Symmetric Computation: Lecture 5

Anuj Dawar and Gregory Wilsenach

Department of Computer Science and Technology, University of Cambridge

ESSLLI, August 2021

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 $\nu_{\mathsf{XOR\text{-}SAT}} = \Omega(\sqrt{n})$.

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 *constraints* 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 CSP(Γ) is *not* definable in Datalog, then $\nu_{\mathsf{CSP}(\Gamma)}$ is *unbounded*.

(BK) show a *sufficient, algebraic* condition for CSP(Γ) to be of bounded width.
(ABD) shows that in the absence of these conditions, XOR-SAT can be reduced to CSP(Γ) by means of *definable reductions*.

These reductions can be made *linear*.

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

Definability Dichotomy

Bulatov-Zhuk Dichotomy Theorem: For every Γ, *either* $\mathsf{CSP}(\Gamma)$ is in P *or* $\mathsf{CSP}(\Gamma)$ is NP-complete.

Definability Dichotomy: For every Γ

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

Note: all problems in *(1)* are in P.
Some problems in *(2)* (such as XOR-SAT) 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. $\mathcal{K}^* \subseteq \mathrm{Las}^{\pi}_t(\mathcal{K})$.
2. $\mathrm{Las}_0(\mathcal{K}) \supseteq \mathrm{Las}_1(\mathcal{K}) \supseteq \ldots \supseteq \mathrm{Las}_{|V|}(\mathcal{K})$.
3. $\mathrm{Las}^{\pi}_0(\mathcal{K}) \subseteq \mathcal{K}$, and $\mathcal{K}^* = \mathrm{Las}^{\pi}_{|V|}(\mathcal{K})$.

Lasserre and Definability

(D., Wang 2017):

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))*.*

NP Optimization Problems

MAX 3SAT:
We are given a *Boolean formula* φ in 3CNF, i.e. a conjunction of clauses with three literals per clause.

Say φ has n Boolean variables and m clauses.

Let m^* denote the *maximum* number such that some assignment of values to the Boolean variables makes m^* clauses of φ true.

Algorithmic Problems:

- *Find* an assignment of values to the variables that makes m^* clauses of φ true;
- *Determine* the value of m^*;
- c-*approximate* m^* for some constant $0 < c < 1$, i.e. give a value m' with a guarantee that $cm^* \leq m' \leq m^*$.

Lower Bounds

NP-completeness (Cook; Levin 1973):
Unless $\mathsf{P} = \mathsf{NP}$, there is no *polynomial-time algorithm* that can determine m^*.

PCP Theorem (Arora et al. 1998):
There is a constant $c < 1$ such that, unless $\mathsf{P} = \mathsf{NP}$, there is no *polynomial-time algorithm* that can c-approximate m^*.

(Håstad 2001):
Unless $\mathsf{P} = \mathsf{NP}$, for every $\epsilon > 0$ there is no *polynomial-time algorithm* that can $(\frac{7}{8} + \epsilon)$-approximate m^*.

> *Note: This is optimal since there is a trivial algorithm that can $\frac{7}{8}$-approximate m^*.*

Why 7/8?

Given a 3CNF clause, say $(\overline{x} \vee y \vee z)$

$\frac{7}{8}$ *of all Boolean assignments satisfy it.*

A simple *averaging* argument then shows that there is an assignment of values to the Boolean variables that satisfies $\frac{7}{8}$ of the clauses.

MAX 3XOR

We are given a Boolean formula φ in 3XOR, i.e. a conjunction of clauses each of which is the *exclusive or* ($\oplus$) of three literals.

Say φ has n Boolean variables and m clauses.

Let m^* denote the *maximum* number such that some assignment of values to the Boolean variables makes m^* clauses of φ true.

- determining whether $m^* = m$ can be done in polynomial-time, by Gaussian elimination;
- determining the exact value of m^* is MAX SNP-complete.

(Håstad 2001):
Unless $\mathsf{P} = \mathsf{NP}$, for every $\epsilon > 0$ there is no *polynomial-time algorithm* that can $(\frac{1}{2} + \epsilon)$-approximate m^*.

> *This is optimal since there is a trivial algorithm that can $\frac{1}{2}$-approximate m^*.*

Vertex Cover

In a graph $G = (V, E)$, $S \subseteq V$ is a *vertex cover* if each edge in E has at least one endpoint in S.

$\mathsf{vc}(G)$ is the *size* of the smallest vertex cover in G.

(Dinur-Safra 2005):
Unless $\mathsf{P} = \mathsf{NP}$, there is no polynomial-time algorithm that can approximate $\mathsf{vc}(G)$ up to a factor of $10\sqrt{5} - 21 \approx 1.36$.

> *Note 1: Since this is a minimization problem, the approximation ratio is a constant $c > 1$.*
> *Note 2: This has very recently been improved to $\sqrt{2}$* **(Khot, Minzer, Safra 2018+)***.*

There are polynomial-time algorithms that can approximate $\mathsf{vc}(G)$ up to a factor of 2.

Conjecture:
Unless $\mathsf{P} = \mathsf{NP}$, for every $\epsilon > 0$ there is no polynomial-time algorithm that can approximate $\mathsf{vc}(G)$ up to a factor of $2 - \epsilon$.

Methods

Say that a 3CNF formula is c-*satisfiable* if $m^* > cm$.

The proof of the PCP theorem gives (for some constant c) a reduction from 3SAT to itself which:

- maps a satisfiable formula to a satisfiable formula; and
- maps an unsatisfiable formula to one that is not c-satisfiable.

As a consequence, any class $\mathcal{C}$ of formulas that includes the satisifiable ones and excludes the ones that are *not* c-satisfiable, is NP-hard to decide.

We say that the class of satisfiable formulas is not *efficiently separable* from the ones that are not c-satisfiable.

Amplification

The gap is amplified by further reductions, such as **Håstad's** long-code reductions.

In the case of 3XOR:

> *For any $\epsilon > 0$, any class $\mathcal{C}$ of formulas that includes the $(1 - \epsilon)$-satisfiable ones and excludes the ones that are not $(\frac{1}{2} + \epsilon)$-satisfiable, is NP-hard to decide.*

From this, by reduction, we also obtain the optimal lower bound for MAX 3SAT, but this can be improved to *perfect completeness*:

> *For any $\epsilon > 0$, any class $\mathcal{C}$ of 3CNF formulas that includes the satisfiable ones and excludes the ones that are not $(\frac{7}{8}+\epsilon)$-satisfiable, is NP-hard to decide.*

Vertex Cover

The *textbook* proof of the NP-completeness of vertex cover, takes a 3CNF formula φ (with n variables and m clauses) and gives a graph G with:

- $3m$ vertices; and
- $\mathsf{vc}(G) = 2m^*$.

This shows (using the Håstad bound on MAX 3SAT) that the class of graphs with $\mathsf{vc}(G) \leq (\frac{7}{12} + \epsilon)|V(G)|$ is not *efficiently separable* from the graphs with $\mathsf{vc}(G) \geq \frac{2}{3}|V(G)|$.

This yields an inapproximability bound of $\frac{8}{7}$.
A better bound of $\frac{7}{6}$ can obtained by a reduction from 3XOR.
The bounds of 1.36 and $\sqrt{2}$ are obtained by much more sophisticated reductions (but still *gadgets*).

Results

For any $\epsilon > 0$ there is no term of FPC which, interpreted in a 3CNF formula φ, defines a number guaranteed to be within $\frac{7}{8} + \epsilon$ of $m^*(\varphi)$.

For any $\epsilon > 0$ there is no term of FPC which, interpreted in a 3XOR formula φ, defines a number guaranteed to be within $\frac{1}{2} + \epsilon$ of $m^*(\varphi)$.

There is no term of FPC which, interpreted in a graph G, defines a value guaranteed to be within a factor 1.36 of $\mathsf{vc}(G)$.

New Challenges for Duplicator

The results are established by showing *definability gaps*:

> *If $\mathcal{C}$ is any class of 3XOR formulas that includes the satisfiable ones and excludes those that are not $(\frac{1}{2}+\epsilon)$-satisfiable, then $\mathcal{C}$ has counting width $\Omega(n)$ for some.*

Note perfect completeness.

Then, by reduction:

> *If $\mathcal{C}$ is any class of 3CNF formulas that includes the satisfiable ones and excludes those that are not $(\frac{7}{8}+\epsilon)$-satisfiable, then $\mathcal{C}$ has counting width $\Omega(n)$.*

Initial Gap

Unlike the PCP theorem, we establish an initial gap for 3XOR:

> *If $\mathcal{C}$ is any class of 3XOR formulas that includes the satisfiable ones and excludes those that are not $(\frac{1}{2}+\epsilon)$-satisfiable, then $\mathcal{C}$ has counting width $\Omega(n)$.*

We then extend it to 3SAT and *vertex cover* by means of *reductions* definable in *first-order logic*.

This involves showing that known polynomial-time reductions in the literature can be done in first-order logic.

k-local Consistency

Given a 3XOR formula φ and $k \in \mathbb{N}$, consider the following game:
At stage 0:

- *Challenger* chooses a set X_0 of at most k clauses from φ
- *Prover* gives an assignment α_0 of values to the variables satisfying all clauses in X_0

At stage $i + 1$:

- *Challenger* chooses a set X_{i+1} of at most k clauses from φ
- *Prover* gives an assignment α_{i+1} of values to the variables, that agrees with α_i on all variables in $X_i \cap X_{i+1}$ and satisfies all clauses in X_{i+1}

We say that φ is *k-locally consistent* if *Prover* has a strategy to play forever.

CFI construction

We can treat a formula φ of 3XOR as a system of equations over $\mathbb{Z}_2$:

$$x + y + z = b \qquad b \in \{0, 1\}$$

Define the system $\mathsf{cfi}(\varphi)$ to be the system obtained by replacing each variable x with two variables x^0 and x^1. and each equation $x + y + z = b$ with eight equations:

$$x^i + y^j + z^k = b + i + j + k$$

Also, define φ_0—the *homogeneous companion* of φ—to be the system obtained from φ by replacing b by 0 in all equations.

Claim: if φ is k-locally consistent, then $\mathsf{cfi}(\varphi) \equiv^k \mathsf{cfi}(\varphi_0)$.

Random 3XOR

For a set V of n *variables*, choose uniformly at random, a collection of $m > n$ subsets $\{x_1, x_2, x_3\}$ of V of three elements.

> *With high probability, the resulting bipartite graph has certain expansion properties.*

Construct a system of equations $x_1 + x_2 + x_3 = b$ where the left-hand sides are the chosen sets and b is 0 or 1 based on the toss of a coin.

> *With high probability, the system is not $(\frac{1}{2} + \epsilon)$-satisfiable.*
>
> *The expansion properties guarantee that it is k-locally consistent for $k = \Omega(n)$.*

CFI Again

If φ is satisfiable, then $\text{cfi}(\varphi)$ is satisfiable.

If φ is not c-satisfiable then $\text{cfi}(\varphi)$ is not $(\frac{1}{2} + \frac{c}{2})$ satisfiable.

This means that for a k-locally consistent φ that is not $(\frac{1}{2} + \epsilon)$-satisfiable:

- $\text{cfi}(\varphi_0)$ is satisfiable;
- $\text{cfi}(\varphi)$ is not $(\frac{3}{4} + \epsilon)$-satisfiable; and
- $\text{cfi}(\varphi) \equiv^k \text{cfi}(\varphi_0)$.

A more careful analysis of the probabilistic construction actually shows that for a random φ, with high probability, $\text{cfi}(\varphi)$ is not $(\frac{1}{2} + \epsilon)$ satisfiable.

First-Order Reductions

An FO *interpretation* θ of a structure $\mathbb{B}$ in $\mathbb{A}$ is a family of first-order formulas which define the *universe* and *relations* of $\mathbb{B}$ when interpreted in $\mathbb{A}$. We write $\mathbb{B} = \theta(\mathbb{A})$.

An FO *reduction* of a class of structures $\mathcal{C}$ to a class $\mathcal{D}$ is a single FO interpretation θ such that $\mathbb{A} \in \mathcal{C}$ if, and only if, $\theta(\mathbb{A}) \in \mathcal{D}$.
We write $\mathcal{C} \leq_{\mathsf{FO}} \mathcal{D}$.

If $\mathcal{C}$ has *unbounded* counting width and $\mathcal{C} \leq_{\mathsf{FO}} \mathcal{D}$, then $\mathcal{D}$ has unbounded counting width.

Moreover, if the reduction from $\mathcal{C}$ to $\mathcal{D}$ is *linearly bounded* (i.e. the size of $\theta(\mathbb{A})$ is linear in $\mathbb{A}$), then if the counting width of $\mathcal{C}$ is $\Omega(n)$ so is the width of $\mathcal{D}$.

3SAT and Vertex Cover

The reduction from 3XOR to 3SAT just takes each clause $x \oplus y \oplus z$ to the set of four clauses:

$$(\overline{x} \vee \overline{y} \vee z); (\overline{x} \vee y \vee \overline{z}); (x \vee \overline{y} \vee \overline{z}); \text{ and } (x \vee y \vee z)$$

We are also able to show that the **Håstad** long-code reductions, the standard reductions to vertex cover, as well as the **Dinur-Safra** reduction can all be expressed by FO reductions.

Choiceless Polynomial Time

Choiceless Polynomial Time (C̃PT) is a class of computational problems defined by **Blass, Gurevich and Shelah**.

It is based on a *machine model (Gurevich Abstract State Machines)* which can be seen as an extension of the *relational machines*.

The machine can access the collection of hereditarily finite sets over the universe of the structure.

C̃PT is the polynomial time and space restriction of the machines.

C̃PT is strictly more expressive than FP, but still cannot express counting properties.

Consider C̃PT(Card)—the extension of C̃PT with counting.

Does it express all properties in P?

Choiceless Polynomial Time

$\tilde{\mathrm{C}}\mathrm{PT}$ can express the property of **Cai, Fürer and Immerman**.

Any program of $\tilde{\mathrm{C}}\mathrm{PT}(\mathrm{Card})$ that expresses the CFI property must use sets of *unbounded rank*.

FPC can be translated to programs of $\tilde{\mathrm{C}}\mathrm{PT}(\mathrm{Card})$ of bounded rank.
(D., Richerby and Rossman 2008)

Rank Logics

FPC cannot define solvability of linear equations over finite fields

e.g. *XOR-SAT.*

FPrk—*fixed-point logic with rank* is an extension of FP with *matrix rank operators*

$$[\mathrm{rk}_\pi \mathbf{x}\mathbf{y}]\eta(\mathbf{x},\mathbf{y})$$

denotes in a structure $\mathbb{A}$ the *rank* in $\mathbb{F}_p$ of the $A^k \times A^k$-matrix M where

$$M_{\mathbf{a},\mathbf{b}} = \eta^{\mathbb{A}}[\mathbf{a},\mathbf{b}] \pmod{p}$$

Here, π is a numerical term denoting the prime p.

Equivalence Induced by Rank Logic

FPrk properly extends the expressive power of FPC. In particular, it can express XOR-SAT; solvability of linear equations on finite fields; CFI graph.

We can define R^k, an extension of first-order logic with *matrix rank quantifiers* and limited to k variables.
And the corresponding equivalence relation $\equiv^{R^k}$.

For every formula φ of FPrk, there is a k such that the class of structures defined by φ is invariant under $\equiv^{R^k}$.

These equivalence relations are not as well behaved as $\equiv^{C^k}$

> *In particular, it is not known if $\equiv^{R^k}$ can be decided in polynomial time, for fixed k.*

Invertible Map Equivalence

A better behaved, and more robust equivalence is obtained as a *refinement* of $\equiv^{R^k}$.

These are the *invertible map equivalences* $\equiv^{\mathrm{IM}^k}$.

They correspond to equivalence in the k-variable fragment of an extension of first-order logic with *all linear algebraic* operators.

The relation $\equiv^{\mathrm{IM}^k}$ is decidable in time $n^{O(k)}$.

The equivalence relations $\equiv^{\mathrm{IM}^k}$ have a clean characterization in terms of *algebraic pebble games*.

Lower Bound Result

Lichter (LICS 2021) has shown that FPrk can be *separated* from P. That is, there is a polynomial-time decidable class of structures not definable in FPrk.

The separating example is systems of equations over a (*variable*) finite ring $\mathbb{Z}_{2^i}$.
The proof is an application of the *invertible map games*.

As a consequence, combining with results of **D., Grädel,Pakusa (2019)** we get that $\equiv^{\mathrm{IM}^k}$ is not the same as isomorphism for any fixed k.

Circuits for Rank Logic

We can define a *circuit model* for FPrk just as we did for FPC.

This is based on *symmetric circuits* with *matrix rank gates*.

Such gates are not *symmetric* in the sense of being invariant under all permutations of their inputs.
Thus, the circuit definition requires imposing additional structure on the circuits.

The proof of equivalence with FPrk is again based on a *support theorem*, which has to take account of this extra structure.

Concluding Remarks

The class of problems definable in FPC forms a *robust* class within P of problems solvable by *symmetric* polynomial time algorithms.

The robustness is demonstrated by the variety of equivalent characterizations:

- *logic*;
- *machines*;
- *circuits*;
- *linear programs*

This is combined with a method for proving *undefinability* based on *games* which gives *concrete lower bounds* for powerful and natural algorithmic methods in *constraint satisfaction* and *optimization*.

There is scope for extending the methods beyond the expressive power of FPC.