Linear Algebra and the Quest for a Logic for Polynomial Time

Erich Grädel

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 structures, there is a sentence $\psi \in L$ such that $S = \{\mathfrak{A} \in \mathrm{Fin} : \mathfrak{A} \models \psi\}$.

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 structures, there is a sentence $\psi \in L$ such that $S = \{\mathfrak{A} \in \mathrm{Fin} : \mathfrak{A} \models \psi\}$.

The precise definition of what may constitute a logic for PTIME is a bit more subtle. It includes certain 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.

Least fixed point logic LFP

Syntax. LFP extends FO by fixed point rule:

- For every formula $\psi(T, x_1 \dots x_k) \in \mathrm{LFP}[\tau \cup \{T\}]$,
  T k-ary relation variable, occuring only positive in ψ,
  build formulae $[\mathbf{lfp}\, T\overline{x} \,.\, \psi](\overline{x})$ and $[\mathbf{gfp}\, T\overline{x} \,.\, \psi](\overline{x})$

Semantics. On τ-structure $\mathfrak{A}$, $\psi(T, \overline{x})$ defines monotone operator

$$\begin{aligned} \psi^{\mathfrak{A}} : \mathcal{P}(A^k) &\longrightarrow \mathcal{P}(A^k) \\ T &\longmapsto \{\overline{a} : (\mathfrak{A}, T) \vDash \psi(T, \overline{a})\} \end{aligned}$$

- $\mathfrak{A} \vDash [\mathbf{lfp}\, T\overline{x} \,.\, \psi(T, \overline{x})](\overline{a}) \;:\Longleftrightarrow\; \overline{a} \in \mathbf{lfp}(\psi^{\mathfrak{A}})$
  $\mathfrak{A} \vDash [\mathbf{gfp}\, T\overline{x} \,.\, \psi(T, \overline{x})](\overline{a}) \;:\Longleftrightarrow\; \overline{a} \in \mathbf{gfp}(\psi^{\mathfrak{A}})$

LFP and polynomial time

Theorem (Immerman, Vardi)
On ordered finite structures, LFP captures PTIME.

On arbitrary finite structures, LFP can express certain PTIME-complete problems (such as winning regions of reachability games), but fails to express all of PTIME.

LFP and polynomial time

Theorem (Immerman, Vardi)
On ordered finite structures, LFP captures PTIME.

On arbitrary finite structures, LFP can express certain PTIME-complete problems (such as winning regions of reachability games), but fails to express all of PTIME.

LFP is unable to count.

For instance the class of graphs with an even number of vertices is not LFP-definable.

LFP and polynomial time

Theorem (Immerman, Vardi)
On ordered finite structures, LFP captures PTIME.

On arbitrary finite structures, LFP can express certain PTIME-complete problems (such as winning regions of reachability games), but fails to express all of PTIME.

LFP is unable to count.

For instance the class of graphs with an even number of vertices is not LFP-definable.

Immerman suggested to extend fixed-point logics by a counting mechanism.

Fixed-point logic with counting

(FP + C): A 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 arithmetic operations $+$, $\cdot$ and $<$ are available, but variables must be explicitely bounded and only take 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 $\varphi(a)$.

Least or inflationary fixed point operator defining formulae of the form

$$[\mathbf{fp}\, R\overline{x\mu}_{\leq \bar{t}} \,.\, \psi(R, \overline{x}, \overline{\mu})](\overline{y}, \overline{\nu}).$$

Infinitary logic with counting

(FP + C) can be embedded into the infinitary logic $C^{\omega}_{\infty\omega}$, which extends first-order logic by allowing

- counting quantifiers $\exists^i x$: there exist at least i elements x such that…
- infinitary conjunctions and disjunctions:
  $\bigvee \Phi$ and $\bigwedge \Phi$ for any set Φ of formulae.
- only finitely many distinct variables in each formula.

Infinitary logic with counting

(FP + C) can be embedded into the infinitary logic $C^{\omega}_{\infty\omega}$, which extends first-order logic by allowing

- counting quantifiers $\exists^i x$: there exist at least i elements x such that…
- infinitary conjunctions and disjunctions:
  $\bigvee \Phi$ and $\bigwedge \Phi$ for any set Φ of formulae.
- only finitely many distinct variables in each formula.

$C^{k}_{\infty\omega}$ is the k-variable-fragment of $C^{\omega}_{\infty\omega}$.

Infinitary logic with counting

(FP + C) can be embedded into the infinitary logic $C^{\omega}_{\infty\omega}$, which extends first-order logic by allowing

- counting quantifiers $\exists^i x$: there exist at least i elements x such that…
- infinitary conjunctions and disjunctions:
  $\bigvee \Phi$ and $\bigwedge \Phi$ for any set Φ of formulae.
- only finitely many distinct variables in each formula.

$C^{k}_{\infty\omega}$ is the k-variable-fragment of $C^{\omega}_{\infty\omega}$.

Why are we interested in this infinitary logic?

Infinitary logic with counting

(FP + C) can be embedded into the infinitary logic $C^{\omega}_{\infty\omega}$, which extends first-order logic by allowing

- counting quantifiers $\exists^i x$: there exist at least i elements x such that…
- infinitary conjunctions and disjunctions:
  $\bigvee \Phi$ and $\bigwedge \Phi$ for any set Φ of formulae.
- only finitely many distinct variables in each formula.

$C^{k}_{\infty\omega}$ is the k-variable-fragment of $C^{\omega}_{\infty\omega}$.

Why are we interested in this infinitary logic?

$C^{k}_{\infty\omega}$-equivalence of finite structures, and hence non-expressibility results for (FP+C) can be proved via appropriate variants of Ehrenfeucht-Fraïssé games, as for instance Hella's k-pebble bijection games.

Fixed-point logic with counting versus polynomial time

Theorem. (FP+C) $\subsetneq$ PTIME (Cai, Fürer, Immerman 1992)

Fixed-point logic with counting versus polynomial time

Theorem. (FP+C) $\subsetneq$ PTIME (Cai, Fürer, Immerman 1992)

It is easy to see that every (FP+C)-definable property of finite structures is decidable in polynomial time.

On the other side, Cai, Fürer, and Immerman constructed sequences $(G_n)_{n\in\omega}$ and $(H_n)_{n\in\omega}$ of graphs such that

(1) There is a class of graphs, that is decidable in polynomial time, which includes all G_n and excludes all H_n

(2) $G_n \equiv^{C^n_{\infty\omega}} H_n$, for all n.

Fixed-point logic with counting versus polynomial time

Theorem. (FP+C) $\subsetneq$ PTIME (Cai, Fürer, Immerman 1992)

It is easy to see that every (FP+C)-definable property of finite structures is decidable in polynomial time.

On the other side, Cai, Fürer, and Immerman constructed sequences $(G_n)_{n\in\omega}$ and $(H_n)_{n\in\omega}$ of graphs such that

(1) There is a class of graphs, that is decidable in polynomial time, which includes all G_n and excludes all H_n

(2) $G_n \equiv^{C^n_{\infty\omega}} H_n$, for all n.

Although the Cai-Fürer-Immerman construction is sophisticated, the property separating the G_n from the H_n seemed somewhat artificial.
Might it be the case that (FP+C) captures all "natural" PTIME-properties of finite structures?

Capturing polynomial time sometimes

Fixed-point logic with counting captures PTIME on certain interesting classes of structures, such as

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

Further (FP+C) captures PTIME almost everywhere, i.e. on a class of random structures with asymptotic probability one. (Hella, Kolaitis, Luosto)

Formula that define matrices

A formula $\varphi(x, y)$ defines, when evaluated on a finite structure $\mathfrak{A}$, a square matrix $M_\varphi^{\mathfrak{A}} = (m_{a,b})_{a,b \in \mathfrak{A}}$, with entries

$$m_{a,b} := \begin{cases} 1 & \text{if } \mathfrak{A} \models \varphi(a,b) \\ 0 & \text{if } \mathfrak{A} \not\models \varphi(a,b) \end{cases}$$

Also more general matrices, for instance with entries in an arbitrary finite commutative ring, can be defined by appropriate (collections of) formulae.

Since we assume our structures to be unordered, these matrices are defined only up to permutations of the rows and columns.

A next frontier: linear algebra

Most concepts in linear algebra can be formulated in terms of matrices.

We are interested in matrix properties and functions on matrices that are

(1) invariant under permutations of the rows and columns

(2) computable in polynomial time

This includes arithmetic operations on matrices, singularity, rank and determinant, characteristic polynomials, minimal polynomials, solvablity of linear equation systems, normal forms, …

Question: Which of these properties and operations are definable in (FP+C)?

The answer may depend on the underlying ring.
One may consider $\mathbb{Q}$, $\mathbb{Z}$, finite fields, or arbitrary finite commutative rings.

Linear algebra in (FP+C)

Actually a fair amount of linear algebra can be defined in (FP + C):

- matrix addition and matrix multiplication
- matrix exponentiation M^k (with k given in binary notation)
- (non-)singularity of matrices
- determinant (over finite fields, $\mathbb{Q}$, and $\mathbb{Z}$)
- characteristic polynomials
- minimal polynomials (on fields)
- matrix rank over $\mathbb{Q}$

(Blass, Gurevich, Shelah, Rossman, Dawar, Grohe, Holm, Laubner, Kopczynski, Pakusa, ….)

Linear algebra outside (FP+C)

However, there are some fundamental polynomial-time properties in linear algebra that are not definable in (FP+C)

- solvability of linear equation systems, over any finite commutative ring
- the rank of a matrix (over a finite field)
- similarity of matrices (over a finite field)
- rank equality (over a finite field)

Linear algebra outside (FP+C)

However, there are some fundamental polynomial-time properties in linear algebra that are not definable in (FP+C)

- solvability of linear equation systems, over any finite commutative ring
- the rank of a matrix (over a finite field)
- similarity of matrices (over a finite field)
- rank equality (over a finite field)

These are also sources of new operators to extend the logics.

From graphs to systems of linear equations

Take a connected 3-regular graph G, and a finite commutative ring R.

For every vertex u and $a \in R$, we define the equation system G_a^u, with:

Variables: $x_r^{v,e}$, for every $r \in R$, every v and every edge e incident to v.

Edge equations: $x_r^{u,e} + x_s^{v,e} = r + s$

for every edge $e = \{u, v\}$ and all $r, s \in R$,

Vertex equations: $$x_r^{v,e} + x_s^{v,f} + x_t^{v,g} = \begin{cases} r + s + t & \text{for } v \neq u \\ r + s + t + a & \text{for } v = u \end{cases}$$

for all vertices v, with incident edges e, f, g, and all $r, s, t \in R$

Solvability of the equation systens G_a^u

Proposition. The equation system G_a^u is solvable for $a = 0$.

The equation system G_a^u consists of

Edge equations: $x_r^{u,e} + x_s^{v,e} = r + s$

Vertex equations: $x_r^{v,e} + x_s^{v,f} + x_t^{v,g} = \begin{cases} r+s+t & \text{for } v \neq u \\ r+s+t+a & \text{for } v = u \end{cases}$

For $a = 0$, all vertex equations have the form

$$x_r^{v,e} + x_s^{v,f} + x_t^{v,g} = r + s + t$$

and the system is solved by the assignment $x_r^{v,e} \mapsto r$ for all v, e, r.

Solvability of the equation systems G_a^u

Proposition. The equation system G_a^u is unsolvable for $a \neq 0$.

Consider the subsystem of equations that involve only variables $x_0^{w,e}$.

Edge equations: $x_0^{u,e} + x_0^{v,e} = 0$

Vertex equations: $x_0^{v,e} + x_0^{v,f} + x_0^{v,g} = \begin{cases} 0 & \text{for } v \neq u \\ a & \text{for } v = u \end{cases}$

Solvability of the equation systems G_a^u

Proposition. The equation system G_a^u is unsolvable for $a \neq 0$.

Consider the subsystem of equations that involve only variables $x_0^{w,e}$.

Edge equations: $x_0^{u,e} + x_0^{v,e} = 0$

Vertex equations: $x_0^{v,e} + x_0^{v,f} + x_0^{v,g} = \begin{cases} 0 & \text{for } v \neq u \\ a & \text{for } v = u \end{cases}$

Here, every variable $x_0^{w,e}$ appears exactly once in an edge equation and exactly once in a vertex equations.

Sum of all edge equations: $\sum_{w,e} x_0^{w,e} = 0$

Sum of all vertex equations: $\sum_{w,e} x_0^{w,e} = a$

Thus, for $a \neq 0$, the subsystem, and hence also G_a^u, is unsolvable.

Equation systems as finite structures

The equation systems G_a^u can be viewed as finite structures

$$G_a^u = (X, (E_r)_{r \in R}, (V_r)_{r \in R})$$

- X is the set of variables occurring in the system
- $E_r = \{(x, y) : x + y = r \text{ is an edge equation in the system}\}$
- $V_r = \{(x, y, z) : x + y + z = r \text{ is a vertex equation in the system}\}$

Clearly the (structures that represent) equation systems G_a^u are first-order interpretable in G.

Proposition. For all nodes u, v of a connected, 3-regular graph G, we have $G_a^u \cong G_a^v$.

Equation systems and the Cai-Fürer-Immerman construction

Theorem. (Atserias, Bulatov, Dawar) , (Dawar and Richerby), (Cai, Fürer, Immerman) Let G be a connected, 3-regular graph of tree width $> k$. Then for every node u and every ring element $a \in R$

$$G_a^u \equiv^{C^k_{\infty\omega}} G_0^u$$

Equation systems and the Cai-Fürer-Immerman construction

Theorem. (Atserias, Bulatov, Dawar) , (Dawar and Richerby), (Cai, Fürer, Immerman) Let G be a connected, 3-regular graph of tree width $> k$. Then for every node u and every ring element $a \in R$

$$G_a^u \equiv^{C_{\infty\omega}^k} G_0^u$$

This can either be shown directly by playing the k-pebble bijection game on G_a^u and G_0^u, or by an interpretation from the Cai-Fürer-Immerman graphs.

Equation systems and the Cai-Fürer-Immerman construction

Theorem. (Atserias, Bulatov, Dawar) , (Dawar and Richerby), (Cai, Fürer, Immerman) Let G be a connected, 3-regular graph of tree width $> k$. Then for every node u and every ring element $a \in R$

$$G_a^u \equiv^{C_{\infty\omega}^k} G_0^u$$

This can either be shown directly by playing the k-pebble bijection game on G_a^u and G_0^u, or by an interpretation from the Cai-Fürer-Immerman graphs.

So actually the Cai-Fürer-Immerman property of graphs is closely related to an extremely natural problem: the solvability of linear equation systems.

Solvability of linear equation systems

Let $S(R)$ be the collection finite structures representing a linear equation system over the commutative ring R.

$$\mathrm{Solv}(R) := \{\mathfrak{A} \in S(R) : \text{ the system over } R \text{ represented by } \mathfrak{A} \text{ is solvable}\}$$

Theorem. (Atserias, Bulatov, Dawar)
For any finite commutative ring R, $\mathrm{Solv}(R) \notin (\mathrm{FP} + \mathrm{C})$

Solvability of linear equation systems

Let $S(R)$ be the collection finite structures representing a linear equation system over the commutative ring R.

$$\mathrm{Solv}(R) := \{\mathfrak{A} \in S(R) : \text{ the system over } R \text{ represented by } \mathfrak{A} \text{ is solvable}\}$$

Theorem. (Atserias, Bulatov, Dawar)
For any finite commutative ring R, $\mathrm{Solv}(R) \notin (\mathrm{FP} + \mathrm{C})$

Corollary. The rank of matrices over a finite field is not (FP + C)-definable.

An equation system $Ax = b$ over a field is solvable iff $\mathrm{rk}(A) = \mathrm{rk}(A|b)$. Hence definability of matrix rank would imply definability of Solv.

Fixed-point logic with rank

(FP + rk): Extend (two-sorted) fixed-point logic by rank operators, where $\mathrm{rk}_q\varphi$ defines the rank (over the field $\mathbb{F}_q$) of the matrix defined by φ.

Rank is more general than counting:

$$\#_x\varphi(x) = \mathrm{rk}_2(x = y \wedge \varphi(x))$$

Open problem. Does (FP + rk) capture PTIME ?

All known problems on fields that are in PTIME but not in (FP+C) are definable in (FP + rk).

Operators for finite commutative rings

On rings that are not fields, the situation is more complicated.

Theorem. (Arvind, Vijayaraghavan)

For any finite commutative ring R, $\mathrm{Solv}(R) \in$ PTIME

Operators for finite commutative rings

On rings that are not fields, the situation is more complicated.

Theorem. (Arvind, Vijayaraghavan)
For any finite commutative ring R, Solv$(R) \in$ PTIME

There are several different notions of matrix ranks over rings: row rank, column rank, McCoy rank, inner rank .... It is unclear whether any of these notions is of help for solving linear equation systems on finite rings.

It is also unclear whether ranks over finite rings are PTIME-computable.

Hence, for appropriate rings R, Solv(R) might be a candidate for separating (FP+rk) from PTIME.

But we can also add the solvability of linear equation systems directly to FP or to (FP+C) to get logic (FP + slv). Or (FP + rk + slv) ...

Logics with solvability operators

Logical reducibility among solvability problems.

Solvability of linear equations systems over arbitrary Abelian groups or rings reduces, via LFP-reductions, to solvability over commutative rings, and to local rings.

Normal forms for logics with solvablity operators over finite fields

Every formula in (FO + slv_F) is equivalent to a formula of form

$$\mathrm{slv}(\overline{x}, \overline{y})[\varphi_M(\overline{x}, \overline{\ }), \mathbf{1}] \quad \text{with } \varphi_M \text{ quantifier-free}$$

(Dawar,Grädel, Holm, Kopczynski, Pakusa)

The permutation group membership problem (GM)

Given: Permutations $\pi_1, \ldots, \pi_k$ and π on a set Ω

Question: Is $\pi \in \langle \pi_1, \ldots, \pi_k \rangle$?

The permutation group membership problem (GM)

Given: Permutations $\pi_1, \ldots, \pi_k$ and π on a set Ω

Question: Is $\pi \in \langle \pi_1, \ldots, \pi_k \rangle$?

Theorem (Babai, Luks, Seress) GM and #GM are in PTIME

The permutation group membership problem (GM)

Given: Permutations $\pi_1, \ldots, \pi_k$ and π on a set Ω

Question: Is $\pi \in \langle \pi_1, \ldots, \pi_k \rangle$?

Theorem (Babai, Luks, Seress) GM and #GM are in PTIME

All solvability problems for linear equation systems over fields, rings, and Abelian groups, as well as matrix rank over fields, reduce to GM and #GM.

Theorem

$$\begin{array}{ccc} \mathrm{Solv(R)} & \overset{\mathrm{FO}}{\longmapsto} & \mathrm{GM}\ (\pi \in \langle \pi_1, \ldots, \pi_k \rangle ?) \\ \mathrm{rk(F)} & \overset{\mathrm{FO}}{\longmapsto} & \#\mathrm{GM}(\text{Compute: } |\langle \pi_1, \ldots, \pi_k \rangle|) \end{array}$$

The permutation group membership problem (GM)

Given: Permutations $\pi_1, \ldots, \pi_k$ and π on a set Ω

Question: Is $\pi \in \langle \pi_1, \ldots, \pi_k \rangle$?

Theorem (Babai, Luks, Seress) GM and #GM are in PTIME

All solvability problems for linear equation systems over fields, rings, and Abelian groups, as well as matrix rank over fields, reduce to GM and #GM.

Theorem

$$\begin{array}{ccc} \mathrm{Solv(R)} & \overset{\mathrm{FO}}{\longmapsto} & \mathrm{GM}\ (\pi \in \langle \pi_1, \ldots, \pi_k \rangle ?) \\ \mathrm{rk(F)} & \overset{\mathrm{FO}}{\longmapsto} & \#\mathrm{GM}(\text{Compute: } |\langle \pi_1, \ldots, \pi_k \rangle|) \end{array}$$

Can one define a nice logic (FP + GM) based on the permutation group membership problem? Would such a logic capture PTIME?

Choiceless Polynomial Time

introduced by Blass, Gurevich, and Shelah in 1999

BGS-machines: 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

Disallow choices that violate the inherent symmetry of the input structure

compensated by parallelism (explore all possible choices in parallel) and the machinery of set theory (building sets using comprehension terms)

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 operatorion $\varnothing, \in, \bigcup, \ldots$
- a cardinality operator $x \mapsto |x|$ (as a von Neumann ordinal)
- comprehension terms of form $\{t(x) : x \in t' : \varphi(x)\}$

Evaluation: $[\![t(\overline{x})]\!]^{\mathfrak{A}} \in \mathrm{HF}(\mathfrak{A}), \qquad [\![\varphi(\overline{x})]\!]^{\mathfrak{A}} \in \{\mathrm{false}, \mathrm{true}\} = \{\varnothing, \{\varnothing\}\}$

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 operatorion $\varnothing, \in, \bigcup, \ldots$
- a cardinality operator $x \mapsto |x|$ (as a von Neumann ordinal)
- comprehension terms of form $\{t(x) : x \in t' : \varphi(x)\}$

Evaluation: $[\![t(\overline{x})]\!]^{\mathfrak{A}} \in \mathrm{HF}(\mathfrak{A}), \qquad [\![\varphi(\overline{x})]\!]^{\mathfrak{A}} \in \{\mathrm{false}, \mathrm{true}\} = \{\varnothing, \{\varnothing\}\}$

For a comprehension term $s(\overline{y}) := \{t(x, \overline{y}) : x \in t'(\overline{y}) : \varphi(x, \overline{y})\}$ and a tuple $\overline{b}$ in $\mathrm{HF}(\mathfrak{A})$, we have

$$[\![s(\overline{b})]\!]^{\mathfrak{A}} := \{[\![t(a, \overline{b}]\!]^{\mathfrak{A}} : a \in [\![t'(\overline{b})]\!]^{\mathfrak{A}} \wedge [\![\varphi(a, \overline{b})]\!]^{\mathfrak{A}} = \mathit{true}\}$$

Definition of choiceless polynomial time

A BGS-programme is a triple $\Pi = (\Pi_{\text{step}}(x), \Pi_{\text{halt}}(x), \Pi_{\text{out}}(x))$ of a BGS-term and two BGS-formulae.

Its computation on a finite structure $\mathfrak{A}$ is a sequence $x_0, x_1, \ldots$ of sets in $\mathrm{HF}(\mathfrak{A})$ with $x_0 = \varnothing$ and $x_{i+1} := [\![\Pi_{\text{step}}(x_i)]\!]^{\mathfrak{A}}$.

$$\Pi(\mathfrak{A}) := \begin{cases} \bot & \text{if } [\![\Pi_{\text{halt}}(x_i)]\!]^{\mathfrak{A}} = \text{false, for all } i \\ [\![\Pi_{\text{out}}(x_t)]\!]^{\mathfrak{A}} & \text{for the minimal } t < \omega \text{ with } [\![\Pi_{\text{halt}}(x_t)]\!]^{\mathfrak{A}} = \text{true} \end{cases}$$

Definition of choiceless polynomial time

A BGS-programme is a triple $\Pi = (\Pi_{\text{step}}(x), \Pi_{\text{halt}}(x), \Pi_{\text{out}}(x))$ of a BGS-term and two BGS-formulae.

Its computation on a finite structure $\mathfrak{A}$ is a sequence $x_0, x_1, \ldots$ of sets in $\mathrm{HF}(\mathfrak{A})$ with $x_0 = \varnothing$ and $x_{i+1} := [\![\Pi_{\text{step}}(x_i)]\!]^{\mathfrak{A}}$.

$$\Pi(\mathfrak{A}) := \begin{cases} \bot & \text{if } [\![\Pi_{\text{halt}}(x_i)]\!]^{\mathfrak{A}} = \text{false, for all } i \\ [\![\Pi_{\text{out}}(x_t)]\!]^{\mathfrak{A}} & \text{for the minimal } t < \omega \text{ with } [\![\Pi_{\text{halt}}(x_t)]\!]^{\mathfrak{A}} = \text{true} \end{cases}$$

CPT: Choiceless polynomial time (with counting) is the set of properties computable by BGS-programmes that

- operate in polynomial time
- activate only a polynomial number of sets in their computation.

The power of choiceless polynomial time

CPT is a proper extension of (FP + C)

CPT can define any polynomial time property of small definable subsets 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 subsets 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

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

CPT can solve certain systems of linear equations (with a preorder on the variables) that cannot be solved in (FP+C)

The power of choiceless polynomial time

CPT is a proper extension of (FP + C)

CPT can define any polynomial time property of small definable subsets 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

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

CPT can solve certain systems of linear equations (with a preorder on the variables) that cannot be solved in (FP+C)

Open problem: Does CPT capture polynomial time ?

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 $\text{FO}[\tau, \sigma]$-interpretation defining from the input structure $\mathfrak{A}$ an initial state $\mathfrak{A}_0 := I_{\text{init}}(\mathfrak{A})$
- I_{step} is a $\text{FO}[\sigma, \sigma]$-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 $\text{FO}[\tau, \sigma]$-interpretation defining from the input structure $\mathfrak{A}$ an initial state $\mathfrak{A}_0 := I_{\text{init}}(\mathfrak{A})$
- I_{step} is a $\text{FO}[\sigma, \sigma]$-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

In this form PIL is equivalent to CPT without counting. To get the full power of choiceless polynomial time, interpretations have to be equipped with counting constructs, such as cardinality comparison.

Conclusion

The problem of whether there exists a logic for PTIME is still one of the most intriguing problems of logic in computer science.

Conclusion

The problem of whether there exists a logic for PTIME is still one of the most intriguing problems of logic in computer science.

Although fixed-point logic with counting fails to capture PTIME it is the logic of reference in this area.

Conclusion

The problem of whether there exists a logic for PTIME is still one of the most intriguing problems of logic in computer science.

Although fixed-point logic with counting fails to capture PTIME it is the logic of reference in this area.

Linear algebra is a source of problems to measure the power of extensions of (FP + C). Linear algebra may also provide new logical operators to define such extensions. The group membership problem seems to be promissing for such an approach.

Conclusion

The problem of whether there exists a logic for PTIME is still one of the most intriguing problems of logic in computer science.

Although fixed-point logic with counting fails to capture PTIME it is the logic of reference in this area.

Linear algebra is a source of problems to measure the power of extensions of (FP + C). Linear algebra may also provide new logical operators to define such extensions. The group membership problem seems to be promissing for such an approach.

Choiceless polynomial time is still an interesting candidate for a logic for PTIME. Its power is difficult to understand. Perhaps its characterization via intepretations will help to provide new methods for studying CPT.