▸ AMIRHOSSEIN AKBAR TABATABAI, NEIL THAPEN, *Predicative Set Theory: A New Foundation for Feasible Mathematics.*
Bernoulli Institute, University of Groningen, The Netherlands.
E-mail: `amir.akbar@gmail.com`.
Institute of Mathematics, Czech Academy of Sciences, Czechia.
E-mail: `thapen@math.cas.cz`.

Absolute predicativism is a philosophical standpoint that rejects *all* self-referential definitions, including those of inductive structures like the natural numbers and binary strings. Various formalizations of absolutely predicative reasoning [9, 4] and computation [2, 3, 5, 8, 6, 7, 1] have been proposed. However, each theory or computational scheme tends to be tailored to a specific structure and often relies on some syntactic tricks. As a result, a unified, natural, and systematic foundation for such reasoning and computation is still absent.

In this talk, we introduce a unified foundational system called *Predicative Set Theory*, denoted by PST. This theory is a two-sorted weak set theory, with one sort for sets and another for Σ_1-definable classes. Our main goal is to argue that PST, as a predicative set theory, provides a *structural foundation* for *feasible mathematics*. It is worth noting that, since no explicit bounds appear anywhere in the theory, there is no obvious connection to computational complexity at first glance.

To support this main claim, we first show that, like any sufficiently strong set theory, PST can define natural numbers following the von Neumann style. As a consequence, it can also formalize a binary string of length n as a subset of the natural number $n = \{0, 1, \ldots, n-1\}$. Then, we show that the binary strings in PST form a model of S^1_2, the arithmetical theory formalizing feasible reasoning. This yields an interpretation of S^1_2 into PST by relativizing quantifiers to binary strings, demonstrating that PST is strong enough to fully formalize feasible arithmetic. In particular, every polynomial-time computable function is Σ_1-definable and provably total over binary strings in PST. Conversely, we interpret the Σ_1-fragment of PST on binary strings inside S^1_2, implying that the provably total Σ_1-definable functions over binary strings in PST are precisely the polynomial-time computable ones. Therefore, putting together, these results support the claim that PST provides a foundation for feasible computation on binary strings.

In the following, we present the details of the ideas outlined above. Let $\mathcal{L}$ be the first-order language with two sorts: one for sets and one for classes. The language includes the predicates $\{\in, =\}$ on the first sort, and a binary relation $X(x)$ between the sorts. For simplicity, we sometimes write $x \in X$ instead of $X(x)$. A first sort quantifier is called *bounded* if it is of the form $\forall y \in x\, \varphi(x, y)$ or $\exists y \in x\, \varphi(x, y)$. A formula is called *bounded* if it contains no second sort variables and all its first sort quantifiers are bounded. We denote the class of bounded formulas by Δ_0. We define the class Σ_1 as the smallest class of formulas that includes Δ_0 and atomic formulas of the form $X(x)$, and is closed under conjunction, disjunction, and existential quantification. By Σ_1^-, we mean the class of formulas in Σ_1 with no second sort variables.

Definition 1. The theory PST consists of standard first-order logic over $\mathcal{L}$, along with the usual equality axioms for sets and the following non-logical axioms:
- **Extensionality**: $\forall ab\, (\forall x\, (x \in a \leftrightarrow x \in b) \to a = b)$
- **Empty Set**: $\exists x\, \forall y\, \neg(y \in x)$
- **Pairing**: $\forall ab\, \exists y\, \forall u\, (u \in y \leftrightarrow u = a \vee u = b)$
- **Union**: $\forall a\, \exists y\, \forall x\, (x \in y \leftrightarrow \exists u \in a\, (x \in u))$
- Δ_0-**Separation**: $\forall a\, \exists y\, \forall x\, [x \in y \leftrightarrow x \in a \wedge \varphi(x)]$, for all formulas $\varphi(x) \in \Delta_0$, where y is not free in $\varphi(x)$.
- Σ_1-**Comprehension**: $\exists X\, \forall x\, [X(x) \leftrightarrow \varphi(x)]$, where $\varphi \in \Sigma_1$ and X is free in φ.

The theory PST has a canonical model in which the sets are interpreted as elements of V_ω, i.e., the hereditarily finite sets, and the classes are interpreted as Σ_1^--definable subsets of V_ω. We refer to this model as the *standard model* of PST, denoted by $\mathcal{V}_\omega$.

To represent natural numbers in PST, we take the empty set 0 as the number zero and define the successor function by $s(x) = x \cup \{x\}$. A natural candidate for the predicate "being a natural number" is $\mathbb{N}(x) = \forall X(\mathrm{Ind}(X) \to x \in X)$, where the formula $\mathrm{Ind}(X)$ is $0 \in X \wedge \forall z(z \in X \to s(z) \in X)$ which expresses that X is an *inductive class*. We sometimes write $\mathbb{N}(x)$ as $x \in \mathbb{N}$. This means that x is a natural number if it belongs to the intersection of all inductive classes. Note that the elements satisfying $\mathbb{N}(x)$ form neither a set nor a class, as $\mathbb{N}(x)$ is a Π_1^1 formula (i.e., essentially a formula beginning with some universal second sort quantifiers, followed only by first sort quantifiers). This captures the essence of predicativity: each definition of classes and then of $\mathbb{N}$ raises the level, avoiding vicious circles. However, with Π_1^1-comprehension, $\mathbb{N}$ satisfies the full induction axiom, essentially because it becomes an inductive class itself.

Having natural numbers, we can interpret "x is a binary string", denoted by $\mathbb{W}(x)$, as x being a pair consisting of a natural number n and a subset of n. The first component represents the length of the string, while the second encodes the bits of x. We sometimes write $\mathbb{W}(x)$ as $x \in \mathbb{W}$. Our first main theorem is:

THEOREM 2. *If $S_2^1 \vdash \varphi$ then* $\mathrm{PST} \vdash \varphi^{\mathbb{W}}$, *where $\varphi^{\mathbb{W}}$ is the suitable interpretation of φ relativizing all quantifiers to the elements of $\mathbb{W}$.*

As a corollary, Theorem 2 shows that all polynomial-time computable functions are Σ_1^--definable, provably total, and unique in PST over $\mathbb{W}$:

COROLLARY 3. *For any polynomial-time computable function f, there exists $\varphi(\bar{x}, y) \in \Sigma_1^-$ such that $\mathcal{V}_\omega \vDash \forall \bar{x} \in \mathbb{W}\, \varphi(\bar{x}, f(\bar{x}))$ and* $\mathrm{PST} \vdash \forall \bar{x} \in \mathbb{W}\, \exists! y \in \mathbb{W}\, \varphi(\bar{x}, y)$.

The second main contribution is to prove a witnessing theorem showing the converse of Theorem 2:

THEOREM 4. *Let $\varphi(\bar{x}) \in \Sigma_1^-$ such that* $\mathrm{PST} \vdash \forall \bar{x} \in \mathbb{W}\, \varphi(\bar{x})$. *Then, $S_2^1 \vdash \forall \bar{x}\, \varphi^D(\bar{x})$, where φ^D is an appropriate encoding of sets via binary strings in S_2^1.*

This implies that only Σ_1^--definable total functions over $\mathbb{W}$ in PST are polynomial time computable:

COROLLARY 5. *If* $\mathrm{PST} \vdash \forall \bar{x} \in \mathbb{W}\, \exists y\, \varphi(\bar{x}, y)$, *where $\varphi(\bar{x}, y) \in \Sigma_1^-$. Then, there is a polynomial time computable function f such that $\mathcal{V}_\omega \vDash \forall \bar{x} \in \mathbb{W}\, \varphi(\bar{x}, f(\bar{x}))$.*

Here are some remarks about PST and the above-mentioned results. First, PST employs an *expressive* set-theoretical language that enables the formalization of predicative reasoning and computation on various inductive objects *at once*—from numbers and binary strings to well-founded trees. Second, the *structural* approach of PST to feasible reasoning, in which no explicit bounds appear in the theory, offers an alternative synthetic and structural understanding of feasible computation and helps extend the notion of feasible computation from binary strings to *arbitrary sets*. Third, our foundational approach reveals the unexpected *foundational role* of feasible computation and suggests that computational explosions and vicious circles are two sides of the same coin. Finally, similar to [4], since PST defines numbers and binary strings in the usual set-theoretic way, our results show that what makes computation in ZFC infeasible is the *wild comprehension* it allows—such as powersets or classes defined by Π_1^1-formulas. Without such existential axioms, the base setting of computation remains feasible.

[1] Toshiyasu Arai. Predicatively computable functions on sets. Archive for Mathematical Logic, 54(3- 4):471–485, 2015.

[2] Stephen Bellantoni and Stephen Cook. A new recursion-theoretic characterization of the polytime functions. Computational Complexity, 2(2):97–110, Jun 1992.

[3] Stephen J. Bellantoni. Predicative recursion and computational complexity. PhD thesis, University of Toronto, Canada, 1992.

[4] Daniel Leivant. A foundational delineation of computational feasibility. In Proceedings 1991 Sixth Annual IEEE Symposium on Logic in Computer Science, pages 2–3. IEEE Computer Society, 1991.

[5] Daniel Leivant. Ramified recurrence and computational complexity. I. Word recurrence and poly- time. In Feasible mathematics, II (Ithaca, NY, 1992), volume 13 of Progress in Computer Science and Applied Logic, pages 320–343. Birkh¨auser Boston, Boston, MA, 1995.

[6] Daniel Leivant. Ramified recurrence and computational complexity. III. Higher type recurrence and elementary complexity. volume 96, pages 209–229. 1999. Festschrift on the occasion of Professor Rohit Parikh's 60th birthday.

[7] Daniel Leivant. Feasible functionals and intersection of ramified types. Electronic Notes in Theoretical Computer Science, 70(1):149–162, 2003.

[8] Daniel Leivant and Jean-Yves Marion. Ramified recurrence and computational complexity. II. Substitution and poly-space. In Computer science logic (Kazimierz, 1994), volume 933 of Lecture Notes in Computer Science, pages 486–500. Springer, Berlin, 1995.

[9] Edward Nelson. Predicative arithmetic, volume 32 of Mathematical Notes. Princeton University Press, Princeton, NJ, 1986.