

FORMALIZING RANDOMIZED MATCHING ALGORITHMS

DAI TRI MAN LÊ AND STEPHEN A. COOK

Department of Computer Science, University of Toronto
e-mail address: {ledt, sacook}@cs.toronto.edu

ABSTRACT. Using Jeřábek’s framework for probabilistic reasoning, we formalize the correctness of two fundamental RNC^2 algorithms for bipartite perfect matching within the theory VPV for polytime reasoning. The first algorithm is for testing if a bipartite graph has a perfect matching, and is based on the Schwartz-Zippel Lemma for polynomial identity testing applied to the Edmonds polynomial of the graph. The second algorithm, due to Mulmuley, Vazirani and Vazirani, is for finding a perfect matching, where the key ingredient of this algorithm is the Isolating Lemma.

1. INTRODUCTION

There is a substantial literature on theories such as PV , S_2^1 , VPV , V^1 which capture polynomial time reasoning [8, 4, 17, 7]. These theories prove the existence of polynomial time functions, and in many cases they can prove properties of these functions that assert the correctness of the algorithms computing these functions. But in general these theories cannot prove the existence of probabilistic polynomial time relations such as those in ZPP , RP , BPP because defining the relevant probabilities involves defining cardinalities of exponentially large sets. Of course stronger theories, those which can define $\#\text{P}$ or PSPACE functions can treat these probabilities, but such theories are too powerful to capture the spirit of feasible reasoning.

Note that we cannot hope to find theories that exactly capture probabilistic complexity classes such as ZPP , RP , BPP because these are ‘semantic’ classes which we suppose are not recursively enumerable (cf. [30]). Nevertheless there has been significant progress toward developing tools in weak theories that might be used to describe some of the algorithms in these classes.

Paris, Wilkie and Woods [25] and Pudlák [26] observed that we can simulate *approximate counting* in bounded arithmetic by applying variants of the weak pigeonhole principle. It seems unlikely that any of these variants can be proven in the theories for polynomial time, but they can be proven in Buss’s theory S_2 for the polynomial hierarchy. The first connection between the weak pigeonhole principle and randomized algorithms was noticed by Wilkie (cf. [17]), who showed that randomized polytime functions witness Σ_1^b -consequences

1998 ACM Subject Classification: F.2.2, F.4.1.

Key words and phrases: bounded arithmetic, computational complexity, weak pigeonhole principle, probabilistic reasoning, randomized algorithms, polynomial identity testing.

of $S_2^1 + \text{sWPHP}(PV)$ (i.e., Σ_1^B -consequences of $V^1 + \text{sWPHP}(\mathcal{L}_{\text{FP}})$ in our two-sorted framework), where $\text{sWPHP}(PV)$ denotes the surjective weak pigeonhole principle for all PV functions (i.e. polytime functions).

Building on these early results, Jeřábek [13] showed that we can “compare” the sizes of two bounded P/poly definable sets within VPV by constructing a surjective mapping from one set to another. Using this method, Jeřábek developed tools for describing algorithms in ZPP and RP. He also showed in [13, 14] that the theory $VPV + \text{sWPHP}(\mathcal{L}_{\text{FP}})$ is powerful enough to formalize proofs of very sophisticated derandomization results, e.g. the Nisan-Wigderson theorem [24] and the Impagliazzo-Wigderson theorem [12]. (Note that Jeřábek actually used the single-sorted theory $PV_1 + \text{sWPHP}(PV)$, but these two theories are isomorphic.)

In [15], Jeřábek developed an even more systematic approach by showing that for any bounded P/poly definable set, there exists a suitable pair of surjective “counting functions” which can approximate the cardinality of the set up to a polynomially small error. From this and other results he argued convincingly that $VPV + \text{sWPHP}(\mathcal{L}_{\text{FP}})$ is the “right” theory for reasoning about probabilistic polynomial time algorithms. However so far no one has used his framework for feasible reasoning about specific interesting randomized algorithms in classes such as RP and RNC^2 .

In the present paper we analyze (in VPV) two such algorithms using Jeřábek’s framework. The first one is the RNC^2 algorithm for determining whether a bipartite graph has a perfect matching, based on the Schwartz-Zippel Lemma [27, 33] for polynomial identity testing applied to the Edmonds polynomial [9] associated with the graph. The second algorithm, due to Mulmuley, Vazirani and Vazirani [22], is in the function class associated with RNC^2 , and uses the Isolating Lemma to find such a perfect matching when it exists. Proving correctness of these algorithms involves proving that the probability of error is bounded above by $1/2$. We formulate this assertion in a way suggested by Jeřábek’s framework (see Definition 2.1). This involves defining polynomial time functions from $\{0, 1\}^n$ onto $\{0, 1\} \times \Phi(n)$, where $\Phi(n)$ is the set of random bit strings of length n which cause an error in the computation. We then show that VPV proves that the function is a surjection.

Our proofs are carried out in the theory VPV for polynomial time reasoning, without the surjective weak pigeonhole principle $\text{sWPHP}(\mathcal{L}_{\text{FP}})$. Jeřábek used the $\text{sWPHP}(\mathcal{L}_{\text{FP}})$ principle to prove theorems justifying the above definition of error probability, but we do not need it to apply his definition.

Many proofs concerning determinants are based on the *Lagrange expansion* (also known as the Leibniz formula)

$$\text{Det}(A) = \sum_{\sigma \in S_n} \text{sgn}(\sigma) \prod_{i=1}^n A(i, \sigma(i))$$

where the sum is over exponentially many terms. Since our proofs in VPV can only use polynomial time concepts, we cannot formalize such proofs, and must use other techniques. In the same vein, the standard proof of the Schwartz-Zippel Lemma assumes that a multivariate polynomial given by an arithmetic circuit can be expanded to a sum of monomials. But this sum in general has exponentially many terms so again we cannot directly formalize this proof in VPV .

2. PRELIMINARIES

2.1. Basic bounded arithmetic. The theory VPV for polynomial time reasoning used here is a two-sorted theory described by Cook and Nguyen [7]. The two-sorted language has variables x, y, z, \dots ranging over \mathbb{N} and variables X, Y, Z, \dots ranging over finite subsets of \mathbb{N} , interpreted as bit strings. Two sorted vocabulary \mathcal{L}_A^2 includes the usual symbols $0, 1, +, \cdot, =, \leq$ for arithmetic over \mathbb{N} , the length function $|X|$ for strings, the set membership relation \in , and string equality $=_2$ (subscript 2 is usually omitted). We will use the notation $X(t)$ for $t \in X$, and think of $X(t)$ as the t^{th} bit in the string X .

The number terms in the base language \mathcal{L}_A^2 are built from the constants $0, 1$, variables x, y, z, \dots and length terms $|X|$ using $+$ and \cdot . The only string terms are string variables, but when we extend \mathcal{L}_A^2 by adding string-valued functions, other string terms will be built as usual. The atomic formulas are $t = u, X = Y, t \leq u, t \in X$ for any number terms u, t and string variables X, Y . Formulas are built from atomic formulas using \wedge, \vee, \neg and $\exists x, \exists X, \forall x, \forall X$. Bounded number quantifiers are defined as usual, and bounded string quantifier $\exists X \leq t, \varphi$ stands for $\exists X (|X| \leq t \wedge \varphi)$ and $\forall X \leq t, \varphi$ stands for $\forall X (|X| \leq t \rightarrow \varphi)$, where X does not appear in term t .

Σ_0^B is the class of all \mathcal{L}_A^2 -formulas with no string quantifiers and only bounded number quantifiers. Σ_1^B -formulas are those of the form $\exists \vec{X} < \vec{t} \varphi$, where $\varphi \in \Sigma_0^B$ and the prefix of the bounded quantifiers might be empty. These classes are extended to Σ_i^B (and Π_i^B) for all $i \geq 0$, in the usual way.

Two-sorted complexity classes contain relations $R(\vec{x}, \vec{X})$, where \vec{x} are number arguments and \vec{X} are string arguments. In defining complexity classes using machines or circuits, the number arguments are represented in unary notation and the string arguments are represented in binary. The string arguments are the main inputs, and the number arguments are auxiliary inputs that can be used to index the bits of strings.

In the two sorted setting, we can define AC^0 to be the class of relations $R(\vec{x}, \vec{X})$ such that some alternating Turing machine accepts R in time $\mathcal{O}(\log n)$ with a constant number of alternations, where n is the sum of all the numbers in \vec{x} and the total length of all the string arguments in \vec{X} . Then from the descriptive complexity characterization of AC^0 , it can be shown that a relation $R(\vec{x}, \vec{X})$ is in AC^0 iff it is represented by some Σ_0^B -formula $\varphi(\vec{x}, \vec{X})$.

Given a class of relations C , we associate a class FC of string-valued functions $F(\vec{x}, \vec{X})$ and number functions $f(\vec{x}, \vec{X})$ with C as follows. We require that these functions to be p -bounded, i.e., the length of the outputs of F and f is bounded by a polynomial in x and $|X|$. Then we define FC to consist of all p -bounded number functions whose graphs are in C and all p -bounded string functions whose bit graphs are in C .

We write $\Sigma_i^B(\mathcal{L})$ to denote the class of Σ_i^B -formulas which may have function and predicate symbols from $\mathcal{L} \cup \mathcal{L}_A^2$. A string function is $\Sigma_0^B(\mathcal{L})$ -definable if it is p -bounded and its bit graph is represented by a $\Sigma_0^B(\mathcal{L})$ -formula. Similarly, a number function is Σ_0^B -definable from \mathcal{L} if it is p -bounded and its graph is represented by a $\Sigma_0^B(\mathcal{L})$ -formula.

The theory V^0 for AC^0 is the basis to develop theories for small complexity classes within P in [7]. The theory V^0 consists of the vocabulary \mathcal{L}_A^2 and axiomatized by the sets of 2 -*BASIC* axioms as given in Figure 1, which express basic properties of symbols in \mathcal{L}_A^2 ,

B1. $x + 1 \neq 0$ B2. $x + 1 = y + 1 \rightarrow x = y$ B3. $x + 0 = x$ B4. $x + (y + 1) = (x + y) + 1$ B5. $x \cdot 0 = 0$ B6. $x \cdot (y + 1) = (x \cdot y) + x$ B7. $(x \leq y \wedge y \leq x) \rightarrow x = y$	B8. $x \leq x + y$ B9. $0 \leq x$ B10. $x \leq y \vee y \leq x$ B11. $x \leq y \leftrightarrow x < y + 1$ B12. $x \neq 0 \rightarrow \exists y \leq x (y + 1 = x)$ L1. $X(y) \rightarrow y < X $ L2. $y + 1 = X \rightarrow X(y)$
--	---

SE. $(|X| = |Y| \wedge \forall i < |X| (X(i) = Y(i))) \rightarrow X = Y$

Figure 1: The 2-BASIC axioms

together with the *comprehension* axiom schema

$$\Sigma_0^B(\mathcal{L}_A^2)\text{-COMP:} \quad \exists X \leq y \forall z < y (X(z) \leftrightarrow \varphi(z)),$$

where $\varphi \in \Sigma_0^B(\mathcal{L}_A^2)$ and X does not occur free in φ .

In [7, Chapter 5], it was shown that V^0 is finitely axiomatizable and a p -bounded function is in FAC^0 iff it is provably total in V^0 . A universally-axiomatized conservative extension $\overline{V^0}$ of V^0 was also obtained by introducing function symbols and their defining axioms for all FAC^0 functions.

In [7, Chapter 9], Cook and Nguyen showed how to associate a theory VC to each complexity class $C \subseteq \text{P}$, where VC extends V^0 with an additional axiom asserting the existence of a solution to a complete problem for C . General techniques are also presented for defining a universally-axiomatized conservative extension \overline{VC} of VC which has function symbols and defining axioms for every function in FC , and VC admits induction on open formulas in this enriched vocabulary. It follows from Herbrand's Theorem that the provably-total functions in \overline{VC} (and hence in VC) are precisely the functions in FC . Using this framework, Cook and Nguyen defined explicitly theories for various complexity classes within P .

Since we need some basic linear algebra in this paper, we are interested in the two-sorted theory $V\#L$ and its universal conservative extension $\overline{V\#L}$ from [6]. Recall that $\#L$ is usually defined as the class of functions f such that for some nondeterministic logspace Turing machine M , $f(x)$ is the number of accepting computations of M on input x . Since counting the number of accepting paths of nondeterministic logspace is AC^0 -equivalent to matrix powering, $V\#L$ was defined to be the extension of the base theory V^0 with an additional axiom stating the existence of powers A^k for every matrix A over \mathbb{Z} . The closure of $\#L$ under AC^0 -reductions is called DET . It turns out that computing the determinant of integer matrices is complete for DET under AC^0 -reductions. In fact Berkowitz's algorithm can be used to reduce the determinant to matrix powering. Moreover, $\overline{V\#L}$ proves that the function Det , which computes the determinant of integer matrices based on Berkowitz's algorithm, is in the language of $\overline{V\#L}$. Unfortunately it is an open question whether the theory $\overline{V\#L}$ also proves the *cofactor expansion formula* and other basic properties of determinants. However from results in [29] it follows that $\overline{V\#L}$ proves that the usual properties

of determinants follow from the Cayley-Hamilton Theorem (which states that a matrix satisfies its characteristic polynomial).

In this paper, we are particularly interested in the theory VPV for polytime reasoning [7, Chapter 8.2] since we will use it to formalize all of our theorems. The universal theory VPV is based on Cook's single-sorted theory PV [8], which was historically the first theory designed to capture polytime reasoning. A nice property of PV (and VPV) is that their universal theorems translate into families of propositional tautologies with polynomial size proofs in any extended Frege proof system.

The vocabulary \mathcal{L}_{FP} of VPV extends that of $\overline{V^0}$ with additional symbols introduced based on Cobham's machine independent characterization of FP [5]. Let $Z^{<y}$ denote the first y bits of Z . Formally the vocabulary \mathcal{L}_{FP} of VPV is the smallest set satisfying

- (1) \mathcal{L}_{FP} contains the vocabulary of $\overline{V^0}$
- (2) For any two function $G(\vec{x}, \vec{X})$, $H(y, \vec{x}, \vec{X}, \vec{Z})$ over \mathcal{L}_{FP} and a \mathcal{L}_A^2 -term $t = t(y, \vec{x}, \vec{X})$, if F is defined by *limited recursion* from G , H and t , i.e.,

$$\begin{aligned} F(0, \vec{x}, \vec{X}) &= G(\vec{x}, \vec{X}), \\ F(y+1, \vec{x}, \vec{X}) &= H(y, \vec{x}, \vec{X}, F(y, \vec{x}, \vec{X}))^{<t(y, \vec{x}, \vec{X})}, \end{aligned}$$

then $F \in \mathcal{L}_{\text{FP}}$.

We will often abuse the notation by letting \mathcal{L}_{FP} denote the set of function symbols in \mathcal{L}_{FP} .

The theory VPV can then be defined to be the theory over \mathcal{L}_{FP} whose axioms are those of $\overline{V^0}$ together with defining axioms for every function symbols in \mathcal{L}_{FP} . VPV proves the scheme $\Sigma_0^B(\mathcal{L}_{\text{FP}})$ -*COMP* and the following schemes

$$\begin{aligned} \Sigma_0^B(\mathcal{L}_{\text{FP}})\text{-IND:} & \quad (\varphi(0) \wedge \forall x(\varphi(x) \rightarrow \varphi(x+1))) \rightarrow \forall x\varphi(x) \\ \Sigma_0^B(\mathcal{L}_{\text{FP}})\text{-MIN:} & \quad \varphi(y) \rightarrow \exists x(\varphi(x) \wedge \neg \exists z < x\varphi(z)) \\ \Sigma_0^B(\mathcal{L}_{\text{FP}})\text{-MAX:} & \quad \varphi(0) \rightarrow \exists x \leq y(\varphi(x) \wedge \neg \exists z \leq y(z > x \wedge \varphi(z))) \end{aligned}$$

where φ is any $\Sigma_0^B(\mathcal{L}_{\text{FP}})$ -formula. It follows from Herbrand's Theorem that the provably-total functions in VPV are precisely the functions in \mathcal{L}_{FP} .

Observe that VPV extends $\overline{V\#L}$ since matrix powering can easily be carried out in polytime, and thus all theorems of $\overline{V\#L}$ from [6, 29] are also theorems of VPV . From results in [29] (see page 44 of [14] for a correction) it follows that VPV proves the Cayley-Hamilton Theorem, and hence the cofactor expansion formula and other usual properties of determinants of integer matrices.

In our introduction, we mentioned V^1 , the two sorted version of Buss's S_2^1 theory [4]. The theory V^1 is also associated with polytime reasoning in the sense that the provably total functions of V^1 are FP functions, and V^1 is Σ_1^B -conservative over VPV . However, there is evidence showing that V^1 is stronger than VPV . For example, the theory V^1 proves the Σ_1^B -*IND*, Σ_1^B -*MIN* and Σ_1^B -*MAX* schemes while VPV cannot prove these Σ_1^B schemes, assuming the polynomial hierarchy does not collapse [18]. In this paper we do not use V^1 to formalize our theorems, since the weaker theory VPV suffices for all our needs.

2.2. Notation. If φ is a function of n variables, then let $\varphi(\alpha_1, \dots, \alpha_{n-1}, \bullet)$ denote the function of one variable resulting from φ by fixing the first $n-1$ arguments to $\alpha_1, \dots, \alpha_{n-1}$. We write $\psi : \Delta \twoheadrightarrow \Phi$ to denote that ψ is a surjection from Δ onto Φ .

We use $[X, Y)$ to denote $\{Z \in \mathbb{Z} \mid X \leq Z < Y\}$, i.e., the interval of integers between X and $Y - 1$, where strings code integers using signed binary notation. We also use the standard notation $[n]$ to denote the set $\{1, \dots, n\}$.

Given a square matrix M , we write $M[i|j]$ to denote the (i, j) -minor of M , i.e., the square matrix formed by removing the i th row and j th column from M .

We write \vec{x} to denote number sequence $\langle x_1, \dots, x_k \rangle$ and write \vec{X} to denote string sequence $\langle X_1, \dots, X_k \rangle$. We write $\vec{x}_{k \times k}$ and $\vec{X}_{k \times k}$ to denote that \vec{x} and \vec{X} have k^2 elements and are treated as two-dimensional arrays $\langle x_{i,j} \mid 1 \leq i, j \leq k \rangle$ and $\langle X_{i,j} \mid 1 \leq i, j \leq k \rangle$ respectively, where the elements of these two-dimensional arrays are listed by rows. Note that $\vec{x}_{k \times k}$ and $\vec{X}_{k \times k}$ can be simply encoded as integer matrices, and thus we will use matrix notation freely on them.

We write the notation “ $(T \vdash)$ ” in front of the statement of a theorem to indicate that the statement is formulated and proved within the theory T .

2.3. The weak pigeonhole principle. The *surjective weak pigeonhole principle* for a function F , denoted by $\text{sWPHP}(F)$, states that F cannot map $[0, nA)$ onto $[0, (n+1)A)$. Thus, the surjective weak pigeonhole principle for the class of VPV functions, denoted by $\text{sWPHP}(\mathcal{L}_{\text{FP}})$, is the schema

$$\{\text{sWPHP}(F) \mid F \in \mathcal{L}_{\text{FP}}\}.$$

Note that this principle is believed to be weaker than the usual surjective “strong” pigeonhole principle stating that we cannot map $[0, A)$ onto $[0, A+1)$. For example, $\text{sWPHP}(\mathcal{L}_{\text{FP}})$ can be proven in the theory V^3 (the two-sorted version of Buss’s theory S_2^3) for $\text{FP}^{\Sigma_3^P}$ reasoning (cf. [30]), but it is not known if the usual surjective pigeonhole principle for VPV functions can be proven within the theory $\bigcup_{i \geq 1} V^i$ for the polynomial hierarchy (the two-sorted version of Buss’s theory $S_2 := \bigcup_{i \geq 1} S_2^i$ in [4]).

2.4. Jeřábek’s framework for probabilistic reasoning. In this section, we give a brief and simplified overview of Jeřábek’s framework [13, 14, 15] for probabilistic reasoning within $VPV + \text{sWPHP}(\mathcal{L}_{\text{FP}})$. For more complete definitions and results, the reader is referred to Jeřábek’s work.

Let $F(R)$ be a VPV 0-1 valued function (which may have other arguments). We think of F as defining a relation on binary numbers R . Let $\Phi(n) = \{R < 2^n \mid F(R) = 1\}$. Observe that bounding the probability $\Pr_{R < 2^n} [F(R) = 1]$ from above by the ratio s/t is the same as showing that $t \cdot |\Phi(n)| \leq s \cdot 2^n$. More generally, many probability inequalities can be restated as inequalities between cardinalities of sets. This is problematic since even for the case of polytime definable sets, it follows from Toda’s theorem [31] that we cannot express their cardinalities directly using bounded formulas (assuming that the polynomial hierarchy does not collapse). Hence we need an alternative method to compare the sizes of definable sets without exact counting.

The method proposed by Jeřábek in [13, 14, 15] is based on the following simple observation: if $\Gamma(n)$ and $\Phi(n)$ are definable sets and there is a function F mapping $\Gamma(n)$ onto $\Phi(n)$, then the cardinality of $\Phi(n)$ is at most the cardinality of $\Gamma(n)$. Thus instead of counting the sets $\Gamma(n)$ and $\Phi(n)$ directly, we can compare the sizes of $\Gamma(n)$ and $\Phi(n)$ by showing the existence of a surjection F , which in many cases can be easily carried out within weak theories of bounded arithmetic. In this paper we will restrict our discussion to

the case when the sets are bounded polytime definable sets and the surjections are polytime functions, all of which can be defined within VPV , since this is sufficient for our results.

The remaining challenge is then to formally verify that the definition of cardinality comparison through the use of surjections is a meaningful and well-behaved definition. The basic properties of surjections like “any set can be mapped onto itself” and “surjectivity is preserved through function compositions” roughly correspond to the usual reflexivity and transitivity of cardinality ordering, i.e., $|\Phi| \leq |\Phi|$ and $|\Phi| \leq |\Gamma| \leq |\Lambda| \rightarrow |\Phi| \leq |\Lambda|$ for all bounded definable sets Φ, Γ and Λ . However more sophisticated properties, e.g., dichotomy $|\Phi| \leq |\Gamma| \vee |\Gamma| \leq |\Phi|$ or “uniqueness” of cardinality, turn out to be much harder to show.

As a result, Jeřábek proposed in [15] a systematic and sophisticated framework to justify his definition of size comparison. He observed that estimating the size of a P/poly definable set $\Phi \subseteq [0, 2^n)$ within an error $2^n/\text{poly}(n)$ is the same as estimating $\Pr_{X \in [0, 2^n)}[X \in \Phi]$ within an error $1/\text{poly}(n)$, which can be solved by drawing $\text{poly}(n)$ independent random samples $X \in [0, 2^n)$ and check if $X \in \Phi$. This gives us a polytime random sampling algorithm for approximating the size of Φ . Since a counting argument [13] can be formalized within $VPV + \text{sWPHP}(\mathcal{L}_{\text{FP}})$ to show the existence of suitable average-case hard functions for constructing Nisan-Wigderson generators, this random sampling algorithm can be derandomized to show the existence of an *approximate cardinality* S of Φ for any given error $E = 2^n/\text{poly}(n)$ in the following sense. The theory $VPV + \text{sWPHP}(\mathcal{L}_{\text{FP}})$ proves the existence of S, y and a pair of P/poly “counting functions” (F, G)

$$\begin{aligned} F &: [0, y) \times (\Phi \uplus [0, E)) \rightarrow [0, y \cdot S) \\ G &: [0, y \cdot (S + E)) \rightarrow [0, y) \times \Phi \end{aligned}$$

Intuitively the pair (F, G) witnesses that $S - E \leq |\Phi| \leq S + E$. This allows him to show many properties, expected from cardinality comparison, that are satisfied by his method within $VPV + \text{sWPHP}(\mathcal{L}_{\text{FP}})$ (see Lemmas 2.10 and 2.11 in [15]). It is worth noting that proving the uniqueness of cardinality within some error seems to be the best we can do within bounded arithmetic, where exact counting is not available.

For the present paper, the following definition is all we need to know about Jeřábek’s framework.

Definition 2.1. Let $\Phi(n) = \{R < 2^n \mid F(R) = 1\}$, where $F(R)$ is a VPV function (which may have other arguments) and let s, t be VPV terms. Then

$$\Pr_{R < 2^n}[R \in \Phi(n)] \lesssim s/t$$

means that either $\Phi(n)$ is empty, or there exists a VPV function $G(n, \bullet)$ mapping the set $[s] \times 2^n$ onto the set $[t] \times \Phi(n)$.

Since we are not concerned with justifying the above definition, our theorems can be formalized in VPV without sWPHP .

3. EDMONDS’ THEOREM

Let G be a bipartite graph with two disjoint sets of vertices $U = \{u_1, \dots, u_n\}$ and $V = \{v_1, \dots, v_n\}$. We use a pair (i, j) to encode the edge $\{u_i, v_j\}$ of G . Thus the edge relation of the graph G can be encoded by a boolean matrix $E_{n \times n}$, where we define $(i, j) \in E$, i.e. $E(i, j) = 1$, iff $\{u_i, v_j\}$ is an edge of G .

Each perfect matching in G can be encoded by an $n \times n$ permutation matrix M satisfying $M(i, j) \rightarrow E(i, j)$ for all $i, j \in [n]$. Recall that a *permutation matrix* is a square boolean matrix that has exactly one entry of value 1 in each row and each column and 0's elsewhere.

Let $A_{n \times n}$ be the matrix obtained from G by letting $A_{i,j}$ be an indeterminate $X_{i,j}$ for all $(i, j) \in E$, and let $A_{i,j} = 0$ for all $(i, j) \notin E$. The matrix of indeterminates $A(\vec{X})$ is called the *Edmonds matrix* of G , and $\text{Det}(A(\vec{X}))$ is called the *Edmonds polynomial* of G . In general this polynomial has exponentially many monomials, so for the purpose of proving its properties in *VPV* we consider $\text{Det}(A(\vec{X}))$ to be a function which takes as input an integer matrix $\vec{W}_{n \times n}$ and returns an integer $\text{Det}(A(\vec{W}))$. Thus $\text{Det}(A(\vec{X})) \equiv 0$ means that this function is identically zero.

The following theorem draws an important connection between determinants and matchings. The standard proof uses the *Lagrange expansion* which has exponentially many terms, and hence cannot be formalized in *VPV*. However we will give an alternative proof which can be so formalized.

Theorem 3.1 (Edmonds' Theorem [9]). (*VPV*⊢) *Let $\text{Det}(A(\vec{X}))$ be the Edmonds polynomial of the bipartite graph G . Then G has a perfect matching iff $\text{Det}(A(\vec{X})) \neq 0$ (i.e. iff there exists an integer matrix \vec{W} such that $\text{Det}(A(\vec{W})) \neq 0$).*

Proof. For the direction (\Rightarrow) we need the following lemma.

Lemma 3.2. (*VPV*⊢) $\text{Det}(M) \in \{-1, 1\}$ for any permutation matrix M .

Proof of Lemma 3.2. We will construct a sequence of matrices

$$N_n, N_{n-1}, \dots, N_1,$$

where $N_n = M$, $N_1 = (1)$, and we construct N_{i-1} from N_i by choosing j_i satisfying $N(i, j_i) = 1$ and letting $N_{i-1} = N_i[i | j_i]$.

From the way the matrices N_i are constructed, we can easily show by $\Sigma_0^B(\mathcal{L}_{\text{FP}})$ induction on $\ell = n, \dots, 1$ that the matrices N_ℓ are permutation matrices. Finally, using the cofactor expansion formula, we prove by $\Sigma_0^B(\mathcal{L}_{\text{FP}})$ induction on $\ell = 1, \dots, n$ that $\text{Det}(N_\ell) \in \{-1, 1\}$. \square

From the lemma we see that if M is the permutation matrix representing a perfect matching of G , then *VPV* proves $\text{Det}(A(M)) = \text{Det}(M) \in \{1, -1\}$, so $\text{Det}(A(\vec{X}))$ is not identically 0.

For the direction (\Leftarrow) it suffices to describe a polytime function F that takes as input an integer matrix $B_{n \times n} = A(\vec{W})$, where $A(\vec{X})$ is the Edmonds matrix of a bipartite graph G and $\vec{W}_{n \times n}$ is an integer value assignment, and reason in *VPV* that if $\text{Det}(B) \neq 0$, then F outputs a perfect matching of G .

Assume $\text{Det}(B) \neq 0$. Note that finding a perfect matching of G is the same as extracting a nonzero diagonal, i.e., a sequence of nonzero entries $B(1, \sigma(1)), B(2, \sigma(2)), \dots, B(n, \sigma(n))$, where σ is a permutation of the set $[n]$. For this purpose, we construct a sequence of matrices

$$B_n, B_{n-1}, \dots, B_1,$$

as follows. We let $B_n = B$. For $i = n, \dots, 2$, we let $B_{i-1} = B_i[i | j_i]$ and the index j_i is chosen using the following method. Suppose we already know B_i satisfying $\text{Det}(B_i) \neq 0$.

By the cofactor expansion along the last row of B_i ,

$$\text{Det}(B_i) = \sum_{j=1}^i B_i(i, j)(-1)^{i+j} \text{Det}(B_i[i | j]).$$

Thus, since $\text{Det}(B_i) \neq 0$, at least one of the terms in the sum on the right-hand side is nonzero. Thus, we can choose the least index j_i such that $B_i(i, j_i) \cdot \text{Det}(B_i[i | j_i]) \neq 0$.

To extract the perfect matching, we let Q be an $n \times n$ matrix, where $Q(i, j) = j$. Then we construct a sequence of matrices

$$Q_n, Q_{n-1}, \dots, Q_1,$$

where $Q_n = Q$ and $Q_{i-1} = Q_i[i | j_i]$, i.e., we delete from Q_i exactly the row and column we deleted from B_i . We define a permutation σ by letting $\sigma(i) = Q_i(i, j_i)$. Then $\sigma(i)$ is the column number in B which corresponds to column j_i in B_i , and the set of edges

$$\{(i, \sigma(i)) \mid 1 \leq i \leq n\}$$

is our desired perfect matching. □

4. SCHWARTZ-ZIPPEL LEMMA

The Schwartz-Zippel Lemma [27, 33] is one of the most fundamental tools in the design of randomized algorithms. The lemma provides us a `coRP` algorithm for the *polynomial identity testing problem* (PIT): given an arithmetic circuit computing a multivariate polynomial $P(\vec{X})$ over a field \mathbb{F} , we want to determine if $P(\vec{X})$ is identically zero. The PIT problem is important since many problems, e.g., primality testing [1], perfect matching [22], and software run-time testing [32], can be reduced to PIT. Moreover, many fundamental results in complexity theory like $\text{IP} = \text{PSPACE}$ [28] and the PCP theorem [2, 3] make heavy use of PIT in their proofs. The Schwartz-Zippel lemma can be stated as follows.

Theorem 4.1 (Schwartz-Zippel Lemma). *Let $P(X_1, \dots, X_n)$ be a non-zero polynomial of degree $D \geq 0$ over a field (or integral domain) \mathbb{F} . Let S be a finite subset of \mathbb{F} and let \vec{R} denote the sequence $\langle R_1, \dots, R_n \rangle$. Then*

$$\Pr_{\vec{R} \in S^n} [P(\vec{R}) = 0] \leq \frac{D}{|S|}.$$

Using this lemma, we have the following `coRP` algorithm for the PIT problem when $\mathbb{F} = \mathbb{Z}$. Given a polynomial

$$P(X_1, \dots, X_n)$$

of degree at most D , we choose a sequence $\vec{R} \in [0, 2D]^n$ at random. If P is given implicitly as a circuit, the degree of P might be exponential, and thus the value of $P(\vec{R})$ might require exponentially many bits to encode. In this case we use the method of Ibarra and Moran [10] and let Y be the result of evaluating $P(\vec{R})$ using arithmetic modulo a random integer from the interval $[1, D^k]$ for some fixed k . If $Y = 0$, then we report that $P \equiv 0$. Otherwise, we report that $P \not\equiv 0$. (Note that if P has small degree, then we can evaluate $P(\vec{R})$ directly.)

Unfortunately the Schwartz-Zippel Lemma seems hard to prove in bounded arithmetic. The main challenge is that the degree of P can be exponentially large. Even in the special case when P is given as the symbolic determinant of a matrix of indeterminates and hence the degree of P is small, the polynomial P still has up to $n!$ terms. Thus, we will focus

on a much weaker version of the Schwartz-Zippel Lemma that involves only Edmonds' polynomials since this will suffice for us to establish the correctness of a FRNC² algorithm for deciding if a bipartite graph has a perfect matching.

4.1. Edmonds' polynomials for complete bipartite graphs. In this section we will start with the simpler case when every entry of an Edmonds matrix is a variable, since it clearly demonstrates our techniques. This case corresponds to the Schwartz-Zippel Lemma for Edmonds' polynomials of complete bipartite graphs.

Let A be the full $n \times n$ Edmonds' matrix A , where $A_{i,j} = X_{i,j}$ for all $1 \leq i, j \leq n$. We consider the case that S is the interval of integers $S = [0, s)$ for $s \in \mathbb{N}$, so $|S| = s$. Then $\text{Det}(A(\vec{X}))$ is a nonzero polynomial of degree exactly n , and we want to show that

$$\Pr_{\vec{r} \in S^{n^2}} [\text{Det}(A(\vec{r})) = 0] \lesssim \frac{n}{s}.$$

Let

$$\mathcal{Z}(n, s) := \{\vec{r} \in S^{n^2} \mid \text{Det}(A(\vec{r})) = 0\},$$

i.e., the set of zeros of the Edmonds polynomial $\text{Det}(A(\vec{X}))$. Then by Definition 2.1, it suffices to exhibit a *VPV* function mapping $[n] \times S^{n^2}$ onto $S \times \mathcal{Z}(n, s)$. For this it suffices to give a *VPV* function mapping $[n] \times S^{n^2-1}$ onto $\mathcal{Z}(n, s)$. We will define a *VPV* function

$$F(n, s, \bullet) : [n] \times S^{n^2-1} \rightarrow \mathcal{Z}(n, s),$$

so $F(n, s, \bullet)$ takes as input a pair (i, \vec{r}) , where $i \in [n]$ and $\vec{r} \in S^{n^2-1}$ is a sequence of $n^2 - 1$ elements.

Let B be an $n \times n$ matrix with elements from S . For $i \in [n]$ let B_i denote the leading principal submatrix of B that consists of the $i \times i$ upper-left part of B . In other words, $B_n = B$, and $B_{i-1} := B_i[i \mid i]$ for $i = n, \dots, 2$. The following fact follows easily from the least number principle $\Sigma_0^B(\mathcal{L}_{\text{FP}})$ -MIN.

Fact 4.2. (*VPV* \vdash) If $\text{Det}(B) = 0$, then there is $i \in [n]$ such that $\text{Det}(B_j) = 0$ for all $i \leq j \leq n$, and either $i = 1$ or $i > 1$ and $\text{Det}(B_{i-1}) \neq 0$.

We claim that given $\text{Det}(B) = 0$ and given i as in the fact, the element $B(i, i)$ is uniquely determined by the other elements in B . Thus if $i = 1$ then $B(i, i) = 0$, and if $i > 1$ then by the cofactor expansion of $\text{Det}(B_i)$ along row i ,

$$0 = \text{Det}(B_i) = B_i(i, i) \cdot \text{Det}(B_{i-1}) + \text{Det}(B'_i) \quad (4.1)$$

where B'_i is obtained from B_i by setting $B'(i, i) = 0$. This equation uniquely determines $B_i(i, i)$ because $\text{Det}(B_{i-1}) \neq 0$.

The output of $F(n, s, (i, \vec{r}))$ is defined as follows. Let B be the $n \times n$ matrix determined by the $n^2 - 1$ elements in \vec{r} by inserting the symbol $*$ (for unknown) in the position for $B(i, i)$. Try to use the method above to determine the value of $* = B_i(i, i)$, assuming that $*$ is chosen so that $\text{Det}(B) = 0$. This method could fail because $\text{Det}(B_{i-1}) = 0$. In this case, or if the solution to the equation (4.1) gives a value for $B_i(i, i)$ which is not in S , output the default “dummy” zero sequence $\vec{0}_{n \times n}$. Otherwise let C be B with $*$ replaced by the obtained value of $B_i(i, i)$. If $\text{Det}(C) = 0$ then output C , otherwise output the dummy zero sequence.

Theorem 4.3. (VPV \vdash) Let $A(\vec{X})$ be the Edmonds matrix of a complete bipartite graph $K_{n,n}$. Let S denote the set $[0, s)$. Then the function $F(n, s, \bullet)$ defined above is a polytime surjection that maps $[n] \times S^{n^2-1}$ onto $\mathcal{Z}(n, s)$.

Proof. It is easy to see that $F(n, s, \bullet)$ is polytime (in fact it belongs to the complexity class DET). To see that F is surjective, let C be an arbitrary matrix in $\mathcal{Z}(n, s)$, so $\text{Det}(C) = 0$. Let $i \in [n]$ be determined by Fact 4.2 when $B = C$. Let \vec{r} be the sequence of $n^2 - 1$ elements consisting of the rows of C with $C(i, i)$ deleted. Then the algorithm for computing $F(n, s, (i, \vec{r}))$ correctly computes the missing element $C(i, i)$ and outputs C . \square

4.2. Edmonds' polynomials for general bipartite graphs. For general bipartite graphs, an entry of an Edmonds matrix A might be 0, so we cannot simply use leading principal submatrices in our construction of the surjection F . However given a sequence $\vec{W}_{n \times n}$ making $\text{Det}(A(\vec{W})) \neq 0$, it follows from Theorem 3.1 that we can find a perfect matching M in polytime. Thus, the nonzero diagonal corresponding to the perfect matching M will play the role of the main diagonal in our construction. The rest of the proof will proceed similarly. Thus, we have the following theorem.

Theorem 4.4. (VPV \vdash) There is a VPV function $H(n, s, A, \vec{W}, \bullet)$ where $A_{n \times n}$ is the Edmonds matrix for an arbitrary bipartite graph and \vec{W} is a sequence of n^2 (binary) integers, such that if $\text{Det}(A(\vec{W})) \neq 0$ then $H(n, s, A, \vec{W}, \bullet)$ maps $[n] \times S^{n^2-1}$ onto $\{\vec{r} \in S^{n^2} \mid \text{Det}(A(\vec{r})) = 0\}$, where $S = [0, s)$.

In other words, it follows from Definition 2.1 that the function $H(n, s, A, \vec{W}, \bullet)$ in the theorem witnesses that

$$\Pr_{\vec{r} \in S^{n^2}} [\text{Det}(A(\vec{r})) = 0] \lesssim \frac{n}{s}.$$

Proof. Assume $\text{Det}(A(\vec{W})) \neq 0$. Then the polytime function described in the proof of Theorem 3.1 produces an $n \times n$ permutation matrix M such that for all $i, j \in [n]$, if $M(i, j) = 1$ then the element $A(i, j)$ in the Edmonds matrix A is not zero. We apply the algorithm in the proof of Theorem 4.3, except that the sequence of principal submatrices of B used in Fact 4.2 is replaced by the sequence B_n, B_{n-1}, \dots, B_1 determined by M as follows. We let $B_n = B$, and for $i = n, \dots, 2$ we let $B_{i-1} = B_i[i \mid j_i]$, where the indices j_i are chosen the same way as in the proof of Theorem 3.1 when constructing the perfect matching M . \square

We note that the mapping $H(n, s, A, \bullet)$ in this case may not be in DET since the construction of M depends on the sequential polytime algorithm from Theorem 3.1 for extracting a perfect matching.

4.3. Formalizing the RNC² algorithm for the bipartite perfect matching decision problem. An instance of the *bipartite perfect matching decision* problem is a bipartite graph G encoded by a matrix $E_{n \times n}$, and we are to decide if G has a perfect matching. Here is an RDET algorithm for the problem. The algorithm is essentially due to Lovász [20]. From E , construct the Edmonds matrix $A(\vec{X})$ for G and choose a random sequence $\vec{r}_{n \times n} \in [2n]^{n^2}$. If $\text{Det}(A(\vec{r})) \neq 0$ then we report that G has a perfect matching. Otherwise, we report G does not have a perfect matching.

We claim that *VPV* proves correctness of this algorithm. The correctness assertion states that if G has a perfect matching then the algorithm reports NO with probability at most $1/2$, and otherwise it certainly reports NO. Theorem 3.1 shows that *VPV* proves the latter. Conversely, if G has a perfect matching given by a permutation matrix M then the function $H(n, 2n, A, M, \bullet)$ of Theorem 4.4 witnesses that the probability of $\text{Det}(A(\vec{r})) = 0$ is at most $1/2$, according to Definition 2.1, where A is the Edmonds matrix for G . Hence *VPV* proves the correctness of this case too.

Since $\text{RDET} \subseteq \text{FRNC}^2$, this algorithm (which solves a decision problem) is also an RNC^2 algorithm.

5. FORMALIZING THE HUNGARIAN ALGORITHM

The Hungarian algorithm is a combinatorial optimization algorithm which solves the *maximum-weight bipartite matching* problem in polytime and anticipated the later development of the powerful *primal-dual method*. The algorithm was developed by Kuhn [19], who gave the name ‘‘Hungarian method’’ since it was based on the earlier work of two Hungarian mathematicians: D. Kőnig and J. Egerváry. Munkres later reviewed the algorithm and showed that it is indeed polytime [23]. Although the Hungarian algorithm is interesting by itself, we formalize the algorithm since we need it in the *VPV* proof of the Isolating Lemma for perfect matchings in Section 6.1.

The Hungarian algorithm finds a maximum-weight matching for any weighted bipartite graph. The algorithm and its correctness proof are simpler if we make the two following changes. First, since edges with negative weights can never be in a maximum-weight matching, and thus can be safely deleted, we can assume that every edge has *nonnegative* weight. Second, by assigning zero weight to every edge not present, we only need to consider weighted *complete* bipartite graphs.

Let $G = (X \uplus Y, E)$ be a complete bipartite graph, where $X = \{x_i \mid 1 \leq i \leq n\}$ and $Y = \{y_i \mid 1 \leq i \leq n\}$, and let \vec{w} be an integer weight assignment to the edges of G , where $w_{i,j} \geq 0$ is the weight of the edge $\{x_i, y_j\} \in E$.

A pair of integer sequences $\vec{u} = \langle u_i \rangle_{i=1}^n$ and $\vec{v} = \langle v_i \rangle_{i=1}^n$ is called a *weight cover* if

$$\forall i, j \in [n], w_{i,j} \leq u_i + v_j. \quad (5.1)$$

The *cost* of a cover is $\text{cost}(\vec{u}, \vec{v}) := \sum_{i=1}^n (u_i + v_i)$. We also define $w(M) := \sum_{(i,j) \in M} w_{i,j}$. The Hungarian algorithm is based on the following important observation.

Lemma 5.1. (*VPV* \vdash) *For any matching M and weight cover (\vec{u}, \vec{v}) , we have $w(M) \leq \text{cost}(\vec{u}, \vec{v})$.*

Proof. Since the edges in a matching M are disjoint, summing the constraints $w_{i,j} \leq u_i + v_j$ over all edges of M yields $w(M) \leq \sum_{(i,j) \in M} (u_i + v_j)$. Since no edge has negative weight, we have $u_i + v_j \geq 0$ for all $i, j \in [n]$. Thus,

$$w(M) \leq \sum_{(i,j) \in M} (u_i + v_j) \leq \text{cost}(\vec{u}, \vec{v})$$

for every matching M and every weight cover (\vec{u}, \vec{v}) . □

Given a weight cover (\vec{u}, \vec{v}) , the *equality subgraph* $H_{\vec{u}, \vec{v}}$ is the subgraph of G whose vertices are $X \uplus Y$ and whose edges are precisely those $\{x_i, y_j\} \in E$ satisfying $w_{i,j} = u_i + v_j$.

Theorem 5.2. (VPV \vdash) *Let $H = H_{\vec{u}, \vec{v}}$ be the equality subgraph, and let M be a maximum cardinality matching of H . Then the following three statements are equivalent*

- (1) $w(M) = \text{cost}(\vec{u}, \vec{v})$.
- (2) M is a maximum-weight matching of G and (\vec{u}, \vec{v}) is a minimum-weight cover of G .
- (3) M is a perfect matching of the equality subgraph H .

(cf. Appendix A.3 for the full proof of this theorem.)

Below we give a simplified version of the Hungarian algorithm which runs in polynomial time when the edge weights are small (i.e. presented in unary notation). The correctness of the algorithm easily follows from Theorem 5.2.

Algorithm 5.3 (The Hungarian algorithm). We start with an arbitrary weight cover (\vec{u}, \vec{v}) with small weights: e.g. let

$$u_i = \max\{w_{i,j} \mid 1 \leq j \leq n\}$$

and $v_i = 0$ for all $i \in [n]$. If the equality subgraph $H_{\vec{u}, \vec{v}}$ has a perfect matching M , we report M as a maximum-weight matching of G . Otherwise, change the weight cover (\vec{u}, \vec{v}) as follows. Since the maximum matching M is not a perfect matching of H , the Hall's condition fails for H . Thus it is not hard (cf. Corollary 1 from Appendix A.2) to construct in polytime a subset $S \subseteq X$ satisfying $|N(S)| < |S|$, where $N(S)$ denotes the neighborhood of S . Hence we can calculate the quantity

$$\delta = \min\{u_i + v_j - w_{i,j} \mid x_i \in S \wedge y_j \notin N(S)\},$$

and decrease u_i by δ for all $x_i \in S$ and increase v_j by δ for all $y_j \in N(S)$ without violating the weight cover property (5.1). This strictly decreases the sum $\sum_{i=1}^n (u_i + v_i)$. Thus this process can only repeat at most as many time as the initial cost of the cover (\vec{u}, \vec{v}) . Assuming that all edge weights are small (i.e. presented in unary), the algorithm terminates in polynomial time. Finally we get an equality subgraph $H_{\vec{u}, \vec{v}}$ containing a perfect matching M , which by Theorem 5.2 is also a maximum-weight matching of G .

When formalizing the Isolating Lemma for bipartite matchings, we need a *VPV* function Mwpm that takes as inputs an edge relation $E_{n \times n}$ of a bipartite graph G and a nonnegative weight assignment \vec{w} to the edges in E , and outputs a minimum-weight perfect matching if such a matching exists, or outputs \emptyset to indicate that no perfect matching exists. Recall that the Hungarian algorithm returns a maximum-weight matching, and not a *minimum-weight perfect* matching. However we can use the Hungarian algorithm to compute $\text{Mwpm}(n, E, \vec{w})$ as follows.

Algorithm 5.4 (Finding a minimum-weight perfect matching).

- 1: Let $c = n \cdot \max\{w_{i,j} \mid (i, j) \in E\} + 1$
- 2: Construct the sequence \vec{w}' as follows

$$w'_{i,j} = \begin{cases} c - w_{i,j} & \text{if } (i, j) \in E \\ 0 & \text{otherwise.} \end{cases}$$

- 3: Run the Hungarian algorithm on the complete bipartite graph $K_{n,n}$ with weight assignment \vec{w}' to get a maximum-weight matching M .

4: **if** M contains an edge that is not in E **then**
 5: return the empty matching \emptyset
 6: **else**
 7: return M
 8: **end if**

Note that since we assign zero weights to the edges not present and very large weights to other edges, the Hungarian algorithm will always prefer the edges that are present in the original bipartite graph. More formally for any perfect matching M and non-perfect matching N we have

$$\begin{aligned} w'(M) &\geq nc - n \cdot \max\{w_{i,j} \mid (i,j) \in E\} \\ &= (n-1)c + (c - n \cdot \max\{w_{i,j} \mid (i,j) \in E\}) \\ &= (n-1)c + 1 \\ &> (n-1)c \\ &\geq w'(N) \end{aligned}$$

The last inequality follows from the fact that $w'_{i,j} \leq c$ for all $(i,j) \in N$. Thus, if the Hungarian algorithm returns a matching M with at least one edge not in E , then the original graph cannot have a perfect matching. Also from the way the weight assignment \bar{w}' was defined, every maximum-weight perfect matching of $K_{n,n}$ with weight assignment \bar{w}' is a minimum-weight matching of the original bipartite graph.

It is straightforward to check that the above argument can be formalized in *VPV*, so *VPV* proves the correctness of Algorithm 5.4 for computing the function *Mwpm*.

6. FRNC² ALGORITHM FOR FINDING A BIPARTITE PERFECT MATCHING

Below we recall the elegant FRNC² (or more precisely RDET) algorithm due to Mulmuley, Vazirani and Vazirani [22] for finding a bipartite perfect matching. Although the original algorithm works for general undirected graphs, we will only focus on bipartite graphs in this paper.

Let G be a bipartite graph with two disjoint sets of vertices $U = \{u_1, \dots, u_n\}$ and $V = \{v_1, \dots, v_n\}$. We first consider the *minimum-weight bipartite perfect matching problem*, where each edge $(i,j) \in E$ is assigned an integer weight $w_{i,j} \geq 0$, and we want to find a minimum-weight perfect matching of G . It turns out there is a DET algorithm for this problem under two assumptions: the weights must be polynomial in n , and the minimum-weight perfect matching must be *unique*. We let $A(\vec{X})$ be an Edmonds matrix of the bipartite graph. Replace $X_{i,j}$ with $W_{i,j} = 2^{w_{i,j}}$ (this is where we need the weights to be small). We then compute $\text{Det}(A(\vec{W}))$ using Berkowitz's FNC² algorithm. Assume that there exists exactly one (unknown) minimum-weight perfect matching M . We will show in Theorem 6.5 that $w(M)$ is exactly the position of the least significant 1-bit, i.e., the number of trailing zeros, in the binary expansion of $\text{Det}(A(\vec{W}))$. Once having $w(M)$, we can test if an edge $(i,j) \in E$ belongs to the unique minimum-weight perfect matching M as follows. Let w' be the position of the least significant 1-bit of $\text{Det}(A[i|j](\vec{W}))$. We will show in Theorem 6.6 that the edge (i,j) is in the perfect matching if and only if w' is precisely $w(M) - w_{i,j}$. Thus, we can test all edges in parallel. Note that up to this point, everything

can be done in $\text{DET} \subseteq \text{FNC}^2$ since the most expensive operation is the Det function, which is complete for DET.

What we have so far is that, assuming that the minimum-weight perfect matching exists and is unique, there is a DET algorithm for finding this minimum-weight perfect matching. But how do we guarantee that if a minimum-weight perfect matching exists, then it is unique? It turns out that we can assign every edge $(u_i, v_j) \in E$ a random weight $w_{i,j} \in [2m]$, where $m = |E|$, and use the Isolating Lemma [22] to ensure that the graph has a *unique* minimum-weight perfect matching with probability at least $1/2$.

The $\text{RDET} \subseteq \text{FRNC}^2$ algorithm for finding a perfect matching is now complete: assign random weights to the edges, and run the DET algorithm for the unique minimum-weight perfect matching problem. If a perfect matching exists, with probability at least $1/2$, this algorithm returns a perfect matching.

6.1. Isolating a perfect matching . We will recall the Isolating Lemma [22], the key ingredient of Mulmuley-Vazirani-Vazirani FRNC^2 algorithm for finding a perfect matching. Let X be a set with m elements $\{a_1, \dots, a_m\}$ and let \mathcal{F} be a family of subsets of X . We assign a weight w_i to each element $a_i \in X$, and define the weight of a set $Y \in \mathcal{F}$ to be $w(Y) := \sum_{a_i \in Y} w_i$. Let *minimum-weight* be the minimum of the weights of all the sets in \mathcal{F} . Note that several sets of \mathcal{F} might achieve minimum-weight. However, if minimum-weight is achieved by a unique $Y \in \mathcal{F}$, then we say that the weight assignment $\vec{w} = \langle w_i \rangle_{i=1}^m$ is *isolating* for \mathcal{F} . (Every weight assignment is isolating if $|\mathcal{F}| \leq 1$.)

Theorem 6.1 (Isolating Lemma [22]). *Let \mathcal{F} be a family of subsets of an n -element set $X = \{a_1, \dots, a_m\}$. Let $\vec{w} = \langle w_i \rangle_{i=1}^m$ be a random weight assignment to the elements in X . Then*

$$\Pr_{\vec{w} \in [k]^m} [\vec{w} \text{ is not isolating for } \mathcal{F}] \leq \frac{m}{k}.$$

To formalize the Isolating Lemma in *VPV* it seems natural to present the family \mathcal{F} by a polytime algorithm. This is difficult to do in general (see Remark 6.3 below), so we will formalize a special case which suffices to formalize the FRNC^2 algorithm for finding a bipartite perfect matching. Thus we are given a bipartite graph G , and the family \mathcal{F} is the set of perfect matchings of G . We want to show that if we assign random weights to the edges, then the probability that this weight assignment does not isolate a perfect matching is small. Note that although the family \mathcal{F} here might be exponentially large, \mathcal{F} is polytime definable, since recognizing a perfect matching is easy.

Theorem 6.2 (Isolating a Perfect Matching). *(VPV \vdash) Let \mathcal{F} be the family of perfect matchings of a bipartite graph G with edges $E = \{e_1, \dots, e_m\}$. Let \vec{w} be a random weight assignment to the edges in E . Then*

$$\Pr_{\vec{w} \in [k]^m} [\vec{w} \text{ is not isolating for } \mathcal{F}] \lesssim m/k.$$

For brevity, we will call a weight assignment \vec{w} “bad” if \vec{w} is not isolating for \mathcal{F} . Let

$$\Phi := \{\vec{w} \in [k]^m \mid \vec{w} \text{ is bad for } \mathcal{F}\}.$$

Then to prove Theorem 6.2, it suffices to construct a *VPV* function mapping $[m] \times [k]^{m-1}$ onto Φ . Note that the upper bound m/k is independent of the size n of the two vertex sets.

The set Φ is polytime definable since $\vec{w} \in \Phi$ iff

$$\exists i, j \in [n] \left(\begin{array}{l} E(i, j) \text{ and } M(i, j) \text{ and } \neg M'(i, j) \text{ and } M, M' \\ \text{encode two perfect matchings with the same weight} \end{array} \right),$$

where M denotes the output produced by applying the Mwpm function (Algorithm 5.4) on G , and M' denotes the output produced by applying Mwpm on the graph obtained from G by deleting the edge (i, j) .

Proof of Theorem 6.2. By Definition 2.1 we may assume that Φ is nonempty, so there is an element $\delta \in \Phi$. (We will use δ as a “dummy” element.) It suffices for us to construct explicitly a VPV function φ mapping $[m] \times [k]^{m-1}$ onto Φ . For each $i \in [k]$ we interpret the set $\{i\} \times [k]^{m-1}$ as the set of all possible weight assignments to the $m - 1$ edges $E \setminus \{e_i\}$. Our function φ will map each set $\{i\} \times [k]^{m-1}$ onto the set of those bad weight assignments \vec{w} such that the graph G contains two *distinct* minimum-weight perfect matchings M and M' with $e_i \in M \setminus M'$.

The function φ takes as input a sequence

$$\langle i, w_1, \dots, w_{i-1}, w_{i+1}, \dots, w_m \rangle$$

from $[m] \times [k]^{m-1}$ and does the following. Use the function Mwpm (defined by Algorithm 5.4) to find a minimum-weight perfect matching M' of G with the edge e_i deleted. Use Mwpm to find a minimum-weight perfect matching M_1 of the subgraph $G \setminus \{u_j, v_\ell\}$, where u_j and v_ℓ are the two endpoints of e_i . If both perfect matchings M' and M_1 exist and satisfy $w(M') - w(M_1) \in [k]$, then φ outputs the sequence

$$\langle w_1, \dots, w_{i-1}, w(M') - w(M_1), w_{i+1}, \dots, w_m \rangle. \quad (6.1)$$

Otherwise φ outputs the dummy element δ of Φ .

Note that if both M' and M_1 exist, then (6.1) is a bad weight assignment, since M' and $M = M_1 \cup \{e_i\}$ are distinct minimum-weight perfect matchings of G under this assignment.

To show that φ is surjective, consider an arbitrary bad weight assignment $\vec{w} = \langle w_i \rangle_{i=1}^m \in \Phi$. Since \vec{w} is bad, there are two distinct minimum-weight perfect matchings M and M' and some edge $e_i \in M \setminus M'$. Thus from how φ was defined,

$$\langle i, w_1, \dots, w_{i-1}, w_{i+1}, \dots, w_m \rangle \in [m] \times [k]^{m-1}$$

is an element that gets mapped to the bad weight assignment \vec{w} . \square

Remark 6.3. The above proof uses the fact that there is a polytime algorithm for finding a minimum-weight perfect matching (when one exists) in an edge-weighted bipartite graph. This suggests limitations on formalizing a more general version of Theorem 6.2 in VPV . For example, if \mathcal{F} is the set of Hamiltonian cycles in a complete graph, then finding a minimum weight member of \mathcal{F} is NP hard.

6.2. Extracting the unique minimum-weight perfect matching. Let G be a bipartite graph and assume that G has a perfect matching. Then in Section 6.1 we formalized a version of the Isolating Lemma, which with high probability gives us a weight assignment \vec{w} for which G has a unique minimum-weight perfect matching. This is the first step of the Mulmuley-Vazirani-Vazirani algorithm. Now we proceed with the second step, where we need to output this minimum-weight perfect matching using a DET function.

Let B be the matrix we get by substituting $W_{i,j} = 2^{w_{i,j}}$ for each nonzero entry (i, j) of the Edmonds matrix A of G . We want to show that if M is the unique minimum weight

perfect matching of G with respect to \vec{w} , then the weight $w(M)$ is exactly the position of the least significant 1-bit in the binary expansion of $\text{Det}(B)$. The usual proof of this fact is not hard, but it uses properties of the Lagrange expansion for the determinant, which has exponentially many terms and hence cannot be formalized in VPV . Our proof avoids using the Lagrange expansion, and utilizes properties of the cofactor expansion instead.

Lemma 6.4. ($VPV \vdash$) *There is a VPV function that takes as inputs an $n \times n$ Edmonds' matrix A and a weight sequence*

$$\vec{W} = \langle W_{i,j} = 2^{w_{i,j}} \mid 1 \leq i, j \leq n \rangle.$$

And if $B = A(\vec{W})$ satisfies $\text{Det}(B) \neq 0$ and p is the position of the least significant 1-bit of $\text{Det}(B)$, then the VPV function outputs a perfect matching M of weight at most p .

It is worth noting that the lemma holds regardless of whether or not the bipartite graph corresponding to A and weight assignment \vec{W} has a unique minimum-weight perfect matching.

The proof of Lemma 6.4 is very similar to that of Theorem 3.1. Recall that in Theorem 3.1, given a matrix B satisfying $\text{Det}(B) \neq 0$, we want to extract a nonzero diagonal of B . In this lemma, we are given the position p of the least significant 1-bit of $\text{Det}(B)$, and we want to get a nonzero diagonal of B whose product has the least significant 1-bit at position at most p . For this, we can use the same method of extracting the nonzero diagonal from Theorem 3.1 with the following modification. When choosing a term of the Lagrange expansion on the recursive step, we will also need to make sure the chosen term produces a nonzero sub-diagonal of B that will not contribute too much weight to the diagonal we are trying to extract. This ensures that the least significant 1-bit of the weight of the chosen diagonal is at most p .

For the rest of this section, we define $\text{numz}(Y)$ to be the position of the least significant 1-bit of the binary string Y . Thus if $\text{numz}(Y) = q$ then $Y = \pm 2^q Z$ for some positive odd integer Z .

Proof of Lemma 6.4. We construct a sequence of matrices

$$B_n, B_{n-1}, \dots, B_1$$

where $B_n = B$ and $B_{i-1} = B_i[i \mid j_i]$ for $i = n \dots, 2$ where the index j_i is chosen as follows. Define

$$T_i := \left(\prod_{\ell=i+1}^n B_\ell(\ell, j_\ell) \right) \text{Det}(B_i).$$

Assume we are given j_n, \dots, j_{i+1} such that $\text{numz}(T_i) \leq p$. We want to choose j_i such that $\text{numz}(T_{i-1}) \leq p$, where by definition

$$T_{i-1} = \left(\prod_{\ell=i}^n B_\ell(\ell, j_\ell) \right) \text{Det}(B_{i-1}) = \left(\prod_{\ell=i}^n B_\ell(\ell, j_\ell) \right) \text{Det}(B_i[i \mid j_i]).$$

This can be done as follows. From the cofactor expansion of $\text{Det}(B_i)$, we have

$$T_i = \sum_{j=1}^i (-1)^{i+j} \left(\prod_{\ell=i+1}^n B_\ell(\ell, j_\ell) \right) B_i(i, j) \text{Det}(B_i[i \mid j]).$$

Since $\text{numz}(T_i) \leq p$, at least one of the terms in the sum must have its least significant 1-bit at position at most p . Thus, we can choose j_i such that

$$\text{numz} \left(\left(\prod_{\ell=i+1}^n B_\ell(\ell, j_\ell) \right) B_i(i, j_i) \text{Det}(B_i[i | j_i]) \right) = \text{numz}(T_{i-1})$$

is minimized, which guarantees that $\text{numz}(T_{i-1}) \leq p$.

Since by assumption $\text{numz}(T_n) = \text{numz}(\text{Det}(B_n)) = p$, *VPV* proves by $\Sigma_0^B(\mathcal{L}_{\text{FP}})$ induction on $i = n, \dots, 1$ that

$$\text{numz}(T_i) \leq p.$$

If we define $j_1 = 1$, then when $i = 1$ we have

$$T_1 = \left(\prod_{\ell=2}^n B_\ell(\ell, j_\ell) \right) \text{Det}(B_1) = \prod_{\ell=1}^n B_\ell(\ell, j_\ell).$$

Thus it follows that $\text{numz}(T_1) = \text{numz}(\prod_{\ell=1}^n B_\ell(\ell, j_\ell)) \leq p$.

Similarly to the proof of Theorem 3.1, we can extract a perfect matching with weight at most p by letting Q be a matrix, where $Q(i, j) = j$ for all $i, j \in [n]$. Then we compute another sequence of matrices

$$Q_n, Q_{n-1}, \dots, Q_1,$$

where $Q_n = Q$ and $Q_{i-1} = Q_i[i | j_i]$, i.e., we delete from Q_i exactly the row and column we deleted from B_i .

To prove that $M = \{(\ell, Q_\ell(\ell, j_\ell)) \mid 1 \leq \ell \leq n\}$ is a perfect matching, we note that whenever a pair (i, k) is added to the matching M , we delete the row i and column j_i , where j_i is the index satisfying $Q_i(i, j_i) = k$. So we can never match any other vertex to k again.

It remains to show that $w(M) \leq p$. Since

$$\prod_{\ell=1}^n B_\ell(\ell, j_\ell) = 2^{w(M)},$$

the binary expansion of $\prod_{\ell=1}^n B_\ell(\ell, j_\ell)$ has a unique one at position $w(M)$ and zeros elsewhere. Thus it follows from how the matching M was constructed that $w(M) \leq p$. \square

The next two theorems complete our description and justification of our RDET algorithm for finding a perfect matching. For these theorems we are given a bipartite graph $G = (U \uplus V, E)$, where we have $U = \{u_1, \dots, u_n\}$ and $V = \{v_1, \dots, v_n\}$, and each edge $(i, j) \in E$ is assigned a weight $w_{i,j}$ such that G has a *unique* minimum-weight perfect matching (see Theorem 6.2). Let $\vec{W}_{n \times n}$ be a sequence satisfying $W_{i,j} = 2^{w_{i,j}}$ for all $(i, j) \in E$. Let A be the Edmonds matrix of G , and let $B = A(\vec{W})$. Let M denote the unique minimum weight perfect matching of G .

Theorem 6.5. (*VPV* \vdash) *The weight $p = w(M)$ is exactly $\text{numz}(\text{Det}(B))$.*

If in Lemma 6.4 we tried to extract an appropriate nonzero diagonal of B using the determinant and minors of B as our guide, then in the proof of this theorem we do the reverse. From a minimum-weight perfect matching M of G , we want to rebuild in polynomially many steps suitable minors of B until we fully recover the determinant of B . We can then prove by $\Sigma_0^B(\mathcal{L}_{\text{FP}})$ induction that in every step of this process, each ‘‘partial determinant’’ of B has the least significant 1-bit at position p . Note that the technique we used to

prove this theorem does have some similarity to that of Lemma 3.2, even though the proof of this theorem is more complicated.

Proof. Let Q be a matrix, where $Q(i, j) = j$ for all $i, j \in [n]$. For $1 \leq i \leq n$ let B_i be the result of deleting rows $i + 1, \dots, n$ and columns $M(i + 1), \dots, M(n)$ from B and let Q_i be Q with the same rows and columns deleted. We can construct these matrices inductively in the form of two matrix sequences

$$B_n, B_{n-1}, \dots, B_1 \qquad Q_n, Q_{n-1}, \dots, Q_1$$

where

- we let $B_n = B$ and $Q_n = Q$, and
- for $i = n, n - 1, \dots, 2$, define j_i to be the unique index satisfying

$$M(i) = Q_i(i, j_i),$$

and then let $B_{i-1} = B_i[i \mid j_i]$ and $Q_{i-1} = Q_i[i \mid j_i]$.

Then (setting $j_1 = 1$)

$$B_i(i, j_i) = B(i, M(i)) = 2^{w_{i, M(i)}}, \quad 1 \leq i \leq n \tag{6.2}$$

Claim: $\text{numz}(\text{Det}(B_i)) = \sum_{\ell=1}^i w_{\ell, M(\ell)}$ for all $i \in [n]$.

The theorem follows from this by setting $i = n$. We will prove the claim by induction on i . The base case $i = 1$ follows from (6.2).

For the induction step, it suffices to show

$$\text{numz}(\text{Det}(B_{i+1})) = \text{numz}(\text{Det}(B_i)) + w_{i+1, M(i+1)}$$

From the cofactor expansion formula we have

$$\text{Det}(B_{i+1}) = \sum_{j=1}^{i+1} (-1)^{(i+1)+j} B_{i+1}(i+1, j) \text{Det}(B_{i+1}[i+1 \mid j])$$

Since $B_{i+1}(i+1, j_{i+1}) = 2^{w_{i+1, M(i+1)}}$ by (6.2), and $\text{Det}(B_{i+1}[i+1 \mid j_{i+1}]) = \text{Det}(B_i)$, it suffices to show that if

$$R := B_{i+1}(i+1, j_{i+1}) \text{Det}(B_{i+1}[i+1 \mid j_{i+1}])$$

then

$$\text{numz}(R) < \text{numz}(B_{i+1}(i+1, j) \text{Det}(B_{i+1}[i+1 \mid j]))$$

for all $j \neq j_{i+1}$.

Suppose for a contradiction that there is some $j' \neq j_{i+1}$ such that

$$\text{numz}(B_{i+1}(i+1, j') \text{Det}(B_{i+1}[i+1 \mid j'])) \leq \text{numz}(R).$$

Then, we can extend the set of edges

$$\{(n, M(n)), \dots, (i+2, M(i+2)), (i+1, j')\}$$

with i edges extracted from $B_{i+1}[i+1 \mid j']$ (using the method from Lemma 6.4) to get a perfect matching of G with weight at most p , which contradicts that M is the unique minimum-weight perfect matching of G . \square

To extract the edges of M in DET, we need to decide if an edge (i, j) belongs to the unique minimum-weight perfect matching M without knowledge of other edges in M . The next theorem, whose proof follows directly from Lemma 6.4 and Theorem 6.5, gives us that method.

Theorem 6.6. (*VPV*⊢) *For every edge $(i, j) \in E$, we have $(i, j) \in M$ if and only if*

$$w(M) - w_{i,j} = \text{numz}(\text{Det}(B[i | j])).$$

Proof. (\Rightarrow): Assume $(i, j) \in M$. Then the bipartite graph $G' = G \setminus \{u_i, v_j\}$ must have a unique minimum-weight perfect matching of weight $w(M) - w_{i,j}$. Thus from Theorem 6.5,

$$\text{numz}(\text{Det}(B[i | j])) = w(M) - w_{i,j}.$$

(\Leftarrow): We prove the contrapositive. Assume $(i, j) \notin M$. Suppose for a contradiction that

$$w(M) - w_{i,j} = \text{numz}(\text{Det}(B[i | j])).$$

Then by Lemma 6.4 we can extract from the submatrix $B[i | j]$ a perfect matching Q of the bipartite graph $G' = G \setminus \{u_i, v_j\}$ with weight at most $w(M) - w_{i,j}$. But then $M' = Q \cup \{(i, j)\}$ is another perfect matching of G with $w(M') \leq w(M)$, a contradiction. \square

Theorems 6.2, 6.5, and 6.6 complete the description and justification of our RDET algorithm for finding a perfect matching in a bipartite graph. Since these are theorems of *VPV*, it follows that *VPV* proves the correctness of the algorithm.

6.3. Related bipartite matching problems. The correctness of the Mulmuley-Vazirani-Vazirani algorithm can easily be used to establish the correctness of RDET algorithms for related matching problems, for example, the maximum (cardinality) bipartite matching problem and the minimum-weight bipartite perfect matching problem, where the weights assigned to the edges are small. We refer to [22] for more details on these reductions.

7. CONCLUSION AND FUTURE WORK

We have only considered randomized matching algorithms for *bipartite* graphs. For general undirected graphs, we need Tutte's matrix (cf. [22]), a generalization of Edmonds' matrix. Since every Tutte matrix is a skew symmetric matrix where each variable appears exactly twice, we cannot directly apply our technique for Edmonds' matrices, where each variable appears at most once. However, by using the recursive definition of the Pfaffian instead of the cofactor expansion, we believe that it is also possible to generalize our results to general undirected graphs. We also note that the Hungarian algorithm only works for weighted bipartite graphs. To find a maximum-weight matching of a weighted undirected graph, we need to formalize Edmonds' *blossom algorithm* (cf. [16]). Once we have the correctness of the blossom algorithm, the proof of the Isolating Lemma for undirected graph perfect matchings will be the same as that of Theorem 6.2. We leave the detailed proofs for the general undirected graph case for future work.

It is worth noticing that symbolic determinants of Edmonds' matrices result in very special polynomials, whose structures can be used to define the *VPV* surjections witnessing the probability bound in the Schwartz-Zippel Lemma as demonstrated in this paper. It remains an open problem whether we can prove the full version of the Schwartz-Zippel Lemma using Jeřábek's method within the theory *VPV*.

We have shown that the correctness proofs for several randomized algorithms can be formalized in the theory VPV for polynomial time reasoning. But some of the algorithms involved are in the subclass DET of polynomial time, where DET is the closure of $\#L$ (and also of the integer determinant function Det) under AC^0 reductions. As mentioned in Section 2.1 the subtheory $\overline{V\#L}$ of VPV can define all functions in DET , but it is open whether $\overline{V\#L}$ can prove properties of Det such as the expansion by minors. However the theory $\overline{V\#L} + CH$ can prove such properties, where CH is an axiom stating the Cayley-Hamilton Theorem. Thus in the statements of Lemma 3.2 and Theorem 4.3 we could have replaced $(VPV \vdash)$ by $(\overline{V\#L} + CH \vdash)$. We could have done the same for Theorem 4.4 if we changed the argument \vec{W} of the function H to M , where M is a permutation matrix encoding a perfect matching for the underlying bipartite graph. This modified statement of the theorem still proves the interesting direction of the correctness of the bipartite perfect matching algorithm in Section 4.3, since the function H is used only to bound the error assuming that G does have a perfect matching.

We leave open the question of whether any of the other correctness proofs can be formalized in $\overline{V\#L} + CH$.

We believe that Jeřábek's framework deserves to be studied in greater depth since it helps us to understand better the connection between probabilistic reasoning and weak systems of bounded arithmetic. We are working on using Jeřábek's ideas to formalize constructive aspects of fundamental theorems in finite probability in the spirit of the recent beautiful work by Moser and Tardos [21], Impagliazzo and Kabanets [11], etc.

APPENDIX A. FORMALIZING THE HUNGARIAN ALGORITHM

Before proceeding with the Hungarian algorithm, we need to formalize the two most fundamental theorems for bipartite matching: Berge's Theorem and Hall's Theorem.

A.1. Formalizing Berge's Theorem and the augmenting-path algorithm. Let $G = (X \uplus Y, E)$ be a bipartite graph, where $X = \{x_i \mid 1 \leq i \leq n\}$ and $Y = \{y_i \mid 1 \leq i \leq n\}$. Formally to make sure that X and Y are disjoint, we can let $x_i := i$ and $y_i := n + i$. We encode the edge relation E of G by a matrix $E_{n \times n}$, where $E(i, j) = 1$ iff x_i is adjacent to y_j . Note that we often abuse notation and write $\{u, v\} \in E$ to denote that u and v are adjacent in G , which formally means either

$$\begin{aligned} u \in X \wedge v \in Y \wedge E(u, v - n), \text{ or} \\ v \in X \wedge u \in Y \wedge E(v, u - n). \end{aligned}$$

This complication is due to the usual convention of using an $n \times n$ matrix to encode the edge relation of a bipartite graph with $2n$ vertices.

An $n \times n$ matrix M encodes a matching of G iff M is a permutation matrix satisfying

$$\forall i, j \in [n], M(i, j) \rightarrow E(i, j).$$

We represent a path by a sequence of vertices $\langle v_1, \dots, v_k \rangle$ with $\{v_i, v_{i+1}\} \in E$ for all $i \in [k]$.

Given a matching M , a vertex v is M -saturated if v is incident with an edge in M . We will say v is M -unsaturated if it is not M -saturated. A path $P = \langle v_1, \dots, v_k \rangle$ is an M -alternating path if P alternates between edges in M and edges in $E \setminus M$. More formally, P is an M -alternating path if either of the following two conditions holds:

- For every $i \in \{1, \dots, k-1\}$, $\{v_i, v_{i+1}\} \in E \setminus M$ if i is odd, and $\{v_i, v_{i+1}\} \in M$ if i is even.
- For every $i \in \{1, \dots, k-1\}$, $\{v_i, v_{i+1}\} \in E \setminus M$ if i is even, and $\{v_i, v_{i+1}\} \in M$ if i is odd.

An M -alternating path $\langle v_1, \dots, v_k \rangle$ is an M -augmenting path if the vertices v_1 and v_k are M -unsaturated.

Theorem A.1 (Berge's Theorem). (*VPV*) Let $G = (X \uplus Y, E)$ be a bipartite graph. A matching M is maximum iff there is no M -augmenting path in G .

Proof. (\Rightarrow): Assume that all matchings N of E satisfy $|N| \leq |M|$. Suppose for a contradiction that there is an M -augmenting path P . Let $M \oplus P$ denote the symmetric difference of two sets of edges M and P . Then $M' = M \oplus P$ is a matching greater than M , a contradiction.

(\Leftarrow): We will prove the contrapositive. Assume there is another matching M' satisfying $|M'| > |M|$. We want to construct an M -augmenting path in G .

Consider $Q = M' \oplus M$. Since $|M'| > |M|$, it follows that $|M' \setminus M| > |M \setminus M'|$, and thus

$$|Q \cap M'| > |Q \cap M| \tag{A.1}$$

Note that we can compute cardinalities of the sets directly here since all the sets we are considering here are small. Now let H be the graph whose edge relation is Q and whose vertices are simply the vertices of G . We then observe the following properties of H :

- Since Q is constructed from two matchings M and M' , every vertex of H can only be incident with at most two edges: one from M and another from M' . So every vertex of H has degree at most 2.
- Any path of H must alternate between the edges of M and M' .

We will provide a polytime algorithm to extract from the graph H an augmenting path with respect to M , which gives us the contradiction.

- 1: Initialize $K = H$ and $i = 1$
- 2: **while** $K \neq \emptyset$ **do**
- 3: Pick the least vertex $v \in K$
- 4: Compute the connected component C_i containing v
- 5: **if** C_i is an M -augmenting path **then**
- 6: return C_i and halt.
- 7: **end if**
- 8: Update $K = K \setminus C_i$ and $i = i + 1$.
- 9: **end while**

Note that since H has n vertices, the while loop can only iterate at most n times. It only remains to show the following.

Claim: The algorithm returns an M -augmenting path assuming $|M'| > |M|$.

Suppose for a contradiction that the algorithm would never produce any M -augmenting path. Since H has degree at most two, in every iteration of the while loop, we know that the connected component C_i is

- either a cycle, which means $|C_i \cap M| = |C_i \cap M'|$, or
- a path but not an M -augmenting path, which implies that $|C_i \cap M| \geq |C_i \cap M'|$.

Since $Q = M' \oplus M = \bigcup_i C_i$ and all C_i are disjoint, we have

$$|Q \cap M| = \left| \bigcup_i (C_i \cap M) \right| \geq \left| \bigcup_i (C_i \cap M') \right| = |Q \cap M'|.$$

But this contradicts (A.1). \square

Algorithm A.2 (The augmenting-path algorithm). As a corollary of Berge's Theorem, we have the following simple algorithm for finding a maximum matching of a bipartite graph G . We start from any matching M of G , say empty matching. Repeatedly locate an M -augmenting path P and augment M along P and replace M by the resulting matching. Stop when there is no M -augmenting path. Then we know that M is maximum. Thus, it remains to show how to search for an M -augmenting path given a matching M of G .

Algorithm A.3 (The augmenting-path search algorithm). First, from G and M we construct a directed graph H , where the vertices V_H of H are exactly the vertices $X \uplus Y$ of G , and the edge relation E_H of H is a $2n \times 2n$ matrix defined as follows:

$$E_H := \{(x, y) \in X \times Y \mid \{x, y\} \in E \setminus M\} \cup \{(y, x) \in Y \times X \mid \{y, x\} \in M\}.$$

The key observation is that t is reachable from s by an M -alternating path in the bipartite graph G iff t is reachable from s in the directed graph H .

After constructing the graph H , we can search for an M -augmenting path using the *breadth first search* algorithm as follows. Let s be an M -unsaturated vertex in X . We construct two $2n \times 2n$ matrices S and T as follows.

- 1: The row $\text{Row}(1, S)$ of S encodes the set $\{s\}$, the starting point of our search.
- 2: From $\text{Row}(i, S)$, the set $\text{Row}(i + 1, S)$ is defined as follows: $j \in \text{Row}(i + 1, S) \leftrightarrow$ there exists some $k \in [n]$,

$$\text{Row}(i, S)(k) \wedge E_H(k, j) \wedge \forall \ell \in [i], \neg \text{Row}(\ell, S)(k).$$

After finish constructing $\text{Row}(i + 1, S)$, we can update T by setting $T(j, k)$ to 1 for every $k \in \text{Row}(i, S)$ and $j \in \text{Row}(i + 1, S)$ satisfying $E_H(k, j)$.

Intuitively, $\text{Row}(i, S)$ encodes the set of vertices that are of distance $i - 1$ from s , and T is the auxiliary matrix that can be used to recover a path from s to any vertex $j \in \text{Row}(i, S)$ for all $i \in [2n]$.

Remark A.4. Let $R = \bigcup_{i=2}^n \text{Row}(i, S)$, then R the set of vertices reachable from s by an M -alternating path. This follows from the fact that our construction mimics the construction of the formula δ_{CONN} , which was used to define the theory VNL for NL in [7, Chapter 9]. The matrix T is constructed to help us trace a path for every vertex $v \in R$ to s .

By induction on i , we can prove that $\text{Row}(i, S) \subseteq X$ for every odd $i \in [2n]$, and $\text{Row}(i, S) \subseteq Y$ for every even $i \in [2n]$. Thus, after having S and T , we choose the largest even i^* such that $\text{Row}(i^*, S) \neq \emptyset$, and then search the set $\text{Row}(i^*, S) \cap Y$ for an M -unsaturated vertex t . If such vertex t exists, then we use T to trace back a path to s . This path will be our M -augmenting path. If no such vertex t exists, we report that there is no M -augmenting path.

A.2. Formalizing Hall's Theorem .

Theorem A.5 (Hall's Theorem). (*VPV*⊢) *Let $G = (X \uplus Y, E)$ be a bipartite graph. Then G has a perfect matching iff for every subset $S \subseteq X$, $|S| \leq |N(S)|$, where $N(S)$ denotes the neighborhood of S in G .*

The condition $\forall S \subseteq X, |S| \leq |N(S)|$, which is necessary and sufficient for a bipartite graph to have a perfect matching, is called Hall's condition. We encode a set $S \subseteq X$ in the theorem as a binary string of length n , where $S(i) = 1$ iff $x_i \in S$. Similarly, we encode the neighborhood $N(S)$ as a binary string of length n , and we define

$$N(S) := \bigcup \{ \text{Row}(i, E) \mid x_i \in S \},$$

where the union can be computed by taking the disjunction of all binary vectors in the set

$$\{ \text{Row}(i, E) \mid x_i \in S \}$$

componentwise. Note that we can compute the cardinalities of S and $N(S)$ directly since both of these sets are subsets of small sets X and Y .

Proof. (\Rightarrow): Assume M is a perfect matching of G . Given a subset $S \subseteq X$, the vertices in S are matched to the vertices in some subset $T \subseteq Y$ by the perfect matching M , where $|S| = |T|$. Since $T \subseteq N(S)$, we have $|N(S)| \geq |T| = |S|$.

(\Leftarrow): We will prove the contrapositive. Assume G does not have a perfect matching, we want to construct a subset $S \subseteq X$ such that $|N(S)| < |S|$.

Let M be a maximum but not perfect matching constructed by the augmenting-path algorithm. Since M is not a perfect matching, there is some M -unsaturated vertex $s \in X$. Let S and T be the result of running the "augmenting-path search" algorithm from s , then $R := \bigcup_{i=2}^n \text{Row}(i, S)$ is the set of all vertices reachable from s by an M -alternating path. Since there is no M -augmenting path, all the vertices in R are M -saturated. We want to show the following two claims.

Claim 1: The vertices in $R \cap X$ are all matched to the vertices in $R \cap Y$ by M , and

$$|R \cap X| = |R \cap Y|.$$

Suppose for a contradiction that some vertex $v \in R$ is not matched to any vertex $u \in R$ by M . Since we already know that all vertices in R are M -saturated, v is matched by some vertex $w \notin R$ by M . But this is a contradiction since w must be reachable from s by an alternating path, and so the augmenting-path search algorithm must already have added w to R . Thus, the vertices in $R \cap X$ are all matched to the vertices in $R \cap Y$ by M , which implies that $|R \cap X| = |R \cap Y|$.

Claim 2: $N(R \cap X) = R \cap Y$.

Since $R \cap X$ are matched to $R \cap Y$, we know $N(R \cap X) \supseteq R \cap Y$. Suppose for a contradiction that $N(R \cap X) \supset R \cap Y$. Let $v \in N(R \cap X) \setminus R \cap Y$, and $u \in R \cap X$ be the vertex adjacent to v . Since u is reachable from s by an M -alternating path P , we can extend P to get an M -alternating path from s to v , which contradicts that v is not added to R .

We note that $N(\{s\} \cup (R \cap X)) = R \cap Y$. Then $S = \{s\} \cup (R \cap X)$ is the desired set since

$$|N(S)| = |R \cap Y| = |R \cap X| < |S|.$$

□

From the proof of Hall's Theorem, we have the following corollary saying that if a bipartite graph does not have a perfect matching, then we can find in polytime a subset of vertices violating Hall's condition.

Corollary 1. (*VPV*⊢) There is a *VPV* function that, on input a bipartite graph G that does not have a perfect matching, outputs a subset $S \subseteq X$ such that $|S| > |N(S)|$.

A.3. Proof of Theorem 5.2 . Let $H = H_{\vec{u}, \vec{v}}$ be the equality subgraph for the weight cover (\vec{u}, \vec{v}) , and let M be a maximum cardinality matching of H . Recall Theorem 5.2 wants us to show that *VPV* proves equivalence of the following three statements:

- (1) $w(M) = \text{cost}(\vec{u}, \vec{v})$
- (2) M is a maximum-weight matching and the cover (\vec{u}, \vec{v}) is a minimum-weight cover of G
- (3) M is a perfect matching of H

Proof of Theorem 5.2. (1) \Rightarrow (2): Assume that $\text{cost}(\vec{u}, \vec{v}) = w(M)$. By Lemma 5.1, no matching has weight greater than $\text{cost}(\vec{u}, \vec{v})$, and no cover with weight less than $w(M)$.

(2) \Rightarrow (3): Assume M is a maximum-weight matching and (\vec{u}, \vec{v}) is a minimum-weight cover of G . Suppose for a contradiction that the maximum matching M is not a perfect matching of H . We will construct a weight cover whose cost is strictly less than $\text{cost}(\vec{u}, \vec{v})$, which contradicts that (\vec{u}, \vec{v}) is a minimum-weight cover.

Since the maximum matching M is not a perfect matching of H , by Corollary 1, we can construct in polytime a subset $S \subseteq X$ satisfying

$$|N(S)| < |S|.$$

Then we calculate the quantity

$$\delta = \min\{u_i + v_j - w_{i,j} \mid x_i \in S \wedge y_j \notin N(S)\}.$$

Note that $\delta > 0$ since H is the equality subgraph. Next we construct a pair of sequences $\vec{u}' = \langle u'_i \rangle_{i=1}^n$ and $\vec{v}' = \langle v'_i \rangle_{i=1}^n$, as follows:

$$u'_i = \begin{cases} u_i - \delta & \text{if } x_i \in S \\ u_i & \text{if } x_i \notin S \end{cases} \quad v'_i = \begin{cases} v_j + \delta & \text{if } y_j \in N(S) \\ v_j & \text{if } y_j \notin N(S) \end{cases}$$

We claim that (\vec{u}', \vec{v}') is again a weight cover. The condition $w_{i,j} \leq u'_i + v'_j$ might only be violated for $x_i \in S$ and $y_j \notin N(S)$. But since we chose $\delta \leq u_i + v_j - w_{i,j}$, it follows that

$$w_{i,j} \leq (u_i - \delta) + v_j = u'_i + v'_j.$$

Since

$$\begin{aligned} \text{cost}(\vec{u}', \vec{v}') &= \sum_{i=1}^n (u'_i + v'_i) \\ &= \underbrace{\sum_{i=1}^n (u_i + v_i)}_{=\text{cost}(\vec{u}, \vec{v})} + \underbrace{\delta|N(S)| - \delta|S|}_{<0}, \end{aligned}$$

it follows that $\text{cost}(\vec{u}', \vec{v}') < \text{cost}(\vec{u}, \vec{v})$.

(3) \Rightarrow (1): Suppose M is a perfect matching of H . Then $w_{i,j} = u_i + v_j$ holds for all edges in M . Summing equalities $w_{i,j} = u_i + v_j$ over all edges of M yields the equality $\text{cost}(\vec{u}, \vec{v}) = w(M)$. \square

ACKNOWLEDGEMENT

We would like to thank Emil Jeřábek for answering our questions regarding his framework. We also thank the referees and Lila Fontes for their constructive comments. This work was financially supported by the Ontario Graduate Scholarship and the Natural Sciences and Engineering Research Council of Canada.

REFERENCES

- [1] M. Agrawal and S. Biswas. Primality and identity testing via chinese remaindering. *Journal of the ACM*, 50(4):429–443, 2003.
- [2] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, 1998.
- [3] S. Arora and S. Safra. Probabilistic checking of proofs: A new characterization of NP. *Journal of the ACM*, 45(1):122, 1998.
- [4] S.R. Buss. *Bounded arithmetic*. Bibliopolis, Naples, 1986.
- [5] A. Cobham. The Intrinsic Computational Difficulty of Functions. In *Proceedings of the 1995 International Congress for Logic, Methodology and Philosophy of Science*, pages 24–30. North-Holland, 1965.
- [6] S. Cook and L. Fontes. Formal Theories for Linear Algebra. In *Computer Science Logic*, pages 245–259. Springer, 2010.
- [7] S. Cook and P. Nguyen. *Logical foundations of proof complexity*. Cambridge University Press, 2010.
- [8] S.A. Cook. Feasibly constructive proofs and the propositional calculus (preliminary version). In *Proceedings of the seventh annual ACM symposium on Theory of computing*, pages 83–97. ACM, 1975.
- [9] J. Edmonds. Systems of distinct representatives and linear algebra. *J. Res. Natl. Bureau of Standards*, 71A:241–245, 1967.
- [10] O.H. Ibarra and S. Moran. Probabilistic Algorithms for Deciding Equivalence of Straight-Line Programs. *Journal of the ACM (JACM)*, 30(1):217–228, 1983.
- [11] R. Impagliazzo and V. Kabanets. Constructive proofs of concentration bounds. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 617–631, 2010.
- [12] R. Impagliazzo and A. Wigderson. P= BPP if E requires exponential circuits: Derandomizing the XOR lemma. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 220–229. ACM, 1997.
- [13] E. Jeřábek. Dual weak pigeonhole principle, Boolean complexity, and derandomization. *Annals of Pure and Applied Logic*, 129(1–3):1–37, 2004.
- [14] E. Jeřábek. *Weak pigeonhole principle, and randomized computation*. PhD thesis, Faculty of Mathematics and Physics, Charles University, Prague, 2005.
- [15] E. Jeřábek. Approximate counting in bounded arithmetic. *Journal of Symbolic Logic*, 72(3):959–993, 2007.
- [16] B.H. Korte and J. Vygen. *Combinatorial optimization: theory and algorithms*. Springer Verlag, 2008.
- [17] J. Krajíček. *Bounded arithmetic, propositional logic, and complexity theory*. Cambridge University Press, 1995.
- [18] Jan Krajíček, Pavel Pudlák, and Gaisi Takeuti. Bounded Arithmetic and the Polynomial Hierarchy. *Annals of Pure and Applied Logic*, 52:143–153, 1991.
- [19] H.W. Kuhn. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.
- [20] L. Lovász. On determinants, matchings and random algorithms. In L. Budach, editor, *Fundamentals of Computation Theory*, pages 565–574. Akademia-Verlag, 1979.
- [21] R.A. Moser and G. Tardos. A constructive proof of the general Lovász Local Lemma. *Journal of the ACM (JACM)*, 57(2):1–15, 2010.
- [22] K. Mulmuley, U.V. Vazirani, and V.V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7(1):105–113, 1987.
- [23] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, pages 32–38, 1957.
- [24] N. Nisan and A. Wigderson. Hardness vs randomness. *Journal of Computer and System Sciences*, 49(2):149–167, 1994.

- [25] J.B. Paris, A.J. Wilkie, and A.R. Woods. Provability of the pigeonhole principle and the existence of infinitely many primes. *Journal of Symbolic Logic*, 53(4):1235–1244, 1988.
- [26] P. Pudlák. Ramsey’s theorem in bounded arithmetic. In *Computer Science Logic*, pages 308–317. Springer, 1991.
- [27] J.T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM (JACM)*, 27(4):717, 1980.
- [28] A. Shamir. $IP = PSPACE$. *Journal of the ACM*, 39(4):869–877, 1992.
- [29] M. Soltys and S. Cook. The proof complexity of linear algebra. *Annals of Pure and Applied Logic*, 130:277–323, 2004.
- [30] N. Thapen. *The weak pigeonhole principle in models of bounded arithmetic*. PhD thesis, University of Oxford, 2002.
- [31] S. Toda. PP is as Hard as the Polynomial-Time Hierarchy. *SIAM Journal on Computing*, 20(5):865–877, 1991.
- [32] H. Wasserman and M. Blum. Software reliability via run-time result-checking. *Journal of the ACM*, 44(6):826–849, 1997.
- [33] R. Zippel. An explicit separation of relativised random polynomial time and relativised deterministic polynomial time. *Information Processing Letters*, 33(4):207–212, 1989.