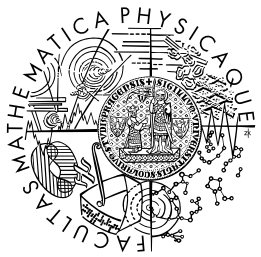

FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

DOCTORAL THESIS

Azza Gaysin

Proof complexity of CSP

Department of Algebra

Supervisor of the doctoral thesis: Prof. RNDr. Jan Krajíček, DrSc.

Study programme: Mathematics

Study branch: Algebra, Theory of Numbers and Mathematical Logic

Prague 2023

I declare that I carried out this doctoral thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In ..... date ....

Author's signature

I would like to express my gratitude to Jan Krajíček, the best supervisor I could ever have, for his wise guidance, a great deal of encouragement, and unwavering optimism. I thank Michael Kompatscher for his patience and a number of helpful discussions on universal algebra. I wish to thank Emil Jeřábek, Pavel Pudlák, Neil Thapen, Moritz Müller, and Albert Atserias for their expert comments and inspiration. I am grateful to my friends, my mother, and my partner for their constant support and belief in me.

Title: Proof complexity of CSP

Author: Azza Gaysin

Department: Department of Algebra

Supervisor: Prof. RNDr. Jan Krajíček, DrSc., Department of Algebra

Abstract:

In this thesis we formalize Zhuk's decision algorithm solving in p-time tractable constraint satisfaction problems (CSPs) in a weak theory of bounded arithmetic W_1^1. As a consequence, we show that tautologies that express the negative instances of such CSPs have polynomial proofs in the quantified propositional calculus G.

Keywords: bounded arithmetic, constraint satisfaction problem, proof complexity, universal algebra

Contents

Introduction

Constraint satisfaction problems (CSPs) form a wide class of decision problems, first studied in 1998 by Feder and Vardi in their attempt to discover a large subclass of NP that exhibits a dichotomy [4]. The problem CSP(Γ) consists of a finite set D and a finite collection $\Gamma = \{R_1, ..., R_n\}$ of relations on D, or constraint language. The question is, given as input a list of variables V and a list of constraints $\mathcal{C} = \{C_1, ..., C_m\}$ - pairs of tuples of distinct variables and relations R_i, whether there is an assignment of variables to values in D satisfying the given constraints. If such an assignment exists, an instance is called satisfiable and unsatisfiable otherwise. The equivalent definition is formulated as a homomorphism problem between relational structures. The obvious example is the $\mathcal{H}$-coloring problem, where $\mathcal{H}$ is a simple undirected graph without loops, whose vertices are considered as different colors. The $\mathcal{H}$-coloring of a graph $\mathcal{G}$ is an assignment of colors to the vertices of $\mathcal{G}$ such that adjacent vertices of $\mathcal{G}$ obtain adjacent colors. The generalization of this problem is a homomorphism problem from an input directed graph to a fixed target digraph. It is known that the latter problem is universal: for any constraint language Γ, CSP(Γ) is logspace equivalent to a homomorphism problem for some digraph $\mathcal{H}$ [1].

The first dichotomy result was formulated by Schaefer in 1978 for a problem over a binary domain called Generalized Satisfiability [8]. The second result of this form was proved by Hell and Nešetřil in 1990 for the $\mathcal{H}$-coloring problem [6]. Feder and Vardi conjectured that the whole class of CSPs satisfies the general dichotomy between P and NP [4], i.e. for a given Γ the problem is either in P or is NP-complete. The proof of this conjecture was presented in 2017 by Zhuk [10] and Bulatov [2].

Zhuk's algorithm solves the problem of whether there exists a homomorphism from $\mathcal{X}$ to $\mathcal{A}$ for any tractable CSP($\mathcal{A}$) in polynomial time in size of $\mathcal{X}$, for any fixed $\mathcal{A}$. If an instance is satisfiable, then the algorithm produces a solution, i.e. a polynomial-size witness of an affirmative answer that one can independently check in polynomial time. The qualification 'independent' means that we can check the validity of the witness irrespective of how it was obtained, i.e. not understanding anything about Zhuk's algorithm. That is not the case for unsatisfiable instances. The only apparent polynomial size witness is the particular computation of Zhuk's algorithm on the instance.

In our work, we use some proof complexity methods (formalization in theories of bounded arithmetic, propositional translations, etc.) to show that the algorithm may be appended to provide an independent proof of the correctness of the algorithm for negative answers too. The witness in the case of the homomorphism problem between relational structures is a short propositional proof of a formula $\neg HOM(\mathcal{X}, \mathcal{A})$, encoding that there is no homomorphism from $\mathcal{X}$ to $\mathcal{A}$, in a particular well-known proof system, namely the quantified propositional calculus G [7]. When relational structures are directed graphs, for example, for a given digraph $\mathcal{X}$, the non-existence of a homomorphism from $\mathcal{X}$ to $\mathcal{A}$ can be expressed by the fact that a simple and transparent (see also Definition 2.2 in Chapter 1) set of clauses is not satisfiable: it is built from propositional atoms p_{ij}, one for each vertex i in $\mathcal{X}$ and vertex j in $\mathcal{A}$, and says that the set of pairs (i, j) for which p_{ij} is true is the graph of a map from $\mathcal{X}$ to $\mathcal{A}$, which is a homomorphism. Namely, for any two digraphs $\mathcal{X} = (V_\mathcal{X}, E_\mathcal{X})$, $\mathcal{A} = (V_\mathcal{A}, E_\mathcal{A})$, consider the following set of clauses:

- a clause $\bigvee_{j \in V_\mathcal{A}} p_{i,j}$ for each $i \in V_\mathcal{X}$ (every vertex of $\mathcal{X}$ is sent to some vertex of $\mathcal{A}$);
- a clause $\neg p_{i,j_1} \vee \neg p_{i,j_2}$ for each $i \in V_\mathcal{X}$ and $j_1, j_2 \in V_\mathcal{A}$ with $j_1 \neq j_2$ (the map is well-defined);

- a clause $\neg p_{i_1,j_1} \vee \neg p_{i_2,j_2}$ for every edge $(i_1, i_2) \in E_{\mathcal{X}}$ and $(j_1, j_2) \notin E_{\mathcal{A}}$ (a map is indeed a homomorphism).

A propositional refutation of these clauses in a transparent propositional calculus, whose soundness is obvious, thus indeed serves as a simple (and simple to check) witness for a negative answer to the algorithm.

The thesis consists of three chapters. The first two are papers we wrote while working on the project, the third one is a mathematically complete last part of the project that will be developed into a paper. The notation is consistent throughout the thesis except for minor differences, which are nonetheless mentioned and explained in each part. In particular, in the third part, the reasoning is based on the notation introduced earlier.

The bulk of the work consists of formalizing the soundness of Zhuk's algorithm in bounded arithmetic theory. By soundness, we mean the formula $Reject_{\mathcal{A}}(\mathcal{X}, W) \longrightarrow \neg HOM(\mathcal{X}, \mathcal{A})$, where $Reject_{\mathcal{A}}(\mathcal{X}, W)$ formalizes naturally that W is the algorithm computation on input $\mathcal{X}$ that results in rejection. The formalization is for each specific $\mathcal{A}$ separate; i.e. $\mathcal{A}$ does not feature as a variable in the formalization (and neither does it in Zhuk's algorithm).

The first chapter consists of the paper '$\mathcal{H}$-Colouring Dichotomy in Proof Complexity' [5] that deals with a specific case of the $\mathcal{H}$-coloring problem and is based on the result of Hell and Nešetřil [6]. The soundness of the associated algorithm can be formalized in a simple two-sorted theory V^0 [3], which yields short propositional proofs in $R^*(log)$, a mild extension of resolution.

The general case is divided into the second and third chapters. The second chapter consists of the paper 'Proof complexity of CSP' and we prove there over theory V^1 [3] that the soundness of Zhuk's algorithm follows from three axiom schemes stating non-trivial facts from universal algebra.

These axiom schemes are then proved in the third chapter, 'Proof complexity of universal algebra in a proof of the CSP dichotomy'. Here, we had to use a stronger theory than V^1, namely theory W_1^1 [9]. Using this theory yields as witnesses propositional proofs in calculus G. We show in this chapter that all notions used in the proof of the soundness of Zhuk's algorithm in [10] can be formalized using bounded quantifier formulas (two or three sorted) and that the statements about them can be proved in the theory. To show the latter, we selected a number of statements whose proofs represent all types of argument (in particular, all types of inductive argument) in [10]. To show that the formalization exists, we write the formal definitions of all objects used. This is because we need to know their quantifier complexity (various bounded arithmetic theories differ mainly in the class of formulas for which they assume induction).

The thesis is concluded with a short chapter that points out possible continuations of this research.

Bibliography

[1] 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.

[2] 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.

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

[4] Tomás Feder and Moshe Y. Vardi. The computational structure of monotone monadic snp and constraint satisfaction: A study through datalog and group theory. *SIAM Journal on Computing*, 28(1):57–104, 1998.

[5] Azza Gaysin. H-colouring dichotomy in proof complexity. *Journal of Logic and Computation*, 31(5):1206–1225, 2021.

[6] Pavol Hell and Jaroslav Nešetřil. On the complexity of h-coloring. *Journal of Combinatorial Theory, Series B*, 48(1):92–110, 1990.

[7] Jan Krajíček and Pavel Pudlák. Quantified propositional calculi and fragments of bounded arithmetic. *Mathematical Logic Quarterly*, 36(1):29–46, 1990.

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

[9] Alan Skelley. A third-order bounded arithmetic theory for pspace. In Jerzy Marcinkowski and Andrzej Tarlecki, editors, *Computer Science Logic*, pages 340–354, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

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

1. $\mathcal{H}$-Colouring Dichotomy in Proof Complexity

This chapter is formed by the paper '$\mathcal{H}$-Colouring Dichotomy in Proof Complexity' published in Journal of Logic and Computation. The numbering of definitions and statements is adjusted to make the thesis consistent.

$\mathcal{H}$-colouring Dichotomy in Proof Complexity

Azza Gaysin

Department of Algebra, Faculty of Mathematics and Physics
Charles University in Prague, Prague 186 00, Czech Republic.
E-mail: azza.gaysin@karlov.mff.cuni.cz

Abstract

The $\mathcal{H}$-colouring problem for undirected simple graphs is a computational problem from a huge class of the constraint satisfaction problems (CSP): an $\mathcal{H}$-colouring of a graph $\mathcal{G}$ is just a homomorphism from $\mathcal{G}$ to $\mathcal{H}$ and the problem is to decide for fixed $\mathcal{H}$, given $\mathcal{G}$, if a homomorphism exists or not.

The dichotomy theorem for the $\mathcal{H}$-colouring problem was proved by Hell and Nešetřil [9] in 1990 (an analogous theorem for all CSPs was recently proved by Zhuk [14] and Bulatov [3]) and it says that for each $\mathcal{H}$ the problem is either p-time decidable or NP-complete. Since negations of unsatisfiable instances of CSP can be expressed as propositional tautologies, it seems to be natural to investigate the proof complexity of CSP.

We show that the decision algorithm in the p-time case of the $\mathcal{H}$-colouring problem can be formalized in a relatively weak theory and that the tautologies expressing the negative instances of such $\mathcal{H}$ have polynomial proofs in the propositional proof system $R^*(log)$, a mild extension of resolution. In fact, when the formulas are expressed as unsatisfiable sets of clauses, they have p-size resolution proofs. To establish this, we use a well-known connection between theories of bounded arithmetic and propositional proof systems. This upper bound follows also from a different construction in [1].

We complement this result with a lower bound result that holds for many weak proof systems for a special example of NP-complete case of the $\mathcal{H}$-colouring problem, using known results about the proof complexity of the Pigeonhole Principle.

The main goal of our work is to start the development of some of the theories beyond the CSP dichotomy theorem in bounded arithmetic. We aim eventually – in subsequent work – to formalize in such a theory the soundness of Zhuk's algorithm from [14], extending the upper bound proved here from undirected simple graphs to the general case of directed graphs in some logical calculi.

1.1 Introduction

The constraint satisfaction problem (CSP) is a computational problem. The problem is in finding an assignment of values to a set of variables such that this assignment satisfies some specified feasibility conditions. If such an assignment exists, we call the instance of CSP satisfiable and unsatisfiable otherwise. One can also define CSP through the homomorphism between relational structures: in the constraint satisfaction problem associated

Vol. 31, No. 5, © The Author(s) 2021. Published by Oxford University Press. All rights reserved. For permissions, please e-mail: journals.permission@oup.com. Advance Access Publication on 24 April 2021 https://doi.org/10.1093/logcom/exab028

with a structure $\mathcal{H}$, denoted by CSP($\mathcal{H}$) the question is, given a structure $\mathcal{G}$ over the same vocabulary, whether there exists a homomorphism from $\mathcal{G}$ to $\mathcal{H}$. It turns out that all CSPs can be classified with only two complexity classes: there are either polynomial-time CSPs, or NP-complete CSPs. This dichotomy was conjectured by Feder and Vardi in 1998 [7] and recently proved by Zhuk [14] and Bulatov [3].

The $\mathcal{H}$-colouring problem is essentially CSP($\mathcal{H}$) on relational structures that are undirected graphs. Its computational complexity was investigated years ago, and the dichotomy theorem for the $\mathcal{H}$-colouring problem was proved by Hell and Nešetřil [9] in 1990.

Theorem 1 (The dichotomy theorem for the $\mathcal{H}$-colouring problem, [9])**.** *If $\mathcal{H}$ is bipartite, then the $\mathcal{H}$-colouring problem is in P. Otherwise, the $\mathcal{H}$-colouring problem is NP-complete.*

There is an easy $\mathcal{H}$-colourability test when $\mathcal{H}$ is bipartite.

Lemma 1 ([9])**.** *For all graphs $\mathcal{G}, \mathcal{H}$, if $\mathcal{H}$ is bipartite, then $\mathcal{G}$ is $\mathcal{H}$-colorable if and only if $\mathcal{G}$ is bipartite graph.*

Instances of CSP($\mathcal{H}$) can be expressed by propositional formulas: denote by $\alpha(\mathcal{G}, \mathcal{H})$ the propositional formula expressing that there is a homomorphism from $\mathcal{G}$ to $\mathcal{H}$ (see Definition 2). If the instance of CSP is unsatisfiable, then $\neg\alpha(\mathcal{G}, \mathcal{H})$ is a tautology (for the $\mathcal{H}$-colouring problem we get a tautology any time we consider the bipartite graph $\mathcal{H}$ and the nonbipartite graph $\mathcal{G}$). From this point of view, it is natural to ask about the proof complexity of those instances. A common way to do this is to formalize the sentence in some weak theory of bounded arithmetic, and first prove that this universal statement is valid in all finite structures. Then it could be translated into a family of propositional tautologies that will have short proofs in the corresponding proof system. The simpler the theory, the weaker the propositional proof system will be.

If $\mathcal{H}$-colouring is NP-complete, then the negative instances (graphs $\mathcal{G}$ that cannot be $\mathcal{H}$-colored) form a $coNP$-complete set, and hence, unless $NP = coNP$, they cannot have poly-size proofs in any propositional proof system. In the case when $\mathcal{H}$-colouring is tractable (i.e. we have a p-time algorithm distinguishing positive and negative instances), we shall prove that the negative instances, when represented by unsatisfiable sets of clauses, actually have p-size resolution refutations. A resolution proof is a much more rudimentary object than a run of a p-time algorithm: it operates just on clauses. (In fact, the algorithm can be reconstructed from the proof via feasible interpolation, see Section 1.3.3.2).

In this paper, we show that the decision algorithm in the p-time case of the $\mathcal{H}$-colouring problem (i.e. the case where $\mathcal{H}$ is a bipartite graph) can be formalized in a relatively weak two-sorted theory V^0 [5], which is quite convenient for formalizing sets of vertices and relations between them, and proved by using only formulas of restricted complexity in the Induction scheme. Therefore, tautologies that express negative instances of such $\mathcal{H}$ hence have polynomial proofs in the propositional proof system $R^*(log)$, a mild extension of resolution. In fact, when the formulas are expressed as unsatisfiable sets of clauses, they have p-size resolution proofs. We are interested in a more narrow interpretation of the problem, namely, in the case when a bipartite graph $\mathcal{H}$ is fixed. What we prove is in fact more general: our arguments work for variable bipartite graphs, but we do not expect that something similar could happen for general CSP.

Although the use of the theory of bounded arithmetic for establishing this result (i.e. an upper bound) may seem redundant (indeed, one could directly construct short propositional proofs for the p-case of the $\mathcal{H}$-colouring problem), we believe that this approach

provides the following advantages. The known proofs of CSP dichotomy for general relational structures (see [14], [3]) use advanced notions from universal algebra, such as polymorphism, weak near-unanimity operation, cycle-consistency, absorption, and so forth, that cannot be easily handled directly in propositional logic. To establish the analogous result for the general CSP, one will require the framework allowing one to formalize these advanced notions, and the apparatus of bounded arithmetic is capable of doing that.

We shall complement the upper bound for the $\mathcal{H}$-colouring problem by a lower bound by giving examples of graphs $\mathcal{H}$ and $\mathcal{G}$ for which CSP($\mathcal{H}$) is NP-complete and for which any proof of the tautologies expressing that $\mathcal{G} \notin$ CSP($\mathcal{H}$) must have exponential size length in the constant-depth Frege system (which contains $R^*(log)$) and some other well-known proof systems. This is based on the proof complexity of the Pigeonhole Principle.

The paper is organized as follows. In Section 1.2 we give some common definitions from propositional proof complexity and theory of bounded arithmetic, the definition of CSP in terms of homomorphisms, and explain how to express instances of CSP by propositional formulas. In Section 1.3 we formalize the $\mathcal{H}$-colouring problem in theory V^0 and prove all auxiliary lemmas and the main universal statement. Then we proceed with the translation of the main universal statement into propositional tautologies and prove that for any non-bipartite graph $\mathcal{G}$ and bipartite graph $\mathcal{H}$ the propositional family, expressing that there is no homomorphism from $\mathcal{G}$ to $\mathcal{H}$, has polynomial size bounded depth Frege proofs. Some definitions and material here about translations are quite standard in proof complexity but maybe not so in the CSP community, hence we decided to include them explicitly. We end the section with some remarks about the collateral results and minor improvement of the upper bound. In Section 1.4 we consider NP-complete case of the $\mathcal{H}$-colouring problem and known lower bounds for one suitable example. In Section 1.5 we discuss open questions and the future direction of research.

1.2 Preliminaries

1.2.1 Constraint satisfaction problems and the $\mathcal{H}$-colouring problem

There are many equivalent definitions of the constraint satisfaction problem. Here, we will use the definition in terms of homomorphisms.

Definition 1 (Constraint satisfaction problem)**.**

- 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 the tuple $\mathcal{H} = (H, R_1^{\mathcal{H}}, ..., R_n^{\mathcal{H}})$ such that H is a non-empty set, called the *universe* of $\mathcal{H}$, and each $R_i^{\mathcal{H}}$ is a relation on H having the same arity as the symbol R_i.
- For $\mathcal{G}$, $\mathcal{H}$ being relational structures over the same vocabulary R_1,..., R_n a homomorphism from $\mathcal{G}$ to $\mathcal{H}$ is a mapping $\phi : \mathcal{G} \to \mathcal{H}$ from the universe G to H such that for every m-ary relation $R^{\mathcal{G}}$ and every tuple $(a_1, ..., a_m) \in R^{\mathcal{G}}$ we have $(\phi(a_1), ..., \phi(a_m)) \in R^{\mathcal{H}}$.

Let $\mathcal{H}$ be a relational structure over a vocabulary R_1,..., R_n. In the *constraint satisfaction problem* associated with $\mathcal{H}$, denoted by CSP($\mathcal{H}$) the question is, given a structure $\mathcal{G}$ over the same vocabulary, whether there exists a homomorphism from $\mathcal{G}$ to $\mathcal{H}$. If the answer is positive, then we call the instance $\mathcal{G}$ *satisfiable* and *unsatisfiable* otherwise [2].

The $\mathcal{H}$-colouring problem could be described as follows: let $\mathcal{H} = (V_\mathcal{H}, E_\mathcal{H})$ be a simple undirected graph without loops, whose vertices we consider as different colors. An $\mathcal{H}$-colouring of a simple undirected graph $\mathcal{G} = (V_\mathcal{G}, E_\mathcal{G})$ without loops is an assignment of colors to the vertices of $\mathcal{G}$ such that adjacent vertices of $\mathcal{G}$ obtain adjacent colors. Since a graph homomorphism $h : \mathcal{G} \to \mathcal{H}$ is a mapping of $V_\mathcal{G}$ to $V_\mathcal{H}$ such that if g, g' are adjacent vertices of $\mathcal{G}$, then so are $h(g), h(g')$, it is easy to see that an $\mathcal{H}$-colouring of $\mathcal{G}$ is just a homomorphism $\mathcal{G} \to \mathcal{H}$. A graph $\mathcal{H}$ can be considered as a relational structure $\mathcal{H} = (V_\mathcal{H}, E_\mathcal{H})$ with only one binary symmetric irreflexive relation $E_\mathcal{H}(i, j)$ (to i, j be adjacent vertices). Thus, the problem of $\mathcal{H}$-colouring of a graph $\mathcal{G}$ is equivalent to CSP($\mathcal{H}$).

To express an instance of CSP($\mathcal{H}$) by the propositional formula, we use the following construction [1]. For any sets $V_\mathcal{G}$ and $V_\mathcal{H}$ by $V(V_\mathcal{G}, V_\mathcal{H})$ we denote a set of propositional variables: for every $v \in V_\mathcal{G}$ and every $u \in V_\mathcal{H}$ there is a variable $x_{v,u}$ in the set $V(V_\mathcal{G}, V_\mathcal{H})$. A variable $x_{v,u}$ is assigned the truth value 1 if and only if the vertex v is mapped to vertex u. To every graph $\mathcal{G} = (V_\mathcal{G}, E_\mathcal{G})$ we assign a set of clauses $CNF(\mathcal{G}, \mathcal{H})$ over the variables in $V(V_\mathcal{G}, V_\mathcal{H})$ in such a way that there is a one-to-one correspondence between the truth valuations of the variables in $V(V_\mathcal{G}, V_\mathcal{H})$ satisfying this set and the homomorphisms from $\mathcal{G}$ to $\mathcal{H}$.

Definition 2. For any two graphs $\mathcal{G} = (V_\mathcal{G}, E_\mathcal{G})$, $\mathcal{H} = (V_\mathcal{H}, E_\mathcal{H})$, by $CNF(\mathcal{G}, \mathcal{H})$ we denote the following set of clauses,

- a clause $\bigvee_{u \in V_\mathcal{H}} x_{v,u}$ for each $v \in V_\mathcal{G}$;
- a clause $\neg x_{v,u_1} \vee \neg x_{v,u_2}$ for each $v \in V_\mathcal{G}$ and $u_1, u_2 \in V_\mathcal{H}$ with $u_1 \neq u_2$;
- a clause $\neg x_{v_1,u_1} \vee \neg x_{v_2,u_2}$ for any adjacent vertices $v_1, v_2 \in V_\mathcal{G}$ and non-adjacent vertices $u_1, u_2 \in V_\mathcal{H}$.

It is easy to see that if we exchange the last item with a more general definition:

- a clause $\bigvee_{i \in [r]} \neg x_{v_i,u_i}$ for each natural number r, each relation symbol R of arity r, each $(v_1, v_2, ..., v_r) \in R^\mathcal{G}$, and each $(u_1, u_2, ..., u_r) \notin R^\mathcal{H}$,

we get the set of clauses $CNF(\mathcal{G}, \mathcal{H})$ for a common CSP on any relational structure.

1.2.2 Bounded Arithmetic

Some definitions, examples, and results are adapted from [5]. In our work, we use *two-sorted first-order* (sometimes called second-order) set-up as a framework for the theory. Here there are two kinds of variables: the variables $x, y, z, ...$ of the first sort are called *number variables* and range over the natural numbers, and the variables $X, Y, Z, ...$ of the second sort are called *set (or also string) variables* and range over finite subsets of natural numbers (which represent binary strings). Functions and predicate symbols may involve both sorts and there are two kinds of functions: the number-valued functions (or just *number functions*) and the string-valued functions (or just *string functions*). Quantifiers over number variables are called *number quantifiers*, and quantifiers over string variables are called *string quantifiers*.

The usual language of arithmetic for two-sorted first-order theories is the extension of the standard language for Peano Arithmetic $\mathcal{L}_{\mathcal{PA}}$.

Definition 3 ($\mathcal{L}^2{}_{\mathcal{PA}}$)**.** $\mathcal{L}^2{}_{\mathcal{PA}} = \{0, 1, +, \cdot, |\,|; =_1, =_2, \leq, \in\}$.

Here the symbols $0, 1, +, \cdot, =_1$ and $\leq$ are well-known and are from $\mathcal{L}_{\mathcal{PA}}$: they are function and predicate symbols over the first sort. The function $|X|$ (*the length of* X) is a number-valued function and is intended to denote the least upper bound of the set X (the length of the corresponding string). The binary predicate $\in$ for a number and a set denotes set membership, and $=_2$ is the equality predicate for sets. The defining properties of all symbols from language ${\mathcal{L}^2}_{\mathcal{PA}}$ are described by a set of basic axioms denoted by 2-$BASIC$ [5], which we do not present here.

Notation 1. *We will use the abbreviation*

$$X(t) =_{def} t \in X,$$

where t is a number term. Thus, we think of $X(i)$ as the ith bit of a binary string X of length $|X|$.

To define the theory V^0, in which we will formalize the $\mathcal{H}$-colouring problem, we need the following definitions.

Definition 4 (Bounded formulas)**.** Let $\mathcal{L}$ be a two-sorted vocabulary. If x is a number variable and X is a string variable that do not occur in the $\mathcal{L}$-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 $\forall x(x \leq t \rightarrow \phi)$, $\exists X \leq t\phi$ stands for $\exists X(|X| \leq t \wedge \phi)$ and $\forall X \leq t\phi$ stands for $\forall X(|X| \leq t \rightarrow \phi)$. Quantifiers that occur in this form are said to be *bounded*, and a *bounded formula* is one in which every quantifier is bounded.

Notation 2. *We will use the following abbreviations: $\exists \bar{x} \leq \bar{t}\phi$ stands for $\exists x_1 \leq t_1, ...,$ $\exists x_k \leq t_k\phi$ for some k, where no x_i occurs in any t_j (even if $i < j$). Similarly, for $\forall \bar{x} \leq \bar{t}$, $\exists \bar{X} \leq \bar{t}$, $\forall \bar{X} \leq \bar{t}$.*

Definition 5 (Σ^B_i and Π^B_i formulas in ${\mathcal{L}^2}_{\mathcal{PA}}$)**.** We will define Σ^B_i and Π^B_i formulas recursively as follows.

- $\Sigma^B_0 = \Pi^B_0$ is the set of ${\mathcal{L}^2}_{\mathcal{PA}}$-formulas whose only quantifiers are bounded number quantifiers (there can be free string variables);
- For $i \geq 0$, Σ^B_{i+1} (resp. Π^B_{i+1}) is the set of formulas of the form $\exists \bar{X} \leq \bar{t}\phi(\bar{X})$ (resp. $\forall \bar{X} \leq \bar{t}\phi(\bar{X})$), where ϕ is a Π^B_i formula (resp. Σ^B_i formula), and $\bar{t}$ is a sequence of ${\mathcal{L}^2}_{\mathcal{PA}}$-terms that do not involve any variable from $\bar{X}$.

Definition 6 (Comprehension Axiom)**.** If Φ is a set of formulas, then the *comprehension axiom scheme* for Φ, denoted by Φ-$COMP$, is the set of formulas

$$\exists X \leq y \forall z < y(X(z) \longleftrightarrow \phi(z)), \tag{1.1}$$

where $\phi(z)$ is any formula in Φ, X does not occur free in $\phi(z)$, and $\phi(z)$ may have free variables of both sorts, in addition to z.

Definition 7 (V^0)**.** *The theory V^0* has the vocabulary ${\mathcal{L}^2}_{\mathcal{PA}}$ and is axiomatized by 2-$BASIC$ and Σ^B_0-$COMP$.

There is no explicit Induction axiom scheme in V^0, but it is known [4] that $V^0 \vdash \Sigma^B_0$-IND, where Φ-IND is Number Induction Axiom.

Definition 8 (Number Induction Axiom)**.** If Φ is a set of two-sorted formulas, then Φ-IND axioms are the formulas

$$(\phi(0) \wedge \forall x(\phi(x) \rightarrow \phi(x+1))) \rightarrow \forall z \phi(z), \tag{1.2}$$

where ϕ is a formula in Φ.

1.2.3 Propositional Proof Complexity

In this section we define the propositional proof systems R, $R(log)$, and their tree-like versions. Some definitions and results are adopted from [10],[12].

Definition 9 (Propositional proof system, [6])**.** A *propositional proof system* is a polynomial time function P whose range is the set $TAUT$. For a tautology $\tau \in TAUT$, any string w such that $P(w) = \tau$ is called a P-proof of τ.

Proof systems are usually defined by a finite number of inference rules of a particular form, and a proof is created by applying them step by step. The complexity of a proof is measured by its size and number of steps.

The *resolution system* R operates with atoms and their negations and has no other logical connectives. The basic object is a *clause*, a disjunction of a finite set of literals. The *resolution rule* allows us to derive new clause $C_1 \cup C_2$ from two clauses $C_1 \cup \{p\}$ and $C_2 \cup \{\neg p\}$.

$$\frac{C_1 \cup \{p\} \quad C_2 \cup \{\neg p\}}{C_1 \cup C_2}. \tag{1.3}$$

If we manage to derive the *empty clause* $\emptyset$ from the initial set of clauses $\mathcal{C}$, the clauses in the set $\mathcal{C}$ are not simultaneously satisfiable. Thus, the resolution system can be interpreted as a *refutation proof system*: instead of proving that a formula is a tautology, it proves that a set of clauses $\mathcal{C} = \{C_1, C_2, ..., C_n\}$ is not satisfiable, and therefore formula $\alpha = \bigvee_{i=1}^{n} \neg C_i$ is a tautology.

Definition 10 (An R-proof)**.** Let $\mathcal{C}$ be a set of clauses, an *R-refutation* of $\mathcal{C}$ is a sequence of clauses $D_1, ..., D_k$ such that

- for each $i \leq k$, either $D_i \in \mathcal{C}$ or there are $u, v < i$ such that D_i follows from D_u, D_v by the resolution rule,
- $D_k = \emptyset$.

The number of steps in the refutation is k.

The *DNF-resolution* (denoted by DNF-R) is a proof system extending R by allowing in clauses not only literals but also their conjunctions [12]. DNF-R has the following inference rules.

$$\frac{C \cup \{\bigwedge_j l_j\} \quad D \cup \{\neg l'_1, ..., \neg l'_t\}}{C \cup D}, \tag{1.4}$$

if $t \geq 1$ and all l'_i occur among l_j, and

$$\frac{C \cup \{\bigwedge_{j \leq s} l_j\} \quad D \cup \{\bigwedge_{s < j \leq t} l_j\}}{C \cup D \cup \{\bigwedge_{i \leq s+t} l_i\}}. \tag{1.5}$$

Notice that the constant-depth Frege systems generalize the resolution and DNF-R systems, which are depth one and depth two systems, respectively.

Let $f : \mathbb{N}^+ \to \mathbb{N}^+$ be a non-decreasing function. Define $R(f)$-size of a DNF-R refutation π to be the minimum s such that

- π has at most s steps (i.e. clauses), and
- every logical term occurring in π has size at most $f(s)$.

Thus, a size s $R(log)$-refutation may contain terms of size up to $log(s)$.

Definition 11 (Tree-like proof systems)**.** A proof is called *tree-like* if every step of the proof is a part of the hypotheses of at most one inference in the proof (each line in the proof can be used only once as a hypothesis for an inference rule). For a proof system P by P^* we denote the proof system whose proofs are exactly tree-like P-proofs, for example R^* and $R^*(log)$.

Definition 12 (p-simulation)**.** Let P and Q be two propositional proof systems. A p-time function $f : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ is a p-simulation of Q by P if and only if for all strings ω, α

$$Q(\omega, \alpha) \to P(f(\omega, \alpha), \alpha).$$

Lemma 2 (5.7.2 in [12])**.** *R p-simulates $R^*(log)$ with respect to refutations of sets of clauses.*

We also introduce Definition 13, which we will use at the end of Section 1.3.3.

Definition 13 (DNF_1-formula)**.** A *basic formula* is an atomic formula or the negation of an atomic formula. A *DNF_1-formula* is a formula that is built from basic formulas by

- first apply any number of conjunctions and bounded universal quantifiers,
- then apply any number of disjunctions and bounded existential quantifiers.

1.3 Formalization of the $\mathcal{H}$-colouring problem in V^0

1.3.1 Defining Relations

In this section we define all the notions we need to formalize the decision algorithm in the p-time case of the $\mathcal{H}$-colouring problem, i.e. the notions of a graph, bipartite and non-bipartite graphs and a homomorphism between graphs, in the vocabulary $\mathcal{L}^2{}_{\mathcal{PA}}$ and using only basic axioms of V^0. To do this, we extend our theory with new predicate and function symbols, and for each of them, we add defining axioms which ensure that they receive their standard interpretations in a model of V^0.

Definition 14 (Representable/Definable relations)**.** Let $\mathcal{L} \supseteq \mathcal{L}^2{}_{\mathcal{PA}}$ be a two-sorted vocabulary, and let ϕ be a $\mathcal{L}$-formula. Then we say that $\phi(\bar{x}, \bar{X})$ represents (or defines) a relation $R(\bar{x}, \bar{X})$ if

$$R(\bar{x}, \bar{X}) \longleftrightarrow \phi(\bar{x}, \bar{X}). \tag{1.6}$$

If Φ is a set of $\mathcal{L}$-formulas, then we say that $R(\bar{x}, \bar{X})$ is Φ-*representable* (or Φ-definable) if it is represented by some $\phi \in \Phi$.

Definition 15 (Definable number functions)**.** Let T be a theory with a two-sorted vocabulary $\mathcal{L} \supseteq \mathcal{L}^2{}_{\mathcal{PA}}$, and let Φ be a set of $\mathcal{L}$-formulas. A number function f is Φ-*definable* in T if there is a formula $\phi(\bar{x}, y, \bar{X})$ in Φ such that

$$T \vdash \forall \bar{x} \forall \bar{X}\, \exists! y\, \phi(\bar{x}, y, \bar{X}), \tag{1.7}$$

and

$$y = f(\bar{x}, \bar{X}) \longleftrightarrow \phi(\bar{x}, y, \bar{X}). \tag{1.8}$$

The auxiliary predicate and function symbols, which we will use further to define different notions in V^0, are the following.

Definition 16 (Divisibility)**.** The relation of *divisibility* is defined by

$$x|y \longleftrightarrow \exists z \leq y(x \cdot z = y). \tag{1.9}$$

Definition 17 (Pairing function)**.** If $x, y \in \mathbb{N}$ we define the *pairing function* $\langle x, y\rangle$ as the following term in V^0.

$$\langle x, y\rangle = (x + y)(x + y + 1) + 2y. \tag{1.10}$$

Since the formula for pairing function is just a term in the standard vocabulary for the theory V^0, it is obvious that V^0 proves the condition (1.7). It is also easy to prove in V^0 that the pairing function is a one-one function, i.e.

$$V^0 \vdash \forall x_1, x_2, y_1, y_2\ \langle x_1, y_1\rangle = \langle x_2, y_2\rangle \rightarrow x_1 = x_2 \wedge y_1 = y_2. \tag{1.11}$$

Using the pairing function, we can code a pair of numbers x, y by one number $\langle x, y\rangle$, and the sequence of pairs by a subset of numbers. To define a graph on n vertices, consider a string $V_{\mathcal{G}}$ where $|V_{\mathcal{G}}| = n$ and $\forall i < n\ V_{\mathcal{G}}(i)$. We say that $V_{\mathcal{G}}$ is the set of n vertices of the graph $\mathcal{G}$. Then we define string $E_{\mathcal{G}}$ of length $|E_{\mathcal{G}}| < 4n^2$ to be the set of edges of the graph $\mathcal{G}$ as following: if there is an edge between vertices i, j, then, using the pairing function, set $E_{\mathcal{G}}(\langle i, j\rangle)$ and $\neg E_{\mathcal{G}}(\langle i, j\rangle)$ otherwise.

Notation 3. *Instead of $E_{\mathcal{G}}(\langle i, j\rangle)$ we will write just $E_{\mathcal{G}}(i, j)$ to denote that there is an edge between i and j, and sometimes instead of $(V_{\mathcal{G}}, E_{\mathcal{G}})$ we will write $\mathcal{G}$.*

Definition 18 (Undirected graph $\mathcal{G}$ without loops)**.** A pair of sets $\mathcal{G} = (V_{\mathcal{G}}, E_{\mathcal{G}})$ with $|V_{\mathcal{G}}| = n$ denotes an undirected graph without loops if it satisfies the following relation.

$$\begin{gathered} GRAPH(V_{\mathcal{G}}, E_{\mathcal{G}}) \longleftrightarrow \forall i < n(V_{\mathcal{G}}(i)) \wedge \forall i < j < n \\ (E_{\mathcal{G}}(i, j) \longleftrightarrow E_{\mathcal{G}}(j, i)) \wedge \forall i < n\ \neg(E_{\mathcal{G}}(i, i)). \end{gathered} \tag{1.12}$$

Talking about graphs, we will consider only pairs of strings $\mathcal{G} = (V_{\mathcal{G}}, E_{\mathcal{G}})$ that satisfy the above relation. Since we formalize the $\mathcal{H}$-colouring problem, we need to define the homomorphism on graphs in the vocabulary $\mathcal{L}^2{}_{\mathcal{PA}}$. Consider two graphs $\mathcal{G} = (V_{\mathcal{G}}, E_{\mathcal{G}})$ and $\mathcal{H} = (V_{\mathcal{H}}, E_{\mathcal{H}})$, where $|V_{\mathcal{G}}| = n$, $|V_{\mathcal{H}}| = m$. Firstly, we define a map between two sets of vertices $V_{\mathcal{G}}, V_{\mathcal{H}}$, i.e. between sets $[0, n-1]$ and $[0, m-1]$. We again use the pairing function: consider a set $Z < \langle n-1, m-1\rangle + 1$, where $Z(\langle i, j\rangle)$ means that ith vertex is mapped to jth vertex. For Z to be a well-defined map, it should satisfy the following Σ^B_0-definable relation $MAP(n, m, Z)$.

Definition 19 (Map between two sets)**.** We say that a set Z is a *well-defined map* between two sets $[0, n-1]$ and $[0, m-1]$ if it satisfies the relation

$$\begin{gathered} MAP(n, m, Z) \longleftrightarrow \forall i < n \exists j < m\, Z(\langle i, j\rangle) \wedge \\ \forall i < n \forall j_1, j_2 < m(Z(\langle i, j_1\rangle) \wedge Z(\langle i, j_2\rangle) \rightarrow j_1 = j_2). \end{gathered} \tag{1.13}$$

Now we can formalize the standard notion of the existence of a homomorphism between two graphs $\mathcal{G}$ and $\mathcal{H}$ (here the homomorphism is formalized by a set Z with certain properties).

Definition 20 (The existence of a homomorphism between graphs $\mathcal{G}$ and $\mathcal{H}$)**.** There is a *homomorphism between two graphs $\mathcal{G} = (V_{\mathcal{G}}, E_{\mathcal{G}})$ and $\mathcal{H} = (V_{\mathcal{H}}, E_{\mathcal{H}})$* with $|V_{\mathcal{G}}| = n$, $|V_{\mathcal{H}}| = m$ if they satisfy the relation

$$\begin{gathered} HOM(\mathcal{G}, \mathcal{H}) \longleftrightarrow \exists Z \leq \langle n-1, m-1\rangle (MAP(n, m, Z) \wedge \\ \forall i_1, i_2 < n, \forall j_1, j_2 < m \\ (E_{\mathcal{G}}(i_1, i_2) \wedge Z(\langle i_1, j_1\rangle) \wedge Z(\langle i_2, j_2\rangle) \rightarrow E_{\mathcal{H}}(j_1, j_2))). \end{gathered} \tag{1.14}$$

Note that the relation $HOM(\mathcal{G},\mathcal{H})$ is a Σ_1^B-definable relation.

Finally, we need to formalize what it means to be a bipartite or a non-bipartite graph. The notion of being bipartite is Σ_1^B-definable in ${\mathcal{L}^2}_{\mathcal{PA}}$.

Definition 21 (Bipartite graph $\mathcal{H}$)**.** A graph $\mathcal{H} = (V_\mathcal{H}, E_\mathcal{H})$ with $|V_\mathcal{H}| = m$ is *bipartite* if it satisfies the relation

$$\begin{gathered} BIP(\mathcal{H}) \longleftrightarrow \exists W_\mathcal{H}, U_\mathcal{H} \leq m(\forall i < m(W_\mathcal{H}(i) \leftrightarrow \neg U_\mathcal{H}(i)) \wedge \\ \forall i < j < m(E_\mathcal{H}(i,j) \rightarrow (W_\mathcal{H}(i) \wedge U_\mathcal{H}(j)) \vee (W_\mathcal{H}(j) \wedge U_\mathcal{H}(i)))). \end{gathered} \tag{1.15}$$

To define a non-bipartite graph we use a commonly known characterization of non-bipartite graphs (to contain an odd cycle, or, more generally, to allow a homomorphism from an odd cycle). The reason here is to get a Σ_1^B-definable relation for a non-bipartite graph. This makes the formula in the main statement in the next section Π_1^B, and hence translatable into propositional logic. First, we define a cycle.

Definition 22 (Cycle $\mathcal{C}_k$)**.** A graph $\mathcal{C}_k = (V_{\mathcal{C}_k}, E_{\mathcal{C}_k})$ with $V_{\mathcal{C}_k} = \{0, 1, ..., k-1\}$ is a *cycle of length* k if it satisfies the relation

$$\begin{gathered} CYCLE(\mathcal{C}_k) \longleftrightarrow E_{\mathcal{C}_k}(0, k-1) \wedge \forall i < (k-1)\, E_{\mathcal{C}_k}(i, i+1) \wedge \\ \forall i, j < (k-1)(j \neq i+1 \rightarrow \neg E_{\mathcal{C}_k}(i,j)). \end{gathered} \tag{1.16}$$

Definition 23 (Non-bipartite graph $\mathcal{G}$)**.** A graph $\mathcal{G} = (V_\mathcal{G}, E_\mathcal{G})$ with $|V_\mathcal{G}| = n$ is *non-bipartite* if it satisfies the following Σ_1^B-definable relation

$$\begin{gathered} NONBIP(\mathcal{G}) \longleftrightarrow \exists k \leq n(2|(k-1)) \exists V_{\mathcal{C}_k} = k, \exists E_{\mathcal{C}_k} < 4k^2 \\ CYCLE(V_{\mathcal{C}_k}, E_{\mathcal{C}_k}) \wedge HOM(\mathcal{C}_k, \mathcal{G}). \end{gathered} \tag{1.17}$$

1.3.2 Proving in theory V^0

Lemma 3 (Homomorphism transitivity)**.** *For all graphs $\mathcal{G}, \mathcal{H}, \mathcal{S}$, V^0 proves the property of a homomorphism to be transitive.*

$$V^0 \vdash \forall \mathcal{G}, \mathcal{H}, \mathcal{S}\, (HOM(\mathcal{G},\mathcal{H}) \wedge HOM(\mathcal{H},\mathcal{S}) \rightarrow HOM(\mathcal{G},\mathcal{S})). \tag{1.18}$$

Proof. Consider the graphs $\mathcal{G} = (V_\mathcal{G}, E_\mathcal{G}))$, $\mathcal{H} = (V_\mathcal{H}, E_\mathcal{H})$ and $\mathcal{S} = (V_\mathcal{S}, E_\mathcal{S})$, where $|V_\mathcal{G}| = n$, $|V_\mathcal{H}| = m$ and $|V_\mathcal{S}| = t$. Since $HOM(\mathcal{G},\mathcal{H})$ and $HOM(\mathcal{H},\mathcal{S})$, there exist two sets $Z \leq \langle n-1, m-1\rangle$ and $Z' \leq \langle m-1, t-1\rangle$ that satisfy the homomorphism definition. We need to prove that there exists a set $Z'' \leq \langle n-1, t-1\rangle$ such that

$$\begin{gathered} MAP(n, t, Z'') \wedge \forall i_1, i_2 < n, \forall k_1, k_2 < t \\ (E_\mathcal{G}(i_1, i_2) \wedge Z''(\langle i_1, k_1\rangle) \wedge Z''(\langle i_2, k_2\rangle) \rightarrow E_\mathcal{S}(k_1, k_2)). \end{gathered} \tag{1.19}$$

Consider the set $Z'' \leq \langle n-1, t-1\rangle$ which we define by the formula

$$Z''(\langle i,k\rangle) \longleftrightarrow \exists j < m(Z(\langle i,j\rangle) \wedge Z'(\langle j,k\rangle)). \tag{1.20}$$

This set should exist due to Comprehension Axiom Σ_0^B-$COMP$ since the formula $\phi(\langle i,k\rangle)$ $= \exists j < m\ (Z(\langle i,j\rangle) \wedge Z'(\langle j,k\rangle)) \in \Sigma_0^B$. It is easy to check that the set Z'' satisfies the homomorphism relation between graphs $\mathcal{G}$ and $\mathcal{S}$. □

Notation 4. *K_2 will denote the complete graph on two vertices.*

In the following two lemmas, we prove that there is always a homomorphism from a bipartite graph to K_2 and there is no homomorphism from a non-bipartite graph to K_2.

Lemma 4. *For all bipartite graphs $\mathcal{H}$, V^0 proves the existence of a homomorphism from $\mathcal{H}$ to $\mathcal{K}_2$.*

$$V^0 \vdash \forall \mathcal{H}\,(BIP(\mathcal{H}) \to HOM(\mathcal{H}, \mathcal{K}_2)). \tag{1.21}$$

Proof. Consider a bipartite graph $\mathcal{H} = (V_{\mathcal{H}}, E_{\mathcal{H}})$ with $|V_{\mathcal{H}}| = n$. We need to show that there exists a homomorphism from $\mathcal{H}$ to $\mathcal{K}_2$, i.e. an appropriate set $Z \leq \langle n-1, 2\rangle$. Since $\mathcal{H}$ is bipartite, it follows that there exist two subsets $W_{\mathcal{H}}$ and $U_{\mathcal{H}}$ such that $(W_{\mathcal{H}}(i) \leftrightarrow \neg U_{\mathcal{H}}(i))$. Consider a set $Z \leq \langle n-1, 2\rangle$ such that

$$\begin{cases} Z(\langle i, 0\rangle) \longleftrightarrow W_{\mathcal{H}}(i), \\ Z(\langle i, 1\rangle) \longleftrightarrow U_{\mathcal{H}}(i). \end{cases}$$

This set also exists due to Comprehension Axiom Σ_0^B-$COMP$ since formula $\phi(\langle i, j\rangle) = (j = 0 \wedge W_{\mathcal{H}}(i)) \vee (j = 1 \wedge U_{\mathcal{H}}(i)) \in \Sigma_0^B$. Obviously, since $(W_{\mathcal{H}}(i) \leftrightarrow \neg U_{\mathcal{H}}(i))$, by the definition of Z we have $MAP(n, 2, Z)$. Consider any $i_1, i_2 < n$ such that $E_{\mathcal{H}}(i_1, i_2)$. Then $(W_{\mathcal{H}}(i_1) \wedge U_{\mathcal{H}}(i_2))$ or $(W_{\mathcal{H}}(i_2) \wedge U_{\mathcal{H}}(i_1))$. In the first case we have $Z(\langle i_1, 0\rangle) \wedge Z(\langle i_2, 1\rangle)$, in the second case $Z(\langle i_2, 0\rangle) \wedge Z(\langle i_1, 1\rangle)$, and in both cases $E_{\mathcal{K}_2}(0, 1)$. Thus, Z is a homomorphism from $\mathcal{H}$ to $\mathcal{K}_2$. □

Lemma 5. *For all non-bipartite graphs $\mathcal{G}$, V^0 proves that there is no homomorphism from $\mathcal{G}$ to $\mathcal{K}_2$.*

$$V^0 \vdash \forall \mathcal{G}\ (NONBIP(\mathcal{G}) \to \neg HOM(\mathcal{G}, \mathcal{K}_2)). \tag{1.22}$$

Proof. Suppose that a graph $\mathcal{G} = (V_{\mathcal{G}}, E_{\mathcal{G}})$, $|V_{\mathcal{G}}| = n$ is non-bipartite, i.e. there exist $k \leq n$, $\mathcal{C}_k = (V_{\mathcal{C}_k}, H_{\mathcal{C}_k})$ with $|V_{\mathcal{C}_k}| = k$ such that $2|(k-1)$, $CYCLE(\mathcal{C}_k)$ and $HOM(\mathcal{C}_k, \mathcal{G})$.

Assume that there exists a homomorphism from $\mathcal{G}$ to $\mathcal{K}_2$. Due to Lemma 3, by transitivity there is also a homomorphism $Z \leq \langle k-1, 2\rangle$ from $\mathcal{C}_k$ to $\mathcal{K}_2$. Since it is a homomorphism from $\mathcal{C}_k$ to $\mathcal{K}_2$, for every $0 \leq i \leq (k-1)$ either $Z(\langle i, 0\rangle)$ or $Z(\langle i, 1\rangle)$.

Without loss of generality, suppose that $Z(\langle 0, 0\rangle)$ and let us prove that $Z(\langle k-1, 0\rangle)$. Since $2|(k-1)$, it follows that $k > 2$. Due to $CYCLE(\mathcal{C}_k)$, $E_{\mathcal{C}_k}(0, 1)$ and $E_{\mathcal{C}_k}(1, 2)$. We claim that for every $i < k$, if $2|i$, then $Z(\langle i, 0\rangle)$ and $Z(\langle i, 1\rangle)$ otherwise. Consider the formula:

$$\phi(i, Z) = (2|i \to Z(\langle i, 0\rangle)) \wedge (2 \nmid i \to Z(\langle i, 1\rangle)). \tag{1.23}$$

Since $\phi(i, Z) \in \Sigma_0^B$, we can prove this claim by induction on i, because V^0 proves Σ_0^B-IND.

$$(\phi(0, Z) \wedge \forall i < k(\phi(i, Z) \to \phi(i+1, Z)) \to \forall j < k\, \phi(j, Z). \tag{1.24}$$

The base case is considered above. For the step of induction, suppose that it is true for $(i-1)$ and consider i. We have two options. If $2|(i-1)$, then by the induction hypothesis $Z(\langle i-1, 0\rangle)$. Thus, since for $(i-1)$ by $CYCLE(\mathcal{C}_k)$ we have $E_{\mathcal{C}_k}(i-1, i)$, by the definition of homomorphism $Z(\langle i, 1\rangle)$. Analogously, if $2 \nmid (i-1)$, then $Z(\langle i, 0\rangle)$.

Hence $Z(\langle 0, 0\rangle)$ and $Z(\langle k-1, 0\rangle)$. But since there is an edge between the vertices 0 and $(k-1)$ in the graph $\mathcal{C}_k$, Z cannot be a homomorphism between $\mathcal{C}_k$ and K_2. Therefore, our assumption leads to a contradiction, and there is no homomorphism from $\mathcal{G}$ to $\mathcal{K}_2$. □

The main result of this paper is an immediate conclusion from the previous lemmas.

Theorem 2 (The main universal statement)**.** *For all non-bipartite graphs $\mathcal{G}$ and bipartite graphs $\mathcal{H}$, V^0 proves that there is no homomorphism from $\mathcal{G}$ to $\mathcal{H}$.*

$$V^0 \vdash \forall \mathcal{G}, \mathcal{H}(BIP(\mathcal{H}) \wedge NONBIP(\mathcal{G}) \to \neg HOM(\mathcal{G}, \mathcal{H})). \tag{1.25}$$

Proof. Suppose that there exists a homomorphism from $\mathcal{G}$ to $\mathcal{H}$. According to Lemma 4, since $\mathcal{H}$ is bipartite, there exists a homomorphism from $\mathcal{H}$ to K_2. Thus, due to Lemma 3, by the transitivity there exists a homomorphism from $\mathcal{G}$ to K_2. But this is in contradiction with Lemma 5. □

1.3.3 Translating into tautologies

1.3.3.1 Translation of the main universal statement

In this section we proceed with the translation of the main universal statement in the theory V^0 into propositional tautologies. There is a well-known translation of Σ_0^B formulas into propositional calculus formulas: we can translate each formula $\phi(\bar{x}, \bar{X}) \in \Sigma_0^B$ into a family of propositional formulas [5].

$$||\phi(\bar{x}, \bar{X})|| = \{\phi(\bar{x}, \bar{X})[\bar{m}, \bar{n}] : \bar{m}, \bar{n} \in \mathbb{N}\}. \tag{1.26}$$

Lemma 6 ([5])**.** *For every $\Sigma_0^B(\mathcal{L}^2{}_{\mathcal{PA}})$ formula $\phi(\bar{x}, \bar{X})$, there is a constant $d \in \mathbb{N}$ and a polynomial $p(\bar{m}, \bar{n})$ such that for all $\bar{m}, \bar{n} \in \mathbb{N}$, the propositional formula $\phi(\bar{x}, \bar{X})[\bar{m}, \bar{n}]$ has depth at most d and size at most $p(\bar{m}, \bar{n})$ [5].*

There is a theorem that establishes a connection between Σ_0^B-fragment of the theory V^0 and constant-depth Frege proof system.

Theorem 3 (V^0 Translation, [5])**.** *Suppose that $\phi(\bar{x}, \bar{X})$ is a Σ_0^B formula such that $V^0 \vdash \forall\bar{x}\forall\bar{X}\phi(\bar{x}, \bar{X})$. Then the propositional family $||\phi(\bar{x}, \bar{X})||$ has polynomial size bounded depth Frege proofs. That is, there are a constant d and a polynomial $p(\bar{m}, \bar{n})$ such that for all $1 \leq \bar{m}, \bar{n} \in \mathbb{N}$, $\phi(\bar{x}, \bar{X})[\bar{m}, \bar{n}]$ has a d-Frege proof of size at most $p(\bar{m}, \bar{n})$. Further, there is an algorithm that finds a d-Frege proof of $\phi(\bar{x}, \bar{X})[\bar{m}, \bar{n}]$ in time bounded by a polynomial in $(\bar{m}, \bar{n})$ [5].*

Consider the Π_1^B-formula $\phi(\mathcal{G}, \mathcal{H})$ from Theorem 2 which expresses that there is no homomorphism from a non-bipartite graph $\mathcal{G}$ to a bipartite graph $\mathcal{H}$.

$$\begin{aligned} \phi(\mathcal{G}, \mathcal{H}) = \neg GRAPH(\mathcal{G}) \vee \neg GRAPH(\mathcal{H}) \vee \\ \vee \neg BIP(\mathcal{H}) \vee \neg NONBIP(\mathcal{G}) \vee \neg HOM(\mathcal{G}, \mathcal{H}). \end{aligned} \tag{1.27}$$

For the graphs $\mathcal{G} = (V_\mathcal{G}, E_\mathcal{G})$ with $|V_\mathcal{G}| = n$ and $\mathcal{H} = (V_\mathcal{H}, E_\mathcal{H})$ with $|V_\mathcal{H}| = m$ we can rewrite this formula as follows:

$$\phi(V_\mathcal{G}, E_\mathcal{G}, V_\mathcal{H}, E_\mathcal{H}) = \tag{1.28}$$
$$\exists i < n\, \neg V_\mathcal{G}(i) \vee \exists i < j < n \tag{1.29}$$
$$((\neg E_\mathcal{G}(i,j) \vee \neg E_\mathcal{G}(j,i)) \wedge (E_\mathcal{G}(i,j) \vee E_\mathcal{G}(j,i))) \vee \exists i < n\, E_\mathcal{G}(i,i) \tag{1.30}$$
$$\vee \tag{1.31}$$
$$\exists i < m\, \neg V_\mathcal{H}(i) \vee \exists i < j < m \tag{1.32}$$
$$((\neg E_\mathcal{H}(i,j) \vee \neg E_\mathcal{H}(j,i)) \wedge (E_\mathcal{H}(i,j) \vee E_\mathcal{H}(j,i))) \vee \exists i < n\, E_\mathcal{H}(i,i) \tag{1.33}$$
$$\vee \tag{1.34}$$
$$\forall W_\mathcal{H}, U_\mathcal{H} \leq m (\exists i < m\, ((\neg W_\mathcal{H}(i) \vee U_\mathcal{H}(i)) \wedge (W_\mathcal{H}(i) \vee \neg U_\mathcal{H}(i))) \vee \tag{1.35}$$
$$\vee \exists i < j < m \tag{1.36}$$
$$(E_\mathcal{H}(i,j) \wedge (\neg W_\mathcal{H}(i) \vee \neg U_\mathcal{H}(j)) \wedge (\neg W_\mathcal{H}(j) \vee \neg U_\mathcal{H}(i)))) \tag{1.37}$$
$$\vee \tag{1.38}$$

$$\forall k \le n(2|(k-1))\,\forall V_{\mathcal{C}_k} = k,\, \forall E_{\mathcal{C}_k} < 4k^2((\exists i < k\, \neg V_{\mathcal{C}_k}(i) \vee \tag{1.39}$$

$$\vee \exists i < j < k((\neg E_{\mathcal{C}_k}(i,j) \vee \neg E_{\mathcal{C}_k}(j,i)) \wedge (E_{\mathcal{C}_k}(i,j) \vee E_{\mathcal{C}_k}(j,i))) \vee \tag{1.40}$$

$$\vee \exists i < k\, E_{\mathcal{C}_k}(i,i)) \vee (\neg E_{\mathcal{C}_k}(0,k-1) \vee \exists i < (k-1) \tag{1.41}$$

$$\neg E_{\mathcal{C}_k}(i,i+1) \vee \exists i,j < (k-1)\,(j \neq i+1 \wedge E_{\mathcal{C}_k}(i,j))) \vee \tag{1.42}$$

$$\vee (\forall Z \le \langle k-1, n-1\rangle\,(\neg MAP(k,n,Z) \vee \exists i_1, i_2 < k \exists j_1, j_2 < n \tag{1.43}$$

$$E_{\mathcal{C}_k}(i_1,i_2) \wedge Z(\langle i_1,j_1\rangle) \wedge Z(\langle i_2,j_2\rangle) \wedge \neg E_{\mathcal{G}}(j_1,j_2)))) \tag{1.44}$$

$$\vee \tag{1.45}$$

$$\forall Z' \le \langle n-1, m-1\rangle(\neg MAP(n,m,Z') \vee \exists i_1, i_2 < n, \exists j_1, j_2 < m \tag{1.46}$$

$$(E_{\mathcal{G}}(i_1,i_2) \wedge Z'(\langle i_1,j_1\rangle) \wedge Z'(\langle i_2,j_2\rangle) \wedge \neg E_{\mathcal{H}}(j_1,j_2))). \tag{1.47}$$

In strict form (with all string quantifiers occurring in front) the formula $\phi(V_{\mathcal{G}}, E_{\mathcal{G}}, V_{\mathcal{H}}, E_{\mathcal{H}})$ looks like

$$\begin{gathered}\phi(V_{\mathcal{G}}, E_{\mathcal{G}}, V_{\mathcal{H}}, E_{\mathcal{H}}) = \forall W_{\mathcal{H}}, U_{\mathcal{H}} \le m, \forall V_{\mathcal{C}_k} \le n, \forall E_{\mathcal{C}_k} \le 4n^2, \\ \forall Z \le \langle k-1, n-1\rangle, \forall Z' \le \langle n-1, m-1\rangle \\ [\psi(n, m, V_{\mathcal{G}}, V_{\mathcal{H}}, W_{\mathcal{H}}, U_{\mathcal{H}}, V_{\mathcal{C}_k}, E_{\mathcal{G}}, E_{\mathcal{H}}, E_{\mathcal{C}_k}, Z, Z')],\end{gathered} \tag{1.48}$$

where

$$\psi(n, m, V_{\mathcal{G}}, V_{\mathcal{H}}, W_{\mathcal{H}}, U_{\mathcal{H}}, V_{\mathcal{C}_k}, E_{\mathcal{G}}, E_{\mathcal{H}}, E_{\mathcal{C}_k}, Z, Z')$$

is the Σ_0^B-formula. Thus, by Lemma 22 one can translate it into a family of short propositional formulas. For every free string variable X, $|X| = n_X$ in the formula ψ we introduce propositional variables $p_0^X, p_1^X, ..., p_{n_{(X-1)}}^X$ where p_i^X is intended to mean $X(i)$. The first two parts (1.29)-(1.30), (1.32)-(1.33) of the formula $\phi(V_{\mathcal{G}}, E_{\mathcal{G}}, V_{\mathcal{H}}, E_{\mathcal{H}})$ say that $\mathcal{G}, \mathcal{H}$ are not graphs. The free number variables here are n, m, free string variables are $V_{\mathcal{G}}, V_{\mathcal{H}}, E_{\mathcal{G}}, E_{\mathcal{H}}$. For graph $\mathcal{G}$, (1.29)-(1.30) translates into

$$\begin{gathered}[\bigvee_{i=0}^{n-1}(\neg p_i^{V_{\mathcal{G}}})] \vee [\bigvee_{j=0}^{n-1}\bigvee_{i=0}^{j-1}(\neg p_{\langle i,j\rangle}^{E_{\mathcal{G}}} \vee \neg p_{\langle j,i\rangle}^{E_{\mathcal{G}}}) \wedge (p_{\langle i,j\rangle}^{E_{\mathcal{G}}} \vee p_{\langle j,i\rangle}^{E_{\mathcal{G}}})] \vee \\ [\bigvee_{i=0}^{n-1}(p_{\langle i,i\rangle}^{E_{\mathcal{G}}})].\end{gathered} \tag{1.49}$$

And for graph $\mathcal{H}$, (1.32)-(1.33) translates into

$$\begin{gathered}[\bigvee_{i=0}^{m-1}(\neg p_i^{V_{\mathcal{H}}})] \vee [\bigvee_{j=0}^{m-1}\bigvee_{i=0}^{j-1}(\neg p_{\langle i,j\rangle}^{E_{\mathcal{H}}} \vee \neg p_{\langle j,i\rangle}^{E_{\mathcal{H}}}) \wedge (p_{\langle i,j\rangle}^{E_{\mathcal{H}}} \vee p_{\langle j,i\rangle}^{E_{\mathcal{H}}})] \vee \\ [\bigvee_{i=0}^{m-1}(p_{\langle i,i\rangle}^{E_{\mathcal{H}}})].\end{gathered} \tag{1.50}$$

The third part (1.35)-(1.37) of the formula $\phi(V_{\mathcal{G}}, E_{\mathcal{G}}, V_{\mathcal{H}}, E_{\mathcal{H}})$ is about the graph $\mathcal{H}$ not being bipartite, free number variable here is m, free string variables are $W_{\mathcal{H}}, U_{\mathcal{H}}, E_{\mathcal{H}}$. The translation of (1.35)-(1.37) is

$$\begin{gathered}[\bigvee_{i=0}^{m-1}(\neg p_i^{W_{\mathcal{H}}} \vee p_i^{U_{\mathcal{H}}}) \wedge (p_i^{W_{\mathcal{H}}} \vee \neg p_i^{U_{\mathcal{H}}})] \vee \\ [\bigvee_{j=0}^{m-1}\bigvee_{i=0}^{j-1} p_{\langle i,j\rangle}^{E_{\mathcal{H}}} \wedge (\neg p_i^{W_{\mathcal{H}}} \vee \neg p_j^{U_{\mathcal{H}}}) \wedge (\neg p_j^{W_{\mathcal{H}}} \vee \neg p_i^{U_{\mathcal{H}}})].\end{gathered} \tag{1.51}$$

The fourth part (1.39)-(1.44) of the formula $\phi(V_{\mathcal{G}}, E_{\mathcal{G}}, V_{\mathcal{H}}, E_{\mathcal{H}})$ expresses that $\mathcal{G}$ is not a non-bipartite graph. Free number variable here is n, free string variables are $V_{\mathcal{C}_k}, E_{\mathcal{C}_k}, Z$. This complex subformula we split into parts. Firstly, the part of subformula saying that $\mathcal{C}_k$ is not a graph is translated into

$$[\bigvee_{i=0}^{k-1}(\neg p_i^{V_{\mathcal{C}_k}})] \vee [\bigvee_{j=0}^{k-1}\bigvee_{i=0}^{j-1}(\neg p_{\langle i,j\rangle}^{E_{\mathcal{C}_k}} \vee \neg p_{\langle j,i\rangle}^{E_{\mathcal{C}_k}}) \wedge (p_{\langle i,j\rangle}^{E_{\mathcal{C}_k}} \vee p_{\langle j,i\rangle}^{E_{\mathcal{C}_k}})] \vee [\bigvee_{i=0}^{n-1}(p_{\langle i,i\rangle}^{E_{\mathcal{C}_k}})]. \tag{1.52}$$

Then the part saying that $\mathcal{C}_k$ is not a cycle translates into

$$[\neg p_{\langle 0,k-1\rangle}^{E_{\mathcal{C}_k}}] \vee [\bigvee_{i=0}^{k-2}\neg p_{\langle i,i+1\rangle}^{E_{\mathcal{C}_k}}] \vee [\bigvee_{i=0}^{k-2}\bigvee_{j=0,\, j\neq i+1}^{k-2} p_{\langle j,i\rangle}^{E_{\mathcal{C}_k}}]. \tag{1.53}$$

And the part saying that Z is not a map or not a homomorphism between $\mathcal{C}_k$ and $\mathcal{G}$, is translated into

$$\begin{aligned}[\bigvee_{i=0}^{k-1}\bigwedge_{j=0}^{n-1}(\neg p_{\langle i,j\rangle}^{Z})] \vee [\bigvee_{i=0}^{k-1}\bigvee_{j_2=0}^{n-1}\bigvee_{j_1=0,\, j_1\neq j_2}^{n-1}(p_{\langle i,j_1\rangle}^{Z} \wedge p_{\langle i,j_2\rangle}^{Z})] \vee \\ [\bigvee_{i_1,i_2=0}^{k-1}\bigvee_{j_1,j_2=0}^{n-1}(p_{\langle i_1,i_2\rangle}^{E_{\mathcal{C}_k}} \wedge p_{\langle i_1,j_1\rangle}^{Z} \wedge p_{\langle i_2,j_2\rangle}^{Z} \wedge \neg p_{\langle j_1,j_2\rangle}^{E_{\mathcal{G}}})].\end{aligned} \tag{1.54}$$

Finally, to get the translation of the whole subformula, we need first to make a disjunction of all formulas (1.52)-(1.54) and then make a conjunction on k.

$$\begin{aligned}\bigwedge_{k=3,\,2|(k-1)}^{n-1}\Big[&[\bigvee_{i=0}^{k-1}(\neg p_i^{V_{\mathcal{C}_k}})] \vee [\bigvee_{j=0}^{k-1}\bigvee_{i=0}^{j-1}(\neg p_{\langle i,j\rangle}^{E_{\mathcal{C}_k}} \vee \neg p_{\langle j,i\rangle}^{E_{\mathcal{C}_k}}) \wedge (p_{\langle i,j\rangle}^{E_{\mathcal{C}_k}} \vee p_{\langle j,i\rangle}^{E_{\mathcal{C}_k}})] \vee \\ &[\bigvee_{i=0}^{n-1}(p_{\langle i,i\rangle}^{E_{\mathcal{C}_k}})] \vee [\neg p_{\langle 0,k-1\rangle}^{E_{\mathcal{C}_k}}] \vee [\bigvee_{i=0}^{k-2}\neg p_{\langle i,i+1\rangle}^{E_{\mathcal{C}_k}}] \vee [\bigvee_{i=0}^{k-2}\bigvee_{j=0,\, j\neq i+1}^{k-2} p_{\langle j,i\rangle}^{E_{\mathcal{C}_k}}] \vee \\ &[\bigvee_{i=0}^{k-1}\bigwedge_{j=0}^{n-1}(\neg p_{\langle i,j\rangle}^{Z})] \vee [\bigvee_{i=0}^{k-1}\bigvee_{j_2=0}^{n-1}\bigvee_{j_1=0,\, j_1\neq j_2}^{n-1}(p_{\langle i,j_1\rangle}^{Z} \wedge p_{\langle i,j_2\rangle}^{Z})] \vee \\ &[\bigvee_{i_1,i_2=0}^{k-1}\bigvee_{j_1,j_2=0}^{n-1}(p_{\langle i_1,i_2\rangle}^{E_{\mathcal{C}_k}} \wedge p_{\langle i_1,j_1\rangle}^{Z} \wedge p_{\langle i_2,j_2\rangle}^{Z} \wedge \neg p_{\langle j_1,j_2\rangle}^{E_{\mathcal{G}}})].\Big]\end{aligned} \tag{1.55}$$

And the fifth part (1.46)-(1.47) of the formula $\phi(V_{\mathcal{G}}, E_{\mathcal{G}}, V_{\mathcal{H}}, E_{\mathcal{H}})$ saying that there is no homomorphism from $\mathcal{G}$ to $\mathcal{H}$, with free number variables n, m, free string variables $Z', E_{\mathcal{G}}, E_{\mathcal{H}}$, is translated into

$$\begin{aligned}[\bigvee_{i=0}^{n-1}\bigwedge_{j=0}^{m-1}(\neg p_{\langle i,j\rangle}^{Z'})] \vee [\bigvee_{i=0}^{n-1}\bigvee_{j_2=0}^{m-1}\bigvee_{j_1=0,\, j_1\neq j_2}^{m-1}(p_{\langle i,j_1\rangle}^{Z'} \wedge p_{\langle i,j_2\rangle}^{Z'})] \vee \\ [\bigvee_{i_1,i_2=0}^{n-1}\bigvee_{j_1,j_2=0}^{m-1}(p_{\langle i_1,i_2\rangle}^{E_{\mathcal{G}}} \wedge p_{\langle i_1,j_1\rangle}^{Z'} \wedge p_{\langle i_2,j_2\rangle}^{Z'} \wedge \neg p_{\langle j_1,j_2\rangle}^{E_{\mathcal{H}}})].\end{aligned} \tag{1.56}$$

The family of propositional formulas $||\psi(n, m, V_{\mathcal{G}}, V_{\mathcal{H}}, W_{\mathcal{H}}, U_{\mathcal{H}}, V_{\mathcal{C}_k}, E_{\mathcal{G}}, E_{\mathcal{H}},\ E_{\mathcal{C}_k},\ Z, Z')||$ is therefore the disjunction of formulas (1.49)-(1.56) for all possible n, m, $n_{V_{\mathcal{G}}}$, $n_{V_{\mathcal{H}}}$, $n_{W_{\mathcal{H}}}$,

$n_{U_\mathcal{H}}$, $n_{V_{\mathcal{C}_k}}$, $n_{E_\mathcal{G}}$, $n_{E_\mathcal{H}}$, $n_{E_{\mathcal{C}_k}}$, n_Z, $n_{Z'} \in \mathbb{N}$. By Theorem 3, this family of tautologies has a polynomial-size bounded depth Frege proof.

We are now ready to prove our main goal: to show that the formulas $||\neg HOM(\mathcal{G},\mathcal{H})||$ for any non-bipartite graph $\mathcal{G}$ and bipartite graph $\mathcal{H}$, have short propositional proofs. Note that the propositional family $||\neg HOM(\mathcal{G},\mathcal{H})||$ is logically equivalent to $\neg \bigwedge CNF(\mathcal{G},\mathcal{H})$, which we introduced in Definition 2. The upper bound stated next is also a consequence of the results in Section 5 of [1] that use different methods.

Theorem 4 (Upper Bound)**.** *For any non-bipartite graph $\mathcal{G}$ and bipartite graph $\mathcal{H}$ the propositional family $||\neg HOM(\mathcal{G},\mathcal{H})||$ has polynomial size bounded depth Frege proofs.*

Proof. By the construction above and Theorem 3 the translation of the formula (1.27) has the p-size constant-depth Frege proof. If $\mathcal{G}$ and $\mathcal{H}$ are graphs, then the translations of the first two disjuncts in (1.27) are propositional sentences that evaluate to 0 and thus can be computed in the proof system.

Further, because $\mathcal{H}$ is bipartite, we can find its two parts $W_\mathcal{H}, U_\mathcal{H}$ and evaluate accordingly the atoms in the translation of $\neg BIP(\mathcal{H})$ corresponding to $W_\mathcal{H}$ and $U_\mathcal{H}$ so that the whole translation of the disjunct $\neg BIP(\mathcal{H})$ becomes false. That is, as before, it is a propositional sentence that evaluates to 0. The analogous argument removes the translation of the disjunct $\neg NONBIP(\mathcal{G})$: substitute for the atoms corresponding to a homomorphism from an odd cycle for some k values determined by an actual homomorphism from $\mathcal{C}_k$ into $\mathcal{G}$. This will turn the translation of the fourth disjunct $\neg NONBIP(\mathcal{G})$ into a sentence equal to 0 as well.

To summarize: after these substitutions the first four disjuncts in the translation of the formula (1.27) become propositional sentences evaluated to 0 and thus the entire translation of the formula (1.27) is equivalent to the translation of $\neg HOM(\mathcal{G},\mathcal{H})$. That is, we obtained a polynomial-size constant-depth Frege proof of $||\neg HOM(G,H)||$. □

1.3.3.2 Other Remarks

Actually, we can slightly improve our upper bound result from Section 1.3.3.1. To reason about graphs, we used a convenient for this purpose set-up of a two-sorted theory V^0, including the Comprehension axiom. However, actually, we can avoid using it in both the proofs of Lemmas 3 and 4. For example, in the proof of Lemma 3 instead of declaring the existence of the set $Z''(\langle i,k\rangle) \longleftrightarrow \exists j < m(Z(\langle i,j\rangle) \wedge Z'(\langle j,k\rangle))$ by the Comprehension axiom we can derive that there always exists such $j < m$ that $Z(\langle i,j\rangle)$ and $Z'(\langle j,k\rangle)$ (since $MAP(n,m,Z) \wedge MAP(m,t,Z')$) and therefore we just manually construct the appropriate set Z''. Thus, we can switch between the theory V^0 and the weaker theory $I\Sigma_0^{1,b}$, which is axiomatized by 2-$BASIC$ and $I\Sigma_0^{1,b}$-IND (where $I\Sigma_0^{1,b}$ denotes the class of $\mathcal{L}^2{}_{\mathcal{PA}}$-formulas with all number quantifiers bounded and without string quantifiers) when it is needed. Moreover, we can further restrict the complexity of formulas in the Induction scheme from the full class $I\Sigma_0^{1,b}$ to its subclass Σ_1^b (which allows only bounded existential number quantifiers) since we use the Induction scheme only once for the Σ_1^b-formula (1.23) in the proof of Lemma 5.

Denote by $T_1^1(\alpha)$ the two-sorted theory in the vocabulary $\mathcal{L}^2{}_{\mathcal{PA}}$, containing the 2-$BASIC$ and IND scheme for Σ_1^b-formulas. There is then a theorem.

Theorem 5 ([12])**.** *Suppose that $\phi(\bar{x},\bar{X})$ is a Σ_0^B, DNF_1-formula such that $T_1^1(\alpha) \vdash \forall\bar{x}\forall\bar{X}\phi(\bar{x},\bar{X})$. Then the propositional family $||\phi(\bar{x},\bar{X})||$ has polynomial size $R^*(log)$-proofs. That is, there is a polynomial $p(\bar{m},\bar{n})$ such that for all $1 \leq \bar{m},\bar{n} \in \mathbb{N}$, $\neg\phi(\bar{x},\bar{X})[\bar{m},\bar{n}]$ has an $R^*(log)$-refutation of size at most $p(\bar{m},\bar{n})$. Furthermore, there is an algorithm that finds a $R^*(log)$-refutation of $\neg\phi(\bar{x},\bar{X})[\bar{m},\bar{n}]$ in time bounded by a polynomial in $(\bar{m},\bar{n})$.*

It is obvious that we can modify little the formula $\psi(...)$ in (1.48) to become DNF_1: to transform it to DNF we use the limited extension introduced by Tseitin, and to remove all existential quantifiers after universal ones we use Herbrandization (i.e. Skolemization of the negation, see Section 13.2 of [12]). Thus, the negations of the family of tautologies expressing that there is no homomorphism from a non-bipartite graph $\mathcal{G}$ to a bipartite graph $\mathcal{H}$ have polynomial $R^*(log)$-refutation in the $R^*(log)$ system, which is essentially a constant depth Frege system with depth 2 and narrow logical terms.

Another note is that one of our auxiliary lemmas, Lemma 5, gives us a collateral result. The Π_1^B-formula (1.22)

$$\phi(\mathcal{G}) = \neg NONBIP(\mathcal{G}) \vee \neg HOM(\mathcal{G}, \mathcal{K}_2),$$

expressing that there is no homomorphism from a non-bipartite graph $\mathcal{G}$ to a complete graph $\mathcal{K}_2$, also could be rewritten in strict form as the universal statement of the Σ_0^B-fragment of V^0. Thus, the family of tautologies into which one can translate this universal statement also has polynomial size $R^*(log)$-proofs. Essentially, the formula (1.22) means that the sets of bipartite and non-bipartite graphs are disjoint since we can define a bipartite graph $\mathcal{H}$ as

$$BIP(\mathcal{H}) \longleftrightarrow HOM(\mathcal{H}, \mathcal{K}_2). \tag{1.57}$$

We know that resolution R p-simulates $R^*(log)$ system (see Lemma 2). Thus, due to the feasible interpolation Theorem 6, there is a p-time algorithm separating bipartite and non-bipartite graphs. Of course, this is well-known, but here we obtain the algorithm as a consequence of the existence of polynomial resolution proofs.

Theorem 6 (The feasible interpolation theorem, [12])**.** *Assume that the set of clauses* $\{A_1, ..., A_m, B_1, ..., B_l\}$ *for all* $i \leq m, j \leq l$ *satisfies*

$$A_i \subseteq \{p_1, \neg p_1, ..., p_n, \neg p_n, q_1, \neg q_1, ..., q_s, \neg q_s\};$$
$$B_j \subseteq \{p_1, \neg p_1, ..., p_n, \neg p_n, r_1, \neg r_1, ..., r_t, \neg r_t\},$$

and has a resolution refutation with k *clauses. Then the implication*

$$\bigwedge_{i \leq m} (\bigvee A_i) \to \neg \bigwedge_{j \leq l} (\bigvee B_j)$$

has an interpolating circuit $I(\bar{p})$ *whose size is* $O(kn)$*. If the refutation is tree-like,* I *is a formula. Moreover, if all atoms* $\bar{p}$ *occur only positively in all* A_i*, then there is a monotone interpolating circuit (or a formula in the tree-like case) whose size is* $O(kn)$*.*

1.4 Lower Bounds

In this section we consider another side of the dichotomy of the $\mathcal{H}$-colouring problem, namely, NP-complete case for non-bipartite graphs $\mathcal{H}$. Since the consequence of this section is rather an observation than an independent result, we will not define proof systems from Theorems 7-10: the reader can find the definitions in [8], [10], [13], [11] if desired.

A well-studied example of the $\mathcal{H}$-colouring problem is the $\mathcal{K}_n$-colouring problem, which is essentially the n-colouring problem, where $\mathcal{K}_n$ is a complete graph on $n > 2$ vertices. One of the obvious negative instances for $\mathrm{CSP}(\mathcal{K}_n)$ is the graph $\mathcal{K}_{n+1}$: it is impossible to n-color complete graph with $n + 1$ vertices. The propositional formula, expressing that there is no homomorphism from $\mathcal{K}_{n+1}$ to $\mathcal{K}_n$, is logically equivalent to the Pigeonhole

Principle formula PHP_n^{n+1}, because essentially trying to find a homomorphism from $\mathcal{K}_{n+1}$ to $\mathcal{K}_n$ is trying to injectively map the set $[0, n+1]$ to the set $[0, n]$. The PHP_n^{n+1} formula is as follows:

$$\neg[\bigwedge_i \bigvee_j p_{ij} \wedge \bigwedge_i \bigwedge_{j \neq j'} (\neg p_{ij} \vee \neg p_{ij'}) \wedge \bigwedge_{i \neq i'} \bigwedge_j (\neg p_{ij} \vee \neg p_{i'j})], \tag{1.58}$$

where $(n+1)n$ atoms p_{ij} with $i \in [n+1]$ and $j \in [n]$ expressing that i is mapped to j. For PHP_n^{n+1}, there are a lot of known lower bounds in different weak proof systems.

Theorem 7 ([8]). *There exists a constant c, $c > 1$, so that, for sufficiently large n, every resolution refutation of $\neg PHP_n^{n+1}$ contains at least c^n different clauses.*

Theorem 8 (Ajtai 1988, Beame et al. 1992, [10]). *Assume that F is a Frege proof system and d is a constant, and let $n > 1$. Then in every depth d F-proof of the formula PHP_n^{n+1} at least $2^{n^{(1/6)^d}}$ different formulas must occur. In particular, each depth d F-proof of PHP_n^{n+1} must have size at least $2^{n^{(1/6)^d}}$ and must have at least $\Omega(2^{n^{(1/6)^d}})$ proof steps.*

We also can consider weak variants of the PHP principle, PHP_n^m, where the number m of pigeons is larger than $n+1$ (which will be equivalent to the non-existence of homomorphism from $\mathcal{K}_m$ to $\mathcal{K}_n$).

Theorem 9 ([13]). *For $m > n$ PHP_n^m has no polynomial calculus refutation of degree $d \leq \lceil n/2 \rceil$.*

Theorem 10 ([11]). *Let c, d and a prime p be fixed, and let q be a number not divisible by p. Then there is $\delta > 0$ such that for all n large enough it holds: there is $m \leq n$ such that in every tree-like $F_d^c(MOD_p)$-proof of PHP_n^{n+m} at least $exp(n^\delta)$ different formulas must occur.*

Thus, we see that even for such an elementary negative instance of NP-complete case of the $\mathcal{H}$-colouring problem, $\mathrm{CSP}(\mathcal{K}_n)$, the tautology, expressing that there is no homomorphism from $\mathcal{K}_m$ to $\mathcal{K}_n$, $m \geq n+1$, has no short proofs in many weak proof systems.

1.5 Conclusion

We have constructed in Section 1.3.3 short proofs of propositional statements expressing that $\mathcal{G} \notin CSP(\mathcal{H})$ for non-bipartite graphs $\mathcal{G}$ and bipartite graphs $\mathcal{H}$ by translating into propositional logic a suitable formalization of the algorithm for the p-time case of the $\mathcal{H}$-colouring problem. Note that while this algorithm is very simple, it is not AC^0-computable (parity is easily AC^0-reducible to the question of whether or not a graph is bipartite), while our propositional proofs operate only with clauses and are thus, in this respect, more rudimentary than the decision algorithm is.

The conditions for the p-time case of the $\mathcal{H}$-colouring problem (and the algorithm) are so simple that one could perhaps directly construct short propositional proofs, and the use of bounded arithmetic may seem redundant. However, we think of this work as a stepping stone towards proving an analogous result for the full dichotomy theorem. Its known proofs rely on universal algebra, and formalizing them in a suitable bounded arithmetic theory ought to be accessible, while direct propositional formalization looks unlikely. For this reason, we use bounded arithmetic here as a common framework. Moreover, this

framework generally allows us to obtain some collateral results that help to compose a complete picture of the problem.

In this work, we aimed to develop the language of reasoning about the CSP dichotomy in the theory of bounded arithmetic. The eventual goal is to formalize in such a theory the soundness of Zhuk's algorithm from [14], and translate it into a corresponding proof system, extending the upper bound proved here from undirected graphs to the full CSP in some logical calculi.

An interesting issue that we left out is to prove a lower bound not just for a suitable $\mathcal{H}$ (as we did in Section 1.4) but for all $\mathcal{H}$ that fall under the NP-complete case of the dichotomy theorem. If CSP($\mathcal{H}$) is NP-complete, then, unless $NP = coNP$, no proof system can prove in p-size all valid statements $\mathcal{G} \notin \text{CSP}(\mathcal{H})$. In addition, if the NP-completeness of the class can be formalized in a theory T and we have a lower bound for the proof system corresponding to T (see [12] for this topic), then one can use it to construct $\mathcal{G}$ for which the lower bound holds. This uses a well-known part of proof complexity, but we do feel that it adds to our understanding of the proof complexity of CSP; it is rather a transposition of known results via known techniques. For this reason, we do not pursue this avenue of research here.

Funding: This work was partly supported by the project SVV-2020-260589 and by the Charles University Research Centre program [UNCE/SCI/022].

Acknowledgements: I would like to thank my supervisor Jan Krajíček for the helpful comments that resulted in many improvements to this paper. I am also grateful to Pavel Pudlák, Neil Thapen, and others for the opportunity to present this work at the Institute of Mathematics of the Czech Academy of Sciences and for further discussion. Finally, I would like to thank Albert Atserias for responding to our queries about [1].

Bibliography

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

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

[3] 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.

[4] Stephen A. Cook. Feasibly constructive proofs and the propositional calculus. In *Proceedings of the Seventh Annual ACM Symposium on Theory of Computing*, STOC '75, page 83–97, New York, NY, USA, 1975. Association for Computing Machinery.

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

[6] Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *Journal of Symbolic Logic*, 44(1):36–50, 1979.

[7] Tomás Feder and Moshe Y. Vardi. The computational structure of monotone monadic snp and constraint satisfaction: A study through datalog and group theory. *SIAM Journal on Computing*, 28(1):57–104, 1998.

[8] Armin Haken. The intractability of resolution. *Theoretical Computer Science*, 39:297 – 308, 1985. Third Conference on Foundations of Software Technology and Theoretical Computer Science.

[9] Pavol Hell and Jaroslav Nešetřil. On the complexity of h-coloring. *Journal of Combinatorial Theory, Series B*, 48(1):92–110, 1990.

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

[11] Jan Krajiček. Lower bounds for a proof system with an exponential speed-up over constant-depth frege systems and over polynomial calculus. In Igor Prívara and Peter Ružička, editors, *Mathematical Foundations of Computer Science 1997*, pages 85–90, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg.

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

[13] Alexander A. Razborov. Lower bounds for the polynomial calculus. *Computational complexity*, 7(4):291–324, 1998.

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

2. Proof complexity of CSP

This chapter is formed by the paper 'Proof complexity of CSP'. The numbering of definitions and statements is adjusted to make the thesis consistent.

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.

2.1 Introduction

An important class of NP problems are the constraint satisfaction problems (CSP). We will give its definition in Section 2.2.2, but a universal formulation is as follows: in a constraint satisfaction problem 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 CSP($\mathcal{A}$). A celebrated theorem of Zhuk [19] and Bulatov [7] states that for each constraint language $\mathcal{A}$, 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 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 CSP($\mathcal{A}$), as that would imply that NP is closed under complementation. However, when 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 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

This work was partly supported by the project SVV-2020-260589, the project "Grant Schemes at CU" (reg. no. CZ.02.2.69/0.0/0.0/19_073/0016935) and by Charles University Research Centre program [UNCE/SCI/022].

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 2.2 we recall the necessary background from universal algebra, CSP theory, proof complexity, and bounded arithmetic. In Section 2.3 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 2.4. Section 2.5 is devoted to the soundness of Zhuk's algorithm and is divided into three principal parts. In Sections 2.5.1 - 2.5.3 we introduce the framework, formalize most of the notions used in the algorithm, and define a new theory of bounded arithmetic. In Section 2.5.4 we prove the soundness of consistency reductions in the theory V^1. Finally, in Section 2.5.5 we consider the linear case of the algorithm. The main theorem is formulated in Section 2.5.6 and the summary of the proof is presented there.

2.2 Preliminaries

2.2.1 Basic notions from universal algebra

This section 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 11 ([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 12 (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 $E(X) = X \cup \{f(a_1, ..., a_n) : f \text{ is a basic operation on } A, a_1, ..., a_n \in X\}$, 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 13 (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.*

2.2.2 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 24 (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 25 (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 7 ([17]). *For any idempotent WNU operation Ω on a finite set, there exists a special WNU operation $\Omega' \in Clone(\Omega)$.*

Theorem 14 (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 12 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 26 (*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 15. *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-constructs each other.*

2.2.3 Characterization of a CSP instance

This section 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. CSP(Γ) is p-equivalent to 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* $\ddot{\mathcal{A}} = (V_{\ddot{\mathcal{A}}}, E^{ij}_{\ddot{\mathcal{A}}} : 0 \leq i, j < n)$, 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 $(a, b) \in E^{ij}_{\ddot{\mathcal{A}}}$ (relations $E^{ij}_{\ddot{\mathcal{A}}}$ 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 $(a, b) \in E^{ij}_{\ddot{\mathcal{A}}}$ 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 $a \in D_i$, $E^{ij}_{\ddot{\mathcal{A}}}(a, b)$, there is an edge $(a', b) \in E^{ij}_{\ddot{\mathcal{A}}}$ for every $a' \in D_i$. Note that for a 1-consistent CSP instance this means that $E^{ij}_{\ddot{\mathcal{A}}}$ 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.

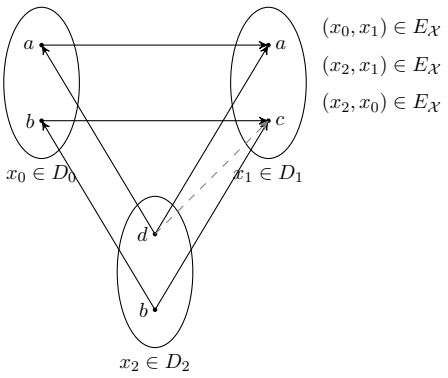

Figure 2.1: Cycle-consistent, non-linked instance.

Compare as examples two CSP instances in Figure 2.1 and Figure 2.2. 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, $E^{01}_{\ddot{\mathcal{A}}} = \{(a, a), (b, c)\})$, $E^{21}_{\ddot{\mathcal{A}}} = \{(d, a), (b, c)\})$ and $E^{20}_{\ddot{\mathcal{A}}} = \{(d, a), (b, b)\}$. 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,

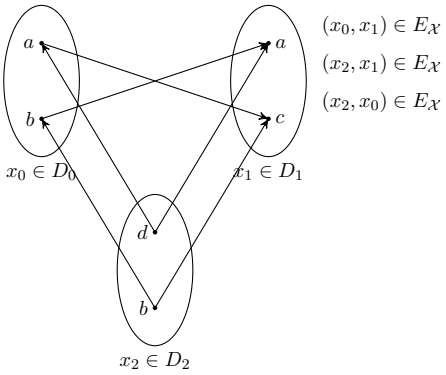

Figure 2.2: Linked, not cycle-consistent instance.

for example, there is no path connecting a and b in D_0. However, if we add one more edge (d, c) to $E^{21}_{\ddot{\mathcal{A}}}$, the new instance will be linked. On the contrary, the second instance in Figure 2.2 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 $E^{ij}_{\ddot{\mathcal{A}}}$ (i.e. by making a richer relation of the same arity). The algorithm never increases the domains.

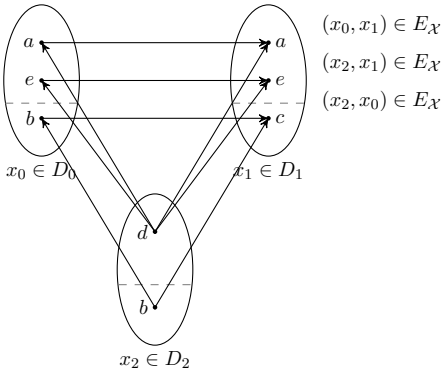

Figure 2.3: Division into linked components.

We conclude this section with Lemma 8 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 8 ([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 2.3. 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 $\{b\}$.

2.2.4 The theory V^1

In this section 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 5. *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 27 (2-BASIC, second-sort axioms)**.** The set 2-BASIC for the second-sort variables contains the following axioms:

1. $X(y) \rightarrow y < |X|$.
2. $y + 1 =_1 |X| \rightarrow X(y)$.
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 6. *Sometimes for a set* A*, an element* x *and a formula* ϕ *instead of* $\exists x < |A|\, A(x) \wedge \phi$ *and* $\forall x < |A|\, A(x) \rightarrow \phi$ *we will write* $\exists x \in A\, \phi$ *and* $\forall x \in A\, \phi$*.*

Definition 28 (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 $\forall x(x \leq t \rightarrow \phi)$, $\exists X \leq t\phi$ stands for $\exists X(|X| \leq t \wedge \phi)$ and $\forall X \leq t\phi$ stands for $\forall X(|X| \leq t \rightarrow \phi)$. Quantifiers that occur in this form are said to be *bounded*, and a *bounded formula* is one in which every quantifier is bounded.

Definition 29 (Number Induction axioms)**.** If Φ is a set of two-sorted formulas, then Φ-IND axioms are the formulas

$$\phi(0) \wedge \forall x(\phi(x) \rightarrow \phi(x+1)) \rightarrow \forall z \phi(z), \tag{2.1}$$

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

Definition 30 (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

$$\phi(y) \rightarrow \exists x \leq y(\phi(x) \wedge \neg \exists H < x\, \phi(z)), \tag{2.2}$$

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

$$\phi(0) \rightarrow \exists x \leq y(\phi(x) \wedge \neg \exists H \leq y(x < z \wedge \phi(z))), \tag{2.3}$$

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

$$\forall x \exists X \leq x\, \forall y < x\, y \in X \equiv \phi(y), \tag{2.4}$$

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 32 (The theory V^1)**.** $\Sigma^{1,b}_0 = \Pi^{1,b}_0$-formulas are formulas with all number quantifiers bounded and with no set-sort quantifiers. Classes $\Sigma^{1,b}_1$ and $\Pi^{1,b}_1$ are the smallest classes of ${\mathcal{L}^2}_{\mathcal{PA}}$-formulas such that:

1. $\Sigma^{1,b}_0 \cup \Pi^{1,b}_0 \subseteq \Sigma^{1,b}_1 \cap \Pi^{1,b}_1$,
2. both $\Sigma^{1,b}_1$ and $\Pi^{1,b}_1$ are closed under $\vee$ and $\wedge$,
3. the negation of a formula $\Sigma^{1,b}_1$ is in $\Pi^{1,b}_1$ and vice versa,
4. if $\phi \in \Sigma^{1,b}_1$, then also $\exists X \leq t\, \phi \in \Sigma^{1,b}_1$,
5. if $\phi \in \Pi^{1,b}_1$, then also $\forall X \leq t\, \phi \in \Pi^{1,b}_1$.

The theory $I\Sigma^{1,b}_0$ is a second-order theory and it is axiomatized by 2-BASIC and the IND scheme for all $\Sigma^{1,b}_0$-formulas. The teory V^0 expands $I\Sigma^{1,b}_0$ by having also bounded comprehension axioms $\Sigma^{1,b}_0$-CA. The theory V^0 is a conservative extension of $I\Sigma^{1,b}_0$ with respect to $\Sigma^{1,b}_0$-consequences: if γ is a $\Sigma^{1,b}_0$-formula and V^0 proves its universal closure, so does $I\Sigma^{1,b}_0$. Finally, the theory V^1 extends V^0 by accepting the IND scheme for all $\Sigma^{1,b}_1$-formulas.

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

In this section, 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

$$subS(B,A) \iff |A| = |B| \wedge \forall i < |B| \, (B(i) \to A(i)). \tag{2.5}$$

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

$$\begin{aligned} PsubS(B,A) \iff |A| = |B| \wedge \forall i < |B| \, (B(i) \to A(i)) \wedge \\ \exists j < |A|, B(j) \wedge \exists i < |A|, \, A(i) \wedge \neg B(i). \end{aligned} \tag{2.6}$$

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

$$\langle x, y \rangle = \frac{(x+y)(x+y+1)}{2} + y. \tag{2.7}$$

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

- $\forall x_1, x_2, y_1, y_2 \; (\langle x_1, y_1 \rangle = \langle x_2, y_2 \rangle \to x_1 = x_2 \wedge y_1 = y_2)$,
- $\forall z \exists x, y \; (\langle x, y \rangle = z)$,
- $\forall x, y \;\; (x, y \leq \langle x, y \rangle < (x+y+1)^2)$.

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

$$\langle x_1, x_2, ..., x_k \rangle = \langle ... \langle \langle x_1, x_2 \rangle, x_3 \rangle, ..., x_k \rangle, \tag{2.8}$$

where

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

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

Notation 7. *For any set H, $m \geq 2$: $H(x_1, ..., x_m) =_{def} H(\langle x_1, ..., x_m \rangle)$.*

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 $Z : X_1, ..., X_n \to Y$ 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

$$\begin{aligned} MAP(A, n, B, m, H) \iff \forall i \in A \exists j \in B \wedge H(i) = j \wedge \\ \forall i \in A \, \forall j_1, j_2 \in B \, (H(i) = j_1 \wedge H(i) = j_2 \to j_1 = j_2). \end{aligned} \tag{2.9}$$

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:

$$\begin{array}{c} Census(X, \#X) \iff \#X \leq \langle |X|, |X| \rangle \wedge \#X(0) = 0 \wedge \forall x < |X| \\ (x \in X \to \#X(x+1) = \#X(x) + 1 \wedge x \notin X \to \#X(x+1) = \#X(x)). \end{array} \tag{2.10}$$

Lemma 9. *For any set X, V^1 proves that there exists its census function.*

Proof. Given any set X, consider $\Sigma^{1,b}_1$-induction on $n \leq |X|$ for the formula

$$\begin{array}{c} \phi(n) = \exists H \leq \langle n, n \rangle \, H(0) = 0 \wedge \forall \, 0 \leq x < n \\ (x \in X \to H(x+1) = H(x) + 1 \wedge x \notin X \to H(x+1) = H(x)). \end{array} \tag{2.11}$$

□

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

$$x|y \iff \exists z \leq y(xz = y). \tag{2.12}$$

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

$$Prime(p) \iff 1 < p \wedge \forall y < p \forall z < p \, (yz \neq p). \tag{2.13}$$

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

$$c = a \dot{-} b \longleftrightarrow ((b + c = a) \vee (a \leq b \wedge c = 0)), \tag{2.14}$$

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

$$c = a/b \longleftrightarrow (bc \leq a \wedge a < b(c+1)). \tag{2.15}$$

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

$$a \, mod \, p = a \dot{-} (p \cdot a/p). \tag{2.16}$$

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

$$\begin{array}{c} c_1 \dot{-} (p \cdot c_1/p) = c_2 \dot{-} (p \cdot c_2/p), \\ c_2 \dot{-} c_1 = p(c_2/p \dot{-} c_1/p), \end{array} \tag{2.17}$$

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 $y_1 \equiv y_2 (mod \, p)$,

$$\begin{array}{c} (x_1 + y_1) \equiv (x_2 + y_2) (mod \, p), \\ (x_1 y_1) \equiv (x_2 y_2) (mod \, p). \end{array} \tag{2.18}$$

2.3 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.

2.3.1 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 29 gives a necessary and sufficient condition of polynomial completeness.

Theorem 16 ([5])**.** *A finite algebra is polynomially complete if and only if it has the ternary discriminator as a polynomial operation.*

2.3.2 Linear algebras: properties and examples on digraphs

Definition 33 (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 10 ([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

$$\begin{aligned} \Omega(x_1, ..., x_i, x_{i+1}, 0, ..., 0) = \Omega(x_1, ..., x_i, 0, 0, ..., 0) + h(x_{i+1}) = \\ = h(x_1) + ... + h(x_i) + h(x_{i+1}). \end{aligned} \tag{2.19}$$

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. □

If we additionally assume that Ω is special (by Lemma 7), then $t = 1$:

$$\begin{aligned} \Omega(x, ..., x, \Omega(x, ..., x, y)) &= \Omega(x, ..., x, y), \\ \underbrace{tx + ... + tx}_{m-1} + t\Omega(x, ..., x, y) &= \underbrace{tx + ... + tx}_{m-1} + ty, \\ t(\underbrace{tx + ... + tx}_{m-1} + ty) &= ty, \\ \underbrace{tx + ... + tx}_{m-1} + ty + tx &= y + tx \\ x + ty = y + tx \implies t &= 1. \end{aligned} \tag{2.20}$$

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 17 (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 $\mathbb{Z}_{p_i}$.

$$\begin{aligned} f(x,y,z) = x - y + z\, (mod\, p_i) = \Omega(x, z, 0, ..., 0) + \Omega(y, ..., y, 0, 0) = \\ = x + z + y + ... + y = \Omega(x, z, y, ..., y). \end{aligned} \tag{2.21}$$

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. □

In the remainder of this section 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 11, 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.

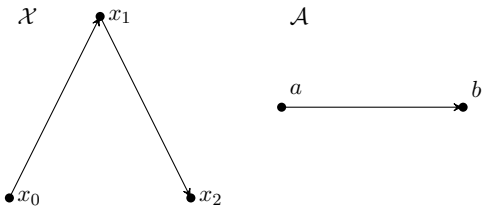

Figure 2.4: Example 1.

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 2.4, 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:

$$\begin{aligned} \Omega(a,a,a) = a, \;\; \Omega(b,b,b) = b, \\ \Omega(b,a,a) = \Omega(a,b,a) = \Omega(a,a,b) = b, \\ \Omega(a,b,b) = \Omega(b,a,b) = \Omega(b,b,a) = a. \end{aligned} \tag{2.22}$$

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

$$\begin{aligned} \Omega(a,a,\Omega(a,a,b)) &= \Omega(a,a,b) = b, \\ \Omega(b,b,\Omega(b,b,a)) &= \Omega(b,b,a) = a. \end{aligned} \tag{2.23}$$

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

$$E_{\mathcal{X}}(x_0, x_1) \iff \begin{cases} x_0 = a, \\ x_1 = b. \end{cases}$$

$$E_{\mathcal{X}}(x_1, x_2) \iff \begin{cases} x_1 = a, \\ x_2 = b. \end{cases}$$

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 2.5, 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.

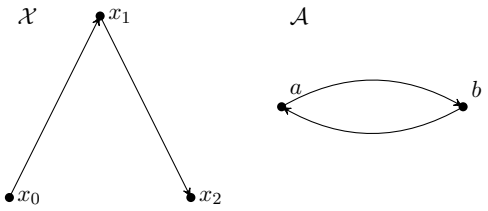

Figure 2.5: Example 2.

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

$$\begin{aligned} E_{\mathcal{X}}(x_0, x_1) &\iff x_0 + x_1 = b; \\ E_{\mathcal{X}}(x_1, x_2) &\iff x_1 + x_2 = b. \end{aligned} \tag{2.24}$$

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.

2.3.3 Zhuk's four-cases theorem

Zhuk's algorithm is based on the following theorem:

Theorem 18 ([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,*

- $\mathbb{A}$ *has a nontrivial centrally absorbing subuniverse,*
- $\mathbb{A}$ *has a nontrivial PC quotient,*
- $\mathbb{A}$ *has a nontrivial affine quotient.*

2.4 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 sections. 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 7, 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 $\mathrm{CSP}(\Gamma')$ is an instance of $\mathrm{CSP}(\Gamma)$.

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

2.4.1 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 $\mathrm{CSP}(\Gamma)$, $\Theta = (X, D, C)$. 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 19, 20 and 21, proved by Zhuk in [19], if the reduced instance has no solution, then so does the initial one.

Theorem 19 ([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 20 ([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 21 ([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 18 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.

2.4.2 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 37 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

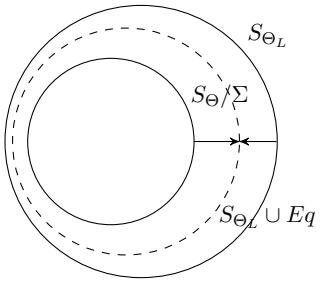

Figure 2.6: Solution sets.

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 37, $S_{\Theta_L} \subseteq (\mathbb{Z}_{p_1})^{n_1} \times ... \times (\mathbb{Z}_{p_k})^{n_k}$ 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.

2.5 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.

2.5.1 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 Sections 2.2.4 and 2.2.5.

2.5.1.1 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 15. 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.

2.5.1.2 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 $(\Gamma^1_{\mathcal{A}}, \Gamma^2_{\mathcal{A}})$ where $\Gamma^1_{\mathcal{A}}$ is the set which encodes all unary relations from $\Gamma_{\mathcal{A}}$, and $\Gamma^2_{\mathcal{A}}$ encodes binary relations,

$$\Gamma^1_{\mathcal{A}}(j,a) \iff D^1_j(a) \text{ and } \Gamma^2_{\mathcal{A}}(i,a,b) \iff E^2_i(a,b).$$

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 $\Gamma^2_{\mathcal{A}}$ 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

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

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

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

$$\begin{gathered} OP_m(F,A) \iff \forall x_0,...,x_{m-1} \in A, \exists y \in A,\ F(x_0,...,x_{m-1}) = y \wedge \\ \wedge \forall y_1, y_2 \in A\ (F(x_0,...,x_{m-1}) = y_1 \wedge F(x_0,...,x_{m-1}) = y_2 \to y_1 = y_2). \end{gathered} \tag{2.25}$$

An idempotent operation F is defined straightforwardly:

$$IDM_m(F,A) \iff OP_m(F,A) \wedge \forall a \in A\ F(a,a,...,a) = a. \tag{2.26}$$

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

$$\begin{gathered} wNU_m(\Omega,A) \iff OP_m(\Omega,A) \wedge \forall a,b \in A, \exists c \in A, \forall x_0,...,x_{m-1} \in A \\ \bigwedge_{t<m} (x_t = a \wedge \forall j \neq t < m,\ x_j = b \to \Omega(x_0,...,x_{m-1}) = c). \end{gathered} \tag{2.27}$$

A special WNU operation is defined as follows:

$$\begin{gathered} SwNU_m(\Omega,A) \iff wNU_m(\Omega,A) \wedge IDM_m(\Omega,A) \\ \forall a,b \in A, \exists c \in A,\ \Omega(a,...,a,b) = c \wedge \Omega(a,...,a,c) = c. \end{gathered} \tag{2.28}$$

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

$$\begin{gathered} Pol_{m,2}(F,A,R) \iff OP_m(F,A) \wedge \forall a^0_1,...,a^{m-1}_1, a^0_2,...,a^{m-1}_2 \in A, \\ \forall b_1,b_2 \in A,\ R(a^0_1,a^0_2) \wedge ... \wedge R(a^{m-1}_1,a^{m-1}_2) \wedge \\ \wedge F(a^0_1,...,a^{m-1}_1) = b_1 \wedge F(a^0_2,...,a^{m-1}_2) = b_2 \to R(b_1,b_2). \end{gathered} \tag{2.29}$$

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

$$\begin{gathered} Pol_{m,1}(F,A,R) \iff OP_m(F,A) \wedge \forall a^0, a^1, ..., a^{m-1} \in A, \\ \forall b \in A\ R(a^0) \wedge ... \wedge R(a^{m-1}) \wedge \\ \wedge F(a^0, ..., a^{m-1}) = b \rightarrow R(b). \end{gathered} \tag{2.30}$$

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

2.5.1.4 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

$$subTA(\mathbb{B}, \mathbb{A}) \iff |B| = |A| \wedge \forall i < l, B(i) \rightarrow A(i) \wedge SwNU(\Omega, B). \tag{2.31}$$

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

$$\begin{gathered} DP_{m,k}(B, \Omega, A_0, \Omega_0, ..., A_{k-1}, \Omega_{k-1}) \iff \forall a_0 \in A_0, ..., a_{k-1} \in A_{k-1}, \\ B(a_0, ..., a_{k-1}) \wedge \forall a_0^1, a_0^2, ..., a_0^m \in A_0, ..., a_{k-1}^1, a_{k-1}^2, ..., a_{k-1}^m \in A_{k-1} \\ \Omega(a_0^1, a_0^2, ..., a_0^m, ..., a_{k-1}^1, a_{k-1}^2, ..., a_{k-1}^m) = \\ = (\Omega_0(a_0^1, a_0^2, ..., a_0^m), ..., \Omega_{k-1}(a_{k-1}^1, a_{k-1}^2, ..., a_{k-1}^m)). \end{gathered} \tag{2.32}$$

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:

$$\begin{gathered} subDR_k(R, A_0, ..., A_{k-1}) \iff \bigwedge_{i<k} \forall a_i \in A_i, \exists a_0 \in A_0, ..., a_{i-1} \in A_{i-1}, \\ a_{i+1} \in A_{i+1}, ..., a_{k-1} \in A_{k-1}, R(a_0, ..., a_{i-1}, a_i, a_{i+1}, ..., a_{k-1}.) \end{gathered} \tag{2.33}$$

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

$$\begin{gathered} Cong_m(A, \Omega, \sigma) \iff Pol_{m,2}(\Omega, A, \sigma) \wedge \\ \forall a \in A,\ \sigma(a,a) \wedge \forall a, b \in A, (\sigma(a,b) \leftrightarrow \sigma(b,a)) \wedge \\ (\forall a, b, c \in A,\ \sigma(a,b) \wedge \sigma(b,c) \rightarrow \sigma(a,c)). \end{gathered} \tag{2.34}$$

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

$$(\exists x, y \in A\, \neg\sigma(x,y)) \wedge (\exists x \neq y \in A\ \sigma(x,y)), \tag{2.35}$$

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:

$$\begin{gathered} maxCong_m(A, \Omega, \sigma) \iff Cong_m(A, \Omega, \sigma) \wedge \exists a, b \in A,\ \neg\sigma(a,b) \wedge \\ \wedge [\forall \sigma' < \langle l, l \rangle,\ (Cong_m(A, \Omega, \sigma') \wedge \exists a, b \in A,\ \neg\sigma'(a,b)) \rightarrow \\ \rightarrow \exists a, b \in A,\ \sigma(a,b) \wedge \neg\sigma'(a,b)]. \end{gathered} \tag{2.36}$$

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:

$$\begin{gathered} FS_m(A/\sigma, A, \Omega, \sigma) \iff Cong_m(A, \Omega, \sigma) \wedge \\ \forall a, b \in A,\ (\sigma(a,b) \wedge (a < b) \to \neg A/\sigma(b)) \\ \wedge (\forall a \in A (\forall a' \in A,\ \sigma(a, a') \to a \leq a') \to A/\sigma(a)). \end{gathered} \tag{2.37}$$

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

$$\begin{gathered} Rep_m(a, a/\sigma, A/\sigma, A, \Omega, \sigma) \iff FS_m(A/\sigma, A, \Omega, \sigma) \wedge \\ \sigma(a, a/\sigma) \wedge A/\sigma(a). \end{gathered} \tag{2.38}$$

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

$$\begin{gathered} FA_m(A/\sigma, \Omega/\sigma, A, \Omega, \sigma) \iff FS_m(A/\sigma, A, \Omega, \sigma) \wedge \\ \wedge (\forall a_1, ..., a_m, c \in A, \forall a_1/\sigma, ..., a_m/\sigma, c/\sigma \in A, \\ \Omega(a_1/\sigma, ..., a_m/\sigma) = c/\sigma \wedge Rep_m(c, c/\sigma, A/\sigma, A, \Omega, \sigma) \wedge \\ \wedge \bigwedge_{i<m} Rep_m(a_i, a_i/\sigma, A/\sigma, A, \Omega, \sigma) \to \Omega/\sigma(a_1, ..., a_m) = c). \end{gathered} \tag{2.39}$$

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

2.5.1.5 Encoding digraphs and CSP properties

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

Definition 34. 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 $E^{ij}_{\ddot{\mathcal{A}}}(a, b)$ (there is an edge (a, b) between D_i and D_j):

$$\begin{gathered} E_{\ddot{\mathcal{A}}}(u, v) \to \exists i, j < n\, \exists a, b < l\ u = \langle i, a \rangle \wedge v = \langle j, b \rangle \wedge \\ D_i(a) \wedge D_j(b). \end{gathered} \tag{2.40}$$

Sometimes we will use the notation $E^{ij}_{\ddot{\mathcal{A}}}(a, b)$ 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 $\mathrm{DG}(\Theta)$, 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 $E^{ij}_{\ddot{\mathcal{A}}}$ and different domains for all vertices $x_1, ..., x_n$.

Definition 35 (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

$$\begin{gathered} H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}}) \iff \exists H < \langle n, \langle n, l\rangle\rangle (MAP(V_{\mathcal{X}}, n, V_{\ddot{\mathcal{A}}}, \langle n, l\rangle, H) \wedge \\ (\forall i < n, s < \langle n, l\rangle \; H(i) = s \to \exists a < l, s = \langle i, a\rangle \wedge D_i(a)) \wedge \\ \forall i_1, i_2 < n, \forall j_1, j_2 < \langle n, l\rangle \\ (E_{\mathcal{X}}(i_1, i_2) \wedge H(i_1) = j_1 \wedge H(i_2) = j_2 \to E_{\ddot{\mathcal{A}}}(j_1, j_2)). \end{gathered} \tag{2.41}$$

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

$$\begin{gathered} HOM(\mathcal{G}, \mathcal{H}) \iff \exists H < \langle n, m\rangle (MAP(V_{\mathcal{G}}, n, V_{\mathcal{H}}, m, H) \wedge \\ \forall i_1, i_2 < n, \forall j_1, j_2 < m \\ (E_{\mathcal{G}}(i_1, i_2) \wedge H(i_1) = j_1 \wedge H(i_2) = j_2 \to E_{\mathcal{H}}(j_1, j_2))). \end{gathered} \tag{2.42}$$

Notation 8. *Sometimes we will write $\exists H < \langle n, m\rangle, HOM(\mathcal{G}, \mathcal{H}, H)$ and $\exists H < \langle n, \langle n, l\rangle\rangle$, $H\dot{O}M(\mathcal{X}, \ddot{\mathcal{A}}, H)$ 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

$$\begin{gathered} subInst(\mathcal{X}', \mathcal{X}) \iff subS(V_{\mathcal{X}'}, V_{\mathcal{X}}) \wedge subS(E_{\mathcal{X}'}, E_{\mathcal{X}}) \wedge \\ (E_{\mathcal{X}'}(x_1, x_2) \to x_1, x_2 \in V_{\mathcal{X}'}). \end{gathered} \tag{2.43}$$

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

$$\begin{gathered} CYCLE(\mathcal{C}_t) \iff (E_{\mathcal{C}_t}(0, t-1) \vee E_{\mathcal{C}_t}(t-1, 0)) \wedge \\ \forall i < (t-1) \, (E_{\mathcal{C}_t}(i, i+1) \vee E_{\mathcal{C}_t}(i+1, i)) \wedge \\ \forall i, j < (t-1)(j \neq i+1 \to (\neg E_{\mathcal{C}_t}(i, j) \wedge \neg E_{\mathcal{C}_t}(j, i)). \end{gathered} \tag{2.44}$$

We will define cycle-consistency through two homomorphisms.

Definition 36 (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

$$\begin{gathered} 1C(\mathcal{X}, \ddot{\mathcal{A}}) \iff \forall i < n, \forall a \in D_i, \forall j < n, \\ (E_{\mathcal{X}}(i, j) \to \exists b \in D_j, E^{ij}_{\ddot{\mathcal{A}}}(a, b)) \wedge (E_{\mathcal{X}}(j, i) \to \exists b \in D_j, E^{ji}_{\ddot{\mathcal{A}}}(b, a)). \end{gathered} \tag{2.45}$$

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

$$\begin{gathered} CCInst(\mathcal{X}, \ddot{\mathcal{A}}) \iff 1C(\mathcal{X}, \ddot{\mathcal{A}}) \wedge \forall k < n, \forall a \in D_k, \forall t < n, \forall V_{\mathcal{C}_t} = t, \\ \forall E_{\mathcal{C}_t} \leq 4t^2, \forall H < \langle t, n\rangle, \big[CYCLE(V_{\mathcal{C}_t}, E_{\mathcal{C}_t}) \wedge HOM(\mathcal{C}_t, \mathcal{X}, H) \wedge H(0,k) \\ \to \exists H' < \langle t, \langle t, l\rangle\rangle, H\ddot{O}M(\mathcal{C}_t, \ddot{\mathcal{A}}, H') \wedge \\ \wedge \forall i < n, j < t\,(H(j) = i \to \exists b \in D_i,\ H'(j) = \langle i, b\rangle) \wedge H'(0) = \langle k, a\rangle\big]. \end{gathered} \tag{2.46}$$

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 $D_i, D_j,\ E^{ij}_{\ddot{\mathcal{A}}}$ 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

$$\begin{gathered} PATH(\mathcal{P}_t) \longleftrightarrow \forall i < t\,(E_{\mathcal{P}_t}(i, i+1) \vee E_{\mathcal{P}_t}(i+1, i)) \wedge \\ \forall i < t, j \leq t (j \neq i+1 \to (\neg E_{\mathcal{C}_t}(i,j) \wedge \neg E_{\mathcal{C}_t}(j,i)). \end{gathered} \tag{2.47}$$

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:

$$\begin{gathered} V_{\mathcal{P}_t} = V_{\mathcal{P}_t^{-1}} = (t+1) \wedge \forall i < t, \\ E_{\mathcal{P}_t^{-1}}(i, i+1) \leftrightarrow E_{\mathcal{P}_t}(t-i, t-(i+1)) \wedge E_{\mathcal{P}_t^{-1}}(i+1, i) \leftrightarrow E_{\mathcal{P}_t}(t-(i+1), t-i). \end{gathered} \tag{2.48}$$

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

$$\begin{gathered} V_{\mathcal{P}_t \circ \mathcal{P}_m} = (t+m+1) \wedge \\ \wedge \forall i < t,\ E_{\mathcal{P}_t \circ \mathcal{P}_m}(i, i+1) \leftrightarrow E_{\mathcal{P}_t}(i, i+1) \wedge E_{\mathcal{P}_t \circ \mathcal{P}_m}(i+1, i) \leftrightarrow E_{\mathcal{P}_t}(i+1, i) \wedge \\ \wedge \forall t \leq j < (t+m), \\ E_{\mathcal{P}_t \circ \mathcal{P}_m}(j, j+1) \leftrightarrow E_{\mathcal{P}_m}(j-t, j+1-t)) \wedge \\ \wedge E_{\mathcal{P}_t \circ \mathcal{P}_m}(j+1, j) \leftrightarrow E_{\mathcal{P}_m}(j+1-t, j-t)). \end{gathered} \tag{2.49}$$

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 $H(t) = j$:

$$\begin{gathered} Path(i, j, \mathcal{X}) \iff \exists t < n, \exists V_{\mathcal{P}_t} = t, \exists E_{\mathcal{P}_t} \leq 4t^2,\ PATH(V_{\mathcal{P}_t}, E_{\mathcal{P}_t}) \wedge \\ \wedge \exists H \leq \langle t, n\rangle, HOM(\mathcal{P}_t, \mathcal{X}, H) \wedge (H(0,i) \wedge H(t,j)). \end{gathered} \tag{2.50}$$

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

$$\begin{gathered} Linked(a, b, i, j, \Theta) \iff \exists t < nl, \exists V_{\mathcal{P}_t} = t, \exists E_{\mathcal{P}_t} \leq 4t^2, \\ \exists H \leq \langle t, n\rangle,\ PATH(V_{\mathcal{P}_t}, E_{\mathcal{P}_t}) \wedge HOM(\mathcal{P}_t, \mathcal{X}, H) \wedge (H(0,i) \wedge H(t,j)) \wedge \\ \wedge \exists H' \leq \langle t, \langle t, l\rangle\rangle, H\ddot{O}M(\mathcal{P}_t, \ddot{\mathcal{A}}, H') \wedge \\ \wedge (\forall k < n, p < t,\ (H(p,k) \to \exists c \in D_k,\ H'(p) = \langle k, c\rangle)) \\ \wedge H'(0) = \langle i, a\rangle \wedge H'(t) = \langle j, b\rangle. \end{gathered} \tag{2.51}$$

Notation 9. *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 37 (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

$$LinkedInst(\mathcal{X}, \ddot{\mathcal{A}}) \iff \forall i < n, \forall a, b \in D_i,\ Linked(a,b,i,i,\Theta). \tag{2.52}$$

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

Definition 38 (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.

$$\begin{gathered} FragmInst(\mathcal{X}, \ddot{\mathcal{A}}) \iff \exists V^1_{\mathcal{X}}, \exists V^2_{\mathcal{X}}, V^1_{\mathcal{X}} = V^2_{\mathcal{X}} = n \wedge \\ \wedge PsubS(V^1_{\mathcal{X}}, V_{\mathcal{X}}) \wedge PsubS(V^2_{\mathcal{X}}, V_{\mathcal{X}}) \wedge (\forall i < n,\ V^1_{\mathcal{X}}(i) \leftrightarrow \neg V^2_{\mathcal{X}}(i)) \wedge \\ \wedge \forall i \in V^1_{\mathcal{X}}, \forall j \in V^2_{\mathcal{X}},\ \neg E_{\mathcal{X}}(i,j) \wedge \neg E_{\mathcal{X}}(j,i). \end{gathered} \tag{2.53}$$

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

$$\begin{gathered} subDSSInst(\mathcal{X}, \ddot{\mathcal{A}}) \iff \forall i < n \forall a \in D_i, \exists H' < \langle n, \langle n, l\rangle\rangle \\ H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}}, H) \wedge H(i) = \langle i, a\rangle. \end{gathered} \tag{2.54}$$

Now we are ready to define irreducibility.

Definition 39 (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

$$\begin{gathered} IRDInst(\mathcal{X}, \ddot{\mathcal{A}}) \iff \forall \mathcal{X}' = (V_{\mathcal{X}'}, E_{\mathcal{X}'}), \forall V_{\mathcal{X}'} = n, \forall E_{\mathcal{X}'} < 4n^2, \\ (subInst(\mathcal{X}', \mathcal{X}) \rightarrow \\ \rightarrow FragmInst(\mathcal{X}', \ddot{\mathcal{A}}) \vee LinkedInst(\mathcal{X}', \ddot{\mathcal{A}}) \vee subDSSInst(\mathcal{X}', \ddot{\mathcal{A}})). \end{gathered} \tag{2.55}$$

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 $E^{ij}_{\ddot{\mathcal{A}}}$ from the list.

Definition 40. 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.

$$\begin{gathered} Inst(\Theta, \Gamma_{\mathcal{A}}) \iff DG(\Theta) \wedge \forall i < n, |D_i| = l \wedge \\ \wedge \forall i, j < n, a, b < l, \exists s < |\Gamma_{\mathcal{A}}|, E_{\ddot{\mathcal{A}}}(\langle i, a\rangle, \langle j, b\rangle) \leftrightarrow \Gamma^2_{\mathcal{A}}(s, a, b) \wedge \\ \wedge \forall i < n, a < l, \exists s < |\Gamma_{\mathcal{A}}|, D_i(a) \leftrightarrow \Gamma^1_{\mathcal{A}}(s, a). \end{gathered} \tag{2.56}$$

2.5.2 Universal algebra axiom schemes

In this section 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 19, 20 and 21 (for the soundness we do not need the 'if' implication). For this section we will consider CSP instances alongside the corresponding algebras and suppose that any algebra is finite and has a special WNU term.

2.5.2.1 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:

$$T \in Clone(\Omega) \iff Pol_2(T, A, \Gamma_{\mathcal{A}}). \tag{2.57}$$

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:

$$\begin{aligned} BAsubS(B, A, T) \iff subS(B, A) \wedge \forall a \in A, \forall b \in B, \exists c_1, c_2 \in B, \\ T(a, b) = c_1 \wedge T(b, a) = c_2. \end{aligned} \tag{2.58}$$

We will formalize the 'only if' implication of Theorem 19 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 41 ($\mathrm{BA}_{\mathcal{A}}$-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 $\mathrm{BA}_{\mathcal{A}}$-axioms and consists of the finitely many formulas of the following form

$$\begin{aligned} BA_{\mathcal{A},B,D} =_{def} \forall \mathcal{X} = (V_{\mathcal{X}}, E_{\mathcal{X}}), \forall \ddot{\mathcal{A}} = (V_{\ddot{\mathcal{A}}}, E_{\ddot{\mathcal{A}}}, D_0, ..., D_{n-1}), \\ (PsubS(B, D) \wedge SwNU_m(\Omega, D) \wedge SwNU_m(\Omega, B) \wedge \\ \wedge \exists T < (3l)^{2^3}, Pol_2(T, D, \Gamma_{\mathcal{A}}) \wedge BAsubS(B, D, T) \wedge \\ \wedge Inst(\Theta, \Gamma_{\mathcal{A}}) \wedge CCInst(\mathcal{X}, \ddot{\mathcal{A}}) \wedge IRDInst(\mathcal{X}, \ddot{\mathcal{A}}) \wedge \\ \exists i < n, D_i = D \wedge \\ H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}})) \to H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}} = (V_{\ddot{\mathcal{A}}}, E_{\ddot{\mathcal{A}}}, D_0, ..., B, ..., D_{n-1})). \end{aligned} \tag{2.59}$$

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.

2.5.2.2 Central subuniverse axiom scheme

We will formalize the 'only if' implication of Theorem 20 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

$$\begin{gathered} E(X) = X \cup \{\Omega(a_1, ..., a_m) : a_1, ..., a_m \in X\} \\ \forall t \geq 0, E^0(X) = X, E^{t+1}(X) = E(E^t(X)). \end{gathered} \tag{2.60}$$

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 $E^{l^2}_X$ up to l^2

$$\begin{gathered} \forall b, c < l,\ E^0_X(b, c) \iff X(b, c) \wedge \\ \wedge \forall 0 < t < l^2, \forall b, c < l,\ E^t_X(b, c) \iff E^{t-1}_X(b, c) \vee \\ \vee \exists b_1, ..., b_m, c_1, ..., c_m \in A, E^{t-1}_X(b_1, c_1) \wedge ... \wedge E^{t-1}_X(b_m, c_m) \wedge \\ \wedge \Omega(b_1, ..., b_m) = b \wedge \Omega(c_1, ..., c_m) = c. \end{gathered} \tag{2.61}$$

The existence of this set follows from $\Sigma^{1,b}_1$-induction. A central subuniverse must be an absorbing subuniverse, namely a ternary absorbing subuniverse [20]. For any three sets A, C, S the following $\Pi^{1,b}_1$-definable relation expresses that the subset C of A is central under ternary term operation S.

$$\begin{gathered} CRsubS(C, A, S) \iff subS(C, A) \wedge \forall c_1, c_2 \in C, \forall a \in A, \exists c'_1, c'_2, c'_3 \in C, \\ S(c_1, c_2, a) = c'_1 \wedge S(c_1, a, c_2) = c'_2 \wedge S(a, c_1, c_2) = c'_3 \wedge \\ \wedge \forall a \in A \backslash C, \forall X < \langle l, l \rangle,\ ((X(a, c) \wedge X(c, a) \leftrightarrow c \in C) \rightarrow \neg E^{l^2}_X(a, a)). \end{gathered} \tag{2.62}$$

Definition 42 ($\mathrm{CR}_{\mathcal{A}}$-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 $\mathrm{CR}_{\mathcal{A}}$-axioms. The scheme embraces the finitely many formulas of the following form

$$\begin{gathered} CR_{\mathcal{A},D,C} =_{def} \forall \mathcal{X} = (V_{\mathcal{X}}, E_{\mathcal{X}}), \forall \ddot{\mathcal{A}} = (V_{\ddot{\mathcal{A}}}, E_{\ddot{\mathcal{A}}}, D_0, ..., D_{n-1}), \\ \big(PsubS(C, D) \wedge SwNU_m(\Omega, D) \wedge SwNU_m(\Omega, C) \wedge \\ \exists S < (4l)^{2^4},\ Pol_3(S, D, \Gamma_{\mathcal{A}}) \wedge CRsubS(C, D, S) \wedge \\ \wedge Inst(\Theta, \Gamma_{\mathcal{A}}) \wedge CCInst(\mathcal{X}, \ddot{\mathcal{A}}) \wedge IRDInst(\mathcal{X}, \ddot{\mathcal{A}}) \wedge \\ \exists i < n, D_i = D \wedge \\ H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}})) \rightarrow H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}} = (V_{\ddot{\mathcal{A}}}, E_{\ddot{\mathcal{A}}}, D_0, ..., C, ..., D_{n-1})). \end{gathered} \tag{2.63}$$

The formula is analogous to $\mathrm{BA}_{\mathcal{A}}$-axioms, it is again the universal closure of $\Sigma^{1,b}_2$-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.

2.5.2.3 Polynomially complete axiom scheme

Theorem 29 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:

$$Polynom(\mathbb{A}) = Clone(\Omega, a_1, ..., a_{|A|}). \tag{2.64}$$

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 $\Gamma_{\mathcal{A}}$. 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

$$\begin{gathered}\Gamma_{\mathcal{A}}^{1,diag}(j,a) \iff \Gamma_{\mathcal{A}}^{1}(j,a) \wedge (\forall b \in A, \Gamma_{\mathcal{A}}^{1}(j,b)) \\ \Gamma_{\mathcal{A}}^{2,diag}(i,a,b) \iff \Gamma_{\mathcal{A}}^{2}(i,a,b) \wedge (\forall c \in A, \Gamma_{\mathcal{A}}^{2}(j,c,c)).\end{gathered} \tag{2.65}$$

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.

$$P \in Polynom(\mathbb{A}) \iff Pol_n(P, A, \Gamma_{\mathcal{A}}^{diag}). \tag{2.66}$$

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

$$\begin{gathered}PCD(A,P) \iff \forall a,b,c \in A, \\ (a = b \wedge P(a,b,c) = c) \vee (a \neq b \wedge P(a,b,c) = a).\end{gathered} \tag{2.67}$$

Before the formalization of the 'only if' implication of Theorem 21 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:

$$\begin{gathered}\Gamma_{\mathcal{A}}^{1}/\sigma(j,a) \iff \forall a/\sigma \in A,\ Rep_m(a, a/\sigma, A/\sigma, A, \Omega, \sigma) \wedge \Gamma_{\mathcal{A}}^{1}(j, a/\sigma) \\ \Gamma_{\mathcal{A}}^{2}/\sigma(i,a,b) \iff \forall a/\sigma, b/\sigma \in A,\ \Gamma_{\mathcal{A}}^{2}(i, a/\sigma, b/\sigma) \wedge \\ \wedge Rep_m(a, a/\sigma, A/\sigma, A, \Omega, \sigma) \wedge Rep_m(b, b/\sigma, A/\sigma, A, \Omega, \sigma).\end{gathered} \tag{2.68}$$

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 43 (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

$$\begin{gathered}PC_{\mathcal{A},D,E} =_{def} \forall \mathcal{X} = (V_{\mathcal{X}}, E_{\mathcal{X}}), \forall \ddot{\mathcal{A}} = (V_{\ddot{\mathcal{A}}}, E_{\ddot{\mathcal{A}}}, D_0, ..., D_{n-1}), \\ ([\forall j < n, \forall B < l, \forall T < (3l)^{2^3}, Pol_2(T, D_j, \Gamma_{\mathcal{A}}) \to \neg BAsubS(B, D_j, T) \wedge \\ \wedge \forall j < n, \forall C < l, \forall S < (4l)^{2^4}, Pol_3(S, D_j, \Gamma_{\mathcal{A}}) \to \neg CRsubS(C, D_j, S)] \\ \wedge \exists \sigma < \langle l, l \rangle, \exists D/\sigma < l, \exists \Omega/\sigma < (ml)^{2^{m+1}}, FA_m(D/\sigma, \Omega/\sigma, D, \Omega, \sigma) \wedge \\ \wedge \exists P < (4l)^{2^4},\ Pol_3(P, D/\sigma, \Gamma_{\mathcal{D}}^{diag}/\sigma) \wedge PCD(D/\sigma, P) \wedge \\ SwNU_m(\Omega, D) \wedge PsubS(E, D) \wedge (\forall a \in E, \forall b \in D, \sigma(a,b) \leftrightarrow b \in E) \wedge \\ \wedge Inst(\Theta, \Gamma_{\mathcal{A}}) \wedge CCInst(\mathcal{X}, \ddot{\mathcal{A}}) \wedge IRDInst(\mathcal{X}, \ddot{\mathcal{A}}) \wedge \\ \exists i < n, D_i = D \wedge \\ H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}})) \to H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}} = (V_{\ddot{\mathcal{A}}}, E_{\ddot{\mathcal{A}}}, D_0, ..., E, ..., D_{n-1})).\end{gathered} \tag{2.69}$$

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.

2.5.3 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 44 (Theory $V^1_{\mathcal{A}}$)**.**

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

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 $V^1_{\mathcal{A}}$ proves the soundness of Zhuk's algorithm.

2.5.4 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.

2.5.4.1 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

$$E_{\ddot{\mathcal{A}}}(u,v) \longrightarrow \exists i,j < n\, \exists a,b < l\ u = \langle i,a\rangle \wedge v = \langle j,b\rangle \wedge \\ D_i(a) \wedge D_j(b), \tag{2.70}$$

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

$$E_{\ddot{\mathcal{R}}}(u,v) \iff \exists i,j < n\, \exists a,b < l\ u = \langle i,a\rangle \wedge v = \langle j,b\rangle \wedge \\ D_i(a) \wedge D_j(b). \tag{2.71}$$

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

$$E^{ij}_{\ddot{\mathcal{R}}_0}(a,b) \iff (a \in D_i \wedge b \in D_j) \wedge \\ \wedge (E_{\mathcal{X}}(i,j) \to E^{ij}_{\ddot{\mathcal{A}}}(a,b)) \wedge (E_{\mathcal{X}}(j,i) \to E^{ji}_{\ddot{\mathcal{A}}}(b,a)). \tag{2.72}$$

Note that if there are no constraints $E_{\mathcal{X}}(i,j)$ and $E_{\mathcal{X}}(j,i)$, then at this point both $E^{ij}_{\ddot{\mathcal{R}}_0}$ and $E^{ji}_{\ddot{\mathcal{R}}_0}$ 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 $E^{ij}_{\ddot{\mathcal{R}}_0}$ on variable i:

$$Pr_1(i,a) \iff a \in D_i \wedge \forall j < n,\ E_{\mathcal{X}}(i,j) \to \exists b \in D_j,\ E^{ij}_{\ddot{\mathcal{R}}_0}(a,b) \wedge \\ \forall k < n,\ E_{\mathcal{X}}(k,i) \to \exists c \in D_k,\ E^{ki}_{\ddot{\mathcal{R}}_0}(c,a). \tag{2.73}$$

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

$$V_{\ddot{\mathcal{R}}_1}(i,a) \iff Pr_1(i,a), \tag{2.74}$$

and

$$E^{ij}_{\ddot{\mathcal{R}}_1}(a,b) \iff Pr_1(i,a) \wedge Pr_1(j,b) \wedge E^{ij}_{\ddot{\mathcal{R}}_0}(a,b). \tag{2.75}$$

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:

$$R_1(i,j,a,b) \iff E^{ij}_{\ddot{\mathcal{R}}_1}(a,b), \tag{2.76}$$

and for $t > 1$

$$R_t(i,j,a,b) \iff R_{t-1}(i,j,a,b) \wedge \\ \forall k < n \exists c < l\ Pr_1(k,c) \wedge (R_{t-1}(i,k,a,c) \wedge R_{t-1}(k,j,c,b)). \tag{2.77}$$

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 $E^{ij}_{\ddot{\mathcal{R}}_t}$ 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 $\forall i,j < n, \forall a,b < k,\ R_t(i,j,a,b) \to R_{t-1}(i,j,a,b)$. 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

$$\begin{gathered} R_{q'+f+1}(i,j,a,b) \iff R_{q'-1}(i,j,a,b) \wedge \\ \forall k < n \exists c < l,\ Pr_1(k,c) \wedge (R_{q'-1}(i,k,a,c) \wedge R_{q'-1}(k,j,c,b)). \end{gathered} \tag{2.78}$$

Now suppose that for every $1 < t \leq (q+1)$, $\neg(R_{t-1}(i,j,a,b) \to R_t(i,j,a,b))$, 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.

$$\begin{gathered} Pr_{cc}(i,a) \iff Pr_1(i,a) \wedge \forall j < n, E_{\mathcal{X}}(i,j) \to \exists b, Pr_1(j,b) \wedge E^{ij}_{\ddot{\mathcal{R}}_{q+1}}(a,b) \\ \wedge \forall k < n,\ E_{\mathcal{X}}(k,i) \to \exists c,\ Pr_1(k,c) \wedge E^{ki}_{\ddot{\mathcal{R}}_{q+1}}(c,a). \end{gathered} \tag{2.79}$$

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

$$V_{\ddot{\mathcal{A}}_{cc}}(i,a) \iff Pr_{cc}(i,a), \tag{2.80}$$

and

$$E^{ij}_{\ddot{\mathcal{A}}_{cc}}(a,b) \iff (Pr_{cc}(i,a) \wedge Pr_{cc}(j,b)) \wedge E^{ij}_{\ddot{\mathcal{R}}_{q+1}}(a,b). \tag{2.81}$$

Remark 2. In Zhuk's algorithm, the original function CheckCycleConsistency in [19] reduces one domain D_i at a time (as if in (2.79) 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 $E^{ij}_{\ddot{\mathcal{A}}}$: 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 11. *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 (2.79)-(2.81), 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 $j < t, k < n, H(j) = k \to H'(j) = \langle k, b\rangle$ 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 (2.78), the formula

$$\exists b_1, b_2, ..., b_{t-1} < l,\ \tilde{E}^{ik_1}_{\ddot{\mathcal{A}}_{cc}}(a, b_1) \wedge \tilde{E}^{k_1k_2}_{\ddot{\mathcal{A}}_{cc}}(b_1, b_2) \wedge ... \wedge \tilde{E}^{k_{t-1}i}_{\ddot{\mathcal{A}}_{cc}}(b_{t-1}, a) \tag{2.82}$$

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

$$\begin{gathered} E^{ii}_{\ddot{\mathcal{A}}_{cc}}(a, a) \to \exists b_{t-1} < l,\ E^{ik_{t-1}}_{\ddot{\mathcal{A}}_{cc}}(a, b_{t-1}) \wedge E^{k_{t-1}i}_{\ddot{\mathcal{A}}_{cc}}(b_{t-1}, a), \\ ... \\ E^{ik_3}_{\ddot{\mathcal{A}}_{cc}}(a, b_3) \to \exists b_2 < l,\ E^{ik_2}_{\ddot{\mathcal{A}}_{cc}}(a, b_2) \wedge E^{k_2k_3}_{\ddot{\mathcal{A}}_{cc}}(b_2, b_3), \\ E^{ik_2}_{\ddot{\mathcal{A}}_{cc}}(a, b_2) \to \exists b_1 < l,\ E^{ik_1}_{\ddot{\mathcal{A}}_{cc}}(a, b_1) \wedge E^{k_1k_2}_{\ddot{\mathcal{A}}_{cc}}(b_1, b_2). \end{gathered} \tag{2.83}$$

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

Lemma 12. *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 $E^{ij}_{\ddot{\mathcal{A}}}(a, b)$ for $a \in D_i, b \in D_j$. Due to the definition of a homomorphism for both a and b, $E^{ij}_{\ddot{\mathcal{A}}}$ must satisfy (2.72)-(2.75) and we do not lose any solution after the intersection of all constraints. That is, instead of the set $\{\mathcal{X} \to \ddot{\mathcal{A}}\}$ we can consider set $\{\mathcal{X} \to \ddot{\mathcal{R}}_1\}$.

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 $a, b, c < l$

$$\begin{gathered} \phi(t) = H\ddot{O}M(\mathcal{X}, \mathcal{R}'_1, H) \wedge H(i) = \langle i, a\rangle \wedge H(j) = \langle j, b\rangle \wedge H(k) = \langle k, c\rangle \longrightarrow \\ (E^{ij}_{\ddot{\mathcal{R}}_t}(a, b) \wedge E^{ik}_{\ddot{\mathcal{R}}_t}(a, c) \wedge E^{kj}_{\ddot{\mathcal{R}}_t}(c, b)). \end{gathered} \tag{2.84}$$

For $t = 1$ this is true. For every constraint $E_{\mathcal{X}}(i, j)$ the implication $E^{ij}_{\ddot{\mathcal{R}}_1}(a, b)$ follows from the definition of a homomorphism. For any i, j such that $\neg E_{\mathcal{X}}(i, j)$ the implication $E^{ij}_{\ddot{\mathcal{R}}_1}(a, b)$ follows from the definition of $E^{ij}_{\ddot{\mathcal{R}}_0}$ and (2.73)-(2.75): 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 (2.77). Hence, $\{\mathcal{X} \to \ddot{\mathcal{A}}\} \subseteq \{\mathcal{X} \to \ddot{\mathcal{A}}_{cc}\}$. The opposite inclusion is trivial. □

2.5.4.2 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 $E^{jk}_{\ddot{\mathcal{A}}}$ (or $E^{kj}_{\ddot{\mathcal{A}}}$) or domains D_j, D_k. On domain D_k of such variable k, the algorithm generates relation σ_k as follows:

$$E_{\mathcal{X}}(j,k) : \sigma_k(a,b) \iff \exists a', b' \in D_j,\ \sigma_j(a',b') \wedge E^{jk}_{\ddot{\mathcal{A}}}(a,a') \wedge E^{jk}_{\ddot{\mathcal{A}}}(b,b'). \tag{2.85}$$

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 (2.85). 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:

$$\begin{gathered}
\forall a,b < l,\ I_{0,i,q}(i,a,b) \iff \sigma_i^q(a,b) \wedge \\
\wedge \forall 0 < t < n, k < n, a, b < l,\ I_{t,i,q}(k,a,b) \iff I_{t-1,i,q}(k,a,b) \vee \\
\vee \exists j < n, a', b' < l,\ I_{t-1,i,q}(j,a',b') \wedge \\
\wedge (E_{\mathcal{X}}(j,k) \wedge E^{jk}_{\ddot{\mathcal{A}}}(a',a) \wedge E^{jk}_{\ddot{\mathcal{A}}}(b',b)) \vee (E_{\mathcal{X}}(k,j) \wedge E^{kj}_{\ddot{\mathcal{A}}}(a,a') \wedge E^{kj}_{\ddot{\mathcal{A}}}(b,b')) \wedge \\
\wedge \neg [\exists c,d \in D_j, \exists e \in D_k,\ \neg I_{t-1,i,q}(j,c,d) \wedge \\
\wedge (E_{\mathcal{X}}(j,k) \wedge E^{jk}_{\ddot{\mathcal{A}}}(c,e) \wedge E^{jk}_{\ddot{\mathcal{A}}}(d,e)) \vee (E_{\mathcal{X}}(k,j) \wedge E^{kj}_{\ddot{\mathcal{A}}}(e,c) \wedge E^{kj}_{\ddot{\mathcal{A}}}(e,d))].
\end{gathered} \tag{2.86}$$

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

$$\begin{gathered}\forall V_{\mathcal{X}'} < n, \forall E_{\mathcal{X}'} < 4n^2,\ \mathcal{X}' = (V_{\mathcal{X}'}, E_{\mathcal{X}'}),\\ ((\forall j < n, \exists a, b < l,\ V_{\mathcal{X}'}(j) \leftrightarrow I_{n,i,q}(j,a,b)) \wedge\\ \wedge(\forall s, s' < n,\ E_{\mathcal{X}'}(s,s') \to s, s' \in V_{\mathcal{X}'}) \wedge\\ \wedge(\forall s, s' \in V_{\mathcal{X}'},\ E_{\mathcal{X}'}(s,s') \leftrightarrow E_{\mathcal{X}}(s,s'))) \to subDSSInst(\mathcal{X}', \ddot{\mathcal{A}}_{ir}).\end{gathered} \tag{2.87}$$

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 13. *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 $Linked(a, b, i, i, \Theta)$.

$$\begin{gathered}Linked(a,b,i,i,\Theta) \iff \exists t < nl, V_{\mathcal{P}_t} = t, E_{\mathcal{P}_t} \leq t^2,\\ PATH(V_{\mathcal{P}_t}, E_{\mathcal{P}_t}) \wedge \exists H \leq \langle t, n\rangle, HOM(\mathcal{P}_t, \mathcal{X}, H) \wedge (H(0,i) \wedge H(t,i)) \wedge\\ \wedge \exists H' \leq \langle t, \langle t, l\rangle\rangle, H\ddot{O}M(\mathcal{P}_t, \ddot{\mathcal{A}}, H') \wedge\\ \wedge(\forall k < n, p < t,\ (H(p,k) \to \exists c \in D_k,\ H'(p) = \langle k, c\rangle))\\ \wedge H'(0) = \langle i, a\rangle \wedge H'(t) = \langle i, b\rangle.\end{gathered} \tag{2.88}$$

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 $\Gamma^2_{\mathcal{A}}$. □

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 14. *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 $\theta(t)$

$$\begin{gathered}\theta(t) =_{def} t < n,\ i \in V^1_{\mathcal{X}}, j \in V^2_{\mathcal{X}} \wedge V^1_{\mathcal{X}} = V^2_{\mathcal{X}} = n \wedge \#V^2_{\mathcal{X}}(n) = t \wedge \\ \wedge PsubS(V^1_{\mathcal{X}}, V_{\mathcal{X}}) \wedge PsubS(V^2_{\mathcal{X}}, V_{\mathcal{X}}) \wedge (\forall k < n,\ V^1_{\mathcal{X}}(k) \leftrightarrow \neg V^1_{\mathcal{X}}(k)) \wedge \\ \wedge \exists m \leq t, \exists \mathcal{P}_m,\ V_{\mathcal{P}_m} = m, E_{\mathcal{P}_m} < m^2, PATH(V_{\mathcal{P}_m}, E_{\mathcal{P}_m}) \wedge \\ \wedge \exists H \leq \langle m, n\rangle, HOM(\mathcal{P}_m, \mathcal{X}, H) \wedge H(0,j) \wedge H(m,i') \wedge i' \in V^1_{\mathcal{X}}.\end{gathered} \tag{2.89}$$

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 $V^2_{\mathcal{X}}$, $\#V^2_{\mathcal{X}}(n) = (s+1)$ there are two elements $i' \in V^1_{\mathcal{X}}$ and $j' \in V^2_{\mathcal{X}}$ such that there is an edge $E_{\mathcal{X}}(i', j')$ or $E_{\mathcal{X}}(j', i')$. Then consider two sets $V^1_{\mathcal{X}} \cup \{j'\}$ and $V^2_{\mathcal{X}} \backslash \{j'\}$. Since $\#V^2_{\mathcal{X}} \backslash \{j'\} = s$, 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 $i'' \in V^1_{\mathcal{X}} \cup \{j'\}$. 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 $i'' \in V^1_{\mathcal{X}}$. Finally, it also must be true for $t = n - 1$. □

Lemma 15. *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 14 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:

$$\begin{aligned} V_{\mathcal{C}_{2s}} = 2m \wedge \forall k < m,\, E_{\mathcal{C}_{2s}}(k, k+1) \leftrightarrow E_{\mathcal{P}_s}(k, k+1) \wedge \\ \wedge \leftrightarrow E_{\mathcal{C}_{2s}}(k+1, k) \leftrightarrow E_{\mathcal{P}_s}(k+1, k) \wedge \\ \wedge \forall r < (s-1),\, E_{\mathcal{C}_{2s}}(s+r, s+r+1) \leftrightarrow E_{\mathcal{P}_s^{-1}}(r, r+1) \wedge \\ \wedge E_{\mathcal{C}_{2s}}(s+r+1, s+r) \leftrightarrow E_{\mathcal{P}_s^{-1}}(r+1, r) \wedge \\ \wedge E_{\mathcal{C}_{2s}}(2s-1, 0) \leftrightarrow E_{\mathcal{P}_s^{-1}}(s-1, s) \wedge E_{\mathcal{C}_{2m}}(0, 2s-1) \leftrightarrow E_{\mathcal{P}_s^{-1}}(s, s-1). \end{aligned} \tag{2.90}$$

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. □

Remark 4. Note that in proof of Lemma 15 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 16. *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 15, 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:

$$\begin{aligned} \forall j < n, \exists a, b < l,\, V_{\mathcal{X}''}(j) \leftrightarrow I_{n,i,q}(j, a, b) \wedge \\ \wedge \forall s, s' < n,\, E_{\mathcal{X}''}(s, s') \to s, s' \in V_{\mathcal{X}''} \wedge \\ \wedge \forall s, s' \in V_{\mathcal{X}''},\, E_{\mathcal{X}''}(s, s') \leftrightarrow E_{\mathcal{X}}(s, s'). \end{aligned} \tag{2.91}$$

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 $a', b' \in D_j$,

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

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

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

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 14 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

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

and the only thing we have to check due to defining equation (2.86) 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

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

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 $s = t \to s = (t + 1)$, 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 (2.86) hold. Note that for $s = 0, 1$ we established $Linked(c, c', i, k, \Theta') \wedge Linked(d, d', i, k, \Theta')$, 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 $s = t \to s = (t + 1)$ 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 $I_{n,i,q}(i, a, b) \wedge Linked(a, a', i, k, \Theta) \wedge Linked(b, b', i, k, \Theta)$ 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. □

Lemma 17. *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 $\sigma^q_{i_0}$ 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 $E^{st}_{\ddot{\mathcal{A}}}$ 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 $\mathcal{X}'$.

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 $H \restriction_{X'}$ from $\mathcal{X}'$ to $\ddot{\mathcal{A}}$ as follows:

$$\forall i \in \{i_0, i_1, ..., i_g\},\ H \restriction_{X'} (i) = \langle i, a\rangle \iff H(i) = \langle i, a\rangle. \tag{2.92}$$

That $H \restriction_{X'}$ is a homomorphism follows right from the definition of H. Therefore, $\{H \restriction_{X'}\} \subseteq \{\mathcal{X}' \to \ddot{\mathcal{A}}\}$. 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 $\langle j, b_1\rangle, ..., \langle j, b_s\rangle$. □

2.5.4.3 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 $E^{ij}_{\ddot{\mathcal{A}}}$ for an edge $E_{\mathcal{X}}(i, j)$. Since Θ is cycle-consistent, projections $pr_i(E^{ij}_{\ddot{\mathcal{A}}})$ and $pr_j(E^{ij}_{\ddot{\mathcal{A}}})$ are equal to D_i, D_j. The algorithm never increases domains, so we weaken only binary constraints and replace each $E^{ij}_{\ddot{\mathcal{A}}}$ 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 $E^{ij}_{\ddot{\mathcal{A}}}$ 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 $E^{ij}_{\ddot{\mathcal{A}}}$ 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 18. *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. □

2.5.5 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.

2.5.5.1 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:

$$c = a +_{(mod\, p)} b \longleftrightarrow c < p \wedge c \equiv (a+b)\,(mod\, p). \tag{2.93}$$

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:

$$c = m \cdot_{(mod\, p)} a \longleftrightarrow c < p \wedge c \equiv (ma)\,(mod\, p). \tag{2.94}$$

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}:

$$c = a^{-1} \longleftrightarrow c < p \wedge c \neq 0 \wedge c \cdot_{(mod\, p)} a = a \cdot_{(mod\, p)} c = 1. \tag{2.95}$$

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

$$C = A + B \longleftrightarrow MX_{m\times n,p}(C) \wedge \forall i < m, j < n\ C_{ij} = A_{ij} +_{(mod\, p)} B_{ij}, \tag{2.96}$$

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

$$C = bA \longleftrightarrow MX_{m\times n,p}(C) \wedge \forall i < m, j < n\ C_{ij} = b \cdot_{(mod\, p)} A_{ij}. \tag{2.97}$$

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

$$\begin{gathered}\phi(i, j, t, A, B) =_{def} \exists X < \langle t, p\rangle,\ X_0 = A_{i0} \cdot_{(mod\, p)} B_{0j} \wedge \\ \forall 0 < k < t\, X_k = X_{k-1} +_{(mod\, p)} A_{ik} \cdot_{(mod\, p)} B_{kj}.\end{gathered} \tag{2.98}$$

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:

$$\begin{gathered} C = AB \longleftrightarrow MX_{m\times s,p}(C) \wedge \forall i < m, j < s \\ C_{ij} = A_{i0} \cdot_{(mod\, p)} B_{0j} +_{(mod\, p)} \dots +_{(mod\, p)} A_{i(n-1)} \cdot_{(mod\, p)} B_{(n-1)j}. \end{gathered} \tag{2.99}$$

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

2.5.5.2 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

$$\begin{gathered} E_{\ddot{\mathcal{A}}_L}(s, r) \iff \exists i, j < n\, \exists a, b < l,\ s = \langle i, a\rangle \wedge r = \langle j, b\rangle \wedge \\ D_i/\sigma_i(a) \wedge D_j/\sigma_j(b) \wedge (\exists c, d < l,\ \sigma(i, a, c) \wedge \sigma(j, b, d) \wedge E^{ij}_{\ddot{\mathcal{A}}}(c, d)). \end{gathered} \tag{2.100}$$

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 $E^{ij}_{\ddot{\mathcal{A}}} \cap [a]/\sigma_i \times [b]/\sigma_j \neq \emptyset$. 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 22. *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 $\Sigma^{1,b}_0$-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

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

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. □

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

$$\begin{gathered} EL^I_{m\times m,p}(E) \iff MX_{m\times m,p}(E) \wedge \exists i' \neq j' < m \forall i, j < m \\ (i \neq i' \wedge i \neq j' \to E_{ii} = 1) \wedge (i \neq i' \wedge j \neq j' \wedge i \neq j \to E_{ij} = 0) \\ \wedge (E_{i'i'} = 0 \wedge E_{j'j'} = 0 \wedge E_{i'j'} = 1 \wedge E_{j'i'} = 1), \end{gathered} \tag{2.101}$$

the second one corresponds to row-multiplying transformations

$$\begin{gathered} EL^{II}_{m\times m,p}(E) \iff MX_{m\times m,p}(E) \wedge \exists a \neq 0 \in \mathbb{Z}_p \exists i' < m \\ \forall i, j < m (i \neq i' \to E_{ii} = 1) \wedge (i \neq j \to E_{ij} = 0) \wedge E_{i'i'} = a, \end{gathered} \tag{2.102}$$

and the last one corresponds to row-addition transformations

$$\begin{gathered} EL^{III}_{m\times m,p}(E) \iff MX_{m\times m,p}(E) \wedge \exists a \neq 0 \in \mathbb{Z}_p \exists i', j' < m \forall i, j < m \\ (E_{ii} = 1 \wedge (i \neq j \wedge i \neq i' \wedge j \neq j' \to E_{ij} = 0) \wedge E_{i'j'} = a. \end{gathered} \tag{2.103}$$

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 19. *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. □

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 Section 2.4.2). 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 $E^{ij}_{\ddot{\mathcal{A}}}$ (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 $E^{ij}_{\ddot{\mathcal{A}}}$ 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 23. *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 $\theta(t)$,

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

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

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. □

Lemma 20. *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 $\Theta_{t,s}$. □

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

Theorem 24. *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 23. 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

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

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

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 $(q - r) \leq \#E_{\mathcal{X}_r}$. 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 $V_{\ddot{\mathcal{A}}}$. □

2.5.6 The main result

Theorem 25 (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 $V^1_{\mathcal{A}}$ 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 $V^1_{\mathcal{A}}$ proves that W_i has a solution only if W_{i+1} has a solution.

In Section 2.5.4 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 12);

- the instance Θ has a solution only if it has a solution after irreducible reduction (Lemma 17);
- the instance Θ has a solution only if it has a solution after the weaker instance reduction (Lemma 18).

The three universal algebra axiom schemes $\mathrm{BA}_{\mathcal{A}}$-axioms, $\mathrm{CR}_{\mathcal{A}}$-axioms, and $\mathrm{PC}_{\mathcal{A}}$-axioms defined in Section 2.5.2 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 2.5.5 we have shown that V^1 validates:

- factorization of the instance by minimal linear congruences (Theorem 22);
- Gaussian elimination (Lemma 19);
- decreasing of the solution set to the factorized instance (Theorems 23, 24, Lemma 20).

This completes the proof. □

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.

2.6 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 15 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.

Acknowledgements: I would like to thank my supervisor Jan Krajíček for many helpful comments that resulted in many improvements to this paper. I thank Dmitriy Zhuk for answering my questions about his results. Also, I'm grateful to Michael Kompatscher for a number of discussions on universal algebra. Finally, I would like to thank Emil Jeřábek for the expert remarks on formalization in bounded arithmetic.

Bibliography

[1] Albert Atserias and Joanna Ochremiak. Proof complexity meets algebra. *ACM Trans. Comput. Logic*, 20(1), 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. *ArXiv*, abs/2210.07383, 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 Sankappanavar. *A Course in Universal Algebra*, volume 91. 01 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. The computational structure of monotone monadic snp and constraint satisfaction: A study through datalog and group theory. *SIAM Journal on Computing*, 28(1):57–104, 1998.

[12] R. Freese, R. McKenzie, R.M. Kenzie, and S.P.G.N.J. Hitchin. *Commutator Theory for Congruence Modular Varieties*. Lecture note series / London mathematical society. Cambridge University Press, 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, 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.

3. Proof complexity of universal algebra in a proof of CSP dichotomy

This chapter is the third part of the longer project, aiming to establish the soundness of Zhuk's algorithm. By soundness, we mean the formula $Reject_{\mathcal{A}}(\mathcal{X}, W) \implies \neg HOM(\mathcal{X}, \mathcal{A})$, where $Reject_{\mathcal{A}}(\mathcal{X}, W)$ formalizes naturally that W is the algorithm computation on input $\mathcal{X}$ that results in rejection, and $\neg HOM(\mathcal{X}, \mathcal{A})$ means that there is no homomorphism from $\mathcal{X}$ to $\mathcal{A}$. In Chapter 2 (ref. [6]) we have shown that for any fixed relational structure $\mathcal{A}$ that corresponds to an algebra $\mathbb{A}$ with WNU operation, the theory $V^1_{\mathcal{A}}$ proves the soundness of Zhuk's algorithm, where $V^1_{\mathcal{A}}$ extends the theory V^1 with three universal algebra axiom schemes $\mathrm{BA}_{\mathcal{A}}$-axioms, $\mathrm{CR}_{\mathcal{A}}$-axioms, and $\mathrm{PC}_{\mathcal{A}}$-axioms. These axiom schemes consist of finitely many $\forall\Sigma^{1,b}_2$-formulas and reflect the main theorems in Zhuk's paper [15], namely Theorems 5.5, 5.6. Informally, they state that by reducing a domain of an instance to its binary absorbing subuniverse, central subuniverse, or PC subuniverse (see [16]), the algorithm does not lose all the solutions to the instance. For the formal definition of these axiom schemes in Section 3.2.9, we first repeat the formalization of all the necessary notions.

In this chapter, we have used the bounded arithmetic W^1_1 introduced in [10] to formalize the proofs of these three axiom schemes. Formalization, together with the known relation of the theory to propositional calculus G, completes the proof of the following main theorem. Recall the translation of first-order formulas to propositional ones [4].

Theorem 26 (The main theorem)**.** *For any particular relational structure $\mathcal{A}$ such that CSP($\mathcal{A}$) is in P:*

1. *Theory W^1_1 proves the soundness of Zhuk's algorithm. That is, the theory proves the formula $Reject_{\mathcal{A}}(\mathcal{X}, W) \implies \neg HOM(\mathcal{X}, \mathcal{A})$.*
2. *There exists a p-time algorithm F such that for any unsatisfiable instance $\mathcal{X}$, i.e. such that $\neg HOM(\mathcal{X}, \mathcal{A})$, the output $F(\mathcal{X})$ of F on $\mathcal{X}$ is a propositional proof of the proposition translation of formula $\neg HOM(\mathcal{X}, \mathcal{A})$ in propositional calculus G.*

3.1 Preliminaries

3.1.1 Auxiliary relations and functions

For any two relations R_1, R_2 of the same arity, we will use standard denotations for $R_1 \subseteq R_2$, $R_1 \subsetneq R_2$, $R_1 = R_2$, $R_1 \neq \emptyset$, as opposed to Chapter 1 (ref. [5]) and Chapter 2 (ref. [6]), where we wanted to stay in the classical low-level relational-set notation of bounded arithmetic. We introduce the following relations and functions as in [4]. The string function $row(i, Z)$, or Z_i, representing the row i of a binary array Z, has a bit-defining axiom:

$$Z_i(a) = row(i, Z)(a) \iff (a < |Z| \wedge Z(i, a)). \tag{3.1}$$

We can use row to represent a tuple $Z_1, ..., Z_k$ of strings by a single string Z. We use a similar idea to allow Z coding a sequence $y_0, y_1, ...$ of numbers. Now y_i is the smallest element of Z_i, or $|Z|$ if Z_i is empty. The number function $seq(i, Z)$ (also denoted by z_i) has the following defining axiom:

$$\begin{aligned} a = seq(i, Z) \iff (a < |Z| \wedge Z(i, a) \wedge \forall b < a, \neg Z(i, b)) \vee \\ \vee (\forall b < |Z|, \neg Z(i, b) \wedge a = |Z|). \end{aligned} \tag{3.2}$$

To get the maximum or minimum elements of the set R, we define functions max and min naturally:

$$\begin{aligned} max(R) = |R| - 1, \\ min(R) = x \iff \forall y < |R|,\ R(y) \to x \leq y. \end{aligned} \tag{3.3}$$

We define the ordering relation for strings as follows:

$$\begin{aligned} X \leq Y \iff X = Y \vee (|X| \leq |Y| \wedge \exists z \leq |Y| (Y(z) \wedge \neg X(z) \wedge \\ \wedge \forall u \leq |Y|, z < u \to (X(u) \to Y(u)))). \end{aligned} \tag{3.4}$$

That is, we compare strings based on numbers they represent as binary coding (the greater the number, the greater the string). Finally, we give $\Sigma_0^{1,b}$ bit-definitions of the string functions $\emptyset$ (constant empty string) and $S(X)$ (successor):

$$\emptyset(z) \iff z < 0, \tag{3.5}$$

and

$$S(X)(i) \iff (i \leq |X| \wedge ((X(i) \wedge \exists j < i, \neg X(j)) \vee (\neg X(i) \wedge \forall j < i, X(j)))). \tag{3.6}$$

3.1.2 A Third-Order Language

To refer to exponentially large objects such as power sets, congruences on products of algebras of size comparable to $|A| = l$, and so forth, we use the setting introduced in [10]. In addition to free and bound variables of first and second sorts, we consider variables of a third sort that represent finite sets of finite sets, named $\mathscr{A}, \mathscr{B}, \mathscr{C}, ...$ and $\mathscr{X}, \mathscr{Y}, \mathscr{Z}, ...$ We refer to second-sort objects as 'strings', and to third-sort objects as 'classes' (note that in the original setting in [10] classes were referred to as 'superstring', but this name does not reflect the type of objects we discuss). The language ${\mathcal{L}^3}_{\mathcal{PA}}$ contains an additional symbol for the third-order membership predicate $A \in_3 \mathscr{B}$,

$${\mathcal{L}^3}_{\mathcal{PA}} = \{0, 1, +, \cdot, |\,|, =_1, =_2, \leq, \in_2, \in_3\}.$$

Classes can be also thought of as strings of bits, where each bit is indexed by a set referred to as *bit-index*. There is no length-function analog for classes, so the 'length' of a class in this setting is the lexicographically maximal bit-index under consideration. Number terms are defined as in V^1, not including any reference to third-order variables, while formulas additionally may have third-order variables and quantifiers. We extend the hierarchy $\Sigma_i^{1,b}$ of second-order formulas to third-order classes $\Sigma_i^{\mathscr{B}}$ that consist of those formulas with arbitrarily many bounded first-order and second-order quantifiers, and exactly i alternating unbounded third-order quantifiers, the outermost being restricted, i.e. equivalent to the existential quantifier. We also define a specific class $\forall^2\Sigma_i^{\mathscr{B}}$ of formulas consisting of a single bounded universal second-order quantifier followed by a $\Sigma_i^{\mathscr{B}}$-formula.

Although third-order variables are unbounded because of the absence of a length function, they will be implicitly bounded, in the sense that the bounds on first-order and second-order quantifiers will limit the part of the class that affects the truth value of a given formula.

Theory W_1^0 and W_1^1 presented in [10] have $\forall^2\Sigma_0^{\mathscr{B}}$-IND or $\forall^2\Sigma_1^{\mathscr{B}}$-IND induction axiom scheme respectively (the unusual class of formulas is a technicality, W_1^1 admits full $\Sigma_1^{\mathscr{B}}$-induction), and the following two comprehension axiom schemes, namely $\Sigma_0^{\mathscr{B}}$-2COMP:

$$\exists Y \le t(\bar{x},\bar{X})(\forall z \le s(\bar{x},\bar{X}))\big(\phi(\bar{x},\bar{X},\bar{\mathscr{X}},z) \iff Y(z)\big),$$

and $\Sigma_0^{\mathscr{B}}$-3COMP:

$$\exists \mathscr{Y}(\forall Z \le t(\bar{x},\bar{X}))\big(\phi(\bar{x},\bar{X},\bar{\mathscr{X}},Z) \iff \mathscr{Y}(Z)\big),$$

where in each case $\phi \in \Sigma_0^{\mathscr{B}}$ subjects to the restriction that neither Y nor $\mathscr{Y}$ occur free in ϕ. Recall that V is a two-sorted version of the theory S_2 from [3].

Lemma 21 ([11])**.** *W_1^0 is a conservative extension of V, the two-sorted theory for the poly-time hierarchy.*

Thus, $\Sigma_0^{\mathscr{B}}$-definable functions of number and string arguments are usual p-time hierarchy functions. $\Sigma_1^{\mathscr{B}}$-definable functions of W_1^1 are exactly FPSPACE$^+$, a third-order analogue of PSPACE functions (see [11]). W_1^1 can $\Sigma_0^{\mathscr{B}}$-define all number and string-valued functions of number and string arguments from the polynomial-time hierarchy. We can add pairing functions for second-order objects, such as $\langle X,Y\rangle$ and $\langle x,Y\rangle$, using the following natural definitions.

$$\langle X,Y\rangle = Z(i,a) \iff (i=0 \wedge X(a)) \vee (i=1 \wedge Y(a)), \tag{3.7}$$

and

$$\langle x,Y\rangle = Z(i,a) \iff i = x \wedge Y(a). \tag{3.8}$$

For a third-order variable $\mathscr{X}$ define $\mathscr{X}^{[x]}(Y) \equiv \mathscr{X}(\langle x,Y\rangle)$ and $\mathscr{X}^{[X]}(Y) \equiv \mathscr{X}(\langle X,Y\rangle)$. This notation allows us to consider $\mathscr{X}$ as an array with rows indexed by numbers or strings, where each row is a third-order object. Note that as opposed to a string-valued function $row(i,Z)$, this notation is just an abbreviation of the formula, not a class-valued function. However, if we can bound the size of all strings in a class we are interested in by some value s, then we can define a string-valued function $r\tilde{o}w(\cdot)$ analogous to $row(\cdot)$,

$$\begin{aligned} r\tilde{o}w(i,\mathscr{X},s) = Y \iff & (|Y| < s \wedge \mathscr{X}^{[i]}(Y) \wedge \forall Y' < Y\, \neg\mathscr{X}^{[i]}(Y')) \vee \\ & \vee (\forall Y' < s\, \neg\mathscr{X}^{[i]}(Y') \wedge \forall a < s,\, \neg Y(a) \wedge |Y| = s \wedge Y(s-1)), \end{aligned} \tag{3.9}$$

where $Y' < Y$ is string ordering relation (3.4). Thus, the function returns the minimum string (due to string ordering) Y of length less than s such that $\mathscr{X}^{[i]}(Y)$ or, if such a string does not exist, the string Y of length s with the only element $s-1 \in Y$. An analogous function $r\tilde{o}w(X,\mathscr{X},s)$ can be defined for string indexing.

3.1.3 Quantified propositional calculus G and its correspondence to W^1_1

In this section we recall the definition of the sequent calculus G. We adopt definitions from [9].

The class of quantified propositional formulas, denoted by Σ^q_∞, is the smallest class of formulas containing atoms $0, 1$, and closed under logical connectives and Boolean quantification: if $\phi(x)$ is a formula in Σ^q_∞, then so are $\exists x\phi(x)$ and $\forall x\phi(x)$, and the meaning is $\phi(0) \vee \phi(1)$ and $\phi(0) \wedge \phi(1)$, respectively. The proof system G for quantified propositional formulas extends a classical proof system, the sequent calculus LK. For the sake of completeness, we recall the definition of LK here.

Definition 45 (Sequent Calculus LK)**.** A line in an LK-proof is a *sequent*: it is an ordered pair of finite, possibly empty sequences of formulas Γ, Δ written as

$$\Gamma \longrightarrow \Delta,$$

that is satisfiable if and only if there is an assignment that either makes some formula in Γ false or makes some formula in Δ true. Thus, if $\Gamma = \phi_1, ..., \phi_m$ and $\Delta = \psi_1, ..., \psi_n$, then the sequent is equivalent to the formula

$$\neg\psi_1 \vee ... \vee \neg\phi_m \vee \psi_1 \vee ... \vee \psi_n.$$

The inference rules of the sequent calculus LK are the following:

1. *Initial sequents* are the following:

$$p \longrightarrow p, \quad 0 \longrightarrow, \quad \longrightarrow 1,$$

   where p is a variable.

2. *Structural rules* are:

   - the weakening rules

$$\text{left: } \frac{\Gamma \longrightarrow \Delta}{\phi, \Gamma \longrightarrow \Delta} \qquad \text{right: } \frac{\Gamma \longrightarrow \Delta}{\Gamma \longrightarrow \Delta, \phi}$$

   - the exchanging rules

$$\text{left: } \frac{\Gamma_1, \phi, \psi, \Gamma_2 \longrightarrow \Delta}{\Gamma_1, \psi, \phi, \Gamma_2 \longrightarrow \Delta} \qquad \text{right: } \frac{\Gamma \longrightarrow \Delta_1, \phi, \psi, \Delta_2}{\Gamma \longrightarrow \Delta_1, \psi, \phi, \Delta_2}$$

   - the contraction rules

$$\text{left: } \frac{\Gamma_1, \phi, \phi, \Gamma_2 \longrightarrow \Delta}{\Gamma_1, \phi, \Gamma_2 \longrightarrow \Delta} \qquad \text{right: } \frac{\Gamma \longrightarrow \Delta_1, \phi, \phi, \Delta_2}{\Gamma \longrightarrow \Delta_1, \phi, \Delta_2}$$

3. *Logical rules* are:

   - $\neg$-introduction rules

$$\text{left: } \frac{\Gamma \longrightarrow \Delta, \phi}{\neg\phi, \Gamma \longrightarrow \Delta} \qquad \text{right: } \frac{\phi, \Gamma \longrightarrow \Delta}{\Gamma \longrightarrow \Delta, \neg\phi}$$

   - $\wedge$-introduction rules

$$\text{left: } \frac{\phi, \Gamma \longrightarrow \Delta}{\phi \wedge \psi, \Gamma \longrightarrow \Delta} \qquad \text{right: } \frac{\Gamma \longrightarrow \phi, \Delta \quad \Gamma \longrightarrow \Delta, \psi}{\Gamma \longrightarrow \Delta, \phi \wedge \psi}$$

- $\vee$-introduction rules

$$\text{left: } \frac{\phi, \Gamma \longrightarrow \Delta \quad \psi, \Gamma \longrightarrow \Delta}{\phi \vee \psi, \Gamma \longrightarrow \Delta} \qquad \text{right: } \frac{\Gamma \longrightarrow \phi, \Delta}{\Gamma \longrightarrow \Delta, \phi \vee \psi}$$

4. *The cut-rule* is:

$$\frac{\Gamma \longrightarrow \Delta, \phi \quad \phi, \Gamma \longrightarrow \Delta}{\Gamma \longrightarrow \Delta}$$

An *LK-proof* of a sequent S is a sequence of sequents that starts with initial sequents and ends with S, where each sequent in the proof is produced from previous ones by using the inference rules.

Definition 46 (Sequent calculus G)**.** Quantified propositional calculus G extends system LK by allowing Σ^q_∞-formulas in sequents and by accepting two quantifier rules:

1. $\forall$:introduction

$$\text{left: } \frac{\phi(\psi), \Gamma \longrightarrow \Delta}{\forall x \phi(x), \Gamma \longrightarrow \Delta} \qquad \text{right: } \frac{\Gamma \longrightarrow \Delta, \phi(p)}{\Gamma \longrightarrow \Delta, \forall x \phi(x)}$$

2. $\exists$:introduction

$$\text{left: } \frac{\phi(p), \Gamma \longrightarrow \Delta}{\exists x \phi(x), \Gamma \longrightarrow \Delta} \qquad \text{right: } \frac{\Gamma \longrightarrow \Delta, \phi(\psi)}{\Gamma \longrightarrow \Delta, \exists x \phi(x)}$$

where ψ is any formula such that no variable occurrence free in ψ becomes quantified in $\phi(\psi)$, and variable p does not occur in lower sequences of inference rules.

In [4] there is presented the well-known translation of any $\phi(\bar{x}, \bar{X}) \in \Sigma^{1,b}_0$ into a family of propositional formulas,

$$||\phi(\bar{x}, \bar{X})|| = \{\phi(\bar{x}, \bar{X})[\bar{m}, \bar{n}] : \bar{m}, \bar{n} \in \mathbb{N}\} \tag{3.10}$$

such that the following lemma holds:

Lemma 22 ([4])**.** *For every $\Sigma^{1,b}_0(\mathcal{L}^2{}_{\mathcal{PA}})$ formula $\phi(\bar{x}, \bar{X})$, there is a constant $d \in \mathbb{N}$ and a polynomial $p(\bar{m}, \bar{n})$ such that for all $\bar{m}, \bar{n} \in \mathbb{N}$, the propositional formula $\phi(\bar{x}, \bar{X})[\bar{m}, \bar{n}]$ has depth at most d and size at most $p(\bar{m}, \bar{n})$.*

Propositional translation of formulas $\Sigma^{1,b}_0(\mathcal{L}^2{}_{\mathcal{PA}})$ can be extended to the translation of any bounded $\mathcal{L}^2{}_{\mathcal{PA}}$-formula into a quantified propositional formula, using strings of Boolean quantifiers to represent second-order quantifiers, [4]. Let us denote the class of all bounded $\mathcal{L}^2{}_{\mathcal{PA}}$ formulas by

$$\Sigma^{1,b}_\infty = \bigcup_i \Sigma^{1,b}_i = \bigcup_i \Pi^{1,b}_i.$$

The following theorem establishes the correspondence between theory W^1_1 and quantified propositional calculus G. It follows from [10], specifically from Theorems 12, 13.

Theorem 27. *Suppose that $\phi(\bar{x}, \bar{X})$ is a $\Sigma^{1,b}_\infty$ formula such that $W^1_1 \vdash \phi(\bar{x}, \bar{X})$. Then the propositional family $||\phi(\bar{x}, \bar{X})||$ has quantified propositional calculus proofs of polynomial size. That is, there is a polynomial $p(\bar{m}, \bar{n})$ such that for all $1 \leq \bar{m}, \bar{n} \in \mathbb{N}$, $\phi(\bar{x}, \bar{X})[\bar{m}, \bar{n}]$ has a G-proof of size at most $p(\bar{m}, \bar{n})$. Furthermore, there is an algorithm that finds a G-proof of $\phi(\bar{x}, \bar{X})[\bar{m}, \bar{n}]$ in time bounded by a polynomial in $(\bar{m}, \bar{n})$.*

3.2 Formalization of notions

In this section we shall use the formalization of notions introduced in Chapter 2 (ref. [6]) and we shall formalize all the remaining notions of universal algebra that will be used to prove the three axiom schemes from Chapter 2 (ref. [6]), as well as some specific notions introduced by Zhuk in [15]. Notions not formalized in this section are defined earlier in Chapter 2 (ref. [6]). We can repeat some definitions from Chapter 2 (ref. [6]) for consistency of the text if we introduce some new notation or objects based on previous ones. We know of no way to prove that a formalization exists other than actually doing it. This leads to a quite formal (and occasionally tedious) text with long formulas. Writing the formulas explicitly allows us to see that their bounded quantifier complexity is what is claimed.

Notation 10. *To simplify the notation for the reader, we will denote relations on numbers using capital letters and functions using lowercase letters. We sometimes omit arguments in relations that are obviously implied and do not affect the content of the relation. We index elements of sets starting with* 0*, while all indices not related to elements of sets (for example, a sequence of relations) start from* 1*.*

3.2.1 $\mathbb{A}$—Monster Set: objects we have in advance

In this section we describe the list of objects we will further refer to as objects given in advance. The algorithm works for any finite algebra having a weak near unanimity (WNU) term and uses the fact that this term and all the properties of the algebra are known. From now on, we fix the algebra $\mathbb{A} = (A, \Omega)$, fix l to be its size, and suppose that the only basic operation of $\mathbb{A}$ is an idempotent special WNU m-ary operation Ω. To be consistent with Zhuk's paper, we do not use bold font for subuniverses of $\mathbb{A}$. For the corresponding relational structure, we fix the notation $\mathcal{A} = (A, \Gamma_{\mathcal{A}})$, where $\Gamma_{\mathcal{A}}$ is explained below.

Let $Sound(\mathbb{A})$ denote the soundness of Zhuk's algorithm for algebra $\mathbb{A}$. Recall from Chapter 2 (ref. [6]) that this formalizes that if the algorithm rejects an instance, then the instance has no solution. In theory T we can consider proving not just $Sound(\mathbb{A})$ but more generally an implication of the form

$$Cond(\mathbb{A}) \implies Sound(\mathbb{A}),$$

where $Cond(\mathbb{A})$ is any recursively enumerable property of algebra $\mathbb{A}$. It can be written as

$$\exists Y Cond_0(\mathbb{A}, Y),$$

where Y, in general, cannot be bounded (even recursively), and $Cond_0$ can be a second-order bounded formula. In our case, Y is a list of various objects such as subuniverses, binary relations preserved by Ω on A or any subuniverse D of $\mathbb{A}$, ternary operations on A, isomorphisms from subalgebras (D, Ω) of $\mathbb{A}$ to products of finite fields, etc. together with V^0-proofs of their various $\Sigma_0^{1,b}$-properties. The proofs are given simply by exhaustive searching, unwinding all quantifiers. Therefore, the size of the monster list Y may be huge, in particular exponential, in the size of $\mathbb{A}$, but in general it does not matter: whatever function of l it is, it is a constant for fixed l.

Note that if W witnesses $Cond(\mathbb{A})$, we can prove $Cond_0(\mathbb{A}, W)$ in V^0 (a constant size proof) and apply modus ponens to the implication above to deduce $Sound(\mathbb{A})$, which is what we really want to prove in T. This argument applies whenever T contains V^0, which is true in our case.

The use of this can be illustrated as follows. Assume $P(D)$ and $Q(D)$ are two bounded properties of a subuniverse D of $\mathbb{A}$ and assume that in the monster set Y we have two lists of all subuniverses together with proofs that they do or do not satisfy P and Q respectively. A universal statement

$$\forall D, subAlgebra(D, A), P(D) \to Q(D)$$

can then be simply proved by going through Y and checking that every D in the list of those satisfying P is also in the list of those satisfying Q; this uses a composition of proofs listed in Y. Another example of use is the following. The properties of Z that may involve second-order universal bounded quantifiers, as, for example, in

$$\forall D, subAlgebra(D, A),\ \phi(Z, D)$$

with $\phi \in \Sigma_0^{1,b}$, can be rewritten as $\Sigma_0^{1,b}$-formulas: replace the universal quantifier by a large (but constant size) conjunction over all subalgebras of $\mathbb{A}$ as listed in the monster set Y.

This allows us to use well-known facts from universal algebra, as well as facts proved by Zhuk in [15], without proving them in a theory of bounded arithmetic when it comes to the formalization of objects related exclusively to algebra $\mathbb{A}$. For example, we will not prove that any PC congruence σ on $\mathbb{A}$ is maximal or that polynomially complete algebra $\mathbb{A}/\sigma$ is simple. Although we believe that all the properties of different objects on $\mathbb{A}$ needed in the argument can be proved in $\Sigma_1^{1,b}$-reasoning even with $\mathbb{A}$ variable, it is not necessary for our purpose. On the contrary, we shall prove any property related to an input structure since this structure is variable.

We further list all the given in advance objects related to $\mathbb{A}$ we will use in the formalization, so-called $\mathbb{A}$-Monster set. All of them will be defined in detail in the corresponding sections.

- All subuniverses of $\mathbb{A}$ and any of its subuniverse D, the lists $\Gamma^1_{\mathcal{A}}$, $\Gamma^1_{\mathcal{D}}$;
- All binary relations on A and any of its subuniverse D, compatible with Ω, the lists $\Gamma^2_{\mathcal{A}}$, $\Gamma^2_{\mathcal{D}}$;
- All congruences σ on A and any of its subuniverse D, the lists $\Sigma_{\mathcal{A}}$, $\Sigma_{\mathcal{D}}$;
- All factor sets for congruences σ on A and any of its subuniverse D, A/σ and D/σ, and all operations Ω/σ, the lists $\mathrm{A}_{\mathcal{A}}(i, A/\Sigma_{\mathcal{A},i}, \Omega/\Sigma_{\mathcal{A},i})$, $\mathrm{A}_{\mathcal{D}}(i, D/\Sigma_{\mathcal{D},i}, \Omega/\Sigma_{\mathcal{D},i})$;
- All maximal congruences on A and any of its subuniverse D, the lists $\Sigma^{max}_{\mathcal{A}}$, $\Sigma^{max}_{\mathcal{D}}$;
- For all congruences σ on A and any of its subuniverse D, the lists of all unary and binary quotient relations on A and D, compatible with Ω/σ. We will denote the lists by $\Gamma_{\mathcal{A}}/\sigma$, $\Gamma_{\mathcal{D}}/\sigma$.
- The sets of all binary and ternary polymorphisms on A and any of its subuniverse D, the lists $\Pi^2_{\mathcal{A}}, \Pi^3_{\mathcal{D}}$;
- For all congruences σ on A and any of its subuniverse D, the sets of all binary and ternary polymorphisms on A/σ and D/σ, the lists $\Pi^2_{\mathcal{A}/\sigma}, \Pi^3_{\mathcal{D}/\sigma}$;
- For all congruences σ on A and any of its subuniverse D, the sets of all maps H from A/σ to Z_{p_0}, all maps H from A/σ to $Z_{p_0} \times Z_{p_1}$,..., all maps H from A/σ to $Z_{p_0} \times Z_{p_1} \times ... \times Z_{p_{s-1}}$, for $s = log_2 l$ and any prime $p_0, ..., p_{s-1}$, $p_0 \cdot ... \cdot p_{s-1} \leq l$. We will denote these lists by $\mathrm{M}_{A,\sigma,p_0,...,p_{t-1}}$, $\mathrm{M}_{D,\sigma,p_0,...,p_{t-1}}$.

- The set of all linear congruences on A and any of its subuniverse D, the lists $\Sigma^{lin}_{\mathcal{A}}$, $\Sigma^{lin}_{\mathcal{D}}$;

- For any subuniverse C of A and any of its subuniverse D, all sets of the form $X = \{\{a\} \times C, C \times \{a\}\}$ for all $a \in A \backslash C$. We denote the lists by $\mathrm{X}_{\mathcal{A}}$, $\mathrm{X}_{\mathcal{D}}$;

- The set of all PC congruences on A and any of its subuniverse D, the lists $\Sigma^{PC}_{\mathcal{A}}$, $\Sigma^{PC}_{\mathcal{D}}$.

- For all congruences θ on A and any of its subuniverse D, and for all PC congruences $\sigma_0, ..., \sigma_{s-1}$ on A and any of its subuniverse D, the sets of all maps H from A/θ to A/σ_{j_0}, all maps H from A/θ to $A/\sigma_{j_0} \times A/\sigma_{j_1}$,..., all maps H from A/θ to $A/\sigma_{j_0} \times A/\sigma_{j_1} \times ... \times A/\sigma_{j_{s-1}}$, for $s = log_2 l$. We denote these lists by $\mathrm{M}_{A,\theta,\sigma_{j_0},\sigma_{j_1},...,\sigma_{j_{t-1}}}$, $\mathrm{M}_{D,\theta,\sigma_{j_0},\sigma_{j_1},...,\sigma_{j_{t-1}}}$.

- For all subuniverses D_i, D_j of A, all congruences σ_i, σ_j on D_i, D_j, the set of all bridges from σ_i to σ_j, the set of all reflexive bridges and the set of all optimal bridges, the lists Ξ_{σ_i,σ_j}, $\Xi^{\leftrightarrow}_{\sigma_i,\sigma_j}$ and $\Xi^{opt}_{\sigma_i,\sigma_j}$.

Due to the definitions of all these sets (given in the corresponding sections), they may be empty.

Notation 11. *In formulas, we use notation* $\bigwedge_{\Sigma^{max}_{\mathcal{A},i}}$ *or* $\bigvee_{\Sigma_{\mathcal{D},i}}$ *meaning* $\bigwedge_{\Sigma^{max}_{\mathcal{A},i} \neq \emptyset}$ *or* $\bigvee_{\Sigma_{\mathcal{D},i} \neq \emptyset}$*: here we consider conjunction over all maximal congruences on* A *or disjunction over all congruences on its subuniverse* D*. Sometimes, we also write* $\bigwedge_{\sigma \in \Sigma^{max}_{\mathcal{A}}}$ *or* $\bigvee_{B \in \Gamma^1_{\mathcal{A}}} \bigvee_{T \in \Pi^2_{\mathcal{A}}}$ *with the same meaning. When needed for better clarity, we use the explicit notation* $\exists j < 2^{l^2}, ...\Sigma^{PC}_{\mathcal{A},j}....$

3.2.2 Encoding directed graphs and CSP instances

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

Definition 47. 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 a pair of sets $\ddot{\mathcal{A}} = (V_{\ddot{\mathcal{A}}}, E_{\ddot{\mathcal{A}}})$, where:

- $V_{\ddot{\mathcal{A}}} < \langle n, l \rangle$ is the set corresponding to the superdomain; we denote set $V_{\ddot{\mathcal{A}},i}$ by D_i and call it domain subset for variable x_i;

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

$$E_{\ddot{\mathcal{A}}}(u, v) \rightarrow \exists i, j < n \, \exists a, b < l \; u = \langle i, a \rangle \wedge v = \langle j, b \rangle \wedge \\ D_i(a) \wedge D_j(b). \tag{3.11}$$

Sometimes we consider set $D = \{D_0, ..., D_{n-1}\}$. We use the notation $E^{ij}_{\ddot{\mathcal{A}}}(a, b)$ instead of $E_{\ddot{\mathcal{A}}}(\langle i, a \rangle, \langle j, b \rangle)$ for simplicity. We will denote a pair of sets $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$, satisfying all the above conditions, by $\mathrm{DG}(\Theta)$, and we will call Θ an instance. This representation allows us to construct a homomorphism from $\mathcal{X}$ to $\ddot{\mathcal{A}}$ with respect to different relations $E^{ij}_{\ddot{\mathcal{A}}}$ and different domains for all vertices $x_1, ..., x_n$.

Definition 48. A pair of sets $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$ is a CSP instance on n domains over constraint language $\Gamma_{\mathcal{A}}$ if

$$\begin{gathered} Inst(\Theta, \Gamma_{\mathcal{A}}) \iff DG(\Theta) \wedge \forall i < n, |D_i| = l \wedge \\ \wedge \forall i, j < n, a, b < l, \exists s < |\Gamma_{\mathcal{A}}|, E_{\ddot{\mathcal{A}}}(\langle i, a\rangle, \langle j, b\rangle) \leftrightarrow \Gamma^2_{\mathcal{A}}(s, a, b) \wedge \\ \wedge \forall i < n, a < l, \exists s < |\Gamma_{\mathcal{A}}|, D_i(a) \leftrightarrow \Gamma^1_{\mathcal{A}}(s, a). \end{gathered} \tag{3.12}$$

When considering the direct product $D_0 \times ... \times D_{n-1}$, we can refer to it as a set of solutions to a CSP instance $\Theta_{null} = (\mathcal{X}_{null}, \ddot{\mathcal{A}}_{null})$, where

- $V_{\mathcal{X}_{null}} = n$ and for all $i < n, V_{\mathcal{X}_{null}}(i)$;
- for all $i, j < n$, $\neg E_{\mathcal{X}_{null}}(i, j)$ (i.e. the instance digraph $\mathcal{X}_{null}$ has no edges at all);
- for all $a < l$, $V_{\ddot{\mathcal{A}}_{null}}(i, a) \iff D_i(a)$;
- for all $a, b < l$, for all $i, j < n$, $\neg E^{ij}_{\ddot{\mathcal{A}}_{null}}(a, b)$ (i.e. the target digraph $\ddot{\mathcal{A}}_{null}$ has no edges at all).

We will denote a pair of sets $\Theta_{null} = (\mathcal{X}_{null}, \ddot{\mathcal{A}}_{null})$ satisfying all the above conditions by $DG_{null}(\Theta_{null})$. Since as domains we consider only subuniverses D_i of $\mathbb{A} = (A, \Omega)$, Θ_{null} is also a CSP instance over constraint language $\Gamma_{\mathcal{A}}$. Sometimes, we will work with the so-called factorized instances, where we factorize all domains D_i by congruences σ_i.

Definition 49. A pair of sets $\Theta' = (\mathcal{X}', \ddot{\mathcal{A}}')$ is a factorized CSP instance by list of n congruences Σ on n domains from a CSP instance Θ over constraint language $\Gamma_{\mathcal{A}}$ if

$$\begin{gathered} FInst(\Theta', \Sigma, \Theta, \Gamma_{\mathcal{A}}) \iff Inst(\Theta, \Gamma_{\mathcal{A}}) \wedge \mathcal{X} = \mathcal{X}' \wedge \forall i < n,\ FS_m(D'_i, D_i, \Omega, \Sigma_i) \wedge \\ \wedge \forall i, j < n,\ E^{ij}_{\ddot{\mathcal{A}}'}(a, b) \leftrightarrow D'_i(a) \wedge D'_j(b) \wedge (\exists c, d < l,\ \Sigma_i(a, c) \wedge \Sigma_j(b, d) \wedge E^{ij}_{\ddot{\mathcal{A}}}(c, d)), \end{gathered} \tag{3.13}$$

where definition of the relation FS_m can be found in Chapter 2 (ref. [6]).

Definition 50 (Homomorphism from digraph $\mathcal{X}$ to digraph with domains $\ddot{\mathcal{A}}$)**.** A map H is a *homomorphism between the input digraph $\mathcal{X} = (V_{\mathcal{X}}, E_{\mathcal{X}})$, $V_{\mathcal{X}} = n$ and the target digraph with domains $\ddot{\mathcal{A}} = (V_{\ddot{\mathcal{A}}}, E_{\ddot{\mathcal{A}}})$, $V_{\mathcal{A}} < \langle n, l\rangle$* 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 H can be expressed by the following $\Sigma^{1,b}_1$-formula.

$$\begin{gathered} H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}}) \iff \exists H < \langle n, \langle n, l\rangle\rangle (MAP(V_{\mathcal{X}}, n, V_{\ddot{\mathcal{A}}}, \langle n, l\rangle, H) \wedge \\ (\forall i < n, s < \langle n, l\rangle\ H(i) = s \rightarrow \exists a < l, s = \langle i, a\rangle \wedge D_i(a)) \wedge \\ \forall i_1, i_2 < n, \forall j_1, j_2 < \langle n, l\rangle \\ (E_{\mathcal{X}}(i_1, i_2) \wedge H(i_1) = j_1 \wedge H(i_2) = j_2 \rightarrow E_{\ddot{\mathcal{A}}}(j_1, j_2)). \end{gathered} \tag{3.14}$$

In addition to 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 $V_{\mathcal{G}} < n$, $V_{\mathcal{G}} < m$ is again a $\Sigma^{1,b}_1$-formula.

$$\begin{gathered} HOM(\mathcal{G}, \mathcal{H}) \iff \exists H < \langle n, m\rangle (MAP(V_{\mathcal{G}}, n, V_{\mathcal{H}}, m, H) \wedge \\ \forall i_1, i_2 < n, \forall j_1, j_2 < m \\ (E_{\mathcal{G}}(i_1, i_2) \wedge H(i_1) = j_1 \wedge H(i_2) = j_2 \rightarrow E_{\mathcal{H}}(j_1, j_2))). \end{gathered} \tag{3.15}$$

Notation 12. *Sometimes, we will write $\exists(\forall) H < \langle n, m\rangle, HOM(\mathcal{G}, \mathcal{H}, H)$ and $\exists(\forall) H < \langle n, \langle n, l\rangle\rangle$, $H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}}, H)$ to omit repetitions. Note that $HOM(\mathcal{G}, \mathcal{H})$ and $H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}})$ are $\Sigma^{1,b}_1$-formulas, while $HOM(\mathcal{G}, \mathcal{H}, H)$ and $H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}}, H)$ are $\Sigma^{1,b}_0$.*

3.2.3 Subalgebras and Solution sets to a CSP instance

Recall that we encode the fixed algebra $\mathbb{A}$ with a pair of sets (A, Ω), where $|A| = l$, $A(i)$ for every i, and Ω is a set of size $((m+1)l)^{2^{m+1}}$ representing a special m-ary WNU operation on A, while all subuniverses of $\mathbb{A}$ are encoded by subsets of A closed under Ω. To define the direct and subdirect products of k algebras for constant k, we first define a universe set for the product. For any sets $D_0, ..., D_{k-1}$ of size bounded by l we will denote by $D_0 \times ... \times D_{k-1}$ a k-ary set of the form

$$D_0 \times ... \times D_{k-1}(a_0, ..., a_{k-1}) \iff a_0 \in D_0 \wedge ... \wedge a_{k-1} \in D_{k-1}. \tag{3.16}$$

As any m-ary operation $F : D^m \to D$ on a set D in Chapter 2 (ref. [6]), we define an m-ary operation $F : (D_0 \times ... \times D_{k-1})^m \to D_0 \times ... \times D_{k-1}$ on a set $D_0 \times ... \times D_{k-1}$. Denote $\langle a_i^0, ..., a_i^{k-1} \rangle$ by $\bar{a}_i^k$, then

$$\begin{gathered} OP_m(F, D_0 \times ... \times D_{k-1}) \iff \forall \bar{a}_1^k, ..., \bar{a}_m^k \in D_0 \times ... \times D_{k-1}, \\ \exists \bar{b}^k \in D_0 \times ... \times D_{k-1},\ F(\bar{a}_1^k, ..., \bar{a}_m^k, \bar{b}^k) \wedge \forall \bar{b}_1^k, \bar{b}_2^k \in A_0 \times ... \times D_{k-1}, \\ (F(\bar{a}_1^k, ..., \bar{a}_m^k, \bar{b}_1^k) \wedge F(\bar{a}_1^k, ..., \bar{a}_m^k, \bar{b}_2^k) \to \bar{b}_1^k = \bar{b}_2^k). \end{gathered} \tag{3.17}$$

In the same fashion, we can formalize a special idempotent WNU operation Ω on the set $D_0 \times ... \times D_{k-1}$, and can define a subuniverse R of algebra $(D_0 \times ... \times D_{k-1}, \Omega)$:

$$\begin{gathered} subTA(R, D_0 \times ... \times D_{k-1}, \Omega) \iff |R| = |D_0 \times ... \times D_{k-1}| \wedge \\ \forall i < (kl)^{2^k}, R(i) \to D_0 \times ... \times D_{k-1}(i) \wedge SwNU_m(\Omega, R). \end{gathered} \tag{3.18}$$

We say that an algebra $\mathbb{D} = (D, \Omega)$ is a direct product of k algebras (D_0, Ω_0), ..., (D_{k-1}, Ω_{k-1}) of the same type (with m-ary operations) if

$$\begin{gathered} DP_{m,k}(D, \Omega, D_0, \Omega_0, ..., D_{k-1}, \Omega_{k-1}) \iff D = D_0 \times ... \times D_{k-1} \wedge \\ \wedge \forall a_1^0, ..., a_m^0 \in D_0, ..., \forall a_1^{k-1}, ..., a_m^{k-1} \in D_{k-1} \exists b^0 \in D_0, ..., \exists b^{k-1} \in D_{k-1} \\ \Omega(\bar{a}_1^k, ..., \bar{a}_m^k, \langle b^0, ..., b^{k-1} \rangle) \wedge \Omega_0(a_1^0, ..., a_m^0, b^0) \wedge ... \wedge \Omega_{k-1}(a_1^{k-1}, ..., a_m^{k-1}, b^{k-1}). \end{gathered} \tag{3.19}$$

A subdirect product (R, Ω) of k algebras (D_0, Ω_0), ...,(D_{k-1}, Ω_{k-1}) is encoded as follows:

$$\begin{gathered} subDP_{m,k}(R, \Omega, D_0, \Omega_0, ..., D_{k-1}, \Omega_{k-1}) \iff subTA(R, D_0 \times ... \times D_{k-1}, \Omega) \\ \wedge DP_{m,k}(D_0 \times ... \times D_{k-1}, \Omega, D_0, \Omega_0, ..., D_{k-1}, \Omega_{k-1}) \wedge \\ \wedge \bigwedge_{i<k} \forall a_i \in D_i, \exists a_0 \in D_0, ..., \exists a_i \in D_{i-1}, \exists a_{i+1} \in D_{i+1}, ..., \exists a_{k-1} \in A_{k-1}, \\ R(a_1, ..., a_{i-1}, a_i, a_{i+1}, ..., a_k). \end{gathered} \tag{3.20}$$

To move to the solution set to a CSP instance, we recall the following theorem [1].

Theorem 28. *Let $\mathcal{B} = (B, \Gamma)$ be a finite relational structure, and let $R \subseteq B^n$ be a non-empty relation. Then R is preserved by all polymorphisms of Γ if and only if R is pp-definable from Γ.*

Note that the set of solutions to any instance of CSP(Γ) can be viewed as a subuniverse of power of B. Every n-ary relation R on B^n preserved by all polymorphisms of $Pol(\Gamma)$ is a solution set to some CSP instance Θ with n variables over the language $RelClone(\Gamma)$. Thus, a relation R on B^n can be *pp*-defined from a set of relations Γ if it is equal to some projection of the set of solutions to some instance of CSP(Γ). However, the instance itself

can be exponential in n (see the construction in [1]). Furthermore, we cannot define a subalgebra R of B^n as an n-ary set $R(b_1, ..., b_n)$ as it requires $(ln)^{2^n}$ length. We shall stress that since most of the theorems in the universal algebra part of Zhuk's algorithm that are proved for any subalgebras were used in the algorithm only for solution sets [15], whenever possible, we restrict ourselves upward to solution sets to some CSP instances over $\Gamma_{\mathcal{A}}$.

For definitions, we use the $\Sigma_0^{\mathscr{B}}$-3COMP axiom scheme. We can consider any n-ary relation R on A^n as a third-order object – a class of maps $\mathscr{R}$ from $[n]$ to $[A, A, ..., A]$. Analogously, any $R \leq D_0 \times ... \times D_{n-1}$ is a class of maps from $[n]$ to $[D_0, D_1, ..., D_{n-1}]$. In terms of digraphs,

$$\mathscr{R}(H) \implies MAP(V_{\mathcal{X}}, n, V_{\ddot{\mathcal{A}}}, \langle n, l\rangle, H), \tag{3.21}$$

which is $\Sigma_0^{\mathscr{B}}$-formula, and

$$\mathscr{D}_0 \times ... \times \mathscr{D}_{n-1}(H) \iff MAP(V_{\mathcal{X}}, n, V_{\ddot{\mathcal{A}}}, \langle n, l\rangle, H), \tag{3.22}$$

which is already $\Sigma_0^{1,b}$-formula. To make from $\mathscr{D}_0 \times ... \times \mathscr{D}_{n-1}$ an algebra, we define a third-order object representing a basic m-ary function $\mathscr{F}_{\Omega_0,...,\Omega_{n-1}}$ on $\mathscr{D}_0 \times ... \times \mathscr{D}_{n-1}$ (again, $\Sigma_0^{1,b}$-formula):

$$\begin{gathered}\mathscr{F}_{\Omega_0,...,\Omega_{n-1}}(H_1, ..., H_m, H) \iff \forall i < n, \exists a_1^i, ..., a_m^i, a^i < l,\ \Omega_i(a_1^i, ..., a_m^i) = a^i \wedge \\ \wedge H_1(i) = \langle i, a_1^i\rangle \wedge ... \wedge H_m(i) = \langle i, a_m^i\rangle \wedge H(i) = \langle i, a^i\rangle.\end{gathered} \tag{3.23}$$

If for all D_i there is the same operation Ω, we denote this class by $\mathscr{F}_{\Omega}$. Let us consider in this section all D_i being subalgebras of $\mathbb{A} = (A, \Omega)$. For subuniverses, we require that $\mathscr{R}(H)$ be closed under $\mathscr{F}_{\Omega}$ by the definition. In fact, we can express this requirement remaining in the second-order setting. For any n, we introduce a string function ω of m maps from $[n]$ to $[A, A, ..., A]$ returning a new map H by its bit-definition for all $i < n, a < l$:

$$\begin{gathered}\omega(H_1, ..., H_m)(\langle i, \langle i, a\rangle\rangle) \iff \exists a_1, ..., a_m < l,\ \Omega(a_1, ..., a_m) = a \wedge \\ \wedge H_1(i) = \langle i, a_1\rangle \wedge ... \wedge H_m(i) = \langle i, a_m\rangle.\end{gathered} \tag{3.24}$$

Note that ω is an actual function, not a set of sets, and it is based on the fixed set Ω. In the same fashion, we can introduce a string function $usepol_k$ for any k-ary polymorphism F on $\mathbb{A}$ by its bit-definition for all $i < n, a < l$:

$$\begin{gathered}usepol_k(F, H_1, ..., H_k)(\langle i, \langle i, a\rangle\rangle) \iff \exists a_1, ..., a_k < l,\ F(a_1, ..., a_k) = a \wedge \\ \wedge H_1(i) = \langle i, a_1\rangle \wedge ... \wedge H_k(i) = \langle i, a_k\rangle.\end{gathered} \tag{3.25}$$

We will denote such functions restricted by $\mathscr{R}$ by $\omega^{\mathscr{R}}$ and $usepol_k^{\mathscr{R}}$. Thus, for any class $\mathscr{R}$ representing subalgebra on A^n, and any maps $H_1, ..., H_m$:

$$\mathscr{R}(H_1) \wedge ... \wedge \mathscr{R}(H_m) \implies \mathscr{R}(\omega(H_1, ..., H_m)). \tag{3.26}$$

To consider a projection of subalgebra $\mathscr{R}$ to some subset of coordinates $i_1, ..., i_s, s < n$, we introduce a partial map from $[n]$ to $(D_0, ..., D_{n-1})$:

$$\begin{gathered}\mathscr{R}^{i_1,...,i_s}(H) \iff \bigwedge_{i \in \{i_1,...,i_s\}} \exists! a \in D_i,\ H(i) = \langle i, a\rangle \wedge \\ \bigwedge_{i \notin \{i_1,...,i_s\}} \forall a < l,\ \neg H < \langle i, \langle i, a\rangle\rangle \wedge \\ \exists H' < \langle n, \langle n, l\rangle\rangle, \mathscr{R}(H') \wedge \bigwedge_{i \in i_1,...,i_s} H(i) = H'(i).\end{gathered} \tag{3.27}$$

There are $2^n = \sum_{s=0}^{n} \binom{n}{s}$ such different classes, but we do not need to define them all; we will define the required occasionally. Note that ω and $usepol_k$ are well-defined for such partial maps for $i \in \{i_1, ..., i_s\}$.

The solution set to the instance $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$ of CSP over $\Gamma_{\mathcal{A}}$, $\ddot{\mathcal{A}} = (D_0, ..., D_{n-1}, E_{\ddot{\mathcal{A}}})$, is a set of homomorphisms $\{\mathcal{X} \to \ddot{\mathcal{A}}\} = \{H_1, H_2, ..., H_s\}$. Let us denote it as $\mathscr{R}_\Theta$. Note that the definition is a $\Sigma_0^{1,b}$-formula.

$$\mathscr{R}_\Theta(H) \iff H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}}, H). \tag{3.28}$$

In these terms, the product $\mathscr{D}_0 \times ... \times \mathscr{D}_{n-1}$ can be considered as $\mathscr{R}_{\Theta_{null}}$. The projection $\mathscr{R}_\Theta^{i_1,...,i_s}$ of the solution set to some subset of coordinates is defined analogously to (3.27), we call $H \in \mathscr{R}_\Theta^{i_1,...,i_s}$ a partial homomorphism from $\mathcal{X}$ to $\ddot{\mathcal{A}}$.

Lemma 23. *For any $k > 0$, V^1 proves that for any CSP instance Θ, any k-ary operation $F \in Pol_k(F, A, \Gamma_{\mathcal{A}})$, and any k homomorphisms $H_1, ..., H_k$ from $\mathcal{X}$ to $\ddot{\mathcal{A}}$ (and for any $i \in \{i_1, ..., i_s\}$) a map $H = usepol_{n,k}(F, H_1, ..., H_k)$ is again a homomorphism from $\mathcal{X}$ to $\ddot{\mathcal{A}}$ (a partial homomorphism from $\mathcal{X}$ to $\ddot{\mathcal{A}}$).*

Proof. Recall that any polymorphism preserves all relations from $\Gamma_{\mathcal{A}}$. Every relation $E_{\mathcal{A}}^{ij}$ (set of edges from D_i to D_j) is a subalgebra of $D_i \times D_j$ (since it is compatible with Ω). The proof then goes by contradiction: suppose that there is an edge $(x_i, x_j) \in E_{\mathcal{X}}$ (with $i, j \in \{i_1, ..., i_s\}$) such that H fails to map it to an edge in $\ddot{\mathcal{A}}$. Since all (partial for $i \in \{i_1, ..., i_s\}$) homomorphisms $H_1, ..., H_k$ map (x_i, x_j) to some edge in $E_{\ddot{\mathcal{A}}}^{ij}$, it is possible only if F does not preserve the relation $E_{\ddot{\mathcal{A}}}^{ij}$. □

Corollary 1. *V^1 proves that a solution set $\mathscr{R}_\Theta$ and a projection $\mathscr{R}_\Theta^{i_1,...,i_s}$ for any subset of coordinates $\{i_1, ..., i_s\}$ for a CSP instance $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$ on n variables are subuniverses of A^n and $A^{\{0,1,...,n\}\setminus\{i_1,...,i_s\}}$ respectively.*

We say that subuniverse $\mathscr{R}$ is subdirect if

$$subDSSInst(\mathscr{R}) \iff \forall i < n \forall a \in D_i, \exists H < \langle n, \langle n, l\rangle\rangle, H \in \mathscr{R} \wedge H(i) = \langle i, a\rangle. \tag{3.29}$$

Note that this is a $\Sigma_0^{\mathscr{B}}$-formula. If we consider solution set $\mathscr{R}_\Theta$, then the definition becomes a $\Sigma_1^{1,b}$-formula:

$$\begin{gathered} subDSSInst(\mathscr{R}_\Theta) \iff \forall i < n, \forall a \in D_i, \\ \exists H < \langle n, \langle n, l\rangle\rangle, H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}}, H) \wedge H(i) = \langle i, a\rangle. \end{gathered} \tag{3.30}$$

Whenever possible, we refer to $\mathscr{R}_\Theta$ as a set of homomorphisms $\{\mathcal{X} \to \ddot{\mathcal{A}}, \} = \{H_1, H_2, ..., H_s\}$, i.e. we use $\forall H \leq \langle n, \langle n, i\rangle\rangle$, $H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}}, H)$ since this allows us to avoid third-sorted objects.

Remark 5. Note that we cannot prove that any subalgebra of A^n is a solution set to some CSP instance over $\Gamma_{\mathcal{A}}$ (this is simply not true, otherwise we do not need existential quantification in *pp*-definitions).

3.2.4 Congruence and congruence on products

We define relations $Cong_m$, $pCong_m$ to be a congruence and a proper congruence as in Chapter 2 (ref. [6]). A maximal congruence on an algebra (D, Ω) can be defined by the following $\Sigma_0^{1,b}$-formula:

$$\begin{aligned} maxCong_m(D, \Omega, \sigma) \iff Cong_m(D, \Omega, \sigma) \wedge \exists a, b \in D, \neg\sigma(a, b) \wedge \\ \wedge[\bigwedge_{\Sigma_{\mathcal{D},i}} (\exists a, b \in D, \neg\Sigma_{\mathcal{D},i}(a, b) \to \exists a, b \in D, \sigma(a, b) \wedge \neg\Sigma_{\mathcal{D},i}(a, b))]. \end{aligned} \tag{3.31}$$

Note that the standard definition of a maximal congruence σ for any (not fixed) algebra $\mathbb{B}$ of size n is $\Pi_1^{1,b}$:

$$\begin{aligned} maxCong_m(B,\Omega,\sigma) \iff Cong_m(B,\Omega,\sigma) \wedge \exists a,b \in B,\ \neg\sigma(a,b) \wedge \\ \wedge[\forall \sigma' < \langle n,n\rangle,\ (Cong_m(B,\Omega,\sigma') \wedge \exists a,b \in B,\ \neg\sigma'(a,b)) \rightarrow \\ \rightarrow \exists a,b \in B,\ \sigma(a,b) \wedge \neg\sigma'(a,b)]. \end{aligned} \tag{3.32}$$

Analogously we can define a minimal congruence σ, by relation $minCong_m(D,\Omega,\sigma)$. Each block of a factor set, denoted by D/σ, is represented by its minimum element (it exists by the $\Sigma_0^{1,b}$-MIN principle). Therefore, we also think of the factorized object D/σ as a set of numbers. When we consider any congruence σ on D, we do not need to claim the existence of sets D/σ and Ω/σ - there is a simple algorithm to construct them, and the construction is unique. First, we define the following string function

$$\begin{aligned} factorset(D,\sigma)(a) = D/\sigma(a) \iff a < |D| \wedge a \in D \wedge \\ \wedge(\forall a' \in D, \sigma(a,a') \rightarrow a \leq a'). \end{aligned} \tag{3.33}$$

To represent an element we define a number function $rep(a/\sigma,D,\sigma)$

$$a = rep(a/\sigma,D,\sigma) \iff \sigma(a,a/\sigma) \wedge factorset(D,\sigma)(a). \tag{3.34}$$

Finally, we can define a string function returning Ω/σ using a bit-defining axiom:

$$\begin{aligned} factor\omega(D,\Omega,\sigma)(b) = \Omega/\sigma(b) \iff \exists a_1 ... \exists a_m \exists c \in D/\sigma,\ b = \langle a_1,...,a_m,c\rangle \wedge \\ \exists a_1/\sigma ... \exists a_m/\sigma \exists c/\sigma \in D,\ c = rep(c/\sigma,D,\sigma) \wedge \bigwedge_{i<m} a_i = rep(a_i/\sigma,D,\sigma) \wedge \\ \wedge\Omega(a_1/\sigma,...,a_m/\sigma,c/\sigma). \end{aligned} \tag{3.35}$$

The following two claims follow straightforwardly from the definitions of congruence and WNU operation.

Claim 1. *Consider an algebra $\mathbb{D} = (D,\Omega_D)$, its subuniverse B and a congruence σ on D. Then V^0 proves that σ restricted to B is a congruence on B.*

Claim 2. *Consider an algebra $\mathbb{D} = (D,\Omega_D)$ with Ω being a special WNU operation, and a congruence σ on D. Then V^0 proves that for all $a \in D$, a congruence block $[a]/\sigma$ is a subuniverse of D.*

For any congruence σ on algebra $\mathbb{D} = (D,\Omega)$, for factor algebra $\mathbb{D}/\sigma$ we will define the quotient set of relation $\Gamma_{\mathcal{D}}/\sigma$ as follows:

$$\begin{aligned} \Gamma^1_{\mathcal{D}}/\sigma(j,a) \iff \forall a/\sigma \in D,\ Rep_m(a,a/\sigma,D/\sigma,D,\Omega,\sigma) \wedge \Gamma^1_{\mathcal{D}}(j,a/\sigma) \\ \Gamma^2_{\mathcal{D}}/\sigma(i,a,b) \iff \forall a/\sigma,b/\sigma \in D,\ \Gamma^2_{\mathcal{D}}(i,a/\sigma,b/\sigma) \wedge \\ \wedge Rep_m(a,a/\sigma,D/\sigma,D,\Omega,\sigma) \wedge Rep_m(b,b/\sigma,D/\sigma,D,\Omega,\sigma). \end{aligned} \tag{3.36}$$

Note that for some i,j, $\Gamma^1_{\mathcal{D},j}/\sigma$ and $\Gamma^2_{\mathcal{D},i}/\sigma$ are empty sets. We will use it in the definition of PC subuniverses. The formulas (3.36) follow from log-space reduction from CSP($\mathbb{D}/\sigma$) to CSP($\mathbb{D}$), see [2]. We want to stress it directly here, not to repeat it many times. The relation signatures of the structures corresponding to D and D/σ differ, and the relation $R \in \Gamma_{\mathcal{D}}/\sigma$ lifts to the relation $R' \in \Gamma_{\mathcal{D}}$ by the rule $\bar{a} \in R' \iff \bar{a}/\sigma \in R$. Thus, for any binary or unary relation preserved by Ω/σ on D/σ its corresponding lifted relation is preserved by Ω on D. That is, we already have all such relations in $\Gamma_{\mathcal{A}}$. Moreover, for any

binary relation R on $D_i/\sigma_i \times D_j/\sigma_j$ preserved by $\Omega/\sigma = (\Omega/\sigma_i, \Omega/\sigma_j)$ its lifted relation on $D_i \times D_j$ is preserved by Ω under the same rule.

We define a binary relation $\mathscr{C}_\sigma$ on $D_0 \times ... \times D_{n-1}$ as a third-order object for all maps from $[n]$ to $(D_0, ..., D_{n-1})$, $\mathscr{C}_\sigma(H_1, H_2)$. For $\mathscr{C}_\sigma$ being compatible with Ω, we require that for any $H_1, ..., H_m, H'_1, ..., H'_m$:

$$\mathscr{C}_\sigma(H_1, H'_1) \wedge ... \wedge \mathscr{C}_\sigma(H_m, H'_m) \implies \mathscr{C}_\sigma(\omega(H_1, ..., H_m), \omega(, H'_1, ..., H'_m)). \tag{3.37}$$

For $\mathscr{C}_\sigma$ being a congruence, we additionally require that for any three maps H_1, H_2, H_3,

$$\begin{aligned} &\mathscr{C}_\sigma(H_1, H_1) \wedge (\mathscr{C}_\sigma(H_1, H_2) \leftrightarrow \mathscr{C}_\sigma(H_2, H_1)) \wedge \\ &\wedge (\mathscr{C}_\sigma(H_1, H_2) \wedge \mathscr{C}_\sigma(H_2, H_3) \rightarrow \mathscr{C}_\sigma(H_1, H_3)). \end{aligned} \tag{3.38}$$

We can restrict $\mathscr{C}_\sigma$ to any subuniverse $\mathscr{R}$ (we will call $\mathscr{C}^{\mathscr{R}}_\sigma$ a congruence restricted to $\mathscr{R}$) by requiring for all H, H',

$$\mathscr{C}^{\mathscr{R}}_\sigma(H, H') \implies \mathscr{R}(H') \wedge \mathscr{R}(H'). \tag{3.39}$$

Now we return to second-order congruences and extend them to third-order objects. The next three relations are expressed by $\Sigma^{1,b}_0$-formulas. For any congruence σ_i on D_i we say that two maps H_1, H_2 are in the same equivalence block on $D_0 \times ... \times D_{n-1}$ if

$$\begin{aligned} &1EqClass(i, H_1, H_2, D_i, \sigma_i) \iff \forall a_{i_1}, a_{i_2} < l, \\ &H_1(i) = \langle i, a_{i_1} \rangle \wedge H_2(i) = \langle i, a_{i_2} \rangle \rightarrow \sigma_i(a_{i_1}, a_{i_2}). \end{aligned} \tag{3.40}$$

Then for any congruence σ_i on D_i we define an extended relation $\mathscr{C}_{\sigma_i^{ext}}$ as follows:

$$\mathscr{C}_{\sigma_i^{ext}}(H_1, H_2) \iff 1EqClass(i, H_1, H_2, D_i, \sigma_i). \tag{3.41}$$

Analogously, for any $\sigma_0, ..., \sigma_{n-1}$ where each σ_i is a congruence on D_i, we define a relation $\mathscr{C}_{\cap_n \sigma_i^{ext}}$ on $D_0 \times ... \times D_{n-1}$ as follows:

$$\mathscr{C}_{\cap_n \sigma_i^{ext}}(H_1, H_2) \iff \forall i < n,\ 1EqClass(i, H_1, H_2, D_i, \sigma_i). \tag{3.42}$$

Notice that some congruences σ_i can be ∇_{D_i} or Δ_{D_i}. Obviously, for any three maps H_1, H_2, H_3 the relation $1EqClass$ is reflexive, symmetric, and transitive. Compatibility can be proved easily again. Consider $2m$ maps $H_1, ..., H_m$ and $H'_1, ..., H'_m$ such that for every $j < m$, $1EqClass(i, H_j, H'_j, D_i, \sigma_i)$. Then, due to defining equation (3.24) of ω,

$$1EqClass(i, \omega(H_1, ..., H_m), \omega(H'_1, ..., H'_2), D_i, \sigma_i).$$

Note that in (3.42) we define a third-order object using only its second-order properties. Thus, we have proved the following claims.

Claim 3. *Consider $D_0 \times ... \times D_{n-1}$, and binary relations $\sigma_0, ..., \sigma_{n-1}$ where σ_i is a congruence of D_i for every i. Then V^1 proves that any $\mathscr{C}_{\sigma_i^{ext}}$ and $\mathscr{C}_{\cap_n \sigma_i^{ext}}$ are congruences on $\mathscr{D}_0 \times ... \times \mathscr{D}_{n-1}$.*

Claim 4. *Consider $\mathscr{R}_\Theta \leq D_0 \times ... \times D_{n-1}$, and binary relations $\sigma_0, ..., \sigma_{n-1}$ where σ_i is a congruence of D_i for every i. Then V^1 proves that any $\mathscr{C}_{\sigma_i^{ext}}$ and $\mathscr{C}_{\cap_n \sigma_i^{ext}}$ restricted to $\mathscr{R}_\Theta$ are congruences on $\mathscr{R}_\Theta$.*

We get the following lemma with Claim 2.

Lemma 24. *Consider $\mathscr{R}_\Theta \leq D_0 \times ... \times D_{n-1}$, and binary relations $\sigma_0, ..., \sigma_{n-1}$ where σ_i is a congruence of D_i for every i. Suppose that E_i is a congruence block of σ_i for all i. Then V^1 proves that $(E_0 \times ... \times E_{n-1}) \cap \mathscr{R}_\Theta$ is a subuniverse of $\mathscr{R}_\Theta$.*

We need to define factor sets and factor operations for third-order objects. We first show how to define them for solution set $\mathscr{R}_\Theta$ and extended congruence $\mathscr{C}_{\sigma_i^{ext}}$ and the intersection of extended congruences $\mathscr{C}_{\cap_n \sigma_i^{ext}}$. To be compatible with the definition of factor sets for second-order objects, we need to choose not the existing map of $\mathscr{R}_\Theta$, but that sending i to $a_i = rep(a_i/\sigma_i, D_i, \sigma_i)$ for every σ_i.

$$\begin{gathered} factorset(\mathscr{R}_\Theta, \mathscr{C}_{\cap_n \sigma_i^{ext}})(H) = \mathscr{R}_\Theta/\mathscr{C}_{\cap_n \sigma_i^{ext}}(H) \iff \exists H' \leq \langle n, \langle n, l\rangle\rangle, H' \in \mathscr{R}_\Theta \wedge \\ \wedge \mathscr{C}_{\cap_n \sigma_i^{ext}}(H, H') \wedge \\ \wedge \forall i < n, \exists a_i \in D_i,\ H'(i) = \langle i, a_i/\sigma_i\rangle \wedge H(i) = \langle i, rep(a_i/\sigma_i, D_i, \sigma_i)\rangle. \end{gathered} \tag{3.43}$$

Note that the value of the function is a third-order object, but its definition is again essentially second-order. To define factor set for $\mathscr{C}_{\sigma_i^{ext}}$, consider $\mathscr{C}_{\cap_n \sigma_i^{ext}}$ where for each $j \neq i$, σ_j is ∇_{D_j}. To represent an element H' we define a string function $rep(H', \mathscr{R}_\Theta, \mathscr{C}_{\cap_n \sigma_i^{ext}})$:

$$H = rep(H', \mathscr{R}_\Theta, \mathscr{C}_{\cap_n \sigma_i^{ext}}) \iff \mathscr{C}_{\cap_n \sigma_i^{ext}}(H, H') \wedge factorset(\mathscr{R}_\Theta, \mathscr{C}_{\cap_n \sigma_i^{ext}})(H). \tag{3.44}$$

Finally, we define third-order valued function $factor\omega$:

$$\begin{gathered} factor\omega(\mathscr{R}_\Theta, \mathscr{F}_\Omega, \mathscr{C}_{\cap_n \sigma_i^{ext}})(B) = \mathscr{F}_{\Omega/\cap_n \sigma_i^{ext}}(B) \iff \exists H_1 ... \exists H_m \in \mathscr{R}_\Theta/\mathscr{C}_{\cap_n \sigma_i^{ext}}, \\ \exists H \in \mathscr{R}_\Theta/\mathscr{C}_{\cap_n \sigma_i^{ext}},\ B = \langle H_1, ..., H_m, H\rangle \wedge \\ \wedge \exists H_1' ... \exists H_m' \exists H' \in \mathscr{R}_\Theta \wedge H = rep(H', \mathscr{R}_\Theta, \mathscr{C}_{\sigma_i^{ext}}) \wedge \bigwedge_{i<m} H_i = rep(H_i'/\sigma, D, \sigma) \wedge \\ \mathscr{F}_{\Omega/\sigma_0, ..., \Omega/\sigma_{n-1}}(H_1, ..., H_m, H). \end{gathered} \tag{3.45}$$

As a factor algebra we consider a pair of classes $(\mathscr{R}_\Theta/\mathscr{C}_{\cap_n \sigma_i^{ext}}, \mathscr{F}_{\Omega/\cap_n \sigma_i^{ext}})$.

Now, to define a factor set for the general third-order subalgebra $\mathscr{R}$ and the congruence relation $\mathscr{C}_\sigma$, we need to choose a representant of a congruence block. It can be done by choosing the minimum string (in the sense of (3.4)) that represents maps from the block. The rest are defined analogously.

3.2.5 Homomorphism and isomorphism between second and third order objects

We say that there exists a homomorphism between two subalgebras (B, Ω_B), (C, Ω_C) of algebra $\mathbb{A}$ if

$$\begin{gathered} HOM_{alg}(B, \Omega_B, C, \Omega_C) \iff \exists H < \langle l, l\rangle, MAP(B, l, C, l, H) \wedge \\ \wedge \forall b_1, ..., b_m, b \in B, \Omega_B(b_1, ..., b_m, b) \leftrightarrow \Omega_C(H(b_1), ..., H(b_m), H(b)). \end{gathered} \tag{3.46}$$

The image and kernel of B under H can be returned by string-valued functions defined as follows:

$$\begin{gathered} img(B, H)(i) \iff i \in C \wedge \exists j \in B,\ H(j) = i, \\ ker(H)(i, j) \iff i, j \in B \wedge H(i) = H(j). \end{gathered} \tag{3.47}$$

We can easily formalize embedding, epimorphism, and isomorphism:

$$\begin{gathered} ISO_{alg}(B, \Omega_B, C, \Omega_C) \iff \exists H < \langle l, n\rangle, HOM_{alg}(B, \Omega_B, C, \Omega_C) \wedge \\ \wedge \forall i_1, i_2 \in B,\ (H(i_1) = H(i_2) \to i_1 = i_2) \wedge \forall j \in C, \exists i \in B, H(i) = j. \end{gathered} \tag{3.48}$$

Now, we define a relation of being isomorphic between third-order objects and second-order objects. This relation assumes the existence of the third-order object – a class of maps. We say that $\mathscr{M}$ is a *well-defined map* between a class $\mathscr{R}$ and a set D if

$$\begin{gathered} MAP^{3,2}(\mathscr{R}, D, \mathscr{M}) \iff \forall H \in \mathscr{R} \exists a \in D,\, \mathscr{M}(H, a) \wedge \\ \forall H \in \mathscr{R}\, \forall a, b \in D\, (\mathscr{M}(H, a) \wedge \mathscr{M}(H, b) \to a = b). \end{gathered} \tag{3.49}$$

We say that $\mathscr{H}$ is a homomorphism from a class $(\mathscr{R}, \mathscr{F})$ to an algebra (D, Ω) if

$$\begin{gathered} HOM^{3,2}_{alg}(\mathscr{R}, \mathscr{F}, D, \Omega, \mathscr{H}) \iff MAP^{3,2}(\mathscr{R}, D, \mathscr{H}) \wedge \\ \wedge \forall H_1, ..., H_m, H \in \mathscr{R}, \mathscr{F}(H_1, ..., H_m, H) \leftrightarrow \Omega(\mathscr{M}(H_1), ..., \mathscr{M}(H_m), \mathscr{M}(H)), \end{gathered} \tag{3.50}$$

and that the class $(\mathscr{R}, \mathscr{F})$ is isomorphic to the set (D, Ω) if

$$\begin{gathered} ISO^{3,2}_{alg}(\mathscr{R}, \mathscr{F}, D, \Omega) \iff \exists \mathscr{H},\, HOM^{3,2}(\mathscr{R}, \mathscr{F}, D, \Omega, \mathscr{H}) \wedge \forall H_1, H_2 \in \mathscr{R}, \\ (\mathscr{H}(H_1) = \mathscr{H}(H_2) \to H_1 = H_2) \wedge \forall a \in D, \exists H \in \mathscr{R}, \mathscr{H}(H) = a. \end{gathered} \tag{3.51}$$

Analogously, we can define a map $MAP^{2,3}$, a homomorphism $HOM^{2,3}_{alg}$, and an isomorphism $ISO^{2,3}_{alg}$ from a set to a class. Finally, we define an isomorphism between third-order objects. We say that $\mathscr{M}$ is a *well-defined map* between a class $\mathscr{R}$ and a class $\mathscr{R}'$ if

$$\begin{gathered} MAP^{3,3}(\mathscr{R}, \mathscr{R}', \mathscr{M}) \iff \forall H \in \mathscr{R} \exists H' \in \mathscr{R},\, \mathscr{M}(H, H') \wedge \\ \forall H \in \mathscr{R}\, \forall H_1, H_2 \in \mathscr{R}'\, (\mathscr{M}(H, H_1) \wedge \mathscr{M}(H, H_2) \to H_1 = H_2). \end{gathered} \tag{3.52}$$

We say that $\mathscr{H}$ is a homomorphism from a class $(\mathscr{R}, \mathscr{F})$ to a class $(\mathscr{R}', \mathscr{F}')$ if

$$\begin{gathered} HOM^{3,3}_{alg}(\mathscr{R}, \mathscr{F}, \mathscr{R}', \mathscr{F}', \mathscr{H}) \iff MAP^{3,3}(\mathscr{R}, \mathscr{R}', \mathscr{H}) \wedge \\ \wedge \forall H_1, ..., H_m, H \in \mathscr{R}, \mathscr{F}(H_1, ..., H_m, H) \leftrightarrow \mathscr{F}'(\mathscr{M}(H_1), ..., \mathscr{M}(H_m), \mathscr{M}(H)), \end{gathered} \tag{3.53}$$

and, finally,

$$\begin{gathered} ISO^{3,3}_{alg}(\mathscr{R}, \mathscr{F}, \mathscr{R}', \mathscr{F}') \iff \exists \mathscr{H},\, HOM^{3,3}(\mathscr{R}, \mathscr{F}, \mathscr{R}', \mathscr{F}', \mathscr{H}) \wedge \forall H_1, H_2 \in \mathscr{R}, \\ (\mathscr{H}(H_1) = \mathscr{H}(H_2) \to H_1 = H_2) \wedge \forall H' \in \mathscr{R}', \exists H \in \mathscr{R}, \mathscr{H}(H) = H'. \end{gathered} \tag{3.54}$$

For every domain D_i and any of its subuniverse B_i, we define its extension $\mathscr{B}^{ext}_i$ to third-order object, as a set of maps from $[n]$ to $[D_0, ..., D_{n-1}]$ such that it contains all maps sending i to elements of B_i:

$$\mathscr{B}^{ext}_i(H) \iff \exists a_i \in B_i,\, H(i) = \langle i, a_i \rangle. \tag{3.55}$$

3.2.6 Auxiliary definitions from Zhuk's algorithm

Some notions, which were used in Zhuk's algorithm mainly in relation to constraints, we will give both for binary relations and n-ary relations.

3.2.6.1 Crucial Instance

For an instance Θ a constraint C is called *crucial* in $D^{(\perp)} = (D_0^{(\perp)}, ..., D_{n-1}^{(\perp)})$, where $D_i^{(\perp)} \subseteq D_i$ for each i, if it does not have dummy variables, Θ has no solutions in $D^{(\perp)}$, but the replacement of C by all weaker constraints gives an instance with a solution in $D^{(\perp)}$. A CSP instance Θ is crucial in $D^{(\perp)}$ if every constraint of Θ is crucial in $D^{(\perp)}$. In this section we will formalize this notion.

Definition 51 (Reduction of the domain set)**.** For an instance $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$ with domain set $D = (D_0, ..., D_{n-1})$ we say that a set $D^{(\perp)} = (D_0^{(\perp)}, ..., D_{n-1}^{(\perp)})$ is a reduction of D if $D_i^{(\perp)}$ is a subuniverse of D_i for every i.

$$Red(D^{(\perp)}, D) \iff \forall i \leq n,\ subTA(D_i^{(\perp)}, D_i). \tag{3.56}$$

In the definition, we can additionally require that equal domains be reduced to equal domains, i.e.

$$\forall i \forall j,\ D_i = D_j \to D_i^{(\perp)} = D_j^{(\perp)}. \tag{3.57}$$

We shall use it later considering different modifications of the instance to avoid abuse of the notation.

Definition 52 (Instance after reduction)**.** For an instance $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$, we need to define an instance $\Theta^{(\perp)} = (\mathcal{X}^{(\perp)}, \ddot{\mathcal{A}}^{(\perp)})$ after the reduction of a domain set of a target digraph $\ddot{\mathcal{A}} = (V_{\ddot{\mathcal{A}}}, E_{\ddot{\mathcal{A}}})$ from $D = (D_0, ..., D_{n-1})$ to $D^{(\perp)} = (D_0^{(\perp)}, ..., D_{n-1}^{(\perp)})$. Since there is a unique way to construct a reduction of an instance, we actually can define a string function (using a bit-defining axiom) that returns a reduced instance:

$$\begin{gathered} redinst(\Theta, D^{(\perp)})(\mathcal{X}^{(\perp)}, \ddot{\mathcal{A}}^{(\perp)}) = \Theta^{(\perp)}(\mathcal{X}^{(\perp)}, \ddot{\mathcal{A}}^{(\perp)}) \iff Red(D^{(\perp)}, D) \wedge \\ \wedge (\mathcal{X}^{(\perp)} = \mathcal{X}) \wedge \\ \wedge (\forall i, j < n, \forall a, b < l, E^{ij}_{\ddot{\mathcal{A}}^{(\perp)}}(a, b) \leftrightarrow E^{ij}_{\ddot{\mathcal{A}}}(a, b) \wedge a \in D_i^{(\perp)} \wedge b \in D_j^{(\perp)}). \end{gathered} \tag{3.58}$$

We say that $D^{(\perp)}$ is a 1-consistent reduction if the instance $\Theta^{(\perp)}$ is 1-consistent, $1C(\Theta^{(\perp)})$.

Definition 53 (Linked component)**.** Sometimes when we work with a non-linked instance, we need to produce its *linked component*, i.e. elements that can be connected by a path in the instance. Recall that two elements $a \in D_i, b \in D_j$ are linked, if there exists 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 a $\Sigma_1^{1,b}$-formula,

$$\exists \mathcal{P}_t < (nl, 4(nl)^2), Linked(a, b, i, j, \Theta, \mathcal{P}_t).$$

We have formalized in Chapter 2 (ref. [6]) that $Linked(a, b, i, i, \Theta)$ is a congruence relation on D_i, and that for a non-fragmented instance, this congruence provides a partition into linked components. That is, each linked component can be viewed as the same CSP instance on smaller domains. We define a string function $linkcomp(\Theta, D_i, a)$ that produces a linked reduction of the domain set based on an element a in the domain D_i.

$$\begin{gathered} linkcomp(\Theta, D_i, a)(j, b) = V_{\ddot{\mathcal{A}}}^{link,i,a}(j, b) \iff \exists \mathcal{P}_t < (nl, 4(nl)^2), \\ Linked(a, b, i, j, \Theta, \mathcal{P}_t). \end{gathered} \tag{3.59}$$

Then a $\Sigma_1^{1,b}$-function

$$redinst(\Theta, linkcomp(\Theta, D_i, a))$$

produces a linked reduction of instance Θ, containing the element a in the domain D_i.

Definition 54 (Dummy variable)**.** 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 $a \in D_i$, $E^{ij}_{\ddot{\mathcal{A}}}(a, b)$, there is an edge $(a', b) \in E^{ij}_{\ddot{\mathcal{A}}}$ for every $a' \in D_i$.

$$Dum_2(E^{ij}_{\ddot{\mathcal{A}}}, i) \iff \forall b \in D_j, (\exists a \in D_i, E^{ij}_{\ddot{\mathcal{A}}}(a, b) \to \forall a' \in D_i, E^{ij}_{\ddot{\mathcal{A}}}(a', b)). \tag{3.60}$$

Note that for a 1-consistent CSP instance this means that $E^{ij}_{\ddot{\mathcal{A}}}$ is a full relation:

$$FullRel(E^{ij}_{\ddot{\mathcal{A}}}) \iff \forall a \in D_i, \forall b \in D_j, E^{ij}_{\ddot{\mathcal{A}}}(a, b). \tag{3.61}$$

We also introduce the notion of being a dummy variable for a solution set $\mathscr{R}_\Theta$. We say that a variable x_i is dummy if the following $\Pi^{1,b}_2$-relation holds:

$$\begin{gathered} Dum(\mathscr{R}_\Theta, i) \iff \forall H \leq \langle n, \langle n, l\rangle\rangle, (H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}}, H) \to \forall a \in D_i, \\ \exists H' \leq \langle n, \langle n, l\rangle\rangle, H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}}, H') \wedge H'(i) = \langle i, a\rangle \wedge \forall j \neq i < n,\, H'(j) = H(j)). \end{gathered} \tag{3.62}$$

Definition 55 (Weakening of a constraint)**.** For a binary constraint $E^{ij}_{\ddot{\mathcal{A}}}$ we have only two types of weaker constraints: domains D_i, D_j which are weaker constraints of less arity (which we never increase), and all binary constraints from the list $\Gamma_{\mathcal{A}}$ containing $E^{ij}_{\ddot{\mathcal{A}}}$, including the full relation on $D_i \times D_j$ (as if we remove a constraint at all). We say that E is a *weaker constraint* than $E^{ij}_{\ddot{\mathcal{A}}}$ if

$$\begin{gathered} Weaker(E, E^{ij}_{\ddot{\mathcal{A}}}) \iff Pol_{m,2}(\Omega, A, E) \wedge (\forall a, b < l, E(a, b) \to \\ \to a \in D_i \wedge b \in D_j) \wedge \big[FullRel(E) \vee \\ \vee \big((\forall a \in D_i, \forall b \in D_j,\, E^{ij}_{\ddot{\mathcal{A}}}(a, b) \to E(a, b)) \wedge (\exists a \in D_i, \\ \exists b \in D_j,\, E(a, b) \wedge \neg E^{ij}_{\ddot{\mathcal{A}}}(a, b))\big)\big]. \end{gathered} \tag{3.63}$$

Note that for any constraint $E^{ij}_{\ddot{\mathcal{A}}}$ there exists at least one weaker constraint (namely the full relation). Any time we weaken a constraint $E^{ij}_{\ddot{\mathcal{A}}}$ we replace it with all weaker constraints simultaneously. That is, we consider an intersection of all weaker constraints. But since in the list $\Gamma_{\mathcal{A}}$ we have all *pp*-definable binary relations invariant under Ω, there exists $k < 2^{l^2}$ such that $\Gamma^2_{\mathcal{A},k}$ is that intersection. A problem here arises when the intersection of all weaker constraints of a constraint is the constraint itself: it just means that there are several incomparable intersections of weaker constraints. In this case, we can choose one of them arbitrarily, and we will choose the one with the smallest $k < 2^{l^2}$. We first define a string function that returns the list of such intersections:

$$\begin{gathered} weakerlist(E^{ij}_{\ddot{\mathcal{A}}})(k) \iff Weaker(\Gamma^2_{\mathcal{A},k}, E^{ij}_{\ddot{\mathcal{A}}}) \wedge \forall g \neq k < 2^{l^2}, \\ \neg(Weaker(\Gamma^2_{\mathcal{A},g}, E^{ij}_{\ddot{\mathcal{A}}}) \wedge Weaker(\Gamma^2_{\mathcal{A},k}, \Gamma^2_{\mathcal{A},g})). \end{gathered} \tag{3.64}$$

Then we define a string function that returns the first intersection from this list. We will call it the *weakening of the constraint* $E^{ij}_{\ddot{\mathcal{A}}}$ and denote by $E^{ij}_{\ddot{\mathcal{A}},w}$:

$$\begin{gathered} weakening(E^{ij}_{\ddot{\mathcal{A}}})(a, b) = E^{ij}_{\ddot{\mathcal{A}},w}(a, b) \iff \exists i < 2^{l^2}, (weakerlist(E^{ij}_{\ddot{\mathcal{A}}})(i) \wedge \\ \wedge \Gamma^2_{\mathcal{A},i}(a, b)) \wedge \forall j < i, \neg weakerlist(E^{ij}_{\ddot{\mathcal{A}}})(j). \end{gathered} \tag{3.65}$$

The function is well-defined due to the Number Minimization axiom $\Sigma^{1,b}_0$-MIN (see Chapter 2, ref. [6]) and since $weakerlist(E^{ij}_{\ddot{\mathcal{A}}})$ is never empty. Thus, we can uniquely define the instance after weakening of a constraint $E^{ij}_{\ddot{\mathcal{A}}}$:

Definition 56 (Instance after weakening)**.** For an instance $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$, the instance $\Theta_{w^{ij}} = (\mathcal{X}_{w^{ij}}, \ddot{\mathcal{A}}_{w^{ij}})$ after the weakening of a constraint $E^{ij}_{\ddot{\mathcal{A}}}$ is defined by the following string function:

$$\begin{gathered} weakinst(\Theta, E^{ij}_{\ddot{\mathcal{A}}})(\mathcal{X}_{w^{ij}}, \ddot{\mathcal{A}}_{w^{ij}}) = \Theta_{w^{ij}}(\mathcal{X}_{w^{ij}}, \ddot{\mathcal{A}}_{w^{ij}}) \iff (D_{w^{ji}} = D) \wedge \\ \wedge (V_{\mathcal{X}_{w^{ij}}} = V_{\mathcal{X}}) \wedge (\forall t \neq i < n, \forall r \neq j < n, (E_{\mathcal{X}_{w^{ij}}}(t,r) \leftrightarrow E_{\mathcal{X}}(t,r)) \wedge \\ \wedge (E^{tr}_{\ddot{\mathcal{A}}_{w^{ij}}} = E^{tr}_{\ddot{\mathcal{A}}})) \wedge \\ \wedge E^{ij}_{\ddot{\mathcal{A}}_{w^{ij}}} = weakening(E^{ij}_{\ddot{\mathcal{A}}}) \wedge (FullRel(E^{ij}_{\ddot{\mathcal{A}}_{w^{ij}}}) \leftrightarrow \neg E_{\mathcal{X}}(i,j)). \end{gathered} \tag{3.66}$$

Note that the last line ensures that if the only weaker binary relation to $E^{ij}_{\ddot{\mathcal{A}}}$ is a full relation, then we remove an edge from $\mathcal{X}$.

Definition 57 (Crucial instance)**.** Let $D_i^{(\perp)} \subseteq D_i$ for every i, and let $D^{(\perp)}$ be a reduction of D. A constraint $E^{ij}_{\ddot{\mathcal{A}}}$ of instance Θ is called crucial in $D^{(\perp)}$ if

$$\begin{gathered} CrucConst(E^{ij}_{\ddot{\mathcal{A}}}, \Theta, D^{(\perp)}) \iff \neg Dum_2(E^{ij}_{\ddot{\mathcal{A}}}, i) \wedge \neg Dum_2(E^{ij}_{\ddot{\mathcal{A}}}, j) \wedge \\ \neg H\ddot{O}M(\mathcal{X}^{(\perp)}, \ddot{\mathcal{A}}^{(\perp)}) \wedge H\ddot{O}M(\mathcal{X}^{(\perp)}_{w^{ij}}, \ddot{\mathcal{A}}^{(\perp)}_{w^{ij}}). \end{gathered} \tag{3.67}$$

We say that a CSP instance $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$ is crucial in $D^{(\perp)}$ if

$$CrucInst(\Theta, D^{(\perp)}) \iff \forall j, i < n, E_{\mathcal{X}}(i,j) \to CrucConst(E^{ij}_{\ddot{\mathcal{A}}}, \Theta, D^{(\perp)}). \tag{3.68}$$

Note that all formulas used in the definitions of this section except Definition 57, are $\Sigma_0^{1,b}$. Formulas (3.67) and (3.68) are from the class $\mathfrak{B}(\Sigma_1^{1,b})$, Boolean combinations of $\Sigma_1^{1,b}$-formulas.

3.2.6.2 Covering and expanded covering

We can consider a CSP instance Θ on n variables as a set of constraints of the form $E^{ij}_{\ddot{\mathcal{A}}}$ for all $i, j < n$ (we do not consider domains here as constraints).

Definition 58 (Tree-instance)**.** We say that an instance Θ is a *tree-formula* if there is no path $z_1 - C_1 - z_2 - ... - z_{l-1} - C_l - z_l$ such that $l \leq 3$, $z_1 = z_l$, and all the constraints $C_1, C_2, ..., C_l$ are different. Since in our setting for any $i, j < n$ we suppose that there is only one constraint relation $E^{ij}_{\ddot{\mathcal{A}}}$ (we can do this because we have any intersection of any invariant relations in our list $\Gamma_{\mathcal{A}}$), an instance $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$ is a tree-formula if it does not contain cycles. It can be expressed by the following $\Pi_1^{1,b}$-formula:

$$\begin{gathered} TreeInst(\mathcal{X}, \ddot{\mathcal{A}}) \iff \forall t < n^2, \forall V_{\mathcal{C}_t} = t, \forall E_{\mathcal{C}_t} \leq 4t^2, \forall H < \langle t, n \rangle, \\ CYCLE(V_{\mathcal{C}_t}, E_{\mathcal{C}_t}) \wedge HOM(\mathcal{C}_t, \mathcal{X}, H) \to \exists i_1 \neq j_1 < t, \exists i_2 \neq j_2 < t, \exists k_1, k_2 < n, \\ E_{\mathcal{C}_t}(i_1, i_2) \wedge E_{\mathcal{C}_t}(j_1, j_2) \wedge H(i_1, k_1) \wedge H(i_2, k_2) \wedge H(j_1, k_1) \wedge H(j_2, k_2). \end{gathered} \tag{3.69}$$

That is, for any cycle $\mathcal{C}_t$, any homomorphism from $\mathcal{C}_t$ to $\mathcal{X}$ must glue at least two different edges of $\mathcal{C}_t$.

Definition 59 (Subinstance)**.** For instance $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$ we call $\Theta' = (\mathcal{X}', \ddot{\mathcal{A}})$ a *subinstance* of Θ if Θ' is a subset of the variables together with some subset of constraints from Θ that only involve these variables, i.e.:

$$\begin{gathered} subInst(\Theta', \Theta) \iff \ddot{\mathcal{A}}' = \ddot{\mathcal{A}} \wedge V_{\mathcal{X}'} \subseteq V_{\mathcal{X}} \wedge E_{\mathcal{X}'} \subseteq E_{\mathcal{X}} \wedge \\ (E_{\mathcal{X}'}(x_1, x_2) \to x_1, x_2 \in V_{\mathcal{X}'}). \end{gathered} \tag{3.70}$$

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 $V_{\mathcal{X}'}$. The solution to such a subinstance is a partial homomorphism.

Consider instance Θ as a set of constraints $\{E^{ij}_{\ddot{\mathcal{A}}} : i, j < n\}$. Then consider a subset Θ' of such constraints, a subinstance of Θ. We need to define uniquely a subinstance $\Theta\backslash\Theta'$ as a string function:

$$\begin{gathered} dif(\Theta,\Theta')(\mathcal{X}_{\Theta\backslash\Theta'},\ddot{\mathcal{A}}_{\Theta\backslash\Theta'}) = \Theta\backslash\Theta'(\mathcal{X}_{\Theta\backslash\Theta'},\ddot{\mathcal{A}}_{\Theta\backslash\Theta'}) \iff \ddot{\mathcal{A}}_{\Theta\backslash\Theta'} = \ddot{\mathcal{A}}\wedge \\ \wedge\forall i,j<n,\ E_{\mathcal{X}\backslash\mathcal{X}'}(i,j) \leftrightarrow E_{\mathcal{X}}(i,j)\wedge\neg E_{\mathcal{X}'}(i,j)\wedge \\ \wedge V_{\mathcal{X}\backslash\mathcal{X}'}\subseteq V_{\mathcal{X}}\wedge(\forall i<n, V_{\mathcal{X}\backslash\mathcal{X}'}(i)\leftrightarrow\exists j<n,\ \neg E_{\mathcal{X}'}(i,j)\wedge\neg E_{\mathcal{X}'}(i,j)\wedge \\ \wedge(E_{\mathcal{X}}(i,j)\vee E_{\mathcal{X}}(i,j)). \end{gathered} \tag{3.71}$$

Note that Θ' and $\Theta\backslash\Theta'$ can share common variables, so the third line in the formula places to $V_{\mathcal{X}\backslash\mathcal{X}'}$ only variables that are involved in some constraint not in Θ'. We also lose all the variables that are not involved in any constraint, neither in Θ' nor in Θ.

For the rest part of this section and sometimes further when we talk about (expanded) covering and substitutions, we will use labels for vertex sets instead of elements. For any instance Θ with a vertex set $V_{\mathcal{X}}$ we can introduce as many labels as we want using two-dimensional strings Y, Z, W, and the function $seq(i, Y) = y_i$. They are bounded on the first coordinate by the number of vertices and on the second coordinate by some reasonable number of labels. Let us denote this bound for n variables by b_n. We will use $y_i, z_j, w_k < b_n$ in the formulas when appropriate. When we write $\forall i < n$, $R(y_i)$, this is an abbreviation for

$$\forall i < n,\ R(seq(i, Y)).$$

The representation of the entire vertex set for a digraph $\mathcal{X}$ is $V_{\mathcal{X}}(i, x_i)$, and the representation of the set of vertices of a digraph $\ddot{\mathcal{A}}$ is $V(\langle seq(i, V_{\mathcal{X}}), a\rangle)$, which does not differ much from our usual representations. For an instance Λ and two sets of variables $z_1, ..., z_k$ and $y_1, ..., y_k$ by $\Lambda^{y_1,...,y_k}_{z_1,...,z_k}$ we denote the instance obtained from instance Λ by replacing every variable z_i by y_i. This can be expressed by a $\Sigma^{1,b}_0$ string function

$$substitute(\Lambda, Y, Z)(\mathcal{X}_{\Lambda^{y_1,...,y_k}_{z_1,...,z_k}},\ddot{\mathcal{A}}_{\Lambda^{y_1,...,y_k}_{z_1,...,z_k}}) = \Lambda^{y_1,...,y_k}_{z_1,...,z_k}(\mathcal{X}_{\Lambda^{y_1,...,y_k}_{z_1,...,z_k}},\ddot{\mathcal{A}}_{\Lambda^{y_1,...,y_k}_{z_1,...,z_k}}),$$

which definition is rather tedious than interesting, so we do not present it here. We also need to define a union of two sets of constraints, i.e. a union of two instances $\Theta_{\mathcal{X}} = (\mathcal{X}, \ddot{\mathcal{A}})$ with $x_0, ...x_{n-1}$ variables, $\ddot{\mathcal{A}} = (V_{\ddot{\mathcal{A}}}, E_{\ddot{\mathcal{A}}})$, and $\Theta_{\mathcal{Y}} = (\mathcal{Y}, \ddot{\mathcal{B}})$ with $y_0, ..., y_{m-1}$ variables, $\ddot{\mathcal{B}} = (V_{\ddot{\mathcal{B}}}, E_{\ddot{\mathcal{B}}})$. The problem here is that they can share common variables (that are labeled by the same number). To be safe and to easily track the number of variables, we just copy the common variables x_i, y_j, and set $E^{x_i y_j}_{\ddot{\mathcal{A}}\cup\ddot{\mathcal{B}}}$ to be equality relation (we have it since in our list $\Gamma_{\mathcal{A}}$ we have all relations *pp*-definable from Γ). To copy variables without collisions, we first define a number function.

$$maxlabel(V_{\mathcal{X}}) = s \iff \forall i < n,\ seq(i, V_{\mathcal{X}}) \leq s \wedge \exists i < n,\ seq(i, V_{\mathcal{X}}) = s, \tag{3.72}$$

and use a label $z_i = y_i + maxlabel(V_{\mathcal{X}})$ for every $i < m$ in the following definition. Then

we define a string function *uni* on two arguments by its bit-definition:

$$\begin{gathered}
uni(\Theta_{\mathcal{X}},\Theta_{\mathcal{Y}})(\mathcal{X}\cup\mathcal{Y},\ddot{\mathcal{A}}\cup\ddot{\mathcal{B}})=\Theta_{\mathcal{X}}\cup\Theta_{\mathcal{Y}}(\mathcal{X}\cup\mathcal{Y},\ddot{\mathcal{A}}\cup\ddot{\mathcal{B}})\iff\\
\forall i<n,\,V_{\mathcal{X}\cup\mathcal{Y}}(i,x_i)\wedge\forall i<m,\,V_{\mathcal{X}\cup\mathcal{Y}}(n+i,z_i)\wedge\\
\wedge\forall i,j<n,\,E_{\mathcal{X}\cup\mathcal{Y}}(x_i,x_j)\leftrightarrow E_{\mathcal{X}}(x_i,x_j)\wedge\forall i,j<m,\,E_{\mathcal{X}\cup\mathcal{Y}}(z_i,z_j)\leftrightarrow E_{\mathcal{Y}}(y_i,y_j)\wedge\\
\wedge\forall i<n,\forall a<l,\,V_{\ddot{\mathcal{A}}\cup\ddot{\mathcal{B}}}(x_i,a)\leftrightarrow V_{\ddot{\mathcal{A}}}(x_i,a)\wedge\\
\wedge\forall i<m,\forall a<l,\,V_{\ddot{\mathcal{A}}\cup\ddot{\mathcal{B}}}(z_i,a)\leftrightarrow V_{\ddot{\mathcal{B}}}(y_i,a)\wedge\\
\wedge\forall i,j<n,\forall a,b<l,\,E^{x_ix_j}_{\ddot{\mathcal{A}}\cup\ddot{\mathcal{B}}}(a,b)\leftrightarrow E^{x_ix_j}_{\ddot{\mathcal{A}}}(a,b)\wedge\\
\wedge\forall i,j<m,\forall a,b<l,\,E^{z_iz_j}_{\ddot{\mathcal{A}}\cup\ddot{\mathcal{B}}}(a,b)\leftrightarrow E^{y_iy_j}_{\ddot{\mathcal{B}}}(a,b)\wedge\\
\wedge\forall i<n\forall j<m,\,x_i=y_j\rightarrow\forall a\in D_{x_i},\,E^{x_iz_j}_{\ddot{\mathcal{A}}\cup\ddot{\mathcal{B}}}(a,a).
\end{gathered}\tag{3.73}$$

Due to this definition, function *uni* is not commutative, but $\Theta_{\mathcal{X}}\cup\Theta_{\mathcal{Y}}$ and $\Theta_{\mathcal{Y}}\cup\Theta_{\mathcal{X}}$ are obviously isomorphic. We can iteratively define $\Theta_{\mathcal{X}_1}\cup\Theta_{\mathcal{X}_2}\cup...\cup\Theta_{\mathcal{X}_n}=(\Theta_{\mathcal{X}_1}\cup\Theta_{\mathcal{X}_2}\cup...\cup\Theta_{\mathcal{X}_{n-1}})\cup\Theta_{\mathcal{X}_n}$.

A *pp*-formula $\exists y_0,...,y_{k-1}\Theta'(x_0,...,x_{n-1})$, where $y_0,...,y_{k-1}$ are the only variables occurring in Θ' except $x_0,...,x_{n-1}$, is called a *subconstraint* of Θ if $\Theta'\subseteq\Theta$ and Θ' and $\Theta\backslash\Theta'$ do not have common variables except for $x_0,...,x_{n-1}$. We can consider Θ' as a subinstance that involves variables $x_0,...,x_{n-1},y_0,...,y_{k-1}$, and Θ as an instance on variables $x_0,...,x_{n-1},y_0,...,y_{k-1},z_0,...,z_{s-1}$. Constraints involving variables $y_0,...,y_{k-1}$ occur only in Θ', and constraints involving $z_0,...,z_{s-1}$ occur only in $\Theta\backslash\Theta'$. We code common variables by a set X. Then we can define a subconstraint in the following way:

$$\begin{gathered}
subConst(\Theta,\Theta',X)\iff subInst(\Theta',\Theta)\wedge\forall i,j,k<(n+k+s),\\
E_{\mathcal{X}'}(i,j)\wedge E_{\mathcal{X}\backslash\mathcal{X}'}(j,k)\rightarrow\exists s<(n+k+s),j=X(s,x_s).
\end{gathered}\tag{3.74}$$

Here, for brevity's sake, we abbreviate by $E_{\mathcal{X}}(i,j)\wedge E_{\mathcal{X}\backslash\mathcal{X}'}(j,k)$ all four combinations of non-symmetric constraints. Then $\exists y_0,...,y_{k-1}\Theta'(x_0,...,x_{n-1})$ defines a projection of solution set to CSP instance Θ' on the coordinates $x_0,...,x_{n-1}$, $\mathscr{R}^{x_0,...,x_{n-1}}_{\Theta'}$.

Definition 60 (Covering and Expanded covering)**.** For an instance $\Theta_{\mathcal{X}}=(\mathcal{X},\ddot{\mathcal{A}})$ with $x_0,...x_{n-1}$ variables, $\ddot{\mathcal{A}}=(V_{\ddot{\mathcal{A}}},E_{\ddot{\mathcal{A}}})$, we say that an instance $\Theta_{\mathcal{Y}}=(\mathcal{Y},\ddot{\mathcal{B}})$ with $y_0,...,y_{m-1}$ variables and $\ddot{\mathcal{B}}=(V_{\ddot{\mathcal{B}}},E_{\ddot{\mathcal{B}}})$ is a covering of Θ if the following $\Sigma_1^{1,b}$-relation holds:

$$\begin{gathered}
Cov(\Theta_{\mathcal{Y}},\Theta_{\mathcal{X}})\iff\exists H<\langle b_m,b_n\rangle,HOM(\mathcal{Y},\mathcal{X})\wedge\\
\wedge\forall i<m,\,H(y_i)=x_j\rightarrow D_{y_i}=D_{x_j}\wedge\\
\wedge\forall i,j<m,\,E_{\mathcal{Y}}(y_i,y_j)\wedge H(y_i)=x_k\wedge H(y_j)=x_p\rightarrow\forall a\in D_{y_i},\forall b\in D_{y_j},\\
E^{y_iy_j}_{\ddot{\mathcal{B}}}(a,b)\leftrightarrow E^{x_kx_p}_{\ddot{\mathcal{A}}}(a,b)\wedge\forall i<m,\forall j<n,\,y_i=x_j\rightarrow H(y_j)=x_i.
\end{gathered}\tag{3.75}$$

That is, for our purpose, a covering is another instance with different $\mathcal{X}'$ and $\ddot{\mathcal{A}}'$ such that

1. The domain of every variable y_i in $\Theta_{\mathcal{Y}}$ is equal to the domain of $H(y_i)$ in $\Theta_{\mathcal{X}}$.

2. There is a homomorphism from $\mathcal{Y}$ to $\mathcal{X}$ (for any constraint $(y_i,y_j;E_{\ddot{\mathcal{B}}})$) of $\Theta_{\mathcal{Y}}$, $(H(y_i),H(y_j);E_{\ddot{\mathcal{A}}})$ is a constraint of $\Theta_{\mathcal{X}}$ such that $E_{\ddot{\mathcal{A}}}$ and $E_{\ddot{\mathcal{B}}}$ here are the same relation but for different variables.

3. If a variable y appears in both $\Theta_{\mathcal{X}}$ and $\Theta_{\mathcal{Y}}$, we just assume that $H(y)=y$.

We say that $\Theta_{\mathcal{Y}}$ is an expanded covering if

$$\begin{gathered} ExpCov(\Theta_{\mathcal{Y}}, \Theta_{\mathcal{X}}) \iff \exists H < \langle b_m, b_n \rangle, HOM(\mathcal{Y}, \mathcal{X}) \wedge \\ \wedge \forall i < n,\, H(y_i) = x_j \to D_{y_i} = D_{x_j} \wedge \forall i, j < m, \\ ((E_{\mathcal{Y}}(y_i, y_j) \wedge H(y_i) = x_k \neq H(y_j) = x_p \to \\ \to (\forall a \in D_{x_k}, \forall b \in D_{x_p}, E_{\ddot{\mathcal{A}}}^{x_k x_p}(a, b) \to E_{\ddot{\mathcal{B}}}^{y_i y_j}(a, b))) \wedge \\ \wedge ((E_{\mathcal{Y}}(y_i, y_j) \wedge H(y_i) = H(y_j) = x_k \to \forall a \in D_{y_i},\, E_{\ddot{\mathcal{B}}}^{y_i y_j}(a, a))). \end{gathered} \tag{3.76}$$

That is, for our purpose, an expanded covering is another instance with different $\mathcal{X}'$ and $\ddot{\mathcal{A}}'$ such that:

1. The domain of every variable y_i in $\Theta_{\mathcal{Y}}$ is equal to the domain of $H(y_i)$ in $\Theta_{\mathcal{X}}$.

2. There is a homomorphism from $\mathcal{Y}$ to $\mathcal{X}$, but in this case:

   - $\mathcal{X}$ can 'have loops'. When $H(y_i) = H(y_j)$, then we need for any a in $D_{y_i} = D_{y_j}$, $(a, a) \in E_{\ddot{\mathcal{B}}}$;
   - When $H(y_i) \neq H(y_j)$, then $E_{\mathcal{X}}(H(y_i), H(y_j))$ is an edge but $E_{\ddot{\mathcal{B}}}^{y_i y_j}$ is weaker or equivalent to $E_{\ddot{\mathcal{A}}}^{H(y_i), H(y_j)}$ (in our case it is always a richer relation of the same arity, more edges between D_{y_i}, D_{y_j} in $\Theta_{\mathcal{Y}}$ than between $D_{H(y_i)}, D_{H(y_j)}$) in $\Theta_{\mathcal{X}}$.

3. If a variable y appears in both $\Theta_{\mathcal{X}}$ and $\Theta_{\mathcal{Y}}$, we just assume that $H(y) = y$.

Then it is obvious that:

1. Any time we replace some constraints with weaker constraints, we get an expanded covering of the original instance: we remove some edges from $\mathcal{X}$ and add some edges to $\ddot{\mathcal{A}}$.

2. Any solution to the original instance can be naturally expanded to a solution to a covering (expanded covering): consider a homomorphism H from $\mathcal{X}$ to $\ddot{\mathcal{A}}$, and a homomorphism H' from $\mathcal{Y}$ to $\mathcal{X}$ and then construct $H \circ H'$ (and it will be a homomorphism from $\mathcal{Y}$ to $\ddot{\mathcal{B}}$).

3. The union (union of all constraints) of two coverings (expanded coverings) is also a covering (expanded covering): consider digraphs $\mathcal{Y}_1 \cup \mathcal{Y}_2$ and $\ddot{\mathcal{B}}_1 \cup \ddot{\mathcal{B}}_2$.

4. A covering (expanded covering) of a covering (expanded covering) is a covering (expanded covering): consider a superposition of homomorphism.

5. Suppose $\Theta_{\mathcal{Y}}$ is a covering (expanded covering) of a 1-consistent instance and $\Theta_{\mathcal{Y}}$ is a tree formula. Then the solution set to $\Theta_{\mathcal{Y}}$ is subdirect (there are no cycles in $\mathcal{Y}$).

The following lemma can be easily proved (see [15]).

Lemma 25 (Lemma 6.1, [15])**.** *Suppose $\Theta_{\mathcal{X}}$ is a cycle-consistent irreducible CSP instance and $\Theta_{\mathcal{Y}}$ is an expanded covering. Then $\Theta_{\mathcal{Y}}$ is cycle-consistent and irreducible.*

3.2.6.3 Relations and properties

A binary relation R is called *critical* if it cannot be represented as an intersection of other binary relations on $D_i \times D_j$ and it has no dummy variables. Since in our list $\Gamma_{\mathcal{A}}$ there is any invariant binary relation on $D_i \times D_j$, we define $Critical_2(R)$ as follows:

$$\begin{gathered} Critical_2(R) \iff \neg Dum_2(R,i) \wedge \neg Dum_2(R,j) \wedge \exists a \in D_i, \exists b \in D_j, \forall k < 2^{l^2}, \\ R \subsetneq \Gamma^2_{\mathcal{A},k} \to (\Gamma^2_{\mathcal{A},k}(a,b) \wedge \neg R(a,b)). \end{gathered} \tag{3.77}$$

For a critical binary relation R, the minimal relation R' such that $R \subsetneq R'$ is called the *cover of* R:

$$Cover_2(R',R) \iff Critical_2(R) \wedge R' = weakening(R). \tag{3.78}$$

Further notions we will consider in connection to both binary and n-ary relations, so we will define them both for constraints and solution sets $\mathscr{R}_\Theta$. We use constant subscripts to highlight the difference between the definitions, but we do not use n in subscripts for higher arity since the definitions do not depend on variable n. For a congruence σ on D_i we say that the ith variable of a unary relation $E \leq D_i$ and a binary relation $R \leq D_i \times D_j$ is *stable* under σ if

$$\begin{gathered} Stable_1(E,\sigma) \iff \forall a,a' \in D_i,\ E(a) \wedge \sigma(a,a') \to E(a'); \\ Stable_2(R,i,\sigma) \iff \forall a,a' \in D_i, \forall b \in D_j, R(a,b) \wedge \sigma(a,a') \to R(a',b). \end{gathered} \tag{3.79}$$

Remark 6. Note that a unary relation E stable under some congruence σ on D is just a union of that congruence blocks, it does not have to be a subuniverse of D. A binary relation R such that both its variables on D are stable under σ is a full relation between some set of congruence blocks on the first variable and some (not necessarily the same) set of congruence blocks on the second variable. A congruence $\sigma \subseteq D \times D$ is stable under itself, in the sense that all elements from one congruence block on the first coordinate are connected with all elements from the same block on the second coordinate.

We say that the ith variable of the solution set $\mathscr{R}_\Theta \leq D_0 \times ... \times D_{n-1}$ is stable under congruence σ on D_i if

$$\begin{gathered} Stable(\mathscr{R}_\Theta,i,\sigma) \iff \forall H,H' \leq \langle n,\langle n,l\rangle\rangle, \forall a_i,a_i' \in D_i, (\sigma(a_i,a_i') \wedge (\forall j \neq i, \\ H(j) = H'(j)) \wedge H(i) = \langle i,a_i\rangle \wedge H'(i) = \langle i,a_i'\rangle \wedge H\ddot{O}M(\mathcal{X},\ddot{\mathcal{A}},H)) \to \\ \to H\ddot{O}M(\mathcal{X},\ddot{\mathcal{A}},H'). \end{gathered} \tag{3.80}$$

Note that this is $\Pi_1^{1,b}$-formula. If every variable of R or $\mathscr{R}_\Theta$ is stable under σ we say that R or $\mathscr{R}_\Theta$ is *stable* under σ and write $Stable(\mathscr{R}_\Theta,\sigma)$. We say that a binary relation $R \leq D_i \times D_j$ *has a parallelogram property* if

$$ParlPr_2(R) \iff \forall a,c \in D_i, \forall b,d \in D_j, R(a,b) \wedge R(c,b) \wedge R(c,d) \to R(a,d). \tag{3.81}$$

A relation has the parallelogram property if any way of grouping its coordinates into two groups gives a binary relation with the parallelogram property. That is, we say that a solution set $\mathscr{R}_\Theta \leq D_0 \times ... \times D_{n-1}$ to some CSP instance Θ has a *parallelogram property* if the following $\Pi_2^{1,b}$-relation holds:

$$\begin{gathered} ParlPr(\mathscr{R}_\Theta) \iff \forall V_1,V_2 < n,\ (\forall i < n,\ V_1(i) \leftrightarrow \neg V_2(i)) \wedge \\ \wedge \forall H_1,H_2,H_3 \leq \langle n,\langle n,l\rangle\rangle, \\ (H\ddot{O}M(\mathcal{X},\ddot{\mathcal{A}},H_1) \wedge H\ddot{O}M(\mathcal{X},\ddot{\mathcal{A}},H_2) \wedge H\ddot{O}M(\mathcal{X},\ddot{\mathcal{A}},H_3) \wedge \\ \wedge(\forall i \in V_1,\ H_3(i) = H_2(i) \wedge \forall j \in V_2,\ H_3(j) = H_1(j))) \to \\ \to \exists H_4 \leq \langle i,\langle i,a\rangle\rangle,\ H\ddot{O}M(\mathcal{X},\ddot{\mathcal{A}},H_4) \wedge \\ \wedge(\forall i \in V_1,\ H_4(i) = H_1(i) \wedge \forall j \in V_2,\ H_4(j) = H_2(j)). \end{gathered} \tag{3.82}$$

For a binary relation $R \leq D_i \times D_j$ by $Con_2^{(R,i)}$ we denote the following relation:

$$\begin{aligned} Con_2^{(R,i)}(a,a') &\iff \exists b \in D_j, R(a,b) \wedge R(a',b) \\ Con_2^{(R,j)}(b,b') &\iff \exists a \in D_i, R(a,b) \wedge R(a,b'). \end{aligned} \tag{3.83}$$

For a constraint $C = (x_i, x_j; R)$ we will denote $Con_2^{(R,i)}$ by $Con_2^{(C,i)}$. For a set of constraints Θ we denote by $Con_2^{(\Theta,i)}$ the set of all $Con_2^{(C,i)}$. In the case of a CSP instance Θ, for any $i < n$ this set is of the form

$$\begin{aligned} \forall j < n, \forall a, b < l,\, Con_2^{(\Theta,i)}(0,j,a,b) &\iff E_{\mathcal{X}}(i,j) \wedge Con_2^{(E_{\mathcal{X}}(i,j),i)}(a,b), \\ \forall j < n, \forall a, b < l,\, Con_2^{(\Theta,i)}(j,0,a,b) &\iff E_{\mathcal{X}}(j,i) \wedge Con_2^{(E_{\mathcal{X}}(j,i),i)}(a,b). \end{aligned} \tag{3.84}$$

and its size is bounded by $\langle n, n, l, l\rangle$. We say that the ith variable of the binary relation R is *rectangular* if

$$\begin{aligned} RectPr_2(R,i) &\iff \forall a, a' \in D_i, \forall b \in D_j, \\ &(Con_2^{(R,i)}(a,a') \wedge R(a,b) \to R(a',b)). \end{aligned} \tag{3.85}$$

For a solution set $\mathscr{R}_\Theta \leq D_0 \times ... \times D_{n-1}$ to some CSP instance Θ by $Con^{([\mathscr{R}_\Theta],i)}$ we define the binary relation

$$\begin{aligned} Con^{([\mathscr{R}_\Theta],i)}(a,a') &\iff \exists H_1, H_2 \leq \langle n, \langle n, l\rangle\rangle,\, H_1(i) = a \wedge H_2(i) = a' \wedge \\ &\wedge \forall j \neq i < n,\, H_1(j) = H_2(j) \wedge H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}}, H_1) \wedge H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}}, H_2). \end{aligned} \tag{3.86}$$

We say that the ith variable of the solution set R is *rectangular* if

$$\begin{aligned} RectPr(\mathscr{R}_\Theta, i) &\iff \forall a, a' \in D_i,\, \forall H_1 \leq \langle n, \langle n, l\rangle\rangle, \\ H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}}, H_1) \wedge H_1(i) = \langle i, a\rangle \wedge Con_n^{([\mathscr{R}_\Theta],i)}(a,a') &\to \exists H_2 \leq \langle n, \langle n, l\rangle\rangle, \\ H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}}, H_2) \wedge H_2(i) = \langle i, a'\rangle \wedge (\forall j \neq i < n,\, H_1(j) &= H_2(j)). \end{aligned} \tag{3.87}$$

Finally, we say that the solution set $\mathscr{R}_\Theta$ to Θ is *rectangular* if all its variables are rectangular:

$$RectInst(\mathscr{R}_\Theta) \iff \forall i < n,\, RectPr(\mathscr{R}_\Theta, i). \tag{3.88}$$

Note that $RectPr(\mathscr{R}_\Theta, i)$ is $\Sigma_2^{1,b}$-formula.

Remark 7. Note that the parallelogram property implies rectangularity, and if ith coordinate of the relation R is rectangular, then $Con^{([\mathscr{R}_\Theta],i)}$ is a congruence.

A binary relation $R \leq D_i \times D_j$ is called *essential* if it cannot be represented as a conjunction of relations with smaller arities. A pair $(a_i, a_j) \in D_i \times D_j$ is called *essential for* R if

$$\begin{aligned} EssPair(a_i, a_j, R) &\iff \neg R(a_i, a_j) \wedge \exists b_i \in D_i, \exists b_j \in D_j, \\ &R(a_i, b_j) \wedge R(b_i, a_j). \end{aligned} \tag{3.89}$$

It is known [14] that for a relation R being an essential is equivalent to having an essential pair. Thus, we can define an essential binary relation R as follows:

$$EssRel_2(R) \iff \exists a_i \in D_i, \exists a_j \in D_j,\, EssPair(a_i, a_j, R). \tag{3.90}$$

For a solution set $\mathscr{R}_\Theta$ we define an essential tuple by the following $\Sigma_1^{1,b}$-formula:

$$\begin{aligned} EssTuple(H, \mathscr{R}_\Theta) &\iff \neg H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}}, H) \wedge \forall i < n, \exists b < l, \exists H' \leq \langle n, \langle n, l\rangle\rangle, \\ &H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}}, H') \wedge H'(i) = \langle i, b\rangle \wedge \forall j \neq i < n,\, H'(j) = H(j). \end{aligned} \tag{3.91}$$

Thus, $\mathscr{R}_\Theta$ is essential if there exists an essential tuple.

$$EssRel(\mathscr{R}_\Theta) \iff \exists H \leq \langle n, \langle n, l\rangle\rangle,\ EssTuple(H, \mathscr{R}_\Theta). \tag{3.92}$$

We say that a relation $\mathscr{R} \leq D_0 \times ... \times D_{n-1}$ is $(C_0, ..., C_{n-1})$-essential if $\mathscr{R} \cap (C_0, ..., C_{n-1}) = \emptyset$, but for every $i \leq k$, $\mathscr{R} \cap (C_0, ..., C_{i-1}, A_i, C_{i+1}, ..., C_{n-1}) \neq \emptyset$. We can formalize the tuple $(C_0, ..., C_{n-1})$ as usual, by one set $C(i, a) \iff C_i(a)$.

$$\begin{gathered} EssRel(\mathscr{R}, C) \iff \neg(\exists H \in \mathscr{R},\ \forall i < n, \exists c_i \in C_i,\ H(i) = \langle i, c_i\rangle) \wedge \\ \forall i < n, \exists a_i \in D_i \backslash C_i, \exists H \in \mathscr{R},\ H(i) = \langle i, a_i\rangle \wedge \\ \wedge \forall j \neq i, j < n,\ \exists c_j \in C_j,\ H(j) = \langle j, c_j\rangle. \end{gathered} \tag{3.93}$$

This is $\Sigma_0^{\mathscr{B}}$-formula, but if we restrict ourselves to solution sets, we get a Boolean combination of $\Sigma_1^{1,b}$ and $\Pi_1^{1,b}$ formulas.

Finally, to define a *key relation*, we first present a *unary vector function that preserves the relation*. Suppose $R \leq A_0 \times ... \times A_{s-1}$ and define a tuple $\Psi = (\psi_0, ..., \psi_{s-1})$, where $\psi_i : A_i \to A_i$, is called a unary-vector function. We say that ψ preserves R if $(\psi_0(a_0), ..., \psi_{s-1}(a_{s-1})) \in R$ for every $(a_0, ..., a_{s-1}) \in R$. We say that R is a *key relation* if there exists a tuple $(b_0, ..., b_{s-1}) \notin R$ such that for any tuple $(c_0, ..., c_{s-1}) \notin R$ there exists a vector function Ψ which preserves R and gives $\psi_i(c_i) = b_i$ for any $i < s$. For a binary relation $R \leq D_i \times D_j$ there is a pair of unary functions ψ_i, ψ_j, represented by two-dimensional sets, such that:

$$\begin{gathered} VecFun_2(R, \psi_i, \psi_j) \iff MAP(D_i, l, D_i, l, \psi_i) \wedge MAP(D_j, l, D_j, l, \psi_j) \wedge \\ \wedge \forall a_i, b_i \in D_i, \forall a_j, b_j \in D_j,\ R(a_i, a_j) \wedge \psi_i(a_i, b_i) \wedge \psi_j(a_j, b_j) \to R(b_i, b_j). \end{gathered} \tag{3.94}$$

Obviously, both ψ_i, ψ_j are polymorphisms. We say that a binary relation R is a *key relation* if there exists a tuple $(b_i, b_j) \notin R$ such that for every $(c_i, c_j) \notin R$ there exists a unary vector function represented by sets ψ_i, ψ_j that preserves R and $\psi(c_i, b_i)$ and $\psi_j(c_j, b_j)$:

$$\begin{gathered} KeyRel_2(R) \iff \exists b_i, b_j < l, \forall c_i, c_j < l \\ \neg R(b_i, b_j) \wedge \neg R(c_i, c_j) \to \bigvee_{\psi_i, \psi_j < l^2} VecFun_2(R, \psi_i, \psi_j) \wedge \psi_i(c_i, b_i) \wedge \psi_j(c_j, b_j). \end{gathered} \tag{3.95}$$

Note that already for binary relations it would be $\Sigma_1^{1,b}$-formula if we do not fix the algebra $\mathbb{A}$. But in our case, we can go through all possible endomorphisms on D_i, D_j. For a solution set $\mathscr{R}_\Theta \leq D_0 \times ... \times D_{n-1}$ we can represent a unary vector function as a three-dimensional set $\Psi(i, a, b)$, where each Ψ_i represents a function from D_i to D_i. Consider the following $\Pi_1^{1,b}$-formula:

$$\begin{gathered} VecFun(\mathscr{R}_\Theta, \Psi) \iff \forall i < n,\ MAP(D_i, l, D_i, l, \Psi_i) \wedge \\ \wedge \forall H, H' \leq \langle n, \langle n, l\rangle\rangle, H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}}, H) \wedge \\ \wedge (\forall i < n, \forall a_i, b_i \in D_i,\ H(i) = \langle i, a_i\rangle \wedge \Psi_i(a_i, b_i) \to H'(i) = \langle i, b_i\rangle) \to \\ \to H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}}, H'). \end{gathered} \tag{3.96}$$

Then for a solution set $\mathscr{R}_\Theta \leq D_0 \times ... \times D_{n-1}$ we have the following $\Sigma_4^{1,b}$-formula:

$$\begin{gathered} KeyRel(\mathscr{R}_\Theta) \iff \exists H \leq \langle n, \langle n, l\rangle\rangle, \forall H' \leq \langle n, \langle n, l\rangle\rangle,\ \neg H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}}, H) \wedge \\ \wedge \neg H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}}, H') \to \exists \Psi \leq \langle n, l, l\rangle,\ VecFun(\mathscr{R}_\Theta, \Psi) \wedge \\ \wedge (\forall i < n, \forall a_i, b_i < l,\ H'(i) = \langle i, a_i\rangle \wedge \Psi_i(a_i, b_i) \to H(i) = \langle i, b_i\rangle). \end{gathered} \tag{3.97}$$

3.2.6.4 Bridges and connectivity

Definition 61 (Irreducible congruence)**.** We say that a congruence σ on D_i is *irreducible* if it is proper, and it cannot be represented as an intersection of other binary relations stable under σ.

$$\begin{gathered} irCong_m(\sigma, D_i) \iff pCong_m(D_i, \Omega, \sigma) \wedge \exists a, b \in D_i, \forall j < 2^{l^2}, \\ \sigma \subsetneq \Gamma^2_{\mathcal{D},j} \wedge Stable_2(\Gamma^2_{\mathcal{D},j}, \sigma) \to (\Gamma^2_{\mathcal{D},j}(a, b) \wedge \neg\sigma(a, b)). \end{gathered} \tag{3.98}$$

We denote the set of all irreducible congruences on D by $\Sigma^{ir}_{\mathcal{D}}$. For an irreducible congruence σ on set D by σ^* is denoted the minimal binary relation $\sigma \subsetneq \sigma^*$ stable under σ. We can define a string function

$$\begin{gathered} {}^{\cdot^*}(\sigma)(a, b) = \sigma^*(a, b) \iff \bigvee_{\sigma' \leq \langle l,l \rangle} Stable_2(\sigma', \sigma) \wedge \sigma \subsetneq \sigma' \wedge \forall j < 2^{l^2}, \\ Stable_2(\Gamma^2_{\mathcal{D},j}, \sigma) \wedge \sigma \subsetneq \Gamma^2_{\mathcal{D},j} \to \sigma' \subseteq \Gamma^2_{\mathcal{D},j} \wedge \sigma(a, b). \end{gathered} \tag{3.99}$$

Remark 8. Any congruence σ' containing σ is stable under σ, but a binary relation stable under σ does not need to be a congruence.

Definition 62 (Bridge)**.** For two domains D_i, D_j and congruences on them σ_i, σ_j respectively, we say that a 4-ary relation $\rho \subseteq D_i^2 \times D_j^2$ is a *bridge from* σ_i *to* σ_j if the first two variables of ρ are stable under σ_1 and the last two variables of ρ are stable under σ_2, $\sigma_1 \subsetneq pr_{1,2}(\rho)$ and $\sigma_2 \subsetneq pr_{3,4}(\rho)$, and $(a_1, a_2, a_3, a_4) \in \rho$ implies

$$(a_1, a_2) \in \sigma_1 \iff (a_3, a_4) \in \sigma_2.$$

We can define it by $\Sigma^{1,b}_0$-formula:

$$\begin{gathered} Bridge(\rho, \sigma_i, \sigma_j) \iff (\exists a, a' \in D_i, \exists b, b' \in D_j, \\ pr_{1,2}(\rho)(a, a') \wedge \neg\sigma_i(a, a') \wedge pr_{3,4}(\rho)(b, b') \wedge \neg\sigma_j(b, b')) \wedge \\ \wedge Stable_2(\rho, 1, \sigma_i) \wedge Stable_2(\rho, 2, \sigma_i) \wedge Stable_2(\rho, 3, \sigma_j) \wedge Stable_2(\rho, 4, \sigma_j) \wedge \\ \wedge(\forall a, a' \in D_i, \forall b, b' \in D_j,\ \rho(a, a', b, b') \to (\sigma_i(a, a') \leftrightarrow \sigma_j(b, b'))). \end{gathered} \tag{3.100}$$

In words, the projection of ρ to the first two coordinates strictly contains σ_i and is a full relation between some set of congruence blocks on the first coordinate and some set of blocks on the second coordinate, and the same for the projection of ρ to the last two coordinates, and the first two coordinates contain elements from one congruence block of σ_i if and only if the last two coordinates also contain elements from one congruence block of σ_j. A bridge $\rho \subseteq D^4$ is called *reflexive* if $(a, a, a, a) \in \rho$ for every $a \in D$. For a bridge ρ, denote by $\tilde{\rho}$ the binary relation defined by $\tilde{\rho}(x, y) = \rho(x, x, y, y)$, we define it as a string function:

$$\tilde{}(\rho)(x, y) = \tilde{\rho}(x, y) \iff \rho(x, x, y, y). \tag{3.101}$$

A reflexive bridge ρ from an irreducible congruence σ_i to an irreducible congruence σ_j is called *optimal* if there is no a reflexive bridge ρ' from σ_i to σ_j such that $\tilde{\rho} \subsetneq \tilde{\rho}'$, i.e. a bridge that contains more congruence blocks than ρ.

$$\begin{gathered} OptBridge(\rho, \sigma_i, \sigma_j) \iff irCong_m(\sigma_i, D) \wedge irCong_m(\sigma_j, D) \wedge \\ \wedge\neg(\bigvee_{\rho' \leq (4l)^{2^4}} Bridge(\rho', \sigma_i, \sigma_j) \wedge \forall a \in D,\ \rho'(a, a, a, a) \wedge \tilde{\rho} \subsetneq \tilde{\rho}'). \end{gathered} \tag{3.102}$$

If ρ is optimal, then $\tilde{\rho}$ is a congruence. For an irreducible congruence σ, define a string function *opt* as

$$opt(\sigma)(x,y) \iff \bigvee_{\rho \le (4l)^{2^4}} OptBridge(\rho,\sigma,\sigma) \wedge \tilde{\rho}(x,y). \tag{3.103}$$

It returns the congruence $\tilde{\rho}$ for an optimal bridge ρ from σ to σ, which is well-defined since we can compose two reflexive bridges. For a set of irreducible congruences $\Sigma^{ir}_{\mathcal{D}}$, we define a string function *optset* that returns the set of $opt(\sigma)$ for all $\sigma \in \Sigma^{ir}_{\mathcal{D}}$:

$$optset(\Sigma^{ir}_{\mathcal{D}})(i,a,b) \iff \Sigma^{ir}_{\mathcal{D},i} \neq \emptyset \wedge opt(\Sigma^{ir}_{\mathcal{D},i})(a,b). \tag{3.104}$$

We say that two congruences σ_i, σ_j on a set D are *adjacent* if there exists a reflexive bridge from σ_i to σ_j. Since we consider only finite and fixed set of binary constraints $\Gamma^2_{\mathcal{A}}$, including the set of all congruences on A and all its subuniverses, we know in advance all bridges for all congruences, the list denoted by Ξ:

$$Adj(\sigma_i,\sigma_j) \iff \bigvee_{\rho \in \Xi} Bridge(\rho,\sigma_i,\sigma_j) \wedge \forall a \in D,\, \rho(a,a,a,a). \tag{3.105}$$

Note that $Adj(\sigma_i,\sigma_j)$ is $\Sigma^{1,b}_0$-formula. We say that two rectangular constraints C_1, C_2 are *adjacent* in a common variable x if $Con_2^{(C_1,x)}$ and $Con_2^{(C_2,x)}$ are adjacent. A formula is called *connected* if every constraint in the formula is critical and rectangular, and the graph, whose vertices are constraints and edges are adjacent constraints, is connected. To define connectivity, recall 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 $H(t) = j$:

$$\begin{aligned} Path(i,j,\mathcal{X}) \iff \exists t < n, \exists V_{\mathcal{P}_t} = t, \exists E_{\mathcal{P}_t} \le 4t^2,\, PATH(V_{\mathcal{P}_t}, E_{\mathcal{P}_t}) \wedge \\ \wedge \exists H \le \langle t,n \rangle, HOM(\mathcal{P}_t,\mathcal{X},H) \wedge (H(0,i) \wedge H(t,j)). \end{aligned} \tag{3.106}$$

We short it as $\exists \mathcal{P}_t < \langle n, 4n^2 \rangle, Path(i,j,\mathcal{X},\mathcal{P}_t)$. For an instance Θ, we define the following $\Sigma^{1,b}_1$-relation of being connected:

$$\begin{aligned} Connected(\Theta) \iff \forall i,j < n,\, E_{\mathcal{X}}(i,j) \to Critical_2(E^{ij}_{\ddot{\mathcal{A}}}) \wedge \\ \wedge RectPr_2(E^{ij}_{\ddot{\mathcal{A}}},i) \wedge RectPr_2(E^{ij}_{\ddot{\mathcal{A}}},j) \wedge \\ \wedge \forall i,j,k,s < n,\, E_{\mathcal{X}}(i,j) \wedge E_{\mathcal{X}}(k,s) \to \exists \mathcal{P}_t < \langle n, 4n^2 \rangle, \exists H \le \langle t,n \rangle,\, PATH(V_{\mathcal{P}_t}) \wedge \\ \wedge HOM(\mathcal{P}_t,\mathcal{X},H) \wedge (H(0,i) \wedge H(t,s)) \wedge \\ \wedge \forall p \le t-2,\, Adj(Con_2^{(E_{\mathcal{X}}(H(p),H(p+1)),H(p+1))}, Con_2^{(E_{\mathcal{X}}(H(p+1),H(p+2)),H(p+1))}), \end{aligned} \tag{3.107}$$

where by $E_{\mathcal{X}}(H(p),H(p+1)))$ and $E_{\mathcal{X}}(H(p+1),H(p+2))$ we abbreviate all four combinations of non-symmetric constraints.

3.2.7 One-of-four subuniverses

In this section we will define 4 different subuniverses for an algebra $\mathbb{D} = (D,\Omega)$. For D being a subuniverse for a fixed algebra $\mathbb{A} = (A,\Omega)$, all these definitions are $\Sigma^{1,b}_0$-formulas.

3.2.7.1 Binary absorption subuniverse

If $\mathbb{B} = (B, F_B)$ is a subalgebra of $\mathbb{D} = (D, F_D)$, then B *absorbs* $\mathbb{D}$ if there exists an n-ary term operation $f \in Clone(F_D)$ 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* D if there exists a binary term operation $f \in Clone(F_D)$ such that $f(a, b) \in B$ and $f(b, a) \in B$ for any $a \in D$ and $b \in B$. Consider the algebra $\mathbb{A} = (A, \Omega)$ and its subalgebra $\mathbb{B} = (B, \Omega)$, where Ω is m-ary basic operation. 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:

$$T \in Clone(\Omega) \iff Pol_2(T, A, \Gamma_{\mathcal{A}}). \tag{3.108}$$

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

$$\begin{gathered} BAsubS(B, D, T) \iff subS(B, D) \wedge \forall a \in D, \forall b \in B, \exists c_1, c_2 \in B, \\ T(a, b) = c_1 \wedge T(b, a) = c_2. \end{gathered} \tag{3.109}$$

If we want to define a subuniverse, then

$$\begin{gathered} BAsubU(B, D, T, \Omega) \iff SwNU(\Omega, B) \wedge Pol_2(T, D, \Gamma_{\mathcal{A}}) \wedge \\ \wedge BAsubS(B, D, T). \end{gathered} \tag{3.110}$$

Recall that a binary absorbing subuniverse can be trivial, i.e. $B = D$.

3.2.7.2 Central subuniverse

A subuniverse C of D is called *central* if it is an absorbing subuniverse and for every $a \in D \backslash C$ we have $(a, a) \notin Sg(\{a\} \times C \cup C \times \{a\})$. Every central subuniverse is a ternary absorbing subuniverse.

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

$$\begin{gathered} Cl(X) = X \cup \{\Omega(a_1, ..., a_m) : a_1, ..., a_m \in X\} \\ \forall t \geq 0, Cl^0(X) = X, Cl^{t+1}(X) = Cl(Cl^t(X)). \end{gathered} \tag{3.111}$$

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 Cl because at any application we either add to the set at least one element or, after some t, $Cl^t(X) = Cl^{t+r}(X)$ 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 $Cl_X^{l^2}$ up to l^2:

$$\begin{gathered} \forall b, c < l,\ Cl_X^0(b, c) \iff X(b, c) \wedge \\ \wedge \forall 0 < t < l^2, \forall b, c < l,\ Cl_X^t(b, c) \iff Cl_X^{t-1}(b, c) \vee \\ \vee \exists b_1, ..., b_m, c_1, ..., c_m \in A, Cl_X^{t-1}(b_1, c_1) \wedge ... \wedge Cl_X^{t-1}(b_m, c_m) \wedge \\ \wedge \Omega(b_1, ..., b_m) = b \wedge \Omega(c_1, ..., c_m) = c. \end{gathered} \tag{3.112}$$

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 [16]. For any three sets

D, C, S the following $\Sigma_0^{1,b}$-definable relation (D and C are bounded by l) expresses that the subset C of D is central under ternary term operation S:

$$\begin{aligned} CRsubS(C,D,S) \iff & subS(C,D) \wedge \forall c_1, c_2 \in C, \forall a \in D, \exists c_1', c_2', c_3' \in C, \\ & S(c_1,c_2,a) = c_1' \wedge S(c_1,a,c_2) = c_2' \wedge S(a,c_1,c_2) = c_3' \wedge \\ \wedge & \bigwedge_{a \in D \setminus C} \bigwedge_{X < \langle l,l \rangle} ((X(a,c) \wedge X(c,a) \leftrightarrow c \in C) \rightarrow \neg Cl_X^{l^2}(a,a)). \end{aligned} \tag{3.113}$$

Note that for not fixed algebra $\mathbb{B} = (B, \Omega)$, this relation is $\Pi_1^{1,b}$ since the size of B would not be bounded, and therefore we could not use large conjunction. If we want to define a subuniverse, then

$$\begin{aligned} CRsubU(C,D,S,\Omega) \iff & SwNU(\Omega, C) \wedge Pol_2(S, D, \Gamma_{\mathcal{A}}) \wedge \\ & \wedge CRsubS(C,D,S). \end{aligned} \tag{3.114}$$

Recall that a central subuniverse can be trivial, i.e. $C = D$.

3.2.7.3 PC subuniverse

We call an algebra $\mathbb{D} = (D, F_D)$ *polynomially complete* (PC) if its polynomial clone is the clone of all operations on D, $O(D)$. Polynomially complete algebras are necessarily simple [7], i.e. they have no non-trivial congruence relations. A classical result on 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 29 gives a necessary and sufficient condition of polynomial completeness.

Theorem 29 ([2])**.** *A finite algebra is polynomially complete if and only if it has the ternary discriminator as a polynomial operation.*

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

$$Polynom(\mathbb{D}) = Clone(\Omega, a_1, ..., a_{|D|}). \tag{3.115}$$

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 constant operations, any unary relation has to contain the entire set D, and any binary relation has to contain the diagonal relation Δ_D. For the algebra $\mathbb{D} = (D, \Omega)$ denote by $\Gamma_{\mathcal{D}}^{diag} = (\Gamma_{\mathcal{D}}^{1,diag}, \Gamma_{\mathcal{D}}^{2,diag})$ the pair of sets such that

$$\begin{aligned} \Gamma_{\mathcal{D}}^{1,diag}(j,a) \iff & \Gamma_{\mathcal{D}}^{1}(j,a) \wedge (\forall b \in A, \Gamma_{\mathcal{D}}^{1}(j,b)) \\ \Gamma_{\mathcal{D}}^{2,diag}(i,a,b) \iff & \Gamma_{\mathcal{D}}^{2}(i,a,b) \wedge (\forall c \in A, \Gamma_{\mathcal{D}}^{2}(j,c,c)). \end{aligned} \tag{3.116}$$

Note that for some i, j, $\Gamma_{\mathcal{D},j}^{1,diag}$ and $\Gamma_{\mathcal{D},i}^{2,diag}$ are empty sets. An n-ary operation P on algebra $\mathbb{D}$ is a polynomial operation if it is a polymorphism for relations from $\Gamma_{\mathcal{D}}^{diag}$, i.e.

$$P \in Polynom(\mathbb{D}) \iff Pol_n(P, D, \Gamma_{\mathcal{D}}^{diag}). \tag{3.117}$$

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

$$\begin{gathered} PCD(D,P) \iff \forall a,b,c \in D, \\ (a=b \wedge P(a,b,c)=c) \vee (a \neq b \wedge P(a,b,c)=a). \end{gathered} \tag{3.118}$$

To formalize a PC subuniverse we need the following definition.

Definition 63 (Polynomially complete algebra)**.** We say that an algebra $\mathbb{D}=(D,\Omega)$ is polynomially complete if

$$PCA(D,\Omega) \iff \bigvee_{P \in \Pi^3_{\mathcal{D}}} Pol_3(P,D,\Gamma^{diag}_{\mathcal{D}}) \wedge PCD(D,P). \tag{3.119}$$

Definition 64 (Polynomially complete algebra without a non-trivial binary absorbing or central subuniverse)**.** We say that an algebra $\mathbb{D}=(D,\Omega)$ is an algebra *without a non-trivial binary absorbing or central subuniverse* if it satisfies the following $\Sigma_0^{1,b}$-definable relation:

$$\begin{gathered} subTA_{\neg BACR}(D,\Omega) \iff subTA(D,A,\Omega) \wedge \\ \wedge \bigwedge_{B<l} \bigwedge_{T<(3l)^{2^3}} PsubS(B) \wedge Pol_2(T,D,\Gamma_{\mathcal{D}}) \to \neg BAsubU(B,D,T) \wedge \\ \wedge \bigwedge_{C<l} \bigwedge_{S<(4l)^{2^4}} PsubS(C) \wedge Pol_3(S,D,\Gamma_{\mathcal{D}}) \to \neg CRsubU(C,D,S). \end{gathered} \tag{3.120}$$

Note that for a not fixed algebra $\mathbb{B}=(B,\Omega)$, this relation would be $\Pi_2^{1,b}$ since the size of B would not be bounded, and therefore we could not use big conjunctions for sets and $\neg CRsubU(C,D,S)$ would be $\Sigma_1^{1,b}$-formula. We say that an algebra is polynomially complete algebra without a non-trivial binary absorbing or central subuniverse if

$$PCA_{\neg BACR}(D,\Omega) \iff PCA(D,\Omega) \wedge subTA_{\neg BACR}(D,\Omega). \tag{3.121}$$

Definition 65 (PC congruence)**.** We say that a set $\sigma < \langle l,l \rangle$ is a PC congruence on algebra $\mathbb{D}=(D,\Omega)$ of size bounded by l if

$$PCCong_m(D,\Omega,\sigma) \iff PCA_{\neg BACR}(D/\sigma,\Omega/\sigma). \tag{3.122}$$

Note that in this definition we apply notions from (3.116) to (3.121) to algebra $(D/\sigma$, $\Omega/\sigma)$ and relations from $\Gamma_{\mathcal{D}}/\sigma$, recall (3.36).

Recall that for algebra $\mathbb{A}=(A,\Omega)$ we denoted the set of all congruences on $\mathbb{A}$ and all its subuniverses by $\Sigma_{\mathcal{A}}$. Using this list we can define the list of congruence on $\mathbb{A}$ and all its subuniverses of any type, for example:

$$\begin{aligned} \Sigma^{max}_{\mathcal{A}}(i,a,b) &\iff \Sigma_{\mathcal{A}}(i,a,b) \wedge maxCong_m(A,\Omega,\Sigma_{\mathcal{A},i}); \\ \Sigma^{PC}_{\mathcal{D}}(i,a,b) &\iff \Sigma_{\mathcal{D}}(i,a,b) \wedge PCCong_m(D,\Omega,\Sigma_{\mathcal{D},i}). \end{aligned} \tag{3.123}$$

In these definitions we do not enumerate elements in the lists from the beginning, we thin out the existing lists $\Sigma_{\mathcal{A}}$, $\Sigma_{\mathcal{D}}$. That is, for some $i < 2^{l^2}$ the new lists can be empty. Then we say that σ is an intersection of all PC congruences on $\mathbb{D}$ if it satisfies the following $\Sigma_0^{1,b}$ relation:

$$\begin{gathered} CongPC(D,\Omega,\sigma) \iff Cong_m(D,\Omega,\sigma) \wedge (\forall i < 2^{l^2}, \Sigma^{PC}_{\mathcal{D},i} \neq \emptyset \to \sigma \subseteq \Sigma^{PC}_{\mathcal{D},i}) \wedge \\ \wedge \bigwedge_{\sigma' < \langle l,l \rangle} (Cong_m(D,\Omega,\sigma') \wedge \sigma \subsetneq \sigma') \to \exists j < 2^{l^2}, \Sigma^{PC}_{\mathcal{D},j} \neq \emptyset, \exists a,b \in D, \\ \sigma'(a,b) \wedge \neg \Sigma^{PC}_{\mathcal{D},j}(a,b). \end{gathered} \tag{3.124}$$

A subuniverse $E \subseteq D$ is called a *PC subuniverse* if $E = E_0 \cap ... \cap E_{s-1}$ where each E_i is an equivalence class of some PC congruence.

Definition 66 (PC subuniverse, I definition)**.** For an algebra $\mathbb{D} = (D, \Omega)$, E is a PC subuniverse if

$$\begin{aligned} PCsubU(E, D, \Omega) \iff & (E = \emptyset) \vee (E = D) \vee \\ & \vee ((\exists j < 2^{l^2}, \forall a, b \in E, \Sigma_{\mathcal{D}}^{PC}(j, a, b)) \wedge \\ & \wedge (\exists i < 2^{l^2}, \forall a \in E, \forall b \in D, \Sigma_{\mathcal{D}}(i, a, b) \leftrightarrow b \in E)). \end{aligned} \tag{3.125}$$

A PC subuniverse can be empty or full ($E = D$). The condition in the second line ensures that the entire E is inside exactly one block of any number of PC congruences (since we do not restrict the number of different j's in any way) and the condition in the third line ensures that E is indeed a block of some congruence (not necessarily PC congruence since due to the maximality, intersection of any number of PC congruences is not a PC congruence).

We give a second definition of a PC subuniverse in this section straightaway. Lemma 7.13.1 is proved in [15].

Lemma 26 (Lemma 7.13.1, [15])**.** *Suppose that $\sigma_1, ..., \sigma_k$ are all PC congruences on A. Put $A_i = A/\sigma_i$, and define $\psi : A \to A_1 \times ... \times A_k$ by $\psi(a) = (a/\sigma_1, ..., a/\sigma_k)$. Then*

1. *ψ is surjective, hence $A/\cap_i \sigma_i \cong A_1 \times ... \times A_k$;*
2. *the PC subuniverses are the sets of the form $\psi^{-1}(S)$ where $S \subseteq A_1 \times ... \times A_k$ is a relation definable by unary constraints of the form $x_i = a_i$;*
3. *for each non-empty PC subuniverse B of A there is a congruence θ of A such that B is an equivalence class of θ and A/θ is isomorphic to a product of PC algebras having no non-trivial binary absorbing subuniverse or center. That is, $A/\theta \cong A_{j_1} \times ... \times A_{j_s}$ where each A_{j_i} is a PC algebra that has no non-trivial binary absorbing subuniverse or center.*

Since for a fixed algebra $\mathbb{A} = (A, \Omega)$ and all its subalgebras D we know the list of all PC congruences, we do not need to prove this lemma, the algorithm can just check it. Then, since the algebra $\mathbb{A}$ and all its subalgebras are bounded by size l, the maximal possible number of quotients in the direct product $D_0 \times ... \times D_{k-1}$ is $s = log_2 l$.

Definition 67 (PC subuniverse, II definition)**.** For an algebra $\mathbb{D} = (D, \Omega)$, $s = log_2 l$, E is a PC subuniverse if

$$\begin{aligned} PCsubU(E, D, \Omega) \iff & (E = \emptyset) \vee (E = D) \vee \big((\exists i < 2^{l^2}, \forall a \in E, \forall b \in D, \\ & \Sigma_{\mathcal{D}}(i, a, b) \leftrightarrow b \in E) \wedge \\ \wedge & \bigvee_{(\sigma_0 \in \Sigma_D^{PC})} \bigvee_{(H \in \mathrm{M}_{D, \Sigma_{D,i}, \sigma_0})} ISO_{alg}(D/\Sigma_{D,i}, \Omega/\Sigma_{D,i}, D/\sigma_0, \Omega/\sigma_0, H) \vee \\ \vee & \bigvee_{(\sigma_0, \sigma_1 \in \Sigma_D^{PC})} \bigvee_{(H \in \mathrm{M}_{D, \Sigma_{D,i}, \sigma_0, \sigma_1})} ISO_{alg}(D/\Sigma_{D,i}, \Omega/\Sigma_{D,i}, D/\sigma_0 \times D/\sigma_1, \\ & \Omega/\sigma_0 \cap \sigma_1, H) \vee ... \\ ... \vee & \bigvee_{(\sigma_0, ..., \sigma_{s-1} \in \Sigma_D^{PC})} \bigvee_{(H \in \mathrm{M}_{D, \Sigma_{D,i}, \sigma_0, ..., \sigma_{s-1}})} ISO_{alg}(D/\Sigma_{D,i}, \Omega/\Sigma_{D,i}, D/\sigma_0 \times ... \\ & ... \times D/\sigma_{s-1}, \Omega/\cap_i \sigma_i, H)). \end{aligned} \tag{3.126}$$

3.2.7.4 Linear subuniverse

An idempotent finite algebra $\mathbb{D} = (D, \Omega)$, where Ω is an m-ary idempotent special WNU operation, is called *linear* if it is isomorphic to $(\mathbb{Z}_{p_1} \times ... \times \mathbb{Z}_{p_k}, x_1 + ... + x_m)$ for prime (not necessarily distinct) numbers $p_1, ..., p_k$. 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.

Since the algebra $\mathbb{A}$ and all its subalgebras are bounded by size l, the maximal possible number of prime fields in the prime product $\mathbb{Z}_{p_0} \times ... \times \mathbb{Z}_{p_{k-1}}$ is $s = log_2 l$. We formalize a finite abelian group $\mathbb{Z}_p = (Z_p, 0, +, -)$, where p is prime, as a set Z_p, $|Z_p| = p \wedge \forall a < p, Z_p(a)$, and $+_{(mod\, p)}$, $-_{(mod\, p)}$. For any direct product up to $k \leq log_2 l$ abelian groups $\mathbb{Z}_{p_0} \times ... \times \mathbb{Z}_{p_{k-1}}$ we define a set $Z_{p_0} \times ... \times Z_{p_{k-1}} \leq (p_0 + ... + p_{k-1} + 1)^{2^k}$ such that for all $a_0 < p_0, ..., a_{k-1} < p_{k-1}$, $(a_0, ..., a_{k-1}) \in Z_{p_0} \times ... \times Z_{p_{k-1}}$ and

$$\begin{aligned} &\forall a_0, b_0 < p_0, ..., \forall a_{k-1}, b_{k-1} < p_{k-1}, \\ &+((a_0, ..., a_{k-1}), (b_0, ..., b_{k-1})) = (a_0 +_{(mod\, p_0)} b_0, ..., a_{k-1} +_{(mod\, p_{k-1})} b_{k-1}), \\ &-((a_0, ..., a_{k-1}), (b_0, ..., b_{k-1})) = (a_0 -_{(mod\, p_0)} b_1, ..., a_{k-1} -_{(mod\, p_{k-1})} b_{k-1}). \end{aligned} \tag{3.127}$$

We will denote elements $(a_0, ..., a_{k-1})$ of $Z_{p_0} \times ... \times Z_{p_{k-1}}$ by $\bar{a}^k$, and will omit index $_{(mod\, p)}$ when it does not lead to confusion. Also, we allow the use of trivial algebras (with one element 0) in a product, so $Prime'(p) \iff Prime(p) \vee p = 1$.

Definition 68 (Linear algebra of size at most $|A|$)**.** For an algebra $\mathbb{D} = (D, \Omega)$, $s = log_2 l$, we say that it is a *linear algebra* if

$$\begin{aligned} LinA(D, \Omega) \iff &\bigvee_{\substack{p_0 \leq l \\ Prime'(p_0)}} \bigvee_{H \in \mathrm{M}_{D, p_0}} ISO_{alg}(D, \Omega, Z_{p_0}, \bar{a}_1^1 + ... + \bar{a}_m^1, H) \vee \\ \vee &\bigvee_{\substack{p_0 \cdot p_1 \leq l \\ Prime'(p_0), Prime'(p_1)}} \bigvee_{H \in \mathrm{M}_{D, p_0, p_1}} ISO_{alg}(D, \Omega, Z_{p_0} \times Z_{p_1}, \bar{a}_1^2 + ... + \bar{a}_m^2, H) \vee ... \\ ... \vee &\bigvee_{\substack{p_0 \cdot ... \cdot p_{s-1} \leq l \\ Prime'(p_0), ..., Prime'(p_k) \\ s = log_2 l}} \bigvee_{H \in \mathrm{M}_{D, p_0, ..., p_{s-1}}} ISO_{alg}(D, \Omega, Z_{p_0} \times ... \times Z_{p_{s-1}}, \\ &\bar{a}_1^s + ... + \bar{a}_m^s, H). \end{aligned} \tag{3.128}$$

Definition 69 (Linear congruence)**.** We say that a set $\sigma < \langle l, l \rangle$ is a *linear congruence* on algebra $\mathbb{D} = (D, \Omega)$ if

$$LinCong_m(D, \Omega, \sigma) \iff LinA(D/\sigma, \Omega/\sigma). \tag{3.129}$$

We can also check that any linear congruence of algebra $\mathbb{A}$ (or its subalgebras) bounded by size l is a linear congruence for any subalgebra of $\mathbb{A}$ (or their subalgebras). Let us define the set of all linear congruences on $\mathbb{D}$ as:

$$\Sigma_{\mathcal{D}}^{lin}(i, a, b) \iff \Sigma_{\mathcal{D}}(i, a, b) \wedge LinCong_m(D, \Omega, \Sigma_{\mathcal{D}, i}). \tag{3.130}$$

Then we say that σ is *the* minimal linear congruence (an intersection of all linear congruences) on D if

$$CongLin(D, \Omega, \sigma) \iff \exists i < 2^{l^2}, \sigma = \Sigma_{\mathcal{D}, i}^{lin} \wedge \forall j < 2^{l^2}, \Sigma_{\mathcal{D}, j}^{lin} \neq \emptyset \to \sigma \subseteq \Sigma_{\mathcal{D}, j}^{lin}. \tag{3.131}$$

Note that the definition of $CongLin$ differs from the definition of $CongPC$ since any intersection of linear congruences is again a linear congruence. A subuniverse $L \subseteq D$ is called a *linear subuniverse* if it is stable under $CongLin$:

$$\begin{aligned} LNsubU(L,D,\Omega) \iff SwNU(\Omega,L) \wedge \bigwedge_{\sigma \leq \langle l,l \rangle} CongLin(D,\Omega,\sigma) \to \\ \to Stable_1(L,\sigma). \end{aligned} \tag{3.132}$$

Remark 9. A linear subuniverse is a union of classes of $CongLin$. However, not every union of such classes needs to be a subuniverse. For example, for a linear algebra (D,Ω) that is isomorphic to $(\mathbb{Z}_p, x_1 + ... + x_m)$, and a minimal linear congruence Δ every element of $\mathbb{Z}_p$ is a subuniverse (since Ω is idempotent), but not any other proper subset of $\mathbb{Z}_p$ is a subuniverse. From here, we get that there are no non-trivial congruences on (D,Ω) (every congruence block must be a subuniverse).

3.2.7.5 One-of-four and minimal subuniverse

All the following formulas in this section are $\Sigma_0^{1,b}$ (they would not if $\mathbb{A}$ is not fixed). We say that B is *one-of-four* subuniverse of D if

$$\begin{aligned} 1of4subU(B,D,\Omega) \iff PCsubU(B,D,\Omega) \vee LNsubU(B,D,\Omega) \vee \\ \vee \bigvee_{T<(3l)^8} BAsubU(B,D,T,\Omega) \vee \bigvee_{S<(4l)^{16}} CRsubU(B,D,S,\Omega). \end{aligned} \tag{3.133}$$

We say that a subuniverse is minimal if it is non-trivial and inclusion minimal (does not contain any other subuniverse of the same type). For example,

$$\begin{aligned} minBAsubU(B,D,T,\Omega) \iff BAsubU(B,D,T,\Omega) \wedge \bigwedge_{B'<l} \bigwedge_{T'<(3l)^8} B' \subsetneq B \to \\ \to \neg BAsubU(B',D,T',\Omega); \\ minLNsubU(B,D,\Omega) \iff LNsubU(B,D,\Omega) \wedge \bigwedge_{B'<l} B' \subsetneq B \to \\ \to \neg LNsubU(B',D,\Omega). \end{aligned} \tag{3.134}$$

For linear and PC subuniverses we also will use the fact that a minimal linear/ PC subuniverse is a block of $CongLin$/ $CongPC$. We denote a block B of a congruence σ as

$$Block(B,D,\sigma) \iff \forall a \in B, \forall b \in D, \sigma(a,b) \leftrightarrow b \in B. \tag{3.135}$$

Then

$$\begin{aligned} minPCsubU^B(B,D,\Omega) \iff \bigwedge_{\sigma<(2l)^4} CongPC(D,\Omega,\sigma) \to \\ \to Block(B,D,\sigma); \\ minLNsubU^B(B,D,\Omega) \iff \bigwedge_{\sigma<(2l)^4} CongLin(D,\Omega,\sigma) \to \\ \to Block(B,D,\sigma). \end{aligned} \tag{3.136}$$

3.2.8 Reductions

Note that all further definitions for all types of reductions and strategies are $\Sigma_0^{1,b}$-formulas.

Definition 70 (Different types of reductions)**.** For an instance $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$ with domain set $D = (D_0, ..., D_{n-1})$ we say that a set $D^{(\perp)} = (D_0^{(\perp)}, ..., D_{n-1}^{(\perp)})$ is an absorbing reduction of D if there exists a term operation T such that $D_i^{(\perp)}$ is a binary absorbing subuniverse of D_i with the term operation T for every i:

$$BARed(D^{(\perp)}, D) \iff Red(D^{(\perp)}, D) \wedge \bigvee_{T<(3l)^8} Pol_2(T, D, \Gamma_{\mathcal{D}}) \wedge \\ \wedge \forall i < n,\, BAsubU(D_i^{(\perp)}, D_i, T, \Omega). \tag{3.137}$$

We say that $D^{(\perp)} = (D_0^{(\perp)}, ..., D_{n-1}^{(\perp)})$ is a central reduction if D_i is a central subuniverse for every i:

$$CRRed(D^{(\perp)}, D) \iff Red(D^{(\perp)}, D) \wedge \forall i < n, \\ \bigvee_{S_i<(4l)^{16}} Pol_3(S_i, D, \Gamma_{\mathcal{D}}) \wedge CRsub(D_i^{(\perp)}, D_i, S_i). \tag{3.138}$$

We say that $D^{(\perp)} = (D_0^{(\perp)}, ..., D_{n-1}^{(\perp)})$ is a PC reduction if

$$PCRed(D^{(\perp)}, D) \iff Red(D^{(\perp)}, D) \wedge \forall i < n, \\ PCsubU(D_i^{(\perp)}, D_i, \Omega) \wedge subTA_{\neg BACR}(D_i, \Omega). \tag{3.139}$$

We say that $D^{(\perp)} = (D_0^{(\perp)}, ..., D_{n-1}^{(\perp)})$ is a linear reduction if

$$LNRed(D^{(\perp)}, D) \iff Red(D^{(\perp)}, D) \wedge \forall i < n, \\ LNsubU(D_i^{(\perp)}, D_i, \Omega) \wedge subTA_{\neg BACR}(D_i, \Omega). \tag{3.140}$$

In an obvious way, we can define a minimal absorbing/ central/ PC/ linear reduction, a non-linear reduction $nonLNRed(D^{(\perp)}, D)$ and one-of-four reduction $1of4Red(D^{(\perp)}, D)$.

Remark 10. A CSP instance $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$ is a set

$$\Theta(\langle \underbrace{\langle n, \langle n, n\rangle\rangle}_{\mathcal{X}}, \underbrace{\langle\langle n, l\rangle, \langle\langle n, l\rangle, \langle n, l\rangle\rangle\rangle}_{\ddot{\mathcal{A}}} \rangle)$$

Let us denote the length of Θ as a number function $instsize(n, l) = |\Theta|$.

Definition 71 (A strategy for a CSP instance)**.** A *strategy* for a CSP instance $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$ with domain set D is a sequence of reductions $D^{(0)}, ..., D^{(s)}$, where $D^{(j)} = (D_0^{(j)}, ..., D_{n-1}^{(j)})$, such that $D^{(0)} = D$ and $D^{(j)}$ is a one-of-four 1-consistent reduction of $\Theta^{(j-1)}$ for every $j \geq 1$. A strategy is called *minimal* if every reduction in the sequence is minimal.

Since after any reduction we decrease at least one domain by at least one element, to represent the entire strategy it is enough to consider a set (matrix) with nl rows, each row representing a reduction of the CSP instance. We need to track both domain reductions (the set $V_{\ddot{\mathcal{A}}}$, or D) and restrictions of the constraint relations (the set $E_{\ddot{\mathcal{A}}}$). An input digraph $\mathcal{X}$ does not change, but for consistency, we will track it as well. Thus, a strategy for an instance Θ up to $s \leq nl$ step is a set $\Theta_{Str} < \langle nl, instsize(n, l)\rangle$ such that:

$$Strategy(\Theta, \Theta_{Str}, s) \iff \Theta_{Str}^0 = \Theta \wedge \forall 1 \leq j \leq s, 1C(\Theta_{Str}^{(j)}) \wedge \\ \wedge redinst(\Theta^{(j-1)}, D^{(j)}) = \Theta_{Str}^{(j)} \wedge \forall 1 \leq j \leq s,\, 1of4Red(D_{Str}^{(j)}, D). \tag{3.141}$$

A strategy is called minimal if

$$\begin{aligned} minStrategy(\Theta, \Theta_{Str}, s) \iff Strategy(\Theta, \Theta_{Str}, s) \wedge \forall 1 \leq j \leq s, \\ min1of4Red(D^{(j)}_{Str}, D). \end{aligned} \tag{3.142}$$

When we want to consider the domain strategy separately, we will refer to it as $D_{Str} < \langle nl, \langle n, l \rangle \rangle$, each jth row representing $D^{(j)}$.

3.2.9 Three universal algebra axiom schemes

We are now ready to recall formally the three universal algebra axiom schemes from Chapter 2 (ref. [6]). These schemes are formulated for any constraint language $\Gamma_{\mathcal{A}}$ over the set A of size l, fixed algebra $\mathbb{A} = (A, \Omega)$ with Ω being a special m-ary WNU operation. They consist of finitely many $\forall\Sigma_2^{1,b}$-formulas (for all possible subuniverses of $\mathbb{A}$). The relations $CCInst(\mathcal{X}, \ddot{\mathcal{A}})$ and $IRDInst(\mathcal{X}, \ddot{\mathcal{A}})$ are $\Pi_2^{1,b}$, corresponding to $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$ being a cycle-consistent and irreducible instance, respectively, see Chapter 2 (ref. [6]).

$\mathrm{BA}_{\mathcal{A}}$-axiom scheme reflects that if Θ is a cycle-consistent irreducible CSP instance, and B is a non-trivial binary absorbing subuniverse of D_i, then Θ has a solution only if Θ has a solution with $x_i \in B$ (Theorem 5.5 in [15]):

$$\begin{aligned} BA_{\mathcal{A},B,D} =_{def} \forall \mathcal{X} = (V_{\mathcal{X}}, E_{\mathcal{X}}), \forall \ddot{\mathcal{A}} = (V_{\ddot{\mathcal{A}}}, E_{\ddot{\mathcal{A}}}, D_0, ..., D_{n-1}), \\ \big(B \subsetneq D \wedge SwNU_m(\Omega, D) \wedge SwNU_m(\Omega, B) \wedge \\ \wedge \exists T < (3l)^{2^3}, Pol_2(T, D, \Gamma_{\mathcal{A}}) \wedge BAsubS(B, D, T) \wedge \\ \wedge Inst(\Theta, \Gamma_{\mathcal{A}}) \wedge CCInst(\mathcal{X}, \ddot{\mathcal{A}}) \wedge IRDInst(\mathcal{X}, \ddot{\mathcal{A}}) \wedge \\ \exists i < n, D_i = D \wedge \\ H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}})) \implies H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}} = (V_{\ddot{\mathcal{A}}}, E_{\ddot{\mathcal{A}}}, D_0, ..., B, ..., D_{n-1})). \end{aligned} \tag{3.143}$$

$\mathrm{CR}_{\mathcal{A}}$-axiom scheme states that if Θ is a cycle-consistent irreducible CSP instance, and C is a non-trivial central subuniverse of D_i, then Θ has a solution only if Θ has a solution with $x_i \in C$ (Theorem 5.5 in [15]):

$$\begin{aligned} CR_{\mathcal{A},D,C} =_{def} \forall \mathcal{X} = (V_{\mathcal{X}}, E_{\mathcal{X}}), \forall \ddot{\mathcal{A}} = (V_{\ddot{\mathcal{A}}}, E_{\ddot{\mathcal{A}}}, D_0, ..., D_{n-1}), \\ \big(C \subsetneq D \wedge SwNU_m(\Omega, D) \wedge SwNU_m(\Omega, C) \wedge \\ \exists S < (4l)^{2^4}, Pol_3(S, D, \Gamma_{\mathcal{A}}) \wedge CRsubS(C, D, S) \wedge \\ \wedge Inst(\Theta, \Gamma_{\mathcal{A}}) \wedge CCInst(\mathcal{X}, \ddot{\mathcal{A}}) \wedge IRDInst(\mathcal{X}, \ddot{\mathcal{A}}) \wedge \\ \exists i < n, D_i = D \wedge \\ H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}})) \implies H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}} = (V_{\ddot{\mathcal{A}}}, E_{\ddot{\mathcal{A}}}, D_0, ..., C, ..., D_{n-1})). \end{aligned} \tag{3.144}$$

Finally, $\mathrm{PC}_{\mathcal{A}}$-axioms says that if Θ is a cycle-consistent irreducible CSP instance, there does not exist a non-trivial binary absorbing subuniverse or a non-trivial 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 only if Θ has a solution with $x_i \in E$ (Theorem 5.6 in [15]):

$$\begin{gathered}
PC_{\mathcal{A},D,E} =_{def} \forall \mathcal{X} = (V_{\mathcal{X}}, E_{\mathcal{X}}), \forall \ddot{\mathcal{A}} = (V_{\ddot{\mathcal{A}}}, E_{\ddot{\mathcal{A}}}, D_0, ..., D_{n-1}), \\
([\forall j < n, \forall B < l, \forall T < (3l)^{2^3}, Pol_2(T, D_j, \Gamma_{\mathcal{A}}) \to \neg BAsubS(B, D_j, T) \wedge \\
\wedge \forall j < n, \forall C < l, \forall S < (4l)^{2^4}, Pol_3(S, D_j, \Gamma_{\mathcal{A}}) \to \neg CRsubS(C, D_j, S)] \\
\wedge \exists \sigma < \langle l, l \rangle, \exists D/\sigma < l, \exists \Omega/\sigma < (ml)^{2^{m+1}}, FA_m(D/\sigma, \Omega/\sigma, D, \Omega, \sigma) \wedge \\
\wedge \exists P < (4l)^{2^4},\ Pol_3(P, D/\sigma, \Gamma_{\mathcal{D}}^{diag}/\sigma) \wedge PCD(D/\sigma, P) \wedge \\
SwNU_m(\Omega, D) \wedge E \subsetneq D \wedge (\forall a \in E, \forall b \in D, \sigma(a, b) \leftrightarrow b \in E) \wedge \\
\wedge Inst(\Theta, \Gamma_{\mathcal{A}}) \wedge CCInst(\mathcal{X}, \ddot{\mathcal{A}}) \wedge IRDInst(\mathcal{X}, \ddot{\mathcal{A}}) \wedge \\
\exists i < n, D_i = D \wedge \\
H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}})) \implies H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}} = (V_{\ddot{\mathcal{A}}}, E_{\ddot{\mathcal{A}}}, D_0, ..., E, ..., D_{n-1})).
\end{gathered} \tag{3.145}$$

3.3 Formalization of proofs of the three axiom schemes

In this section we do not differentiate between a solution set and its projection to any subset of coordinates since the proofs do not differ.

3.3.1 Formalization of some auxiliary lemmas and theorems

We are going to present the formalization of a number of selected statements and their proofs used in the proof of the soundness of the algorithm. We selected those that genuinely represent various types of arguments encountered in [15], [16]. It should be sufficiently clear that other statements of a similar nature can be formalized analogously.

3.3.1.1 Properties of a binary absorbing subuniverse on $\mathbb{A}^n$

We say that a solution set to a CSP instance Θ over $\Gamma_{\mathcal{A}}$ on n variables, $\mathscr{R}_\Theta \leq D_0 \times ... \times D_{n-1}$ is a binary absorbing subuniverse of $D_0 \times ... \times D_{n-1}$ if there exists a binary term operation $T \in Pol(\Gamma_{\mathcal{A}})$ such that for any two maps $H, H' : [n] \to (D_0, ..., D_{n-1})$ where $H \notin \mathscr{R}_\Theta$ and $H' \in \mathscr{R}_\Theta$, the maps $usepol_2(T, H, H')$ and $usepol_2(T, H', H)$ are in $\mathscr{R}_\Theta$. An analogous definition can be formulated for any projection of the set of solutions $\mathscr{R}_\Theta^{i_1,...,i_s}$.

Lemma 27 (Lemma 7.1, [15])**.** *Suppose that $\mathscr{R}_\Theta$ is defined by a pp-formula $\Theta(x_0, ..., x_{n-1})$ and Θ' is obtained from Θ by replacing some constraint relations $\rho_1, ..., \rho_s$ by constraint relations $\rho'_1, ..., \rho'_s$ such that ρ'_k absorbs ρ_k with a term operation T for every k. Then V^1 proves that the relation $\mathscr{R}_{\Theta'}$ defined by $\Theta'(x_0, ..., x_{n-1})$ absorbs $\mathscr{R}_\Theta$ with the term operation T.*

Proof. Let us consider two CSP instances $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$ and $\Theta' = (\mathcal{X}', \ddot{\mathcal{A}}')$, where $\mathcal{X}' = \mathcal{X}$ (the analogous reasoning can be applied to projections). Suppose that there exists a binary term $T \in Pol(\Gamma_{\mathcal{A}})$ such that for each $i < n$, $D'_i \subseteq D_i$ binary absorbs D_i and for all $i, j < n$ with $E_{\mathcal{X}}(i, j)$, $E^{ij}_{\ddot{\mathcal{A}}'} \subseteq E^{ij}_{\ddot{\mathcal{A}}}$ binary absorbs $E^{ij}_{\ddot{\mathcal{A}}}$:

$$\begin{gathered}
\forall a \in D'_i, \forall b \in D_i, \exists c_1, c_2 \in D'_i,\ T(a, b) = c_1 \wedge T(b, a) = c_2 \wedge \\
\forall a_1, b_1 < l, \forall a_2, b_2 < l,\ (E^{ij}_{\ddot{\mathcal{A}}}(a_1, b_1) \wedge E^{ij}_{\ddot{\mathcal{A}}'}(a_2, b_2)) \to \\
\to \exists a_3, a_4 < l, \exists b_3, b_4 < l,\ E^{ij}_{\ddot{\mathcal{A}}'}(a_3, b_3) \wedge E^{ij}_{\ddot{\mathcal{A}}'}(a_4, b_4) \wedge \\
\wedge T(a_1, a_2) = a_3 \wedge T(a_2, a_1) = a_4 \wedge T(b_1, b_2) = b_3 \wedge T(b_2, b_1) = b_4.
\end{gathered} \tag{3.146}$$

Note that for some $i, j < n$, D'_i and $E^{ij}_{\ddot{\mathcal{A}}'}$ could be equal to D_i and $E^{ij}_{\ddot{\mathcal{A}}}$. If $\mathscr{R}_{\Theta'}$ or/ and $\mathscr{R}_{\Theta}$ are empty, we are done ($\mathscr{R}_{\Theta'}$ is an empty subuniverse). Suppose that both instances have solutions. Every solution to the instance Θ' is a solution to the instance Θ. Consider any two solutions to Θ and Θ', homomorphisms H and H' respectively. Consider two maps $H_1 = usepol_2(T, H, H')$ and $H_2 = usepol_2(T, H', H)$. We need to prove that these maps are homomorphisms from $\mathcal{X}'$ to $\ddot{\mathcal{A}}'$. Suppose that H_1 (or H_2) is not a homomorphism. Then there exists an edge in $\mathcal{X}'$, $E_{\mathcal{X}'}(i, j)$ such that H_1 fails to map it to an edge in $\ddot{\mathcal{A}}'$. But this contradicts with (3.146). □

Corollary 2 (Corollary 6.1.3, [16]). *Suppose $\mathscr{R}_{\Theta} \leq D_0 \times ... \times D_{n-1}$ and B_i is an absorbing subuniverse in A_i with a term T for every i. Then V^1 proves that $(B_0 \times ... \times B_{n-1}) \cap \mathscr{R}_{\Theta}$ is an absorbing subuniverse of $\mathscr{R}_{\Theta}$ with the term T.*

Corollary 3 (Corollary 7.1.2, [15]). *Suppose that $\mathscr{R}_{\Theta} \leq D_0 \times ... \times D_{n-1}$ is a relation such that $pr_0(\mathscr{R}_{\Theta}) = D_0$ and $B = pr_0((B_0 \times ... \times B_{n-1}) \cap \mathscr{R}_{\Theta})$, where B_i is an absorbing subuniverse in D_i with a term T for every i. Then V^1 proves that B is an absorbing subuniverse in D_0 with the term T.*

Proof. Consider $\mathscr{R}_{\Theta}$ as a solution set to some CSP instance $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$. Since every B_i is an absorbing subuniverse of D_i, $(B_0 \times ... \times B_{n-1}) \cap \mathscr{R}_{\Theta}$ is a solution set for an instance Θ' defined similarly to (3.146) with a domain set $D' = \{B_0, ..., B_{n-1}\}$ (for all $i, j < n$, $E^{ij}_{\ddot{\mathcal{A}}'} = E^{ij}_{\ddot{\mathcal{A}}}$). Then from $pr_0(\mathscr{R}_{\Theta}) = D_0$ it follows that B is an absorbing subuniverse of D_0. □

Lemma 28 (Lemma 7.3, [15]). *Suppose that $\mathscr{R}_{\Theta}$ is a non-trivial absorbing subuniverse of $D_0 \times ... \times D_{n-1}$. Then V^2 proves that for some i there exists a non-trivial absorbing subuniverse B_i in D_i with the same term.*

Proof. The lemma is proved by induction on the arity of $\mathscr{R}_{\Theta}$. $\mathscr{R}_{\Theta}$ is a non-trivial binary absorbing subuniverse of $D_0 \times ... \times D_{n-1}$ if there exists a solution to Θ_{null} that is not a solution to Θ, and there exists a binary term T such that for any two homomorphisms H from $\mathcal{X}$ to $\ddot{\mathcal{A}}$ and H_{null} from $\mathcal{X}_{null}$ to $\ddot{\mathcal{A}}_{null}$, both maps $usepol_2(T, H, H_{null})$ and $usepol_2(T, H_{null}, H)$ are homomorphisms from $\mathcal{X}$ to $\ddot{\mathcal{A}}$.

Consider the following $\mathfrak{B}(\Sigma^{1,b}_1)$-formula $\phi(t)$. Here, as fixed parameters, we use $\Gamma_{\mathcal{A}}$ and $\mathbb{A} = (A, \Omega)$. Induction goes on the size of the instance, $V_{\mathcal{X}} = V_{\mathcal{X}_{null}} = t$. Witnesses ($\forall$ quantification) in $\Sigma^{1,b}_2$-induction corresponding to t are sets $E_{\mathcal{X}}, E_{\mathcal{X}_{null}}$, $\ddot{\mathcal{A}}$ with the set of vertices of size $\langle t, l \rangle$.

$$\begin{aligned}
\phi(t) := (V_{\mathcal{X}} = V_{\mathcal{X}_{null}} = t \wedge DG(\Theta) \wedge DG_{null}(\Theta_{nul}) \wedge \\
\wedge Inst(\Theta, \Gamma_{\mathcal{A}}) \wedge Inst(\Theta_{null}, \Gamma_{\mathcal{A}})) \wedge \\
\wedge (\bigvee_{T \in \Pi^2_{\mathcal{A}}} \forall H, H_{null} \leq \langle t, \langle t, l \rangle \rangle,\ H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}}, H) \wedge H\ddot{O}M(\mathcal{X}_{null}, \ddot{\mathcal{A}}_{null}, H_{null}) \rightarrow \\
\rightarrow H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}}, usepol_2(T, H, H_{null})) \wedge H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}}, usepol_2(T, H_{null}, H)) \\
\wedge (\exists H' \leq \langle t, \langle t, l \rangle \rangle,\ H\ddot{O}M(\mathcal{X}_{null}, \ddot{\mathcal{A}}_{null}, H') \wedge \neg H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}}, H')) \implies \\
\implies \exists i < t, \bigvee_{B < l} B_i \subsetneq A_i \wedge BAsubU(B_i, A_i, T, \Omega).
\end{aligned} \tag{3.147}$$

The expression in the first brackets says that Θ and Θ_{null} are CSP instances over $\Gamma_{\mathcal{A}}$ on t variables, and Θ_{null} is an empty instance. This is obviously true for $t = 1$. Suppose that it is true for $t = s$ and consider $t = s + 1$. If the projection of $\mathscr{R}_{\Theta}$ on $s + 1$ coordinate

is not D_{s+1}, then $\mathscr{R}_{\Theta}^{s+1}$ is a binary absorbing subuniverse due to the definition of a non-trivial absorbing subuniverse. Otherwise, choose any element $a \in D_{s+1}$ such that $\mathscr{R}_{\Theta}$ does not contain all homomorphisms sending $s+1$ to a and consider the new s-ary relation $\mathscr{R}_{\Theta'} = \{(a_0, ..., a_s) | (a_0, ..., a_s, a) \in \mathscr{R}_{\Theta}\}$. Due to Lemma 27, it is a non-trivial binary absorbing subuniverse for $D_0 \times ... \times D_s$ (T is idempotent). But note that $\mathscr{R}_{\Theta'}$ is also a solution set to a specific CSP instance Θ' on s variables: we just remove from $E_{\mathcal{X}}$ of Θ all edges adjacent to x_{s+1} and for all $j < s+1$ restrict $E_{\ddot{\mathcal{A}}}^{j(s+1)}$ and $E_{\ddot{\mathcal{A}}}^{(s+1)j}$ to pairs ending and starting with a. Thus, we can apply the induction hypothesis. □

The following lemma from [8] gives us a more precise theory.

Lemma 29 ([8])**.** *For all $i \geq 1$, the theory T_2^i proves the induction scheme IND for $\mathcal{B}(\Sigma_i^b)$-formulas.*

Lemma 30 (Lemma 7.5, [15])**.** *Suppose $D^{(1)}$ is an absorbing reduction of a CSP instance Θ and a relation $\mathscr{R}_{\Theta} \leq D_0 \times ... \times D_{n-1}$ is subdirect. Then $\mathscr{R}_{\Theta}^{(1)}$ is not empty.*

Proof. Suppose the opposite. Then $\mathscr{R}_{\Theta}^{(1)} \cap D_0^{(1)} \times ... \times D_{n-1}^{(1)} = \emptyset$, where for each $i < n$, $D_i^{(1)}$ is a binary absorbing subuniverse of D_i with the term T. This means that for every homomorphism $H_{i_1} \in \mathscr{R}_{\Theta}$ there exists $i < n$ such that $H_{i_1}(i) = \langle i, a \rangle$, where $a \in D_i \backslash D_i^{(1)}$. Since $\mathscr{R}_{\Theta}$ is subdirect, for any $i < n$ and for any $b \in D_i^{(1)}$ there exists a homomorphism H_{i_2} such that $H_{i_2}(i) = \langle i, b \rangle$. Composing these homomorphisms, since T is a polymorphism, we again get a homomorphism $H_{i_3} = usepol_2(T, H_{i_1}, H_{i_2})$ such that $H_{i_3}(i) = \langle i, c \rangle$ for some $c \in D_i^{(1)}$. Consider any $j \neq i < n$ such that $H_{i_3}(j) = \langle j, d \rangle$ with $d \notin D_j^{(1)}$. Again, since $\mathscr{R}_{\Theta}$ is subdirect, there must be a homomorphism H_{i_4} such that $H_{i_4}(j) = \langle j, e \rangle$ for some $e \in D_j^{(1)}$. We compose these two homomorphisms and get $H_{i_5} = usepol_2(T, H_{i_3}, H_{i_4})$ such that $H_{i_5}(i) = \langle i, f \rangle$ and $H_{i_5}(j) = \langle j, g \rangle$ with $f \in D_i^{(1)}$ and $g \in D_j^{(1)}$. Applying this procedure at most n times, we will get a homomorphism in $\mathscr{R}_{\Theta}^{(1)}$, contradiction. □

3.3.1.2 Properties of a central subuniverse on $\mathbb{A}^n$

We say that a solution set to a CSP instance Θ over $\Gamma_{\mathcal{A}}$ on n variables, $\mathscr{R}_{\Theta} \leq D_0 \times ... \times D_{n-1}$ is a central absorbing subuniverse of $D_0 \times ... \times D_{n-1}$ if there exists a ternary term operation $S \in Pol(\Gamma_{\mathcal{A}})$ such that $\mathscr{R}_{\Theta}$ absorbs $D_0 \times ... \times D_{n-1}$ with S and for any map $H \notin \mathscr{R}_{\Theta}$ such that for all $i < n$, $H(i) = \langle i, a_i \rangle$, the following conditions hold. If we construct two new CSP instances, Θ_L and Θ_R as follows:

- We double the number of variables, $D_L = D_R = \{D_0, ..., D_{2n-1}\}$ with $D_i = D_{n+i}$ for every $i < n$;
- For the instance Θ_L we copy digraph $\mathcal{X}$ for the first n variables, and we join it with a path $\mathcal{P}_n$ of length n, namely $E_{\mathcal{P}_n}(x_n - 1, x_n), ..., E_{\mathcal{P}_n}(x_{2n-2}, x_{2n-1})$. We copy $\ddot{\mathcal{A}}$ for the first n variables and set the constraints for the path: $E_{\ddot{\mathcal{A}}_n}^{x_n - 1, x_n}$ as a full relation and for the next edges $E_{\ddot{\mathcal{A}}_n}^{x_n, x_{n+1}} = \{(a_0, a_1)\}$, $E_{\ddot{\mathcal{A}}_n}^{x_{n+1}, x_{n+2}} = \{(a_1, a_2)\}$,..., $E_{\ddot{\mathcal{A}}_n}^{x_{2n-2}, x_{2n-1}} = \{(a_{n-2}, a_{n-1})\}$. Note that such a CSP instance is a CSP instance over $\Gamma_{\mathcal{A}}$ since Ω is idempotent and every single element of $D_0 \times ... \times D_{n-1}$ is a subuniverse;
- For the instance Θ_R we do the same but in inversed manner (copy $\mathcal{X}$ and $\ddot{\mathcal{A}}$ for the variables $n, ..., 2n-1$);

Then from solutions to Θ_L and Θ_R by applying Ω we cannot generate the map H such that for $i < n$, $H(i) = H(n+i) = \langle i, a_i \rangle$. Note that to define this fact, we need the third-order induction. The number of maps on $2n$ variables is l^{2n}, so to define the generated algebra $\mathscr{C}_{\mathscr{R}_{\Theta_L} \cup \mathscr{R}_{\Theta_R}}$ it is needed to consider at most l^{2n} steps which we encode by strings:

$$\begin{gathered}
\forall H \leq \langle 2n\langle 2n, l\rangle\rangle,\ \mathscr{C}^{[\emptyset]}_{\mathscr{R}_{\Theta_L} \cup \mathscr{R}_{\Theta_R}}(H) \iff \mathscr{R}_{\Theta_L}(H) \vee \mathscr{R}_{\Theta_R}(H) \wedge \\
\forall \emptyset \leq T < 2n\lceil log_2 l \rceil, \forall H \leq \langle 2n\langle 2n, l\rangle\rangle, \\
\mathscr{C}^{[S(T)]}_{\mathscr{R}_{\Theta_L} \cup \mathscr{R}_{\Theta_R}}(H) \iff \mathscr{C}^{[T]}_{\mathscr{R}_{\Theta_L} \cup \mathscr{R}_{\Theta_R}}(H) \vee \\
\vee \exists H_1, ..., H_m \leq \langle 2n\langle 2n, l\rangle\rangle,\ \mathscr{C}^{[T]}_{\mathscr{R}_{\Theta_L} \cup \mathscr{R}_{\Theta_R}}(H_1) \wedge ... \wedge \mathscr{C}^{[T]}_{\mathscr{R}_{\Theta_L} \cup \mathscr{R}_{\Theta_R}}(H_m) \wedge \\
\wedge \omega(H_1, ..., H_m) = H.
\end{gathered} \tag{3.148}$$

The analogous definition can be formulated for any projection of the solution set $\mathscr{R}^{i_1,...,i_s}_{\Theta}$.

Lemma 31 (Composed Lemma 7.6, [15], and Theorem 6.9, [16])**.** *Suppose $\mathscr{R}_\Theta$ is defined by a pp-formula $\Theta(x_0, ..., x_{n-1})$ and Θ' is obtained from Θ by replacement of some constraint relations $\rho_1, ..., \rho_s$ by constraint relations $\rho'_1, ..., \rho'_s$ such that ρ'_k is a central subuniverse for ρ_k with a term operation S for every k. Then V^1 proves that the relation $\mathscr{R}_{\Theta'}$ defined by $\Theta'(x_0, ..., x_{n-1})$ is a central subuniverse for $\mathscr{R}_\Theta$ with the term operation S.*

Proof. Consider two CSP instances $\Theta = (\mathcal{X}, \ddot{\mathcal{A}})$ and $\Theta' = (\mathcal{X}', \ddot{\mathcal{A}}')$, where $\mathcal{X}' = \mathcal{X}$ (again, the analogous reasoning can be applied to projections). Due to the assumption, there exists a ternary term $S \in Pol(\Gamma_\mathcal{A})$ such that for each $i < n$, $D'_i \subseteq D_i$ ternary absorbs D_i and for all $i, j < n$ with $E_\mathcal{X}(i, j)$, $E^{ij}_{\ddot{\mathcal{A}}'} \subseteq E^{ij}_{\ddot{\mathcal{A}}}$ ternary absorbs $E^{ij}_{\ddot{\mathcal{A}}}$. The defining relation is analogous to (3.146). Also, for each $i < n$, and for any $a \in D_i \backslash D'_i$, $(a, a) \notin Sg(X^i_{(a)})$, where

$$X^i_{(a)} = \{\{a\} \times D'_i, D'_i \times \{a\}\}, \tag{3.149}$$

and for all $i, j < n$ with $E_\mathcal{X}(i, j)$, and for every $(a, b) \in E^{ij}_{\ddot{\mathcal{A}}} \backslash E^{ij}_{\ddot{\mathcal{A}}'}$ we have $(a, b, a, b) \notin Sg(X^{ij}_{(a,b)})$, where

$$X^{ij}_{(a,b)} = \{(a, b)\} \times E^{ij}_{\ddot{\mathcal{A}}'} \cup E^{ij}_{\ddot{\mathcal{A}}'} \times \{(a, b)\}. \tag{3.150}$$

We will show how to define $Sg(X^{ij}_{(a,b)})$ analogously to a central subuniverse we defined before, using the closure operator $Cl(X^{ij}_{(a,b)})$. Since $\mathbb{A}$ is finite of size l and $|X^{ij}_{(a,b)}| = 2|E^{ij}_{\ddot{\mathcal{A}}'}|$, we do not need more than $(l^4 - 2|E^{ij}_{\ddot{\mathcal{A}}'}|)$ steps. Not to depend on $E^{ij}_{\ddot{\mathcal{A}}'}$, choose the value l^4. For set $X^{ij}_{(a,b)} \leq \langle\langle l, l\rangle, \langle l, l\rangle\rangle$, iteratively define the following set $Cl^t_{X^{ij}_{(a,b)}}$ up to l^4:

$$\begin{gathered}
\forall c, d, e, f < l,\ Cl^0_{X^{ij}_{(a,b)}}(c, d, e, f) \iff X^{ij}_{(a,b)}(c, d, e, f) \wedge \\
\wedge \forall 0 < t < l^4, \forall c, d, e, f < l,\ Cl^t_{X^{ij}_{(a,b)}}(c, d, e, f) \iff Cl^{t-1}_{X^{ij}_{(a,b)}}(c, d, e, f) \vee \\
\vee \exists c_1, ..., c_m, d_1, ..., d_m, e_1, ..., e_m, f_1, ..., f_m \in A, \\
Cl^{t-1}_{X^{ij}_{(a,b)}}(c_1, d_1, e_1, f_1) \wedge ... \wedge Cl^{t-1}_{X^{ij}_{(a,b)}}(c_m, d_m, e_m, f_m) \wedge \\
\wedge \Omega(c_1, ..., c_m) = c \wedge \Omega(d_1, ..., d_m) = d \wedge \Omega(e_1, ..., e_m) = e \wedge \Omega(f_1, ..., f_m) = f.
\end{gathered} \tag{3.151}$$

Therefore,

$$\forall i, j < n, \forall a, b < l,\ (E_\mathcal{X}(i, j) \wedge E^{ij}_{\ddot{\mathcal{A}}}(a, b) \wedge \neg E^{ij}_{\ddot{\mathcal{A}}'}(a, b)) \to \neg Cl^{l^4}_{X^{ij}_{(a,b)}}(a, b, a, b). \tag{3.152}$$

That $\mathscr{R}_{\Theta'}$ absorbs $\mathscr{R}_\Theta$ with ternary operation S can be proved as in Lemma 27. Suppose that $\mathscr{R}_{\Theta'}$ is not a central subuniverse of $\mathscr{R}_\Theta$. Then there exists a homomorphism H from $\mathcal{X}$ to $\ddot{\mathcal{A}}$, sending each $i < n$ to $\langle i, a_i\rangle$, such that it is not a homomorphism from $\mathcal{X}'$ to $\ddot{\mathcal{A}}'$, and if we construct two instances Θ'_L and Θ'_R as above, the subalgebra generated by these two instances contains a homomorphism H' sending both i and $n+i$ to $\langle i, a_i\rangle$ for each $i < n$. It would mean that for all $i, j < n$ such that $E_{\mathcal{X}}(i, j)$, elements of the form $(a_i, a_j, a_i, a_j) \in D_i \times D_j \times D_{n+i} \times D_{n+j}$ must belong to sets $Cl^{l^4}_{X^{ij}_{(a_i,a_j)}}$. But at least one (a_i, a_j) must be from $E^{ij}_{\ddot{\mathcal{A}}} \backslash E^{ij}_{\ddot{\mathcal{A}}'}$ (otherwise H is a homomorphism from $\mathcal{X}'$ to $\ddot{\mathcal{A}}'$). That contradicts with (3.152). □

Remark 11. Note that in the above proof, we do not even use the definition of the third-order object. We lowered requirements to second-order objects and showed the contradiction. Thus, we can remain in V^1.

Corollary 4 (Corollary 6.9.2, [16])**.** *Suppose $\mathscr{R}_\Theta \leq D_0 \times ... \times D_{n-1}$ is a relation such that $pr_0(\mathscr{R}_\Theta) = D_0$ and $C = pr_0((C_0 \times ... \times C_{n-1}) \cap \mathscr{R}_\Theta)$, where C_i is a central subuniverse in D_i for every i. Then V^1 proves that C is a central subuniverse in D_0.*

Corollary 5 (Corollary 6.9.3, [16])**.** *Suppose $\mathscr{R}_\Theta \leq D_0 \times ... \times D_{n-1}$ and C_i is a central subuniverse for every i. Then V^1 proves that $(C_0 \times ... \times C_{n-1}) \cap \mathscr{R}_\Theta$ is a central subuniverse for $\mathscr{R}_\Theta$.*

Lemma 32 (Lemma 7.7, [15])**.** *Suppose $\mathscr{R}_\Theta$ is a non-trivial center of $D_0 \times ... \times D_{n-1}$. Then V^2 proves that for some i there exists a non-trivial center C_i of D_i.*

3.3.1.3 Properties of a PC subuniverse on $\mathbb{A}^n$

While considering the properties of a PC subuniverse on $D_0 \times ... \times D_{n-1}$ we keep in mind that we further use all auxiliary lemmas in the proof of the main statements about the next reduction, and all reductions are constructed based on the separate domains upward to subuniverses on their product. That is, there is no need to consider arbitrary PC subuniverses on $D_0 \times ... \times D_{n-1}$ or on the solution set $\mathscr{R}_\Theta$. The problem here is that we have not even a definition of a PC algebra for an arbitrary product or a PC congruence on that product (recall that in the definition of PC algebra, we use fixed constraint language $\Gamma_{\mathcal{A}}$). Thus, any time in the proofs we consider an arbitrary algebra $\mathbb{D}$ and its arbitrary PC congruence σ, we may assume that $\mathbb{D}$ is a subuniverse $\mathscr{R} \leq D_0 \times ... \times D_{n-1}$ and σ is an extended congruence for some PC congruence σ_i on a domain D_i.

Lemma 33 (Lemma 6.20, [16])**.** *Suppose $\mathscr{R}_\Theta$ is a subdirect relation on $D_0 \times ... \times D_{n-1}$ and E_i is a PC subuniverse of D_i for every i. Then W^0_1 proves that $(E_0 \times ... \times E_{n-1}) \cap \mathscr{R}_\Theta$ is a PC subuniverse of $\mathscr{R}_\Theta$.*

Proof. Due to the assumption, there is a PC subuniverse E_i of D_i for every i. If some E_i is empty, then we are done. Otherwise, each E_i is a block of some congruence $\delta_i = \cap_j \sigma_{i_j}$ on D_i, where $\sigma_{i_1}, ..., \sigma_{i_s}$ are PC congruences on D_i. We can extend every σ_{i_j} to the product $D_0 \times ... \times D_{n-1}$, and consider the congruence $\mathscr{C}_{\theta_i} = \cap_{k \neq i} \mathscr{C}_{\nabla^{ext}_{D_k}} \cap \mathscr{C}_{\sigma^{ext}_{ij}}$. Since $\mathscr{R}_\Theta$ is subdirect, it is easy to check that a third-order map $\mathscr{H}$ that sends every $H \in \mathscr{R}_\Theta / \mathscr{C}_{\theta_i}$ to $a \in D_i/\sigma_{ij}$ such that $H(i) = \langle i, a\rangle$, is an isomorphism from $(\mathscr{R}_\Theta / \mathscr{C}_{\theta_i}, \mathscr{F}_{\Omega/\theta_i})$ to $(D_i/\sigma_{ij}, \Omega/\sigma_{ij})$. So $(\mathscr{R}_\Theta / \mathscr{C}_{\theta_i}, \mathscr{F}_{\Omega/\theta_i})$ acts like a PC algebra, and we will call $\mathscr{C}^{\mathscr{R}_\Theta}_{\sigma^{ext}_{i_j}}$ restricted to $\mathscr{R}_\Theta$ an extended PC congruence for $(\mathscr{R}_\Theta, \mathscr{F}_\Omega)$.

It follows that for each E_i, $\mathscr{E}_i^{ext} \cap \mathscr{R}_\Theta$ is an intersection of blocks of extended PC congruences restricted to $(\mathscr{R}_\Theta, \mathscr{F}_\Omega)$, as well as $\mathscr{E}_0^{ext} \cap ... \cap \mathscr{E}_{n-1}^{ext} \cap \mathscr{R}_\Theta$, and we can call it an extended PC subuniverse. By the definition of every $\mathscr{E}_i^{ext}$, it is just $E_0 \times ... \times E_{n-1} \cap \mathscr{R}_\Theta$. □

Note that from the same reasoning, it follows that $E_0 \times ... \times E_{n-1}$ is an extended PC subuniverse of $D_0 \times ... \times D_{n-1}$.

Lemma 34 (Lemma 6.18, [16]). *Suppose that D is a PC algebra and $\mathscr{R}_\Theta \leq D^n$ contains all constant tuples $(a, a, ..., a)$. Then V^0 proves that $\mathscr{R}_\Theta$ can be represented as a conjunction of binary relations of the form $x_i = x_j$.*

Proof. Since $\mathscr{R}_\Theta \leq D^n$ contains all constant tuples $(a, a, ..., a)$, every domain D_i of a CSP instance Θ is equal to D, and every binary constraint $E^{ij}_{\ddot{\mathcal{D}}}$ contains a diagonal relation Δ_{ij}. Recall that an algebra is PC if there exists a ternary discriminator P such that $Pol_3(P, D, \Gamma_{\mathcal{D}}^{diag})$. Therefore, P must preserve $\mathscr{R}_\Theta$. We want to show that every $E^{ij}_{\ddot{\mathcal{D}}}$ that is not a full binary relation is equal to relation Δ_{ij}. Suppose that for some $i, j < n$ $E^{ij}_{\ddot{\mathcal{D}}}$ is neither a full nor diagonal relation. Then there must exist some $a \neq b$ such that $E^{ij}_{\ddot{\mathcal{D}}}(a, b)$. But since P preserves $E^{ij}_{\ddot{\mathcal{D}}}$, $(P(a, a, b), P(a, b, b)) = (b, a) \in E^{ij}_{\ddot{\mathcal{D}}}$, and for every $c \neq b$, $(P(a, b, c), P(a, a, c)) = (a, c) \in E^{ij}_{\ddot{\mathcal{D}}}$. Thus, $E^{ij}_{\ddot{\mathcal{D}}}$ is a full relation. □

The following lemma follows from the previous lemma and some additional reasoning.

Lemma 35 (Lemma 6.19, [16]). *Suppose that $\mathscr{R}_\Theta \leq D_0 \times ... \times D_{n-1}$ is subdirect, D_i is a PC algebra without non-trivial binary absorbing and central subuniverses for every $i \in \{1, ..., n-1\}$, and D_0 has no non-trivial central subuniverse. Then V^1 proves that $\mathscr{R}_\Theta$ can be represented as a conjunction of binary relations $\delta_1(x_{i_1}, x_{j_1}), ..., \delta_s(x_{i_s}, x_{j_s})$, where for every $l \leq s$ the first variable of δ_l is uniquely determined whenever $i_l \neq 1$ and the second variable of δ_l is uniquely determined whenever $j_l \neq 1$.*

Lemma 36 (Lemma 6.21, [16]). *Suppose that $\mathscr{R}_\Theta$ is a subdirect relation on $D_0 \times ... \times D_{n-1}$, E_i is a PC subuniverse of D_i for all i and $E = pr_0((E_0 \times ... \times E_{n-1}) \cap \mathscr{R}_\Theta)$. Then V^1 proves that E_0 is a PC subuniverse of D_0.*

Proof. We consider $\mathscr{R}_\Theta$ to be a solution set to some CSP instance Θ on n domains such that for all $i, j < n$ there is a constraint $E_{\mathcal{X}}(i, j)$, but for some of them $E^{ij}_{\ddot{\mathcal{A}}}$ are full relations. By Lemma 26 (or by Definition 67) each E_i is a block of congruence δ_i such that there are PC congruences $\sigma_{i_0}, ..., \sigma_{i_{k-1}}$ with $k \leq log_2 l$ and

$$D_i/\delta_i \cong D_i/\sigma_{i_0} \times ... \times D_i/\sigma_{i_{k-1}}.$$

Then we can consider the factorized instance Θ_{PC}, constructed as follows. The instance digraph $\mathcal{X}_{PC} = \mathcal{X}$ does not change. For a target digraph $\ddot{\mathcal{A}}_{PC}$, domain set is $D_{PC} = \{D_0, D_1/\delta_1, ..., D_{n-1}/\delta_{n-1}\}$, i.e. we factorize every domain except the first one (or equivalently factorize it by Δ_0). Constraint relations are defined by a set $E_{\ddot{\mathcal{A}}_{PC}}$ such that

$$\begin{aligned}
\forall 0 < i, j < n,\ E^{ij}_{\ddot{\mathcal{A}}_{PC}}(a, b) \iff\ & D_i/\delta_i(a) \wedge D_j/\delta_j(b) \wedge \\
& (\exists c, d < l,\ \delta_i(a, c) \wedge \delta_j(b, d) \wedge E^{ij}_{\ddot{\mathcal{A}}}(c, d)), \\
\forall i < n,\ E^{1i}_{\ddot{\mathcal{A}}_{PC}}(a, b) \iff\ & D_0(a) \wedge D_i/\delta_i(b) \wedge (\exists c < l,\ \delta_i(b, c) \wedge E^{1i}_{\ddot{\mathcal{A}}}(a, c)), \\
\forall i < n,\ E^{i1}_{\ddot{\mathcal{A}}_{PC}}(a, b) \iff\ & D_0(a) \wedge D_i/\delta_i(b) \wedge (\exists c < l,\ \delta_i(b, c) \wedge E^{i1}_{\ddot{\mathcal{A}}}(c, a)).
\end{aligned} \tag{3.153}$$

By Theorem 12 in Chapter 2 (ref. [6]), there is a canonical homomorphism $H_c \leq \langle\langle l, n\rangle, \langle l, n\rangle\rangle$ from the target digraph $\ddot{\mathcal{A}}$ to the target digraph $\ddot{\mathcal{A}}_{PC}$ such that for any homomorphism H from $\mathcal{X}$ to $\ddot{\mathcal{A}}$, the map H_{PC} defined as

$$H_{PC}(i) = \langle i, a\rangle \iff \exists b < l, (H(i) = \langle i, b\rangle \wedge H_c(\langle i, b\rangle) = \langle i, a\rangle),$$

is a homomorphism from $\mathcal{X}$ to $\ddot{\mathcal{A}}_{PC}$. It is easy to see that $pr_0((E_0 \times E_1 \times ... \times E_{n-1}) \cap \mathscr{R}_\Theta) = pr_0((E_0 \times e_1 \times ... \times e_{n-1}) \cap \mathscr{R}_{\Theta_{PC}})$ where for each $0 < i < n$, e_i is the representative of the class E_i, and that $\mathscr{R}_{\Theta_{PC}}$ is subdirect.

For any $0 < i < n$, we can find $M_i \in \mathrm{M}_{D_i, \delta_i \sigma_{i_0}, ..., \sigma_{i_{k-1}}}$ such that

$$ISO_{alg}(D_i/\delta_i, \Omega/\delta_i, D_i/\sigma_{i_0} \times ... \times D_i/\sigma_{i_{k-1}}, \Omega/\cap_j \sigma_{i_j}, M_i).$$

Let us combine all of these maps into one set M. For every i such that $M_i \in \mathrm{M}_{D_i, \delta_i \sigma_{i_0}, ..., \sigma_{i_{k-1}}}$ for $k < s = log_2 l$ we add to the end $s - k$ trivial algebras $D_{i_k}, ..., D_{i_{s-1}}$ (containing one element 0), set $\sigma_{i_k}, ..., \sigma_{i_{s-1}}$ to be trivial congruences and extend M_i to $s+1$-ary set. Then for all $a_0, ..., a_{s-1} < l$,

$$\begin{gathered}\forall 0 < i < n, \forall a \in D_i/\delta_i,\ M(a) = (a_0, ..., a_{s-1}) \iff M_i(a) = (a_0, ..., a_{s-1}) \\ i = 0, \forall a \in D_0,\ M(a) = (a, 0, ..., 0).\end{gathered} \tag{3.154}$$

Consider a CSP instance Θ'_{PC} on ns variables, with domain set

$$D'_{PC} = \{D_0, 0, ..., 0, D_1/\sigma_{1_0}, ..., D_1/\sigma_{1_{s-1}}, ..., D_{n-1}/\sigma_{n-1_0}, ..., D_{n-1}/\sigma_{n-1_{s-1}}\}$$

such that $H'_{PC} \in \mathscr{R}_{\Theta'_{PC}}$ if and only if $H_{PC} \in \mathscr{R}_{\Theta_{PC}}$ where H'_{PC} is a map from $[ns]$ to $[D'_{PC}]$ defined from a homomorphism H_{PC} as follows:

$$\begin{gathered}i = 0,\ H'_{PC}(0) = \langle 0, a\rangle \iff H_{PC}(0) = \langle 0, a\rangle, \\ \forall 0 < i < s,\ H'_{PC}(i) = \langle i, 0\rangle, \\ \forall 0 < j < n, \forall k < s,\ \forall a \in D_j/\sigma_{j_k},\ H'_{PC}(js + k) = \langle js + k, a\rangle \iff \\ \iff \exists b \in D_j/\delta_j, \exists b_0 \in D_j/\sigma_{j_0}, ..., \exists b_{k-1} \in D_j/\sigma_{j_{k-1}}, \\ \exists b_{k+1} \in D_j/\sigma_{j_{k+1}}, ..., \exists b_{s-1} \in D_j/\sigma_{j_{s-1}}, \\ H_{PC}(j) = \langle j, b\rangle \wedge M(b) = (b_0, ..., b_{k-1}, a, b_{k+1}, ..., b_{s-1})\end{gathered} \tag{3.155}$$

Strictly speaking, the instance Θ' is not an instance over language $\Gamma_{\mathcal{A}}$, but we still can define every $2s$-ary relation $R^{ij}_{\ddot{\mathcal{A}}'_{PC}}$,

$$\begin{gathered}R^{ij}_{\ddot{\mathcal{A}}'_{PC}}(a_0, ..., a_{s-1}, b_0, ..., b_{s-1}) \iff \exists a \in D/\delta_i \exists b \in D_j/\delta_j,\ E^{ij}_{\ddot{\mathcal{A}}_{PC}}(a, b) \wedge \\ \wedge M(a) = (a_0, ..., a_{s-1}) \wedge M(b) = (b_0, ..., b_{s-1}).\end{gathered} \tag{3.156}$$

Moreover, we can apply to $R^{ij}_{\ddot{\mathcal{A}}'_{PC}}$ a similar reasoning as in Lemma 34. The solution set $\mathscr{R}_{\Theta'_{PC}}$ is subdirect since $\mathscr{R}_{\Theta_{PC}}$ is subdirect. Since every D_j/σ_{j_k} is a PC algebra, it follows that by Lemma 35, $\mathscr{R}_{\Theta'_{PC}}$ can be represented as a conjunction of binary relations $\delta_1(x_{i_1}, x_{j_1}), ..., \delta_s(x_{i_s}, x_{j_s})$, where for every $l \leq s$ the first variable of δ_l is uniquely-determined whenever $i_l \neq 1$ and the second variable of δ_l is uniquely-determined whenever $j_l \neq 1$. Thus, for all $0 < j < n, k < s$, for every $a \in D_0$ there exists a unique $b \in D_j/\delta_{j_k}$ such that $E^{1(js+k)}_{\ddot{\mathcal{A}}'_{PC}}(a, b)$ (and analogously for the relation $E^{(js+k)1}_{\ddot{\mathcal{A}}'_{PC}}$). It follows that any such relation divides D_0 into $|D_j/\delta_{j_k}|$ classes, and we can check that this is a PC congruence on D_0. Thus, $pr_0((E_0 \times e_1 \times ... \times e_{n-1}) \cap \mathscr{R}_{\Theta_{PC}})$ can be represented as an intersection of blocks of ns PC congruences (some of them can be trivial). □

3.3.1.4 Properties of a linear subuniverse on $\mathbb{A}^n$

It is known that the ability to simulate an affine CSP (or historically the ability to count) adds substantial complexity to the problem. Structures that cannot count are all tractable and can even be solved by a simple constraint propagation algorithm.

The proof complexity of linear algebra (in the sense of a branch of mathematics) was well studied in [12], or in [13]. In particular, Gaussian elimination was considered and was shown to be formalizable in theory V^1. The proof of the following lemma can be easily formalized in V^1, see the detailed proof, for example, in Chapter 2 (ref. [6]).

Notation 13. *For linear algebras, we shall adhere to the following notation. We will continue to denote elements of different domains D_i, D_j by a_i, a_j. If we consider several elements of the same domain, we add an index after the index of the domain, for example, $a_{i1}, ..., a_{im}$. To represent an element a_{ij} as an element of the product of $k \leq log_2 l$ prime fields $Z_{p_0}, ..., Z_{p_{k-1}}$, we add the superscript $\bar{a}^k_{ij} = (a^0_{ij}, ..., a^{k-1}_{ij})$.*

Lemma 37 (Lemma 7.20, [15])**.** *Suppose that the relation $\mathscr{R}_\Theta \leq (\mathbb{Z}_{p_1})^{n_1} \times ... \times (\mathbb{Z}_{p_k})^{n_k}$ 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 V^1 proves that $\mathscr{R}_\Theta = L_1 \times ... \times L_k$, where each L_i is an affine subspace of $(\mathbb{Z}_{p_i})^{n_i}$.*

Proof. Consider any CSP instance Θ on n variables such that each D_i is a linear algebra, i.e. $LinA(D_i, \Omega)$. That is, for every $i < n$ there are some $k \leq log_2 l$, primes $p_0, ..., p_{k-1} < l$, and an isomorphism $M_i \in \mathrm{M}_{A,\Delta_i,p_0,...,p_{k-1}}$ from (D_i, Ω) to $(Z_{p_0} \times ... \times Z_{p_{k-1}}, \bar{a}^k_{i1} + ... + \bar{a}^k_{im})$ such that

$$M_i(a_{ij}) = \bar{a}^k_{ij} = (a^0_{ij}, ..., a^{k-1}_{ij}).$$

As for the PC subuniverses, to unify all M_i, for every $k < s = log_2 l$, we add $s - k$ trivial algebras $\mathbb{Z}_{p_k}, ..., \mathbb{Z}_{p_{s-1}}$, representing their elements as 0's. Thus, $M_i(a_{ij}) = \bar{a}^s_{ij}$. Then we can construct M. For all $a^0_i, ..., a^{s-1}_i < l$,

$$\forall 0 < i < n, \forall a_i \in D_i,\ M(a_i) = (a^0_i, ..., a^{s-1}_i) \iff M_i(a_i) = (a^0_i, ..., a^{s-1}_i). \tag{3.157}$$

We consider CSP instance Θ_L on ns domains such that every solution to Θ, translated naturally (analogously to (3.155)) is a solution to Θ_L and vice versa. Again, it is not a CSP instance over language $\Gamma_{\mathcal{A}}$, but all $2s$-ary relations $R^{ij}_{\ddot{\mathcal{A}}_L}$ can be easily defined; see (3.156). The most important thing is that these relations are preserved by m-ary sum.

For this proof, we do not need to collect equal Z_{p_j} from different domains to a group. We define a vector space on $Z_{p_0} \times ... \times Z_{p_{ns-1}}$ as any subset of maps from $[ns]$ to $[D_L] = [Z_{p_0}, ..., Z_{p_{ns-1}}]$ such that it contains 'zero map' H sending all $i < ns$ to 0, and is closed under $+$, i.e. for any two maps H_1, H_2, the map $H_3 = H_1 + H_2$ such that

$$\begin{gathered}(H_1 + H_2)(i) = H_3(i) = \langle i, a\rangle \iff \exists b, c \in D_i, \\ H_1(i) = \langle i, b\rangle \wedge H_2(i) = \langle i, c\rangle \wedge a = b + c,\end{gathered} \tag{3.158}$$

is also in that set. We define an affine subspace $\mathscr{R}_{\Theta_L}$ of $Z_{p_0} \times ... \times Z_{p_{ns-1}}$ as a shift of some linear subspace, i.e. such a set that for any map $H \in \mathscr{R}_{\Theta_L}$, the set of all maps H' such that $H' + H \in \mathscr{R}_{\Theta_L}$ contains zero map and is closed under $+$. Note that when we are talking about solution sets, it is a second-order definition:

$$\begin{aligned}AffSubS(\mathscr{R}_{\Theta_L}) \iff \forall H \leq \langle n\langle n, l\rangle\rangle, \forall H_1, H_2 \leq \langle n\langle n, l\rangle\rangle,\ H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}}, H) \wedge \\ \wedge H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}}, H + H_1) \wedge H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}}, H + H_2) \rightarrow \\ \rightarrow H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}}, H + (H_1 + H_2)).\end{aligned} \tag{3.159}$$

Then it is easy to show that the solution to Θ_L is an affine subspace defining $x - y + z(mod\, p_j) = \Omega(x, z, y, ..., y)$. □

Remark 12. Recall well-known facts from linear algebra. The number of k-dimensional subspaces of n-dimensional vector space over finite field $\mathbb{Z}_p$ is equal to Gaussian coefficient,

$$\begin{bmatrix} n \\ r \end{bmatrix}_p = \frac{(p^n - 1)(p^n - p)...(p^n - p^{k-1})}{(p^k - 1)(p^k - p)...(p^k - p^{k-1})},$$

and the number of k-dimensional affine subspaces, i.e. in our case k-subuniverses of $(\mathbb{Z}_p^n, x_1 + ...x_m)$, is equal to

$$p^{n-k} \begin{bmatrix} n \\ r \end{bmatrix}_p.$$

Remark 13. Any affine subspace of $(\mathbb{Z}_{p_i})^{n_i}$ is a linear translation of some vector subspace, and any two subspaces of $(\mathbb{Z}_{p_i})^{n_i}$ of the same dimension are isomorphic (elementary p-subgroups). Thus, the only way to get a subgroup (or a quotient) is to 'lose' some $\mathbb{Z}_{p_i}$ from the product.

To prove some auxiliary lemmas and theorems, in addition to Definition 68, we need to define a linear algebra on a product $D_0 \times ... \times D_{n-1}$ of algebras of size at most $|A|$. We could define a linear algebra $\mathscr{R}_\Theta$ on n domains, each of which is isomorphic to a product of $k \leq log_2 l$ prime fields, as a solution set to some CSP instance Θ or as a set of maps closed under $x_1 + ... + x_m$. This defining relation is second-sorted and it would give us that V^1 proves that the set of linear algebras is closed under taking subalgebras. The problem here is that the resulting direct product isomorphic to relation $\mathscr{R}_\Theta$ will not be related to the initial CSP instance Θ, it is considered exclusively as an algebra. We know that any relation preserved by $x_1 + ... + x_m$ can be represented as a system of linear equations over at most ns variables, where $s = log_2 l$, and this system can be solved by Gaussian elimination. Different domains $\mathbb{Z}_{p_i}$ cannot be mixed in one subsystem of equations, but variables over the same $\mathbb{Z}_{p_i}$ representing different x_i, x_j can. Suppose that domains D_i, D_j are both isomorphic to $\mathbb{Z}_3$, and the solution to CSP instance Θ for these two variables contains the affine subspace (coset) $\{(1,1),(2,0),(0,2)\}$. This is isomorphic to $(\mathbb{Z}_3, \Omega)$, but we must lose one domain. The same would happen while taking the quotient and in the proofs of some results about linear algebras on n domains, we need the closeness of a set of linear algebras under taking quotients, especially extended ones. So we cannot avoid an isomorphism between third-order objects. Recall that we allow trivial algebras $\mathbb{Z}_1$ and that solution set to any CSP instance Θ_{null} on n domains is a full relation.

Definition 72 (Linear algebra on a product of algebras of size at most l)**.** For an algebra $(\mathscr{R}, \mathscr{F})$ defined on a set of maps from $[k]$ to $[D'_0, ..., D'_{k-1}]$, where $k \leq n$, we say that $\mathscr{R}$ is a linear algebra on n domains if there exists a set of n domains D and the set of n m-ary operations F on these domains such that every algebra (D_i, F_i) is linear and there is an isomorphism from $(\mathscr{R}, \mathscr{F})$ to an algebra $(\mathscr{D}_0 \times ... \times \mathscr{D}_{n-1}, \mathscr{F}_{F_0,...,F_{n-1}})$, where $\mathscr{D}_0 \times ... \times \mathscr{D}_{n-1}$ and $\mathscr{F}_{F_0,...,F_{n-1}}$ are defined as in (3.22)-(3.23):

$$\begin{aligned} LinA(\mathscr{R}, \mathscr{F}) \iff \exists D \leq \langle n, l \rangle, \exists F \leq \langle n, \underbrace{l, l, ..., l}_{m \text{ times}} \rangle, \forall i < n,\ LinA(D_i, F_i) \wedge \\ \wedge ISO^{3,3}_{alg}(\mathscr{R}, \mathscr{F}, \mathscr{D}_0 \times ... \times \mathscr{D}_{n-1}, \mathscr{F}_{F_0,...,F_{n-1}}). \end{aligned} \tag{3.160}$$

Note that the defining relation is third-order because of the relation $ISO^{3,3}_{alg}$. In an obvious way, we can define a linear and a minimal linear congruence $\mathscr{C}_\theta$ for any algebra on n domains $(\mathscr{R}, \mathscr{F})$.

From Lemma 37, definition of linear algebras and definition of linear algebras on products, we conclude the following corollary.

Corollary 6. *V^0 proves that the set of linear algebras is closed under taking subalgebras and quotients. W_1^1 proves that the set of linear algebras on a product of algebras of size at most $|A|$ is closed under taking subalgebras and quotients.*

In the presence of the third-order definition of linear algebra on n domains, the following lemmas can be proved almost exactly as in [15].

Lemma 38 (Lemma 7.21, [15])**.** *W_1^1 proves that a linear algebra has no non-trivial absorbing subuniverse, non-trivial central subuniverse, or non-trivial PC subuniverse.*

Lemma 39 (Lemma 7.24.1, [15])**.** *Suppose that $\mathscr{R}_\Theta \leq D_0 \times ... \times D_{n-1}$ is a relation such that $pr_0(\mathscr{R}_\Theta) = D_0$, there are no non-trivial binary absorbing subuniverses on D_0, and $L = pr_0((L_1 \times ... \times L_{n-1}) \cap \mathscr{R}_\Theta)$ where L_i is a linear subuniverse of D_i for every $i < n$. Then W_1^1 proves that L is a linear subuniverse of D_0.*

The following lemma can be proved similarly to Lemma 33.

Lemma 40 (Lemma 7.25, [15])**.** *Suppose $\mathscr{R}_\Theta$ is a subdirect relation on $D_0 \times ... \times D_{n-1}$ and L_i is a linear subuniverse of D_i for every i. Then W_1^0 proves that $(L_0 \times ... \times L_{n-1}) \cap \mathscr{R}_\Theta$ is a linear subuniverse of $\mathscr{R}_\Theta$.*

3.3.1.5 Common properties and Interaction between subuniverses

The common property for subuniverses $C_0, ..., C_{n-1}$ of a fixed type (any but linear) is that there does not exist $(C_0, ..., C_{n-1})$-essential relation $\mathscr{R}$ of any arity greater than 2. For PC subuniverses we additionally require the relation to be subdirect.

Lemma 41 (Lemma 7.27, [15])**.** *Suppose C_i is a non-trivial binary absorbing subuniverse of D_i with a term T for all $i \in \{0, 1, 2, ..., n\}$ and $n > 1$. Then V^1 proves that there does not exist a $(C_0, ..., C_{n-1})$-essential solution set $\mathscr{R}_{Theta} \leq D_0 \times ... \times D_{n-1}$.*

Proof. Suppose that such solution set $\mathscr{R}_\Theta$ to some CSP instance over $\Gamma_\mathcal{A}$ exists. Consider two solutions, $H_1 \in (D_0 \times C_1 \times ... \times C_{n-1}) \cap R$ and $H_2 \in (C_0 \times C_1 \times ... \times D_{n-1}) \cap R$. Then $usepol_2(T, H_1, H_2)$ is a new solution to Θ and it is in $C_0 \times ... \times C_{n-1}$. □

Lemma 42 (Lemma 6.11, [16])**.** *Suppose C_i is a central subuniverse of D_i for all $i \in \{0, 1, 2, ..., n\}$ and $n > 2$. Then V^1 proves that there does not exist a $(C_0, ..., C_{n-1})$-essential solution set $\mathscr{R}_\Theta \leq D_0 \times ... \times D_{n-1}$.*

Lemma 43 (Corollary 7.13.3, [15])**.** *Suppose C_i is a PC subuniverse of D_i for all $i \in \{0, 1, 2, ..., n\}$ and $n > 2$. Then V^1 proves that there does not exist a $(C_0, ..., C_{n-1})$-essential subdirect solution set $\mathscr{R}_\Theta \leq D_0 \times ... \times D_{n-1}$.*

For our purposes, the last two lemmas can be proved by an exhaustive search. For any subuniverses D_0, D_1, D_2 of the fixed algebra $\mathbb{A} = (A, \Omega)$, for any of their central /PC subuniverses C_0, C_1, C_2 and for any subalgebras $\mathscr{R}$ of $D_0 \times D_1 \times D_2$ check that $\mathscr{R}$ is not (C_0, C_1, C_2)-essential or (C_0, C_1, C_2)-essential and subdirect. For relations of higher arity just consider a projection for any of its three coordinates.

The following three lemmas about the interaction of subuniverses of different types are formulated for an arbitrary algebra D, i.e. for example, they can be used for $D = D_0 \times ... \times D_{n-1}$ and its subuniverses $\mathscr{R}_1$ and $\mathscr{R}_2$. For these cases, we can think about D

as of a domain set, and about B_1 and B_2 as of reductions $D_0^{(i)} \times ... \times D_{n-1}^{(i)}$ or solution sets $\mathscr{R}_\Theta$. Sometimes we consider $D = \mathscr{R}_\Theta \leq D_0 \times ... \times D_{n-1}$ and $B_1 = \mathscr{R}_\Theta \cap D_0^{(\bot)} \times ... \times D_{n-1}^{(\bot)}$, $B_2 = \mathscr{R}_\Theta \cap D_0^{(\top)} \times ... \times D_{n-1}^{(\top)}$, where $D^{(\bot)}, D^{(\top)}$ are reductions of some (different) types. The proof of these lemmas is based on simple universal algebra reasoning, and in the presence of all third-order objects, their formalization in W_1^1 does not differ much from [15], [16].

Lemma 44 (Lemma 7.28, [15], Lemma 6.25)**.** *Suppose B_1 is a binary absorbing, central, or linear subuniverse of D, B_2 is a subuniverse of D. Then $B_1 \cap B_2$ is a binary absorbing, central, or linear subuniverse of B_2, respectively.*

Lemma 45 (Lemma 7.29, [15])**.** *Suppose B_1 and B_2 are non-empty one-of-four subuniverses of D, $B_1 \cap B_2 = \emptyset$. Then B_1 and B_2 are subuniverses of the same type.*

Lemma 46 (Theorem 7.30, [15])**.** *Suppose B_1 and B_2 are one-of-four subuniverses of D of types $\mathcal{T}_1$ and $\mathcal{T}_2$. Then $B_1 \cap B_2$ is a one-of-four subuniverse of B_2 of type $\mathcal{T}_1$.*

The following lemma is proved by induction and is used for third-order objects, namely for reductions in strategies.

Lemma 47 (Lemma 7.31, [15])**.** *Suppose $\mathscr{A}_0 = \mathscr{B}_0$, $s \geq 1$, $t \geq 0$, $\mathscr{A}_i$ is a one-of-four subuniverse of $\mathscr{A}_{i-1}$ for every $i \in \{1, ..., s\}$, and $\mathscr{B}_i$ is a one-of-four subuniverse of $\mathscr{B}_{i-1}$ for every $i \in \{1, ..., t\}$. Then W_1^1 proves that $\mathscr{A}_s \cap \mathscr{B}_t$ is a one-of-four subuniverse of $\mathscr{A}_{s-1} \cap \mathscr{B}_t$ of the same type as $\mathscr{A}_s$.*

Proof. The proof of the claim goes by induction on $s + t$. If $t = 0$, then the claim follows from the statement. If $t \leq 1$, then by the inductive assumption, $\mathscr{A}_{s-1} \cap \mathscr{B}_t$ and $\mathscr{A}_s \cap \mathscr{B}_{t-1}$ are both one-of-four subuniverses of $\mathscr{A}_{s-1} \cap \mathscr{B}_{t-1}$, and the second one is of type $\mathcal{T}$. Then by Theorem 46 their intersection $\mathscr{A}_s \cap \mathscr{B}_t$ is a one-of-four subuniverses of $\mathscr{A}_{s-1} \cap \mathscr{B}_t$ of type $\mathcal{T}$.

We will formalize the proof for the specific case that we further need in the proofs of auxiliary lemmas about strategies. Suppose that $\mathscr{A}_0 = \mathscr{B}_0 = \mathscr{R}_\Theta \leq D_0 \times ... \times D_{n-1}$ for some CSP instance Θ with domain set D, where $\mathscr{R}_\Theta$ is its solution set, and for each $i \in \{1, ..., s\}$,

$$\mathscr{A}_i = \mathscr{R}_\Theta \cap D_0^{(i)} \times ... \times D_{n-1}^{(i)}$$

where $D = D^{(0)}, D^{(1)}, ..., D^{(s)}$ is some strategy for Θ, and analogously, for each $i \in \{1, ..., t\}$,

$$\mathscr{B}_i = \mathscr{R}_\Theta \cap D_0^{(i)'} \times ... \times D_{n-1}^{(i)'}$$

for some (other) strategy $D = D^{(0)'}, D^{(1)'}, ..., D^{(t)'}$ for Θ. Recall that we can formalize any strategy by one set $\Theta_{Str} < \langle nl, instsize(n, l) \rangle$. By Corollaries 2, 5 and Lemmas 33, 40 we know that W_1^1 proves that $\mathscr{A}_i$ is a one-of-four subuniverse of $\mathscr{A}_{i-1}$ and $\mathscr{B}_i$ is a one-of-four subuniverse of $\mathscr{B}_{i-1}$. Since in any step we reduce at least one domain, the number of steps t, s cannot be greater than nl and $t + s < 2nl$. This induction is available in W_1^1: the formula itself is $\Sigma_0^{1,b}$, but in the proof of the induction step we use the results proved in W_1^1. □

Lemma 48 (Lemma 7.32, [15])**.** *Suppose $\mathscr{R}_\Theta \subseteq \mathscr{A}_0 \times \mathscr{B}_0$ is a subdirect relation, $\mathscr{B}_i$ is a one-of-four subuniverse of $\mathscr{B}_{i-1}$ for every $i \in \{1, 2, ..., s\}$, $\mathscr{A}_1$ is a one-of-four subuniverse of $\mathscr{A}_0$. Then W_1^1 proves that $pr_1(\mathscr{R}_\Theta \cap (\mathscr{A}_1 \times \mathscr{B}_s))$ is a one-of-four subuniverse of $pr_1(\mathscr{R}_\Theta \cap (\mathscr{A}_1 \times \mathscr{B}_{s-1}))$ of the same type as $\mathscr{B}_s$.*

Proof. The statement of this lemma will eventually be used in the proof of Lemma 8.28 [15], which is used further for constraints and subconstraints in proofs of Theorem 36 and Theorem 37. So we will formalize the proof of the lemma for one specific case of Lemma 8.28 [15]. Consider some subdirect solution set $\mathscr{R}_\Theta \leq D_0 \times ... \times D_{n-1}$ (for constraints and projections the reasoning is similar). Let $D = D^{(0)}, D^{(1)}, ..., D^{(s)}$ be some strategy for Θ, and set for $i = 0, 1$

$$\mathscr{A}_i = pr_{0,1,...t-1}(D_0^{(i)} \times ... \times D_{t-1}^{(i)} \times D_t \times ... \times D_{n-1}),$$

and for $i = 0, ..., t$

$$\mathscr{B}_i = pr_{t,t+1,...n-1}(D_0 \times ... \times D_{t-1} \times D_t^{(i)} \times ... \times D_{n-1}^{(i)}).$$

Then by Corollaries 2, 5 and Lemmas 33, 40, W_1^1 proves that $\mathscr{R}_\Theta \cap (\mathscr{A}_0 \times \mathscr{B}_i)$ is a one-of-four subuniverse of $\mathscr{R}_\Theta \cap (\mathscr{A}_0 \times \mathscr{B}_{i-1})$ of the same type as $\mathscr{B}_i$, and $\mathscr{R}_\Theta \cap (\mathscr{A}_1 \times \mathscr{B}_0)$ is a one-of-four subuniverse of $\mathscr{R}_\Theta$. By Lemma 48, $\mathscr{R}_\Theta \cap (\mathscr{A}_1 \times \mathscr{B}_s)$ is a one-of-four subuniverse of $\mathscr{R}_\Theta \cap (\mathscr{A}_1 \times \mathscr{B}_{s-1})$ of the same type as $\mathscr{B}_s$. Consider a congruence $\mathscr{C}_\sigma$ on $\mathscr{R}_\Theta \cap (\mathscr{A}_1 \times \mathscr{B}_0)$ such that two elements are equivalent whenever their projections on the second coordinate are equal,

$$\mathscr{C}_\sigma(H_1, H_2) \iff \forall t \leq i < n,\ H_1(i) = H_2(i).$$

Note that this is $\Sigma_0^{1,b}$ definition. Then for every coordinate $i = 0, 1$, $Stable_1(\mathscr{R}_\Theta \cap (\mathscr{A}_1 \times \mathscr{B}_t), \mathscr{C}_\sigma)$, which means that if $\mathscr{R}_\Theta \cap (\mathscr{A}_1 \times \mathscr{B}_t)$ contains one element of the block of $\mathscr{C}_\sigma$, then it contains the entire block. We now need Lemma 7.26 from [15], which is used just once in this proof. Therefore, we will formalize it only for this specific case as a claim.

Claim 5. *Suppose $\mathscr{C}_\sigma$ is a congruence on $\mathscr{R}_\Theta \cap (\mathscr{A}_1 \times \mathscr{B}_0)$, $\mathscr{R}_\Theta \cap (\mathscr{A}_1 \times \mathscr{B}_t)$ is a one-of-four subuniverse of $\mathscr{R}_\Theta \cap (\mathscr{A}_1 \times \mathscr{B}_0)$ stable under $\mathscr{C}_\sigma$. Then W_1^1 proves that $\{H/\mathscr{C}_\sigma | H \in \mathscr{R}_\Theta \cap (\mathscr{A}_1 \times \mathscr{B}_t)\}$ is a one-of-four subuniverse of $\mathscr{R}_\Theta \cap (\mathscr{A}_1 \times \mathscr{B}_0)/\mathscr{C}_\sigma$ of the same type as $\mathscr{R}_\Theta \cap (\mathscr{A}_1 \times \mathscr{B}_t)$.*

The proof of the claim is as in [15]. The statement follows immediately from the claim. □

3.3.1.6 Some technical lemmas

In the following two lemmas, $\Theta(z)$ is the set of all $a \in D_z$ such that there is a solution to Θ with $z = a$. Analogously, $\Theta^{(1)}(z)$ is the set of all $a \in D_z^{(1)}$ such that there is a solution to $\Theta^{(1)}$ with $z = a$.

Lemma 49 (Lemma 8.1, [15])**.** *Suppose $D^{(1)}$ is a one-of-four reduction for an instance Θ of type $\mathcal{T}$, which is not of the PC type. Then W_1^1 proves that $\Theta^{(1)}(z)$ is a one-of-four subuniverse of $\Theta(z)$ of type $\mathcal{T}$ for every variable z.*

Proof. Consider a CSP instance Θ on n domains and its solution set $\mathscr{R}_\Theta$. Since $\mathscr{R}_\Theta$ is preserved by ω, $\Theta(i)$ is a subuniverse of D_i for every i, and by the definition of reductions, $D_i^{(1)}$ is a subuniverse of type $\mathcal{T}$. Thus, by Lemma 44, $\Theta(i) \cap D_i^{(1)}$ is a subuniverse of $\Theta(i)$ of type $\mathcal{T}$ for every $i < n$. Consider the reduction Θ' of Θ to the domain set $[\Theta(0), ..., \Theta(n-1)]$, $\mathscr{R}_{\Theta'}$ is a subdirect relation. Then, by Corollaries 3, 4 and Lemma 39, $\Theta^{(1)}(z)$ is a one-of-four subuniverse of $\Theta(z)$ of type $\mathcal{T}$. □

Lemma 50 (Lemma 8.2, [15])**.** *Suppose $D^{(1)}$ is a PC reduction for a 1-consistent instance Θ, for every variable y appearing at least twice in Θ the pp-formula $\Theta(y)$ defines D_y and $\Theta(z)$ defines D_z for a variable z. Then V^1 proves that $\Theta^{(1)}(z)$ is a PC subuniverse of D_z.*

Proof. For the proof, we first rename all variables in Θ so that every variable occurs just once. This instance is denoted as Θ_0. Then, step by step, we identify each two variables back to obtain the original instance, by the sequence $\Theta_0, \Theta_1, ...\Theta_s = \Theta$. We show that these transformations can be held in V^1.

Recall that we are allowed to have only one constraint relation for any two variables x, y (in that order). That is, for instance Θ with n variables, the number of possible constraints that involve one variable is at most $(2n-1)$ (and the number of all possible constraints is at most n^2). First, define the set of all variables that occur in Θ more than once:

$$\forall x < n,\, S(x) \iff \exists y \neq z < n,\, (E_{\mathcal{X}}(x,y) \vee E_{\mathcal{X}}(y,x)) \wedge (E_{\mathcal{X}}(x,z) \vee E_{\mathcal{X}}(z,x)). \quad (3.161)$$

Note that if we have two edges of the form $E_{\mathcal{X}}(x,y), E_{\mathcal{X}}(y,x)$, we need to rename one x to x' and one y to y', and do it at different steps. To perform this, we further define two sets of variables for any such x, S^{in}, S^{out}:

$$\begin{aligned} \forall x, y < n,\, S^{out}(x,y) &\iff S(x) \wedge E_{\mathcal{X}}(x,y), \\ \forall x, z < n,\, S^{in}(x,z) &\iff S(x) \wedge E_{\mathcal{X}}(z,x). \end{aligned} \quad (3.162)$$

When we rename every occurrence of a variable x, we can get at most $2n$ new variables (there are at most $(2n-1)$ constraints with x, and one of them could be a loop $E_{\mathcal{X}}(x,x)$). It works for every of n variables, so the maximal number of steps is $2n^2$. We now set $s = 2n^2$, $\Theta_s = \Theta$, $S_s = S, S^{out}_s = S^{out}, S^{in}_s = S^{in}$, and then for any $t = 1, ..., 2n^2$ we will define a new CSP instance Θ_{s-t} based on the following rules. If the set $S_{s-(t-1)}$ is empty (neither of the variables occurs at least twice) we just replicate the instance $\Theta_{s-(t-1)}$. Otherwise, for odd t, we consider the set $S^{in}_{(s-(t-1))}$ (and for even the set $S^{out}_{(s-(t-1))}$). If it is empty, replicate the instance and move on. If not, choose elements x, y such that $\langle x, y\rangle$ is the minimum element of $S^{in}_{(s-(t-1))}(x,y)$. Rename a variable x in that constraint by the next number after the maximum element in $V_{\mathcal{X}_{s-(t-1)}}$. For this construction, for each odd step t we consider additional sets $L_{(s-(t-1))}$ and $R_{(s-(t-1))}$, defined as follows:

$$\begin{aligned} L_{(s-(t-1))}(x) &\iff \exists y < max(V_{\mathcal{X}_{s-(t-1)}}) + 1,\, min(S^{in}_{(s-(t-1))}) = \langle x, y\rangle, \\ R_{(s-(t-1))}(y) &\iff \exists x < max(V_{\mathcal{X}_{s-(t-1)}}) + 1,\, min(S^{in}_{(s-(t-1))}) = \langle x, y\rangle. \end{aligned} \quad (3.163)$$

Now we are ready to define an instance digraph for step t:

$$\begin{aligned} \forall x < max(V_{\mathcal{X}_{s-(t-1)}}) + 1,\, V_{\mathcal{X}_{s-t}}(x) &\iff V_{\mathcal{X}_{s-(t-1)}}(x), \\ V_{\mathcal{X}_{s-t}}(max(V_{\mathcal{X}_{s-(t-1)}}) + 1) & \\ \forall x, y < max(V_{\mathcal{X}_{s-(t-1)}}) + 1,\, E_{\mathcal{X}_{s-t}}(x,y) &\iff E_{\mathcal{X}_{s-(t-1)}}(y,x) \wedge \\ \wedge(\neg L_{(s-(t-1))}(x) \vee \neg R_{(s-(t-1))}(y)) & \\ E_{\mathcal{X}_{s-t}}(y, max(V_{\mathcal{X}_{s-(t-1)}}) + 1) &\iff L_{(s-(t-1))}(x) \wedge R_{(s-(t-1))}(y). \end{aligned} \quad (3.164)$$

In parallel, we define a target digraph $\ddot{\mathcal{A}}_{s-t}$ by adding there a new domain $D_{max(V_{\mathcal{X}_{s-(t-1)}})+1}$ equal to D_x for a new variable $max(V_{\mathcal{X}_{s-(t-1)}})+1$ and $E^{yx}_{\ddot{\mathcal{A}}}$ as a constraint for a new edge $E_{\mathcal{X}_{s-t}}(y, max(V_{\mathcal{X}_{s-(t-1)}}) + 1)$. Eventually, we will get an instance Θ_0. Since we consider all sets in a particular order and address only sets from the previous step $t-1$, all of them exist by $\Sigma^{1,b}_1$ induction.

The proof of the statement then goes by induction on s, and the implication $s \to s+1$ follows from the reasoning that can be easily formalized in V^1. We refer the reader to the source [15]. □

For a relation $\mathscr{R}$ of arity n denote by $UnPol^{\mathscr{R}}$ the set of all unary vector functions preserving the relation $\mathscr{R}$. For a solution set $\mathscr{R}_\Theta$ for some CSP instance Θ, due to (3.96)

$$\Psi \in UnPol^{\mathscr{R}_\Theta} \iff VecFun(\mathscr{R}_\Theta, \Psi), \tag{3.165}$$

which is a $\Pi_1^{1,b}$-formula. For every map H from $[n]$ to $[D_0, ..., D_{n-1}]$, and every unary vector function Ψ, we can define a map $\Psi(H)$ using bit-definition:

$$\Psi(H)(\langle i, \langle i, a\rangle\rangle) = H^\Psi(\langle i, \langle i, a\rangle\rangle) \iff \exists b \in D_i,\ H(\langle i, \langle i, b\rangle\rangle) \wedge \Psi(i, b, a). \tag{3.166}$$

Lemma 51 (Lemma 8.12, [15]). *Suppose a pp-formula $\Lambda(x_0, ..., x_{n-1})$ defines a relation $\mathscr{R}_\Lambda$, $H \in D_{x_0} \times ... \times D_{x_{n-1}}$, and $\mathscr{R}' = \{H^\Psi : \Psi \in UnPol^{\mathscr{R}_\Lambda}\}$. Then W_1^1 proves that there exists $\Upsilon \in Covering(\Lambda)$ such that $\Upsilon(x_0, ..., x_{n-1})$ defines $\mathscr{R}'$.*

Proof. The idea of the universal algebra proof is the following. Consider any relation $\mathscr{R}$ on n variables. Suppose that there are l elements in each domain, $d_0, ..., d_{l-1}$. Then the formula

$$\mathscr{S}(x_0^{d_0}, ..., x_0^{d_{l-1}}, ..., x_{n-1}^{d_0}, ..., x_{n-1}^{d_{l-1}}) = \bigwedge_{(b_0,...,b_{n-1})\in\mathscr{R}} \mathscr{R}(x_0^{b_0}, ..., x_{n-1}^{b_{n-1}})$$

expresses that the vector-function preserves $\mathscr{R}$ (we think about $x_i^{b_i}$ as about x_i being sending to b_i). Then, if we consider any tuple $\alpha = (a_0, ..., a_{n-1})$, the projection of $\mathscr{S}$ to $x_0^{a_0}, ..., x_{n-1}^{a_{n-1}}$ defines the relation $\{f(\alpha) : f \in UnPol^{\mathscr{R}}\}$.

We will consider Λ as a CSP instance on n variables, $|V_{\mathcal{X}_\Lambda}| = n$ (for projections the reasoning is analogous). Suppose that for some $a_0, ..., a_{n-1}$, for all $i < n$, $H(i) = \langle i, a_i\rangle$. We need to define a new CSP instance Υ such that the projection of its solution set to some subset of vertices is exactly $\mathscr{R}'$. Consider a CSP instance Υ_{null} on nl variables, where for $i < n, a < l$ we think about vertex $il + a$ as about vertex i that was sent to $\langle i, a\rangle$ (or if we use labels, $x_i \to a \in D_i$). Then for every $H' \in \mathscr{R}_\Lambda$ such that for $i < n, b_i < l$, $H'(i) = \langle i, b_i\rangle$ we copy instance Λ to domains $D_{b_0}, D_{l+b_1}, D_{2l+b_2}, ..., D_{(n-1)l+b_{n-1}}$. Denote the resulting instance by Υ. It is clear that $\Upsilon \in Covering(\Lambda)$. Then the projection $\mathscr{R}_\Upsilon^{a_0, l+a_1, ..., (n-1)l+a_{n-1}}$ defines $\mathscr{R}'$.

The algorithm of the construction is clear, but to perform it we need the number of steps that is bounded only by l^n (the number of possible homomorphisms from $[n]$ to $[D_0, ..., D_{n-1}]$). Since every homomorphism H is expressed by a string of length $\langle n, \langle n, l\rangle\rangle < n^4$, we encode the number of steps by strings $\emptyset < T < n^4$ of length up to n^4, run the algorithm (if T represents some homomorphism to the instance Λ, copy Λ to corresponding domains), and then use $\Sigma_1^{\mathscr{B}}$-induction to show that such instance exists. □

Corollary 7 (Corollary 8.12.1, [15]). *Suppose a pp-formula $\Lambda(x_0, ..., x_{n-1})$ defines a relation $\mathscr{R}_\Lambda$ without a tuple $H \in D_{x_0} \times ... \times D_{x_{n-1}}$, Σ is the set of all relations defined by $\Upsilon(x_0, ..., x_{n-1})$ where $\Upsilon \in Covering(\Lambda)$, and $\mathscr{R}_\Lambda$ is an inclusion-maximal relation in Σ without the tuple H. Then W_1^1 proves that H is a key tuple for $\mathscr{R}_\Lambda$.*

Proof. Consider Λ as a CSP instance on n variables, let S be any map from $[n]$ to $[D_0, ..., D_{n-1}]$ that is not in $\mathscr{R}_\Lambda$. Then by Lemma 51 the set of maps $\mathscr{R}' = \{S^\Psi : \Psi \in UnPol^{\mathscr{R}_\Lambda}\}$ is a projection of the solution set to some $\Upsilon \in Covering(\Lambda)$. Since Ψ can be constant mapping to a homomorphism of Λ and identity mapping, $\mathscr{R}_\Lambda \subsetneq \mathscr{R}'$, and since $\mathscr{R}_\Lambda$ is inclusion-maximal, $H \in \mathscr{R}'$. By the definition, H is a key tuple for $\mathscr{R}_\Lambda$. □

Lemmas we consider next in this section are

1. either related exclusively to binary relations since we consider languages with at most binary constraint relations,

2. or are used in the further proofs only for constant arity relations,

3. or related to constant arity relations and constant sizes over algebra $\mathbb{A}$.

In the first and the second cases, they can be formalized and proved in V^1 exactly as they are proved in [15]. In the third case, such properties must be listed in the $\mathbb{A}$-Monster set. For these reasons, we will mention a few examples, but for the proofs and the rest, we refer the reader to the source [15].

Lemma 52 (Lemma 7.19, [15]). *Suppose $R \subseteq D \times B \times B$ is a subdirect relation, D is a PC algebra without a non-trivial binary absorbing or central subuniverse, and for every $b \in B$ there exists $a \in A$ such that $(a, b, b) \in R$. Then V^1 proves that for every $a \in A$ there exists $b \in B$ such that $(a, b, b) \in R$.*

The following two lemmas are formulated in [15] for t variables, but then they are only used for relations on one and two variables, so instead of using induction on t we can consider just cases $\Theta(x_0)$ and $\Theta(x_0, x_1)$.

Lemma 53 (Lemma 8.3, [15]). *Suppose $D^{(1)}$ is a minimal absorbing, central or linear reduction for an instance Θ, and $\Theta(x_0, x_1)$ defines a full relation. Then $\Theta^{(1)}(x_0, x_1)$ defines a full or empty relation.*

Lemma 54 (Lemma 8.4, [15]). *Suppose $D^{(1)}$ is a minimal PC reduction for a 1-consistent instance Θ. For every variable y appearing at least twice in Θ the pp-formula $\Theta(y)$ defines D_y and $\Theta(x_0, x_1)$ defines a full relation. Then $\Theta^{(1)}(x_0, x_1)$ defines a full or empty relation*

The next examples of technical lemmas are the following.

Lemma 55 (Lemma 8.10, [15]). *Suppose $R \leq D_i \times D_j$ is a critical rectangular binary relation, and R' is a cover of R. Then V^1 proves that $Con_2^{(R,i)} \subsetneq Con_2^{(R,i)}$.*

Lemma 56 (Theorem 8.15, [15]). *Suppose $R \leq D^4$ is a strongly rich relation preserved by an idempotent WNU. Then V^1 proves that there exists an abelian group $(D, +)$ and bijective mappings $\phi_0, \phi_1, \phi_2, \phi_3 : D \to D$ such that*

$$R = \{(x_0, x_1, x_2, x_3) : \phi_0(x_0) + \phi_1(x_1) + \phi_2(x_2) + \phi_3(x_3) = 0\}.$$

Lemma 57 (Theorem 8.17, [15]). *Suppose $\sigma \subseteq D^2$ is a congruence, ρ is a bridge from σ to σ such that $\tilde{\rho}$ is a full relation, $pr_{1,2}(\rho) = \omega$, ω is a minimal relation stable under σ such that $\sigma \subsetneq \omega$. Then V^1 proves that there exists a prime number p and a relation $\zeta \subseteq D \times D \times \mathbb{Z}_p$ such that $pr_{1,2}\zeta = \omega$ and $(a_1, a_2, b) \in \zeta$ implies that $(a_1, a_2) \in \sigma \iff (b = 0)$.*

Lemma 58 (Lemma 8.18, [15]). *Suppose $\rho \subseteq D^4$ is an optimal bridge from σ_1 to σ_2, and σ_1 and σ_2 are different irreducible congruences. Then V^1 proves that $\sigma_2 \subseteq \tilde{\rho}$.*

Lemma 59 (Lemma 8.20, [15]). *Suppose $R \leq D_i \times D_j$ is a subdirect rectangular relation, and there exist $(b_i, a_j), (a_i, b_j) \in R$ such that $(a_i, a_j) \notin R$. Then V^1 proves that there exists a bridge δ from $Con_2^{(R,i)}$ to $Con_2^{(R,j)}$ such that $\tilde{\delta} = R$.*

These and some other lemmas imply the following result about CSP instances. Recall that cycle-consistency is formalized by $\Pi_2^{1,b}$-formula, and $Linked_{[i,i,\Theta]}$ is a $\Sigma_1^{1,b}$-relation, see Chapter 2 (ref. [6]).

Lemma 60 (Lemma 8.22, [15]). *Suppose Θ is a cycle-consistent connected instance. Then V^1 proves that for any constraints C, C' with variables x, x' there exists a bridge δ from $Con^{(C,x)}$ to $Con^{(C',x')}$ such that $\tilde{\delta}$ contains all pairs of elements linked in Θ. Moreover, if $Con^{(C'',x'')} \neq Linked_{[x'',x'',\Theta]}$ for some constraint $C'' \in \Theta$ and a variable x'', then δ can be chosen so that $\tilde{\delta}$ contains all pairs of elements linked in Θ', where Θ' is obtained from Θ by replacing every constraint relation by its cover.*

The next lemma is proved easily by the application of the definition of the crucial instance and expanded covering. If we replace any constraint in a crucial instance Θ with all weaker constraints, we get a solution. All relations in expanded covering Θ' are either diagonal relations or weaker or equivalent to relations in Θ.

Lemma 61 (Lemma 8.24, [15]). *Suppose $\Theta_{\mathcal{X}} = (\mathcal{X}, \ddot{\mathcal{A}})$ is a crucial instance in $D^{(1)}$, $\Theta_{\mathcal{Y}} = (\mathcal{Y}, \ddot{\mathcal{B}}) \in ExpCov(\Theta_{\mathcal{X}})$ via the homomorphism H from $\mathcal{Y}$ to $\mathcal{X}$, and $\Theta_{\mathcal{Y}}$ has no solution in $D^{(1)}$. Then V^1 proves that for every constraint $E^{x_i x_j}_{\ddot{\mathcal{A}}}$ in $\Theta_{\mathcal{X}}$ there exists a constraint $E^{y_k y_p}_{\ddot{\mathcal{B}}}$ such that $H(y_k) = x_i$, $H(y_p) = x_j$ and $E^{x_i x_j}_{\ddot{\mathcal{A}}} = E^{y_k y_p}_{\ddot{\mathcal{B}}}$.*

Lemmas 61 and 60 imply Lemma 62.

Lemma 62 (Lemma 8.25, [15]). *Suppose $\Theta_{\mathcal{X}} = (\mathcal{X}, \ddot{\mathcal{A}})$ is a crucial instance in $D^{(1)}$, $\Theta_{\mathcal{Y}} = (\mathcal{Y}, \ddot{\mathcal{B}}) \in ExpCov(\Theta_{\mathcal{X}})$ has no solution in $D^{(1)}$, every constraint relation of $\Theta_{\mathcal{X}}$ is a critical rectangular relation, and $\Theta_{\mathcal{Y}}$ is connected. Then V^1 proves that $\Theta_{\mathcal{X}}$ is connected.*

3.3.2 Formalization of the main theorems

3.3.2.1 The existence of the next reduction

Lemma 63 (Lemma 9.1, [15]). *Suppose $D^{(0)}, D^{(1)}, ..., D^{(s)}$ is a strategy for a 1-consistent CSP instance Θ, and $D^{(\perp)}$ is a reduction of $\Theta^{(s)}$. Then V^1 proves that:*

1. *if there exists a 1-consistent reduction contained in $D^{(\perp)}$ and $D^{(s+1)}$ is maximal among such reductions, then for every variable x of Θ there exists a tree-formula $\Upsilon_x \in Coverings(\Theta)$ such that $\Upsilon_x^{(\perp)}(x)$ defines $D_x^{(s+1)}$;*
2. *otherwise, there exists a tree-formula $\Upsilon \in Coverings(\Theta)$ such that $\Upsilon^{(\perp)}$ has no solutions.*

Proof. The proof of this theorem is based on constraint propagation. At the beginning for every variable x we consider an empty tree formula Υ_x. Then $\Upsilon_x^{(\perp)}$ defines the reduction $D^{(\perp)}$. Then the recursive algorithm works as follows: if at some step the reduction defined by these tree formulas is 1-consistent, it stops. Otherwise, it considers any constraint $C = R(x_1, ..., x_t)$ that breaks 1-consistency. The current restrictions of variables $x_1, ..., x_t$ in C imply stronger restriction of some variable x_i, and the algorithm changes the formula Υ_{x_i} as follows:

$$\Upsilon_{x_i} =_{def} C \wedge \Upsilon_{x_1} \wedge ... \wedge \Upsilon_{x_t}.$$

To keep the formula Υ_{x_i} tree, any time the algorithm joins Υ_{x_j} and Υ_{x_k} it renames the variables so that they do not have common variables. Finally, for each Υ_{x_j} we consider the reduction of this instance on the domain set $D^{(\perp)}$. Projection of the solution set to $\Upsilon_{x_j}^{(\perp)}$ on variable x_j, $\Upsilon_{x_j}^{(\perp)}(x_j)$ defines $D_{x_j}^{(s+1)}$. That will be a maximal 1-consistent reduction since it is defined by tree-formulas.

Let us formalize this algorithm. The formalization is based on a 1-consistency algorithm (see [1]). Recall that any CSP instance can be converted in polynomial time to

a 1-consistent one with the same set of solutions. Moreover, any implementation of a 1-consistency algorithm derives the same unary constraints. Thus, we can first define recursive sets of edges and vertices, based on which we can construct our tree formulas.

For the steps $t = 0, t = 1$ for any $i, j < n$ we set

$$\begin{aligned} E^{ij}_{\ddot{\mathcal{A}},0}(a,b) &\iff E^{ij}_{\ddot{\mathcal{A}}}(a,b), \\ E^{ij}_{\ddot{\mathcal{A}},1}(a,b) &\iff a \in D_i^{(\perp)} \wedge b \in D_j^{(\perp)} \wedge E^{ij}_{\ddot{\mathcal{A}}}(a,b). \end{aligned} \tag{3.167}$$

For any further step $t > 1$ we will propagate constraints recursively until we cannot change any domain further (i.e. until the instance is 1-consistent) or some domain is empty. For any $i, j < n$ we set:

$$\begin{aligned} E^{ij}_{\ddot{\mathcal{A}},t}(a,b) \iff & [(\forall p, r < n, \exists e, f < l,\ E_{\mathcal{X}}(p,r) \to E^{pr}_{\ddot{\mathcal{A}},t-1}(e,f)) \wedge \\ & \wedge E^{ij}_{\ddot{\mathcal{A}},t-1}(a,b) \wedge \forall q < n,\ E_{\mathcal{X}}(i,q) \to \exists d \in D_q^{(\perp)},\ E^{iq}_{\ddot{\mathcal{A}},t-1}(a,d) \wedge \\ & \wedge \forall k < n,\ E_{\mathcal{X}}(k,i) \to \exists c \in D_k^{(\perp)},\ E^{ki}_{\ddot{\mathcal{A}},t-1}(c,a) \wedge \\ & \wedge \forall q < n,\ E_{\mathcal{X}}(j,q) \to \exists d \in D_q^{(\perp)},\ E^{jq}_{\ddot{\mathcal{A}},t-1}(b,d) \wedge \\ & \wedge \forall k < n,\ E_{\mathcal{X}}(k,j) \to \exists c \in D_k^{(\perp)},\ E^{kj}_{\ddot{\mathcal{A}},t-1}(c,b)] \vee \\ \vee [(\exists p, r < n, \forall e, f < l,\ & E_{\mathcal{X}}(p,r) \wedge \neg E^{pr}_{\ddot{\mathcal{A}},t-1}(e,f)) \wedge E^{ij}_{\ddot{\mathcal{A}},t-1}(a,b)]. \end{aligned} \tag{3.168}$$

The expression in the first square brackets holds when neither of the relations at the previous step is empty, and the expression in the second square brackets holds otherwise (it would mean that some domain of the instance at this step is already empty). In both cases after some step t the relation set $E^{ij}_{\ddot{\mathcal{A}},t-1}$ stops changing. The maximal number of edges in a directed graph with loops on n vertices is n^2. Therefore, the maximal number of edges in the instance Θ is n^2l^2 and since at each step we reduce some relation at least by one edge, it is enough to consider at most n^2l^2 steps. Moreover, since we remove an edge if at least one of its endpoints $a \in D_i^{(\perp)}$ violates 1-consistency (so within one step we remove all edges in $E^{ij}_{\ddot{\mathcal{A}},t-1}$ connected with a for all $j < n$), the actual number of steps is nl (the number of elements). The existence of this set is ensured by $\Sigma^{1,b}_1$-induction: consider the formula

$$\begin{aligned} \phi(t) = \exists E_{\ddot{\mathcal{A}}} < \langle t, \langle\langle n, l\rangle, \langle n, l\rangle\rangle\rangle,\ & \forall i, j < n, \forall a, b < l,\ E^{ij}_{\ddot{\mathcal{A}},1}(a,b) \leftrightarrow (3.167) \wedge \\ & \wedge \forall 1 < p < t, \forall i, j < n, \forall a, b < l,\ E^{ij}_{\ddot{\mathcal{A}},p}(a,b) \leftrightarrow (3.168). \end{aligned} \tag{3.169}$$

Since (3.168) is a $\Sigma^{1,b}_0$-formula, to provide the implication $\phi(t) \to \phi(t+1)$ we can use comprehension axiom $\Sigma^{1,b}_0$-CA.

Note that in (3.168) we do not need to track the domain's changes separately (they are all recorded in the relations $E^{ij}_{\ddot{\mathcal{A}},t}$). We will proceed with recursive propagation of the domain set $V_{\ddot{\mathcal{A}}}$ after this procedure based on the resulting relation set. For any $i < n$, for steps $t = 0, t = 1$ we set

$$\begin{aligned} V_{\ddot{\mathcal{A}},0}(i,a) &\iff D_i(a), \\ V_{\ddot{\mathcal{A}},1}(i,a) &\iff D_i^{(\perp)}(a), \end{aligned} \tag{3.170}$$

and for all $1 < t < nl$

$$\begin{aligned} V_{\ddot{\mathcal{A}},t}(i,a) \iff & V_{\ddot{\mathcal{A}}'t-1}(i,a) \wedge (\forall j < n,\ E_{\mathcal{X}}(i,j) \to (\exists b < l,\ V_{\ddot{\mathcal{A}}'t-1}(j,b) \wedge \\ & \wedge E^{ij}_{\ddot{\mathcal{A}},t}(a,b))) \wedge (\forall k < n,\ E_{\mathcal{X}}(k,i) \to (\exists c < l,\ V_{\ddot{\mathcal{A}},t-1}(k,c) \wedge \\ & \wedge E^{ki}_{\ddot{\mathcal{A}},t}(c,a))). \end{aligned} \tag{3.171}$$

Again, this set exists due to $\Sigma_1^{1,b}$-induction. Note that in (3.171) for the step $t < nl$ we use $E_{\ddot{\mathcal{A}},t-1}$ and not $E_{\ddot{\mathcal{A}},nl}$: we need recursive changing of the domains for further reconstruction of the tree formulas Υ_i. We also define a set C_i^{list} that for any step $0 < t < nl$ collect elements that were deleted from $V_{\ddot{\mathcal{A}},t,i}$:

$$\forall 0 < t < nl, \forall a < l,\, C_i^{list}(t,a) \iff V_{\ddot{\mathcal{A}},t-1}(i,a) \wedge \neg V_{\ddot{\mathcal{A}},t}(i,a). \tag{3.172}$$

Further, we need to construct tree-instances Υ_i. We want them to be coverings, so for each i we only need to define an instance graph $\mathcal{X}_i$ and remember parents for renamed variables. To do it, we again use recursion. For each $i < n$, we start with an instance $\Upsilon_{i,0}$ with a domain set D and with an empty set of constraints. Then for further steps u we:

- either do nothing with instance $\Upsilon_{i,u}$ - if for any $k < n$ and for some t constraints $E^{ik}_{\ddot{\mathcal{A}},t}$ and $E^{ki}_{\ddot{\mathcal{A}},t}$ that violate 1-consistency do not imply stronger restriction of a domain D_i,
- or we need to consider a union of two CSP instances (that corresponds to the intersection of their constraints) for every constraint $E^{ij}_{\ddot{\mathcal{A}},t}$ (or $E^{ji}_{\ddot{\mathcal{A}},t}$) restricting domain D_i, namely
$$\Upsilon_{i,u} := E^{ij}_{\ddot{\mathcal{A}}} \wedge \Upsilon_{i,u-1} \wedge \Upsilon_{j,u-1}.$$

Note that while our next move depends on reduced by the step constraint $E^{ik}_{\ddot{\mathcal{A}},t}$, to the instance $\Upsilon_{i,u}$ we add original constraint $E^{ik}_{\ddot{\mathcal{A}}}$. After we add enough such constraints and previous instances, the reduction of the resulted instance $\Upsilon_{i,u}$ to $D^{(\bot)}$ will give us the same projection to the coordinate D_i as it gives (3.171).

Also note that if two constraints at step t reduce domain D_i by the same values, we do not need to and we will not construct both intersections. Recall that when joining any two $\Upsilon_{i,t-1}, \Upsilon_{j,t-1}$ we have to rename all variables to retain the instances tree. Since we have at most binary relations and for any two variables there can be only two constraints containing them, namely $E^{ij}_{\ddot{\mathcal{A}}}$ and $E^{ji}_{\ddot{\mathcal{A}}}$, at the first step of the recursion process, we can add to each $\Upsilon_{i,1}$ at most $2n$ new vertices (if for every $j < n$ there are both $E_{\mathcal{X}}(i,j)$ and $E_{\mathcal{X}}(j,i)$). Then for every new constraint restricting at step t, we can at most double the number of variables of the largest instance of step $(t-1)$. Still, it will not make sense after the first l intersections for every Υ_i since in this case we will get an empty domain set D_i and thus justify the case 2 of the theorem. Thus, even if we start with instances $\Upsilon_{i,1}$ on $2n$ variables, after l intersections we will not need more than $2^l 2n$ variables.

First, for every $\Upsilon_{i,0}$, define $V_{\mathcal{X}_i,0}$ as a set of length $2^l 2n$ that contains only one element i, and $E_{\mathcal{X}_i,0}$ as an empty set of length $\langle 2^l 2n, 2^l 2n\rangle$. By $V_{\ddot{\mathcal{A}}_i,0}$ denote the set of length $\langle 2^l 2n, l\rangle$, with only one non-empty domain $V_{\ddot{\mathcal{A}}_i,0} = D_i$. By $E_{\ddot{\mathcal{A}}_i,0}$ denote an empty set of length $\langle\langle 2^l 2n, l\rangle, \langle 2^l 2n, l\rangle\rangle$.

Remark 14. Strictly speaking, we are not allowed to use empty sets of some length. But we can bypass it by choosing a set, for example, for $V_{\mathcal{X}_i,0}$ with two elements, i, and that we will never properly use, $2^l 2n + 1$. Further, we also consider the number function $max'(V_{\mathcal{X}_i,u})$ with the following value:

$$max'(V_{\mathcal{X}_i,u}) = m \iff m \neq 2^l 2n + 1 \wedge \forall u \in V_{\mathcal{X}_i,u}, (u \neq 2^l 2n + 1 \to u \leq m). \tag{3.173}$$

We construct Υ_0,...,Υ_{n-1} simultaneously. The entire construction takes $0 < u < 2n(nl)$ steps. Each $\Upsilon_{i,u}$ consists of

- set $V_{\mathcal{X}_i,u}$, representing the current number of vertices,
- set $E_{\mathcal{X}_i,u}$, representing the current set of edges,
- set $V_{\ddot{\mathcal{A}}_i,u}$, representing domains for current variables,
- set $E_{\ddot{\mathcal{A}}_i,u}$, representing constraint relations for current variables,
- and set $C^{erase}_{i,u}$ that keeps track of elements that must be deleted from the domain D_i during each outer step t and erases them one by one if we need to change the instance $\Upsilon_{i,u}$.

We consider these sets in the above order. The description of the algorithm is as follows. For every Υ_i, within any step $1 < t < nl$ we run $2n$ internal steps. At the beginning of each new internal iteration, for some $2nt$ step, we check the list $C^{list}_{i,t+1}$ for elements that we will exclude from D_i by adding new constraints to the instance during this internal iteration. We write them down to $C^{erase}_{i,2nt}$. For each step $u = 2nt + j$ we consider the constraint $E^{ij}_{\ddot{\mathcal{A}},t+1}$ (for step $2nt + j + 1$ we consider the opposite constraint $E^{ji}_{\ddot{\mathcal{A}},t+1}$) and decide whether it kills any of the elements from $C^{erase}_{i,2nt+j}$. For any $a \in C^{erase}_{i,2nt+j}$ it happens when there exists at least one edge $(a, b) \in E^{ij}_{\ddot{\mathcal{A}},t}$ and no edges connected with a in $E^{ij}_{\ddot{\mathcal{A}},t+1}$:

$$\exists a \in C^{erase}_{i,2nt+j}(\exists b < l,\, E^{ij}_{\ddot{\mathcal{A}},t}(a, b) \wedge \forall b < l,\, \neg E^{ij}_{\ddot{\mathcal{A}},t+1}(a, b)).\ \text{[formula (3.168)]} \tag{3.174}$$

If it is the case, we define our instance $\Upsilon_{i,u+1}$ as follows: we first replicate instance $\Upsilon_{i,u}$ and then add all vertices of $\Upsilon_{j,u}$ to part from $max(V_{\mathcal{X}_i,u}) + 1$ as well as all edges of $\Upsilon_{j,u}$, and add an edge $E_{\mathcal{X}_i,u+1}(i, max(V_{\mathcal{X}_i,u}) + 1 + j)$.

$$\begin{gathered}
\forall k < max'(V_{\mathcal{X}_i,u}) + 1,\, V_{\mathcal{X}_i,u+1}(k) \iff V_{\mathcal{X}_i,u}(k),\\
\forall k_1, k_2 < max'(V_{\mathcal{X}_i,u}) + 1,\, E_{\mathcal{X}_i,u+1}(k_1, k_2) \iff E_{\mathcal{X}_i,u}(k_1, k_2),\\
\forall max'(V_{\mathcal{X}_i,u}) < k < 2^l 2n + 1,\, V_{\mathcal{X}_i,u+1}(max'(V_{\mathcal{X}_i,u}) + 1 + k) \iff V_{\mathcal{X}_j,u}(k),\\
\forall max'(V_{\mathcal{X}_i,u}) < k_1, k_2 < 2^l 2n + 1,\\
E_{\mathcal{X}_i,u+1}(max'(V_{\mathcal{X}_i,u}) + 1 + k_1, max'(V_{\mathcal{X}_i,u}) + 1 + k_2) \iff E_{\mathcal{X}_j,u}(k_1, k_2),\\
E_{\mathcal{X}_i,u+1}(i, max'(V_{\mathcal{X}_i,u}) + 1 + j).
\end{gathered} \tag{3.175}$$

In the same way, we define a target digraph $\ddot{\mathcal{A}}_{i,u+1}$ by adding new domains for new variables (from the list $\{D_0, ..., D_{n-1}\}$) and $E^{ij}_{\ddot{\mathcal{A}}}$ as a constraint for the new edge $E_{\mathcal{X}_i,u+1}(i, max(V_{\mathcal{X}_i,u}) + 1 + j)$.

If this is not the case, we just replicate instance $\Upsilon_{i,u}$ to $\Upsilon_{i,u+1}$. Finally, we either replicate the set $C^{erase}_{i,u}$ or change it to $C^{erase}_{i,u+1}$ as follows:

$$\forall a < l,\, C^{erase}_{i,(2nt+j)+1}(a) \iff C^{erase}_{i,2nt+j}(a) \wedge (\exists b < l,\, E^{ij}_{\ddot{\mathcal{A}},t+1}(a, b)). \tag{3.176}$$

Thus, after we pass constraint $E^{ji}_{\ddot{\mathcal{A}},t+1}$ we leave in $C^{erase}_{i,(2nt+j)+1}$ those elements that will be deleted in $V_{\ddot{\mathcal{A}},t+1}$ but because of another constraint that will lose all edges adjacent to them. We keep track of already deleted elements for outer step t not to intersect the instances with constraints that kill the same set of vertices – because we want to stop after l intersection with an empty domain.

At each step sets $V_{\mathcal{X}_i,u}$, $V_{\ddot{\mathcal{A}}_i,u}$, $E_{\mathcal{X}_i,u}$, $E_{\ddot{\mathcal{A}}_i,u}$ and $C^{erase}_{i,u}$ address to themselves and each other in previous steps. They also address to different levels of already defined set $C^{list}_i(t, a)$ based on $V_{\ddot{\mathcal{A}},t}$ and to $E^{ij}_{\ddot{\mathcal{A}},t}$. The existence of them is given by $\Sigma^{1,b}_1$-induction. At some point, we stop with tree-instances Υ_i, each of them defining $D^{(s+1)}_i$ on $D^{(\perp)}$. □

The next three theorems follow from Lemma 63 and some previous results, formalized in W_1^1.

Theorem 30 (Theorem 9.2, [15]). *Suppose $D^{(0)}, D^{(1)}, ..., D^{(s)}$ is a strategy for a cycle-consistent CSP instance Θ. Then W_1^1 proves that:*

1. *if $D_x^{(s)}$ has a non-trivial binary absorbing subuniverse B then there exists a 1-consistent absorbing reduction $D^{(s+1)}$ of $\Theta^{(s)}$ with $D_x^{(s+1)} \subseteq B$;*
2. *if $D_x^{(s)}$ has a non-trivial central subuniverse C then there exists a 1-consistent central reduction $D^{(s+1)}$ of $\Theta^{(s)}$ with $D_x^{(s+1)} \subseteq B$;*
3. *if $D^{(s)}$ has no non-trivial binary absorbing or central subuniverse for every y but there exists a non-trivial PC subuniverse B in $D_x^{(s)}$ for some x, then there exists a 1-consistent PC reduction $D^{(s+1)}$ of $\Theta^{(s)}$ with $D_x^{(s+1)} \subseteq B$.*

Theorem 31 (Theorem 9.3, [15]). *Suppose that $D^{(0)}, D^{(1)}, ..., D^{(s)}$ is a strategy for a 1-consistent CSP instance Θ, and $D^{(\perp)}$ is a non-linear 1-consistent reduction of $\Theta^{(s)}$. Then W_1^1 proves that there exists a 1-consistent minimal reduction $D^{(s+1)}$ of $\Theta^{(s)}$ of the same type such that $D_i^{(s+1)} \subseteq D_i^{(\perp)}$ for every variable i.*

Theorem 32 (Theorem 9.4, [15]). *Suppose $D^{(\perp)}$ is a 1-consistent PC reduction for a cycle-consistent irreducible CSP instance Θ, Θ is not linked and not fragmented. Then W_1^1 proves that there exist a reduction $D^{(1)}$ of Θ and a minimal strategy $D^{(1)}, ..., D^{(s)}$ for $\Theta^{(1)}$ such that the solution set to $\Theta^{(1)}$ is subdirect, the reductions $D^{(2)}, ..., D^{(s)}$ are non-linear, $D_x^{(s)} \subseteq D_x^{(\perp)}$ for every variable x.*

3.3.2.2 Main theorems proved by induction

In this section we consider the main five theorems, proved simultaneously by induction on the size of the domain set (to be defined further). We will not consider the formalization of their proofs since it is based on the formalization of previous results. However, some reasoning from the proofs is used for the formalization of the theorems.

Remark 15. We will use the same notation $D^{(s)}$ for the reductions of the initial instance, its subinstances, subconstraints, differences, unions, and both coverings and expanded coverings to avoid unnecessary indices. These, of course, cannot be the same sets of domains, but once given $D^{(s)}$ for an instance $\Theta_{\mathcal{X}}$ we can easily construct a similar reduction for any of these objects, denoted by $\Theta_{\mathcal{Y}}$, under the simple rule

$$\forall x_i \forall y_j,\ D_{x_i} = D_{y_j} \implies D_{x_i}^{(s)} = D_{y_j}^{(s)}.$$

This is well-defined since we can additionally require in the reduction $D^{(s)}$ of instance $\Theta_{\mathcal{X}}$ that equal domains be reduced to equal domains (see 3.57). In a minimal 1-consistent one-of-four reduction, every $D_{x_i}^{(s)}$ must be minimal by inclusion.

Theorem 33 (Theorem 9.5, [15]). *Suppose $D^{(1)}$ is a minimal 1-consistent one-of-four reduction of a cycle-consistent irreducible CSP instance Θ, $\Lambda(x_0, ..., x_{n-1})$ is a subconstraint of Θ, the solution set to $\Lambda^{(1)}$ is subdirect, $\Theta \backslash \Lambda$ has a solution in $D^{(1)}$, and Θ has no solutions in $D^{(1)}$. Then W_1^1 proves that there exist instances $\Upsilon_1, ..., \Upsilon_t \in Coverings(\Lambda)$ such that $\Phi = (\Theta \backslash \Lambda) \cup \Upsilon_1 \cup ... \cup \Upsilon_t$ has no solutions in $D^{(1)}$, each $\Upsilon_i(x_0, ..., x_{n-1})$ is a subconstraint of Φ, and $\Upsilon_i^{(1)}(x_0, ..., x_{n-1})$ defines a subdirect key relation with the parallelogram property for every i.*

The formalization of the theorem will be based on its proof. Since $\Lambda(x_0,...,x_{n-1})$ is a subconstraint of Θ, it follows that Λ is a subinstance of Θ that involves variables $x_0,...,x_{n-1}$, $y_0,...,y_{k-1}$, and Θ as an instance on variables $x_0,...,x_{n-1},y_0,...,y_{k-1},z_0,...,z_{s-1}$ such that $\Theta\backslash\Lambda$ involves variables $x_0,...,x_{n-1},z_0,...,z_{s-1}$. $\Upsilon_i(x_0,...,x_{n-1})$ here denotes all tuples $(a_0,...,a_{n-1})$ such that instance Υ_i has a solution with $x_0=a_0$,...,$x_{n-1}=a_{n-1}$. That is, it is a projection of the solution set to Υ_i onto coordinates $x_0,...,x_{n-1}$, which can be expressed by the formula

$$\exists y_0^i ... \exists y_{m_i-1}^i \Upsilon_i(x_0,...,x_{n-1},y_0^i,...,y_{m_i-1}^i).$$

$\Upsilon_i^{(1)}(x_0,...,x_{n-1})$ thus expressed the projection of the solution set to the instance $\Upsilon_i^{(1)}$ after the reduction $D^{(1)}$. We can denote this projection using a third-order object $\mathscr{R}_{\Upsilon_i^{(1)}}^{x_0,...,x_{n-1}}$. Note that when we talk not about a solution set to an instance but about projection to the solution set, we add to the formula an additional second-sorted existential quantifier, see (3.27).

Since both Λ and $\Theta\backslash\Lambda$ have solutions in $D^{(1)}$, but Θ does not, it follows that $\Lambda^{(1)}(x_0,...,x_{n-1})$ and $\Theta\backslash\Lambda^{(1)}(x_0,...,x_{n-1})$ define relations $\mathscr{R}_{\Lambda^{(1)}}^{x_0,...,x_{n-1}}$ and $\mathscr{R}_{\Theta\backslash\Lambda^{(1)}}^{x_0,...,x_{n-1}}$ that do not intersect. Every solution to $\Lambda^{(1)}$ is a solution to any $\Upsilon^{(1)}$ from $Coverings(\Lambda)$. According to the proof of the theorem, for every tuple H_i of the relation $\mathscr{R}_{\Theta\backslash\Lambda^{(1)}}^{x_0,...,x_{n-1}}$, we find an instance $\Upsilon_i^{(1)}$ such that the relation $\mathscr{R}_{\Upsilon_i^{(1)}}^{x_0,...,x_{n-1}}$ defined by $\Upsilon_i^{(1)}(x_0,...,x_{n-1})$ is an inclusion-maximal relation that contains $\mathscr{R}_{\Lambda^{(1)}}^{x_0,...,x_{n-1}}$ and does not contain H_i. Thus, $\Phi=(\Theta\backslash\Lambda)\cup\Upsilon_1\cup...\cup\Upsilon_t$ does not have solutions in $D^{(1)}$, but if we replace any Υ_i by a weaker instance Υ that produces a greater relation $\mathscr{R}_{\Upsilon^{(1)}}^{x_0,...,x_{n-1}}$, we get an instance with solution H_i. That is, the number of coverings $\Upsilon_1,...,\Upsilon_t$ is bounded by the number of tuples in $\mathscr{R}_{\Theta\backslash\Lambda^{(1)}}^{x_0,...,x_{n-1}}$, which is bounded by $l^n/2-|\mathscr{R}_{\Lambda^{(1)}}^{x_0,...,x_{n-1}}|$. Note that we do not need to know the precise number of Υ_i to write the formula; some of them can be repeated as many times as necessary. So we stick to the bound l^n, since it can be conveniently rewritten as $(2^n)^{log_2 l}$. Then, following the reasoning from Lemma 51, we can roughly bound the number of variables in each Υ_i by $(n+k)+nl$ (we introduce a new variable x_i^a for all $i\in\{0,...,n-1\}$ and $a<l$). Thus, every instance Υ_i can be bound by a unique number $b_\Lambda=instsize((n+k)+nl,l)$. It follows that we can encode the set of all $\Upsilon_0,...,\Upsilon_{(2^n)^{log_2 l}}$ by one class $\mathscr{Y}$, where each Υ_i is encoded by a string X of length at most nv, with $v=\lceil log_2 l\rceil$, $\mathscr{Y}(X,\Upsilon)$. Then $r\tilde{o}w(X,\mathscr{Y},b_\Lambda)=\Upsilon$, which we denote as $\Upsilon_{[X]}$.

Due to the assumption, each Υ_i is a covering for Λ on some set of variables $x_0^i,...,x_{n-1}^i$, $y_0^i,...,y_{m_i-1}^i$ such that for all $j<n$, $x_j^i=x_j$. Therefore, there is a homomorphism H from $\mathcal{X}_{\Upsilon_i}$ to $\mathcal{X}_\Lambda$ that sends x_j^i to x_j. Each Υ_i is a subconstraint of Φ, hence it has no common variables with $\Theta\backslash\Lambda$ and any Υ_j except for $x_0,...,x_{n-1}$. Recall that for the union of instances we have the function uni well-defined by $\Sigma_0^{1,b}$-formula, as well as the function dif for the difference. In the union of two instances, we add all variables of the second instance after all variables of the first instance, shift their labels, and add equality constraints between vertices with labels that were the same. The problem here is that when we join Υ_i to $(\Theta\backslash\Lambda)\cup\Upsilon_1\cup...\cup\Upsilon_{i-1}$ we rename all the vertices and since the number of Υ_i can be exponential, we have no space to represent Φ as a second-order object. We cannot represent Φ as a third-sorted object either (with vertices labeled by strings) since the statement that there is no solution to Φ in $D^{(1)}$ would be the $\Pi_1^{\mathscr{B}}$-formula. To avoid this problem, we will not define Φ, but define the preconditions that lead to the situation where Φ does not have a solution in $D^{(1)}$. By Corollary 7 these preconditions also lead to each $\mathscr{R}_{\Upsilon_i^{(1)}}^{x_0,...,x_{n-1}}$ being a key relation (so we do not need to write it down explicitly in the

formula). We order all projections to $x_0, ..., x_{n-1}$ of solutions to an instance $\Theta\backslash\Lambda^{(1)}$ in one class $\mathscr{H}$, where each H is encoded by a string X of length at most nv, denoted $H_{[X]}$. That $H_{[X]}$ will correspond to $\Upsilon_{[X]}$ in the sense that $\mathscr{R}_{\Upsilon^{(1)}_{[X]}}$ is an inclusion-maximal relation that does not contain $H_{[X]}$. This is reflected in square brackets in the formula.

The function *redinst* is definable by the $\Sigma^{1,b}_0$-formula and returns the reduction of an instance on $D^{(1)}$. Thus,

$$\Theta^{(1)} = redinst(\Theta, D^{(1)})$$
$$\Theta\backslash\Lambda^{(1)} = redinst(\Theta\backslash\Lambda, D^{(1)})$$
$$\Upsilon^{(1)}_i = redinst(\Upsilon_i, D^{(1)}).$$

We also cannot use the relation $subConst_n(\Phi, \Upsilon_i, X)$ since we cannot define Φ and technically Υ_i is not a subinstance of Φ. But we can explicitly write this condition for each Υ_i. In the 7th line of the formula (3.177) we require that the first n variables in each Υ_i be labeled exactly by $x_0, ..., x_{n-1}$ (we are talking about the existence), in the $8-9$th lines we ensure that the common variables of any Υ_i and $\Theta\backslash\Lambda$ are only $x_0, ..., x_{n-1}$, and in the last two lines we require the same for each pair Υ_i, Υ_j.

The relation $subD(\mathscr{R}^{x_0,...,x_{n-1}}_{\Upsilon^{(1)}_i})$ is $\Sigma^{1,b}_1$, the relation $ParlPr(\mathscr{R}^{x_0,...,x_{n-1}}_{\Upsilon^{(1)}_i})$ remains $\Pi^{1,b}_2$. Relations $min1of4Red(D^{(1)}, D)$, $1C(\Theta^{(1)})$ and $subConst_n(\Theta, \Lambda, x_0, ..., x_{n-1})$ are described by $\Sigma^{1,b}_0$ formulas, relations $subDSSInst(\Lambda^{(1)})$, $Cov(\Upsilon_i, \Lambda)$ and $H\ddot{O}M(\mathcal{X}_{\Theta\backslash\Lambda^{(1)}}, \ddot{\mathcal{A}}_{\Theta\backslash\Lambda^{(1)}})$ are $\Sigma^{1,b}_1$, relation $\neg H\ddot{O}M(\mathcal{X}_{\Theta^{(1)}}, \ddot{\mathcal{A}}_{\Theta^{(1)}})$ becomes $\Pi^{1,b}_1$, and $CCInst(\Theta)$ and $IRDInst(\Theta)$ are $\Pi^{1,b}_2$-formulas. This gives us the $\Sigma^{\mathscr{B}}_1$-formula.

$$\begin{aligned}
T9.5(\Theta, D^{(1)}, \Lambda, X) := (&CCInst(\Theta) \wedge IRDInst(\Theta) \wedge min1of4Red(D^{(1)}, D) \wedge \\
&\wedge 1C(\Theta^{(1)}) \wedge subConst(\Theta, \Lambda, X) \wedge subDSSInst(\Lambda^{(1)}) \wedge \\
&\wedge H\ddot{O}M(\mathcal{X}_{\Theta\backslash\Lambda^{(1)}}, \ddot{\mathcal{A}}_{\Theta\backslash\Lambda^{(1)}}) \wedge \neg H\ddot{O}M(\mathcal{X}_{\Theta^{(1)}}, \ddot{\mathcal{A}}_{\Theta^{(1)}})) \implies \exists\mathscr{H}\exists\mathscr{Y}\forall X < nv, \\
&[Cov(\Upsilon_{[X]}, \Lambda) \wedge H_{[X]} \in \mathscr{R}^{x_0,...,x_{n-1}}_{\Theta\backslash\Lambda^{(1)}} \wedge H_{[X]} \notin \mathscr{R}^{x_0,...,x_{n-1}}_{\Upsilon^{(1)}_{[X]}} \wedge \forall\Upsilon < b_\Lambda \\
&(Cov(\Upsilon, \Lambda) \wedge \mathscr{R}^{x_0,...,x_{n-1}}_{\Upsilon^{(1)}_{[X]}} \subsetneq \mathscr{R}^{x_0,...,x_{n-1}}_{\Upsilon^{(1)}}) \to H_{[X]} \in \mathscr{R}^{x_0,...,x_{n-1}}_{\Upsilon^{(1)}})] \wedge \\
&\wedge \forall X < nv,\ subDSSInst(\mathscr{R}^{x_0,...,x_{n-1}}_{\Upsilon^{(1)}_{[X]}}) \wedge ParlPr(\mathscr{R}^{x_0,...,x_{n-1}}_{\Upsilon^{(1)}_{[X]}}) \wedge \\
&\wedge \forall X < nv, \forall j < n,\ V_{\mathcal{X}_{\Upsilon_{[X]}}}(j, x_j) \wedge \\
&\wedge \forall X < nv, \forall r < (s+n), \forall g < b_{n+s}, \forall p < (n+k) + nl, \\
&V_{X_{\Theta\backslash\Lambda}}(r, g) \wedge V_{\mathcal{X}_{\Upsilon_{[X]}}}(p, g) \to (g = x_0 \vee ... \vee g = x_{n-1}) \wedge \\
&\wedge \forall X < nv, \forall X' < nv, \forall r, p < (n+k) + nl, \forall g < b_{(n+k)+nl}, \\
&V_{\mathcal{X}_{\Upsilon_{[X]}}}(r, g) \wedge V_{\mathcal{X}_{\Upsilon_{[X']}}}(p, g) \to (g = x_0 \vee ... \vee g = x_{n-1}) \wedge
\end{aligned} \tag{3.177}$$

Theorem 34 (Theorem 9.6, [15]). *Suppose $D^{(1)}$ is a minimal 1-consistent one-of-four reduction of a cycle-consistent irreducible CSP instance Θ, Θ is crucial in $D^{(1)}$ and is not connected. Then W^1_1 proves that there exists an instance $\Theta' \in ExpCov(\Theta)$ that is crucial in $D^{(1)}$ and contains a linked connected component whose solution set is not subdirect.*

To formalize this theorem, we first have to formalize some additional notions used in its proof since we need a bound on the instance Θ'. For every variable x of instance Θ,

all constraints of which are critical and rectangular, we assign the pair of sets $\xi^{\Theta,x} = (\Sigma^{\Theta,1}_{\mathcal{D}_x}, \Sigma^{\Theta,2}_{\mathcal{D}_x})$ such that for all $i < 2^{l^2}$ and $a, b < l$

$$\begin{gathered}\Sigma^{\Theta,1}_{\mathcal{D}_x}(i,a,b) \iff \Sigma_{\mathcal{D}_x}(i,a,b) \wedge \big((\exists y < n, \forall a', b' < l,\\ Con_2^{(\Theta,x)}(0,y,a',b') \leftrightarrow \Sigma_{\mathcal{D}_x}(i,a',b'))\wedge\\ \wedge(\forall z \neq y < n, (E_{\mathcal{X}}(x,z) \to Con_2^{(E_{\mathcal{X}}(x,z),x)} \not\subseteq \Sigma_{\mathcal{D}_x,i}\wedge\\ \wedge E_{\mathcal{X}}(z,x) \to Con_2^{(E_{\mathcal{X}}(z,x),x)} \not\subseteq \Sigma_{\mathcal{D}_x,i})))\vee\\ ((\exists y < n, \forall a', b' < l,\, Con_2^{(\Theta,x)}(y,0,a',b') \leftrightarrow \Sigma_{\mathcal{D}_x}(i,a',b'))\wedge\\ \wedge(\forall z \neq y < n, (E_{\mathcal{X}}(x,z) \to Con_2^{(E_{\mathcal{X}}(x,z),x)} \not\subseteq \Sigma_{\mathcal{D}_x,i}\wedge\\ E_{\mathcal{X}}(z,x) \to Con_2^{(E_{\mathcal{X}}(z,x),x)} \not\subseteq \Sigma_{\mathcal{D}_x,i}))).\end{gathered} \tag{3.178}$$

and

$$\begin{gathered}\Sigma^{\Theta,2}_{\mathcal{D}_x}(i,a,b) \iff \Sigma^{\Theta,1}_{\mathcal{D}_x}(i,a,b) \wedge \forall j < 2^{l^2}\\ (j \neq i \wedge \Sigma^{\Theta,1}_{\mathcal{D}_x,j} \neq \emptyset \to \neg Adj(\Sigma^{\Theta,1}_{\mathcal{D}_x,i}, \Sigma^{\Theta,1}_{\mathcal{D}_x,j})).\end{gathered} \tag{3.179}$$

Thus, $\Sigma^{\Theta,1}_{\mathcal{D}_x}$ is the set of all minimal congruences among the set $Con_2^{(\Theta,x)}$, and $\Sigma^{\Theta,2}_{\mathcal{D}_x}$ is the set of all minimal congruences among the congruences of $Con_2^{(\Theta,x)}$ that are not adjacent with any other congruence from $\Sigma^{\Theta,1}_{\mathcal{D}_x}$. That is, both sets contain mutually non-inclusive congruences among $Con_2^{(\Theta,x)}$ (note that a minimal congruence among $Con_2^{(\Theta,x)}$ is not necessarily a minimal congruence on D_x). The lists can be empty for some i. We call $\xi^{\Theta,x}$ a characteristic of x. Next, we define a partial order on such characteristics. We further consider only sets of irreducible congruences. For two sets Σ and Σ', define relations $\leq$ and $<$ as follows:

$$\begin{gathered}\Sigma \leq \Sigma' \iff \forall i < 2^{l^2}, (\Sigma_i \neq \emptyset \to \exists j < 2^{l^2}, \Sigma'_j \subseteq \Sigma_i),\\ \Sigma < \Sigma' \iff \Sigma \leq \Sigma' \wedge \neg \Sigma' \leq \Sigma.\end{gathered} \tag{3.180}$$

Relations $\Sigma \backslash \Sigma'$ and $\Sigma = \Sigma'$ are defined in the usual way. Also, for any set of irreducible congruences $\Sigma'_{\mathcal{D}_x}$ define a function $\uparrow optset$ that returns the set of all congruences σ on D_x such that $\delta \subseteq \sigma$ for some $\delta \in optset(\Sigma'_{\mathcal{D}_x})$:

$$\begin{gathered}\uparrow optset(\Sigma'_{\mathcal{D}_x})(i,a,b) \iff \Sigma_{\mathcal{D}_x}(i,a,b)\wedge\\ \wedge(\exists j < 2^{l^2}, optset(\Sigma'_{\mathcal{D}_x,j}) \neq \emptyset \wedge optset(\Sigma'_{\mathcal{D}_x,j}) \subseteq \Sigma_{\mathcal{D}_x,i}).\end{gathered} \tag{3.181}$$

Finally, if (Σ_1, Σ_2) and (Σ'_1, Σ'_2) are two characteristics, then define a relation $\lesssim$ as

$$\begin{gathered}(\Sigma_1, \Sigma_2) \lesssim (\Sigma'_1, \Sigma'_2) \iff (\Sigma_1 < \Sigma'_1) \vee (\Sigma_1 = \Sigma'_1 \wedge \Sigma_2 \leq \Sigma'_2)\vee\\ \vee(\Sigma_1 = \Sigma'_1 \wedge \neg\Sigma_2 \leq \Sigma'_2 \wedge \neg\Sigma'_2 \leq \Sigma_2 \wedge \Sigma_2\backslash(\uparrow optset(\Sigma_1)) < \Sigma'_2\backslash(\uparrow optset(\Sigma_1))).\end{gathered} \tag{3.182}$$

That is, we say that $(\Sigma_1, \Sigma_2) \lesssim (\Sigma'_1, \Sigma'_2)$ if

1. either every congruence in Σ_1 contains some congruence of Σ'_1 and these sets are not equal;
2. or Σ_1 is equal to Σ'_1 and every congruence in Σ_2 contains some congruence of Σ'_2;
3. or Σ_1 is equal to Σ'_1, sets Σ_2 and Σ'_2 are incomparable (there exists a congruence in Σ_2 that does not contain any congruence of Σ_2 and vise versa) and every congruence in $\Sigma_2\backslash(\uparrow optset(\Sigma_1))$ contains some congruence of $\Sigma_2\backslash(\uparrow optset(\Sigma_1))$ and these sets are not equal.

When we decrease a characteristic of a variable, we can decrease the number of congruences in either of sets Σ_1, Σ_2, $\Sigma_2 \backslash (\uparrow optset(\Sigma_1))$ or enlarge congruences. Since for an algebra $\mathbb{A}$ and all its subuniverses we have at most 2^{l^2} congruences, we can decrease a characteristic of a variable at most $2 \cdot 2^{l^2}$ times, which is a constant.

We then define three transformations of an instance, giving an expanded covering of the original instance. These transformations do not increase the characteristics of related variables. The first transformation T_1 makes an instance crucial in some reduction $D^{(1)}$: it replaces constraints by all weaker constraints until the instance is crucial in $D^{(1)}$. The second transformation T_2 splits a variable x based on two congruences on D_x. Finally, the third transformation T_3 makes some changes for a connected component of an instance. Transformations T_1, T_2, T_3 are not unique, but we do not need them to be unique and therefore can formalize them as relations. Thus, for two instances Θ and Θ' we say that Θ' is a T_1 transformation of Θ if

$$\begin{aligned} T_1(\Theta', \Theta) \iff & V_{\mathcal{X}} = V_{\mathcal{X}'} \wedge \forall i, j < n,\ E_{\mathcal{X}'}(i,j) \to E_{\mathcal{X}}(i,j) \wedge \forall i, j < n, \\ & Weaker(E^{ij}_{\ddot{\mathcal{A}}'}, E^{ij}_{\ddot{\mathcal{A}}}) \wedge CrucInst(\Theta', D^{(1)}). \end{aligned} \tag{3.183}$$

For the second transformation T_2, we choose a variable x, choose two congruences σ_1, σ_2 on D_x, and define two subsets of constraints in Θ containing x, $\Lambda_1 = \{C_1^1, C_1^2, ..., C_1^k\}$ and $\Lambda_2 = \{C_2^1, C_2^2, ..., C_2^s\}$ such that $Con_2^{(C_1^i, x)} = \sigma_1$ and $Con_2^{(C_2^i, x)} = \sigma_2$. Denote by Λ_0 all constraints in $\Theta \backslash \Lambda_1 \cup \Lambda_2$ containing x. Then the instance Θ is transformed as follows. We choose two new variables x_1, x_2 and

1. rename x by x_1 in all constraints from Λ_0 and Λ_1;
2. rename x by x_2 in all constraints from Λ_2;
3. add the constraints $\sigma_1^*(x_1, x_2)$ and $\sigma_2^*(x_1, x_2)$;
4. for every $\sigma \in Con_2^{(\Lambda_0, x)}$ add the constraint $\sigma(x_1, x_2)$.

Both x_1, x_2 are *children* for x. To formalize this transformation, we will use labels for variables. We choose new labels $x_1 = max(V_{\mathcal{X}}) + 1, x_2 = max(V_{\mathcal{X}}) + 2$. To simplify the following formula, we abbreviate by $E_{\mathcal{X}}(x,y)$ both $E_{\mathcal{X}}(x,y)$ and $E_{\mathcal{X}}(y,x)$.

$$\begin{aligned} T_2(\Theta', \Theta, \sigma_1, \sigma_2, x) \iff & irCong_m(\sigma_1, D_x) \wedge irCong_m(\sigma_2, D_x) \wedge \\ & \wedge ExpCov(\Theta', \Theta) \wedge \forall t, s < b_n,\ t \neq x \wedge s \neq x \to \\ & \to (E_{\mathcal{X}'}(t,s) \leftrightarrow E_{\mathcal{X}}(t,s) \wedge E^{ts}_{\ddot{\mathcal{A}}'}(a,b) \leftrightarrow E^{ts}_{\ddot{\mathcal{A}}}(a,b)) \wedge \\ \wedge (\forall y < b_n,\ E_{\mathcal{X}}(x,y) \wedge Con_2^{(E_{\mathcal{X}}(x,y),x)} = \sigma_2 \to E_{\mathcal{X}'}(x_2, y)) \wedge & (\forall y < b_n,\ E_{\mathcal{X}}(x,y) \wedge \\ \wedge (Con_2^{(E_{\mathcal{X}}(x,y),x)} = \sigma_1 \vee (Con_2^{(E_{\mathcal{X}}(x,y),x)} \neq \sigma_1 \wedge Con_2^{(E_{\mathcal{X}}(x,y),x)} \neq \sigma_2)) \to & \\ \to E_{\mathcal{X}'}(x_1, y)) \wedge & \\ \wedge E_{\mathcal{X}'}(x_1, x_2) \wedge \forall a, b < l,\ E^{x_1 x_2}_{\mathcal{A}'}(a,b) \leftrightarrow \sigma_1^*(a,b) \wedge \sigma_2^*(a,b) \wedge & \\ \wedge (\forall y < b_n,\ E_{\mathcal{X}}(x,y) \wedge (Con_2^{(E_{\mathcal{X}}(x,y),x)} \neq \sigma_1 \wedge Con_2^{(E_{\mathcal{X}}(x,y),x)} \neq \sigma_2) \to & \\ \to Con_2^{(E_{\mathcal{X}}(x,y),x)}(a,b)). & \end{aligned} \tag{3.184}$$

The second and third lines of the formula (3.184) reflect the fact that we do not change any constraint not containing x. The last three lines add to the instance new constraints from items $3, 4$ (recall that we allowed to have only one constraint relation for any two variables x_1, x_2 and instead of the set of constraints consider its intersection).

Finally, the third transformation T_3 uses as an argument a connected component $\Lambda \subseteq \Theta$. By $\mathrm{MinVar}(\Lambda, \Theta) = \{x_1, ..., x_s\}$, where $s \geq 1$, we denote the set of all variables x_i such that there exists $\sigma \in Con_2^{(\Lambda, x_i)}$ that is minimal among $Con_2^{(\Theta, x_i)}$. Then the new instance Θ' is defined in the following way. We choose new variables $x'_1, ..., x'_s$ and

1. rename the variables $x_1, ..., x_s$ by $x'_1, ..., x'_s$ in $\Theta \backslash \Lambda$;
2. add the covers of all constraints from Λ with $x'_1, ..., x'_s$ instead of $x_1, ..., x_s$;
3. for every j and every $\sigma \in Con_2^{(\Theta \backslash \Lambda, x_j)}$ add the constraint $\sigma^*(x_j, x'_j)$;
4. for every j and $\sigma \in Con_2^{(\Theta \backslash \Lambda, x_j)}$ such that $Linked(a, b, x_j, x_j, \Lambda) \nsubseteq \sigma$, add the constraint $\delta_j(x_j, x'_j)$, where $\{\delta_j\} = optset(Con^{(\Lambda, x_j)})$.

We call each x_i a *parent* for x'_i. We can formalize transformation T_3 as a relation $T_3(\Theta', \Theta, \Lambda)$ in V^1 in the same way as the previous two ones, and we do not perform it here. The complexity of all these formulas does not exceed $\Sigma_2^{1,b}$. All transformations T_1, T_2, T_3 produce expanded coverings. The important thing is that the transformation T_1 does not change the number of variables, transformation T_2 increases the number of variables by 1, and transformation T_3 increases the number of variables by $s \leq n$.

In the proof of Theorem 34, we consider a sequence of instances $\Theta_1, \Theta_2, ..., \Theta_k, \Theta_{k+1}...$ such that every Θ_{i+1} is produced from Θ_i either by composition of transformations T_1T_2, or by composition T_1T_3. Due to some auxiliary lemmas in [15], compositions T_1T_2 and T_1T_3 do not increase a characteristic of any variable. Composition T_1T_2, splitting a variable x to x_1 and x_2, decreases the number of minimal congruences among $Con_2^{(\Theta, x)}$ by one for both x_1, x_2. Since the number of all congruences on A (and any of its subuniverse D) is bounded by 2^{l^2}, the number of total new variables that we can produce from x by applying T_1T_2 to it and all its children is bounded by $2^{2^{l^2}}$. Composition T_1T_3 also decreases a characteristic of x by enlarging all congruences for x', thus the number of descendants in one chain is also bounded by 2^{l^2}.

Let us call a variable x in instance Θ *stable* if all congruences in $Con_2^{(\Theta, x)}$ are adjacent. Also, two variables x_1, x_2 are *friends* if there is $E_{\mathcal{X}}(x_1, x_2)$ or $E_{\mathcal{X}}(x_2, x_1)$. By applying T_1T_3, we also decrease characteristics of all non-stable y's in $\mathrm{MinVar}(\Lambda_i, \Theta_i)$, and we can reuse every non-stable variable at most 2^{l^2} times. Thus, after at most 2^{l^2} steps, every variable in instance Θ_i for some i is stable. A stable variable y cannot be a friend with both a variable z' and its parent z. Considering the set of friends of y in Θ_j for $j > i$, we thus see that going from Θ_j to Θ_{j+1} we can replace an old friend of y with at most 2 new weaker friends and cannot add a new friend keeping its parent. Therefore, after Θ_i with n_i variables, at any step $j > i$ any variable y will have at most $(n_i - 1)2^{2^{l^2}}$ friends. Since any instance in the sequence $\Theta_1, \Theta_2, ..., \Theta_k, \Theta_{k+1}...$ is not fragmented, from some auxiliary axioms in [15] it follows that there is an instance Θ_s for some s that satisfies all conditions posed on instance Θ' in Theorem 34.

Considering all the above, we can conclude that the number of instances in a sequence $\Theta_1, \Theta_2, ..., \Theta_s$ cannot exceed the exponential bound, and the size of any instance Θ_i has some bound b_Θ polynomial in n which we will not calculate precisely. The important thing is that we can formalize the sequence by a third-order class $\mathcal{Y}$, where each instance Θ_i for $1 \leq i \leq s$ is encoded by a string X of length at most v, $\mathcal{Y}(X, \Theta)$. We denote such instance by $\Theta_{[X]}$.

In the formula (3.185), $redinst(\Theta', linkcomp(\Theta', D_i, a))$ is a composed function, where *linkcomp* is expressed by $\Sigma_1^{1,b}$-formula and returns the reduction of the domain set. The

complexity of the relations $min1of4Red(D^{(1)}, D)$ and $1C(\Theta^{(1)})$ is $\Sigma_0^{1,b}$. The complexity of the relations

$$\neg subDSSInst(redinst(\Theta', linkcomp(\Theta', D_i, a)))$$

and $\neg Connected(\Theta)$ is $\Pi_1^{1,b}$. $CrucInst(\Theta, D^{(1)})$ and $CrucInst(\Theta', D^{(1)})$ are expressed by formulas from $\mathfrak{B}(\Sigma_1^{1,b})$. The complexity of $ExpCov(\Theta', \Theta)$ is $\Sigma_1^{1,b}$. Finally, the complexity of relations $CCInst(\Theta)$ and $IRDInst(\Theta)$ is $\Pi_2^{1,b}$. This gives us $\Sigma_1^{\mathscr{B}}$-formula.

$$\begin{aligned} T9.6(\Theta, D^{(1)}) := \big[CCInst(\Theta) \wedge IRDInst(\Theta) \wedge min1of4Red(D^{(1)}, D) \wedge \\ \wedge 1C(\Theta^{(1)}) \wedge CrucInst(\Theta, D^{(1)}) \wedge \neg Connected(\Theta)\big] \implies \exists \mathscr{Y}, \Theta_{[\emptyset]} = \Theta \wedge \\ \forall X < v, \big[\Theta_{[S(X)]} = \Theta_{[X]} \vee (\exists x < v, \exists \sigma_1 < \langle l, l\rangle, \exists \sigma_2 < \langle l, l\rangle, \exists \Theta < b_\Theta \\ (T_2(\Theta, \Theta_{[X]}, \sigma_1, \sigma_2, x) \wedge T_1(\Theta_{[S(X)]}, \Theta))) \vee \\ \vee (\exists \Lambda < b_\Theta, \exists \Theta < b_\Theta (T_3(\Theta, \Theta_{[X]}, \Lambda) \wedge T_1(\Theta_{[S(X)]}, \Theta))) \wedge \\ \wedge ExpCov(\Theta_{[S(X)]}, \Theta_{[X]}) \wedge CrucInst(\Theta_{[S(X)]}, D^{(1)})\big] \wedge \forall X < v,\ S(X) = v \rightarrow \\ \rightarrow \exists a \in D_0, \neg subDSSInst(redinst(\Theta_{[X]}, linkcomp(\Theta_{[X]}, D_i, a))). \end{aligned} \tag{3.185}$$

Theorem 35 (Theorem 9.7, [15]). *Suppose $D^{(1)}$ is a 1-consistent non-linear reduction of a cycle-consistent irreducible instance Θ. If Θ has a solution, then W_1^1 proves that it has a solution in $D^{(1)}$.*

The complexity of relations $nonLNRed(D^{(1)}, D)$, $1C(\Theta^{(1)})$, $H\ddot{O}M(\mathcal{X}^{(1)}, \ddot{\mathcal{A}}^{(1)}, H')$ and $H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}}, H))$ is $\Sigma_0^{1,b}$, and the complexity of relations $CCInst(\Theta)$ and $IRDInst(\Theta)$ is $\Pi_2^{1,b}$. This gives us $\Sigma_2^{1,b}$-formula.

$$\begin{aligned} T9.7(\Theta, D^{(1)}) := \big(CCInst(\Theta) \wedge IRDInst(\Theta) \wedge \\ \wedge nonLNRed(D^{(1)}, D) \wedge 1C(\Theta^{(1)}) \wedge \\ \wedge \exists H < \langle n, \langle n, l\rangle\rangle, H\ddot{O}M(\mathcal{X}, \ddot{\mathcal{A}}, H)) \implies \exists H' < \langle n, \langle n, l\rangle\rangle, H\ddot{O}M(\mathcal{X}^{(1)}, \ddot{\mathcal{A}}^{(1)}, H'). \end{aligned} \tag{3.186}$$

Theorem 36 (Theorem 9.8, [15]). *Suppose $D^{(0)}, ..., D^{(s)}$ is a minimal strategy for a cycle-consistent irreducible CSP instance Θ, and a constraint $\rho(x_0, ..., x_{n-1})$ of Θ is crucial in $D^{(s)}$. Then W_1^1 proves that ρ is a critical relation with the parallelogram property.*

Since we consider only binary constraints of an instance Θ, relations $Critical_2(E^{ij}_{\ddot{\mathcal{A}}})$, $ParlPr_2(E^{ij}_{\ddot{\mathcal{A}}})$, and $minStrategy(\Theta, \Theta_{Str}, s)$ are $\Sigma_0^{1,b}$, relation $CrucConst(E^{ij}_{\ddot{\mathcal{A}}}, \Theta, D_Str^{(s)})$ is a Boolean combination of $\Sigma_1^{1,b}$ and $\Pi_1^{1,b}$ formulas, and relations $CCInst(\Theta)$, $IRDInst(\Theta)$ are $\Pi_2^{1,b}$. This gives us $\Sigma_2^{1,b}$-formula.

$$\begin{aligned} T9.8(\Theta, \Theta_{Str}) := \big(CCInst(\Theta) \wedge IRDInst(\Theta) \wedge minStrategy(\Theta, \Theta_{Str}, s) \wedge \\ \exists i, j < n,\ E_{\mathcal{X}}(i, j) \wedge CrucConst(E^{ij}_{\ddot{\mathcal{A}}}, \Theta, D_Str^{(s)})\big) \implies \\ \implies Critical_2(E^{ij}_{\ddot{\mathcal{A}}}) \wedge ParlPr_2(E^{ij}_{\ddot{\mathcal{A}}}). \end{aligned} \tag{3.187}$$

Theorem 37 (Theorem 9.9, [15]). *Suppose $D^{(0)}, ..., D^{(s)}$ is a minimal strategy for a cycle-consistent irreducible CSP instance Θ, $\Upsilon(x_0, ..., x_{n-1})$ is a subconstraint if Θ, the solution set to $\Upsilon^{(s)}$ is subdirect, $k \in \{0, 2, ..., n-2\}$, $Var(\Upsilon) = \{x_0, ..., x_{n-1}, u_0, ..., u_{t-1}\}$,*

$$\Lambda = \Upsilon^{y_0,...,y_{k-1},v_0,...,v_{t-1}}_{x_0,...,x_{k-1},u_1,...,u_{t-1}} \wedge \Upsilon^{y_k,...,y_{n-1},v_t,...,v_{2t-1}}_{x_k,...,x_{n-1},u_0,...,u_{t-1}} \wedge \Upsilon^{y_0,...,y_{n-1},v_{2t},...,v_{3t-1}}_{x_0,...,x_{n-1},u_0,...,u_{t-1}} = \Upsilon_1 \wedge \Upsilon_2 \wedge \Upsilon_3$$

and $\Theta^{(s)}$ has no solutions. Then W_1^1 proves that $(\Theta \backslash \Upsilon) \cup \Lambda$ has no solutions in $D^{(s)}$.

To get Υ_1, Υ_2 and Υ_3 we use function $substitute_k$, $substitute_{n-k}$ and $substitute_n$ that has $\Sigma_0^{1,b}$ definition. After the substitution,

$$Var(\Upsilon_1) = \{y_0, ..., y_{k-1}, x_k, ..., x_{n-1}, v_0, ..., v_{t-1}\},$$
$$Var(\Upsilon_2) = \{x_0, ..., x_{k-1}, y_k, ..., y_{n-1}, v_t, ..., v_{2t-1}\},$$

and

$$Var(\Upsilon_3) = \{y_0, ..., y_{k-1}, y_k, ..., y_{n-1}, v_{2t}, ..., v_{3t-1}\}.$$

The instance Λ here is just an intersection of all constraints of three new instances, i.e. the union $\Upsilon_1 \cup \Upsilon_2 \cup \Upsilon_3$. Relations $subConst_n(\Theta, \Upsilon, x_0, ..., x_{n-1})$ and $minStrategy(\Theta, \Theta_{Str}, s)$ are expressed by $\Sigma_0^{1,b}$-formulas, relation $subDSSInst(\Upsilon^{(s)})$ is $\Sigma_1^{1,b}$-formula. Relations $\neg H\ddot{O}M(\mathcal{X}_{\Theta^{(s)}}, \ddot{\mathcal{A}}_{\Theta^{(s)}})$ and $\neg H\ddot{O}M(\mathcal{X}_{(\Theta\setminus\Upsilon)\cup\Lambda^{(s)}}, \ddot{\mathcal{A}}_{(\Theta\setminus\Upsilon)\cup\Lambda^{(s)}})$ are $\Pi_1^{1,b}$. Finally, relations $CCInst(\Theta)$ and $IRDInst(\Theta)$ are $\Pi_2^{1,b}$. This gives us $\Sigma_2^{1,b}$-formula:

$$\begin{aligned} T9.9(\Theta, \Theta_{Str}, \Upsilon, X) := (&CCInst(\Theta) \wedge IRDInst(\Theta) \wedge minStrategy(\Theta, \Theta_{Str}, s) \wedge \\ &\wedge \forall i < n,\, V_{\mathcal{X}_\Upsilon}(i, x_i) \wedge \forall i < t,\, V_{\mathcal{X}_\Upsilon}(n+i, u_i) \wedge subConst(\Theta, \Upsilon, X) \wedge \\ &\wedge subDSSInst(\Upsilon^{(s)}) \wedge \neg H\ddot{O}M(\mathcal{X}_{\Theta^{(s)}}, \ddot{\mathcal{A}}_{\Theta^{(s)}})) \implies \\ &\implies \neg H\ddot{O}M(\mathcal{X}_{(\Theta\setminus\Upsilon)\cup\Lambda^{(s)}}, \ddot{\mathcal{A}}_{(\Theta\setminus\Upsilon)\cup\Lambda^{(s)}}). \end{aligned} \tag{3.188}$$

The proof of the above 5 theorems goes by induction simultaneously on the size of domain sets. For this, the partial order on domain sets is introduced. For every domain set D we assign a tuple of integers $Size(D) = (|D_{i_1}|, |D_{i_2}|, ..., |D_{i_t}|)$, where $D_{i_1}, ..., D_{i_t}$ is the set of all *different domains* of D ordered by their size starting from the largest one. That is, if for two variables x_i, x_j we have $D_i = D_j$, these domains will be represented by one integer in $Size(D)$, but for different domains $D_i \neq D_j$ such that $|D_i| = |D_j|$ there will be two equal integers. Then the lexicographic order on tuples of integers induces a partial order on domain sets, i.e. we say that $(a_1, ..., a_k) < (b_1, ..., b_l)$ if there exists $j \in \{1, 2, ..., min(k+1, l)\}$ such that $a_i = b_i$ for all $i < j$, and $a_j < b_j$ or $j = k+1$. That is, $(a_1, ..., a_k) < (b_1, ..., b_l)$ in two cases:

- these tuples are of any lengths and there is the first $j < min(k+1, l)$ such that $a_j < b_j$;
- $k < l$ and for all $i \leq k$, $a_i = b_i$.

It follows from the definition that $\leq$ is transitive and there does not exist an infinite descending chain of reductions. Also, duplicating domains does not affect this partial order, so the size of a domain set of any covering of the instance is not larger than the size of a domain set of the instance. If we consider any minimal (proper) one-of-four reduction $D^{(1)}$ of the instance with a domain set $D^{(0)}$, then $Size(D^{(1)}) < Size(D^{(0)})$ since we reduce equal domains simultaneously. Further, we will use the induction on the size of domain sets exclusively either for reductions of an instance or for instance and its (expanded) coverings. We never compare domain sets of totally unrelated instances.

The string induction can be formalized as follows. For every CSP instance Θ with domain set $D = V_{\ddot{\mathcal{A}}}$ we define two new sets $D_{dif,D}$, $D_{size,D}$ (here we suppose that none of the domains $D_0, ..., D_{n-1}$ is empty). First, we need to remove duplicated domains:

$$\begin{aligned} &\forall a < l,\, D_{dif,D}(0, a) \iff D_0(a) \wedge \\ \wedge \forall 0 < i < n, \forall a < l,\, &D_{dif,D}(i, a) \iff (D_i(a) \wedge (\forall j < i \exists a < l, \\ &(D_i(a) \wedge \neg D_{dif,D}(j, a)) \vee (\neg D_i(a) \wedge D_{dif,D}(j, a))). \end{aligned} \tag{3.189}$$

That is, if for any $i < n$ such that there is $j < i$, $D_i = D_j$, we define $D_{dif,D,i}$ to be an empty set. $D_{dif,D}$ exists due to $\Sigma_1^{1,b}$-induction up to n. Based on this set we define $D_{size,D}$ using the census function:

$$\forall i < n, \forall s < l,\ D_{size,D}(i,s) \iff \#D_{dif,D,i} = s. \tag{3.190}$$

Then we can sort a given sequence of natural numbers $D_{size,D}$ using a number function $rank(i, n-1, D_{size,D})$, where $s = rank(i, n-1, D_{size,D})$ is the number that appears at the ith position when $D_{size,D}$ is sorted in non-increasing order (see [4]):

$$\forall i < n, \forall s < l,\ D_{\geq size,D}(i,s) \iff s = rank(i, n-1, D_{size,D}). \tag{3.191}$$

The formalization of the order on domain sets is straightforward (at the end of the string $D_{\geq size,D}$ there could be 0s, but this does not affect the order). We will denote this order between strings by $\leq_{size}$. It is easy to see that if we view a string X as a number $\sum_i X(i)2^i$, then for any two domain sets D, D' such that either Θ' is a minimal reduction of CSP instance Θ or $\Theta' \in ExpCov(\Theta)$,

$$D_{\geq size,D'} \leq_{size} D_{\geq size,D} \implies \sum_i D_{\geq size,D'}(i)2^i \leq \sum_i D_{\geq size,D}(i)2^i.$$

The problem can arise only if we compare domain sets of two totally unrelated instances (for example, $D_{\geq size,D}(x) \iff x = \langle 0, l\rangle$ (all domains are A, $|A| = l$) and $D_{\geq size,D'}(i, l-1)$ for all $i < n$), but we never do. Thus, here we can use order on strings (viewed as the binary representation of numbers), successor function, and string minimization axiom (see [4]).

It turns out that one can reduce string induction in this case to number induction. We again consider sets $D_{dif,D}$ and $D_{size,D}$. Since we work in a fixed algebra $\mathbb{A} = (A, \Omega)$ of size l, there are $k_0 \leq t$ domains of size 0, $k_1 \leq t$ domains of size 1, $k_2 \leq t$ domains of size 2,...,$k_l \leq t$ domains of size l, with $k_0 + ... + k_l = t \leq n$, where t is the number of different domains of the instance,

$$t = \#D_{dif,D}.$$

Then let us define l sets, $K_1, K_2, ..., K_l$ in the following way:

$$\begin{aligned} K_s(0,0) \wedge \forall 0 < i < n, \forall r < t,\ (K_s(i-1,r) \wedge D_{\geq size,D}(i,s) \to K_s(i,r+1)) \wedge \\ \wedge (K_s(i-1,r) \wedge \neg D_{\geq size,D}(i,s) \to K_s(i,r)). \end{aligned} \tag{3.192}$$

Such sets exist due to $\Sigma_1^{1,b}$-induction. Define $k_0 := 0$ and $k_s = seq(n-1, K_s)$ for every $0 < s \leq l$. Then the tuple

$$size(D) = \langle k_l, k_{l-1}, ..., k_0\rangle = \langle ...\langle\langle k_l, k_{l-1}\rangle, k_{l-2}\rangle, ..., k_0\rangle < (n(l+1)+1)^{2^{l+1}} \tag{3.193}$$

codes the size of the domain set D by one integer.

It is easy to see that for any $\Theta' \in ExpCov(\Theta)$, $\langle k'_l, k'_{l-1}, ..., k'_0\rangle \leq \langle k_l, k_{l-1}, ..., k_0\rangle$. Consider a minimal reduction D' of D, suppose we reduced one domain D_{i_j} from the list $D_{i_1}, ..., D_{i_t}$ of the size q to the size p. Thus, the tuple coding the size of D' is

$$\langle k_l, ..., k_q - 1, ..., k_p + 1, ..., k_0\rangle.$$

Recall the ordering property of the pairing function

$$\langle x_1, x_2\rangle < \langle y_1, y_2\rangle \iff x_1 + x_2 < y_1 + y_2 \vee x_1 + x_2 = y_1 + y_2 \wedge x_2 < y_2. \tag{3.194}$$

Since we never consider the trivial case with domains of size 1, there must be at least three integers, $\langle k_2, k_1, k_0\rangle$, and we never decrease k_1. Thus, for every reduction D' we have the following situation:

$$\langle ...\langle\langle ...\langle a, k_q - 1\rangle, ...\rangle, k_p + 1, ..., \rangle k_0\rangle$$

for some $a \geq 0, k_q > 1$. Since

$$\langle a, k_q\rangle - \langle a, k_q - 1\rangle = a + 1 + k_q,$$

for any $b > \langle 0, 1\rangle$ and any $b > c \geq 0$ we have $\langle c, k_p + 1\rangle < \langle b, k_p\rangle$. It follows that for any two domain sets D, D' such that either Θ' is a minimal reduction of CSP instance Θ or $\Theta' \in ExpCov(\Theta)$

$$\langle k'_l, k'_{l-1}, ..., k'_0\rangle < \langle k_l, k_{l-1}, ..., k_0\rangle.$$

Thus, we can use the standard Number induction axiom available in W_1^1.

Lemma 64. *For any CSP instance Θ, induction on size(D) follows in W_1^1.*

The proof of all theorems goes simultaneously by the induction on the size of the domain sets. We assume that formulas T9.5, T9.6, T9.7 hold on instances Θ with a domain set $D^{(0)}$ if $Size(D^{(0)}) < Size(D^{(\perp)})$, and formulas T9.8 and T9.9 hold on instances Ψ with a domain set $D^{(s)}$ if $Size(D^{(s)}) < Size(D^{(\perp)})$. The induction step proves formulas T9.5, T9.6, T9.7 on instances Θ with a domain set $D^{(0)}$ if $Size(D^{(0)}) = Size(D^{(\perp)})$, and formulas T9.8 and T9.9 on instances Ψ with a domain set $D^{(s)}$ if $Size(D^{(s)}) = Size(D^{(\perp)})$. Consider the following $\Sigma_1^{\mathscr{B}}$-formula ϕ:

$$\begin{aligned}\phi(t) := T.9.5(\Theta, D^{(1)}_{\Theta}, \Lambda, X) \wedge T9.6(\Theta, D^{(1)}_{\Theta}) \wedge T9.7(\Theta, D^{(1)}_{\Theta}) \wedge \\ \wedge T9.8(\Psi, \Psi_{Str}) \wedge T9.9(\Psi, \Psi_{Str}, \Upsilon, X) \wedge \\ \wedge size(D^{(1)}_{\Theta}) = t \wedge size(D^{(s)}_{\Psi}) = t.\end{aligned} \tag{3.195}$$

With the application of the Strong Induction Scheme,

$$\forall x((\forall t < x\phi(t)) \rightarrow \phi(x)) \rightarrow \forall z\phi(z), \tag{3.196}$$

we can formulate the following result.

Theorem 38. *Theory W_1^1 proves Theorems 33, 34, 35, 36 and 37.*

It follows immediately from Theorem 35 that W_1^1 proves three universal algebra axiom schemes.

Theorem 39. *For any fixed relational structure $\mathcal{A}$ which corresponds to an algebra with WNU operation and therefore leads to a p-time-solvable CSP, the theory W_1^1 proves universal algebra axiom schemes $BA_{\mathcal{A}}$-axioms, $CR_{\mathcal{A}}$-axioms, and $PC_{\mathcal{A}}$-axioms.*

This, together with Theorem 27, proves Theorem 1:

Theorem 1 (The main theorem)**.** *For any particular relational structure $\mathcal{A}$ such that CSP($\mathcal{A}$) is in P:*

1. *Theory W_1^1 proves the soundness of Zhuk's algorithm. That is, the theory proves the formula $Reject_{\mathcal{A}}(\mathcal{X}, W) \implies \neg HOM(\mathcal{X}, \mathcal{A})$.*
2. *There exists a p-time algorithm F such that for any unsatisfiable instance $\mathcal{X}$, i.e. such that $\neg HOM(\mathcal{X}, \mathcal{A})$, the output $F(\mathcal{X})$ of F on $\mathcal{X}$ is a propositional proof of the proposition translation of formula $\neg HOM(\mathcal{X}, \mathcal{A})$ in propositional calculus G.*

3.4 Closing notes

In this chapter, we have proved three universal algebra axiom schemes $\mathrm{BA}_{\mathcal{A}}$-axioms, $\mathrm{CR}_{\mathcal{A}}$-axioms, and $\mathrm{PC}_{\mathcal{A}}$-axioms in the theory of bounded arithmetic W^1_1. In Section 3.2 we first formalized in the third-sorted setting all the universal algebra notions used in the proof and then, in Section 3.3, showed the formalization of the proofs of theorems and lemmas themselves, concentrating on the key statements and arguments in [15].

We did not treat those statements whose formalization is straightforward by literally translating the original notions into bounded arithmetic language: the formalization of proofs would just exactly repeat the universal algebra reasoning of Zhuk's paper (although we have considered a few examples of those too). Also, we did not consider proofs that require nothing that thorough and tedious formalization of all tiny details in V^1, though they require a lot of imagination from a universal algebra point of view. Our goal was not to mechanically rewrite all the proofs in the language of the theory of bounded arithmetic. Instead, we wanted to clearly show the idea of formalization and treat notions and statements whose formalization needs some additional idea.

As far as we were able, we tried to keep the formalization, even for exponentially large objects, in the second-sorted level, working with definitions of the objects rather than with the objects themselves. For this, we used some tricks and simplifications that were allowed by the fact that although some universal algebraic statements hold in general, we needed to treat only the special instances applied in [15], i.e. related to the objects formed from bottom to top from the domains for variables and constraint relations. If we could stay in the second-order setup all the way, our formalization would stay in theory V^1. However, eventually, we still were forced to use third-order objects since even elementary (and seems to be unavoidable) from a universal algebra point of view reasoning about factor algebras when algebra is not a constant product requires exponential size.

In retrospect, our initial aim to provide short propositional proofs of the soundness of the algorithm for the general p-time CSPs in the Extended Frege proof system, corresponding to propositional reasoning, seems to be beyond the formalization of Zhuk's proof. It is likely that to achieve it, one would need to make changes in the level of the proof itself before the formalization. Finding a formalization in a weaker theory than W^1_1, possibly in V^1 itself, is in our view the most interesting avenue for further research.

Bibliography

[1] 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.

[2] Zarathustra Brady. Notes on csps and polymorphisms. *ArXiv*, abs/2210.07383, 2022.

[3] Samuel R. Buss. Bounded arithmetic and propositional proof complexity. In Helmut Schwichtenberg, editor, *Logic of Computation*, pages 67–121, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg.

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

[5] Azza Gaysin. H-colouring dichotomy in proof complexity. *Journal of Logic and Computation*, 31(5):1206–1225, 2021.

[6] Azza Gaysin. Proof complexity of csp, submitted, arxiv: https://arxiv.org/abs/2201.00913, 2023.

[7] M. Istinger and H.K Kaiser. A characterization of polynomially complete algebras. *Journal of Algebra*, 56(1):103–110, 1979.

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

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

[10] Alan Skelley. A third-order bounded arithmetic theory for pspace. In Jerzy Marcinkowski and Andrzej Tarlecki, editors, *Computer Science Logic*, pages 340–354, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[11] Alan Ramsay Skelley. *Theories and Proof Systems for Pspace and the Exp-Time Hierarchy*. PhD thesis, University of Toronto, Computer Center Toronto, Ont. M5S 1A1, Canada, 2006. AAINR15761.

[12] Michael Soltys and Stephen Cook. The proof complexity of linear algebra. *Annals of Pure and Applied Logic*, 130(1):277–323, 2004. Papers presented at the 2002 IEEE Symposium on Logic in Computer Science (LICS).

[13] Neil Thapen and Michael Soltys. Weak theories of linear algebra. *Archive for Mathematical Logic*, 44(2):195–208, 2005.

[14] Dmitriy Zhuk. Key (critical) relations preserved by a weak near-unanimity function. *Algebra Univers*, 77:191–235, 2017.

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

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

Conclusion

We have shown that Zhuk's algorithm, solving any tractable CSP($\mathcal{A}$) in polynomial time, may be augmented so that it also provides independent witnesses – propositional proofs – for negative answers. Witnesses of the non-existence of a solution, i.e. the non-existence of a homomorphism for relational structures, are proofs in the propositional proof system G. To get an even more transparent proof system, e.g. the Extended Frege system (which is equivalent to the usual textbook Hilbert calculus with the substitution rule), one would need to formalize the soundness of the algorithm in a weaker theory, e.g. in V^1.

The most interesting open question is whether the formalization of the algorithm in a weaker theory of bounded arithmetic is possible, and whether it can be done without changes in the level of Zhuk's proof of the CSP dichotomy theorem itself. Another question is whether the restriction of the types of algebras considered within the algorithm would lead to some weakening of the theory. Bulatov [1] uses different methods of universal algebra to prove the dichotomy. The problem of formalizing Bulatov's algorithm in theory of bounded arithmetic is another open problem of particular interest.

The bounded formulas involved in the formalization are sometimes quite long. It may be that a formalization that does not use formal arithmetic, but rather one of the modern (computer-oriented) systems for formalization, namely proof assistants, such as Lean or Isabelle, would be more suitable for this. However, the link between these systems and propositional logic is currently missing. We think this could be another interesting avenue for future research.

Bibliography

[1] 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.

List of Publications

[1] Azza Gaysin, *Proof complexity of CSP*, submitted (2022).

[2] Azza Gaysin, *$\mathcal{H}$-colouring dichotomy in proof complexity*, Journal of Logic and Computation, 31(5):1206–1225, (2021).

[3] Azza Gaysin, Mikhail V. Volkov, *Block-Groups and Hall Relations*, Semigroups, Categories, and Partial Algebras, 25–32. Springer Singapore, (2021).