Topics in Logic and Complexity
Handout 2

Anuj Dawar

MPhil Advanced Computer Science, Lent 2012

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)

Enumerating Queries

For a given structure $\mathbb{A}$ with n elements, there may be as many as $n!$ distinct strings $[\mathbb{A}]_<$ encoding it.

Given $(M_0, p_0), \ldots, (M_i, p_i), \ldots$—an enumeration of polynomially-clocked Turing machines.

Can we enumerate a subsequence of those that compute graph properties, i.e. are *encoding invariant*, while including all such properties?

Recursive Indexability

We say that P is *recursively indexable*, if there is a recursive set $\mathcal{I}$ and a Turing machine M such that:

- on input $i \in \mathcal{I}$, M produces the code for a machine $M(i)$ and a polynomial p_i
- $M(i)$, accepts a class of structures in P.
- $M(i)$ runs in time bounded by p_i
- for each class of structures $C \in$ P, there is an i such that $M(i)$ accepts C.

Canonical Labelling

We say that a machine M *canonically labels* graphs, if

- on any input $[G]_<$, the output of M is $[G]_{<'}$ for some ordering $<'$; and
- if $[G]_{<_1}$ and $[G]_{<_2}$ are two encodings of the same graph, then $M([G]_{<_1}) = M([G]_{<_2})$.

It is an open question whether such a polynomial-time machine exists.

If so, then P is recursively indexable, by enumerating machines $M \to M_i$.

If not, $\mathsf{P} \neq \mathsf{NP}$.

Interpretations

Given two relational signatures σ and τ, where $\tau = \langle R_1, \ldots, R_r \rangle$, and arity of R_i is n_i

A *first-order interpretation of τ in σ* is a sequence:

$$\langle \pi_U, \pi_1, \ldots, \pi_r \rangle$$

of first-order σ-formulas, such that, for some k,:

- the free variables of π_U are among $x_1, \ldots, x_k$,
- and the free variables of π_i (for each i) are among $x_1, \ldots, x_{k \cdot n_i}$.

k is the width of the interpretation.

Interpretations II

An interpretation of τ in σ maps σ-structures to τ-structures.

If $\mathbb{A}$ is a σ-structure with universe A, then

$\pi(\mathbb{A})$ is a structure $(B, R_1, \ldots, R_r)$ with

- $B \subseteq A^k$ is the relation defined by π_U.
- for each i, R_i is the relation on B defined by π_i.

Reductions

Given:

- C_1 – a class of structures over σ; and
- C_2 – a class of structures over τ

π is a *first-order reduction* of C_1 to C_2 if, and only if,

$$\mathbb{A} \in C_1 \Leftrightarrow \pi(\mathbb{A}) \in C_2.$$

If such a π exists, we say that C_1 is first-order reducible to C_2.

NP-complete Problems

First-order reductions are, in general, much weaker than *polynomial-time reductions*.

Still, there are NP-complete problems under such reductions.

> Every problem in NP is first-order reducible to *SAT*
> **(Lovàsz and Gàcs 1977)**

> *CNF-SAT*, *Hamiltonicity* and *Clique* are NP-complete via first-order reductions
> **(Dahlhaus 1984)**

But, *3-colourability* is not NP-complete via first-order reductions.
(D.-Grädel 1995)

and the question is open for *3SAT*.

CNF-SAT

We formulate the problem *CNF-SAT* (of deciding whether a Boolean formula in *CNF* is satisfiable) as a class of structures.

Universe $V \cup C$ where V is the set of variables and C the set of clauses.

Unary Relation V for the set of variables

Binary Relations $P(v, c)$ to indicate that variable v occurs positively in c and $N(v, c)$ to indicate that v occurs negatively in c.

NP-completeness

Consider any ESO sentence ϕ. It can be transformed (by Skolemization) to a sentence

$$\exists R_1 \cdots \exists R_k \, \exists F_1 \cdots \exists F_l (\bigwedge_{i=1}^{l} \forall \mathbf{x}_i \exists y \, F_i(\mathbf{x}_i, y)) \wedge \forall \mathbf{y} \, \theta$$

where θ is quantifier-free (in *CNF*).

Now, given a finite structure $\mathbb{A}$, we construct a *CNF* Boolean formula $\phi_{\mathbb{A}}$ which is satisfiable if, and only if,

$$\mathbb{A} \models \phi.$$

Boolean Formula

The formula $\phi_{\mathbb{A}}$ contains variables $R_i^{\mathbf{a}}$ and $F_j^{\mathbf{a}}$ for every $1 \leq i \leq k$, every $1 \leq j \leq l$ and every tuple $\mathbf{a}$ of the appropriate length.

$$(\bigwedge_{i=1}^{l} \bigwedge_{\mathbf{a}} \bigvee_{a} F_i^{\mathbf{a}a}) \wedge \bigwedge_{\mathbf{a}} \theta^{\mathbf{a}}$$

The translation $\mathbb{A} \mapsto \phi_{\mathbb{A}}$ can be given by a first-order interpretation.

P-complete Problems

If there is any problem that is complete for P with respect to first-order reductions, then there is a logic for P.

If Q is such a problem, we form, for each k, a quantifier Q^k.

The sentence

$$Q^k(\pi_U, \pi_1, \ldots, \pi_s)$$

for a k-ary interpretation $\pi = (\pi_U, \pi_1, \ldots, \pi_s)$ is defined to be true on a structure $\mathbb{A}$ just in case

$$\pi(\mathbb{A}) \in Q.$$

The collection of such sentences is then a logic for P.

Conversely,

Theorem
If the polynomial time properties of graphs are recursively indexable, there is a problem complete for P under first-order reductions.

(D. 1995)

Proof Idea:

Given a recursive indexing $((M_i, p_i) | i \in \omega)$ of P

Encode the following problem into a class of finite structures:

$$\{(i, x) | M_i \text{ accepts } x \text{ in time bounded by } p_i(|x|)\}$$

To ensure that this problem is still in P, we need to pad the input to have length $p_i(|x|)$.

Constructing the Complete Problem

Suppose M is a machine which on input $i \in \omega$ gives a pair (M_i, p_i) as in the definition of recursive indexing. Let g a recursive bound on the running time of M.

Q is a class of structures over the signature $(V, E, \preceq, I)$.

$\mathbb{A} = (A, V, E, \preceq, I)$ is in Q if, and only if,

1. $\preceq$ is a linear pre-order on A;
2. if $a, b \in I$, $a \preceq b$ and $b \preceq a$, i.e. I picks out one equivalence class from the pre-order (say the i^{th});
3. $|A| \geq p_i(|V|)$;
4. the graph (V, E) is accepted by M_i; and
5. $g(i) \leq |A|$.

Finite Variable Logic

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

$$\mathbb{A} \equiv^k \mathbb{B}$$

denotes that $\mathbb{A}$ and $\mathbb{B}$ agree on all sentences of L^k.

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

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

For a tuple $\mathbf{a}$ in $\mathbb{A}$, $\mathsf{Type}^k(\mathbb{A}, \mathbf{a})$ denotes the collection of all formulas $\phi \in L^k$ such that $\mathbb{A} \models \phi[\mathbf{a}]$.

Finite Variable Logic

For any k,

$$\mathbb{A} \equiv^k \mathbb{B} \quad \Rightarrow \quad \mathbb{A} \equiv_k \mathbb{B}$$

However, for any q, there are $\mathbb{A}$ and $\mathbb{B}$ such that

$$\mathbb{A} \equiv_q \mathbb{B} \quad \text{and} \quad \mathbb{A} \not\equiv^2 \mathbb{B}.$$

Take $\mathbb{A}$ and $\mathbb{B}$ to be linear orders longer than 2^q.

Stages

For every formula ϕ of LFP, there is a k such that the query defined by ϕ is closed under $\equiv^k$.

Consider a formula $\psi(R, \mathbf{x})$ defining an operator.

Let the variables occurring in ψ be $x_1, \ldots, x_k$, with $\mathbf{x} = (x_1, \ldots, x_l)$, and $y_1, \ldots, y_l$ be new.

Stages

Define, by induction, the formulas ψ^m.

$$\psi^0 = \exists x\, x \neq x$$

ψ^{m+1} is obtained from $\psi(R, \mathbf{x})$ by replacing all sub-formulas $R(t_1, \ldots, t_l)$ with

$$\exists y_1 \ldots \exists y_l \,(\bigwedge_{1 \leq i \leq l} y_i = t_i) \wedge \phi^m(\mathbf{y})$$

Note that each ψ^m has at most $k+1$ variables.

LFP

If $(\mathbb{A}, \mathbf{a}) \equiv^{k+l} (\mathbb{B}, \mathbf{b})$, then *for all* m:

$$\mathbb{A} \models \psi^m[\mathbf{a}] \quad \text{if, and only if,} \quad \mathbb{B} \models \psi^m[\mathbf{b}].$$

So, $(\mathbb{A}, \mathbf{a})$ and $(\mathbb{B}, \mathbf{b})$ are not distinguished by $\mathbf{lfp}_{R,\mathbf{x}}\psi$.

Pebble Games

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 anelement 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)**

Using Pebble Games

To show that a class of structures S is not definable in first-order logic:

$$\forall k\ \forall q\ \exists \mathbb{A}, \mathbb{B}\ (\mathbb{A} \in S \wedge \mathbb{B} \notin S \wedge \mathbb{A} \equiv_q^k \mathbb{B})$$

Since $\mathbb{A} \equiv_q^q \mathbb{B} \Rightarrow \mathbb{A} \equiv_q \mathbb{B}$, we can ignore the parameter k

To show that S is not closed under any $\equiv^k$ (and hence not definable in LFP):

$$\forall k\ \exists \mathbb{A}, \mathbb{B}\ \forall q\ (\mathbb{A} \in S \wedge \mathbb{B} \notin S \wedge \mathbb{A} \equiv_q^k \mathbb{B})$$

If $\mathbb{A} \equiv_q^k \mathbb{B}$ holds for all q, then *Duplicator* actually wins an *infinite* game. That is, she has a strategy to play forever.

Evenness

To show that *Evenness* is not definable in LFP, 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

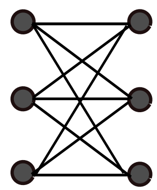
$$\mathbb{A} \equiv^k \mathbb{B}.$$

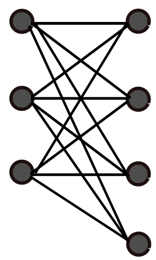
It is easily seen that *Duplicator* has a strategy to play forever when one structure is a set containing k elements (and no other relations) and the other structure has $k + 1$ elements.

Hamiltonicity

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 Hamiltonian cycle, and the other does not.