On Symmetric Circuits and Fixed-Point Logics

Matthew Anderson and Anuj Dawar

University of Cambridge Computer Laboratory
15 JJ Thomson Ave, Cambridge, CB3 0FD, UK
`firstname.lastname@cl.cam.ac.uk`

Abstract

We study properties of relational structures such as graphs that are decided by families of Boolean circuits. Circuits that decide such properties are necessarily invariant to permutations of the elements of the input structures. We focus on families of circuits that are symmetric, i.e., circuits whose invariance is witnessed by automorphisms of the circuit induced by the permutation of the input structure. We show that the expressive power of such families is closely tied to definability in logic. In particular, we show that the queries defined on structures by uniform families of symmetric Boolean circuits with majority gates are exactly those definable in fixed-point logic with counting. This shows that inexpressibility results in the latter logic lead to lower bounds against polynomial-size families of symmetric circuits.

1998 ACM Subject Classification F.1.1 Models of Computation, F.1.3 Complexity Measures and Classes, F.4.1 Mathematical Logic

Keywords and phrases symmetric circuit, fixed-point logic, majority, counting, uniformity

1 Introduction

A property of graphs on n vertices can be seen as a Boolean function which takes as inputs the $\binom{n}{2}$ potential edges (each of which can be 0 or 1) and outputs either 0 or 1. For the function to determine a property of the graph, rather than of a particular presentation of the graph, it must be invariant under re-ordering the vertices of the graph. That is, permuting the $\binom{n}{2}$ inputs according to some permutation of $[n]$ leaves the value of the function unchanged. We call such Boolean functions *invariant*. Note that this does not require the function to be invariant under *all* permutations of its inputs, which would mean that it was entirely determined by the number of inputs that are set to 1.

It is a long-standing open problem in descriptive complexity to give a characterisation of the polynomial-time properties of finite relational structures (or, indeed, just graphs) as the classes of structures definable in some suitable logic (see, for instance, [7, Chapter 11]). It is known that fixed-point logic FP and its extension with counting FPC are strictly less expressive than deterministic polynomial time P [3]. It is easy to see that every polynomial-time property of graphs is decided by a P-uniform family of circuits that are *invariant* in the sense above. On the other hand, when a property of graphs is expressed in a formal logic, it gives rise to a family of circuits that are *explicitly invariant* or *symmetric*. By this we mean that their invariance is witnessed by the automorphisms of the circuits themselves. For instance, any sentence of FP translates into a polynomial-size family of symmetric Boolean circuits, while any sentence of FPC translates into a polynomial-size family of symmetric Boolean circuits with majority gates.

Concretely, a circuit C_n consists of a directed acyclic graph whose internal gates are marked by operations from a basis (e.g., the standard Boolean basis $\mathbb{B}_{\text{std}} := \{\text{AND, OR, NOT}\}$ or the majority basis $\mathbb{B}_{\text{maj}} = \mathbb{B}_{\text{std}} \cup \{\text{MAJ}\}$) and input gates which are marked with pairs of vertices representing potential edges of an n-vertex input graph. Such a circuit is *symmetric*

if C_n has an automorphism π induced by each permutation σ of the n vertices, i.e., π moves the input gates of C_n according to σ and preserves operations and wiring of the internal gates of C_n. Clearly, any symmetric circuit is invariant.

Are symmetric circuits a weaker model of computation than invariant circuits? We aim at characterising the properties that can be decided by uniform families of symmetric circuits. Our main result shows that, indeed, any property that is decided by a uniform polynomial-size family of symmetric majority circuits can be expressed in FPC.

▶ **Theorem 1.** *A graph property is decided by a* P*-uniform family of symmetric majority circuits if, and only if, it is defined by a fixed-point with counting sentence.*

A consequence of this result is that inexpressibility results that have been proved for FPC can be translated into lower bound results for symmetric circuits. For instance, it follows (using [4]) that there is no polynomial-size family of symmetric majority circuits deciding 3-colourability or Hamiltonicity of graphs.

We also achieve a characterisation similar to Theorem 1 of symmetric Boolean circuits.

▶ **Theorem 2.** *A graph property is decided by a* P*-uniform family of symmetric Boolean circuits if, and only if, it is defined by a fixed-point sentence interpreted in $\mathcal{G} \oplus \langle [n], \leq \rangle$, i.e., the structure that is the disjoint union of an n-vertex graph $\mathcal{G}$ with a linear order of length n.*

Note that symmetric majority circuits can be transformed into symmetric Boolean circuits. But, since FP, even interpreted over $\mathcal{G} \oplus \langle [n], \leq \rangle$, is strictly less expressive than FPC, our results imply that any such translation must involve a super-polynomial blow-up in size. Similarly, our results imply with [3] that *invariant* Boolean circuits cannot be transformed into symmetric circuits (even with majority gates) without a super-polynomial blow-up in size. On the other hand, it is clear that symmetric majority circuits can still be translated into *invariant* Boolean circuits with only a polynomial blow-up.

Support. The main technical tool in establishing the translation from uniform families of symmetric circuits to sentences in fixed-point logics is a *support theorem* (stated informally below) that establishes properties of the stabiliser groups of gates in symmetric circuits.

We say that a set $X \subseteq [n]$ *supports* a gate g in a symmetric circuit C on an n-element input structure if every automorphism of C that is generated by a permutation of $[n]$ fixing X also fixes g. It is not difficult to see that for a family of symmetric circuits obtained from a given first-order formula ϕ there is a constant k such that all gates in all circuits of the family have a support of size at most k. To be precise, the gates in such a circuit correspond to subformulas ψ of ϕ along with an assignment of values from $[n]$ to the free variables of ψ. The set of elements of $[n]$ appearing in such an assignment forms a support of the gate and its size is bounded by the number of free variables ψ. Using the fact that any formula of FP is equivalent, on structures of size n, to a first-order formula with a constant bound k on the number of variables and similarly any formula of FPC is equivalent to a first-order formula *with majority quantifiers* (see [9]) and a constant bound on the number of variables, we see that the resulting circuits have supports of constant-bounded size. Our main technical result is that the existence of supports of bounded size holds, in fact, for all polynomial-size families of symmetric circuits. In its general form, we show the following theorem in Section 3 via an involved combinatorial argument.

▶ **Theorem 3** (Informal Support Thm)**.** *Let C be a symmetric circuit with s gates over a graph of size n. If n is sufficiently large and s is sub-exponential in n, then every gate in C has a support of size $O\Big(\frac{\log s}{\log n}\Big)$.*

In the typical instantiation of the Theorem 3 the circuit C contains a polynomial number of gates $s = \text{poly}(n)$ and hence the theorem implies that every gate has a support that is bounded in size by a constant. The proof of the Theorem 3 mainly relies on the structural properties of symmetric circuits and is largely independent of the semantics of such circuits; this means it may be of independent interest for other circuit bases and in other settings.

Symmetric Circuits and FP**.** In Section 4 we show that each polynomial-size family $\mathcal{C}$ of symmetric circuits can be translated into a formula of fixed-point logic. If the family $\mathcal{C}$ is P-uniform, by the Immerman-Vardi Theorem [12, 8] there is an FP-definable interpretation of the circuit C_n in the ordered structure $\langle [n], \leq \rangle$. We show that the support of a gate is computable in polynomial time, and hence we can also interpret the support of each gate in $\langle [n], \leq \rangle$. The circuit C_n can be evaluated on an input graph $\mathcal{G}$ by fixing a bijection between $[n]$ and the universe U of $\mathcal{G}$. We associate with each gate of g of C_n the set of those bijections that cause g to evaluate to 1 on $\mathcal{G}$. This set of bijections admits a compact (i.e., polynomial-size) representation as the set of injective maps from the support of g to U. We show that these compact representations can be inductively defined by formulas of FP, or FPC if the circuit also admits majority gates.

Thus, we obtain that P-uniform families of symmetric Boolean circuits can be translated into formulas of FP interpreted in $\mathcal{G}$ combined with a disjoint linear order $\langle [|\mathcal{G}|], \leq \rangle$, while families containing majority gates can be simulated by sentences of FPC. The reverse containment follows using classical techniques. As a consequence we obtain the equivalences of Theorems 1 & 2, and a number of more general results as this sequence of arguments naturally extends to: (i) inputs given as an arbitrary relational structure, (ii) outputs defining arbitrary relational queries, and (iii) non-uniform circuits, provided the logic is allowed additional advice on the disjoint linear order.

Related Work. The term "symmetric circuit" is used by Denenberg et al. in [6] to mean what we call invariant circuits. They give a characterisation of first-order definability in terms of a restricted invariance condition, namely circuits that are invariant and whose relativisation to subsets of the universe remains invariant. Our definition of symmetric circuits follows that in [10] where Otto describes it as the "natural and straightforward combinatorial condition to guarantee generic or isomorphism-invariant performance." He combines it with a size restriction on the orbits of gates along with a strong uniformity condition, which he calls "coherence", to give an exact characterisation of definability in infinitary logic. A key element is the proof that if the orbits of gates in such a circuit are polynomially bounded in size then they have supports of bounded size. We remove the assumption of coherence from this argument and show that constant-size supports exist in any polynomial-size symmetric circuit. This requires a generalisation of what Otto calls a "base" to supporting partitions. See Section 5 for more discussion of connections with prior work.

Due to space limitations, full proofs are omitted and may be found in [1].

2 Preliminaries

Let $[n]$ denote the set of positive integers $\{1, \ldots, n\}$. Let Sym_S denote the group of all permutations of the set S. When $S = [n]$, we write Sym_n for $\text{Sym}_{[n]}$.

2.1 Vocabularies, Structures, and Logics

A *relational vocabulary* (always denoted by τ) is a finite sequence of relation symbols $(R_1^{r_1}, \ldots, R_k^{r_k})$ where for each $i \in [k]$ the relation symbol R_i has an associated arity $r_i \in \mathbb{N}$.

A τ-*structure* $\mathcal{A}$ is a tuple $\langle A, R_1^{\mathcal{A}}, \ldots, R_k^{\mathcal{A}} \rangle$ consisting of (i) a non-empty set A called the *universe* of $\mathcal{A}$, and (ii) relations $R_i^{\mathcal{A}} \subseteq A^{r_i}$ for $i \in [k]$. Members of the universe A are called *elements* of $\mathcal{A}$. A *multi-sorted* structure is one whose universe is given as a disjoint union of several distinct *sorts*. Define the *size* of a structure $|\mathcal{A}|$ to be the cardinality of its universe. All structures considered in this paper are *finite*, i.e., their universes have finite cardinality. Let $\mathrm{fin}[\tau]$ denote the set of all finite τ-structures.

First-Order and Fixed-Point Logics. Let $\mathrm{FO}(\tau)$ denote *first-order logic* with respect to the vocabulary τ. The logic $\mathrm{FO}(\tau)$ is the set of formulas whose atoms are formed using the relation symbols in τ, an equality symbol $=$, an infinite sequence of variables $(x, y, z \ldots)$, and that are closed under the Boolean connectives ($\wedge$ and $\vee$), negation ($\neg$), and universal and existential quantification ($\forall$ and $\exists$). Let *fixed-point logic* $\mathrm{FP}(\tau)$ denote the extension of $\mathrm{FO}(\tau)$ to include an inflationary fixed-point operator ifp. Assume standard syntax and semantics for FO and FP (see the textbook [7] for more background). For a formula ϕ write $\phi(x)$ to indicate that x is the tuple of the free variables of ϕ. For a logic $\mathcal{L}$, a formula $\phi(x) \in \mathcal{L}(\tau)$ with k free variables, $\mathcal{A} \in \mathrm{fin}[\tau]$, and tuple $a \in A^k$ write $\mathcal{A} \models_{\mathcal{L}} \phi[a]$ to express that the tuple a makes the formula ϕ true in the structure $\mathcal{A}$ with respect to the logic $\mathcal{L}$. We usually drop the subscript $\mathcal{L}$ and write $\mathcal{A} \models \phi[a]$ when no confusion would arise.

Logics with Disjoint Advice. Let τ_{arb} be a relational vocabulary without a binary relation symbol $\leq$. Let $\Upsilon : \mathbb{N} \to \mathrm{fin}[\tau_{\mathrm{arb}} \uplus \{\leq^2\}]$ be an *advice function*, where for $n \in \mathbb{N}$, $\Upsilon(n)$ has universe $[n]$ naturally ordered by $\leq$. Let $(\mathrm{FP} + \Upsilon)(\tau)$ denote the set of formulas of $\mathrm{FP}(\tau')$ where $\tau' := \tau \uplus \tau_{\mathrm{arb}} \uplus \{\leq^2\}$ and τ is a vocabulary disjoint from $\tau_{\mathrm{arb}} \uplus \{\leq^2\}$. For a structure $\mathcal{A} \in \mathrm{fin}[\tau]$ define the semantics of $\phi \in (\mathrm{FP} + \Upsilon)(\tau)$ to be $\mathcal{A} \models_{(\mathrm{FP}+\Upsilon)} \phi$ iff $\mathcal{A}^{\Upsilon} \models_{\mathrm{FP}} \phi$, where $\mathcal{A}^{\Upsilon} := \mathcal{A} \oplus \Upsilon(|\mathcal{A}|)$ is the multi-sorted τ'-structure formed by taking the disjoint union of $\mathcal{A}$ with a structure coding a linear order of corresponding cardinality endowed with interpretations of the relations in τ_{arb}. The universe of the multi-sorted structure $\mathcal{A}^{\Upsilon}$ is written as $A \uplus [|A|]$; refer to A as the *point sort* of $\mathcal{A}^{\Upsilon}$ and to $[|A|]$ as the *number sort* of $\mathcal{A}^{\Upsilon}$. We are primarily interested in the special case when τ_{arb} is empty and hence $\Upsilon(|\mathcal{A}|) = \langle [|A|], \leq \rangle$ is simply a linear order. Denote formulas of this logic by $(\mathrm{FP} + \leq)(\tau)$ and extended structures by $\mathcal{A}^{\leq}$ to emphasise the disjoint linear order. Let $\mathrm{FPC}(\tau)$ denote the extension of $(\mathrm{FP} + \leq)(\tau)$ with a counting operator $\#_x$ where x is a point or number variable. For a structure $\mathcal{A} \in \mathrm{fin}[\tau]$ and a formula $\phi(x) \in \mathrm{FPC}(\tau)$, $\#_x \phi(x)$ is a term denoting the element in the number sort corresponding to $|\{a \in \mathcal{A} \mid \mathcal{A} \models \phi[a]\}|$. See [7, Section 8.4.2] for more details. Finally, we consider the extension of fixed-point logic with both advice functions and counting quantifiers $(\mathrm{FPC} + \Upsilon)(\tau)$.

2.2 Symmetric and Uniform Circuits

A *Boolean basis* (always denoted by $\mathbb{B}$) is a finite set of Boolean functions from $\{0,1\}^*$ to $\{0,1\}$. We consider only bases containing symmetric functions, i.e., for all $f \in \mathbb{B}$, $f(x) = f(y)$ for all $n \in \mathbb{N}$ and $x, y \in \{0,1\}^n$ with the same number of ones. The *standard Boolean basis* $\mathbb{B}_{\mathrm{std}}$ consists of unbounded fan-in AND, OR, and unary NOT operators. The *majority basis* $\mathbb{B}_{\mathrm{maj}}$ extends the standard basis with an operator MAJ which is one iff the number of ones in the input is at least the number of zeroes.

▶ **Definition 4** (Circuits on Structures). A *Boolean* $(\mathbb{B}, \tau)$*-circuit* C *with universe* U *computing a* q*-ary query* Q is a structure $\langle G, W, \Omega, \Sigma, \Lambda \rangle$.

- G is a set called the *gates* of C. The *size* of C is $|C| := |G|$.

- $W \subseteq G \times G$ is a binary relation called the *wires* of the circuit. We require that (G, W) forms a *directed acyclic graph*. Call the gates with no incoming wires *input gates*, and all other gates *internal gates*. Gates h with $(h, g) \in W$ are called the *children* of g.
- Ω is an injective function from U^q to G. The gates in the image of Ω are called the *output gates*. When $q = 0$, Ω is a constant function mapping to a single output gate.
- Σ is a function from G to $\mathbb{B} \uplus \tau \uplus \{0, 1\}$ which maps input gates into $\tau \uplus \{0, 1\}$ with $|\Sigma^{-1}(0)|, |\Sigma^{-1}(1)| \leq 1$ and internal gates into $\mathbb{B}$. Call the input gates marked with a relation from τ *relational gates* and the input gates marked with 0 or 1 *constant gates*.
- Λ is a sequence of injective functions $(\Lambda_R)_{R \in \tau}$ where for each $R \in \tau$, Λ_R maps each relational gate g with $R = \Sigma(g)$ to $\Lambda_R(g) \in U^r$ where r is the arity of R. Where no ambiguity arises, we write $\Lambda(g)$ for $\Lambda_R(g)$.

Let C be a Boolean $(\mathbb{B}, \tau)$-circuit with universe U, $\mathcal{A} \in \text{fin}[\tau]$ with $|\mathcal{A}| = |U|$, and $\gamma : A \to U$ be a bijection. Let $\gamma\mathcal{A}$ denote the τ-structure over the universe U obtained by relabelling the universe of $\mathcal{A}$ according to γ. Recursively evaluate C on $\gamma\mathcal{A}$ by determining a value $C[\gamma\mathcal{A}](g)$ for each gate g: (i) a constant gate evaluates to the bit given by $\Sigma(g)$, (ii) a relational gate evaluates to 1 iff $\gamma\mathcal{A} \models \Sigma(g)(\Lambda_{\Sigma(g)}(g))$, and (iii) an internal gate evaluates to the result of applying the Boolean operation $\Sigma(g)$ to the values for g's children. C defines the q-ary query $Q \subseteq \mathcal{A}^q$ where $a \in Q$ iff $C[\gamma\mathcal{A}](\Omega(\gamma a)) = 1$.

▶ **Definition 5** (Invariant Circuit)**.** Let C be a $(\mathbb{B}, \tau)$-circuit with universe U computing a q-ary query. The circuit C is *invariant* if for every $\mathcal{A} \in \text{fin}[\tau]$ with $|\mathcal{A}| = |U|$, $a \in \mathcal{A}^q$, and bijections γ_1, γ_2 from A to U, $C[\gamma_1\mathcal{A}](\Omega(\gamma_1 a)) = C[\gamma_2\mathcal{A}](\Omega(\gamma_2 a))$.

Invariance indicates that C computes a property of τ-structures which is invariant to presentations of the structure. Moreover, for an invariant circuit C only the size of U matters and we often write $C = C_n$, for emphasis, when the universe is size n. A *family* $\mathcal{C} = (C_n)_{n \in \mathbb{N}}$ of invariant $(\mathbb{B}, \tau)$-circuits naturally computes a q-ary query on τ-structures. When $q = 0$ the family computes a Boolean property of structures. We now discuss an algebraic property of circuits called *symmetry* that implies invariance.

Symmetric Circuits. Permuting a circuit's universe may induce automorphisms of the circuit.

▶ **Definition 6** (Induced Automorphism)**.** Let $C = \langle G, W, \Omega, \Sigma, \Lambda \rangle$ be a $(\mathbb{B}, \tau)$-circuit with universe U computing a q-ary query. Let $\sigma \in \text{Sym}_U$. If there is a bijection π from G to G such that

- for all gates $g, h \in G$, $W(g, h)$ iff $W(\pi(g), \pi(h))$,
- for all output tuples $x \in U^q$, $\pi\Omega(x) = \Omega(\sigma(x))$,
- for all gates $g \in G$, $\Sigma(g) = \Sigma(\pi(g))$, and
- for each relational gate $g \in G$, $\sigma\Lambda(g) = \Lambda(\pi(g))$,

we say σ *induces the automorphism* π of C.

The principle goal of this paper is to understand the computational power of circuit classes with the following type of algebraic symmetry.

▶ **Definition 7** (Symmetric)**.** A circuit C with universe U is called *symmetric* if for every permutation $\sigma \in \text{Sym}_U$, σ induces an automorphism of C.

It is not difficult to see that, for a symmetric circuit C, there is a homomorphism $h : \text{Sym}_U \to \text{Aut}(C)$ (where $\text{Aut}(C)$ denotes the automorphism group of C) such that $h(\sigma)$ is an automorphism induced by σ.

To avoid certain trivialities we restrict ourselves to circuits which are *rigid*.

▶ **Definition 8** (Rigid). Let $C = \langle G, W, \Omega, \Sigma, \Lambda \rangle$ be a $(\mathbb{B}, \tau)$-circuit with universe U. Call C *rigid* if there do not exist distinct gates $g, g' \in G$ with $\Sigma(g) = \Sigma(g')$, $\Lambda(g) = \Lambda(g')$, $\Omega^{-1}(g) = \Omega^{-1}(g')$, and for every $g'' \in G$, $W(g'', g)$ iff $W(g'', g')$.

For a rigid symmetric circuit C it is easy to show that the group of automorphisms of C is exactly Sym_U acting faithfully. We shall therefore abuse notation and use these interchangeably. In particular, we shall write σg to denote the image of a gate g in C under the action of the automorphism induced by a permutation σ in Sym_U.

An examination of the definitions suffices to show that symmetry implies invariance. In symmetric circuits it is useful to consider those permutations which induce automorphisms that fix gates. Let $\mathcal{P}$ be a partition of a set U. Let the *pointwise stabiliser* of $\mathcal{P}$ be $\mathrm{Stab}_U(\mathcal{P}) := \{\sigma \in \mathrm{Sym}_U \mid \forall P \in \mathcal{P}, \sigma P = P\}$, and similarly define the *setwise stabiliser* $\mathrm{Stab}_U\{\mathcal{P}\} := \{\sigma \in \mathrm{Sym}_U \mid \forall P \in \mathcal{P}, \sigma P \in \mathcal{P}\}$. For a gate g in a rigid symmetric circuit C with universe U, let the *stabiliser* of g be $\mathrm{Stab}_U(g) := \{\sigma \in \mathrm{Sym}_U \mid \sigma g = g\}$, and let the *orbit* of g under the automorphism group $\mathrm{Aut}(C)$ of C be $\mathrm{Orb}(g) := \{\sigma g \mid \sigma \in \mathrm{Sym}_U\}$. In each case, when $U = [n]$, we write Stab_n instead of $\mathrm{Stab}_{[n]}$.

Uniform Circuits. One natural class of circuits are those with polynomial-size descriptions that can be generated by a deterministic polynomial-time machine.

▶ **Definition 9** (P and P/poly-Uniform). A $(\mathbb{B}, \tau)$-circuit family $\mathcal{C} = (C_n)_{n \in \mathbb{N}}$ computing a q-ary query is P/poly-*uniform* if there exists an integer $t \geq q$ and function $\Upsilon : \mathbb{N} \to \{0,1\}^*$ which takes an integer n to a binary string $\Upsilon(n)$ such that $|\Upsilon(n)| = \mathrm{poly}(n)$, and $\Upsilon(n)$ describes[1] the circuit C_n whose gates are indexed by t-tuples of $[n]$, inputs are labelled by t-tuples of $[n]$, and outputs are labelled by q-tuples of $[n]$. Moreover, if there exists a deterministic Turing machine M that for each integer n computes $\Upsilon(n)$ from 1^n in time $\mathrm{poly}(n)$ call $\mathcal{C}$ P-*uniform*.

Note that such uniform families implicitly have polynomial size.

Over ordered structures neither P-uniform nor P/poly-uniform circuits need compute invariant queries as their computation may implicitly depend on the order associated with $[n]$. To obtain invariance for such circuits we assert symmetry. The next section proves a natural property of symmetric circuits that ultimately implies that *symmetric* P-uniform circuits coincide with FP definitions on the standard and majority bases.

3 Symmetry and Support

In this section we analyse the algebraic properties of symmetric circuits.

▶ **Definition 10** (Support). Let C be a rigid symmetric circuit with universe U and let g be a gate in C. A set $X \subseteq U$ *supports* g if $\mathrm{Stab}_U(X) \subseteq \mathrm{Stab}_U(g)$.

We now show how to associate supports of constant size in a canonical way to all gates in *any* rigid symmetric circuit of polynomial size. Indeed, our result is more general as it associates moderately growing supports to gates in circuits of sub-exponential size. We first introducing the more general notion of a *supporting partition* for a permutation group, which can be canonically associated with any permutation group G, and obtain bounds on the size of such a partition based on the index of G in the symmetric group. These results are then

[1] Formally one must define a particular way of encoding circuits via binary strings. However, since the details of the representation are largely irrelevant for our purposes we omit them.

used to bound the size of supports of stabiliser groups of gates in rigid symmetric circuits as a function of circuit size. This proves our main technical result—the Support Theorem.

A supporting partition generalises the notion of a support of a gate by replacing the set with a partition and the stabiliser group of the gate with an arbitrary permutation group.

▶ **Definition 11** (Supporting Partition)**.** Let $G \subseteq \mathrm{Sym}_U$ be a group and $\mathcal{P}$ a partition of U. We say that $\mathcal{P}$ is a *supporting partition* of G if $\mathrm{Stab}_U(\mathcal{P}) \subseteq G$.

For intuition consider two extremes. When G has supporting partition $\mathcal{P} = \{U\}$, it indicates $G = \mathrm{Sym}_U$. Saying that G has supporting partition $\mathcal{P} = \{\{u_1\}, \{u_2\}, \ldots, \{u_{|U|}\}\}$ indicates only that G contains the identity permutation, which is always true.

A natural partial order on partitions is the coarseness relation, i.e., $\mathcal{P}'$ is as coarse as $\mathcal{P}$, denoted $\mathcal{P}' \supseteq \mathcal{P}$, if every part in $\mathcal{P}$ is contained in some part of $\mathcal{P}'$. A proof is similar to that of (*) on page 379 of [10] implies the following lemma.

▶ **Lemma 12.** *Each permutation group $G \subseteq \mathrm{Sym}_U$ has a unique coarsest supporting partition.*

We write $\mathsf{SP}(G)$ for *the unique coarsest partition supporting G*. By analysing how supporting partitions are affected by the conjugacy action of Sym_U it is easy to show that any group G is sandwiched between the pointwise and setwise stabilisers of $\mathsf{SP}(G)$.

▶ **Lemma 13.** *For any group $G \subseteq \mathrm{Sym}_U$, we have $\mathrm{Stab}_U(\mathsf{SP}(G)) \subseteq G \subseteq \mathrm{Stab}_U\{\mathsf{SP}(G)\}$.*

Note that these bounds need not be tight. For example, if G is the alternating group on U (or, indeed, any transitive, primitive subgroup of Sym_U), then $\mathsf{SP}(G)$ is the partition of U into singletons. In this case, $\mathrm{Stab}_U(\mathsf{SP}(G))$ is the trivial group while $\mathrm{Stab}_U\{\mathsf{SP}(G)\} = \mathrm{Sym}_U$.

We now use the bounds given by Lemma 13, in conjunction with bounds on G to obtain size bounds on $\mathsf{SP}(G)$. Recall that the index of G in Sym_U, denoted $[\mathrm{Sym}_U : G]$ is the number of cosets of G in Sym_U or, equivalently, $\frac{|\mathrm{Sym}_U|}{|G|}$. The next lemma, proved via a involved combinatorial argument, says that if $\mathcal{P}$ is a partition of $[n]$ where the index of $\mathrm{Stab}_n\{\mathcal{P}\}$ in Sym_n is sufficiently small then (i) the number of parts in $\mathcal{P}$ is either very small or very big, and (ii) if the number of parts in $\mathcal{P}$ is small, then it must have a large part.

▶ **Lemma 14.** *Let ϵ and n be such that $0 \leq \epsilon < 1$ and $\log n \geq \frac{8}{\epsilon^2}$. Let $\mathcal{P}$ be a partition of $[n]$, $s := [\mathrm{Sym}_n : \mathrm{Stab}_n\{\mathcal{P}\}]$ and $n \leq s \leq 2^{n^{1-\epsilon}}$.*

1. *Let $k := |\mathcal{P}|$, then $\min\{k, n-k\} \leq \frac{8}{\epsilon}\frac{\log s}{\log n}$.*
2. *If $|\mathcal{P}| \leq \frac{n}{2}$, then $\mathcal{P}$ contains a part with at least $n - \frac{33}{\epsilon} \cdot \frac{\log s}{\log n}$ elements.*

We leverage the above combinatorial lemmas to show that in symmetric circuits of polynomial size, each gate has a small supporting partition, and hence has a small support. Let g be a gate in a rigid symmetric circuit C over universe U, we abuse notation and write $\mathsf{SP}(g)$ for $\mathsf{SP}(\mathrm{Stab}_U(g))$. Note that, if P is any part in $\mathsf{SP}(g)$, then $U \setminus P$ is a support of g in the sense of Definition 10. We write $\|\mathsf{SP}(g)\|$ to denote the smallest value of $|U \setminus P|$ over all parts P in $\mathsf{SP}(g)$. Also, let $\mathsf{SP}(C)$ denote the maximum of $\|\mathsf{SP}(g)\|$ over all gates g in C.

By the orbit-stabiliser theorem, $|\mathrm{Orb}(g)| = [\mathrm{Sym}_U : \mathrm{Stab}_U(g)]$. By Lemma 13, we have that $\mathrm{Stab}_U(g) \subseteq \mathrm{Stab}_U\{\mathsf{SP}(g)\}$ and thus, if s is an upper bound on $|\mathrm{Orb}(g)|$, $s \geq [\mathrm{Sym}_U : \mathrm{Stab}_U(g)] \geq [\mathrm{Sym}_U : \mathrm{Stab}_U\{\mathsf{SP}(g)\}]$. Then, by Part 2 of Lemma 14, g has a support of small size provided that (i) s is sub-exponential, and (ii) $\mathsf{SP}(g)$ has fewer than $n/2$ parts. Thus, to prove our main technical theorem, which formalises Theorem 3 from the introduction, it suffices to show that if s is sufficiently sub-exponential, (ii) holds.

▶ **Theorem 15** (Support Theorem). *For any ϵ and n with $\frac{2}{3} \leq \epsilon \leq 1$ and $n > 2^{\frac{56}{\epsilon^2}}$, if C is a rigid symmetric circuit over universe U with $|U| = n$ and $s := \max_{g \in C} |\mathrm{Orb}(g)| \leq 2^{n^{1-\epsilon}}$, then, $\mathsf{SP}(C) \leq \frac{33}{\epsilon} \frac{\log s}{\log n}$.*

Proof. Suppose $1 \leq s < n$. C cannot have relational inputs, because each relational gate must have an orbit of size at least n, so each gate of C computes a constant Boolean function. The support of every gate g in C must be $\{U\}$, and hence $0 = \|\mathsf{SP}(g)\| = \mathsf{SP}(C)$. Therefore assume $s \geq n$.

To conclude the theorem from Part 2 of Lemma 14 it suffices to argue that for all gates g, $|\mathsf{SP}(g)| \leq \frac{n}{2}$. Suppose g is a constant gate, then, because g is the only gate with its label, it is fixed under all permutations and hence $|\mathsf{SP}(g)| = |\{U\}| = 1 < \frac{n}{2}$. If g is a relational gate, then it is fixed by any permutation that fixes all elements appearing in $\Lambda(g)$ and moved by all others. Thus, $\mathsf{SP}(g)$ must contain singleton parts for each element of U in $\Lambda(g)$ and a part containing everything else. Thus, if $|\mathsf{SP}(g)| > \frac{n}{2}$, $\mathsf{SP}(g)$ contains at least $\frac{n}{2}$ singleton parts, there is a contradiction using the bounds on s, n, and ϵ, $s \geq |\mathrm{Orb}(g)| \geq \|\mathsf{SP}(g)\|! \cdot \binom{n}{\|\mathsf{SP}(g)\|} \geq \lfloor \frac{n}{2} \rfloor ! \geq 2^{\lfloor \frac{n}{4} \rfloor} > 2^{n^{1-\epsilon}}$.

It remains to consider internal gates. For the sake of contradiction let g be a topologically first internal gate such that $\mathsf{SP}(g)$ has more than $\frac{n}{2}$ parts. Part 1 of Lemma 14 implies, along with the assumptions on s, n, and ϵ, that $n - |\mathsf{SP}(g)| \leq k' := \left\lceil \frac{8}{\epsilon} \frac{\log s}{\log n} \right\rceil \leq \frac{1}{4} n^{1-\epsilon} < \frac{n}{2}$.

Let H denote the children of g. Because g is a topologically first gate with $|\mathsf{SP}(g)| > \frac{n}{2}$, for all $h \in H$, $\mathsf{SP}(h)$ has at most $\frac{n}{2}$ parts. As before, we argue a contradiction with the upper bound on s. This is done by demonstrating that there is a set of gate-automorphism pairs $S = \{(h, \sigma) \mid h \in H, \sigma \in \mathrm{Sym}_U\}$ that are: (i) *useful* – the automorphism moves the gate out of the set of g's children, i.e., $\sigma h \notin H$, and (ii) *independent* – each child and its image under the automorphism are fixed points of the other automorphisms in the set, i.e., for all $(h, \sigma), (h', \sigma') \in S$, $\sigma' h = h$ and $\sigma' \sigma h = \sigma h$. Note that sets which are useful and independent contain tuples whose gate and automorphism parts are all distinct. The set S describes elements in the orbit of H with respect to Sym_U.

▶ **Claim 16.** *Let S be useful and independent, then $|\mathrm{Orb}(H)| \geq 2^{|S|}$.*

Proof. Let R be any subset of S. Derive an automorphism from R: $\sigma_R := \prod_{(h,\sigma) \in R} \sigma$ (since automorphisms need not commute, fix an arbitrary ordering of S).

Let R and Q be distinct subsets of S where without loss of generality $|R| \geq |Q|$. Pick any $(h, \sigma) \in R \backslash Q \neq \emptyset$. Because S is independent $\sigma_R h = \sigma h$ and $\sigma_Q \sigma h = \sigma h$. Since S is useful, $\sigma h \notin H$. Thus $\sigma h \in \sigma_R H$, but $\sigma h \notin \sigma_Q H$. Hence $\sigma_R H \neq \sigma_Q H$. Therefore each subset of S can be identified with a distinct element in $\mathrm{Orb}(H)$ and hence $|\mathrm{Orb}(H)| \geq 2^{|S|}$. ◀

Thus to reach a contradiction it suffices to construct a sufficiently large set S of gate-automorphism pairs. To this end, divide U into $\lfloor \frac{|U|}{k'+2} \rfloor$ disjoint sets S_i of size $k' + 2$ and ignore the elements left over. Observe that for each i there is a permutation σ_i which fixes $U \backslash S_i$ but σ_i moves g, because otherwise the supporting partition of g could be smaller $(n - (k' + 2) + 1)$. Since g is moved by each σ_i and C is rigid, there must be an associated child $h_i \in H$ with $\sigma_i h_i \notin H$. Thus let (h_i, σ_i) be the gate-automorphism pair for S_i, these pairs are *useful*. Let Q_i be the union of all but the largest part of $\mathsf{SP}(h_i)$. Observe that for any σ which fixes Q_i pointwise σ also fixes both h_i and $\sigma_i h_i$, by the definition of support.

Define a directed graph K on the sets S_i as follows. Include an edge from S_i to S_j, with $i \neq j$, if $Q_i \cap S_j \neq \emptyset$. An edge in K indicates a potential lack of independence between (h_i, σ_i) and (h_j, σ_j), and on the other hand if there are no edges between S_i and S_j, the associated pairs are independent. Thus it remains to argue that K has a large independent set. This is

possible because the out-degree of S_i in K is bounded by $|Q_i| = \|\mathsf{SP}(h_i)\| \leq \frac{33}{\epsilon}\frac{\log s}{\log n}$ as the sets S_i are disjoint and Part 2 of Lemma 14 can be applied to h_i. Thus the average total degree (in-degree + out-degree) of K is at most $9k'$. Greedily select a maximal independent set in K by repeatedly selecting the S_i with the lowest total degree and eliminating it and its neighbours. This action does not effect the bound on the average total degree of K and hence determines an independent set I in K of size at least

$$\frac{\lfloor \frac{|U|}{k'+2} \rfloor}{9k'+1} \geq \frac{n-(k'+2)}{(9k'+1)(k'+2)} \geq \frac{\frac{n}{2}-1}{9k'^2+10k'+2} \geq \frac{\frac{7}{16}n}{9k'^2+10k'+2} \geq \frac{n}{(7k')^2}$$

where the first inequality follows by expanding the floored expression, the second follows because $k' < \frac{n}{2}$, the third follows from the lower bound on n, and the last follows because $k' \geq 1$ as it is the ceiling of a positive non-zero quantity by definition.

Take $S := \{(h_i, \sigma_i) \mid S_i \in I\}$. By the argument above S is useful and independent. By Claim 16, conclude that $s \geq |\mathrm{Orb}(g)| \geq |\mathrm{Orb}(H)| \geq 2^{|S|} \geq 2^{\frac{n}{(7k')^2}}$. For $\epsilon \geq \frac{2}{3}$, $s \leq 2^{n^{1-\epsilon}}$, and $\frac{\epsilon}{56}\log n > 1$ the following is a contradiction $\log s \geq n \cdot (\frac{56}{\epsilon}\frac{\log s}{\log n})^{-2} > n \cdot (n^{1-\epsilon})^{-2} = n^{2\epsilon-1} \geq n^{1-\epsilon}$. Thus $|\mathsf{SP}(g)| \leq \frac{n}{2}$ for all $g \in C$ and the proof is complete by Part 2 of Lemma 14. ◀

Observe that when s is polynomial in n the support of a rigid symmetric circuit family is asymptotically constant. This is the case for polynomial-size families.

▶ **Corollary 17.** *Let $\mathcal{C}$ be a polynomial-size rigid symmetric circuit family, then $\mathsf{SP}(\mathcal{C}) = O(1)$.*

4 Translating Symmetric Circuits to Formulas

In this section, we deploy the Support Theorem to show that P-uniform families of symmetric circuits can be translated into formulas of fixed-point logic. We can show that there is a polynomial-time algorithm that takes a symmetric circuit and outputs an equivalent rigid symmetric circuit together with the supporting partitions of each gate.

▶ **Lemma 18.** *Let C be a symmetric $(\mathbb{B}, \tau)$-circuit with universe U. There is a deterministic algorithm which runs in time $\mathrm{poly}(|C|)$ and outputs a rigid symmetric $(\mathbb{B}, \tau)$-circuit C' computing the same query as C along with coarsest supporting partitions for every gate of C'.*

Let $\mathcal{C} = (C_n)_{n\in\mathbb{N}}$ be a family of P-uniform symmetric $(\mathbb{B}, \tau)$-circuits computing a q-ary query. Let $\mathcal{A} \in \mathrm{fin}[\tau]$ be an input structure with universe U of size n. By Lemma 18 and the Immerman-Vardi theorem, we have an FP interpretation defining a rigid symmetric circuit equivalent to C_n over the number sort of $\mathcal{A}^{\leq}$, i.e., a tuple of formulas of $\mathrm{FP}(\leq)$ that define the circuit when interpreted in $\langle [n], \leq \rangle$. Moreover, the interpretation provides the coarsest supporting partitions of the gates in C_n. Note that C_n is defined over the universe $[n]$.

By Theorem 15, there is a constant bound k so that for each gate g in C_n the union of all but the largest part of the coarsest partition supporting g, $\mathsf{SP}(g)$, has at most k elements. Moreover, this union is a *support* of g in the sense of Definition 10. We call it the *canonical support* of g and denote it by $\mathrm{sp}(g)$. To describe the evaluation of the circuit C_n with a formula of fixed-point logic, we show that the evaluation of a gate g in C_n with respect to the structure $\mathcal{A}$ depends only on how its universe U is mapped to the canonical support of g.

For any set $X \subseteq [n]$, let U^X denote the set of *injective* functions from X to U. For $X, Y \subseteq [n]$ and $\alpha \in U^X, \beta \in U^Y$, we say α and β are *consistent*, denoted $\alpha \sim \beta$, if for all $z \in X \cap Y, \alpha(z) = \beta(z)$, and for all $x \in X \backslash Y$ and $y \in Y \backslash X$, $\alpha(x) \neq \beta(y)$. Recall that any bijection $\gamma : U \to [n]$ determines an evaluation of the circuit C_n on the input structure $\mathcal{A}$ which assigns to each gate g the Boolean value $C_n[\gamma\mathcal{A}](g)$. Let g be a gate and let

$\Gamma(g) := \{\gamma \mid C_n[\gamma\mathcal{A}](g) = 1\}$. The following claim establishes that the membership of γ in $\Gamma(g)$ (moreover, the number of 1s input to g) depends only on what γ maps to $\mathrm{sp}(g)$.

▶ **Claim 19.** *Let g be a gate in C_n with children H. Let $\alpha \in U^{\mathrm{sp}(g)}$, then for all $\gamma_1, \gamma_2 : U \to [n]$ with $\gamma_1^{-1} \sim \alpha$ and $\gamma_2^{-1} \sim \alpha$,*

1. *$\gamma_1 \in \Gamma(g)$ iff $\gamma_2 \in \Gamma(g)$.*
2. *$|\{h \in H \mid \gamma_1 \in \Gamma(h)\}| = \sum_{h \in H} \frac{|A_h \cap \mathrm{EV}_h|}{|A_h|}$, where for $h \in H$, $A_h := \{\beta \in U^{\mathrm{sp}(h)} \mid \alpha \sim \beta\}$.*

We associate with each gate g a set of injective functions $\mathrm{EV}_g \subseteq U^{\mathrm{sp}(g)}$ defined by $\mathrm{EV}_g := \{\alpha \in U^{\mathrm{sp}(g)} \mid \exists \gamma \in \Gamma(g) \wedge \alpha \sim \gamma^{-1}\}$ and note that, by Claim 19, this completely determines $\Gamma(g)$. Since $[n]$ is linearly ordered, $X \subseteq [n]$ inherits this order and we write $\vec{X}$ for the ordered $|X|$-tuple consisting of the elements of X in the inherited order. For $\alpha \in U^X$ write $\vec{\alpha} \in U^{\vec{X}}$ for the tuple $\alpha(\vec{X})$. This allows us to encode injective functions as tuples over U e.g., $\vec{\mathrm{EV}}_g := \{\vec{\alpha} \mid \alpha \in \mathrm{EV}_g\}$. Using Claim 19 we can construct $\vec{\mathrm{EV}}_g$ inductively over C_n.

- Let g be a constant input gate, then $\mathrm{sp}(g)$ is empty. If $\Sigma(g) = 0$, then $\Gamma(g) = \emptyset$ and $\vec{\mathrm{EV}}_g = \emptyset$. Otherwise $\Sigma(g) = 1$, then $\Gamma(g)$ is all bijections and $\vec{\mathrm{EV}}_g = \{\langle\rangle\}$, i.e., the set containing the empty tuple.
- Let g be a relational gate with $\Sigma(g) = R \in \tau$, then $\mathrm{sp}(g)$ is the set of elements in the tuple $\Lambda_R(g)$. By definition we have $\vec{\mathrm{EV}}_g = \{\vec{\alpha} \in U^{\vec{\mathrm{sp}}(g)} \mid \alpha(\Lambda_R(g)) \in R^{\mathcal{A}}\}$.
- Let $\Sigma(g) = \mathrm{AND}$ and consider $\vec{\alpha} \in U^{\vec{\mathrm{sp}}(g)}$. By Claim 19, $\vec{\alpha} \in \vec{\mathrm{EV}}_g$ iff $\vec{A_h} = \vec{\mathrm{EV}}_h$ for *every* child h of g, i.e., for *every* child h and *every* $\beta \in U^{\mathrm{sp}(h)}$ with $\alpha \sim \beta$, we have $\vec{\beta} \in \vec{\mathrm{EV}}_h$.
- Let $\Sigma(g) = \mathrm{OR}$ and consider $\vec{\alpha} \in U^{\vec{\mathrm{sp}}(g)}$. By Claim 19, $\vec{\alpha} \in \vec{\mathrm{EV}}_g$ iff there is a child h of g where $\vec{A_h} \cap \vec{\mathrm{EV}}_h$ is non-empty, i.e., for *some* child h of g and *some* $\beta \in U^{\mathrm{sp}(h)}$ with $\alpha \sim \beta$, we have $\vec{\beta} \in \vec{\mathrm{EV}}_h$.
- Let $\Sigma(g) = \mathrm{NOT}$ and consider $\vec{\alpha} \in U^{\vec{\mathrm{sp}}(g)}$. g has exactly one child h. Claim 19 implies that $\vec{\alpha} \in \vec{\mathrm{EV}}_g$ iff $\vec{A_h} \neq \vec{\mathrm{EV}}_h$, i.e., for *some* $\beta \in U^{\mathrm{sp}(h)}$ with $\alpha \sim \beta$, we have $\vec{\beta} \notin \vec{\mathrm{EV}}_h$.
- Let $\Sigma(g) = \mathrm{MAJ}$ and consider $\vec{\alpha} \in U^{\vec{\mathrm{sp}}(g)}$. Let H be the set of children of g and let $A_h := \{\beta \in U^{\mathrm{sp}(h)} \mid \beta \sim \alpha\}$. Then Claim 19 implies that $\vec{\alpha} \in \vec{\mathrm{EV}}_g$ if, and only if,

$$\sum_{h \in H} \frac{|\vec{A_h} \cap \vec{\mathrm{EV}}_h|}{|\vec{A_h}|} \geq \frac{|H|}{2}. \tag{1}$$

From $\vec{\mathrm{EV}}$ we can recover the query Q computed by C_n on the input structure $\mathcal{A}$ because the support of an output gate g is exactly the set of elements in the marking of g by Λ_Ω. In particular: $Q = \{\overline{a} \in U^q \mid \exists g \in G, \vec{\alpha} \in \vec{\mathrm{EV}}_g \text{ such that } \Lambda_\Omega(\alpha^{-1}(\overline{a})) = g\}$.

It is then straightforward (if laborious) to turn the inductive construction of $\vec{\mathrm{EV}}_g$ given above to a fixed-point formula defining the relation $V \subseteq [n]^t \times U^k$ by $V(g, \overline{a})$ if, and only if, $\tilde{a} \in \vec{\mathrm{EV}}_g$ in the structure $\mathcal{A}^{\leq}$, where $\tilde{a}$ is the restriction of the tuple $\overline{a}$ to $|\mathrm{sp}(g)|$ elements. Here t is the arity of the FP-interpretation of the circuit C_n in the structure $\mathcal{A}^{\leq}$. From this we get a formula defining the query Q given by the circuit family $\mathcal{C}$.

The only use of counting operators in the construction of the formula is in translating the inductive step corresponding to majority gates. Thus, the formula we obtain is one of $\mathrm{FP} + \leq$ if $\mathbb{B}$ is the standard basis and of FPC if $\mathbb{B}$ is the majority basis. Moreover, if the family $\mathcal{C} = (C_n)_{n \in \mathbb{N}}$ is not P-uniform, but given by an advice function Υ, we get an equivalent formula of $\mathrm{FP} + \Upsilon$ (for the standard basis) or $\mathrm{FPC} + \Upsilon$ (for the majority basis).

On the other hand, formulas of $\mathrm{FP} + \leq$ can be translated into P-uniform families of symmetric Boolean circuits by standard methods and similar translations hold for FPC and $\mathrm{FP} + \Upsilon$. Putting this all together gives us our main theorem.

▶ **Theorem 20** (Main). *The following pairs of classes define the same queries on structures:*
1. *Symmetric* P*-uniform Boolean circuits and* $\mathrm{FP} + \leq$.
2. *Symmetric* P*-uniform majority circuits and* FPC.
3. *Symmetric* P/poly*-uniform Boolean circuits and* $\mathrm{FP} + \Upsilon$.
4. *Symmetric* P/poly*-uniform majority circuits and* $\mathrm{FPC} + \Upsilon$.

One consequence is that properties of graphs which we know not to be definable in FPC are also not decidable by P-uniform families of symmetric circuits. The results of Cai-Fürer-Immerman [3] give graph properties that are polynomial-time decidable, but not definable in FPC. Furthermore, there are a number of natural NP-complete graph problems known not to be definable in FPC, including Hamiltonicity and 3-colourability (see [4]). Indeed, all these proofs actually show that these properties are not even definable in the infinitary logic with a bounded number of variables and counting ($C^{\omega}_{\infty\omega}$—see [9]). Since it is not difficult to show that formulas of $\mathrm{FPC} + \Upsilon$ can be translated into $C^{\omega}_{\infty\omega}$, we have the following.

▶ **Corollary 21.** *Hamiltonicity and 3-colourability of graphs are not decidable by families of* P/poly*-uniform symmetric majority circuits.*

5 Coherent and Locally Polynomial Circuits

Otto [10] studies families of rigid symmetric Boolean circuits deciding properties of structures where the families satisfy two uniformity properties. Informally, a circuit family $\mathcal{C} := (C_n)_{n\in\mathbb{N}}$ is *coherent* if C_n appears as a subcircuit consisting of exactly the gates fixed by $\mathrm{Sym}_{[m]\setminus[n]}$ of all but finitely many of the circuits C_m at input length $m > n$. Second, $\mathcal{C}$ is *locally polynomial of degree* k if the size of the orbit of every wire in C_n is at most n^k. The main result [10, Theorem 6] is that coherent locally-polynomial of degree k families of symmetric $(\mathbb{B}_{\mathrm{std}}, \tau)$-circuits computing Boolean properties of $\mathrm{fin}[\tau]$ correspond to infinitary FO with k variables. In Otto's definition, individual circuits in the family may themselves be infinite, as the only size restriction is on the orbits of wires. The theorem also shows that if the circuit families are constant depth they correspond to the fragment of FO with k variables.

In all notions of uniformity we consider the circuits are of polynomial size. The Support Theorem can be used to establish a direct connection between polynomial-size symmetric circuit families and the locally-polynomial coherent symmetric families.

▶ **Proposition 22** (Informal). *Let* $\mathcal{C} := (C_n)_{n\in\mathbb{N}}$ *be a family of rigid symmetric Boolean circuits.*
1. *If* $\mathcal{C}$ *is locally-polynomial and coherent, then* $\mathcal{C}$ *is polynomial size.*
2. *If* $\mathcal{C}$ *is polynomial size, then* $\mathcal{C}$ *is locally polynomial.*

Since there are properties definable in an infinitary logic with finitely many variables that are not decidable by polynomial-size circuits, it follows from the above proposition that the use of infinite circuits is essential in Otto's result.

Proposition 22 implies that all uniform circuit families we consider are locally polynomial. However, they are not necessarily coherent. Indeed there are Boolean circuit families uniformly definable in $\mathrm{FO} + \leq$ that are not coherent. To see this observe that such circuit families may include gates that are completely indexed by the number sort and hence are fixed under all automorphisms induced by permutations of the point sort. Moreover the number of such gates may increase as a function of input length. However, under the definition of coherence, the number of gates in each circuit of a coherent family that are not moved by any automorphism must be identical. Thus there are uniform circuits that are not coherent.

6 Future Directions

One of the original motivations for studying symmetric majority circuits was the hope that they had the power of choiceless polynomial time with counting (CPTC) [2], and that, perhaps, techniques from circuit complexity could improve our understanding of the relationship between CPTC and the invariant queries definable in polynomial time. However, because FPC $\subsetneq$ CPTC [5], our results indicate that symmetry is too much of a restriction on P-uniform circuit families to recover CPTC.

A natural way to weaken the concept of symmetry is to require that induced automorphisms exist only for a certain subgroup of the symmetric group. This interpolates between our notion of symmetric circuits and circuits on linearly-ordered structures, with the latter case occurring when the subgroup is the identity.

The Support Theorem is a fairly general statement about the structure of symmetric circuits and is largely agnostic to the particular semantics of the basis. To that end the Support Theorem may find application to circuits over bases not consider here. The Support Theorem can be applied to arithmetic circuits computing invariant properties of matrices over a field; e.g., the Permanent polynomial is invariant and one standard way to compute it is as a symmetric arithmetic circuit, i.e., Ryser's formula [11]. Finally, the form of the Support Theorem can, perhaps, be improved as the particular upper bound required on the orbit size does not appear to be fundamental to the conclusion it reaches.

Acknowledgments. The authors thank Dieter van Melkebeek for looking at an early draft of this paper. This research was supported by EPSRC grant EP/H026835.

References

1 M. Anderson and A. Dawar. On symmetric circuits and fixed-point logics. *arXiv 1401.1125*, 2014.
2 A. Blass, Y. Gurevich, and S. Shelah. Choiceless polynomial time. *Annals of Pure and Applied Logic*, 100:141–187, 1999.
3 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.
4 A. Dawar. A restricted second order logic for finite structures. *Information and Computation*, 143:154–174, 1998.
5 A. Dawar, D. Richerby, and B. Rossman. Choiceless polynomial time, counting and the Cai–Fürer–Immerman graphs. *Annals of Pure and Applied Logic*, 152(1):31–50, 2008.
6 L. Denenberg, Y. Gurevich, and S. Shelah. Definability by constant-depth polynomial-size circuits. *Information and Control*, 70(2):216–240, 1986.
7 H.D. Ebbinghaus and J. Flum. *Finite Model Theory*. Springer, 2006.
8 N. Immerman. Relational queries computable in polynomial time. *Information and Control*, 68(1-3):86–104, 1986.
9 M. Otto. *Bounded Variable Logics and Counting: A Study in Finite Models*, volume 9 of *Lecture Notes in Logic*. Springer-Verlag, 1997.
10 M. Otto. The logic of explicitly presentation-invariant circuits. In Dirk van Dalen and Marc Bezem, editors, *Computer Science Logic*, volume 1258 of *Lecture Notes in Computer Science*, pages 369–384. Springer Berlin Heidelberg, 1997.
11 H.J. Ryser. *Combinatorial Mathematics*. Mathematical Association of America, 1963.
12 M. Vardi. The complexity of relational query languages. In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, pages 137–146. ACM, 1982.