Mathematical Logic in Computer Science
(eds. Dömölki et al.), 26.
Colloq. Math.

An Algorithmic Model of Strict Finitism[1]

Strict finitism does not yet exist as a clearly delineated mathematical theory, all we have to go on are various formulations of its basic tenets,[1] which are summarized in section 1 below. In a fashion similar to that of Finitism and Intuitionism it has its roots in an attitude towards the problem of securing the foundations of mathematics. One of the tasks of the logician is to capture such conceptually presented foundational approaches into a formal logico-mathematical system – as nearly true to the original conceptions as is possible.

The goal of the present paper is to present, motivate, and discuss a technical Ansatz for a system of strict finitism. We do this with the purpose of exhibiting some of the difficulties in sustaining the original basic attitudes against the criticism that arises once these attitudes are made precise.[2]

We do not wish to discuss here in any detail the recent papers,[3] which arose from a similar questioning of the role of the arbitrarily large finite in the foundations and in proof theory. As far as we are able to determine, the present approach is the most radically restrictive among these.

In any case, the outcome of our present work is that strict finitism can do no more than produce a mathematical system which may (or may not) be interesting in itself[4]. It is not, in our opinion, a reasonably tenable position in foundation. See the arguments in the last section.

1 The standpoint

In strict finitism we envision a radicalization of the constructionist viewpoint. This radicalization is motivated by the observation that many of the "constructions" allowed by the constructionists are constructions only in the sense of being potentially executable - indeed, executable only in an idealized world of infinitely patient and

[1]The manuscript of this paper, written in 1971, was lost until recently. We were encouraged to publish it now, because a number of papers with related positions and results have been put forward, and our paper may contribute to this discussion.

[1]For example in Bernays [1, pp. 280-281], Wang [9, pp. 473-476].

[2]The author is indebted to Bernays, Kreisel, G.H. Mueller and others for discussion of the basic issues involved.

[3]We mention in particular the work of Yessenin-Volpin [11], [12], Geiser[3], Goguen [4], Parikh [7], and Williamsen [10]; see also footnotes 7 and 8.

[4]For example, it may be useful for providing a foundation for a Strict Finitist theory of computation.

gigantic machines. How convincing is such a world for a firm foundation of mathematics? Would it not be more realistic to base this foundation on a realm of actually feasible processes? Strict finitism, then, is conceived as treating of the ideas that variously restricted finite beings develop about the concrete mathematical structure which they consider. It thus takes the form of a metatheory, in a classical framework, whose object-theory (or theories) are the sets of ideas of restricted beings referred to above.

Let us therefore consider beings, mathematicians as it were, who try to gain knowledge about a basic mathematical structure, say set theory. The way to gain such knowledge is to perform thought experiments. And, indispensable with scientific experiments, these should be reproducible, hence governed by fixed and communicable programs. Strictly finitist mathematicians operate under restrictions which we could formulate conceptually as follows.

(a) The sets that are considered, i.e. that enter the experiment, are in reality always finite and so are their elements and elements of elements, etc.

(b) Each individual mathematician thinks only during a restricted period of time, and has only restricted imagination.

Our goal now is to construct a series of increasingly intelligent and patient (models of) mathematicians, to investigate what each one's ideas would be about set theory, and on what properties of sets these mathematicians are able to come to a consensus. This consensus is what we call strict finitist truth. Our hope is that the model is realistic enough so that this consensus has a large overlap with classical set theory. For example, our model should explain why, and in what fashion, finite minds can perceive infinite totalities.

2 A technical realization of the standpoint

Let F be the set of *hereditarily finite sets*, i.e. let $F = \bigcup_{i<\omega} R(i)$ where $R(0) = \emptyset$, $R(i+1) = P(R(i)) =$ power set of $R(i)$.

We envision experiments within the relational structure $\mathcal{F} = \langle F, \in \rangle$, conducted according to *programs* that are written in a fixed programming language similar to ALGOL. The exact details of the structure of this language are unimportant here; we shall mostly present programs in the form of flow-diagrams (which are self-explanatory).

By a *complexity measure* on programs we understand a function $\mu : \Pi \to I\!N$ from the set Π of all programs to the set $I\!N$ of natural numbers with the property that each set $C_i = \{\pi \in \Pi : \mu(\pi) \leq i\}$ is finite.

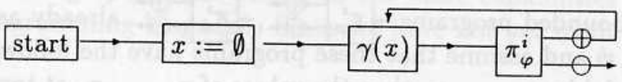
A program $\gamma(x)$, containing the variable x as indicated, is called a *generating program* (at x) if there exists a sequence $a_0, a_1, a_2, \ldots$ of sets such that $a_0 = \emptyset, F = \{a_0, a_1, a_2, \ldots\}$ and such that $x = a_{i+1}$ is the output of the program for the input $x = a_i$; and $x = a_0$ is the output for input $x = \emptyset$.

For each positive integer i and program π let π^i denote the modified program which arises when each loop in π is allowed to be run through at most i times. The program π^i has one additional exit which is taken if any one of the loops is about to be entered an $i+1$'st time.

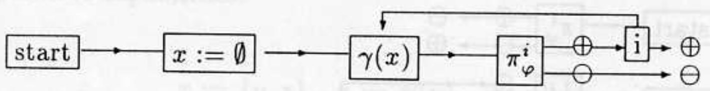
The bounding of loops is one of the two main devices by which we implement the inherent restriction of finite minds in our model. Let, for example, the formula $\varphi(x)$ of first-order predicate calculus express some properties of sets. The i-th mathematician will accept $\varphi(x)$ as true for an assignment $x := a$ if the thought-experiment which he associates with φ is successful for input $x := a$. When will he accept $(\forall x)\varphi(x)$?

Assume for the moment that we already know how to associate to the formula φ and any positive integer i such a thought experiment. That is, assume we are given a program $\to \boxed{\pi^i_\varphi}$ $\oplus$ $\ominus$ with two exits, and assume that this program takes exit $\oplus$ on input $x := a$ exactly if $\varphi[a]$ holds, (otherwise it terminates in exit $\ominus$).

Consider now the program

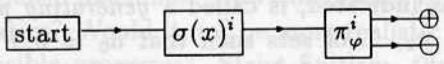


Since $\gamma(x)$ is a generator at x this program does *not* terminate exactly when all sets are such that the i-th mathematician would accept $\varphi[a]$ as true once he had tested a. However, the i-th mathematician does not have this patience and is willing to accept $(\forall x)\varphi(x)$ already after a limited amount of experimentation. We realize this restriction by letting him accept $(\forall x)\varphi(x)$ iff the i-bounded program



takes exit $\oplus$.

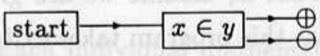
A second device is needed if we wish to implement the restrictions that adhere to strictly finitist statements of existence. Assume as above that we have associated π^i_φ to $\varphi(x)$. When will the i-th mathematician accept $(\exists x)\varphi(x)$? Obviously only if he can think of a program which constructs a set a satisfying φ. Now, the imagination of the i-th mathematician is bounded: he can think up only programs of complexity $\leq i$. For each such construction program $\sigma(x)$ consider the composite program



This program terminates in $\oplus$ exactly if the program $\sigma(x)$ indeed constructs an element such that the i-th mathematician accepts that it satisfies φ. There are only finitely many programs σ of complexity $\leq i$. By combining the above composite programs for each such σ, it is, therefore, easy to construct a single program $\pi^i_{\exists x\varphi(x)}$ such that the i-th mathematician accepts $(\exists x)\varphi(x)$ as true if this program terminates in $\oplus$.

We still owe the description of the passage between arbitrary first-order formulas $\varphi(x, y, \ldots)$ of set theory and programs π^i_φ. This procedure is defined recursively as follows:

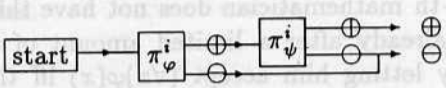
(1) To the formula $x \in y$ we associate the program

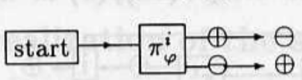


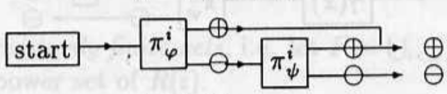
consisting of one conditional instruction

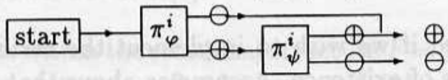
1 : *if* $x \in y$ *then go to 2 else go to 3.*

(2) Suppose that there are i-bounded programs π^i_φ, π^i_ψ already associated to the formulas φ, ψ and assume that these programs have the following property: If φ has free variables $x_1, \ldots, x_n$ then the values of $x_1, \ldots, x_n$ at termination of π^i_φ are the same as at the start. The programs for $\varphi \wedge x, \neg\varphi, \varphi \vee x, \varphi \rightarrow x$ are found as follows:

$\pi^i_{\varphi\wedge x}$ is 

$\pi^i_{\neg\varphi}$ is 

$\pi^i_{\varphi\vee\psi}$ is 

$\pi^i_{\varphi\rightarrow\psi}$ is 

(3) For the formulas $(\forall x)\varphi$ and $(\exists x)\varphi$ we have already described the passage to the corresponding programs.

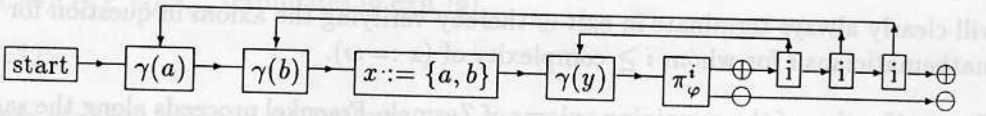
The set of *strict finitistically true sentences* can now be defined as the set of all sentences φ such that for all sufficiently large i the i-th mathematician accepts φ.

Clearly, this set does not depend on a particular complexity measure that we may have chosen.

Let us now investigate whether some of the familiar statements of set theory are strictly finitistically true. First we consider a typical existential axiom, the axiom of pairsets:

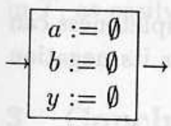
$$\forall a \forall b \exists x \forall y (y \in x \equiv (y = a \vee y = b))$$

Let φ be the quantifier-free part of the above axiom and let π^i_φ be the corresponding verification program of the i-th mathematician. Then



is the program with which the i-th mathematician verifies the pair-set axiom whenever the complexity measure of the program $x := \{a, b\}$ is $\leq i$.

In the case of the *pairset axiom*, and similarly for the powerset and union axioms, the strict finitistical-truth of the statement hinges essentially on the fact that corresponding operations belong to the basic capabilities (i.e. basic instructions) of the programming language. Since we have not yet made a list of these, it is time to do so:



Operative capabilities:

$$x := \{y, z\}, \quad x := P(y), \quad x := \bigcup y, \quad x := \emptyset, \quad x := y$$

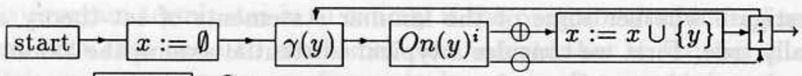
Conditional capabilities:

$$x \in y, \quad x = y$$

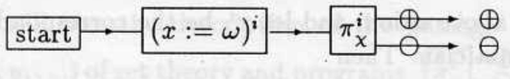
For the axiom of infinity:

$$\exists x (\emptyset \in x \wedge \forall y (y \in x \rightarrow y \cup \{y\} \in x)),$$

we need a program that would do something like $x := \omega$ (which is, of course, not included in the basic capabilities). Consider then the following program "$(x := \omega)^i$":



where $\rightarrow \boxed{On(y)^i} \rightrightarrows^{\oplus}_{\ominus}$ is the obvious program whereby the i-th mathematician checks whether y is an ordinal or not. Let us abbreviate the quantifier-free part of the axiom by $\chi(x)$ and let π^i_χ be the verifying program for $\chi(x)$ then



will clearly always terminate in exit $\oplus$ thereby verifying the axiom in question for all mathematicians i for whom $i \geq$ complexity of $(x := \omega)$.

The verification of the remaining axioms of Zermelo-Fraenkel proceeds along the same lines. But before we investigate what this result means, let us discuss quite generally the properties of the set T of sentences which are strictly finitistically true.

A first, obvious observation is that this set T does not contain φ and $\neg\varphi$ simultaneously for any sentence φ. But T is neither consistent nor complete *in the classical sense*. Namely, the formula

$$(\exists x)(\emptyset{\in}x \wedge (\forall y)(y{\in}x \rightarrow \{y\}{\in}x)) \wedge (\forall y)\neg(\emptyset{\in}x \wedge (\forall y) \wedge (y{\in}x{\rightarrow}\{y\}{\in}x))$$

is in T, but is - classically - of the form $\varphi \wedge \neg\varphi$, a contradiction. Incompleteness can be illustrated by the fact that neither the statement that ω is even nor its negation is in T.[5]

In order to facilitate the following discussion, let us assume that our programming language also admits variables that range over natural numbers, together with some simple capabilities with respect to computations on natural numbers (which we shall introduce as the need arises). Let there be given a Gödel-numbering of sentences, φ_p denoting the sentence with Gödel number p. We have not found a truth-definition of T in T in the sense of Tarski. This would be a formula $\sigma(x)$ such that $(\sigma(p) \rightarrow \varphi_p) \in T$ for all p. However, there is a quasi truth-definition, i.e. there exists a formula $\tau(x)$ such that $\tau(p) \in T$ iff $\varphi_p \in T$.

This can be seen as follows. For each natural number j let $U(j) = \{a : a = \overbrace{\gamma\gamma\ldots}^{i}(\emptyset)$ for some $i < j\}$, and assume that the operation $i_x := U(j)$ belongs to the basic capabilities of the programming language.

[5]This was pointed out to the author by the audience when the present paper was read before the colloquium at Heidelberg (1971). Other examples are easily found.

Furthermore, for each natural numbers p, k, m let $S(p,k,m) = \{\langle a_1, \ldots, a_m \rangle \in U(k)^m :$ $\rightarrow \boxed{\pi^k_{\varphi_p}} \begin{smallmatrix}\rightarrow \oplus \\ \rightarrow \ominus\end{smallmatrix}$ terminates in $\oplus$ on input $\langle a_1, \ldots, a_m \rangle\}$.

Assume again that the operation $y := S(p,k,m)$ is among the basic capabilities of the programming language. Now, let $\pi_\tau(z)^i$ be the program

$$\boxed{\text{start}} \rightarrow \boxed{(x := \omega)^i} \rightarrow \boxed{y := S(p,x,1)} \rightarrow \boxed{z := U(x)} \rightarrow \boxed{y = z}$$

Let p be the Gödel-number of a sentence φ_p. Then $\tau(p) \in T$

iff $(\exists i_o)(\forall i \geq i_o)(\pi_\tau(p)^i$ terminates in exit $\oplus)$,

iff $(\exists i_o)(\forall i \geq i_o)(\pi^i_{\varphi_p}$ terminates in exit $\oplus)$,

iff $\varphi^\in T$.

Hence $\tau(x)$ is a quasi truth-definition as stated.

If we try - in analogy to the proof of Tarski's theorem - to formulate the Liar's paradox, we again obtain an example of a sentence that is not in T and neither is its negation. Namely, let $d(n)$ be the Gödel-number of the result of substituting the constant symbol for n into the formula φ_n with Gödel-number n (assuming this formula has exactly the free variable x; for other formulas let $d(n) = 0$). Assume that $x := d(y)$ belongs to the basic operational capabilities of our programming language. Let $\varphi(x)$ be the formula $\neg\tau(d(x))$ whose Gödel-number is m, say. Then neither $\varphi(m)$ nor $\neg\varphi(m)$ are in T, as easily verified.

3 Conclusion

The fact that T contains both formulas $(\exists x)\varphi(x)$ and $\neg(\forall x)\neg\varphi(x)$ for some φ is surprising. A priori one would expect the axioms and rules of intuitionistic logic (at least) to hold for strict finitistical truth: in particular the intuitionistic theorem $(\exists x)\varphi(x) \rightarrow \neg(\forall x)\neg\varphi(x)$ should be acceptable. Thus, with modus ponens, T would be contradictory - which it is not. Since modus ponens can hardly be disputed by a strict finitist,[6] the question arises of how plausible the quoted intuitionistic theorem is for him. For the strict finitist it seems reasonable to interpret the above theorem thus: "If there is a construction of an element which all sufficiently large finite minds can perform and such that almost all finite minds are convinced that it has property φ, then it is absurd that almost all finite minds can at the same time be convinced that all elements do not have this property φ". There is no obvious conceptual reason for a strict finitist not to accept the above statement as true. Indeed, the above

[6] See, however, Goguen [4].

process of interpreting a formula could serve as a way to motivate a formal system of strict finitism (a process that is similar to that which is sometimes used to motivate formal systems of intuitionism). But the point is that the concepts that enter this "interpretation" are too vague. Our more precise interpretation, making sharper use of the restrictions of finite minds expressed in Section 1, does thus in fact diverge from it.

Perhaps it is good at this point to review some arguments *against* the plausibility of strict finitism. We have encountered two examples that illustrated the incompleteness of T. The fact that the statement of liar is neither (strictly finitistically) true nor false may please those who seek the way out of the liars paradox by relegating that statement to the realm of pronouncements which - while grammatical - are of indefinite truth value. More puzzling is the fact that there should be no agreement among finite minds on the parity of ω. Mathematicians who, after all, have finite minds, have no trouble in agreeing that ω is even. How can one explain this? It is obvious that mathematicians minds do not function as naively as our above model makes it out. For example, there seems to exist a sort of interplay between findings based on intuition (or "thought-experiments") and logical deductions from them. We might speculate about a process of educating the intuition which then becomes more acute on questions about actual infinite.[7] It is not surprising that our naive model of the mind gives only a poor approximation. It would be beautiful, of course, if an exact theory of mathematicians minds were available, and perhaps worthwhile to work towards one. But it is not realistic, in our opinion, to rely upon such an endeavor to "secure the foundations of mathematics". It would be more realistic to leave the theory - or a more appropriate variant of it - where it arose: in computer science, as a theory of feasible processes on a computer.[8]

[7](Added 1979) This idea was made precise by my former colleague Jeroslow [6] and by Hajek [5] as a so-called "experimental logic".

[8](Added 1979) For notions of "feasible numbers" and "feasible proofs" in computer science cf. the papers by Simon [8] and Cook [2].

References

[1] P. Bernays. On Platonism in Mathematics. In: *Philosophy of Mathematics* (eds Benacerraf and Putnam), Prentice-Hall (1964), 274–286.

[2] C. A. Cook. Feasibly constructive proofs and the propositional calculus. Preliminary version. *Proceedings of the 8th ACM Symposium on the Theory of Computing* (1976), 83–97.

[3] J. R. Geiser. A formalization of Essenin-Volpin's proof theory by means of nonstandards analysis (abstract). *IVth International Congress for Logic, Methodology, and Philosophy of Science*, Bucharest (1971), 72 pp.

[4] J. A. Goguen. The logic of inexact concepts, *Synthese* 19 (1968), 325–373.

[5] P. Hajek. Experimental logics and π_3^o theories. *Journal of Symbolic Logic* 42 (1977), 515–522.

[6] R. J. Jeroslow. Experimental logics and Δ_2^o theories. *Journal of Philosophical Logic* 4 (1975), 253–267.

[7] R. Parikh. Existence and Feasibility in Number Theory, *Journal of Symbolic Logic* 36 (1971), 494–508.

[8] J. Simon. On feasible numbers. (Preliminary version). *Proceedings of the 8th ACM Symposium on the Theory of Computing* (1976), 195–207.

[9] H. Wang. Eighty years of foundational studies, *Dialectica* 12 (1958), 466–497.

[10] J. S. Williamsen. *Induction and artificial intelligence*, Cambridge (1971), Thesis.

[11] A. S. Yessenin-Volpin. Le programme ultra-intuitioniste des fondements des mathematiques. In: *Infinitistic Methods*, Pergamon Press (1961), 201–223.

[12] A. S. Yessenin-Volpin. The ultraintuitionistic criticism and the antirational program for the foundations of mathematics. In: *Intuitionism and Proof Theory* (eds Myhill et al.), North-Holland (1970), 3–45.