Finite and Algorithmic Model Theory IV: Descriptive Complexity

Anuj Dawar

University of Cambridge Computer Laboratory

Simons Institute, 1 September 2016

Review

We have seen tools for establishing limits on the expressive power of :

first-order *sentences*; and

first-order theories with a *bounded number of variables.*

We have linked the expressive power of MSO to that of finite automata giving:

tools for studying the expressive power of MSO; and

tools for limiting the complexity of MSO definable properties on *decomposable* structures.

We analyzed the complexity of first-order logic on *sparse* classes of structures using *locality*.

Descriptive Complexity

Descriptive Complexity provides an alternative perspective on Computational Complexity.

Computational Complexity

- Measure use of resources (space, time, *etc.*) on a machine model of computation;
- Complexity of a language—i.e. a set of strings.

Descriptive Complexity

- Complexity of a class of structures—e.g. a collection of graphs.
- Measure the complexity of describing the collection in a formal logic, using resources such as variables, quantifiers, higher-order operators, *etc.*

There is a fascinating interplay between the views.

Encoding Structures

In order to talk of the complexity of a class of finite structures, we need to fix some way of representing finite structures as strings.

We use an alphabet $\Sigma = \{0, 1, \#\}$.
For a structure $\mathbb{A} = (A, R_1, \ldots, R_m)$, fix a linear order $<$ on $A = \{a_1, \ldots, a_n\}$.
R_i (of arity k) is encoded by a string $[R_i]_<$ of 0s and 1s of length n^k.

$$[\mathbb{A}]_< = \underbrace{1 \cdots 1}_{n} \# [R_1]_< \# \cdots \# [R_m]_<$$

The exact string obtained depends on the choice of order.

Invariance

Note that the decision problem:

Given a string $[\mathbb{A}]_<$ *decide whether* $\mathbb{A} \models \varphi$

has a natural invariance property.

It is invariant under the following equivalence relation

Write $w_1 \sim w_2$ *to denote that there is some structure* $\mathbb{A}$ *and orders* $<_1$ *and* $<_2$ *on its universe such that*

$$w_1 = [\mathbb{A}]_{<_1} \text{ and } w_2 = [\mathbb{A}]_{<_2}$$

Note: deciding the equivalence relation $\sim$ is just the same as deciding structure isomorphism.

First-Order Logic

For a first-order sentence φ, the *computational complexity* of the problem:

> *Given: a structure* $\mathbb{A}$
> *Decide: if* $\mathbb{A} \models \varphi$

is in *logarithmic space* and *polynomial time*.

There are computationally easy properties that are not definable in first-order logic.

- There is no sentence φ of first-order logic such that $\mathbb{A} \models \varphi$ if, and only if, $|A|$ is even.
- There is no formula $\varphi(E, x, y)$ that defines the transitive closure of a binary relation E.

All of these are definable in *second-order logic*

Examples

Evennness

This formula is true in a structure if, and only if, the size of the domain is even.

$$\begin{array}{ll}\exists B \exists S & \forall x \exists y B(x,y) \land \forall x \forall y \forall z B(x,y) \land B(x,z) \rightarrow y = z \\ & \forall x \forall y \forall z B(x,z) \land B(y,z) \rightarrow x = y \\ & \forall x \forall y S(x) \land B(x,y) \rightarrow \neg S(y) \\ & \forall x \forall y \neg S(x) \land B(x,y) \rightarrow S(y)\end{array}$$

Examples

Transitive Closure

Each of the following formulas is true of a pair of elements a, b in a structure if, and only if, there is an E-path from a to b.

$$\forall S\big(S(a) \land \forall x \forall y[S(x) \land E(x,y) \to S(y)] \to S(b)\big)$$

$$\begin{array}{ll} \exists P & \forall x \forall y\, P(x,y) \to E(x,y) \\ & \exists x P(a,x) \land \exists x P(x,b) \land \neg\exists x P(x,a) \land \neg\exists x P(b,x) \\ & \forall x \forall y (P(x,y) \to \forall z (P(x,z) \to y = z)) \\ & \forall x \forall y (P(x,y) \to \forall z (P(z,y) \to x = z)) \\ & \forall x ((x \neq a \land \exists y P(x,y)) \to \exists z P(z,x)) \\ & \forall x ((x \neq b \land \exists y P(y,x)) \to \exists z P(x,z)) \end{array}$$

Examples

3-Colourability

The following formula is true in a graph (V, E) if, and only if, it is 3-colourable.

$$
\begin{aligned}
\exists R \exists B \exists G \quad & \forall x(Rx \lor Bx \lor Gx) \land \\
& \forall x(\quad \neg(Rx \land Bx) \land \neg(Bx \land Gx) \land \neg(Rx \land Gx)) \land \\
& \forall x \forall y(Exy \to (\quad \neg(Rx \land Ry) \land \\
& \qquad\qquad\qquad\qquad \neg(Bx \land By) \land \\
& \qquad\qquad\qquad\qquad \neg(Gx \land Gy)))
\end{aligned}
$$

Fagin's Theorem

Theorem (Fagin)
A class $\mathcal{C}$ of finite structures is definable by a sentence of *existential second-order logic* if, and only if, it is decidable by a *nondeterminisitic machine* running in polynomial time.

$$\mathsf{ESO} = \mathsf{NP}$$

$S = \mathrm{Mod}(\varphi)$ for some φ in ESO *if, and only if,* $\{[\mathbb{A}]_{\leq} \mid \mathbb{A} \in S\}$ is in NP

Fagin's Theorem

If φ is $\exists R_1 \cdots \exists R_m \theta$ for a *first-order* θ.

To decide $\mathbb{A} \models \varphi$, *guess* an interpretation for the relations $R_1, \ldots, R_m$ and then evaluate θ in the expanded structure.

Given a *nondeterministic* machine M and a polynomial p:

> $\exists \leq$ *a linear order*
> $\exists H, T, S$ *that code an accepting computation of* M *of length* p *starting with* $[\mathbb{A}]_{\leq}$.

Is there a logic for P?

The major open question in *Descriptive Complexity* (first asked by Chandra and Harel in 1982) is whether there is a logic $\mathcal{L}$ such that

> *for any class of finite structures $\mathcal{C}$, $\mathcal{C}$ is definable by a sentence of $\mathcal{L}$ if, and only if, $\mathcal{C}$ is decidable by a deterministic machine running in polynomial time.*

Formally, we require $\mathcal{L}$ to be a *recursively enumerable* set of sentences, with a computable map taking each sentence to a Turing machine M and a polynomial time bound p such that (M, p) accepts a *class of structures*.

(Gurevich 1988)

Inductive Definitions

Let $\varphi(R, x_1, \ldots, x_k)$ be a first-order formula in the vocabulary $\sigma \cup \{R\}$
Associate an operator Φ on a given σ-structure $\mathbb{A}$:

$$\Phi(R^{\mathbb{A}}) = \{\mathbf{a} \mid (\mathbb{A}, R^{\mathbb{A}}, \mathbf{a}) \models \varphi(R, \mathbf{x})\}$$

We define the *non-decreasing* sequence of relations on $\mathbb{A}$:

$$\Phi^0 = \emptyset$$

$$\Phi^{m+1} = \Phi^m \cup \Phi(\Phi^m)$$

The *inflationary fixed point* of Φ is the limit of this sequence.
On a structure with n elements, the limit is reached after at most n^k stages.

FP

The logic FP is formed by closing first-order logic under the rule:

If φ *is a formula of vocabulary* $\sigma \cup \{R\}$ *then* $[\mathbf{ifp}_{R,\mathbf{x}}\varphi](\mathbf{t})$ *is a formula of vocabulary* σ.

The formula is read as:

the tuple $\mathbf{t}$ *is in the inflationary fixed point of the operator defined by* φ

LFP is the similar logic obtained using *least fixed points* of *monotone* operators defined by *positive* formulas.
LFP and FP have the same expressive power **(Gurevich-Shelah 1986; Kreutzer 2004)**.

Transitive Closure

The formula

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

defines the *transitive closure* of the relation E

The expressive power of FP properly extends that of first-order logic.

Still, every property definable in FP is decidable in *polynomial time*.

On a structure with n elements, the fixed-point of an induction of arity k is reached in at most n^k steps.

Immerman-Vardi Theorem

Theorem

On structures which come equipped with a linear order FP *expresses exactly the properties that are in* P*.*

(Immerman; Vardi 1982)

Recall from *Fagin's theorem*:

> $\exists \leq$ *a linear order*
> $\exists H, T, S$ *that code an accepting computation of* M *of length* p *starting with* $[\mathbb{A}]_{\leq}$.

FP vs. Ptime

The order cannot be built up inductively.
It is an open question whether a *canonical* string representation of a structure can be constructed in polynomial-time.

> *If it can, there is a logic for* P *(and also graph isomorphism is in* P*.*
> *If not, then* P $\neq \mathrm{NP}$*.*

All P classes of structures can be expressed by a sentence of FP with $<$, which is invariant under the choice of order. The set of all such sentences is not *r.e.*

FP by itself is too weak to express all properties in P.
Evenness is not definable in FP.

Finite Variable Logic

Recall that L^k is first order formulas using only the variables $x_1, \ldots, x_k$.

If $\varphi(R, \mathbf{x})$ has k variables all together, then each of the relations in the sequence:

$$\Phi^0 = \emptyset;\ \Phi^{m+1} = \Phi^m \cup \Phi(\Phi^m)$$

is definable in L^{2k}.

Proof by induction, using *substitution* and *renaming* of bound variables.

On structures of a fixed size n, $[\mathbf{ifp}_{R,\mathbf{x}}\varphi](\mathbf{t})$ is equivalent to a formula of L^{2k}.

For any sentence φ of FP there is a k such that the property defined by φ is invariant under $\equiv^k$.

Inexpressibility in FP

The following are not definable in FP:

- *Evenness*;
- *Perfect Matching*;
- *Hamiltonicity*.

The examples showing these inexpressibility results all involve some form of *counting*.

Fixed-point Logic with Counting

Immerman 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 *non-negative integers*.

If $\varphi(x)$ is a formula with free variable x, then $\#x\varphi$ is a *term* denoting the *number* of elements of $\mathbb{A}$ that satisfy φ.

We have arithmetic operations $(+, \times)$ on *number terms*.

Quantification over number variables is *bounded*: $(\exists x < t)\, \varphi$

Counting Quantifiers

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

- *counting quantifiers*: $\exists^i x\, \varphi$; 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 if $\mathbb{A} \equiv^{C^k} \mathbb{B}$, then

$$\mathbb{A} \models \varphi \quad \text{if, and only if,} \quad \mathbb{B} \models \varphi.$$

Limits of FPC

FPC was proposed by **Immerman** as a possible logic for *capturing* P:

It was proved **(Cai, Fürer, Immerman 1992)** that there are polynomial-time graph properties that are *not* expressible in FPC.

A number of other results about the limitations of FPC followed.

In particular, it has been shown that the problem of solving linear equations over the two element field $\mathbb{Z}_2$ is not definable in FPC.

(Atserias, Bulatov, D. 09)

The problem is clearly solvable in polynomial time by means of Gaussian elimination.

Systems of Linear Equations

We see how to represent systems of linear equations as unordered relational structures.

Consider structures over the domain $\{x_1, \ldots, x_n, e_1, \ldots, e_m\}$, (where $e_1, \ldots, e_m$ are the equations) with relations:

- unary E_0 for those equations e whose r.h.s. is 0.
- unary E_1 for those equations e whose r.h.s. is 1.
- binary M with $M(x, e)$ if x occurs on the l.h.s. of e.

$\mathsf{Solv}(\mathbb{Z}_2)$ is the class of structures representing solvable systems.

Undefinability in FPC

To show that the *satisfiability* of systems of equations is not definable in FPC it suffices to show that for each k, we can construct a two systems of equations

$$E_k \text{ and } F_k$$

such that:

- E_k is satisfiable;
- F_k is unsatisfiable; and
- $E_k \equiv^{C^k} F_k$

Constructing systems of equations

Take G a 4-regular, connected graph.
Define equations $\mathbf{E}_G$ with two variables x_0^e, x_1^e for each edge e.
For each vertex v with edges e_1, e_2, e_3, e_4 incident on it, we have 16 equations:

$$E_v: \qquad x_a^{e_1} + x_b^{e_2} + x_c^{e_3} + x_d^{e_4} \equiv a + b + c + d \pmod 2$$

$\tilde{\mathbf{E}}_G$ is obtained from $\mathbf{E}_G$ by replacing, for exactly one vertex v, E_v by:

$$E'_v: \qquad x_a^{e_1} + x_b^{e_2} + x_c^{e_3} + x_d^{e_4} \equiv a + b + c + d + 1 \pmod 2$$

We can show: $\mathbf{E}_G$ is satisfiable; $\tilde{\mathbf{E}}_G$ is unsatisfiable.

Satisfiability

Lemma $\mathbf{E}_G$ is satisfiable.

by setting the variables x_i^e to i.

Lemma $\tilde{\mathbf{E}}_G$ is unsatisfiable.

Consider the subsystem consisting of equations involving only the variables x_0^e.
The sum of all left-hand sides is

$$2\sum_e x_0^e \equiv 0 \pmod 2$$

However, the sum of right-hand sides is 1.

Now we show that, for each k, we can find a graph G such that $\mathbf{E}_G \equiv^{C^k} \tilde{\mathbf{E}}_G$.

Counting Game

Immerman and Lander (1990) defined a *pebble game* for C^k.
This is again played by *Spoiler* and *Duplicator* using k pairs of pebbles $\{(a_1, b_1), \ldots, (a_k, b_k)\}$ on a pair of structures $\mathbb{A}$ and $\mathbb{B}$

At each move, Spoiler picks i and a set of elements of one structure (say $X \subseteq B$)

Duplicator responds with a set of vertices of the other structure (say $Y \subseteq A$) of the same size.

Spoiler then places a_i on an element of Y and Duplicator must place b_i on an element of X.

Spoiler wins at any stage if the partial map from $\mathbb{A}$ to $\mathbb{B}$ defined by the pebble pairs is not a partial isomorphism

If Duplicator has a winning strategy for p moves, then $\mathbb{A}$ and $\mathbb{B}$ agree on all sentences of C^k of quantifier rank at most p.

Bijection Games

$\equiv^{C^k}$ is also characterised by a k-pebble *bijection game*. **(Hella 96)**.
The game is played on graphs $\mathbb{A}$ and $\mathbb{B}$ with pebbles $a_1, \ldots, a_k$ on $\mathbb{A}$ and $b_1, \ldots, b_k$ on $\mathbb{B}$.

- *Spoiler* chooses a pair of pebbles a_i and b_i;
- *Duplicator* chooses a bijection $h : A \to B$ such that for pebbles a_j and $b_j (j \neq i)$, $h(a_j) = b_j$;
- *Spoiler* chooses $a \in A$ and places a_i on a and b_i on $h(a)$.

Duplicator loses if the partial map $a_i \mapsto b_i$ is not a partial isomorphism.
Duplicator has a strategy to play forever if, and only if, $\mathbb{A} \equiv^{C^k} \mathbb{B}$.

Equivalence of Games

It is easy to see that a winning strategy for *Duplicator* in the bijection game yields a winning strategy in the counting game:

> *Respond to a set $X \subseteq A$ (or $Y \subseteq B$) with $h(X)$ ($h^{-1}(Y)$, respectively).*

For the other direction, consider the partition induced by the equivalence relation

$$\{(a, a') \mid (\mathbb{A}, \mathbf{a}[a/a_i]) \equiv^{C^k} (\mathbb{A}, \mathbf{a}[a'/a_i])\}$$

and for each of the parts X, take the response Y of *Duplicator* to a move where *Spoiler* would choose X.
Stitch these together to give the bijection h.

Cops and Robbers

A game played on an undirected graph $G = (V, E)$ between a player controlling k cops and another player in charge of a robber.

At any point, the cops are sitting on a set $X \subseteq V$ of the nodes and the robber on a node $r \in V$.
A move consists in the cop player removing some cops from $X' \subseteq X$ nodes and announcing a new position Y for them. The robber responds by moving along a path from r to some node s such that the path does not go through $X \setminus X'$.
The new position is $(X \setminus X') \cup Y$ and s. If a cop and the robber are on the same node, the robber is caught and the game ends.

Cops and Robbers on the Grid

If G is the $k \times k$ toroidal grid, than the *robber* has a winning strategy in the k-*cops and robbers* game played on G.

To show this, we note that for any set X of at most k vertices, the graph $G \setminus X$ contains a connected component with at least half the vertices of G.

If all vertices in X are in distinct rows then $G \setminus X$ is connected. Otherwise, $G \setminus X$ contains an entire row and column and in its connected component there are at least $k-1$ vertices from at least $k/2$ columns.

Robber's strategy is to stay in the large component.

Cops, Robbers and Treewidth

Actually, the cops and robbers game *characterizes tree-width*.

> *A connected graph G has tree-width $\geq k$ if, and only if, robber has a winning strategy against a team of k cops on G.*

Cops, Robbers and Bijections

Suppose G is such that the *robber* has a winning strategy in the k-*cops and robbers* game played on G.

We use this to construct a winning strategy for Duplicator in the k-pebble bijection game on $\mathbf{E}_G$ and $\tilde{\mathbf{E}}_G$.

- A bijection $h : \mathbf{E}_G \to \tilde{\mathbf{E}}_G$ is *good bar* v if it is an isomorphism everywhere except at the variables x_a^e for edges e incident on v.
- If h is good bar v and there is a path from v to u, then there is a bijection h' that is good bar u such that h and h' differ only at vertices corresponding to the path from v to u.
- Duplicator plays bijections that are good bar v, where v is the robber position in G when the cop position is given by the currently pebbled elements.

Restricted Graph Classes

If we restrict the class of structures we consider, FPC may be powerful enough to express all polynomial-time decidable properties.

1. FPC captures P on *trees*. **(Immerman and Lander 1990)**.
2. FPC captures P on any class of graphs of *bounded treewidth*. **(Grohe and Mariño 1999)**.
3. FPC captures P on the class of *planar graphs*. **(Grohe 1998)**.
4. FPC captures P on any *proper minor-closed class of graphs*. **(Grohe 2010)**.

In each case, the proof proceeds by showing that for any G in the class, a *canonical, ordered* representaton of G can be interpreted in G using FPC.

Beyond FPC

How do we define logics extending FPC while remaining inside P?

FPrk is fixed-point logic with an operator for *matrix rank* over finite fields.

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

Choiceless Polynomial Time with counting ($\tilde{\mathrm{C}}$PT(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).
$\tilde{\mathrm{C}}$PT(Card) is the polynomial time and space restriction of the machines.

Both of these have expressive power *strictly greater* than FPC.
Their relationship to each other and to P remains unknown.

We need new tools to analyze the *expressive power* of these logics.