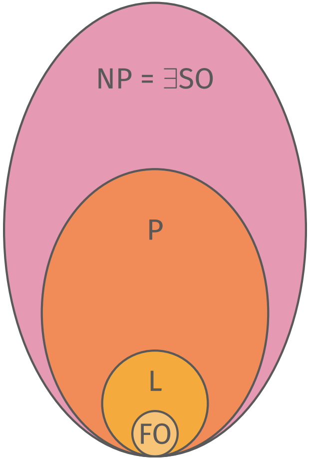
Fixed-point logics

Benedikt Pago [1]

ESSLLI 2025, Bochum

[1]University of Cambridge

UNIVERSITY OF CAMBRIDGE
Department of Computer Science and Technology


NP = ∃SO
P
L
FO

First-order logic is **local**:

Theorem

There is no FO-sentence that expresses whether a graph is connected.

Solution: Extend FO with an *iteration mechanism*.

Fixed-point logic (IFP) extends the syntax of FO with the following operator:

$$[\textbf{ifp}\ R\bar{x}.\ \varphi(\bar{x}; R)](\bar{y}).$$

It holds $\mathfrak{A} \models [\textbf{ifp}\ R\bar{x}.\ \varphi(\bar{x}; R)](\bar{y})$ iff $\bar{y}$ is in the *least-fixed point* defined by φ:

- $R_0 = \emptyset$.
- $R_1 = R_0 \cup \{\bar{a} \in A^{\mathrm{ar}(R)} \mid \mathfrak{A} \models \varphi(\bar{a}; \emptyset)\}$.
- $R_2 = R_1 \cup \{\bar{a} \in A^{\mathrm{ar}(R)} \mid \mathfrak{A} \models \varphi(\bar{a}; R_1)\}$.
- ...
- $R_{\mathrm{fix}} = R_{\mathrm{fix}+1}$.

Theorem

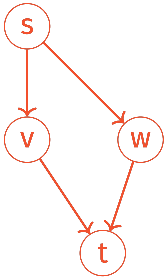
For every sentence $\psi \in \mathrm{IFP}$*, its model-checking problem* $\mathcal{MC}_\psi$ *is in* PTIME*.*

Proof.

The evaluation of $[\mathbf{ifp}\ Rx.\ \varphi(x; R)](y)$ in $\mathfrak{A}$ takes at most $|A|^r$ steps, where r is the arity of R.

Reachability:

Input: A directed graph $G = (V, E, s, t)$.
Question: Is there a path from s to t?



$$\varphi := [\textbf{ifp}\ Rx.\ \underbrace{(x = s \lor \exists y(Ry \land E\,yx))}_{\text{“Add to } R \text{ each vertex } x \text{ that is } s \text{ or has a predecessor in } R\text{”}}\ ](t).$$

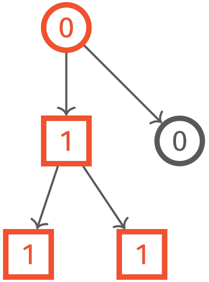
Fixed-point computation:

- $R_0 = \emptyset$.
- $R_1 = \{s\}$.
- $R_2 = \{s, v, w\}$.
- $R_3 = \{s, v, w, t\}$.

Examples of fixed-point computations

Game:

Input: A 2-player game graph $G = (V, V_0, V_1, E)$.
Question: Compute the set of winning positions for Player 0.



$$\varphi(x) := [\mathbf{ifp}\, Rx.\, \underbrace{(V_0\, x \wedge \exists y(Exy \wedge R\, y)) \vee (V_1\, x \wedge \forall y(Exy \rightarrow R\, y)).}_{}\,](x).$$

"A vertex in V_0 is winning if it has a winning successor.
A vertex in V_1 is winning if all its successors are winning."

Does IFP capture PTIME?

Theorem (Immerman-Vardi, 1986 & 1982)

IFP *captures* PTIME *on the class of all linearly ordered finite structures.*

Theorem

IFP *fails to capture* PTIME *on the class of all finite structures.*

Proof of the Immerman-Vardi Theorem

Theorem (Immerman-Vardi, 1986 & 1982)

IFP *captures* PTIME *on the class of all linearly ordered finite structures.*

Proof.

Translate any polynomial-time TM M into an IFP-sentence, similar to Fagin's theorem:

- Use the order to define a string encoding of the input structure.
- Since M uses only polynomial time and space, there is a $k \in \mathbb{N}$ such that a k-ary fixed-point relation can be used to simulate the run of M.

Proving inexpressibility results for IFP

Theorem

IFP *fails to capture* PTIME *on the class of all finite structures.*

Proof structure:

1. Embed IFP into **infinitary FO**.
2. Define a *pebble game* that characterises indistinguishability in infinitary FO.
3. Use this to show that IFP cannot define whether a finite structure has EVEN size (which is clearly in PTIME).

For $k \in \mathbb{N}$, denote by $\mathcal{L}^k_{\infty\omega}$ the ***k*-variable fragment of infinitary FO**.
It extends *k*-variable FO with the following formula formation rules:

- If Φ is an (infinite) set of $\mathcal{L}^k_{\infty\omega}$-formulas, then $\bigvee \Phi$ is an $\mathcal{L}^k_{\infty\omega}$-formula.
- If Φ is an (infinite) set of $\mathcal{L}^k_{\infty\omega}$-formulas, then $\bigwedge \Phi$ is an $\mathcal{L}^k_{\infty\omega}$-formula.

$$\mathcal{L}^\omega_{\infty\omega} = \bigcup_{k \in \mathbb{N}} \mathcal{L}^k_{\infty\omega}.$$

Embedding IFP into infinitary FO

Theorem

For every sentence $\psi \in \text{IFP}$ *there exists a* $k \in \mathbb{N}$ *and a* $\varphi \in \mathcal{L}^k_{\infty\omega}$ *such that* ψ *and* φ *are equivalent on all finite structures.*

Proof.

Let k be the number of variables in $\psi \in \text{IFP}$.

- For any finite structure, we have $\mathfrak{A} \models [\textbf{ifp}\ R\bar{x}.\ \varphi(\bar{x}; R)](\bar{a})$ iff there exists $n \in \mathbb{N}$ such that $\bar{a} \in R^n$, which is the n-th iteration stage.
- For each $n \in \mathbb{N}$, there is a formula $\varphi^n(\bar{x})$ that defines R^n in every finite structure.
- $[\textbf{ifp}\ R\bar{x}.\ \varphi(\bar{x}; R)](\bar{a}) \equiv \bigvee_{n \in \mathbb{N}} \varphi^n(\bar{a})$.

Definition

Let $\mathfrak{A}, \mathfrak{B}$ two structures, $k \in \mathbb{N}$ the number of pebbles.
The *position* after any round is $(\bar{a} \in A^\ell, \bar{b} \in B^\ell)$ with $\ell \leq k$. In each round,

- Spoiler either removes a pebble-pair (a_i, b_i) that is currently on the board, or places an unused pebble on A or B.
- Duplicator: If Spoiler has placed a pebble, then Duplicator places the corresponding pebble on the other structure.
- If $\bar{a} \to \bar{b}$ does *not* define a *local isomorphism* $\mathfrak{A}[\bar{a}] \to \mathfrak{B}[\bar{b}]$, then Spoiler wins.

Duplicator wins if the play continues forever without Spoiler winning.

Theorem

Duplicator has a winning strategy in the k-pebble game on $(\mathfrak{A}, \mathfrak{B})$ if and only if $\mathfrak{A}$ and $\mathfrak{B}$ agree on all sentences of $\mathcal{L}^k_{\infty\omega}$.

Theorem

IFP *cannot express the* EVEN*-query and hence does not capture* PTIME.

Proof.

- Suppose for a contradiction that there is a sentence $\psi \in$ IFP that expresses whether a finite structure has even cardinality.
- There is a $k \in \mathbb{N}$ and an equivalent sentence $\varphi \in \mathcal{L}^k_{\infty\omega}$.
- Duplicator wins the k-pebble game on the structures $(\{1, \dots, k\}, \{1, \dots, k+1\})$ with empty vocabulary.
- Thus, they are not distinguished by φ and hence not by ψ. But one of them is odd, the other even.

Summary

Via model-comparison games, we have shown:

$$\text{FO} \lneq \text{IFP} \lneq \textsc{Ptime}.$$

Alternative argument: 0-1 Laws.

Definition

A logic $\mathcal{L}$ is said to have a **0-1-law** if for every relational vocabulary τ, and every sentence $\psi \in \mathcal{L}[\tau]$,

$$\lim_{n\to\infty} P(\mathfrak{A}_n \models \psi) \in \{0,1\},$$

where $P(\mathfrak{A}_n \models \psi)$ denotes the probability that an n-element τ-structure whose *relations* are chosen uniformly at *random* satisfies ψ.

Theorem (Kolaitis, Vardi, 1992)

The logic $\mathcal{L}^{\omega}_{\infty\omega}$ has a 0-1 law.

$\Longrightarrow$ EVEN is not definable in $\mathcal{L}^{\omega}_{\infty\omega}$ and hence not in IFP.

Constraint Satisfaction Problems

Let $\mathfrak{B}$ be a finite relational τ-structure, called **template**. Then CSP($\mathfrak{B}$) is the following problem.

CSP($\mathfrak{B}$):

Input: A finite τ-structure $\mathfrak{A}$.

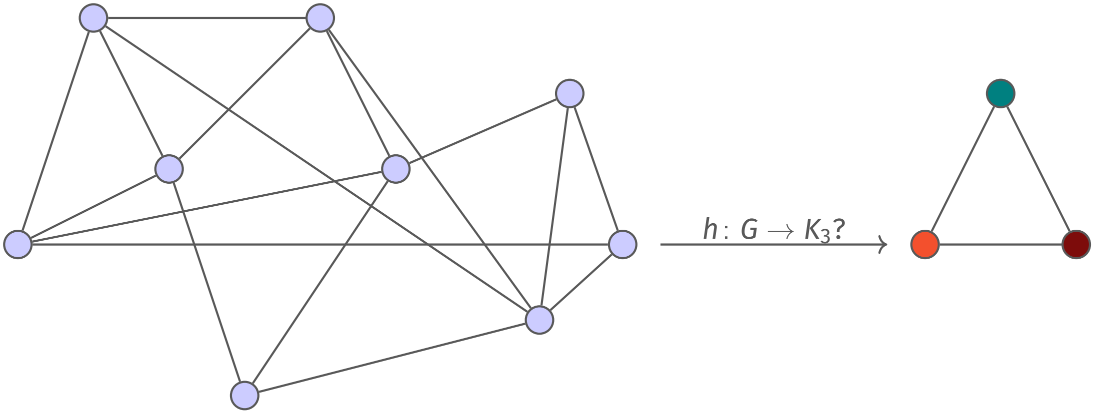
Question: Is there a homomorphism $h\colon \mathfrak{A} \to \mathfrak{B}$?

A *homomorphism* is a map h such that whenever $\bar{a} \in R^{\mathfrak{A}}$, then $h(\bar{a}) \in R^{\mathfrak{B}}$.

Examples:

- Systems of linear equations over finite fields
- Graph k-colourability
- Boolean satisfiability

Example: A graph G is 3-colourable if and only if it admits a homomorphism into K_3.



Theorem (Bulatov-Zhuk, 2017)

Let $\mathfrak{B}$ be a finite relational structure. Then $\mathrm{CSP}(\mathfrak{B})$ *is either in* PTIME *or* NP*-complete.*

Theorem (Barto-Kozik and Atserias-Bulatov-Dawar)

Let $\mathfrak{B}$ be a finite relational structure. Then $\mathrm{CSP}(\mathfrak{B})$ *is solvable in* IFP *if and only if it is solvable by the k-consistency algorithm, for a constant $k \in \mathbb{N}$.*

Note: There are CSPs in PTIME which are not in IFP.

The local consistency method

k**-consistency** for CSP($\mathfrak{B}$):

Input: An instance $\mathfrak{A}$.
Output: Is there a homomorphism $\mathfrak{A} \to \mathfrak{B}$? (answer can be wrong)
For every $X \subseteq A$ with $|A| \leq k$, initialise $\mathcal{H}(X) := \{h \colon \mathfrak{A}[X] \to \mathfrak{B} \mid h \text{ a homomorphism}\}$.
while $\mathcal{H}$ keeps changing **do**
    For every $X \subset Y$, if there is a $h \in \mathcal{H}(X)$ that does not extend to a $h' \in \mathcal{H}(Y)$, remove h from $\mathcal{H}(X)$.
    For every $X \subset Y$, if there is a $h \in \mathcal{H}(Y)$ such that $h|_X \notin \mathcal{H}(X)$, remove h from $\mathcal{H}(Y)$.
end while
If there is an X such that $\mathcal{H}(X) = \emptyset$, **return** UNSAT.
Else, **return** SAT.

Theorem

For every template $\mathfrak{B}$ and every $k \in \mathbb{N}$, there is a sentence $\psi_{k,\mathfrak{B}} \in \mathrm{IFP}$ such that for all instances $\mathfrak{A}$,

$$\mathfrak{A} \models \psi_{k,\mathfrak{B}} \iff k\text{-consistency accepts } \mathfrak{A}.$$

Remark: $\psi_{k,\mathfrak{B}}$ can be taken to be in the existential fragment of IFP, also called DATALOG.

Fixed-point logic with counting

Recall: IFP cannot define whether a structure has even cardinality.

Solution: Add a counting mechanism to the logic.

Fixed-point logic with counting (FPC) is the extension of IFP with *counting terms*.

For a finite structure $\mathfrak{A}$, let $\mathfrak{A}^*$ denote the **2-sorted structure**

$$\mathfrak{A}^* := \mathfrak{A} \uplus (\{0, \dots, |A|\}; <, 0, e),$$

where e is a constant with $e = |A|$. FPC[τ]-formulas use:

- a 2-sorted vocabulary $\tau \uplus \{<, 0, e\}$,
- 2-sorted variables $x, y, z \dots$, and $\lambda, \mu, \nu, \dots$.
- **counting terms:** If $\varphi(x)$ is a formula, then $\#_x[\varphi]$ is a term in the numerical sort.

Semantics: $[\![\#_x[\varphi]]\!]^{\mathfrak{A}} = t \in \{0, \dots, |A|\}$, where $t = |\{a \in A \mid \mathfrak{A} \models \varphi(a)\}|$.

Examples for expressivity of counting terms

Example

Regularity of undirected graphs can be expressed (i.e. every node has the same degree):

$$\forall x \forall y (\#_z[Exz] = \#_z[Eyz]).$$

Example

Isomorphism of equivalence relations E_1, E_2:

$$\forall \mu (\#_x[\#_y[E_1 xy] = \mu] = \#_x[\#_y[E_2 xy] = \mu]).$$

"For every equivalence-class-size μ, equally many elements are in a class of size μ in E_1 and in E_2."

Complexity hierarchy

$$\mathrm{FO} \lneq \mathrm{IFP} \lneq \mathrm{FPC} \lneq \textsc{Ptime}.$$

FPC can express EVEN

shown later

The power of FPC

- FPC can solve linear-algebraic problems over $\mathbb{Q}$ [Holm, 2010].
- FPC can solve the optimization problem for linear programs over $\mathbb{Q}$ [Anderson, Dawar, Holm, 2013].
- **Consequence:** FPC can define the size of a maximum matching in a graph.
- FPC captures PTIME on any proper minor-closed graph class [Grohe, 2014].

Definition

A **canonization** for a class $\mathcal{K}$ of structures is a function f that maps $\mathfrak{A} \in \mathcal{K}$ to an **ordered copy** $f(\mathfrak{A}) = (\mathfrak{A}, <)$ such that for all $\mathfrak{A}, \mathfrak{B} \in \mathcal{K}$,

$$f(\mathfrak{A}) = f(\mathfrak{B}) \iff \mathfrak{A} \cong \mathfrak{B}.$$

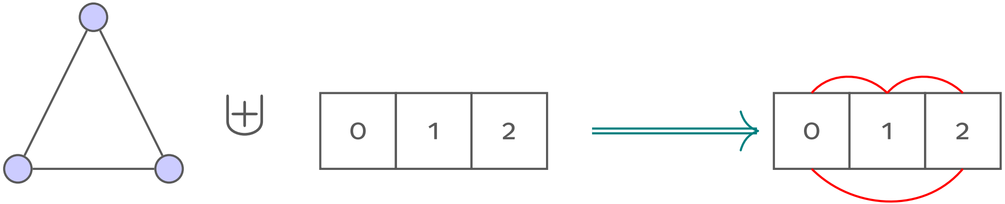
If f can be realized by an $\mathcal{L}$-interpretation, for a logic $\mathcal{L}$, then the canonization is $\mathcal{L}$-definable.

Theorem

If $\mathcal{L}$ is a logic at least as strong as IFP, *then if a class $\mathcal{K}$ of structures admits $\mathcal{L}$-definable canonization, $\mathcal{L}$ captures* PTIME *on $\mathcal{K}$.*

Canonization in FPC

In FPC, structures come with a linearly ordered number sort, in which we may define the canon.



Example: Canonizing directed trees in FPC.

- **Input:** A 2-sorted tree $\mathcal{T}^* = (V, E) \uplus (\{0, \dots, |V|\}, <, 0, e)$.
- Use the fixed-point operator to define a ternary relation $F \subseteq V \times \{1, \dots, |V|\}^2$ such that for every $v \in V$, $F_v := \{(i, j) \mid (v, i, j) \in F\}$ is the edge relation of an ordered copy of the subtree $\mathcal{T}_v$ rooted at v.
- **Inductive step:** Compute F_v assuming $F_{w_1}, \dots, F_{w_m}$ have been computed for the children of v.
- It suffices to *define an order* on $\{w_1, \dots, w_m\}$.
- F_{w_i} is an ordered copy of the subtree $\mathcal{T}_{w_i}$, so $\text{code}(\mathcal{T}_{w_i}, <) \in \{0, 1\}^*$ can be computed, and $\{w_1, \dots, w_m\}$ can be ordered according to $\text{code}(\mathcal{T}_{w_i}, <) \in \{0, 1\}^*$.

Just as IFP can be seen as a fragment of $\mathcal{L}^{\omega}_{\infty\omega}$, FPC is a fragment of $\mathcal{C}^{\omega}_{\infty\omega}$.

For every $k \in \mathbb{N}$, $\mathcal{C}^{k}_{\infty\omega}$ is the extension of $\mathcal{L}^{k}_{\infty\omega}$ with counting quantifiers $\exists^{\geq m}x$ for all $m \in \mathbb{N}$.

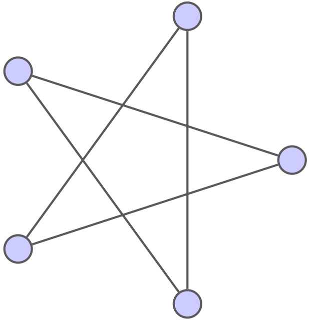
Theorem (Grädel and Otto, 1993)

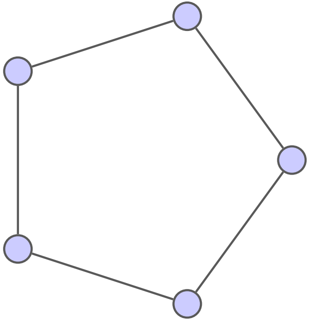
For every sentence $\psi \in \mathrm{FPC}$, there exists a $k \in \mathbb{N}$ and a $\varphi \in \mathcal{C}^{k}_{\infty\omega}$ such that ψ and φ are equivalent on all finite structures.

Graph Isomorphism

Graph Isomorphism:

Input: Two graphs G, H.
Question: Are G and H isomorphic?





Theorem

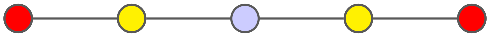
Let $\mathcal{K}$ be a class of graphs. Then the following are equivalent.

1. *The isomorphism problem for graphs in $\mathcal{K}$ can be solved in* FPC.
2. *There is a $k \in \mathbb{N}$ such that for all non-isomorphic $G, H \in \mathcal{K}$, $G \not\equiv_{\mathcal{C}^k} H$.*
3. *The $(k-1)$-dimensional Weisfeiler-Leman algorithm solves the isomorphism problem for graphs in $\mathcal{K}$.*

The Weisfeiler-Leman algorithm

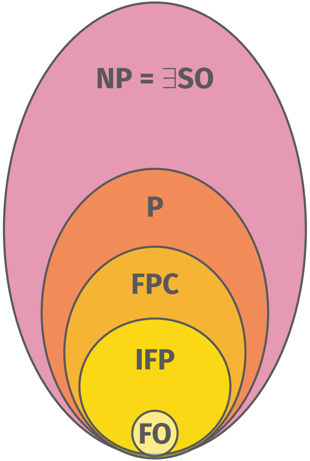
1-dimensional Weisfeiler-Leman:

Input: A graph G.
Output: A colouring of the vertices according to $\mathcal{C}^2$-types.
Initialise every vertex $v \in V(G)$ with the same colour $c(v)$.
while colouring keeps changing **do**
    For each $v \in V$, set $c(v) := \{\{c(w) \mid w \in E(v)\}\}$.
end while



- Generally, *k-dimensional Weisfeiler-Leman* computes a colouring of the k-tuples in a graph G according to their *$\mathcal{C}^{k+1}$-types* in G.
- We say that k-WL **distinguishes** G and H if the computed colourings are different.
- Intuitively, every FPC-sentence can at most distinguish all graphs that can also be distinguished by k-WL for some fixed $k \in \mathbb{N}$.

Summary



≈ Weisfeiler-Leman algorithm for Graph Isomorphism

≈ k-consistency algorithm for CSP

On ordered structures:

- IFP captures PTIME.
- Deterministic transitive closure logic captures LOGSPACE.
- Transitive closure logic captures NL.

[1] Matthew Anderson, Anuj Dawar, and Bjarki Holm. **Maximum Matching and Linear Programming in Fixed-Point Logic with Counting.** 2013. arXiv: 1304.6870 [cs.LO]. https://arxiv.org/abs/1304.6870.

[2] Albert Atserias, Andrei Bulatov, and Anuj Dawar. **"Affine systems of equations and counting infinitary logic".** In: Theoretical Computer Science 410.18 (2009), pp. 1666–1683.

[3] Albert Atserias, Andrei A. Bulatov, and Víctor Dalmau. **"On the Power of k-Consistency".** In: Automata, Languages and Programming, 34th International Colloquium, ICALP 2007, Wroclaw, Poland, July 9-13, 2007, Proceedings. Ed. by Lars Arge et al. 4596. Lecture Notes in Computer Science. Springer, 2007, pp. 279–290. 10.1007/978-3-540-73420-8_26.

[4] Albert Atserias and Joanna Ochremiak. **"Definable ellipsoid method, sums-of-squares proofs, and the isomorphism problem".** In: Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science. 2018, pp. 66–75.

[5] Libor Barto and Marcin Kozik. **"Constraint Satisfaction Problems Solvable by Local Consistency Methods".** In: J. ACM 61.1 (2014), 3:1–3:19. 10.1145/2556646. https://doi.org/10.1145/2556646.

[6] A. Bulatov. **"A Dichotomy Theorem for Nonuniform CSPs".** In: 58th IEEE Annual Symposium on Foundations of Computer Science, FOCS. 2017, pp. 319–330. 10.1109/FOCS.2017.37.

[7] Anuj Dawar. **"The nature and power of fixed-point logic with counting".** In: ACM SIGLOG News 2.1 (2015), pp. 8–21.

[8] Erich Grädel and Martin Otto. **"Inductive Definability with Counting on Finite Structures".** In: Computer Science Logic, 6th Workshop, CSL '92, San Miniato, Italy, September 28 - October 2, 1992, Selected Papers. Ed. by Egon Börger et al. 702. Lecture Notes in Computer Science. Springer, 1992, pp. 231–247. 10.1007/3-540-56992-8_15. https://doi.org/10.1007/3-540-56992-8%5C_15.

[9] Martin Grohe. **Descriptive Complexity, Canonisation, and Definable Graph Structure Theory.** 47. Lecture Notes in Logic. Cambridge University Press, 2017. ISBN: 9781139028868. 10.1017/9781139028868. https://doi.org/10.1017/9781139028868.

[10] Bjarki Holm. **"Descriptive complexity of linear algebra".** PhD thesis. University of Cambridge, UK, 2011. https://ethos.bl.uk/OrderDetails.do?uin=uk.bl.ethos.609435.

[11] Neil Immerman. **"Relational queries computable in polynomial time".** In: Proceedings of the fourteenth annual ACM symposium on Theory of computing. 1982, pp. 147–152.

[12] Sandra Kiefer. **"The Weisfeiler-Leman algorithm: an exploration of its power".** In: ACM SIGLOG News 7.3 (2020), pp. 5–27.

[13] Phokion G. Kolaitis and Moshe Y. Vardi. **"Infinitary Logics and 0-1 Laws".** In: Inf. Comput. 98.2 (1992), pp. 258–294. 10.1016/0890-5401(92)90021-7. https://doi.org/10.1016/0890-5401(92)90021-7.

[14] Martin Otto. **Bounded Variable Logics and Counting.** 9. Lecture Notes in Logic. Springer, 1997.

[15] Moshe Vardi. **“The complexity of relational query languages”.** In: Proceedings of the fourteenth annual ACM symposium on Theory of computing. 1982, pp. 137–146.

[16] D. Zhuk. **“A Proof of the CSP Dichotomy Conjecture”.** In: J. ACM 67 (2020), 30:1–30:78. 10.1145/3402029.