Solving CSPs using weak local consistency *

Marcin Kozik

Theoretical Computer Science Department
The Faculty of Mathematics and Computer Science
Jagiellonian University

June 11, 2018

Abstract

The characterization of all the CSPs solvable by local consistency checking (lcc CSPs for short), proposed by Feder and Vardi [SICOMP'98], was confirmed in [Bulatov'09] and independently in [FOCS'09, JACM'14]. Later, a collapse of the hierarchy of local consistency notions [JLC'14] was achieved by showing that (2,3) minimality solves all the lcc CSPs.

In this paper we show that a weaker consistency notion, *jpq* consistency, is sufficient to solve these problems. Our notion is strictly weaker than (2,3) consistent, (2,3) minimality, path consistency and Singleton Arc Consistency (SAC). This last fact allows us to answers the question posed in [JLC'13] and, as known algorithms work faster for SAC then for the other notions, implies that lcc CSPs can be solved more efficiently.

The definition of jpq consistency is closely related to consistency obtained from rounding an SDP relaxation of a CSP problem. In fact the main result of this paper is necessary in the proof showing that CSPs with a near unanimity polymorphisms admit robust approximation algorithms with polynomial loss [SODA'17]. Finally the *jpq* consistency notion implies new algebraic condition characterizing the algebras of lcc CSPs i.e. the algebras in congruence meet semi-distributive varieties. The new characterization aligns nicely with one of the tractable cases of Promise CSPs [SODA'18].

1 Introduction

Algorithms verifying various notions of local consistency have long history in computer science. The usual objective of such an algorithm is to recognize, at a low computational cost, instances which are unsolvable for "local" reasons. Local consistency checking often serves as one of the first stages of an algorithm and allows to quickly disregard a fraction of inputs.

Local consistency checking algorithms play a specially important role in the context of Constraint Satisfaction Problem. An instance of the Constraint Satisfaction Problem consists of variables and constraints. In the decision version of CSP the question is whether the variables can be evaluated in such a way that all the constraints, often described as a relation constraining a sequence of variables, are satisfied.

In a seminal paper [23] Feder and Vardi proposed to parametrize the problem by restricting the constraining relations allowed in instances. More formally, for every finite relational structure

*Research partially supported by National Science Center under grant UMO-2014/13/B/ST6/01812. Preliminary version of this paper (proving a weaker result) appeared in LICS 2016.

A (called in this context *a template*) the CSP parametrized by **A**, CSP(**A**), is the CSP restricted to instances with all the constraint relations taken from **A**. Clearly, for any **A**, the problem CSP(**A**) is in NP and it is quite easy to construct relational structures which define NP-complete CSPs or CSPs solvable in polynomial time. One of the main problems in the area, the famous *the CSP Dichotomy Conjecture* [23] postulates that every **A** defines CSP(**A**) which is NP-complete or solvable in a polynomial time. Two recent, independent results of Bulatov [14] and Zhuk [31] confirm this conjecture.

The class of problems which can be expressed as a CSP(**A**) is very rich; it is easy to construct relational structure **A** such that CSP(**A**) is 2-colorability of graphs, 3-SAT, 3-Horn-SAT, or a problem of solving systems of linear equations in $\mathbb{Z}_2$. The last problem on this list is a canonical example of a CSP with an *ability to count*. Feder and Vardi [23] conjectured that all the CSPs which do not have the ability to count are solvable by local consistency checking i.e. that some form of local consistency is able to recognize all the instances with no solutions. The algebraic approach to CSP allowed to formalize [28] and eventually confirm this conjecture [1, 5, 13].

The proof [1, 5], however, require (k,l)-consistency (compare Definition 3.1 in [3]) with k,l dependent on the maximal arity of relations in the template. The consistency was substituted by a weaker one in [3] showing that establishing so called $(2,3)$ minimality solves these CSPs. All these results, as well as [13], even when restricted to templates with binary constraints require some version of path consistency.

Path consistency is a local consistency notion which is costly to establish; it introduces auxiliary constraints for each pair of variables which impacts the performance of the algorithm (it runs in time which is cubic with respect to the number of variables [22]) and destroys the structure of an instance. For example big and sparse instances are becoming, in essence, cliques and all the structural advantage is lost. This paper shows that the same can be accomplished using a significantly weaker consistency notion.

Arc consistency is another, perhaps the most recognized consistency notion. Arc consistency is weaker than path consistency and can be computed efficiently. It does not, however, solve all the lcc CSPs (as seen from e.g. classical Schaefer's result [29]). A next, well established, consistency notion is Singleton Arc Consistency (SAC). It is easier to verify than path consistency and the verification do not distort the structure of the instance. In [17] authors discuss applicability of SAC to lcc CSPs and ask if every such CSP is solvable by SAC. The main result of this paper answers the question as the *jpq* consistency is weaker than SAC.

It isn't always the case that the consistency of an instance is verified by a direct application of an algorithm. In case of robust algorithms providing approximation for CSPs (compare e.g. [25, 19, 4]) the consistency is obtained by rounding a solution to an SDP relaxation of a CSP instance. Only specific and usually quite weak consistency notions can be provided by such rounding. In particular the *jpq* consistency notion is the only notion that we were able to provide in order to show that all the templates with near unanimity polymorphisms have robust approximation algorithms with polynomial loss [21].

On the algebraic side of the connection the lcc CSPs correspond to algebras generating varieties with meet semi-distributive congruences (SD($\wedge$) algebras for short). Using the main result of the paper we are able to immediately reprove some recent term condition [26] for SD($\wedge$) algebras. Moreover we are able to provide new, equivalent term conditions which align nicely with a tractability class for a particular family of Promise CSPs [12].

The paper is organized as follows. In the next section we introduce some background, define *jpq* consistency and state the main result. Section 3 contain a comparison of pertinent consistency notions. The algebraic consequences are proved in Section 4. Finally the proof of the main theorem starts in Section 5 and splits into two cases analyzed in Sections 6 and 7. The last section contains acknowledgments.

2 The main result

The computational problem CSP($\mathbf{A}$) is defined for each finite relational structure $\mathbf{A}$ called, in this context, *a template*. *An instance* of CSP over template $\mathbf{A}$ consists of a set of variables and a set of constraints which are pairs of the form $((x_1,\ldots,x_n),R)$ where each x_i is a variable and R is an n-ary relation in $\mathbf{A}$. *A solution* of such an instance is a function f sending variables to the universe of $\mathbf{A}$ (usually denoted by A) in such a way that for every constraint $((x_1,\ldots,x_n),R)$ the tuple $(f(x_1),\ldots,f(x_n)) \in R$. Every such an instance can be equivalently presented as a relational structure in the language of the template; in this presentation solvable instances correspond to relational structures with homomorphisms to $\mathbf{A}$.

Let $\mathcal{I}$ and $\mathcal{J}$ be CSP instances and φ be a function mapping variables of $\mathcal{I}$ to variables of $\mathcal{J}$ and constraints of $\mathcal{I}$ to constraints of $\mathcal{J}$; if every constraint $((x_1,\ldots,x_n),R)$ in $\mathcal{I}$ is mapped by φ to a constraint of $\mathcal{J}$ of the form $((\varphi(x_1),\ldots,\varphi(x_n)),R)$, then φ is the *instance homomorphism* from $\mathcal{I}$ to $\mathcal{J}$. Instance homomorphism from $\mathcal{I}$ to $\mathcal{J}$ can be equivalently defined as a homomorphism between relational structures which $\mathcal{I}$ and $\mathcal{J}$ essentially are.

All of the local consistency notions can be expressed in terms of instance homomorphisms. More precisely a local consistency notion can be defines by family of instances (usually given by a particular structural condition e.g. a bound on the tree-width). Checking consistency of an instance $\mathcal{I}$ is equivalent to verifying if all the instances from this family with instance homomorphisms to $\mathcal{I}$ have solutions (compare e.g. [15]). The notion of *jpq* consistency will be defined in this way as well.

A standard notion of the *adjacency multigraph* of an instance allows us to define structural properties of instances. The vertex set of the adjacency multigraph of an instance consists of all the variables and all the constraints and one edge between a variable vertex and a constraint vertex is introduced for each time the variable appears in the constraint (in particular the multigraph is bipartite). An instance is *a tree (forest) instance* if it's adjacency multigraph is a tree (forest) i.e. has no multiple edges and no cycles. It is easy to see that the arc consistency checking algorithm accepts instance $\mathcal{I}$ if and only if every tree (equivalently forest) instance that maps homomorphically to $\mathcal{I}$ has a solution.

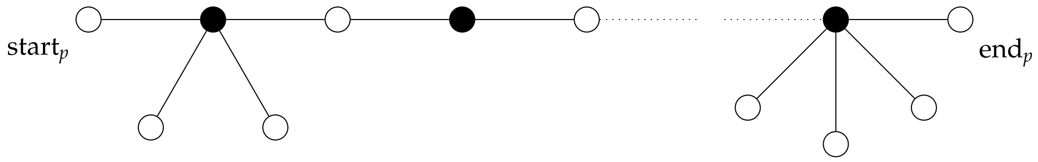
2.1 Patterns in instances

Patterns are instances, with some added structure, which allow easier characterization of local consistency conditions. The definition of a pattern in this paper is semantically very similar to the definitions in [5, 8]. A pattern p in an instance $\mathcal{I}$ is an instance, together with:

- a homomorphism $\hom_p$ to from p to $\mathcal{I}$;
- a set of variables start_p (of the instance p) such that $|\hom_p(\mathrm{start}_p)| = 1$;
- one more distinguished variable end_p which is outside of the set start_p.

Moreover we say that a pattern p is from x to y is $\hom_p(\mathrm{start}_p) = x$ and $\hom_p(\mathrm{end}_p) = y$.

A pattern p is a *tree pattern* if it is defined by a tree instance, and the variables in start_p and end_p are leaves in the adjacency tree of p. A tree pattern p is a *path pattern* if start_p is a one element set and the adjacency graph of p is a path from start_p to end_p with possible hanging variable vertices (note that if all the constraints in a path pattern are binary there is no hanging variables i.e. the adjacency graph of a path pattern is a path). The following graph depicts the structure of a path pattern (solid vertices correspond to constraints):



Let p, q be patterns in the same instance. If $\text{hom}_p(\text{end}_p) = \text{hom}_q(\text{start}_q)$ we define a pattern $p + q$ in the following way:

1. $p + q$ includes q and end_{p+q} is equal to end_q;
2. for each v in start_q we introduce (into $p + q$) a fresh copy of p and
   - identify v with end_p of this fresh copy;
   - add all the vertices of start_p to start_{p+q};
   - define hom_{p+q} on the new vertices according to hom_p.

It is clear that pattern addition is associative i.e. $p + (q + r) = (p + q) + r$ whenever the condition on starts and ends of p, q, r is satisfied. If p, q are path patterns then so is $p + q$, moreover for a path pattern p we denote by $-p$ the same pattern, but with the roles of start_p and end_p exchanged. For path patterns p and q, we put $p - q$ to be $p + (-q)$ and occasionally write kp instead of $p + p + \cdots + p$ (this is well defined for tree patterns as well).

2.2 Propagation via patterns

Let p be a pattern; the set $B + p$ (the *propagation of B via p*) consists of all the values end_p takes in the solutions of p which send every variable in start_p into B. Note that $(B + p) + q = B + (p + q)$ (where in the first case there are two propagations, and in the second one propagation, but with the sum of patterns).

2.3 The *jpq* consistency and the main theorem

One of the most basic and well established (compare e.g. [20]) consistency notions is called *arc consistency* (sometimes *generalized arc consistency* to account for the presence of constraints of arity greater than two).

Definition 2.1 (Arc consistency)**.** *The instance $\mathcal{I}$ is* arc consistent *if for any constraints $((x_1, \ldots, x_n), R)$ and $((y_1, \ldots, y_m), S)$ such that $x_i = y_j$ we have* $\text{proj}_i(R) = \text{proj}_j(S)$*.*

We say that the instance $\mathcal{I}$ is arc consistency with sets $\{A_x\}$ *(where x ranges over variables of $\mathcal{I}$), when $\mathcal{I}$ is arc-consistent and the unique set the arc-consistency assigns to a variable x is A_x.*

The notion of *jpq* consistency is strictly stronger then arc-consistency:

Definition 2.2. *Let $\mathcal{I}$ be an arc consistent instance with sets $\{A_x\}$. The instance $\mathcal{I}$ is jpq consistent if for any variable x, any $a \in A_x$ and any path patterns p, q both starting and ending at x there exists j such that $a \in \{a\} + j(p + q) + p$.*

An algorithm, checking a local consistency notion, verifies whether an instance given on input contains a subinstance which is consistent. *A subinstance* has the same variables and constraints as the original instance; the only difference is that the constraining relations of a subinstance can be smaller (in particular if a subinstance has a solution so does the instance). The algorithm usually proceeds by "trimming" the instance (removing, from constraining relations, tuples which falsify the consistency) until the condition is satisfied or an empty instance is produced. In Section 3 we take a closer look at various consistency notions and the algorithms that produce corresponding consistent instances..

By its nature the local consistency checking algorithms do not produce false-negatives. We say that local consistency *solves CSP* when the positive answers are also correct i.e. each consistent instance has a solution. The CSP problems solvable by local consistency checking were characterized in [13, 1, 5], but each time the consistency notions were rather strong (see discussion in Section 3). The main result of this paper states that establishing *jpq* consistency is sufficient.

Theorem 2.3. *Let* $\mathbf{A}$ *be a relational structure defining* CSP($\mathbf{A}$) *solvable by local consistency checking. Then jpq consistency solves* CSP($\mathbf{A}$).

3 Comparison of (selected) local consistency notions

There exists an extensive literature on various local consistency notions used in CSP (compare e.g. [22, 11, 17]). In this section we present only these notions which are directly pertinent to our approach. DATALOG (with a goal predicate) and its restrictions offer a natural way of defining some local consistency notions.

As discussed in previous section an instance of CSP can be viewed as a relational structure in the signature of the template. A DATALOG program derives facts about this relational structure using a set of rules (which are fixed for a fixed template). The rules are defined using relations of the template called *extensional database* or EDBs enhanced by a number of auxiliary relations called *intensional database* or IDBs. Each rule consists of a *head* which is a single IDB on an appropriate number of variables and the *body* which is a sequence on IDBs and EDBs. The execution of the program updates the head IDB whenever the body of the rule is satisfied. The computation ends when no relation can be updated, or when the goal predicate is reached.

Example. The following DATALOG program operates on digraphs. The edge relation of a digraph (denoted by E) is the single binary EDB in the program. The program uses two IDBs: ODD and $GOAL$ (the goal predicate) and verifies whether the digraph has a directed cycle of odd length.

$$ODD(x,y) :- E(x,y)$$
$$ODD(x,v) :- E(x,y) \wedge E(y,z) \wedge ODD(z,v)$$
$$GOAL :- ODD(x,x)$$

The program computes the relation ODD and fires the $GOAL$ predicate when a directed circle of odd length is found.

3.1 Arc consistency

Arc consistency (introduced in Definition 2.1) is one of the most basic local consistency notions. The algorithm verifying arc consistency is conceptually very simple. It starts with an instance $\mathcal{I}$ and the family $\{A_x\}$ where each A_x is initially equal to the full universe of the template. Whenever, for some constraint $((x_1,\dots,x_n),R)$ in $\mathcal{I}$ and some i, we have $\mathrm{proj}_i(R) \neq A_{x_i}$ we restrict A_{x_i} to $\mathrm{proj}_i(R)$ and update all the constraints involving x_i by removing tuples sending x_i to elements outside the new A_{x_i}. This procedure terminates with a maximal (possibly empty) arc consistent subinstance of $\mathcal{I}$.

An equivalent definition of arc consistency uses DATALOG: if we require that, in the DATALOG program, every IDB is a unary relation its expressive power reduces to verifying exactly arc consistency (note that the goal predicate is fired whenever the instance is inconsistent). Already Schaefer's classification [29] shows that the arc consistency is not strong enough to solve all the lcc CSPs. In fact all the CSPs solvable by arc consistency are characterized in [20].

Note that, for a fixed template, arc-consistency can be computed quite quickly i.e. in the time linear with respect to the number of constrains. But, at the same time, a PTIME-complete CSP: Horn-3SAT is solvable by arc consistency.

3.2 Notions solving all lcc CSPs

The conjecture of Feder and Vardi [23] characterizing CSPs solvable by local consistency checking was first confirmed in [13, 1, 5] but the consistency notions used there were unnecessarily strong. The paper of Barto [3] improved these results, by showing that providing $(2, 3)$ minimality is sufficient for all the CSPs in this class. One of the consequences of this fact was a collapse of hierarchy of CSPs requiring potentially stronger consistency notions.

An algorithm verifying $(2, 3)$ minimality uses the same principle as the arc consistency algorithm in the previous section. The difference is that instead of restricting sets A_x we restrict sets A_{xy} for every pair of variables x, y and that we additionally require that each triple of variables is constrained by some constraint.

For templates containing only binary relations all the notions discussed in this section are equivalent and correspond to well-studied path consistency. Algorithms verifying this consistency work in time cubic with respect to the number of variables [22]; moreover, since the algorithm introduces set A_{xy} for each pair of variables, sparse instances (which could be otherwise analyzed) loose their structure and can become intractable in practice.

More importantly, in some cases, the strong consistency notions just cannot be provided. In [21] instance is constructed by rounding a solution to an SDP relaxation of a CSP instance and the rounding procedure does not produce any of the strong notions discussed in this section.

3.3 Singleton Arc Consistency (SAC)

The notion of singleton arc consistence was introduced in [22]. The simplest algorithm establishing SAC is a modification of the arc consistency algorithm. As with the arc consistency we update sets $\{A_x\}$ (and the instance) by removing elements. In the arc consistency algorithm removing the element was decided by verifying a single constraints, in SAC it is decided by a subroutine. The subroutine takes x and $a \in A_x$ and runs arc consistency on the instance with added constraint "$x = a$"[1]. If the subroutine computes an empty instance a is removed from A_x and the instance is updated.

It is easy to see that every path consistent (binary) instance is necessarily SAC, and that every SAC instance is arc consistent. The reverse implications do not hold. SAC offers some computational advantages over path consistency: it is possible to verify SAC in time bounded by a constant times the number of constraints times the number of variables [10]. More importantly SAC does not alter the structure of the instance and therefore runs efficiently on sparse instances with many variables.

SAC has been studied in [17] where the authors ask whether it solves all lcc CSPs. As the notion of *jpq* consistency is weaker than SAC the result of this paper answers this question in affirmative.

[1] All the changes to the instance performed by the subroutine are disregarded.

3.4 Singleton Linear Arc Consistency (SLAC)

The Singleton Linear Arc Consistency is the notion studied in [27] which is the conference predecessor of this paper. SLAC is defined in a way similar to Definition 2.2 but with only one pattern, say p, and the condition $a \in \{a\} + p$.

The name of this notion is motivated by an algorithm which can be used to provide it: Linear Arc Consistency (LAC) is the consistency that can be computed using a DATALOG program similar to the one for arc consistency, but allowing only the rules with no more than one IDB in the body. SLAC can be provided by an algorithm similar to the one described for SAC in Section 3.3 but with the subroutine running LAC instead of arc consistency. The notion of LAC originates in [11, 18] and has computational advantages over arc consistency, most importantly it can be computed in NL.

3.5 The *jpq* consistency

The *jpq* consistency is the central notion of this paper. It is weaker than the consistencies from Section 3.2, SAC or SLAC. The definition of *jpq* consistency originates in [21], where a CSP instance is obtained from rounding a solution to an SDP relaxation of a CSP instance. We were unable to (in [21]) to construct a rounding that would produce an instance satisfying any of the consistency notions stronger than *jpq* consistency and thus the result of [21] requires Theorem 2.3 (together with some technical facts from Section 7) in the proof of correctness.

The next section contains further applications of Theorem 2.3 and the notion of *jpq* consistency. It reproves a recent result [26] introducing a new term condition for algebras corresponding to lcc CSPs, and introduces new term conditions for the same class.

4 Algebraic consequences

All of the result [13, 1, 5] make heavy use of the tools available via the algebraic approach to CSP. The same techniques will be employed in this paper, but we refrain from introducing them, and instead refer a reader to the excellent survey [9] and to a classical text on universal algebra [16]. Due to the nature of this section it requires more algebraic background than the remainder of the paper. As the results proved here are consequences of the main theorem of the paper and are of algebraic flavor the section can be omitted with no impact on other sections.

At the heart of the classification results for lcc CSPs lies the fact that, for core relational templates, they correspond to algebras generating varieties with meet semi-distributive congruence lattices – SD($\wedge$) algebras for short. Vice versa, the development of consistency notions can be used to obtain new information about the structure of these algebra.

In this section we apply *jpq* consistence to reprove one of the main results of [26] which follows immediately (comp. Corollary 4.2) from Theorem 2.3. Moreover, in Corollary 4.3, we present a new term condition characterizing finite SD($\wedge$) algebras (or locally finite SD($\wedge$) varieties). The condition is interesting due to its similarity to one of the tools providing tractability of PCSPs in [12]. We start the section with a technical lemma.

Lemma 4.1. *Let $\mathbb{A}$ be an SD($\wedge$) algebra, $a, b \in \mathbb{A}$ and $R \subseteq \{a, b\}^n$ be such that* $\mathrm{proj}_{ij}(R)$ *is a graph on $\{a, b\}$ with no sources, no sinks, and loop at a. The subalgebra of $\mathbb{A}^n$ generated by R contains a constant tuple.*

Proof. Let $\mathbb{R}'$ denote the subalgebra of $\mathbb{A}^n$ generated by R. We will show that $((x, \dots, x), \mathbb{R}')$ is a *jpq* instance and use Theorem 2.3 to conclude that it has a solution, which immediately implies that $\mathbb{R}'$ has a constant tuple.

The first step is to show that $((x,\dots,x),R)$ is a *jpq* instance. The arc consistency of the instance is clear. In order to establish *jpq* consistency we investigate the digraphs defined by path patterns in this instance. Let p be a pattern; if the instance of p has single constraint then the set of solutions of p is essentially R, and its restriction to $(\hom_p(\text{start}_p), \hom_p(\text{end}_p))$ is $\text{proj}_{ij}(R)$ for some i,j i.e. a graph on $\{a,b\}$ with no sources, no sinks and loop at a. If p has more than a one constraint then restricting the set of solutions of p to $(\hom_p(\text{start}_p), \hom_p(\text{end}_p))$ we get a graph which is obtained from a composition of the graphs defined by $\text{proj}_{ij}(R)$. Such graphs have no sources, no sinks and loops at a.

There are five graphs on $\{a,b\}$ satisfying our assumptions and it is an easy exercise to verify that in all the cases $a \in \{a\} + (p+q) + p$ and $b \in \{b\} + 2(p+q) + p$ and therefore $((x,\dots,x),R)$ is a *jpq* instance.

Consider an instance with a single constraint $((x,\dots,x),\mathbb{R}')$. It is arc consistent: indeed the projection of $\mathbb{R}'$ to any coordinate is the subalgebra of $\mathbb{A}$ generated by $\{a,b\}$. Let c be any element of this subalgebra, clearly $c = t(a,b)$ for some term t. Now for any patterns patterns p,q we have $a \in \{a\} + 2(p+q) + p$ and $b \in \{b\} + 2(p+q) + p$ and applying (coordinatewise) the term t to the solutions of these pattern we conclude that $c = t(a,b) \in \{t(a,b)\} + 2(p+q) + p$ i.e $((x,\dots,x),\mathbb{R})$ is a *jpq* consistent instance which concludes the proof. ◻

Corollary 4.2 (Theorem 3.2 of [26]). Every finite SD(∧) algebra has an idempotent term t satisfying

$$t(y,x,x,x) = t(x,y,x,x) = t(x,x,y,x) = t(x,x,x,y) = t(y,y,x,x) = t(y,x,y,x) = t(x,y,y,x).$$

Proof. Let $\mathbb{A}$ be a finite SD(∧) algebra; the idempotent reduct of $\mathbb{A}$ is also SD(∧). Let $\mathbb{F}$ be the free algebra on two generators in the variety generated by this idempotent reduct and let $\mathbb{R} \leq_{\text{sub}} \mathbb{F}^7$ be generated by $(y,x,x,x,y,y,x), (x,y,x,x,y,x,y), (x,x,y,x,x,y,y), (x,x,x,y,x,x,x)$. The generators satisfy the assumptions of Lemma 4.1 which implies that $\mathbb{R}$ has a constant tuple. The term generating this tuple is satisfying all the required identities. ◻

The next corollary defines a new term condition for SD(∧). We omit its proof as it is identical to the proof of Corollary 4.2.

Corollary 4.3. Let $\mathbb{A}$ be an SD(∧) algebra. For any odd n there is an n-ary term t satisfying all the identities (in x and y) of the form

$$t(\underbrace{x,x,y,x,y,\dots,x,x}) = t(\underbrace{y,x,y,x,x,\dots,y,x})$$

when x and y appear on both sides, and the same variable has majority on the left and on the right.

Clearly, each such term of arity at least five, characterizes the class of SD(∧) algebras. The corollary can be extended to even arities (adding identities similar to these in Corollary 4.2) at the price of complicating the definition.

5 Into the proof of Theorem 2.3

In order to prove the main results of the paper we follow the general structure of the proofs in [1, 5, 3]. The notion of *jpq* consistency is weaker then the notions used there, and therefore we are unable to simplify the paper by citing appropriate results from earlier papers. In particular the reasoning presented is a stand alone proof of the bounded width conjecture of Feder and Vardi [23] as stated in [28].

In order to prove Theorem 2.3 we perform a number of standard reductions found already in [23]. First reduction allows us to assume that **A** is a *core* that is every endomorphism of **A** is a bijection. Indeed it is easy to see that every finite relational structure maps homomorphically to a core which can be found as its induced substructure. Moreover an instance of **A** is solvable if and only if the appropriate instance for the core of **A** is solvable, and the consistency algorithms report identical results on both.

The proof of Theorem 2.3 proceeds as follows. We start with an arbitrary template **A** solvable by local consistency checking and without loss of generality, assume that **A** is a core. Using [28] we conclude that adding constants to **A** (for each element of A we introduce a unary relation defining this element) does not change the property of being solvable by local consistency algorithm. The algebra associated to such an enriched **A** is idempotent (i.e. all its operations satisfy $t(x, \dots, x) = x$) and therefore (by the hardness result of [28]) needs to be an SD($\wedge$) algebra.

Take any instance over such a template; the local consistency checking algorithm reduces this instance to a *jpq* consistent subinstance (or such an instance can be obtained in another way as in e.g. [21]). The *jpq* subinstance is algebraically closed i.e. it inherits the algebraic structure as the template. If every such instance has a solution then verifying *jpq* consistency solves CSP for **A** enriched with constants and thus for the original **A** as well. This reasoning reduces Theorem 2.3 to Theorem 5.1 below.

Theorem 5.1. *Let* **A** *be a relational structure such that* $\mathbb{A}$ *is idempotent and SD(*$\wedge$*). Every jpq consistent instance over* **A** *has a solution.*

The proof of Theorem 5.1 hinges on a slightly non-standard definition: a subalgebra $\mathbb{B} \leq \mathbb{C}$ is *absorbing* if there is an operation of $\mathbb{C}$, say $f : C^n \to C$, such that $f(c_1, \dots, c_n) \in \mathbb{B}$ if for at most one i we have $c_i \notin \mathbb{B}$.

Proof of Theorem 5.1. We fix a template **A** and the associated algebra $\mathbb{A}$ which is idempotent and SD($\wedge$). Moreover we fix a *jpq* consistent instance $\mathcal{I}$ and sets $\{A_x\}$ so that $\mathcal{I}$ is arc-consistent with $\{A_x\}$. We know that each A_x is a subuniverse of $\mathbb{A}$ (we will denote them by $\mathbb{A}_x$) and that the constraining relations are subpowers of $\mathbb{A}$ as well.

The general structure of the proof is identical to the one in [1, 5, 3] i.e.: if each $\mathbb{A}_x$ is a one-element-algebra the arc-consistency of $\mathcal{I}$ provides a solution; otherwise we produce $\mathbb{A}'_x \leq \mathbb{A}_x$ (we require that for at least one x the subalgebra is proper) so that restricting the instance $\mathcal{I}$ to $\{A'_x\}$'s (i.e. substituting each constraint $((x_1, \dots, x_n), R)$ with $((x_1, \dots, x_n), R \cap \prod_i \mathbb{A}'_{x_i})$) produces a subinstance which is *jpq* consistent. Throughout the proof subinstance is denoted by $\mathcal{I}'$.

As in [1, 5, 3] the reasoning splits into two cases depending on the algebraic structure of $\mathbb{A}_x$'s: either no $\mathbb{A}_x$ has a proper absorbing subalgebra, or at least one does. The first case is tackled in Section 6 and the second in Section 7. ◻

Before we are ready to launch into the proof we need to establish some basic combinatorial properties of *jpq* consistent instances.

5.1 Properties of *jpq* consistent instances

This first corollary of the definition of the *jpq* instance appeared as property (P3) in [3].

Corollary 5.2. Let $\mathcal{I}$ be a *jpq* consistent instance, and p, q patterns starting and ending at x. If $A' + p = A''$ and $A'' + q = A'$ then $A' = A''$.

Proof. Assume, for a contradiction that the corollary fails. If there exists $a \in A' \setminus A''$, then by the assumptions of the corollary $a \notin \{a\} + j(p + q) + p$ for any j (as the set is a subset of A''). This, however, contradicts the definition of *jpq* consistency for $\mathcal{I}$. The case of $a \in A'' \setminus A'$ is exactly the same. □

Which leads to the next corollary.

Corollary 5.3. Let the instance $\mathcal{I}$ be *jpq* consistent with the sets $\{A_x\}$, and p, q patterns starting and ending at x. For all sufficiently large j and for every $a \in A_x$ we have $a \in \{a\} + j(p+q) + p$.

Proof. Fix an arbitrary $a \in A_x$ and consider the sequence of sets $A_0 = \{a\}, A_{2i+1} = A_{2i} + p$ and $A_{2i+2} = A_{2i+1} + q$. If, for some $i < j < k$, we have $A_i = A_k \neq A_j$ we contradict Corollary 5.2 and since A is finite the sequence of sets needs to be eventually constant.

We apply the definition of *jpq* instance to a and the pattern $k(p+q)$ taken in place of both p and q and conclude that the set at which the sequence stabilises contains a. This in turn implies that for all sufficiently large j we have $a \in \{a\} + j(p+q) + p$. Now if j is large enough to work for all $a \in A_x$ we satisfy the claim. □

6 Reduction with no absorption

This section presents a construction of the subinstance in the case when none of $\mathbb{A}_x$ has an absorbing subuniverse. Note that all the algebras appearing in this section belong to the variety generated by $\mathbb{A}$ and therefore are SD($\wedge$) algebras. However the algebraic statements are true independently from the context of this variety and their presentation is greatly simplified by the fact that a product of a finite family of finite SD($\wedge$) algebras is an SD($\wedge$) algebra (compare e.g. [7, 24]).

In a product, say $\mathbb{R} \leq \prod_{i \in I} \mathbb{B}_i$ the projections (e.g. $\mathrm{proj}_J \mathbb{R}$ for some $J \subseteq I$) are defined naturally and inherit the algebraic structure from $\mathbb{R}$. Moreover we let π_i to define the congruence identifying in $\mathbb{R}$ tuples with the same element on the i-th coordinate. Consider $\mathbb{R} \leq_{\mathrm{sub}} \mathbb{B} \times \mathbb{C}$ and the congruence $\pi_1 \vee \pi_2$. It is clear that $\pi_1 \vee \pi_2$ induces a congruence, say β, on $\mathbb{B}$ by putting $b \mathrel{\beta} b'$ if and only if there exists c, c' such that (b, c) and (b', c') are related in $\pi_1 \vee \pi_2$. We call such β *the left congruence of* $\mathbb{R}$ and analogous congruence on $\mathbb{C}$ *the right congruence of* $\mathbb{R}$.

We will often use the fact (which is a basic consequence of the definition of the absorption) that if $\mathbb{R}' \trianglelefteq \mathbb{R} \leq \mathbb{B} \times \mathbb{C}$ and $\pi_1 \vee \pi_2$ is the full congruence on $\mathbb{R}$, then $\pi_1 \vee \pi_2$ (this time the congruences and their join are on $\mathbb{R}'$) is the full congruence on $\mathbb{R}'$. The following algebraic fact [2] is used to define the instance in the case of no absorption.

Theorem 6.1 (Theorem III.6 [2])**.** *Let $\mathbb{R} \leq_{sub} \mathbb{B} \times \mathbb{C}$ be an idempotent Taylor algebra. If $\pi_1 \vee \pi_2$ is the full congruence on $\mathbb{R}$ then*

- $\mathbb{R} = \mathbb{B} \times \mathbb{C}$, *or*
- $\mathbb{B}$ *has a proper absorbing subalgebra or*
- $\mathbb{C}$ *has a proper absorbing subalgebra.*

Note that if Theorem 6.1 provides a proper absorbing subuniverse in $\mathbb{B}$ or in $\mathbb{C}$ we can restrict $\mathbb{R}$ to pairs intersecting this subuniverse and reapply the theorem. After finitely many applications we obtain $\mathbb{B}' \trianglelefteq \mathbb{B}$ and $\mathbb{C}' \trianglelefteq \mathbb{C}$ such that $\mathbb{B} \times \mathbb{C} \trianglelefteq \mathbb{R}$ and neither $\mathbb{B}'$ nor $\mathbb{C}'$ has a proper absorbing subalgebra. The construction is possible since the relation of "being an absorbing subalgebra" is transitive (which can be proved using a composition of terms).

6.1 Defining the subinstance

The beginning of this section bears the closest resemblance to the proof of [1, 5, 3]. Recall that we assumed that none of the $\mathbb{A}_x$'s has a proper absorbing subuniverse.

Claim 6.2. Let p be a path-pattern from x to y and α a congruence on $\mathbb{A}_y$ such that the quotient $\mathbb{A}_y/\alpha$ is simple. Let

$$\mathbb{R} = \{(s(\text{start}_p), s(\text{end}_p)/\alpha) : s \text{ is a solution of } p\}.$$

Then $\mathbb{R} \leq_{\text{sub}} \mathbb{A}_x \times \mathbb{A}_y/\alpha$ and it is

1. either the full product,
2. or a graph of a function.

In the second case α' defined by

$$a\ \alpha'\ a' \text{ if and only if } (a, b), (a', b) \in \mathbb{R} \text{ for some } b$$

is a congruence on $\mathbb{A}_x$ and solutions of p establish an isomorphism from $\mathbb{A}_x/\alpha'$ to $\mathbb{A}_y/\alpha$.

Proof. Take p, α and $\mathbb{R}$ as in the statement of the claim. The fact that $\mathbb{R}$ is subdirect in $\mathbb{A}_x \times \mathbb{A}_y/\alpha$ follows from arc consistency of the instance. Consider the congruence $\pi_1 \vee \pi_2$ on $\mathbb{R}$. If it is the full congruence then, as neither $\mathbb{A}_x$ nor $\mathbb{A}_y/\alpha$ have proper absorbing subalgebras (if $\mathbb{A}_y/\alpha$ had one then $\mathbb{A}_y$ would have one as well), Theorem 6.1 implies that $\mathbb{R}$ is the full product.

In the second case $\pi_1 \vee \pi_2$ is not the full congruence on $\mathbb{R}$. Therefore the right congruence defined by $\mathbb{R}$ is non-full and, as the algebra is simple, it has to be the identity congruence. This implies that every $a \in \mathbb{A}_x$ has a unique b in $\mathbb{A}_y/\alpha$ such that $(a, b) \in \mathbb{R}$. Let α' be as in the statement then $\mathbb{R}' = \{(s(\text{start}_p)/\alpha', s(\text{end}_p)/\alpha) : s \text{ is a solution to } p\}$ is a graph of a bijection and it establishes an isomorphism between the quotients. ◻

In order to define $\mathcal{I}'$ we find a variable y such that $\mathbb{A}_y$ has more than one element and fix a congruence α_y so that $\mathbb{A}_y/\alpha_y$ is simple. Moreover we choose an arbitrary block of α_y and denote it by $\mathbb{A}'_y$. Let p be a path pattern in $\mathcal{I}$ which end at y. We say that a pattern is *non-proper* if, in Claim 6.2, it falls into the scope of case 1; a pattern is *proper* if it falls into the scope of case 2.

Claim 6.3. Let p, q be proper path patterns which start at the same variable x. The congruences and the isomorphisms established in Claim 6.2 by p and q are the same.

Proof. Let α_p (α_q) be the congruence given by Claim 6.2 for pattern p (pattern q respectively). If, for every $a \in \mathbb{A}_x$ we have $a \in a/\alpha_p + p - q$ the claim holds. Indeed we immediately conclude that $a + p$ and $a + q$ are in the same α_y block for any a and the classes of α_p and α_q as well as the isomorphisms defined by p and q must coincide.

If, on the other hand, there is an element a such that $a \notin a/\alpha_p + p - q$ then the last set is an α_q block not containing a. Therefore $a \notin \{a\} + j(p - q + q - p) + p - q$ for all j; indeed the pattern switches between α_p block containing a and a block of α_q not containing it. This contradicts the definition of *jpq* consistency. ◻

Call a variable x of $\mathcal{I}$ *proper* if there exists a proper path pattern p from x to y. Denote by α_x the congruence α' on $\mathbb{A}_x$ and by i_{xy} the isomorphism between $\mathbb{A}_x/\alpha_x$ and $\mathbb{A}_y/\alpha_y$ given by p according to by Claim 6.2. Let $\mathbb{A}'_x$ be a block of α_x such that $i_{xy}(\mathbb{A}'_x) = \mathbb{A}'_y$. By Claim 6.3 α' and i_{xy} do not depend on the choice of the pattern p. For every variable x which is not a proper variable put $\mathbb{A}'_x = \mathbb{A}_x$.

Let $\mathcal{I}'$ be a subinstance of $\mathcal{I}$ obtained by restricting every variable x of $\mathcal{I}$ to $\mathbb{A}'_x$ (by restricting all the constraining relations appropriately). The following technical claim is used to show the consistency of $\mathcal{I}'$.

Claim 6.4. Let x and z be proper variables and let p be a path-pattern from x to z;

1. either, for every $a \in \mathbb{A}_x$, $a/\alpha_x + p = \mathbb{A}_z$, and for every $a' \in \mathbb{A}_z$, $a'/\alpha_z - p = \mathbb{A}_x$ or
2. the pattern p establishes an isomorphism i from $\mathbb{A}_x/\alpha_x$ to $\mathbb{A}_z/\alpha_z$ such that the $i_{xy}(a/\alpha_x) = i_{zy}(i(a/\alpha_x))$ for any $a \in \mathbb{A}_x$, in particular $i(\mathbb{A}'_x) = \mathbb{A}'_z$.

Proof. Apply Claim 6.2 to p and α_z and to $-p$ and α_x; if in both of the applications we are in case 1 then we proved the first case of the current claim.

Note that proving the second case for p is equivalent to proving it for $-p$, and thus without loss of generality we assume that case 2 of Claim 6.2 holds for p and α_z. Let q be any proper pattern from z to y, and note that in this case $p + q$ is a proper pattern from x to y. This means that $p + q$ defines the congruence α_x and the isomorphism i_{xy} (as they are independent on the choice of a proper pattern). This means that p defines an isomorphism i and that $i_{xy}(a/\alpha_x) = i_{zy}(i(a/\alpha_x))$ for any $a \in \mathbb{A}_x$ as required. ◻

Note that in the case where all of the constraints are binary, the last claim establishes arc consistency for $\mathcal{I}'$. In the general case we need to proceed more carefully.

6.2 The subinstance is arc consistent

In order to establish arc consistency, and eventually *jpq* consistency, of the instance $\mathcal{I}'$ we require two facts. The first one is from [5].

Proposition 6.5 (Lemma 5 of [5])**.** *Let $\mathbb{A}_1, \dots, \mathbb{A}_k$ be simple SD(∧) algebras with no proper absorbing subuniverses. If $\mathbb{R} \leq_{sub} \prod_i \mathbb{A}_i$ is such that $\pi_i \vee \pi_j = 1_{\mathbb{R}}$ then $\mathbb{R} = \prod_i \mathbb{A}_i$.*

One particular consequence of the fact is that, under the assumptions of Proposition 6.5, the algebra $\prod_i \mathbb{A}_i$ has no proper absorbing subalgebras. Indeed such a subalgebra, say $\mathbb{R}'$ would need to be subdirect, as none of the $\mathbb{A}_i$'s has an absorbing subuniverse. Moreover $\pi_i \vee \pi_j = 1_{\mathbb{R}'}$ would hold by absorption and Proposition 6.5 implies that $\mathbb{R}' = \prod_i \mathbb{A}_i$. A proof of the following proposition is postponed until Section 6.4.

Proposition 6.6. *Let $\mathbb{A}$ be an SD(∧) algebra and α a congruence on $\mathbb{A}$ such that $\mathbb{A}/\alpha$ has no proper absorbing subuniverses. Then for every pair of congruences β and γ if $\alpha \vee \beta = \alpha \vee \gamma = 1_{\mathbb{A}}$ then $\alpha \vee (\beta \wedge \gamma) = 1_{\mathbb{A}}$.*

Let $(\mathbf{x}, \mathbb{R})$ be an arbitrary constraint of $\mathcal{I}$. Note that, by arc consistency of $\mathcal{I}$, we have $\mathbb{R} \leq_{\text{sub}} \prod_i \mathbb{A}_{\mathbf{x}(i)}$. We fix an arbitrary coordinate, without loss of generality coordinate 1, and will show that

$$\text{any } a \in \mathbb{A}_{\mathbf{x}(1)} \text{ extends to a tuple in } \mathbb{R} \cap \prod_i \mathbb{A}'_{\mathbf{x}(i)}.$$

Once this is proved, we've established arc consistency of $\mathcal{I}'$.

Let i and j be indices such that both $\mathbf{x}(i)$ and $\mathbf{x}(j)$ are proper. Define pattern p by taking a copy of the constraint $(\mathbf{x}, \mathbb{R})$ but making all the variables different, define hom_p in the natural way and set the variable at i-th position to be the start and at j-th to be the end of p. Apply the Claim 6.4 to p; if it results in case 2 we call i and j equivalent. Note that, in this case, every tuple $\mathbf{a} \in \mathbb{R}$ has $\mathbf{a}(i) \in \mathbb{A}'_{\mathbf{x}(i)}$ if and only if $\mathbf{a}(j) \in \mathbb{A}'_{\mathbf{x}(j)}$. Applying, in the same way, Claim 6.4 to any pair of coordinates with proper variables we define an equivalence on the coordinates of $\mathbb{R}$ (but only these which correspond to proper variables in $\mathbf{x}$).

Choose one index from each equivalence class of this equivalence (if $\mathbf{x}(1)$ is proper choose 1 from its equivalence class) and assume without loss of generality that these indices are $2, \dots, k$ (or

$1, \ldots, k$ if $\mathbf{x}(1)$ is proper). Let $\mathbb{S} = \{(a_1, a_2/\alpha_{\mathbf{x}(2)}, \ldots, a_k/\alpha_{\mathbf{x}(k)}) : (a_1, \ldots, a_k) \in \mathrm{proj}_{1\ldots k} \mathbb{R}\}$. The next step is to show that, in the algebra $\mathbb{S}$, we have $\pi_i \vee \pi_j = 1_{\mathbb{S}}$ for all $i \neq j$. If both $i, j > 1$ then they are from different equivalence classes, which means that the pattern defined as in the previous paragraph is in case 1 of Claim 6.4. As every solution of this pattern induces a tuple in $\mathbb{S}$ we are done. In order to confirm that $\pi_1 \vee \pi j = 1_{\mathbb{S}}$ we consider two cases; either $\mathbf{x}(1)$ is proper and the reasoning is the same as for the other i's, or it is non-proper and the failure of $\pi_1 \vee \pi_j = 1_{\mathbb{S}}$ would imply a proper pattern from $\mathbf{x}(1)$ via $\mathbf{x}(j)$. Note that the first consequence of these facts is that $\mathrm{proj}_{2,\ldots,k} \mathbb{S}$ is, by Proposition 6.5, the full product $\prod_{i=2}^{k} \mathbb{A}_{\mathbf{x}(i)}/\alpha_{\mathbf{x}(i)}$.

Multiple applications of Proposition 6.6 (each time with π_1 in place of α) show that $\pi_1 \vee \bigwedge_{i \leq 2} \pi_i = 1_{\mathbb{S}}$. This means that $\mathbb{S}$ treated as a product $\mathbb{A}_{\mathbf{x}(1)}$ and the $\prod_{i=2}^{k} \mathbb{A}_{\mathbf{x}(i)}/\alpha_{\mathbf{x}(i)}$ satisfies the assumption of Theorem 6.1. As neither $\mathbb{A}_{\mathbf{x}(1)}$ nor $\prod_{i=2}^{k} \mathbb{A}_{\mathbf{x}(i)}/\alpha_{\mathbf{x}(i)}$ have absorbing subuniverses (the last one by the discussion after Proposition 6.5) Theorem 6.1 implies that $\mathbb{S}$ is indeed the full product. Now the construction of $\mathbb{S}$ implies that every element in $\mathbb{A}'_{\mathbf{x}(1)}$ extends to a tuple in $\mathbb{R} \cap \prod_i \mathbb{A}'_{\mathbf{x}(i)}$ i.e. the instance $\mathcal{I}'$ is arc consistent.

6.3 The subinstance is *jpq* consistent

The following proposition is, in essence Theorem 6 of [8], can be used to establish *jpq* consistency.

Proposition 6.7. *Let $\mathbb{A}$ be an algebra and $\mathbb{R}, \mathbb{S} \leq_{sub} \mathbb{A}^n$ such that $\mathbb{R} \trianglelefteq \mathbb{S}$, and for every $a \in \mathbb{A}$ the constant tuple $(a, \ldots, a)$ belongs to $\mathbb{S}$. Then $\mathbb{R}$ contains at least one constant tuple.*

The proof of *jpq* consistency of the instance $\mathcal{I}'$ follows the same plan as the proof or arc-consistency, but instead with a single constraint we start with patterns. Let x be any variable and p', q' be patterns, from x to x, in instance $\mathcal{I}'$. Let p, q be counterparts of these patterns in $\mathcal{I}$ i.e. p and p' are almost the same but p' has restricted relations from $\mathcal{I}'$ while p has relations from $\mathcal{I}$. Define pattern $r = j(p+q)+p$ where j is given by Corollary 5.3. Let the set of variables of r, excluding start_r and end_r, be denoted by V. Let $\mathbb{R} \leq_{\mathrm{sub}} \mathbb{A}_x^2 \times \prod_{v \in V} \mathbb{A}_{\mathrm{hom}_r(v)}$ be the algebra of all the solutions of this pattern. The subdirectness of $\mathbb{R}$ follows by arc consistency of $\mathcal{I}$, and Corollary 5.3 implies that, for every $a \in \mathbb{A}_x$, we have $(a, a) \in \mathrm{proj}_{1,2} \mathbb{R}$.

Our goal is to prove that, for any $a \in \mathbb{A}'_y$, there is a tuple in $\mathbb{R} \cap (\mathbb{A}_y^2 \times \prod_{v \in V} \mathbb{A}'_{\mathrm{hom}_r(v)})$ which starts with (a, a). In order to obtain this we proceed almost exactly as in the previous section. Let v and w be variables of r such that both $\mathrm{hom}_r(v)$ and $\mathrm{hom}_r(w)$ are proper. Let r' be the path pattern, obtained by dropping constraints from r, such that $\mathrm{start}_{r'} = v$ and $\mathrm{end}_{r'} = w$. If, applying the Claim 6.4 to this pattern, we are in case 2 then every solution to r, say s, has $s(v) \in \mathbb{A}'_{\mathrm{hom}_r(v)}$ if and only if $s(v) \in \mathbb{A}'_{\mathrm{hom}_r(w)}$. This defines an equivalence on the set of variables v which have proper $\mathrm{hom}_r(v)$.

Let W be a set of variables such that every equivalence class, that does not contain start_r and end_r, has a single representative in W. Take a projection of $\mathbb{R}$ to $\{1, 2\} \cup W$ and let $\mathbb{S}$ be the algebra obtained from this projections by quoting every coordinate $v \in W$ by $\alpha_{\mathrm{hom}_r(v)}$.

Note that $\mathbb{S} \leq_{\mathrm{sub}} \mathbb{A}_x^2 \times \prod_{v \in W} \mathbb{A}_{\mathrm{hom}_r(v)}/\alpha_{\mathrm{hom}_r(v)}$ and that, using reasoning identical as in the previous section, we can show that $\pi_i \vee \pi_j = 1_{\mathbb{S}}$ as long as $i \neq j$ and $\{i, j\} \neq \{1, 2\}$. This implies, among other things, that $\mathrm{proj}_W(\mathbb{S})$ is the full product. Moreover we can apply Proposition 6.6 to show that $\pi_1 \vee \bigwedge_{i \in W} \pi_i = 1_{\mathbb{S}}$ and similarly for 2 in place of 1. As $\mathbb{S}/\bigwedge_{i \in W} \pi_i$ has no proper absorbing subalgebras we can apply Proposition 6.6 once more to get $(\pi_1 \wedge \pi_2) \vee \bigwedge_{i \in W} \pi_i = 1_{\mathbb{S}}$. Applying Theorem 6.1 to $\mathbb{S}$ ($\mathrm{proj}_{12} \mathbb{S}$ on the left and the remaining coordinates on the right) by the discussion after the theorem we get $\mathbb{R}$ such that $\mathbb{R} \times \prod_{v \in W} \mathbb{A}_{\mathrm{hom}_r(v)}/\alpha_{\mathrm{hom}_r(v)}$ is a subalgebra of $\mathbb{S}$.

Note that $\mathbb{R} \trianglelefteq \mathrm{proj}_{1,2} \mathbb{S}$ and, putting $\Delta = \{(a, a) : a \in \mathbb{A}_y\}$ we have $\Delta \leq \mathrm{proj}_{1,2} \mathbb{S}$ by construction. That means that we can apply Proposition 6.7 to obtain a constant pair in $\mathbb{R}$. But then

$\emptyset \neq (\mathbb{R} \cap \Delta) \trianglelefteq (\mathrm{proj}_{12}(\mathbb{S}) \cap \Delta) = \Delta$ and as $\mathbb{A}_y$ has no absorbing subuniverses we conclude that $\mathbb{R} \cap \Delta = \Delta$. This finishes a proof of the fact that the instance $\mathcal{I}'$ is *jpq* consistent.

6.4 A proof of Proposition 6.6

In this section we prove Proposition 6.6. In order to proceed with the proof we require the following theorem, which is a special case of the main result of [6].

Theorem 6.8. *Let $\mathbb{R} \leq_{sub} \mathbb{A}^2$ be an idempotent Taylor algebra. If $\pi_1 \vee \pi_2$ is the full congruence on $\mathbb{R}$ then $(a, a) \in \mathbb{A}$ for some a.*

and its easy consequence

Corollary 6.9. Let $\mathbb{A}, \mathbb{B}$ be idempotent Taylor algebras and both $\mathbb{R}, \mathbb{S}$ be subdirect in $\mathbb{A} \times \mathbb{B}$. If $\pi_1 \vee \pi_2$ is the full congruence in $\mathbb{R}$ then $\mathbb{R} \cap \mathbb{S} \neq \emptyset$.

Proof. Consider the algebra $\mathbb{T} = \{(a, c) : \exists b\ (a, b) \in \mathbb{R} \wedge (c, b) \in \mathbb{S}\}$. The algebra $\mathbb{T}$ is subdirect in $\mathbb{A}^2$ and $\pi_1 \vee \pi_2$ is the full congruence on $\mathbb{T}$: indeed if $(a, b), (a', b) \in \mathbb{R}$ then, choosing c such that $(c, b) \in \mathbb{S}$, we have $(a, c), (a', c) \in \mathbb{T}$. This implies that the left congruence defined by $\mathbb{T}$ is not smaller than the left congruence defined by $\mathbb{R}$ i.e. is full. Applying Theorem 6.8 to $\mathbb{T}$ we get a constant pair which implies that $\mathbb{R} \cap \mathbb{S} \neq \emptyset$. □

The proof hinges on the following lemma.

Lemma 6.10. *Let $\mathbb{R} \leq_{sub} \mathbb{A} \times \mathbb{B}$ be an idempotent SD($\wedge$) algebra and let α be a congruence on $\mathbb{R}$. There is at most one α class which is subdirect in $\mathbb{A} \times \mathbb{B}$.*

Proof. First note that the subset of $\mathbb{R}/\alpha$ consisting of α classes which are subdirect in $\mathbb{A} \times \mathbb{B}$ is a subalgebra of $\mathbb{R}/\alpha$. Indeed let $a \in \mathbb{A}$ and take $(\mathbb{R} \cap (\{a\} \times \mathbb{B}))/\alpha$ — this is a subalgebra of $\mathbb{R}/\alpha$ consisting of α-classes containing a pair with a as first element. Intersecting these subalgebras for all $a \in \mathbb{A}$ and for all $b \in \mathbb{B}$ we get a subalgebra of $\mathbb{R}/\alpha$ consisting of all α-classes subdirect in $\mathbb{A} \times \mathbb{B}$.

We substitute $\mathbb{R}$ with the union of the subdirect α-classes and restrict α to it. The assumptions still hold but now every α class is subdirect. Assume, for a contradiction, that $\mathbb{R}/\alpha$ has more than one element and let $\mathbb{A}'$ be a block of left congruence of $\mathbb{R}$ and $\mathbb{B}'$ be block of right congruence of $\mathbb{R}$ such that $\mathbb{R} \cap (\mathbb{A}' \times \mathbb{B}')$ is linked. The goal is to find a non-full subalgebra of $\mathbb{R}/\alpha$ (say $\mathbb{R}'/\alpha$) so that $\mathbb{R}$ and $\mathbb{R}'$ define the same left and right congruences. If we can accomplish it then $\mathbb{R}' \cap (\mathbb{A}' \times \mathbb{B}')$ is linked and subdirect in $\mathbb{A}' \times \mathbb{B}'$ and for $(a, b) \in \mathbb{R} \setminus \mathbb{R}'$ the class $(a, b)/\alpha \cap (\mathbb{A}' \times \mathbb{B}')$ is subdirect in $\mathbb{A}' \times \mathbb{B}'$. These subalgebras intersect empty which contradicts Corollary 6.9.

If $\mathbb{R}/\alpha$ has a non-trivial absorbing subalgebra, say given by $\mathbb{R}'/\alpha$, then left and right congruences defined by $\mathbb{R}$ and $\mathbb{R}'$ are the same: if $(a, b), (a', b') \in \mathbb{R}'$ and $(a, b') \in \mathbb{R}$ and the absorbing term is t then

$$a = t(a, \ldots, a), t(b', b, \ldots, b), t(a', a, \ldots, a), t(a', a', a, \ldots, a), \ldots, t(a', \ldots, a') = a'$$

is a sequence in $\mathbb{R}'$ witnessing that (a, b) and (a', b') are in $\pi_1 \vee \pi_2$ in $\mathbb{R}'$. The concludes the case of absorption in $\mathbb{R}/\alpha$.

If $\mathbb{R}/\alpha$ has no non-trivial absorbing subuniverses we fix β a maximal congruence above α. Let $(a, b) \in \mathbb{R}$, define an algebra $\mathbb{P} \leq \mathbb{B} \times \mathbb{A}$

$$(b', a') \in \mathbb{P} \text{ iff } (a, b'), (a', b'), (a', b) \in \mathbb{R},$$

and put $\mathbb{Q} = \{((a, b')/\beta, (a', b')/\beta, (a', b)/\beta) : (b', a') \in \mathbb{P}\}$. Note that the tuple constantly equal to $(a, b)/\beta$ is in $\mathbb{Q}$ (indeed $(b, a) \in \mathbb{P}$). Take any class of β; there exists $b' \in \mathbb{B}$ such that

(a, b') belongs to this class (respectively $a' \in \mathbb{A}$ such that (a', b) belongs to this class). Then $((a, b')/\beta, (a, b')/\beta, (a, b)/\beta) \in \mathbb{Q}$ (and respectively $((a, b)/\beta, (a', b)/\beta, (a', b)/\beta)$) which implies that $\mathbb{Q} \leq_{\text{sub}} (\mathbb{R}/\beta)^3$ and that $\pi_i \vee \pi_j$ is the full congruence on $\mathbb{Q}$ for any $1 \leq i \neq j \leq 3$. As $\mathbb{R}/\beta$ is simple and has no absorption Proposition 6.5 implies that $\mathbb{Q}$ is the full product. This means that in any β class there is a connection from a to b i.e. each β class defines left and right congruences identical to those given by $\mathbb{R}$. This is a contradiction in the case of no absorption and the lemma is proved. □

Proof of Proposition 6.6. Let $\mathbb{R}$ be an algebra of triples $\{(a/\alpha, a/\beta, a/\gamma) : a \in \mathbb{A}\}$ which is obviously subdirect in $\mathbb{A}/\alpha \times \mathbb{A}/\beta \times \mathbb{A}/\gamma$. From the assumptions it follows that $\pi_1 \vee \pi_2 = \pi_1 \vee \pi_3 = 1_{\mathbb{R}}$ and in order to confirm the proposition we need to show that $\pi_1 \vee (\pi_2 \wedge \pi_3) = 1_{\mathbb{R}}$.

Now let $\mathbb{R}'$ be minimal absorbing subuniverse of $\mathbb{R}$. Clearly $\pi_1 \vee \pi_2 = \pi_1 \vee \pi_3$ is still a full congruence in $\mathbb{R}'$; moreover $\text{proj}_1(\mathbb{R}') = \mathbb{A}/\alpha$ since $\mathbb{A}/\alpha$ has no absorbing subuniverses. Therefore if we show that $\pi_1 \vee (\pi_2 \wedge \pi_3)$ is the full congruence in $\mathbb{R}'$ the same would holds in $\mathbb{R}$ and the proposition will be proved.

Let us denote $\text{proj}_2(\mathbb{R}')$ by $\mathbb{A}_2$ and $\text{proj}_3(\mathbb{R}')$ by $\mathbb{A}_3$. Note that $\text{proj}_{12}(\mathbb{R}') = \mathbb{A}/\alpha \times \mathbb{A}_2$; indeed $\pi_1 \vee \pi_2$ is the full congruence and neither $\mathbb{A}/\alpha$ nor $\mathbb{A}_2$ have absorbing subuniverse (since $\mathbb{R}'$ doesn't) and thus Theorem 6.1 implies that $\text{proj}_{12}(\mathbb{R}')$ is full. Similarly, $\text{proj}_{13}(\mathbb{R}') = \mathbb{A}/\alpha \times \mathbb{A}_3$.

If $\pi_1 \vee (\pi_2 \wedge \pi_3)$ is not the full congruence on $\mathbb{R}'$ it defines a non-trivial right congruence on $\text{proj}_{23}(\mathbb{R}')$ (here $\mathbb{A}/\alpha$ is on the left and $\text{proj}_{23}(\mathbb{R}')$ is on the right). By the previous paragraph every congruence block of this right congruence is subdirect in $\mathbb{A}_2 \times \mathbb{A}_3$ — this directly contradicts Lemma 6.10 and therefore finishes the proof. □

7 Reduction in the presence of absorption

In this section we tackle the case in which at least one $\mathbb{A}_x$ has a non-trivial absorbing subuniverse. The first step is to construct an arc consistent subinstance of $\mathcal{I}$ which is absorbing i.e. if $(\mathbf{x}, \mathbb{R})$ is a constraint in $\mathcal{I}$ and $(\mathbf{x}, \mathbb{R}')$ is the corresponding constraint in the subinstance then $\mathbb{R}' \trianglelefteq \mathbb{R}$. Additionally we require the subinstance to be minimal arc consistent with sets $\{\mathbb{A}'_x\}$ such that no $\mathbb{A}'_x$ has a proper absorbing subuniverse. This is obtained by repeated applications of Proposition 7.1, which is proved in the next section. In section 7.2 we prove that such a subinstance is indeed *jpq* consistent.

7.1 Shrinking arc consistent subinstances of $\mathcal{I}$

This section is fully devoted to a proof of Proposition 7.1. Once the proposition is proved we can repeatedly apply it to $\mathcal{I}$ (choosing $\mathcal{J} = \mathcal{I}$ in the first step) each time taking the proper subinstance of $\mathcal{J}$ provided by the proposition as the new $\mathcal{J}$.

Proposition 7.1. *Let $\mathcal{I}$ be a jpq instance, and $\mathcal{J}$ be an absorbing subinstance of $\mathcal{I}$ which is arc consistent with sets $\{\mathbb{B}_x\}$. If one of $\mathbb{B}_x$'s has a proper absorbing subuniverse, then there exists a proper absorbing and arc consistent subinstance of $\mathcal{J}$.*

Let $\mathcal{I}, \mathcal{J}$ be as in the statement of the proposition. The vast majority of the proof disregards $\mathcal{I}$, so all the patterns and propagations in this section are (unless explicitly stated otherwise) in $\mathcal{J}$. We begin, as in [8], by defining a preorder on pairs $(x, \mathbb{B})$ satisfying $\mathbb{B} \leq \mathbb{B}_x$ and $\emptyset \neq \mathbb{B} \neq \mathbb{B}_x$. We put $(x, \mathbb{B}) \sqsubseteq (x', \mathbb{B}')$ if there is a tree pattern p satisfying $\text{hom}_p(\text{start}_p) = x$, $\text{hom}_p(\text{end}_p) = x'$ and $\mathbb{B} + p = \mathbb{B}'$. Note that the relation $\sqsubseteq$ is transitive by the addition of tree patterns.

Fix $\mathbb{B}'$ and x' such that $\mathbb{B}'$ is a non-trivial absorbing subuniverse of $\mathbb{B}_{x'}$ provided by the assumption of the proposition. Let $\mathcal{R}$ be set of elements in the preorder satisfying the following properties:

- for every $(y, \mathbb{B}) \in \mathcal{R}$ we have $(x', \mathbb{B}') \sqsubseteq (y, \mathbb{B})$;
- for every $(y, \mathbb{B}), (y', \mathbb{B}') \in \mathcal{R}$ we have $(y, \mathbb{B}) \sqsubseteq (y', \mathbb{B}')$ and
- if $(y, \mathbb{B}) \in \mathcal{R}$ and $(y, \mathbb{B}) \sqsubseteq (y', \mathbb{B}')$ then $(y', \mathbb{B}') \in \mathcal{R}$.

Such a set can be easily found by starting from $(x', \mathbb{B}')$ and following the relation "up" to a top equivalence class of $\sqsubseteq$. Note that, by arc consistency of instance $\mathcal{J}$, if $(y, \mathbb{B}) \in \mathcal{R}$ then $\mathbb{B} \trianglelefteq \mathbb{B}_y$. We call a variable of $\mathcal{I}$ *proper* if it appears in a pair in $\mathcal{R}$, and postpone a proof of the following claim until later.

Claim 7.2. For every proper x there is an algebra $\mathbb{B}$ such that:

- $(x, \mathbb{B}) \in \mathcal{R}$, and
- for every $\mathbb{B}'$ if $(x, \mathbb{B}') \in \mathcal{R}$ then $\mathbb{B} \leq \mathbb{B}'$.

For each proper variable x put $\mathbb{B}'_x$ to be the unique smallest algebra provided by the previous claim (denoted there by $\mathbb{B}$) and for remaining x put $\mathbb{B}'_x = \mathbb{B}_x$. Let $\mathcal{K}$ be the subinstance of $\mathcal{J}$ to obtained by restricting to $\{\mathbb{B}'_x\}$. The instance $\mathcal{K}$ is a proper subinstance of $\mathcal{J}$ by the choice of $\mathcal{R}$, and is absorbing since all the algebras $\mathbb{B}'_x$ are. In order to finish a proof of Proposition 7.1 we will prove that it is arc consistent with the sets $\{\mathbb{B}'_x\}$.

Take an arbitrary constraint $((x_1, \ldots, x_n), \mathbb{R})$; our goal is to show that $\mathbb{R} \cap \prod_{i=1}^n \mathbb{B}'_{x_i} \leq_{\text{sub}} \prod_{i=1}^n \mathbb{B}'_{x_i}$. For simplicity of presentation we will show only that $\text{proj}_n(\mathbb{R} \cap \prod_{i=1}^n \mathbb{B}'_{x_i}) = \mathbb{B}'_{x_n}$ (as the order of coordinates does not alter the reasoning) and assume that x_1 is proper (if none of the $x_1, \ldots, x_{n-1}$ is proper the fact follows from arc consistency of $\mathcal{J}$ and if one is we permute the constraint). Let k be the largest number such that,

$$\text{proj}_k\left(\text{proj}_{1,\ldots,k}(\mathbb{R}) \cap \prod_{i \in \{1,\ldots,k\}} \mathbb{B}'_{x_i}\right) = \mathbb{B}'_{x_k}.$$

Clearly $k > 1$ (as $\mathcal{I}$ is arc consistent) and if k is smaller than n we obtain a contradiction using the following, initially empty, tree pattern p:

1. take the constraint $((y_1, \ldots, y_n), \mathbb{R})$ (make all y_i's pairwise different, and note that x_i's don't need to be) add it to p and set
   - $\text{hom}_p((y_1, \ldots, y_n), \mathbb{R}) = ((x_1, \ldots, x_n), \mathbb{R})$,
   - $\text{hom}_p(y_i) = x_i$
   - $\text{start}_p = \{y_1\}$ and $\text{end}_p = x_{k+1}$
2. for $i = 2, \ldots, k$ consider x_i
   (a) if $x_i = x_1$ add y_i to start_p;
   (b) if x_i is proper and different than x_1:
   - take a fresh copy of tree pattern q witnessing $(x_1, \mathbb{B}'_{x_1}) \sqsubseteq (x_i, \mathbb{B}'_{x_i})$,
   - adjoin it to p identifying end_q with y_i (extend hom_p to new variables and constraints in the natural way),
   - add start_q to start_p.

(c) if x_i isn't proper do nothing;

Let $\mathbb{B}' = \mathbb{B}'_{x_1} + p$; the algebra $\mathbb{B}' \neq \mathbb{B}_{x_{k+1}}$ as it would contradict the maximality of k. Therefore $(x_1, \mathbb{B}'_{x_1}) \sqsubseteq (x_{k+1}, \mathbb{B}')$ but in this case, using Claim 7.2, we get $\mathbb{B}'_{x_{k+1}} \leq \mathbb{B}'$ which contradicts the maximality of k as well. This finishes, modulo a proof of Claim 7.2, the proof showing that the instance $\mathcal{I}'$ obtained by restricting to $\{\mathbb{B}'_x\}$ is arc consistent.

The remaining part of this subsection contains a proof of Claim 7.2. Suppose, for a contradiction, that the claim fails for a variable x and let $\mathcal{M}$ be the set of minimal under inclusion elements of $\{\mathbb{B} : (x, \mathbb{B}) \in \mathcal{R}\}$.

First we show that if $\mathbb{B} \in \mathcal{M}$ and $(x, \mathbb{C}) \in \mathcal{R}$ then either $\mathbb{B} \cap \mathbb{C} = \emptyset$ or $\mathbb{B} \leq \mathbb{C}$. Suppose otherwise and let $\mathbb{B}' = \mathbb{B} \cap \mathbb{C}$, we will show that $(x, \mathbb{B}') \in \mathcal{R}$ which contradicts the fact that $\mathbb{B} \in \mathcal{M}$. Indeed let patterns p, q be such that $\mathbb{B} + p = \mathbb{C}$ and $\mathbb{C} + q = \mathbb{B}$. Define a new pattern r by identifying the ends of p and $p + q$; we have $\mathbb{B} + r = (\mathbb{B} \cap \mathbb{C})$ showing that $(x, \mathbb{B}) \sqsubseteq (x, \mathbb{B} \cap \mathbb{C})$ i.e. $(x, \mathbb{B} \cap \mathbb{C}) \in \mathcal{R}$ which contradicts the minimality of $\mathbb{B}$.

Now fix $\mathbb{C}$ such that:

1. $(x, \mathbb{C}) \in \mathcal{R}$,

2. there exists $\mathbb{B} \in \mathcal{M}$ such that $\mathbb{B} \cap \mathbb{C} = \emptyset$

3. $\mathbb{C}$ is maximal, under inclusion, among the algebras satisfying the two previous conditions.

Let $\mathbb{B}_1, \ldots, \mathbb{B}_n$ be the elements of $\mathcal{M}$ which intersect empty with $\mathbb{C}$. Fix tree patterns p_i such that $\mathbb{B}_i + p_i = \mathbb{B}_{i+1}$ and patterns $q, q' : \mathbb{C} + q = \mathbb{B}_1, \mathbb{B}_n + q' = \mathbb{C}$.

Let pattern $p = q + p_1 + \cdots + p_n + q'$ and note that that $\mathbb{C} + p = \mathbb{C}$. Next we iteratively modify the pattern p: take an element $v \in \text{start}_p$ let p' be identical to p with the only exception that $\text{start}'_p = \text{start}_p \setminus \{v\}$. If $\mathbb{C} + p' = \mathbb{C}$ we substitute p for p' and repeat the procedure. Note that, as instance $\mathcal{J}$ is arc consistent, we will not remove all the elements from start_p.

Fix any $v \in \text{start}_p$ and let $\mathbb{E} \leq \mathbb{B}_x^2$ be the projection to (v, end_p) of all the solutions to p which send all the variable in $\text{start}_p \setminus \{v\}$ to $\mathbb{C}$. The following claim lists some basic properties of such an algebra $\mathbb{E}$.

Subclaim 1. The following hold:

1. $\text{proj}_2(\mathbb{E} \cap (\mathbb{C} \times \mathbb{B}_x)) = \mathbb{C}$;

2. every $\mathbb{B}_i$ is a subset of $\text{proj}_2(\mathbb{E})$;

3. for every i and every $a \in \mathbb{B}_i$ there exists $a' \in \bigcup_j \mathbb{B}_j$ such that $(a', a) \in \mathbb{E}$.

Proof. For the first item of the claim note that the algebra $\text{proj}_2(\mathbb{E} \cap (\mathbb{C} \times \mathbb{B}_x))$ is equal to the set $\mathbb{C} + p$ which is $\mathbb{C}$.

For the second let p' be identical to p with the only exception that $\text{start}_{p'} = \text{start}_p \setminus \{v\}$. The algebra $\text{proj}_2(\mathbb{E})$ is equal to $\mathbb{C} + p'$ and if it is $\mathbb{B}_x$ we are done. Otherwise $(x, \mathbb{C} + p') \in \mathcal{R}$ and the same follows from the maximality of $\mathbb{C}$ ($\mathbb{C} + p' \supsetneq \mathbb{C}$ as otherwise we would remove v from start_p while refining p).

For the last item we fix i and proceed independently on the choice of $a \in \mathbb{B}_i$. First we define p' by letting, initially, $p' = p$ and:

- removing v from $\text{start}_{p'}$ and setting $\text{end}_{p'} = v$;

- taking a fresh copy of q witnessing $(x, \mathbb{C}) \sqsubseteq (x, \mathbb{B}_i)$, adding it to p' by identifying end_q with end_p (which is no longer $\text{end}_{p'}$), adding start_q to $\text{start}_{p'}$ and adjusting $\text{hom}_{p'}$ accordingly.

Now $\mathbb{C} + p'$ is either $\mathbb{B}_x$ or $(x, \mathbb{C} + p') \in \mathcal{R}$ (it cannot be empty by the reasoning in the previous paragraph) and in either case $\mathbb{B}_j \subseteq \mathbb{C} + p'$ for some j. Let us define p'' starting, as usual, with $p'' = p$ and perform the following modifications:

- remove v from $\text{start}_{p''}$;
- take a fresh copy, say q, witnessing $(x, \mathbb{C}) \sqsubseteq (x, \mathbb{B}_j)$, add it to p'' by identifying end_q with v, add start_q to $\text{start}_{p''}$ and adjust $\text{hom}_{p''}$ accordingly.

We claim that $\mathbb{C} + p''$ is a superset of $\mathbb{B}_i$. Indeed $(\mathbb{C} + p'') \cap \mathbb{B}_i \neq \emptyset$ as $\mathbb{C} + p' = \mathbb{B}_j$ and therefore either $\mathbb{C} + p'' = \mathbb{B}_x$ or $(x, \mathbb{C} + p'') \in \mathcal{R}$. In the first case obviously $\mathbb{B}_i \leq \mathbb{C} + p''$ and in the second it follows from the minimality of $\mathbb{B}_i$. But this means that every $a \in \mathbb{B}_i$ has a solution of p'' mapping $\text{end}_{p''} \mapsto a$; this solution maps v to an element of $\mathbb{B}_j$ which can be chosen for a'. □

Now take pattern q to be identical to p but substitute the constraining relations from $\mathcal{J}$ with their counterparts in $\mathcal{I}$. More precisely let $(\mathbf{y}, \mathbb{R})$ be a constraint of p and $(\text{hom}_p(\mathbf{y}), \mathbb{R})$ a constraint of $\mathcal{J}$ if the corresponding constraint of $\mathcal{I}$ has constraining relation $\mathbb{S}$ we have $(\mathbf{y}, \mathbb{S})$ in q (instead of $(\mathbf{y}, \mathbb{R})$). Now let $\mathbb{F} \leq \mathbb{A}_x^2$ be similar to $\mathbb{E}$, i.e. defined by projections on (v, end_q) of the solutions to q, but this time we place no restrictions on evaluations of $\text{start}_q \setminus \{v\}$.

Subclaim 2. The following hold:

1. $\mathbb{F} \leq_{\text{sub}} \mathbb{A}_x^2$;
2. for every $a \in \mathbb{A}_x$ we have (a, a) in $\mathbb{F}$ composed with itself sufficiently many times;
3. for any i and any $b \in \mathbb{B}_i$ there exists $c \in \mathbb{C}$ such that (c, b) is in $\mathbb{F}$ composed with itself sufficiently many times.

Proof. Item 1. holds by the arc consistency of $\mathcal{I}$. For item 2. let r be the path pattern in q which connects v to end_q. We have $a \in \{a\} + j(r + r) + r$ for some j by the definition of jpq consistency. Let $a = a_0$ and choose $a_i \in a_{i-1} + r$ so that $a_{2j+1} = a$. Arc consistency of $\mathcal{I}$ implies that the solutions to r can be extended to solutions of q and thus $(a_i, a_{i+1}) \in \mathbb{F}$ for every i and item 2. is proved.

For item 3. fix i and $b \in \mathbb{B}_i$. Let r be as before but we split it into two parts: r' from v to the end_{p_i} (i.e. the variable which, while constructing p, was end_{p_i} on the path from v to end_p) and r'' from the end_{p_i} to end_q. Let j be such that $b \in \{b\} + j(r'' + r') + r''$ (by jpq consistency) and $c \in \mathbb{C}$ such that $b \in \{c\} + r'$ (such a c exists by the definition of p). Let $c = b_0$ and choose $b_i \in b_{i-1} + r' + r''$ such that $b_{j+1} = b$; now, using arc consistency of $\mathcal{I}$, we finish the proof using the reasoning identical to the one for item 2. □

First we argue that $\mathbb{E} \trianglelefteq \mathbb{F}$. As all the absorptions can be witnessed by a single term, the solutions to p absorb solutions to q. Moreover $\mathbb{C} \trianglelefteq \mathbb{B}_x \trianglelefteq \mathbb{A}_x$ and therefore the solutions to p with elements of $\text{start}_p \setminus \{v\}$ in $\mathbb{C}$ also absorb all the solutions of q. This implies that $\mathbb{E} \trianglelefteq \mathbb{F}$. In order to finish the proof we introduce a more shorthand notation: we write $\mathbb{E}^{(k)}$ for $\mathbb{E} \circ \cdots \circ \mathbb{E}$ (where there is k copies of $\mathbb{E}$ composed) and similarly for $\mathbb{F}$.

Take an $a \in \bigcup_i \mathbb{B}_i$ and, using Subclaim 1 item 3, find $a' \in \bigcup_i \mathbb{B}_i$ such that $(a', a) \in \mathbb{E}$. By repeating this procedure we get $a \in \bigcup_i \mathbb{B}_i$ such that (a, a) is included in $\mathbb{E}^{(k)}$ for some k. By Subclaim 2 item 3 we get $a' \in \mathbb{C}$ such that $(a', a) \in \mathbb{F}^{(l)}$ for some l and using e.g. Subclaim 1 item 1. we find $a'' \in \mathbb{C}$ such that $(a'', a) \in \mathbb{F}^{(kl)}$.

Now we start with $a_0 = a''$ and use Subclaim 1 item 1. to find $a_1 \in \mathbb{C}$ such that $(a_1, a_0) \in \mathbb{E}$. We repeat this procedure until some element repeats itself and then find a''' and m such that $(a''', a''') \in \mathbb{E}^{(m)}$ and $(a''', a'') \in \mathbb{F}^{(kl(m-1))}$.

Finally we got $(a,a), (a''', a''') \in \mathbb{E}^{(klm)}$ and $(a''', a) \in \mathbb{F}^{(klm)}$. Let t be the term witnessing the absorption $\mathbb{E} \trianglelefteq \mathbb{F}$ and thus the absorption $\mathbb{E}^{(klm)} \trianglelefteq \mathbb{F}^{(klm)}$ as well. Consider the sequence

$$a''' = t(a''', \dots, a'''), t(a, a''', \dots, a''), t(a, a, a''', \dots, a'''), \dots, t(a, \dots, a, a'''), t(a, \dots, a) = a.$$

Each pair of consecutive elements belongs to $\mathbb{E}^{(klm)}$ and thus we have (a''', a) in a sufficiently large composition of $\mathbb{E}$ with itself. But as $a''' \in \mathbb{C}$ and $a \notin \mathbb{C}$ this contradicts Subclaim 1 item 1.

7.2 There is a *jpq* instance inside $\mathcal{I}$

To finish the proof we start with $\mathcal{I}$ and, repeatedly, apply Proposition 7.1. The first application is with $\mathcal{J} = \mathcal{I}$ and the proper absorbing subuniverse is provided by the case we are working in. In all the following applications $\mathcal{I}$ is the same and $\mathcal{J}$ is the instance produced by Proposition 7.1 in the previous step.

Let $\mathcal{I}'$ be the subinstance of $\mathcal{I}$ on which we cannot produce any further reductions; say the instance is arc consistent with sets $\{\mathbb{A}'_x\}$ and no $\mathbb{A}'_x$ has a proper absorbing subuniverse. The instance $\mathcal{I}'$ is arc consistent and absorbing, it remains to prove that it is a *jpq* instance. We proceed identically as in the no absorption case. Let x be any variable and p', q' be patterns, from x to x, in instance $\mathcal{I}'$. Let p, q be counterparts of these patterns in $\mathcal{I}$ (obtained in the same way q was obtained from p in the previous section). Define pattern $r = j(p+q) + p$ where j is given by Corollary 5.3.

Let $\mathbb{R} \leq_{\text{sub}} \mathbb{A}_x^2$ be the algebra obtained by taking all the solutions of this pattern and projecting them to $(\text{start}_r, \text{end}_r)$. By the choice of j, for every $a \in \mathbb{A}_x$, we have $(a,a) \in \mathbb{R}$. Let $r' = j(p' + q') + p'$ and let $\mathbb{R}'$ be defined from r' in the same way $\mathbb{R}$ was obtained from R. By arc consistency of $\mathcal{I}'$ we have $\mathbb{R}' \leq_{\text{sub}} (A'_x)^2$.

Now restrict $\mathbb{R}$ to $\mathbb{A}'_x$ and apply Proposition 6.7 to such a restriction and $\mathbb{R}'$. The proposition provides a constant pair in $\mathbb{R}'$. Since $\mathbb{R}' \trianglelefteq \mathbb{R}$ the algebra consisting of constant pairs in $\mathbb{R}'$ absorbs the algebra consisting of constant pairs in $\mathbb{R}$ (which is the algebra of all constant pairs in $\mathbb{A}_x$). Therefore if for some $a \in \mathbb{A}'_x$ we have $(a,a) \notin \mathbb{R}'$ we would obtain a proper absorbing subuniverse of $\mathbb{A}'_x$ — this contradicts our assumptions on $\mathcal{I}'$. Thus $\mathbb{R}'$ has all the constants from $\mathbb{A}'_x$ and the instance $\mathcal{I}'$ is *jpq* consistent which finishes the proof.

8 Acknowledgments

I would like to thank Libor Barto and Jakub Bulin for many fruitful discussions and simplifications of some of the proofs in the paper. I would also like to thank the anonymous reviewers of this paper and an earlier LICS version for pointing out some inconsistencies in the presentation and other comments on improving readability of the paper.

References

[1] L. Barto and M. Kozik. Constraint Satisfaction Problems of bounded width. In *Foundations of Computer Science, 2009. FOCS '09. 50th Annual IEEE Symposium on*, pages 595–603, Oct 2009.

[2] L. Barto and M. Kozik. New conditions for Taylor varieties and CSP. In *Logic in Computer Science (LICS), 2010 25th Annual IEEE Symposium on*, pages 100–109, July 2010.

[3] Libor Barto. The collapse of the bounded width hierarchy. *Journal of Logic and Computation*, 2014.

[4] Libor Barto and Marcin Kozik. Robust satisfiability of Constraint Satisfaction Problems. In *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing*, STOC '12, pages 931–940, New York, NY, USA, 2012. ACM.

[5] Libor Barto and Marcin Kozik. Constraint Satisfaction Problems solvable by local consistency methods. *J. ACM*, 61(1):3:1–3:19, January 2014.

[6] Libor Barto, Marcin Kozik, and Todd Niven. Graphs, polymorphisms and the complexity of homomorphism problems. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, pages 789–796, New York, NY, USA, 2008. ACM.

[7] Libor Barto, Marcin Kozik, and David Stanovský. Mal'tsev conditions, lack of absorption, and solvability. *Algebra universalis*, 74(1):185–206, Sep 2015.

[8] Libor Barto, Marcin Kozik, and Ross Willard. Near unanimity constraints have bounded pathwidth duality. In *Proceedings of the 2012 27th Annual IEEE/ACM Symposium on Logic in Computer Science*, LICS '12, pages 125–134, Washington, DC, USA, 2012. IEEE Computer Society.

[9] Libor Barto, Andrei Krokhin, and Ross Willard. Polymorphisms, and How to Use Them. In Andrei Krokhin and Stanislav Zivny, editors, *The Constraint Satisfaction Problem: Complexity and Approximability*, volume 7 of *Dagstuhl Follow-Ups*, pages 1–44. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2017.

[10] Christian Bessiere and Romuald Debruyne. Theoretical analysis of singleton arc consistency and its extensions. *Artificial Intelligence*, 172(1):29 – 41, 2008.

[11] Manuel Bodirsky and Hubie Chen. Peek arc consistency. *Theoretical Computer Science*, 411(2):445 – 453, 2010.

[12] Joshua Brakensiek and Venkatesan Guruswami. Promise constraint satisfaction: Structure theory and a symmetric boolean dichotomy. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '18, pages 1782–1801, Philadelphia, PA, USA, 2018. Society for Industrial and Applied Mathematics.

[13] Andrei Bulatov. Bounded relational width. 2009. manuscript.

[14] Andrei A. Bulatov. A dichotomy theorem for nonuniform csps. In Umans [30], pages 319–330.

[15] Andrei A. Bulatov, Andrei Krokhin, and Benoit Larose. *Dualities for Constraint Satisfaction Problems*, pages 93–124. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[16] S. Burris and H.P. Sankappanavar. *A course in universal algebra*. Graduate texts in mathematics. Springer-Verlag, 1981.

[17] Hubie Chen, Victor Dalmau, and Berit Grußien. Arc consistency and friends. *Journal of Logic and Computation*, 23(1):87–108, 2013.

[18] Victor Dalmau and Andrei Krokhin. Majority constraints have bounded pathwidth duality. *European Journal of Combinatorics*, 29(4):821 – 837, 2008. Homomorphisms: Structure and Highlights Homomorphisms: Structure and Highlights.

[19] Victor Dalmau and Andrei Krokhin. Robust satisfiability for CSPs: Hardness and algorithmic results. *ACM Trans. Comput. Theory*, 5(4):15:1–15:25, November 2013.

[20] Victor Dalmau and Justin Pearson. Closure functions and width 1 problems. In Joxan Jaffar, editor, *Principles and Practice of Constraint Programming - CP'99*, volume 1713 of *Lecture Notes in Computer Science*, pages 159–173. Springer Berlin Heidelberg, 1999.

[21] VÃŋctor Dalmau, Marcin Kozik, Andrei Krokhin, Konstantin Makarychev, Yury Makarychev, and Jakub OprÅąal. *Robust algorithms with polynomial loss for near-unanimity CSPs*, pages 340–357.

[22] Romuald Debruyne and Christian Bessiere. Some practicable filtering techniques for the Constraint Satisfaction Problem. In *In Proceedings of IJCAI'97*, pages 412–417, 1997.

[23] Tomás Feder and Moshe Y. Vardi. 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, 1998.

[24] Ralph Freese and Ralph McKenzie. Maltsev families of varieties closed under join or maltsev product. *Algebra universalis*, 77(1):29–50, Feb 2017.

[25] Venkatesan Guruswami and Yuan Zhou. Tight bounds on the approximability of almost-satisfiable Horn SAT and exact hitting set. In Dana Randall, editor, *SODA*, pages 1574–1589. SIAM, 2011.

[26] Jelena Jovanović, Petar Marković, Ralph McKenzie, and Matthew Moore. Optimal strong mal'cev conditions for congruence meet-semidistributivity in locally finite varieties. *Algebra universalis*, 76(3):305–325, Nov 2016.

[27] Marcin Kozik. Weak consistency notions for all the csps of bounded width. In Martin Grohe, Eric Koskinen, and Natarajan Shankar, editors, *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5-8, 2016*, pages 633–641. ACM, 2016.

[28] Benoit Larose and László Zádori. Bounded width problems and algebras. *Algebra universalis*, 56(3-4):439–466, 2007.

[29] Thomas J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the Tenth Annual ACM Symposium on Theory of Computing*, STOC '78, pages 216–226, New York, NY, USA, 1978. ACM.

[30] Chris Umans, editor. *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*. IEEE Computer Society, 2017.

[31] Dmitriy Zhuk. A proof of CSP dichotomy conjecture. In Umans [30], pages 331–342.