Topics in Logic and Complexity

Handout 5

Anuj Dawar

http://www.cl.cam.ac.uk/teaching/2324/L15

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, P $\neq$ 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

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

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. $g(i) \leq |A|$.

Fixed-point Logic with Counting

Immerman proposed FPC—the extension of IFP 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 $\phi(x)$ is a formula with free variable x, then $\#x\phi$ 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*: $(\exists x < t)\,\phi$

Evenness

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

$$\exists \nu < \#x\phi(\nu + \nu = \#x\phi)$$

Counting Quantifiers

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

- allowing *counting quantifiers*: $\exists^i x\,\phi$; 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

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

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 $\{(a_1, b_1), \ldots, (a_k, b_k)\}$.

Spoiler picks a subset of the universe (say $X \subseteq B$)

Duplicator responds with $Y \subseteq A$ such that $|X| = |Y|$.

Spoiler then places a b_i pebble on an element of Y and Duplicator must place a_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 q moves, then $\mathbb{A}$ and $\mathbb{B}$ agree on all sentences of C^k of quantifier rank at most q.

Bijection Games

$\equiv^{C^k}$ is also characterised by a k-pebble *bijection game*. **(Hella 96)**.
The game is played on structures $\mathbb{A}$ 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

To show that the games do, indeed, capture $\equiv^{C^k}$, we can show the following series of implications for any structures $\mathbb{A}, \mathbb{B}$ and k-tuples of elements $\mathbf{a}$, $\mathbf{b}$.

1\. $\Rightarrow$ 2. $\Rightarrow$ 3. $\Rightarrow$ 1.

1. $(\mathbb{A}, \mathbf{a}) \not\equiv^{C^k} (\mathbb{B}, \mathbf{b})$
2. *Spoiler* wins the k-pebble counting game starting from $(\mathbb{A}, \mathbf{a})$ and $(\mathbb{B}, \mathbf{b})$.
3. *Spoiler* wins the k-pebble bijection game starting from $(\mathbb{A}, \mathbf{a})$ and $(\mathbb{B}, \mathbf{b})$.

Equivalence of Games

For 1. $\Rightarrow$ 2., from a sentence $\phi \in C^k$ such that

$$\mathbb{A} \models \phi \quad \text{and} \quad \mathbb{B} \not\models \phi$$

construct a winning strategy for *Spoiler* on $\mathbb{A}$ and $\mathbb{B}$.
If ϕ is $\exists^i x\theta$, choose a set X of i elements in $\mathbb{A}$ such that for all $a \in X$:

$$\mathbb{A} \models \theta[a]$$

In *Duplicator* response Y in $\mathbb{B}$, there must be b such that:

$$\mathbb{B} \not\models \theta[b]$$

Equivalence of Games

For 2. $\Rightarrow$ 3., we can show that a winning strategy for *Duplicator* in the bijection game yields a winning strategy in the counting game:

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

Equivalence of Games

For 3. $\Rightarrow$ 1., we show that if $(\mathbb{A}, \mathbf{a}) \equiv^{C^k} (\mathbb{B}, \mathbf{b})$, then *Duplicator* has a winning strategy in the bijection game starting from the position $\mathbf{a}$ and $\mathbf{b}$.

Consider the partition on A 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 the corresponding partition of B.

The condition $(\mathbb{A}, \mathbf{a}) \equiv^{C^k} (\mathbb{B}, \mathbf{b})$ guarantees that the corresponding parts have the same numbers of elements.

Stitch these together to give the bijection h.

Solvability of Linear Equations

We can now use the games to show that some natural problems in P are not definabile in FPC.
We consider the problem of solving linear equations over the two element field $\mathbb{Z}_2$.

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

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

Systems of Linear Equations

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.

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

Constructing systems of equations

Take G a 4-regular, connected graph.
Define equations 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:

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

$\tilde{\mathrm{E}}_G$ is obtained from 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: E_G is satisfiable; $\tilde{\mathrm{E}}_G$ is unsatisfiable.

Satisfiability

Lemma E_G is satisfiable.

by setting the variables x_i^e *to* i*.*

Lemma $\tilde{\mathrm{E}}_G$ 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 $\mathrm{E}_G \equiv^{C^k} \tilde{\mathrm{E}}_G$.

Toroidal Grids

We aim to show that if G is *sufficiently connected*, then $\mathsf{E}_G \equiv^{C^k} \tilde{\mathsf{E}}_G$.

The graph we choose is the $k \times k$ *toroidal grid*.

This has vertex set

$$V = \{(i,j) \mid 0 \leq i,j \leq k-1\}$$

and edges $((i,j),(i',j'))$ whenever

either $i = i'$ *and* $j' = j + 1 \bmod k$

or $j = j'$ *and* $i' = i + 1 \bmod k$

Cops and Robbers

The *cops and robbers* game is a way of measuring the connectivity of a graph.

> *It is 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 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 Bijections

Suppose G is such that the *robber* has a winning strategy in the $2k$-*cops and robbers* game played on G.

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

- A bijection $h : \mathrm{E}_G \to \tilde{\mathrm{E}}_G$ 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.