

# Constraint Satisfaction Problems Solvable by Local Consistency Methods

LIBOR BARTO, McMaster University and Charles University in Prague  
MARCIN KOZIK, Jagiellonian University

We prove that constraint satisfaction problems without the ability to count are solvable by the local consistency checking algorithm. This settles three (equivalent) conjectures: Feder–Vardi [SICOMP’98], Bulatov [LICS’04] and Larose–Zádori [AU’07].

Categories and Subject Descriptors: F.1.3 [Computation by Abstract Devices]: Complexity Measures and Classes—*Relations among complexity classes*; F.2.2 [Analysis of Algorithms and Problem Complexity]: Nonnumerical Algorithms and Problems—*Computations on discrete structures*; F.4.1 [Mathematical Logic and Formal Languages]: Mathematical Logic—*Logic and constraint programming*

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Constraint satisfaction problem, local consistency checking

## ACM Reference Format:

Barto, L. and Kozik, M. 2014. Constraint Satisfaction Problems solvable by local consistency methods. *J. ACM* 61, 1, Article 3 (January 2014), 19 pages.  
DOI: <http://dx.doi.org/10.1145/2556646>

## 1. INTRODUCTION

The Constraint Satisfaction Problem (CSP) asks if, given variables and constraints, there is an assignment such that all the constraints are satisfied. Each constraint is presented by a list of admissible evaluations for a few variables. Deciding if such an assignment exists is NP-complete.

In a seminal paper [Feder and Vardi 1998], a concept of nonuniform CSPs was introduced. A nonuniform CSPs restricts the admissible instances by limiting the allowed constraints – the language of the CSP. Thus, each constraint language defines a computational problem of some complexity. The most important question in this area is the Dichotomy Conjecture of Feder and Vardi [1998] postulating that each constraint language defines a problem which is NP-complete or solvable in polynomial time.

Two major algorithmic principles solve CSPs for wide classes of constraint languages. One of them, generalizations of Gaussian elimination [Berman et al. 2009; Bulatov 2006a; Bulatov and Dalmau 2006; Idziak et al. 2007], is beyond the scope of

---

A preliminary version of this article appeared in *Proceedings of the 50th Symposium on Foundations of Computer Science (FOCS 09)* [Barto and Kozik 2009b].

L. Barto was supported by the Grant Agency of the Czech Republic, grant 201/09/P223 and by the Ministry of Education of the Czech Republic, grant MSM 0021620839. M. Kozik was partially supported by National Science Center grant DEC-2011/01/B/ST6/01006.

Authors’ addresses: L. Barto, Department of Algebra, Faculty of Mathematics and Physics, Charles University in Prague, Sokolovská 83, 186 75 Praha 8, Czech Republic; M. Kozik (corresponding author), Theoretical Computer Science Department, Faculty of Mathematics and Computer Science, Jagiellonian University, ul. Prof. St. Łojasiewicza 6, 30-348 Kraków, Poland; email: [marcin.kozik@uj.edu.pl](mailto:marcin.kozik@uj.edu.pl).

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or [permissions@acm.org](mailto:permissions@acm.org).

© 2014 ACM 0004-5411/2014/01-ART3 \$15.00

DOI: <http://dx.doi.org/10.1145/2556646>

this article. The other algorithm, called the local consistency checking algorithm, determines whether an instance has a consistent set of local solutions. For a wide variety of constraint languages the existence of such a set implies a global solution. These constraint languages are said to have bounded width [Feder and Vardi 1998].

The CSPs of bounded width appear naturally in different approaches to the subject. The class can be equivalently described as the CSPs having complements recognizable by a Datalog program with a goal predicate, using pebble games, or as problems with a bounded tree-width duality.

An obstruction to having bounded width was recognized in Feder and Vardi [1998] and called the ability to count. A constraint language has this property if it can, in some sense, simulate relations encoding linear equations over a finite field. Feder and Vardi [1998] proved that a bounded width constraint language cannot have the ability to count and conjectured that lack of this ability implies bounded width. A different conjecture characterizing bounded width was stated by Bulatov [2004b] by means of his very successful [Bulatov 2003, 2004a, 2006c] technique based on studying the local structure of the constraint language. Finally, Larose and Zádori [2007] used Tame Congruence Theory [Hobby and McKenzie 1988] to propose a characterization using the types in the variety associated with the constraint language. These three conjectures were shown to be equivalent in Larose et al. [2009] and Bulatov and Valeriote [2008].

The progress toward proving the bounded width conjecture(s) [Barto and Kozik 2009a; Bulatov 2006b; Carvalho et al. 2009; Kiss and Valeriote 2007] culminated in its confirmation announced by the authors [Barto and Kozik 2009b] and independently by Bulatov [2009]. This article contains a full proof of the result announced in Barto and Kozik [2009b] presented in a self-contained way (modulo basic algebraic and computational notions). The result is stated in a slightly more general way than in Barto and Kozik [2009b]. This change is motivated by recent developments in robust approximability of CSPs [Barto and Kozik 2012].<sup>1</sup> The presented proof is also more elementary than Barto and Kozik [2009b] in the sense that we do not require the algebraic results from Maróti and McKenzie [2008]. A generalization of our result in a different direction appears in Barto [2013].

The material presented in this article is divided in the following way. The next section contains preliminary information on CSPs, universal algebra and the connection between constraint languages and algebras. In Section 3, we introduce the local consistency checking algorithm and use it to define CSPs of bounded width. Section 4 states the three conjectures. Sections 5 and 6 contain reductions of the conjecture of Larose and Zádori to the case of special instances with binary constraints only. In Section 7, we introduce tools and concepts of universal algebra necessary for the proof. In Section 8, we use these tools to prove the conjecture.

## 2. PRELIMINARIES

The preliminaries are split into three parts.

### 2.1. Constraint Satisfaction Problems

The following definition is standard for nonuniform Constraint Satisfaction Problems.

*Definition 2.1.* An instance of the CSP is a triple  $\mathcal{I} = (V, D, C)$  with  $V$  a finite set of variables,  $D$  a finite domain, and  $C$  a finite list of constraints, where each constraint is

<sup>1</sup>The characterization of CSPs admitting robust approximation, given in Barto and Kozik [2012], requires a result stronger than the one provided in Barto and Kozik [2009b].

a pair  $C = (S, R)$  with  $S$  a tuple of variables of length  $k$ , called the *scope* of  $C$ , and  $R$  a  $k$ -ary relation on  $D$  (i.e., a subset of  $D^k$ ), called the *constraint relation* of  $C$ .

An *assignment* for  $\mathcal{I}$  is a mapping  $F : V \rightarrow D$ . We say that  $F$  *satisfies* a constraint  $C = (S, R)$  if  $F(S) \in R$  (where  $F$  is applied component-wise). An assignment which satisfies all the constraints of the instance is a *solution*.

A finite set of relations  $\mathbb{D}$  over a common domain  $D$  is called a *constraint language* (the arities of the relations form the *signature* of the language). An *instance* of  $\text{CSP}(\mathbb{D})$  is an instance of the CSP such that all the constraint relations are from  $\mathbb{D}$ . The *decision problem* for  $\text{CSP}(\mathbb{D})$  asks whether an input instance  $\mathcal{I}$  of  $\text{CSP}(\mathbb{D})$  has a solution.

The definition of the CSP does not specify how the constraint relations are given on the input. Note, however, that for a nonuniform problem  $\text{CSP}(\mathbb{D})$ , where  $\mathbb{D}$  is finite, standard representations lead to log-space equivalent problems. We can, for instance, represent relations by listing all their tuples in every constraint (this is how relations are usually represented when the constraint language is not finite), or we can just use names of the relations.

## 2.2. Universal Algebra

The following paragraphs cover most of the algebraic concepts which are used in this paper, for a more exhaustive introduction refer to Burris and Sankappanavar [1981] and Bergman [2011].

An *algebra* denoted in boldface, for example  $\mathbf{A}$ , consists of a set  $A$  (the *universe* of  $\mathbf{A}$ ) and operations (sometimes called the *basic operations*) – functions from finite powers of  $A$  to  $A$ . The symbols and arities of the operations of an algebra form the *signature* of this algebra, for example, algebras  $\mathbf{A}$  and  $\mathbf{B}$  in the same signature containing an  $n$ -ary symbol  $t$  have basic operations  $t^{\mathbf{A}} : A^n \rightarrow A$  and  $t^{\mathbf{B}} : B^n \rightarrow B$  respectively.<sup>2</sup>

A *subuniverse* of  $\mathbf{A}$  is a nonempty set  $B$ , contained in  $A$ , and such that every operation of  $\mathbf{A}$  evaluated on arguments from  $B$  produces a result in  $B$ . An algebra  $\mathbf{B}$  is a *subalgebra* of  $\mathbf{A}$  (denoted by  $\mathbf{B} \leq \mathbf{A}$ ) if  $B$  is a subuniverse of  $\mathbf{A}$  and the operations of  $\mathbf{B}$  are the operations of  $\mathbf{A}$  restricted to  $B$ .

The *product* of algebras  $\mathbf{A}$  and  $\mathbf{B}$  (of the same signature), denoted by  $\mathbf{A} \times \mathbf{B}$ , is the algebra with universe  $A \times B$  and operations computed coordinatewise. A product of more than two, or an infinite number of algebras is defined analogically. A *power* of  $\mathbf{A}$  is a product of copies of  $\mathbf{A}$  and a *subpower* of  $\mathbf{A}$  is a subalgebra of a power of  $\mathbf{A}$ . A subalgebra  $\mathbf{B}$  of a product  $\mathbf{A}_1 \times \cdots \times \mathbf{A}_n$  is called *subdirect* if the projection of  $B$  on each coordinate is full (i.e., the projection on the  $i$ th coordinate is equal to  $A_i$ ), in such a case we write  $\mathbf{B} \leq_{sd} \mathbf{A}_1 \times \cdots \times \mathbf{A}_n$ .

A function  $f$  from  $A$  to  $B$  is a *homomorphism* from  $\mathbf{A}$  to  $\mathbf{B}$  (the signatures of  $\mathbf{A}$  and  $\mathbf{B}$  are identical) if, for every operation  $t$  with arity  $n$ , we have  $f(t^{\mathbf{A}}(a_1, \dots, a_n)) = t^{\mathbf{B}}(f(a_1), \dots, f(a_n))$ , for every choice of  $a_1, \dots, a_n \in A$ .

An equivalence relation  $\alpha$  is a *congruence* of  $\mathbf{A}$  if every operations of  $\mathbf{A}$ , computed on two coordinatewise  $\alpha$ -related tuples, produces a pair in  $\alpha$ . The congruences of an algebra  $\mathbf{A}$  form a lattice:  $\alpha \wedge \beta$  is the largest congruence contained in both  $\alpha$  and  $\beta$ , and  $\alpha \vee \beta$  is the smallest congruence containing both. An algebra with just two (or one) congruences (i.e., the identity congruence and the full congruence) is called *simple*. If  $\alpha$  is a congruence of  $\mathbf{A}$  one can form the *quotient* algebra  $\mathbf{A}/\alpha$ : the universe of  $\mathbf{A}/\alpha$  is  $A/\alpha$  and the operations are derived from the operations on  $\mathbf{A}$  by taking representatives of the congruence classes (since  $\alpha$  is a congruence the operations are well defined). If  $\mathbf{B} \leq \mathbf{A}_1 \times \cdots \times \mathbf{A}_n$ , then the kernel of projection to  $\mathbf{A}_i$  is denoted by  $\pi_i$  and is a

<sup>2</sup>We will skip the superscript, writing  $t$  instead of  $t^{\mathbf{A}}$ , whenever the algebra is clear from the context.

congruences of  $\mathbf{B}$ . If, on the other hand,  $\alpha$  is a congruence of such  $\mathbf{B}$ , then the projection of  $\alpha$  to  $i$ th coordinate is the smallest congruence of  $\mathbf{A}_i$  containing all  $(a_i, b_i)$  such that  $((a_1, \dots, a_n), (b_1, \dots, b_n)) \in \alpha$ .

A *term*  $t$  is a syntactical description of composition of operations in a given signature. For a given algebra  $\mathbf{A}$  (in this signature) a *term operation*  $t^{\mathbf{A}}$  is an operation obtained by composition of basic operation according to  $t$ . A class of algebras is a *variety* if it is closed under taking products, subalgebras and homomorphic images of algebras in the class (quotients of algebras in a variety are also inside the variety). The smallest variety containing an algebra  $\mathbf{A}$  is called the variety generated by  $\mathbf{A}$  and denoted by  $\mathcal{V}(\mathbf{A})$ . By Birkhoff's theorem [Birkhoff 1935], each variety  $\mathcal{V}$  is determined by a set of pairs of terms  $s \approx t$  (called *identities*) in the following way: an algebra  $\mathbf{A}$  is in  $\mathcal{V}$  if and only if  $t^{\mathbf{A}} = s^{\mathbf{A}}$  for each pair of terms from the set.

### 2.3. Polymorphisms

Much of the recent progress on the complexity of the decision problem for CSP was achieved by the algebraic approach. The notion linking relations and operations is at the center of these developments.

*Definition 2.2.* An  $l$ -ary operation  $f$  on  $D$  is a *polymorphism* of a  $k$ -ary relation  $R$  (or  $R$  is *compatible* with  $f$ ), if

$$(f(a_1^1, \dots, a_1^l), f(a_2^1, \dots, a_2^l), \dots, f(a_k^1, \dots, a_k^l)) \in R$$

whenever  $(a_1^1, \dots, a_k^1), (a_1^2, \dots, a_k^2), \dots, (a_1^l, \dots, a_k^l) \in R$ .

An operation on  $D$  is a *polymorphism* of a constraint language  $\mathbb{D}$  (with domain  $D$ ) if it is a polymorphism of every relation in  $\mathbb{D}$ .

The set of all polymorphisms of a constraint language  $\mathbb{D}$  will be denoted by  $\text{Pol}(\mathbb{D})$ . Among the earliest results of the algebraic approach to the CSP is a theorem [Jeavons et al. 1997] stating that adding to  $\mathbb{D}$  relations compatible with all operations of  $\text{Pol}(\mathbb{D})$  produces a CSP computationally equivalent to  $\text{CSP}(\mathbb{D})$ . Earlier results [Bodnarčuk et al. 1969; Geiger 1968] show that these are exactly the relations definable from  $\mathbb{D}$  by pp-formulas – formulas using existential quantification, conjunction and equality. Moreover, adding such relations does not alter the set of polymorphisms of the constraint language.

With each constraint language  $\mathbb{D}$ , an algebra  $\mathbf{D}$  is associated. The algebra  $\mathbf{D}$  has universe  $D$  and the set of operations  $\text{Pol}(\mathbb{D})$ . The relations compatible with all the operations of  $\text{Pol}(\mathbb{D})$  (from the previous paragraph) are, using algebraic terms, the subpowers of  $\mathbf{D}$  (i.e., subalgebras of powers of  $\mathbf{D}$ ).

Further developments in the algebraic approach [Bulatov et al. 2000, 2005; Larose and Tesson 2009] showed that, given  $\mathbb{D}$ , for any algebra  $\mathbf{E}$  in the variety generated by  $\mathbf{D}$ , any constraint language with domain  $E$  and with subpowers of  $\mathbf{E}$  as relations defines a CSP reducible (in log-space) to  $\text{CSP}(\mathbb{D})$ . Therefore, the complexity of the decision problem for  $\text{CSP}(\mathbb{D})$  depends only on the variety generated by  $\mathbf{D}$ , that is, by the result of Birkhoff, on the identities that hold in  $\mathbf{D}$ .

A constraint language  $\mathbb{D}$  is a *core*, if all its unary polymorphisms are bijections. It is clear that, for any constraint language  $\mathbb{D}$ , there is a core constraint language  $\mathbb{D}'$  (the *core of*  $\mathbb{D}$ ) such that  $\text{CSP}(\mathbb{D}) = \text{CSP}(\mathbb{D}')$ . Further, if  $\mathbb{D}'$  is a core one can construct  $\mathbb{D}''$  by adding, for every  $d \in D'$ , the unary constraint relation  $\{d\}$  (this relation allows to fix an evaluation of a variable to  $d$ ). The constraint languages  $\mathbb{D}'$  and  $\mathbb{D}''$  are log-space equivalent [Bulatov et al. 2005]. Any  $t \in \text{Pol}(\mathbb{D}'')$  has the property that  $t(d, \dots, d) = d$  for all  $d \in D$  which means that  $\mathbf{D}$  satisfies the identity  $t(x, \dots, x) \approx x$  for every operation  $t$ . Such algebras are called *idempotent*. Summarizing, every constraint language

**ALGORITHM 1:** The  $(k, l)$ -consistency checking algorithm

---

**Input:** An instance  $\mathcal{I} = (V, D, \mathcal{C})$

```

1  $\mathcal{F} =$  all functions from at most  $l$ -elements subsets of  $V$  into  $D$ ;
2 for  $f \in \mathcal{F}$  do
3   for  $((x_1, \dots, x_n), R) \in \mathcal{C}$  do
4     if  $x_1, \dots, x_n \in \text{dom } f$  and  $(f(x_1), \dots, f(x_n)) \notin R$  then
5        $\mathcal{F} = \mathcal{F} \setminus \{f\}$ ;
6       break;
7 repeat
8   for  $f \in \mathcal{F}$  do
9     foreach  $W$  at most  $l$ -element subset of  $V$  do
10      if
11         $(|\text{dom } f| \leq k, \text{dom } f \subseteq W$  and there is no  $g \in \mathcal{F}$  with  $\text{dom } g = W$  and
12           $g|_{\text{dom } f} = f)$  or
13           $(W \subseteq \text{dom } f$  and  $f|_W \notin \mathcal{F})$ 
14      then
15         $\mathcal{F} = \mathcal{F} \setminus \{f\}$ ;
16        break;                                     // proceed to the next  $f \in \mathcal{F}$ 
17 until  $\mathcal{F}$  was not altered;
18 if  $\mathcal{F} = \emptyset$  then return NO else return YES

```

---

has a computationally equivalent constraint language associated with an idempotent algebra. This fact is essential to the algebraic classification of CSPs.

### 3. PROBLEMS OF BOUNDED WIDTH

In the seminal paper [Feder and Vardi 1998], the class of CSPs of bounded width was introduced. A constraint language  $\mathbb{D}$  has, according to Feder and Vardi [1998], bounded width if the complement of  $\text{CSP}(\mathbb{D})$  can be recognized by a Datalog program. The class can be also described as the CSPs recognizable by certain pebble games or having bounded tree-width duality. In this article, the class of CSPs of bounded width is introduced as the CSPs solvable by the local consistency checking algorithm. A more exhaustive overview of this and other related classes of problems is provided in Bulatov et al. [2008].

The local consistency algorithm, presented as Algorithm 1, is parametrized by two natural numbers  $k \leq l$ . The algorithm starts with the set  $\mathcal{F}$  of all partial assignments of variables (of at most  $l$  variables) into the domain  $D$ . In the first loop (lines 2–6), the assignments which falsify constraints are removed. In the second loop (lines 7–15), an assignment  $f \in \mathcal{F}$  is removed if it falsifies one of two conditions:

- (1) if  $|\text{dom } f| \leq k$ , for any set consisting of at most  $l$  variables and containing  $\text{dom } f$ , there is an assignment in  $\mathcal{F}$  which extends  $f$  to this set (line 11), and
- (2) every restriction of  $f$  to a subset of its domain is in  $\mathcal{F}$  (line 12).

Finally, the algorithm answers *NO* if  $\mathcal{F}$  is empty and *YES* otherwise (note that, at this stage, if  $\mathcal{F}$  contains no functions with domain  $W$ , and  $|W| \leq l$ , then  $\mathcal{F}$  is empty).

Intuitively, the algorithm constructs a set of partial assignments (restricted to at most  $l$  variables) which are consistent on small (at most  $k$  element) sets of variables. If the instance on the input of the algorithm has a global solution, then none of its restrictions can be removed from  $\mathcal{F}$ . Thus, if the local consistency checking algorithm outputs *NO* there is no solution. Constraint languages of bounded width are those, for which the *YES* answer of the algorithm is always correct.



**Definition 3.1.** A constraint language  $\mathbb{D}$  has *width*  $(k, l)$  if  $(k, l)$ -consistency checking correctly decides  $\text{CSP}(\mathbb{D})$ . A constraint language  $\mathbb{D}$  (or the problem  $\text{CSP}(\mathbb{D})$ ) has *bounded width* if it has width  $(k, l)$  for some  $k, l$ .

Let  $\mathbb{D}$  be a constraint language and let  $\mathcal{I}$  be an instance of  $\text{CSP}$  in this language. Let us assume that the  $(k, l)$ -consistency checking algorithm answered *YES* and  $\mathcal{F}$  is the set of partial assignments of variables at the end of the run. Note that, for a fixed set of variables  $\{x_1, \dots, x_n\}$ , all the functions in  $\mathcal{F}$  with domain  $\{x_1, \dots, x_n\}$  can be viewed as a set, say  $E$ , of tuples from  $D^n$  (by putting  $(f(x_1), \dots, f(x_n))$  into  $E$  for each such  $f$ ). It is crucial for the proof, and a part of folklore, that such obtained  $E$  is a subpower of  $\mathbf{D}$ , that is,  $E$  is a relation preserved by all<sup>3</sup> the polymorphisms of  $\mathbb{D}$ .

Finally, if a constraint language  $\mathbb{D}$  has bounded width, then so do the languages defined (in the same way as in the fourth paragraph of Section 2.3) on algebras in the variety generated by  $\mathbf{D}$  [Larose and Zádori 2007]. Therefore, again by Birkhoff's result, the property of having bounded width is determined by identities in the algebra of polymorphisms.

#### 4. THE THREE CONJECTURES

The conjectures in this section are mostly concerned with constraint languages which are cores. In the last paragraph of Section 2.3 we argued that this restriction does not decrease the generality of the statements.

Feder and Vardi [1998] introduced the notion of *ability to count*. A constraint language has the ability to count if it can simulate the set  $\{1, \dots, p\}$  with two relations similar to the relations “sum of  $x, y$  and  $z$  is equal to 1” and “ $x$  is 0” in a cyclic group. The precise definition can be found in Larose et al. [2009]. Constraint languages with the ability to count are on the opposite end of the spectra from those of bounded width (Feder and Vardi [1998] called them languages “in Datalog”). The conjecture of Feder and Vardi in the original form states the following.

**CONJECTURE 4.1 (CONJECTURE 1 IN FEDER AND VARDI [1998]).** *A constraint-satisfaction problem is not in Datalog if and only if the associated core  $\mathbb{T}$  can simulate a core  $\mathbb{T}'$  consisting of two relations  $C, Z$  that give the ability to count. This is equivalent to simulating either  $\mathbb{Z}_p$  or one-in-three SAT.*

It is shown in Feder and Vardi [1998] that if a constraint language can simulate one-in-three SAT or  $\mathbb{Z}_p$  then the associated CSP does not have bounded width. The conjecture postulates that the reverse implication holds as well.

Given a constraint language  $\mathbb{D}$ , Bulatov [2004b] constructs a graph  $\mathcal{G}(\mathbb{D})$  with colored edges and a vertex set  $D$ . Two elements of the domain  $a, b \in D$  are connected with a blue edge if the smallest subalgebra of  $\mathbf{D}$  containing  $a$  and  $b$ , say  $\mathbf{E}$ , has a congruence  $\alpha$  and there is an affine operation of  $\mathbf{E}/\alpha$  (and no congruence on this algebra produces a quotient with a semilattice or majority operation).

**CONJECTURE 4.2 (CONJECTURE 2 IN BULATOV [2004B]).** *Let  $\mathbb{D}$  be a constraint language which is a core, and let  $\mathbb{D}'$  consist of all the relations from  $\mathbb{D}$  plus all the single element relations  $\{a\}$  for  $a \in D$ . Then,  $\text{CSP}(\mathbb{D})$  has bounded width if and only if*

- (1) *for every algebra  $\mathbf{A} \leq \mathbf{D}'$  and congruence  $\alpha$  of  $\mathbf{A}$  the algebra  $((\text{Pol}(\mathbb{D}'))_{|\mathbf{A}})/\alpha$  contains an operation which is not essentially unary, and*

<sup>3</sup>The reason is that, when we apply all polymorphisms of  $\mathbb{D}$  to a set  $\mathcal{F}$  satisfying (1) and (2), we get a set satisfying the same conditions. On the other hand, the  $(k, l)$ -consistency checking algorithm produces the largest set  $\mathcal{F}$  of partial assignments satisfying the constraints and conditions (1) and (2). Therefore, the set  $\mathcal{F}$  is closed under all polymorphisms.

(2) the graph  $\mathcal{G}(\mathbb{D})$  has no blue edges,

Similarly, as in the case of the conjecture of Feder and Vardi, the implication from left to right is proved [Bulatov 2004b].

Larose and Zádori [2007] used Tame Congruence Theory (TCT) to conjecture the classification of problems of bounded width. TCT, developed in Hobby and McKenzie [1988], introduces five types of local behaviors of algebras. TCT then classifies varieties according to types of the behavior present in its finite members. The TCT-types **1** and **2** correspond to essentially unary and affine behavior, respectively.

**CONJECTURE 4.3 (CONJECTURE IN LAROSE AND ZÁDORI [2007]).** *For a core  $\mathbb{D}$ , the  $\text{CSP}(\mathbb{D})$  has bounded width if and only if  $\mathbf{D}$  generates a variety omitting TCT-types **1** and **2**.*

Also in this case, the authors show that if type **1** or **2** appears in the variety then the CSP in question cannot have bounded width. The reverse implication is the question asked.

The equivalence of the conjectures above has been shown [Bulatov and Valeriote 2008; Larose et al. 2009] which allows us to focus on proving the conjecture of Larose and Zádori. The knowledge of types of Tame Congruence Theory is not required for the proof presented in this article. Our proof uses a condition from Hobby and McKenzie [1988] stated, in Section 7, as Theorem 7.1.

## 5. REDUCTION TO BINARY RELATIONS

In this section we prove a reduction of Conjecture 4.3 to the case of languages with binary constraints only. The precise form of the binary instances we produce will simplify the remainder of the proof. We say that an instance of CSP is *syntactically simple* if

- every constraint is binary, that is, the scope of each constraint is a pair of *distinct* variables,
- for every pair  $(x, y)$  of distinct variables there is at most one constraint with scope  $(x, y)$ . The corresponding constraint relation is denoted by  $R_{x,y}$ ,<sup>4</sup> and
- if  $(x, y)$  is the scope of some constraint, then so is  $(y, x)$  and  $R_{y,x} = \{(b, a) : (a, b) \in R_{x,y}\}$ .

Let  $\mathbb{D}$  be a constraint language with the maximum arity of a constraint relation equal to  $n$  and such that the algebra  $\mathbf{D}$  generates a variety omitting types **1** and **2**. Let  $\mathcal{I}$  be an instance in the language  $\mathbb{D}$ . We run the  $(2\lceil \frac{n}{2} \rceil, 3\lceil \frac{n}{2} \rceil)$ -consistency checking algorithm on  $\mathcal{I}$ . If the answer is *NO*, there is no solution to  $\mathcal{I}$ . If the answer is *YES*, we produce a new instance  $\mathcal{I}'$  in a language  $\mathbb{D}'$  such that:

- (1) the instance  $\mathcal{I}'$  is syntactically simple (with one constraint  $((x, y), R'_{x,y})$  for every pair  $(x, y)$  of distinct variables),
- (2) the  $(2, 3)$ -consistency checking algorithm on  $\mathcal{I}'$  answers *YES* and stops with a set  $\mathcal{F}'$  such that  $R'_{x,y} = \{(f'(x), f'(y)) : f' \in \mathcal{F}' \text{ and } \text{dom } f' = \{x, y\}\}$ ,
- (3) the algebra  $\mathbf{D}'$  generates a variety omitting types **1** and **2**, and
- (4)  $\mathcal{I}$  has a solution if and only if  $\mathcal{I}'$  does.

The construction of  $\mathcal{I}'$  was presented in Section 5 of Barto and Kozik [2009a] (although with a different notation). For the sake of completeness, we now give a sketch.

<sup>4</sup>The reduction of Conjecture 4.3 to Theorem 6.6 presented in this section produces an instance with constraints  $((x, y), R_{x,y})$  for every pair  $(x, y)$  of distinct variables. The statement of Theorem 6.6 in full generality, that is, with some of the constraints missing, is referred, and required, in Barto and Kozik [2012].

Assume that the  $(2^{\lceil \frac{n}{2} \rceil}, 3^{\lceil \frac{n}{2} \rceil})$ -consistency checking algorithm on  $\mathcal{I}$  returns *YES* and stops with the set  $\mathcal{F}$  of partial assignments of variables. The constraint language  $\mathbb{D}'$  has domain  $D^{\lceil \frac{n}{2} \rceil}$  and the relations in the new, syntactically simple instance  $\mathcal{I}'$  are defined as follows:

- (1) for every  $\lceil \frac{n}{2} \rceil$ -tuple of variables in  $\mathcal{I}$  we introduce a variable in  $\mathcal{I}'$ , and
- (2) if  $x$  is a variable for  $(x_1, \dots, x_{\lceil \frac{n}{2} \rceil})$  and  $y$  for  $(y_1, \dots, y_{\lceil \frac{n}{2} \rceil})$ ,  $x \neq y$ , we introduce a constraint  $((x, y), R'_{x,y})$  where

$$R'_{x,y} = \{((a_1, \dots, a_{\lceil \frac{n}{2} \rceil}), (b_1, \dots, b_{\lceil \frac{n}{2} \rceil})) : \exists f \in \mathcal{F} f(x_i) = a_i \text{ and } f(y_i) = b_i\}.$$

Let  $\mathcal{F}'$  be the set of assignments obtained by the (2, 3)-consistency checking for  $\mathcal{I}'$ . It is clear that, after the first loop (lines 2–6 of Algorithm 1)  $f'$  is in  $\mathcal{F}'$  if and only if there is  $f \in \mathcal{F}$  such that for every variable  $x \in \text{dom } f'$  with corresponding tuple  $(x_1, \dots, x_{\lceil \frac{n}{2} \rceil})$  we have  $f'(x)_i = f(x_i)$ . That is, condition (2) imposed on  $\mathcal{F}'$  and  $R'_{x,y}$  in  $\mathcal{I}'$  holds after the first loop of the algorithm. The second loop of Algorithm 1 does not remove any functions from  $\mathcal{F}'$ . Therefore, the (2, 3)-consistency checking algorithm for  $\mathcal{I}'$  answers *YES* and condition (2) required for  $\mathcal{I}'$  holds.

The domain  $D^{\lceil \frac{n}{2} \rceil}$  is the universe of the algebra  $\mathbf{D}^{\lceil \frac{n}{2} \rceil}$  (a power of  $\mathbf{D}$ ) which lies in the variety generated by  $\mathbf{D}$ . Each relation

$$\{(a_1, \dots, a_{\lceil \frac{n}{2} \rceil}, b_1, \dots, b_{\lceil \frac{n}{2} \rceil}) : \exists f \in \mathcal{F} f(x_i) = a_i \text{ and } f(y_i) = b_i\}$$

is a subpower of  $\mathbf{D}$  (by the discussion in the fourth paragraph of Section 3), and thus all  $R'_{x,y}$ 's are subpowers of  $\mathbf{D}^{\lceil \frac{n}{2} \rceil}$ . The variety generated by  $\mathbf{D}^{\lceil \frac{n}{2} \rceil}$  is contained in the variety generated by  $\mathbf{D}$  and therefore omits types **1** and **2**. The algebra of polymorphisms of  $\mathbb{D}'$  contains all the operations of  $\mathbf{D}^{\lceil \frac{n}{2} \rceil}$  and thus omits types **1** and **2** as well [Hobby and McKenzie 1988].

## 6. REDUCTION TO WEAK PRAGUE INSTANCES

From this point on, all instances, without mentioning it, are assumed to be syntactically simple. By considerations of Section 5, this assumption does not decrease the generality of the result.

We define two consistency notions very important for the proof. The first notion is the weaker one.

**Definition 6.1.** An instance  $\mathcal{I} = (V, D, C)$  is called *1-minimal*, if there are sets  $P_x$ ,  $x \in V$ , such that the projection of each  $R_{x,y}$  on the first coordinate is  $P_x$  and on the second  $P_y$ .

In order to define a stronger consistency notion, a weak Prague instance, we define patterns and realizations.

**Definition 6.2 (Pattern and Step).** A *step* in an instance  $\mathcal{I}$  is a pair of variables which is the scope of a constraint in  $\mathcal{I}$ . A *pattern from  $x$  to  $y$*  (in  $\mathcal{I}$ ) is a sequence of variables  $p = (x = x_1, x_2, \dots, x_k = y)$  such that every pair  $(x_i, x_{i+1})$ ,  $1 \leq i \leq k - 1$ , is a step.

For a pattern  $p = (x_1, \dots, x_k)$ , we define  $-p = (x_k, \dots, x_1)$ . If  $p = (x_1, \dots, x_k)$ ,  $q = (y_1, \dots, y_l)$ ,  $x_k = y_1$ , then the sum of  $p$  and  $q$  is the pattern  $p + q = (x_1, x_2, \dots, x_k = y_1, y_2, \dots, y_l)$ . For a pattern  $p$  from  $x$  to  $x$  and a natural number  $k$ ,  $kp$  denotes the  $k$ -fold sum of  $p$  with itself.

Note that the addition of patterns is allowed only if the terminal point of the first pattern coincides with the initial point of the second one. Observe also that from the



definition of a syntactically simple instance it follows that  $-p$  is a pattern whenever  $p$  is.

**Definition 6.3 (Realization, Addition).** Let  $p = (x = x_1, x_2, \dots, x_k = y)$  be a pattern from  $x$  to  $y$  in an instance  $\mathcal{I}$ . A *realization* of  $p$  is a sequence  $(a_1, \dots, a_k) \in D^k$  such that  $(a_i, a_{i+1}) \in R_{x_i, x_{i+1}}$  for every  $1 \leq i \leq k-1$ .

For a subset  $A \subseteq D$ , we define  $A+p$  as the set of the last elements of those realizations of  $p$  whose first element is in  $A$ , that is,

$$A + p = \{b \in D : (\exists a_1, \dots, a_{k-1} \in D) a_1 \in A \text{ and } (a_1, \dots, a_{k-1}, b) \text{ is a realization of } p\}.$$

Finally, we define  $A - p = A + (-p)$ .

The addition of patterns is associative, that is,  $(A + p) + q = A + (p + q)$ . Also note that, in a 1-minimal instance, we have  $A \subseteq A + p - p$  for any  $A \subseteq P_x$  and any pattern  $p$  from  $x$ , in particular,  $A + p$  is nonempty when  $A \neq \emptyset$ . A weak Prague instance is a 1-minimal instance with additional requirements concerning addition of patterns.

**Definition 6.4 (Weak Prague Instance).** An instance  $\mathcal{I}$  is a *weak Prague instance* if

(P1)  $\mathcal{I}$  is 1-minimal (with sets  $P_x$  from Definition 6.1),

(P2) for every  $A \subseteq P_x$  and every pattern  $p$  from  $x$  to  $x$ , if  $A + p = A$ , then  $A - p = A$ , and

(P3) for any patterns  $p, q$  from  $x$  to  $x$  and every  $A \subseteq P_x$ , if  $A + p + q = A$ , then  $A + p = A$ .

The instance  $\mathcal{I}$  is *nontrivial*, if  $P_x \neq \emptyset$  for every  $x \in V$ .

To clarify the definition, let us consider the following digraph  $\mathbb{G}$  for a 1-minimal instance: vertices of  $\mathbb{G}$  are all the pairs  $(A, x)$  with  $x \in V$  and  $A \subseteq P_x$ , and  $((A, x), (B, y))$  forms a directed edge iff  $A + (x, y) = B$ . Condition (P3) means that no strong component of  $\mathbb{G}$  contains  $(A, x)$  and  $(A', x)$  with  $A \neq A'$ . Condition (P2) is equivalent (albeit this fact requires a reasoning) to the fact that every strong component of  $\mathbb{G}$  contains only “undirected” edges (that is, if  $((A, x), (B, y))$  is an edge then so is  $((B, y), (A, x))$ ).

**LEMMA 6.5.** *The instance  $\mathcal{I}'$  constructed in Section 5 is a weak Prague instance.*

**PROOF.** Let  $\mathcal{F}'$  be the set of partial assignments of variables after the (2,3)-consistency checking algorithm answered *YES* on  $\mathcal{I}'$ . We put  $P'_x = \{f(x) : f \in \mathcal{F}' \text{ and } \text{dom} f = \{x\}\}$ .

To see that condition (P1) is satisfied, let  $x$  and  $y$  be arbitrary variables. If  $a \in P'_x$ , then there is  $f \in \mathcal{F}'$  with domain  $\{x\}$  and  $f(x) = a$ . Then some function  $f'$  with  $\text{dom} f' = \{x, y\}$  and  $f'(x) = a$  belongs to  $\mathcal{F}'$  (as otherwise  $f$  would be removed by condition on line 11 of Algorithm 1). But this implies that  $a$  is in the first projection of  $R'_{x,y}$ . On the other hand, if  $a$  is in the first projection of  $R'_{x,y}$ , then there is  $f \in \mathcal{F}'$  with  $f(x) = a$  and thus  $f|_{\{x\}} \in \mathcal{F}'$  provides  $a \in P'_x$  ( $f|_{\{x\}} \in \mathcal{F}'$  as otherwise  $f$  would be dropped from  $\mathcal{F}'$  by condition on line 12 of Algorithm 1). The proof for the second projection is analogical.

To show (P2) and (P3), we first prove that for any  $(a, b)$  in  $R'_{x,y}$  and for any pattern  $p$  from  $x$  to  $y$ , we have  $b \in \{a\} + p$ . The proof is by induction on the length of  $p$ . If  $p$  is a step, then  $p = (x, y)$  and the claim is obvious. For the induction step from  $n$  to  $n+1$ , take any pattern  $p = (x, \dots, w, v, y)$  of length  $n+1$ . By induction hypothesis used for the pattern  $(x, \dots, w, y)$ , there is  $c \in \{a\} + (x, \dots, w)$  such that  $(c, b) \in R'_{w,y}$ . Therefore, there exists  $f \in \mathcal{F}'$  such that  $\text{dom} f = \{w, y\}$ ,  $f(w) = c$  and  $f(y) = b$ . By condition on line 11 of Algorithm 1, we can find  $f' \in \mathcal{F}'$  which extends  $f$  to  $\{w, v, y\}$ . Then, the element  $d = f'(v)$  satisfies  $(c, d) \in R'_{w,v}$ ,  $(d, b) \in R'_{v,y}$  (by condition on line 12) which implies  $b \in c + (w, v, y)$ , and  $b \in \{a\} + p$ .

To see (P3), we note that for any  $A \subseteq P'_x$  and any pattern  $p$  from  $x$  to  $x$  we have  $A \subseteq A + p$ . Indeed, take  $a \in A$ , and let  $p = (x, \dots, y, x)$ . As our instance is 1-minimal,

there is some  $b \in P'_y$  with  $(a, b) \in R'_{x,y}$ . By the previous paragraph,  $b \in \{a\} + (x, \dots, y)$  and therefore  $a \in \{a\} + p$  and  $A \subseteq A + p$ . Now condition (P3) follows: if  $A + p + q = A$ , then  $A \subseteq A + p \subseteq A + p + q = A$ .

Finally, for (P2), let  $A \subseteq P'_x$  and  $A + (x, \dots, y, x) = A$ . It suffices to show that  $A + (x, \dots, y) = A - (y, x)$  as we can apply the same argument to the sets  $A' = A + p'$  for initial segments  $p'$  of  $p$  (the condition  $A' + p'' = A'$  will be satisfied for a cyclic shift  $p''$  of  $p$ ). The inclusion  $A + (x, \dots, y) \subseteq A - (y, x)$  follows from 1-minimality as  $A + (x, \dots, y) \subseteq A + (x, \dots, y) + (y, x) - (y, x)$  and  $A + (x, \dots, y) + (y, x) = A$ . For the reverse inclusion, we take an arbitrary  $b \in A - (y, x)$ . Then,  $(a, b) \in R'_{x,y}$  for some  $a \in A$ , and, by a paragraph above,  $b \in \{a\} + (x, \dots, y)$  as required.  $\square$

In order to confirm Conjecture 4.3, it remains to prove the following theorem.

**THEOREM 6.6.** *Every nontrivial weak Prague instance in a constraint language  $\mathbb{D}$ , with  $\mathbf{D}$  in a variety omitting types **1** and **2**, has a solution.*

## 7. ALGEBRAIC TOOLS

Our proof relies heavily on the algebraic properties implied by omitting types **1** and **2**. We use the following characterization [Hobby and McKenzie 1988]:

**THEOREM 7.1.** *Let  $\mathcal{V}$  be a variety generated by a finite algebra. The following are equivalent:*

- (1)  $\mathcal{V}$  omits types **1** and **2**;
- (2) for every  $\mathbf{A} \in \mathcal{V}$  and three congruences  $\alpha, \beta, \gamma$  on  $\mathbf{A}$ , if  $\alpha \wedge \beta = \alpha \wedge \gamma$ , then  $\alpha \wedge \beta = \alpha \wedge (\beta \vee \gamma)$ .

In this section, we tacitly assume that all the algebras are idempotent, that is, every operation of every algebra satisfies the identity  $t(x, \dots, x) \approx x$ .

### 7.1. Absorption

One of the main algebraic concepts behind the proof is the notion of an absorbing subuniverse.

*Definition 7.2.* We say that  $B$  is an *absorbing subuniverse* of an algebra  $\mathbf{A}$  (denoted by  $B \triangleleft \mathbf{A}$ ) if  $B \leq \mathbf{A}$  and there exists a term  $t$  of  $\mathbf{A}$  such that  $t(B, B, \dots, B, A, B, B, \dots, B) \subseteq B$  for any position of  $A$ .

In varieties omitting type **1**, lack of absorption has interesting consequences for a particular type of subdirect subalgebras.

*Definition 7.3.* A subdirect subalgebra  $\mathbf{R}$  of  $\mathbf{A} \times \mathbf{B}$  is called *linked*, if  $\pi_1 \vee \pi_2 = 1_{\mathbf{R}}$ .

The set  $R \subseteq A \times B$  can be viewed as a bipartite graph with partite sets  $A$  and  $B$ . Then,  $\mathbf{R}$  is linked if and only if this graph is connected.

One of the main tools of Barto and Kozik [2010] is extensively used in the proof.

**THEOREM 7.4 (ABSORPTION THEOREM).** *If  $\mathbf{A}$  and  $\mathbf{B}$  are algebras in a variety omitting type **1**,  $R \leq_{sd} \mathbf{A} \times \mathbf{B}$  is linked and  $R \neq \mathbf{A} \times \mathbf{B}$ , then  $\mathbf{A}$  or  $\mathbf{B}$  has a proper absorbing subuniverse.*

### 7.2. Pointed Terms

The second algebraic tool – pointed terms – is studied in more detail in Barto et al. [2013]. Here we provide only the facts necessary for the proof of the main result of the paper.

**Definition 7.5.** Let  $\mathbf{A}$  be an algebra and let  $a \in A$ . An  $n$ -ary term  $t$  of  $\mathbf{A}$  *points to  $a$*  if there exist  $a_1, \dots, a_n \in A$  such that

$$t(b_1, \dots, b_n) = a \text{ whenever } b_i \in A \text{ and } |\{i : a_i \neq b_i\}| \leq 1.$$

In the remaining part of this section, we will be working towards a proof of the following lemma.

**LEMMA 7.6.** *Let  $\mathbf{A}$  be a simple algebra with no proper absorbing subuniverse, generating a variety omitting types **1** and **2**. Then, for every  $a \in A$ , there exists a term of  $\mathbf{A}$  which points to  $a$ .*

We begin with a basic property of (absorbing) subuniverses of algebras.

**LEMMA 7.7.** *Let  $R \leq_{sd} \mathbf{A} \times \mathbf{B}$  and let  $C$  be an (absorbing) subuniverse of  $\mathbf{A}$ . Then the set  $\{d \in B : \exists c \in C (c, d) \in R\}$  is an (absorbing) subuniverse of  $\mathbf{B}$ . The absorption is realized by the same term.*

**PROOF.** Let  $C$  be a subuniverse of  $\mathbf{A}$ . Put  $D = \{d \in B : \exists c \in C (c, d) \in R\}$  and let  $t$  be a  $n$ -ary term. For any  $d_1, \dots, d_n \in D$ , we can find  $c_1, \dots, c_n \in C$  such that  $(c_i, d_i) \in R$  for all  $i$ . Therefore,  $t((c_1, d_1), \dots, (c_n, d_n)) = (t(c_1, \dots, c_n), t(d_1, \dots, d_n)) \in R$  and, since  $C$  is a subuniverse of  $\mathbf{A}$ , we get  $t(d_1, \dots, d_n) \in D$  as required – this proves that  $D$  is a subuniverse of  $\mathbf{B}$ .

If  $C$  were absorbing  $\mathbf{A}$  with an  $m$ -ary term  $r$ , then for,  $d_1, \dots, d_{k-1}, b, d_{k+1}, \dots, d_m$  with  $d_i \in D$ , we find  $c_1, \dots, c_{k-1}, c_{k+1}, \dots, c_m$  as previously stated and use subdirectness of  $R$  to get  $a \in A$  with  $(a, b) \in R$ . We proceed as in the previous case and use absorption to conclude that  $r(c_1, \dots, c_{k-1}, a, c_{k+1}, \dots, c_m) \in C$ .  $\square$

The following lemma states that a product of algebras with no proper absorbing subuniverses has no proper absorbing subuniverse.

**LEMMA 7.8.** *Let  $\mathbf{A}_1, \dots, \mathbf{A}_n$  be algebras and let  $\mathbf{R} \not\leq \prod_{i=1}^n \mathbf{A}_i$ . If  $\mathbf{R} \triangleleft \prod_{i=1}^n \mathbf{A}_i$ , then some  $\mathbf{A}_i$  contains a proper absorbing subalgebra.*

**PROOF.** Suppose, for a contradiction, that the lemma holds for  $n - 1$  and fails for  $\mathbf{R} \leq \prod_{i=1}^n \mathbf{A}_i$ . The projection of  $\mathbf{R}$  on the first coordinate is an absorbing subuniverse of  $\mathbf{A}_1$ , so it has to be equal to  $\mathbf{A}_1$ . Therefore, there exists  $a \in A_1$  such that

$$\emptyset \neq R' = \{(b_2, \dots, b_n) \mid (a, b_2, \dots, b_n) \in R\} \neq \prod_{i=2}^n \mathbf{A}_i.$$

Since  $\mathbf{A}_1$  is idempotent,  $R'$  is a subuniverse of  $\prod_{i=2}^n \mathbf{A}_i$ . Moreover, since  $\mathbf{R}$  absorbs the full product with an idempotent term, it is easily seen that  $R'$  absorbs  $\prod_{i=2}^n \mathbf{A}_i$  with the same term. Using the induction hypothesis for  $R'$ , we get a proper absorbing subuniverse of one of the  $\mathbf{A}_i$ 's for some  $i > 1$  which is a contradiction.  $\square$

Next, we argue that certain products of algebras (in varieties omitting types **1** and **2**) have no nontrivial subdirect subuniverses.

**LEMMA 7.9.** *Let  $\mathbf{A}_1, \dots, \mathbf{A}_n$  be simple algebras with no proper absorbing subuniverses and in a variety omitting types **1** and **2**. If  $\mathbf{R} \leq_{sd} \prod_{i=1}^n \mathbf{A}_i$  and  $\pi_i \vee \pi_j = 1_{\mathbf{R}}$  for every  $i \neq j$ , then  $\mathbf{R} = \prod_{i=1}^n \mathbf{A}_i$ .*

**PROOF.** First, we prove that there exists  $j$  such that  $\bigwedge_{i \neq j} \pi_i \neq 0_{\mathbf{R}}$ , unless  $0_{\mathbf{R}} = 1_{\mathbf{R}}$  (which implies that  $\mathbf{R}$  has one element and the lemma is trivially true). Suppose otherwise, that is, for every  $j$ , we have  $\bigwedge_{i \neq j} \pi_i = 0_{\mathbf{R}}$ . We prove by induction on  $n - |I|$  that  $\bigwedge_{i \in I} \pi_i = 0_{\mathbf{R}}$  for every  $I \subseteq \{1, \dots, n\}$ . By the assumption, it is true for  $|I| = n - 1$ ;

suppose it holds for all  $I$  with  $|I| = k$  and let  $J$  be such that  $|J| = k - 1$ . Let  $l \notin J, m \notin J, l \neq m$ . Theorem 7.1 (with  $\alpha = \bigwedge_{i \in J} \pi_i, \beta = \pi_l, \gamma = \pi_m$ ) provides

$$0_{\mathbf{R}} = \bigwedge_{i \in J \cup \{l\}} \pi_i = \bigwedge_{i \in J} \pi_i \wedge (\pi_l \vee \pi_m) = \bigwedge_{i \in J} \pi_i$$

and the induction step is proved. Finally, for  $|I| = 1$ , we get  $\pi_1 = \pi_2 = 0_{\mathbf{R}}$  which contradicts  $\pi_1 \vee \pi_2 = 1_{\mathbf{R}}$ .

Without loss of generality, we assume that  $\bigwedge_{i \neq 1} \pi_i \neq 0_{\mathbf{R}}$ . The projection of  $\pi_1 \vee \bigwedge_{i \neq 1} \pi_i$  to the first coordinate cannot be  $0_{\mathbf{A}_1}$  (since for some  $a \neq b$  we have  $(a, b) \in \bigwedge_{i \neq 1} \pi_i$ , we immediately get that the projections of  $a$  and  $b$  on the first coordinate are different and related by the projection of  $\pi_1 \vee \bigwedge_{i \neq 1} \pi_i$ ). Since  $\mathbf{A}_1$  is simple,  $\pi_1 \vee \bigwedge_{i \neq 1} \pi_i = 1_{\mathbf{R}}$ .

Suppose, for a contradiction, that the lemma holds for  $n - 1$  and fails for  $\mathbf{R} \leq_{sd} \prod_{i=1}^n \mathbf{A}_i$ . By this assumption, the projection of  $\mathbf{R}$  to the coordinates  $2, \dots, n$  is equal to  $\prod_{i=2}^n \mathbf{A}_i$  and  $\mathbf{R}$  is a subdirect product of  $\mathbf{A}_1$  and  $\prod_{i=2}^n \mathbf{A}_i$ . By the previous paragraph,  $\mathbf{R}$  is linked and therefore, by the Absorption Theorem,  $\mathbf{R}$  is either equal to  $\mathbf{A}_1 \times \prod_{i=2}^n \mathbf{A}_i$ , or  $\mathbf{A}_1$  contains a proper absorbing subuniverse – which contradicts the hypotheses, or  $\prod_{i=2}^n \mathbf{A}_i$  contains a proper absorbing subuniverse – which contradicts Lemma 7.8.  $\square$

Finally, we argue that if a simple algebra with no proper absorbing subuniverses generates a variety omitting types **1** and **2**, then it has many term operations. In the statement, we use the notation  $Sg_{\mathbf{A}}(B)$  for the subalgebra of  $\mathbf{A}$  generated by  $B$ , that is, the smallest subalgebra of  $\mathbf{A}$  containing  $B$ .

**LEMMA 7.10.** *Let  $\mathbf{A}$  be a simple algebra in a variety omitting types **1** and **2**, with no proper absorbing subuniverse. Let  $n$  be an arbitrary positive integer, and let  $\mathcal{Z} \subseteq A^n$  be such that*

$$\forall a \neq b \in \mathcal{Z} \exists i, j (a_i = a_j \wedge b_i \neq b_j) \vee (a_i \neq a_j \wedge b_i = b_j)$$

and

$$\forall a \in \mathcal{Z} Sg_{\mathbf{A}}(\{a_1, \dots, a_n\}) = \mathbf{A}.$$

Then every function from  $\mathcal{Z}$  to  $A$  is a restriction of some  $n$ -ary term operation of  $\mathbf{A}$ .

**PROOF.** Let  $\mathbf{F} \leq \mathbf{A}^{A^n}$  be the free algebra on  $n$ -generators (the universe consists of the  $n$ -ary term operations of  $\mathbf{A}$ ).

Choose  $a, b$  to be two arbitrary tuples from  $\mathcal{Z}$  and let  $\mathbf{R}$  be the projection of  $\mathbf{F}$  to the coordinates  $a, b$ . Since  $\mathbf{F}$  is generated by projections, we have  $(a_i, b_i) \in \mathbf{R}$  for every  $i \leq n$ . By the second condition on  $\mathcal{Z}$ , the algebra  $\mathbf{R}$  is subdirect in  $\mathbf{A} \times \mathbf{A}$ . The first condition provides  $i, j$  with, say,  $a_i = a_j$  and  $b_i \neq b_j$  (the other case is similar). Thus, the projection of  $\pi_1 \vee \pi_2$  (computed in  $\mathbf{R}$ ) to the second coordinate is not the equality congruence, and, by simplicity of  $\mathbf{A}$ , it is the full congruence. That means that  $\mathbf{R}$  is linked, so  $\pi_a \vee \pi_b = 1_{\mathbf{F}}$ .

By the previous paragraph, the restriction of  $\mathbf{F}$  to  $\mathcal{Z}$  satisfies the hypotheses of Lemma 7.9, and therefore is the full relation. In other words, every function from  $\mathcal{Z}$  to  $\mathbf{A}$  extends to a term operation of  $\mathbf{A}$ .  $\square$

To finish the proof of Lemma 7.6, we take a list  $a_1, \dots, a_n$  of all elements in  $\mathbf{A}$ . We put  $b = (a_1, a_1, a_2, a_2, \dots, a_n, a_n) \in \mathbf{A}^{2n}$  and

$$\mathcal{Z} = \{c \in \mathbf{A}^{2n} : |\{i : c_i \neq b_i\}| \leq 1\}.$$

It is straightforward to verify that the set  $\mathcal{Z}$  satisfies the hypotheses of Lemma 7.10 and therefore, for every  $a \in A$ , the constant function mapping  $\mathcal{Z}$  to  $a$  extends to a term operation. The term associated with this operation points to  $a$ .

### 7.3. Additional Properties of Absorbing Subuniverses

We require the following observation.

**LEMMA 7.11.** *Let  $\mathbf{A}$  and  $\mathbf{B}$  be algebras in a variety omitting types **1** and **2** and such that neither  $\mathbf{A}$  nor  $\mathbf{B}$  has a proper absorbing subuniverse. Moreover, let  $R \leq_{sd} \mathbf{A} \times \mathbf{B}$  and let  $\alpha$  be a maximal congruence of  $\mathbf{A}$ . Then*

- (1) *either  $(a, b), (a', b) \in R$  implies that  $(a, a') \in \alpha$  for all  $a, a' \in A$ , or*
- (2) *for every  $a \in A$  and  $b \in B$  there exists  $a' \in A$  such that  $(a, a') \in \alpha$  and  $(a', b) \in R$ .*

**PROOF.** Consider the subdirect product  $R' = \{(a/\alpha, b) : (a, b) \in R\} \leq_{sd} \mathbf{A}/\alpha \times \mathbf{B}$ . The algebra  $\mathbf{A}/\alpha$  has no absorbing subuniverse (since it is easily seen that the preimage of an absorbing subuniverse of  $\mathbf{A}/\alpha$  is an absorbing subuniverse of  $\mathbf{A}$ ) and is simple (as  $\alpha$  is maximal).

If (1) is not satisfied, then the projection of  $\pi_1 \vee \pi_2$  to the first coordinate (i.e.,  $\mathbf{A}/\alpha$ ) is not the equality congruence, therefore, since  $\mathbf{A}/\alpha$  is simple,  $\pi_1 \vee \pi_2 = 1_{R'}$ , so  $R'$  is linked. By the Absorption Theorem,  $R' = A/\alpha \times B$  which is a restatement of (2).  $\square$

## 8. A PROOF OF THEOREM 6.6

Theorem 6.6 is a generalization of a result in Barto and Kozik [2009b]. The more general version is required for the approximability results which appear in Barto and Kozik [2012].

To prove Theorem 6.6, we work with a weak Prague instance  $\mathcal{I}$  in a constraint language  $\mathbb{D}$  such that the associated algebra  $\mathbf{D}$  generates a variety omitting types **1** and **2**. Since  $\mathcal{I}$  is syntactically simple, the only constraints are of the form  $((x, y), R_{x,y})$  and each  $R_{x,y}$  is a subuniverse of  $\mathbf{D}^2$ . Similarly, each  $P_x$  from the definition of 1-minimality is a subuniverse of  $\mathbf{D}$ .

Our last preliminary step is to drop, from  $\mathbf{D}$ , all the nonidempotent operations. The algebra  $\mathbf{D}'$  obtained in this way is idempotent and, still, generates a variety omitting types **1** and **2** (it follows from part (2) of Theorem 9.10 in Hobby and McKenzie [1988]). All  $R_{x,y}$ 's and  $P_x$ 's are subuniverses of  $(\mathbf{D}')^2$  and  $\mathbf{D}'$  respectively. This allows us to substitute, in the remainder of the proof,  $\mathbf{D}$  with  $\mathbf{D}'$  and work within a variety generated by an idempotent algebra. In particular, all the results of Section 7 can be applied in this variety.

### 8.1. Pointed Decomposition

Given a weak Prague instance, we proceed by reducing it to a smaller induced subinstance.

**Definition 8.1.** Let  $\mathcal{J}$  be an instance. An instance  $\mathcal{J}'$  is an *induced subinstance* of  $\mathcal{J}$  with potatoes  $P'_x \leq \mathbf{D}$ ,  $x \in V$ , if  $R'_{x,y} = R_{x,y} \cap (P'_x \times P'_y)$ .

In order to succeed with such a reduction, we first decompose the instance.

**Definition 8.2.** A *decomposition* of a 1-minimal instance  $\mathcal{J}$  consists of induced subinstances  $\mathcal{J}^1, \dots, \mathcal{J}^l$  of  $\mathcal{J}$  with potatoes  $P_x^i \leq \mathbf{P}_x$ , relations  $R_{x,y}^i = R_{x,y} \cap (P_x^i \times P_y^i)$ ,  $x, y \in V$ ,  $i \leq l$  and a subset  $X$  of  $V$  such that

- (1) if  $x \notin X$ , then  $P_x^i = P_x$  for all  $i \leq l$ ,
- (2) if  $x \in X$ , then



- (a)  $P_x^i \cap P_x^j = \emptyset$  for all  $i \neq j$ , and
- (b) for any step  $(x, y)$ , either  $P_x^i + (x, y) = P_y^i$  for all  $i \leq l$ , or  $P_x^i + (x, y) = P_y$  for all  $i \leq l$ .

Note that any decomposition of a 1-minimal  $\mathcal{J}$  consists of instances which are 1-minimal as well.

**Definition 8.3.** We say that the decomposition is *pointed*, if there exists a term  $t$  of  $\mathbf{D}$  of arity  $m$  and indices  $k_1, \dots, k_m$  such that

$$\forall i \leq m \forall x \in V t(P_x^{k_1}, \dots, P_x^{k_{i-1}}, P_x, P_x^{k_{i+1}}, \dots, P_x^{k_m}) \subseteq P_x^1.$$

The decomposition is *proper* if  $\mathcal{J}^1$  is a nontrivial instance and  $P_x^1 \subsetneq P_x$  for some  $x \in V$ .

In the following two sections, we prove:

**THEOREM 8.4.** *Every nontrivial weak Prague instance  $\mathcal{J}$  with  $|P_x| > 1$  for some  $x \in V$  has a proper pointed decomposition.*

In the last section, we show that  $\mathcal{J}^1$  is a weak Prague instance. As  $P_x^1$ 's are assumed to be subuniverses of  $\mathbf{D}$ , then  $R_{x,y}^1$ 's are subuniverses of  $\mathbf{D}^2$ . That means that throwing all the  $R_{x,y}^1$ 's into  $\mathbb{D}$  does not affect polymorphisms of  $\mathbb{D}$  and that we reduced  $\mathcal{I}$  to a strictly smaller instance still satisfying hypotheses of Theorem 6.6. Continuing this process, we obtain a minimal induced subinstance of  $\mathcal{I}$  and it must have  $|P_x| = 1$  for all  $x \in V$ . Any such instance, trivially, has a solution which sends  $x$  to the unique element of  $P_x$ . This finishes the proof of Theorem 6.6.

## 8.2. Pointed Decomposition when Absorption Is Present

We prove Theorem 8.4 when some algebra  $\mathbf{P}_x$  has a proper absorbing subuniverse  $A$ . In this case, we find a decomposition into a single subinstance  $\mathcal{J}^1$ .

We define a preorder  $\sqsubseteq$  on the set of all pairs  $(C, y)$ ,  $C \subsetneq P_y$ , by  $(C, y) \sqsubseteq (D, z)$  if  $(C, y) = (D, z)$  or there exists a pattern  $p$  from  $y$  to  $z$  such that  $C + p = D$ . Among the equivalence classes of this preorder which are greater or equal to the equivalence class containing  $(A, x)$ , we choose a maximal one and denote it by  $\mathcal{M}$ . Let  $X$  denote the set of all  $y \in V$  for which there exists some  $C$  such that  $(C, y) \in \mathcal{M}$ .

**CLAIM 1.** *The set  $C$  is uniquely determined by  $y$ .*

**PROOF.** If  $(C, y), (D, y) \in \mathcal{M}$ , then by the fact that  $(C, y)$  and  $(D, y)$  are in the same equivalence class of  $\sqsubseteq$  we get  $C = D$ , or  $C + p = D$  and  $D + q = C$  for some patterns  $p$  and  $q$  from  $y$  to  $y$ , and, by (P3),  $C = D$  again.  $\square$

For  $y \in X$  we define  $P_y^1$  as the unique set with  $(P_y^1, y) \in \mathcal{M}$ . For  $y \notin X$ , we put  $P_y^1 = P_y$ . Note that, as  $\mathcal{J}$  is 1-minimal, we have  $P_y^1 \neq \emptyset$  for all  $y \in V$ . Conditions (1) and (2a) from Definition 8.2 hold by the construction; it remains to verify condition (2b) and find a term witnessing that the decomposition is pointed.

To show condition (2b) of Definition 8.2, let  $y \in X$  be arbitrary and let  $(y, z)$  be any step. If  $(P_y^1 + (y, z), z) \in \mathcal{M}$ , then  $P_y^1 + (y, z) = P_z^1$  as required. On the other hand, if  $(P_y^1 + (y, z), z) \notin \mathcal{M}$ , then this pair is outside of the preorder; therefore,  $P_y^1 + (y, z) = P_z$ .

It remains to show that the induced subinstance  $\mathcal{J}^1$  of  $\mathcal{J}$  with potatoes  $P_y^1$  for  $y \in V$  is a pointed decomposition of  $\mathcal{J}$ .

Note that Lemma 7.7 implies that, for an absorbing subuniverse  $B$  of  $P_y$  and a step  $(y, v)$ , the set  $B + (y, v)$  is an absorbing subuniverse of  $P_v$ . By induction, we get that  $P_y^1$  (which is equal to  $A + p$  for some  $p$  from  $x$  to  $y$ ) is an absorbing subuniverse of  $P_y$  and the absorbing term is the same as for  $A$  and  $\mathbf{P}_x$ . This term satisfies the condition in Definition 8.2 with  $k_i = 1$  for all  $i$ .

### 8.3. Pointed Decomposition when Absorption Is Missing

Now we prove Theorem 8.4 in the case that none of the algebras  $\mathbf{P}_x, x \in V$  has a proper absorbing subuniverse.

**LEMMA 8.5.** *Let  $\alpha$  be a maximal congruence on some  $\mathbf{P}_x$ . Then, for every step  $(x, y)$*   
*(case1) either  $(a/\alpha + (x, y)) \cap (b/\alpha + (x, y)) \neq \emptyset$  implies  $(a, b) \in \alpha$  for all  $a, b \in P_x$ ,*  
*(case2) or  $a/\alpha + (x, y) = P_y$  for all  $a \in P_x$ .*

**PROOF.** The lemma is a restatement of Lemma 7.11. □

If  $\mathbf{P}_x$  and  $\alpha$  are as in the previous lemma, then we can define an equivalence relation  $\alpha + (x, y)$ . The equivalence is given by the following partition of  $P_y$

$$\{a/\alpha + (x, y) : a \in P_x\}.$$

This defines a trivial partition  $\{P_y\}$  in (case2) of Lemma 8.5 or a proper partition in (case1). The equivalence  $\alpha + (x, y)$  is easily seen to be a congruence of  $\mathbf{P}_y$ . Moreover, in (case1), the function

$$\{(a/\alpha, b/(\alpha + (x, y))) : (a, b) \in R_{x,y}\}$$

provides an isomorphism between  $\mathbf{P}_x/\alpha$  and  $\mathbf{P}_y/(\alpha + (x, y))$ . Therefore, in this case,  $\alpha + (x, y)$  is a maximal congruence of  $\mathbf{P}_y$  and  $(\alpha + (x, y)) + (y, x) = \alpha$ .

**LEMMA 8.6.** *Let  $\alpha$  be a maximal congruence on  $\mathbf{P}_x$ ,  $a \in P_x$ , and let  $p$  be a pattern from  $x$  to  $y$ . If  $a/\alpha + p \subsetneq P_y$ , then for any  $a' \in P_x$  we have  $a'/\alpha + p - p = a'/\alpha$ .*

**PROOF.** The proof is by induction on the length of  $p$ . Let  $p = (x, z) + p'$ . Then,  $a/\alpha + (x, z) \subsetneq P_z$  and (case1) of Lemma 8.5 (used for the step  $(x, z)$ ) applies. Therefore  $\alpha + (x, z)$  is a maximal congruence on  $\mathbf{P}_z$ , both  $a/\alpha + (x, z)$  and  $a'/\alpha + (x, z)$  are its congruence classes and  $a'/\alpha + (x, z) + (z, x) = a'/\alpha$ . By induction hypothesis,  $(a'/\alpha + (x, z)) + p' - p' = (a'/\alpha + (x, z))$  and the lemma is proved. □

If  $\mathbf{P}_x$  and  $\alpha$  are as in the previous lemma,  $a/\alpha + p \subsetneq P_y$ , and  $q$  is another pattern from  $x$  to  $y$  such that  $a/\alpha + q \subsetneq P_y$ , then, by the previous lemma,  $(a/\alpha + p) - p + q = a/\alpha + q$  and  $(a/\alpha + q) - q + p = a/\alpha + p$ , and using (P3), we get  $a/\alpha + p = a/\alpha + q$ , that is, the result of addition is independent of the pattern.

Now we are ready to define the decomposition. We assume that  $P_x$  has at least two elements and put  $\alpha$  to be a maximal congruence on  $\mathbf{P}_x$ . We denote by  $P_x^1, \dots, P_x^l$  the equivalence classes of  $\alpha$  and will decompose  $\mathcal{J}$  into  $\mathcal{J}^1, \dots, \mathcal{J}^l$ . We include  $y$  into  $X$  if there is a pattern  $p$  from  $x$  to  $y$  such that  $P_x^1 + p \subsetneq P_y$  and, in this case, we set  $P_y^i = P_x^i + p$  for all  $i \leq l$  (by the discussion after Lemma 8.6, the definition is independent on the choice of  $p$  as long as  $P_x^i + p \subsetneq P_y$ ). If  $y \notin X$ , we put  $P_y^i = P_y$ . Condition (1) of Definition 8.2 holds by the construction.

Let  $y$  be an arbitrary variable in  $X$ . From Lemma 8.6, it follows that  $P_y^i \cap P_y^{i'} = \emptyset$  for any  $i \neq i'$ , that is, condition (2a) of Definition 8.2 holds. To prove condition (2b), take any step  $(y, z)$ , and suppose that  $P_y^i + (y, z) \neq P_z^i$  for some  $i \leq l$ . By Lemma 8.5, the sets  $(P_y^j + (y, z))$  (for different  $j$ 's) are either pairwise disjoint, or all equal to  $P_z$ . If they

were disjoint,  $P_y^i + (y, z)$  would be a candidate for  $P_z^i$  (given by a different path). This is impossible (by the discussion after Lemma 8.6) and therefore  $P_y^i + (y, z) = P_z$  for all  $j \leq l$  – this proves condition (2b) of Definition 8.2. Therefore, the induced subinstances  $\mathcal{J}^1, \dots, \mathcal{J}^l$  with potatoes  $P_x^i, x \in V$  form a decomposition.

It remains to show that the decomposition is pointed. Clearly, for any  $i$ ,  $P_x^i$  is a subuniverse of  $\mathbf{P}_x$  and therefore, by Lemma 7.7,  $P_y^i$  is a subuniverse of  $\mathbf{P}_y$ . In order to find a pointed term for this decomposition consider  $\mathbf{P}_x/\alpha$ . This algebra is simple, has no proper absorbing subuniverses (since  $\mathbf{P}_x$  does not) and lies in a variety omitting types 1 and 2. Therefore, by Lemma 7.6, there exists a term  $t(x_1, \dots, x_m)$  in  $\mathbf{P}_x/\alpha$  pointing to  $P_x^1/\alpha$ .

This term clearly satisfies the required condition for the variable  $x$  and for all the variables outside the set  $X$ . On the other hand, if  $y \in X$ , then  $\mathbf{P}_y/(\alpha + p)$  is isomorphic to  $\mathbf{P}_x/\alpha$  via an isomorphism sending  $\mathbf{P}_x^1$  to  $\mathbf{P}_y^1$  and therefore  $t(x_1, \dots, x_m)$  satisfies the condition for  $\mathbf{P}_y$  as well. This finishes the case when there is no absorption in the weak Prague instance.

#### 8.4. The Reduction to $\mathcal{J}^1$

The following theorem finishes the proof of the main result.

**THEOREM 8.7.** *Let  $\mathcal{J}$  be a weak Prague instance with a pointed decomposition  $\mathcal{J}^1, \dots, \mathcal{J}^l$  with potatoes  $P_x^i, x \in V, i \leq l$ . Then,  $\mathcal{J}^1$  is a weak Prague instance.*

In this section, we need to consider realizations and addition in the instance  $\mathcal{J}$  as well as in the subinstance  $\mathcal{J}^1$ . To distinguish them, we write  $A + p$  for addition in  $\mathcal{J}$  and  $A +^1 p$  for addition in  $\mathcal{J}^1$  (and similarly for subtraction).

When we keep adding a pattern  $p$  from  $x$  to  $x$  to a set  $A \subseteq P_x$ , the process will stabilize on a set containing  $A$ .

**LEMMA 8.8.** *Let  $x \in V, A \subseteq P_x$  and let  $p$  be a pattern from  $x$  to  $x$ . There exists a natural number  $i$  such that  $A + ip + jp = A + ip$  for every integer<sup>5</sup>  $j$  and, moreover,  $A \subseteq A + ip$ .*

**PROOF.** Since the domain is finite, there exist  $i, i' > 0$  such that  $A + ip + i'p = A + ip$ . Put  $B = A + ip$ . Then,  $B + p + (i' - 1)p = B$  and, by (P3),  $B + p = B$ . From (P2) we get that  $B - p = B$  and now clearly  $B + jp = B$  for every integer  $j$ . Finally, we have  $A \subseteq A + ip - ip = B - ip = B$ .  $\square$

For every  $A \subseteq P_x$  and pattern  $p$  from  $x$  to  $x$ , the set  $A + ip$  given by Lemma 8.8 is denoted by  $[A]_p$ . We have  $A \subseteq [A]_p$  and  $[A]_p + jp = [A]_p$  for every integer  $j$ .

Theorem 8.7 is a consequence of the following lemma.

**LEMMA 8.9.** *Let  $x \in V, A \subseteq P_x^1$  and let  $p$  be a pattern from  $x$  to  $x$ . If  $A +^1 p = A$ , then  $A = [A]_p \cap P_x^1$ .*

Using this lemma, we prove Theorem 8.7.

**PROOF OF THEOREM 8.7.** It follows from the definition of decomposition that  $\mathcal{J}^1$  (as well as  $\mathcal{J}^2, \dots, \mathcal{J}^l$ ) is 1-minimal.

(P2). If  $A +^1 p = A$ , where  $A \subseteq P_x$  and  $p$  is a pattern from  $x$  to  $x$ , then  $A \subseteq A +^1 p -^1 p = A -^1 p$ . To prove the reverse inclusion, we use the properties of  $[A]_p$  stated after Lemma 8.8 and Lemma 8.9. Since  $A \subseteq [A]_p = [A]_p - p$ , we have  $A -^1 p \subseteq A - p \subseteq [A]_p - p = [A]_p$ , and then  $A -^1 p \subseteq [A]_p \cap P_x^1 = A$ . Thus,  $A = A -^1 p$  as required.

<sup>5</sup>Compare Definition 6.2 for pattern multiplicities.

(P3). Let  $A \subseteq P_x^1$  and let  $p, q$  be patterns from  $x$  to  $x$  such that  $A +^1 p +^1 q = A$ . Using (P3) for the instance  $\mathcal{J}$  and  $[A]_{p+q} = [A]_{p+q} + p + q$ , we get  $[A]_{p+q} + p = [A]_{p+q}$ . Now  $A +^1 p \subseteq A + p \subseteq [A]_{p+q} + p = [A]_{p+q}$  and, by Lemma 8.9, we obtain  $A +^1 p \subseteq [A]_{p+q} \cap P_x^1 = A$ .

Similarly, the set  $A = (A +^1 p) +^1 q$  is a subset of  $A +^1 p$ , and then  $A = A +^1 p$  as required.  $\square$

The remaining part of this section is devoted to the proof of Lemma 8.9. Let  $x, A, p$  be as in the statement of the lemma.

For any  $i > 0$ , we have  $A +^1 ip = A +^1 p$  and  $[A]_p = [A]_{ip}$ , therefore we can, without loss of generality, replace  $p$  with the pattern  $ip$ . We do it for large enough  $i$  so that  $B + p = [B]_p$  for every  $B \subseteq P_x$ .

As  $A \subseteq [A]_p$ , the inclusion  $A \subseteq [A]_p \cap P_x^1$  is satisfied and we proceed to prove the reverse containment. Let  $b \in [A]_p \cap P_x^1$  be arbitrary. We need to show that  $b \in A$ . As  $[A]_p = A + p$ , there exists  $a \in A$  such that  $b \in \{a\} + p$  and we fix such an element  $a$ .

We split the pattern  $p$  into parts. Let  $p = (x = x_1, x_2, \dots, x_n = x)$  and define  $r, s$ :

- let  $r$  be the largest index such that  $P_{x_z}^1 + (x_z, x_{z+1}) = P_{x_{z+1}}^1$  for all  $z < r$ ,
- let  $s$  be the smallest index such that  $P_{x_z}^1 + (x_z, x_{z-1}) = P_{x_{z-1}}^1$  for all  $z > s$ .

If it is not true that  $r + 1 \leq s - 1$ , then  $b \in \{a\} + p$  implies  $b \in \{a\} +^1 p \in A$  since no realization of  $p$  can leave  $\mathcal{J}^1$  and return to it, thus in this case the proof is concluded.

Let  $t$  be an  $m$ -ary term and  $k_1, \dots, k_m$  be indices from the definition of pointed decomposition. We find a matrix of domain elements with  $m$  rows such that its  $i$ th row

$$(a = a_{11}^i, a_{12}^i, \dots, a_{1n}^i = a_{21}^i, \dots, a_{2n}^i = a_{31}^i, \dots, a_{(m-1)n}^i = a_{m1}^i, \dots, a_{mn}^i = b)$$

is a realization of the pattern  $mp$  in the instance  $\mathcal{J}$  satisfying the following conditions for every  $1 \leq j \leq m$ .

- (1) The elements  $a_{j1}^i, \dots, a_{jr}^i, a_{js}^i, \dots, a_{jn}^i$  lie in the instance  $\mathcal{J}^1$  (i.e., more precisely,  $a_{j1}^i \in P_{x_1}^1, \dots, a_{jr}^i \in P_{x_r}^1$ , etc.), and
- (2) if  $j \neq i$ , then  $a_{j(r+1)}^i, \dots, a_{j(s-1)}^i$  lie in the instance  $\mathcal{J}^{k_i}$  (i.e., for all  $z$  such that  $r < z < s$ , we have  $a_{jz}^i \in P_{x_z}^{k_i}$ ).

First, we find the initial part  $(a_{11}^i, \dots, a_{i1}^i)$ .

The first segment  $(a_{11}^i, \dots, a_{1n}^i)$  is found as follows. We start with  $a_{11}^i = a$ . By 1-minimality of  $\mathcal{J}^1$ , we can choose elements  $a_{12}^i \in P_{x_2}^1, \dots, a_{1r}^i \in P_{x_r}^1$  one by one so that  $(a_{1z}^i, a_{1(z+1)}^i)$  is a realization of  $(x_z, x_{z+1})$  in the instance  $\mathcal{J}^1$  for all  $z < r$ . By the choice of  $r$ , we have  $P_{x_r}^1 + (x_r, x_{r+1}) \neq P_{x_{r+1}}^1$  and, by the definition of decomposition,  $P_{x_r}^1 + (x_r, x_{r+1}) = P_{x_{r+1}}$ . In particular,  $P_{x_{r+1}}^{k_i} + (x_{r+1}, x_r)$  intersects  $P_{x_r}^1$ . If  $k_i \neq 1$ , then it follows from the definition of decomposition that  $P_{x_{r+1}}^{k_i} + (x_{r+1}, x_r) = P_{x_r}$ , and we can therefore choose  $a_{1(r+1)}^i \in P_{x_{r+1}}^{k_i}$  such that  $(a_{1r}^i, a_{1(r+1)}^i)$  is a realization of  $(x_r, x_{r+1})$ . If  $k_i = 1$ , we can find  $a_{1(r+1)}^i$  from 1-minimality of  $\mathcal{J}^1$ . Using 1-minimality of  $\mathcal{J}^{k_i}$ , we find  $a_{1(r+2)}^i, \dots, a_{1(s-1)}^i$  such that  $(a_{1(r+1)}^i, \dots, a_{1(s-1)}^i)$  is a realization of  $(x_{r+1}, \dots, x_{s-1})$  in the instance  $\mathcal{J}^{k_i}$ . By the choice of  $s$ , we have  $P_{x_s}^1 + (x_s, x_{s-1}) \neq P_{x_{s-1}}^1$  and, by the definition of decomposition again, we get  $P_{x_s}^1 + (x_s, x_{s-1}) = P_{x_{s-1}}$ ; therefore, there exists

$a_{1s}^i \in P_{x_s}^1$  such that  $(a_{1(s-1)}^i, a_{1s}^i)$  is a realization of  $(x_{s-1}, x_s)$ . Finally, by 1-minimality of  $\mathcal{J}^1$ , we find  $a_{1(s+1)}^i, \dots, a_{1n}^i$  such that  $(a_{1s}^i, \dots, a_{1n}^i)$  is a realization of  $(x_s, \dots, x_n)$  in  $\mathcal{J}^1$ .

By the same argument, we construct the remaining  $(i - 2)$  segments of  $(a_{11}^i, \dots, a_{(i-1)n}^i = a_{i1}^i)$ . These elements satisfy, by construction, both (1) and (2) for every  $j < i$ . Using an analogical reasoning starting from  $b = a_{mn}^i$  (and following the pattern in reverse direction), we can find a realization  $(a_{in}^i = a_{(i+1)1}^i, \dots, a_{mn}^i)$  of the pattern  $(m - i)p$  satisfying (1) and (2) for every  $j > i$ .

It remains to fill in the middle part  $(a_{i1}^i, \dots, a_{in}^i)$  of the  $i$ th row of the matrix. Let  $a' = a_{i1}^i$  and  $b' = a_{in}^i$ . By construction,  $a', b' \in P_x^1$ ,  $a' \in \{a\} + (i - 1)p$  and  $b' \in \{b\} - (m - i)p$ . We observe that  $[\{a'\}]_p$  contains  $b'$ . Indeed, since  $a \in \{a'\} - (i - 1)p$ , we have  $a \in [\{a'\}]_p - (i - 1)p = [\{a'\}]_p$ , then, using  $b \in \{a\} + p$ , we get  $b \in [\{a'\}]_p + p = [\{a'\}]_p$ , and, by  $b' \in \{b\} - (m - i)p$ , we obtain  $b' \in [\{a'\}]_p - (m - i)p = [\{a'\}]_p$ . Recall that  $[B]_p = B + p$  for every  $B \subseteq P_x$ , therefore  $b' \in \{a'\} + p$  and hence we can find a realization  $(a' = a_{i1}^i, a_{i2}^i, \dots, a_{in}^i = b')$  of the pattern  $p$  in the instance  $\mathcal{J}$ . By the choice of  $r$ , we have  $P_{x_z}^1 + (x_z, x_{z+1}) = P_{x_{z+1}}^1$  for all  $z < r$ , which guarantees  $a_{iz}^i \in P_{x_z}^1$  for all  $z \leq r$ . Similarly, by the choice of  $s$ , we have  $a_{iz}^i \in P_{x_z}^1$  for  $z \geq s$  and the construction of the matrix is concluded.

Finally, we apply the operation  $t$  to the columns of the matrix and obtain a tuple  $(b_{11}, \dots, b_{mn})$ . Note that  $b_{11} = a$  and  $b_{mn} = b$  from idempotency of  $t$ . Since every row of the matrix is a realization of the pattern  $mp$  and  $t$  is a polymorphism of the relations  $R_{x_1, x_2}, \dots, R_{x_{n-1}, x_n}$ , the tuple  $(b_{11}, \dots, b_{mn})$  is a realization of  $mp$ . Every element  $b_{jz}$  lies in  $P_{x_z}^1$  since either  $t$  was applied to elements of  $P_{x_z}^1$  (in the case that  $z \leq r$  or  $z \geq s$ ), or  $t$  was applied to  $(a_{jz}^1, \dots, a_{jz}^m)$  where  $a_{jz}^i \in P_{x_z}^{k_i}$  for all  $i \neq j$  (in the case that  $r < z < s$ ) and we can then use the property of  $t$  from the definition of pointed decomposition. Thus  $(a = b_{11}, \dots, b_{mn} = b)$  is a realization of  $mp$  in the instance  $\mathcal{J}^1$ ; therefore,  $b \in \{a\} + {}^1 mp \subseteq A + {}^1 mp = A$  as required.

## REFERENCES

- Libor Barto. 2013. The collapse of the bounded width hierarchy. In preparation.
- Libor Barto and Marcin Kozik. 2009a. Congruence distributivity implies bounded width. *SIAM J. Comput.* 39, 4, 1531–1542. DOI:http://dx.doi.org/10.1137/080743238.
- Libor Barto and Marcin Kozik. 2009b. Constraint satisfaction problems of bounded width. In *Proceedings of the 50th Symposium on Foundations of Computer Science (FOCS'09)*. 595–603.
- Libor Barto and Marcin Kozik. 2010. New conditions for Taylor varieties and CSP. In *Proceedings of the 25th Annual IEEE Symposium on Logic in Computer Science (LICS'10)*. IEEE Computer Society, Los Alamitos, CA, 100–109. DOI:http://dx.doi.org/10.1109/LICS.2010.34.
- Libor Barto and Marcin Kozik. 2012. Robust satisfiability of constraint satisfaction problems. In *Proceedings of the 44th Symposium on Theory of Computing (STOC'12)*. ACM, New York, 931–940. DOI:http://dx.doi.org/10.1145/2213977.2214061.
- Libor Barto, Marcin Kozik, and David Stanovský. 2013. Mal'tsev conditions, lack of absorption, and solvability. Submitted.
- C. Bergman. 2011. *Universal Algebra: Fundamentals and Selected Topics*. Taylor and Francis. <http://books.google.ca/books?id=QXi3BZW0MRwC>.
- Joel Berman, Pawel Idziak, Petar Marković, Ralph McKenzie, Matthew Valeriote, and Ross Willard. 2009. Varieties with few subalgebras of powers. *Trans. Amer. Math. Soc.* 362, 1445–1473. DOI:http://dx.doi.org/10.1090/S0002-9947-09-04874-0.
- Garrett Birkhoff. 1935. On the structure of abstract algebras. *Proc. Cambridge Philos. Soc.* 31, 433–454.
- V. G. Bodnarčuk, L. A. Kalužnin, V. N. Kotov, and B. A. Romov. 1969. Galois theory for Post algebras. I, II. *Kibernetika (Kiev)* 3, 1–10; *ibid.* 1969, no. 5, 1–9.
- Andrei Bulatov. 2006a. Complexity of Maltsev constraints. *Algebra i Logika* 26, 1, (In Russian).



- Andrei Bulatov. 2009. Bounded relational width. Manuscript.  
<http://www.cs.sfu.ca/~abulatov/papers/relwidth.pdf>.
- Andrei Bulatov and Víctor Dalmau. 2006. A simple algorithm for Mal'tsev constraints. *SIAM J. Comput.* 36, 1, 16–27 (electronic).
- Andrei Bulatov, Peter Jeavons, and Andrei Krokhin. 2005. Classifying the complexity of constraints using finite algebras. *SIAM J. Comput.* 34, 720–742. Issue 3.  
 DOI:<http://dx.doi.org/10.1137/S0097539700376676>.
- Andrei A. Bulatov. 2003. Tractable conservative constraint satisfaction problems. In *Proceedings of the 18th Annual IEEE Symposium on Logic in Computer Science*. IEEE Computer Society, Los Alamitos, CA, 321–. DOI:<http://dl.acm.org/citation.cfm?id=788023.789067>.
- A. A. Bulatov. 2004a. Complexity of the conservative generalized satisfiability problem. *Dokl. Akad. Nauk* 397, 5, 583–585.
- Andrei A. Bulatov. 2004b. A graph of a relational structure and constraint satisfaction problems. In *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science (LICS'04)*. IEEE Computer Society Press, Los Alamitos, CA, 448–457.
- Andrei A. Bulatov. 2006b. Combinatorial problems raised from 2-semilattices. *J. Algebra* 298, 2, 321–339.
- Andrei A. Bulatov. 2006c. A dichotomy theorem for constraint satisfaction problems on a 3-element set. *J. ACM* 53, 1, 66–120.
- Andrei A. Bulatov and Matthew Valeriote. 2008. Recent results on the algebraic approach to the CSP. In *Complexity of Constraints*, Nadia Creignou, Phokion G. Kolaitis, and Heribert Vollmer Eds., Lecture Notes in Computer Science, vol. 5250, Springer, 68–92.  
<http://dblp.uni-trier.de/db/conf/dagstuhl/coc2008.html#BulatovV08>.
- Andrei A. Bulatov, Andrei A. Krokhin, and Peter Jeavons. 2000. Constraint satisfaction problems and finite algebras. In *Automata, Languages and Programming*. Lecture Notes in Computer Science, vol. 1853, Springer, Berlin, 272–282.
- Andrei A. Bulatov, Andrei Krokhin, and Benoît Larose. 2008. Dualities for constraint satisfaction problems. In *Complexity of Constraints*, Springer-Verlag, Berlin, 93–124.  
 DOI:[http://dx.doi.org/10.1007/978-3-540-92800-3\\_5](http://dx.doi.org/10.1007/978-3-540-92800-3_5).
- Stanley N. Burris and H. P. Sankappanavar. 1981. *A Course in Universal Algebra*. Graduate Texts in Mathematics, Vol. 78, Springer-Verlag, New York.
- Catarina Carvalho, Víctor Dalmau, Petar Marković, and Miklós Maróti. 2009. CD(4) has bounded width. *Algebra Universalis* 60, 3, 293–307.
- Tomás Feder and Moshe Y. Vardi. 1998. The computational structure of monotone monadic SNP and constraint satisfaction: A study through datalog and group theory. *SIAM J. Comput.* 28, 1, 57–104.  
 DOI:<http://dx.doi.org/10.1137/S0097539794266766>.
- David Geiger. 1968. Closed systems of functions and predicates. *Pacific J. Math.* 27, 95–100.
- David Hobby and Ralph McKenzie. 1988. *The Structure of Finite Algebras*. Contemporary Mathematics, Vol. 76, American Mathematical Society, Providence, RI.
- Paweł Idziak, Petar Marković, Ralph McKenzie, Matthew Valeriote, and Ross Willard. 2007. Tractability and learnability arising from algebras with few subpowers. In *Proceedings of the 22nd Annual IEEE Symposium on Logic in Computer Science (LICS'07)*. IEEE Computer Society Press, Los Alamitos, CA, 213–222.
- Peter Jeavons, David Cohen, and Marc Gyssens. 1997. Closure properties of constraints. *J. ACM* 44, 4, 527–548.
- Emil Kiss and Matthew Valeriote. 2007. On tractability and congruence distributivity. *Log. Meth. Comput. Sci.* 3, 2, 2:6, (electronic).
- Benoît Larose and Pascal Tesson. 2009. Universal algebra and hardness results for constraint satisfaction problems. *Theoret. Comput. Sci.* 410, 18, 1629–1647.  
 DOI:<http://dx.doi.org/10.1016/j.tcs.2008.12.048>.
- Benoît Larose and László Zádori. 2007. Bounded width problems and algebras. *Algebra Universalis* 56, 3–4, 439–466.
- Benoît Larose, Matt Valeriote, and László Zádori. 2009. Omitting types, bounded width and the ability to count. *Internat. J. Algebra Comput.* 19, 5, 647–668.  
 DOI:<http://dx.doi.org/10.1142/S021819670900524X>.
- Miklós Maróti and Ralph McKenzie. 2008. Existence theorems for weakly symmetric operations. *Algebra Universalis* 59, 3–4, 463–489.

Received March 2012; revised February 2013; accepted November 2013