Symmetric Circuits and Fixed-Point Logics

Anuj Dawar

University of Cambridge Computer Laboratory
visiting RWTH Aachen

joint work with Matthew Anderson

Aachen, 8 November 2013
Is There a Logic for P?

The question of whether or not there is a logic expressing exactly the P properties of *(unordered)* relational structures is the central problem in *Descriptive Complexity*.

If we assume structures are *ordered*, then FP, the extension of first-order logic with least fixed points suffices. *(Immerman; Vardi 1982)*

In the absence of order FP fails to express simple cardinality properties such as *evenness*. 
Immerman had proposed FPC—the extension of FP with a mechanism for *counting*

Two sorts of variables:
- $x_1, x_2, \ldots$ range over $|A|$—the domain of the structure;
- $\nu_1, \nu_2, \ldots$ which range over *numbers* in the range $0, \ldots, |A|$

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

We also have the order $\nu_1 < \nu_2$, which allows us (using recursion) to define arithmetic operations.
Expressive Power of FPC

Most “obviously” polynomial-time algorithms can be expressed in FPC.

This includes \( \mathbf{P} \)-complete problems such as

CVP—\textit{the Circuit Value Problem}

\textit{Input: a circuit, i.e. a labelled DAG with source labels from \{0, 1\}, internal node labels from \{\lor, \land, \neg\}.}

\textit{Decide: what is the value at the output gate.}

CVP is expressible in FP.

It is expressible in FPC for circuits that may include \textit{threshold or counting gates}.
Expressive Power of FPC

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 2012)

• FPC can express *linear programming* problems; *max-flow* and *maximum matching* on graphs. (Anderson, D., Holm 2013)

But some cannot be expressed:

• There are polynomial-time decidable properties of graphs that are not definable in FPC. (Cai, F"urer, Immerman, 1992)

• Solvability of a system of linear equations over a finite field cannot be expressed in FPC. (Atserias, Bulatov, D. 2009)
Circuit Complexity

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

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

Each $f_n$ may be given by a *circuit* $C_n$ made up of Boolean gates, with $n$ Boolean inputs and one output.

If the size of $C_n$ is bounded by a polynomial in $n$, the language $L$ is in the class $P/\text{poly}$.

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

*Note:* it makes no difference if the circuits only use $\{\text{AND, OR, NOT}\}$ or a richer basis with *unbounded fan-in or threshold gates*. 

Anuj Dawar

November 2013
Circuit Lower Bounds

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

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

No result separates $\text{NP}$ from constant-depth, polynomial-size families of circuits with threshold gates.
A property of *graphs* (or other relational structures) in $P$ is recognised by a family of Boolean circuits $C_n$:

- inputs to $C_n$ are $n^2$ potential edges, each taking value 0 or 1;
- the size of $C_n$ is bounded by a polynomial $p(n)$;
- the family is uniform, so the function $n \mapsto C_n$ is in $P$ (or DLogTime).

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

Anuj Dawar November 2013
Say $C_n$ is **symmetric** if any permutation of $[n]$ applied to its inputs can be extended to an automorphism of $C_n$.

- Any symmetric circuit is invariant, but *not* conversely.
  
  Consider the obvious circuit for deciding if the number of edges in an $n$-vertex graph is even.

- Any formula of *first-order logic* translates into a uniform family of *constant-depth, polynomial-size symmetric* Boolean circuits.

  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.*
Symmetric Circuits

• Any formula $\phi$ of FP translates into a uniform family of polynomial-size symmetric Boolean circuits.

For each $n$, $\phi$ translates into a first-order formula of depth polynomial in $n$ and with a constant bound $k$ on the number of free variables in a sub-formula.

• Any formula of FPC translates into a uniform family of polynomial-size symmetric threshold (or majority) circuits.
Symmetric Circuits

- There is trivially a polynomial-size family of symmetric circuits $C_n$ deciding whether $n$ is even.
- Is there a polynomial-size family of symmetric Boolean circuits deciding if an $n$ vertex graph has an even number of edges? No – as we shall see.
- Are polynomial-size families of uniform symmetric threshold circuits more powerful than Boolean circuits? Yes – follows from above.
- Can every invariant circuit be translated into an equivalent symmetric threshold circuit, with only polynomial blow-up? No – as we shall see.
Main Results

**Theorem**

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

**Theorem**

A class of graphs is accepted by a $\mathbf{P}$-uniform, polynomial-size, symmetric family of threshold circuits if, and only if, it is definable in $\mathbf{FPC}$. 
Some Consequences

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

Inexpressiblity results for FPC are also lower bound results against a natural circuit class.
Technical Tools – Support

For a symmetric circuit $C_n$ we assume \textit{w.l.o.g.} that the automorphism group is $S_n$ acting in the natural way.

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

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

Say a set $X \subseteq [n]$ \textit{supports} $g$ if

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

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

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

Anuj Dawar  November 2013
We want 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 gate $g$ in a symmetric circuit $C_n$, say that a partition $\mathcal{P}$ of $[n]$ supports $g$ if every permutation that fixes each $P \in \mathcal{P}$ also fixes $g$:

**Lemma:** There is a *coarsest* partition (denote it $\text{Sup}(g)$) 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.
Writing $\text{Stab}^\bullet(\text{Sup}(g))$ for the the group of permutations that fix each part in $\text{Sup}(g)$ and $\text{Stab}(\text{Sup}(g))$ for the group of permutations that fix the partition $\text{Sup}(g)$ \textit{setwise}, we have:

$$\text{Stab}^\bullet(\text{Sup}(g)) \subseteq \text{Stab}(g) \subseteq \text{Stab}(\text{Sup}(g))$$

The first inclusion is by definition. The second follows from the fact that for any permutation $\pi \in S_n$, $\text{Sup}(\pi(g)) = \pi(\text{Sup}(g))$.

By the \textit{orbit-stabilizer} theorem, the size of the orbit of $g$ is $\frac{n!}{|\text{Stab}(g)|}$. So, an upper bound on $\text{Stab}(g)$ gives us a lower bound on the \textit{orbit} of $g$.

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

Anuj Dawar
November 2013
Support Theorem

For a circuit $C$, $\text{Sup}(C)$ denotes the maximum over all gates $g$ in $C$ of the size of the union of all but the largest part in $\text{Sup}(g)$.

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

$$\text{Sup}(C) \leq \frac{33 \log s}{\epsilon \log n}.$$  

Corollary
Polynomial-size symmetric circuits have constant support.
Proof Sketch of Support Theorem – 1

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

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

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

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

so, $|\text{Orb}(g)| = \frac{n!}{|\text{Stab}(g)|} \geq \frac{n!}{|\text{Stab}(\text{Sup}(g))|} > s.$

Say that $\text{Sup}(g)$ is small if it has at most $\frac{8 \log s}{\epsilon \log n}$ parts and big otherwise.
Claim: If $\text{Sup}(g)$ is small then the largest part has size at least $n - \frac{33 \log s}{\epsilon \log n}$

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

\[
\frac{n!}{|\text{Stab}(\mathcal{P})|} > s
\]
Claim: \( \text{Sup}(g) \) is small.

Suppose that \( g \) is a minimal gate (in the DAG-order of the circuit) with \( \text{Sup}(g) \) large.

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

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 \( \text{Orb}(g) \) must be bigger than \( s \).
Translating Symmetric Circuits to Formulas

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

1. There are formulas of \text{FP} interpreted on $([n], <)$ that defines the structure $C_n$.
2. We can also compute in polynomial time, $\text{Orb}(g)$ and $\text{Sup}(g)$ for each gate $g$.
3. These can be turned into tuples of elements \textit{labelling} the gates, with some duplication.
4. With these as labels for gates, the edge relation of $C_n$ can be expressed in terms of just equality.
5. Evaluate circuit on unordered universe (in \text{FP} for a Boolean circuit, in \text{FPC} for one with threshold gates.)
FP with Rank Operators

FP_{rk} is fixed-point logic with an operator for \textit{matrix rank} over finite fields.

(D., Grohe, Holm, Laubner, 2009)

We have, as with FPC, terms of \textit{element sort} and \textit{numeric sort}.

We interpret \(\eta(x, y)\)—a term of numeric sort—in \(G = (V, E)\) as defining a matrix with rows and columns indexed by elements of \(V\) with entries \(\eta[a, b]\).

\(\text{rk}_{x, y}\eta\) is a term denoting the number that is the rank of the matrix defined by \(\eta(x, y)\).

To be precise, we have, for each finite field \(\mathbb{F}_q\) (\(q\) prime), an operator \(\text{rk}^q\) which defines the rank of the matrix with entries \(\eta[a, b](\text{mod} q)\).
Choiceless Polynomial Time with counting ($\tilde{\text{CPT}}(\text{Card})$) is a class of computational problems defined by (Blass, Gurevich and Shelah 1999).

It is based on a machine model (Gurevich Abstract State Machines) that works directly on a graph or relational structure (rather than on a string representation).

The machine can access the collection of hereditarily finite sets with the vertices of the graph as atoms, and can perform counting operations.

$\tilde{\text{CPT}}(\text{Card})$ is the polynomial time and space restriction of the machines.
Beyond FPC

FPrk can express the *CFI property* and solvability of systems of linear equations on finite fields. (D., Grohe, Holm, Laubner, 2009)

ČPT(Card) can express the *CFI property* (but requires sets of unbounded rank). (D., Richerby, Rossman, 2008)

The relationship between the two (and their relationship to P) remains open.
Big Picture

<table>
<thead>
<tr>
<th>Logic</th>
<th>Circuits</th>
</tr>
</thead>
<tbody>
<tr>
<td>FP on structures with a disjoint number sort ([n], &lt;).</td>
<td>Poly-size <em>symmetric</em> Boolean circuits.</td>
</tr>
<tr>
<td>Additional predicates on number sort.</td>
<td>Non-uniformity (of function ( n \mapsto C_n )).</td>
</tr>
<tr>
<td>Connections between element sort and number sort (FPC and FPrk).</td>
<td>Additional gates (<em>counting</em> and <em>rank</em>).</td>
</tr>
<tr>
<td>ĖPT(Card).</td>
<td>Breaking symmetry (how?).</td>
</tr>
</tbody>
</table>