

# SEMANTICAL MUTATION, ALGORITHMS AND PROGRAMS

André Porto

Universidade Federal de Goiás

**RESUMO:** O presente artigo oferece uma explicação para aquela que talvez seja a tese mais estranha e menos intuitiva de Wittgenstein – a tese da mutação semântica – segundo a qual nunca se consegue responder a uma conjectura matemática porque a nova prova altera os próprios significados dos termos envolvidos na pergunta original. Ao invés de basearmos nossa justificação na distinção entre mero cálculo e demonstrações de proposições isoladas, característica do período intermediário, nós a generalizamos de forma a incluir procedimentos efetivos de cálculo também.

**ABSTRACT:** The present article offers an explanation of perhaps Wittgenstein's strangest and least intuitive thesis – the semantical mutation thesis – according to which one can never answer a mathematical conjecture because the new proof alters the very meanings of the terms involved in the original question. Instead of basing our justification on the distinction between mere calculation and proofs of isolated propositions, characteristic of Wittgenstein's intermediary period, we generalize it to include conjectures involving effective procedures as well.

Some of Wittgenstein's remarks on the philosophy of mathematics are notoriously extravagant, even hostile. Consider, say, his famous comment on “*the disastrous invasion' of mathematics by logic*” (Wittgenstein, 1983, p. 281).<sup>1</sup> None perhaps seems so unintuitive, downright outrageous as the Semantical mutation Thesis, though. By Semantical Mutation Thesis we mean Wittgenstein's strange idea that no mathematician could ever settle any mathematical conjectures because, according to him, all novel proofs strictly speaking alter the very concepts in terms of which the original conjectures were framed. In other words, mathematical advances are always answers to different

---

<sup>1</sup> This paper is part of the second chapter of a book I'm writing on Wittgenstein's Philosophy of Mathematics.

questions than their original motivating conjectures. Again according to Wittgenstein there will always be a “semantical gap” between conjectures and the rule they engender. One can never “level up” the two groups; a new mathematical advance always represents a *conceptual modification*. Thus the name we’ve proposed to this idea.

There is no doubt that Wittgenstein did maintain these eccentric ideas in his so-called middle period:

... once a proof has been supplied, it in no way proves what had been conjectured, ... you can’t conjecture the proof until you’ve got it, and not then, either. (Wittgenstein, 2005, pp. 418-9)

Why do I say that we don’t “discover” a proposition like the fundamental theorem of algebra, but instead “construct” it? - Because in proving it we give it a new sense that it didn’t have before. (Wittgenstein, 2005, p. 428)

The “medical proof ” didn’t incorporate the hypothesis it proved into a new calculus, and so didn’t give it a new sense; a mathematical proof incorporates the mathematical proposition into a new calculus, and alters its position in mathematics. (Wittgenstein, 2005, p. 426)

There is less agreement whether Wittgenstein did continue to support these ideas in his latter, mature years. Apparently, the majority of the authors (Wright, 1980, p. Chp 3; Shanker, 1987, p. Chp 3; Diamond, 1991, p. Chp 10; Glock, 1996) (Marion & Okada, 2012; Schroeder, 2012; Säätelä, 2011) are “continuists”. For them Wittgenstein never changed his mind about these odd opinions. But there is a smaller group which suggests quite the opposite:

The calculus conception [from the middle period] was unable to account for the change and growth of mathematics, while the language-game conception [from the latter period] emphasizes this. The bizarre views [on semantical mutation] that have earned many commentators’ derision drop out. (Gerrard, 1991, p. 132)<sup>2</sup>

---

<sup>2</sup> See also (Frascolla, 2004, pp. 180-1; Panjvani, 2006, pp. 420-1). I thank Severin Schroeder for indicating me some of these earlier sources.

For an exegete trying to make sense of Wittgenstein's ideas, it might look as a gain to classify the semantical mutation thesis as a "temporary speculation" within "transitory period" of the philosopher's development. One would still have to explain the source of these strange ideas - perhaps referring to this period's stringent verificationism - but one would be relieved of any further need of justifying this seemingly outrageous proposal. Unfortunately the latter textual evidence does not seem to bear this out. In Wittgenstein's *Remarks on the Foundations of Mathematics* one reads:

Now how about this - ought I to say that the same sense can only have *one* proof? Or that when a proof is found the sense alters?

Of course some people would oppose this and say: "Then the proof of a proposition cannot ever be found, for, if it has been found, it is no longer the proof of *this* proposition". (Wittgenstein, 1983, p. 366)

When I said that a proof introduces a new concept, I meant something like: the proof puts a new paradigm among the paradigms of the language ...One would like to say: the proof changes the grammar of our language, changes our concepts. (Wittgenstein, 1983, p. 166)

I should like to say: the proof shows me a new connection, and hence it also gives me a new concept. (Wittgenstein, 1983, p. 297)

The early dismissal of the semantical mutation thesis as a "temporary fling" doesn't appear to be really available. Thus, the only option left open for the exegete would be to squarely face the challenge of *making sense* of the horrible idea. The task appears daunting, though. Consider the recent discussion on the foundations of mathematics. One finds various proposals for alternative foundations of mathematical knowledge, besides the classical set theoretical one. There are the new categorical foundations, various type-theoretic versions, the intuitionistic reconstructions, even the ultra-finitists. Despite such diversity, there is one point about which all of them seem to agree: the precept that a mathematician is basically a *proof-producer*. It is

precisely this last common ground, perhaps the only common ground shared by all participants of the debate that Wittgenstein's proposal appears to threaten.

Of course, we could insist that Wittgenstein is not discarding the notion of *proof*. He is just altering the way we construe the connection between a *conjecture* and its *settling proof*. But Wittgenstein's proposal seems to undermine the idea that no proof can "come out of the blue". Any new proof was already sort of "potentially there", "prefigured in the very meanings of the mathematical concepts involved". A conjecture-settler (say, Andrew Wiles on Fermat's last theorem) doesn't downright *invent* his proof. The elements he needed for his achievement were "already there", somehow dormant in some theory (apparently called "the Theory of Elliptic Curves"). No one realized just how that could be done. He was the first to discover a way to do it, thus his claim to glory. All of this is constitutive of the very self-image of the mathematician, his sense of worth, his pride.

If the proof of a conjecture is never an answer to the question which preceded it, if each new proof is merely a semantical mutation which does not come out of any preexisting system which potentially prompts it, then new mathematical advances would appear *arbitrary*. The whole of mathematics would lose its necessary character. Is this what Wittgenstein is proposing? Of course once again our entire problem here is just how to characterize the notion of a system *determining* its inferential consequences. Just as in the example of the uncalculated decimal place of Pi (Porto, 2013, p. 83)<sup>3</sup>, Wile's settling of Fermat's conjecture seemed "*already somehow determined by a preceding system*". Paraphrasing Wittgenstein's famous remark, it's as if the system "ran ahead of us and fixed all its future *logical consequences* much before any actual mathematician ever realized that opportunity". We are clearly back to where we've started, the crucial question within the rule-following considerations: exactly *how* does a *function* determine its *extension*, how does a *program* prefigure its *execution* and, a *system* predetermine its "*logical consequences*"? What should we

---

<sup>3</sup> Any one beyond the 5,000,000,000,001<sup>st</sup>, according to 2010's attempt.

say about this “forward abstract inferential potency”, this anticipated fixation of all possible mathematical advances?

### **Wittgenstein’s pet example of a “Conjecture Settling”**

*In 1: 7 gibt es ein endliches Problem und ein unendliches.*  
(Wittgenstein, 1979, p. 183)

Let us return to Wittgenstein’s texts. To be sure, even the hardheaded philosopher has to accept, “at least a sociological fact”, the “phenomenon of conjecture”. Mathematicians do seem to answer some old mathematical challenges, even centenary ones, like in Wile’s case. Even stubborn Wittgenstein cannot but accept that:

Sometimes it seems as though mathematical discoveries are made by performing what one might call a mathematical experiment. For example, the mathematician first notices a certain regularity and then proves that it had to be so.  
(Wittgenstein, 1976, p. 92)

Yes: Wittgenstein himself concedes that “sometimes it seems as though” mathematical conjectures do occur. How could he deny that? One notices an apparent regularity and then demonstrates that that regularity was not accidental at all: it had to happen. The philosopher even has a sort of “pet example” of such phenomenon, the “discovery of the period” in a division such as  $1 \div 3$  or  $1 \div 7$ :

Suppose a person divides 1 by 3 to see whether 4 turns up in the development. I tell him "You will never get 4; it is hopeless", and draw his attention to the fact that the dividend and remainder are the same. This may never have struck him. Here it looks as though by drawing attention to this fact we did not perform an operation but showed what was already there. It also looks as though this is a quick way of showing what could be shown by carrying on the division to an enormous number of places and concluding that it is hopeless to look for a 4. The use of the rule to show by a short cut that a 4 cannot be found looks very much like giving up looking for a pine tree on being told that pines never grow in the soil of that neighborhood. (WITTGENSTEIN, 1979, p. 182)

To begin our discussion, there is an important point which deserves to be stressed here: his choice of example. In contemporary terms, it involves regularities of an *effectively computable operation*, a *recursive function* (the division  $1 \div 3$ ). There is surprisingly little discussion in the secondary literature on Wittgenstein's rather extensive discussions of these recursive processes, the periodic divisions.<sup>4</sup> Perhaps the usual accusation of a lack of mathematical sophistication by the philosopher is at work here.<sup>5</sup>

Be that as it may, the first thing we would like to stress about Wittgenstein's example is the fact that his rather humble question (whether the number 4 will ever turn up in the development of the quotient  $1 \div 3$ ) is carefully chosen to concern mathematical object about which all the participants of the foundational debate on mathematics, be they *classicists*, *intuitionists*, *finitists* or even *ultra-finitists*, recognize as a *genuine mathematical object*. Wittgenstein's choice of "conjecture" is anything but unsophisticated. Intuitionists sometimes discard the very meaningfulness of some classical mathematical statements and construals. In consonance with that, some classical mathematical conjectures (say, about inaccessible cardinals) are rejected as pseudo-questions by the *intuitionists* and even more so by the *finitists* and *ultra-finitists*. But the legitimacy of the object in question – a non-exponential recursive process – cannot be questioned, even by an ultra-finitist. The acceptability of Wittgenstein petty conjecture is beyond dispute, even from the point of view of these more "wild" alternatives. His example lays within the (rather narrow) area about which all parties in the foundational debate are in *agreement*. It's about the very hard core of mathematics.

There is also another important aspect of Wittgenstein's somewhat unassuming pet example which should be pointed out here. His "conjecture" – whether the reappearance of 3's was accidental, or whether one could ever obtain a different number, say, a 4 in the quotient's expansion – is strikingly reminiscent of a most illustrious

---

<sup>4</sup> (Wittgenstein, 1975, pp. 223-34), (Wittgenstein, 2005, pp. 466-7), (Wittgenstein, 1979, pp. 183-6) and (Wittgenstein, 1976, pp. 122-30)

<sup>5</sup> Cf (Bernays, 1986, p. 173; Mac Lane, 1986, p. 444; Putnam, 2007, p. 246)

problem in Classical Computability Theory: the famous Halting Question. All we would have to do to transmute his conjecture into the halting question would be to shift one's attention from the *quotient expansion* to that of the *remainder's expansion* and ask: does a number 0 ever turns up in that other expansion? Once again one can gage the centrality of Wittgenstein's investigation: his choice aims at the very heart of the foundational debate: the notions of *effective procedure*, of *infinity* and that of *computability*.

### **What are those conjectures about?**

Let us commence our effort to live up to our discouraging task: making sense of Wittgenstein's preposterous-looking thesis of semantical mutation within the very constrictive arena of effective computable functions. We have an operation, the unpretentious division  $1 \div 3$  and we have a "conjecture": whether there will ever be a 4 in the quotient's expansion 0,333 .... We notice that this is true for the initial decimals, and we make an hypothesis: this will always be like that, only 3's will ever turn up in that expansion, it is hopeless to look for a 4. Later, so the tale goes, we "settle our conjecture": we discover that that division is periodic and the period - a single digit "3" - does not include 4 (such as it happens, say, in the cycle of  $1 \div 7$ ).

Our hopeless-looking task, the task confronting *any Wittgenstein's wholehearted devotee*, would (unfortunately) be: to explain just how "the master" intends to argue in favor of a "semantical mutation" in such direct and transparent cases. What semantical mutation could Wittgenstein be talking about there? What possible conceptual change could have ever taken place in the diminutive transition from the "conjecture" ("*Will there ever be a 4?*") to its "settling" ("*It's periodic, There won't be any 4's!*"). Just what concept (if any) suffered that so-called "semantical mutation" and what exactly was that mutation?

As we pointed out before, our conjecture has almost the same logical structure as a most distinguished question, the *Halting question*. If Wittgenstein's example would have been just slightly different, we

would have in front of us precisely a very clear cut instantiation of the famous question about the *infinity* or *finiteness* of that very process. In other words, Wittgenstein's (and the devotee's) challenge will be to somehow show us that even the "halting settling" fails to answer the "halting conjecture": not a very promising task indeed.

We will have to move on by steps, one at a time. We are still not ready to meet our "main challenge" yet. As our first move toward that goal though, we have to begin by pointing out a difference in the way a *classicist* and an *intuitionist* construe those questions (the occurrence of 4's and the halting question). The difference concerns the *nominal part* of that "conjecture": what are we *conjecturing about*? The classical and intuitionistic construals diverge. For the classicist we are talking, of course, about an *extension*, a *set* of order pairs, along the lines of, say:

$$\{(n, q_n) : (1 \div 3)_{(n)} = q_n\}^6$$

or expressed in characteristically extensional notation:

$$\{(0, 0), (1, 3), (2, 3), (3, 3), \dots\}$$

All reference to the original *generating process*, the *operation* involved, is dropped in the classical proposal. In its place we are offered a strange postulation: the idea of a "final complete list of all partial answers" thoroughly independent of any epistemological act, a pure "record of a complete final execution of that operation" (an operation not operated by any one at no particular time). It is about that "pure list" that those conjectures are construed on, according to the classicist. In our example, the question regarding the occurrence of a 4 in the quotient's expansion would be construed as simply the appearance of that digit as the second component of some ordered pair within that list. No reference to a process, just pure "platonian actuality".

Intuitionists notoriously rejects such complete discarding of the operational component. For them, the list of partial answers is secondary, the operation, primary. In fact, as we all know, that "list"

---

<sup>6</sup> " $q_n$ " being of course the  $n^{\text{th}}$  decimal place of the quotient's development.

will end up being “infinite” (the “list” of partial results of the operation  $1 \div 3$ ). What would then be the point of talking about a “list” involving infinite entries, complains intuitionists? How *could we ever list* all partial answers in such extreme cases? In a typical intuitionist’s reaction to those classical proposals, Dummett writes:

... to grasp an infinite structure is to grasp the process which generates it, to refer to such a structure is to refer to that process, and to recognize the structure as being infinite is to recognize that the process will not terminate. (Dummett, 1977, p. 56)<sup>7</sup>

For an intuitionist, in all such cases (as in our “conjectures”) we are not talking, or conjecturing, about *extensional objects*, these strange “lists of correct partial answers”. We are referring to “methods of obtainment”, in our example, the method of obtaining, say, *the third part of 1 written in decimal notation*. Thus, instead of an *extensional object*, the classical notion of *set*:

$$\{(n, q_n) : (1 \div 3)_{(n)} = q_n\}$$

for a intuitionist our conjectures should be construed as referring to an *intentional object*, something which could be expressed in the lambda-notion as:

$$[\lambda n : \mathbb{N} . (1 \div 3)_{(n)}]$$

Another way to make a similar point would be to recall that intuitionists do have some notion of *totality* which is reminiscent of the classical notion of *set*. One can think of Brouwer’s *Species* (Van Heijenoort, 1967, p. 448) or, more contemporarily, Bishop’s and Martin-Löf’s *Sets* (Bishop, 1986, p. 410; Martin-Löf, 1984, p. 5). But these are much weaker than the classical *Sets*. Fundamentally intuitionism rejects the classical way of forming a totality *via* the Axiom of Comprehension, i.e., as the *extension of a property*, any

---

<sup>7</sup> One could call this the *ordinal* construal of infinity, as opposed to the classical *cardinal* one.

property.<sup>8</sup> Thus, only *some of classical totalities* are also *sets*, say, in Martin-Löf's sense, *but not all of them*. According to that author, a classical totality could only be called a *Set* (or a *Domain*) if one could offer exhaustive rules for forming its objects, the “canonical objects” and for deciding “when two noncanonical elements are equal”. (Martin-Löf, 1984, p. 5). We are back to the requirement of a “method of construction” again, not only regarding the *construction of the objects themselves*, but also regarding their *discrimination*.

One last point should be stressed here, before we move on. This time, our observation is not going to be about something on which classicists and intuitionists *diverge*. Quite the contrary, it is a point about which they are all in fact in perfect agreement: the *abstract nature* of mathematical objects. Whether *intensionally* or *extensionally* construed, the occupants of the nominal part of our conjectures

<i>Halts?</i>	<i>Rekurs?</i>	<i>Contains a 4?</i>
(      )	(      )	(      )

are always *abstract entities*. Both intuitionists and classicists agree that we are not framing these “conjectures” directly about *ordinary empirical processes*, i.e., specific implementations executed by some calculating agent at a particular occasion (and the partial answers that agent ends up obtaining). Our problems are mathematical conjectures, not empirical hypothesis. So according to them these conjectures are not about *concrete implementations*, but about *abstractly conceived objects* (be they intensional or extensionally construed).

This is important because as we will see even though Wittgenstein *agrees* with both parties that the distinction between *empirical propositions* and *mathematical assertions* – his *rules* – is *crucial*, for him the appeal to *abstract entities* as a way of construing such distinction is completely misguided. It is not enough to merely restrict the abstract operation of forming totalities, say, by requiring both a *method of construal* and a *method of discrimination*, as

---

<sup>8</sup> Of course, even the classicist has to agree that an unrestricted appeal to this abstraction-operation would lead to Russell's paradox. Zermelo's Separation Axiom Schema can be seen as a form of restriction on that operation, even within the classical camp.

intuitionists insist, as the intuitionist proposes. The very idea of an (abstract) *object*, or of an *abstract operation*, be it intensionally or extensionally construed, is still the result of confusion between *empirical propositions* and *mathematical rules* according to them. We will have to come back to this crucial point later, of course.

### **Methods as Programs: nothing has changed!**

Let us go back to our illustrative conjectures about the division  $1 \div 3$  and let us adopt for a while the more demanding *intuitionist construal* of such questions (the intuitionist's insistence on *methods* is certainly comprehensible to a classicist, he only believes it might be dropped as inessential in the end). Our question will then be: exactly how should we construe the intuitionistic notion of *method*, of *operation*, then? In other words, what *kinds of entities* should we take lambda terms such as

$$[\lambda n : \mathbb{N}. (1 \div 3)_{(n)}]$$

to be referring to? How should we construe the very idea of a *method of obtainment*?

One direct answer to this question is given by Martin-Löf himself:

Think of  $(\lambda x) b(x)$  as a name of the program  $b(x)$ . (Martin-Löf, 1984, p. 29)

The suggestion is thus to construe the rather abstract-looking *lambda-terms* very concretely, as *computer programs*. One must agree this is indeed a very natural proposal indeed. A *program* can be seen as *description of an operation*, its *definition*. More than that: differently from a simple description, one can even “run it”, so we can check if it's correctly constructed, if everything is alright, no missteps involved).

So, according to that proposal, when one conjectures about such properties as “*Halts?*” or “*Contains a 4?*”, one would be talking about *programs*, the “embodied definitions of those operations”. In the

context of our illustrative case we could even illustrate this with something like the pseudo-code:

```

{
  int divisor, dividend, quo, rem;
  dividend = 1;
  divisor = 3;
  quo = dividend ÷ divisor;
  rem = dividend - (quo × divisor);
  print (quo);
  while (rem ≠ 0)
  {
    quo = (10 × rem) ÷ divisor;
    rem = (10 × rem) - (quo × divisor);
    print (quo);
  }
}

```

So, if we accept such proposed identification of intuitionist’s *methods obtainment* with *computer programs*, our “conjectures” would be very concretely construed to be about *programs*. Our questions would thus be framed concerning them. Let us take, say, our question about the appearance of the digit 4. What we would really be asking, according to this proposal, would be the (apparently) very reasonable question: will we ever get such and such digits as the output of our program? Similarly with our “halting question”.

To recapitulate all that we have obtained so far, here we have a very clear-cut version of the exegete’s challenge of making some sense out of Wittgenstein’s Semantical mutation Thesis. We do frame mathematical conjectures, and in the case of recursive operations the meanings of these conjectures could be very concretely construed as being about *programs* and their *implementations*, so even the fussy intuitionists would have to accept that. What we are asking would then concern the behavior of machines while running these programs.

But now here, within this very concrete context, Wittgenstein’s insistence of some sort of “semantical mutation” appears to be completely absurd! The program’s text remains exactly the same, *before* and *after* we “settled our conjectures”. Nothing’s changed. We merely discover some new *properties* about that particular operation: it is

*cyclic*, and thus, *infinite* (and the digit 4 is not part of the quotient's cycle). That is all. No "conceptual change", no "new meaning", nothing! We merely uncovered some new attributes (*recurrence, non-halting*) of *one and the same operation, the same program*. Wittgenstein seems to be wrong. In fact, the situation is even worse than that. His proposal seems to be *utterly absurd!* What could he be thinking when he intimated some sort of "conceptual mutation" here? What mysterious "conceptual change" could have taken place, aside from our own mathematical improvement?

### What is an Effective Procedure? The intuitionistic stance

In the beginning of last section we've agreed on identifying intuitionist's lambda-terms such as

$$[\lambda n : \mathbb{N} . (1 \div 3)_{(n)}]$$

with *programs*. But, was that a good idea? Should we have accepted identifying intuitionist's *methods of obtainment* directly with *programs*? *From a purely extensional point of view* one could readily accept that there is no crucial difference between a *program* and a *method*, but should that also be the case when we regard the situation from an *intensional point of view*? Should the notion of *method* (or the notion of *algorithm*, for that matter) be identified with the notion of *program*?

We would like to begin our discussion by stressing the fact that any investigation of the idea of identifying *operations* and *programs* will end up including also an evaluation of Church's (or Turing's) thesis. The reason is immediate: any such proposals (Church's, Turing's) is bound to include some sort of identification of *intensional notions* (such as *operations, effective procedures, methods of obtainment*) with some *formal counterparts*, i.e., *documents written in some specially chosen formal language*. Thus, in Church's case the selected language would be his own *lambda notation*. In Turing's case we would have of course his famous *Turing Machines*. If we move on to a more recent proposal, Yury Gurevich's for example, despite all his

rather more general and abstract approach, we still find a crucial identification of an algorithm with texts written in a specific formal language, in his case an *algebraic fragment of first order languages*. (Gurevich & Deschowitz, 2008, p. 314)<sup>9</sup>

In all such proposals one finds an essential identification of typically *intensional notions*, such as an *operation*, with some specific *formal texts* written in some higher or lower level *programming languages*. This identification is going to be *rejected* by Wittgenstein. He is going to reject any identification of *rules* and (formal) *texts*. But before we examine his ideas, it might be useful to quickly recall that *full-blooded intuitionists also tend to reject Church's Thesis*.

Differently from Wittgenstein, their difficulty is connected to the very idea of *defining constructivity*. The notion of *method of obtainment* should be *final, absolute, primitive*. All classical attempts of defining the notion of *computable function* involve the idea of an *extensional equivalence*. But this notion is not available to an intuitionist, of course. The equivalence between ordinary arguments and formal proofs within some system could not be extensionally construed. One could not simply offer a general  $\forall\exists$  formulation, *for all* ordinary mathematical operations, *there is*, say, a Turing machine such that... The existential quantifier would have to be intuitionistically understood. One would have to actually exhibit some such method. Thus, Prawitz writes:

It seems that the notion of constructive procedure used here must be taken as a primitive notion. ... it is not possible to define it as a Turing machine that yields the value when applied to an argument; the  $\forall\exists$  in this definition must then be understood intuitionistically, (Prawitz, 1977, p. 27).<sup>10</sup>

And here again there is a rather subtle question involved: the notion *existence of a method* will have to be further distinguish from the notion of *objectual existence*, the notion of existence expressed by the intuitionistic *existential quantifier*. The idea of *existence of a method*

---

<sup>9</sup> I thank Kim Solin for calling my attention to Gurevich's work.

<sup>10</sup> Apparently this argument was first introduced by (Peter, 1959). It is also found in (Skolem, 1952, p. 123) and in (Prawitz, 1977, p. 27)

will have to be *concretely construed*, in cannot be a mere *in principle, theoretical possibility of construction* as in the case of intuitionistic *objectual existence*. Objectual *propositions of existence* (such as “ $\exists x : D (P(x))$ ”) are explained by intuitionists as the possession of a *demonstration* which assures us that  $P(a)$  is true for some inhabitant  $a$  in  $D$ . But the general idea of *demonstration* (as opposed to that of a *canonical proof*) is further explain as the possession of a method for producing a canonical proof of that proposition.

On pain of vicious regress (or circularity), one cannot go back and explain the notion of *existence of a method* appealing once again to the idea of *objectual, existentially quantified propositions*. Thus Prawitz writes:

One may ask whether this knowledge [of a method or a procedure] should not consist of a description of the procedure together with a proof that this procedure has the property required... But this would lead to an infinite regress... (Prawitz, 1977, p. 27)

*Existence of method, of a constructive procedure*, cannot involve any appeal to a *further demonstration*, as in the case of the intuitionistic interpretation of an existential quantifier. It has to be *final*. For the idea of *demonstration* itself is to be explained in terms of *method of obtainment*, and not *vice versa*.

Intuitionistically, truth of a proposition is analyzed as existence of proof... It is quite clear that the notion of existence that enters here is not the notion of existence that is expressed by means of the existential quantifier. (Martin-Löf, 1991, p. 141)

We are back to the idea of actually exhibiting some such method of translating “all possible ordinary (intuitionistic) proofs” into some specified formal system. But, so complains the intuitionist, is it reasonable to even talk about “the totality of all past, present, future intuitionistic proofs”? Would such predicate really circumscribe an *intuitionistically acceptable totality*? And further on: does it make sense to expect a translation method from the pseudo-totality of all

intuitionistically acceptable demonstrations into a specific formal system? Dummett writes:

As for Church's Thesis, this is not particularly plausible from an intuitionistic standpoint. ... (Dummett, 1977, p. 264)

This thesis [a mathematical proof or construction is essentially a mental entity] is rejection of the idea that there can even be an isomorphism between the totality of possible proofs of statements within some mathematical theory and any determinately specified totality of symbolic structures, i.e. proofs within any formal system. (Dummett, 1978, p. 200)

Differently from a classicist, an intuitionist cannot even accept the idea of a true absolutely undecidable proposition.

An intuitionist like him not only rejects any reference to a *totality of all intuitionistically sound arguments*, but he would also reject as implausible the idea of a method of translation of all those arguments into any one given formal system.

These notions [*propositions* and *judgments*] have later been replaced by the formalistic notions of formula and theorem (in a formal system), respectively. Contrary to formulas, propositions are not defined inductively. So to speak, they form an open concept. (Martin-Löf, 1984, p. 2)

### **What is an Algorithm?**

As we've anticipated before, according to our reconstruction of Wittgenstein's thought, the author of the *Philosophical Investigations* should definitively *reject all proposals of identification of algorithms and programs*. But his rejection has nothing to do with the interpretation of the existential quantifiers or the notion of *totality*. Rather his problem has to do with a key *intensional element* which is present both in the notions of *method* and of *algorithm*, and which is completely absent in the concept of *program*.

An *algorithm*, as well as a *method*, are both *teleological* notions. They are both means to a desired end. The idea of a *program* is not. A program is just some (grammatically well-constructed) *text*. There is no

*teleological component*, no reference to a *goal*, no *task to be fulfilled*, just a general constraining criterion: *grammaticality*, *well-formedness*. Thus, there is here a sharp contrast with the notions of *algorithm*, of *method*. A *method* is always a method of *attaining something*. Likewise with the notion of *algorithm*: at least in its original version (before Kleene's proposal of a new, enlarged notion of *effective operation*) something could properly be called an *algorithm only in so far* as it provided a strategy for *solving some targeted question*. This was the construal behind Hilbert's famous *Entscheidungsverfahren*, a decision-method for solving a given mathematical question (say, regarding the validity of logical formula):

...a procedure that, for a given logical expression, allows one to decide the general validity or satisfiability through a finite number of operations. (Hilbert & Ackermann, 1928, p. 72)

Kleene himself introduces the notion of *algorithm* also as a *method of obtainment* (of an answer):

In mathematics, we are often interested in having a general *method* or *procedure* for answering anyone of a certain class of questions; for example, the class of questions, "Is  $n$  a prime number?".

After the procedure has been described, if we select any question from the class, the procedure will then tell us how to perform successive steps, so that after a finite number of them we will have the answer to the question selected. ... Such methods as I have just described have had a prominent role in mathematics. They have been called "algorithms". (Kleene, 1995, pp. 15,16)

The emphasis lays entirely on the final obtainment of the intended answer:

After our performing any step to which the procedure has led us, the rules or instructions will either enable us to recognize that now we have the answer before us and to read it off, or else that we do not yet have the answer before us... (Kleene, 1995, p. 16)

Just as it was in the case of Hilbert's *Entscheidungsverfahren*, for something to be called a “*decision procedure*” it should always be crowned by the *final obtainment of the intended answer* “*after a finite number of operations*”, as Hilbert said, or, in the words of Kleene: “*after a finite number of [steps] we will have the answer to the question selected*”. Thus, the original concept of an *effective procedure* involved just that: *effectiveness*, i.e., “*the quality of being able to bring about the intended effect*”, i.e., the *answer*.

This is important because as we know it was Kleene himself who in 1938 proposed a new, enlarged concept of an *effective procedure*. Instead of concentrating on the *goal*, the *answer*, the focus was transferred to that of a *formal* (“*grammatical well formed*”) *characterization* of the “*nature of the process and the set of [its] instructions*”:

An effective or constructive operation on the objects of an enumerable class is one for which a fixed set of instructions can be chosen such that, for each of the infinitely many objects (or  $n$ -tuples of objects), the operation can be completed by a finite process in accordance with the instructions. *This notion is made exact by specifying the nature of the process and set of instructions. It appears possible to do so without loss of generality. ... If we omit the requirement that the computation process always terminate, we obtain a more general class of functions, each function of which is defined over a subset (possibly null or total) of the  $n$ -tuples of natural numbers, and possesses the property of effectiveness when defined. These functions we call partial recursive.* (Kleene, 1938, pp. 150, 151) (My emphasis)

The result, of course, was Kleene's famous new concept of a “*partially effective function*”:

... let us now call a function from any subset (proper or improper) of the  $n$ -tuples of the natural numbers to the natural numbers a *partial function*. ... For an  $n$ -tuple  $x_1, \dots, x_n$  for which  $\phi$  has a natural number as value, we say  $\phi \dots$  is defined; for an  $n$ -tuple  $x_1, \dots, x_n$  for which  $\phi$  has no natural number as value, we say  $\phi \dots$  is *undefined* ... The *range of definition* of a partial function is the set of the  $n$ -tuples

$x_1, \dots, x_n$  for which  $\phi(x_1, \dots, x_n)$  is defined. When this consists of all  $n$ -tuples, we have an ordinary (*completely defined*) number theoretic function; otherwise an *incompletely defined* function. When it is empty, we have the *completely undefined* function. (Kleene. *Introduction to Metamathematics*. Pg. 325-6)

In a quite curious “conceptual metamorphosis”, with Kleene’s new notion of *partial (effective) function* one could now have something rather surprising indeed. One could claim to have an “*effective procedure*” even though it was *completely undefined*, i.e., with the curious quirk of *failing to deliver an answer to all possible intended questions!* The constraint was not focused on *effectivity* anymore, but instead on the *formalism itself* (in Kleene’s case, general recursive formalism). Something could be an “*effective procedure*” even though it failed (completely) to be effective.<sup>11</sup>

### **Aleatory Programing, Interruption, Conclusiveness and Finalization**

The central, teleological element present in the notion of *algorithm* was substituted by a purely *extensional, formal* criterion: grammaticality. But there is a second, all-important move present in Kleene’s new notion of an “*effective procedure*”: the crucial role of the notion of *infinity*. What about “non-terminating” procedures? Should we claim that they are *not really algorithms*, just because they don’t deliver an answer in any “*finite number of steps*”? Should we discard an algorithm, just because it happens not terminate?

There would be a lot to say about this recourse to the (negative) notion of *infinity* within our characterization of an *effective procedure*. Let us now be content with commencing our discussion here the much simpler (positive) notion of *finiteness*. There is a curious ambiguity involving this term which is quite important for us. *Finiteness* can

---

<sup>11</sup> Carol Cleland makes a similar point in (Cleland, 2001). Her discussion there is somewhat hindered there because of her failure in carefully distinguishing empirical descriptions and normative criteria though.

mean *interruption* (say, of an empirical process), but it can also mean *conclusiveness*, a *final accomplishment* (say, of a task).

The first concept is clearly *empirical*, the second is not. Thus one can say of an concrete process or event (such as raining, or even the workings of computer), that it has been *interrupted* or *ceased*, that is, it is not *unfolding in time* any more. But such statement would not involve any implicature of *accomplishment*. One can *interrupt* the functioning of a desktop computer, say, simply by pulling off its plug, but this would of course have nothing to do with this machine having *successfully accomplished* any particular operation. It just means *cessation of activities*, that's all.

The situation can get even more confusing because there is a *third, normative notion* involved, that of *finalization*, i.e., *the laying down of "purely formal criteria" for closure* within a programming language. This *prescriptive notion* should not be confused, neither with the *descriptive, empirical* concept of *interruption, cessation*, nor with the teleological idea of *accomplishment*. Thus in most contemporary computer languages one finds a "*normative counterpart*" to the empirical notion of *interruption of an operation*, but this time employed, not to *describe*, but to *prescribe*, say, the transferal of control back to the main program or all the way to the operating system (C's control statement "*return*" would be a good example of this directive).

One can find this "prescriptive clauses" even in purely abstract formal languages. Let us take for example Martin Davis formal specification of a Turing Machine in his well-known book *Computability and Unsolvability*. Despite the rather misleading terminology of "machines", one cannot but carefully distinguish between the *formal language* being introduced and any concrete (*partial*<sup>12</sup>) realization of it in a concrete apparatus (a concrete machine). In his Definition 1.8 Davis introduces a "purely formal" notion of *closure*.

---

<sup>12</sup> The idea of an "infinite tape" would of course mark of the *purely abstract* character of the proposal.

DEFINITION 1.8. *An instantaneous description:  $\alpha$  is called terminal with respect to  $Z$  if for no  $\beta$  do we have  $\alpha \rightarrow \beta (Z)$ . [footnote of the author: Thus, the machine interprets the absence of an instruction as a “stop” order.] (Davis, 1982, p. 7)*

The purely formal (i.e., *normative*) character of the notion *closure* is even clearer in Post’s original formulation from 1936. There, instead of *closure* being left *implicitly* as mere *absence of further instruction*, the concept is *explicitly* listed as one of *three main kinds of operations*<sup>13</sup> of his “machine”:

- (A) Perform operation  $O_i$  [ $O_i = (a), (b), (c), \text{ or } (d)$ ] and then follow direction  $j_i$ ,
- (B) Perform operation (e) and according as the answer is *yes* or *no* correspondingly follow direction  $j_i'$  or  $j_i''$ ,
- (C) Stop. (Post, 2004, p. 290)

Now, is this normative notion of *closure* the same as the teleological concept of *accomplishment*? Of course not! How could it be equivalent to the *teleological notion* if there is no *teleology* involved? We are dealing here with a purely *grammatical determination*: “ $\alpha$  is [to be] called **terminal** with respect to  $Z$  if...”, no goal-orientation is mentioned or even considered as relevant. The notion of *closure* is of course purely grammatical and does not involve any *telos*, any means to an end.

One can better appreciate the difference between the *teleological* notion of *accomplishment* and the *purely formal* notion of *closure* (and, in general, between the teleological notion of *algorithm* and the grammatical notion of a *program*) if one considers the idea of “*aleatory programming*”. Let us use Turing’s formalism once more as our illustrative case. The “grammatical constraints” on the notion of a “Turing program” are very lenient:

---

<sup>13</sup> (a) is “marking the box”, (b) is “erasing the mark in the box”, (c) is “moving to the box on the right” and (d) is “moving to the box on the left”. (e) is “determining whether the box his is in, is or is not marked”. Thus, (a), (b) (c) and (d) would be a sort of “active capabilities” of the “machine”, and (e) would be a sort of “perceptive capability” of the “machine”. (Post, 2004, p. 289)

DEFINITION 1.3. A **Turing machine** is a finite (nonempty) set of quadruples that contains no two quadruples whose first two symbols are the same. [Footnote of the author: *This last restriction is one of consistency. That is, it guarantees that no Turing machine will ever be confronted with two different instructions at the same time.*] (Davis, 1982, p. 5)

Considering such light grammatical constraints, it is not hard for us to imagine something like our “*aleatory programming*” taking place. Let us call the first two arguments of a Turing quadruple the “*conditional arguments*” (as opposed to the last two arguments, the *active ones*). Any old list of quadruples with the sole restriction of excluding repetition of the two initial, conditional arguments would do. We could just, say, shuffle a (consistent) set of quadruples and organize them in a sequence. The important point is that here we are obviously very far from any *teleological* considerations, any task being *targeted*. The whole idea was a purely random, aimless “programming”.

Of course, regarding one of these final “aleatory program-texts”, one could still inquire whether the *formal closure clause* happened to be effective (or it belonged to the “trash-clauses”). But it is important to keep in mind here that we are in quite far any case from any notion of *accomplishment*, of a *method of obtainment*, thus from the notion of *algorithm*.

### **Back to our “conjectures”**

We now have all the elements we need to begin facing our self-imposed challenge: making sense of Wittgenstein’s strange idea of a *semantical mutation*, even when restricted to the highly confined context of recursive processes such as the periodical division  $1 \div 3$ . As we have stressed above, one could readily agree that our *program’s text* remained exactly the same, before and after our having realized recurrence of the remainder and thus the cycle involved in that operation. How could it have changed if it hasn’t been rewritten? But should we go on to say the same about the original, teleological and intensional notion of *algorithm*? Should we say that our understanding

of that operation, *qua an algorithm* (as opposed a *program's text*) remained exactly the same, before and after the realization of the cycle?

As we've seen, nothing is more crucial regarding the (teleological) concept of algorithm than its *goal*. Thus, the centrality of its *concluding clause*: the clause that determines nothing less than our finally having accomplished task which justifies us calling by that name. In the words of Kleene above, the rules of instructions which allow us recognize

... that now we have the answer before us and to read it off, or else that we do not yet have the answer before us... (Kleene, 1995, p. 16)

So let us consider the concluding clause of our algorithm for the obtainment of the *third part of 1*. In the case of that algorithm, the *concluding criterion* was clearly the final obtainment of a 0 as the remainder. The *text* of that embodied that clause, "*while (rem ≠ 0) do*" obviously didn't get transmuted just because of our realization of the cycle ("*it just sits there on the page*", in Hehner's words above). But what about our *views on the effectiveness* of that clause? Did that remained the same? It was stated there that we *should conclude* (a normative *should!*) our computation *when we've found a digit "0"* in the expansion of the remainder. But, after our realization of the cycle, what should we say about this *conclusive clause*? Does it even *make sense* now? How does one "look for a digit 0" in an expansion which cannot possibly contain anything else rather than 1's?

A direct consequence of realization of the cycle was that our *very conclusive clause*, "*while (Rem<sub>(n)</sub> ≠ 0) do*", was really *totally misconceived!* How could one ever find a digit "0" in a cyclic expansion which *does not include that digit* in its cycle? Wittgenstein writes:

Similarly, suppose someone were looking for a product whose middle digit was 4, and I said "Multiply 19 by 34".

$$\begin{array}{r}
 19 \\
 \times 34 \\
 \hline
 76 \\
 57 \\
 \hline
 646
 \end{array}$$

What would it be like to look for a 5 in this result? I can imagine what it would be like to find a £5 note in a book, but can I imagine what it would be like to find a 5 in the result of this multiplication? I could rub out 4 and write 5; but this won't do. You could object that I have *written* 5, but not *found* 5. (Wittgenstein, Wittgenstein's Lectures: Cambridge 1932-1935, 1979, p. 183)

As Wittgenstein emphasizes over and over again, mathematical rules are *not descriptive*, they are *prescriptive, normative*. We are not saying that someone, say, "little Jonny" won't ever find a "0" in *his expansion* of the remainder of the division  $1 \div 3$  (and thus, quite proudly claim to have finally finished that operation!). What we are saying is that if any *computing agent* (either human or electronic) who "supposedly implements" that operation and ends up *implementing that clause* (and thus "*halting*"), the implementation would be immediately *disqualified*. Whatever happens to that agent, it couldn't possibly qualify at the same time as having "*implemented that operation*" and having "*implemented its concluding clause*". The two conditions are completely *incompatible!* This was our realization when we finally hit on the notion of *cycle*.

Another way of saying what we want to point out is that we have now comprehended that our conclusive clause *while* ( $Rem_{(n)} \neq 0$ ) *do* is *completely innocuous within our program*. One could even "operationally erase it". We could substituted it, say, by a completely non-restrictive "1" (for "*true*"), as in "*while* (1) *do*!" The conditional clause would then simply be removed from our algorithm. It would not impose any condition whatsoever. But, based on our new mathematical insight, this new "program" would have (a normative "have") to prescribe the same operation, even though its conclusive clause has been erased! This is what we meant by calling it *innocuous*.

Of course, there is nothing in the notion of a *program* which demands any kind of *effectiveness* from it. A program is not something teleologically conceived. One can even have "random programming", as above. Its statements need merely to be *grammatically correct*, not *effective* in any way. One can have any amount of "*trash-clauses*" (i.e., *innocuous* ones) within programs, that would present no problem as far as its qualification as a *program*. It might not constitute a good

programming technique, but from a purely “extensional point of view”, it is ok. Trash clauses can still be part of the *program-text*. But should we also understand them as being part of our *algorithm*, even if they are completely ineffective? Can *innocuous-clauses* be part of a *method of obtainment* of something?

One does not usually find extensive discussions in the literature on the distinction between *trash* versus *effective* clauses. An important exception is:

Call a state  $X$  of an algorithm  $A$  *reachable* if  $X$  occurs in some run of  $A$ . ... We do not assume, however, that  $S(A)$  [the states of the algorithm  $A$ ] consists of reachable states only. ... Often it is simpler than the set of reachable states and thus more convenient to deal with. (Gurevich, 2000, p. 83)

Gurevich suggest disregarding the distinction between *trash* clauses and *effective* ones on account of *convenience* and *simplicity*. Should we accept his proposal based only on these flimsy grounds? Shouldn't we consider Gurevich's whole strategy of disregarding all *intensional notions* with no *extensional counterpart* as the very heart of the tendency we are opposing? As we said above, from the teleological point of view the acceptance of trash clauses amounts exactly to the blurring of the distinction of *formal programs* and *teleological algorithms*. In other words, according to our proposed version of Church's thesis as advocating precisely such identification, the problem regarding these clauses amounts to nothing less then accepting or rejecting Church's proposal! Trash clauses can belong to *programs*, but not to *algorithms*. This is the alternative construal we've been encouraging all along.

### **The Semantical Mutation Thesis**

Let us pause for a while and review some of the important points we've achieved so far, trying to contrast the different approaches involved. Both classicists and intuitionists accept a completely abstract notion of an *operation*. The classicist construes these *abstract entities* directly as *extensions*, i.e., as an “abstract complete lists of partial

results”, the “final, correct results of an operation operated by no one at no particular time”.

Intuitionists reject such construal as “abstract, platonic extravaganza”. They insist on the idea that all talk of infinity should always be construed through reference to an underlying *method*. One cannot talk about an infinite expansion, an “infinite list of answers”, except as a potential result of the application of a *generating method*. But one should not be deceived by concrete sounding terms such as, say, “construction”, “generating method”, etc.. The abstract character of this notion becomes clear, for example, when we consider the intuitionist’s notion of *existence* (the one expressed by the intuitionist’s *existential quantifier*, to be sure).

Once one has a *method* (a “constructive operation”, say, for “generating” Pi’s decimal expansion) and *any argument* belonging to the appropriate *argument-domain* (the natural numbers in this case) one cannot but conclude that the application of that operation on that argument somehow “*exists*” in a purely “abstract sense”. Despite Dummett’s famous protests regarding Pi’s expansion:

It seems that we ought to interpose between the platonist and the constructivist picture [a reference to Wittgenstein] an intermediate picture, say of objects springing into being in response to our probing. We do not make the objects but must accept them as we find them (this corresponds to the proof imposing itself on us); but they were not already there for our statements to be true or false of before we carried out the investigations which brought them into being” (Dummett, 1978, p. 185)

an intuitionist cannot but conclude that *there does exist* a digit corresponding, say, to the  $10^{10^{10^{10}}}$  decimal place of Pi’s decimal expansion, even if its effective calculation would demand more seconds than the history of the universe since the *Big Bang*. We have a *method* and we have a *correctly typed argument*: intuitionistically we have to accept *existence*, some sort of “abstract existence”.

The important question for us here is this: whereof does this “abstract confidence” of intuitionists (and classicists) comes from? The

answer we've suggested above was: from a pretense ability of a little program-text such as:

```

{
  int divisor, dividend, quo, rem;
  dividend = 1;
  divisor = 3;
  quo = dividend ÷ divisor;
  rem = dividend - (quo × divisor);
  print (quo);
  while (rem > 0)
  {
    quo = (10 × rem) ÷ divisor;
    rem = (10 × rem) - (quo × divisor);
    print (quo);
  }
}

```

to “run ahead of us and fix the intended “extension of a function” much before anyone ever actually calculates this or that value” - even the  $10^{10^{10^{10}}}$  decimal place of that expansion we've mentioned above. All this “abstract existence” is fully endorsed by the crucial intuitionist's notion of *possibility in principle*.

From our point of view, both the intuitionist's and the classicist's proposals are thus based on a postulation of a sort of “*definitional power*” a little text like that is supposed to have, an ability of singularly fix the entire intended expansion-list of the new operation, completely insulated from all future mathematical upheavals, of all novel mathematical revolutions. Thus according to those views, the identity of the defined-operation would be eternally secured by such *syntactical constructions*, such *program-texts*. In a sort of aristotelian vision, those documents would somehow possess a “*potentiality property*” incarnated in their very structure which would allow them to fix once and for all the very expansion intended in all future and even counterfactual occasions.

Wittgenstein's retort would be: shouldn't one accept that new mathematical advances (such as our humble realization of the division's cycle) *may have some impact* on our very discrimination of

what to classify as a (correct) implementation of the operations, quite independently of their original definitional texts? Should we really reject the idea that anyone of these new mathematical insights (*recursion*, *infinity*, etc.) could have any bearing on our judgment of implementation of those operations?

Let us imagine a full class of youngsters in the beginning of a new academic year, say, struggling to learn decimal division, but still with no concept of *recursion*. And let us picture them now, towards the end of that semester, after having realized the crucial cyclic property. Should we accept that the new mathematical insight would have no impact on their ability as implementers of that operation? Of course, one could never say that the “correction-standards” have actually *changed*: if an old calculation is *now* recognized to contain a mistake, it will be considered as *always having been wrong*, no matter the date of the new mathematical advances. It didn’t “*get to be wrong*” or “*became wrong*”.

Wittgenstein quite agrees with that: mathematical rules are *atemporal*, they don’t prescribe standards of correction only for this or that period of time.

“The 100 apples in this box consist of 50 and 50” – here the non-temporal character of “consists” is important. For it doesn’t mean that *now*, or just for a time, they consist of 50 and 50. (Wittgenstein, 1983, pp. 74, § 101)

For him, mathematical advances do have a *retroactive impact*: they can overrule past attempts as *errors*. This is part of their “grammar”, as the philosopher would say. We don’t even have to evoke Wittgenstein for that point. Concerning proofs, instead of calculations, Dag Prawitz writes:

If it turns out that we have made a mistake when trying to prove an assertion and that, in fact, what is asserted is false, then we say that we did not really have a proof. We only thought that we had a proof, but we did not really have one. This is how we use the notion of proof. (Prawitz. *Proofs Verifying Programs and Programs Producing Proofs: A Conceptual Analysis*. Pg. 83)

But the important question here is this: even if we do accept the “grammatical point” above – as Wittgenstein surely does – should one advance one step ahead then and posit an “*abstract entity*” which would somehow “incarnate the intended operation” once and forever, “eternally”? Should we follow the classicist’s metaphor and imagine a “marvelous complete list of answers”, a sort of “mathematical meter-of-Paris, carefully preserved in some platonic ideal museum”? Or, alternatively, should we insist, as the intuitionists do, on introducing a “*quasi-empirical force*”, an “*abstract trigger*”, which would remain *eternally operative* and, once again, function as a sort of “*abstract paradigm*” for that operation? *Do we need* to posit the “*list*”, or else, the “*force*”? Could we not remain satisfied solely with the “grammatical note” above – mathematical advances have a retroactive impact – and let the rest go?

One could adopt Church’s intentional lambda notation in order to further elucidate Wittgenstein’s alternative view vis-à-vis the more traditional proposals. According to both intuitionists and classicists, we are dealing with one and the same abstract object, either *before* or *after* the “mathematical improvement”. As we have seen, for the classicists this abstract object would be an extensional construed entity, a *Set*:

$$\{(n, q_n) : (1 \div 3)_{(n)} = q_n\}$$

whereas for an intuitionist the enduring equally abstract object would be an intentionally construed entity, a *Method of obtainment*:

$$[\lambda n : \mathbb{N}. (1 \div 3)_{(n)}]$$

With the intentional notion at our disposal, one could represent Wittgenstein’s *semantical mutation* quite graphically by saying that, contrary to the more familiar proposals, in Wittgenstein’s construal of the mathematical episode, one has a different concept of *division*, before and after the mathematical improvement. Before the advance, one would equate the concept of the *division* with whatever *satisfied the algorithm*. Thus, employing Church’s intentional lambda notion, we could represent that concept as:

$[\lambda ep : Emp_{op}. \textit{Satisfaction of the Algorithm}(ep)]$

After the proof, according to Wittgenstein, a number of *new definitional notes* (new *Merkmale*) were added to the original construal of that operation, new definitional criteria for its implementation, thus completely revamping the old notion into a quite novel conception of that simple operation. Once again we could represent this new concept as:

$[\lambda ep : Emp_{op}. \textit{Sat the Algthm}(ep) \wedge \textit{Rekurs}(ep) \wedge \textit{NonTerminates}(wp) \wedge \dots]$

We could then finally fulfill our self-imposed task of explaining just what conceptual change Wittgenstein had in mind in a conjecture-settling (in a situation like that of the division  $1 \div 3$ ) by directly “*drawing*” the intended conceptual mutation, say, as:

$[\lambda ep : Emp_{op}. \textit{Satisfaction of the Algorithm}(ep)]$       **Before the mutation**

$[\lambda ep : Emp_{op}. \textit{Sat the Algthm}(ep) \wedge \textbf{Rekurs}(ep) \wedge \textbf{NonTerminates}(wp) \wedge \dots]$       **After the mutation**

As we pointed out above, even our views on the *text* of the original definition (the *program*) have changed. The very conclusive clause of the original definition of that operation was discarded as *misconceived*; it was *necessarily inapplicable*. We have thus a completely new intentional stance towards that operation, an intentional view which according to Wittgenstein should never be equated to the original conceptions.

And the discovery of the periodicity is really the construction of a new sign and a new calculus. For it is misleading to say that this discovery consists in our having

*noticed* that the first remainder is equal to the dividend. For if we had asked someone unacquainted with periodic division: "Is the first remainder in this division equal to the dividend?" of course he would have answered "Yes"; so he would have noticed this. But in noticing it he wouldn't have had to notice the periodicity; that is, in noticing it he would not have discovered the calculus with the signs  $a \div b = c$ . (Wittgenstein, 2005, p. 451)

## Bibliography

- Bernays, P. (1986). Comments on Remarks on the Foundations of Mathematics. In S. Shanker, Ludwig Wittgenstein: Critical Assessments (pp. 165-182). London: Routledge.
- Bishop, E. (1986). Selected Papers. Philadelphia: World Scientific.
- Cleland, C. (2001). Recipes, Algorithms, and Programs. *Minds and Machines*, 11, pp. 219-237.
- Davis, M. (1982). Computability and Unsolvability. New York: Dover.
- Diamond, C. (1991). The Realistic Spirit. Cambridge: The Mit Press.
- Dummett, M. (1977). Elements of Intuitionism. Oxford: Clarendon.
- Dummett, M. (1978). Truth and Other Enigmas. Cambridge: Harvard University Press.
- Frascolla, P. (2004). Wittgenstein on Mathematical Proof. In A. Coliva, & E. Picardi, Wittgenstein Today (pp. 167-184). Padova: Poligrafo.
- GERRARD, S. (1991). Wittgenstein's Philosophies of mathematics. *Synthese*, 87, pp. 125-142.
- GLOCK, H.-J. (1996). A Wittgenstein Dictionary. London: Wiley-Blackwell.
- GUREVICH, Y. (2000, July). Sequential Abstract-State Machines Capture Sequential Algorithms. *ACM Transactions on Computational Logic*, 1 (1), pp. 77-111.
- Gurevich, Y., & Deschowitz, N. (2008). A Natural Axiomatization of Computability and a Proof of Church's Thesis. *Bulletin of Symbolic Logic*, 14 (3), pp. 299-350.
- HILBERT, D., & Ackermann, W. (1928). Grundzüge der Theoretischen Logik (1st ed.). Berlin: Springer.
- KLEENE, S. (1938). Notation for Ordinal Numbers. *Journal of Symbolic Logic*, 3 (4), pp. 150-5.

- KLEENE, S. (1995). Turing's Analysis of Computability, and Major Applications of it. In R. Herken, *The Universal Turing Machine: A Half-Century Survey*. Wien: Springer-Verlag.
- Mac Lane, S. (1986). *Mathematics: Form and Function*. New York: Springer-Verlag.
- MARION, M., & Okada, M. (2012). Wittgenstein et le lien entre la signification d'un énoncé mathématique et sa preuve. *Philosophiques* , 39 (1), pp. 101-124.
- MARTIN-LÖF, P. (1991). A Path from Logic to Metaphysics. In *Atti del Congresso Nouvi della logica e della filosofia della scienza* (pp. 141-149). Bologna: Clueb.
- (1984). *Intuitionistic Type Theory*. Naples: Bibliopolis.
- PANJVANI, C. (2006). Wittgenstein and Strong Mathematical Verificationism. *The Philosophical Quarterly* , 56 (224), pp. 406-425.
- PETER, R. (1959). *Rekursivität und Konstruktivität*. In A. Heyting, *Constructivity in Mathematics*. Amsterdam: North-Holland.
- PORTO, A. (2013). Rule Following and Functions. *O que nos faz pensar* , 33, pp. 63-109.
- POST, E. (2004). Finite Combinatory Processes. Formulation I. In M. Davis, *The Undecidable: Basic Papers on Undecidable Propositions, Unsolvability Problems and Computable Functions* (pp. 288-291). Mineola: Dover Publications.
- PRAWITZ, D. (1977, April). The conflict between Classical and Intuitionistic Logic. *Theoria* , 43, pp. 2-40.
- PUTNAM, H. (2007). Wittgenstein and the Real Numbers. In A. Crary, *Wittgenstein and the Moral Life Essays in Honor of Cora Diamond* (pp. 235-250). Cambridge: MIT.
- Säätelä, S. (2011). From Logical Method to 'Messing About': Wittgenstein on 'Open Problems' in Mathematics. In M. McGinn, & O. Kuusela, *The Oxford Handbook on Wittgenstein* (pp. 162-182). Oxford: Oxford.
- Schroeder, S. (2012). Conjecture, Proof, and Sense, in Wittgenstein's Philosophy of Mathematics. In C. Jäger , & W. Löffler, *Epistemology: Contexts, Values, Disagreement. Proceedings of the 34th International Ludwig Wittgenstein Symposium in Kirchberg* (pp. 459-471). Frankfurt: Ontos.

- SHANKER, S. (1987). Wittgenstein and the Turning-point in the Philosophy of Mathematics. London: Croom Helm.
- Skolem, T. (1952). Sobre la Naturaleza del Razonamento Matematico. *Gazeta Matematica* , 4, pp. 113-124.
- Van Heijenoort, J. (1967). From Frege to Gödel: a source book in Mathematical Logic, 1879-1931. Cambridge: Harvard.
- WITTGENSTEIN, L. (1976). Lectures on the Foundations of Mathematics, Cambridge, 1939. Chicago: The University of Chicago Press.
- (1975). *Philosophical Remarks*. Oxford: Basil Blackwell.
- (1983). *Remarks on the Foundations of Mathematics*. Cambridge: Mit.
- (2005). *The Big Typescript*. Oxford: Basil Blackwell.
- (1979). *Wittgenstein's Lectures: Cambridge 1932-1935*. Totowa: Rowman & Littlefield.
- (1979). *Wittgenstein's Lectures: Cambridge 1932-1935*. Totowa: Rowman & Littlefield.
- WRIGHT, C. (1980). *Wittgenstein on the Foundations of Mathematics*. Cambridge: Harvard.