppose that $i \in U$ and $j \in \overline{U} \equiv V - U$. If $(c, x) \in \rho_{ij}$ where $c \in A_i \cup B_i$ and $x \in D_j$ then $(d, x) \in \rho_{ij}$ for all $d \in A_i \cup B_i$.

To complete the iteration, we modify sets M_i and operations $\sqcap_i, \sqcup_i$ for each $i \in U$ as follows:

— add all pairs $\{a, b\}$ to M_i where $a \in A_i$, $b \in B_i$.
— redefine $a \sqcap_i b = b \sqcap_i a = a$, $a \sqcup_i b = b \sqcup_i a = b$ for all $a \in A_i$, $b \in B_i$

LEMMA 7.4. *The new collection $\langle \sqcap, \sqcup \rangle$ of pairs of operations is a (multi-sorted) multimorphism of f.*

A proof of Lemmas 7.3 and 7.4 is given in the next section. They imply that all steps are well-defined, and upon termination the algorithm produces a pair $\langle \sqcap, \sqcup \rangle$ which is a multi-sorted STP multimorphism of f. Clearly, the number of iterations is at most $|V| \cdot \frac{|D|(|D|-1)}{2}$ where D is the initial domain, and so Stage 2 takes a polynomial time.

Stage 3: Reduction to submodular minimisation

At this stage we have an instance that admits a multi-sorted STP multimorphism. Theorem 2.5 applies to VCSP instances with cost functions admitting a (non-multi-sorted) STP multimorphism. However, the proof of Theorem 2.5 [Cohen et al. 2008, Theorem 8.2] also works in our case. In particular, the proof of Theorem 2.5 consists of (i) establishing strong 3-consistency and (ii) finding a total order on the domains of the variables (both the domains and the orders are allowed to be different for different variables) that renders the objective function submodular [Schrijver 2000]. The same argument as in [Cohen et al. 2008] shows that the objective function f is submodular and thus minimisable in polynomial time [Schrijver 2000]. (The tractability of minimising submodular functions with different variables having different (distributive lattice) orders was first studied in [Krokhin and Larose 2008].)

7.1. Algorithm's correctness: proof of Lemmas 7.3 and 7.4

First, we show that the following holds at any moment during Stage 2.

PROPOSITION 7.5. *If $\{a,b\} \in \overline{M}_i$, $\{a',b'\} \in P_j$ and $(a,a'),(b,b') \in \rho_{ij}$, where i,j are distinct nodes in V, then exactly one of the following holds:*

(i) $(a,b'),(b,a') \in \rho_{ij}$
(ii) $(a,b'),(b,a') \notin \rho_{ij}$ *and* $\{a',b'\} \in \overline{M}_j$

PROOF. First, suppose that $\{a',b'\} \in M_j$. We need to show that case (i) holds. Operations $\sqcap_i, \sqcup_i$ are non-commutative on $\{a,b\}$, while $\sqcap_j, \sqcup_j$ are commutative on $\{a',b'\}$. It is easy to check that

$$\{(a,a') \sqcap (b,b'), (b,b') \sqcap (a,a'), (a,a') \sqcup (b,b'), (b,b') \sqcup (a,a')\} = \{(a,a'),(a,b'),(a',b),(a',b')\}$$

Since $\sqcap$, $\sqcup$ are polymorphisms of ρ_{ij}, all assignments involved in the equation above belong to ρ_{ij}. Thus, (i) holds.

Now suppose $\{a',b'\} \in \overline{M}_j$. We then have

$$\mathtt{Mn}_3((a,a'),(b,b'),(a,b')) = (b,a') \qquad \mathtt{Mn}_3((a,a'),(b,b'),(b,a')) = (a,b')$$

$\mathtt{Mn}_3$ is a polymorphism of ρ_{ij}, therefore if one of the assignments $(a,b'),(b,a')$ belongs to ρ_{ij} then the other one also belongs to ρ_{ij}. This proves the proposition. □

7.1.1. Proof of Lemma 7.3(a-c). It follows from construction that during all stages of the algorithm there holds

$$\rho_{ki}(A_k,\cdot) = A_i\ , \quad \rho_{ki}(B_k,\cdot) = B_i \qquad\qquad \forall i \in U - \{k\} \tag{37}$$

Strong 3-consistency also implies that sets A_i, B_i for $i \in U$ are non-empty. Clearly, properties (a) and (b) of Lemma 7.3 hold after initialization (line 2). Let us prove that each step of the algorithm preserves these two properties. Note, property (a) together with (37) imply that $(a,b') \notin \rho_{ki}$ if $a \in A_k$, $b' \in B_i$, and $(b,a') \notin \rho_{ki}$ if $b \in B_k$, $a' \in A_i$, where $i \in U - \{k\}$.

First, consider line 4, i.e. adding i to U with $A_i = \rho_{ki}(A_k,\cdot)$, $B_i = \rho_{ki}(B_k,\cdot)$. Property (a) for node i follows from the precondition of line 3; let us show (b) for node i. Suppose

that $a' \in A_i$, $b' \in B_i$, then there exist $a \in A_k$, $b \in B_k$ such that $(a,a'),(b,b') \in \rho_{ki}$. We have $(a,b') \notin \rho_{ki}$, so by Proposition 7.5 we get $\{a',b'\} \in \overline{M}$.

Now consider line 6, i.e. adding a to A_k and updating A_j for $j \in U - \{k\}$ accordingly. (The analysis of line 9 will be symmetrical, and we omit it.) We denote by A_j° and A_j respectively the old and the new set for node $j \in U$. There must exist node $i \in U - \{k\}$ and element $a' \in A_i^\circ$ such that $(a,a') \in \rho_{ki}$. We prove below that properties (a) and (b) are preserved for nodes k, i and all nodes $j \in U - \{k,i\}$.

Node k It is clear that $a \notin B_k$, otherwise we would have $a' \in \rho_{ki}(B_k,\cdot) = B_i$ contradicting condition $A_i^\circ \cap B_i = \varnothing$. Thus, property (a) for node k holds. Consider an element $b \in B_k$. By arc-consistency there exists an element $b' \in \rho_{ki}(b,\cdot) \subseteq B_i$. From property (b) we get $\{a',b'\} \in \overline{M}_i$. We also have $(b,a') \notin \rho_{ki}$ since $A_i^\circ \cap \rho_{ki}(B_k,\cdot) = A_i^\circ \cap B_i = \varnothing$. By Proposition 7.5 we get $\{a,b\} \in \overline{M}_k$. Thus, property (b) holds for node k.

Node i Let us prove that $A_i \cap B_i = \varnothing$. Suppose not, then $(a,b') \in \rho_{ki}$ for some $b' \in B_i$. There must exist $b \in B_k$ with $(b,b') \in \rho_{ki}$. We have $\rho_{ki} \cap (\{a,b\} \times \{a',b'\}) = \{(a,a'),(b,b'),(a,b')\}$ and $\{a',b'\} \in \overline{M}_i$, which is a contradiction by Proposition 7.5. This proves property (a) for node i.

Property (b) for node i follows from property (a) for nodes k,i, property (b) for node k, and Proposition 7.5.

Node $j \in U - \{k,i\}$ Let us prove that $A_j \cap B_j = \varnothing$. Suppose not, then $(a,y) \in \rho_{kj}$ for some $y \in B_j$. There must exist $b \in B_k$ with $(b,y) \in \rho_{kj}$, and $b' \in B_i$ with $(b,b') \in \rho_{ki}$. We also have $a' \in A_i^\circ = \rho_{ki}(A_k^\circ,\cdot)$, therefore there must exist $c \in A_k^\circ$ with $(c,a') \in \rho_{ki}$, and $x \in A_j^\circ$ with $(c,x) \in \rho_{kj}$. It can be seen that

$$\begin{aligned} \rho_{ki} \cap (\{a,b,c\} \times \{a',b'\}) &= \{(a,a'),(c,a'),(b,b')\} \\ \rho_{kj} \cap (\{a,b,c\} \times \{x,y\}) &= \{(a,y),(c,x),(b,y)\} \end{aligned}$$

Indeed, all listed assignments belong to ρ_{ki} or ρ_{kj} by construction; we need to show that remaining assignments do not belong to these relations. We have $(a,b'),(c,b'),(b,a') \notin \rho_{ki}$ since we have already established property (a) for nodes k and i. We also have $(c,y),(b,x) \notin \rho_{kj}$ since $A_k^\circ \cap B_k = \varnothing$ and $A_j^\circ \cap B_j = \varnothing$. Combining it with the fact that $\{x,y\} \in \overline{M}$ and using Proposition 7.5 gives that $(a,x) \notin \rho_{kj}$.

Consider relation $\beta_{ij} = \rho'_{ik} \circ \rho_{kj}$ where $\rho'_{ik} = \{(d',d) \in \rho_{ik} \mid d \in \{a,b,c\}\}$. It is easy to check that $(a',x),(a',y),(b',y) \in \beta_{ij}$ and $(b',x) \notin \beta_{ij}$. We have $\{a',b'\} \in \overline{M}_i$ and $\{x,y\} \in \overline{M}_j$, so $\mathtt{Mn}_3((a',x),(a',y),(b',y)) = (b',x)$. Clearly, $\mathtt{Mn}_3$ is a polymorphism of ρ'_{ik} and β_{ij}, therefore we must have $(b',x) \in \beta_{ij}$ - a contradiction. This proves property (a) for node j.

Property (b) for node j follows from property (a) for nodes k,j, property (b) for node k, and Proposition 7.5.

Concluding remark We showed that throughout the algorithm sets U, A_i, B_i satisfy properties (a,b) and equation (37). It is easy to see that after running lines 5-7 we also have $\rho_{ki}(\cdot,A_i) = A_k$, and after running lines 8-10 we have $\rho_{ki}(\cdot,B_i) = B_k$. Thus, property (c) holds upon termination, which concludes the proof of Lemma 7.3(a-c).

7.1.2. Proof of Lemma 7.3(d). First, we will prove the following claim:

PROPOSITION 7.6. *Suppose that* $(a,x),(b,x),(c,y) \in \rho_{ij}$ *where* $i \in U$, $j \in \overline{U}$, $a \in A_i$, $b \in B_i$, $c \in A_i \cup B_i$, $x,y \in D_j$. *Then* $(a,y),(b,y),(c,x) \in \rho_{ij}$.

PROOF. We claim that there exists a relation $\gamma_{ii} \subseteq D_i \times D_i$ with the following properties:

(i) γ_{ii} is an equivalence relation on D_i such that A_i and B_i are among its classes.

(ii) Operation $\mathtt{Mn}_{3i}$ is a polymorphism of γ_{ii}.

Indeed, for $i = k$ such relation can be constructed as follows. Let us set $\gamma_{kk} = \{(a, a) | a \in D_k\}$ and iteratively update it via $\gamma_{kk} := \gamma_{kk} \circ \rho_{ki} \circ \rho_{ik}$ for $i \in U - \{k\}$. Set γ_{ii} will never shrink; we stop when no such operation can change γ_{kk}. Clearly, at this point γ_{ii} is an equivalence relation. By comparing this scheme with lines 5-10 of the algorithm we conclude that (i) holds. Finally, (ii) follows from the fact that polymorphisms are preserved under compositions. If $i \in U - \{k\}$ then we take $\gamma_{ii} = \rho_{ik} \circ \gamma_{kk} \circ \rho_{ki}$; (i)-(ii) then follow from property (c) of Lemma 7.3.

We are now ready to prove Proposition 7.6. We can assume that $x \neq y$, otherwise the claim is trivial. Assume that $c \in A_i$ (the case $c \in B_i$ is analogous). Suppose that $(b, y) \notin \rho_{ij}$. We have $\{b, c\} \in \overline{M}$, so Proposition 7.5 implies that $\{x, y\} \in \overline{M}$. Consider relation $\gamma'_{ii} = \{(x', y') \in \gamma_{ii} \mid y' \notin B_i - \{b\})\}$. Operation $\mathtt{Mn}_{3i}$ is conservative, therefore it is a polymorphism of γ'_{ii} as well. Define relation $\beta_{ij} = \gamma'_{ii} \circ \rho_{ij} \subseteq D_i \times D_j$, then $\mathtt{Mn}_3$ is a polymorphism of β_{ij}. It is easy to check that $(a, y), (a, x), (b, x) \in \beta_{ij}$. Operation $\mathtt{Mn}_3$ is a polymorphism of β_{ij} and it acts as the minority operation on $\{a, b\} \in \overline{M}$ and $\{x, y\} \in \overline{M}$, therefore $\mathtt{Mn}_3((a, y), (a, x), (b, x)) = (b, y) \in \beta_{ij}$. This implies that $(b, y) \in \rho_{ij}$, contradicting the assumption made earlier. We showed that we must have $(b, y) \in \rho_{ij}$. The fact that $\{a, b\} \in \overline{M}$ and Proposition 7.5 then imply that $(a, y) \in \rho_{ij}$. Finally, the fact that $\{c, b\} \in \overline{M}$ and Proposition 7.5 imply that $(c, x) \in \rho_{ij}$. Proposition 7.6 is proved. □

We can now prove Lemma 7.3(d) under the following assumption:

(∗) Sets $\rho_{ij}(A_i, \cdot)$ and $\rho_{ij}(B_i, \cdot)$ have a non-empty intersection.

(This assumption clearly holds if $i = k$, otherwise the algorithm wouldn't have terminated; we will later show that (∗) holds for nodes $i \in U - \{k\}$ as well.)

First, let us prove that $\rho_{ij}(A_i, \cdot) = \rho_{ij}(B_i, \cdot)$. Suppose that $y \in \rho_{ij}(A_i, \cdot)$, then $(c, y) \in \rho_{ij}$ for some $c \in A_i$. From assumption (∗) we get that there exist $a \in A_i$, $b \in B_i$, $x \in D_j$ such that $(a, x), (b, x) \in \rho_{ij}$. Proposition 7.6 implies that $(b, y) \in \rho_{ij}$, and thus $\rho_{ij}(A_i, \cdot) \subseteq \rho_{ij}(B_i, \cdot)$. By symmetry we also have $\rho_{ij}(B_i, \cdot) \subseteq \rho_{ij}(A_i, \cdot)$, implying $\rho_{ij}(A_i, \cdot) = \rho_{ij}(B_i, \cdot)$.

Second, let us prove that if $(a, x) \in \rho_{ij}$ where $a \in A_i$, $x \in D_j$ then $(c, x) \in \rho_{ij}$ for all $c \in B_i$. (We call this claim [AB]). As we showed in the previous paragraph, there exists $b \in B_i$ such that $(b, x) \in \rho_{ij}$. We can also select $y \in D_j$ such that $(c, y) \in \rho_{ij}$. Proposition 7.6 implies that $(c, x) \in \rho_{ij}$, as desired.

A symmetrical argument shows that if $(b, x) \in \rho_{ij}$ where $b \in B_i$, $x \in D_j$ then $(c, x) \in \rho_{ij}$ for all $c \in A_i$ [BA]. By combining facts [AB] and [BA] we obtain that if $(a, x) \in \rho_{ij}$ where $a \in A_i$, $x \in D_j$ then $(c, x) \in \rho_{ij}$ for all $c \in A_i$ [AA], and also that if $(b, x) \in \rho_{ij}$ where $b \in B_i$, $x \in D_j$ then $(c, x) \in \rho_{ij}$ for all $c \in B_i$ [BB].

We have proved Lemma 7.3(d) assuming that (∗) holds (and in particular, for $i = k$). It remains to show that (∗) holds for $i \in U - \{k\}$. Let us select $(a', x) \in \rho_{ij}$ where $a' \in A_i$, $x, y \in D_j$. By strong 3-consistency there exists $a \in D_k$ such that $(a, a') \in \rho_{ki}$ and $(a, x) \in \rho_{kj}$. By Lemma 7.3(c) we get that $a \in A_k$. As we have just shown, there exists $b \in B_k$ such that $(b, x) \in \rho_{kj}$. By strong 3-consistency there exists $b' \in D_i$ such that $(b, b') \in \rho_{ki}$ and $(b', x) \in \rho_{ij}$. By Lemma 7.3(c) we get that $b' \in B_i$. We have shown that $x \in \rho_{ij}(A_i, \cdot)$ and $x \in \rho_{ij}(B_i, \cdot)$, which proves (∗).

7.1.3. Proof of Lemma 7.4. Suppose we have an arc- and path-consistent instance with an STP on M and MJN on $\overline{M}$ and non-empty subset U with $A_i, B_i \subseteq D_i$ for $i \in U$ that satisfy properties (a-d) of Lemma 7.3 (where node $k \in U$ is fixed). Let us denote by M° and M the set before and after the update respectively. Similarly, $\langle \sqcap^\circ, \sqcup^\circ \rangle$ and $\langle \sqcap, \sqcup \rangle$ denote operations before and after the update. We need to show that

$$f(\boldsymbol{x} \sqcap \boldsymbol{y}) + f(\boldsymbol{x} \sqcup \boldsymbol{y}) \leq f(\boldsymbol{x}) + f(\boldsymbol{y}) \qquad \text{if } \boldsymbol{x}, \boldsymbol{y} \in \texttt{dom} f \tag{38}$$

For a vector $\boldsymbol{z} \in \mathcal{D}$ and subset $S \subseteq V$ we denote by $\boldsymbol{z}^S$ the restriction of $\boldsymbol{z}$ to S. Given $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{D}$, denote

$$\delta(\boldsymbol{x}, \boldsymbol{y}) = \begin{cases} 0 & \text{if } \boldsymbol{x}^U \sqcap \boldsymbol{y}^U = \boldsymbol{x}^U \sqcap^\circ \boldsymbol{y}^U \\ 1 & \text{otherwise} \end{cases} \qquad \Delta(\boldsymbol{x}, \boldsymbol{y}) = \{i \in \overline{U} \mid x_i \neq y_i\}$$

Let us introduce a partial order $\preceq$ on pairs $(\boldsymbol{x}, \boldsymbol{y})$ as the lexicographical order on vector $(|\Delta(\boldsymbol{x}, \boldsymbol{y})|, \delta(\boldsymbol{x}, \boldsymbol{y}))$ (the first component is more significant than the second). We use induction on this order. The base of the induction is given by the following lemma.

LEMMA 7.7. *Suppose that $\boldsymbol{x}, \boldsymbol{y} \in \texttt{dom} f$ and either $|\Delta(\boldsymbol{x}, \boldsymbol{y})| \leq 1$ or $\delta(\boldsymbol{x}, \boldsymbol{y}) = 0$. Then condition* (38) *holds.*

PROOF. If $\delta(\boldsymbol{x}, \boldsymbol{y}) = 0$ then $\boldsymbol{x} \sqcap \boldsymbol{y} = \boldsymbol{x} \sqcap^\circ \boldsymbol{y}$ and $\boldsymbol{x} \sqcup \boldsymbol{y} = \boldsymbol{x} \sqcup^\circ \boldsymbol{y}$, so the claim is trivial. Suppose that $|\Delta(\boldsymbol{x}, \boldsymbol{y})| \leq 1$ and $\delta(\boldsymbol{x}, \boldsymbol{y}) = 1$.

There exists node $i \in U$ such that either $x_i \in A_i, y_i \in B_i$ or $x_i \in B_i, y_i \in A_i$. Lemma 7.3(c) implies that either $x_i \in A_i, y_i \in B_i$ for all $i \in U$ or $x_i \in B_i, y_i \in A_i$ for all $i \in U$. Therefore, from the definition of operations $\sqcap, \sqcup$ we get $\{\boldsymbol{x}^U \sqcap \boldsymbol{y}^U, \boldsymbol{x}^U \sqcup \boldsymbol{y}^U\} = \{\boldsymbol{x}^U, \boldsymbol{y}^U\}$. Also, we have $\boldsymbol{x} \sqcap^\circ \boldsymbol{y}, \boldsymbol{x} \sqcup^\circ \boldsymbol{y} \in \texttt{dom} f$, so Lemma 7.3(c) gives $\{\boldsymbol{x}^U \sqcap^\circ \boldsymbol{y}^U, \boldsymbol{x}^U \sqcup^\circ \boldsymbol{y}^U\} = \{\boldsymbol{x}^U, \boldsymbol{y}^U\}$.

If $|\Delta(\boldsymbol{x}, \boldsymbol{y})| = 0$ then $\{\boldsymbol{x} \sqcap \boldsymbol{y}, \boldsymbol{x} \sqcup \boldsymbol{y}\} = \{\boldsymbol{x}, \boldsymbol{y}\}$ and so the claim holds trivially. Let us assume that $\Delta(\boldsymbol{x}, \boldsymbol{y}) = \{j\}$. We will write $\boldsymbol{x} = (\boldsymbol{x}^U, x_j, \boldsymbol{z})$ and $\boldsymbol{y} = (\boldsymbol{y}^U, y_j, \boldsymbol{z})$ where $\boldsymbol{z} = \boldsymbol{x}^{\overline{U}-\{j\}} = \boldsymbol{y}^{\overline{U}-\{j\}}$. Denote $\boldsymbol{z}^{01} = (\boldsymbol{x}^U, y_j, \boldsymbol{z})$ and $\boldsymbol{z}^{10} = (\boldsymbol{y}^U, x_j, \boldsymbol{z})$. Clearly, we have either $\{\boldsymbol{x} \sqcap \boldsymbol{y}, \boldsymbol{x} \sqcup \boldsymbol{y}\} = \{\boldsymbol{x}, \boldsymbol{y}\}$ or $\{\boldsymbol{x} \sqcap \boldsymbol{y}, \boldsymbol{x} \sqcup \boldsymbol{y}\} = \{\boldsymbol{z}^{01}, \boldsymbol{z}^{10}\}$. We can assume that the latter condition holds, otherwise (38) is a trivial equality. We claim that $\boldsymbol{z}^{01}, \boldsymbol{z}^{10} \in \texttt{dom} f$. Indeed, let us show the first claim (the second one is analogous). Labelling $\boldsymbol{z}^{01}$ differs from $\boldsymbol{x} \in \texttt{dom} f$ only at node j; thus, it suffices to show that $(z^{01}_s, z^{01}_j) \equiv (x_s, y_j) \in \rho_{sj}$ for all $s \in V - \{j\}$, and then use (36). For $s \in U$ the fact $(x_s, y_j) \in \rho_{sj}$ follows from Lemma 7.3(d), and for $s \notin U$ the fact $(x_s, y_j) \equiv (y_s, y_j) \in \rho_{sj}$ holds since $\boldsymbol{y} \in \texttt{dom} f$.

We also observe that $(x_i, y_i) \in \overline{M}$ for all $i \in U$ by Lemma 7.3(b). We now consider two possible cases:

Case 1 $\{x_j, y_j\} \in M_j$, so $\sqcap^\circ_j, \sqcup^\circ_j$ are commutative on $\{x_j, y_j\}$. Thus, we must have either $\{\boldsymbol{x} \sqcap^\circ \boldsymbol{y}, \boldsymbol{x} \sqcup^\circ \boldsymbol{y}\} = \{\boldsymbol{z}^{01}, \boldsymbol{z}^{10}\}$ or $\{\boldsymbol{y} \sqcap^\circ \boldsymbol{x}, \boldsymbol{y} \sqcup^\circ \boldsymbol{x}\} = \{\boldsymbol{z}^{01}, \boldsymbol{z}^{10}\}$. Using the fact that $\langle \sqcap^\circ, \sqcup^\circ \rangle$ is a multimorphism of f, we get in each case the desired inequality:

$$f(\boldsymbol{z}^{01}) + f(\boldsymbol{z}^{01}) \leq f(\boldsymbol{x}) + f(\boldsymbol{y})$$

Case 2 $\{x_j, y_j\} \in \overline{M}_j$. It can be checked that applying operations $\langle \texttt{Mj}_1, \texttt{Mj}_2, \texttt{Mn}_3 \rangle$ to $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}^{01})$ gives $(\boldsymbol{z}^{01}, \boldsymbol{z}^{01}, \boldsymbol{z}^{10})$, therefore

$$f(\boldsymbol{z}^{01}) + f(\boldsymbol{z}^{01}) + f(\boldsymbol{z}^{10}) \leq f(\boldsymbol{x}) + f(\boldsymbol{y}) + f(\boldsymbol{z}^{01})$$

which is equivalent to (38). □

PROPOSITION 7.8. *If $\boldsymbol{x}, \boldsymbol{y} \in \texttt{dom} f$ and $\delta(\boldsymbol{x}, \boldsymbol{y}) = 1$ then either $\delta(\boldsymbol{x} \sqcup \boldsymbol{y}, \boldsymbol{y}) = 0$ or $\delta(\boldsymbol{x}, \boldsymbol{x} \sqcup \boldsymbol{y}) = 0$.*

PROOF. Using the same argument as in the proof of Lemma 7.7 we conclude that $\{\boldsymbol{x}^U \sqcap \boldsymbol{y}^U, \boldsymbol{x}^U \sqcup \boldsymbol{y}^U\} = \{\boldsymbol{x}^U, \boldsymbol{y}^U\}$. If $\boldsymbol{x}^U \sqcup \boldsymbol{y}^U = \boldsymbol{y}^U$ then $\delta(\boldsymbol{x} \sqcup \boldsymbol{y}, \boldsymbol{y}) = 0$, and if $\boldsymbol{x}^U \sqcup \boldsymbol{y}^U = \boldsymbol{x}^U$ then $\delta(\boldsymbol{x}, \boldsymbol{x} \sqcup \boldsymbol{y}) = 0$. □

We now proceed with the induction argument. Suppose that $\Delta(\boldsymbol{x},\boldsymbol{y}) \geq 2$ and $\delta(\boldsymbol{x},\boldsymbol{y}) = 1$. Denote

$$X = \{i \in \Delta(\boldsymbol{x},\boldsymbol{y}) \mid (x_i \sqcap y_i, x_i \sqcup y_i) = (x_i, y_i)\}$$
$$Y = \{i \in \Delta(\boldsymbol{x},\boldsymbol{y}) \mid (x_i \sqcap y_i, x_i \sqcup y_i) = (y_i, x_i)\}$$

We have $|X \cup Y| \geq 2$, so by Proposition 7.8 at least one of the following holds:
(1) $|X| \geq 2$;
(2) $|Y| \geq 2$;
(3) $|X| = |Y| = 1$, $\delta(\boldsymbol{x} \sqcup \boldsymbol{y}, \boldsymbol{y}) = 0$;
(4) $|X| = |Y| = 1$, $\delta(\boldsymbol{x}, \boldsymbol{x} \sqcup \boldsymbol{y}) = 0$.
These cases are analysed below.

Cases 1,3 It can be checked that $(\boldsymbol{x} \sqcup \boldsymbol{y}) \sqcap \boldsymbol{y} = \boldsymbol{y}$. Therefore, if we define $\boldsymbol{x}' = \boldsymbol{x} \sqcup \boldsymbol{y}$, $\boldsymbol{y}' = \boldsymbol{y}$ then the following identities hold:

$$\boldsymbol{x} \sqcap \boldsymbol{y}' = \boldsymbol{x} \sqcap \boldsymbol{y} \qquad \boldsymbol{x} \sqcup \boldsymbol{y}' = \boldsymbol{x}' \qquad \boldsymbol{x}' \sqcap \boldsymbol{y} = \boldsymbol{y}' \qquad \boldsymbol{x}' \sqcup \boldsymbol{y} = \boldsymbol{x} \sqcup \boldsymbol{y} \tag{39}$$

Let us select node $s \in X$ and modify $\boldsymbol{x}', \boldsymbol{y}'$ by setting $(x'_s, y'_s) = (x_s, x_s)$. It can be checked that (39) still holds. We have

— $(\boldsymbol{x}, \boldsymbol{y}') \prec (\boldsymbol{x}, \boldsymbol{y})$ since $\Delta(\boldsymbol{x}, \boldsymbol{y}') = \Delta(\boldsymbol{x}, \boldsymbol{y}) - \{s\}$, and
— $(\boldsymbol{x}', \boldsymbol{y}) \prec (\boldsymbol{x}, \boldsymbol{y})$ since $\Delta(\boldsymbol{x}', \boldsymbol{y}) = \Delta(\boldsymbol{x}, \boldsymbol{y}) - (X - \{s\})$; if $X - \{s\}$ is empty (i.e. Case 3 holds) then $\delta(\boldsymbol{x}', \boldsymbol{y}) = 0 < \delta(\boldsymbol{x}, \boldsymbol{y}) = 1$. Indeed, we have $\delta(\boldsymbol{x} \sqcup \boldsymbol{y}, \boldsymbol{y}) = 0$ (by the assumption of Case 3) and $(\boldsymbol{x} \sqcup \boldsymbol{y})^U = (\boldsymbol{x}')^U$ (since $s \notin U$), and therefore $\delta(\boldsymbol{x}', \boldsymbol{y}) = 0$.

Thus, by the induction hypothesis

$$f(\boldsymbol{x} \sqcap \boldsymbol{y}) + f(\boldsymbol{x}') \leq f(\boldsymbol{x}) + f(\boldsymbol{y}') \tag{40}$$

assuming that $\boldsymbol{y}' \in \mathtt{dom} f$, and

$$f(\boldsymbol{y}') + f(\boldsymbol{x} \sqcup \boldsymbol{y}) \leq f(\boldsymbol{x}') + f(\boldsymbol{y}) \tag{41}$$

assuming that $\boldsymbol{x}' \in \mathtt{dom} f$. If $\boldsymbol{y}' \in \mathtt{dom} f$ then Inequality (40) implies that $\boldsymbol{x}' \in \mathtt{dom} f$, and the claim then follows from summing (40) and (41). We now assume that $\boldsymbol{y}' \notin \mathtt{dom} f$; Inequality (41) then implies that $\boldsymbol{x}' \notin \mathtt{dom} f$.

Assume for simplicity of notation that s corresponds to the first argument of f. Define instance $\hat{\mathcal{I}}$ with the set of nodes $\hat{V} = V - \{s\}$ and cost function

$$g(\boldsymbol{z}) = \min_{a \in D_s} \{u(a) + f(a, \boldsymbol{z})\} \qquad \forall \boldsymbol{z} \in \hat{\mathcal{D}} \equiv \times_{i \in \hat{V}} D_i$$

where $u(a)$ is the following unary cost function: $u(x_s) = 0$, $u(y_s) = C$ and $u(a) = 2C$ for $a \in D - \{x_s, y_s\}$. Here C is a sufficiently large constant, namely $C > f(\boldsymbol{x}) + f(\boldsymbol{y})$. We denote $\langle \hat{\sqcap}^\circ, \hat{\sqcup}^\circ \rangle$ and $\langle \hat{\sqcap}, \hat{\sqcup} \rangle$ to be the restrictions of respectively $\langle \sqcap^\circ, \sqcup^\circ \rangle$ and $\langle \sqcap, \sqcup \rangle$ to $\hat{V}$.

LEMMA 7.9. *If $\boldsymbol{u}, \boldsymbol{v} \in \mathtt{dom} g$ and $(\boldsymbol{u}, \boldsymbol{v}) \prec (\boldsymbol{x}, \boldsymbol{y})$ then $g(\boldsymbol{u} \mathbin{\hat{\sqcap}} \boldsymbol{v}) + g(\boldsymbol{u} \mathbin{\hat{\sqcup}} \boldsymbol{v}) \leq g(\boldsymbol{u}) + g(\boldsymbol{v})$ (assuming that the induction hypothesis holds).*

PROOF. It is straightforward to check that unary relations $D_i, i \in \hat{V}$ and binary relations $\rho_{ij}, i, j \in \hat{V}, i \neq j$ are the unique arc- and path-consistent relations for g, i.e.

$$\rho_i = \{z_i \mid \boldsymbol{z} \in \mathtt{dom} g\} \quad \forall i \in \hat{V}\ , \qquad \rho_{ij} = \{(z_i, z_j) \mid \boldsymbol{z} \in \mathtt{dom} g\} \quad \forall i, j \in \hat{V}, i \neq j$$

This implies that set $U \subseteq \hat{V}$ and sets A_i, B_i for $i \in U$ satisfy conditions (a-d) of Lemma 7.3 for instance $\hat{\mathcal{I}}$. The appropriate restrictions of $\langle \sqcap^\circ, \sqcup^\circ \rangle$ and $\langle \mathtt{Mj}_1, \mathtt{Mj}_2, \mathtt{Mn}_3 \rangle$ are multimorphisms of functions u (since they are conservative) and f (by assumption), therefore they are also multimorphisms of g. Furthermore, if the modification in

Stage 2 had been applied to instance $\hat{\mathcal{I}}$ and sets U, A_i, B_i then the pair $\langle \hat{\sqcap}^\circ, \hat{\sqcup}^\circ \rangle$ would be changed to the pair $\langle \hat{\sqcap}, \hat{\sqcup} \rangle$.

Thus, the conditions stated in the first paragraph of Section 7.1.3 hold for instance $\hat{\mathcal{I}}$, and so the induction hypothesis applies. The lemma is proved. □

Let $\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{x}}', \hat{\boldsymbol{y}}'$ be restrictions of respectively $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{x}', \boldsymbol{y}'$ to $\hat{V}$. We can write

$$\begin{aligned} g(\hat{\boldsymbol{y}}) = g(\hat{\boldsymbol{y}}') &= u(y_s) + f(y_s, \hat{\boldsymbol{y}}) = f(\boldsymbol{y}) + C \qquad \text{(since } (x_s, \hat{\boldsymbol{y}}) = \boldsymbol{y}' \notin \mathtt{dom} f\text{)} \\ g(\hat{\boldsymbol{x}}) &= f(x_s, \hat{\boldsymbol{x}}) = f(\boldsymbol{x}) \end{aligned}$$

We have $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}) \prec (\boldsymbol{x}, \boldsymbol{y})$ (since $|\Delta(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})| < |\Delta(\boldsymbol{x}, \boldsymbol{y})|$), so Lemma 7.9 gives

$$g(\hat{\boldsymbol{x}} \hat{\sqcap} \hat{\boldsymbol{y}}) + g(\hat{\boldsymbol{x}} \hat{\sqcup} \hat{\boldsymbol{y}}) \le g(\hat{\boldsymbol{x}}) + g(\hat{\boldsymbol{y}}) = f(\boldsymbol{x}) + f(\boldsymbol{y}) + C \tag{42}$$

We have $g(\hat{\boldsymbol{x}} \hat{\sqcup} \hat{\boldsymbol{y}}) < 2C$, so we must have either $g(\hat{\boldsymbol{x}} \hat{\sqcup} \hat{\boldsymbol{y}}) = f(x_s, \hat{\boldsymbol{x}} \hat{\sqcup} \hat{\boldsymbol{y}})$ or $g(\hat{\boldsymbol{x}} \hat{\sqcup} \hat{\boldsymbol{y}}) = f(y_s, \hat{\boldsymbol{x}} \hat{\sqcup} \hat{\boldsymbol{y}}) + C = f(\boldsymbol{x} \sqcup \boldsymbol{y}) + C$. The former case is impossible since $(x_s, \hat{\boldsymbol{x}} \hat{\sqcup} \hat{\boldsymbol{y}}) = \boldsymbol{x}' \notin \mathtt{dom} f$, so $g(\hat{\boldsymbol{x}} \hat{\sqcup} \hat{\boldsymbol{y}}) = f(\boldsymbol{x} \sqcup \boldsymbol{y}) + C$. Combining it with (42) gives

$$g(\hat{\boldsymbol{x}} \hat{\sqcap} \hat{\boldsymbol{y}}) + f(\boldsymbol{x} \sqcup \boldsymbol{y}) \le f(\boldsymbol{x}) + f(\boldsymbol{y}) \tag{43}$$

This implies that $g(\hat{\boldsymbol{x}} \hat{\sqcap} \hat{\boldsymbol{y}}) < C$, so we must have $g(\hat{\boldsymbol{x}} \hat{\sqcap} \hat{\boldsymbol{y}}) = f(x_s, \hat{\boldsymbol{x}} \hat{\sqcap} \hat{\boldsymbol{y}}) = f(\boldsymbol{x} \sqcap \boldsymbol{y})$. Thus, (43) is equivalent to (38).

Cases 2,4 It can be checked that $\boldsymbol{x} \sqcap (\boldsymbol{x} \sqcup \boldsymbol{y}) = \boldsymbol{x}$. Therefore, if we define $\boldsymbol{x}' = \boldsymbol{x}$, $\boldsymbol{y}' = \boldsymbol{x} \sqcup \boldsymbol{y}$ then the following identities hold:

$$\boldsymbol{x}' \sqcap \boldsymbol{y} = \boldsymbol{x} \sqcap \boldsymbol{y} \qquad \boldsymbol{x}' \sqcup \boldsymbol{y} = \boldsymbol{y}' \qquad \boldsymbol{x} \sqcap \boldsymbol{y}' = \boldsymbol{x}' \qquad \boldsymbol{x} \sqcup \boldsymbol{y}' = \boldsymbol{x} \sqcup \boldsymbol{y} \tag{44}$$

Let us select node $s \in Y$ and modify $\boldsymbol{x}', \boldsymbol{y}'$ by setting $(x'_s, y'_s) = (y_s, y_s)$. It can be checked that (44) still holds. We have

— $(\boldsymbol{x}', \boldsymbol{y}) \prec (\boldsymbol{x}, \boldsymbol{y})$ since $\Delta(\boldsymbol{x}', \boldsymbol{y}) = \Delta(\boldsymbol{x}, \boldsymbol{y}) - \{s\}$, and
— $(\boldsymbol{x}, \boldsymbol{y}') \prec (\boldsymbol{x}, \boldsymbol{y})$ since $\Delta(\boldsymbol{x}, \boldsymbol{y}') = \Delta(\boldsymbol{x}, \boldsymbol{y}) - (Y - \{s\})$; if $Y - \{s\}$ is empty (i.e. case 4 holds) then $\delta(\boldsymbol{x}, \boldsymbol{y}') = 0 < \delta(\boldsymbol{x}, \boldsymbol{y}) = 1$. Indeed, we have $\delta(\boldsymbol{x}, \boldsymbol{x} \sqcup \boldsymbol{y}) = 0$ (by the assumption of Case 4) and $(\boldsymbol{x} \sqcup \boldsymbol{y})^U = (\boldsymbol{y}')^U$ (since $s \notin U$), and therefore $\delta(\boldsymbol{x}, \boldsymbol{y}') = 0$.

Thus, by the induction hypothesis

$$f(\boldsymbol{x} \sqcap \boldsymbol{y}) + f(\boldsymbol{y}') \le f(\boldsymbol{x}') + f(\boldsymbol{y}) \tag{45}$$

assuming that $\boldsymbol{x}' \in \mathtt{dom} f$, and

$$f(\boldsymbol{x}') + f(\boldsymbol{x} \sqcup \boldsymbol{y}) \le f(\boldsymbol{x}) + f(\boldsymbol{y}') \tag{46}$$

assuming that $\boldsymbol{y}' \in \mathtt{dom} f$. The rest of the proof proceeds analogously to the proof for the Cases 1,3, distinguishing whether or not $\boldsymbol{x}'$ belongs to $\mathtt{dom} f$.

Acknowledgements

We thank the anonymous referees for careful proofreading of the manuscript and catching typos and mistakes.

REFERENCES

BAKER, K. AND PIXLEY, A. 1975. Polynomial Interpolation and the Chinese Remainder Theorem. *Mathematische Zeitschrift 143,* 2, 165–174.

BARTO, L. 2011. The dichotomy for conservative constraint satisfaction problems revisited. In *Proceedings of the 26th IEEE Symposium on Logic in Computer Science (LICS'11)*. IEEE Computer Society, 301–310.

BARTO, L. AND KOZIK, M. 2009. Constraint Satisfaction Problems of Bounded Width. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS'09)*. IEEE Computer Society, 461–471.

BARTO, L. AND KOZIK, M. 2010. New Conditions for Taylor Varieties and CSP. In *Proceedings of the 25th IEEE Symposium on Logic in Computer Science (LICS'10)*. 100–109.

BARTO, L., KOZIK, M., MARÓTI, M., AND NIVEN, T. 2009a. CSP dichotomy for special triads. *Proceedings of the American Mathematical Society 137,* 9, 2921–2934.

BARTO, L., KOZIK, M., AND NIVEN, T. 2009b. The CSP dichotomy holds for digraphs with no sources and no sinks (a positive answer to a conjecture of Bang-Jensen and Hell). *SIAM Journal on Computing 38,* 5, 1782–1802.

BERMAN, J., IDZIAK, P., MARKOVIĆ, P., MCKENZIE, R., VALERIOTE, M., AND WILLARD, R. 2010. Varieties with few subalgebras of powers. *Transactions of the American Mathematical Society 362,* 3, 1445–1473.

BERTELÉ, U. AND BRIOSHI, F. 1972. *Nonserial dynamic programming*. Academic Press.

BISTARELLI, S., MONTANARI, U., AND ROSSI, F. 1997. Semiring-based Constraint Satisfaction and Optimisation. *Journal of the ACM 44,* 2, 201–236.

BULATOV, A. 2006. A dichotomy theorem for constraint satisfaction problems on a 3-element set. *Journal of the ACM 53,* 1, 66–120.

BULATOV, A. AND JEAVONS, P. 2003. An Algebraic Approach to Multi-sorted Constraints. In *Proceedings of the 9th International Conference on Principles and Practice of Constraint Programming (CP'03)*. Lecture Notes in Computer Science Series, vol. 2833. Springer, 183–198.

BULATOV, A., KROKHIN, A., AND JEAVONS, P. 2005. Classifying the Complexity of Constraints using Finite Algebras. *SIAM Journal on Computing 34,* 3, 720–742.

BULATOV, A. A. 2003. Tractable Conservative Constraint Satisfaction Problems. In *Proceedings of the 18th IEEE Symposium on Logic in Computer Science (LICS'03)*. 321–330.

BULATOV, A. A. 2011. Complexity of conservative constraint satisfaction problems. *ACM Transactions on Computational Logic 12,* 4. Article 24.

CHEN, H. 2006. A rendezvous of logic, complexity, and algebra. *SIGACT News 37,* 4, 85–114.

CHEN, X., DYER, M. E., GOLDBERG, L. A., JERRUM, M., LU, P., MCQUILLAN, C., AND RICHERBY, D. 2012. The complexity of approximating conservative counting CSPs. In *Proceedings of the 30th International Symposium on Theoretical Aspects of Computer Science (STACS'13)*.

COHEN, D. A., COOPER, M. C., AND JEAVONS, P. G. 2008. Generalising submodularity and Horn clauses: Tractable optimization problems defined by tournament pair multimorphisms. *Theoretical Computer Science 401,* 1-3, 36–51.

COHEN, D. A., COOPER, M. C., JEAVONS, P. G., AND KROKHIN, A. A. 2006. The Complexity of Soft Constraint Satisfaction. *Artificial Intelligence 170,* 11, 983–1016.

COOPER, M. 1989. An Optimal k-consistency Algorithm. *Artificial Intelligence 41,* 1, 89–95.

COOPER, M. C. AND ŽIVNÝ, S. 2011. Hybrid tractability of valued constraint problems. *Artificial Intelligence 175,* 9-10, 1555–1569.

COOPER, M. C. AND ŽIVNÝ, S. 2012. Tractable triangles and cross-free convexity in discrete optimisation. *Journal of Artificial Intelligence Research 44*, 455–490.

CREIGNOU, N. 1995. A dichotomy theorem for maximum generalized satisfiability problems. *Journal of Computer and System Sciences 51,* 3, 511–522.

CREIGNOU, N., KHANNA, S., AND SUDAN, M. 2001. *Complexity Classification of Boolean Constraint Satisfaction Problems*. SIAM Monographs on Discrete Mathematics and Applications Series, vol. 7. SIAM.

CREIGNOU, N., KOLAITIS, P. G., AND ZANUTTINI, B. 2008. Structure identification of Boolean relations and plain bases for co-clones. *Journal of Computer and System Sciences 74,* 7, 1103–1115.

DALMAU, V., KOLAITIS, P. G., AND VARDI, M. Y. 2002. Constraint Satisfaction, Bounded Treewidth, and Finite-Variable Logics. In *Proceedings of the 8th International Conference on Principles and Practice of Constraint Programming (CP'02)*. Lecture Notes in Computer Science Series, vol. 2470. Springer, 310–326.

DECHTER, R. 2003. *Constraint Processing*. Morgan Kaufmann.

DECHTER, R. AND PEARL, J. 1988. Network-based Heuristics for Constraint Satisfaction Problems. *Artificial Intelligence 34,* 1, 1–38.

DECHTER, R. AND PEARL, J. 1992. Structure Identification in Relational Data. *Artificial Intelligence 58,* 1-3, 237–270.

DEINEKO, V., JONSSON, P., KLASSON, M., AND KROKHIN, A. 2008. The approximability of Max CSP with fixed-value constraints. *Journal of the ACM 55,* 4. Article 16.

FEDER, T. AND VARDI, M. Y. 1998. The Computational Structure of Monotone Monadic SNP and Constraint Satisfaction: A Study through Datalog and Group Theory. *SIAM Journal on Computing 28,* 1, 57–104.

GAREY, M. R. AND JOHNSON, D. S. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman.

GOTTLOB, G., GRECO, G., AND SCARCELLO, F. 2009. Tractable Optimization Problems through Hypergraph-Based Structural Restrictions. In *Proceedings of the 36th International Colloquium on Automata, Languages and Programming (ICALP'09), Part II*. Lecture Notes in Computer Science Series, vol. 5556. Springer, 16–30.

GROHE, M. 2007. The complexity of homomorphism and constraint satisfaction problems seen from the other side. *Journal of the ACM 54,* 1, 1–24.

GROHE, M. AND MARX, D. 2006. Constraint solving via fractional edge covers. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'06)*. 289–298.

GUTIN, G., RAFIEY, A., YEO, A., AND TSO, M. 2006. Level of Repair Analysis and Minimum Cost Homomorphisms of Graphs. *Discrete Applied Mathematics 154,* 6, 881–889.

HELL, P. AND NEŠETŘIL, J. 2008. Colouring, constraint satisfaction, and complexity. *Computer Science Review 2,* 3, 143–163.

IDZIAK, P. M., MARKOVIC, P., MCKENZIE, R., VALERIOTE, M., AND WILLARD, R. 2010. Tractability and learnability arising from algebras with few subpowers. *SIAM Journal on Computing 39,* 7, 3023–3037.

IWATA, S., FLEISCHER, L., AND FUJISHIGE, S. 2001. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *Journal of the ACM 48,* 4, 761–777.

JEAVONS, P., COHEN, D., AND COOPER, M. C. 1998. Constraints, Consistency and Closure. *Artificial Intelligence 101,* 1–2, 251–265.

JEAVONS, P. G. 1998. On the Algebraic Structure of Combinatorial Problems. *Theoretical Computer Science 200,* 1-2, 185–204.

JEAVONS, P. G., COHEN, D. A., AND GYSSENS, M. 1997. Closure Properties of Constraints. *Journal of the ACM 44,* 4, 527–548.

JONSSON, P., KUIVINEN, F., AND THAPPER, J. 2011. Min CSP on Four Elements: Moving Beyond Submodularity. In *Proceedings of the 17th International Conference on Principles and Practice of Constraint Programming (CP'11)*. Lecture Notes in Computer Science Series, vol. 6876. Springer, 438–453.

KHANNA, S., SUDAN, M., TREVISAN, L., AND WILLIAMSON, D. 2001. The approximability of constraint satisfaction problems. *SIAM Journal on Computing 30,* 6, 1863–1920.

KOLAITIS, P. G. AND VARDI, M. Y. 2000. Conjunctive-Query Containment and Constraint Satisfaction. *Journal of Computer and System Sciences 61,* 2, 302–332.

KOLMOGOROV, V. 2012. The power of linear programming for valued CSPs: a constructive characterization. Tech. rep. July. arXiv:1207.7213.

KROKHIN, A. AND LAROSE, B. 2008. Maximizing Supermodular Functions on Product Lattices, with Application to Maximum Constraint Satisfaction. *SIAM Journal on Discrete Mathematics 22,* 1, 312–328.

KUN, G. AND SZEGEDY, M. 2009. A New Line of Attack on the Dichotomy Conjecture. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC'09)*. 725–734.

LAURITZEN, S. L. 1996. *Graphical Models*. Oxford University Press.

MARX, D. 2010a. Approximating fractional hypertree width. *ACM Transactions on Algorithms 6,* 2. Article 29.

MARX, D. 2010b. Tractable hypergraph properties for constraint satisfaction and conjunctive queries. In *Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC'10)*. 735–744.

MONTANARI, U. 1974. Networks of Constraints: Fundamental properties and applications to picture processing. *Information Sciences 7*, 95–132.

RAGHAVENDRA, P. 2008. Optimal algorithms and inapproximability results for every CSP? In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC'08)*. 245–254.

RAGHAVENDRA, P. AND STEURER, D. 2009. How to Round Any CSP. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS'09)*. 586–594.

SCHAEFER, T. J. 1978. The Complexity of Satisfiability Problems. In *Proceedings of the 10th Annual ACM Symposium on Theory of Computing (STOC'78)*. ACM, 216–226.

SCHIEX, T., FARGIER, H., AND VERFAILLIE, G. 1995. Valued Constraint Satisfaction Problems: Hard and Easy Problems. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI'95)*. 631–637.

SCHRIJVER, A. 2000. A Combinatorial Algorithm Minimizing Submodular Functions in Strongly Polynomial Time. *Journal of Combinatorial Theory, Series B 80,* 2, 346–355.

TAKHANOV, R. 2010a. A Dichotomy Theorem for the General Minimum Cost Homomorphism Problem. In *Proceedings of the 27th International Symposium on Theoretical Aspects of Computer Science (STACS'10)*. 657–668.

TAKHANOV, R. 2010b. Extensions of the Minimum Cost Homomorphism Problem. In *Proceedings of the 16th International Computing and Combinatorics Conference (COCOON'10)*. Lecture Notes in Computer Science Series, vol. 6196. Springer, 328–337.

THAPPER, J. AND ŽIVNÝ, S. 2012. The power of linear programming for valued CSPs. In *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS'12)*. IEEE.

THAPPER, J. AND ŽIVNÝ, S. 2013. The complexity of finite-valued CSPs. To appear in *Proceedings of the 45th ACM Symposium on the Theory of Computing (STOC'13)*.

ULLMAN, J. D. 1989. *Principles of Database and Knowledge-Base Systems*. Vol. 1 & 2. Computer Science Press.

WAINWRIGHT, M. J. AND JORDAN, M. I. 2008. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning 1,* 1-2, 1–305.

Received October 2011; revised October 2012; accepted January 2013