Complexity and the Expressive Power of Logics

Anuj Dawar

University of Cambridge

CUSPOMMS, 9 March 2007

Mathematical Logic

Mathematical logic seeks to formalise the process of mathematical reasoning and turn this process itself into a subject of mathematical enquiry.

It investigates the relationships among:

- Structure
- Language
- Proof

Proof-theoretic vs. *Model-theoretic* views of logic.

Computation as Logic

If logic aims to reduce reasoning to symbol manipulation,

> *On the one hand, computation theory provides a formalisation of "symbol manipulation".*
>
> *On the other hand, the development of computing machines leads to "logic engineering".*

The validities of first-order logic are r.e.-complete.

Proof Theory in Computation

As all programs and data are strings of symbols in a formal system, one view sees all computation as inference.

For instance, the functional programming view:

- Propositions are types.
- Programs are (constructive) proofs.
- Computation is proof transformation.

Model Theory in Computation

A model-theoretic view of computation aims to distinguish computational *structures* and languages used to talk about them.

Data Structure	Programming Language
Database	Query Language
Program/State Space	Specification Language

The structures involved are rather different from those studied in classical model theory. *Finite Model Theory.*

First-Order Logic

terms – $c, x, f(t_1, \ldots, t_a)$

atomic formulae – $R(t_1, \ldots, t_a), t_1 = t_2$

boolean operations – $\varphi \wedge \psi$, $\varphi \vee \psi$, $\neg\varphi$

first-order quantifiers – $\exists x\varphi$, $\forall x\varphi$

Formulae are interpreted in structures:

$$\mathbb{A} = (A, R_1, \ldots, R_m, f_1, \ldots, f_n, c_1, \ldots, c_n)$$

Success of First-Order Logic

First-order logic is very successful at its intended purpose, the formalisation of mathematics.

Many natural mathematical theories can be expressed as first-order theories.

These include *set theory*, fundamental to the foundations of mathematics.

Gödel's completeness theorem guarantees that the consequences of these theories can be effectively obtained.

Finite Structures

The completeness theorem fails when restricted to finite structures.

> The sentences of first-order logic, valid on finite structures are not recursively enumerable.
>
> (Trakhtenbrot 1950)

On finite structures, first-order logic is both too strong and too weak.

First-Order Logic is too Strong

For every finite structure $\mathbb{A}$, there is a sentence $\varphi_{\mathbb{A}}$ such that

$$\mathbb{B} \models \varphi_{\mathbb{A}} \quad \text{if, and only if,} \quad \mathbb{B} \cong \mathbb{A}$$

For any isomorphism-closed class of finite structures, there is a first-order theory that defines it.

First-Order Logic is too Weak

For any first-order sentence φ, its class of finite models

$$\mathrm{Mod}_{\mathcal{F}}(\varphi) = \{\mathbb{A} \mid \mathbb{A} \text{ finite, and } \mathbb{A} \models \varphi\}$$

is trivially decidable (in LOGSPACE).

There are computationally easy classes that are not defined by any first-order sentence.

- The class of sets with an even number of elements.
- The class of graphs (V, E) that are connected.

Computational Complexity

P—the classes of finite structures for which membership can be decided in polynomial time.

These are often identified as the *feasibly computable* problems.

NP—the classes of finite structures for which there are "membership certificates" that can be verified in polynomial time.

The central problem in computational complexity theory (and one of the more prominent open problems in all of mathematics) is to determine whether $\mathrm{P} = \mathrm{NP}$.

Second-Order Logic

Second-order logic is obtained by adding to the defining rules of first-order logic two further clauses:

atomic formulae – $X(t_1, \ldots, t_a)$, where X is a *second-order variable*

second-order quantifiers – $\exists X \varphi$, $\forall X \varphi$

Second-order logic can express evenness and connectivity as well as properties that are deemed not to be feasibly computable, such as *graph 3-colourability*.

Indeed, it can express every *NP-complete* problem.

Examples

Evenness.

$$\exists B \exists S \quad \forall x \exists y B(x,y) \wedge \forall x \forall y \forall z B(x,y) \wedge B(x,z) \rightarrow y = z$$

$$\forall x \forall y \forall z B(x,z) \wedge B(y,z) \rightarrow x = y$$

$$\forall x \forall y S(x) \wedge B(x,y) \rightarrow \neg S(y)$$

$$\forall x \forall y \neg S(x) \wedge B(x,y) \rightarrow S(y)$$

Examples

3-Colourability

$$\exists R \exists B \exists G \quad \forall x(Rx \vee Bx \vee Gx) \wedge$$
$$\forall x(\quad \neg(Rx \wedge Bx) \wedge \neg(Bx \wedge Gx) \wedge \neg(Rx \wedge Gx)) \wedge$$
$$\forall x \forall y(Exy \rightarrow (\quad \neg(Rx \wedge Ry) \wedge$$
$$\neg(Bx \wedge By) \wedge$$
$$\neg(Gx \wedge Gy)))$$

Descriptive Complexity

Fagin's Theorem:

A class of finite structures is definable in existential second-order logic if, and only if, it is in the class *NP*.

A major open problem in the field of *Descriptive Complexity* has been to establish whether there is a descriptive characterisation of *P*—the class of computational problems decidable in polynomial time.

Is there any extension of first-order logic in which one can express all and only the feasibly computable problems?

Can the class *P* be "build up from below" by finitely many operations?

Inductive Definitions

In computing (and logic), many classes of structures are naturally defined *inductively*.

viz. The definition of the terms and formulae of first-order logic.

Includes definitions of syntax and semantics of most *languages*, of *data structures* (trees, lists, etc.), of *arithmetic functions*.

Definition by Fixed Point

The collection of first-order terms can be defined as *the least set* containing all constants, all variables and such that $f(t_1, \ldots, t_a)$ is a term whenever $t_1, \ldots, t_a$ are terms and f is a function symbol of arity a.

The addition function is defined as the least function satisfying:

$$\begin{aligned} x + 0 &= x \\ x + s(y) &= s(x + y). \end{aligned}$$

In each case, the set defined is the least fixed point of a monotone operator on sets.

From Metalanguage to Language

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

> If φ is a formula, *positive* in the relational variable R, then so is
>
> $$[\mathbf{lfp}_{R,\mathbf{x}}\varphi](\mathbf{t}).$$

The formula is read as:

> the tuple $\mathbf{t}$ is in the least fixed point of the operator that maps R to $\varphi(R, \mathbf{x})$.

Connectivity

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.

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

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

(Immerman; Vardi)

Fixed-point Logic with Counting

LFP $+$ C is a logic formulated to add the ability to count to LFP.

If $\varphi(x)$ is a formula with free variable x, then $\#x\varphi$ is a term denoting the number of elements satisfying φ.

Formulae of LFP $+$ C:

- all atomic formulae as in LFP;
- $\tau_1 < \tau_2$; $\tau_1 = \tau_2$ where τ_i is a term of numeric sort;
- $\exists x\, \varphi$; $\exists \nu\, \varphi$; where ν is a variable ranging over numbers up to the size of the domain;
- $[\mathbf{lfp}_{X,\mathbf{x},\nu}\varphi](\mathbf{t})$; and
- $\varphi \wedge \psi$; $\neg\varphi$.

Evenness

There are an even number of elements satisfying $\varphi(x)$.

$$\exists \nu_1 \exists \nu_2 (\nu_1 = [\#x\varphi] \wedge (\nu_2 + \nu_2 = \nu_1))$$

Further Operations

Cai, Fürer and Immerman (1992) showed that LFP $+$ C is not powerful enough to express all properties in *P*.

The proof involved a contrived construction of a class of graphs on which the graph isomorphism problems is solvable in polynomial time but not definable in LFP $+$ C.

More recently, we have exhibited natural feasibly computable problems that are not definable in LFP $+$ C, such as computing the rank of a 0-1 matrix.

(Atserias, Bulatov, D.)

Extensions of LFP + C

A number of extensions of LFP + C have been proposed:

- *choice operators*—involves a delicate trade-off to preserve symmetry and feasibility.
- *higher-order sets*—a model of "choiceless polynomial time" machines proposed by Blass, Gurevich and Shelah.
- *algebraic operators*—based on matrix computations.

It remains a challenge to show that these do not suffice to capture all of *P*.

In Summary

- Model-theoretic methods concerned with studying the expressive power of logical languages.
- First-order logic does not occupy a central place in finite model theory.
- A variety of fixed-extensions of first-order logic used to study computational complexity.
- A particularly significant question is whether there is such an extension exactly capturing *P*.
- Leads to the consideration of unusual *logical operators* whose expressive power is being studied.