Proof complexity of CSP

Azza Gaysin[*]

Department of Algebra, Faculty of Mathematics and Physics, Charles University in Prague

Abstract

The CSP (constraint satisfaction problems) is a class of problems deciding whether there exists a homomorphism from an instance relational structure to a target one. The CSP dichotomy is a profound result recently proved by Zhuk [19] and Bulatov [7]. It establishes that for any fixed target structure, CSP is either NP-complete or p-time solvable. Zhuk's algorithm solves CSP in polynomial time for constraint languages having a weak near-unanimity polymorphism.

For negative instances of p-time CSPs, it is reasonable to explore their proof complexity. We show that the soundness of Zhuk's algorithm can be proved in a theory of bounded arithmetic, namely in the theory V^1 augmented by three special universal algebra axioms. This implies that any propositional proof system that simulates both Extended Resolution and a theory that proves the three axioms admits p-size proofs of all negative instances of a fixed p-time CSP.

1 Introduction

An important class of NP problems are the constraint satisfaction problems (CSP). We will give its definition in Subsection 2.2, but a universal formulation is as follows: in a constraint satisfaction problem $\mathrm{CSP}(\mathcal{A})$ associated with a relational structure $\mathcal{A}$, for any relational structure over the same vocabulary $\mathcal{X}$ the question is whether $\mathcal{X}$ can be homomorphically mapped into $\mathcal{A}$. The problem $\mathcal{X} \mapsto_? \mathcal{A}$ is an instance of $\mathrm{CSP}(\mathcal{A})$. A celebrated theorem of Zhuk [19] and Bulatov [7] states that for each constraint language $\mathcal{A}$, $\mathrm{CSP}(\mathcal{A})$ is either NP-complete or p-time decidable (see [3],[11] for the history of this theorem and earlier results and conjectures).

The statement that there is no homomorphism from $\mathcal{X}$ into $\mathcal{A}$ can be encoded by a propositional tautology having atoms for the potential edges of a homomorphism. The size of this tautology, to be denoted $\neg HOM(\mathcal{X}, \mathcal{A})$, is polynomial in the sizes of $\mathcal{X}$ and $\mathcal{A}$. When $\mathrm{CSP}(\mathcal{A})$ is NP-complete we cannot hope to have short propositional proofs (in any proof system) of formulas $\neg HOM(\mathcal{X}, \mathcal{A})$ for all unsatisfiable instances $\mathcal{X}$ of $\mathrm{CSP}(\mathcal{A})$, as that would imply that NP is closed under complementation. However, when $\mathrm{CSP}(\mathcal{A})$ is p-time decidable this obstacle is removed.

Zhuk's algorithm solves polynomial time CSPs and provides a tool for the investigation of their proof complexity. In fact, for a satisfiable instance $\mathcal{X}$ of $\mathrm{CSP}(\mathcal{A})$ the algorithm produces a homomorphism from $\mathcal{X}$ to $\mathcal{A}$ as a witness of an affirmative answer. For unsatisfiable instances, on the contrary, one has no witness to the algorithm's correctness

---

other than its run. Our main result is that the soundness of Zhuk's algorithm can be proved in a theory of bounded arithmetic, namely in the theory V^1 augmented with three universal algebra axioms. By the soundness here we mean that all negative answers of the algorithm are correct. Every theory of bounded arithmetic corresponds to some propositional proof system in the sense that if one proves a universal statement in the theory, the propositional translations of this statement will have polynomial proofs in the proof system. Short propositional proofs of the statement $\neg HOM(\mathcal{X}, \mathcal{A})$ can be considered as witnesses for negative instances of CSP($\mathcal{A}$).

To establish the result we uses a modified framework analogous to the framework we explore for our previous result in [14]; there we considered a simple example of relational structures that are undirected graphs (the Hell-Nešetřil dichotomy theorem). Atserias and Ochremiak in [1] studied the relation between universal algebra (and CSP in particular) and proof complexity.

The paper is organized as follows. In Section 2 we recall the necessary background from universal algebra, CSP theory, proof complexity, and bounded arithmetic. In Section 3 we define strong subuniverses and linear algebras, and formulate Zhuk's four cases theorem representing one of the main ideas of the whole algorithm. The outline of Zhuk's algorithm is presented in Section 4. Section 5 is devoted to the soundness of Zhuk's algorithm and is divided into three principal parts. In Sections 5.1 - 5.3 we introduce the framework, formalize most of the notions used in the algorithm, and define a new theory of bounded arithmetic. In Section 5.4 we prove the soundness of consistency reductions in the theory V^1. Finally, in Section 5.5 we consider the linear case of the algorithm. The main theorem is formulated in Section 5.6 and the summary of the proof is presented there.

2 Preliminaries

2.1 Basic notions from universal algebra

This subsection is based on papers [2], [3]. Some definitions and results are adopted from [8].

For our purpose, we will consider only finite objects. For any non-empty domain A and any natural number n we call a mapping $f : A^n \mapsto A$ an n-ary *operation* on A. An *algebra* $\mathbb{A} = (A, f_1, f_2, ...)$ is a pair of a domain A and basic operations $f_1, f_2, ...$ of fixed arities on A from some signature $\Sigma = \{f_1, f_2, ...\}$. A *constraint language* Γ is a set of relations on finite domains. A *relational structure* $\mathcal{A} = (A, R_1, R_2, ...)$ is a pair of a domain A and relations $R_1, R_2, ...$ on A from some constraint language $\Gamma = \{R_1, R_2, ...\}$.

We say that an m-ary operation $f : A^m \to A$ *preservers* an n-ary relation $R \in A^n$ (or f is the *polymorphism* of R, or f is *compatible* with R, or R is *invariant* under f) if $f(\bar{a_1}, ..., \bar{a_m}) \in R$ for all choices of $\bar{a_1}, ..., \bar{a_m} \in R$. For any constraint language Γ and any set of operations O we will denote by $Pol(\Gamma)$ the set of all operations on A preserving each relation from Γ, and by $Inv(O)$ the set of all relations on A invariant under each operation from O.

A *term* in a signature Σ is a formal expression that uses *variables and composition of symbols* from Σ. The set of all term operations of *algebra* $\mathbb{A} = (A, F)$ is called the *clone of term operations* of $\mathbb{A}$, denoted by $Clone(\mathbb{A})$. A well-known theorem from universal algebra establishes the connection between algebras and relational structures.

Theorem 1 ([4]). *For any algebra $\mathbb{A}$ there exists relation structure $\mathcal{A}$ such that $Clone(\mathbb{A}) = Pol(\mathcal{A})$.*

In general, any set of operations O on A is a clone if it contains all projections and is closed under superposition, i.e. for a k-ary operation $f \in O$ and m-ary operations $g_1, ..., g_m \in O$ the superposition $f[g_1, ..., g_k]$ is in O as well. We define $Clone(O)$ to be the smallest clone containing O. The dual object for relations is the so-called relational clone – a set of relations Γ containing the binary equality relation and closed under primitive positive definitions (relations defined by relations from Γ, conjunction, and existential quantifiers). If we define $RelClone(\Gamma)$ to be the smallest relational clone containing Γ, then the following theorem expresses a one-to-one correspondence between relational clones and clones.

Theorem 2 (Galois correspondence for constraint languages).

1. For any finite domain A, any constraint language Γ on A, $Inv(Pol(\Gamma)) = RelClone(\Gamma)$.

2. For any finite domain A, any set of operations O on A, $Pol(Inv(O)) = Clone(O)$.

For an algebra $\mathbb{A}$ a subset $B \subseteq A$ is a *subuniverse* if it is closed under all operations of $\mathbb{A}$. Given a subuniverse B we can form the subalgebra $\mathbb{B} \leq \mathbb{A}$ by restriction of all the operations of $\mathbb{A}$ to the set B. Given an algebra $\mathbb{A}$ for every subset $X \subseteq A$ we denote by $Sg(X)$ the minimal subalgebra of A containing X, i.e. the subalgebra generated by X. If we define a closure operator $E(X)$ to be $E(X) = X \cup \{f(a_1, ..., a_n) : f$ is a basic operation on $A, a_1, ..., a_n \in X\}$, and $E^t(X)$ for $t \geq 0$ by $E^0(X) = X, E^{t+1}(X) = E(E^t(X))$, then

$$Sg(X) = X \cup E(X) \cup E^2(X) \cup ...$$

An equivalence relation σ on $\mathbb{A}$ is a *congruence* if any term operation on $\mathbb{A}$ is compatible with σ. Two trivial congruences on $\mathbb{A}$ are the diagonal relation $\Delta_A = \{(a, a) : a \in A\}$ and full relation $\nabla_A = A^2$. A congruence is a *maximal congruence* if it is not contained in any other congruence except ∇_A. A congruence σ allows one to introduce a *quotient, or factor, algebra* $\mathbb{A}/\sigma$. It has as the universe the set of σ-classes and the operations are defined using arbitrary representatives from these classes. Note that the congruence σ forms a subalgebra of $\mathbb{A}^2$: applying any term operation to elements from σ coordinatewise, due to the compatibility property, we again get an element from σ. In general, any n-ary relation R on $\mathbb{A}$ invariant under all term operations is a subalgebra of $\mathbb{A}^n$.

A nonempty class K of algebras of the same type (same signature) is called a *variety* if it is closed under subalgebras $S(K)$, homomorphic images $H(K)$, and direct products $P(K)$. It is known that the smallest variety containing K is equal to $HSP(K)$. For a pair of terms s, t over a signature Σ, we say that a class of algebras K in the signature Σ satisfies the identity $s \approx t$ if every algebra in the class does. For any set of identities Ξ of the type Σ, define $M(\Xi)$ to be the class of algebras K satisfying Ξ. A class K of algebras is an equational class if there is a set of identities Ξ such that $K = M(\Xi)$. In this case, we say that K is defined, or axiomatized, by Ξ.

Theorem 3 (Birkhoff). *K is an equational class if and only if K is a variety. In other words, classes of algebras defined by identities are precisely those that are closed under $H, S,$ and P.*

2.2 CSP basics

In this section, we will give two different definitions of the Constraint satisfaction problem (CSP) and will formulate the CSP dichotomy conjecture. Some definitions, examples, and results are adapted from [3], [19], and [21].

Definition 1 (CSP over finite domains [19])**.** The *Constraint satisfaction problem* is a problem of deciding whether there is an assignment to a set of variables that satisfies some specified constraints. An *instance of CSP problem* over finite domains is defined as a triple $\Theta = (X, D, C)$, where

- $X = \{x_0, ..., x_{n-1}\}$ is a finite set of variables,

- $D = \{D_0, ..., D_{n-1}\}$ is a set of non-empty finite domains,

- $C = \{C_0, ..., C_{t-1}\}$ is a set of constraints,

where each variable x_i can take on values in the non-empty domain D_i, and every constraint $C_j \in C$ is a pair $(\vec{x}_j, \rho_j)$ with $\vec{x}_j$ being a tuple of variables of some length m_j, called a *constraint scope*, and ρ_j being an m_j-ary relation on the product of the corresponding domains, called a *constraint relation*. The question is whether there exists a solution to Θ, i.e. an assignment to every variable x_i such that for each constraint C_j the image of the constraint scope is a member of the constraint relation.

A *constraint satisfaction problem* associated with constraint language Γ, to be denoted CSP(Γ), is a subclass of CSP defined by the property that any constraint relation in any instance of CSP(Γ) must belong to Γ.

The equivalent definition of CSP can be formulated in terms of homomorphisms between relational structures.

Definition 2 (CSP [6])**.**

- A *vocabulary* is a finite set of relational symbols R_1,..., R_n, each of which has a fixed arity.

- A *relational structure* over the vocabulary R_1,..., R_n is a tuple $\mathcal{A} = (A, R_1^{\mathcal{A}}, ..., R_n^{\mathcal{A}})$ such that A is a non-empty set, called the *universe* of $\mathcal{A}$, and each $R_i^{\mathcal{A}}$ is a relation on A having the same arity as the symbol R_i.

- For $\mathcal{X}$, $\mathcal{A}$, being relational structures over the same vocabulary R_1,..., R_n, a *homomorphism* from $\mathcal{X}$ to $\mathcal{A}$ is a mapping $\phi : \mathcal{X} \to \mathcal{A}$ from the universe X to the universe A such that for every m-ary relation $R^{\mathcal{X}}$ and every tuple $(x_1, ..., x_m) \in R^{\mathcal{X}}$ we have $(\phi(x_1), ..., \phi(x_m)) \in R^{\mathcal{A}}$.

Let $\mathcal{A}$ be a relational structure over a vocabulary R_1,..., R_n. In the *constraint satisfaction problem* associated with $\mathcal{A}$, denoted by CSP($\mathcal{A}$), the question is, given a structure $\mathcal{X}$ over the same vocabulary, whether there exists a homomorphism from $\mathcal{X}$ to $\mathcal{A}$. If the answer is positive, then we call the instance $\mathcal{X}$ *satisfiable* and *unsatisfiable* otherwise. We call $\mathcal{A}$ the *target structure* and $\mathcal{X}$ the *instance (or input) one*.

The idea of translation from the homomorphism form to the constraint form is the following: consider the domain X of the structure $\mathcal{X}$ as a set of variables and every tuple $(x_1, ..., x_m) \in R^{\mathcal{X}}$ as a constraint $C = (x_1, ..., x_m; R^{\mathcal{A}})$. For the translation back, consider the set of variables X as a domain of the instance structure, the set A as a domain of the target structure, and each constraint $C = (x_1, ..., x_m; R^{\mathcal{A}})$ as a relation $R^{\mathcal{X}}$ on X.

It was conjectured years ago by Feder and Vardi [11] and recently proved by Zhuk [19] and Bulatov [7] that there is a dichotomy: each CSP($\mathcal{A}$) is either NP-complete or polynomial time solvable. The dichotomy depends on the following. We call an operation Ω on a set A the *weak-near unanimity operation* (WNU) if it satisfies $\Omega(y, x, x, ..., x) = \Omega(x, y, x, ..., x) = ... = \Omega(x, x, ..., x, y)$ for all $x, y \in A$. Furthermore, Ω is called *idempotent* if $\Omega(x, ..., x) = x$ for every $x \in A$, and is called *special* if for all $x, y \in A$, $\Omega(x, ..., x, \Omega(x, ..., x, y)) = \Omega(x, ..., x, y)$.

Lemma 1 ([17]). *For any idempotent WNU operation Ω on a finite set, there exists a special WNU operation $\Omega' \in Clone(\Omega)$.*

Theorem 4 (CSP dichotomy theorem [19]). *Suppose Γ is a finite set of relations on a set A. Then CSP(Γ) can be solved in polynomial time if there exists a WNU operation Ω on A preserving Γ; CSP(Γ) is NP-complete otherwise.*

In terms of complexity, instead of Γ it is more convenient to consider richer languages since they considerably reduce the variety of languages to be studied. For example, if we consider the language $RelClone(\Gamma)$ that contains the binary equality relation and is closed under *pp*-definitions over Γ, we do not increase the complexity of the problem since CSP($RelClone(\Gamma)$) is log-space reducible to CSP(Γ). Note that due to Theorem 2 all relations *pp*-definable over Γ are invariant under all polymorphisms preserving Γ.

Apart from *pp*-definability, there are other modifications of constraint languages that do not increase their complexity (i.e. allow log-space reduction) such as *pp*-interpretability, homomorphic equivalence, and singleton expansion of a core constraint language, see [3]. The beauty of the so-called algebraic approach to CSP is that these modifications to constraint languages represent classical algebraic constructions. Indeed, homomorphic equivalence and singleton expansion put together ensure that the algebra corresponding to the constraint language is idempotent. *Pp*-interpretations correspond to taking homomorphic images, subalgebras, and products over the algebras of polymorphisms of the constraint languages, where an algebra of polymorphisms is $Pol(\Gamma)$ with elements being polymorphisms and the operation being a superposition.

It turns out that a constraint language $\mathcal{D}$ *pp*-interpreters a constraint language $\mathcal{E}$ if and only if in $Pol(\mathcal{E})$ there exist operations satisfying all the identities that are satisfied by operations in $Pol(\mathcal{D})$ [2]. Since a variety of algebras is defined by its identities, the variety of algebra corresponding to the language $\mathcal{D}$ contains the variety of algebra corresponding to the language $\mathcal{E}$. Thus, *pp*-interpretability does not change the structure or the properties of the corresponding algebras.

Pp-constructibility combines all previous modifications.

Definition 3 (*Pp*-constructibility [3]). A constraint language $\mathcal{D}$ over a domain D *pp*-constructs a constraint language $\mathcal{E}$ over a domain E if there is a sequence of constraint languages $\mathcal{D} = \mathcal{C}_1, ..., \mathcal{C}_k = \mathcal{E}$ such that for each $1 \leq i \leq k$

- $\mathcal{C}_i$ *pp*-interprets $\mathcal{C}_{i+1}$, or

- $\mathcal{C}_i$ is homomorphically equivalent to $\mathcal{C}_{i+1}$, or

- $\mathcal{C}_i$ is a core and $\mathcal{C}_{i+1}$ is its singleton expansion.

The last theorem in this section is very useful since it allows one to work with at most binary constraints, which often simplifies representation and analysis of CSP. For the sake of clarity, we will further restrict the discussion to constraint languages with at most binary relations. It must be stressed that all results in the paper can be extrapolated to any other finite constraint languages (with possibly more tedious representation).

Theorem 5. *For any constraint language Γ there is a constraint language Γ' such that all relations in Γ' are at most binary and Γ and Γ' pp-construct each other.*

2.3   Characterization of a CSP instance

This subsection introduces some properties of a CSP instance that will be used in Zhuk's algorithm [19] and provides their interpretations in terms of constraint languages with at most binary relations.

We say that a variable y_i of a constraint $C_j = (y_1, ..., y_k; R)$ is *dummy* if R does not depend on its i-th variable. A relation $R \subseteq D_0 \times ... \times D_{n-1}$ is *subdirect* if for every i the projection of R onto the i-th coordinate is the whole D_i. A CSP instance Θ with a domain set D is called 1-*consistent* (or *arc consistent*) if for every constraint C_i of the instance the corresponding relation $R_i \subseteq D_{i_1} \times ... \times D_{i_k}$ is subdirect. An arbitrary instance can be turned into 1-consistent instance with the same set of solutions by a simple algorithm [3].

Another type of consistency is related to the notion of a path. Let D_y denote the domain of the variable $y \in \{x_1, ..., x_n\}$. We say that the sequence $y_1 - C_1 - y_2 - ... - y_{l-1} - C_{l-1} - y_l$ is a *path* in a CSP instance if $\{y_i, y_{i+1}\}$ are in the scope of C_i for every $i < l$ (we do not care in what order variables y_i, y_{i+1} occur in C_i). We say that the path *connects* b and c if there exists $a_i \in D_{y_i}$ for every i such that $a_1 = b$, $a_l = c$ and the projection of C_i onto $\{y_i, y_{i+1}\}$ contains the tuple (a_i, a_{i+1}). We say that a CSP instance is *cycle-consistent* if it is 1-consistent and for every variable y and $a \in D_y$ *any* path starting and ending with y connects a and a. A CSP instance is called *linked* if for every variable y occurring in the scope of a constraint C and for all $a, b \in D_y$ there *exists* a path starting and ending with y in Θ that connects a and b.

A fragmented CSP instance can be divided into several nontrivial instances: an instance is *fragmented* if the set of variables X can be divided into 2 disjoint sets X_1 and X_2 such that each of them is non-empty, and the constraint scope of any constraint of Θ either has variables only from X_1, or only from X_2. We call an instance $\Theta = (X, D, C)$ *irreducible* if any instance $\Theta' = (X', D', C')$ such that $X' \subseteq X$, $D'_x = D_x$ for every $x \in X'$, and every constraint of Θ' is a projection of a constraint from Θ on *some* subset of variables from X' is fragmented, or linked, or its solution set is subdirect.

One of the important notions of Zhuk's algorithm is a weaker constraint: by weakening some constraints we make an instance weaker (i.e. possibly having more solutions). We say that a constraint $C_1 = ((y_1, ..., y_t), \rho_1)$ is *weaker or equivalent* to a constraint $C_2 = ((z_1, ..., z_s), \rho_2)$ if $\{y_1, ..., y_t\} \subseteq \{z_1, ..., z_s\}$ and C_2 implies C_1, i.e the solution set to $\Theta_1 = (\{z_1, ..., z_s\}, (D_{z_1}, ..., D_{z_s}), C_1)$ contains the solution set to $\Theta_2 = (\{z_1, ..., z_s\}, (D_{z_1}, ..., D_{z_s}), C_2)$. We say that C_1 is *weaker* than C_2 (denoted $C_1 \leq C_2$) if C_1 is weaker or equivalent to C_2, but C_1 does not imply C_2. There can be 2 types of weaker constraints. We say that $C_1 = ((y_1, ..., y_t), \rho_1) \leq C_2 = ((z_1, ..., z_s), \rho_2)$ with $\{y_1, ..., y_t\} \subseteq \{z_1, ..., z_s\}$ if one of the following conditions holds:

1. The arity of relation ρ_1 is less than the arity of relation ρ_2 and for any tuple $(a_{z_1}, ..., a_{z_s}) \in \rho_2$, $(a_{y_1}, ..., a_{y_t}) \in \rho_1$.

2. The arities of relations ρ_1 and ρ_2 are equal and $\rho_2 \subsetneq \rho_1$.

All the above-mentioned properties have simple interpretations in terms of constraint languages with at most binary relations. Generally, CSP is defined as having a single common "superdomain" D for all variables. However, even though domains can be all equal at the beginning, Zhuk's algorithm will create different domains for individual variables. We require each $D_i, i \in \{0, ..., n-1\}$ to be *pp*-definable over the constraint language Γ, i.e. $\mathrm{CSP}(\Gamma)$ is *p*-equivalent to $\mathrm{CSP}(\Gamma, D_0, ..., D_{n-1})$. Any constraint for the CSP instance is either $C = (x_i; D_i)$, where D_i is a restriction on the domain for the variable x_i, or $C = (x_i, x_j; E^{ij})$. Every unary relation can be viewed as a domain and every binary

relation - as an edge, where the order corresponds to the direction. So it is natural to refer to these relational structures as some sort of digraphs and to the CSP problem as a homomorphism problem between relational structures.

In our case, an input relational structure is a classical digraph $\mathcal{X} = (V_{\mathcal{X}}, E_{\mathcal{X}})$ with $V_{\mathcal{X}} = \{x_1, ..., x_n\}$. Let us call a target relational structure a *digraph with domains* $\ddot{\mathcal{A}} = (V_{\ddot{\mathcal{A}}}, E_{\ddot{\mathcal{A}}}^{ij} : 0 \leq i, j < n)$, where $V_{\ddot{\mathcal{A}}} = \{D_0, ..., D_{n-1}\}$. The problem is in finding a homomorphism such that it sends every x_i to the domain D_i and every edge $(x_i, x_j) \in E_{\mathcal{X}}$ to an edge $(a, b) \in E_{\ddot{\mathcal{A}}}^{ij}$ (relations $E_{\ddot{\mathcal{A}}}^{ij}$ can differ for all i, j). We will denote the corresponding instance by $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$.

In this setting, a 1-consistent CSP instance is an instance in which for every edge (x_i, x_j) from $E_{\mathcal{X}}$, for any element $a \in D_i$ there is an element $b \in D_j$ such that $(a, b) \in E_{\ddot{\mathcal{A}}}^{ij}$ and vice versa. A variable x_i of an edge $(x_i, x_j) \in E_{\mathcal{X}}$ is dummy if for every $b \in D_j$ such that there exists $a \in D_i$, $E_{\ddot{\mathcal{A}}}^{ij}(a, b)$, there is an edge $(a', b) \in E_{\ddot{\mathcal{A}}}^{ij}$ for every $a' \in D_i$. Note that for a 1-consistent CSP instance this means that $E_{\ddot{\mathcal{A}}}^{ij}$ is a full relation.

Since we work with digraphs, by *undirected path or cycle* in the paper are meant any path or cycle with edges not necessarily directed in the same direction. A path $y_0 - C_0 - y_1 - ... - y_{t-1} - C_{t-1} - y_t$ is an undirected path in digraph $\mathcal{X}$ (where some variables y_i, y_j can be the same). Consider this path as a separate digraph $\mathcal{P}_t$ with new (all different) vertices $s_0 - C_0 - s_1 - ... - s_{t-1} - C_{t-1} - s_t$, and consider a homomorphism H from $\mathcal{P}_t$ to $\mathcal{X}$ such that for each $i \leq t$, $H(s_i) = y_i$. We say that path $\mathcal{P}_t$ connects elements $b \in D_{y_0}$ and $c \in D_{y_t}$ if it can be homomorphically mapped to $\ddot{\mathcal{A}}$ in such a way that for each $i \leq t$ homomorphism $H' : \mathcal{P}_t \to \ddot{\mathcal{A}}$ sends s_i to some $a_i \in D_{y_i}$ and $H'(s_0) = b$, $H'(s_t) = c$. An instance is linked if for any $a, b \in D_y$ there exists an undirected path that connects a and b. Cycle-consistency in these terms means that the instance is 1-consistent and for any $a \in D_y$ and any $y \in \{x_0, ..., x_{n-1}\}$ any undirected path that is a cycle connects a and a. In other words, an instance is cycle-consistent if any undirected cycle in $\mathcal{X}$ can be homomorphically mapped onto a cycle in $\ddot{\mathcal{A}}$ for any element $a \in D_y$ and any $y \in \{x_0, ..., x_{n-1}\}$ that occurs in this cycle.
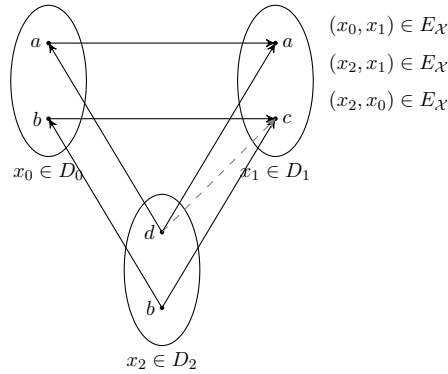

Figure 1: Cycle-consistent, non-linked instance.

Compare as examples two CSP instances in Figure 1 and Figure 2. The input digraph $\mathcal{X}$ is the same for both instances, $V_{\mathcal{X}} = \{x_0, x_1, x_2\}$, $E_{\mathcal{X}} = \{(x_0, x_1), (x_2, x_1), (x_2, x_0)\}$. The first CSP instance has three constraint relations, $E_{\ddot{\mathcal{A}}}^{01} = \{(a, a), (b, c)\})$, $E_{\ddot{\mathcal{A}}}^{21} = \{(d, a), (b, c)\})$ and $E_{\ddot{\mathcal{A}}}^{20} = \{(d, a), (b, b)\}$. This instance is cycle-consistent since it is 1-consistent (each constraint of the instance is subdirect) and for every variable x and $e \in D_x$ any path starting and ending with x connects e and e. But it is not linked since,
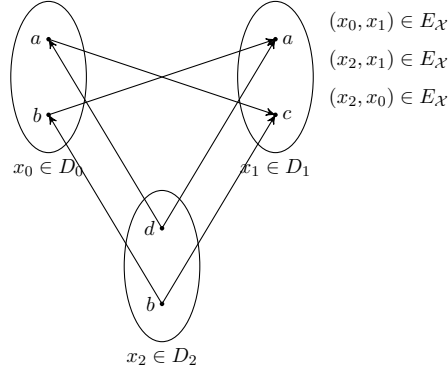
7

Figure 2: Linked, not cycle-consistent instance.

for example, there is no path connecting a and b in D_0. However, if we add one more edge (d, c) to $E_{\ddot{\mathcal{A}}}^{21}$, the new instance will be linked. On the contrary, the second instance in Figure 2 is linked, but not cycle-consistent.

A fragmented instance in terms of digraphs and digraphs with domains is such an instance where $\mathcal{X}$ is a disconnected digraph. Finally, if an instance is not irreducible, then there exists a subgraph $\mathcal{X}'$ (a digraph formed from subsets of vertices $V_{\mathcal{X}'} \subseteq V_{\mathcal{X}}$ and edges $E_{\mathcal{X}'} \subseteq E_{\mathcal{X}}$) such that the resulting instance $\Theta = (\mathcal{X}', \ddot{\mathcal{A}})$ is not fragmented, is not linked, and its solution set is not subdirect.

Since there are two types of weaker constraints (of less arity or of richer relation of the same arity), we can weaken the CSP instance $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$ either by removing an edge $(x_i, x_j) \in E_{\mathcal{X}}$ from $\mathcal{X}$ (i.e. by reducing the arity of a constraint) or by adding edges to a relation $E_{\ddot{\mathcal{A}}}^{ij}$ (i.e. by making a richer relation of the same arity). The algorithm never increases the domains.
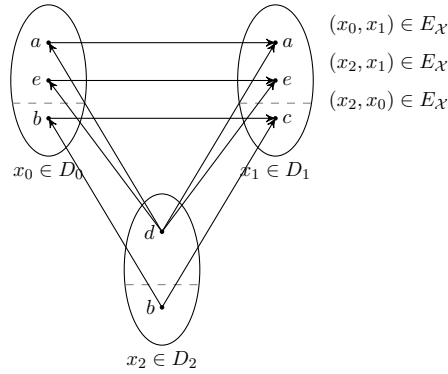

Figure 3: Division into linked components.

We conclude this subsection with Lemma 2 to be used further for the formalization of Zhuk's algorithm. For an instance Θ and its variable x let $Linked(\Theta, x)$ denote the binary relation on D_x defined as follows: $(a, b) \in Linked(\Theta, x)$ if there exists a path in Θ that connects a and b.

Lemma 2 ([19])**.** *Suppose Θ is a cycle-consistent CSP instance such that every its variable $x \in X$ actually occurs in some constraint of Θ. Then for every $x \in X$ there exists a path*

in Θ connecting all pairs $(a, b) \in Linked(\Theta, x)$ and $Linked(\Theta, x)$ is a congruence.

For example, consider cycle-consistent non-linked instance Θ in Figure 3. Binary relation $Linked(\Theta, x)$ divides each domain into two classes: D_0 into $\{a, e\}$ and $\{b\}$, D_1 into $\{a, e\}$ and $\{c\}$, and D_2 into $\{d\}$ and $\{b\}$.

2.4 The theory V^1

In this subsection most definitions and results are adapted from [9], [15], [16].

Second-order (or *two-sorted* first-order) theories of bounded arithmetic use the following setup. The variables are of two kinds: variables $x, y, H, ...$ of the first kind are called *number variables* and range over the natural numbers, and variables $X, Y, H, ...$ of the second kind are called *set variables* and range over finite subsets of natural numbers (which can be represented as binary strings). Functions and predicate symbols can use both number and set variables, and there are *number-valued* functions and *set-valued* functions. Also, there are two types of quantifiers: quantifiers over number variables are called *number quantifiers*, and quantifiers over set variables are called *string quantifiers*. The language for the second-order theory of bounded arithmetic is an extension of the standard language for Peano Arithmetic $\mathcal{L}_{\mathcal{PA}}$,

$$\mathcal{L}^2_{\mathcal{PA}} = \{0, 1, +, \cdot, |\,|, =_1, =_2, \leq, \in\}.$$

The symbols $0, 1, +, \cdot, =_1$ and $\leq$ are function and predicate symbols over the number variables. The function $|X|$ (called the *length of X*) is a number-valued function and it denotes the length of the corresponding string X (i.e. the upper bound for the set X). The binary predicate $\in$ for a number and a set variables denotes set membership, and $=_2$ is the equality predicate for sets.

Notation 1. *We will use the abbreviation $X(t) =_{def} t \in X$, where t is a number term. We thus think of $X(i)$ as of the i-th bit of binary string X of length $|X|$.*

There is a set of axioms 2-BASIC [9] that defines basic properties of symbols from $\mathcal{L}^2_{\mathcal{PA}}$. Here we present only axioms of the second sort:

Definition 4 (2-BASIC, second-sort axioms)**.** The set 2-BASIC for the second-sort variables contains the following axioms:

1. $X(y) \to y < |X|$.

2. $y + 1 =_1 |X| \to X(y)$.

3. $(|X| =_1 |Y| \wedge \forall i < |X|(X(i) \leftrightarrow X(i))) \iff X =_2 Y$.

We will skip the indices $=_1, =_2$ as there is no danger of confusion.

Notation 2. *Sometimes for a set A, an element x and a formula ϕ instead of $\exists x < |A|\, A(x) \wedge \phi$ and $\forall x < |A|\, A(x) \to \phi$ we will write $\exists x \in A\, \phi$ and $\forall x \in A\, \phi$.*

Definition 5 (Bounded formulas)**.** Let $\mathcal{L}$ be the two-sorted vocabulary. If x is a number variable, X is a string variable that do not occur in an $\mathcal{L}$-number term t, then $\exists x \leq t\phi$ stands for $\exists x(x \leq t \wedge \phi)$, $\forall x \leq t\phi$ stands for $\forall x(x \leq t \to \phi)$, $\exists X \leq t\phi$ stands for $\exists X(|X| \leq t \wedge \phi)$ and $\forall X \leq t\phi$ stands for $\forall X(|X| \leq t \to \phi)$. Quantifiers that occur in this form are said to be *bounded*, and a *bounded formula* is one in which every quantifier is bounded.

Definition 6 (Number Induction axioms)**.** If Φ is a set of two-sorted formulas, then Φ-IND axioms are the formulas

$$\phi(0) \wedge \forall x(\phi(x) \rightarrow \phi(x+1)) \rightarrow \forall z \phi(z), \tag{1}$$

where ϕ is any formula in Φ. The formula $\phi(x)$ may have other free variables than x of both sorts.

Definition 7 (Number Minimization and Maximization axioms)**.** The number minimization axioms (or the least number principle axioms) for a set Φ of formulas are denoted by Φ-MIN and consist of the formulas

$$\phi(y) \rightarrow \exists x \leq y(\phi(x) \wedge \neg \exists H < x \, \phi(z)), \tag{2}$$

where ϕ is a formula in Φ. Similarly, the number maximization axioms for Φ are denoted by Φ-MAX and consist of the formulas

$$\phi(0) \rightarrow \exists x \leq y(\phi(x) \wedge \neg \exists H \leq y(x < z \wedge \phi(z))), \tag{3}$$

where ϕ is a formula in Φ. In the above definitions, ϕ is permitted to have free variables of both sorts, in addition to x.

Definition 8 (Comprehension axioms)**.** If Φ is a set of two-sorted formulas, then Φ-COMP is the set of all formulas

$$\forall x \exists X \leq x \, \forall y < x \, y \in X \equiv \phi(y), \tag{4}$$

where ϕ is any formula in Φ, and X does not occur free in $\phi(y)$. The formula $\phi(y)$ may have other free variables than y of both sorts.

Finally, we can define the theory V^1, which is the key theory for our work.

Definition 9 (The theory V^1)**.** $\Sigma_0^{1,b} = \Pi_0^{1,b}$-formulas are formulas with all number quantifiers bounded and with no set-sort quantifiers. Classes $\Sigma_1^{1,b}$ and $\Pi_1^{1,b}$ are the smallest classes of $\mathcal{L}^2_{\mathcal{PA}}$-formulas such that:

1. $\Sigma_0^{1,b} \cup \Pi_0^{1,b} \subseteq \Sigma_1^{1,b} \cap \Pi_1^{1,b}$,

2. both $\Sigma_1^{1,b}$ and $\Pi_1^{1,b}$ are closed under $\vee$ and $\wedge$,

3. the negation of a formula $\Sigma_1^{1,b}$ is in $\Pi_1^{1,b}$ and vice versa,

4. if $\phi \in \Sigma_1^{1,b}$, then also $\exists X \leq t \, \phi \in \Sigma_1^{1,b}$,

5. if $\phi \in \Pi_1^{1,b}$, then also $\forall X \leq t \, \phi \in \Pi_1^{1,b}$.

The theory $I\Sigma_0^{1,b}$ is a second-order theory and it is axiomatized by 2-BASIC and the IND scheme for all $\Sigma_0^{1,b}$-formulas. The teory V^0 expands $I\Sigma_0^{1,b}$ by having also bounded comprehension axioms $\Sigma_0^{1,b}$-CA. The theory V^0 is a conservative extension of $I\Sigma_0^{1,b}$ with respect to $\Sigma_0^{1,b}$-consequences: if γ is a $\Sigma_0^{1,b}$-formula and V^0 proves its universal closure, so does $I\Sigma_0^{1,b}$. Finally, the theory V^1 extends V^0 by accepting the IND scheme for all $\Sigma_1^{1,b}$-formulas.

2.5 Auxiliary functions, relations, and axioms in V^1

In this subsection, we will present some general auxiliary functions and relations, which help to express the bounds of the theory V^1.

For any two sets A, B, we say that a set B is a *subset* of A if

$$subS(B, A) \iff |A| = |B| \wedge \forall i < |B| \, (B(i) \rightarrow A(i)). \tag{5}$$

We say that a set B is a *proper subset* of A if

$$PsubS(B, A) \iff |A| = |B| \wedge \forall i < |B| \, (B(i) \rightarrow A(i)) \wedge$$
$$\exists j < |A|, B(j) \wedge \exists i < |A|, A(i) \wedge \neg B(i). \tag{6}$$

If $x, y \in \mathbb{N}$, we define the *pairing function* $\langle x, y \rangle$ to be the following term

$$\langle x, y \rangle = \frac{(x+y)(x+y+1)}{2} + y. \tag{7}$$

One can easily prove in V^0 that for the pairing function the following is true:

- $\forall x_1, x_2, y_1, y_2 \, (\langle x_1, y_1 \rangle = \langle x_2, y_2 \rangle \rightarrow x_1 = x_2 \wedge y_1 = y_2)$,

- $\forall z \exists x, y \, (\langle x, y \rangle = z)$,

- $\forall x, y \;\; (x, y \le \langle x, y \rangle < (x + y + 1)^2)$.

We can iterate the pairing function to code triples, quadruples, and so forth for any k, inductively setting

$$\langle x_1, x_2, ..., x_k \rangle = \langle ... \langle \langle x_1, x_2 \rangle, x_3 \rangle, ..., x_k \rangle, \tag{8}$$

where

- $\forall x_1, x_2, ..., x_k \;\; x_1, x_2, ..., x_k \le \langle x_1, x_2, ..., x_k \rangle < (x_1 + x_2 + ... + x_k + 1)^{2^k}$.

We refer to the term $\langle x_1, x_2, ..., x_k \rangle$ as the *tupling function*.

Notation 3. *For any set H, $m \ge 2$: $H(x_1, ..., x_m) =_{def} H(\langle x_1, ..., x_m \rangle)$.*

We will use the tupling function to code a function as a set. We can then express that H is a function from sets $X_1, ..., X_n$ to a set Y by stating

$$\forall x_1 \in X_1, ..., \forall x_n \in X_n \exists! y \in Y \, H(x_1, ..., x_n, y).$$

We will abbreviate it as $Z : X_1, ..., X_n \rightarrow Y$ and $H(x_1, ..., x_n) = y$. Using the pairing function (or encoding of k-tuples), with finite sets we can also code binary (or k-ary) relations. Finite functions can be represented by their digraphs. For example, to represent an $m \times n$ matrix A with natural number entries we think of it as of a function from $[m] \times [n]$ into N. The matrix is thus encoded by the set $A(i, j, a)$, and we write $A_{ij} = a$ for the corresponding entry.

We say that a set H is a *well-defined map* between two sets $A, |A| = n$ and $B, |B| = m$ if it satisfies the relation

$$MAP(A, n, B, m, H) \iff \forall i \in A \exists j \in B \wedge H(i) = j \wedge$$
$$\forall i \in A \, \forall j_1, j_2 \in B \, (H(i) = j_1 \wedge H(i) = j_2 \rightarrow j_1 = j_2). \tag{9}$$

11

The counting axiom allows one to count the number of elements in a set. Given a set X, the *census function* $\#X(n)$ for X is a number function defined for $n \leq |X|$ such that $\#X(n)$ is the number of $x < n$, $x \in X$. Thus, $\#X(|X|)$ is the number of elements in X. The following relation says that $\#X$ is the census function for X:

$$Census(X, \#X) \iff \#X \leq \langle |X|, |X| \rangle \wedge \#X(0) = 0 \wedge \forall x < |X|$$
$$(x \in X \rightarrow \#X(x+1) = \#X(x) + 1 \wedge x \notin X \rightarrow \#X(x+1) = \#X(x)). \tag{10}$$

Lemma 3. *For any set X, V^1 proves that there exists its census function.*

Proof. Given any set X, consider $\Sigma_1^{1,b}$-induction on $n \leq |X|$ for the formula

$$\phi(n) = \exists H \leq \langle n, n \rangle \, H(0) = 0 \wedge \forall 0 \leq x < n$$
$$(x \in X \rightarrow H(x+1) = H(x) + 1 \wedge x \notin X \rightarrow H(x+1) = H(x)). \tag{11}$$

$\square$

We will now remind the reader a few well-known number-theoretic functions and relations, mainly to fix the notation. They are all definable in a weak subtheory of V^1 and the stated properties are proved in [9],[15]. The relation of *divisibility* can be defined by the formula

$$x|y \iff \exists z \leq y (xz = y). \tag{12}$$

We say that p is a *prime number* if it satisfies the relation

$$Prime(p) \iff 1 < p \wedge \forall y < p \forall z < p \, (yz \neq p). \tag{13}$$

It is easily seen that V^1 proves that any $x > 0$ is uniquely representable by a product of powers of primes. The *limited subtraction* $a \dot{-} b = max\{0, a - b\}$ can be defined by

$$c = a \dot{-} b \longleftrightarrow ((b + c = a) \vee (a \leq b \wedge c = 0)), \tag{14}$$

and the *division* $\lfloor a/b \rfloor$ for $b \neq 0$ can be defined as follows:

$$c = \lfloor a/b \rfloor \longleftrightarrow (bc \leq a \wedge a < b(c+1)). \tag{15}$$

Finally, the *remainder* of a after being divided by p can be defined by the formula

$$a \, mod \, p = a \dot{-} (p \cdot \lfloor a/p \rfloor). \tag{16}$$

We say that two numbers are *congruent modulo p*, denoted $c_1 \equiv c_2 (mod \, p)$ if $c_1 \, mod \, p = c_2 \, mod \, p$. It means that if $c_1 < c_2$, then

$$c_1 \dot{-} (p \cdot \lfloor c_1/p \rfloor) = c_2 \dot{-} (p \cdot \lfloor c_2/p \rfloor),$$
$$c_2 \dot{-} c_1 = p(\lfloor c_2/p \rfloor \dot{-} \lfloor c_1/p \rfloor), \tag{17}$$

i.e. the difference $c_2 - c_1$ is divisible by p. Note that it is straightforward to show in V^1 that for all $x_1 \equiv x_2 (mod \, p)$ and $y_1 \equiv y_2 (mod \, p)$,

$$(x_1 + y_1) \equiv (x_2 + y_2)(mod \, p),$$
$$(x_1 y_1) \equiv (x_2 y_2)(mod \, p). \tag{18}$$

3 Zhuk's four cases

One of the two main ideas of Zhuk's algorithm is based on strong subalgebras. In this section we will give the definitions of absorbing subuniverse, center and central subuniverse, and polynomially complete algebra and briefly mention their main properties. Further, we consider the notion of linear algebras as introduced in [19] and give two elementary examples of relational structures corresponding to linear algebras. Finally, we will formulate Zhuk's four-cases theorem.

3.1 Absorption, center and polynomial complete algebras

If $\mathbb{B} = (B, F_B)$ is a subalgebra of $\mathbb{A} = (A, F_A)$, then B *absorbs* $\mathbb{A}$ if there exists an n-ary term operation $f \in Clone(F_A)$ such that $f(a_1, ..., a_n) \in B$ whenever the set of indices $\{i : a_i \notin B\}$ has at most one element. B *binary absorbs* A if there exists a binary term operation $f \in Clone(F_A)$ such that $f(a, b) \in B$ and $f(b, a) \in B$ for any $a \in A$ and $b \in B$.

If $\mathbb{A} = (A, \Omega_A)$ is a finite algebra with a special WNU operation, then $C \subseteq A$ is a *center* if there exists an algebra $\mathbb{B} = (B, \Omega_B)$ with a special WNU operation of the same arity and a subdirect subalgebra $\mathbb{D} = (D, \Omega_D)$ of $\mathbb{A} \times \mathbb{B}$ such that there is no nontrivial binary absorbing subuniverse in $\mathbb{B}$ and $C = \{a \in A | \forall b \in B : (a, b) \in D\}$. Every center is a ternary absorbing subuniverse. A weaker notion, suggested by Zhuk in [20], is a central subuniverse. A subuniverse C of $\mathbb{A}$ is called *central* if it is an absorbing subuniverse and for every $a \in A \backslash C$ we have $(a, a) \notin Sg(\{a\} \times C \cup C \times \{a\})$. A central subuniverse has all the good properties of a center and can be used in Zhuk's algorithm instead of the center. Both algorithms, with the center or central universe, will correctly answer whether an instance has a solution, or not.

For any set A denote by $O_n(A)$ the set of all n-ary operations on A. The clone of all operations on A is denoted by $O(A) = \{O_n(A) | n \geq 0\}$. An n-ary operation f on algebra $\mathbb{A} = (A, F_A)$ is called *polynomial* if there exist some $(n + t)$-ary operation $g \in Clone(F_A)$ and constants $a_1, ..., a_t \in A$ such that for all $x_1, ..., x_n \in A$, $f(x_1, ..., x_n) = g(x_1, ..., x_n, a_1, ..., a_m)$. Denote the clone generated by F_A and all the constants on A (i.e. the set of all polynomial operations on $\mathbb{A}$) by $Polynom(\mathbb{A})$. We call an algebra $\mathbb{A} = (A, F_A)$ *polynomially complete* (PC) if its polynomial clone is the clone of all operations on A, $O(A)$. In simple words, a universal algebra $\mathbb{A}$ is polynomially complete if every function on A with values in A is a polynomial function. A classical result about polynomial completeness is based on the following notion. The *ternary discriminator function* is the function t defined by the identities

$$t(x, y, z) = \begin{cases} z, \ x = y, \\ x, \ x \neq y. \end{cases}$$

Then Theorem 6 gives a necessary and sufficient condition of polynomial completeness.

Theorem 6 ([5]). *A finite algebra is polynomially complete if and only if it has the ternary discriminator as a polynomial operation.*

3.2 Linear algebras: properties and examples on digraphs

Definition 10 (Linear algebra, [19]). An idempotent finite algebra $\mathbb{A} = (A, \Omega)$, where Ω is an m-ary idempotent special WNU operation, is called *linear* if it is isomorphic to $(\mathbb{Z}_{p_1} \times ... \times \mathbb{Z}_{p_s}, x_1 + ... + x_m)$ for prime (not necessarily distinct) numbers $p_1, ..., p_s$. For every finite idempotent algebra, there exists the smallest congruence (not necessarily proper), called the *minimal linear congruence*, such that the factor algebra is linear.

To understand how linear algebras appear in Zhuk's algorithm, and to establish some of their properties, we consider the notion of an affine algebra. An algebra $\mathbb{A} = (A, F)$ is called *affine* if there is an abelian group $\mathbb{A}' = (A, 0, -, +)$ such that the relation $R = \{(x, y, z, u) : (x + y = z + u)\}$ is preserved by all operations of $\mathbb{A}$ [12]. Affine algebra is polynomially equivalent (has the same polynomial clone) to a module. It means that each term operation of algebra $\mathbb{A}$ is affine with respect to the abelian group $\mathbb{A}'$, i.e. to say, for any given n-ary operation $f \in F$ there are endomorphisms $\alpha_1, ..., \alpha_n$ of $\mathbb{A}$ and an element $a \in A$ such that f can be expressed identically as in [12]:

$$f(x_1, ..., x_n) = \sum_{i=1}^{n} \alpha_i(x_i) + a.$$

The following lemma establishes one important property of an affine algebra in case there is an idempotent WNU operation on $\mathbb{A}$. We will provide its proof as in [18], to make some notes further.

Lemma 4 ([18]). *Suppose $\mathbb{A}' = (A, 0, -, +)$ is a finite abelian group, the relation $R \subseteq A^4$ is defined by $R = \{(x, y, z, u) : (x + y = z + u)\}$, R is preserved by an idempotent WNU m-ary operation Ω. Then $\Omega(x_1, ...x_m) = tx_1 + ... + tx_m$ for some $t \in \mathbb{N}$.*

Proof. Define $h(x) = \Omega(0, 0, ..., 0, x)$. We will prove the equation

$$\Omega(x_1, ..., x_i, 0, ..., 0) = h(x_1) + ... + h(x_i)$$

by induction on i. For $m = 1$ it follows from the definition and properties of WNU. We know that

$$\Omega \begin{pmatrix} x_1 & x_2 & ... & x_i & x_{i+1} & 0 & ... & 0 \\ 0 & 0 & ... & 0 & 0 & 0 & ... & 0 \\ x_1 & x_2 & ... & x_i & 0 & 0 & ... & 0 \\ 0 & 0 & ... & 0 & x_{i+1} & 0 & ... & 0 \end{pmatrix} \in R$$

is in R, which by the inductive assumption gives

$$\begin{aligned} \Omega(x_1, ..., x_i, x_{i+1}, 0, ..., 0) = \Omega(x_1, ..., x_i, 0, 0, ..., 0) + h(x_{i+1}) = \\ = h(x_1) + ... + h(x_i) + h(x_{i+1}). \end{aligned} \tag{19}$$

We thus know that $\Omega(x_1, ..., x_m) = h(x_1) + ... + h(x_m)$. Let p be the maximal order of an element in group $\mathbb{A}' = (A, 0, -, +)$. Then for any element a in A, the order of a divides p, and in particular $pa = 0$. For every $a \in A$ we have $\underbrace{h(a) + h(a) + ... + h(a)}_{m} = \Omega(a, a, ..., a) = a$. Thus, for any element $a \neq 0$, $m \cdot h(a) \neq 0$, hence m does not divide an order of any element in $\mathbb{A}'$ and therefore m and p are coprime. Hence m has the multiplicative inverse modulo p and there is some integer t such that $tm = 1$, $m \cdot h(x) = h(x)/t = x$, and $h(x) = tx$ for every x. $\square$

If we additionally assume that Ω is special (by Lemma 1), then $t = 1$:

$$\Omega(x, ..., x, \Omega(x, ..., x, y)) = \Omega(x, ..., x, y),$$
$$\underbrace{tx + ... + tx}_{m-1} + t\Omega(x, ..., x, y) = \underbrace{tx + ... + tx}_{m-1} + ty,$$
$$t(\underbrace{tx + ... + tx}_{m-1} + ty) = ty, \tag{20}$$
$$\underbrace{tx + ... + tx}_{m-1} + ty + tx = y + tx$$
$$x + ty = y + tx \implies t = 1.$$

14

Consider any finite affine algebra $\mathbb{A}$. Due to the well-known Classification theorem [13] every finite abelian group is isomorphic to a product of cyclic groups whose orders are all prime powers. Thus $A = \mathbb{Z}_{p_1^{r_1}} \times ... \times \mathbb{Z}_{p_s^{r_s}}$ for some not necessarily distinct primes $p_1, ..., p_s$. If p is the maximal order of an element in $\mathbb{A}'$, then, by the above proof, $m = 1 \, (mod \, p)$. Therefore, since every p_i has to divide p, every p_i also divides $(m - 1)$. If there is an idempotent WNU operation on $\mathbb{A}$, then there exists the minimal linear congruence σ such that $\mathbb{A}/\sigma$ is isomorphic to a linear algebra.

Finally, we will formulate and prove an important theorem used in Zhuk's algorithm.

Theorem 7 (Affine subspaces [19])**.** *Suppose that relation $\rho \subseteq (\mathbb{Z}_{p_1})^{n_1} \times ... \times (\mathbb{Z}_{p_k})^{n_k}$ is preserved by $x_1 + ... + x_m$, where $p_1, ..., p_k$ are distinct prime numbers dividing $m - 1$ and $\mathbb{Z}_{p_i} = (\mathbb{Z}_{p_i}, x_1 + ... + x_m)$ for every i. Then $\rho = L_1 \times ... \times L_k$, where each L_i is an affine subspace of $(\mathbb{Z}_{p_i})^{n_i}$.*

Proof. We first derive a ternary operation on every $\mathbb{Z}_{p_i}$.

$$f(x, y, z) = x - y + z \, (mod \, p_i) = \Omega(x, z, 0, ..., 0) + \Omega(y, ..., y, 0, 0) =$$
$$= x + z + y + ... + y = \Omega(x, z, y, ..., y). \tag{21}$$

Thus, $f(x, y, z)$ preserves ρ. Now consider the relation $\rho \subseteq (\mathbb{Z}_{p_1})^{n_1} \times ... \times (\mathbb{Z}_{p_k})^{n_k}$ and choose any element $a \in \rho$. The set $\vec{V} = \{v | a + v \in \rho\}$ obviously contains 0. Moreover, it is closed under $+$. Consider any $v_1, v_2 \in \vec{V}$, $a + v_1, a + v_2 \in \rho$. Then $v_1 + v_2 \in \vec{V}$ since $f(a + v_1, a, a + v_2) = a + v_1 + v_2 \in \rho$. Thus, $\vec{V}$ is a linear subspace and ρ is therefore an affine subspace. $\qquad\square$

In the remainder of this subsection we will give two elementary examples of constraint languages corresponding to linear algebras. We will consider classical digraphs, relational structures with unique binary relation of being an edge. Due to Theorem 1, each relational structure $\mathcal{A}$ corresponds to an algebra $\mathbb{A}$ such that $Clone(\mathbb{A}) = Pol(\mathcal{A})$. We can assume that for both CSP instances there is a special WNU operation Ω of some arity m, which is a polymorphism for all constraint relations.
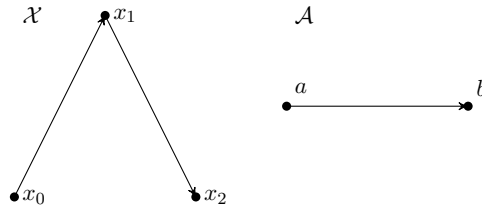

Figure 4: Example 1.

Consider CSP($\mathcal{A}$), where $\mathcal{A} = (V_{\mathcal{A}}, E_{\mathcal{A}})$ is the digraph on two vertices and $E_{\mathcal{A}} = \{(a, b)\}$. An instance of CSP($\mathcal{A}$), depicted in Figure 4, is the digraph $\mathcal{X} = (V_{\mathcal{X}}, E_{\mathcal{X}})$, where $V_{\mathcal{X}} = \{x_0, x_1, x_2\}$ and $E_{\mathcal{X}} = \{(x_0, x_1), (x_1, x_2)\}$. It is obvious that there is no homomorphism from $\mathcal{X}$ to $\mathcal{A}$. Let us define a 3-ary operation Ω on $V_{\mathcal{A}}$ as follows:

$$\Omega(a, a, a) = a, \;\; \Omega(b, b, b) = b,$$
$$\Omega(b, a, a) = \Omega(a, b, a) = \Omega(a, a, b) = b, \tag{22}$$
$$\Omega(a, b, b) = \Omega(b, a, b) = \Omega(b, b, a) = a.$$

Ω preserves $E_{\mathcal{A}}$ and is clearly idempotent, WNU and special:

$$\begin{aligned}
\Omega(a, a, \Omega(a, a, b)) &= \Omega(a, a, b) = b, \\
\Omega(b, b, \Omega(b, b, a)) &= \Omega(b, b, a) = a.
\end{aligned} \tag{23}$$

We can define an operation $+$ on $V_{\mathcal{A}}$ as $(a+x) = (x+a) = x$ (i.e. a is zero) and $(b+b) = a$ (i.e. b is an inverse element to itself). Hence $\mathbb{A} = (V_{\mathcal{A}}, +)$ is a finite abelian group, namely $\mathbb{Z}_2$, and the algebra $(V_{\mathcal{A}}, \Omega)$ is isomorphic to linear algebra $(\mathbb{Z}_2, x + y + z)$.

The instance has two constraints, $E_{\mathcal{X}}(x_0, x_1) \subseteq \mathbb{Z}_2 \times \mathbb{Z}_2$ and $E_{\mathcal{X}}(x_1, x_2) \subseteq \mathbb{Z}_2 \times \mathbb{Z}_2$. Since $E_{\mathcal{A}} = \{(a, b)\}$ is an affine subspace of $\mathbb{Z}_2 \times \mathbb{Z}_2$, we can express constraints as a conjunction of the linear equations

$$E_{\mathcal{X}}(x_0, x_1) \iff \left\{ \begin{array}{l} x_0 = a, \\ x_1 = b. \end{array} \right.$$

$$E_{\mathcal{X}}(x_1, x_2) \iff \left\{ \begin{array}{l} x_1 = a, \\ x_2 = b. \end{array} \right.$$

The instance can be viewed as a system of linear equations in different fields and it has no solution.

Now consider a different example in Figure 5, where $\mathcal{A} = (V_{\mathcal{A}}, E_{\mathcal{A}})$ is the digraph on two vertices with $E_{\mathcal{A}} = \{(a, b), (b, a)\}$, and the instance digraph $\mathcal{X} = (V_{\mathcal{X}}, E_{\mathcal{X}})$ is the same.
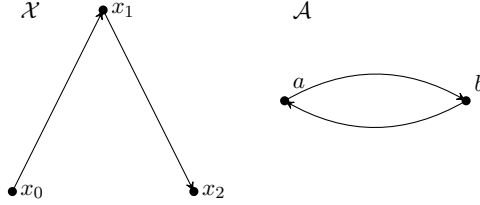

Figure 5: Example 2.

Since the constraint relation $E_{\mathcal{A}}$ is still preserved by above defined Ω, $(V_{\mathcal{A}}, \Omega)$ is isomorphic to $(\mathbb{Z}_2, x + y + z)$. But $E_{\mathcal{A}}$ differs from the relation in the previous example, so we can express constraints as the linear equations

$$\begin{aligned}
E_{\mathcal{X}}(x_0, x_1) &\iff x_0 + x_1 = b; \\
E_{\mathcal{X}}(x_1, x_2) &\iff x_1 + x_2 = b.
\end{aligned} \tag{24}$$

This system has two solutions, $S_1 = \{x_0 = x_2 = a, x_1 = b\}$ and $S_2 = \{x_0 = x_2 = b, x_1 = a\}$, and the instance is therefore satisfiable.

3.3 Zhuk's four-cases theorem

Zhuk's algorithm is based on the following theorem:

Theorem 8 ([19]). *If $\mathbb{A}$ is a nontrivial finite idempotent algebra with WNU operation, then at least one of the following is true:*

- $\mathbb{A}$ *has a nontrivial binary absorbing subuniverse,*

- 𝔸 *has a nontrivial centrally absorbing subuniverse,*

- 𝔸 *has a nontrivial PC quotient,*

- 𝔸 *has a nontrivial affine quotient.*

4 Zhuk's algorithm

Here we will briefly sketch the leading ideas of Zhuk's algorithm without any details. All details necessary for the formalization will be given directly in the corresponding subsections. For more information we send the reader to the original paper [19].

In this section we will consider an arbitrary constraint language (since the algorithm is designed for all finite languages). Before running the algorithm, it is necessary to make a slight modification of the constraint language. Suppose we have a finite language Γ' that is preserved by an idempotent WNU operation Ω'. By Lemma 1, Γ' is therefore also preserved by a special WNU operation Ω. Let k' be the maximal arity of the relations in Γ' and denote by Γ the set of all relations of arity at most k' that are preserved by Ω. Hence all *pp*-definable relations of arity at most k' are in Γ, and CSP(Γ') is an instance of CSP(Γ).

The common property of all parts of the algorithm is that any time when it reduces or restricts domains, the algorithm uses recursion.

4.1 Outline of the general part

The key notion of the general part of Zhuk's algorithm is reduction, which is divided into several procedures. Consider a CSP instance of CSP(Γ), $\Theta = (X, D, C)$. In this part, the algorithm gradually reduces different domains until it terminates in the linear case. At every step, it either produces a reduced domain or moves to the other type of reduction, or answers that there is no solution (if some domain is empty after one of the procedures). After outputting any reduced domain, the algorithm runs all from the beginning for the same instance Θ but with a smaller domain D'.

First, the algorithm reduces domains until the instance is cycle-consistent. Then it checks irreducibility: again, if the instance is not irreducible, the algorithm can produce a reduction to some domain. The next step is to check a weaker instance that is produced from the instance by simultaneously replacing all constraints with all weaker constraints: if the solution set to such an instance is not subdirect, then some domain can be reduced.

After these types of consistency, the algorithm checks whether some domains have a nontrivial binary absorbing subuniverse or a nontrivial center. If any of them does, the algorithm reduces the domain to the subuniverse or to the center. Then it checks whether there is a proper congruence on any domain such that its factor algebra is polynomially complete. If there is such a congruence, then the algorithm reduces the domain to an equivalence class of the congruence.

By Theorems 9, 10 and 11, proved by Zhuk in [19], if the reduced instance has no solution, then so does the initial one.

Theorem 9 ([19])**.** *Suppose Θ is a cycle-consistent irreducible CSP instance, and B is a nontrivial binary absorbing subuniverse of D_i. Then Θ has a solution if and only if Θ has a solution with $x_i \in B$.*

Theorem 10 ([19])**.** *Suppose Θ is a cycle-consistent irreducible CSP instance, and B is a nontrivial center of D_i. Then Θ has a solution if and only if Θ has a solution with $x_i \in B$.*

Theorem 11 ([19])**.** *Suppose Θ is a cycle-consistent irreducible CSP instance, there does not exist a nontrivial binary absorbing subuniverse or a nontrivial center on D_j for every j, $(D_i, \Omega)/\sigma_i$ is a polynomially complete algebra, and E is an equivalence class of σ_i. Then Θ has a solution if and only if Θ has a solution with $x_i \in E$.*

Finally, if the algorithm cannot reduce any domain of the CSP instance Θ any further, by Theorem 8 it means that every domain D_i of size greater than 1 has a nontrivial affine quotient. Since we consider the special WNU operation Ω, for every domain D_i there exists a congruence σ_i such that $(D_i, \Omega)/\sigma_i$ is isomorphic to $(\mathbb{Z}_{p_1} \times ... \times \mathbb{Z}_{p_l}, x_1 + ... + x_m)$ for some prime numbers $p_1, ..., p_l$. The algorithm then proceeds with procedures embraced in the linear case.

4.2   Outline of the linear case

The linear case of Zhuk's algorithm is adopted from [19]. Suppose that on every domain D_i there exists the proper minimal linear congruence σ_i such that $(D_i, \Omega)/\sigma_i$ is linear, i.e. isomorphic to $(\mathbb{Z}_{p_1} \times ... \times \mathbb{Z}_{p_l}, x_1 + ... + x_m)$ for some prime numbers $p_1, ..., p_l$, where m is the arity of Ω.

Denote each D_i/σ_i by L_i and define a new CSP instance Θ_L with domains $L_1, ..., L_n$ as follows: to every constraint $(x_{i_1}, ..., x_{i_s}; R) \in \Theta$ assign a constraint $(x'_{i_1}, ..., x'_{i_s}; R')$, where $R' \in L_{i_1} \times ... \times L_{i_s}$ and a tuple of blocks of congruences $(E_1, ..., E_s) \in R' \iff (E_1 \times ... \times E_s) \cap R \neq \emptyset$. From now we will refer to the instance Θ as the initial instance, and to Θ_L as the factorized one.

Since each $L_i = D_i/\sigma_i$ is isomorphic to some $\mathbb{Z}_{s_1} \times ... \times \mathbb{Z}_{s_l}$, we can define a natural bijective mapping $\psi : \mathbb{Z}_{p_1} \times ... \times \mathbb{Z}_{p_r} \to L_1 \times ... \times L_n$ and assign a variable z_i to every $\mathbb{Z}_{p_i}$. By Theorem 7 every relation on $\mathbb{Z}_{p_1} \times ... \times \mathbb{Z}_{p_r}$ preserved by $\Omega(x_1, ..., x_m) = x_1 + ... + x_m$ is an affine subspace, the instance Θ_L can thus be viewed as a system of linear equations over $z_1, ..., z_r$. Every linear equation is an equation in $\mathbb{Z}_{p_i}$, and only variables ranging over the same field $\mathbb{Z}_{p_i}$ may appear in one equation.

The algorithm compares two sets: the solution set to the initial instance Θ factorized by congruences (let us denote it by S_Θ/Σ) and the solution set to the factorized instance, S_{Θ_L}. It is known that $S_\Theta/\Sigma \subseteq S_{\Theta_L}$. We do not know S_Θ/Σ, but we can efficiently calculate S_{Θ_L} using Gaussian Elimination (since Gaussian Elimination is strongly polynomial [10]). If Θ_L has no solution, then so does the initial instance. If the solution has no independent variables (i.e. there is only one solution and the dimension of the solution set is 0), the algorithm checks whether the initial instance Θ has the solution corresponding to this solution by restricting every domain D_i of Θ to the corresponding congruence blocks and recursively calling the algorithm for these smaller domains. Otherwise, the algorithm arbitrarily chooses independent variables $y_1, ..., y_k$ of the general solution to Θ_L (the dimension of the solution set S_{Θ_L} is k).

The set S_{Θ_L} can be defined as an affine mapping $\phi : \mathbb{Z}_{q_1} \times ... \times \mathbb{Z}_{q_k} \to L_1 \times ... \times L_n$. Thus, any solution to Θ_L can be obtained as $\phi(a_1, ..., a_k)$ for some $(a_1, ..., a_k) \in \mathbb{Z}_{q_1} \times ... \times \mathbb{Z}_{q_k}$.

The algorithm denotes an empty set of linear equations by Eq. The following steps will be repeated until the algorithm either finds a solution or answers that S_Θ/Σ is empty. The idea is to add equations iteratively to the solution set S_{Θ_L} maintaining the property $S_\Theta/\Sigma \subseteq S_{\Theta_L} \cup Eq$. Since the dimension of S_{Θ_L} is k, and at every iteration the algorithm reduces the dimension by at least one, the process will eventually stop.

First of all, the algorithm checks whether Θ has a solution corresponding to $\phi(0, ..., 0)$ by recursively calling the algorithm for smaller domains. If it does, the algorithm stops with a solution, if it does not, it has established the property $S_\Theta/\Sigma \subsetneq S_{\Theta_L}$. Then the
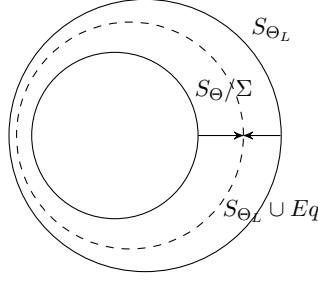
Figure 6: Solution sets.

algorithm starts to decrease the solution set S_{Θ_L}. It always starts with the initial instance Θ, gradually makes it weaker and at every weakening checks whether the solution set to this new weaker instance is equal to S_{Θ_L}.

To make Θ weaker, the algorithm arbitrarily chooses a constraint C and replaces it with all weaker constraints without dummy variables simultaneously. Let us denote this instance by Θ'. To check whether the solution set $S_{\Theta'}/\Sigma$ to Θ' factorized by congruences is equal to S_{Θ_L}, one needs to check whether Θ' has solutions corresponding to $\phi(a_1, ..., a_k)$ for every $(a_1, ..., a_k) \in \mathbb{Z}_{q_1} \times ... \times \mathbb{Z}_{q_k}$ (using recursion for smaller domains). Since $S_{\Theta'}/\Sigma$ and S_{Θ_L} are subuniverses of $L_1 \times ... \times L_n$, it is enough to check the existence of solutions corresponding to $\phi(0, ..., 0)$ and $\phi(0, ..., 1, ..., 0)$ for any position of 1. If the solution set to the weaker instance Θ' does not contain S_{Θ_L}, the algorithm proceeds with weakening the instance Θ' step by step until it cannot make the instance weaker without obtaining $S_{\Theta_L} \subseteq S_{\Theta'}/\Sigma$ (at this point the algorithm checks that whichever constraint it weakens, every solution to Θ_L will be a solution to Θ'). It means that there exists some $(b_1, ..., b_k)$ $\in \mathbb{Z}_{q_1} \times ... \times \mathbb{Z}_{q_k}$ such that Θ' has no solution corresponding to $\phi(b_1, ..., b_k)$. However, if we replace any constraint $C \in \Theta'$ with all weaker constraints simultaneously, then we get an instance that has a solution corresponding to $\phi(a_1, .., a_k)$ for every $(a_1, ..., a_k) \in \mathbb{Z}_{q_1} \times ... \times \mathbb{Z}_{q_k}$.

Finally, the algorithm finds the solution set $S_{\Theta'}/\Sigma$ to the instance Θ' factorized by congruences by finding new equations additional to the set S_{Θ_L}. There are different strategies for linked and non-linked instances Θ'. For linked instance, it is known that $S_{\Theta'}/\Sigma \subsetneq S_{\Theta_L}$ is of codimension 1, so we can find only one equation and add it to S_{Θ_L}. For non-linked instance Θ' we find all equations that describe $S_{\Theta'}/\Sigma$, and then intersect these equations with S_{Θ_L} (see [19]). After new equations are found, the algorithm adds them to the set Eq, solves $S_{\Theta_L} \cup Eq$ using Gaussian Elimination, and runs another iteration.

Remark 1. By Theorem 7, $S_{\Theta_L} \subseteq (\mathbb{Z}_{p_1})^{n_1} \times ... \times (\mathbb{Z}_{p_k})^{n_k}$ is an affine subspace. The solution set S_{Θ}/Σ to the initial instance factorized by congruences is also an affine subspace: the relation that describes it is a subset of S_{Θ_L}, i.e. it is also preserved by Ω. Moreover, when we consider the solution set $S_{\Theta'}/\Sigma$ to the weaker instance Θ' factorized by congruences, it is also an affine subspace since all weaker constraints are in Γ.

5 Soundness of Zhuk's algorithm in a theory of bounded arithmetic

To prove the soundness of Zhuk's algorithm in some theory of bounded arithmetic, it is sufficient to prove that after every step of the algorithm one does not lose all the solutions to the initial instance. Consider any relational structure $\mathcal{A}$ with at most binary

relations and some negative instance $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$ of CSP($\mathcal{A}$), and suppose that there is a homomorphism from $\mathcal{X}$ to $\ddot{\mathcal{A}}$. If the elected theory of bounded arithmetic proves that after every step of the algorithm the new modified instance has solutions only if the previous one does, and the algorithm terminates with no solution, then the theory proves - by its level of bounded induction - that $\mathcal{X}$ is unsatisfiable, and hence that $\neg HOM(\mathcal{X}, \ddot{\mathcal{A}})$ is a tautology.

Consider computation of the algorithm on $(\mathcal{X}, \ddot{\mathcal{A}})$, $W = (W_1, W_2, ..., W_k)$, where:

- $W_1 = (\mathcal{X}, \ddot{\mathcal{A}})$;

- $W_{i+1} = (\mathcal{X}_{i+1}, \ddot{\mathcal{A}}_{i+1})$ is obtained from $W_i = (\mathcal{X}_i, \ddot{\mathcal{A}}_i)$ by one algorithmic step ($\mathcal{X}_{i+1}$ and $\ddot{\mathcal{A}}_{i+1}$ are some modifications of relational structures $\mathcal{X}_i, \ddot{\mathcal{A}}_i$);

- W_k has no solution.

We need to prove, for all types of algorithmic modifications, that if W_i has a solution, then W_{i+1} also has a solution. This will prove that if the algorithm terminates with no solution, then there is no homomorphism from $\mathcal{X}$ to $\ddot{\mathcal{A}}$. Note that it is unnecessary to prove the opposite direction when considering soundness. Moreover, neither it is necessary to prove that the algorithm is well-defined. The transcription of the algorithm's computation can include all auxiliary necessary information.

In the formalization of the algorithm we will incorporate some modifications and adjustments suggested by Zhuk in his later paper [20]. We also sometimes will omit some intermediate steps and other technicalities not affecting the result. We will explicitly highlight all points that distinguish this version of the algorithm from the original one.

In the paper we shall prove the soundness of Zhuk's algorithm in a new theory of bounded arithmetic, namely V^1 augmented with three universal algebra axioms, which will be defined in the next section.

5.1 Defining a new theory of bounded arithmetic

In this section we will define a new theory of bounded arithmetic that will extend the theory V^1. Before moving to this section, we recommend that the reader recall subsections 2.4 and 2.5.

5.1.1 Arrangements before the run of the algorithm

We will consider only relational structures that contain at most binary relations, and algebras corresponding to them, see Theorem 5. The algorithm works for any finite algebra having a WNU term and uses the fact that this term and all the properties of the algebra are known in advance. From here on out we fix algebra $\mathbb{A} = (A, \Omega)$ and suppose that the only basic operation on $\mathbb{A}$ is idempotent special WNU operation Ω. Algebras with richer signatures can be treated in a similar way, extending all conditions imposed on Ω to other (know in advance) basic operations.

Since at the beginning Zhuk's algorithm adds to a constraint language Γ all relations preserved by Ω, of the arity up to the maximal arity of relations in Γ, we will consider the finite set of *all* relations of arity at most 2, invariant under Ω, which we know in advance. Let us denote this set by $\Gamma_{\mathcal{A}}$, and the relational structure by $\mathcal{A} = (A, \Gamma_{\mathcal{A}})$. Any time when in formulas we claim something about this set, it means that we claim this about each relation in this set.

A new theory of bounded arithmetic will extend the theory V^1. Before we introduce this theory, we need to define in V^1 notions from different areas of mathematics.

5.1.2 Encoding relational structure

We encode the finite universe A of size l by the set $A, \forall i < l, A(i)$, and $\Gamma_\mathcal{A}$ as a pair of sets $(\Gamma_\mathcal{A}^1, \Gamma_\mathcal{A}^2)$ where $\Gamma_\mathcal{A}^1$ is the set which encodes all unary relations from $\Gamma_\mathcal{A}$, and $\Gamma_\mathcal{A}^2$ encodes binary relations,

$$\Gamma_\mathcal{A}^1(j, a) \iff D_j^1(a) \text{ and } \Gamma_\mathcal{A}^2(i, a, b) \iff E_i^2(a, b).$$

Note that in the list of $\Gamma_\mathcal{A}$ there are all possible subalgebras of $\mathbb{A}$ (i.e. all possible domains and strong subsets), and all possible *pp*-definitions constructed from unary and binary relations preserved by Ω. When consider a subset D of A, we will denote by $\Gamma_\mathcal{D}$ the set of unary and binary relations from $\Gamma_\mathcal{A}$ restricted to the set D.

Among binary relations $\Gamma_\mathcal{A}^2$ there are all congruences on $\mathbb{A}$ and on all its subalgebras. Let us denote this set by $\Sigma_\mathcal{A}$. Since for any subalgebra $\mathbb{D}$ any congruence of $\mathbb{A}$ is also a congruence of $\mathbb{D}$, the formula

$$D_j^1(a) \wedge D_j^1(b) \wedge \Sigma_\mathcal{A}(i, a, b)$$

defines a congruence on some $\mathbb{D}$. The number of all possible congruences on $\mathbb{A}$ is bounded by $2^{|A|^2}$.

5.1.3 Encoding special WNU operation and polymorphism

We can define a special WNU operation of fixed arity m on some set A in the theory V^1 in several steps. We say that a set F is an m-ary *operation* $F : A^m \to A$ on a set A if it satisfies the relation

$$OP_m(F, A) \iff \forall x_0, ..., x_{m-1} \in A, \exists y \in A, \ F(x_0, ..., x_{m-1}) = y \wedge \tag{25}$$
$$\wedge \forall y_1, y_2 \in A \ (F(x_0, ..., x_{m-1}) = y_1 \wedge F(x_0, ..., x_{m-1}) = y_2 \to y_1 = y_2).$$

An idempotent operation F is defined straightforwardly:

$$IDM_m(F, A) \iff OP_m(F, A) \wedge \forall a \in A \ F(a, a, ..., a) = a. \tag{26}$$

We say that a set Ω is a WNU operation of arity m on the set A if it satisfies the relation

$$wNU_m(\Omega, A) \iff OP_m(\Omega, A) \wedge \forall a, b \in A, \exists c \in A, \forall x_0, ..., x_{m-1} \in A$$
$$\bigwedge_{t < m} (x_t = a \wedge \forall j \neq t < m, \ x_j = b \to \Omega(x_0, ..., x_{m-1}) = c). \tag{27}$$

A special WNU operation is defined as follows:

$$SwNU_m(\Omega, A) \iff wNU_m(\Omega, A) \wedge IDM_m(\Omega, A)$$
$$\forall a, b \in A, \exists c \in A, \ \Omega(a, ..., a, b) = c \wedge \Omega(a, ..., a, c) = c. \tag{28}$$

Since we work with relations of arity at most 2, we will define polymorphisms only for relations of this arity. We say that a set F is an operation of arity m on the set A that preserves 2-ary relation R on A if it satisfies the following relation

$$Pol_{m,2}(F, A, R) \iff OP_m(F, A) \wedge \forall a_1^0, ..., a_1^{m-1}, a_2^0, ..., a_2^{m-1} \in A,$$
$$\forall b_1, b_2 \in A, \ R(a_1^0, a_2^0) \wedge ... \wedge R(a_1^{m-1}, a_2^{m-1}) \wedge \tag{29}$$
$$\wedge F(a_1^0, ..., a_1^{m-1}) = b_1 \wedge F(a_2^0, ..., a_2^{m-1}) = b_2 \to R(b_1, b_2).$$

Finally, operation F preserves 1-ary relation R on A if

$$
\begin{aligned}
Pol_{m,1}(F, A, R) \iff & OP_m(F, A) \wedge \forall a^0, a^1, ..., a^{m-1} \in A, \\
& \forall b \in A \ R(a^0) \wedge ... \wedge R(a^{m-1}) \wedge \\
& \wedge F(a^0, ..., a^{m-1}) = b \to R(b).
\end{aligned}
\tag{30}
$$

We will omit the second index i in $Pol_{m,i}$ when we refer to the whole set of relations $\Gamma_{\mathcal{A}}$.

5.1.4 Encoding notions from universal algebra

A finite algebra with special WNU operation of size l is a pair of sets $\mathbb{A} = (A, \Omega)$, where $|A| = l$, $A(i)$ for every i, and Ω is a $((m+1)l)^{2^{m+1}}$ set representing a special WNU operation on A. We will call this pair a Taylor algebra and denote it by $TA(A, \Omega)$. From here on out under algebra we mean Taylor algebra. We say that $\mathbb{B} = (B, \Omega)$ is a subalgebra of algebra $\mathbb{A}$ if

$$
subTA(\mathbb{B}, \mathbb{A}) \iff |B| = |A| \wedge \forall i < l, B(i) \to A(i) \wedge SwNU(\Omega, B).
\tag{31}
$$

Note that condition $SwNU(\Omega, B)$ ensures that B is closed under operation Ω. The difference between fixed algebra $\mathbb{A}$ and all its subalgebras and factor algebras is that the size of all these objects is bounded by l, but since it is not necessary that for all $i < l, B(i)$, we will measure their size by census function, $\#B(l)$. We say that a pair of sets $\mathbb{B} = (B, \Omega)$ is a direct product of k algebras $\mathbb{A}_0 = (A_0, \Omega_0), ..., \mathbb{A}_{k-1} = (A_{k-1}, \Omega_{k-1})$ of the same type if

$$
\begin{aligned}
DP_{m,k}(B, \Omega, A_0, \Omega_0, ..., A_{k-1}, \Omega_{k-1}) \iff & \forall a_0 \in A_0, ..., a_{k-1} \in A_{k-1}, \\
B(a_0, ..., a_{k-1}) \wedge \forall a_0^1, a_0^2, ..., a_0^m \in A_0, ..., a_{k-1}^1, & a_{k-1}^2, ..., a_{k-1}^m \in A_{k-1} \\
\Omega(a_0^1, a_0^2, ..., a_0^m, ..., a_{k-1}^1, a_{k-1}^2, ..., a_{k-1}^m) = & \\
= (\Omega_0(a_0^1, a_0^2, ..., a_0^m), ..., \Omega_{k-1}(a_{k-1}^1, & a_{k-1}^2, ..., a_{k-1}^m)).
\end{aligned}
\tag{32}
$$

We will denote (B, Ω) by $(A_0 \times ... \times A_{k-1}, \Omega)$. A subdirect k-ary relation R on $A_0 \times ... \times A_{k-1}$ is encoded as follows:

$$
\begin{aligned}
subDR_k(R, A_0, ..., A_{k-1}) \iff & \bigwedge_{i<k} \forall a_i \in A_i, \exists a_0 \in A_0, ..., a_{i-1} \in A_{i-1}, \\
& a_{i+1} \in A_{i+1}, ..., a_{k-1} \in A_{k-1}, R(a_0, ..., a_{i-1}, a_i, a_{i+1}, ..., a_{k-1}.)
\end{aligned}
\tag{33}
$$

We say that a set $\sigma < l^2$ is a congruence relation on the algebra $\mathbb{A} = (A, \Omega)$ if it satisfies the following relation

$$
\begin{aligned}
Cong_m(A, \Omega, \sigma) \iff & Pol_{m,2}(\Omega, A, \sigma) \wedge \\
\forall a \in A, \sigma(a, a) \wedge \forall a, b \in A, (\sigma(a, b) & \leftrightarrow \sigma(b, a)) \wedge \\
(\forall a, b, c \in A, \sigma(a, b) \wedge \sigma(b, c) & \to \sigma(a, c)).
\end{aligned}
\tag{34}
$$

Condition $Pol_{m,2}(\Omega, A, \sigma)$ ensures that σ is from $Inv(Pol(\Gamma_{\mathcal{A}}))$. Recall that all congruences on $\mathbb{A}$ are listed in $\Sigma_{\mathcal{A}}$. If we additionally require that

$$
(\exists x, y \in A \neg \sigma(x, y)) \wedge (\exists x \neq y \in A \ \sigma(x, y)),
\tag{35}
$$

the congruence σ will be proper. A maximal congruence (a congruence over which there is no other congruences except the full binary relation ∇) can be defined as follows:

$$
\begin{aligned}
maxCong_m(A, \Omega, \sigma) \iff & Cong_m(A, \Omega, \sigma) \wedge \exists a, b \in A, \neg \sigma(a, b) \wedge \\
\wedge [\forall \sigma' < \langle l, l \rangle, (Cong_m(A, \Omega, \sigma') \wedge \exists a, b & \in A, \neg \sigma'(a, b)) \to \\
\to \exists a, b \in A, \sigma(a, b) & \wedge \neg \sigma'(a, b)].
\end{aligned}
\tag{36}
$$

Note that this is a $\Pi_1^{1,b}$-formula. A factor set is the set of all equivalence classes under the congruence σ and it will be denoted by A/σ. We can represent each block of σ by its minimal element (it exists by the Minimal principle). Therefore, we think of the factorized object A/σ as of a set of numbers as well:

$$FS_m(A/\sigma, A, \Omega, \sigma) \iff Cong_m(A, \Omega, \sigma) \wedge$$
$$\forall a, b \in A, (\sigma(a,b) \wedge (a < b) \rightarrow \neg A/\sigma(b)) \tag{37}$$
$$\wedge (\forall a \in A(\forall a' \in A, \sigma(a, a') \rightarrow a \leq a') \rightarrow A/\sigma(a)).$$

We say that a is a represent of the class a/σ (where a/σ is just a notation, it is any element of A) if

$$Rep_m(a, a/\sigma, A/\sigma, A, \Omega, \sigma) \iff FS_m(A/\sigma, A, \Omega, \sigma) \wedge$$
$$\sigma(a, a/\sigma) \wedge A/\sigma(a). \tag{38}$$

Finally, we can define the factor algebra $\mathbb{A}/\sigma = (A/\sigma, \Omega/\sigma)$:

$$FA_m(A/\sigma, \Omega/\sigma, A, \Omega, \sigma) \iff FS_m(A/\sigma, A, \Omega, \sigma) \wedge$$
$$\wedge (\forall a_1, ..., a_m, c \in A, \forall a_1/\sigma, ..., a_m/\sigma, c/\sigma \in A,$$
$$\Omega(a_1/\sigma, ..., a_m/\sigma) = c/\sigma \wedge Rep_m(c, c/\sigma, A/\sigma, A, \Omega, \sigma) \wedge \tag{39}$$
$$\wedge \bigwedge_{i < m} Rep_m(a_i, a_i/\sigma, A/\sigma, A, \Omega, \sigma) \rightarrow \Omega/\sigma(a_1, ..., a_m) = c).$$

Thus, we define the operation Ω/σ on minimal elements of the congruence classes.

5.1.5 Encoding digraphs and CSP properties

We will code a CSP instance on relational structures with at most binary relations in the following way.

Definition 11. A *directed input graph* is a pair $\mathcal{X} = (V_{\mathcal{X}}, E_{\mathcal{X}})$ with $V_{\mathcal{X}}(i)$ for all $i < V_{\mathcal{X}} = n$ and $E_{\mathcal{X}}(i, j)$ being a binary relation on $V_{\mathcal{X}}$ (there is an edge from i to j). A *target digraph with domains* is an $(n+2)$-tuple of sets $\ddot{\mathcal{A}} = (V_{\ddot{\mathcal{A}}}, E_{\ddot{\mathcal{A}}}, D_0, ..., D_{n-1})$, where

- $V_{\ddot{\mathcal{A}}} < \langle n, l \rangle$ is the set corresponding to the superdomain,

- $\forall i < n$, $D_i < l$ is the subset of length l corresponding to the domain of variable x_i,

- $V_{\ddot{\mathcal{A}}}(i, a) \iff D_i(a)$,

- $E_{\ddot{\mathcal{A}}} < \langle \langle n, l \rangle, \langle n, l \rangle \rangle$ is the set encoding relations $E_{\ddot{\mathcal{A}}}^{ij}(a, b)$ (there is an edge (a, b) between D_i and D_j):

$$E_{\ddot{\mathcal{A}}}(u, v) \rightarrow \exists i, j < n \, \exists a, b < l \; u = \langle i, a \rangle \wedge v = \langle j, b \rangle \wedge$$
$$D_i(a) \wedge D_j(b). \tag{40}$$

Sometimes we will use the notation $E_{\ddot{\mathcal{A}}}^{ij}(a, b)$ instead of $E_{\ddot{\mathcal{A}}}(\langle i, a \rangle, \langle j, b \rangle)$ for brevity sake. We will denote a pair of sets $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$, satisfying all above conditions, by DG(Θ), and will call Θ an instance. This representation will allow us to construct a homomorphism from $\mathcal{X}$ to $\ddot{\mathcal{A}}$ with respect to different relations $E_{\ddot{\mathcal{A}}}^{ij}$ and different domains for all vertices $x_1, ..., x_n$.

Definition 12 (Homomorphism from digraph $\mathcal{X}$ to digraph with domains $\ddot{\mathcal{A}}$). A map H is a *homomorphism between input digraph* $\mathcal{X} = (V_\mathcal{X}, E_\mathcal{X})$, $V_\mathcal{X} = n$ *and target digraph with domains* $\ddot{\mathcal{A}} = (V_{\ddot{\mathcal{A}}}, E_{\ddot{\mathcal{A}}}, D_0, ..., D_{n-1})$, $V_\mathcal{A} < \langle n, l \rangle$ if H is a homomorphism from $\mathcal{X}$ to $\ddot{\mathcal{A}}$ sending each $i \in V_\mathcal{X}$ to domain D_i in $V_{\ddot{\mathcal{A}}}$. The statement that there exists such an H can be expressed by the $\Sigma_1^{1,b}$-formula

$$
\begin{aligned}
H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}}) \iff & \exists H < \langle n, \langle n, l \rangle \rangle (MAP(V_\mathcal{X}, n, V_{\ddot{\mathcal{A}}}, \langle n, l \rangle, H) \wedge \\
& (\forall i < n, s < \langle n, l \rangle \ H(i) = s \rightarrow \exists a < l, s = \langle i, a \rangle \wedge D_i(a)) \wedge \\
& \forall i_1, i_2 < n, \forall j_1, j_2 < \langle n, l \rangle \\
& (E_\mathcal{X}(i_1, i_2) \wedge H(i_1) = j_1 \wedge H(i_2) = j_2 \rightarrow E_{\ddot{\mathcal{A}}}(j_1, j_2)).
\end{aligned}
$$

(41)

Besides a homomorphism between two digraphs of different types, we will also need a classical homomorphism between digraphs of the same type. The existence of such a homomorphism between digraphs $\mathcal{G}$ and $\mathcal{H}$ with $V_\mathcal{G} < n$, $V_\mathcal{G} < m$ can be expressed by the following $\Sigma_1^{1,b}$-formula

$$
\begin{aligned}
HOM(\mathcal{G}, \mathcal{H}) \iff & \exists H < \langle n, m \rangle (MAP(V_\mathcal{G}, n, V_\mathcal{H}, m, H) \wedge \\
& \forall i_1, i_2 < n, \forall j_1, j_2 < m \\
& (E_\mathcal{G}(i_1, i_2) \wedge H(i_1) = j_1 \wedge H(i_2) = j_2 \rightarrow E_\mathcal{H}(j_1, j_2))).
\end{aligned}
$$

(42)

Notation 4. *Sometimes we will write* $\exists H < \langle n, m \rangle, HOM(\mathcal{G}, \mathcal{H}, H)$ *and* $\exists H < \langle n, \langle n, l \rangle \rangle$, $H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}}, H)$ *to omit repetitions.*

For an instance $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$ we call an instance $\Theta' = (\mathcal{X}', \ddot{\mathcal{A}})$ a subinstance of Θ if

$$
\begin{aligned}
subInst(\mathcal{X}', \mathcal{X}) \iff & subS(V_{\mathcal{X}'}, V_\mathcal{X}) \wedge subS(E_{\mathcal{X}'}, E_\mathcal{X}) \wedge \\
& (E_{\mathcal{X}'}(x_1, x_2) \rightarrow x_1, x_2 \in V_{\mathcal{X}'}).
\end{aligned}
$$

(43)

That is, the target digraph with domains $\ddot{\mathcal{A}}$ does not change, the set of vertices $V_{\mathcal{X}'}$ is a subset of $V_\mathcal{X}$, and the set of constraints $E_{\mathcal{X}'}$ is a subset of $E_\mathcal{X}$ defined only on $V_{\mathcal{X}'}$.

We need to encode three properties of a CSP instance: cycle-consistency, being a linked instance, and irreducibility. In order to certify the quantification complexity of the formulas, we will introduce them explicitly. Recall that we refer to any path or cycle with the edges not necessarily directed in the same direction as an undirected path or cycle. We say that a digraph $\mathcal{C}_t = (V_{\mathcal{C}_t}, E_{\mathcal{C}_t})$ with $V_{\mathcal{C}_t} = \{0, 1, ..., t-1\}$ is an *undirected cycle of length t* if it satisfies the following $\Sigma_0^{1,b}$-definable relation

$$
\begin{aligned}
CYCLE(\mathcal{C}_t) \iff & (E_{\mathcal{C}_t}(0, t-1) \vee E_{\mathcal{C}_t}(t-1, 0)) \wedge \\
& \forall i < (t-1) \, (E_{\mathcal{C}_t}(i, i+1) \vee E_{\mathcal{C}_t}(i+1, i)) \wedge \\
& \forall i, j < (t-1)(j \neq i+1 \rightarrow (\neg E_{\mathcal{C}_t}(i, j) \wedge \neg E_{\mathcal{C}_t}(j, i)).
\end{aligned}
$$

(44)

We will define cycle-consistency through two homomorphisms.

Definition 13 (Cycle-consistent instance). An instance $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$ with $V_\mathcal{X} = n$, $V_{\ddot{\mathcal{A}}} < \langle n, l \rangle$ is 1-consistent if it satisfies the following $\Sigma_0^{1,b}$-definable relation

$$
\begin{aligned}
1C(\mathcal{X}, \ddot{\mathcal{A}}) \iff & \forall i < n, \forall a \in D_i, \forall j < n, \\
& (E_\mathcal{X}(i, j) \rightarrow \exists b \in D_j, E_{\ddot{\mathcal{A}}}^{ij}(a, b)) \wedge (E_\mathcal{X}(j, i) \rightarrow \exists b \in D_j, E_{\ddot{\mathcal{A}}}^{ji}(b, a)).
\end{aligned}
$$

(45)

The instance $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$ is cycle-consistent if it is 1-consistent and any undirected cycle $\mathcal{C}_t$ that can be homomorphically mapped into $\mathcal{X}$ with $H(0) = x_k$ can be homomorphically

mapped into $\ddot{\mathcal{A}}$ for any $a \in D_k$. Cycle-consistency is expressed by the following $\Pi_2^{1,b}$-formula

$$
\begin{aligned}
CCInst(\mathcal{X}, \ddot{\mathcal{A}}) &\iff 1C(\mathcal{X}, \ddot{\mathcal{A}}) \wedge \forall k < n, \forall a \in D_k, \forall t < n, \forall V_{\mathcal{C}_t} = t, \\
&\forall E_{\mathcal{C}_t} \le 4t^2, \forall H < \langle t, n \rangle, [CYCLE(V_{\mathcal{C}_t}, E_{\mathcal{C}_t}) \wedge HOM(\mathcal{C}_t, \mathcal{X}, H) \wedge H(0, k) \\
&\to \exists H' < \langle t, \langle t, l \rangle \rangle, H\ddot{O}M(\mathcal{C}_t, \ddot{\mathcal{A}}, H') \wedge \\
&\wedge \forall i < n, j < t\, (H(j) = i \to \exists b \in D_i,\, H'(j) = \langle i, b \rangle) \wedge H'(0) = \langle k, a \rangle].
\end{aligned}
$$

(46)

Note that for any cycle-consistent instance $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$, any its subinstance $\Theta' = (\mathcal{X}', \ddot{\mathcal{A}})$ is also cycle-consistent. For any $i, j \in X'$ the constraint relations D_i, D_j, $E_{\ddot{\mathcal{A}}}^{ij}$ remain the same. We have just removed some vertices from $\mathcal{X}$ and have removed some edges from $E_{\mathcal{X}}$. This does not affect the cycle-consistency property: for any $i \in X'$, any $a \in D_i$, any *existing* in Θ' path starting and ending in i must connect a and a.

We say that a digraph $\mathcal{P}_t = (V_{\mathcal{P}_t}, E_{\mathcal{P}_t})$ with $V_{\mathcal{P}_t} = \{0, 1, ..., t\}$ is an *undirected path of length t* if it satisfies the $\Sigma_0^{1,b}$-definable relation

$$
\begin{aligned}
PATH(\mathcal{P}_t) &\longleftrightarrow \forall i < t\, (E_{\mathcal{P}_t}(i, i+1) \vee E_{\mathcal{P}_t}(i+1, i)) \wedge \\
&\forall i < t, j \le t(j \ne i+1 \to (\neg E_{\mathcal{C}_t}(i, j) \wedge \neg E_{\mathcal{C}_t}(j, i)).
\end{aligned}
$$

(47)

For any two paths $\mathcal{P}_t$ and $\mathcal{P}_m$ of length t and m we will define the following notions. The reverse path $\mathcal{P}_t^{-1}$ is defined as:

$$
\begin{aligned}
V_{\mathcal{P}_t} = V_{\mathcal{P}_t^{-1}} = (t+1) \wedge \forall i < t, \\
E_{\mathcal{P}_t^{-1}}(i, i+1) \leftrightarrow E_{\mathcal{P}_t}(t-i, t-(i+1)) \wedge E_{\mathcal{P}_t^{-1}}(i+1, i) \leftrightarrow E_{\mathcal{P}_t}(t-(i+1), t-i).
\end{aligned}
$$

(48)

The glued path $\mathcal{P}_t \circ \mathcal{P}_m$ is defined as:

$$
\begin{aligned}
V_{\mathcal{P}_t \circ \mathcal{P}_m} = (t+m+1) \wedge \\
\wedge \forall i < t,\, E_{\mathcal{P}_t \circ \mathcal{P}_m}(i, i+1) \leftrightarrow E_{\mathcal{P}_t}(i, i+1) \wedge E_{\mathcal{P}_t \circ \mathcal{P}_m}(i+1, i) \leftrightarrow E_{\mathcal{P}_t}(i+1, i) \wedge \\
\wedge \forall t \le j < (t+m), \\
E_{\mathcal{P}_t \circ \mathcal{P}_m}(j, j+1) \leftrightarrow E_{\mathcal{P}_m}(j-t, j+1-t)) \wedge \\
\wedge E_{\mathcal{P}_t \circ \mathcal{P}_m}(j+1, j) \leftrightarrow E_{\mathcal{P}_m}(j+1-t, j-t)).
\end{aligned}
$$

(49)

We say that there is a path from i to j in the input digraph $\mathcal{X}$ if there exists a path $\mathcal{P}_t$ of some length t that can be homomorphically mapped to $\mathcal{X}$ such that $H(0) = i$ and $H(t) = j$:

$$
\begin{aligned}
Path(i, j, \mathcal{X}) &\iff \exists t < n, \exists V_{\mathcal{P}_t} = t, \exists E_{\mathcal{P}_t} \le 4t^2,\, PATH(V_{\mathcal{P}_t}, E_{\mathcal{P}_t}) \wedge \\
&\wedge \exists H \le \langle t, n \rangle, HOM(\mathcal{P}_t, \mathcal{X}, H) \wedge (H(0, i) \wedge H(t, j)).
\end{aligned}
$$

(50)

We say that the path $\mathcal{P}_t$ connects i and j. Also, we can encode what it means to be linked for two elements $a \in D_i, b \in D_j$. In words, there must exist a path $\mathcal{P}_t$ of some length t connecting i, j with homomorphism H such that there exists a homomorphism H' from $\mathcal{P}_t$ to $\ddot{\mathcal{A}}$ sending 0 to $\langle i, a \rangle$ and t to $\langle j, b \rangle$, and for every element $p < t$, $H(p) = k$ implies that $H(p) = \langle k, c \rangle$ for some $c \in D_k$. We can express it by the $\Sigma_1^{1,b}$-formula

$$
\begin{aligned}
Linked(a, b, i, j, \Theta) &\iff \exists t < nl, \exists V_{\mathcal{P}_t} = t, \exists E_{\mathcal{P}_t} \le 4t^2, \\
\exists H \le \langle t, n \rangle,\, PATH(V_{\mathcal{P}_t}, E_{\mathcal{P}_t}) &\wedge HOM(\mathcal{P}_t, \mathcal{X}, H) \wedge (H(0, i) \wedge H(t, j)) \wedge \\
&\wedge \exists H' \le \langle t, \langle t, l \rangle \rangle, H\ddot{O}M(\mathcal{P}_t, \ddot{\mathcal{A}}, H') \wedge \\
&\wedge (\forall k < n, p < t,\, (H(p, k) \to \exists c \in D_k,\, H'(p) = \langle k, c \rangle)) \\
&\wedge H'(0) = \langle i, a \rangle \wedge H'(t) = \langle j, b \rangle.
\end{aligned}
$$

(51)

25

Notation 5. *Sometimes we will write* $\exists \mathcal{P}_t < \langle n, 4n^2 \rangle, Path(i, j, \mathcal{X}, \mathcal{P}_t)$ *and* $\exists \mathcal{P}_t < \langle nl, 4(nl)^2 \rangle, Linked(a, b, i, j, \Theta, \mathcal{P}_t)$ *to omit repetitions.*

Definition 14 (Linked instance). We say that an instance $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$ with $V_{\mathcal{X}} = n$, $V_{\ddot{\mathcal{A}}} < \langle n, l \rangle$ is linked if it satisfies the following $\Sigma_1^{1,b}$-relation

$$LinkedInst(\mathcal{X}, \ddot{\mathcal{A}}) \iff \forall i < n, \forall a, b \in D_i, Linked(a, b, i, i, \Theta). \tag{52}$$

To define irreducibility we need to encode a fragmented instance and a subdirect solution set.

Definition 15 (Fragmented instance). A fragmented instance is an instance whose input digraph $\mathcal{X}$ is not connected. For an instance $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$ with $V_{\mathcal{X}} = n$ we define this by the following $\Sigma_1^{1,b}$-definable relation, where PSS encodes a proper subset.

$$FragmInst(\mathcal{X}, \ddot{\mathcal{A}}) \iff \exists V_{\mathcal{X}}^1, \exists V_{\mathcal{X}}^2, V_{\mathcal{X}}^1 = V_{\mathcal{X}}^2 = n \wedge$$
$$\wedge PsubS(V_{\mathcal{X}}^1, V_{\mathcal{X}}) \wedge PsubS(V_{\mathcal{X}}^2, V_{\mathcal{X}}) \wedge (\forall i < n, V_{\mathcal{X}}^1(i) \leftrightarrow \neg V_{\mathcal{X}}^2(i)) \wedge \tag{53}$$
$$\wedge \forall i \in V_{\mathcal{X}}^1, \forall j \in V_{\mathcal{X}}^2, \neg E_{\mathcal{X}}(i, j) \wedge \neg E_{\mathcal{X}}(j, i).$$

We say that the instance $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$ has a subdirect solution set if there is a solution to the instance for all $a \in D_i$, $i \in \{0, ..., n-1\}$. It can be expressed by the $\Sigma_1^{1,b}$-formula

$$subDSSInst(\mathcal{X}, \ddot{\mathcal{A}}) \iff \forall i < n \forall a \in D_i, \exists H' < \langle n, \langle n, l \rangle \rangle$$
$$H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}}, H) \wedge H(i) = \langle i, a \rangle. \tag{54}$$

Now we are ready to define irreducibility.

Definition 16 (Irreducible instance). We say that an instance $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$ with $V_{\mathcal{X}} = n$, $V_{\ddot{\mathcal{A}}} < \langle n, l \rangle$ is irreducible if any its subinstance is fragmented, or linked, or its solution set is subdirect. To express it we use the $\Pi_2^{1,b}$-formula

$$IRDInst(\mathcal{X}, \ddot{\mathcal{A}}) \iff \forall \mathcal{X}' = (V_{\mathcal{X}'}, E_{\mathcal{X}'}), \forall V_{\mathcal{X}'} = n, \forall E_{\mathcal{X}'} < 4n^2,$$
$$(subInst(\mathcal{X}', \mathcal{X}) \to \tag{55}$$
$$\to FragmInst(\mathcal{X}', \ddot{\mathcal{A}}) \vee LinkedInst(\mathcal{X}', \ddot{\mathcal{A}}) \vee subDSSInst(\mathcal{X}', \ddot{\mathcal{A}})).$$

Finally, we will introduce the relation indicating that $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$ is an instance of $CSP(\Gamma_{\mathcal{A}})$ for the relational structure $\mathcal{A} = (A, \Gamma_{\mathcal{A}})$. Since $\Gamma_{\mathcal{A}}$ contains at most binary relations and is closed under *pp*-definition, we indeed can identify all constraints posed on variables x_i, x_j with two unary relations (domains D_i, D_j) and one binary relation $E_{\ddot{\mathcal{A}}}^{ij}$ from the list.

Definition 17. A pair of sets $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$ is a CSP instance over constraint language $\Gamma_{\mathcal{A}}$ on A of size l if the following $\Sigma_0^{1,b}$-relation is true.

$$Inst(\Theta, \Gamma_{\mathcal{A}}) \iff DG(\Theta) \wedge \forall i < n, |D_i| = l \wedge$$
$$\wedge \forall i, j < n, a, b < l, \exists s < |\Gamma_{\mathcal{A}}|, E_{\ddot{\mathcal{A}}}(\langle i, a \rangle, \langle j, b \rangle) \leftrightarrow \Gamma_{\mathcal{A}}^2(s, a, b) \wedge \tag{56}$$
$$\wedge \forall i < n, a < l, \exists s < |\Gamma_{\mathcal{A}}|, D_i(a) \leftrightarrow \Gamma_{\mathcal{A}}^1(s, a).$$

5.2 Universal algebra axiom schemes

In this subsection we will encode absorbing and central subuniverses and polynomially complete algebras in V^1, and formulate three universal algebra axioms reflecting the "only if" implications of Theorems 9, 10 and 11 (for the soundness we do not need the "if" implication). For this subsection we will consider CSP instances alongside the corresponding algebras and suppose that any algebra is finite and has a special WNU term.

26

5.2.1 Binary absorption axiom scheme

Consider any algebra $\mathbb{A} = (A, \Omega)$ and its subalgebra $\mathbb{B} = (B, \Omega)$, where Ω is m-ary basic operation. Suppose that the corresponding relational structure to $\mathbb{A}$ is $\mathcal{A} = (A, \Gamma_{\mathcal{A}})$, where $\Gamma_{\mathcal{A}}$ is at most binary part of a relational clone. Due to Galois correspondence, $Clone(\Omega) = Pol(\Gamma_{\mathcal{A}})$. Thus, for any binary term operation T over A the condition $T \in Clone(\Omega)$ can be encoded as:

$$T \in Clone(\Omega) \iff Pol_2(T, A, \Gamma_{\mathcal{A}}). \tag{57}$$

For any three sets A, B, T the following $\Sigma_0^{1,b}$-definable relation indicates that the subset B absorbs A with binary operation T:

$$
\begin{aligned}
BAsubS(B, A, T) \iff\ & subS(B, A) \wedge \forall a \in A, \forall b \in B, \exists c_1, c_2 \in B, \\
& T(a, b) = c_1 \wedge T(b, a) = c_2.
\end{aligned} \tag{58}
$$

We will formalize the "only if" implication of Theorem 9 in the theory V^1 as Binary absorption axioms, BA-axioms. For any algebra $\mathbb{A} = (A, \Omega)$ corresponding to the constraint language $\Gamma_{\mathcal{A}}$ of a CSP instance, it is enough to consider only finitely many axioms since there are finitely many subalgebras $\mathbb{D}$ of $\mathbb{A}$ and finitely many strong subsets B of $\mathbb{D}$).

Definition 18 (BA$_{\mathcal{A}}$-axioms). For any constraint language $\Gamma_{\mathcal{A}}$ over set A of size l, fixed algebra $\mathbb{A} = (A, \Omega)$ with Ω being an m-ary special WNU operation, and finitely many subuniverses D of $\mathbb{A}$ and binary absorbing subuniverses B of $\mathbb{D}$ the *binary absorption axiom scheme* is denoted by BA$_{\mathcal{A}}$-axioms and consists of the finitely many formulas of the following form

$$
\begin{aligned}
BA_{\mathcal{A}, B, D} =_{def}\ & \forall \mathcal{X} = (V_{\mathcal{X}}, E_{\mathcal{X}}), \forall \ddot{\mathcal{A}} = (V_{\ddot{\mathcal{A}}}, E_{\ddot{\mathcal{A}}}, D_0, ..., D_{n-1}), \\
& (PsubS(B, D) \wedge SwNU_m(\Omega, D) \wedge SwNU_m(\Omega, B) \wedge \\
& \wedge \exists T < (3l)^{2^3}, Pol_2(T, D, \Gamma_{\mathcal{A}}) \wedge BAsubS(B, D, T) \wedge \\
& \wedge Inst(\Theta, \Gamma_{\mathcal{A}}) \wedge CCInst(\mathcal{X}, \ddot{\mathcal{A}}) \wedge IRDInst(\mathcal{X}, \ddot{\mathcal{A}}) \wedge \\
& \exists i < n, D_i = D \wedge \\
& H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}})) \rightarrow H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}} = (V_{\ddot{\mathcal{A}}}, E_{\ddot{\mathcal{A}}}, D_0, ..., B, ..., D_{n-1})).
\end{aligned} \tag{59}
$$

Variables here are an input digraph $\mathcal{X}$ with $V_{\mathcal{X}} = n$ and a target digraph with domains $\ddot{\mathcal{A}}$, Θ stands for $(\mathcal{X}, \ddot{\mathcal{A}})$. The second line of the formula ensures that B is a proper subset of D and both B and D are closed under Ω (relation $SwNU_m$), i.e. both are subuniverses. The third line claims that there exists a binary operation T defined on the subuniverse D and compatible with all relations from $\Gamma_{\mathcal{A}}$ such that B absorbs D with T. The fourth line says that Θ is a CSP instance over constraint language $\Gamma_{\mathcal{A}}$, and this instance is cycle-consistent and irreducible. Finally, the rest of the formula says that if D coincides with a domain D_i for some variable i, all the above-mentioned conditions hold and there is a solution to the instance Θ, then there is a solution to the instance Θ with D_i restricted to B.

In strict form (with all string quantifiers occurring in front) and after regrouping them in such a way that all universal quantifiers will precede existential ones, we will eventually get the universal closure of $\Sigma_2^{1,b}$-formula.

5.2.2 Central subuniverse axiom scheme

We will formalize the "only if" implication of Theorem 10 not for a center, but for a central subuniverse. Recall that a central subuniverse has all the good properties of a center, and

we will use it in the algorithm instead of the latter. To define a central subuniverse C of an algebra $\mathbb{A} = (A, \Omega)$ we need to encode a set Sg for the subset $X = \{\{a\} \times C, C \times \{a\}\}$ of A^2 for any $a \in A$. Recall that $Sg(X)$ can be constructed by the closure operator

$$E(X) = X \cup \{\Omega(a_1, ..., a_m) : a_1, ..., a_m \in X\}$$
$$\forall t \geq 0, E^0(X) = X, E^{t+1}(X) = E(E^t(X)). \tag{60}$$

Since $\mathbb{A}$ is finite of size l and $|X| = 2|C|$, we do not need more than $(l^2 - 2|C|)$ applications of the closure operator E since at every application we either add to the set at least one element or after some t, $E^t(X) = E^{t+r}(X)$ for any r. Not to depend on C, let us choose the value l^2. Thus, for any set $X \leq \langle l, l \rangle$, we will iteratively define the following set $E_X^{l^2}$ up to l^2

$$\forall b, c < l, E_X^0(b, c) \iff X(b, c) \wedge$$
$$\wedge \forall 0 < t < l^2, \forall b, c < l, E_X^t(b, c) \iff E_X^{t-1}(b, c) \vee$$
$$\vee \exists b_1, ..., b_m, c_1, ..., c_m \in A, E_X^{t-1}(b_1, c_1) \wedge ... \wedge E_X^{t-1}(b_m, c_m) \wedge \tag{61}$$
$$\wedge \Omega(b_1, ..., b_m) = b \wedge \Omega(c_1, ..., c_m) = c.$$

The existence of this set follows from $\Sigma_1^{1,b}$-induction. A central subuniverse must be an absorbing subuniverse, namely a ternary absorbing subuniverse [20]. For any three sets A, C, S the following $\Pi_1^{1,b}$-definable relation expresses that the subset C of A is central under ternary term operation S.

$$CRsubS(C, A, S) \iff subS(C, A) \wedge \forall c_1, c_2 \in C, \forall a \in A, \exists c_1', c_2', c_3' \in C,$$
$$S(c_1, c_2, a) = c_1' \wedge S(c_1, a, c_2) = c_2' \wedge S(a, c_1, c_2) = c_3' \wedge \tag{62}$$
$$\wedge \forall a \in A \backslash C, \forall X < \langle l, l \rangle, ((X(a, c) \wedge X(c, a) \leftrightarrow c \in C) \rightarrow \neg E_X^{l^2}(a, a)).$$

Definition 19 (CR$_{\mathcal{A}}$-axioms)**.** For any constraint language $\Gamma_{\mathcal{A}}$ over set A of size l, fixed algebra $\mathbb{A} = (A, \Omega)$, with Ω being an m-ary special WNU operation, and finitely many subuniverses D of $\mathbb{A}$ and central subuniverses C of $\mathbb{D}$ we denote the *central subuniverse axiom scheme* by CR$_{\mathcal{A}}$-axioms. The scheme embraces the finitely many formulas of the following form

$$CR_{\mathcal{A},D,C} =_{def} \forall \mathcal{X} = (V_{\mathcal{X}}, E_{\mathcal{X}}), \forall \ddot{\mathcal{A}} = (V_{\ddot{\mathcal{A}}}, E_{\ddot{\mathcal{A}}}, D_0, ..., D_{n-1}),$$
$$(PsubS(C, D) \wedge SwNU_m(\Omega, D) \wedge SwNU_m(\Omega, C) \wedge$$
$$\exists S < (4l)^{2^4}, Pol_3(S, D, \Gamma_{\mathcal{A}}) \wedge CRsubS(C, D, S) \wedge \tag{63}$$
$$\wedge Inst(\Theta, \Gamma_{\mathcal{A}}) \wedge CCInst(\mathcal{X}, \ddot{\mathcal{A}}) \wedge IRDInst(\mathcal{X}, \ddot{\mathcal{A}}) \wedge$$
$$\exists i < n, D_i = D \wedge$$
$$H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}})) \rightarrow H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}} = (V_{\ddot{\mathcal{A}}}, E_{\ddot{\mathcal{A}}}, D_0, ..., C, ..., D_{n-1})).$$

The formula is analogous to BA$_{\mathcal{A}}$-axioms, it is again the universal closure of $\Sigma_2^{1,b}$-formula and the only line that differs is the third one: it claims that there exists a ternary term operation S defined on subuniverse D and compatible with all relations from $\Gamma_{\mathcal{A}}$ such that C is a central subuniverse under S.

5.2.3 Polynomially complete axiom scheme

Theorem 6 claims that a finite algebra is polynomially complete if and only if it has the ternary discriminator as a polynomial operation. Consider an algebra $\mathbb{A} = (A, \Omega)$. The

clone of all polynomials over $\mathbb{A}$, $Polynom(\mathbb{A})$ is defined as the clone generated by Ω and all constants on A, i.e. nullary operations:

$$Polynom(\mathbb{A}) = Clone(\Omega, a_1, ..., a_{|A|}). \tag{64}$$

Constants as nullary operations with constant values, composed with 0-many n-ary operations are n-ary operations with constant values. Thus, to be preserved by all constants operations, any unary relation has to contain the whole set A, and any binary relation has to contain the diagonal relation Δ_A. We can impose these conditions on the set Γ_A. For the algebra $\mathbb{A}$ denote by $\Gamma_{\mathcal{A}}^{diag} = (\Gamma_{\mathcal{A}}^{1,diag}, \Gamma_{\mathcal{A}}^{2,diag})$ the pair of sets such that

$$\begin{aligned}
\Gamma_{\mathcal{A}}^{1,diag}(j,a) &\iff \Gamma_{\mathcal{A}}^1(j,a) \wedge (\forall b \in A, \Gamma_{\mathcal{A}}^1(j,b)) \\
\Gamma_{\mathcal{A}}^{2,diag}(i,a,b) &\iff \Gamma_{\mathcal{A}}^2(i,a,b) \wedge (\forall c \in A, \Gamma_{\mathcal{A}}^2(j,c,c)).
\end{aligned} \tag{65}$$

An n-ary operation P on algebra $\mathbb{A}$ is a polynomial operation if it is a polymorphism for relations from $\Gamma_{\mathcal{A}}^{diag}$, i.e.

$$P \in Polynom(\mathbb{A}) \iff Pol_n(P, A, \Gamma_{\mathcal{A}}^{diag}). \tag{66}$$

For any two sets A and P the following $\Sigma_0^{1,b}$-definable relation claims that P is a ternary discriminator on A:

$$\begin{aligned}
PCD(A,P) &\iff \forall a,b,c \in A, \\
(a = b \wedge P(a,b,c) = c) &\vee (a \neq b \wedge P(a,b,c) = a).
\end{aligned} \tag{67}$$

Before the formalization of the "only if" implication of Theorem 11 as the polynomially complete axiom scheme, we need to encode one more notion. For any congruence σ on algebra $\mathbb{A} = (A, \Omega)$, for factor algebra $\mathbb{A}/\sigma$ we will define the quotient set of relation $\Gamma_{\mathcal{A}}/\sigma$ as follows:

$$\begin{aligned}
\Gamma_{\mathcal{A}}^1/\sigma(j,a) &\iff \forall a/\sigma \in A, \, Rep_m(a, a/\sigma, A/\sigma, A, \Omega, \sigma) \wedge \Gamma_{\mathcal{A}}^1(j, a/\sigma) \\
\Gamma_{\mathcal{A}}^2/\sigma(i,a,b) &\iff \forall a/\sigma, b/\sigma \in A, \, \Gamma_{\mathcal{A}}^2(i, a/\sigma, b/\sigma) \wedge \\
\wedge Rep_m(a, a/\sigma, A/\sigma, A, \Omega, \sigma) &\wedge Rep_m(b, b/\sigma, A/\sigma, A, \Omega, \sigma).
\end{aligned} \tag{68}$$

The definition follows from log-space reduction from $CSP(\mathbb{A}/\sigma)$ to $CSP(\mathbb{A})$. Note, that for some i,j, $\Gamma_{\mathcal{A},j}^1/\sigma$ and $\Gamma_{\mathcal{A},i}^2/\sigma$ are empty sets, as well as $\Gamma_{\mathcal{A},j}^{1,diag}$ and $\Gamma_{\mathcal{A},i}^{2,diag}$.

Definition 20 (PC$_{\mathcal{A}}$-axioms)**.** For any constraint language $\Gamma_{\mathcal{A}}$ over set A of size l, fixed algebra $\mathbb{A} = (A, \Omega)$ with Ω being an m-ary special WNU operation, and finitely many subuniverses D of $\mathbb{A}$ and congruence blocks E of $\mathbb{D}$ the *polynomially complete axiom scheme* is denoted by PC$_{\mathcal{A}}$-axioms and consists of the finitely many formulas of the following form

$$\begin{aligned}
PC_{\mathcal{A},D,E} =_{def} &\forall \mathcal{X} = (V_{\mathcal{X}}, E_{\mathcal{X}}), \forall \ddot{\mathcal{A}} = (V_{\ddot{\mathcal{A}}}, E_{\ddot{\mathcal{A}}}, D_0, ..., D_{n-1}), \\
&([\forall j < n, \forall B < l, \forall T < (3l)^{2^3}, Pol_2(T, D_j, \Gamma_{\mathcal{A}}) \to \neg BAsubS(B, D_j, T) \wedge \\
&\wedge \forall j < n, \forall C < l, \forall S < (4l)^{2^4}, Pol_3(S, D_j, \Gamma_{\mathcal{A}}) \to \neg CRsubS(C, D_j, S)] \\
&\wedge \exists \sigma < \langle l, l \rangle, \exists D/\sigma < l, \exists \Omega/\sigma < (ml)^{2^{m+1}}, FA_m(D/\sigma, \Omega/\sigma, D, \Omega, \sigma) \wedge \\
&\wedge \exists P < (4l)^{2^4}, \, Pol_3(P, D/\sigma, \Gamma_{\mathcal{D}}^{diag}/\sigma) \wedge PCD(D/\sigma, P) \wedge \\
&SwNU_m(\Omega, D) \wedge PsubS(E, D) \wedge (\forall a \in E, \forall b \in D, \sigma(a,b) \leftrightarrow b \in E) \wedge \\
&\wedge Inst(\Theta, \Gamma_{\mathcal{A}}) \wedge CCInst(\mathcal{X}, \ddot{\mathcal{A}}) \wedge IRDInst(\mathcal{X}, \ddot{\mathcal{A}}) \wedge \\
&\exists i < n, D_i = D \wedge \\
&H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}})) \to H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}} = (V_{\ddot{\mathcal{A}}}, E_{\ddot{\mathcal{A}}}, D_0, ..., E, ..., D_{n-1})).
\end{aligned} \tag{69}$$

In this $\forall\Sigma_2^{1,b}$-formula, the first and the second lines in square brackets say that for any domain D_j of instance Θ there are no binary absorbing or central subuniverses. The fourth and fifth lines claim that there exists a congruence σ on D and the corresponding factor algebra $\mathbb{D}/\sigma = (D/\sigma, \Omega/\sigma)$ such that this factor algebra is polynomially complete. Note that we define a discriminator P on factor set D/σ, and require that P is a polymorphism for all relations from the quotient set of relation $\Gamma_\mathcal{D}^{diag}/\sigma$. The sixth line says that D is closed under Ω, E is a proper subset of D and E is a congruence class of σ. Finally, the rest of the formula says that if D coincides with a domain D_i for some variable i, all the above-mentioned conditions hold and there is a solution to the instance Θ, then there is a solution to the instance Θ with D_i restricted to the congruence class E.

5.3   A new theory of bounded arithmetic

For any relational structure $\mathcal{A}$ let us define a new theory of bounded arithmetic extending the theory V^1, as follows.

Definition 21 (Theory $V_\mathcal{A}^1$)**.**

$$V_\mathcal{A}^1 =_{def} V^1 + \{\mathrm{BA}_\mathcal{A}\text{-axioms, } \mathrm{CR}_\mathcal{A}\text{-axioms, } \mathrm{PC}_\mathcal{A}\text{-axioms}\}.$$

Each of the universal algebra axiom schemes $\mathrm{BA}_\mathcal{A}$-axioms, $\mathrm{CR}_\mathcal{A}$-axioms, and $\mathrm{PC}_\mathcal{A}$-axioms consists of a finitely many $\forall\Sigma_2^{1,b}$-formulas for the fixed finite algebra $\mathbb{A} = (A, \Omega)$ with a special WNU operation Ω corresponding to the relational structure $\mathcal{A} = (A, \Gamma_\mathcal{A})$.

We are going to show that for any structure $\mathcal{A}$ which leads to p-time solvable CSP, the theory $V_\mathcal{A}^1$ proves the soundness of Zhuk's algorithm.

5.4   Consistency reductions

Consistency reductions of Zhuk's algorithm precede all other reductions and the linear case and include cycle-consistency reduction (function CheckCycleConsistency), irreducibility reduction (function CheckIrreducibility) and weaker instance reduction (function CheckWeakerInstance), see [19]. Consider a CSP instance $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$ with domain set $D = \{D_0, ..., D_{n-1}\}$. During consistency reductions the algorithm works with some modifications of an input digraph $\mathcal{X}$ and a target digraph with domains $\ddot{\mathcal{A}}$. At the end of every procedure, the output is either "No solution" (some domain is empty after reduction), or "OK" (the algorithm cannot reduce any domain since the instance satisfies the property we are checking), or the reduction (i, D_i') of the first domain in a line that we can reduce.

At the beginning of every procedure, for simplicity we will refer to every input instance as the initial one, $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$. It makes sense: after every reduction (i, D_i') we start the algorithm all from the beginning with the same input digraph (the same set of variables and the same set of constraints) but with a smaller domain set $D' = \{D_0, ..., D_i', ..., D_{n-1}\}$: we remove some vertices from $\ddot{\mathcal{A}}$, which induces removing some edges. If the algorithm moves to another procedure, it means that the previous one cannot reduce any domain - so technically, we proceed with the same instance from the beginning of the current step of recursion.

5.4.1   Cycle-consistency

In this section we will formalize the modification of the function CheckCycleConsistency suggested by Zhuk in his latter paper [20]. In short, the algorithm first intersects all

constraints and then uses constraint propagation to ensure a type of consistency called $(2,3)$-consistency. In words, $(2,3)$-consistency means that for any variables i, j, k every edge (i, j) extends to a triangle by edges (i, k) and (k, j). These two properties taken together provide cycle-consistency. We explain the procedure in detail alongside the formalization.

Consider a CSP instance $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$. First, for any two variables i, j the algorithm defines a full relation $R_{i,j}$ on domains $D_i \times D_j$. We define a new target digraph with domains $\ddot{\mathcal{R}} = (V_{\ddot{\mathcal{R}}}, E_{\ddot{\mathcal{R}}}, D_0, ..., D_{n-1})$, where $V_{\ddot{\mathcal{R}}} = V_{\ddot{\mathcal{A}}}$, but while

$$
\begin{aligned}
E_{\ddot{\mathcal{A}}}(u, v) \longrightarrow & \exists i, j < n\, \exists a, b < l\ u = \langle i, a \rangle \wedge v = \langle j, b \rangle \wedge \\
& D_i(a) \wedge D_j(b),
\end{aligned}
\tag{70}
$$

for $E_{\ddot{\mathcal{R}}}$ we have

$$
\begin{aligned}
E_{\ddot{\mathcal{R}}}(u, v) \iff & \exists i, j < n\, \exists a, b < l\ u = \langle i, a \rangle \wedge v = \langle j, b \rangle \wedge \\
& D_i(a) \wedge D_j(b).
\end{aligned}
\tag{71}
$$

That is, for all $i, j \in \{0, ..., n-1\}$, $E_{\ddot{\mathcal{R}}}^{ij}$ is the full binary relation on $D_i \times D_j$ (even for those i, j, for which $\neg E_{\mathcal{X}}(i, j)$ and $\neg E_{\mathcal{X}}(j, i)$).

Then for all $i, j \in \{0, ..., n-1\}$ the algorithm intersects each $E_{\ddot{\mathcal{R}}}^{ij}$ with projections of all constraints onto the variables i, j. In our case, for i, j we have only constraints D_i, D_j, $E_{\ddot{\mathcal{A}}}^{ij}$, and $E_{\ddot{\mathcal{A}}}^{ji}$, i.e. we intersect $E_{\ddot{\mathcal{R}}}^{ij}$ only with $E_{\ddot{\mathcal{A}}}^{ij}$ and $E_{\ddot{\mathcal{A}}}^{ji}$. Let us denote new relations by $E_{\ddot{\mathcal{R}}_0}^{ij}$:

$$
\begin{aligned}
E_{\ddot{\mathcal{R}}_0}^{ij}(a, b) \iff & (a \in D_i \wedge b \in D_j) \wedge \\
\wedge (E_{\mathcal{X}}(i, j) \to E_{\ddot{\mathcal{A}}}^{ij}(a, b)) & \wedge (E_{\mathcal{X}}(j, i) \to E_{\ddot{\mathcal{A}}}^{ji}(b, a)).
\end{aligned}
\tag{72}
$$

Note that if there are no constraints $E_{\mathcal{X}}(i, j)$ and $E_{\mathcal{X}}(j, i)$, then at this point both $E_{\ddot{\mathcal{R}}_0}^{ij}$ and $E_{\ddot{\mathcal{R}}_0}^{ji}$ are still $D_i \times D_j$, $D_j \times D_i$. Then denote by $Pr_1(i, a)$ the intersection of the projections of all constraints $E_{\ddot{\mathcal{R}}_0}^{ij}$ on variable i:

$$
\begin{aligned}
Pr_1(i, a) \iff & a \in D_i \wedge \forall j < n,\ E_{\mathcal{X}}(i, j) \to \exists b \in D_j,\ E_{\ddot{\mathcal{R}}_0}^{ij}(a, b) \wedge \\
& \forall k < n,\ E_{\mathcal{X}}(k, i) \to \exists c \in D_k,\ E_{\ddot{\mathcal{R}}_0}^{ki}(c, a).
\end{aligned}
\tag{73}
$$

Let us define a new digraph $\ddot{\mathcal{R}}_1$ with domains by setting

$$
V_{\ddot{\mathcal{R}}_1}(i, a) \iff Pr_1(i, a),
\tag{74}
$$

and

$$
E_{\ddot{\mathcal{R}}_1}^{ij}(a, b) \iff Pr_1(i, a) \wedge Pr_1(j, b) \wedge E_{\ddot{\mathcal{R}}_0}^{ij}(a, b).
\tag{75}
$$

Then the algorithm produces iterative propagation of constraints until it cannot change any further relation. For every step of propagation $t > 1$, for all $i, j \in \{0, ..., n-1\}$ we define a new set R_t as follows:

$$
R_1(i, j, a, b) \iff E_{\ddot{\mathcal{R}}_1}^{ij}(a, b),
\tag{76}
$$

and for $t > 1$

$$
\begin{aligned}
R_t(i, j, a, b) \iff & R_{t-1}(i, j, a, b) \wedge \\
\forall k < n \exists c < l\ Pr_1(k, c) \wedge & (R_{t-1}(i, k, a, c) \wedge R_{t-1}(k, j, c, b)).
\end{aligned}
\tag{77}
$$

The existence of this set is ensured by $\Sigma_1^{1,b}$-induction. For every step of propagation $t > 1$, $R_t(i, j, a, b)$ corresponds to the relation $E_{\ddot{\mathcal{R}}_t}^{ij}$ and thus induces the next digraph with domains $\ddot{\mathcal{R}}_t$. The process will eventually stop since on every step $t > 1$ we remove some edges from $\ddot{\mathcal{R}}_{t-1}$, and the number of edges in $\ddot{\mathcal{R}}_1$ is bounded by some polynomial of n and l, $p(n, l)$. Let us prove it.

Denote the number of edges in $\ddot{\mathcal{R}}_1$ by $q = \#E_{\ddot{\mathcal{R}}_1}$, i.e. the number of elements in $R_1(i, j, a, b)$ is q. For every $t \leq (q+1)$ due to definition $\forall i, j < n, \forall a, b < k$, $R_t(i, j, a, b) \rightarrow R_{t-1}(i, j, a, b)$. Suppose that for some $t = q' < q + 1$ we have

$$\forall i, j < n, \forall a, b < l, \ R_{q'}(i, j, a, b) \iff R_{q'-1}(i, j, a, b).$$

Then it means that the part

$$\forall k < n \exists c < l, \ Pr_1(k, c) \wedge (R_{t-1}(i, k, a, c) \wedge R_{t-1}(k, j, c, b))$$

is always true when $t = q'$. By induction on s we can prove that in this case

$$\forall i, j < n, \forall a, b < l, \ R_{q'+s}(i, j, a, b) \iff R_{q'-1}(i, j, a, b)$$

since for $s = 0$ it is a suggestion, and if it is true for $s = f$, then we can rewrite the definition of $R_{q'+f+1}$ using equivalent sets

$$
\begin{aligned}
R_{q'+f+1}(i, j, a, b) &\iff R_{q'-1}(i, j, a, b) \wedge \\
\forall k < n \exists c < l, \ Pr_1(k, c) &\wedge (R_{q'-1}(i, k, a, c) \wedge R_{q'-1}(k, j, c, b)).
\end{aligned}
\tag{78}
$$

Now suppose that for every $1 < t \leq (q + 1)$, $\neg(R_{t-1}(i, j, a, b) \rightarrow R_t(i, j, a, b))$, i.e. for every t there exist $i, j < n, a, b < l$ such that $R_{t-1}(i, j, a, b) \wedge \neg R_t(i, j, a, b)$, i.e. $\#R_t < \#R_{t-1}$. Then by induction on t we can prove that $\#R_t \leq q - (t - 1)$, therefore $\#R_{q+1} \leq 0$ (the "worst" case - we removed all edges from $\ddot{\mathcal{R}}_1$). In both cases we proved that for every $t > q$, $R_{t+1}(i, j, a, b) \iff R_t(i, j, a, b)$.

After the end of propagation, we reduce domains for the second time.

$$
\begin{aligned}
Pr_{cc}(i, a) &\iff Pr_1(i, a) \wedge \forall j < n, E_{\mathcal{X}}(i, j) \rightarrow \exists b, Pr_1(j, b) \wedge E_{\ddot{\mathcal{R}}_{q+1}}^{ij}(a, b) \\
&\wedge \forall k < n, E_{\mathcal{X}}(k, i) \rightarrow \exists c, Pr_1(k, c) \wedge E_{\ddot{\mathcal{R}}_{q+1}}^{ki}(c, a).
\end{aligned}
\tag{79}
$$

We denote the new (cycle-consistent) target digraph with domains by $\ddot{\mathcal{A}}_{cc}$ and set

$$V_{\ddot{\mathcal{A}}_{cc}}(i, a) \iff Pr_{cc}(i, a), \tag{80}$$

and

$$E_{\ddot{\mathcal{A}}_{cc}}^{ij}(a, b) \iff (Pr_{cc}(i, a) \wedge Pr_{cc}(j, b)) \wedge E_{\ddot{\mathcal{R}}_{q+1}}^{ij}(a, b). \tag{81}$$

Remark 2. In Zhuk's algorithm, the original function CheckCycleConsistency in [19] reduces one domain D_i at a time (as if in (79) we fix some i), outputs the result (x_i, D_i') and starts all from the beginning. The modified function CheckCC in [20] returns all reduced domains at once. Both do not return the reduced relations $E_{\ddot{\mathcal{A}}}^{ij}$: the algorithm applies the function to the initial instance again and again until it cannot produce any further reduction. Nonetheless, it does not affect the final result (we cannot produce two different cycle-consistent reductions), so we omit these technical intermediate steps here.

Now we need to prove the following two statements:

1. The instance $\Theta_{cc} = (\mathcal{X}, \ddot{\mathcal{A}}_{cc})$ is a cycle-consistent instance (according to definition).

2. If the initial instance $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$ has a solution, then Θ_{cc} has a solution.

Lemma 5. V^1 *proves that if none of the domains* $V_{\ddot{\mathcal{A}}_{cc}}(i), i < n$ *is empty, then the instance* $\Theta_{cc} = (\mathcal{X}, \ddot{\mathcal{A}}_{cc})$ *is cycle-consistent.*

Proof. Due to definitions (79)-(81), the instance Θ_{cc} is 1-consistent. For any $i < n$, any $a \in V_{\ddot{\mathcal{A}}_{cc}}(i)$ consider any cycle $\mathcal{C}_t$ that can be homomorphically mapped into $\mathcal{X}$ with $H(0) = i$ and define the set $H' < \langle t, \langle t, l \rangle \rangle$ such that $H'(0) = \langle i, a \rangle$ and for all $j < t, k < n, H(j) = k \rightarrow H'(j) = \langle k, b \rangle$ for some $b \in V_{\ddot{\mathcal{A}}_{cc}}(k)$ (it exists since none of the domains is empty). We need to prove that there is b_k for each $k < n$ such that H' is a homomorphism from $\mathcal{C}_t$ to $\ddot{\mathcal{A}}$. For this, it is enough to note that by the construction (78), the formula

$$\exists b_1, b_2, ..., b_{t-1} < l, \ \tilde{E}_{\ddot{\mathcal{A}}_{cc}}^{ik_1}(a, b_1) \wedge \tilde{E}_{\ddot{\mathcal{A}}_{cc}}^{k_1 k_2}(b_1, b_2) \wedge ... \wedge \tilde{E}_{\ddot{\mathcal{A}}_{cc}}^{k_{t-1} i}(b_{t-1}, a) \tag{82}$$

where $\tilde{E}_{\ddot{\mathcal{A}}_{cc}}^{k_i k_{i+1}}(b_i, b_{i+1})$ is either $E_{\ddot{\mathcal{A}}_{cc}}^{k_i k_{i+1}}(b_i, b_{i+1})$ or $E_{\ddot{\mathcal{A}}_{cc}}^{k_{i+1} k_i}(b_{i+1}, b_i)$ depending on the cycle $\mathcal{C}_t$, is always true since for any $a \in V_{\ddot{\mathcal{A}}_{cc}}(i)$:

$$E_{\ddot{\mathcal{A}}_{cc}}^{ii}(a, a) \rightarrow \exists b_{t-1} < l, \ E_{\ddot{\mathcal{A}}_{cc}}^{ik_{t-1}}(a, b_{t-1}) \wedge E_{\ddot{\mathcal{A}}_{cc}}^{k_{t-1} i}(b_{t-1}, a),$$
$$...$$
$$E_{\ddot{\mathcal{A}}_{cc}}^{ik_3}(a, b_3) \rightarrow \exists b_2 < l, \ E_{\ddot{\mathcal{A}}_{cc}}^{ik_2}(a, b_2) \wedge E_{\ddot{\mathcal{A}}_{cc}}^{k_2 k_3}(b_2, b_3), \tag{83}$$
$$E_{\ddot{\mathcal{A}}_{cc}}^{ik_2}(a, b_2) \rightarrow \exists b_1 < l, \ E_{\ddot{\mathcal{A}}_{cc}}^{ik_1}(a, b_1) \wedge E_{\ddot{\mathcal{A}}_{cc}}^{k_1 k_2}(b_1, b_2).$$

Set $H'(i) = \langle k_i, b_i \rangle$ for all $0 < i < t$. This completes the proof. $\square$

Lemma 6. V^1 *proves that instance* $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$ *has a solution if and only if* $\Theta_{cc} = (\mathcal{X}, \ddot{\mathcal{A}}_{cc})$ *has a solution.*

Proof. Suppose that there is a homomorphism H from $\mathcal{X}$ to $\ddot{\mathcal{A}}$ and it sends edge $E_{\mathcal{X}}(i, j)$ to $E_{\ddot{\mathcal{A}}}^{ij}(a, b)$ for $a \in D_i, b \in D_j$. Due to the definition of a homomorphism for both a and b, $E_{\ddot{\mathcal{A}}}^{ij}$ must satisfy (72)-(75) and we do not lose any solution after the intersection of all constraints. That is, instead of the set $\{\mathcal{X} \rightarrow \ddot{\mathcal{A}}\}$ we can consider set $\{\mathcal{X} \rightarrow \ddot{\mathcal{R}}_1\}$.

Consider a formula $\phi(t)$ which says that if H is a homomorphism from $\mathcal{X}$ to $\mathcal{R}'_1$, then for every step t of propagation, for all $i, j, k \in \{0, 1, ..., n-1\}$, all $a, b, c < l$

$$\phi(t) = H\ddot{O}M(\mathcal{X}, \mathcal{R}'_1, H) \wedge H(i) = \langle i, a \rangle \wedge H(j) = \langle j, b \rangle \wedge H(k) = \langle k, c \rangle \longrightarrow$$
$$(E_{\ddot{\mathcal{R}}_t}^{ij}(a, b) \wedge E_{\ddot{\mathcal{R}}_t}^{ik}(a, c) \wedge E_{\ddot{\mathcal{R}}_t}^{kj}(c, b)). \tag{84}$$

For $t = 1$ this is true. For every constraint $E_{\mathcal{X}}(i, j)$ the implication $E_{\ddot{\mathcal{R}}_1}^{ij}(a, b)$ follows from the definition of a homomorphism. For any i, j such that $\neg E_{\mathcal{X}}(i, j)$ the implication $E_{\ddot{\mathcal{R}}_1}^{ij}(a, b)$ follows from the definition of $E_{\ddot{\mathcal{R}}_0}^{ij}$ and (73)-(75): we do not remove edges from $\ddot{\mathcal{R}}_1$ between domains not connected in a constraint without removing vertices. Thus, if there remain some vertices, there will remain all edges between these vertices as well.

If $\phi(t)$ is true for $t = s$, then it is true for $t = (s + 1)$ due to construction (77). Hence, $\{\mathcal{X} \rightarrow \ddot{\mathcal{A}}\} \subseteq \{\mathcal{X} \rightarrow \ddot{\mathcal{A}}_{cc}\}$. The opposite inclusion is trivial. $\square$

5.4.2 Irreducibility

Consider a cycle-consistent instance $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$ with a domain set $D = \{D_0, ..., D_{n-1}\}$. The algorithm chooses a variable i and some maximal congruence σ_i on D_i and denotes by

$I = \{i\}$ the set of the indices. Then it considers all other variables k such that $k \notin I$ and for some $j \in I$ there is a projection of some constraint C onto j, k. Since we consider at most binary relations, and the instance is cycle-consistent, it follows that the projection of any constraint $E_{\mathcal{X}}(j, k)$ (or $E_{\mathcal{X}}(k, j)$) onto j, k is either the constraint relation $E_{\mathcal{A}}^{jk}$ (or $E_{\mathcal{A}}^{kj}$) or domains D_j, D_k. On domain D_k of such variable k, the algorithm generates relation σ_k as follows:

$$E_{\mathcal{X}}(j, k) : \sigma_k(a, b) \iff \exists a', b' \in D_j, \sigma_j(a', b') \wedge E_{\mathcal{A}}^{jk}(a, a') \wedge E_{\mathcal{A}}^{jk}(b, b'). \qquad (85)$$

That is, the algorithm defines a partition on D_k according to the partition on D_j. Since this new relation is constructed from relations compatible with Ω by *pp*-definition, it is also compatible with Ω, and therefore is a congruence. If this congruence is proper, then we have the same number of equivalence classes on D_k as on D_j, and elements from one class in D_k are connected with elements only from one class in D_j. Otherwise, σ_j is not maximal since we can define a new congruence on D_j in an analogous way as in (85). The algorithm collects all such D_k with proper congruences σ_k into the list of indices I, and then considers the projection $\Theta_{prX'}$ of the initial instance onto $X' = \{k | k \in I\}$. This projection can be split into instances on smaller domains (corresponding to connected classes in different domains), and these instances can be solved by recursion.

Remark 3. If there is no domain D_k such that σ_i generates on it a proper congruence, the algorithm moves first to another maximal congruence σ_i' on D_i and then to $i + 1 \in \{0, 1, ..., n-1\}$.

For every $k \in I$ we thus can check if the solution set to the projection $\Theta_{prX'}$ is subdirect. If not, and for some $k \in I$ there are $b_1, ..., b_s$ such that there is no solution to $\Theta_{prX'}$, then the algorithm return $D_k' = D_k \backslash \{b_1, ..., b_s\}$ and runs from the beginning. If for all $b \in D_k$ there is no solution to $\Theta_{prX'}$, then the algorithm returns "No solution". If the solution set to $\Theta_{prX'}$ is subdirect, then the algorithm moves to another maximal congruence on D_i, and then to $i + 1 \in \{0, 1, ..., n-1\}$. If the algorithm cannot reduce any domain D_i, and none of the domains is empty, the algorithm returns "OK".

For the formalization of the function CheckIrreducibility, for every domain D_i let us denote by $\sigma_i(q, a, b)$ the list of all maximal congruences on D_i (we know them in advance). The number of all congruences on D_i is some constant $q_i \leq 2^{l^2}$. Then for every variable $i \in X$, and every maximal congruence $\sigma_i^q(a, b)$ on D_i we iteratively define the following set of elements $I_{t,i,q}(j, a, b)$, where t is the iteration step, i is fixed domain, q is fixed maximal congruence, j is the considered domain and a, b are elements in one congruence class:

$$\forall a, b < l, \ I_{0,i,q}(i, a, b) \iff \sigma_i^q(a, b) \wedge$$
$$\wedge \forall 0 < t < n, k < n, a, b < l, \ I_{t,i,q}(k, a, b) \iff I_{t-1,i,q}(k, a, b) \vee$$
$$\vee \exists j < n, a', b' < l, \ I_{t-1,i,q}(j, a', b') \wedge$$
$$\wedge (E_{\mathcal{X}}(j, k) \wedge E_{\mathcal{A}}^{jk}(a', a) \wedge E_{\mathcal{A}}^{jk}(b', b)) \vee (E_{\mathcal{X}}(k, j) \wedge E_{\mathcal{A}}^{kj}(a, a') \wedge E_{\mathcal{A}}^{kj}(b, b')) \wedge \qquad (86)$$
$$\wedge \neg [\exists c, d \in D_j, \exists e \in D_k, \ \neg I_{t-1,i,q}(j, c, d) \wedge$$
$$\wedge (E_{\mathcal{X}}(j, k) \wedge E_{\mathcal{A}}^{jk}(c, e) \wedge E_{\mathcal{A}}^{jk}(d, e)) \vee (E_{\mathcal{X}}(k, j) \wedge E_{\mathcal{A}}^{kj}(e, c) \wedge E_{\mathcal{A}}^{kj}(e, d))].$$

At step $t = 0$ the set $I_{0,i,q}$ contains only index i and (a, b) such that $a, b \in D_i$ are in the same congruence class of σ_i^q. At each further step $t > 0$ we add to $I_{t,i,q}$ all elements from $I_{t-1,i,q}$ and indices of the domains connected to elements from $I_{t-1,i,q}$ such that σ_i^q generates proper partitions on those domains. Lines 3-5 consider a connection between j and k and define a partition on $I_{t,i,q}(k)$, and lines 6-8 in square brackets checks that this

partition is proper, i.e. no elements $c, d \in D_j$ from different congruence classes connected in D_k. Since we cannot add more than n elements to I, $I_{n,i,q}$ contains all wanted elements. The existence of this set is provided by induction on t on $\Sigma_1^{1,b}$-formula, and the implication $t \to (t+1)$ follows from comprehension axiom scheme $\Sigma_0^{1,b}$-CA.

Suppose that the algorithm returns "OK". We will denote the new target digraph with domains after irreducibility reduction by $\ddot{\mathcal{A}}_{ir}$. Due to the algorithm, for each subinstance Θ'_{ir} of Θ_{ir}, considered by the function CheckIrreducibility, the solution set to Θ'_{ir} is subdirect. It is obvious that Θ'_{ir} is not fragmented and not linked. We can formalize the properties of the instance $\Theta_{ir} = (\mathcal{X}, \ddot{\mathcal{A}}_{ir})$ as follows: for every $i \in V_{\mathcal{X}}$ and every maximal congruence σ_i^q

$$\forall V_{\mathcal{X}'} < n, \forall E_{\mathcal{X}'} < 4n^2, \mathcal{X}' = (V_{\mathcal{X}'}, E_{\mathcal{X}'}),$$
$$((\forall j < n, \exists a, b < l, V_{\mathcal{X}'}(j) \leftrightarrow I_{n,i,q}(j, a, b)) \wedge (\forall s, s' < n, E_{\mathcal{X}'}(s, s') \to s, s' \in V_{\mathcal{X}'}) \wedge \quad (87)$$
$$\wedge (\forall s, s' \in V_{\mathcal{X}'}, E_{\mathcal{X}'}(s, s') \leftrightarrow E_{\mathcal{X}}(s, s'))) \to subDSSInst(\mathcal{X}', \ddot{\mathcal{A}}_{ir}).$$

We need to prove two statements:

1. The instance $\Theta_{ir} = (\mathcal{X}, \ddot{\mathcal{A}}_{ir})$ is irreducible due to definition.

2. The initial instance $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$ has a solution only if Θ_{ir} has a solution.

We start with several technical lemmas.

Lemma 7. V^1 *proves that for any cycle-consistent instance $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$, for any $i \in X$, relation $Linked(a, b, i, i, \Theta)$ is a congruence on D_i.*

Proof. Recall the definition of $Linked(a, b, i, i, \Theta)$:

$$Linked(a, b, i, i, \Theta) \iff \exists t < nl, V_{\mathcal{P}_t} = t, E_{\mathcal{P}_t} \le t^2,$$
$$PATH(V_{\mathcal{P}_t}, E_{\mathcal{P}_t}) \wedge \exists H \le \langle t, n \rangle, HOM(\mathcal{P}_t, \mathcal{X}, H) \wedge (H(0, i) \wedge H(t, i)) \wedge$$
$$\wedge \exists H' \le \langle t, \langle t, l \rangle \rangle, H\ddot{O}M(\mathcal{P}_t, \ddot{\mathcal{A}}, H') \wedge \quad (88)$$
$$\wedge (\forall k < n, p < t, (H(p, k) \to \exists c \in D_k, H'(p) = \langle k, c \rangle))$$
$$\wedge H'(0) = \langle i, a \rangle \wedge H'(t) = \langle i, b \rangle.$$

First of all, for any $a \in D_i$ we have $Linked(a, a, i, i, \Theta)$. Indeed, since the instance is cycle-consistent, it follows that for any cycle $\mathcal{C}_t$ that can be mapped to $\mathcal{X}$ with $H(0, i)$, we will have a homomorphism H' from $\mathcal{C}_t$ to $\ddot{\mathcal{A}}$ such that

$$\forall j < n, k < t \, (H(k, j) \to \exists b \in D_j, H'(i) = \langle j, b \rangle) \wedge H'(0) = \langle i, a \rangle.$$

Instead of cycle $\mathcal{C}_t$ consider a path $\mathcal{P}_t$ such that for all $i < (t-1)$

$$E_{\mathcal{P}_t}(i, i+1) \leftrightarrow E_{\mathcal{C}_t}(i, i+1) \wedge E_{\mathcal{P}_t}(i+1, i) \leftrightarrow E_{\mathcal{C}_t}(i+1, i),$$

and for $i = (t-1)$

$$E_{\mathcal{P}_t}(i, i+1) \leftrightarrow E_{\mathcal{C}_t}(i, 0) \wedge E_{\mathcal{P}_t}(i+1, i) \leftrightarrow E_{\mathcal{C}_t}(0, i),$$

and set $H(t, i), H'(t) = \langle i, a \rangle$. Thus, $Linked(a, b, i, i, \Theta)$ is indeed a relation on the whole D_x, and a reflexive one. To prove that the relation is symmetric, for any a, b such that

$$\exists \mathcal{P}_t < \langle nl, (nl)^2 \rangle, Linked(a, b, i, i, \Theta, \mathcal{P}_t),$$

consider the inverse path $\mathcal{P}_t^{-1}$ and define a new homomorphisms M, M such that for all $j \leq t, k < n, c < l$

$$M(j,k) \leftrightarrow H(t-j,k) \wedge M'(j) = \langle k, c \rangle \leftrightarrow H'(t-j) = \langle k, c \rangle.$$

Finally, if for $a, b, c \in D_i$, there are

$$\exists \mathcal{P}_t < \langle nl, (nl)^2 \rangle, Linked(a, b, i, i, \Theta, \mathcal{P}_t),$$

$$\exists \mathcal{P}_m < \langle nl, (nl)^2 \rangle, Linked(b, c, i, i, \Theta, \mathcal{P}_m),$$

we can consider the glued path $\mathcal{P}_t \circ \mathcal{P}_m$, and use on the first and second parts of the path homomorphisms corresponding to $\mathcal{P}_t$ and $\mathcal{P}_m$ respectively. Thus, the relation is transitive.

It remains to show that the relation is compatible with operation Ω, i.e. $Pol_{m,2}(\Omega, D_i, Linked_{[i,i,\Theta]})$. But it follows from the fact that the set of all pairs $(a, b) \in Linked_{[i,i,\Theta]}$ can be defined by a pp-positive formula (see [19]), and therefore is in the list $\Gamma_{\mathcal{A}}^2$. $\qquad\square$

Note that since for every variable $i \in X$ the algorithm checks every maximal congruence on D_i, it follows that $Linked_{[i,i,\Theta]}$ is either contained in some maximal congruence or is a maximal congruence itself. Also, for any cycle-consistent instance $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$, for any its subinstance $\Theta' = (\mathcal{X}', \ddot{\mathcal{A}})$ and any $D_i, i \in X'$

$$Linked(a, b, i, i, \Theta') \rightarrow Linked(a, b, i, i, \Theta),$$

i.e. the congruence relation $Linked_{[i,i,\Theta]}$ of the instance Θ contains the congruence relation $Linked_{[i,i,\Theta']}$ of any its subinstance Θ'. By adding any new variable $j \in X \backslash X'$ to X' with all induced edges from $\mathcal{X}$, we cannot make relation $Linked_{[i,i,\Theta']}$ smaller since when it comes down to being linked we consider the existence of a path, and for any $a, b \in D_i$ in $Linked_{[i,i,\Theta']}$ the path already exists. But we can add some new paths, making $Linked_{[i,i,\Theta']}$ larger.

Lemma 8. V^1 *proves that if an instance* $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$ *is not fragmented, then for any* $i, j \in V_{\mathcal{X}}$ *there exist* $t < n$ *and a path* $\mathcal{P}_t$ *such that*

$$\exists H \leq \langle t, n \rangle, HOM(\mathcal{P}_t, \mathcal{X}, Z) \wedge H(0) = i \wedge H(t) = j.$$

Proof. Consider the formula $\theta(t)$

$$\theta(t) =_{def} t < n, i \in V_{\mathcal{X}}^1, j \in V_{\mathcal{X}}^2 \wedge V_{\mathcal{X}}^1 = V_{\mathcal{X}}^2 = n \wedge \#V_{\mathcal{X}}^2(n) = t \wedge$$
$$\wedge PsubS(V_{\mathcal{X}}^1, V_{\mathcal{X}}) \wedge PsubS(V_{\mathcal{X}}^2, V_{\mathcal{X}}) \wedge (\forall k < n, V_{\mathcal{X}}^1(k) \leftrightarrow \neg V_{\mathcal{X}}^1(k)) \wedge$$
$$\wedge \exists m \leq t, \exists \mathcal{P}_m, V_{\mathcal{P}_m} = m, E_{\mathcal{P}_m} < m^2, PATH(V_{\mathcal{P}_m}, E_{\mathcal{P}_m}) \wedge \qquad (89)$$
$$\wedge \exists H \leq \langle m, n \rangle, HOM(\mathcal{P}_m, \mathcal{X}, H) \wedge H(0, j) \wedge H(m, i') \wedge i' \in V_{\mathcal{X}}^1.$$

For $t = 1$, the formula is true since $\neg FragmInst(\mathcal{X}, \ddot{\mathcal{A}})$. If $\theta(t)$ is true for $t = s$, then it is also true for $t = (s+1)$. Indeed, since the instance is not fragmented, it follows that for $V_{\mathcal{X}}^2, \#V_{\mathcal{X}}^2(n) = (s+1)$ there are two elements $i' \in V_{\mathcal{X}}^1$ and $j' \in V_{\mathcal{X}}^2$ such that there is an edge $E_{\mathcal{X}}(i', j')$ or $E_{\mathcal{X}}(j', i')$. Then consider two sets $V_{\mathcal{X}}^1 \cup \{j'\}$ and $V_{\mathcal{X}}^2 \backslash \{j'\}$. Since $\#V_{\mathcal{X}}^2 \backslash \{j'\} = s$, there has to be a path $\mathcal{P}_m, m \leq s$, and $H \leq \langle m, n \rangle$ with $H(0) = j, H(m) = i''$ for some $i'' \in V_{\mathcal{X}}^1 \cup \{j'\}$. If $i'' = j'$, we get a path of length $m \leq (s+1)$ from j to i'. If $i'' \neq j'$, then there is a path of length $m \leq s$ from j to some element $i'' \in V_{\mathcal{X}}^1$. Finally, it also must be true for $t = n - 1$. $\qquad\square$

Lemma 9. V^1 *proves that if a cycle-consistent instance* $\Theta = (\mathcal{X}, \mathcal{A})$ *is not fragmented and not linked, then for all* D_i *there exist* $a, b \in D_i$ *such that* $\neg Linked(a, b, i, i, \Theta)$.

Proof. Since the instance Θ is not linked, by definition there exist $i \in V_{\mathcal{X}}$ and $a, b \in D_i$ such that $\neg LinkedCon(a, b, i, i, \Theta)$. Suppose that there exists D_j such that for any $a', b' \in D_j$ we have $LinkedCon(a', b', j, j, \Theta)$, i.e. there exist some path $\mathcal{P}_t$ and a homomorphism H' from $\mathcal{P}_t$ to $\ddot{\mathcal{A}}$ connecting a' and b'. Since the instance is not fragmented, due to Lemma 8 it follows that there exists a path $\mathcal{P}_s$ from i to j. Consider the reverse path $\mathcal{P}_s^{-1}$ and define a cycle C_{2s} as follows:

$$V_{\mathcal{C}_{2s}} = 2m \wedge \forall k < m, \; E_{\mathcal{C}_{2s}}(k, k+1) \leftrightarrow E_{\mathcal{P}_s}(k, k+1) \wedge$$
$$\wedge \leftrightarrow E_{\mathcal{C}_{2s}}(k+1, k) \leftrightarrow E_{\mathcal{P}_s}(k+1, k) \wedge$$
$$\wedge \forall r < (s-1), \; E_{\mathcal{C}_{2s}}(s+r, s+r+1) \leftrightarrow E_{\mathcal{P}_s^{-1}}(r, r+1) \wedge \tag{90}$$
$$\wedge E_{\mathcal{C}_{2s}}(s+r+1, s+r) \leftrightarrow E_{\mathcal{P}_s^{-1}}(r+1, r) \wedge$$
$$\wedge E_{\mathcal{C}_{2s}}(2s-1, 0) \leftrightarrow E_{\mathcal{P}_s^{-1}}(s-1, s) \wedge E_{\mathcal{C}_{2m}}(0, 2s-1) \leftrightarrow E_{\mathcal{P}_s^{-1}}(s, s-1).$$

That is, in C_{2s} we glued together the start and the end of paths $\mathcal{P}_s$ and $\mathcal{P}_s^{-1}$ respectively, and vice versa. This cycle can obviously be mapped into $\mathcal{X}$, and due to cycle-consistency for $a, b \in D_i$ there exist homomorphisms H'_a, H'_b from $\mathcal{C}_{2s}$ to $\ddot{\mathcal{A}}$ such that $H'_a(0) = \langle i, a \rangle, H'_b(0) = \langle i, b \rangle$. Suppose that $H'_a(s) = \langle j, a' \rangle$ and $H'_b(s) = \langle j, b' \rangle$ and consider a path $\mathcal{P}_s \circ \mathcal{P}_t \circ \mathcal{P}_s^{-1}$. Then use homomorphism H'_a for $\mathcal{P}_s$, H' for $\mathcal{P}_t$ and H'_b for $\mathcal{P}_s^{-1}$. Thus, we have a path and a new homomorphism connecting a and b in D_i. That is a contradiction. $\qquad\square$

Remark 4. Note that in proof of Lemma 9 we have to use cycle-consistency. We can ensure a path from i to j in $\mathcal{X}$ due to the fact that the instance is not fragmented, but without cycle-consistency (or linked property) we cannot ensure that this path has proper evaluation in $\ddot{\mathcal{A}}$.

Lemma 10. V^1 *proves that the instance* $\Theta_{ir} = (\mathcal{X}, \ddot{\mathcal{A}}_{ir})$ *is irreducible.*

Proof. Suppose that there exists a subinstance $\Theta' = (\mathcal{X}', \ddot{\mathcal{A}}_{ir})$ such that $\mathcal{X}' = (V_{\mathcal{X}'}, E_{\mathcal{X}'})$, $V_{\mathcal{X}'} < n, E_{\mathcal{X}'} < 4n^2$, $V_{\mathcal{X}'}$ is a subset of $V_{\mathcal{X}}$, $E_{\mathcal{X}'}$ is a subset of $E_{\mathcal{X}}$, and

$$E_{\mathcal{X}'}(x_1, x_2) \to x_1, x_2 \in V_{\mathcal{X}'},$$

and this instance is not fragmented, and not linked, and its solution set is not subdirect. We need to prove that any such subinstance must be included in some subinstance generated by the algorithm (and therefore must have a subdirect solution set).

Due to Lemma 9, for any $i \in V_{\mathcal{X}'}$ there exist $a, b \in D_i$, $(a, b) \notin Linked_{[i,i,\Theta']}$, thus any such congruence is proper. Fix some $i \in X'$, and consider a maximal congruence $\sigma_i^q(a, b)$ for some $q < q_i$ on D_i that contains $Linked_{[i,i,\Theta']}$. Consider subinstance $\Theta'' = (\mathcal{X}'', \ddot{\mathcal{A}})$, defined as:

$$\forall j < n, \exists a, b < l, \; V_{\mathcal{X}''}(j) \leftrightarrow I_{n,i,q}(j, a, b) \wedge$$
$$\wedge \forall s, s' < n, \; E_{\mathcal{X}''}(s, s') \to s, s' \in V_{\mathcal{X}''} \wedge \tag{91}$$
$$\wedge \forall s, s' \in V_{\mathcal{X}''}, \; E_{\mathcal{X}''}(s, s') \leftrightarrow E_{\mathcal{X}}(s, s').$$

We need to show two points:

1. For every $j \in X'$ there exist $a', b' \in D_j$ such that $I_{n,i,q}(j, a', b')$ (i.e. X' is a subset of X'').

2. For every $j \in X'$, for all $a', b' \in D_j$,

$$I_{n,i,q}(j, a', b') \longrightarrow \exists a, b \in D_i, I_{n,i,q}(i, a, b) \land$$
$$\land Linked(a, a', i, j, \Theta) \land Linked(b, b', i, j, \Theta),$$

and for all $a, b \in D_i$, for all $j \in X'$, $a', b' \in D_j$

$$I_{n,i,q}(i, a, b) \land Linked(a, a', i, j, \Theta) \land Linked(b, b', i, j, \Theta) \rightarrow I_{n,i,q}(j, a', b').$$

This means that in Θ' the congruence $\sigma_i^q(a, b)$ generates the same partition on each domain as in Θ''.

For the first claim, note that since the instance Θ' is not fragmented, due to Lemma 8 it follows that V^1 proves that for any $j \in V_{\mathcal{X}'}$ there exist $s < n$ and a path $\mathcal{P}_s$ connecting i and j. We go by the induction on the length of that path. For $s = 0$ we have $I_{0,i,g}(i, a, b)$, for $s = 1$ consider some k such that $E_{\mathcal{X}'}(i, k)$ (or $E_{\mathcal{X}'}(k, i)$). Since the instance is 1-consistent, there exist some $c, d \in D_i$, $c', d' \in D_k$ such that

$$E_{\mathcal{X}'}(i, k) \land E_{\ddot{\mathcal{A}}}^{ik}(c, c') \land E_{\ddot{\mathcal{A}}}^{ik}(d, d'),$$

and the only thing we have to check due to defining equation (86) is that there are no $c, d \in D_i, e \in D_k$ such that $\neg I_{0,i,g}(i, c, d)$ and

$$E_{\mathcal{X}'}(i, k) \land E_{\ddot{\mathcal{A}}}^{ik}(c, e) \land E_{\ddot{\mathcal{A}}}^{ik}(d, e).$$

It follows immediately from the fact that if such c, d, e exist, then $Linked(c, d, i, i, \Theta')$ and therefore $I_{0,i,q}(i, c, d)$ (the congruence $\sigma_i^q(a, b)$ contains $Linked_{[i,i,\Theta']}$). For the implication $s = t \rightarrow s = (t + 1)$, suppose that for every $k \in X'$ such that there exists a path of length t connecting i and k, there exist $j \in X'$, $c, d \in D_j$, $c', d' \in D_k$ such that $I_{t-1,i,g}(j, c, d)$, and all other conditions of (86) hold. Note that for $s = 0, 1$ we established $Linked(c, c', i, k, \Theta') \land Linked(d, d', i, k, \Theta')$, so we can assume that this is true for $s = t$ as well. Then use the same reasoning.

The first implication of claim 2 follows from the above. For the second implication we again use induction on the length of a path. For $s = 0, 1$ it follows from the definition of $I_{n,i,q}$. For the implication $s = t \rightarrow s = (t + 1)$ suppose that for every $k \in X'$ such that there exists a path of length t connecting i and k, for any $a, b \in D_i$ and any $a', b' \in D_k$ such that $I_{n,i,q}(i, a, b) \land Linked(a, a', i, k, \Theta) \land Linked(b, b', i, k, \Theta)$ we have $I_{n,i,q}(k, a', b')$. But since we can consider any path of length $(t + 1)$ as glued paths of length t and 1, the implication for $s = (t + 1)$ again follows straightaway from the definition of $I_{i,n,q}$. This completes the proof. $\qquad \square$

Lemma 11. V^1 *proves that* $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$ *has a solution only if* $\Theta_{ir} = (\mathcal{X}, \ddot{\mathcal{A}}_{ir})$ *has a solution.*

Proof. It is sufficient to show that if Θ has a solution, then Θ has a solution on domains $D_0, ..., D_{j-1}, D_j \backslash \{b_1, ..., b_s\}, D_{j+1}, ..., D_{n-1}$ after irreducibility reduction of one domain D_j. This is straightforward. Fix some i_0 and suppose that the maximal congruence $\sigma_{i_0}^q$ divides D_{i_0} to t equivalence classes. To make a reduction we consider some subgraph $\mathcal{X}'$ of digraph $\mathcal{X}$ containing vertex i_0 and such that it is connected and contains only vertices for which domains $D_{i_1}, ..., D_{i_g}$ congruence σ_{i_0} generates proper congruences. Since instance Θ is cycle-consistent, therefore for any s, t projection of $E_{\ddot{\mathcal{A}}}^{st}$ onto D_s, D_t Is subdirect. Thus, we construct a subinstance $\Theta_{prX'} = (\mathcal{X}', \ddot{\mathcal{A}})$ of instance Θ with the same target digraph with domains (and the same domain set), but with another input digraph $\mathcal{X}'$.

38

Suppose that there is a homomorphism from $\mathcal{X}$ to $\ddot{\mathcal{A}}$. For every $H \in \{\mathcal{X} \to \ddot{\mathcal{A}}\}$ define a new homomorphism $H \upharpoonright_{X'}$ from $\mathcal{X}'$ to $\ddot{\mathcal{A}}$ as follows:

$$\forall i \in \{i_0, i_1, ..., i_g\}, \, H \upharpoonright_{X'} (i) = \langle i, a \rangle \iff H(i) = \langle i, a \rangle. \tag{92}$$

That $H \upharpoonright_{X'}$ is a homomorphism follows right from the definition of H. Therefore, $\{H \upharpoonright_{X'}\} \subseteq \{\mathcal{X}' \to \ddot{\mathcal{A}}\}$. If for some $j \in \{i_0, i_1, ..., i_g\}$ and some $b_1, ..., b_s$ there is no homomorphism $H' \in \{\mathcal{X}' \to \ddot{\mathcal{A}}\}$ such that $H'(j) = \langle j, b_1 \rangle, ..., H'(j) = \langle j, b_s \rangle$, then no homomorphism from $\{\mathcal{X} \to \ddot{\mathcal{A}}\}$ sends j to $\langle j, b_1 \rangle, ..., \langle j, b_s \rangle$. $\qquad \square$

5.4.3 Weaker instance

When the algorithm runs the function CheckWeakerInstance it makes a copy of $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$ and simultaneously replaces every constraint in the instance with all weaker constraints *without dummy variables.* Then for every $i \in \{0, 1, ..., n-1\}$ it checks if the obtained weaker instance has a solution for $x_i = b$, for every $b \in D_i$ (by recursively calling the algorithm on a smaller domain). That is, the algorithm checks if the solution set to the weaker instance is subdirect. Suppose that the algorithm considers some i, set $D_i' = \emptyset$. It fixes the value $x_i = b$ and solves the weaker instance with domain set $D_0, ..., D_{i-1}, \{b\}, D_{i+1}, ..., D_{n-1}$. If there is a solution, then it adds b to D_i' and proceeds with another $b' \in D_i$. If there are solutions for all $b \in D_i$, the algorithm proceeds with $i + 1$. If for each $b \in D_i$ there is no solution, the algorithm answers that the initial instance has no solution. If there are some $b_1, ..., b_k \in D_i$ for which there is no solution to the weaker instance, the algorithm reduces domain D_i to $D_i' = D \backslash \{b_1, ..., b_s\}$, returns (x_i, D_i') and starts from the beginning.

Consider a cycle-consistent irreducible instance $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$. Any constraint in Θ is either a domain D_i for a variable i, or a relation $E_{\ddot{\mathcal{A}}}^{ij}$ for an edge $E_{\mathcal{X}}(i, j)$. Since Θ is cycle-consistent, projections $pr_i(E_{\ddot{\mathcal{A}}}^{ij})$ and $pr_j(E_{\ddot{\mathcal{A}}}^{ij})$ are equal to D_i, D_j. The algorithm never increases domains, so we weaken only binary constraints and replace each $E_{\ddot{\mathcal{A}}}^{ij}$ by two different types of weaker constraints:

1. D_i, D_j - weaker constraints of less arity;

2. All binary constraints from the list $\Gamma_{\mathcal{A}}$ containing $E_{\ddot{\mathcal{A}}}^{ij}$ except the full relation on $D_i \times D_j$.

Consider the intersection of all the above weaker constraints. Note that for any i we have the same domain D_i. We can lose some edges (i, j) from $E_{\mathcal{X}}$ (when the only binary relation containing $E_{\ddot{\mathcal{A}}}^{ij}$ is the full relation on $D_i \times D_j$) and can add some edges to $\ddot{\mathcal{A}}$. Let us denote the obtained weaker instance by $\Theta_{weak} = (\mathcal{X}_{weak}, \ddot{\mathcal{A}}_{weak})$.

Lemma 12. V^1 *proves that a CSP instance $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$ has a solution only if Θ has a solution after the weaker instance reduction.*

Proof. It is obvious that if instance Θ has a solution, then Θ_{weak} has a solution (we did not remove any edge or vertex from $\ddot{\mathcal{A}}$ and probably removed some edges from $\mathcal{X}$: just take the same homomorphism). That is, $\{\mathcal{X} \to \ddot{\mathcal{A}}\} \subseteq \{\mathcal{X}_{weak} \to \ddot{\mathcal{A}}_{weak}\}$.

Suppose that for some i there are $b_1, ..., b_s \in D_i$ such that there is no solution to Θ_{weak}, i.e. there is no homomorphism H in $\{\mathcal{X}_{weak} \to \ddot{\mathcal{A}}_{weak}\}$ such that $H(i) = \langle i, b_1 \rangle, ..., H(i) = \langle i, b_s \rangle$. It is needed to show that if Θ has a solution, then Θ has a solution on domains $D_0, ..., D_{i-1}, D_i \backslash \{b_1, ..., b_s\}, D_{i+1}, ..., D_{n-1}$. But it is trivial. $\qquad \square$

5.5 Linear case

In this section we will formalize and prove the soundness of the linear case of Zhuk's algorithm in the theory V^1 using $\Sigma_1^{1,b}$-induction.

5.5.1 Formalization of the linear case in V^1

For the linear case of Zhuk's algorithm, we need to define in V^1 some additional notions, namely finite abelian groups and matrices over finite fields.

To formalize the finite abelian group $\mathbb{Z}_p = \{0, 1, ..., p-1\}$ we define sum operation $+_{(mod\,p)}$ as follows:

$$c = a +_{(mod\,p)} b \longleftrightarrow c < p \wedge c \equiv (a+b)\,(mod\,p). \tag{93}$$

We define the identity element to be 0 and the inverse element for any $a < p$, denoted $-_{(mod\,p)}a$, to be $p \dot{-} a$. Furthermore, for any $m \in \mathbb{N}$ and any $a \in \mathbb{Z}_p$ we can define $\cdot_{(mod\,p)}$ as follows:

$$c = m \cdot_{(mod\,p)} a \longleftrightarrow c < p \wedge c \equiv (ma)\,(mod\,p). \tag{94}$$

For fields (when p is a prime number) we can also define the multiplicative inverse for any $a \neq 0, a \in \mathbb{Z}_p$, denoted by a^{-1}:

$$c = a^{-1} \longleftrightarrow c < p \wedge c \neq 0 \wedge c \cdot_{(mod\,p)} a = a \cdot_{(mod\,p)} c = 1. \tag{95}$$

It is clear that $+_{(mod\,p)}, -_{(mod\,p)}, \cdot_{(mod\,p)}$ and 0 can be defined in a weak subtheory of V^1 and satisfy all properties of a finite abelian group. A weak subtheory of V^1 can also define the multiplicative inverse modulo a prime and hence, in particular, V^1 proves that Z_p is a field. In our case, primes p_i are even fixed constants.

An $m \times n$ matrix A over $\mathbb{Z}_p$ is encoded by a relation $A(i, j, a)$, we write $A_{ij} = a$ for the corresponding entry. We will denote by $MX_{m \times n, p}(A)$ a relation that A is an $m \times n$ matrix over $\mathbb{Z}_p$. The sum of two $m \times n$ matrices A and B can be defined by a set-valued function

$$C = A + B \longleftrightarrow MX_{m \times n, p}(C) \wedge \forall i < m, j < n\ C_{ij} = A_{ij} +_{(mod\,p)} B_{ij}, \tag{96}$$

and the scalar multiplication bA of a number $b \in \mathbb{Z}_p$ and an $m \times n$ matrix A can be defined as:

$$C = bA \longleftrightarrow MX_{m \times n, p}(C) \wedge \forall i < m, j < n\ C_{ij} = b \cdot_{(mod\,p)} A_{ij}. \tag{97}$$

The definability of matrix addition and scalar multiplication in V^1 is obvious. Finally, to define the matrix multiplication, we will use the fact that V^1 defines the summation of long sums, i.e. if C is a function with domain $\{0, ..., n-1\}$, then V^1 defines the sum $\sum_{i<n} C(i)$ and proves its basic properties.

Indeed, consider $\Sigma_1^{1,b}$-induction on $t \leq n$, where t is the number of elements in formula

$$\phi(i, j, t, A, B) =_{def} \exists X < \langle t, p \rangle,\ X_0 = A_{i0} \cdot_{(mod\,p)} B_{0j} \wedge$$
$$\forall 0 < k < t\ X_k = X_{k-1} +_{(mod\,p)} A_{ik} \cdot_{(mod\,p)} B_{kj}. \tag{98}$$

Here X encodes the sequence of t partial sums, and by X_k we denote $X(k)$. For $t = 1$, $\phi(i, j, t, A, B)$ is true (since $\cdot_{(mod\,p)}$ is definable in V^1), and $\phi(i, j, t+1, A, B)$ follows from $\phi(i, j, t, A, B)$ since $+_{(mod\,p)}$ is also definable in V^1. This uses $\Sigma_1^{1,b}$ induction.

We can thus define the multiplication of an $m \times n$ matrix A and an $n \times s$ matrix B as follows:

$$C = AB \longleftrightarrow MX_{m \times s, p}(C) \wedge \forall i < m, j < s$$
$$C_{ij} = A_{i0} \cdot_{(mod\, p)} B_{0j} +_{(mod\, p)} ... +_{(mod\, p)} A_{i(n-1)} \cdot_{(mod\, p)} B_{(n-1)j}. \qquad (99)$$

We will further use notation $+, -, \cdot$ instead of $+_{(mod\, p)}, -_{(mod\, p)}$ and $\cdot_{(mod\, p)}$ since it does not lead to confusion.

5.5.2 Soundness of the linear case in V^1

We will call an instance $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$, produced by the algorithm before the linear case, the initial instance. As the first modification of the instance, we need to define a factorized instance Θ_L: at this step, we change the target digraph $\ddot{\mathcal{A}}$ and do not change instance digraph $\mathcal{X}$. The algorithm factorizes each domain separately and due to the assumption for every domain D_i there is the minimal linear congruence σ_i such that D_i/σ_i is isomorphic to linear algebra. Denote by $\sigma < nl^2$ the set representing all congruences σ_i, $\sigma(i, a, b) \iff \sigma_i(a, b)$. The factorized target digraph with domains $\ddot{\mathcal{A}}_L$ can be represented as an $(n+2)$-tuple $(V_{\ddot{\mathcal{A}}_L}, E_{\ddot{\mathcal{A}}_L}, D_0/\sigma_0, ..., D_{n-1}/\sigma_{n-1})$, where $V_{\ddot{\mathcal{A}}_L} < \langle n, l \rangle$, $V_{\ddot{\mathcal{A}}_L}(i, a) \iff \mathcal{D}/\sigma_i(a)$ and $E_{\ddot{\mathcal{A}}_L}$ such that

$$E_{\ddot{\mathcal{A}}_L}(s, r) \iff \exists i, j < n \, \exists a, b < l, \ s = \langle i, a \rangle \wedge r = \langle j, b \rangle \wedge$$
$$D_i/\sigma_i(a) \wedge D_j/\sigma_j(b) \wedge (\exists c, d < l, \ \sigma(i, a, c) \wedge \sigma(j, b, d) \wedge E_{\ddot{\mathcal{A}}}^{ij}(c, d)). \qquad (100)$$

In words, there is an edge between elements a, b representing classes $[a]/\sigma_i$ and $[b]/\sigma_j$ in $\ddot{\mathcal{A}}_L$ any time $E_{\ddot{\mathcal{A}}}^{ij} \cap [a]/\sigma_i \times [b]/\sigma_j \neq \emptyset$. In the factorized target digraph constructed in such a way, we actually can lose some edges (for example, when we glue all edges between elements in $[a]/\sigma_i$ and $[b]/\sigma_j$ in one edge), but we also can get new solutions (for example, when we get new cycles). We thus increase the set of solutions by simplifying the structure of the target digraph with domains.

Theorem 12. V^1 *proves that an instance* $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$ *has a solution only if* $\Theta_L = (\mathcal{X}, \ddot{\mathcal{A}}_L)$ *has a solution.*

Proof. Consider a CSP instance $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$ with $V_{\mathcal{X}} = n$, $V_{\ddot{\mathcal{A}}} < \langle n, l \rangle$. Suppose that the instance has a solution, i.e. there exists a homomorphism H from $\mathcal{X}$ to $\ddot{\mathcal{A}}$. Construct the factorized instance as mentioned above.

We first construct the canonical homomorphism H_c between the target digraph $\ddot{\mathcal{A}}$ and the factorized digraph $\ddot{\mathcal{A}}_L$, and then show that there is a homomorphism from $\mathcal{X}$ to $\ddot{\mathcal{A}}_L$. Define H_c as follows: for every $u \in V_{\ddot{\mathcal{A}}}$, and every $v \in V_{\ddot{\mathcal{A}}_L}$

$$H_c(u, v) \iff \exists i < n, a, b < l, \ u = \langle i, a \rangle, \ v = \langle i, b \rangle \wedge \sigma(i, b, a) \wedge D_i/\sigma_i(b).$$

That is, we send a vertex a to a vertex b in $\ddot{\mathcal{A}}_L$ in the factorized domain D_i/σ_i if and only if $b \in D_i$, b and a are in the same congruence class under σ_i, and b is a represent of the class a/σ_i (the smallest element). This set exists due to $\Sigma_0^{1,b}$-comprehension axiom. Moreover, it satisfies the relation of being a well-defined map between two sets $V_{\ddot{\mathcal{A}}}$ and $V_{\ddot{\mathcal{A}}_L}$. The existence of b is ensured by the property of congruence relation σ_i (reflexivity), and the uniqueness by our choice of representation of the factor set by the minimal element in the class. It is left to show that

$$\forall u_1, u_1, v_1, v_2 < \langle n, l \rangle (E_{\ddot{\mathcal{A}}}(u_1, u_2) \wedge Z_c(u_1, v_1) \wedge Z_c(u_2, v_2) \rightarrow E_{\ddot{\mathcal{A}}_L}(v_1, v_2)),$$

41

but this follows straightforwardly from the definition of H_c and $E_{\ddot{\mathcal{A}}_L}$. Finally, to construct a homomorphism from $\mathcal{X}$ to $\mathcal{A}_L$, consider set $H' < \langle n, \langle n, l \rangle \rangle$ such that

$$H'(i) = v \iff \exists u < \langle n, l \rangle (H(i) = u \wedge H_c(u) = v).$$

It is easy to check that set H' satisfies the homomorphism relation between digraphs $\mathcal{X}$ and $\ddot{\mathcal{A}}_L$. Thus, there is a solution to the factorized instance Θ_L. $\qquad\square$

Suppose that there is a solution set to the instance Θ, the set of homomorphisms from $\mathcal{X}$ to $\ddot{\mathcal{A}}$, denoted by $\{\mathcal{X} \to \ddot{\mathcal{A}}\} = \{H_1, H_2, ..., H_s\}$. We will call the set of all homomorphisms, constructed from $H_1, ..., H_s$ by canonical homomorphisms H_c the solution set to Θ factorized by congruences, denoted by $\{\mathcal{X} \to \ddot{\mathcal{A}}\}/\Sigma = \{H'_1, ..., H'_s\}$ (some of the homomorphisms $H'_1, ..., H'_s$ can be equivalent).

By the previous theorem, we established that Θ_L has a solution only if Θ does. Now to find solutions to Θ_L we will use the translation of constraints into a system of linear equations (we suppose that this translation is included in the algorithm's transcription) and run Gaussian Elimination. We thus need to show in V^1 that this process does not reduce the solution set to Θ_L. Let us recall that a matrix A is in the *row echelon form* if it is either a zero matrix or its first non-zero entry of row $i+1$ must be on the right of the first non-zero entry of row i, and these entries must be 1. Consider the system of linear equations $A\bar{x} = \bar{b}$ for an $m \times n$ matrix A. Suppose that we have a sequence of $m \times (n+1)$ matrices $[A_0|B_0], [A_1|B_1], ..., [A_t|B_t]$, where $[A_0|B_0]$ is the original augmented matrix of the system of linear equations, $[A_t|B_t]$ is a matrix in the row echelon form and every next matrix is obtained from the previous one by one of the elementary row operations. Since every elementary row operation can be simulated by left multiplication by an elementary matrix, instead of defining elementary row operations, we define elementary matrices in V^1.

We say that an $m \times m$ matrix E is elementary if E satisfies one of the following three relations. The first of them corresponds to row-switching transformations

$$\begin{aligned}
EL^I_{m \times m, p}(E) &\iff MX_{m \times m, p}(E) \wedge \exists i' \neq j' < m \forall i, j < m \\
(i \neq i' &\wedge i \neq j' \to E_{ii} = 1) \wedge (i \neq i' \wedge j \neq j' \wedge i \neq j \to E_{ij} = 0) \\
&\wedge (E_{i'i'} = 0 \wedge E_{j'j'} = 0 \wedge E_{i'j'} = 1 \wedge E_{j'i'} = 1),
\end{aligned} \qquad (101)$$

the second one corresponds to row-multiplying transformations

$$\begin{aligned}
EL^{II}_{m \times m, p}(E) &\iff MX_{m \times m, p}(E) \wedge \exists a \neq 0 \in \mathbb{Z}_p \exists i' < m \\
\forall i, j < m(i \neq i' &\to E_{ii} = 1) \wedge (i \neq j \to E_{ij} = 0) \wedge E_{i'i'} = a,
\end{aligned} \qquad (102)$$

and the last one corresponds to row-addition transformations

$$\begin{aligned}
EL^{III}_{m \times m, p}(E) &\iff MX_{m \times m, p}(E) \wedge \exists a \neq 0 \in \mathbb{Z}_p \exists i', j' < m \forall i, j < m \\
(E_{ii} = 1 &\wedge (i \neq j \wedge i \neq i' \wedge j \neq j' \to E_{ij} = 0) \wedge E_{i'j'} = a.
\end{aligned} \qquad (103)$$

Let us denote these elementary matrices by T^1, T^2, T^3. If we consider matrix $[A|B]$, then matrices $T^1[A|B]$, $T^2[A|B]$ and $T^3[A|B]$ are matrices produced from $[A|B]$ by elementary row operations. Since V^1 can define long sums it is easy to show that V^1 proves that each of elementary row operations preserves the solution set to $A\bar{x} = \bar{b}$.

Lemma 13. V^1 *proves that for every matrix $[A|B]$ there is a row-echelon matrix $[A'|B']$ having the same solution set.*

Proof. Use $\Sigma_1^{1,b}$-induction. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ $\square$

Suppose now that we have established the solution set to the factorized instance Θ_L, $\{\mathcal{X} \to \ddot{\mathcal{A}}_L\}$, and assume that $\{\mathcal{X} \to \ddot{\mathcal{A}}\}/\Sigma \subsetneq \{\mathcal{X} \to \ddot{\mathcal{A}}_L\}$. We will further proceed with iterative steps of the algorithm, the first iteration (see Subsection 4.2). We arbitrarily choose a constraint $E_{\mathcal{X}}(i,j)$ and replace it with all weaker constraints without dummy variables, making the initial instance weaker. It can be done either by adding some edges to the relation $E_{\ddot{\mathcal{A}}}^{ij}$ (note that new edges have to be preserved by WNU operation Ω) or by removing the edge (i,j) from $\mathcal{X}$ (when the only relation containing $E_{\ddot{\mathcal{A}}}^{ij}$ is the full relation on $D_i \times D_j$). Without loss of generality, suppose that we start with $\mathcal{X}$. We prove the following theorem by induction on the number of edges removed from $\mathcal{X}$. The process of removing can be interrupted by modifications of $\ddot{\mathcal{A}}$ as well, but since this interruption happens only the constant number of times (the number of edges we can add to $\ddot{\mathcal{A}}$ is a constant), we can consider the constant number of separate inductions as one from start to the end.

Theorem 13. *Consider two CSP instances, the initial instance $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$ and the factorized instance $\Theta_L = (\mathcal{X}, \ddot{\mathcal{A}}_L)$, and suppose that the solution set to the initial instance factorized by congruences is a proper subset of the solution set to the factorized instance, i.e. $\{\mathcal{X} \to \ddot{\mathcal{A}}\}/\Sigma \subsetneq \{\mathcal{X} \to \ddot{\mathcal{A}}_L\}$.*

Then V^1 proves that there exists a subsequence of instance digraphs $\mathcal{X} = \mathcal{X}_0, ..., \mathcal{X}_t$ (and a subsequence of target digraphs with domains $\ddot{\mathcal{A}} = \ddot{\mathcal{A}}_0, ..., \ddot{\mathcal{A}}_s$), where $t \leq n(n-1)$ is the number of edges removed from $\mathcal{X}$, $\{\mathcal{X}_t \to \ddot{\mathcal{A}}_s\}/\Sigma \neq \{\mathcal{X} \to \ddot{\mathcal{A}}_L\}$, and if one removes any other edge from $\mathcal{X}_t$, every solution to Θ_L will be a solution to $\{\mathcal{X}_{t+1} \to \ddot{\mathcal{A}}_s\}/\Sigma$.

Proof. Since $\{\mathcal{X} \to \ddot{\mathcal{A}}\}/\Sigma \subsetneq \{\mathcal{X} \to \ddot{\mathcal{A}}_L\}$, there is some point $(a_1, ..., a_k)$ in free variables $y_1, ..., y_k$ such that $\phi(a_1, ..., a_k)$ is a solution to Θ_L, but if we restrict domains $D_0, ..., D_{n-1}$ of Θ to congruences blocks corresponding to $\phi(a_1, ..., a_k)$, there is no solution to Θ. Thus, there is some homomorphism H_L from $\mathcal{X}$ to $\ddot{\mathcal{A}}_L$ such that for any well-defined map H from $\mathcal{X}$ to $\ddot{\mathcal{A}}$, where every x_i is mapped to the corresponding domain D_i and $H_L = H \circ H_c$, there exists an edge $E_{\mathcal{X}}(i_1, i_2)$ in $\mathcal{X}$ that failed to be mapped into an edge in $\ddot{\mathcal{A}}$. The theory V^1 can count the number of elements in every set. Denote by $q = \#E_{\mathcal{X}}$ the number of edges in $\mathcal{X}$, $q \leq n^2$. Consider the following formula $\theta(t)$,

$$\theta(t) =_{def} \exists H_L < \langle n, \langle n, l \rangle \rangle, MAP(V_{\mathcal{X}}, n, V_{\ddot{\mathcal{A}}_L}, \langle n, l \rangle, H_L) \wedge$$
$$\wedge (\forall i < n, w < \langle n, l \rangle,\ H_L(i) = w \to \exists a < l, w = \langle i, a \rangle \wedge D_i/\sigma_i(a)) \wedge$$
$$\wedge \forall i_1, i_2 < n, \forall w_1, w_2 < \langle n, l \rangle$$
$$(E_{\mathcal{X}}(i_1, i_2) \wedge H_L(i_1) = w_1 \wedge H_L(i_2) = w_2 \to E_{\ddot{\mathcal{A}}_L}(w_1, w_2))$$
$$\wedge$$
$$\forall i, j < n,\ E_{\mathcal{X}_t}(i,j) \to E_{\mathcal{X}}(i,j) \wedge (q - t) \leq \#E_{\mathcal{X}_t}(i,j) \wedge$$
$$\wedge \forall u, v < \langle n, l \rangle,\ E_{\ddot{\mathcal{A}}}(u,v) \to E_{\ddot{\mathcal{A}}_s}(u,v)$$
$$\wedge$$
$$MAP(V_{\mathcal{X}}, n, V_{\ddot{\mathcal{A}}}, \langle n, l \rangle, H) \wedge \forall i < n, w < \langle n, l \rangle$$
$$H(i) = w \to \exists a < l, w = \langle i, a \rangle \wedge D_i(a)$$
$$\wedge$$
$$\forall i < n, v < \langle n, l \rangle,\ H_L(i) = v \longleftrightarrow \exists u < \langle n, l \rangle (H(i) = u \wedge H_c(u) = v)$$
$$\implies$$

$$\exists i_1, i_2 < n, \exists w_1, w_2 < \langle n, l \rangle, \neg(E_{\mathcal{X}_t}(i_1, i_2) \wedge H(i_1) = w_1 \wedge H(i_2) = w_2 \rightarrow$$
$$\rightarrow E_{\ddot{\mathcal{A}}_s}(w_1, w_2)).$$

The first part of the formula expresses that there is a homomorphism H_L from $\mathcal{X}$ to $\ddot{\mathcal{A}}_L$. The second part formalizes that the input digraph $\mathcal{X}_t$ is constructed from $\mathcal{X}$ by removing at least t edges (and the target digraph $\ddot{\mathcal{A}}_s$ is constructed from $\ddot{\mathcal{A}}$ by adding some edges). The third and fourth parts say that there is a well-defined map H from $V_{\mathcal{X}}$ to $V_{\ddot{\mathcal{A}}}$ satisfying all restrictions on domains and such that H_L is a composition of H and the canonical homomorphism H_c. And the last part expresses that if all previous conditions are true, then H cannot be a homomorphism from $\mathcal{X}_t$ to $\ddot{\mathcal{A}}_s$.

In the formula $\theta(t)$ as fixed parameters we use $\mathcal{X} = (V_{\mathcal{X}}, E_{\mathcal{X}})$, $q = \#E_{\mathcal{X}}$, the target digraph with domains $\ddot{\mathcal{A}} = (V_{\ddot{\mathcal{A}}}, E_{\ddot{\mathcal{A}}})$, $V_{\ddot{\mathcal{A}}} < \langle n, l \rangle$ and $\#E_{\ddot{\mathcal{A}}} < \langle n, l \rangle^2$, the factorized digraph with domains $\ddot{\mathcal{A}}_L = (V_{\ddot{\mathcal{A}}_L}, E_{\ddot{\mathcal{A}}_L})$ and the canonical homomorphism H_c. Induction goes on variables t and the instance digraph $\mathcal{X}_t = (V_{\mathcal{X}}, E_{\mathcal{X}_t})$ such that $(q - t) = \#E_{\mathcal{X}_t}$. Finally, witnesses in $\Sigma_1^{1,b}$-induction corresponding to t are the target digraph with domains $\ddot{\mathcal{A}}_s = (V_{\ddot{\mathcal{A}}_s}, E_{\ddot{\mathcal{A}}_s})$ and the map H from $V_{\mathcal{X}}$ to $V_{\ddot{\mathcal{A}}}$.

By assumption, the formula $\theta(t)$ is true for $t = 0$. We also know that it is false for $t = q$ since for all $i_1, i_2 < n$ there is $\neg E_{\mathcal{X}_t}(i_1, i_2)$. Since $\theta(t)$ is $\Sigma_1^{1,b}$-formula, we can use the Number maximization axiom:

$$\forall H \leq \langle n, \langle n, l \rangle \rangle, \forall \ddot{\mathcal{A}}_s, [\theta(0) \rightarrow \exists q' \leq q(\theta(q') \wedge \neg \exists q'' \leq q(q' < q'' \wedge \theta(q'')))].$$

This completes the proof. $\qquad \square$

Lemma 14. *Consider two CSP instances, the initial instance $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$ and the instance $\Theta_{t,s} = (\mathcal{X}_t, \ddot{\mathcal{A}}_s)$, where $t \leq n(n-1)$ is the number of edges removed from the initial digraph $\mathcal{X}$ and $s \leq \langle n, l \rangle^2$ is the number of edges added to the target digraph $\ddot{\mathcal{A}}$. V^1 proves that instance Θ has a solution only if $\Theta_{t,s}$ has a solution.*

Proof. Suppose that there is a solution to the instance Θ, a homomorphism H, and the instance $\Theta_{t,s}$ is constructed from Θ by removing t arbitrary edges from $\mathcal{X}$ and adding some s edges to $\ddot{\mathcal{A}}$. Then it is straightforward to check that H is also a solution to $\Theta_{t,s}$. $\qquad \square$

For further iterations of Zhuk's algorithm, we will prove the following theorem.

Theorem 14. *Consider two CSP instances, the initial instance $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$ and the instance $\Theta_{t,s} = (\mathcal{X}_t, \ddot{\mathcal{A}}_s)$, where $t \leq n(n-1)$ is the number of edges removed from the initial digraph $\mathcal{X}$ and $s \leq \langle n, l \rangle^2$ is the number of edges added to the target digraph with domains $\ddot{\mathcal{A}}$. Suppose that the solution set to the initial instance factorized by congruences is a proper subset of the intersection of the solution set to the instance $\Theta_{t,s}$ factorized by congruences and the solution set to the factorized instance Θ_L, i.e. $\{\mathcal{X} \rightarrow \ddot{\mathcal{A}}\}/\Sigma \subsetneq \{\mathcal{X}_t \rightarrow \ddot{\mathcal{A}}_s\}/\Sigma \cap \{\mathcal{X} \rightarrow \ddot{\mathcal{A}}_L\}$.*

Then V^1 proves that there exists a subsequence of instance digraphs $\mathcal{X} = \mathcal{X}_0, ..., \mathcal{X}_r$ (and a subsequence of target digraphs with domains $\ddot{\mathcal{A}} = \ddot{\mathcal{A}}_0, ..., \ddot{\mathcal{A}}_f$), where $r \leq n(n-1)$ is the number of edges removed from $\mathcal{X}$ such that $\{\mathcal{X}_r \rightarrow \ddot{\mathcal{A}}_f\}/\Sigma \neq \{\mathcal{X}_t \rightarrow \ddot{\mathcal{A}}_s\}/\Sigma \cap \{\mathcal{X} \rightarrow \ddot{\mathcal{A}}_L\}$ and if one removes any other edge from $\mathcal{X}_r$, every solution to $\{\mathcal{X}_t \rightarrow \ddot{\mathcal{A}}_s\}/\Sigma \cap \{\mathcal{X} \rightarrow \ddot{\mathcal{A}}_L\}$ will be a solution to $\{\mathcal{X}_{r+1} \rightarrow \ddot{\mathcal{A}}_f\}/\Sigma$.

Proof. The proof is analogous to the proof of Theorem 13. Let us define a slightly modified formula $\theta'(r)$. We now consider two homomorphisms, H_L from $\mathcal{X}$ to $\ddot{\mathcal{A}}_L$, and $H_{t,s}$ from $\mathcal{X}_t$ to $\ddot{\mathcal{A}}_s$ such that H_L is a composition of $H_{t,s}$ and canonical homomorphism H_c (it is

44

equivalent to the condition that solutions to both instances are in $\{\mathcal{X}_t \to \ddot{\mathcal{A}}_s\}/\Sigma \cap \{\mathcal{X} \to \ddot{\mathcal{A}}_L\})$.

$$\theta(r) =_{def} \exists H_L < \langle n, \langle n, l \rangle \rangle, MAP(V_{\mathcal{X}}, n, V_{\ddot{\mathcal{A}}_L}, \langle n, l \rangle, H_L) \wedge$$
$$\wedge (\forall i < n, w < \langle n, l \rangle, \ H_L(i) = w \to \exists a < l, w = \langle i, a \rangle \wedge D_i/\sigma_i(a)) \wedge$$
$$\wedge \forall i_1, i_2 < n, \forall w_1, w_2 < \langle n, l \rangle,$$
$$(E_{\mathcal{X}}(i_1, i_2) \wedge H_L(i_1) = w_1 \wedge H_L(i_2) = w_2 \to E_{\ddot{\mathcal{A}}_L}(w_1, w_2))$$
$$\wedge$$
$$\exists H_{t,s} < \langle n, \langle n, l \rangle \rangle (MAP(V_{\mathcal{X}_t}, n, V_{\ddot{\mathcal{A}}_s}, \langle n, l \rangle, H_{t,s}) \wedge$$
$$\wedge (\forall i < n, w < \langle n, l \rangle \ H_{t,s}(i) = w \to \exists a < k, w = \langle i, a \rangle \wedge D_i(a)) \wedge$$
$$\wedge \forall i_1, i_2 < n, \forall w_1, w_2 < \langle n, l \rangle$$
$$(E_{\mathcal{X}_t}(i_1, i_2) \wedge H_{t,s}(i_1) = w_1 \wedge H_{t,s}(i_2) = w_2 \to E_{\ddot{\mathcal{A}}_s}(w_1, w_2))$$
$$\wedge$$
$$\forall i < n, v < \langle n, l \rangle \ H_L(i) = v \longleftrightarrow \exists u < \langle n, k \rangle (H_{t,s}(i) = u \wedge H_c(u) = v)$$
$$\wedge$$
$$\forall i, j < n \ E_{\mathcal{X}_r}(i, j) \to E_{\mathcal{X}}(i, j) \wedge (q - r) \le \#E_{\mathcal{X}_r}(i, j) \wedge$$
$$\wedge \forall u, v < \langle n, l \rangle \ E_{\ddot{\mathcal{A}}}(u, v) \to E_{\ddot{\mathcal{A}}_f}(u, v)$$
$$\wedge$$
$$MAP(V_{\mathcal{X}}, n, V_{\ddot{\mathcal{A}}}, \langle n, l \rangle, H) \wedge \forall i < n, w < \langle n, l \rangle$$
$$H(i) = w \to \exists a < l, w = \langle i, a \rangle \wedge D_i(a)$$
$$\wedge$$
$$\forall i < n, v < \langle n, l \rangle \ Z_L(i) = v \longleftrightarrow \exists u < \langle n, l \rangle (H(i) = u \wedge H_c(u) = v)$$
$$\implies$$
$$\exists i_1, i_2 < n, \exists w_1, w_2 < \langle n, l \rangle \ \neg(E_{\mathcal{X}_r}(i_1, i_2) \wedge H(i_1) = w_1 \wedge H(i_2) = w_2$$
$$\to E_{\ddot{\mathcal{A}}_f}(w_1, w_2)).$$

In formula $\theta'(r)$ as fixed parameters we use parameters similar to parameters in the formula $\theta(t)$, but add here $\mathcal{X}_t = (V_{\mathcal{X}}, E_{\mathcal{X}_t})$, $q - t = \#E_{\mathcal{X}_t}$ and $\mathcal{A}_s = (V_{\mathcal{A}_s}, E_{\mathcal{A}_s})$, $s < \langle n, l \rangle^2$ as well. Induction goes on variable r and the instance digraph $\mathcal{X}_r = (V_{\mathcal{X}}, E_{\mathcal{X}_r})$ such that $(q - r) \le \#E_{\mathcal{X}_r}$. Witnesses to the induction are the target digraph with domains $\ddot{\mathcal{A}}_f = (V_{\ddot{\mathcal{A}}_f}, E_{\ddot{\mathcal{A}}_f})$ and the map H from $V_{\mathcal{X}}$ to $V_{\ddot{\mathcal{A}}}$. $\qquad\square$

5.6 The main result

Theorem 15 (The main result). *For any fixed relational structure $\mathcal{A}$ which corresponds to an algebra with WNU operation and therefore leads to p-time solvable CSP, the theory $V_{\mathcal{A}}^1$ proves the soundness of Zhuk's algorithm.*

Proof. Consider any unsatisfiable CSP instance $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$. It is sufficient to show that in the computation $W = (W_1, W_2, ..., W_k)$ of the algorithm on $\mathcal{X}$, for all possible types of algorithmic modifications the theory $V_{\mathcal{A}}^1$ proves that W_i has a solution only if W_{i+1} has a solution.

In Section 5.4 we have shown that V^1 proves that:

- the instance Θ has a solution only if it has a solution after cycle-consistency reduction (Lemma 6);

- the instance Θ has a solution only if it has a solution after irreducible reduction (Lemma 11);

- the instance Θ has a solution only if it has a solution after the weaker instance reduction (Lemma 12).

The three universal algebra axiom schemes $\text{BA}_{\mathcal{A}}$-axioms, $\text{CR}_{\mathcal{A}}$-axioms, and $\text{PC}_{\mathcal{A}}$-axioms defined in Section 5.2 by $\forall \Sigma_2^{1,b}$-formulas validate universal algebra reductions of any domain D_i to a binary absorbing subuniverse, central subuniverse or to an arbitrary equivalence class of polynomially complete congruence on D_i.

Finally, in Section 5.5 we have shown that V^1 validates:

- factorization of the instance by minimal linear congruences (Theorem 12);

- Gaussian elimination (Lemma 13);

- decreasing of the solution set to the factorized instance (Theorems 13, 14, Lemma 14).

This completes the proof. $\qquad\square$

The result implies that tautologies $\neg HOM(\mathcal{X}, \mathcal{A})$ for negative instances of any fixed p-time CSP have short proofs in any propositional proof system simulating Extended Resolution and a theory that proves the three universal algebra axioms.

6   Conclusion notes

In the paper we investigate the proof complexity of general CSP. We proved the soundness of Zhuk's algorithm in a new theory of bounded arithmetic defined by augmenting the two-sorted theory V^1 with three universal algebra axioms. These axioms are designed to verify universal algebra reductions, while the soundness of consistency reductions and the linear case of the algorithm is proved directly in the theory V^1.

Consistency reductions open the algorithm and represent its most technical part. Formalization of the consistency reductions uses iteratively defined sets and $\Sigma_1^{1,b}$-induction. The linear case is the last step of Zhuk's algorithm after all reductions of separate domains. However, it does not lead to linear equations straightforwardly: structures in the linear case have to be factorized first. The proof of the soundness of the linear case is based on the formalization of Gaussian elimination and linear factorization and uses $\Sigma_1^{1,b}$-induction.

In contrast, universal algebra axioms stand apart. Despite the fact that they can be defined by $\forall \Sigma_2^{1,b}$-formulas, their proof in a theory of bounded arithmetic requires the formalization of advanced notions from universal algebra and this will be a subject of further research.

Theorem 5 allows one to consider constraint languages with at most binary relations instead of general CSP. We tested how to utilize the framework and strategy of getting short propositional proofs using bounded arithmetic in [14] on an elementary example of undirected graphs (the $\mathcal{H}$-coloring problem). In that case, the theory of bounded arithmetic corresponds to a weak proof system $R^*(log)$, a mild extension of resolution.

Every theory of bounded arithmetic corresponds to some propositional proof system. The theory V^1 stands for polynomial time reasoning and corresponds to the Extended Frege EF proof system (equivalently Extended resolution ER). Our working hypothesis is that the soundness of Zhuk's algorithm can be established utilizing only $\Sigma_1^{1,b}$-induction.

If it is true, then statements $\neg HOM(\mathcal{X}, \ddot{\mathcal{A}})$ for unsatisfiable instances of polynomial time $CSP(\mathcal{A})$ will have short propositional proofs in EF. The next step in our program is to investigate the boundaries of the theory V^1 in formalizing of universal algebra notions.

References

[1] Albert Atserias and Joanna Ochremiak. Proof complexity meets algebra. *ACM Trans. Comput. Logic*, 20(1):1–46, December 2018.

[2] Libor Barto. Constraint satisfaction problem and universal algebra. *ACM SIGLOG News*, 1(2):14–24, oct 2014.

[3] 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.

[4] Clifford Bergman. *Universal algebra: Fundamentals and selected topics.* Chapman and Hall/CRC, 2011.

[5] Zarathustra Brady. Notes on csps and polymorphisms, 2022.

[6] Andrei A. Bulatov. H-coloring dichotomy revisited. *Theoretical Computer Science*, 349(1):31 – 39, 2005. Graph Colorings.

[7] Andrei A. Bulatov. A dichotomy theorem for nonuniform csps. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 319–330, 2017.

[8] Stanley Burris and Hanamantagouda P. Sankappanavar. *A Course in Universal Algebra.* Springer, 1981.

[9] Stephen A. Cook and Phuong Nguyen. *Logical Foundations of Proof Complexity.* Cambridge University Press, USA, 1st edition, 2010.

[10] Jack Edmonds. Systems of distinct representatives and linear algebra. *J. Res. Nat. Bur. Standards Sect. B*, 71(4):241–245, 1967.

[11] Tomás Feder and Moshe Y. Vardi. Monotone monadic snp and constraint satisfaction. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '93, page 612–622, New York, NY, USA, 1993. Association for Computing Machinery.

[12] Ralph Freese and Ralph Mckenzie. Commutator theory for congruence modular varieties. 125, 01 1987.

[13] L. Fuchs, J.P. Kahane, A.P. Robertson, and S. Ulam. *Abelian Groups.* ISSN. Elsevier Science, 2014.

[14] Azza Gaysin. H-Colouring Dichotomy in Proof Complexity. *Journal of Logic and Computation*, 31(5):1206–1225, 04 2021.

[15] Jan Krajicek. *Bounded Arithmetic, Propositional Logic and Complexity Theory.* Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1995.

[16] Jan Krajíček. *Proof Complexity.* Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2019.

[17] Miklós Maróti and Ralph Mckenzie. Existence theorems for weakly symmetric operations. *Algebra Universalis*, 59:463–489, 12 2008.

[18] Dmitriy Zhuk. On key relations preserved by a weak near-unanimity function. In *Proceedings of the 2014 IEEE 44th International Symposium on Multiple-Valued Logic*, ISMVL '14, page 61–66, USA, 2014. IEEE Computer Society.

[19] Dmitriy Zhuk. A proof of the csp dichotomy conjecture. *J. ACM*, 67(5):1–78, August 2020.

[20] Dmitriy Zhuk. Strong subalgebras and the constraint satisfaction problem. *J. Multiple Valued Log. Soft Comput.*, 36(4-5):455–504, 2021.

[21] Stanislav Živný. *The Complexity of Valued Constraint Satisfaction Problems.* Springer Publishing Company, Incorporated, 2012.