The State of the Art in the Quest for a Logic for Polynomial Time

Erich Grädel

Helmut Veith Workshop, Obergurgl, February 2017

The most important problem of Finite Model Theory

Is there a logic that captures PTIME?

The most important problem of Finite Model Theory

Is there a logic that captures PTIME?

Informal definition: A logic L captures PTIME if it defines precisely those properties of finite structures that are decidable in polynomial time:

(1) For every sentence $\psi \in L$, the set of finite models of ψ is decidable in polynomial time.

(2) For every PTIME-property S of finite τ-structures, there is a sentence $\psi \in L$ such that $S = \{\mathfrak{A} \in \mathrm{Fin}(\tau) : \mathfrak{A} \models \psi\}$.

The most important problem of Finite Model Theory

Is there a logic that captures PTIME?

Informal definition: A logic L captures PTIME if it defines precisely those properties of finite structures that are decidable in polynomial time:

(1) For every sentence $\psi \in L$, the set of finite models of ψ is decidable in polynomial time.

(2) For every PTIME-property S of finite τ-structures, there is a sentence $\psi \in L$ such that $S = \{\mathfrak{A} \in \mathrm{Fin}(\tau) : \mathfrak{A} \models \psi\}$.

The precise definition is more subtle. It includes effectiveness requirements to exclude pathological 'solutions'.

First-Order Logic

First-order logic (FO) is far too weak to capture PTIME.

- FO can express only local properties of finite structures
  Theorems of Gaifman and Hanf

  Global properties (e.g. planarity of graphs) are not expressible.

- FO has no mechanism for recursion or unbounded iteration.

  Transitive closures, reachability or termination properties, winning regions in games, etc. are not FO-definable.

- FO can only express properties in AC^0

  AC^0 is constant parallel time with polynomial hardware. In particular, FO $\subseteq$ LOGSPACE.

Second-Order Logic

Second-order logic (SO) is probably too strong to capture PTIME.

Fagin's Theorem. Existential SO captures NP.

Corollary. SO captures the polynomial hierarchy.

Thus SO captures polynomial time if, and only if, P = NP.

Second-Order Logic

Second-order logic (SO) is probably too strong to capture PTIME.

Fagin's Theorem. Existential SO captures NP.

Corollary. SO captures the polynomial hierarchy.

Thus SO captures polynomial time if, and only if, P = NP.

Monadic second-order logic is orthogonal to PTIME:

On words, MSO captures the regular languages, and not all PTIME-languages are regular.

On graphs, MSO can express NP-complete properties, such as 3-colourability.

Fixed-point logic with counting

(FP + C): Two-sorted fixed-point logic with counting terms.

Two sorts of variables:

- $x, y, z, \ldots$ ranging over the domain of the given finite structure
- $\mu, \nu, \ldots$ ranging over natural numbers

On natural numbers, standard operations $+$, $\cdot$ and $<$ are available, but variables must be explicitly restricted to take only polynomially bounded values.

Counting terms: For a formula $\varphi(x)$, the term $\#_x \varphi(x)$ denotes the number of elements a of the structure that satisfy $\varphi(a)$.

Mechanism for polynomial-time relational recursion.

Least or inflationary fixed points of definable update operators $R \mapsto \{(\overline{a}, \overline{m}) : \mathfrak{A} \models \varphi(R, \overline{a}, \overline{m})\}$ on mixed relations $R(\overline{x}, \overline{\mu})$.

Fixed-point logic with counting is close to PTIME

Fixed-point logic with counting is powerful enough to express fundamental algorithmic techniques (such as the ellipsoid method) and captures PTIME on many interesting classes of finite structures, including

- linearly ordered structures (Immerman, Vardi)
- trees (Immerman, Lander) and
  structures of bounded tree-width (Grohe, Marino)
- planar graphs and graphs of bounded genus (Grohe)
- chordal line graphs (Grohe) and interval graphs (Laubner)
- all classes of graphs that exclude a minor (Grohe)

Fixed-point logic with counting is close to PTIME

Fixed-point logic with counting is powerful enough to express fundamental algorithmic techniques (such as the ellipsoid method) and captures PTIME on many interesting classes of finite structures, including

- linearly ordered structures (Immerman, Vardi)
- trees (Immerman, Lander) and
  structures of bounded tree-width (Grohe, Marino)
- planar graphs and graphs of bounded genus (Grohe)
- chordal line graphs (Grohe) and interval graphs (Laubner)
- all classes of graphs that exclude a minor (Grohe)

(FP+C) is the logic of reference in this area!
(see survey by A. Dawar, SIGLOG-News, 2015)

The CFI-query

Given a connected graph $G = (V, E)$, and a subset $T \subseteq E$, construct the CFI-graph $X_T(G)$:

- replace every node v by a gadget $H(v)$, which has two exit points a_{vw} and b_{vw} for every neighbour $w \in vE$
- replace every edge by two edges that connnect corresponding exit points: a_{vw} with a_{wv} and b_{vw} with b_{wv}
- twist the double-edges in T

The CFI-query

Given a connected graph $G = (V, E)$, and a subset $T \subseteq E$, construct the CFI-graph $X_T(G)$:

- replace every node v by a gadget $H(v)$, which has two exit points a_{vw} and b_{vw} for every neighbour $w \in vE$
- replace every edge by two edges that connnect corresponding exit points: a_{vw} with a_{wv} and b_{vw} with b_{wv}
- twist the double-edges in T

Fact: $X_S(G) \cong X_T(G) \iff |S| = |T| \pmod 2$

Thus, for every G, there are up to isomorphism exactly two CFI-graphs:
$X(G) := X_\emptyset(G)$ and $\tilde{X}(G) := X_{\{e\}}(G)$

The CFI-query

Given a connected graph $G = (V, E)$, and a subset $T \subseteq E$, construct the CFI-graph $X_T(G)$:

- replace every node v by a gadget $H(v)$, which has two exit points a_{vw} and b_{vw} for every neighbour $w \in vE$
- replace every edge by two edges that connnect corresponding exit points: a_{vw} with a_{wv} and b_{vw} with b_{wv}
- twist the double-edges in T

Fact: $X_S(G) \cong X_T(G) \iff |S| = |T| \pmod 2$

Thus, for every G, there are up to isomorphism exactly two CFI-graphs: $X(G) := X_\emptyset(G)$ and $\tilde{X}(G) := X_{\{e\}}(G)$

The CFI-query: Given a CFI-graph, determine whether it is $X(G)$ or $\tilde{X}(G)$.

Fixed-point logic with counting versus polynomial time

Theorem. The CFI-query is in PTIME, but not in (FP + C).

(Cai, Fürer, Immerman 1992)

Fixed-point logic with counting versus polynomial time

Theorem. The CFI-query is in PTIME, but not in (FP + C).
(Cai, Fürer, Immerman 1992)

The CFI-construction separating PTIME from (FP+C) is interesting and sophisticated, but originally seemed somewhat artificial.

However, Atserias, Bulatov, and Dawar proved that it very closely related to the fundamental problem of solving linear equation systems over finite Abelian groups, rings, and fields.

Fixed-point logic with counting versus polynomial time

Theorem. The CFI-query is in PTIME, but not in (FP + C).
(Cai, Fürer, Immerman 1992)

The CFI-construction separating PTIME from (FP+C) is interesting and sophisticated, but originally seemed somewhat artificial.

However, Atserias, Bulatov, and Dawar proved that it very closely related to the fundamental problem of solving linear equation systems over finite Abelian groups, rings, and fields.

Today, the CFI-query and its variants and generalizations still provide interesting benchmarks and challenges for any candidate for a logic for polynomial time.

Hard problems for (FP+C)

There are two main sources of natural polynomial-time problems that are hard for logic, and not in (FP+C):

(1) Tractable cases of isomorphism problems

- Structures of colour class size q: coloured by an ordered set such that at most q elements get the same colour
- Multipedes (Blass and Gurevich)
- Graphs of bounded degree

Hard problems for (FP+C)

There are two main sources of natural polynomial-time problems that are hard for logic, and not in (FP+C):

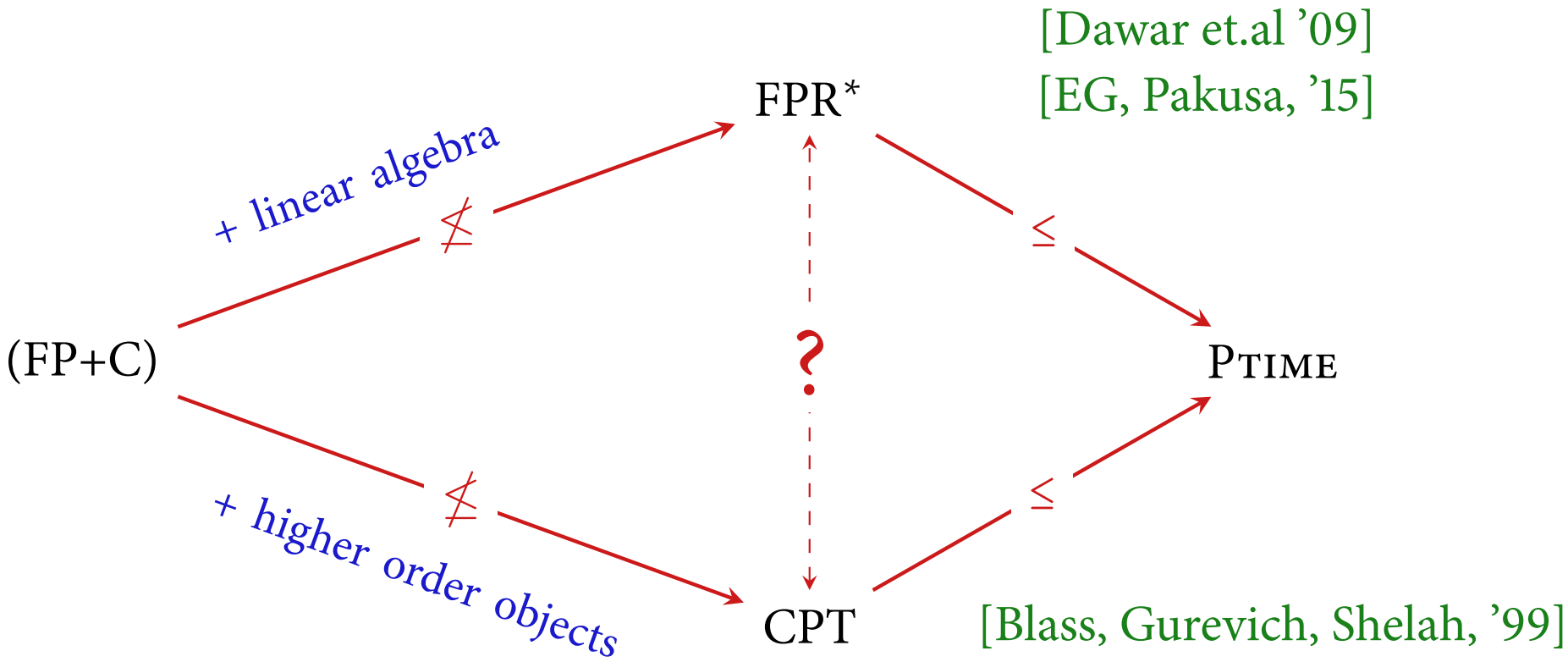
(1) Tractable cases of isomorphism problems

- Structures of colour class size q: coloured by an ordered set such that at most q elements get the same colour
- Multipedes (Blass and Gurevich)
- Graphs of bounded degree

(2) Problems from linear algebra and permutation group theory

- Solving linear equation systems
- Matrix rank over finite fields
- Permutation group membership problem

Candidates for a logic for PTIME



Fixed-point logic with rank

Rank logic FPR: Extend fixed-point logic by rank operators $\mathrm{rk}_p\varphi$, to denote the rank (over the prime field $\mathbb{F}_p$) of the matrix defined by φ.

FPR has been proposed in 2009 by Dawar, Grohe, Holm, Laubner as a potential candidate for a logic for PTIME.

Fixed-point logic with rank

Rank logic FPR: Extend fixed-point logic by rank operators $\mathrm{rk}_p\varphi$, to denote the rank (over the prime field $\mathbb{F}_p$) of the matrix defined by φ.

FPR has been proposed in 2009 by Dawar, Grohe, Holm, Laubner as a potential candidate for a logic for PTIME.

FPR can express the solvability of linear equation systems over finite fields, and thus the isomorphism of CFI-graphs: (FP+C) < FPR ≤ PTIME.

Fixed-point logic with rank

Rank logic FPR: Extend fixed-point logic by rank operators $\mathrm{rk}_p\varphi$, to denote the rank (over the prime field $\mathbb{F}_p$) of the matrix defined by φ.

FPR has been proposed in 2009 by Dawar, Grohe, Holm, Laubner as a potential candidate for a logic for PTIME.

FPR can express the solvability of linear equation systems over finite fields, and thus the isomorphism of CFI-graphs: (FP+C) < FPR ≤ PTIME.

Theorem. (EG, Pakusa, CSL 2015) Rank logic is dead, long live rank logic!

In its original form, FPR fails to capture PTIME, and must be replaced by a stronger variant, FPR*, with a uniform rank operator, taking the prime as an additional input.

Open problem. Does FPR* capture PTIME ?

Choiceless Polynomial Time

introduced by Blass, Gurevich, and Shelah in 1999

Idea. Model for efficient computation on abstract structures that preserves symmetries at every step of the computation. Disallow explicit choice, but permit essentially everything else, including fancy data structures and parallelism (explore all possible choices in parallel).

BGS-machines: model of abstract state machines that operate on hereditarily finite expansions $\mathrm{HF}(\mathfrak{A})$ of finite structures $\mathfrak{A}$

$\mathrm{HF}(\mathfrak{A})$ has universe consisting of
- atoms: the elements of $\mathfrak{A}$
- all finite sets of elements of $\mathrm{HF}(\mathfrak{A})$ with set-theoretic operations such as
  $\varnothing, \in, \bigcup, \ldots$ and a cardinality operator

BGS-logic

BGS-machines can be described and understood in logical terms

BGS-logic: terms $t(\overline{x})$ and formulae $\varphi(\overline{x})$ constructed via

- quantifier-free part of first-order logic
- basic set-theoretic operators $\varnothing, \in, \bigcup, \ldots$
- a cardinality operator $x \mapsto |x|$ (as a von Neumann ordinal)
- comprehension terms of form $\{t(x) : x \in t' : \varphi(x)\}$
- an iteration operator $t(x)^*$

Evaluation: $[\![t(x_1, \ldots, x_n)]\!]^{\mathfrak{A}} : \mathrm{HF}(\mathfrak{A})^n \to \mathrm{HF}(\mathfrak{A})$,

For sentences of BGS-logic, we have $[\![\varphi]\!]^{\mathfrak{A}} \in \{\text{false}, \text{true}\} = \{\varnothing, \{\varnothing\}\}$

Definition of choiceless polynomial time

For an iteration term $t(x)^*$, we get a sequence $x_0, x_1, x_2, \ldots$ of sets in $\mathrm{HF}(\mathfrak{A})$ with $x_0 = \varnothing$ and $x_{i+1} := [\![t]\!](x_i)$, and we let

$$[\![t^*]\!] = x_\ell \text{ for the least } \ell \text{ with } x_{\ell+1} = x_\ell \text{ if it exists, and } \varnothing \text{ otherwise}$$

Definition of choiceless polynomial time

For an iteration term $t(x)^*$, we get a sequence $x_0, x_1, x_2, \ldots$ of sets in $\mathrm{HF}(\mathfrak{A})$ with $x_0 = \varnothing$ and $x_{i+1} := [\![t]\!](x_i)$, and we let

$$[\![t^*]\!] = x_\ell \text{ for the least } \ell \text{ with } x_{\ell+1} = x_\ell \text{ if it exists, and } \varnothing \text{ otherwise}$$

A computation described in BGS-logic is a sequence of hereditarily finite sets.

Choiceless Polynomial Time is the polynomial-time fragment of BGS-logic. It is the set of properties definable in BGS-logic such that

- all iterations have polynomial length
- only a polynomial number of sets are activated.

The power of choiceless polynomial time

CPT is a proper extension of (FP + C)

CPT can define any polynomial time property of small definable substructures X of the input structure $\mathfrak{A}$.

The power of choiceless polynomial time

CPT is a proper extension of (FP + C)

CPT can define any polynomial time property of small definable substructures X of the input structure $\mathfrak{A}$.

Small: $|X|! \leq |A|$. Generate in parallel all linear orders on X and simulate a polynomial time computation on an ordered structure by the usual techniques.

The power of choiceless polynomial time

CPT is a proper extension of (FP + C)

CPT can define any polynomial time property of small definable substructures X of the input structure $\mathfrak{A}$.

Small: $|X|! \leq |A|$. Generate in parallel all linear orders on X and simulate a polynomial time computation on an ordered structure by the usual techniques. For graphs, this has been strengthened by Laubner to $|X| < \log |A|$ implementing a graph canonization algorithm working in time 2^n.

The power of choiceless polynomial time

CPT is a proper extension of (FP + C)

CPT can define any polynomial time property of small definable substructures X of the input structure $\mathfrak{A}$.
Small: $|X|! \leq |A|$. Generate in parallel all linear orders on X and simulate a polynomial time computation on an ordered structure by the usual techniques. For graphs, this has been strengthened by Laubner to $|X| < \log |A|$ implementing a graph canonization algorithm working in time 2^n.

CPT (even without counting) can distinguish the CFI-graphs constructed from ordered graphs (so that the CFI-graphs themselves have a preorder).

The power of choiceless polynomial time

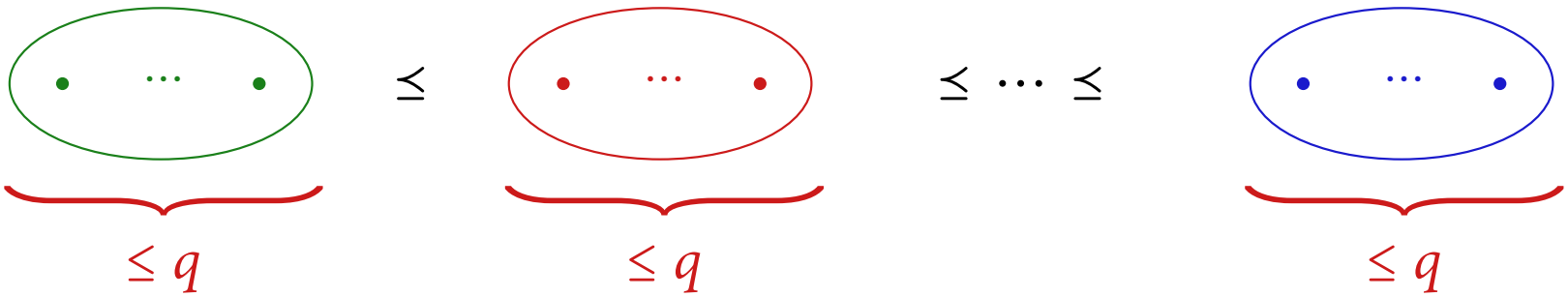
CPT is a proper extension of (FP + C)

CPT can define any polynomial time property of small definable substructures X of the input structure $\mathfrak{A}$.
Small: $|X|! \leq |A|$. Generate in parallel all linear orders on X and simulate a polynomial time computation on an ordered structure by the usual techniques.
For graphs, this has been strengthened by Laubner to $|X| < \log |A|$ implementing a graph canonization algorithm working in time 2^n.

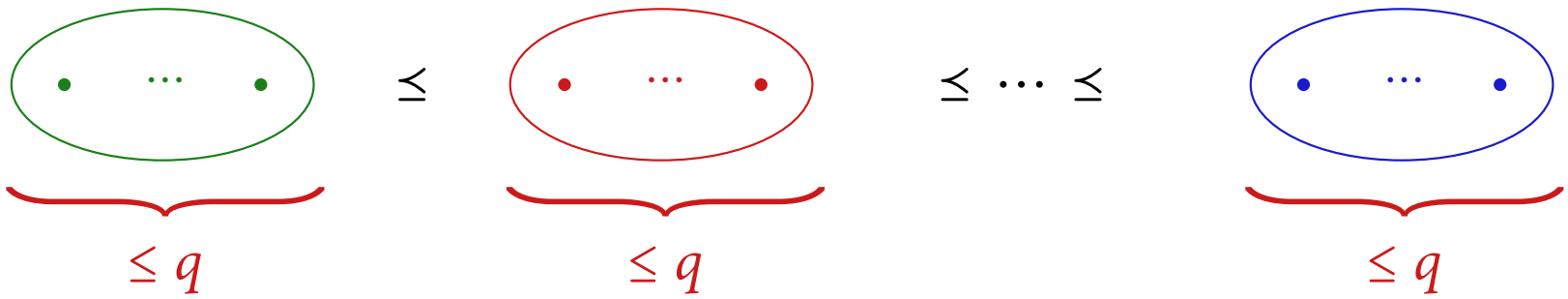
CPT (even without counting) can distinguish the CFI-graphs constructed from ordered graphs (so that the CFI-graphs themselves have a preorder).

This extends to CFI-graphs constructed from graphs with many edges (in particular cliques), and to graphs with logarithmic colour class size
(Pakusa, Schalthöfer, Selman 2016)

Structures of bounded colour class size



Structures of bounded colour class size

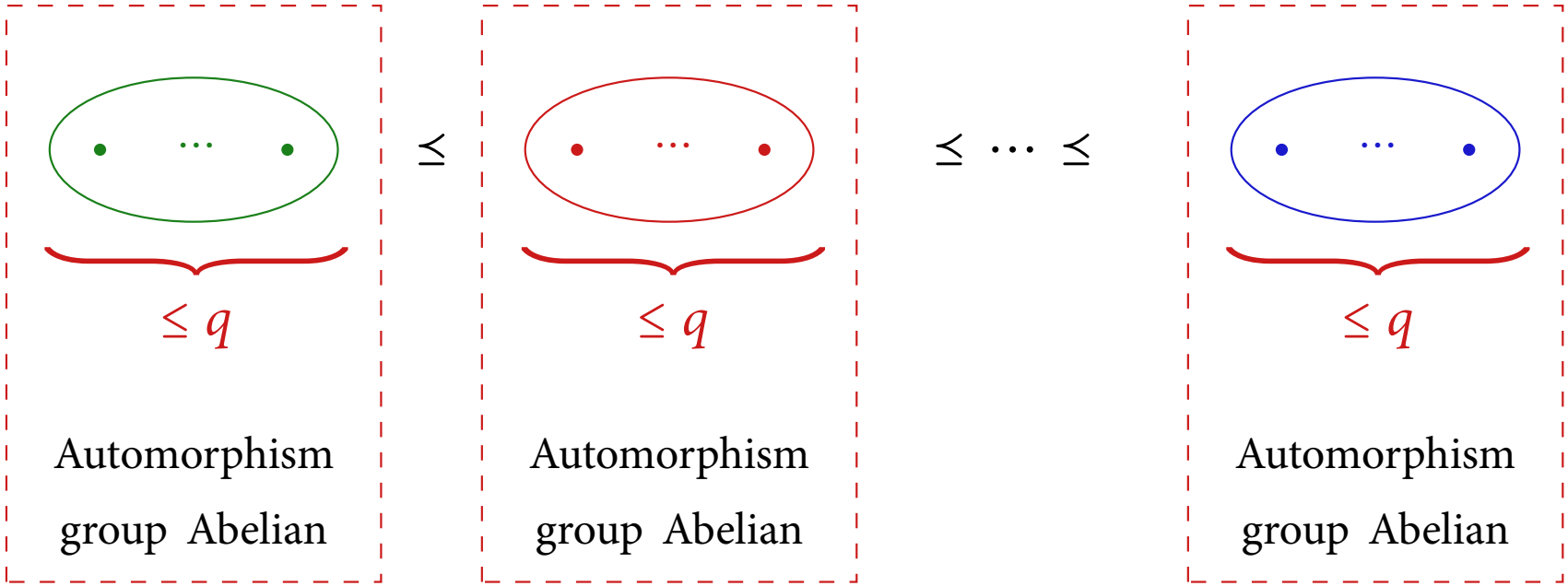


The isomorphism problem for q-bounded structures is solvable in PTIME, for any fixed $q \in \mathbb{N}$.

In general, such isomorphism problems are not (FP+C)-definable.
CFI-graphs (over ordered input graphs) are 2-bounded structures.

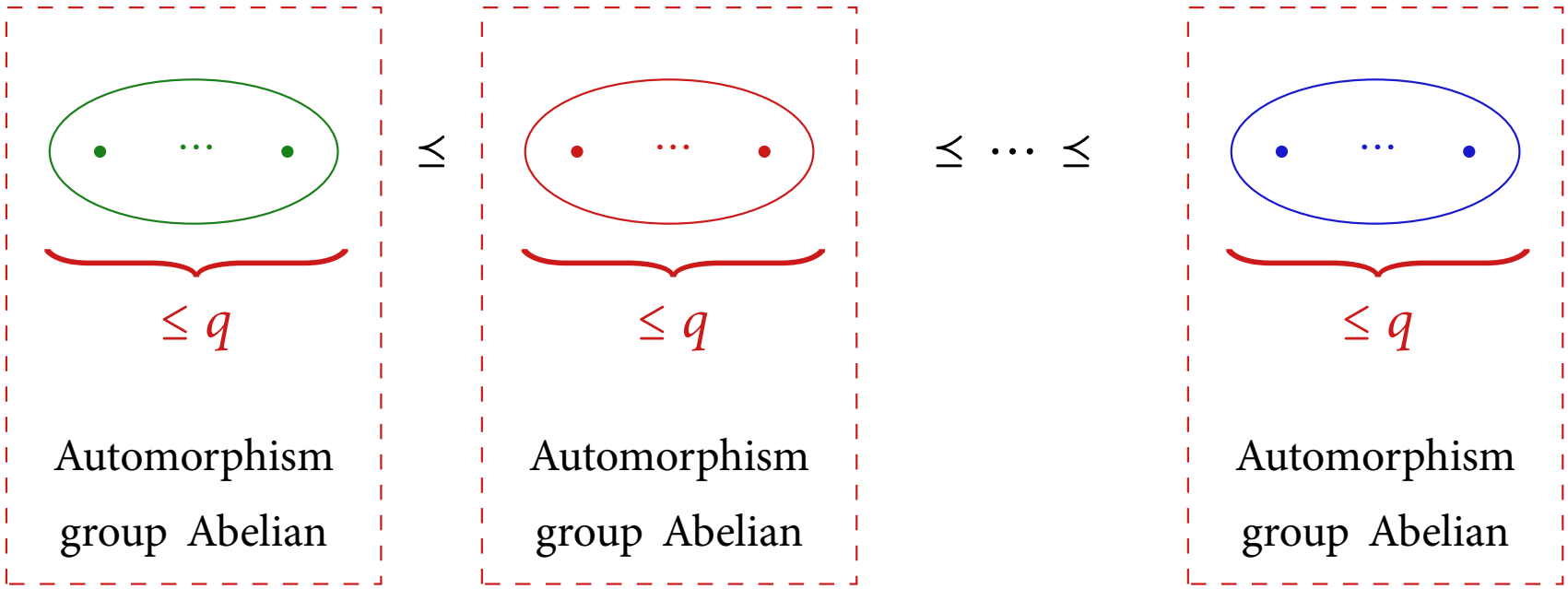
Challenge. Find CPT-canonization procedures for q-bounded structures.

Structures of bounded colour class size



We have CPT-canonization for any such class provided that the symmetry groups on the colour classes are Abelian. (Abu Zaid, EG, Grohe, Pakusa 2014)

Structures of bounded colour class size



We have CPT-canonization for any such class provided that the symmetry groups on the colour classes are Abelian. (Abu Zaid, EG, Grohe, Pakusa 2014)

In particular this holds for 2-bounded structures, which also answers an open problem posed by Blass, Gurevich, and Shelah:
The isomorphism problem for multipedes is solvable in CPT.

Cyclic equation systems

Linear equation systems over an ordered set of variables can be solved in (FP+C). But classical solution algorithms require choice, and cannot be carried out in (FP+C) for unordered sets of variables.

Intermediate class: Cyclic equation systems (CES) over rings $\mathbb{Z}_{p^k}$

- pre-order $\preceq$ on variables
- $\preceq$-equivalent variables are related by equations $x + a = y$
- fixing value of x in a solution fixes values of all variables in the same $\preceq$-class

Theorem. Solvability of CES is definable in Choiceless Polynomial Time.

Cyclic equation systems

Linear equation systems over an ordered set of variables can be solved in (FP+C). But classical solution algorithms require choice, and cannot be carried out in (FP+C) for unordered sets of variables.

Intermediate class: Cyclic equation systems (CES) over rings $\mathbb{Z}_{p^k}$

- pre-order $\preceq$ on variables
- $\preceq$-equivalent variables are related by equations $x + a = y$
- fixing value of x in a solution fixes values of all variables in the same $\preceq$-class

Theorem. Solvability of CES is definable in Choiceless Polynomial Time.

Solving CES is an essential ingredient in the CPT-canonization procedure for q-bounded structures with Abelian colours.

For details, see PhD-Thesis of Wied Pakusa (Aachen, 2015)

Symmetric circuits

A circuit family $(C_n)_{n\in\mathbb{N}}$ decides a property of finite τ-structures if C_n takes as inputs the truth values of atomic τ-formulae of structures with universe $[n] = \{0, \ldots, n-1\}$, and if it is invariant under isomorphisms.

Invariance: Any permutation of $[n]$ induces a permutation of the input gates of C_n. The result of the computation of C_n must be invariant under this.

Symmetric circuits

A circuit family $(C_n)_{n\in\mathbb{N}}$ decides a property of finite τ-structures if C_n takes as inputs the truth values of atomic τ-formulae of structures with universe $[n] = \{0, \ldots, n-1\}$, and if it is invariant under isomorphisms.

Invariance: Any permutation of $[n]$ induces a permutation of the input gates of C_n. The result of the computation of C_n must be invariant under this.

Translate any formula from FO or LFP into a circuit family $C = (C_n)_{n\in\mathbb{N}}$. Then this sequence is

p-uniform: The circuit C_n is polynomial-time computable in n
symmetric: Every permutation of $[n]$ induces an automorphism of C_n

Symmetric circuits are always invariant. The converse is not true.

Symmetric threshold circuits

For logics with counting it is natural to consider circuits with threshold gates.

The extension by threshold gates does not increase the power of polynomial-size circuits. But it can make a difference for restricted classes, such as bounded-depth circuits or symmetric circuits.

Symmetric threshold circuits

For logics with counting it is natural to consider circuits with threshold gates.

The extension by threshold gates does not increase the power of polynomial-size circuits. But it can make a difference for restricted classes, such as bounded-depth circuits or symmetric circuits.

Every formula in (FP+C) can be translated into a p-uniform sequence of symmetric threshold circuits.

Question. Can this also be done for Choiceless Polynomial Time?

Is there a circuit model for CPT?

Theorem (Anderson, Dawar)
p-uniform symmetric threshold circuits are equivalent to (FP+C).

Thus, translations from Choiceless Polynomial Time into equivalent sequences of symmetric threshold circuits are not p-uniform.

Is there a circuit model for CPT?

Theorem (Anderson, Dawar)
p-uniform symmetric threshold circuits are equivalent to (FP+C).

Thus, translations from Choiceless Polynomial Time into equivalent sequences of symmetric threshold circuits are not p-uniform.

To put it differently, p-uniform translations form CPT into threshold circuits must break symmetry in some way. But how?

Challenge: Find a circuit model for CPT, based on a weaker notion of symmetry.

Anderson and Dawar suggest to require induced automorphisms of the circuits only for certain subgroups of the symmetric group on the input universe.

Choiceless polynomial time via interpretations

Idea: Replace the machinery of BGS-terms computing hereditarily finite sets by first-order interpretations.
Instead of a sequence of hereditarily finite sets, a computation then is a sequence of finite structures obtained by repeated application of a fixed first-order interpretation.

Choiceless polynomial time via interpretations

Idea: Replace the machinery of BGS-terms computing hereditarily finite sets by first-order interpretations.
Instead of a sequence of hereditarily finite sets, a computation then is a sequence of finite structures obtained by repeated application of a fixed first-order interpretation.

Interpretations: A FO$[\tau, \sigma]$-interpretation is a sequence

$$I = (\delta(\overline{x}), \varepsilon(\overline{x}, \overline{y}), (\varphi_R(\overline{x}_1, \ldots, \overline{x}_{s(R)})_{R \in \sigma})$$

of FO$[\tau]$-formulae. It maps a τ-structure $\mathfrak{A}$ to a σ-structure

$$I(\mathfrak{A}) = (\delta^{\mathfrak{A}}, (\varphi_R^{\mathfrak{A}})_{R \in \sigma})/\varepsilon^{\mathfrak{A}}$$

Notice that interpretations may change the size of the structures.

Computing by interpretations

Polynomial Time Interpretation Logic PIL$^-$: $\Pi = (I_{\text{init}}, I_{\text{step}}, \varphi_{\text{halt}}, \varphi_{\text{out}})$

- I_{init} is a FO$[\tau, \sigma]$-interpretation defining from the input structure $\mathfrak{A}$ an initial state $\mathfrak{A}_0 := I_{\text{init}}(\mathfrak{A})$
- I_{step} is a FO$[\sigma, \sigma]$-interpretation defining from a state $\mathfrak{A}_i$ the next state $\mathfrak{A}_{i+1} := I_{\text{step}}(\mathfrak{A}_i)$
- the run $\mathfrak{A}_0, \mathfrak{A}_1, \ldots$ of Π in $\mathfrak{A}$ teminates at the first state $\mathfrak{A}_n$ with $\mathfrak{A}_n \models \varphi_{\text{halt}}$
- Π accepts $\mathfrak{A}$ if the run terminates at state $\mathfrak{A}_n$ with $\mathfrak{A}_n \models \varphi_{\text{out}}$

Explicit polynomial bounds on the length of the run and the size of all states are needed to get a polynomial-time variant.

Computing by interpretations

Polynomial Time Interpretation Logic PIL$^-$: $\Pi = (I_{\text{init}}, I_{\text{step}}, \varphi_{\text{halt}}, \varphi_{\text{out}})$

- I_{init} is a FO$[\tau, \sigma]$-interpretation defining from the input structure $\mathfrak{A}$ an initial state $\mathfrak{A}_0 := I_{\text{init}}(\mathfrak{A})$
- I_{step} is a FO$[\sigma, \sigma]$-interpretation defining from a state $\mathfrak{A}_i$ the next state $\mathfrak{A}_{i+1} := I_{\text{step}}(\mathfrak{A}_i)$
- the run $\mathfrak{A}_0, \mathfrak{A}_1, \ldots$ of Π in $\mathfrak{A}$ teminates at the first state $\mathfrak{A}_n$ with $\mathfrak{A}_n \models \varphi_{\text{halt}}$
- Π accepts $\mathfrak{A}$ if the run terminates at state $\mathfrak{A}_n$ with $\mathfrak{A}_n \models \varphi_{\text{out}}$

Explicit polynomial bounds on the length of the run and the size of all states are needed to get a polynomial-time variant.

PIL$^-$ is equivalent to CPT without counting.

PIL: Use (FO+H)-interpretations, where H is the Härtig quantifier.

Equivalence

Theorem CPT ≡ PIL (EG, Kaiser, Pakusa, Schalthöfer, 2015)

Equivalence

Theorem CPT ≡ PIL (EG, Kaiser, Pakusa, Schalthöfer, 2015)

Natural fragments of Interpretation Logic characterize logics that have previously arisen in the quest for a logic for PTIME.

Equivalence

Theorem CPT ≡ PIL (EG, Kaiser, Pakusa, Schalthöfer, 2015)

Natural fragments of Interpretation Logic characterize logics that have previously arisen in the quest for a logic for PTIME.

- two-dimensional PIL is as powerful as full PIL and CPT .

Equivalence

Theorem CPT ≡ PIL (EG, Kaiser, Pakusa, Schalthöfer, 2015)

Natural fragments of Interpretation Logic characterize logics that have previously arisen in the quest for a logic for PTIME.

- two-dimensional PIL is as powerful as full PIL and CPT .
- one-dimensional PIL ≡ (FP+C)

Equivalence

Theorem CPT ≡ PIL (EG, Kaiser, Pakusa, Schalthöfer, 2015)

Natural fragments of Interpretation Logic characterize logics that have previously arisen in the quest for a logic for PTIME.

- two-dimensional PIL is as powerful as full PIL and CPT .
- one-dimensional PIL ≡ (FP+C)
- one-dimensional PIL without counting corresponds to PFP $|_{\text{PTIME}}$, which is equivalent to LFP if, and only if PTIME = PSPACE.

Equivalence

Theorem CPT ≡ PIL (EG, Kaiser, Pakusa, Schalthöfer, 2015)

Natural fragments of Interpretation Logic characterize logics that have previously arisen in the quest for a logic for PTIME.

- two-dimensional PIL is as powerful as full PIL and CPT .
- one-dimensional PIL ≡ (FP+C)
- one-dimensional PIL without counting corresponds to PFP $|_{\text{PTIME}}$, which is equivalent to LFP if, and only if PTIME = PSPACE.
- PIL without counting and without congruences is equivalent to $\text{while}_{\text{new}}|_{\text{PTIME}}$.

Equivalence

Theorem CPT ≡ PIL (EG, Kaiser, Pakusa, Schalthöfer, 2015)

Natural fragments of Interpretation Logic characterize logics that have previously arisen in the quest for a logic for PTIME.

- two-dimensional PIL is as powerful as full PIL and CPT .
- one-dimensional PIL ≡ (FP+C)
- one-dimensional PIL without counting corresponds to PFP $|_{\text{PTIME}}$, which is equivalent to LFP if, and only if PTIME = PSPACE.
- PIL without counting and without congruences is equivalent to $\text{while}_{\text{new}}|_{\text{PTIME}}$.
- On structures of bounded colour-class size, PIL without congruences can be simulated by CPT-programs that access only sets of bounded rank. By Dawar, Richerby, and Rossman, this is too weak for the CFI-query.

The summation problem for Abelian groups and semigroups

Given: A finite (semi)group $(G, +, 0)$ and a subset $X \subset G$.

Question: Determine $\sum X$.

The summation problem for Abelian groups and semigroups

Given: A finite (semi)group $(G, +, 0)$ and a subset $X \subset G$.
Question: Determine $\sum X$.

Algorithmically this a trivial problem. Logically, it is much more delicate, if the semigroup does not come with a linear order, and thus without a canonical way to process the elements of X one by one.

Actually it has been proposed as a candidate for separating CPT from PTIME:

> *"This is the most basic problem I can think of that appears difficult for CPT but is obviously polynomial time. I don't even know the answer when G is an abelian group, or even a direct product of cyclic groups $\mathbb{Z}_2$."* *(Ben Rossman, 2005)*

Definability of the summation problem for semigroups

Actually the summation problem provides yet another example for the surprising expressive power of fixed-point logic with counting.

Theorem. (Abu Zaid, Dawar, EG, Pakusa, 2017)
The summation problem for Abelian semigroups is definable in (FP+C).

For Abelian groups, the summation problem is also definable in the extension of first-order logic by a solvability operator for equation systems over finite rings.

On the other side, the summation problem is not definable in LFP, or even in CPT without counting, not even in the case of Abelian groups. For these results we use probabilistic arguments.

Definability in fixed-point logic with counting

For all $g, h, z \in (G, +, 0)$, let

$$T_i(g,h) := \{(x_1, \ldots, x_i) \in X^i : g + x_1 + \cdots + x_i = h \text{ and } x_k \neq x_\ell \text{ for } k \neq \ell\}$$
$$R_i^{\neq z}(g,h) := \{(x_1, \ldots, x_i) \in T_i(g,h) : x_j \neq z \text{ for all } j \leq i\}$$
$$t_i(g,h) = |T_i(g,h)|, \quad r_i^{\neq z}(g,h) = |R_i^{\neq z}(g,h)|.$$

Let $n = |X|$. Then $\sum X = y$ if, and only if, $t_n(0,y) > 0$.

Definability in fixed-point logic with counting

For all $g, h, z \in (G, +, 0)$, let

$$T_i(g,h) := \{(x_1, \ldots, x_i) \in X^i : g + x_1 + \cdots + x_i = h \text{ and } x_k \neq x_\ell \text{ for } k \neq \ell\}$$
$$R_i^{\neq z}(g,h) := \{(x_1, \ldots, x_i) \in T_i(g,h) : x_j \neq z \text{ for all } j \leq i\}$$
$$t_i(g,h) = |T_i(g,h)|, \quad r_i^{\neq z}(g,h) = |R_i^{\neq z}(g,h)|.$$

Let $n = |X|$. Then $\sum X = y$ if, and only if, $t_n(0,y) > 0$.

Thus, it suffices to provide inductive definitions of the values $t_i(g,h)$ and $r_i^{\neq z}(g,h)$ for all $g, h, z \in G$. For $i = 0$ this is trivial, and

$$t_{i+1}(g,h) = \textstyle\sum_{x \in X} r_i^{\neq x}(g + x, h)$$
$$r_{i+1}^{\neq z}(g,h) = t_{i+1}(g,h) - (i+1) r_i^{\neq z}(g + z, h)$$

Definability in fixed-point logic with counting

For all $g, h, z \in (G, +, 0)$, let

$$T_i(g,h) := \{(x_1, \ldots, x_i) \in X^i : g + x_1 + \cdots + x_i = h \text{ and } x_k \neq x_\ell \text{ for } k \neq \ell\}$$
$$R_i^{\neq z}(g,h) := \{(x_1, \ldots, x_i) \in T_i(g,h) : x_j \neq z \text{ for all } j \leq i\}$$
$$t_i(g,h) = |T_i(g,h)|, \quad r_i^{\neq z}(g,h) = |R_i^{\neq z}(g,h)|.$$

Let $n = |X|$. Then $\sum X = y$ if, and only if, $t_n(0,y) > 0$.

Thus, it suffices to provide inductive definitions of the values $t_i(g,h)$ and $r_i^{\neq z}(g,h)$ for all $g, h, z \in G$. For $i = 0$ this is trivial, and

$$t_{i+1}(g,h) = \textstyle\sum_{x \in X} r_i^{\neq x}(g + x, h)$$
$$r_{i+1}^{\neq z}(g,h) = t_{i+1}(g,h) - (i+1) r_i^{\neq z}(g + z, h)$$

This equation system can be translated into an (FP+C)-definition of the values $t_i(g,h)$ and $r_i^{\neq z}(g,h)$.

A limit law for direct products of cyclic groups

Let $\tau = \{X_1, \ldots, X_\ell\}$ be a relational vocabulary.

$S_n(\mathbb{Z}_p)$: probability space of all expansions of $(\mathbb{Z}_p)^n$ by relations from τ, with uniform probability distribution.

For every sentence ψ of vocabulary $\{+, 0\} \cup \tau$, let

$$\mu_n(\psi) := \Pr_{\mathfrak{A} \in S_n(\mathbb{Z}_p)}[\mathfrak{A} \models \psi]$$

Theorem. For every sentence $\psi \in L^\omega_{\infty\omega}$ of vocabulary $\{+, 0\} \cup \tau$, $\lim_{n\to\infty} \mu_n(\psi) = r/2^\ell$, for $\ell = |\tau|$ and some $r \leq 2^\ell$.

A limit law for direct products of cyclic groups

Let $\tau = \{X_1, \ldots, X_\ell\}$ be a relational vocabulary.

$S_n(\mathbb{Z}_p)$: probability space of all expansions of $(\mathbb{Z}_p)^n$ by relations from τ, with uniform probability distribution.

For every sentence ψ of vocabulary $\{+, 0\} \cup \tau$, let

$$\mu_n(\psi) := \Pr_{\mathfrak{A} \in S_n(\mathbb{Z}_p)}[\mathfrak{A} \models \psi]$$

Theorem. For every sentence $\psi \in L^\omega_{\infty\omega}$ of vocabulary $\{+, 0\} \cup \tau$,

$\lim_{n\to\infty} \mu_n(\psi) = r/2^\ell$, for $\ell = |\tau|$ and some $r \leq 2^\ell$.

The proof uses an appropriate variant of extension axioms, adapted to vector spaces $(\mathbb{Z}_p)^n$.

A limit law for direct products of cyclic groups

Let $\tau = \{X_1, \ldots, X_\ell\}$ be a relational vocabulary.

$S_n(\mathbb{Z}_p)$: probability space of all expansions of $(\mathbb{Z}_p)^n$ by relations from τ, with uniform probability distribution.

For every sentence ψ of vocabulary $\{+, 0\} \cup \tau$, let

$$\mu_n(\psi) := \Pr_{\mathfrak{A} \in S_n(\mathbb{Z}_p)}[\mathfrak{A} \models \psi]$$

Theorem. For every sentence $\psi \in L^\omega_{\infty\omega}$ of vocabulary $\{+, 0\} \cup \tau$, $\lim_{n\to\infty} \mu_n(\psi) = r/2^\ell$, for $\ell = |\tau|$ and some $r \leq 2^\ell$.

The proof uses an appropriate variant of extension axioms, adapted to vector spaces $(\mathbb{Z}_p)^n$.

In fact, using the machinery of strong extension axioms, due to Blass, Gurevich, and Shelah, the limit law can be generalized to CPT without counting.

Non-definability without counting

Let $\varphi(x)$ define the Abelian group summation problem:

$$(G, +, 0, X) \models \varphi(h) \iff \sum X = h.$$

Non-definability without counting

Let $\varphi(x)$ define the Abelian group summation problem:

$$(G, +, 0, X) \models \varphi(h) \iff \sum X = h.$$

Then $\psi := \exists y(\varphi(y) \wedge Xy \wedge X0)$ says that both 0 and $\sum X$ are in X.

Let $G = (\mathbb{Z}_2)^n$. For a random $X \subseteq G$ all $g \in G$ have equal probability to be the sum of all elements of X. The probability that this sum is itself an element of X converges to 1/2. Thus $\lim_{n\to\infty} \mu_n(\psi) = 1/4$.

Non-definability without counting

Let $\varphi(x)$ define the Abelian group summation problem:

$$(G, +, 0, X) \models \varphi(h) \iff \sum X = h.$$

Then $\psi := \exists y(\varphi(y) \wedge Xy \wedge X0)$ says that both 0 and $\sum X$ are in X.

Let $G = (\mathbb{Z}_2)^n$. For a random $X \subseteq G$ all $g \in G$ have equal probability to be the sum of all elements of X. The probability that this sum is itself an element of X converges to 1/2. Thus $\lim_{n\to\infty} \mu_n(\psi) = 1/4$.

But for $\tau = \{X\}$, the asymptotic probability of every sentence $\psi \in L^{\omega}_{\infty\omega}$ is, according to our limit law, either 0, 1, or 1/2. Contradiction.

Theorem. The Abelian group summation problem is not definable in $L^{\omega}_{\infty\omega}$.

This extends to CPT without counting.

Challenges for future research

A characterization of CPT without explicit polynomial bounds.

Definitions of CPT are based on explicit polynomial bounds on length of iterations, number of active elements or size of interpreted structures. Find a presentation that guarantees this by construction, as in fixed-point logics.
Solved by Svenja Schalthöfer (2017)

Symmetric circuits for CPT

Find a circuit model for CPT. Understand better the symmetries inherent in CPT-computations.

CFI-graphs

Can the isomorphism problem for CFI-graphs constructed from unordered input graphs be solved in CPT ?
Actually this might be a candidate for separating CPT from PTIME

Challenges for future research

Constraint Satisfaction Problems (CSP)

Which CSPs are solvable in CPT?
The CSPs solvable in (FP+C) are those with a property called bounded width. CPT can solve certain CSPs of unbounded width, such as cyclic equation systems (CES), which belong to the class of CSPs with Maltsev polymorphisms.

Can CPT solve all CSPs with Maltsev polymorphisms? This would imply that isomorphism of graphs with bounded colour class size is in CPT.

Challenges for future research

Choiceless Polynomial-Time versus Rank Logic

Besides CPT, logics with operators from linear algebra, such as the rank logic FPR*, seem to be the most prominent candidates for a logic for PTIME.
The relationship between CPT and FPR* is unclear but cyclic equation systems (CES) over rings might separate the two logics.

Conjecture. Solvability of CES over $\mathbb{Z}_4$ is definable in CPT but not in FPR*.