$\mathcal{H}$-colouring dichotomy in proof complexity

AZZA GAYSIN, *Department of Algebra, Faculty of Mathematics and Physics, Charles University, 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 (CSPs): 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 (1990, *J. Comb. Theory Ser. B*, 48, 92–110) (an analogous theorem for all CSPs was recently proved by Zhuk (2020, *J. ACM*, 67, 1–78) and Bulatov (2017, *FOCS*, 58, 319–330)), 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 for such $\mathcal{H}$ have polynomial proofs in 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 by 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 a subsequent work—to formalize in such a theory the soundness of Zhuk's algorithm, extending the upper bound proved here from undirected simple graphs to the general case of directed graphs in some logical calculi.

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 CSP associated 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 [7] in 1998 and recently proved by Zhuk [14] and Bulatov [2].

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 1.1
(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.

Vol. 31, No. 5,
Advance Access Publication on 24 April 2021 https://doi.org/10.1093/logcom/exab028

LEMMA 1.2
([9]).
For all graphs $\mathcal{G}, \mathcal{H}$ if $\mathcal{H}$ is bipartite, then $\mathcal{G}$ is $\mathcal{H}$-colourable if and only if $\mathcal{G}$ is a 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 2.2). 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 every time we consider bipartite graph $\mathcal{H}$ and non-bipartite graph $\mathcal{G}$). From this point of view, it is natural to ask about the proof complexity of those instances. Acommon 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 is, the weaker propositional proof system will be.

If $\mathcal{H}$-colouring is *NP*-complete, then the negative instances (graphs $\mathcal{G}$ that cannot be $\mathcal{H}$-coloured) 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; Section 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. The tautologies expressing the negative instances for such $\mathcal{H}$ hence have polynomial proofs in 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 [2, 14]) 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 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 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 2, we give some common definitions from propositional proof complexity and theory of bounded arithmetic and the definition of CSP in terms of homomorphisms, and explain how to express instances of CSP by propositional formulas. In Section 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 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 4, we consider *NP*-complete case of the $\mathcal{H}$-colouring problem and known lower bounds for one suitable example. In Section 5, we discuss open questions and further direction of research.

2 Preliminaries

2.1 CSPs and the $\mathcal{H}$-colouring problem

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

DEFINITION 2.1
(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 $\mathcal{H} = (H, R_1^{\mathcal{H}}, ..., R_n^{\mathcal{H}})$ s.t. H is 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 s.t., 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 *CSP* 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 [3].

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 colours. An $\mathcal{H}$-colouring of a simple undirected graph $\mathcal{G} = (V_{\mathcal{G}}, E_{\mathcal{G}})$ without loops is an assignment of colours to the vertices of $\mathcal{G}$ such that adjacent vertices of $\mathcal{G}$ obtain adjacent colours. 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 $\mathcal{G} \to \mathcal{H}$. 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 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 2.2
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 every 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.

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 2.3
($\mathcal{L}^2{}_{\mathcal{PA}}$).
$\mathcal{L}^2{}_{\mathcal{PA}} = \{0, 1, +, \cdot, |\,|; =_1, =_2, \leq, \in\}$

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 as 2-*BASIC* [5], which we do not present here.

NOTATION 2.4
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 i-th bit of 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 2.5
(Bounded formulas).
Let $\mathcal{L}$ be a two-sorted vocabulary. If x is a number variable, X is a string variable that do not occur in the $\mathcal{L}$-number term t, then $\exists x \le t\phi$ stands for $\exists x(x \le t \wedge \phi)$, $\forall x \le t\phi$ stands for $\forall x(x \le t \to \phi)$, $\exists X \le t\phi$ stands for $\exists X(|X| \le t \wedge \phi)$ and $\forall X \le t\phi$ stands for $\forall X(|X| \le t \to \phi)$. Quantifiers that occur in this form are said to be *bounded* and a *bounded formula* is one in which every quantifier is bounded.

NOTATION 2.6
We will use the following abbreviations: $\exists\bar{x} \le \bar{t}\phi$ stands for $\exists x_1 \le t_1, ..., \exists x_k \le t_k\phi$ for some k, where no x_i occurs in any t_j (even if $i < j$). Similarly for $\forall\bar{x} \le \bar{t}$, $\exists\bar{X} \le \bar{t}$, $\forall\bar{X} \le \bar{t}$.

DEFINITION 2.7
(Σ_i^B and Π_i^B formulas in $\mathcal{L}^2{}_{\mathcal{PA}}$).
We will define Σ_i^B and Π_i^B formulas recursively as follows:

- $\Sigma_0^B = \Pi_0^B$ is the set of $\mathcal{L}^2{}_{\mathcal{PA}}$-formulas whose only quantifiers are bounded number quantifiers (there can be free string variables);
- for $i \ge 0$, Σ_{i+1}^B (resp. Π_{i+1}^B) is the set of formulas of the form $\exists\bar{X} \le \bar{t}\phi(\bar{X})$ (resp. $\forall\bar{X} \le \bar{t}\phi(\bar{X})$), where ϕ is a Π_i^B formula (resp. Σ_i^B formula), and $\bar{t}$ is a sequence of $\mathcal{L}^2{}_{\mathcal{PA}}$-terms not involving any variable from $\bar{X}$.

DEFINITION 2.8
(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

$$\exists X \le y \forall z < y(X(z) \longleftrightarrow \phi(z)), \tag{1}$$

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 2.9
(V^0).
The theory V^0 has the vocabulary $\mathcal{L}^2{}_{\mathcal{PA}}$ and is axiomatized by 2-*BASIC* and Σ_0^B-*COMP*.

There is no explicit induction axiom scheme in V^0, but it is known [4] that $V^0 \vdash \Sigma_0^B$-*IND*, where Φ-*IND* is defined as follows.

DEFINITION 2.10
(Number induction axiom).
If Φ is a set of two-sorted formulas, then Φ-*IND* axioms are the formulas

$$(\phi(0) \wedge \forall x(\phi(x) \to \phi(x+1))) \to \forall z\phi(z), \tag{2}$$

where ϕ is a formula in Φ.

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] and [12].

DEFINITION 2.11
(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 $C_2 \cup \{\neg p\}$:

$$\frac{C_1 \cup \{p\} \quad C_2 \cup \{\neg p\}}{C_1 \cup C_2}. \tag{3}$$

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 the formula $\alpha = \bigvee_{i=1}^{n} \neg C_i$ is a tautology.

DEFINITION 2.12
(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:

$$\frac{C \cup \{\bigwedge_j l_j\} \quad D \cup \{\neg l'_1, ..., \neg l'_t\}}{C \cup D} \tag{4}$$

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

$$\frac{C \cup \{\bigwedge_{j \leq s} l_j\} \quad D \cup \{\bigwedge_{s<j\leq t} l_j\}}{C \cup D \cup \{\bigwedge_{i\leq s+t} l_i\}}. \tag{5}$$

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 (that is 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 the size up to $log(s)$.

DEFINITION 2.13
(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 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, e.g. R^* and $R^*(log)$.

DEFINITION 2.14
(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 2.15
(5.7.2 in [12]).
R p-simulates $R^*(log)$ with respect to refutations of sets of clauses.

We also introduce Definition 2.16, which we will use at the end of Section 3.3.

DEFINITION 2.16
(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, applying any number of conjunctions and bounded universal quantifiers;
- then, applying any number of disjunctions and bounded existential quantifiers.

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

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 3.1
(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

$$R(\bar{x},\bar{X}) \longleftrightarrow \phi(\bar{x},\bar{X}). \tag{6}$$

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 3.2
(Definable number functions).
Let T be a theory with 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

$$T \vdash \forall \bar{x} \forall \bar{X}\, \exists!\, y\, \phi(\bar{x}, y, \bar{X}) \tag{7}$$

and

$$y = f(\bar{x}, \bar{X}) \longleftrightarrow \phi(\bar{x}, y, \bar{X}). \tag{8}$$

Auxiliary predicate and function symbols, which we will use further to define different notions in V^0, are as follows.

DEFINITION 3.3
(Divisibility).
The relation of *divisibility* is defined by

$$x|y \longleftrightarrow \exists z \leq y(x \cdot z = y). \tag{9}$$

DEFINITION 3.4
(Pairing function).
If $x, y \in \mathbb{N}$, we define the *pairing function* $\langle x, y \rangle$ to be the following term in V^0:

$$\langle x, y \rangle = (x + y)(x + y + 1) + 2y. \tag{10}$$

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 (7). It is also easy to prove in V^0 that the pairing function is a one–one function, i.e.

$$V^0 \vdash \forall x_1, x_2, y_1, y_2\ \langle x_1, y_1 \rangle = \langle x_2, y_2 \rangle \to x_1 = x_2 \wedge y_1 = y_2. \tag{11}$$

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 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 follows: 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 3.5
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 3.6
(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:

$$\begin{aligned} GRAPH(V_{\mathcal{G}}, E_{\mathcal{G}}) \longleftrightarrow \forall i < n(V_{\mathcal{G}}(i)) \wedge \forall i < j < n \\ (E_{\mathcal{G}}(i,j) \longleftrightarrow E_{\mathcal{G}}(j,i)) \wedge \forall i < n \neg(E_{\mathcal{G}}(i,i)). \end{aligned} \tag{12}$$

Further, 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 i-th vertex is mapped to j-th vertex. For Z to be a well-defined map, it should satisfy the following Σ_0^B-definable relation $MAP(n,m,Z)$.

DEFINITION 3.7
(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

$$\begin{aligned} MAP(n,m,Z) \longleftrightarrow \forall i < n \exists j < m\, Z(\langle i,j\rangle) \wedge \\ \forall i < n \forall j_1, j_2 < m(Z(\langle i,j_1\rangle) \wedge Z(\langle i,j_2\rangle) \to j_1 = j_2). \end{aligned} \tag{13}$$

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 3.8
(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

$$\begin{aligned} HOM(\mathcal{G}, \mathcal{H}) \longleftrightarrow \exists Z \leq \langle n-1, m-1\rangle \big(MAP(n,m,Z) \wedge \\ \forall i_1, i_2 < n, \forall j_1, j_2 < m \\ (E_{\mathcal{G}}(i_1, i_2) \wedge Z(\langle i_1, j_1\rangle) \wedge Z(\langle i_2, j_2\rangle) \to E_{\mathcal{H}}(j_1, j_2))\big). \end{aligned} \tag{14}$$

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 3.9
(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

$$\begin{aligned} BIP(\mathcal{H}) \longleftrightarrow \exists W_{\mathcal{H}}, U_{\mathcal{H}} \leq m\big(\forall i < m(W_{\mathcal{H}}(i) \leftrightarrow \neg U_{\mathcal{H}}(i)) \wedge \\ \forall i < j < m(E_{\mathcal{H}}(i,j) \rightarrow (W_{\mathcal{H}}(i) \wedge U_{\mathcal{H}}(j)) \vee (W_{\mathcal{H}}(j) \wedge U_{\mathcal{H}}(i)))\big). \end{aligned} \tag{15}$$

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 be Π_1^B and hence translatable into propositional logic. First, we define a cycle.

DEFINITION 3.10
(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

$$\begin{aligned} CYCLE(\mathcal{C}_k) \longleftrightarrow E_{\mathcal{C}_k}(0, k-1) \wedge \forall i < (k-1)\, E_{\mathcal{C}_k}(i, i+1) \wedge \\ \forall i,j < (k-1)(j \neq i+1 \rightarrow \neg E_{\mathcal{C}_k}(i,j)). \end{aligned} \tag{16}$$

DEFINITION 3.11
(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

$$\begin{aligned} NONBIP(\mathcal{G}) \longleftrightarrow \exists k \leq n(2|(k-1))\exists V_{\mathcal{C}_k} = k, \exists E_{\mathcal{C}_k} < 4k^2 \\ CYCLE(V_{\mathcal{C}_k}, E_{\mathcal{C}_k}) \wedge HOM(\mathcal{C}_k, \mathcal{G}). \end{aligned} \tag{17}$$

3.2 Proving in theory V^0

LEMMA 3.12
(Homomorphism transitivity).
For all graphs $\mathcal{G}, \mathcal{H}, \mathcal{S}$, V^0 proves the property of a homomorphism to be transitive:

$$V^0 \vdash \forall \mathcal{G}, \mathcal{H}, \mathcal{S}\, (HOM(\mathcal{G}, \mathcal{H}) \wedge HOM(\mathcal{H}, \mathcal{S}) \rightarrow HOM(\mathcal{G}, \mathcal{S})). \tag{18}$$

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 $Z' \leq \langle m-1, t-1\rangle$ which satisfy the homomorphism definition. We need to prove that there exists a set $Z'' \leq \langle n-1, t-1\rangle$, such that

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

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

$$Z''(\langle i, k\rangle) \longleftrightarrow \exists j < m(Z(\langle i, j\rangle) \wedge Z'(\langle j, k\rangle)). \tag{19}$$

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 $\mathcal{S}$. $\square$

NOTATION 3.13
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 3.14
For all bipartite graphs, $\mathcal{H}$, V^0 proves the existence of a homomorphism from $\mathcal{H}$ to $\mathcal{K}_2$:

$$V^0 \vdash \forall \mathcal{H}\, (BIP(\mathcal{H}) \to HOM(\mathcal{H},\mathcal{K}_2)). \tag{20}$$

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, then 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 the 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 $\mathcal{K}_2$. $\square$

LEMMA 3.15
For all non-bipartite graphs, $\mathcal{G}$, V^0 proves that there is no homomorphism from $\mathcal{G}$ to $\mathcal{K}_2$:

$$V^0 \vdash \forall \mathcal{G}\ (NONBIP(\mathcal{G}) \to \neg HOM(\mathcal{G},\mathcal{K}_2)). \tag{21}$$

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 3.12 by transitivity there also exists 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 $\mathcal{K}_2$, then 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 letus prove that $Z(\langle k-1,0\rangle)$ too. Since $2|(k-1)$, then $k > 2$. 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

$$\phi(i,Z) = (2|i \to Z(\langle i,0\rangle)) \wedge (2 \nmid i \to Z(\langle i,1\rangle)). \tag{22}$$

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*:

$$(\phi(0,Z) \wedge \forall i < k(\phi(i,Z) \to \phi(i+1,Z)) \to \forall j < k\, \phi(j,Z). \tag{23}$$

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 the 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 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 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 3.16
(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 $\mathcal{H}$:

$$V^0 \vdash \forall \mathcal{G}, \mathcal{H}(BIP(\mathcal{H}) \wedge NONBIP(\mathcal{G}) \to \neg HOM(\mathcal{G},\mathcal{H})). \tag{24}$$

PROOF. Suppose that there exists a homomorphism from $\mathcal{G}$ to $\mathcal{H}$. According to Lemma 3.14, since $\mathcal{H}$ is bipartite, then there exists a homomorphism from $\mathcal{H}$ to K_2. Thus, due to Lemma 3.12, by the transitivity, there exists a homomorphism from $\mathcal{G}$ to K_2. But this is a contradiction with Lemma 3.15. □

3.3 Translating into tautologies

3.3.1 Translation of the main universal statement In this section, we proceed with 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]:

$$||\phi(\bar{x},\bar{X})|| = \{\phi(\bar{x},\bar{X})[\bar{m},\bar{n}] : \bar{m},\bar{n} \in \mathbb{N}\}. \tag{25}$$

LEMMA 3.17
([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 3.18
(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 which 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 3.16 which expresses that there is no homomorphism from a non-bipartite graph $\mathcal{G}$ to a bipartite graph $\mathcal{H}$:

$$\begin{aligned}\phi(\mathcal{G},\mathcal{H}) = \neg GRAPH(\mathcal{G}) \vee \neg GRAPH(\mathcal{H}) \vee \\ \neg BIP(\mathcal{H}) \vee \neg NONBIP(\mathcal{G}) \vee \neg HOM(\mathcal{G},\mathcal{H}).\end{aligned} \tag{26}$$

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:

$$\phi(V_{\mathcal{G}}, E_{\mathcal{G}}, V_{\mathcal{H}}, E_{\mathcal{H}}) =$$

$$\exists i < n \neg V_{\mathcal{G}}(i) \vee \exists i < j < n((\neg E_{\mathcal{G}}(i,j) \vee \neg E_{\mathcal{G}}(j,i)) \tag{I}$$
$$\wedge (E_{\mathcal{G}}(i,j) \vee E_{\mathcal{G}}(j,i))) \vee \exists i < n\, E_{\mathcal{G}}(i,i)$$
$$\vee$$
$$\exists i < m \neg V_{\mathcal{H}}(i) \vee \exists i < j < m((\neg E_{\mathcal{H}}(i,j) \vee \neg E_{\mathcal{H}}(j,i)) \tag{II}$$
$$\wedge (E_{\mathcal{H}}(i,j) \vee E_{\mathcal{H}}(j,i))) \vee \exists i < n\, E_{\mathcal{H}}(i,i)$$
$$\vee$$
$$\forall W_{\mathcal{H}}, U_{\mathcal{H}} \leq m\big(\exists i < m\, ((\neg W_{\mathcal{H}}(i) \vee U_{\mathcal{H}}(i)) \wedge (W_{\mathcal{H}}(i) \vee \neg U_{\mathcal{H}}(i)))\vee \tag{III}$$
$$\exists i < j < m(E_{\mathcal{H}}(i,j) \wedge (\neg W_{\mathcal{H}}(i) \vee \neg U_{\mathcal{H}}(j)) \wedge (\neg W_{\mathcal{H}}(j) \vee \neg U_{\mathcal{H}}(i)))\big)$$
$$\vee$$
$$\forall k \leq n(2|(k-1))\, \forall V_{\mathcal{C}_k} = k,\, \forall E_{\mathcal{C}_k} < 4k^2\big((\exists i < k \neg V_{\mathcal{C}_k}(i)\vee$$
$$\exists i < j < k((\neg E_{\mathcal{C}_k}(i,j) \vee \neg E_{\mathcal{C}_k}(j,i)) \wedge (E_{\mathcal{C}_k}(i,j) \vee E_{\mathcal{C}_k}(j,i)))\vee$$
$$\exists i < k\, E_{\mathcal{C}_k}(i,i)) \vee (\neg E_{\mathcal{C}_k}(0, k-1) \vee \exists i < (k-1) \tag{IV}$$
$$\neg E_{\mathcal{C}_k}(i, i+1) \vee \exists i,j < (k-1)\, (j \neq i+1 \wedge E_{\mathcal{C}_k}(i,j)))\vee$$
$$(\forall Z \leq \langle k-1, n-1\rangle\, (\neg MAP(k,n,Z) \vee \exists i_1, i_2 < k \exists j_1, j_2 < n$$
$$E_{\mathcal{C}_k}(i_1, i_2) \wedge Z(\langle i_1, j_1\rangle) \wedge Z(\langle i_2, j_2\rangle) \wedge \neg E_{\mathcal{G}}(j_1, j_2))))\big)$$
$$\vee$$
$$\forall Z' \leq \langle n-1, m-1\rangle\big(\neg MAP(n,m,Z') \vee \exists i_1, i_2 < n, \exists j_1, j_2 < m \tag{V}$$
$$(E_{\mathcal{G}}(i_1, i_2) \wedge Z'(\langle i_1, j_1\rangle) \wedge Z'(\langle i_2, j_2\rangle) \wedge \neg E_{\mathcal{H}}(j_1, j_2))\big).$$

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

$$\begin{aligned} \phi(V_{\mathcal{G}}, E_{\mathcal{G}}, V_{\mathcal{H}}, E_{\mathcal{H}}) = \forall W_{\mathcal{H}}, U_{\mathcal{H}} \leq m, \forall V_{\mathcal{C}_k} \leq n, \forall E_{\mathcal{C}_k} \leq 4n^2,\\ \forall Z \leq \langle k-1, n-1\rangle, \forall Z' \leq \langle n-1, m-1\rangle\\ [\psi(n, m, V_{\mathcal{G}}, V_{\mathcal{H}}, W_{\mathcal{H}}, U_{\mathcal{H}}, V_{\mathcal{C}_k}, E_{\mathcal{G}}, E_{\mathcal{H}}, E_{\mathcal{C}_k}, Z, Z')], \end{aligned} \tag{27}$$

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 3.17, 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, (I) and (II), 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. 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 $\mathcal{G}$, (I) translates into

$$[\bigvee_{i=0}^{n-1}(\neg p_i^{V_{\mathcal{G}}})] \vee [\bigvee_{j=0}^{n-1}\bigvee_{i=0}^{j-1}(\neg p_{\langle i,j\rangle}^{E_{\mathcal{G}}} \vee \neg p_{\langle j,i\rangle}^{E_{\mathcal{G}}}) \wedge (p_{\langle i,j\rangle}^{E_{\mathcal{G}}} \vee p_{\langle j,i\rangle}^{E_{\mathcal{G}}})] \vee \\ [\bigvee_{i=0}^{n-1}(p_{\langle i,i\rangle}^{E_{\mathcal{G}}})]. \tag{28}$$

And for graph $\mathcal{H}$, (II) translates into

$$[\bigvee_{i=0}^{m-1}(\neg p_i^{V_{\mathcal{H}}})] \vee [\bigvee_{j=0}^{m-1}\bigvee_{i=0}^{j-1}(\neg p_{\langle i,j\rangle}^{E_{\mathcal{H}}} \vee \neg p_{\langle j,i\rangle}^{E_{\mathcal{H}}}) \wedge (p_{\langle i,j\rangle}^{E_{\mathcal{H}}} \vee p_{\langle j,i\rangle}^{E_{\mathcal{H}}})] \vee \\ [\bigvee_{i=0}^{m-1}(p_{\langle i,i\rangle}^{E_{\mathcal{H}}})]. \tag{29}$$

The third part (III) 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 and free string variables are $W_{\mathcal{H}}, U_{\mathcal{H}}, E_{\mathcal{H}}$. The translation of (III) is

$$[\bigvee_{i=0}^{m-1}(\neg p_i^{W_{\mathcal{H}}} \vee p_i^{U_{\mathcal{H}}}) \wedge (p_i^{W_{\mathcal{H}}} \vee \neg p_i^{U_{\mathcal{H}}})] \vee \\ [\bigvee_{j=0}^{m-1}\bigvee_{i=0}^{j-1} p_{\langle i,j\rangle}^{E_{\mathcal{H}}} \wedge (\neg p_i^{W_{\mathcal{H}}} \vee \neg p_j^{U_{\mathcal{H}}}) \wedge (\neg p_j^{W_{\mathcal{H}}} \vee \neg p_i^{U_{\mathcal{H}}})]. \tag{30}$$

The fourth part (IV) 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

$$[\bigvee_{i=0}^{k-1}(\neg p_i^{V_{\mathcal{C}_k}})] \vee [\bigvee_{j=0}^{k-1}\bigvee_{i=0}^{j-1}(\neg p_{\langle i,j\rangle}^{E_{\mathcal{C}_k}} \vee \neg p_{\langle j,i\rangle}^{E_{\mathcal{C}_k}}) \wedge (p_{\langle i,j\rangle}^{E_{\mathcal{C}_k}} \vee p_{\langle j,i\rangle}^{E_{\mathcal{C}_k}})] \vee \\ [\bigvee_{i=0}^{n-1}(p_{\langle i,i\rangle}^{E_{\mathcal{C}_k}})]. \tag{31}$$

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

$$[\neg p_{\langle 0,k-1\rangle}^{E_{\mathcal{C}_k}}] \vee [\bigvee_{i=0}^{k-2} \neg p_{\langle i,i+1\rangle}^{E_{\mathcal{C}_k}}] \vee [\bigvee_{i=0}^{k-2} \bigvee_{j=0, j\neq i+1}^{k-2} p_{\langle j,i\rangle}^{E_{\mathcal{C}_k}}]. \tag{32}$$

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

$$\begin{aligned}&[\bigvee_{i=0}^{k-1}\bigwedge_{j=0}^{n-1}(\neg p^{Z}_{\langle i,j\rangle})]\vee[\bigvee_{i=0}^{k-1}\bigvee_{j_2=0}^{n-1}\bigvee_{j_1=0,j_1\neq j_2}^{n-1}(p^{Z}_{\langle i,j_1\rangle}\wedge p^{Z}_{\langle i,j_2\rangle})]\vee\\&[\bigvee_{i_1,i_2=0}^{k-1}\bigvee_{j_1,j_2=0}^{n-1}(p^{E_{\mathcal{C}_k}}_{\langle i_1,i_2\rangle}\wedge p^{Z}_{\langle i_1,j_1\rangle}\wedge p^{Z}_{\langle i_2,j_2\rangle}\wedge\neg p^{E_{\mathcal{G}}}_{\langle j_1,j_2\rangle})].\end{aligned}\tag{33}$$

Finally, to get the translation of the whole subformula, we need first to make a disjunction of all formulas (31)–(33) and then make a conjunction on k:

$$\begin{aligned}\bigwedge_{k=3,\,2|(k-1)}^{n-1}\Big[&[\bigvee_{i=0}^{k-1}(\neg p^{V_{\mathcal{C}_k}}_{i})]\vee[\bigvee_{j=0}^{k-1}\bigvee_{i=0}^{j-1}(\neg p^{E_{\mathcal{C}_k}}_{\langle i,j\rangle}\vee\neg p^{E_{\mathcal{C}_k}}_{\langle j,i\rangle})\wedge(p^{E_{\mathcal{C}_k}}_{\langle i,j\rangle}\vee p^{E_{\mathcal{C}_k}}_{\langle j,i\rangle})]\vee\\&[\bigvee_{i=0}^{n-1}(p^{E_{\mathcal{C}_k}}_{\langle i,i\rangle})]\vee[\neg p^{E_{\mathcal{C}_k}}_{\langle 0,k-1\rangle}]\vee[\bigvee_{i=0}^{k-2}\neg p^{E_{\mathcal{C}_k}}_{\langle i,i+1\rangle}]\vee[\bigvee_{i=0}^{k-2}\bigvee_{j=0,j\neq i+1}^{k-2}p^{E_{\mathcal{C}_k}}_{\langle j,i\rangle}]\vee\\&[\bigvee_{i=0}^{k-1}\bigwedge_{j=0}^{n-1}(\neg p^{Z}_{\langle i,j\rangle})]\vee[\bigvee_{i=0}^{k-1}\bigvee_{j_2=0}^{n-1}\bigvee_{j_1=0,j_1\neq j_2}^{n-1}(p^{Z}_{\langle i,j_1\rangle}\wedge p^{Z}_{\langle i,j_2\rangle})]\vee\\&[\bigvee_{i_1,i_2=0}^{k-1}\bigvee_{j_1,j_2=0}^{n-1}(p^{E_{\mathcal{C}_k}}_{\langle i_1,i_2\rangle}\wedge p^{Z}_{\langle i_1,j_1\rangle}\wedge p^{Z}_{\langle i_2,j_2\rangle}\wedge\neg p^{E_{\mathcal{G}}}_{\langle j_1,j_2\rangle})]\Big].\end{aligned}\tag{34}$$

And the fifth part (V) 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

$$\begin{aligned}&[\bigvee_{i=0}^{n-1}\bigwedge_{j=0}^{m-1}(\neg p^{Z'}_{\langle i,j\rangle})]\vee[\bigvee_{i=0}^{n-1}\bigvee_{j_2=0}^{m-1}\bigvee_{j_1=0,j_1\neq j_2}^{m-1}(p^{Z'}_{\langle i,j_1\rangle}\wedge p^{Z'}_{\langle i,j_2\rangle})]\vee\\&[\bigvee_{i_1,i_2=0}^{n-1}\bigvee_{j_1,j_2=0}^{m-1}(p^{E_{\mathcal{G}}}_{\langle i_1,i_2\rangle}\wedge p^{Z'}_{\langle i_1,j_1\rangle}\wedge p^{Z'}_{\langle i_2,j_2\rangle}\wedge\neg p^{E_{\mathcal{H}}}_{\langle j_1,j_2\rangle})].\end{aligned}\tag{35}$$

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 (28)–(35) 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 3.18, 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 2.2. The upper bound stated next is also a consequence of the results in [1,Section 5] that use different methods.

THEOREM 3.19
(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 3.18, the translation of the formula (26) has p-size constant-depth Frege proof. If $\mathcal{G}$ and $\mathcal{H}$ are graphs, then the translations of the first two disjuncts in (26) 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 $U_{\mathcal{H}}$ such that the whole translation of the disjuct $\neg BIP(\mathcal{H})$ becomes false. That is, as before it is a propositional sentence that evaluates to 0. 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 (26) become propositional sentences evaluated to 0 and thus the whole translation of the formula (26) is equivalent to the translation of $\neg HOM(\mathcal{G},\mathcal{H})$. That is, we obtained a polynomial size constant-depth Frege proof of $||\neg HOM(G,H)||$. □

3.3.2 Other remarks Actually, we can improve a little our upper bound result from Section 3.3.1. To reason about graphs, we used a convenient for this purpose set-up of two-sorted theory V^0, including the comprehension axiom. However, actually we can avoid using it in both proofs of Lemmas 3.12 and 3.14. For example, in the proof of Lemma 3.12 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 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 the $I\Sigma_0^{1,b}$-*IND* (where $I\Sigma_0^{1,b}$ denotes the class of $\mathcal{L}^2{}_{\mathcal{PA}}$-formulas with all number quantifiers bounded and with no string quantifiers) when it is needed. Moreover, we can restrict further 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 Induction scheme only once for the Σ_1^b-formula (22) in the proof of Lemma 3.15.

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

THEOREM 3.20
([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 $p(\bar{m},\bar{n})$. Further, there is an algorithm which finds an $R^*(log)$-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 a little the formula $\psi(...)$ in (27) to become DNF$_1$: to transform it to DNF we use limited extension introduced by Tseitin and to remove all existential quantifiers after universal ones we use Herbrandization (i.e. Skolemization of the negation; see [12, Section 13.2]). 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 $R^*(log)$ 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 3.15, gives us a collateral result. The Π_1^B-formula (21)

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

expressing that there is no homomorphism from a non-bipartite graph $\mathcal{G}$ to 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 (21) means that the sets of bipartite and non-bipartite graphs are disjoint, since we can define a bipartite graph $\mathcal{H}$ as

$$BIP(\mathcal{H}) \longleftrightarrow HOM(\mathcal{H}, \mathcal{K}_2). \tag{36}$$

We know that resolution R p-simulates $R^*(log)$ system (see Lemma 2.15). Thus, due to the feasible interpolation Theorem 3.21, 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 3.21
(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 $O(kn)$.

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 4.1– 4.4: the reader can find the definitions in [8, 10, 11, 13] 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 CSP($\mathcal{K}_n$) is the graph $\mathcal{K}_{n+1}$: it is impossible to n-colour complete graph with $n+1$ vertices. 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 map injectively the set $[0, n+1]$ to the set $[0, n]$. The PHP_n^{n+1} formula is

$$\neg[\bigwedge_i \bigvee_j p_{ij} \wedge \bigwedge_i \bigwedge_{j \neq j'} (\neg p_{ij} \vee \neg p_{ij'}) \wedge \bigwedge_{i \neq i'} \bigwedge_j (\neg p_{ij} \vee \neg p_{i'j})], \tag{37}$$

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 4.1
([8]).
There exists a constant c, $c > 1$, so that, for sufficiently large n, every resolution refutation of $\neg\mathrm{PHP}_n^{n+1}$ contains at least c^n different clauses.

THEOREM 4.2
(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 can also consider weak variants of PHP principle, PHP_n^m, where the number m of pigeons is larger then $n+1$ (which will be equivalent to non-existence of homomorphism from $\mathcal{K}_m$ to $\mathcal{K}_n$).

THEOREM 4.3
([13]).
For $m > n$ PHP_n^m has no polynomial calculus refutation of degree $d \leq \lceil n/2 \rceil$.

THEOREM 4.4
([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, 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.

5 Conclusion

We have constructed in Section 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 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 condition 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 used bounded arithmetic here as a common framework. Moreover, this framework generally allows 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 [14] algorithm 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 which we left out is to prove a lower bound not just for a suitable $\mathcal{H}$ (as we did in Section 4) but for all $\mathcal{H}$ which 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 $\mathcal{G} \notin$ 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 here this avenue of research.

Funding

This work was partly supported by the Specific university research project SVV-2020-260589 and by Charles University Research Centre program (UNCE/SCI/022).

Acknowledgements

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

References

[1] A. Atserias and J. Ochremiak. Proof complexity meets algebra. *ACM Transactions on Computational Logic*, **20**, 1–46, 2018.

[2] A. A. Bulatov. A dichotomy theorem for nonuniform CSPS. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 319–330. IEEE (Institute of Electrical and Electronics Engineers), 2017.

[3] A. A. Bulatov. H-Coloring dichotomy revisited. *Theoretical Computer Science*, **349**, 31–39, 2005. Graph Colorings.

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

[5] S. A. Cook and P. Nguyen. *Logical Foundations of Proof Complexity*, 1st edn. Cambridge University Press, USA, 2010.

[6] S. A. Cook and R. A. Reckhow. The relative efficiency of propositional proof systems. *Journal of Symbolic Logic*, **44**, 36–50, 1979.

[7] T. Feder and M. Y. Vardi. Monotone monadic SNP and constraint satisfaction. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing, STOC '93*, pp. 612–622. Association for Computing Machinery, New York, NY, USA, 1993.

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

[9] P. Hell and J. Nešetřil. On the complexity of h-coloring. *Journal of Combinatorial Theory Series B*, **48**, 92–110, 1990.

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

[11] J. Krajiček. Lower bounds for a proof system with an exponential speed-up over constant-depth Frege systems and over polynomial calculus. In *Mathematical Foundations of Computer Science 1997*, I. Prívara and P. Ružička, eds, pp. 85–90. Springer, Berlin, Heidelberg, 1997.

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

[13] A. A. Razborov. Lower bounds for the polynomial calculus. *Computational Complexity*, **7**, 291–324, 1998.

[14] D. Zhuk. A proof of the CSP dichotomy conjecture. *Journal of the ACM*, **67**, 1–78, 2020.

Received 9 October 2020