On Procedural Postulation

Neil Barton[1], Ethan Russo[2], and Chris Scambler[3]

21. September 2022

[1]IFIKK, Universitetet i Oslo.

[2]New York University

[3]All Souls College, Oxford University

Standard Postulationism

How can one characterize a particular mathematical domain, like the natural numbers, points and lines of Euclidean geometry, or universe of sets?

Standard Postulationism: By laying down a set of *axioms* or *postulates* taken to be true of the domain.

For instance:

- Hilbert's axiomatization of geometry
- Peano Axioms

Standard Postulationism

What are these postulates?

Option 1: Postulates are implicit definitions of the key concepts or relations pertaining to the domain

- So: Hilbert's axioms define what counts as a system of points, lines, etc.
- Question: What guarantees the existence of instances of the notions thus defined?

Option 2: They are principles, which we intuit or otherwise come to know, true of some domain whose existence we intuit or otherwise come to know

- Question: how do we come to know the axioms?

The Idea of Procedural Postulation

How can one characterize a particular mathematical domain, like the natural numbers, points and lines of Euclidean geometry, or universe of sets?

Procedural Postulationism (Kit Fine): By laying down a set of rules or commands for the construction of the domain.

- Introduce a number which is not the successor of anything! Let there be a successor for every number with no successor! Make a set of all sets which do not contain themselves!
- Let i and $-i$ be the square roots of -1
- Let ∞ be a number greater than all reals

The Idea of Procedural Postulation

Procedural Postulationism (perhaps) can answer the worries raised for the two forms of Standard Postulationism.

- Existence of the domain is guaranteed by postulation (or made a non-problem by going modal)
- We come to know truths about the domain by deriving them from the rules of construction
  - Every natural number has a successor because of the way we have postulated that the natural numbers are constructed

The Language of Postulation: Imperatives

Basic imperatives: $!x.\phi$: Make an x that ϕ's! ("Make a set with no members!")

Complex Imperatives:

- $i; j$: Do i, and then do j!
- $\phi \rightarrow i$: If it's the case that ϕ, then do i! ("If it's raining, bring me an umbrella!")
- $\forall x i$: Do i to everything. ("Give every dog a treat!"; "Introduce a successor for every integer!")
  "Everything" is "everything 'currently' existing" (as opposed to existing *after* the command is executed)
- i^R (where R is an extensional well order): "iterate i once for every step in R!" (e.g.: i^4 is "do i, then i, then i, then i.")
- i^*: Do i forever!

The Language of Postulation: Postulational Modalities

How do we formalize the idea that something is so because of an imperative?

Postulational modalities $\langle i \rangle$ and $[i]$ for an imperative i:

- $\langle i \rangle \phi$ if, no matter how i is carried out, ϕ is true
- $[i]\phi$ iff, for some way of carrying out i, ϕ is true

An example: Let $i =$ Let there be a table!

Then: $[i]$ There is a table, $\langle i \rangle$ There is a dinner table, $\langle i \rangle$ There is a coffee table

Postulational Modalities (cont.)

What principles should govern these postulational modalities?

There are some principles we can give for imperatives of particular forms:

- **Make1** $[!x.\phi]\exists x\phi$
- **If1** $\phi \supset \forall p([\phi \to i]p \equiv [i]p)$
- **Then** $[i;j]\phi \equiv [i][j]\phi$

Are there more general principles that all postulational modalities ought to satisfy, like the K axiom or a rule of Necessitation?

Executability

We have to distinguish between imperatives that are *executable* and those that are not.

The intuitive distinction is between commands like $!x.x \neq x$, which cannot be carried out, and those that can.

Formally, we say: i is *executable* iff $\langle i \rangle \top$ (or: $\exists p \langle i \rangle p$), where $\top$ is some tautology like $\forall x(x = x)$

For non-executable commands, everything is necessary and nothing is possible.

We require that executable commands at least satisfy K and necessitation.

Executability Continued

We do not see any good way of telling on something like syntatic grounds whether an imperative will be executable or not.

We can nonetheless try to relate executability of complex imperatives to the executability of the simpler parts.

- $\langle i; j\rangle\top \equiv \langle i\rangle\langle j\rangle\top$
- $\langle p \to i\rangle\top \equiv (p \to \langle i\rangle\top)$
- (?) **Universal Executability.** $\langle\forall x i\rangle\top \equiv \forall x\langle i\rangle\top$

For i^*, the condition is more complicated. Roughly, i^* is executable if the process of iterating i indefinitely eventually comes to a *fixed point*.

Other Resources

We help ourselves to plural logic and propositional quantifiers

We also use another modal operator $\Box$, which we require to satisfy S4 and other principles like the Necessity of Distinctness, the converse Barcan formula:

$$\forall x \Box \exists y (y = x)$$

and the schema: for any i,

$$\Box \phi \supset [i]\phi$$

An intuitive interpretation of $\Box\phi$ is: "no matter how you execute any *possible* i, ϕ."

Suppose we enrich our language to contain a relation symbol S (for *successor*) and a predicate symbol N (for *number*).

We assume a bunch of background 'postulational constraints': things like if something is a number it necessarily is, necessarily no number is its own successor, and so on. These will be black-boxed henceforth.

- Let ζ be the command: $!x.Nx \wedge \forall y \neg S(y, x)$.
  - Intuitively: Make zero!
- Let σ be the command: $\forall x(Nx \rightarrow !y.S(x, y))$.
  - Intuitively: 'For each thing in the domain, check if it's a number, and if so, make a successor for it!'
- Let *Num* be the command: $\zeta; \sigma*$.
  - Intuitively: Make zero! Then, go on making successors forever!

Imperatival Arithmetic

1. We can prove $[Num]\exists x(Nx \wedge \forall y \neg S(y, x))$.
2. We can prove $[Num]\forall x Nx \supset \exists y S(x, y)$, using the fixed point condition.
3. To show induction, we must restrict to 'good' numbers, because we do not require executions to be *economical*.

Imperatival Arithmetic

	⋮
2	&
1	?
0	!
	@
	⋮

Nevertheless, we do have

Theorem

If *Num* is executable, then [*Num*] “The ‘good’ numbers satisfy second-order PA”.

Imperatival Set Theory

Suppose we enrich our language to contain a relation symbol S (for *set*) and a relation symbol $\in$ (for *membership*).

We assume a bunch of background 'postulational constraints': things like set membership is rigid, only sets have members, and so on. These will be black-boxed henceforth.

- Let *Pow* be the command: $\forall X!x.\forall y(y \in x \equiv Xx)$.
  - Intuitively: For any (plural) things, make a set with those things for its elements!

Note: it is crucial that $\forall X$ refers to pluralities existing *before* the acts of introduction take place; otherwise, we get Russell I.

Russell's Paradox II

Suppose it is necessary that *Pow* is executable. Even then, *Pow** is inexecutable.

Pow is a *necessary difference maker*: no matter what the circumstances, doing *Pow* changes something. (This definition uses propositional and imperatival quantification!)

If i is NDM, i^* is not executable.

A command of the form $(p \to i)^*$ is a *hedged iteration* (while p, do i.) This may be executable even if i is NDM.

While-do Lemma

If i is a NDM, $[(p \to i)^*]\neg p$.

Hedged iterations can therefore be used to build models for set theory up to some large cardinal.

By the while do lemma, if H is a 'smallness' property of the universe, then executability of $(H \rightarrow Pow)^*$ entails the possibility of a set-theoretic domain with the 'largeness' property $\neg H$.

Hedged iteration of pow: $(H \rightarrow Pow)^*$ for suitable H.

First pass: $H :=$ the von Neumann ordinals are not strongly inaccessible.

Second pass: $H :=$ the von Neumann ordinals are not strongly inaccessible, or some plurality of cardinality less than the von Neumann ordinals does not form a set.

Let $Set_H = (H \rightarrow Pow)^*$.

If second order ZFC is consistent, then so is the executability of Set_H.

Again, one might hope that the executability of Set_H entailed the sets were a model for ZFC2; again, these hopes are in vain.

Nevertheless, we do have

Theorem

If Set_H is executable, then $[Set_H]$ “the ‘good’ sets satisfy second order ZFC.”

Significance I: Consistency

Fine claims to provide:

> *an inductive proof that [weak executability] will be preserved under the standard (finitary) operations for forming complex [imperatives]. In such a proof we may assume, for example, that the consecutive procedure $\alpha;\beta$, will be [executable] whenever α and β are [executable].*

Let's assess this claim.

Significance I: Consistency

Regarding an inductive proof:

- We know of no such obvious proof.
- However our system does give us the means for formalising such proofs.
- Indeed as we've set things up, executability does 'flow upwards', if α and β are executable, given our definitions we can prove that $\alpha; \beta$ is executable (in particular, it falls right out of **Then** $[i; j]\phi \equiv [i][j]\phi$).
- But this isn't so clear when moving from from $i(x)$ being executable for every x to $\forall x i(x)$ being executable (we just wrote it in).

Significance I: Consistency

- Moreover, the **Universal Executability** assumption $\langle \forall x i(x) \rangle \top \equiv \forall x \langle i(x) \rangle \top$ seems like a substantive assumption. In particular:
  1. It guarantees long iterations.
  2. It seems independently problematic for postulation in general, suppose you have a single hammerhead h and two shafts s_1 and s_2, and consider the imperative "Attach h to x!"
- Perhaps there are stronger assumptions that get us **Universal Executability**, but we don't see any obvious picks.
- Another option is just to view ourselves as restricting the imperatives we allow (e.g. those for which doing i to any x doesn't affect whether you can do i to some other y).

Significance I: Consistency

Fine also claims:

> *...that the rules for the construction of a mathematical domain may be taken to represent our intuitive grasp of that domain and that a demonstration of the above sort may then be seen to represent the role that intuition can play in vindicating the consistency of the axioms for that domain. (p. 99, Fine OKOMO)*

The claim of formalising an intuition of consistency through postulation we find more plausible though.

Significance I: Consistency

- Given that you accept these principles (e.g. that certain commands are iterable into the transfinite) you can show consistency.
  - (**Note:** There's an interesting similarity with proof theory here (e.g. Gentzen): Consistency of theories is linked to possibilities for iteration.)
- Contrast the "one-shot construction"—"Make a model of ZFC_2 / PA_2!".
- Commands like *Num* and Set_H show how we can get a model of the axioms via intuitively executable steps, and the assumption that the iteration as whole is executable.
- There's a sense in which you might think one is getting at the intuition of how to construct a model.

Significance I: Consistency

- Again, though, this might have to be tempered.
- . There's a question, even if we formalise intuition of a structure, as to whether there's a further epistemological gain (e.g. credence increase).
- For example, in the case of PA_2 we can show that *Num* is executable iff there could be a well-order of length ω.
- But this isn't so surprising, it's well known that PA_2 is consistent iff there's an ω-sequence (modulo second-order logic).
- (The situation of set theory is more subtle.)

Significance II: Paradoxes and potentialism

Fine also claims that procedural postulation leads to a distinctive solution of the paradoxes via a set-theoretic potentialism. What to make of this claim?

- As we've given it, procedural postulationism demands that mathematical domains are outputs of terminating construction procedures.
- But because *Pow* is a necessary difference maker, *Pow** is not executable.
- If we assume that *Pow* is executable along any possible well-order, we get something like a Linnebo-style potentialism with a **Collapse** of pluralities to sets.
- However unlike Linnebo, for the postulationist there's a natural interpretation of set theory on which it's about domains obtained by explicitly hedged commands (e.g. $(H \to Pow)^*$), rather than the whole construction process (via the potentialist translation).

Significance II: Paradoxes and potentialism

- But how framework dependent is this solution to the paradoxes? (i.e. is the postulationism doing the work, or the assumptions we made?)
- We just assumed weak executability of Pow^*.
- What if we didn't accept the weak executability of Pow^*?
- One can then have commands like "Make absolutely all the sets!" (i.e. $!X.\Box\forall y Set(y) \supset Xy)$). Call this u.
- This could be executable, if we throw out the weak executability of Pow^*.
- You have to choose between the weak executability of Pow^* and the executability of u.
- Is this any different from the non-postulational picture?

Significance III: Arithmetic vs. set-theoretic potentialism

We now move on a little from Fine's points and discuss some other interesting features of the view.

- Note a difference between arithmetic and set theory: For arithmetic you can just state the 'obvious' principles to make numbers and have it executable (i.e. *Num*), but *Pow** cannot be executable.
- For the Finean postulationist arithmetical potentialism is a choice but set-theoretic potentialism is a mathematical fact of life (cf. Cantor on completable infinity).
- There is (again) the question of how different the situation is from the standard framework.
- Perhaps (?) this (partly) accounts for the difference in uptake between the two views?

Significance IV: Large cardinals and potentialism

Slogan

(In case time is tight.) Because of the hedges involved, the believer in the weak executability of *Pow** has reason to think that there's a link between the never ending sequence of bigger and bigger large cardinals, and potentialism.

Significance IV: Large cardinals and potentialism

- There is a thought in the literature that the never ending definitions of bigger and bigger large cardinals motivates potentialism.
- But this idea is at least somewhat unclear.
- Why not think of this never ending hierarchy as giving us successively more partial information about a single domain, rather than somehow indicating the presence of multiple domains and indefinite extensibility?

Significance IV: Large cardinals and potentialism

- For the Finean: A hedge H is needed to allow the operation Set_H to terminate, and a natural choice for such H is (the negation of) a large cardinal axiom.
- The H-small sets (i.e. those you're guaranteed to have after executing Set_H) can then be isolated using a second-order axiomatisation.
- But there is no terminating command that will guarantee the existence of sets that can be generated by any hedged command Set_H.
- A link between the domains we can talk about and large cardinal axioms is thus expected on this picture.

Summing up

- We think we've:
  (1.) Provided a language and framework in which postulationist principles can be formulated and discussed.
  (2.) Examined some interesting possible upshots of the view.
- We want to close with the following question (similar to ones that repeatedly recur in much of the philosophy of set theory literature):

Question.

How different is this view from the standard declarative picture really? Is it just a different language and underlying idea for formulating the same facts? (Contrast the Button-Linnebo discussion around height potentialism.)

Thanks!

Thanks for listening!