Definability in Counting Logics

Anuj Dawar

University of Cambridge Computer Laboratory

Amsterdam, 21 June 2016

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.

First-Order Logic

Consider *first-order predicate logic*.

> *Fix a vocabulary σ of relation symbols $(R_1, \ldots, R_m)$ and a collection X of variables.*

The formulas are given by

$$R_i(\mathbf{x}) \mid x = y \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \neg\varphi \mid \exists x\varphi \mid \forall x\varphi$$

First-Order Logic

For a first-order sentence φ, we ask what is the *computational complexity* of the problem:

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

In other words, how complex can the collection of finite models of φ be?

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.

Encoding Structures

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.

Naïve Algorithm

The straightforward algorithm proceeds recursively on the structure of φ:

- Atomic formulas by direct lookup.
- Boolean connectives are easy.
- If $\varphi \equiv \exists x\, \psi$ then for each $a \in \mathbb{A}$ check whether

$$(\mathbb{A}, c \mapsto a) \models \psi[c/x],$$

  where c is a new constant symbol.

This runs in time $O(ln^m)$ and $O(m \log n)$ space, where l is the length of φ and m is the nesting depth of quantifiers in φ.

$$\mathrm{Mod}(\varphi) = \{\mathbb{A} \mid \mathbb{A} \models \varphi\}$$

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

Second-Order Logic

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.

Consider second-order logic, extending first-order logic with *relational quantifiers* — $\exists X \varphi$

Examples

Evennness

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

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

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.

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

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.

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

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

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

defines the *transitive closure* of the relation E

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

Theorem

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

(Immerman; Vardi 1982)

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*.*
> *If not, then* $\mathrm{P} \neq \mathrm{NP}$*.*

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

We write L^k for the first order formulas using only the variables $x_1, \ldots, x_k$.

$$(\mathbb{A}, \mathbf{a}) \equiv^k (\mathbb{B}, \mathbf{b})$$

denotes that there is no formula φ of L^k such that $\mathbb{A} \models \varphi[\mathbf{a}]$ and $\mathbb{B} \not\models \varphi[\mathbf{b}]$

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.

Pebble Game

The k-pebble game is played on two structures $\mathbb{A}$ and $\mathbb{B}$, by two players—*Spoiler* and *Duplicator*—using k pairs of pebbles $\{(a_1, b_1), \ldots, (a_k, b_k)\}$.

> *Spoiler moves by picking a pebble and placing it on an element (a_i on an element of $\mathbb{A}$ or b_i on an element of $\mathbb{B}$).*
>
> *Duplicator responds by picking the matching pebble and placing it on an element of the other structure*
>
> *Spoiler wins at any stage if the partial map from $\mathbb{A}$ to $\mathbb{B}$ definedby the pebble pairs is not a partial isomorphism*
>
> *If Duplicator has a winning strategy for q moves, then $\mathbb{A}$ and $\mathbb{B}$ agree on all sentences of L^k of quantifier rank at most q.*
>
> **(Barwise)**

$\mathbb{A} \equiv^k \mathbb{B}$ if, for every q, *Duplicator* wins the q round, k pebble game on $\mathbb{A}$ and $\mathbb{B}$. Equivalently (on finite structures) *Duplicator* has a strategy to play forever.

Evenness

To show that *Evenness* is not definable in FP, it suffices to show that:

> *for every k, there are structures $\mathbb{A}_k$ and $\mathbb{B}_k$ such that $\mathbb{A}_k$ has an even number of elements, $\mathbb{B}_k$ has an odd number of elements and*
>
> $$\mathbb{A} \equiv^k \mathbb{B}.$$

It is easily seen that *Duplicator* has a strategy to play forever when one structure is a set containing k elements (and no other relations) and the other structure has $k+1$ elements.

Matching

In a *graph* $G = (V, E)$ a matching $M \subset E$ is a set of edges such that each vertex is incident on *at most* one edge in M.

A *perfect matching* is a matching M such that each vertex is incident on *exactly* one edge in M

$$\begin{array}{ll} \exists M & \forall x, y[M(x,y) \rightarrow E(x,y)] \wedge \\ & \forall x, y, z[M(x,y) \wedge M(x,z) \rightarrow y = z] \wedge \\ & \forall x \exists y\, M(x,y) \end{array}$$

A classical result of **(Edmonds, 1965)** tells us that the property of having a perfect matching is in P.

Matching

Take $K_{k,k}$—the complete bipartite graph on two sets of k vertices.
and $K_{k,k+1}$—the complete bipartite graph on two sets, one of k vertices, the other of $k+1$.

These two graphs are $\equiv^k$ equivalent, yet one has a perfect matching, and the other does not.

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:

- 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 $\mathcal{G}$ a 3-regular, connected graph.
Define equations $\mathbf{E}_{\mathcal{G}}$ 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 incident on it, we have eight equations:

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

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

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

We can show: $\mathbf{E}_{\mathcal{G}}$ is satisfiable; $\tilde{\mathbf{E}}_{\mathcal{G}}$ 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 $\mathbf{E}_{\mathcal{G}} \equiv^{C^k} \tilde{\mathbf{E}}_{\mathcal{G}}$.