Choiceless Computation and Symmetry

Benjamin Rossman*

June 6, 2010

Abstract

Many natural problems in computer science concern structures like graphs where elements are not inherently ordered. In contrast, Turing machines and other common models of computation operate on strings. While graphs may be *encoded* as strings (via an adjacency matrix), the encoding imposes a linear order on vertices. This enables a Turing machine operating on encodings of graphs to *choose* an arbitrary element from any nonempty set of vertices at low cost (the Augmenting Paths algorithm for Bipartite Matching being an example of the power of choice). However, the outcome of a computation is liable to depend on the external linear order (i.e., the choice of encoding). Moreover, isomorphism-invariance/encoding-independence is an undecidable property of Turing machines. This trouble with encodings led Blass, Gurevich and Shelah [3] to propose a model of computation known as *BGS machines* that operate directly on structures. BGS machines preserve symmetry at every step in a computation, sacrificing the ability to make arbitrary choices between indistinguishable elements of the input structure (hence "choiceless computation"). Blass et al. also introduced a complexity class CPT+C (Choiceless Polynomial Time with Counting) defined in terms of polynomially bounded BGS machines. While every property finite structures in CPT+C is polynomial-time computable in the usual sense, it is open whether conversely every isomorphism-invariant property in P belongs to CPT+C. In this paper we give evidence that CPT+C $\neq$ P by proving the separation of the corresponding classes of *function problems*. Specifically, we show that there is an isomorphism-invariant polynomial-time computable function problem on finite vector spaces ("given a finite vector space V, output the set of hyperplanes in V") that is not computable by any CPT+C program. In addition, we give a new simplified proof of the Support Theorem, which is a key step in the result of [3] that a weak version of CPT+C absent counting cannot decide the parity of sets.

1 Introduction

Is there a "logic" capturing exactly the polynomial-time computable properties of finite structures? This question was raised by Gurevich [9] in the mid 80s, nearly a decade after Fagin [8] showed that the NP properties of finite structures are precisely what can be defined in existential second-order logic. Today this question remains a central open problem in finite model theory.

Addressing this question, Blass, Gurevich and Shelah [3, 4] introduced an logic/complexity class known as CPT+C or Choiceless Polynomial Time with Counting. CPT+C is based on a model of computation known as *BGS machines* (after the inventors). BGS machines operate

*Computer Science and Artificial Intelligence Laboratory, MIT. Supported by the NSF Graduate Research Fellowship.

directly on structures in a manner that preserves symmetry at every step of a computation. By contrast, Turing machines encode structures as strings. This encoding violates the symmetry of structures like graphs (which might possess nontrivial automorphisms) by imposing a linear order on elements. Note that Turing machines are able to exploit this linear order to efficiently *choose* an element from any set constructed in the course of a computation. Thus, it is not uncommon in the high-level description of an algorithm (say, the well-known Augmenting Paths algorithm for Bipartite Matching) to read something along the lines of "let w be any unmatched neighbor of the vertex v". A description like this (implicit) carries a claim that the ultimate outcome of the computation will not dependent on the choice of w. However, the validity of such claims cannot be taken for granted: by Rice's Theorem, encoding-invariance is an undecidable property of Turing machines.

The BGS machine model of computation is said to be "choiceless" because it disallows choices which violate the inherent symmetry of the input structure. Pseudo-instructions of the form "let i be an arbitrary element of the set I" is forbidden. Similarly, "let w be the first neighbor of v" is meaningless (unless referring to an explicitly constructed linear order on vertices). The inability of BGS machines to choose is compensated by *parallelism* (the power to explore all choices in parallel) and the machinery of set theory (the power to build sets using comprehension).

BGS machines may in fact be viewed as the syntactic elements (i.e., formulas) of a logic, whose semantics is well-defined on any structure. One rough description BGS logic is:

> *BGS logic = propositional logic +*
>
> - *a least-fixed-point operator*
> - *a cardinality operator*
> - *basic set-theoretic predicates ($\in$, $\cup$, etc.) and*
> - *comprehension terms of the form* $\{\mathsf{term}_1(x) : x \in \mathsf{term}_2 : \mathsf{formula}(x)\}$
>
> *evaluated in the domain* $\mathrm{HF}(\mathcal{A}) = A \cup \wp(A) \cup \wp(A \cup \wp(A)) \cup \cdots$ *of hereditarily finite objects over a structure $\mathcal{A}$ with universe A.*

The complexity class CPT+C or Choiceless Polynomial Time with Counting (itself also a logic) is obtained by imposing polynomial bounds on the running time and number of processors required to run a BGS machine. (Somewhat more precisely: this involves requiring that fixed-points converge in polynomially many steps and that only polynomial many elements of $\mathrm{HF}(\mathcal{A})$ participate in the evaluation of a formula.) Every CPT+C computable property of structures like graphs can be implemented on a polynomial-time Turing machine (on encodings on such structures). Thus, CPT+C $\subseteq$ P. However, it is open whether, conversely, every isomorphism-invariant polynomial-time property of finite structures is computable in CPT+C.

The main result of this paper (Theorem 6.1) suggests that CPT+C $\neq$ P. We show that there is an isomorphism-invariant polynomial-time computable function problem on finite vector spaces ("given a finite vector space V, output the set of hyperplanes in V") that is not computable by any CPT+C program. An additional result of this paper (Theorem 5.1) is a new simplified proof of the Support Theorem of [3], which is a key step in the result of [3] that a weak version of CPT+C absent counting cannot decide the parity of sets.

Outline of the paper. In §2 we present all relevant definitions. §3 contains a brief summary of the key results on CPT and CPT+C from previous work of Blass et al. [3] and others. In §4 we

introduce some new notions relating BGS machines to the automorphism groups of structures. In §5 we prove that the symmetric group has a certain "support property", leading to a simplified proof of the Support Theorem from the original paper of Blass et al. [3]. In §6 we establish a similar result for the general linear group; as a corollary, we show that there is a polynomial-time computable *function problem* on finite vector spaces that is not computable by any CPT+C program.

2 Definitions

We begin by defining *hereditarily finite expansions* of structures in §2.1. We then define BGS logic and classes CPT and CPT+C in §2.2. The definition of BGS logic presented here differs from the BGS machines of Blass, et al. [3], but classes CPT and CPT+C are exactly the same. BGS logic has a bare-bones syntax that is well-suited for induction, whereas the BGS machines of [3] have an intuitive and attractive syntax (borrowing from Abstract State Machines) that is recommended for the actual description of CPT+C algorithms (see [3] for examples). Let us also mention that CPT(+C) is elsewhere [3, 4, 7] written as $\widetilde{\text{C}}$PT(+C), the tilde over C evoking the "less" in "choiceless".

2.1 Hereditarily Finite Expansion

For every structure $\mathcal{A}$ in a signature σ, we define a structure $\text{HF}(\mathcal{A})$ in an enlarged signature σ^{HF} called the *hereditarily finite expansion* of $\mathcal{A}$.

Definition 2.1 (Hereditarily Finite Objects)**.** Let A be for a set (a special case being the universe of a structure $\mathcal{A}$) whose elements we call *atoms* and assume to be non-set entities (or "ur-elements").

1. $\text{HF}(A)$ denote the (unique) smallest set such that $A \subseteq \text{HF}(A)$ and $B \in \text{HF}(A)$ for every finite subset B of $\text{HF}(A)$. Elements of $\text{HF}(A)$ are called *hereditarily finite objects* (*h.f. objects*) *over* A and elements of $\text{HF}(A) \setminus A$ (i.e., elements of $\text{HF}(A)$ which are sets) are called *hereditarily finite sets* (*h.f. sets*) *over* A. (Note that if A is finite, then $\text{HF}(A)$ is the countable union $A \cup \wp(A) \cup \wp(A \cup \wp(A)) \cup \wp(A \cup \wp(A) \cup \wp(A \cup \wp(A))) \cup \cdots$.)

2. The *rank* of a h.f. object x is defined as 0 if $x \in A \cup \{\emptyset\}$ and $1 + \max_{y \in x} \text{rank}(y)$ otherwise.

3. A h.f. set x is *transitive* if $y \subseteq x$ for all $y \in x$ such that y is a set. The *transitive closure* $\text{TC}(x)$ of a h.f. object x is the (unique) smallest transitive set containing x as an element.

4. The *finite von Neumann ordinals* κ_i for $i < \omega$ are elements of $\text{HF}(\emptyset)$ defined by $\kappa_0 = \emptyset$ and $\kappa_{i+1} = \{\kappa_0, \ldots, \kappa_i\}$.

Definition 2.2 (Group Actions)**.** For a group G acting on A (a special case being the action of $\text{Aut}(\mathcal{A})$ on the universe of a structure $\mathcal{A}$), we shall consider the extension of this action to $\text{HF}(A)$ defined for $g \in G$ and $x \in \text{HF}(A) \setminus A$ inductively by $gx = \{gy : y \in x\}$.

Definition 2.3 (Signatures and Structures)**.** A *signature* σ consists of relation symbols R_i and function symbols f_j (with designated arities) as well as constant symbols c_k. A *σ-structure* $\mathcal{A}$ (or *structure with signature* σ) consists of a set A (called the *universe* of $\mathcal{A}$) together with interpretations for the various symbols in σ, that is, relations $R_i^{\mathcal{A}} \subseteq A^{\text{arity}(R_i)}$, functions $f_j^{\mathcal{A}} : A^{\text{arity}(f_j)} \to A$ and constants $c_k^{\mathcal{A}} \in A$ (or simply R_i, f_j and c_k when $\mathcal{A}$ is known from context).

Definition 2.4 (Hereditarily Finite Expansion)**.** σ^{HF} denotes the disjoint union of σ and $\{\mathsf{In}, \mathsf{EmptySet}, \mathsf{Atoms}, \mathsf{Union}, \mathsf{Pair}, \mathsf{TheUnique}, \mathsf{Card}\}$ where In is a binary relation symbol, $\mathsf{EmptySet}$ and Atoms are constant symbols, Union, $\mathsf{TheUnique}$ and Card unary function symbols, and Pair is a binary function symbol. For a finite σ-structure $\mathcal{A}$, let $\mathrm{HF}(\mathcal{A})$ denote the σ^{HF}-structure with universe $\mathrm{HF}(A)$ where:

- symbols from σ have the same interpretation in $\mathrm{HF}(\mathcal{A})$ as in $\mathcal{A}$, with the convention that functions from σ take value $\emptyset$ whenever any coordinate of the input is not an atom,
- $(x, y) \in \mathsf{In}$ if and only if $x \in y$,
- $\mathsf{EmptySet} = \emptyset, \quad \mathsf{Atoms} = A, \quad \mathsf{Pair}(x, y) = \{x, y\}$,
- $\mathsf{Union}(x) = \bigcup_{y \in x} y$ (in particular, $\mathsf{Union}(x) = \emptyset$ if $x \in \mathsf{Atoms} \cup \{\emptyset\}$),
- $\mathsf{TheUnique}(x) = \begin{cases} y & \text{if } x = \{y\}, \\ \emptyset & \text{if } x \text{ is not a singleton,} \end{cases}$
- $\mathsf{Card}(x) = \begin{cases} |x| \text{ as a von Neumann ordinal} & \text{if } x \text{ is a set,} \\ \emptyset & \text{if } x \text{ is an atom.} \end{cases}$

$\mathrm{HF}(\mathcal{A})$ is called the *hereditarily finite expansion* of $\mathcal{A}$. Structures of the form $\mathrm{HF}(\mathcal{A})$ are called *hereditarily finite structures*.

2.2 BGS Logic and CPT+C

Just like first-order logic, BGS logic (and its weaker cousin BGS^- logic) are defined with respect to a fixed signature σ. Similarly, BGS logic has terms and formulas. However, BGS logic has an additional syntactic element called *programs* (which compute a term $t(t(t(...t(\emptyset)\dots)))$ iteratively using a "step" term $t(\cdot)$ until some "halting" formula is satisfied, whereupon an "output" term is computed).

Definition 2.5 (BGS Logic)**.** ***BGS logic*** over a signature σ consists of *terms*, *formulas* and *programs*, defined below.

1. ***Terms*** and ***formulas*** are defined inductively:
   - (base case) variables are terms;
   - (base case) constant symbols in σ^{HF} are terms;
   - if f is an r-ary function symbol in σ^{HF} and $t_1, \dots, t_r$ are terms, then $f(t_1, \dots, t_r)$ is a term;
   - if R is an r-ary relation symbol in σ^{HF} and $t_1, \dots, t_r$ are terms, then $R(t_1, \dots, t_r)$ is a formula;
   - if t_1 and t_2 are terms, then $t_1 = t_2$ is a formula;
   - if ϕ_1 and ϕ_2 are formulas, then so are $\neg\phi_1$ and $\phi_1 \wedge \phi_2$ and $\phi_1 \vee \phi_2$;
   - if s and t are terms, v is a variable (which is not free in t) and ϕ is a formula, then $\{s(v) : v \in t : \phi(v)\}$ is a term ("the set of $s(v)$ for $v \in t$ such that $\phi(v)$ is true").

Terms of the form $\{s(v) : v \in t : \phi(v)\}$ are called *comprehension terms*.

2. Each occurrence of a variable in a term or formula is either *free* or *bound*, with the comprehension construct $\{s(v) : v \in t : \phi(v)\}$ binding the free occurrences of v within s and ϕ. As a matter of notation, we write $t(v_1, \ldots, v_\ell)$ or $\phi(v_1, \ldots, v_\ell)$ for a term or formula whose free variables are contained among $v_1, \ldots, v_\ell$. A term or formula with no free variables is said to be *ground*.

The *variable rank* of a term t (resp. formula φ) is the maximum number of free variables in any subterm/formula of t (resp. φ).

3. Terms and formulas have the obvious semantics when evaluated on hereditary finite structures with free variables assigned to h.f. objects. For a term $t(v_1, \ldots, v_\ell)$ and elements $x_1, \ldots, x_\ell \in \mathrm{HF}(A)$, the value of t when free variables $v_1, \ldots, v_\ell$ are assigned to $x_1, \ldots, x_\ell$ is denoted by $[\![t(\vec{x})]\!]^{\mathcal{A}} \in \mathrm{HF}(A)$ (or simply $[\![t(\vec{x})]\!]$ if $\mathcal{A}$ is known from context). For a formula $\varphi(v_1, \ldots, v_\ell)$, the value $[\![\varphi(\vec{x})]\!]^{\mathcal{A}}$ is an element of $\{\mathsf{True}, \mathsf{False}\}$ where (following [3]) we identify $\mathsf{True} = \{\emptyset\}$ and $\mathsf{False} = \emptyset$.

We omit a rigorous definition of the semantic operator $[\![\cdot]\!]$. Let us however explicitly state the semantics of comprehension terms (since this construct may be less familiar). For a comprehension term $r(\vec{v}) = \{s(\vec{v}, w) : w \in t(\vec{v}) : \varphi(\vec{v}, w)\}$ and parameters $\vec{x}$ from $\mathrm{HF}(A)$, the value $[\![r(\vec{x})]\!]$ is defined (in the obvious way) as the set of $[\![s(\vec{x}, y)]\!]$ for $y \in [\![t(\vec{x})]\!]$ such that $[\![\varphi(\vec{x}, y)]\!] = \mathsf{True}$.

4. A ***program*** $\Pi = (\Pi_{\mathrm{step}}, \Pi_{\mathrm{halt}}, \Pi_{\mathrm{out}})$ consists of a term $\Pi_{\mathrm{step}}(v)$, a formula $\Pi_{\mathrm{halt}}(v)$, and a term or formula $\Pi_{\mathrm{out}}(v)$ (depending whether Π computes a decision problem or produces an output) all with a single free variable v. If Π_{out} is a formula, then Π is said to be *Boolean*.

The *variable rank* of Π is the maximum of the variable ranks of Π_{step}, Π_{halt} and Π_{out}.

5. A program Π is executed on an input structure $\mathcal{A}$ as follows. Let $x_0 = \emptyset$ and $x_{i+1} = [\![\Pi_{\mathrm{step}}(x_i)]\!]$ for all $i < \omega$. In the event that $[\![\Pi_{\mathrm{halt}}(x_i)]\!] = \mathsf{False}$ for all i, let $\Pi(\mathcal{A}) = \perp$ (and the computation is said to *diverge*). Otherwise, let $\Pi(\mathcal{A}) = [\![\Pi_{\mathrm{out}}(x_i)]\!]$ for the minimal i such that $[\![\Pi_{\mathrm{halt}}(x_i)]\!] = \mathsf{True}$.

6. ***BGS$^-$ logic*** consists of all terms, formulas and programs of BGS logic which exclude unary function symbol Card.

We remark that BGS logic has the ability to carry out bounded existential and universal quantification in $\mathrm{HF}(\mathcal{A})$ (and thus subsumes first-order logic over the base structure $\mathcal{A}$). To see this, note that $(\exists v \in t)\, \phi(v)$ is equivalent to the formula $\{\emptyset : v \in t : \phi(v)\} = \{\emptyset\}$ (pedantically, we should write $\mathsf{EmptySet}$ instead of $\emptyset$ and $\mathsf{Pair}(\mathsf{EmptySet}, \mathsf{EmptySet})$ instead of $\{\emptyset\}$). Similarly, $(\forall v \in t)\, \phi(v)$ is equivalent to the formula $\{\emptyset : v \in t : \neg\phi(v)\} = \emptyset$.

We now define the crucial resource by which we measure the complexity of BGS programs. Informally, a h.f. object $x \in \mathrm{HF}(A)$ is *active* for the operation of a program Π on a structure $\mathcal{A}$ if x is the value of any term involved in the computation of Π on $\mathcal{A}$ (until a halt state is reached). By setting a polynomial bound on the number of active objects, as well as requiring programs to halt on every input, we arrive at classes CPT and CPT+C.

Definition 2.6 (Active Objects and Classes CPT and CPT+C)**.** As in the definition of $[\![\cdot]\!]^{\mathcal{A}}$, we omit the superscript from the *active-element operator* $\langle\!\langle\cdot\rangle\!\rangle^{\mathcal{A}}$ (defined below) when the structure $\mathcal{A}$ is clear from context (as below).

1. For every term $t(v_1,\dots,v_\ell)$ and assignment of free variables to values $x_1,\dots,x_\ell \in \mathrm{HF}(A)$, we define $\langle\!\langle t(x_1,\dots,x_\ell)\rangle\!\rangle \in \mathrm{HF}(A)$ inductively as follows:

- (base case) if term $t(v)$ is precisely the variable v, then $\langle\!\langle t(x)\rangle\!\rangle = \{x\}$ for every $x \in \mathrm{HF}(A)$;
- (base case) if c is a constant symbol in σ^{HF}, then $\langle\!\langle c\rangle\!\rangle = \{c^{\mathrm{HF}(\mathcal{A})}\}$;
- if R is an r-ary relation symbol in σ^{HF} and $t_1,\dots,t_r$ are terms, then

$$\langle\!\langle R(t_1,\dots,t_r)\rangle\!\rangle = \textstyle\bigcup_{i=1}^{r} \langle\!\langle t_i\rangle\!\rangle;$$

- if f is an r-ary function symbol in σ^{HF} and $t_1,\dots,t_r$ are terms, then

$$\langle\!\langle f(t_1,\dots,t_r)\rangle\!\rangle = \big\{[\![f(t_1,\dots,t_r)]\!]\big\} \cup \textstyle\bigcup_{i=1}^{r} \langle\!\langle t_i\rangle\!\rangle;$$

- for logical connectives,

$$\langle\!\langle \neg\phi\rangle\!\rangle = \langle\!\langle \phi\rangle\!\rangle, \quad \langle\!\langle \phi \wedge \psi\rangle\!\rangle = \langle\!\langle \phi \vee \psi\rangle\!\rangle = \langle\!\langle \phi\rangle\!\rangle \cup \langle\!\langle \psi\rangle\!\rangle, \quad \langle\!\langle t_1 = t_2\rangle\!\rangle = \langle\!\langle t_1\rangle\!\rangle \cup \langle\!\langle t_2\rangle\!\rangle;$$

- for comprehension terms,[1]

$$\langle\!\langle \{s(v) : v \in t : \phi(v)\}\rangle\!\rangle = \big\{[\![\{s(v) : v \in t : \phi(v)\}]\!]\big\} \cup \langle\!\langle t\rangle\!\rangle \cup \textstyle\bigcup_{x \in [\![t]\!]} \big(\langle\!\langle \phi(x)\rangle\!\rangle \cup \langle\!\langle s(x)\rangle\!\rangle\big).$$

2. The set $\mathrm{Active}(\Pi, \mathcal{A}) \subseteq \mathrm{HF}(A)$ of *active objects of* Π *on* $\mathcal{A}$ is defined by

$$\mathrm{Active}(\Pi,\mathcal{A}) = \begin{cases} \bigcup_{i<\omega} \langle\!\langle \Pi_{\mathrm{step}}(x_i)\rangle\!\rangle \cup \langle\!\langle \Pi_{\mathrm{halt}}(x_i)\rangle\!\rangle & \text{if } \Pi(\mathcal{A}) = \bot, \\ \langle\!\langle \Pi_{\mathrm{halt}}(x_t)\rangle\!\rangle \cup \langle\!\langle \Pi_{\mathrm{out}}(x_t)\rangle\!\rangle \cup \bigcup_{i=0}^{t-1} \big(\langle\!\langle \Pi_{\mathrm{step}}(x_i)\rangle\!\rangle \cup \langle\!\langle \Pi_{\mathrm{halt}}(x_i)\rangle\!\rangle\big) & \text{otherwise,} \end{cases}$$

where $x_0 = \emptyset$, $x_{i+1} = [\![\Pi_{\mathrm{step}}(x_i)]\!]$ and t is the least nonnegative integer such that $[\![\Pi_{\mathrm{halt}}(x_t)]\!] = \mathsf{True}$.

3. For a function $f(n)$, we denote by $\mathrm{BGS}^{(-)}(f(n))$ the class of $\mathrm{BGS}^{(-)}$ programs Π such that $\Pi(\mathcal{A}) \neq \bot$ and $|\mathrm{Active}(\Pi,\mathcal{A})| \leqslant f(|A|)$ for all finite structures $\mathcal{A}$.

4. Classes CPT and CPT+C are defined by

$$\mathrm{CPT} = \mathrm{BGS}^{-}(n^{O(1)}), \qquad \mathrm{CPT{+}C} = \mathrm{BGS}(n^{O(1)}).$$

We also denote by CPT(+C) the "complexity class" consisting of classes ("languages") of finite structures recognized by a Boolean program $\Pi \in \mathrm{CPT}(+\mathrm{C})$. That is, for a class $\mathscr{C}$ of finite structures, we write $\mathscr{C} \in \mathrm{CPT}(+\mathrm{C})$ if and only if $\mathscr{C} = \{\mathcal{A} : \Pi(\mathcal{A}) = \mathsf{True}\}$ for some CPT(+C) program Π.

[1] There is a reasonable alternative:

$$\langle\!\langle \{s(v) : v \in t : \phi(v)\}\rangle\!\rangle = \big\{[\![\{s(v) : v \in t : \phi(v)\}]\!]\big\} \cup \langle\!\langle t\rangle\!\rangle \cup \textstyle\bigcup_{x \in [\![t]\!]} \langle\!\langle \phi(x)\rangle\!\rangle \cup \bigcup_{x \in [\![t]\!] \,:\, [\![\phi(x)]\!] = \mathsf{True}} \langle\!\langle s(x)\rangle\!\rangle.$$

For the purposes of this paper, either definition is fine (i.e., all results hold just the same).

3 Brief Survey of Results on CPT and CPT+C

For background we give a brief and partial survey of results on CPT and CPT+C. Theorems 3.1, 3.2, 3.3 are from the original paper [3] of Blass, Gurevich and Shelah.

Theorem 3.1. CPT+C $\subseteq$ P.

The idea behind the proof is that a CPT+C program Π can be simulated by a polynomial-time dynamic programming algorithm. The "subproblems" in the dynamic program correspond to terms and formulas occurring in the evaluation of Π, together with assignments of free variables to h.f. objects. The key observation is that, while these terms and formulas like $\Pi_{\text{step}}(\Pi_{\text{step}}(\dots\Pi_{\text{step}}(v)\dots))$ may grow polynomially long, there can only be polynomial many such terms and formulas and, moreover, the number of free variables in any subformula is bounded by a constant (the variable rank of Π).

Theorem 3.2. CPT = CPT+C = P *on structures with a built-in linear order.*

Theorem 3.2 uses the fact that least-fixed-point logic LFP is a subclass of CPT (this is shown in [3]) and that LFP = P on structures with a built-in linear order [10, 12].

Theorem 3.3. *The class* PARITY *of finite sets of even cardinality is not definable in* CPT.

Theorem 3.3 moreover shows that CPT $\neq$ CPT+C, since PARITY is clearly definable in CPT+C. In a significantly strengthening of Theorem 3.3, Shelah [11] proved that CPT has a "zero-one law" (see [1] for an alternative exposition of this result):

Theorem 3.4. *For every relational signature σ and every* CPT*-definable property $\mathcal{P}$ of σ-structures, the probability that a uniform random σ-structure of cardinality n has property $\mathcal{P}$ tends (as a function of n) to* 0 *or* 1.

Although it is open whether CPT+C = P, a number of "choiceless" algorithms have been devised which solve some particular P problems on unordered structures in new and surprising ways.

- The paper [4] contains a CPT+C algorithm for Bipartite Matching among other problems.
- In [2] it is explained how "choicelessly" to implement the Csanky algorithm for computing the determinant of an *unordered matrix*, that is, a function $I \times I \to F$ where I is a finite (unordered) set and F is a finite field.
- The paper [7] presents a CPT algorithm that solves a tractable special case of Graph Isomorphism due to Cai, Fürer and Immerman [5]. (The Cai-Fürer-Immerman problem was used to show that LFP+C $\neq$ P, that is, first-order logic with least-fixed-point and counting operators does not capture P.)

Among polynomial-time problems for which no CPT+C algorithm is known is Perfect Matching (see [4]).

4 The Role of Symmetry

This section introduces a tool for showing that certain h.f. objects in certain structures (with rich automorphism groups) cannot be activated by any CPT+C program.

Definition 4.1. Let G be a group acting faithfully on a finite set A of cardinality n (a special case is $\mathrm{Aut}(\mathcal{A})$ acting on the universe of a structure $\mathcal{A}$). Recall that the action of G extends to $\mathrm{HF}(A)$ via $gx = \{gy : y \in x\}$ for sets $x \in \mathrm{HF}(A) \setminus A$.

1. For $x \in \mathrm{HF}(A)$, the *stabilizer* of x is the subgroup of G defined by $\mathrm{Stab}(x) = \{g \in G : gx = x\}$. If x is a set, the *pointwise stabilizer* of x is the subgroup of G defined by $\mathrm{Stab}^\bullet(x) = \{g \in G : gy = y \text{ for all } y \in x\}$.

2. A set of atoms $B \subseteq A$ is a *support* for a subgroup $H \subseteq G$ if $\mathrm{Stab}^\bullet(B) \subseteq H$. If H has a support of size $\leqslant k$, then H is said to be *k-supported*.

3. For all k and r, the set of *(k,r)-constructible subgroups* of G is the minimal family of subgroups of G such that

   - every k-supported subgroup is (k,r)-constructible, and
   - if $H_1, \ldots, H_r$ are (k,r)-constructible, $H_1 \cap \cdots \cap H_r \subseteq H$ and $[G : H] \leqslant n^k$, then H is (k,r)-constructible.[2]

4. G has the *(k,r)-support property* if every (k,r)-constructible subgroup is k-supported.

We extend the notion of (k,r)-constructibility from subgroups of G to elements of $\mathrm{HF}(A)$.

5. A h.f. object $x \in \mathrm{HF}(A)$ is *(k,r)-constructible* if $\mathrm{Supp}(y)$ is a (k,r)-constructible subgroup of G for all $y \in \mathrm{TC}(x)$.

Note that (k,r)-constructibility is a transitive property: if a h.f. set x is (k,r)-constructible, then so are all $y \in x$. Whenever we speak about "(k,r)-constructible" elements of $\mathrm{HF}(A)$ in the context of a structure $\mathcal{A}$ without mentioning G, let it be understood that G is the automorphism group $\mathrm{Aut}(\mathcal{A})$.

The following lemma gives a condition equivalent to the (k,r)-support property.

Lemma 4.2. *G has the (k,r)-support property if, and only if, every kr-supported subgroup with index $\leqslant n^k$ is k-supported.*

Proof. ($\Longrightarrow$) Suppose G has the (k,r)-support property and H is kr-supported and $[G : H] \leqslant n^k$. Let $B \subseteq A$ be a support for H of size $|B| \leqslant kr$. Fix an arbitrary partition $B = B_1 \cup \cdots \cup B_r$ into sets of size $|B_i| \leqslant k$. For $i \in \{1, \ldots, r\}$, let $H_i = \mathrm{Stab}^\bullet(B_i)$ and note that H_i is (k,r)-constructible (since every k-supported subgroup is (k,r)-constructible). Since $H_1 \cap \cdots \cap H_r = \mathrm{Stab}^\bullet(B) \subseteq H$ and $[G : H] \leqslant n^k$, it follows that H is (k,r)-constructible.

($\Longleftarrow$) Suppose that every kr-supported subgroup with index $\leqslant n^k$ is k-supported. To show that every (k,r)-constructible subgroup is k-supported, assume $H_1, \ldots, H_r$ are k-supported and $H_1 \cap \cdots \cap H_r \subseteq H$ and $[G : H] \leqslant n^k$. It suffices to show that H is k-supported. But note that H is kr-supported (the union of the supports for $H_1, \ldots, H_r$ of size $\leqslant k$ is a support for H). So H is k-supported by assumption. □

[2] Recall that the *index of H in G* is defined by $[G : H] = |G|/|H|$.

The following proposition links these concepts to CPT+C.

Proposition 4.3. *Suppose* Π *is a non-Boolean program in* $\mathrm{BGS}(n^k)$ *with variable rank* $\leqslant r$*. Then* $\Pi(\mathcal{C})$ *is* (k,r)*-constructible for every finite structure* $\mathcal{A}$*.*

The proof follows after two lemma. The first lemma states that the semantics of terms and formulas respects automorphisms of $\mathcal{A}$ in the expected way.

Lemma 4.4. *Let* $\gamma(v_1,\ldots,v_\ell)$ *be any term or formula of BGS logic. For every structure* $\mathcal{A}$*, automorphism* $\alpha \in \mathrm{Aut}(\mathcal{A})$ *and elements* $x_1,\ldots,x_\ell \in \mathrm{HF}(A)$*,*

$$[\![\gamma(\alpha x_1,\ldots,\alpha x_\ell)]\!] = \alpha[\![\gamma(x_1,\ldots,x_\ell)]\!] \quad \textit{and} \quad \langle\!\langle \gamma(\alpha x_1,\ldots,\alpha x_\ell)\rangle\!\rangle = \alpha\langle\!\langle \gamma(x_1,\ldots,x_\ell)\rangle\!\rangle.$$

In particular, $\mathrm{Stab}([\![\gamma]\!]) = \mathrm{Stab}(\langle\!\langle\gamma\rangle\!\rangle) = \mathrm{Aut}(\mathcal{A})$ *for every ground term or formula* γ*.*

The proof (omitted) is a straightforward induction on terms and formulas.

Lemma 4.5. *Suppose* $t(v_1,\ldots,v_\ell)$ *is a term with variable rank* $\leqslant r$ *and* $x_1,\ldots,x_\ell$ *are* (k,r)*-constructible elements of* $\mathrm{HF}(A)$ *such that* $|\{\alpha y : y \in \langle\!\langle t(x_1,\ldots,x_\ell)\rangle\!\rangle, \alpha \in \mathrm{Aut}(\mathcal{A})\}| \leqslant n^k$*. Then* $[\![t(x_1,\ldots,x_\ell)]\!]$ *is* (k,r)*-constructible.*

Proof. The proof is by induction on terms. The bases cases are when t is a constant symbol or a variable; both cases are trivial. For the induction step, we consider the various types of term constructs (see Definition 2.5(1)), namely when t is:

(i) $f(t_1(\vec{v}),\ldots,t_m(\vec{v}))$ where f is an m-ary function symbol f in the signature of $\mathcal{A}$,

(ii) $\mathsf{Pair}(t_1(\vec{v}), t_2(\vec{v}))$,

(iii) $\mathsf{TheUnique}(t_1(\vec{v}))$,

(iv) $\mathsf{Union}(t_1(\vec{v}))$, or

(v) $\{s(\vec{v},w) : w \in t_1(\vec{v}) : \varphi(\vec{v},w)\}$ (i.e., a comprehension term with subterms $t_1(\vec{v})$ and $s(\vec{v},w)$ and subformula $\varphi(\vec{v},w)$).

That is, in each case we assume that the lemma holds for subterms $t_1, t_2, \ldots$ (as well as s in case (v)) and prove that $[\![t(\vec{x})]\!]$ is (k,r)-constructible. For this, it is sufficient to show: first, that every element of $[\![t(\vec{x})]\!]$ is (k,r)-constructible; and second, that $\mathrm{Stab}([\![t(\vec{x})]\!])$ is a (k,r)-constructible subgroup of $\mathrm{Aut}(\mathcal{A})$ (using Lemma 4.4).

As for the first claim that every element of $[\![t(\vec{x})]\!]$ is (k,r)-constructible, we consider cases (i)–(v) separately. Note that every subterm t_i of t has variable rank $\leqslant r$ and satisfies $|\{\alpha y : y \in \langle\!\langle t_i(\vec{x})\rangle\!\rangle, \alpha \in \mathrm{Aut}(\mathcal{A})\}| \leqslant n^k$ since $\langle\!\langle t_i(\vec{x})\rangle\!\rangle \subseteq \langle\!\langle t(\vec{x})\rangle\!\rangle$; therefore, $[\![t_i(\vec{x})]\!]$ is (k,r)-constructible by the induction hypothesis.

- Case (i): $[\![f(t_1(\vec{x}),\ldots,t_m(\vec{x}))]\!]$ is either an atom (if $[\![t_1(\vec{x})]\!],\ldots,[\![t_m(\vec{x})]\!]$ are all atoms) or $\emptyset$ (otherwise). In either case, $[\![f(t_1(\vec{x}),\ldots,t_m(\vec{x}))]\!]$ is (k,r)-constructible.
- Case (ii): By the induction hypothesis, $[\![t_1(\vec{x})]\!]$ and $[\![t_2(\vec{x})]\!]$ are both (k,r)-constructible.
- Case (iii): If $[\![t_1(\vec{x})]\!]$ is not a singleton, then $[\![\mathsf{TheUnique}(t_1(\vec{x}))]\!] = \emptyset$ and hence is (k,r)-constructible. So assume $[\![t_1(\vec{x})]\!]$ is a singleton $\{y\}$. By the induction hypothesis, $[\![t_1(\vec{x})]\!]$ is (k,r)-constructible. By transitivity of (k,r)-constructibility, y is (k,r)-constructible.

- Case (iv): By the induction hypothesis, $[\![t_1(\vec{x})]\!]$ is (k,r)-constructible. By transitivity of (k,r)-constructibility, all elements of $\bigcup[\![t_1(\vec{x})]\!]$ are (k,r)-constructible.

- Case (v): Suppose $t(\vec{v})$ is a comprehension term $\{t_2(\vec{v},w) : w \in t_1(\vec{v}) : \varphi(\vec{v},w)\}$. Recall that

$$[\![t(\vec{x})]\!] = \{[\![s(\vec{x},y)]\!] : y \in [\![t_1(\vec{x})]\!] \text{ such that } [\![\varphi(\vec{x},y)]\!] = \mathsf{True}\}.$$

  By the induction hypothesis, $[\![t_1(\vec{x})]\!]$ is (k,r)-constructible. Therefore, every $y \in [\![s(\vec{x})]\!]$ is (k,r)-constructible (by transitivity of (k,r)-constructibility); it follows that $[\![s(\vec{x},y)]\!]$ is (k,r)-constructible (by the induction hypothesis on s, noting that s has variable rank $\leqslant r$ and $\langle\!\langle s(\vec{x},y)\rangle\!\rangle \subseteq \langle\!\langle t(\vec{x})\rangle\!\rangle$ so $|\{\alpha z : z \in \langle\!\langle s(\vec{x},y)\rangle\!\rangle, \alpha \in \mathrm{Aut}(\mathcal{A})\}| \leqslant n^k$).

To finish the proof, we prove that $\mathrm{Stab}([\![t(x_1,\ldots,x_\ell)]\!])$ is a (k,r)-constructible subgroup of $\mathrm{Aut}(\mathcal{A})$ (in all cases (i)–(v)). Because t has variable rank $\leqslant r$, at most r of the variables $v_1,\ldots,v_\ell$ occur free in t (this is obvious if $\ell \leqslant r$, but we allow $\ell > r$). Let $j_1,\ldots,j_r \in \{1,...,\ell\}$ be such that $v_{j_1},\ldots,v_{j_r}$ are the only variables which occur free in t. Let $H_i = \mathrm{Stab}(x_{j_i})$ for $i = 1,\ldots,r$. Lemma 4.4 implies that

$$H_1 \cap \cdots \cap H_r \subseteq \mathrm{Stab}([\![t(\vec{x})]\!]),$$

that is, every automorphism of $\mathrm{Aut}(\mathcal{A})$ which fixes each of $x_{j_1},\ldots,x_{j_r}$ also fixes $[\![t(\vec{x})]\!]$. Since $H_1,\ldots,H_r$ are (k,r)-constructible subgroups of $\mathrm{Aut}(\mathcal{A})$ (by the assumption that $x_{j_1},\ldots,x_{j_r}$ are (k,r)-constructible elements of $\mathrm{HF}(\mathcal{A})$) and their intersection is contained in $\mathrm{Stab}([\![t(\vec{x})]\!])$, it suffices to show that $[\mathrm{Aut}(\mathcal{A}) : \mathrm{Stab}([\![t(\vec{x})]\!])] \leqslant n^k$. This follows from our assumption that $|\{\alpha y : y \in \langle\!\langle t(\vec{x})\rangle\!\rangle, \alpha \in \mathrm{Aut}(\mathcal{A})\}| \leqslant n^k$, as we have

$$\begin{aligned}
[\mathrm{Aut}(\mathcal{A}) : \mathrm{Stab}([\![t(\vec{x})]\!])] &= |\{\alpha[\![t(\vec{x})]\!] : \alpha \in \mathrm{Aut}(\mathcal{A})\}| \\
&\leqslant |\{\alpha y : y \in \langle\!\langle t(\vec{x})\rangle\!\rangle, \alpha \in \mathrm{Aut}(\mathcal{A})\}| \quad (\text{since } [\![t(\vec{x})]\!] \in \langle\!\langle t(\vec{x})\rangle\!\rangle) \\
&\leqslant n^k \quad (\text{by assumption}).
\end{aligned}$$

□

Finally, we prove Proposition 4.3 using Lemma 4.5.

Proof of Proposition 4.3. Let Π be a non-Boolean program $\mathrm{BGS}(n^k)$ with variable rank $\leqslant r$. For any finite structure $\mathcal{A}$, note that

$$\Pi(\mathcal{A}) = [\![\Pi_{\mathrm{out}}(\underbrace{\Pi_{\mathrm{step}}(\ldots\Pi_{\mathrm{step}}(\emptyset)\ldots)}_{m \text{ times}})]\!]$$

for some finite m. Let t denote this term $\Pi_{\mathrm{out}}(\Pi_{\mathrm{step}}(\ldots\Pi_{\mathrm{step}}(\emptyset)\ldots))$. Whatever m happens to be, t is a ground term with variable rank $\leqslant r$. By Lemma 4.4, $\langle\!\langle t\rangle\!\rangle$ is fixed by all automorphisms of $\mathcal{A}$ (i.e., $\mathrm{Stab}(\langle\!\langle t\rangle\!\rangle) = \mathrm{Aut}(\mathcal{A})$). Thus,

$$\{\alpha y : y \in \langle\!\langle t\rangle\!\rangle, \alpha \in \mathrm{Aut}(\mathcal{A})\} = \{y : y \in \langle\!\langle t\rangle\!\rangle\} \subseteq \mathrm{Active}(\Pi,\mathcal{A}).$$

Since $|\mathrm{Active}(\Pi,\mathcal{A})| \leqslant n^k$ (by definition of $\mathrm{BGS}(n^k)$), we have $|\{\alpha y : y \in \langle\!\langle t\rangle\!\rangle, \alpha \in \mathrm{Aut}(\mathcal{A})\}| \leqslant n^k$. Therefore, $\Pi(\mathcal{A})$ is (k,r)-constructible by Lemma 4.5. □

In the next two sections, we will use Proposition 4.3 to prove that CPT+C programs cannot activate certain h.f. objects over "naked" sets and vector spaces.

5 PARITY $\notin$ CPT

We denote by $[n]$ the "naked" set $\{1, \ldots, n\}$ viewed as structure in the empty signature. Let PARITY denote the class of naked sets with even cardinality (i.e., the "language" of empty sets). Earlier we stated the result of Blass et al. from [3] that PARITY $\notin$ CPT (Theorem 3.3). A key step in the proof is the following so-called Support Theorem (Theorem 24 of [3]).

Theorem 5.1. *For every* $\Pi \in$ CPT*, there is a constant* c *such that for all sufficiently large* n*, every object in* $\mathrm{Active}(\Pi, [n])$ *has a support of cardinality* $\leqslant c$*.*

The original proof of Theorem 5.1 in [3] involves a fairly intricate combinatorial argument. We give an alternative and simpler proof using the support property defined in the previous section. Theorem 5.1 follows directly from Proposition 4.3 and the following proposition.

Proposition 5.2. *For* $n > 2kr$*, the symmetric group* S_n *has the* (k, r)*-support property.*

We remark that S_{kr+1} fails to have the (k, r)-support property, as the alternating subgroup is (k, r)-constructible but not k-supported (its smallest support has size kr). The following lemma and corollary from [3] are also used in the original proof of Theorem 5.1. We include proofs for completeness.

Lemma 5.3. *Let* $H \subseteq S_n$ *and suppose* $U, V \subset [n]$ *such that* $\mathrm{Stab}^\bullet(U), \mathrm{Stab}^\bullet(V) \subseteq H$ *and* $U \cup V \neq [n]$*. Then* $\mathrm{Stab}^\bullet(U \cap V) \subseteq H$*.*

Proof. $\mathrm{Stab}^\bullet(U \cap V)$ is generated by transpositions $(i\ j)$ where $i, j \in [n] \setminus (U \cap V)$. Therefore, it suffices to show that $(i\ j) \in H$ for all $i, j \in [n] \setminus (U \cap V)$. By assumption, there exists $k \in [n] \setminus (U \cup V)$. Since $(i\ j) = (i\ k)(j\ k)(i\ k)$, it suffices to show that $(i\ k) \in H$ for all $i \in [n] \setminus (U \cap V)$. We consider two cases depending whether $i \notin U$ or $i \notin V$: if $i \notin U$, then $(i\ k) \in \mathrm{Stab}^\bullet(U)$ and hence $(i\ k) \in H$ as $\mathrm{Stab}^\bullet(U) \leqslant H$; if $i \notin V$, then $(i\ k) \in \mathrm{Stab}^\bullet(V)$ and hence $(i\ k) \in H$ as $\mathrm{Stab}^\bullet(V) \leqslant H$. □

Corollary 5.4. *If* $H \subseteq S_n$ *has a support of size* $< n/2$*, then* H *has a unique minimal support of size* $< n/2$*.* □

We now give our proof of Proposition 5.2 (which bypasses the lengthy combinatorial argument in [3]).

Proof of Proposition 5.2. Suppose $H_1, \ldots, H_r$ are k-supported subgroups of S_n. Let H be another subgroup of S_n such that $H_1 \cap \cdots \cap H_r \subseteq H$ and $[S_n : H] \leqslant n^k$. By Lemma 4.2, it suffices to show that H is also k-supported. For each $i \in [r]$, fix $U_i \subset [n]$ such that $|U_i| \leqslant k$ and $\mathrm{Stab}^\bullet(U_i) \subseteq H_i$. Let $U = U_1 \cup \cdots \cup U_r$. Note that U is a support for H, as $\mathrm{Stab}^\bullet(U) = \mathrm{Stab}^\bullet(U_1) \cap \cdots \cap \mathrm{Stab}^\bullet(U_r) \subseteq H_1 \cap \cdots \cap H_r \subseteq H$. Also note that $|U| \leqslant rk < n/2$. So by Corollary 5.4, H has a unique minimal support V of size $< n/2$.

We claim that $H \subseteq \mathrm{Stab}(V)$. For contradiction, assume otherwise. Then there exists $h \in H$ such that $hV \neq V$. Note that hV is a support for $hHh^{-1} = H$. Since $V \cup hV \neq [n]$ (as $|V \cup hV| \leqslant 2|V| < n$), the intersection $V \cap hV$ is a support for H by Lemma 5.3. But $V \cap hV \subset V$, which contradicts the minimality of V. Therefore, $H \subseteq \mathrm{Stab}(V)$ as claimed. It follows that $[S_n : H] \geqslant [S_n : \mathrm{Stab}(V)] = \binom{n}{|V|}$. Since $[S_n : H] \leqslant n^k$, we conclude that $|V| \leqslant k$. Therefore, H is k-supported. □

6 CPT+C Cannot Construct the Dual of a Finite Vector Space

Let V be a finite vector space over a fixed finite field F. We view V as a structure with binary operation $+$ and unary operations for scalar multiplication by each element of F. Let $\mathcal{H}(V)$ denote the set of hyperplanes in V. Note that $\mathcal{H}(V)$ is an element of $\mathrm{HF}(V)$ (in particular, $\mathcal{H}(V)$ is a set of subsets of V).

The task of computing $\mathcal{H}(V)$ given V is a polynomial-time *function problem* (as opposed to *decision problem*) in the usual sense of complexity theory ($\mathcal{H}(V)$ has a polynomial-size description as a hereditary finite object, i.e., $|\mathrm{TC}(\mathcal{H}(V))| = O(\mathrm{poly}(|V|))$. Moreover, $\mathcal{H}(V)$ is an invariant of V (i.e., not depending on any extrinsic linear order on V). It is thus reasonable to ask whether any CPT+C program computes the operation $V \longmapsto \mathcal{H}(V)$.

We remark the results of this section hold just the same for the operation $V \longmapsto V^*$ of computing from V the *dual space* V^* of linear functions $V \longrightarrow F$ (suitably represented as an element of $\mathrm{HF}(V \cup F)$).

Theorem 6.1. *No program in* CPT+C *program computes the operation* $V \longmapsto \mathcal{H}(V)$ *over finite vector spaces over a fixed finite field* F*.*

Noting that a hyperplane in an n-dimensional vector space has smallest support size $n-1$, Theorem 6.1 follows from the following vector-space analogue of Proposition 5.2.

Proposition 6.2. *If* V *is a finite vector space of dimension* $> r^2k^2$*, then the group* $\mathrm{GL}(V)$ *of linear automorphisms of* V *has the* (k, r)*-support property.*

To prove Theorem 6.1 from Proposition 6.2, note that if $\Pi \in$ CPT+C, then $\Pi \in \mathrm{BGS}(n^k)$ for some k. Let r be the variable rank Π. Consider a finite vector space V on dimension $> r^2k^2$. By Proposition 4.3, $\Pi(V)$ (the output of Π on V) is (k, r)-constructible. By Proposition 6.2, $\mathrm{GL}(V)$ $(= \mathrm{Aut}(V))$ has the (k, r)-support property. Therefore, $\Pi(V)$ is k-supported. Since $\mathcal{H}(V)$ is not k-supported for any $k < \dim(V)$, we conclude that $\Pi(V) \neq \mathcal{H}(V)$.

The proof of Proposition 6.2 proceeds along similar lines as the proof of Proposition 5.2. We have the following vector-space analogues of Lemma 5.3 and Corollary 5.4.

Lemma 6.3. *If subspaces* $W, W' \subseteq V$ *both support a subgroup* $H \subseteq \mathrm{GL}(V)$ *and if* $W + W' \neq V$*, then the intersection* $W \cap W'$ *supports* H*.*

Proof. Let P, P' and Q denote the pointwise stabilizers of W, W' and $W \cap W'$, respectively. It suffices to show that Q is the subgroup of $\mathrm{GL}(V)$ generated by $P \cup P'$. This is a simple exercise in linear algebra. Choose a basis $v_1, ..., v_n$ for V such that for some $1 \leqslant i \leqslant j \leqslant j' < n = \dim(V)$,

- $v_1, ..., v_j$ span W,
- $v_i, ..., v_{j'}$ span W',
- $v_i, ..., v_j$ span $W \cap W'$.

We now identify particular sets of generators for P, P' and Q. For all $r, s \in \{1, \ldots, n\}$ and

$\lambda \in F^\times$, define $n \times n$ matrix $\sigma_{r,s,\lambda}$ by

$$\sigma_{r,s,\lambda} = \begin{cases} \begin{pmatrix} I_{r-1} & & & & \\ & 1 & & \lambda & \\ & & I_{s-r-1} & & \\ & 0 & & 1 & \\ & & & & I_{n-s} \end{pmatrix} & \text{if } r < s, \\ \begin{pmatrix} I_{r-1} & & \\ & \lambda & \\ & & I_{n-r} \end{pmatrix} & \text{if } r = s, \\ \begin{pmatrix} I_{s-1} & & & & \\ & 1 & & 0 & \\ & & I_{r-s-1} & & \\ & \lambda & & 1 & \\ & & & & I_{n-r} \end{pmatrix} & \text{if } r > s. \end{cases}$$

For all $r, s \in \{1, \ldots, n\}$ with $r < s$, define $n \times n$ matrix $\tau_{r,s}$ by

$$\tau_{r,s} = \begin{pmatrix} I_{r-1} & & & & \\ & 0 & & 1 & \\ & & I_{s-r-1} & & \\ & 1 & & 0 & \\ & & & & I_{n-s} \end{pmatrix}.$$

Linear transformations $\sigma_{r,s,\lambda}$ and $\tau_{r,s}$ are the familiar "row reduction" generators of $\mathrm{GL}(V)$ with respect to the basis $v_1, ..., v_n$. Note that P (respectively: P', Q) is generated by the set of all $\sigma_{r,s,\lambda}$ such that $r \notin \{1, ..., j\}$ (respectively: $r \notin \{i, ..., j'\}$, $r \notin \{i, ..., j\}$), together with all $\tau_{r,s}$ such that $r, s \notin \{1, \ldots, j\}$ (respectively: $r, s \notin \{i, ..., j'\}$, $r, s \notin \{i, ..., j\}$) .

The only generators of Q which are not also generators of P or P' are those of the form $\tau_{r,s}$ where $r \in \{1, ..., i-1\}$ and $s \in \{j+1, ..., j'\}$. Note that $\tau_{r,s} = \tau_{r,n}\tau_{s,n}\tau_{r,n}$ and $\tau_{r,n} \in P'$ and $\tau_{s,n} \in P$. Therefore, $\tau_{r,s}$ is in the subgroup of $\mathrm{GL}(V)$ generated by $P \cup P'$. We conclude that Q is the subgroup of $\mathrm{GL}(V)$ generated by $P \cup P'$. □

Corollary 6.4. *If a subgroup $H \subseteq \mathrm{GL}(V)$ is supported by a subspace of dimension $< n/2$, then H is supported by a unique minimal subspace of dimension $< n/2$.* □

With this corollary, we are ready to prove Proposition 6.2.

Proof of Proposition 6.2. Let $H_1, \ldots, H_r$ be k-supported subgroups of $\mathrm{GL}(V)$. Let H be another subgroup of $\mathrm{GL}(V)$ such that $H \supseteq H_1 \cap \cdots \cap H_r$ and $[\mathrm{GL}(V) : H] \leqslant |V|^k = q^{nk}$ where q is the size of the field F. By Lemma 4.2, it suffices show that H is also k-supported. For each $i \in [r]$, fix $U_i \subset [n]$ such that $|U_i| \leqslant k$ and $\mathrm{Stab}^\bullet(U_i) \subseteq H_i$. Let $U = U_1 \cup \cdots \cup U_r$. Note that U is a support for H, as $\mathrm{Stab}^\bullet(U) = \mathrm{Stab}^\bullet(U_1) \cap \cdots \cap \mathrm{Stab}^\bullet(U_r) \subseteq H_1 \cap \cdots \cap H_r \subseteq H$. Also note that $|U| \leqslant rk < n/2$. So by Corollary 5.4, H is supported by a unique minimal subspace W of dimension $\leqslant rk$.

We claim that $H \subseteq \mathrm{Stab}(W)$. For contradiction, assume otherwise. Then there exists $h \in H$ such that $hW \neq W$. Note that hW is a support for $hHh^{-1} = H$. Since $W + hW \neq V$ (as $\dim(W + hW) \leqslant 2\dim(W) < n$), the intersection $W \cap hW$ is a support for H by Lemma 5.3. But $W \cap hW \subset W$, which contradicts the minimality of W. Therefore, $H \subseteq \mathrm{Stab}(W)$ as claimed. It follows that $[\mathrm{GL}(V) : H] \geqslant [\mathrm{GL}(V) : \mathrm{Stab}(W)] = \#\{\dim(W)\text{-dimensional subspaces of } V\}$. For all $d \leqslant rk$ $(= \sqrt{n})$, we have

$$\begin{aligned}&\#\{d\text{-dimensional subspaces of } V\}\\ &= \prod_{i=0}^{d-1} \frac{q^{n-i}-1}{q^{i+1}-1} \geqslant \prod_{i=0}^{d-1} q^{n-2i-2} = q^{dn-2\binom{d-1}{2}-2} \geqslant q^{dn-(\sqrt{n}-1)(\sqrt{n}-2)-2} = q^{(d-1)n+3\sqrt{n}-4} > |V|^{d-1}.\end{aligned}$$

Since $[\mathrm{GL}(V) : H] \leqslant |V|^k$, it follows that $\dim(W) \leqslant k$. Because every basis for W is a support for H, it follows that H is k-supported. □

Acknowledgements. My thanks to Swastik Kopparty and an anonymous referee for their helpful comments.

References

[1] A. Blass and Y. Gurevich. Strong extension axioms and Shelah's zero-one law for choiceless polynomial time. *Journal of Symbolic Logic*, 68(1):65–131, 2003.

[2] A. Blass and Y. Gurevich. A quick update on the open problems in Blass-Gurevich-Shelah's article "On polynomial time computations over unordered structures". Available at http://research.microsoft.com/en-us/um/people/gurevich/Opera/150a.pdf, December 2005.

[3] A. Blass, Y. Gurevich, and S. Shelah. Choiceless polynomial time. *Annals of Pure and Applied Logic*, 100(1–3):141–187, 1999.

[4] A. Blass, Y. Gurevich, and S. Shelah. On polynomial time computation over unordered structures. *Journal of Symbolic Logic*, 67(3):1093–1125, 2002.

[5] J.-Y. Cai, M. Fürer, and N. Immerman. An optimal lower bound on the number of variables for graph identification. *Combinatorica*, 12(4):389–410, 1992.

[6] A. Chandra and D. Harel. Structure and complexity of relational queries. *Journal of Computer and System Sciences*, 25:99–128, 1982.

[7] A. Dawar, D. Richerby, and B. Rossman. Choiceless polynomial time, counting and the Cai-Fürer-Immerman graphs. *Annals of Pure and Applied Logic*, 152:31–50, 2008.

[8] R. Fagin. Generalized first-order spectra and polynomial-time recognizable sets. In R. M. Karp, editor, *Complexity of Computation*, volume 7 of *SIAM-AMS Proceedings*, pages 43–73, 1974.

[9] Y. Gurevich. Toward logic tailored for computational complexity. In M. M. Richter et al., editor, *Computation and Proof Theory*, pages 175–216. Springer Lecture Notes in Mathematics, 1984.

[10] N. Immerman. Relational queries computable in polynomial time. *Information and Control*, 68(1–3):86–104, 1986.

[11] S. Shelah. Choiceless polynominal time logic: Inability to express. In *CSL*, pages 72–125, 2000.

[12] M. Y. Vardi. The complexity of relational query languages. In *Proc. 14th ACM Symp. on Theory of Computing*, pages 137–146, 1982.