Symmetric Circuits

Anuj Dawar

University of Cambridge Computer Laboratory

joint work with Matthew Anderson

IMSc, Chennai, 8 January 2015

Circuit Complexity

A *language* $L \subseteq \{0,1\}^*$ can be described by a family of *Boolean functions*:

$$(f_n)_{n\in\omega} : \{0,1\}^n \to \{0,1\}.$$

Each f_n may be computed by a *circuit* C_n made up of

- Gates labeled by Boolean operators: $\wedge, \vee, \neg$,
- Boolean inputs: $x_1, \ldots, x_n$, and
- A distinguished gate determining the output.

If there is a polynomial $p(n)$ bounding the *size* of C_n, i.e. the number of gates in C_n, the language L is in the class $\mathrm{P/poly}$.

If, in addition, the function $n \mapsto C_n$ is computable in *polynomial time*, L is in P.

Note: For these classes it makes no difference whether the circuits only use $\{\wedge, \vee, \neg\}$ or a richer basis with *threshold* or *majority* gates.

Circuit Lower Bounds

It is conjectured that $\mathrm{NP} \not\subseteq \mathrm{P/poly}$.

Lower bound results have been obtained by putting further restrictions on the circuits:

- No *constant-depth* (unbounded fan-in), *polynomial-size* family of circuits decides *parity*. **(Furst, Saxe, Sipser 1983)**.
- No *polynomial-size* family of *monotone* circuits decides *clique*. **(Razborov 1985)**.
- No *constant-depth*, $O(n^{\frac{k}{4}})$-*size* family of circuits decides *k-clique*. **(Rossman 2008)**.

No known result separates NP from *constant-depth*, *polynomial-size* families of circuits with *majority gates*.

Circuits for Graph Properties

We want to study families of circuits that decide properties of *graphs* (or other relational structures—for simplicity of presentation we restrict ourselves to graphs).

We have a family of Boolean circuits $(C_n)_{n\in\omega}$ where there are n^2 inputs labelled $(i,j) : i,j \in [n]$, corresponding to the *potential edges*.
Each input takes value 0 or 1;

Graph properties in P are given by such families where:

- the size of C_n is bounded by a polynomial $p(n)$; and
- the family is uniform, so the function $n \mapsto C_n$ is in P (or $\mathrm{DLogTime}$).

Invariant Circuits

C_n is *invariant* if, for every input graph, the output is unchanged under a permutation of the inputs induced by a permutation of $[n]$.

That is, given any input $G : [n]^2 \to \{0,1\}$, and a permutation $\pi \in S_n$,

C_n *accepts* G *if, and only if,* C_n *accepts the input* πG *given*

$$(\pi G)(i,j) = G(\pi(i), \pi(j)).$$

This defines a class of Boolean functions far more general than the *symmetric* ones, including all isomorphism-invariant graph properties such as *connectivity*, *perfect matching*, *Hamiltonicity*, *3-colourability*.

Symmetric Circuits

Say C_n is *symmetric* if any permutation of $[n]$ applied to its inputs can be extended to an automorphism of C_n.

> *i.e., for each $\pi \in S_n$, there is an automorphism of C_n that takes input (i, j) to $(\pi i, \pi j)$.*

Any symmetric circuit is invariant, but *not* conversely.

> *Consider the natural circuit for deciding whether the number of edges in an n-vertex graph is even.*

Any invariant circuit can be converted to a symmetric circuit, but with potentially *exponential blow-up*.

First-Order Logic

We consider *logic* as a language for specifying properties of graphs and the translation of these specifications.

First-order Logic:

A collection X of variables, and formulas:

$$E(x, y) \mid \phi \wedge \psi \mid \phi \vee \psi \mid \neg\phi \mid \exists x\phi \mid \forall x\phi$$

A formula ϕ without free variables specifies a property of graphs.

$$\exists x \exists y \exists z (x \neq y \wedge y \neq z \wedge x \neq z \wedge \neg E(x, y) \wedge \neg E(x, z) \wedge \neg E(y, z))$$

defines the graphs that have an independent set of size 3.

Logic and Circuits

Any formula of ϕ *first-order logic* translates into a uniform family of circuits C_n

> *For each subformula $\psi(\overline{x})$ and each assignment $\overline{a}$ of values to the free variables, we have a gate.*
> *Existential quantifiers translate to big disjunctions, etc.*

The circuit C_n is:

- of *constant* depth (given by the depth of ϕ);
- of size at mose $c \cdot n^k$ where c is the number of subformulas of ϕ and k is the *maximum number of free variables* in any subformula of ϕ.
- *symmetric* by the action of $\pi \in S_n$ that takes $\psi[\overline{a}]$ to $\psi[\pi(\overline{a})]$.

Fixed-Point Logic

The logic FP is formed by adding to first-order logic a mechanism for *inductive definitions*.

The formula

$$\forall u \forall v [\mathbf{lfp}_{T,xy}(x = y \vee \exists z(E(x,z) \wedge T(z,y)))](u,v)$$

is satisfied in a graph (V, E) if, and only if, it is connected.

On structures which come equipped with a linear order FP expresses exactly the classes that are decidable in *polynomial time*.

(Immerman; Vardi)

In the *absence* of order, there is no formula of FP that defines

- the graphs with an even number of *vertices*.
- the graphs with an even number of *edges*.

FP and Circuits

For every sentence ϕ of FP and every n, there is a formula ϕ_n of *first-order logic* that is equivalent to ϕ on all graphs with at most n vertices.

The formula ϕ_n has

- *depth* n^c for some constant c;
- at most k free variables in each sub-formula for some constant k.

It follows that every graph property definable in FP is given by a family of *polynomial-size, symmetric* circuits.

Note: the inexpressibility results for *evenness* really show that it is not definable by any family of formulas with a constant number of variables.

Fixed-Point Logic with Counting

FPC is a logic formulated to add the ability to count to FP.
It was once proposed as a candidate logic for expressing all properties in P.

Two sorts of variables:

- $x_1, x_2, \ldots$ which range over $|A|$—the domain of the structure, and
- $\nu_1, \nu_2, \ldots$ which range over the *numbers* $0, 1, \ldots, |A|$.

If $\phi(x)$ is a formula with free variable x, then $\nu = \#x\phi$ denotes that ν is the number of elements of A that satisfy the formula ϕ.

Counting Quantifiers

C^k is the logic obtained from *first-order logic* by allowing:

- *counting quantifiers*: $\exists^i x\, \phi$; and
- only the variables $x_1, \ldots . x_k$.

Every formula of C^k is equivalent to a formula of first-order logic, albeit one with more variables.

For every sentence ϕ of FPC, there is a k such that for any fixed n, there is a formula of C^k equivalent to ϕ on structures with at most n elements.

It follows that any graph property expressible in FPC is given by a polynomial-size family of *symmetric*, circuits with *counting* (or *threshold*) gates.

Expressive Power of FPC

Most *"obviously"* polynomial-time algorithms can be expressed in FPC.

This includes P-complete problems such as CVP—*the Circuit Value Problem* as well as *2-colourability* and *2-SAT* (all expressible in FP).

Many non-trivial polynomial-time algorithms can be expressed in FPC:

- FPC captures all of P over any *proper minor-closed class of graphs* **(Grohe 2010)**
- FPC can express *linear programming* problems; *max-flow* and *maximum matching* on graphs. **(Anderson, D., Holm 2013)**

Limitations of FPC

There are polynomial-time decidable properties of graphs that are not definable in FPC. **(Cai, Fürer, Immerman, 1992)**

Other inexpressiblity results for FPC follow, either as a consequence of **(Cai, Fürer, Immerman, 1992)** or by similar methods:

- *Hamiltonian Cycle* and *Satisfiability* are not definable in FPC.
- *3-Colourability* is not definable in FPC.

  (D. 1998)
- Solvability of systems of linear equations (over any fixed finite Abelian group) is not definable in FPC

  (Atserias, Bulatov, D. 2009)

All of these are shown, in fact, to be not definable by any (even *non-uniform*) family of C^k formulas, for any k.

Main Results

Theorem

A class of graphs is accepted by a P*-uniform, polynomial-size, symmetric family of Boolean circuits if, and only if, it is definable by an* FP *formula interpreted in* $G \uplus ([n], <)$.

Theorem

A class of graphs is accepted by a P*-uniform, polynomial-size, symmetric family of threshold circuits if, and only if, it is definable in* FPC.

Some Consequences

We get a natural and purely circuit-based characterisation of FPC definability.

Inexpressibility results for FP and FPC yield lower bound results against natural circuit classes.

- There is no polynomial-size family of symmetric Boolean circuits deciding if an n vertex graph has an even number of edges.
- Polynomial-size families of uniform symmetric *threshold circuits* are more powerful than Boolean circuits.
- Invariant circuits *cannot* be translated into equivalent symmetric threshold circuit, with only polynomial blow-up.

Technical Tools – Rigidity

For a symmetric circuit C_n we can assume *w.l.o.g.* that the automorphism group is the symmetric group S_n acting in the natural way.

On the one hand, as long as each element of $[n]$ appears as a label of *some input gate* of C_n, distinct permutations in S_n give rise to distinct autmorphisms of C_n

On the other hand, we can *in polynomial time* transform a symmetric circuit into a *rigid symmetric circuit*—whose only automorphisms are those induced by S_n.

Technical Tools – Support

For a gate g in C_n, $\mathrm{Stab}(g)$ denotes the *stabilizer group of* g, i.e.,

$$\mathrm{Stab}(g) = \{\pi \in S_n \mid \pi(g) = g\}.$$

Say a set $X \subseteq [n]$ *supports* g if

$$\mathrm{Stab}^{\bullet}(X) \subseteq \mathrm{Stab}(g),$$

where $\mathrm{Stab}^{\bullet}(X) := \{\pi \in S_n \mid \pi(x) = x \text{ for all } x \in X\}$ is the *pointwise stabilizer* of X.

Note: For the family of circuits $(C_n)_{n\in\omega}$ obtained from an FPC formula there is a constant k such that all gates in each C_n have a support of size at most k.

Technical Tools—Supporting Partitions

We want to to show that in a symmetric circuit of polynomial size, each gate has support of bounded size. To this end, we introduce *supporting partitions*.

For a permutation group $G \subseteq S_n$, say that a partition $\mathcal{P}$ of $[n]$ *supports* G if every permutation that fixes each $P \in \mathcal{P}$ is in G:

Lemma: There is a *coarsest* partition (denote it $\mathrm{SP}(G)$) that supports G.

Proof sketch: For two partitions $\mathcal{P}$ and $\mathcal{P}'$, let $\mathcal{E}(\mathcal{P}, \mathcal{P}')$ denote the finest partition that is coarser than $\mathcal{P}$ and $\mathcal{P}'$.
Then, any permutation that fixes each part in $\mathcal{E}(\mathcal{P}, \mathcal{P}')$ can be expressed as a composition of permutations fixing all parts in $\mathcal{P}$ and $\mathcal{P}'$ respectively.

Technical Tools – Supporting Partitions

Writing $\mathrm{Stab}^\bullet(\mathrm{SP}(G))$ for the the group of permutations that fix each part in $\mathrm{SP}(G)$ and $\mathrm{Stab}(\mathrm{SP}(G))$ for the group of permutations that fix the partition $\mathrm{SP}(G)$ *setwise*, we have:

$$\mathrm{Stab}^\bullet(\mathrm{SP}(G)) \subseteq G \subseteq \mathrm{Stab}(\mathrm{SP}(G)).$$

The first inclusion is by definition. The second follows from the fact that for any permutation $\pi \in S_n$, $\pi\mathrm{SP}(G)$ is the coarsest supporting partition of the group $\pi G \pi^{-1}$.

By the *orbit-stabilizer* theorem, the size of the *orbit* of any gate g in C_n is $\frac{n!}{|\mathrm{Stab}(g)|}$.

So, an upper bound on $\mathrm{Stab}(g)$ gives us a lower bound on the orbit of g.

Conversely, knowing that the orbit of g is at most polynomial in n gives us bounds on $\mathrm{SP}(\mathrm{Stab}(g))$.

Support Theorem

Our main technical theorem shows that in *sub-exponential size* symmetric circuits, all gates have *small* support.

Theorem

For any $1 > \epsilon \geq \frac{2}{3}$, *let* C *be a symmetric* s*-gate circuit over* $[n]$ *with* $n \geq 2^{\frac{56}{\epsilon^2}}$, *and* $s \leq 2^{n^{1-\epsilon}}$. *Then every gate* g *of* C *has a support of size at most* $\frac{33}{\epsilon}\frac{\log s}{\log n}$.

We prove this by bounding the supporting partitions of the stabliser groups of gates.

Corollary

Polynomial-size symmetric circuits have constant support.

Proof Sketch of Support Theorem – 1

Fix a permutation group G with $[S_n : G] \leq s$.

Claim: If k is the number of parts in $\mathrm{SP}(G)$ then $\min\{k, n-k\} \leq \frac{8}{\epsilon}\frac{\log s}{\log n}$.

This is a computation of the number of permutations that setwise fix a partition $\mathcal{P}$ with k parts.

We can show that, unless $\min\{k, n-k\} \leq \frac{8}{\epsilon}\frac{\log s}{\log n}$,

$$\frac{n!}{|\mathrm{Stab}(\mathcal{P})|} > s.$$

so, $[S_n : G] = \frac{n!}{|G|} \geq \frac{n!}{|\mathrm{Stab}(\mathrm{SP}(G))|} > s$.

Say that $\mathrm{SP}(G)$ is *small* if it has at most $\frac{8}{\epsilon}\frac{\log s}{\log n}$ parts and *big* otherwise.

Proof Sketch of Support Theorem – 2

Claim: If $\mathrm{SP}(G)$ is *small* then the largest part has size at least $n - \frac{33}{\epsilon}\frac{\log s}{\log n}$.

This is again proved by showing that if $\mathcal{P}$ has fewer than $\frac{8}{\epsilon}\frac{\log s}{\log n}$ parts and all of them are smaller than $n - \frac{33}{\epsilon}\frac{\log s}{\log n}$, then there are too few permutations in $\mathrm{Stab}(\mathcal{P})$, i.e.

$$\frac{n!}{|\mathrm{Stab}(\mathcal{P})|} > s$$

Proof Sketch of Support Theorem – 3

Claim: For a gate g in C_n, $\mathrm{SP}(\mathrm{Stab}(g))$ is small.

Suppose that g is a minimal gate (in the *DAG*-order of the circuit) with $\mathrm{SP}(\mathrm{Stab}(g))$ large.

We can show that this implies that g has a large number of immediate predecessors which (*by assumption*) have small supporting partitions.

Using the bounds from the previous claims, we can find a large enough subset of these, and *independently* combine automorphisms that move them.

This is used to show that $\mathrm{Orb}(g)$ must be bigger than s.

Small Supports

Thus, for each g in C, $\mathrm{SP}(\mathrm{Stab}(g))$ has a part with $n - \frac{33}{\epsilon}\frac{\log s}{\log n}$ elements.

The complement of this large part is a *support* of the gate g, which gives us the *support theorem*:

Theorem

For any $1 > \epsilon \geq \frac{2}{3}$, let C be a symmetric s-gate circuit over $[n]$ with $n \geq 2^{\frac{56}{\epsilon^2}}$, and $s \leq 2^{n^{1-\epsilon}}$. Then every gate g of C has a support of size at most $\frac{33}{\epsilon}\frac{\log s}{\log n}$.

We write $\mathrm{sp}(g)$ for the small support of g given by this theorem and note that it can be computed in polynomial time from a symmetric circuit C.

Translating Symmetric Circuits to Formulas

Given a polynomial-time function $n \mapsto C_n$ that generates symmetric circuits:

1. There are formulas of FP interpreted on $([n], <)$ that define the structure C_n.
2. We can also compute in polynomial time (and therefore in FP on $([n], <)$) $\mathrm{sp}(g)$ for each gate g.
3. For an input structure $\mathbb{A}$ and an assignment $\gamma : [n] \to \mathbb{A}$ of the inputs of C_n to elements of $\mathbb{A}$, whether g is made true depends only on $\gamma(\mathrm{sp}(g))$.
4. We define, by induction on the structure of C_n, the set of tuples $\Gamma(g) \subseteq \mathbb{A}^{\mathrm{sp}(g)}$ that represent assignments γ making g true.
5. This inductive definition can be turned into a formula (of FP for a Boolean circuit, of FPC for one with threshold gates.)

Upper and Lower Bounds

The class of properties decided by *symmetric*, *polynomial size*, *threshold* circuits is FPC—a proper subset of FPC.
This has interesting *upper* and *lower* bounds which makes it an interesting object of study.

Upper Bounds	*Lower Bounds*
CVP	SAT
2-Colourability	3-Colourability
2-SAT	3-SAT
Perfect Matching	Hamiltonian Cycle
Linear Programming	XOR-SAT
Isomorphism on planar graphs	Isomorphism on bounded-degree graphs