Is the concept of computation a conceptual fixed point?

Paula Quinon

Abstract

Conceptual engineering is a philosophical activity that involves the reframing or refinement of a given concept to better suit specific scientific, philosophical, or social contexts (Cappelen 2018). Haslanger's (2000) work on the concepts of "gender" and "race" is often cited as a flagship example of conceptual engineering, not only for its methodological interest, but also for Haslanger's social-justice "ameliorative" project, which has prompted sustained debate. More recently, formal or mathematical concepts have also attracted the attention of conceptual engineers. However, not all concepts can be engineered. Some need to be kept as *fixed points* in order to preserve the intended conceptual structure of their respective frameworks, or because they have properties that make them *natural fixed points* that play the same role consistently across different frameworks. Fixed points have been studied in formal contexts where concepts such as "truth," "existence," and "quantifier" are considered to be resistant to engineering (Eklund 2015). In this paper, I explore the theoretical concept of computation as explicated by the Church-Turing Thesis, and argue that it constitutes a natural conceptual fixed point.

Keywords: conceptual engineering, conceptual fixed points, computation, the Church-Turing Thesis, open texture, informal rigour, Carnapian explications

Contents

Introduction

Conceptual engineering is "an activity that many people—not just philosophers—engage in." It is underpinned by an "intellectual attitude" or "cognitive disposition", and leads to a methodological approach to philosophy (and possibly other disciplines) that involves adapting or refining concepts to better serve specific scientific, philosophical, or social contexts (Cappelen 2018 [12]). One example is the concept of gender (as opposed to the concept of sex) which has been (and continues to be) widely discussed in the political context, starting with the proposal to replace the then-current understanding of the concept of woman with the following ameliorative understanding:

> S is a woman iff S is systematically subordinated along some dimension (economic, political, legal, social, etc.), and S is "marked" as a target for this treatment by observed or imagined bodily features presumed to be evidence of a female's biological role in reproduction" (Haslanger 2000 [35, page 39]; see also Eklund 2021 [29, page 16]).

Conceptual engineering was first developed as a philosophical approach to ethical concepts (Burgess, Cappelen & Plunkett 2020 [9]) and has since then attracted the attention of researchers working in other fields, including the areas—which interests me in this paper—the formal philosophy (Eklund

2015 [28]; 2021 [29]; Sharp 2013 [69]) and mathematics (Linnebo & Shapiro 2019 [45]).

As a methodological stance, conceptual engineering is not new; it has been used implicitly and explicitly in philosophy for a long time. Case studies of conceptual engineering can be found in many normative or revisionist projects. It has been compared to other ways of working with concepts, most notably Carnap's method of explication (Carnap 1950 [16]). More recently, Tanswell (2018 [75]) has compared it to Waismann's (1968 [80]) and Shapiro's (2006 [67] and 2013 [68]) conception of open texture, and Dean and Kurokawa (2021 [23]) to Kreisel's (1987 [41]) informal rigour, which I will discuss in more detail later in this paper.

It has been observed that not all concepts can be engineered. Some must be maintained as *fixed points* to preserve the intended conceptual structure of the framework to which they belong. Others are *natural fixed points* that, often due to their formal properties, always play the same role in relation to other concepts within a conceptual framework, regardless of the specific conceptual context. The relationship between fixed points and natural fixed points is that the latter represent a more robust form of conceptual stability, transcending particular frameworks. The notion of conceptual fixed points was primarily formulated in the context of ethics. For example, Cueno & Shafer-Landau (2014 [21, page 405]) argue that propositions such as "it is wrong to engage in the recreational slaughter of a fellow person" must belong to every moral system. This notion is now being explored in other contexts, particularly—and this is crucial for this paper—in formal domains where concepts such as "truth," "existence," or "quantifier" are considered impossible to engineer (Eklund 2015 [28]; 2021 [29]).

Building on previous results in formal philosophy, in this paper I extend the conceptual engineering perspective to the concept of the theoretical concept of computation[1] as explicated by the Church-Turing Thesis. Furthermore, I argue that the concept of computation is a *natural* conceptual fixed point.

In **Section 1**, I introduce conceptual engineering, placing it in historical and philosophical perspective. In **Section 2**, I introduce and comment on

[1]I choose to use the expression "computation" and not "computability" despite the preference for the latter in the community. This decision reflects my intention to emphasize my interest in the process of computing itself, rather than studying what is possible to be computed.

the phenomenon of conceptual fixed points. **Section 3** is devoted to the concept of computation, where, after characterizing this formal concept from several methodological perspectives, I present four reasons that support my thesis that the concept of computation is a natural conceptual fixed point.

1. Conceptual engineering

1.1. Roots of conceptual engineering

What we now call conceptual engineering existed earlier in philosophy as a certain methodological attitude toward conceptual work. As Cappelen says (2018 [12, page 24]), the history of philosophy can be seen as "in large part a battle between descriptivists and revisionists". These has been observed and characterised already by Strawson in the introductory section of *Indiviuals* (1959 [73]):

> Descriptive metaphysics is content to describe the actual structure of our thought about the world, revisionary metaphysics is concerned to produce a better structure ([73, page 9]).

Revisionist projects reappear over the years under different names such as normative or *ameliorative* projects (Haslanger 2000 [35]), or again "meaning negotiations" (Ludlow 2014 [46]).

One of the most frequently cited projects in early (not explicitly qualified as such) conceptual engineering is Haslanger's ameliorative social justice project, in which she works on concepts such as "gender" or "race".[2] The project is *ameliorative* because it is based on the belief that the meanings of the concepts we use influence the thoughts we entertain and shape our vision of the world. Haslanger says the following about her proposed amelioration of gender and race terms (see also Saul 2006 [64, page 138]):

[2]Haslanger's ameliorative definition of "woman" (Haslanger 2000 [35, page 39]) has prompted a lively and multifaceted debate, which it is not the place to report upon in this paper. To mention only a few contributions concerning inclusion, see Jenkins (2016 [38]), Andler (2017 [1]), Barnes (2020 [5]), and Dembroff (2018 [24]; 2020 [25]). Haslanger herself has been open to multiple, purpose-relative gender concepts; see her later remarks that "we need several concepts" for different theoretical purposes (Haslanger 2012 [36, page 230]).

> by appropriating the everyday terminology of race and gender, the analyses I've offered invite us to acknowledge the force of oppressive systems in framing our personal and political identities. Each of us has some investment in our race and gender: I am a White woman. On my accounts, this claim locates me within social systems that in some respects privilege and in some respects subordinate me. Because gender and racial inequality are not simply a matter of public policy but implicate each of us at the heart of our self-understandings, the terminological shift calls us to reconsider who we think we are. (Haslanger 2000 [35, page 47])

I have already mentioned Haslanger's proposal for reformulating the concept of "woman," where the meaning based on biological sex is replaced by a reference to the gender role played in society by the individual referred to as a woman.

> The motivation, put briefly, is that there is oppression of the kind described, and employing a concept that captures that is a way of raising the fact to salience, in a way that helps combat this oppression (Eklund 2021 [29, page 16]).

Conceptual engineering as a distinct methodology first emerged as conceptual ethics, defined as concerned with normative and evaluative issues (see Burgess & Plunkett 2013a [10]; 2013b [11]; and Cappelen & Plunkett 2020 [13] for comprehensive introductions and comparison of the two terms). "Ethics" in "conceptual ethics" refers to a wider context than the field of moral philosophy. It includes "the study of what one should or ought to do (dually, what can permissibly be done)" as well as "the study of which actions and outcomes are good or bad, better or worse". "Thus, this use of "ethics" is not meant to privilege moral/political norms in particular (vs., e.g., those that find their central home in epistemology, metaphysics, aesthetics, etc.)" ([10, page 1094–1095]).

Cappelen and Plunkett (2020 [13]; see also Eklund 2021 [29]) briefly review conceptual engineering-like undertakings from the past. They quote Nietzsche (1901/1968 [50, page 220–221]); they evoke the "founding work of analytic philosophy in the early twentieth century"; they call Frege's *Begriffsschrift* (1879/1967 [31]) an outstanding example of conceptual engineering; they emphasize the role of Wittgenstein (1922/1961 [83]) and, of course,

draw attention to Carnap's work on explication (1950 [16]), because, they argue, "work on explication and language choice are paradigms of conceptual engineering".

In the very influential *The Philosophy of Philosophy* (2007 [82]), Williamson refocuses the central interest of (analytical) philosophy on work on concepts that he calls the "conceptual turn". The expression "conceptual engineering" before becoming the hype notion of the contemporary methodology, was used by Blackburn in *Think* (1999 [6]) who famously declared that he prefers to speak about *conceptual engineering* instead of speaking of *philosophy* which "carries unfortunate connotations: impractical, unworldly, weird."

> I would prefer to introduce myself as doing conceptual engineering. For just as the engineer studies the structure of material things, so the philosopher studies the structure of thought. Understanding the structure involves seeing how parts function and how they interconnect. It means knowing what would happen for better or worse if changes were made. This is what we aim at when we investigate the structures that shape our view of the world. Our concepts or ideas form the mental housing in which we live. We may end up proud of the structures we have built. Or we may believe that they need dismantling and starting afresh. But first, we have to know what they are. (Blackburn 1999 [6, page 8]).

1.2. Characterisation of conceptual engineering

When it comes to conceptual engineering as a general philosophical attitude, there is still a lot of work to be done and debate to be had. Its revisionist, ameliorative aspect, where motivation is more important than its method, has begun to share space with a strong theoretical and speculative aspect. This shift has already taken place in conceptual ethics. One might be interested in conceptual ethics because "one is trying to actually change existing thought and talk", or again the interest might "involve trying to actually change conceptual or linguistic practices", but "someone might be interested in conceptual ethics purely as an interesting part of philosophical theorizing" (Cappelen & Plunkett 2020 [13, page 5]).

> This is parallel to how studying normative political philosophy might help those interested in changing actual existing political institutions, or how studying normative aesthetics might help

> those creating art. But just as it would be a mistake to think of political philosophy solely in terms of the role it might play for the project of creating better political institutions, so too would it be a mistake to think of conceptual ethics solely in terms of the role it might play in practical projects of changing actual conceptual or linguistic practices. (Cappelen & Plunkett 2020 [13, page 5]).

Brandom (2001 [8]) distinguished *conceptual analysis* from *conceptual engineering* in a way that recalls the methodological openness to extra-philosophy resources characteristic of Maddy's *Second Philosophy* (see Maddy 2007 [47]). He observed that conceptual engineering "appeals to the tools of the special sciences (for instance, information theory and evolutionary biology) to describe abstractly, but in criticizable detail, how one might craft a situation in which some state arguably deserves to be characterized as 'representationally contentful' in various important senses" [8, page 587]. This approach contrasts with *conceptual analysis*, which, following the Cartesian *First Philosophy* (Descartes 1641 [26]), involves a purely theoretical manipulation of concepts without recourse to external references.

Cappelen and Plunkett (2020 [13]) think that "conceptual engineering is prior to or more fundamental than all other philosophical disciplines" because "reflection and argumentation in any part of philosophy must rely on concepts" ([13, page 4]). They say that "conceptual engineering is concerned with the assessment and improvement of concepts":

> Conceptual engineering =
>
> (i) The assessment of representational devices [such as concepts, lexical items or semantic values],
> (ii) reflections on and proposal for how to improve representational devices, and
> (iii) efforts to implement the proposed improvements [13, page 3].

Similarly to Brandom, Cappelen and Plunkett speak of conceptual engineering in terms of Maddy's Second Philosophy. Conceptual engineering draws on, they say, "insights from philosophy of language, philosophy of mind, epistemology, political philosophy, philosophy of science, ethics, and other fields" [13, page 4].

Unlike the First Philosophy method, which realizes the ideal of conceptual analysis, conceptual engineering is closer to techniques that use external knowledge to adapt concepts to specific linguistic contexts. The three approaches most closely related to conceptual engineering were also deliberately designed for the context of scientific or formal concepts. Carnap sought to introduce clear concepts into the language of science. Kreiselian "philosophical proofs" were conducted with informal rigor to validate the informed intuitions of experts in logic and mathematics. Shapiro adapted Waismann's idea of concepts having an open or closed texture to a formal context. Even if conceptual engineers were not primarily concerned with formal, scientific, or mathematical concepts, the related 'ameliorative' or 'revisionist' projects mentioned above certainly contributed to awakening an interest in such concepts.

Explicitly recalling conceptual engineering, Scharp (2013 [69]) explores the idea that the concept of truth is inconsistent and as such provokes paradoxes. He argues that it therefore needs to be revised to better serve advanced theoretical work in fields such as linguistics and logic. Tanswell (2018 [75]) engineers set-theoretical concepts. Linnebo and Shapiro (2019 [45]) work with the concept of infinity. Clark and Chalmers (1998 [18]) revise the epistemological concept of belief.

2. Conceptual fixed points in conceptual engineering

The idea that some concepts cannot be engineered recurs regularly in philosophical thinking. Such a role has been assigned to logical constants (Quine 1980 [56]; Quine 1986 [57]; Dummett 1981 [27]); rigid designators; Chalmers (2011 [19]) speaks of "bedrock concepts"; Eklund (2021 [29]) searches for "the thinnest" normative concepts; Linnebo (2018 [44]) uses principles of abstraction as a tool in the search for "the thinnest" formal concepts; in cognitive sciences, core cognitive systems core generate conceptual content that is innate and forms building blocks that are believed to underlie all further conceptual developments (Spelke (2000 [72]), Carey 2009 [14]).

2.1. Conceptual fixed points in ethics

Conceptual fixed points were primarily formulated, again, in the context of ethics, and are now being studied in other contexts. In naturalist moral philosophy, "moral fixed points" are those moral claims that are moral truths that must be contained in a moral system. A normative system that does

not include such claims is not a moral system, but a normative system of a different kind. A flagship example of such a moral fixed point is the claim that "it is wrong to engage in the recreational slaughter of a fellow person" (Cueno & Shafer-Landau 2014 [21, page 405]).

An important work has been done in conceptual ethics where formal ethical concepts such as "good", "right", "ought" are considered the "thinnest" normative words and claimed to be fixed points.

> [C]an there be words non-coextensive with our "good", "right", etc. but with the same evaluative and normative roles? If not, then a certain kind of conceptual engineering isn't possible: that of finding alternative concepts with the same normative roles as our actual concepts but different in extension. (Eklund 2015 [28, page 381])

Eklund (2015 [28]) asks us to imagine the following scenario:

> There is a linguistic community speaking a language much like English, except for the following differences (and whatever differences are directly entailed): while their words "good", "right" and "ought" have the same evaluative and normative roles as our words "good", "right" and "ought" have, their words aren't coextensive with our "good", "right" and "ought". So even if they are exactly right about what is "good" and "right" and what "ought" to be done, in their sense, and they seek to promote and to do what is "good" and "right" and what "ought" to be done in their sense, they do not seek to promote what is good and right and what ought to be done. [28, pages 380-381].

2.2. Conceptual fixed points in formal philosophy

Eklund (2015 [28]) introduces the idea of conceptual fixed points in formal philosophy. Such a concept, which is according to him, impossible to engineer is the concept of truth. Not all researchers stand on this position. As described above, Scharp (2013 [69]) wants to engineer the concept of truth which in its current form is inconsistent and provokes paradoxes. Eklund, on the contrary, writes:

> People care about the truth, and they do not care about some conceptually engineered concept "truth*".

Eklund uses his observation that "people care" to argue that "truth" should not be engineered and it should remain as a natural concept relating to assertions and beliefs, and be regarded as a fixed point that serves as a reference for other elements from a conceptual framework. Similarly, according to Eklund "existence" is a conceptual fixed point, and there are not—as sometimes assumed in contemporary meta-ontological debates—alternative concepts of existence. According to Eklund, the conceptual framework that would emerge from the adaptation of conceptually engineered concepts such as 'truth' or 'existence' would have to accommodate other key concepts in such a manner that the resulting and initial frameworks are isomorphic. Thus, "One cannot, so to speak, *selectively* engineer the quantifier" (2015 [28, chapter 6]; see also Quinon 2020 [61, page 12]).

2.3. Conceptual fixed points in model theory

In order to examine what a conceptual fixed point and a natural conceptual fixed point mean in the case of the concept of computation, it is useful to explain the matter in terms of model theory[3]. The model-theoretic approach examines the mathematical structures, called models, that satisfy a given set of axioms or sentences. It explores how formal language is interpreted within these models. In the philosophical context that is mine, we distinguish:

- A standard model (identified up to isomorphism);
- A more philosophically nuanced intended model (e.g., identified up to computable isomorphism);
- Non-standard models.

This taxonomy provides a framework for understanding the concept of computation as a conceptual fixed point in different contexts.

A formal concept is a fixed point if its meaning cannot further evolve in the standard model; in other words, if a formal concept is a conceptual fixed point, its explicatum (formal definition) is closed-textured. This means it is clear which elements (or tuples of elements) satisfy the formal definition, and this clarity is preserved up to isomorphism in all standard models.

[3]At this point in the paper, I will refrain from committing myself to any particular model theory. Instead, I aim to provide general intuitions that are accessible to the potentially wide range of readers.

Natural conceptual fixed points might shift in meaning but retain their relational framework with neighbouring concepts, even when their interpretation is extended to non-standard models. In these cases, the meaning of concepts co-definable with or closely related to the natural fixed point concept is readjusted. This re-adjustment ensures that the relationships between these concepts remain the same.

If we assume that the concept of computation is a natural fixed point, this means that computation retains a fixed meaning within its intended context, and also that when its meaning shifts in a non-intended context, it retains its position within the conceptual network because it induces shifts of meaning in neighbouring concepts.

A note on terminology: My use of the terms "fixed point" and "diagonalisation" (used later in this paper) deviates from their well-established meanings in mathematical logic, where they are often associated with Cantor's diagonal argument or Gödel's incompleteness theorems. In this paper I adapt these terms to convey a different, though related, sense.

In mathematical logic, a fixed point is a value that remains unchanged by a particular function or operation. This concept is crucial in many areas of mathematics, including topology and functional analysis. In the context of Gödel's work, fixed points refer to self-referential statements in formal systems—statements that, in a sense, "point to themselves". Diagonalisation is a technique, famously used in Cantor's proof that the set of real numbers is uncountable, which involves constructing an element (such as a real number) that is guaranteed to be distinct from every element in a given list. In the context of Gödel's incompleteness theorems, diagonalisation is used in a more abstract sense to construct statements about a formal system that are inherently self-referential, leading to conclusions about the limitations of the system.

When the concept of fixed point, as it is understood in conceptual engineering, is applied to the concept of computation, it is used in a metaphorical way, with a slight allusion to its original mathematical use. One could say that just as a mathematical fixed point remains invariant under certain operations, the concept of computation retains its essence or meaning through various arithmetical models or philosophical contexts. In my study of the concept of computation as a conceptual fixed point, I have adopted the concept of diagonalisation to express the idea that this concept retains its place in the structure of neighbouring concepts. In my view, the technique of diagonalisation reveals inherent limitations or fixed properties within formal

systems, analogous to the way in which computation consistently retains its fixed conceptual identity across different contexts.

In other words, a fixed point consists of the pair: *an un-engineerable concept*, which corresponds to the intended meaning of the concept—or, to use Eklund's expression, the interpretation that "people care about"—and *the intended or standard model*. When it comes to natural conceptual fixed points, the concept retains its intended placement beyond neighbouring concepts and preserves the conceptual framework.

3. Computation

The concept of computation to which my research is devoted—symbolic computation over discrete, countable domains—has been the subject of intense study and formalisation since the early 20th century, culminating in the Church-Turing Thesis (CTT). The intuitive concept of computation has been formalised in many different ways, and it has been shown that all these formalisations are extensionally equivalent, i.e. they all point to the same class of functions, regardless of the specific formalisation.

There has been ongoing debate about whether any one aspect of the informal concept of computation should be privileged and serve as a basis for further inquiry into what it means to "compute" when other intuitions are involved. These discussions began with Gödel favoring Turing's account of computation. Shagrir (2002 [65]) distinguishes between formalizations of what can be done by an idealized human who computes by means of effective procedures (Turing (1936 [78]), Kreisler (1987 [41])) and of what a machine can do (Gandy (1980 [32])). Shapiro (1982 [66]) argues that computation should first be thought of as defined on inscriptions. Rescorla (2007 [62]) argues that computation should first be thought of as defined on abstract objects. Soare (1997 [71]) distinguishes between procedural (defined on Turing Machines) and descriptive computability (defined with recursive functions). (Trakhtenbrot 1995 [77]) distinguishes between computations defined for hardware and computations defined for software. Boker and Dershowitz (2008 [7]) investigate a possibility to define computation without referring to any domain of computation.

An ongoing debate seeks to establish the conditions and modalities of the existence of an absolute core to the concept of computation. Gödel was the first to argue not only for the existence of such a core but even for an independent absolute concept of computation. This debate continues with

proponents of the absolute concept of algorithm (for a recent overview see Papayannopoulos 2023 [51]).

While the concept of computation has not yet been the subject of an explicit conceptual engineering process, it has been examined through approaches that share a similar methodological attitude. This section examines the concept of computation in terms of its open texture, its sharpening through informal rigour, and its explication through Carnapian methods, before arguing that computation is a *natural* conceptual fixed point.

3.1. Open texture

Conceptual engineering has been compared to various philosophical approaches to concepts and language. From my perspective, three are particularly important: Waismann's open texture (as interpreted by Shapiro), Kreisel's informal rigour, and Carnap's explications.

Open texture is a semantic property of concepts in language. Most concepts, its advocates claim, are open-textured, meaning their application cannot be fully determined in advance for all possible cases, particularly unexpected or novel ones. Some concepts are closed-textured, meaning their application is fully determinate in any possible context. A key feature of open texture is that even if we try to make the application conditions of a concept more precise through explicit definitions or rules, this cannot eliminate open texture. Such specifications work only for cases we can currently imagine, but open texture arises precisely when genuinely novel situations emerge that were not foreseeable.

Waismann's original view of open texture primarily concerns empirical terms from ordinary language and, to a lesser extent, empirical terms from scientific language. Theoretical terms, usually having straightforward semantics, remain outside his interest. Empirical concepts often lack a fixed extension, with the meaning of terms shifting according to context, and language users can never be sure that every possible situation in which a term might be used is covered. Waismann discusses "the essential incompleteness of an empirical description" (Waismann 1968 [80, page 121]).[4] He illustrates this with examples such as the everyday concept of a cat:

> if it showed some queer behavior usually not to be found with cats, say, if, under certain conditions it could be revived from

[4]See also Tanswell 2018 [75, pages 833sqq] for a comprehensive introduction.

> death whereas normal cats could not? Shall I, in such a case, say that a new species has come into being? Or that it was a cat with extraordinary properties? [80, page 121].

From this, Waismann concludes that a concept is open-textured if there exist objects about which it is unclear whether they fall under its scope.

Waismann attributes open texture primarily to empirical terms, most extensively from ordinary language (e.g., "cat"), but also from the language of natural sciences (e.g., "gold"). Regarding abstract mathematical terms, Waismann appears to hold the view that mathematical concepts are invariably closed-textured, and thus do not exhibit the open texture characteristic of empirical concepts. Indeed, his counter-examples for incomplete empirical terms are mathematical concepts:

> Goldbach's hypothesis [...] may be undecidable [...] But this in no way detracts from the closed texture of the mathematical concepts. If there is no such thing as the (always present) possibility of the emergence of something new, there could be nothing like the open texture of concepts. [80, pages 123-4].

However, the idea that mathematical concepts can exhibit open texture is a relatively recent addition to the philosophical landscape. Inspired by Lakatos (1976 [42]), who demonstrated that mathematical concepts are subject to changes in meaning through refinement and development, Shapiro (2006 [67]) was the first to systematically apply the idea of open texture to mathematical concepts, focusing on basic terms such as "natural number" and "computation". In detail, Shapiro argues that after a long history of being open-textured and undergoing several transformations, the concept of computation is today sufficiently precisely understood to be considered closed-textured.[5]

Tanswell's (2018 [75])[6] focuses on set-theoretic terms. He compares open texture to conceptual engineering, seeing the latter as providing tools to explore concepts in the context of open and closed texture.

[5]Papayannopoulos (2023 [51]) builds directly on Shapiro's work and argues that the concept of algorithm remains open-textured.

[6]Tanswell (2018 [75, pages 833sqq]) discusses details of the difference between Waismann's and Shapiro's understanding of open texture. The main difference "seems to come down to the difference between the potential for *sharpening* concepts vs. the potential for *extending the domain* of the concepts."

Vecht (2023 [79]) embarks on a general study of open texture and closed texture in mathematics. In order to determine which mathematical concepts have the potential to become closed-textured, he first sharpens the definition of open texture, distinguishing it from vagueness and linking it to notions of analyticity. He then argues that open-textured concepts are characterised by their capacity to be redefined in response to cases outside their standard domain of application, whereas closed-textured concepts are those that can be defined in an algebraic manner, allowing uniform application in any domain.[7]

This transition of the concept of computation from open to closed texture is consistent with the formulation of various formal models of computation, all of which have been proven to be extensionally equivalent. This equivalence suggests a robust, common core to the concept of computation, supporting the idea that it may be a conceptual fixed point.

Shapiro (2006 [67, page 441]) suggests that in the 1930s, and for some time afterward, the notion of computation[8] was subject to open texture. The concept was not delineated with enough precision to decide every possible consideration concerning tools and limitations. The mathematical work of Turing, Church, Post, Kleene, and others served to set the parameters and thus sharpen the original pre-theoretic notion.

> The present conclusion is that the notion of computability in use now is the result of the last seventy years of examining texts like Turing [1936] and of the overwhelming ensuing success of the theory of computability. It is not accurate to think of the historical proofs of CT and related theses as establishing something about absolutely sharp pre-theoretic notions. Rather, the analytical and mathematical work served to sharpen the very notions themselves. Sieg says as much, noting that the analyses result in a "sharpening [of] the informal notion". Once again, this is the norm in mathematics. (Shapiro 2006 [67, page 441])

[7] Vecht (2023 [79]) further observes that closed-textured concepts correspond to structuralist philosophy in mathematics, as they are characterised by the common properties of all systems that satisfy their defining axioms. This structuralist view, Vecht notes, reinforces the algebraic nature of closed-textured concepts by treating them as abstract frameworks rather than objects bound to specific instances.

[8] Shapiro uses the expression "computability", but as by footnote 1 I stick to "computation" throughout this paper.

3.2. Kreisel's informal rigour

Kreisel's method of informal rigour[9]. provides another perspective on the concept of computation. This method, which has been the least discussed in the context of conceptual engineering[10], is particularly interesting in the form proposed by Kreisel, who focused in particular on the concept of computation in the sense of the Church-Turing Thesis.

Informal rigour involves analysing common ideas in mathematical terms, obtaining definitions and properties in a rigorous way: without using formal deduction, but still providing a "philosophical proof". Kreisel focuses strongly on indicating the kind of reasoning that can be used in the process of sharpening or specifying the meaning of a pre-scientific concept. It also suggests that this process is best carried out by a restricted community of experts.

Kreisel states:

> IR is involved in the 2000-year-old tradition of analysing common notions in mathematical terms (current at a given time). In particular, definitions and other properties of such notions are established in rigorous ways. When this way consists of inspection by the mind's eye, one speaks of 'axioms' in the old sense of this word, now called 'informal'. When this way consists of formal deduction from established knowledge, one speaks of formal rigour. (Kreisel (1987 [41, page 502])

It's worth noting that there is no strict definition of the method of informal rigour as Kreisel sees it. Thus, although 'informal rigour' is often taken to be a hallmark of Kreisel's work, there is little consensus about how this method should be characterised or what kinds of questions it should address.

Kreisel considered the concept of computation to be a prime example of informal rigour. The application of informal rigour to computation has led to a "philosophical proof" of what "computation" refers to. This includes intuitions related to the finitude of processes, step-by-step procedures, and the discreteness of the domain of computation.

[9]The model of informal rigour proposed by (Kreisel 1967 [40]; also 1987 [41]), developed by Isaacson (2011 [37]), Barton (2020 [2]), and Dean and Kurokawa (2021 [23]), Leitgeb (2009 [43]) and Rav (1999 [63]); see Tanswell (2024 [76]) for an overview.

[10]Dean and Kurokawa (2021 [23]) mention it briefly.

Shapiro points out examples that, in his view, reached Kreisel's high expectations:

> Turing's argument [...] is [an] instance of what Kreisel calls "informal rigor", and it serves to sharpen a concept of "algorithm". (Shapiro 2006 [67, page 443]).

Shapiro (2006 [67, page 444]) also points at Mendelson's (1990 [49, page 233]) observation:

> The so-called initial functions are clearly [...] computable; we can describe simple procedures to compute them. Moreover, the operations of substitution and recursion and the least-number operator lead from [...] computable functions to [...] computable functions. In each case, we can describe procedures that will compute the new functions. (...) Mendelson concludes that this "simple argument is as clear a proof as I have seen in mathematics, and it is a proof in spite of the fact that it involves the intuitive notion of [...] computability".

Shapiro highlights that he agrees with Mendelson on this point and he calls his argument a rather straightforward example of Kreisel's informal rigor and the possibility to prove things about an intuitive, pre-formal notion.

The epistemological status of the Church-Turing Thesis and similar theses is unclear. However, the community generally agrees that these theses, which make significant use of intuitive pre-scientific language and rely strongly on intuition regarding the correctness of inferential chains, cannot be subject to formal proof. Definitions, implicit definitions, axiomatic systems are classical ways to "philosophically prove" matters about the pre-theoretic notions. (Shapiro 2006 [67, page 430]).

3.3. Carnapian explications

The method of Carnapian explication is the most explicitly regulated method of introducing a new concept into scientific language. It has also been recognized as a paradigmatic form of conceptual engineering (*e.g.*, Cappelen & Plunkett 2020 [13]; Tanswell 2018 [75]).

Carnapian explication transforms an intuitive concept (the *explicandum*) into a precise formal or scientific concept (the *explicatum*) suitable for theoretical purposes. To illustrate this method, consider a pocket knife: extremely useful in many ordinary situations but inadequate for tasks requiring high precision. While it might cut simple shapes in plywood, it would

be unsuitable for creating complex forms in oak. For such tasks, a stainless steel laser woodcutter with a micro-fence would be more appropriate (Carnap 1963 [17, pages 937–938]). The pocket knife, however, remains useful for camping trips. Just as the pocket knife serves everyday needs while specialized tools serve technical purposes, the intuitive explicandum remains valuable in ordinary discourse while the formal explicatum provides the precision required for scientific theorizing.

The method of explication, according to Carnap, consists of two steps: (1) the *clarification* of the explicandum; (2) the *specification* of the explicatum. The often underappreciated clarification stage involves choosing the intension of the intuitive concept that will best serve as a basis for formalization in the targeted context. The specification stage then formulates the exact concept in this context.

In the case of Church's Thesis, the clarification stage involved identifying the intended meaning of computability in the context of functions on natural numbers. This clarification focused on the notion of effective calculability for functions defined on positive integers, aligning with Church's aim to characterize the class of constructively defined functions.

The specification stage then formulated recursive functions within axiomatic number theory, building on work in a well-established mathematical context by Peano, Dedekind, and Hilbert in proof theory and axiomatic number theory.

The specificity of Carnap's proposed method of explication is that it does not require any strict (extensional or intensional) adequacy between explicandum (the informal concept to be clarified) and explicatum (the formal concept that results from the explication), and for this reason cannot be evaluated in terms of right or wrong. Instead, Carnap asks us to check whether the explicatum is satisfactory for the context in which it is formalized, using four general requirements that not only allow us to assess the adequacy of the explicandum, but also facilitate the comparison of different explicata of the same concept explication: *similarity* (explicatum should in some way "correspond" to explicandum), *exactness* (explicatum should be characterised with exact terms), *fruitfulness* (explicatum should enable formulation of universal rules), *simplicity* (explicatum should be formulated with basic terms of the targeted theory) (Carnap 1950 [16, page 7]).

As Quinon (2021 [60]) demonstrates, the Carnapian explication of computability has proven tremendously fruitful. Despite Carnap's own disinterest in this concept, the Church-Turing Thesis exemplifies the transformation

of an informal concept into a formal one, satisfying all adequacy criteria.

3.4. Comparisons

In relation to the concept of computation, the three methodological approaches or methods presented above operate in similar epistemological settings. To facilitate comparison between these approaches, I examine the concept of computation as analysed by each of them. I will use a framework that is explicit in Carnap and less emphasised, but present in Kreisel and Waismann and Shapiro: the concept of computation is understood as consisting of two counterparts: the intuitive or prescientific and the formal. I will also assess whether the approach assumes the existence of an absolute concept or whether the concept is relativised to a specific context. If the latter is the case, I consider the possibility that the concept is formed from the basic part and the part relative to the context. Note that if there is a common basic part, this might be equivalent to saying that there is an absolute concept.

"Open texture" and "closed texture" refer to the semantic status of concepts. A concept has open texture when there exist actual or possible contexts in which it is not decided if the concept would refer or not. It has closed texture when the extension is always determined. Shapiro's main concern is the adequacy between intuitions and formalization, moving back and forth between the two sides of the Church-Turing Thesis, ensuring that intuitions correspond precisely to formalization and vice versa. He discusses how the formalization of the concept in various formal models of computing sharpened intuitions that humans started with, adding constraints such as "computability concernes the (idealized) abilities of humans following algorithms, or of (idealized) mechanical computing devices, or something similar" (Shapiro 2006 [67, page 421]). With a strongly clarified intuitive concept, the formal concept or concepts got identified as precisely reflecting certain intuitions about computing. In contrast to the Carnapian setting, the "ideal" of the closed texture suggests an absolute concept of computation—or at least something common to different formalisations of the concept in different contexts. Carnap, on the other hand, proposes the development of equally adequate formalisations that are relativised to different scientific theories, and his explanations are not intended to uncover the common core of these relativised concepts, nor to identify an absolute concept.

Informal rigour refers declaratively to the intuitive counterpart and, as I see it, aims at a perfect clarification, even if Kreisel was against such a

comparison (1987 [41]). It is an axiological position that points out what kind of methods and who can use them to grant the concept to reach something that could be compared to a formal concept with a closed texture. The absoluteness of the concept that informal rigour seeks to achieve is explained by Kurokawa and Dean as a means by which different domains of concepts can be profitably brought into contact, rather than separated into disjointed frameworks. They write:

> According to the model we have proposed [. . .], an informally rigorous argument is understood as one carried out jointly by using constitutive principles for common concepts, bridging principles which connect them to novel and mathematical concepts, as well as a precise mathematical background theory. It would thus seem that Kreisel regarded informal rigour as a means by which different domains of concepts can be profitably brought into contact via the multi-stage procedure we have attempted to codify via the schema IR rather than one which segregates them concepts into disjoint 'frameworks'. (Dean and Kurokawa 2021 [23, page 57]).

One point that stands out is that Kreisel (1987 [41]) seems at least open to a form of conceptual realism within the mathematical domains that were his primary interest.

Carnapian explication, as described by Carnap himself, is:

> The task of making more exact a vague or not quite exact concept used in everyday life or in an earlier stage of scientific or logical development, or rather of replacing it by a newly constructed, more exact concept, belongs among the most important tasks of logical analysis and logical construction. We call this the task of explicating, or of giving an explication for, the earlier concept. (Carnap 1947/1956 [15, pages 8–9])

The difference between Kreisel's informal rigour and the Carnapian approach became the subject of debate. Bar-Hillel (see comments to [40]) argues that Kreisel places too much faith in the ability to have a game-changing insight into the intuitive counterpart, and compares Kreisel's efforts to the Carnapian method. He equated informal rigour with clarification, and formal rigour with working towards a formal concept. Kreisel disagreed and replied:

> Concerning the equation
>
> clarification of the explicandum = informal rigour
>
> =
>
> providing the explicatum = formal rigour
>
> two things are to be said. First, strictly speaking, the equation does not hold because Carnap certainly denies the possibility of informal rigour or proof; he would not accept the problem of finding the correct explicatum and proving it, but speaks of 'replacing' the prescientific explicandum by an 'adequate' explicatum. Carnap does not reject the possibility of proof outright, but feels convinced of the impossibility or fruitlessness of such a proof as a result of his experience. The examples of the paper are intended to remind us of fruitful cases. (Kreisel 1967 [40, page 176]).

Kreisel highlights two main differences: (1) the lack of criteria for evaluating the clarification process, leading to several equivalently acceptable forms, and (2) that formal concepts replace the intuitive ones in Carnap's approach.

Kurokawa and Dean compare the Carnapian method with Kreisel's ideas, noting similarities in their aim "to make sense of informal concepts for better integration into mathematical or scientific reasoning". However, they also highlight differences in the intended scope, methods, and goals (Dean and Kurokawa 2021 [23, page 56]), drawing on Kreisel's objections.

The concern that the intuitive counterpart becomes unimportant once the corresponding formal concept has been formulated seems to me to be unjustified. Carnap emphasises the importance of keeping informal concepts in use in several instances, such as in the pocket knife illustration above, , where the everyday tool retains its value despite the existence of specialized alternatives. As for the objection that several equivalent forms are possible, this is a matter that cannot be discussed objectively outside a particular philosophical point of view.

An important observation is that both closing the texture and applying informal rigour involve a back and forth process between the intuitive and formal counterparts. In this process, the person engaged in clarification or philosophical proof is constantly inspired by the formal counterpart of the concept under consideration, shaping intuitions on the basis of what is formally possible.

3.5. What does it mean to engineer the concept of computation?

Conceptual engineering focuses on refining or readjusting a concept already integrated in an established context (e.g. formal) to fit this context evolution or for a new context. In essence, it's re-engineering an already clarified or engineered concept.

This paper focuses on the classical concept of computation, as encapsulated today by the Church-Turing Thesis. As demonstrated in previous sections, the earlier versions of the intuitive counterpart underwent significant refinement and development, although this process was not explicitly labelled as "engineering". These efforts, obviously, revealed much about the structure of the intuitive counterpart. Importantly, they also provided insights into the formal counterpart, going beyond merely adapting the concept to different formal contexts. In my view, the concept of computation is multilayered and can be analyzed along two dimensions: the distinction between its intuitive and formal aspects, and the distinction between its absolute or core characteristics and its context-dependent characteristics.

The intuitive concept is clarified in its core aspect according to whether it exhibits open or closed texture and to Kreisel's informal rigour. Here, "clarification" is used in an extra-Carnapian sense, encompassing methodologies beyond Carnapian explication. When a concept determined to be closed-textured through informal rigour is "translated" or expressed in the language of the formal or of another structured language, the concept of computation becomes embedded in a formal structure. This process is elegantly modeled by Carnapian explications.

The concept of computation can be formulated in several specific formal scientific contexts. If yet another formal context were to incorporate the concept of computation, engineering would involve finding a way to formulate it within the language and rules of that new context. This paper argues that there is an interesting feature of the engineering of the concept of computation: whatever the context, the concept behaves in a specific way, maintaining the same relations to other elements of the conceptual structure. This characteristic makes it a very specific kind of conceptual fixed point, called a "natural" fixed point. A fixed point is natural if, as its meaning changes, all the terms in the interconnected network change their meanings in such a way that the structure and internal relations between these terms remain intact.

3.6. Computation as a conceptual fixed point

In my previous work (Quinon 2020 [61]), I argued that to avoid conceptual circularity within the philosophy of arithmetic approached through conceptual analysis[11], one must decide which term to take as primitive or undefined[12]. I suggested that the concept of computation emerges as a viable candidate. These investigations led me to conclude that the concept of computation behaves as a conceptual natural fixed point.

A *conceptual fixed point*[13] is a term that cannot or should not be engineered, given its pivotal role within the interconnected web of related concepts. A *natural conceptual fixed point* is a term that, if its objective meaning changes, all terms within the interconnected web also shift their meanings in such a way that the structure and internal relations between these terms remain intact.

There are several reasons justifying my claim that the concept of computation is a natural fixed point. In what comes, I outline these reasons. While my primary focus is on the formalised concept of computation, I am confident that many of my arguments are also relevant and applicable to the informal clarification of computation. This confidence is informed by Shapiro's insight into the phenomenon of open and closed texture and Kreisel's efforts in applying informal rigour.

3.6.1. Extensional equivalence of the formal models of computation

An argument frequently employed to justify the correctness of the Church-Turing Thesis is the provable extensional equivalence between different models of computation (Cleland 2006 [20, page 133]; Svozil 2006 [74, page 493]). In a similar vein, Firz 2006 [30, page 202] speaks of "robustness," where "robust" suggests a common background for different extensively equivalent

[11]I have carried out my investigation without deciding on any particular formalisation or axiomatisation of arithmetic, and I have referred to arithmetical concepts under their standard or, as we shall soon see, intended interpretations.

[12]This approach differs from what Gödel apparently hoped for. As Church wrote, "His [Gödel's] only idea at the time was that it might be possible, in terms of effective calculability as an undefined term, to state a set of axioms which would embody the generally accepted properties of this notion, and to do something on that basis". Gödel subsequently moved to the idea of absolute concept of computation.

[13]This concept is potentially analogous to an algebraic invariant. See also Vecht (2023 [79]) as observed in the footnote 7 above.

formalizations. Pour-El, while exploring a less extensively studied "Church's Theorem" for Banach spaces, wrote:

> The situation is reminiscent of the one in ordinary recursion theory, when the various definitions, proposed by Turing, Herbrand/Gödel, Church, Post and others, all intuitively convincing, were proved to be equivalent. The notion of a computability structure acts as a unifying concept, since seemingly different definitions of computability, are, in fact, equivalent because of this unicity. (Pour-El 1999 [52, page 450]).

This extensional equivalence suggests that regardless of which intuitions we choose to clarify, in what domain we choose to formalize them, and with what formal vocabulary, the class of computable functions will always remain the same.

3.6.2. Absolute computation

Gödel believed in the existence of a domain-independent "absolute" concept of idealized computation. For him, the work on defining computability undertaken by mathematicians in the 1930s was "an excellent example [...] of a concept which did not appear sharp to us but has become so as a result of a careful reflection" (reported by Wang 1974 [81, page 84]). He saw the culmination of this research in Turing's analysis and claimed that it is "absolutely impossible that anybody who understands the question and knows Turing's definition should decide for a different concept" ([81, page 84]). Gödel confirmed his standpoint in several other places, for instance in (193? [33, page 168]) he wrote that "the correct definition of mechanical computability was established beyond any doubts by Turing". In Gödel's words, "[...] with this concept [Turing's computing] one has for the first time succeeded in giving an absolute definition of an interesting epistemological notion" (Gödel 1946 [34, page 1419]).

Based on his belief in the absolute concept of computation and his conviction that Turing's analysis is the most adequate analysis of the concept of computation, Gödel claimed that—precisely because of equivalence with Turing's analysis—all other explications of the concept of computation are correct (Gödel 193? [33, page 168]).

The second reason supporting my claim that the concept of computation is a conceptual fixed point and as such cannot be engineered is precisely the

idea that within the concept of computation there exists a core part corresponding to the "absolute" concept of computation that cannot be discussed or changed. This ideal functions as what Kant would call a regulative ideal[14] and it is true that Gödel's ideal is far from being realized—we do not know how to account for what it means to compute without reference to the domain of computation—and that many believe it never will be. I argue that the mere existence of the ideal is sufficient to support my claim. Its regulative necessity—the fact that we cannot coherently conceive of computation without presupposing some sort of invariant core—establishes that the concept resists engineering or revision.

3.6.3. Conceptual vicious circles

The third argument supporting my claim that the concept of computation is a conceptual fixed point is based on the observation that the concept of computation, when subjected to conceptual analysis, easily falls into conceptual vicious circles, *i.e.*, it is not possible to explain how to capture the computability of a sequence of symbols—which is necessary to define what "to compute" means—without using the concept of computation in the first place. This suggests that it depends little on other concepts or that it depends only on concepts with which it is inter-definable, such as the concept of natural number. A conceptual analysis of this concept leads to a conceptual vicious circle.

In its simplest version the problem of vicious circle is formulated as follows:

> The core of the problem discussed in this paper is the following: the Church Turing Thesis states that Turing Machines formally explicate the intuitive concept of computability. The description of Turing Machines requires description of the notation used for the input and for the output. The notation used by Turing in the original account and also notations used in contemporary handbooks of computability all belong to the most known, common, widespread notations, such as standard Arabic notation for

[14]Kant's notion of a "regulative ideal", differs sharply from the Platonic Idea. Kant explicitly criticises Plato for treating Ideas as real, supra-sensory entities existing independently of possible experience, thereby detaching reason from the empirical data that give thought its content and limits. Kantian ideals make no such ontological claim; they serve instead to guide inquiry. (Kant 1781/1787 [39, A313/B370—A319/B375])

> natural numbers, binary encoding of natural numbers or stroke notation. The choice is arbitrary and left unjustified. In fact, providing such a justification and providing a general definition of notations, which are acceptable for the process of computations, causes problems. This is so, because the comprehensive definition states that such a notation or encoding has to be computable. Yet, using the concept of computability in a definition of a notation, which will be further used in a definition of the concept of computability yields an obvious vicious circle. (Quinon 2018 [59, page 338])

The vicious circles arise independently of the domain and the theory of formalization of the concept, and also independently of the philosophical standpoint (constructivism, realism, radical realism) (see Quinon 2018 [59]).

Eklund (2015 [28, pages 380–381]) argues, in the context of formal concepts he considers to be conceptual fixed points, that the conceptual framework arising from an adaptation of a conceptually engineered concept of "existence", "truth", or "quantifier" would have to accommodate its other concepts in a way that preserves isomorphism between the resulting and the initial framework. Preserving isomorphism is necessary because what characterizes conceptual fixed points is that, no matter how they are modified, they mean the same thing in the end, or more precisely, they play the same conceptual role, because they enter into the same relations with other concepts of the framework. I argue that the same is true of the concept of computation. The fact that it behaves in a similar manner, entering into similar relations regardless of theory and domain, suggests that it plays the same fixed role in each theoretical context.

3.6.4. Diagonalization

The final argument supporting the claim that the formal concept of computation is a conceptual fixed point, which I will call "diagonalization", demonstrates that computation is not only a conceptual fixed point, as the preceding arguments show, but also a *natural* fixed point. This means that computation retains a fixed meaning within its intended context, and also that when its meaning shifts in a non-intended context, it retains its position within the conceptual network because it induces shifts of meaning in neighbouring concepts.

For example, in a non-standard model of $PA1$, where the computable (or recursive) has a different meaning than in the intended model, it will encapsulate the "natural numbers" of that model. This includes the standard part, reflecting the set $\mathbb{N}$, and a non-standard part, consisting of elements beyond the largest standard natural number, extending indefinitely with no upper limit.[15]

Similarly, if the behaviour of arithmetic functions is governed by second-order logic, so that the theory is categorical (all models are identified up to isomorphism), then the concept of computation on non-intended models (those that form a non-computable sequence) will be non-recursive from the perspective of the intended model, but will be consistent with "natural numbers" from its model.

As I proposed to see it above, a fixed point consists of the pair: an unengineerable concept, corresponding to the intended meaning of the concept that "people care about", and the intended or standard model. Generally, it's often useful to think that each symbol from the language is interpreted only in the relative model or models.

In my study of computation as a conceptual fixed point, I've adopted the concept of diagonalization to express the idea that this concept retains its place in the structure of neighbouring concepts. The technique of diagonalization reveals inherent limitations or fixed properties within formal systems, analogous to how computation consistently retains its fixed conceptual identity across different contexts.

This discussion pertains to the broader question of reference for mathematical terms, a debate famously initiated by Putnam's 1980 paper in response to the Skolem Paradox, relating the relativity of set-theoretic terms.[16] The debate has since expanded to encompass the relativity of arithmetical

[15]It's worth noting that "natural numbers" sometimes refer only to those in the standard part, and sometimes to all elements. The structure of non-standard models is complex and extends beyond the scope of this discussion.

[16]The Skolem Paradox, formulated by Skolem in 1922, posits that the relativity of set-theoretic terms results from the Löwenheim-Skolem theorems, which state that set theory has countable models. Putnam's "model-theoretic argument" challenges the traditional realist view that there exists a single interpretation of theoretical terms that can perfectly describe the world. Instead, Putnam suggests a form of anti-realism where our understanding of the world is shaped not only by the world itself but also by our conceptual frameworks and languages. Knowledge is thus inherently influenced by human perspective and limitations.

concepts and the recursivity of arithmetical functions[17], as discussed by Benacerraf (1965 [3] and 1996 [4]) and further developed in the context of Tennenbaum's theorem's philosophical usefulness.

Summarizing the entire discussion here would obscure my objective. The pertinent aspect is Dean's paper, where the author compares Tennenbaum's theorem to the Löwenheim-Skolem result, highlighting that these theorems, like all theorems, "on their own [have] nothing like a 'reference-fixing' effect for the language of arithmetic [or set-theory]" ([22, page 162]). Thus, in the case of Tennenbaum's theorem, the central significance lies in model-theoretical relativity and the concept of computation. It is simply a fact, akin to other non-definability and non-categoricity results.[18]

A useful distinction, originating from Putnam (1980 [54]), is between interpretation inside the model and interpretation from outside the model.[19] This distinction has been extensively used in the meta-theory of model theory, differentiating between internal and external computation.

Internal computation refers to the operations and computations carried out within a model according to its rules and axioms. When we say "model M thinks that..." it means that within the logical framework and axioms of model M, a certain statement or theorem is *true*. This is an internal perspective, focused on events within the model's universe. *External computation* refers to the analysis, operations, or computations applied to the model from an outside perspective, usually involving a meta-theoretical stance. Here, the model is studied and understood from the viewpoint of an external observer

[17]Putnam in (1980 [54]) stated that he did not think any problems arise in the case of natural numbers.

[18]Dean posits that Tennenbaum's theorem has two consequences for the debate on model-theoretic skepticism. The first, most relevant to my current endeavor, is the "demonstration of our ability to show that if our language were interpreted non-standardly, the extension of predicates like 'recursive' or 'uncountable' would provably diverge from their intended interpretations" ([22, page 162]). The second consequence is that Tennenbaum's theorem influences our understanding of what it means to introduce a model, highlighting that constructing theories with only non-standard interpretations implies that none of such models can have a recursive atomic diagram. This means our ability to refer to non-standard models is mediated by descriptions that are inherently indefinite and can never be made fully constructive. We cannot go beyond linguistic descriptions to characterize a non-standard model of $PA1$.

[19]This distinction, rooted in Gödel's Incompleteness Theorems, helps differentiate between what can be proved within a particular arithmetic system and what can be understood from an external vantage point.

with access to a broader mathematical framework or set of tools than those available within the model itself.

In axiomatic arithmetic, these concepts are particularly important because they distinguish what can be understood or proved within a particular arithmetic system (like Peano arithmetic) from what can be understood or proved from an external vantage point (such as in set theory or a more comprehensive logical system).

The distinction, beyond other sources, comes from Kurt Gödel's Incompleteness Theorems[20]. These theorems underline the fundamental distinction between what can be known or proven within a system (the internal perspective) and what can be understood or proven from an outside perspective (the external perspective). However, the explicit discussion of internal versus external perspectives in model theory, as understood today, has been developed by later logicians and philosophers building upon Gödel's work.

Synthesizing these ideas, we can define internal and external computation more precisely:

- *Internal computation* occurs when:
  - Symbols are interpreted as computable functions and the model of interpretation is intended, or
  - Symbols are interpreted as incomputable and the model is non-standard or standard non-intended.
- *External computation* involves evaluating the behavior of a non-intended model (non-standard or standard) from the perspective of the "real world," maintaining the intended interpretation.

In this framework, internal computing represents a *natural fixed point*, while external computing is a fixed point in the broader sense. Diagonalization demonstrates that if computation cannot be apprehended from a realist perspective of the "real world," it remains a *natural fixed point*.

[20]Gödel's work highlights the difference between what a model can establish internally and what can be understood about the model from an external standpoint. Gödel's First Incompleteness Theorem demonstrates that any sufficiently powerful and consistent formal system cannot prove all truths about arithmetic operations, implying that there are propositions true but unprovable within the system. The Second Incompleteness Theorem asserts that such a system cannot prove its own consistency.

A concept internally defined within a non-intended model can be viewed as an engineered concept. From the external perspective, it is not computable and thus not engineered computing. However, from the internal perspective, its meaning is context-dependent, making it a *natural fixed point*.

Tennenbaum's theorem confirms the existence of these diagonalization pairs. It states that in an enumerable model of arithmetic, if the functions of addition and multiplication are computable, then the model must be the intended model of arithmetic (Quinon & Zdanowski 2007 [58]).[21]

Importantly, the argument from diagonalization holds regardless of our position on the vicious circle argument. This is because, irrespective of the model of interpretation, computation always plays the same role in the overall framework or model.

Conclusions

The primary aim of this paper has been to demonstrate that the concept of computation represents a conceptual fixed point. To support this claim, four key arguments were presented, all of which converge on the notion of conceptual stability.

First, the observed extensional equivalence across various formalizations of computation strongly suggests the existence of a common core underlying these diverse representations. This shared essence points to a fundamental stability in the concept. Second, the idea of an absolute concept of computation, whether realizable or not, indicates that the concept of computation is fixed and resistant to further engineering. Gödel believed this absolute concept was best realized in Turing's account, which is extensionally equivalent to other formalizations. Even if we consider this absolute computation as a Kantian ideal that we can approach but never unambiguously define, its very existence as an ideal suggests that computation in its pure form is a fixed point. This notion of an absolute, albeit potentially unreachable, concept of computation further reinforces its status as a conceptual fixed point, immune to substantial modification through engineering efforts.

Third, the tendency of the concept of computation to invariably lead to vicious circles reinforces its stability. This pattern demonstrates that computation consistently belongs to a particular structure of interdependent

[21] I do not want to provide more technical details in this paper to keep it accessible to a large group of scholars interested in conceptual engineering.

concepts. Lastly, the argument from diagonalization highlights that the concept of computation, as used by people in reference to the intended model of arithmetic, aligns with the concept that people intend to use.

A crucial aspect of these arguments is that they are predominantly based on formal properties rather than pragmatic observations about concept usage. The exception is the diagonalization argument, which does touch on how people intend to use the concept. This formal grounding distinguishes the concept of computation from other fixed points and supports its classification as a *natural fixed point*.

The stability demonstrated by these arguments is not merely a matter of linguistic convention or practical usage. Instead, it reflects deep-seated formal properties that persist across various contexts and formalizations. This persistence suggests that the concept of computation occupies a unique position in our conceptual framework, resistant to engineering efforts that might alter its fundamental nature.

In conclusion, the concept of computation, by virtue of its extensional equivalence, its potential to correspond to an absolute concept (whether realizable or as a Kantian ideal), its role in conceptual structures, and its alignment with intended usage, demonstrates the characteristics of a natural fixed point. This status has significant implications for our understanding of computation in various domains and suggests that attempts to fundamentally alter or "engineer" this concept may be inherently limited. Future research might explore the implications of this fixed point status for related fields such as artificial intelligence, cognitive science, and the philosophy of mind.

References

[1] Andler, D. (2017). Conceptual engineering and gender: A reply to Jenkins. *Journal of Social Ontology*, 3(1): 77–85.

[2] Barton, N. (2020). Forcing and the universe of sets: Must we lose insight? Journal of Philosophical Logic. 49: pages 575–612.

[3] Benacerraf, P. (1965).What Numbers Could not Be. The Philosophical Review 74(1): 47–73.

[4] Benacerraf, P. (1996). Recantation or Any old ω-sequence would do after all. Philosophia Mathematica, 4 (2): pages 184–189.

[5] Barnes, E. (2020). Gender and gender terms. *Noûs*, 54(3): 704–730.

[6] Blackburn, S. (1999). Think: A Compelling Introduction to Philosophy. Cambridge University Press.

[7] Boker, U. and Dershowitz, N. (2008). The Church-Turing Thesis over Arbitrary Domains. In: Avron, A., Dershowitz, N. & Rabinovich, A. (eds.). Pillars of Computer Science: Essays Dedicated to Boris (Boaz) Trakhtenbrot on the Occasion of His 85th Birthday. Lecture Notes in Computer Science, Vol. 4800, Springer-Verlag: 199-229.

[8] Brandom, R. (2001). Modality, Normativity, and Intentionality. Philosophy and Phenomenological Research 63: 587–609.

[9] Burgess A., Cappelen H. & Plunkett D. (eds.) (2020). Conceptual Engineering and Conceptual Ethics. Oxford University Press.

[10] Burgess A. & Plunkett D. (2013). Conceptual Ethics I. Philosophy Compass 8/12: 1091–1101.

[11] Burgess A. & Plunkett D. (2013). Conceptual Ethics II. Philosophy Compass 8/12: 1102–1110.

[12] Cappelen, H. (2018). Fixing Language: An Essay on Conceptual Engineering. Oxford University Press.

[13] Cappelen H. & Plunkett D. (2020). Introduction. A Gided Tour of Conceptual Engineering and Conceptual Ethics. In: Burgess A., Cappelen H. & Plunkett D. (eds.). Conceptual Engineering and Conceptual Ethics. Oxford University Press: 1–26.

[14] Carey, S. (2009). The Origin of Concepts. Oxford University Press.

[15] Carnap, R. (1947/1956). Meaning and Necessity: A Study in Semantics and Modal Logic. University of Chicago Press.

[16] Carnap, R. (1950). Logical Foundations of Probability. Routledge and Kegan Paul.

[17] Carnap, R. (1963). Reply to P. F. Strawson on linguistic naturalism. In: Schilpp, P.A. (ed.), The philosophy of Rudolf Carnap. Open Court Publishing.

[18] Clark, A. & Chalmers, D.J. (1998). The Extended Mind. Analysis 58(1): 7–19.

[19] Chalmers, D. J. (2011). Verbal Disputes. The Philosophical Review 120 (4): 515–566.

[20] Cleland, C.E. (2006). The Church-Turing Thesis. A Last Vestige of a Failed Mathematical Program. In: Olszewski, A., Woleński, J. & Janusz, R. (eds.). Church's Thesis After 70 Years. Ontos-Verlag: 119–146.

[21] Cuneo, T. & Shafer-Landau, R. (2014). The moral fixed points: new directions for moral nonnaturalism. Philosophical Studies 171: 399–443.

[22] Dean, W. (2014), Models and Computability. Philosophia Mathematica 22 (2): 143–16

[23] Dean, W. & Kurokawa H. (2021) On the methodology of informal rigour: Set theory, semantics, and intuitionism, forthcoming in: Anonutti-Marfori, M. & Petrolo, M. (eds.) Intuitionism, Computation, and Proof: Selected themes from the research of G. Kreisel, Springer.

[24] Dembroff, R. (2018). Real talk on the metaphysics of gender. *Philosophical Topics*, 46(2): 21–50.

[25] Dembroff, R. (2020). What is gender? *Philosophers' Imprint*, 20(9): 1–27.

[26] Descartes, R. (1641/1996). Meditations on First Philosophy: With Selections from the Objections and Replies. Cottingham, J. (trans. and ed.). Cambridge University Press 1996.

[27] Dummett, M. (1981). Frege: Philosophy of Language. Harvard University Press.

[28] Eklund, M. (2015). Intuitions, Conceptual Engineering, and Conceptual Fixed Points. In: Daly, C. (ed.) The Palgrave Handbook of Philosophical Methods. Palgrave Macmillan: 363–385.

[29] Eklund, M. (2021). Conceptual Engineering in Philosophy. In: Khoo, J. & Sterken, R.K. (eds.) The Routledge Handbook of Social and Political Philosophy of Language: 15–30.

[30] Fitz, H. (2006). Church's Thesis and Physical Computation. In: Olszewski A., Woleński J. & Janusz R. (eds.) (2006). Church's Thesis After 70 Years. Ontos-Verlag: 175–219.

[31] Frege, G. (1879/1967). Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens, Halle a. S.: Louis Nebert; translated as Concept Script, a formal language of pure thought modelled upon that of arithmetic, by S. Bauer-Mengelberg. In: van Heijenoort J. (ed.). From Frege to Gödel: A Source Book in Mathematical Logic, 1879–1931. Harvard University Press 1967.

[32] Gandy, R. (1980). Church's thesis and principles for mechanisms. In: Barwise, J., Keisler, H. J., & Kunen, K. (Eds.) The Kleene symposium. (pp. 123–148). North-Holland Publishing Company.

[33] Gödel, K. (193?), Undecidable Diophantine Propositions. In: Feferman, S., Dawson, J. & Kleene, S. (eds.). Kurt Gödel: Collected Works, Volume III, Unpublished essays and lectures. Oxford University Press 1995: 164–175.

[34] Gödel, K. (1946), Remarks Before the Princeton Bicentennial Conference on Problems in Mathematics. In: Feferman, S., Dawson, J. & Kleene, S. (eds.). Kurt Gödel: Collected Works Volume II. Oxford University Press 1995: 150–153.

[35] Haslanger, S. (2000). Gender and race: (what) are they? (what) do we want them to be? Noûs 33(13): 459–480.

[36] Haslanger, S. (2012). *Resisting Reality: Social Construction and Social Critique*. Oxford University Press.

[37] Isaacson, D. (2011). The reality of mathematics and the case of set theory. In: Noviak, Z. & Simonyi, A., (eds.) Truth, Reference, and Realism, Central European University Press: 1–75.

[38] Jenkins, K. (2016). Amelioration and inclusion: Gender identity and the concept of woman. *Ethics*, 126(2): 394–421.

[39] Kant, I. (1781/1787 [1999]). Critique of Pure Reason. Guyer, P. & Wood, A. W. (eds. and trans.). Cambridge University Press.

[40] Kreisel, G. (1967). Informal Rigour and Completeness Proofs. In: Lakatos I. (ed.), Problems in the philosophy of mathematics: Proceedings of the International Colloquium in the Philosophy of Science. North-Holland 1967: 138–186.

[41] Kreisel, G. (1987). Church's Thesis and the Ideal of Informal Rigour. Notre Dame Journal of Formal Logic. 28(4): 499–519.

[42] Lakatos, I. (1976). Proofs and Refutation. Cambridge University Press.

[43] Leitgeb, H. (2009). On Formal and Informal Provability. In: Bueno, O. & Linnebo, Ø. (eds.) New Waves in Philosophy of Mathematics. Palgrave Macmillan: pages 263–299.

[44] Linnebo, Ø. (2018). Thin Objects. An Abstractionist Account. Oxford University Press.

[45] Linnebo, Ø. & Shapiro S. (2019). Actual and Potential Infinity. Noûs 53(1): 160–191.

[46] Ludlow, P. (2014). Living Words. Oxford University Press.

[47] Maddy, P. (2007). Second Philosophy. A Naturalistic Method. Oxford University Press.

[48] Makovec, D. & Shapiro, S. (eds.) (2019). Friedrich Waismann. The Open Texture of Analytic Philosophy. Springer.

[49] Mendelson, E. (1990). Second Thoughts about Church's Thesis and Mathematical Proofs. The Journal of Philosophy 87(5): pages 225-233.

[50] Nietzsche, F. (1901/1968). The Will to Power. Random House 1968.

[51] Papayannopoulos, P. (2023). On Algorithms, Effective Procedures, and Their Definitions. Philosophia Mathematica (III) 31(3): 291–329.

[52] Pour-El, M.B. (1999). The structure of computability in analysis and physical theory: An extension of Church's thesis. In: Griffor, E.R. (ed.). Handbook of Computability Theory. Elsevier: 449–472.

[53] Plunkett, D. & Cappelen, H. (2020). A Guided Tour Of Conceptual Engineering and Conceptual Ethics. In: Cappelen, H., Plunkett, D. & Burgess, A. (eds.), Conceptual Engineering and Conceptual Ethics. Oxford University Press (2020): 1–26.

[54] Putnam, H. (1980). Models and Reality. Journal of Symbolic Logic 45(3): 464–482.

[55] Quine, W.V.O. (1970). Philosophy of Logic. Harvard University Press.

[56] Quine, W.V.O. (1980). From a Logical Point of View. Harvard University Press.

[57] Quine, W.V.O. (1986). Philosophy of Logic. Harvard University Press.

[58] Quinon, P. and Zdanowski, K. (2007). Intended Model of Arithmetic. Argument from Tennenbaum's Theorem. In: S.B. Cooper, B. Loewe and A. Sorbi (eds.), Computation and Logic in the Real World, Computability in Europe.

[59] Quinon, P. (2018). Taxonomy of Deviant Encodings. In: Manea, F., Miller, R. & Nowotka, D. (eds.) Sailing Routes in the World of Computation. CiE 2018. Lecture Notes in Computer Science 10936 Springer: 338–348.

[60] Quinon, P. (2021). Can Church's thesis be viewed as a Carnapian explication? Synthese 198 (Suppl 5): 1047–1074.

[61] Quinon, P. (2020). Implicit and Explicit Examples of the Phenomenon of Deviant Encodings. Studies in Logic, Grammar and Rhetoric, 63(1): 53-67.

[62] Rescrola, M. (2007). Church's thesis and the conceptual analysis of computability. Notre Dame Journal of Formal Logic 48 (2): 253–280.

[63] Rav, Y. (1999). Why Do We Prove Theorems? Philosophia Mathematica (III) 7: 5–41.

[64] Saul, J. (2006). Gender and race. Aristotelian Society Supplementary Volume 80(1): 119–43.

[65] Shagrir, O. (2002). Effective Computation by Humans and Machines. Minds and Machines 12: 221-–240.

[66] Shapiro, S. (1982). Acceptable Notation. Notre Dame Journal of Formal Logic 23(1): 14—20.

[67] Shapiro, S. (2006). Computability, Proof, and Open-Texture. In: Olszewski A., Woleński J. & Janusz R. (eds.) (2006). Church's Thesis After 70 Years. Ontos-Verlag: 420–455.

[68] Shapiro, S. (2013). The Open Texture of computability. In B. J. Copeland, C. J. Posy, and O. Shagrir (Eds.), Computability: Turing, Gödel, Church, and Beyond, pp. 153–181. The MIT Press.

[69] Scharp, K. (2013). Replacing Truth. Oxford University Press.

[70] Sieg, W. (1997). Step by recursive step: Church'sanalysis of effective calculability. The Bulletin of Symbolic Logic, 3(2): 154—180.

[71] Soare, R. (1996). Computability and recursion. Bulletin of Symbolic Logic, 2: 284—321.

[72] Spelke, E. (2000). Core knowledge. American Psychologist. 55: 1233–1243.

[73] Strawson, P.F. (1959). Individuals. An Essay in Descriptive Metaphysics. Routledge.

[74] Svozil, K. (2006). Physics and Metaphysics Look at Computation. In: Olszewski A., Woleński J. & Janusz R. (eds.) (2006). Church's Thesis After 70 Years. Ontos-Verlag: 491–517.

[75] Tanswell, F.S. (2018). Conceptual engineering for mathematical concepts. Inquiry 61(8): 881–913.

[76] Tanswell, F.S. (2024). Mathematical rigour and informal proof. Cambridge University Press, 2024.

[77] Trakhtenbrot, B.A. (1995). Comparing the Church and Turing approaches: two prophetical messages. In: Herken R. (ed.) The Universal Turing Machine. A Half-Century Survey. Springer-Verlag: 557–582.

[78] Turing, A. (1936). On Computable Numbers, with an Application to the Entscheidungsproblem. Proceedings of the London Mathematical Society 42: 230–265.

[79] Vecht, J.J., (2023). Open texture clarified. Inquiry 66(6): 1120-1140.

[80] Waismann, F. (1968). Verifiability. In: Flew A. (ed.) Logic and Language. Blackwell.

[81] Wang, H. (1974). From Mathematics to Philosophy. Routledge and Kegan Paul.

[82] Williamson, T. (2007). The Philosophy of Philosophy. Blackwell.

[83] Wittgenstein, L. (1922/1961). Tractatus Logico-Philosophicus. Pears D.F. and McGuinness B. F. (trans.). Humanities Press 1961.