Finite and Algorithmic Model Theory V: Logic and Combinatorial Optimization

Anuj Dawar

University of Cambridge Computer Laboratory

Simons Institute, 2 September 2016

Review

We have developed tools for analyzing the *expressive power* of logics over *finite structures*.

We used these to investigate *logics for polynomial time*.

The logic FPC is a *powerful* and *natural* fragment of P, but it is *not* all of P.

In particular, it cannot express the solvability of *systems of linear equations* over a finite field.

Linear Algebra over Finite Fields

Linear Algebra is a testing ground for exploring the boundary of the expressive power of FPC.

Over the finite field $\mathbb{F}_q$, *matrix multiplication*; *non-singularity* of matrices; the *inverse* of a matrix; are all definable in FPC.

determinants and more generally, the coefficients of the *characteristic polynomial* can be expressed FPC.

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

solvability of systems of equations is *undefinable*.

the *rank* of a matrix is *undefinable*.

Linear Algebra over the Rational Field

Over the rational field $\mathbb{Q}$, we can also define *matrix multiplication*; *non-singularity* of matrices; the *inverse* of a matrix in FPC.

Moreover, we can also define the coefficients of the *characteristic polynomial*

and, we can define the *rank* of a matrix and the *solvability* of systems of equations.

(Holm 2010)

The last result also follows from the stronger result that *optimization of linear programs* is expressible in FPC.

(Anderson, D., Holm 2015)

Representing Rational Numbers

We can take the rational number

$$q = s\frac{n}{d}$$

where $s \in \{1, -1\}$ and $n, d \in \mathbb{N}$
to be given by a structure

$$(B, <, S, N, D)$$

where $<$ is a linear order on the domain B and S, N and D are unary relations.

$S = \emptyset$ *iff* $s = 1$ and N and D code the binary representation of n and d.

Since the domain is ordered, it is straightforward to see that arithmetic, in the form of addition and multiplication of numbers is definable in FPC.

Representing Rational Vectors and Matrices

A *rational vector* indexed by a set I:

$$v : I \to \mathbb{Q}$$

is represented by a structure over domain $I \cup B$ with relations:

- $<$ an order on B;
- $S, N, D \subseteq I \times B$

Similarly, a *rational matrix* $M \in \mathbb{Q}^{I \times J}$ is given by a structure over domain $I \cup J \cup B$ with relations:

- $<$ an order on B;
- $S, N, D \subseteq I \times J \times B$

Weighted Graphs

We use a similar encoding to represent problems over *weighted graphs* where the weights may be integer or rational.

For example, a graph with vertex set V with *non-negative rational* weights might be considered as a relational structure over universe $V \cup B$ where B is bigger than the number of bits required to represent any of the rational weights and we have

- $<$ an order on B;
- *weight relations* $W_n, W_d \subseteq V \times V \times B$

Linear Programming

Linear Programming is an important algorithmic tool for solving a large variety of optimization problems.

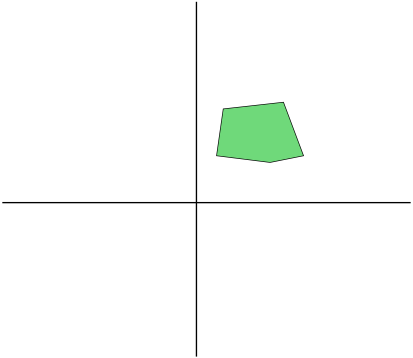
It was shown by **(Khachiyan 1980)** that linear programming problems can be solved in polynomial time.
We have a set C of *constraints* over a set V of *variables*.
Each $c \in C$ consists of $a_c \in \mathbb{Q}^V$ and $b_c \in \mathbb{Q}$.

Feasibility Problem: Given a linear programming instance, determine if there is an $x \in \mathbb{Q}^V$ such that:

$$a_c^T x \leq b_c \quad \text{for all } c \in C$$

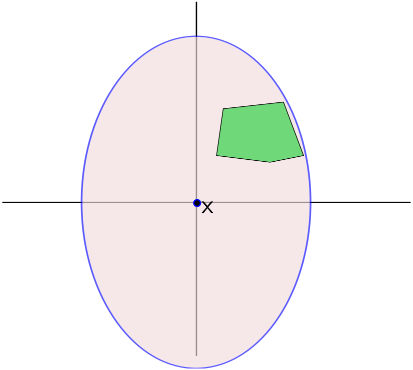
In **Anderson, D., Holm (2013)** we show that this, and the corresponding *optimization problem* are expressible in FPC.

Ellipsoid Method



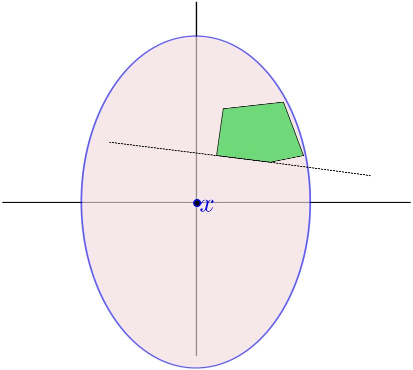
The set of constraints determines a *polytope*

Ellipsoid Method



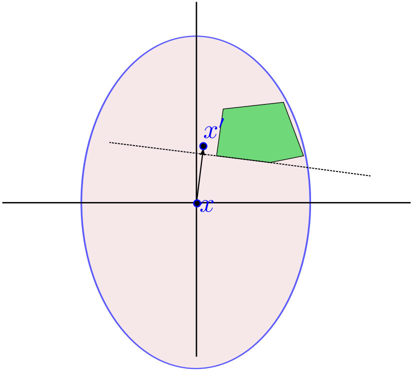
Start at the origin and calculate an *ellipsoid* enclosing it.

Ellipsoid Method



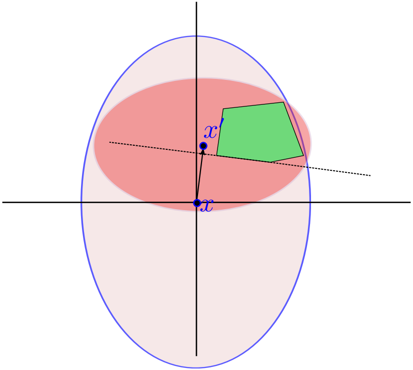
If the centre is not in the polytope, choose a constraint it *violates*.

Ellipsoid Method



Calculate a new *centre*.

Ellipsoid Method



And a new ellipsoid around the centre of at most *half* the volume.

Ellipsoid Method in FPC

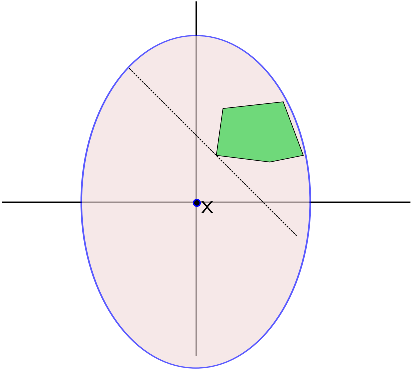
We can encode all the calculations involved in FPC.

This relies on expressing algebraic manipulations of *unordered* matrices.

What is not obvious is how to *choose* the violated constraint on which to project.

However, the ellipsoid method works as long as we can find, at each step, some *separating hyperplane*.

Ellipsoid Method in FPC



Ellipsoid Method in FPC

We can encode all the calculations involved in FPC.

This relies on expressing algebraic manilpulations of *unordered* matrices.

What is not obvious is how to *choose* the violated constraint on which to project.

However, the ellipsoid method works as long as we can find, at each step, some *separating hyperplane*.

So, we can take:

$$(\sum_{c \in S} a_c)^T x \leq \sum_{c \in S} b_c$$

where S is the *set* of all violated constraints.

Separation Oracle

More generally, the ellipsoid method can be used, even when the *constraint matrix* is not given explicitly, as long as we can always determine a *separating hyperplane*.

In particular, the polytope represented may have *exponentially many* facets.

Anderson, D., Holm (2013) shows that as long as the *separation oracle* can be defined in FPC, the corresponding *optimization problem* can be solved in FPC.

Representations of Polytopes

A *representation* of a class $\mathcal{P}$ of *polytopes* is a *relational vocabulary* τ along with a surjective function ν taking τ-structures to polytopes in $\mathcal{P}$, which is isomorphism invariant.

A *separation oracle* for a representation $\nu, \mathcal{P}$ is definable in FPC if there is an FPC formula that given a τ-structure $\mathbb{A}$ and a vector $v \in \mathbb{Q}^V$ either

- determines that $v \in \nu(\mathbb{A})$; or
- defines a hyperplane separating v from $\nu(\mathbb{A})$.

Folding Polytopes

We use the separation oracle to define an *ordered equivalence relation* on the set V of variables.

We also define a *projection* operation on polytopes which either

- preserves feasibility; or
- refines the equivalence relation further.

Graph Matching

Recall, 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.

We saw that the existence of a *perfect matching* is not definable in FP.

(Blass, Gurevich, Shelah 1999) showed that for *bipartite* graphs this is definable in FPC.

We consider the more general problem of determining the *maximum weight* of a matching in a *weighted graph*:

$$G = (V, E) \quad w : E \to \mathbb{Q}_{\geq 0}$$

The Matching Polytope

(Edmonds 1965) showed that the problem of finding a *maximum weight matching* in $G = (V, E)$ $\quad w : \mathbb{Q}^E_{\geq 0}$ can be expressed as the following linear programming problem

$$\max w^\top y \qquad \text{subject to}$$

$$Ay \leq 1^V,$$

$$y_e \geq 0, \;\; \forall e \in E,$$

$$\sum_{e \in E \cap W^2} y_e \leq \frac{1}{2}(|W| - 1), \;\; \forall W \subseteq V \text{ with } |W| \text{ odd},$$

Matching in FPC

We show that a *separation oracle* for this polytope is definable by an FPC formula interpreted in the weighted graph G.

As a consequence, there is an FPC formula defining the *size* of the maximum matching in G.

Note that this does not allow us to define an *actual* matching.

Counting Width

Associate with any class $\mathcal{C}$ of structures the function $\nu_{\mathcal{C}} : \mathbb{N} \to \mathbb{N}$ where $\nu_{\mathbb{C}}(n)$ is the *least* k such that some formula θ of C^k defines exactly the structures in $\mathcal{C}$ with at most n elements.

Note: $\nu_{\mathbb{C}}(n) \leq n$.

If $\mathcal{C}$ is definable in FPC, then $\nu_{\mathcal{C}}$ is bounded by a constant.

Our construction, based on *toroidal grids* shows that $\nu_{\mathsf{Solv}(\mathbb{Z}_2)} = \Omega(\sqrt{n})$.

A construction based on *expander graphs* can improve this lower bound to $\Omega(n)$.

Constraint Satisfaction Problems

A *constraint language* Γ is given by a (finite) domain D and a collection of relations on D.

When Γ is finite, we think of this as a finite relational structure.

CSP(Γ) is defined as the problem of deciding, given a set of *contraints* whether it is satisfiable.

A *constraint* is a pair (v, R) where v is a tuple of variables of length a and R is a relation symbol from Γ of arity a.

So, CSP(Γ) can also be seen as the problem of determining, given an instance I, whether there is a homomorphism to Γ.

Width of CSPs

CSP(Γ) is said to have *bounded width* if

> *The complement of* CSP(Γ) *is definable in Datalog.*

This is the same as saying CSP(Γ) is solvable by *local consistency algorithms.* These are algorithms that construct assignments to the variables. Check consistency for k variables at a time (k fixed) and propagate.

If CSP(Γ) has bounded width, then it is definable in FPC and so $\nu_{\mathsf{CSP}(\Gamma)}$ is bounded by a *constant*.

Width of CSPs

By results of **(Atserias, Bulatov, D.)** and **(Barto and Kozik)**, if $\text{CSP}(\Gamma)$ is *not* definable in Datalog, then $\nu_{\text{CSP}(\Gamma)}$ is *unbounded*.

(BK) show a *sufficient, algebraic* condition for $\text{CSP}(\Gamma)$ to be of bounded width.
(ABD) shows that in the absence of these conditions, $\text{Solv}(\mathbb{Z}_m)$ can be reduced to $\text{CSP}(\Gamma)$ by means of *definable reductions*.

Atserias (based on **Valeriore**) observed that these reductions can be made *linear*.

> *If* $\text{CSP}(\Gamma)$ *is not of bounded width, then* $\nu_{\text{CSP}(\Gamma)} = \Omega(n)$.

Definability Dichotomy

Feder-Vardi Dichotomy Conjecture: For every Γ, *either* CSP(Γ) is in P *or* CSP(Γ) is NP-complete.

Definability Dichotomy: For every Γ

1. *either* $\nu_{\mathsf{CSP}(\Gamma)}$ is constant (and CSP(Γ) is definable in Datalog); *or*
2. $\nu_{\mathsf{CSP}(\Gamma)}$ is $\Omega(n)$ (and CSP(Γ) is *not* definable in FPC.

Note: all problems in *(1)* are in P.
Some problems in *(2)* (such as Solv$(\mathbb{Z}_2)$) are also in P.

Optimization of CSPs

Max-CSP(Γ) is the problem of determining, given an instance I of CSP(Γ) what is the *maximum* number of constraints that can be simultaneously satisfied.

Thapper-Živný dichotomy:

1. If CSP(Γ) is of bounded width, Max-CSP(Γ) is solvable in *polynomial time*, by its *basic linear programming relaxation*.
2. If CSP(Γ) is *not* of bounded width, Max-CSP(Γ) is NP-hard.

e.g. Max-XOR-SAT.

Linear Programming Relaxations

Each instance I of Max-CSP(Γ) can be turned into a linear program: BLP(I)

Set of variables V, domain D, constraints $c = (x, R)$

$$\max \sum_{c \in C} \sum_{d \in R^\Gamma} \lambda_{c,d} \quad \text{where } c = (x, R)\text{, s.t.}$$

$$\sum_{d \in D^{|x|}; d_i = a} \lambda_{c,d} = \mu_{x_i,a} \qquad \forall c \in C, a \in D, i \in [|x|]$$

$$\sum_{a \in D} \mu_{v,a} = 1 \qquad \forall v \in V$$

Lift and Project Hierarchies

Given a *polytope* $\mathcal{K}$ for *integer* optimization problem, we can get a better approximation of the *convex hull* of the integer points by means of *lift-and-project* programs.

The general idea is to add new variables $y_{x_1,\ldots,x_t}$ to denote the product $x_1 \cdots x_t$ and add linear (or semi-definite) constraints to try and force this meaning.

We get hierarchies as t increases:

- *Sherali-Adams*: $\mathsf{SA}_t(\mathcal{K})$
- *Lovasz-Schrijver*: $\mathsf{LS}_t(\mathcal{K})$
- *Lasserre*: $\mathsf{Las}_t(\mathcal{K})$

Of these, the last is the strongest.

Lasserre Hierarchy

Let $\mathcal{K} = \{x \in \mathbb{Q}^V \mid Ax \geq b\}$, and $y \in \mathrm{Las}_t(\mathcal{K})$ for $t \in \{1, \ldots, |V|\}$. Then,

1. $\mathcal{K}^* \subseteq \mathrm{Las}_t^\pi(\mathcal{K})$.
2. $\mathrm{Las}_0(\mathcal{K}) \supseteq \mathrm{Las}_1(\mathcal{K}) \supseteq \ldots \supseteq \mathrm{Las}_{|V|}(\mathcal{K})$.
3. $\mathrm{Las}_0^\pi(\mathcal{K}) \subseteq \mathcal{K}$, and $\mathcal{K}^* = \mathrm{Las}_{|V|}^\pi(\mathcal{K})$.

Lasserre and Definability

(D., Wang 2016):

For each Γ and t, there is an FPC interpretation that takes an instance I of CSP(Γ) to the tth level of the Lasserre hierarchy over BLP(I).

The FPC implementation of the *ellipsoid method* extends to *semdefinite* programs (subject to some technical conditions).

Corollary

If the tth level of the Lasserre hierarchy solves Max-CSP(Γ), then $t = \Omega(\nu_{\mathsf{CSP}(\Gamma)})$.

Corollary

If CSP(Γ) is not of bounded width, then $\Omega(n)$ levels of the Lasserre hierarchy are necessary to obtain the convex hull of the integer solutions BLP(Max-CSP(Γ)).