On the Choice of Gödel Numberings

Léon Probst

Abstract

The choice of Gödel numberings is a source of indeterminacy in metamathematics. For instance, Gödel's second incompleteness theorem fails under certain Gödel numberings. Nevertheless, it is widely believed that our standard metamathematical theorems hold for all *reasonable* Gödel numberings. This article investigates what makes a Gödel numbering reasonable and develops a general framework for addressing philosophical questions about Gödel numberings and, more generally, the arithmetisation of syntax.

Keywords: Arithmetisation, Gödel Numbering, Indeterminacy in Metamathematics, Syntax

Since there is really no reason to distinguish between a formal object and its Gödel number, we shall "identify" the two. (We do not really care exactly what the formulas, proofs etc. of a theory are; the only thing that matters is how they are related to one another.)

— Lindström [49, pp. 9–10]

[This article is forthcoming in *The Review of Symbolic Logic*. This is the penultimate version; the published version may be subject to minor changes. Please cite the published version.]

§1. Introduction. In his 1931 article, Gödel introduced a method for reasoning about the syntax of formal languages within arithmetic. This method, known as the *arithmetisation of syntax*, has become standard practice and is widely used to prove theorems about formal languages. It relies primarily on *Gödel numberings*, which assign natural numbers to syntactic expressions of formal languages, thereby allowing syntax to be represented by numbers.[1]

It is this representational role that allows certain arithmetical results to be interpreted as being about syntax. For instance, in Gödel's second incompleteness theorem (G2), the consistency of a theory T is expressed by an arithmetical sentence $\mathsf{con}(T)$. By itself, such an arithmetical sentence does not express the consistency of T. It does so only through the representation of syntax provided by the chosen Gödel numbering (hereafter simply 'numbering').

It is clear that not every numbering allows us to interpret a proven arithmetical statement as being about syntax. Indeed, not all of the uncountably many numberings represent syntax in an equally reasonable way. For a first gloss, let us call a numbering *reasonable* iff it provides an adequate representation of syntax and its properties. This is deliberately vague, but we do have strong intuitions about specific examples.

Consider, for instance, a numbering that assigns the even numbers to T-provable sentences of a given theory T, and the odd numbers to all other syntactic expressions. With respect to this numbering, the set of T-provable sentences becomes decidable, making it fairly easy to refute

[1] Gödel [25, p. 355], in his 1934 lecture, does not talk of numberings but rather of "representation[s] of the system by a system of positive integers". Church [13, p. 349], in 1936, talks of 'Gödel representation'. The terms 'Gödel numbering' and 'Gödel number' appear in reviews and papers by Rosser [68] in 1936 and later in Bernays [3] and Kleene [44].

versions of G2 for T (see [27, §2.3]). Such a refutation can hardly be regarded as showing that T proves its own consistency. Rather, the numbering fails to provide a genuine representation of syntax, and thus should not count as reasonable.

It follows that whether our standard metamathematical theorems hold depends on the particular numbering we choose. In other words, the choice of numberings is a source of indeterminacy (also sometimes called intensionality) in metamathematics.[2] As Visser [76, pp. 544–545] remarks, such an indeterminacy calls for two distinct kinds of responses.

On the one hand, there are the mathematical questions concerning the invariance of our theorems. Under what conditions do metamathematical theorems remain invariant with respect to the choice of numberings? What is the broadest class of numberings for which our results generalise? These questions have recently been studied (see, e.g., [27, 29, 32, 40]).

On the other hand, there are the more philosophical questions concerning the interpretation of our metamathematical theorems. As Visser [76, p. 544] observes, proving a theorem for a specific numbering "does give the theorem a definite meaning", but fails to capture "the generality of the insight" we have. Indeed, we strongly believe that our standard metamathematical theorems should hold for all reasonable choices, and thus for all reasonable numberings. Only then would we be justified in interpreting our theorems according to their standard and most general interpretation, namely, as results about syntax. This raises the central philosophical questions of this paper: What exactly makes a numbering reasonable for metamathematical investigations? What is supposed to be represented by a numbering? Which properties should be used to define reasonable numberings, and how can they be motivated?[3]

The literature on numberings has so far focused almost exclusively on the mathematical questions, and numerous metamathematical results concerning numberings have recently been proven.[4] For instance, Grabmayr [27, 29] proved the invariance of a version of G2 for all effective numberings. While the requirement of effectivity is standard (as in, e.g., [31, 52]), it is not straightforward to make it precise (see [27, 40, 50, 70, 71] for some options). Other metamathematical results have been obtained by imposing conditions such as monotonicity, regularity, domination, or stricter computability constraints on numberings (see, among others, [27, 29, 32, 37, 39, 40] and [32] for definitions of these properties). In these cases, such properties have been introduced purely in order to prove theorems and with little independent motivation beyond their technical utility.

A general conceptual framework for evaluating numberings and, with it, a philosophical analysis of reasonable numberings, is still lacking. It is widely acknowledged that none of these standard requirements on numberings are sufficient to define the class of reasonable numberings. As Grabmayr [27, p. 70] notes, such constraints are "in general too weak to enforce an adequate arithmetisation of even basic syntactic operations" (see also [36, pp. 363–367]). For instance, these constraints are always tied to a specific presentation of the syntax, and therefore lack the desired generality. Moreover, these constraints are formulated independently of the base theory to which the reasonableness of a numbering is evaluated. Yet the complexity of the formulas arithmetising syntactic notions depends on both the numbering and the base theory (see [36,

[2] See [12, §5.2] and [36, Chap. 12] for some useful overviews of indeterminacy in metamathematics. The terms 'intensionality' and 'intensionally correct' were coined by Feferman in his seminal work [18] and are still widely used in the literature despite being somewhat of a misnomer, as Feferman himself recognised in his later works (see [57, p. 139]). Following [76, p. 544], I favour the more neutral and self-explanatory term 'reasonable' (as well as 'adequate'), and speak more generally of sources of indeterminacy.

[3] Similar questions arise, *mutatis mutandis*, for notations in the philosophy of computation and in ordinal analysis (see, respectively, [7, 30] and [2, 20] with references therein).

[4] A notable exception is Grabmayr [27, §3], who provides philosophical remarks on his notion of admissible numberings. His use of 'admissible' is roughly equivalent to, though perhaps weaker than, my use of 'reasonable'. I discuss several of his claims throughout this paper; in particular, in §5.1, I argue against his general approach to numberings.

p. 364]).

The aim of this article is, therefore, to develop such a framework for the philosophical analysis of numberings and their properties. To this end, I draw on the two standard approaches to indeterminacy in metamathematics relying, respectively, on the resemblance and conditions criteria (see [36, Chap. 12.2] and [37, §2.2]). These criteria have so far been applied only to the indeterminacy arising from formulas, such as provability predicates. The central insight of this article is to apply them to numberings and to show how they can provide a conceptual framework in which to answer philosophical questions about numberings. Within this framework, I defend several necessary conditions for reasonable numberings and provide an analysis of the interplay between reasonable numberings and base theories.

This article proceeds as follows. Given the early stage of this debate, I first introduce some basic definitions in §2 and present the main data for the study of numberings in §3. In §4, I briefly introduce and discuss the resemblance and conditions criteria, and suggest understanding them as complementary in the study of indeterminacy in metamathematics. To apply these criteria to numberings, I first ask what a numbering is supposed to represent in §5. Building on some technical work by Grabmayr [27, 29], I criticise a first standard approach and defend a second algebraic approach. Finally, in §6, I apply the previous analyses to the notion of reasonable numberings. In particular, using resemblance conditions, I identify and defend several necessary properties for a numbering to count as reasonable. I then discuss the conditions criterion for numberings and how the choice of a numbering relates to the choice of the base theory. I briefly conclude in §7 and provide some technical details in Appendix A.

§2. Technical Preliminaries. This section introduces some useful vocabulary and basic definitions for this debate. Some familiarity with the metamathematics of first-order theories of arithmetic is presupposed (see [34] for the standard textbook). Let T be any consistent theory extending Robinson's Arithmetic Q, e.g., $\mathsf{I}\Delta_0$, $\mathsf{I}\Sigma_1$ or PA.

Syntax. Let us work with the *basic language of arithmetic* $\mathcal{L}$ whose alphabet A consists of non-logical symbols $\{\mathsf{0}, \mathsf{s}, +, \times\}$ and of logical symbols $\{\mathsf{v}, ', \bot, =, \to, \forall\}$.[5] The sets of the expressions of $\mathcal{L}$ (i.e., the variables, terms and formulas) are defined inductively over a *background universe* by an *implementation*.

The most common background universe is a universe of strings [1, 6, 16, 24], but other options serve equally well, e.g., labelled trees [17, 74], sets [22, 50, 72], numbers [18, 34], 2-dimensional strings [36]. Before presenting the more general framework, let us define two common examples of implementations on some universe of strings. For the Kleene star A^*, i.e., the set of all finite strings from A, the usual prefix (or Polish) implementation pre on A^* defines the set of the pre-expressions as follows:

$$\begin{aligned} x &::= \mathsf{v} \mid {}'x \\ t &::= x \mid \mathsf{0} \mid \mathsf{s}\,t \mid +\,t\,t \mid \times\,t\,t \\ \varphi &::= \bot \mid =\,t\,t \mid \to\,\varphi\,\varphi \mid \forall x\,\varphi \end{aligned}$$

In this framework, the auxiliary symbols (such as commas or brackets) are part of the background universe. Thus, the standard infix implementation inf defines the set of the inf-expressions on

[5] The choice of the language of arithmetic and of these particular logical symbols is made primarily for simplicity. Our framework extends to arbitrary finite first-order languages. The restriction to a finite alphabet is necessary, e.g., to define the numbering gn_2 discussed below or for Mal'cev's theorem (see [29, Th. 4.20]). For this reason, variables are not taken to form an infinite set of distinct variable symbols; instead, they are generated from v and $'$, yielding, for example, $\mathsf{v}, {}'\mathsf{v}, {}''\mathsf{v}, \ldots$.

the Kleene star on $A \cup \{), (\}$ as follows:

$$\begin{aligned} x &::= \mathsf{v} \mid {'}x \\ t &::= x \mid \mathsf{0} \mid \mathsf{s}\, t \mid (t + t) \mid (t \times t) \\ \varphi &::= \bot \mid (t = t) \mid (\varphi \to \varphi) \mid \forall x\, \varphi \end{aligned}$$

More generally, given a background universe U, the sets of $\imath$-variables, $\imath$-terms, and $\imath$-formulas are defined inductively by the following *$\imath$-constructor functions*:

$$\begin{aligned} \mathsf{v}_\imath &\colon \mathrm{Var}_\imath^0 \to \mathrm{Var}_\imath & {'}_\imath &\colon \mathrm{Var}_\imath \to \mathrm{Var}_\imath \\ \mathsf{0}_\imath &\colon \mathrm{Ter}_\imath^0 \to \mathrm{Ter}_\imath & \mathsf{s}_\imath &\colon \mathrm{Ter}_\imath \to \mathrm{Ter}_\imath \\ +_\imath &\colon \mathrm{Ter}_\imath^2 \to \mathrm{Ter}_\imath & \times_\imath &\colon \mathrm{Ter}_\imath^2 \to \mathrm{Ter}_\imath \\ \bot_\imath &\colon \mathrm{Fml}_\imath^0 \to \mathrm{Fml}_\imath & =_\imath &\colon \mathrm{Ter}_\imath^2 \to \mathrm{Fml}_\imath \\ \to_\imath &\colon \mathrm{Fml}_\imath^2 \to \mathrm{Fml}_\imath & \forall_\imath &\colon \mathrm{Var}_\imath \times \mathrm{Fml}_\imath \to \mathrm{Fml}_\imath \end{aligned}$$

The various $\imath$-constructor functions are defined by operations on U. For instance, ${'}_{\mathsf{pre}}$ is defined by $x \mapsto {'}^\frown x$, where $^\frown$ denotes the operation of concatenation. The set of pre-variables is the smallest set closed under $\mathsf{v}_{\mathsf{pre}}$ and ${'}_{\mathsf{pre}}$ (see, e.g., [1, Chap. 1] on inductively defined sets). The set of $\imath$-expressions $\mathrm{Exp}_\imath$ on U is defined as $\mathrm{Var}_\imath \cup \mathrm{Ter}_\imath \cup \mathrm{Fml}_\imath \subseteq U$. For simplicity, we use an implementation $\imath$ to denote the collection of all $\imath$-constructor functions, and say that $\mathrm{Exp}_\imath$ is defined by $\imath$. More details and some motivation are given in §5.2 and the Appendix A.

In this article, I consider only implementations $\imath$ that satisfy the following two conditions. First, each variable is a term and no term is a formula; thus, $\mathrm{Var}_\imath \subseteq \mathrm{Ter}_\imath$ and $\mathrm{Ter}_\imath \cap \mathrm{Fml}_\imath = \varnothing$. Second, the basic syntactic relations (e.g., *being free in*, *being a sentence*, etc.) are defined by recursion. This requires recursion theorems for each of the sets $\mathrm{Var}_\imath, \mathrm{Ter}_\imath$, and $\mathrm{Fml}_\imath$. Since such theorems hold in general for every inductively and freely generated set, it suffices for our syntactic sets to be *inductively* and *freely generated* (see, e.g., [1, pp. 16–17] or [42, §0.24]). This amounts to considering only those implementations $\imath$ for which the $\imath$-constructor functions are injective and their ranges pairwise disjoint.[6]

Gödel Numberings. Over the set of non-negative integers ω, we define numberings as follows:

Definition 1 (Numbering)**.** Given a set B, a *numbering α of B* is an injective function $\alpha\colon B \to \omega$. The inverse of α is defined in the usual way, $\alpha^{-1}\colon \alpha(B) \to B$.

We say that $\alpha(b)$ is the α-code of b, and that the range $\alpha(B)$ of α is the set of α-codes. For an implementation $\imath$, a *Gödel numbering on $\imath$* is a numbering of B where each $\imath$-expression has an α-code, i.e., $\mathrm{Exp}_\imath \subseteq B$. When the context is clear, I simply use 'numbering' as shorthand for 'Gödel numbering on $\imath$'.

A *tracking function* is used to track, at the level of numbers, a given function on an encoded set.

Definition 2 (α-Tracking Functions)**.** For a numbering α of B and any partial k-ary function $f\colon B^k \to B$, an *α-tracking function* of f is a partial k-ary function $f^\alpha\colon \omega^k \to \omega$ such that:

$$f^\alpha(n_0, \ldots, n_{k-1}) = \alpha(f(\alpha^{-1}(n_0), \ldots, \alpha^{-1}(n_{k-1})))$$

whenever $f(\alpha^{-1}(n_0), \ldots, \alpha^{-1}(n_{k-1}))$ is defined.

[6] In the literature, this second structural requirement is often spelt out in terms of the misnomer *unique readability theorem* as, e.g., in [16]. Note that, in our framework, this requirement rules out certain ways to define the syntax, e.g., de Bruijn indices, Peirce-Quine-Bourbaki's notation, or the negation normal form notation (see [29] for more examples). I discuss these limitations in §5.2.

Let us consider an example of a numbering α on pre. An α-tracking function of the pre-constructor function $\rightarrow_{\mathsf{pre}}$ can be defined by $x, y \mapsto \alpha(\rightarrow_{\mathsf{pre}}(\alpha^{-1}(x), \alpha^{-1}(y)))$ and then the following diagram commutes:

$$\begin{array}{ccc} \mathrm{Fml}_{\mathsf{pre}} \times \mathrm{Fml}_{\mathsf{pre}} & \xrightarrow{\quad\rightarrow_{\mathsf{pre}}\quad} & \mathrm{Fml}_{\mathsf{pre}} \\ \downarrow{\scriptstyle \alpha\times\alpha} & & \downarrow{\scriptstyle \alpha} \\ \alpha(\mathrm{Fml}_{\mathsf{pre}}) \times \alpha(\mathrm{Fml}_{\mathsf{pre}}) & \xrightarrow{\quad\rightarrow^{\alpha}_{\mathsf{pre}}\quad} & \alpha(\mathrm{Fml}_{\mathsf{pre}}) \end{array}$$

where $\alpha\times\alpha\,(x,y) := \langle\alpha(x), \alpha(y)\rangle$. Thus, we have $\rightarrow^{\alpha}_{\mathsf{pre}}(\alpha(\varphi), \alpha(\psi)) = \alpha(\rightarrow \varphi\,\psi)$, which tracks the constructor function $\rightarrow_{\mathsf{pre}}$ at the level of the α-codes.

Finally, to compare numberings, we use a notion of equivalence between numberings. For our purposes, it suffices to define equivalence relative to the class of recursive functions $\mathbb{REC}$.[7]

Definition 3 ($\mathbb{REC}$-Equivalence)**.** For two numberings α and β of B, α is $\mathbb{REC}$-*reducible* to β iff there exists a recursive function f such that for all $n \in \alpha(B)$, $\alpha^{-1}(n) = (\beta^{-1} \circ f)(n)$. α is $\mathbb{REC}$-*equivalent* to β iff α is $\mathbb{REC}$-reducible to β and β is $\mathbb{REC}$-reducible to α.

Roughly, two numberings α and β of B are equivalent if, for each $b \in B$, we can translate (in a computable way) the α-code of b into the β-code of b and *vice versa*.

§3. Some Types of Numberings. This section surveys several paradigmatic types of numberings and discusses whether they are intuitively reasonable. These numberings and their definitions constitute the main data for philosophy of this article.[8]

For simplicity, all numberings introduced below are defined for the implementation pre with A^* as background universe, though they could also be adapted (with more or less ease) to other background universes and implementations. Proofs that any such numbering is indeed a numbering on pre are omitted.

Standard Numberings. The most common numberings in the literature are the *string numberings*. These numberings assign a code to each string of A^* thanks to some basic code and some coding of finite sequences of positive integers. A *basic code* of the alphabet A is a function $\#\colon A \to \omega\backslash\{0\}$. For definiteness, let us fix $\#(\mathsf{v}) = 1, \#(') = 2, \ldots, \#(\forall) = 10$, and for $s_i \in A$ and $n > 0$, let $s \in A^*$ be such that $s \equiv s_0{}^\frown s_1{}^\frown \cdots {}^\frown s_{n-1}$, where $\equiv$ denotes the string identity. Different basic codes and sequential codings yield different string numberings (see [11]). Below, two most common numberings are presented.

The first numbering gn_1 was introduced by Gödel [24] and is still used in many textbooks (e.g., [50, 67, 70]). Relying on the fundamental theorem of arithmetic, it is defined as follows:

$$\mathsf{gn}_1(s) = \pi_0^{\#(s_0)} \cdot \pi_1^{\#(s_1)} \cdot \ldots \cdot \pi_{n-1}^{\#(s_{n-1})} = \prod_{i=0}^{n-1} \pi_i^{\#(s_i)}$$

[7] For $B \subseteq \omega$, a k-ary function $f\colon B^k \to \omega$ is said to be recursive if $\mathrm{dom}(f)$ is decidable and there is a total recursive function $f'\colon \omega^k \to \omega$ such that $f' \upharpoonright \mathrm{dom}(f) = f$. It is well known that any theory $T \supseteq \mathsf{Q}$ binumerates the graph of each recursive function with a Σ_1-formula. A k-ary function f is said to be *provably recursive in* T iff there is a Σ_1-binumeration φ of f in T and $T \vdash \forall x_0, \ldots, x_{k-1}\, \exists! y\, \varphi(x_0, \ldots, x_{k-1}, y)$ (see [34, Chap. I]).

[8] Clearly, our list is not exhaustive. Among the uncountably many numberings, there are of course other types (for more uncommon examples, see, e.g., [51, 64, 79]). I take these numberings to be less relevant to our present discussion and leave them to future work. Note also that some of the standard numberings already presuppose a way to encode finite sequences of positive integers (typically with Gödel's β-function). Such encodings, although usually carried out not with single numbers but, e.g., with pairs of numbers or matrices, can already be seen as encoding some universe of strings (see [4, p. 506]) and are commonly used in the study of weak arithmetics (see, e.g., [4, 8, 53, 54, 79]).

where π_i denotes the $(i+1)$th prime number. A common variant is to encode n, the length of s, as the first term of the encoded sequence (see, e.g., [59]).

Introduced by Smullyan [71], the second numbering gn_2 is commonly used in the study of weak arithmetics (e.g., [79, 81]). For some (often prime) number $b \geqslant \max(\#(A))$, it employs the base-b representation of numbers:

$$\mathsf{gn}_2(s) = \#(s_0) \cdot b^0 + \#(s_1) \cdot b^1 + \ldots + \#(s_{n-1}) \cdot b^{n-1} = \sum_{i=0}^{n-1} \#(s_i) \cdot b^i$$

The length-first lexicographic (or shortlex) numbering is a special case of this numbering, in our case with $b = 10$.

Another common way to define numberings in the literature does not focus on strings, but on the syntactic expressions and their generation. Introduced by Hilbert and Bernays [41], such *compositional numberings* are often preferred when the syntax is not defined on a universe of strings (e.g., [1, 18, 45, 74]).

The third numbering gn_3 is defined recursively on $\mathrm{Var}_{\mathsf{pre}}, \mathrm{Ter}_{\mathsf{pre}}$, and $\mathrm{Fml}_{\mathsf{pre}}$ as follows: For some sequential coding $\langle\cdot\rangle$ and $x \in \mathrm{Var}_{\mathsf{pre}}$,

$$\mathsf{gn}_3(\mathsf{v}) = \langle \#(\mathsf{v}) \rangle \text{ with } \mathsf{gn}_3(\mathsf{'}x) = \langle \#(\mathsf{'}), \mathsf{gn}_3(x) \rangle,$$

for $s, t \in \mathrm{Ter}_{\mathsf{pre}}$,

$$\mathsf{gn}_3(\mathsf{0}) = \langle \#(\mathsf{0}) \rangle \text{ with } \mathsf{gn}_3(\mathsf{s}\, t) = \langle \#(\mathsf{s}), \mathsf{gn}_3(t) \rangle,$$

$$\mathsf{gn}_3(+\, s\, t) = \langle \#(+), \mathsf{gn}_3(s), \mathsf{gn}_3(t) \rangle, \text{ and } \mathsf{gn}_3(\times\, s\, t) = \langle \#(\times), \mathsf{gn}_3(s), \mathsf{gn}_3(t) \rangle$$

and similarly for the formulas. Other definitions not using a sequential coding are also available, as it suffices, for all clauses, to take injective functions whose ranges are pairwise disjoint (see, e.g., [41, pp. 216–218]).[9]

It is clear that the standard numberings $\mathsf{gn}_{1,2,3}$, as well as most numberings found in the literature, are intuitively reasonable. Indeed, it is difficult to see what could go wrong with any of them. However, the reasonableness of a numbering also depends on the choice of base theory. This dependence becomes particularly apparent in weak theories of arithmetic, although it is not limited to such theories.

Consider, for instance, the set of gn_1-codes of the terms. Such a set can be arithmetised by the standard Σ_1-formula $\mathsf{term}(x)$ that mimics a definition of the terms, i.e., x is the gn_1-code of a term iff "there exists a construction sequence for terms which ends with x". Any theory T extending Q binumerates such a set with $\mathsf{term}(x)$, however not all of them prove the basic properties of the terms with $\mathsf{term}(x)$ (see, e.g., [34, p. 312]). For example, $\mathsf{I}\Sigma_1$ does prove them with $\mathsf{term}(x)$ using its induction on Σ_1-formulas. But what about weaker theories without induction on Σ_1-formulas?

The answer depends on whether $\mathsf{term}(x)$ can be proved to be equivalent to a formula of lower complexity, which requires bounding the initial existential quantifier. Under gn_1, the code of such a construction sequence is bounded by an exponential in x. Thus, the theory EA, whose language has a symbol of exponentiation, can prove basic properties of the terms with existentially bounded $\mathsf{term}(x)$, making it $\Delta_0(\mathsf{exp})$. However, the same bounding argument is unavailable in $\mathsf{I}\Delta_0$ (or even $\mathsf{I}\Delta_0 + \Omega_n$), which is formulated in the basic language of arithmetic and does not prove the

[9] It is also common to encode the set of variables using their indices, e.g., with $\langle \#(\mathsf{v}), n \rangle$ for the nth variable or equivalent ways (see, e.g., [74]). However, these numberings are usually defined on a syntax which takes the variables as infinitely many distinct symbols together with some ordering on them, and are therefore not compositional with respect to the variables.

totality of exponentiation. (If the set of gn_1-codes of terms does not belong to the linear time hierarchy, then $\mathsf{I}\Delta_0$ does not Δ_0-binumerate it, see [14, Chap. III.4].)

This is a defect of the combination of the choices of numbering and base theory, not of $\mathsf{I}\Delta_0$ itself. Indeed, by choosing a more efficient numbering, $\mathsf{I}\Delta_0$ can prove the basic properties of the terms [34, Chap. V.3]. (See also [81] on gn_2 and its interaction with $\mathsf{I}\Delta_0 + \Omega_n$.) I return to these issues below. For now, let us turn to more problematic and philosophically significant cases of numberings.

Characteristic Numberings. Another type of numbering is the *characteristic numberings* which encode not only their domain but also additional information about it — typically, the characteristic function of some subset of their domain. Such numberings are similar in many respects to deviant notations in the philosophy of computation.[10]

The fourth numbering gn_4 is defined following Grabmayr [27, p. 55]: For $X \subseteq A^*$ such that both X and $A^*\backslash X$ are infinite, let $(\eta_i)_{i\in\omega}$ and $(\theta_i)_{i\in\omega}$ be, respectively, enumerations without repetitions of X and $A^*\backslash X$. Then, let

$$\mathsf{gn}_4(s) = \begin{cases} 2i & \text{if } s \equiv \eta_i \in X \\ 2i+1 & \text{if } s \equiv \theta_i \in A^*\backslash X \end{cases}$$

Hence, whether or not s is an element of X depends on whether the gn_4-code of s is even. This definition also generalises, with some additional coding, to any n-ary relation on A^*.

It is then straightforward to construct counterexamples to many metamathematical theorems using such characteristic numberings. To extend our sketch in §1, let X be the set of T-provable sentences. By the standard Δ_0-binumeration $\mathsf{even}(x)$ of the set of even numbers, T binumerates X and therefore its own provability. It is easy to see that $\mathsf{even}(x)$ fulfils all Löb's conditions with respect to such a gn_4, but $T \vdash \neg\mathsf{even}(\overline{\mathsf{gn}_4(\bot)})$. Indeed, if T is consistent, then $\bot$ is not provable in T and so, by definition, $\mathsf{gn}_4(\bot)$ is odd. Thus, we obtain a version of $\mathsf{G2}$ with Löb's conditions that fails for T (note that, in this case, the diagonal lemma has to fail [27, p. 55]). Clearly, gn_4 is not a reasonable choice (at least with T).

Nonetheless, not all characteristic numberings appear unreasonable. For instance, allowing X to be the whole (or empty) domain with $(\eta_i)_{i\in\omega}$ (or $(\theta_i)_{i\in\omega}$) given by gn_1, the resulting numberings are essentially just gn_1. More interesting and difficult cases come with non-trivial choices of X. Let X be the set of terms $\mathrm{Ter}_{\mathsf{pre}}$ and consider its characteristic numbering (with some effective enumerations). Working in PA, we then have a straightforward Δ_0-binumeration of the terms with $\mathsf{even}(x)$. However, in PA, this binumeration does not seem to have any substantive advantage over the standard one with gn_1. A similar phenomenon arises if we take X to be the set of $\mathsf{I}\Sigma_1$-provable sentences and work in PA. Trivially, with such a numbering, PA proves a consistency statement for $\mathsf{I}\Sigma_1$, but also with gn_1 (though with much more work and not with the same statement). Still, these last two gn_4 seem mainly to be mathematical tricks, which happen to have no (obvious) dire consequences.

Self-Referential Numberings. The last type of numberings considered here consists of *self-referential numberings*. Both Kripke [46, pp. 77–78] and Feferman [19, p. 80] suggested defining numberings in such a way as to obtain a Gödel sentence directly from the numbering. This idea was later generalised to construct numberings with this self-referential property for all formulas with one free variable (see [38, 39, 47, 75], and [32] for an extensive study).

[10] Given a set $B \subseteq \omega$, a deviant notation for B is a representation of positive integers on a Turing-machine tape that allows the characteristic function of B to be encoded within the representation, so that B is decidable. For an overview of this debate, see [7, 30] and references therein.

The fifth numbering gn_5 is defined following Visser [75, pp. 159–160]: Let $(\theta_i)_{i\in\omega}$ be an enumeration of the Kleene star on $A \cup \{\mathsf{c}\}$, where c is a fresh constant. We define $(\delta_i)_{i\in\omega}$ as an enumeration of A^* (without c) from $(\theta_i)_{i\in\omega}$ by substituting, in each θ_i, every occurrence of c for $\overline{i}$, the standard numeral of i. Thus, $\delta_i \equiv \theta_i[\mathsf{c} := \overline{i}]$ and gn_5 is defined as follows:[11]

$$\mathsf{gn}_5(s) = \min\{n \mid \delta_n \equiv s\}$$

Using such self-referential numberings, the diagonal lemma becomes a triviality. Indeed, for any formula $\varphi(x)$, there is an i such that:

$$\mathsf{gn}_5(\varphi(\overline{i})) = \min\{n \mid \varphi(\overline{i}) \equiv \delta_n \equiv \theta_n[\mathsf{c} := \overline{i}]\} = i$$

Trivially, $T \vdash \overline{\mathsf{gn}_5(\varphi(\overline{i}))} = \overline{i}$ (the strong diagonal lemma), from which the diagonal lemma follows.

As with the preceding examples, such self-referential numberings appear to be a more elaborate version of the characteristic numbering, where the diagonalisation takes place in the construction of the numbering rather than in the theory. Again, this mathematical trick seems to have no dire consequences.[12]

Should we therefore regard these as reasonable numberings? It is unclear to me whether such self-referential numberings should be considered reasonable. As pointed out by a reviewer, self-referential numberings might play an important role in certain contexts with weak theories of arithmetic (see, e.g., the remarks in [48, §6.2]). Although this has not yet been carefully studied, it may count in favour of their reasonableness. On the other hand, these numberings directly affect the consistency of some basic untyped truth theories (see Heck [39, §3] and Schindler [69, §2.2]). In particular, such theories become inconsistent when self-referential numberings are used, while remaining consistent with standard numberings, such as gn_1 (see also [32, §9]). This may seem an unwelcome consequence for the class of reasonable numberings, and therefore favour understanding them as unreasonable.

§4. Indeterminacy Criteria. As shown in the previous section, the adequacy of an arithmetisation of syntax depends, among others, on the choice of numberings. While we have some intuitions about which numberings are reasonable and why, it appears that articulating them is a rather difficult task. A satisfactory definition of reasonable numberings must not only be mathematically precise but also philosophically well-motivated. For instance, defining the reasonable numberings with those for which G2 holds is not satisfactory; one should not disregard a numbering simply because a certain metamathematical result fails (or holds). A better approach, which I follow, is to first identify and argue for the intrinsic mathematical properties that reasonable numberings must possess (or lack), and only then derive invariance results about

[11] This definition assumes that $(\theta_i)_{i\in\omega}$ is monotonic with respect to the substring relation, as, e.g., with the shortlex ordering starting with c. The enumeration $(\delta_i)_{i\in\omega}$ then has some redundancies, as for each θ_i that contains c, $\theta_i[\mathsf{c} := \overline{i}]$ is enumerated twice. Thus, taking the minimum assures us of the functionality of gn_5, and that s gets assigned n, the number of the δ_n resulting from a non-trivial substitution of $\overline{n}$ for c. More efficient definitions are also possible [38, §12], as well as generalisations of self-referential numberings which allow us to trivially prove more general versions of the diagonal lemma.

[12] That a mere choice of a numbering can make a proof substantially different could be an indication that there is something odd about this choice. This observation is, of course, imprecise as it stands. Is the proof of the diagonal lemma using gn_1 (substantially) different from the one using gn_3? Only in the former, the diagonal function is (usually) defined without the help of the substitution function (see, e.g., [70, p. 146]). Similar comments apply to our previous numberings gn_4, which make some proofs about the various sets X ridiculously short and trivial. These remarks touch upon more general issues about the interplay between proofs and theorems in metamathematics, see the short discussion in §4.3 and fn. 16.

this class.[13] Furthermore, as presented in §3, the interplay between classes of numberings and the choice of the base theory must be made explicit.

To this end, I analyse the two standard and general criteria used in the literature to deal with reasonable representation. These are the *resemblance* and the *conditions* (also called *meaning-postulate*) criteria (see [36, Chap. 12.2] and [37, §2.2]). Such criteria usually apply to arithmetical formulas representing relations (e.g., provability or truth). Nonetheless, I suggest that they are also applicable to numberings. In particular, the former lets us specify what a reasonable numbering is, while the latter helps explain how reasonableness interacts with the choice of base theory. For the case of numberings, I discuss these issues in §6. Before doing so, I first ask, in §5, what exactly a numbering is supposed to resemble.

In this section, I outline the resemblance and conditions criteria. Then, I propose an understanding of these two criteria as complementary and relate them to the source of indeterminacy due to the choice of the base theory.

§4.1. Resemblance Criterion. The resemblance criterion is a natural and intuitive criterion that is widely, though often tacitly, used in metamathematics. Consider a relation and its usual description in a metalanguage. To arithmetise this relation, i.e., define an arithmetical formula that represents it, a natural requirement is to ask for this formula to *resemble* its description.

Resemblance Criterion. For a relation R and a description of R, a formula φ is a *reasonable representation* for R iff φ resembles the description of R, in the relevant way.

This presentation assumes that the relation R comes with a preferred description. Typically, the standard PA-provability predicate $\mathsf{prov}_{\mathsf{PA}}(x) \coloneqq \exists y\, \mathsf{prf}_{\mathsf{PA}}(y, x)$ resembles the standard description of the notion of PA-provability. By contrast, the Rosser PA-provability predicate $\mathsf{rprov}_{\mathsf{PA}}(x) \coloneqq \exists y\, (\mathsf{prf}_{\mathsf{PA}}(y, x) \wedge \forall v \leqslant y\, \neg \mathsf{prf}_{\mathsf{PA}}(v, \mathsf{neg}(x)))$ does not, as it contains a second conjunct which does not resemble any part of the description of PA-provability.[14]

As in the previous example, the focus is usually on *structural* resemblance between a formula and a description (see, e.g., [56, §3] or [36, Chap. 12.2]). However, many such descriptions (often implicitly) include computable properties that cannot be captured simply by structural resemblance. This suggests that the resemblance criterion should also be concerned with *computable* resemblance. A natural way to articulate this is, for example, by invoking the Church-Turing thesis and the arithmetical hierarchy. To illustrate, let us consider two examples:

First, the set of terms is usually defined, at least implicitly, in such a way as to be, at least, computable (see, e.g., [34, pp. 21–23] where this requirement is explicit). Take then a Σ_2-formula φ that structurally resembles the description of the terms (e.g., φ is the usual formula but with one of its relevant universal quantifier left unbounded). There is a clear sense that such a formula fails to be a reasonable representation of the terms, as it does not have the same complexity as the set of terms. On this analysis, by the Church-Turing thesis and the arithmetical hierarchy, φ should be a Δ_1-formula.

Second, we often restrict our interest to computably enumerable (c.e.) theories T, meaning that there is a c.e. set of axioms for T. It is then common to require the formula representing

[13] Such reasoning follows the strategy that, in the context of ordinal analysis, Feferman [20, p. 10] advocates: "Though natural well-orderings arise naturally in the proof theory of formal systems, the feeling is that what distinguishes such orderings are certain intrinsic mathematical properties that are independent of their possible use in proof-theoretical work. [...] What we shall do in the following is describe a few systems of natural ordinal representation that have been of significance in proof-theoretical work and try to say in general mathematical terms what is special about them." See also [27, p. 60] for a similar position.

[14] Nonetheless, $\mathsf{rprov}_{\mathsf{PA}}(x)$ is also a numeration of PA-provability in PA (and much weaker). For this reason, it is often said that Rosser's PA-provability predicate is extensionally adequate but fails to be intensionally correct. See [18, pp. 36–37] and fn. 2.

the axiom set of T to be of the same complexity, i.e., to be a Σ_1-formula or even Δ_1^b if one works with p-time decidable theories (see, e.g., [56, §3.2] and [77, p. 68]). For instance, Feferman [18, p. 68]'s representation of the axiom set of PA fails to resemble the usual description of the axioms of PA (at least) on the grounds that this Π_1-formula is not of the right complexity.

Although the previous examples make the resemblance criterion appear intuitive and a good heuristic, this criterion is vague, and it is not obvious how to make it mathematically precise. Indeed, it is far from clear what 'resemblance' means and whether it can be formally captured, especially in the case of structural resemblance. See [37, p. 677] for further difficulties.

§4.2. Conditions Criterion. To address the many difficulties posed by the resemblance criterion, logicians have adopted an alternative criterion. Instead of requiring resemblance, they focus on identifying key conditions that any representation of a relation should fulfil. These conditions are to be proved within a theory T for a class of formulas, which are then taken to represent the relation.

Conditions Criterion For a theory T and a relation R, a formula φ is a *reasonable representation* of R iff T proves the conditions for R with φ.

This criterion, unlike the resemblance criterion, is mathematically well-defined once its conditions are specified. And it has been successfully applied throughout metamathematics. Löb's conditions for the notion of provability are the paradigmatic examples of such a criterion, and are commonly used in the "modal" proof of $\mathsf{G2}$. Other examples of conditions criteria are numerous, e.g., with partial truth [37, pp. 675–676], terms [34, Chap. V.3(g)], or theorems [9, p. 116].

The conditions criterion is often glossed and motivated in anthropomorphic terms. When working with representations, we need to ensure that they are reasonable. This usually requires proving certain results about the representation. For instance, with any implementation of the terms, one should prove that some unique readability theorem holds. In metamathematics, the situation is similar, except that such results must be established in the theory. For a theory T to "know" or "understand" a relation amounts for T to prove the conditions criterion for that relation (see, e.g., [15, p. 45]).[15]

It is extremely difficult to find conditions such that all the formulas that satisfy them in some theory are intuitively reasonable. The main issue with the conditions criterion is that, while it can provide necessary conditions for a formula to be a reasonable representation, it often fails to provide sufficient conditions. For example, any theory T proves Löb's conditions for the trivial $x = x$, yet it obviously does not represent provability.

§4.3. Complementary Criteria. Despite these difficulties, both of these criteria are widely accepted and used throughout the literature. A proper defence and detailed exposition of them would take us too far afield. My aim here is instead to suggest that, *contra* their usual presentation, these two criteria should not be regarded as opposed. On the contrary, I shall argue that their roles are distinct and that both are required for a complete analysis of indeterminacy in metamathematics.

The conditions criterion is given with respect to a theory T. An obvious issue is that different theories might disagree on which conditions are provable. Typically, PA proves Löb's conditions with $\mathsf{prov}_{\mathsf{PA}}$, but Q fails to do so. What should we then conclude from the representation of

[15] See also Niebergall [55, p. 129] for a similar position. Interesting questions concerning (relative) interpretation and the conditions criterion are raised with this position. Does T have access to the representation through an interpretation of a stronger theory that proves the relevant conditions? For example, Q interprets S_2^1, which succeeds in proving Löb's conditions for Q (see [21]). Does Q then have access to the notion of Q-provability via its interpretation of S_2^1? On these questions, see also [23, Chap. 5] and fn. 22.

PA-provability by $\mathsf{prov}_{\mathsf{PA}}$? To answer this worry, we could choose a fixed base theory Base that will determine the reasonableness of any representation. This solution is, however, unattractive. Given some fixed base theory Base, this criterion can only distinguish formulas up to Base-provability, which might be too coarse-grained for our purposes. Take, for example, the standard conditions criterion for terms. Any two Σ_0-formulas satisfying them are equivalent already in $\mathsf{I}\Delta_0$ (in this case, some additional assumptions are at play, see [34, pp. 312–313]). However, it does not follow that all these formulas are equally good representations of the terms. For instance, the representation relying on "counting the brackets" of the terms satisfies these conditions, but one could still argue against its reasonableness. Furthermore, independently of the choice for Base, it is questionable whether a formal theory and its strength should play any role in determining the reasonableness of a representation. See [77, §§3.3–3.4] and [80, §2] for further arguments and a detailed discussion.

In contrast, the resemblance criterion, by lacking any reference to a theory, fails to capture the interplay between a representation and the choice of the base theory. For instance, $\mathsf{con}(\mathsf{PA})$, which is presumably a reasonable representation of the consistency of PA, is unprovable both in PA and Q. Yet these two results are usually interpreted differently. The former is taken as evidence that PA does not prove its own consistency. While the latter is not generally taken as showing that Q fails to prove the consistency of PA, Q being too weak to "understand" $\mathsf{con}(\mathsf{Q})$ (let alone $\mathsf{con}(\mathsf{PA})$) as a consistency statement.[16] There is, more generally, an interesting interplay between the choice of the formula representing some properties and the choice of the theory. Much like the conditions criterion with a fixed base theory, the resemblance condition fails to capture such an interplay.

This analysis suggests that these criteria have two fundamentally distinct roles. First, the resemblance criterion aims to specify the reasonableness of a representation. It is the relation between a formula, i.e., the representation, and the description of some relation, i.e. what is being represented, which informs us whether such representation is reasonable or not. The resemblance criterion simply states that resemblance ensures the reasonableness of the representation. Such a notion is independent of any theory, as it takes place between a formula and a description.

On the other hand, the conditions criterion is not about representation in itself, but rather about reasonable representation *for* a given theory T. It is one thing to have a reasonable representation but another to realise or recognise this representation as reasonable. The conditions criterion is concerned with the latter. On this view, whether a proof of a statement within that theory can be legitimately interpreted as being about what the statement represents also depends on whether the theory recognises it as reasonable. There are many examples where a representation is reasonable, but a theory is too weak to recognise and use it, i.e., too weak to prove the conditions criterion for it. The standard arithmetisation of terms with gn_1 seems reasonable, yet $\mathsf{I}\Delta_0$ fails to prove the conditions for it. Similarly, $\mathsf{con}(\mathsf{PA})$ is a reasonable consistency statement of PA, as well as *for* PA but not *for* Q. Hence, a formula can be a reasonable representation *tout court* without being reasonable *for* T (as well as *vice versa*). If our analysis is correct, then both the resemblance and conditions criteria are needed to fully capture the indeterminacy in metamathematics.

A complete defence and analysis of these claims is beyond the scope of this paper. In the remaining parts of this article, my aim is to apply this understanding to the question of reasonable numbering, which we treat as a case study of the more general thesis. But before applying the

[16] Bezboruah and Shepherdson [4] prove a version of G2 for Q. However, since their proof is substantively different from the usual "modal" proofs, they remark [4, p. 504]: "We must agree with Kreisel that [$\mathsf{Q} \nvdash \mathsf{con}(\mathsf{Q})$] is devoid of any philosophical interest, and that in such a weak system, this formula cannot be said to express consistency, but only an algebraic property which in a stronger system (e.g., Peano arithmetic [PA]) could reasonably be said to express the consistency of [Q]." Visser [80] presents and discusses these issues extensively, see also [23, Chap. 5], [43, 63], and [77].

resemblance and conditions criteria to numberings, it is first necessary to define and motivate what exactly a numbering is supposed represent.

§5. Two Conceptions of the Arithmetisation of Syntax. This section examines two ways to make precise what it means for a numbering to represent syntax, and, more generally, to arithmetise syntax.[17]

Hilbert and Bernays [41, §4] already distinguish two main options for arithmetising the syntax:

> The method of arithmetisation that [Gödel] chooses consists in the *arithmetic imitation of the linear arrangement of signs* in the formulae of formalised mathematics. [...] However, we can also carry out the arithmetisation in a different way, by *arithmetically imitating the grammatical construction* of the formulæ, instead of emulating their writing. The difference in this procedure is that the predicates, symbols [i.e., constants], and function signs [i.e., function symbols], as well as the logical symbols, are associated not with numbers but with arithmetical functions. (Original italics, my translation [41, pp. 216–217].)

On the first option, the arithmetisation of syntax is concerned with the "*arithmetic imitation of the linear arrangement of signs.*" Following Gödel [24], the string numberings such as gn_1 and gn_2 encode any finite sequence of symbols. Hilbert and Bernays suggest an alternative option to carry out the arithmetisation: one may imitate not the "writing" of the formulas, but directly their "*grammatical construction*", which aligns better with the compositional numberings such as gn_3 that recursively encode the syntactic expressions.

As stated, however, these two options provide an incomplete analysis. The syntax can be defined through various implementations across different background universes. Restricting the discussion to a string universe or a particular implementation excludes many, presumably adequate, ways to arithmetise the syntax. Indeed, there is the shared insight that nothing essential depends on these choices, provided that they are themselves reasonable.[18] This is exemplified by the numerous proofs of $\mathsf{G2}$ using different implementations, background universes and numberings. Therefore, a proper and general analysis of the arithmetisation of syntax should not be restricted to a specific choice of background universe or implementation.

In this section, I introduce and discuss the two main conceptions of the arithmetisation of syntax. In §5.1, I present the first conception – the data structures conception – which is a natural generalisation of the first option due to Grabmayr [27, 28, 29]. However, I argue that this conception is inadequate for understanding which numberings are reasonable. In §5.2, I propose a second conception – the proto-syntax conception – and argue in its favour.

[17] In this article, I shall not discuss the reasons for the *arithmetisation* of syntax and, more generally, the use of numberings. Nonetheless, I take this approach to be both historically significant and philosophically motivated. While other objects than numbers can be used to represent syntax, the use of numberings has many advantages. Among others things, we have extensive knowledge of, and trust in, weak arithmetical theories that already seem sufficiently strong to arithmetise syntax. Moreover, the use of natural numbers comes with a robust and notation-free definition of computability via recursive functions, unlike the other common definitions of computability (see, e.g., [30]).

[18] There are mathematical fields that do rely on specific representation, e.g., computable structure theory [52]. However, syntax is usually not taken as one of them. See, e.g., [62] and references therein for some discussions about the objectivity of syntax. Some authors take syntax as being about strings, but such claims seem largely unmotivated. Although the string approach has also been favoured to make precise the notion of a *formal system* (see, e.g., [71, 73]), some alternatives that do not rely on them have also been proposed (see, e.g., [17, 18, 25, 34]). See also [78, §5] and [61, §§1-2] for similar positions about the syntax.

§5.1. The Data Structures Conception. Following Gödel [24], the standard approach to arithmetise syntax assigns a unique number to each element of the background universe. Typically, it "consists in the *arithmetic imitation of the linear arrangement of signs*". The usual definitions of the syntax are then arithmetised with tracking functions of concatenation (see, e.g., [67, §6.2] or [70, Chap. 20]).

Grabmayr [29] proposes generalising this approach with the notion of a data structure. Roughly, a data structure $\mathbf{D}$ is an algebra with a domain D together with some fundamental operations that finitely generate D.[19] Examples include the Kleene star A^* with concatenation, i.e., $\langle A^*, ^\frown\rangle$, the natural numbers and the successor operation, and the hereditary finite sets with the adjunction operation. All these domains provide common background universes for defining our formal expressions.

Since standard background universes can generally be presented as data structure domains, Grabmayr [29] analyses numberings in relation to their data structures. Given a data structure $\mathbf{D}$, a numbering α of D induces a structure isomorphic to $\mathbf{D}$ with $\alpha(D)$ as its domain and the α-tracking function of $\mathbf{D}$'s fundamental operations as its fundamental operations. For instance, with the data structure $\langle A^*, ^\frown\rangle$, our standard numbering gn_1 of A^* induces the isomorphic structure $\langle \mathsf{gn}_1(A^*), ^\frown{}^{\mathsf{gn}_1}\rangle$. Such an arithmetical structure is then taken to be the basis of the arithmetisation of $\mathbf{D}$, i.e., as the representation or simulation of $\mathbf{D}$ induced by the numbering [27, §3.1]. We then say that, for a data structure $\mathbf{D}$, a numbering α of D is a *computable simulation* of $\mathbf{D}$ iff $\alpha(D)$ is decidable and all α-tracking functions of the fundamental operations of $\mathbf{D}$ are recursive.[20] A generalisation of Mal'cev's theorem then shows that any two computable simulations of $\mathbf{D}$ are $\mathbb{REC}$-equivalent, and so they agree on which sets and functions are decidable and recursive, respectively [28, §2.3]. A version of $\mathsf{G2}$ can then be proved to be invariant for numberings that computably simulate their data structures (together with additional conditions on the notation, see [29]).

Under this conception, the arithmetisation of syntax, and hence the evaluation of numberings, is understood primarily through the arithmetisation of data structures. The syntax with its constructor operations is arithmetised and evaluated through the tracking functions of the data structure's fundamental operations. While this approach is widely adopted in many instances of arithmetisation, I consider it to be inadequate for addressing the question of reasonable numberings.

First, the notion of a data structure and its arithmetisation appears to be orthogonal to the arithmetisation of syntax. Although the background universe used to define syntax is often the domain D of a data structure, not every object in D typically represents a syntactic expression. Take $+{\rightarrow}\mathsf{0} \in A^*$, which has no role in, say, the implementation pre. Why should such a string have a code? A numbering α of $A^*\backslash\{+{\rightarrow}\mathsf{0}\}$ that agrees with gn_1 on its domain seems as reasonable as gn_1, and most proofs using α, e.g., of $\mathsf{G2}$, are identical to those using the standard numbering gn_1. As illustrated by our numbering gn_3 on pre, which is defined on $\mathrm{Exp}_{\mathsf{pre}}$ but not on A^*, encoding the set of syntactic expressions is already sufficient. There is thus no need to encode the whole background universe used to define the syntax.

A similar point applies to the notion of computable simulation. For a numbering α on D, it is not clear why we should require the whole $\alpha(D)$ to be decidable, and similarly for the relevant α-tracking functions. Take, for instance, a *non*-recursive function (on the relevant domain) h, and consider the numbering α on pre defined as $2\cdot h(\mathsf{gn}_1(s))+1$ for any $t \in A^*$ and $s \equiv +{\rightarrow}\mathsf{0}t$ and

[19] In more detail, a data structure is a finitely generated many-sorted algebra; all the relevant definitions and further examples can be found in [26, 28, 29]. For simplicity, our presentation and discussion focus on single-sorted data structures, but also apply to the ordered many-sorted case.

[20] This definition is motivated in [27, p. 57]. See also the notion of *computable representation of ordinals* in ordinal analysis [66, pp. 3–4], and [2, 20] for some of its limitations.

as gn_1 otherwise. In this case, the tracking function $^\frown{}^\alpha$ is not recursive, so α is not a computable simulation of $\langle A^*, ^\frown\rangle$. Yet, the sets of the codes of pre-terms and of pre-formulas are still Δ_1 in PA as they can be defined using the same standard definition with gn_1, and the usual α-tracking functions of the pre-constructor functions are all PA-provably recursive, since the corresponding gn_1-tracking functions are.

Second, even if one insists on arithmetising a whole data structure, the focus on data structures adds an additional layer, which introduces further indeterminacy and distracts from the question of reasonable numberings.

The numberings sharing a given mathematical property, and hence the associated invariance results, are relative to the choice of data structure. For a given data structure $\mathbf{D}$, Mal'cev's theorem implies that any two computable simulations of $\mathbf{D}$ are $\mathbb{REC}$-equivalent. However, this does not imply that computable simulations of two distinct data structures are $\mathbb{REC}$-equivalent, even on the same domain. Moreover, a suitable data structure is neither guaranteed to exist nor to be unique: For a numbering α on $\imath$, there might be no data structure $\mathbf{D}$ whose domain is $\mathrm{dom}(\alpha) \supseteq \mathrm{Exp}_\imath$ (as, e.g., with the α defined on $A^*\backslash\{+\!\!\rightarrow\mathsf{0}\}$). And even when such a data structure exists, it need not be unique. For the numbering gn_1 of A^*, there are many different data structures with domain A^*, i.e., many ways to finitely generate A^* (with the usual concatenation operation $^\frown$, with "successor" operations $^\frown a$ for each $a \in A$, etc.). Which of these should we take gn_1 to represent?

This indeterminacy runs even deeper. Among the many data structures, some are deviant (see [29, §5.2] and [30, §7]). Using such deviant data structures, it is possible to make clearly unreasonable numberings, such as our unreasonable gn_4 with X as the set of T-provable sentences, into computable simulations.[21] Thus, unreasonable numberings can still induce a perfectly adequate representation of a deviant data structure. This forces us to consider not only the choice of numberings but also the choice of data structures, thereby introducing a further indeterminacy.

For these reasons, I consider the first conception of arithmetisation inadequate for defining reasonable numberings.[22] I now turn to the second conception.

§5.2. The Proto-Syntax Conception. Following Hilbert and Bernays [41], the second conception of the arithmetisation of syntax focuses on the generation of syntactic expressions. Since an implementation $\imath$ associates the symbols of the alphabet with $\imath$-constructor functions, this conception of arithmetisation associates such functions with arithmetical functions. In the same way a formula is generated by some sequence of $\imath$-constructor functions, its code is generated by the same sequence of *arithmetised* $\imath$-constructor functions.

At first sight, this suggests assessing numberings with respect to their representation of the underlying implementation. However, this would understate the independence of arithmetisation from the specific implementation. Hilbert and Bernays [41, p. 217]'s reference to the "*grammatical construction*" appears to run deeper than any specific implementation of that construction.

[21] See [30, §§6-7] for a discussion of deviant data structures. For the invariance results, Grabmayr [29] avoids such unwanted consequences by considering "notation systems", i.e., a numbering together with a data structure and an implementation: for an implementation $\imath$ on a background universe D, not only must a numbering α be a computable simulation of $\mathbf{D}$, but, in addition, all the α-tracking functions of the $\imath$-constructor functions need to be recursive.

[22] Another alternative approach uses the notion of interpretation, initially proposed by Halbach and sketched in Heck [40, §2.1] (see also [36, pp. 364–365]). The idea is, roughly, to ask the base theory T to interpret a specific theory of concatenation. The numberings then play a central role in defining the translation functions used in the interpretation. Heck [40] uses Grzegorczyk [33]'s theory of concatenation TC, and, presumably, a theory T cannot interpret TC with deviant numberings. It seems that this proposal would run into issues similar to those raised in this section. In addition, the notion of interpretation does not seem to be the correct tool for this debate as it is mainly concerned with structural rather than computable properties (see, e.g., [21, 60]). As I argue in §6, computable properties play a central role in the study of numberings.

Kleene [45, pp. 246–258] adopts a similar position with his notion of "generalized arithmetic entities" which are built so as to "emphasize [the] structure (i.e., modes of generation)" of each syntactic expression. Indeed, we can arithmetise the generation of a formula, e.g., via a compositional numbering, independently of the specific generation of the formula given by the implementation. While some authors do define compositional numberings to mimic their implementation, e.g., our gn_3 on pre or [74, p. 246]'s numbering on inf, it is more common to define them independently of their underlying implementation (see, e.g., [1, p. 259] or [41, pp. 217–219] which both of which implement syntax with inf but arithmetise it by mimicking the simpler pre implementation).

To articulate this approach, I rely on the notion of proto-syntax, which extends our presentation of the syntax in §2. This notion is motivated by the insight that both '$\rightarrow \varphi\, \psi$' and '$(\varphi \rightarrow \psi)$' are different representations, in prefix and infix notation respectively, of the implication from φ to ψ. There is thus, by abstraction, an intuitive notion of implication, and similarly with any constructor function, which guides the implementation of syntax.[23] As Kleene [45, p. 62] nicely puts it: "An intuitive mathematics is necessary even to define the formal mathematics." Such an intuitive mathematics helps explain why we share the insight that our metamathematical results should hold with some generality and independently of the non-essential details of implementation and arithmetisation. Let us then refer to the intuitive conception of syntax that underlies any implementation and arithmetisation as the proto-syntax. Following Grabmayr [29, §§4.1–4.2], let $\Omega(\mathcal{L})$ be the abstract signature of the language $\mathcal{L}$. Since an implementation $\imath$ inductively and freely generates the $\imath$-variables, $\imath$-terms, and $\imath$-formulas, there is a unique free $\Omega(\mathcal{L})$-algebra $\mathbf{S}_\imath$ whose (overall) domain is the set of the $\imath$-expressions and whose fundamental operations are the $\imath$-constructor functions. The structure of the proto-syntax is then the free algebra $\mathbf{S}$, which, for each implementation $\imath$, is isomorphic to $\mathbf{S}_\imath$. An implementation can then be analysed as specifying a particular instantiation of $\mathbf{S}$. More details can be found in the Appendix A.[24]

I suggest understanding the arithmetisation of syntax through the arithmetisation of proto-syntax, and in particular its structure $\mathbf{S}$. A numbering is not taken as representing a given implementation of syntax (or a data structure), but rather as giving its own representation, or instantiation, of the proto-syntax. Since, by definition, every numbering is defined on the set of expressions, the particular representation of the proto-syntax of any numbering can be defined as follows: Given a numbering α on an implementation $\imath$, there is an induced free algebra, whose (overall) domain is $\alpha(\mathrm{Exp}_\imath)$ and whose fundamental operations are the α-tracking functions of the $\imath$-constructor functions. Let us call such an arithmetical structure the α-syntax. Being a free algebra on $\Omega(\mathcal{L})$, the α-syntax is also isomorphic to $\mathbf{S}$, and it is this structure that I take to be the arithmetical representation of the proto-syntax induced by α. See Appendix A for a specific

[23] Avigad [1, p. 199] remarks: "It can be disconcerting that there are often multiple representations available with no strong reason to favour one over another. Such concerns are somewhat addressed by abstracting the properties of the representation, thereby providing a modular interface that specifies the design requirements and encapsulates the details of implementation." Our analysis with the proto-syntax follows precisely what is advocated here by Avigad. Note that this notion of proto-syntax is distinct from Quine [65, §55]'s, who uses it for a string-based view of syntax.

[24] Our notion of the structure of the proto-syntax slightly differs from the one in Grabmayr [29, §4.2] insofar as it uses *ordered* many-sorted algebra. This allows the elements of the sort of variables to also be elements of the sort of terms, and thereby avoids the need for an artificial transfer function from variables to terms (see Grabmayr [29, Def. 4.2]). There are other ways to define the structure of the proto-syntax, in particular with respect to the treatment of variables and quantifiers. Proponents of the Peirce-Quine-Bourbaki notation might, for instance, take it as the basis for the structure of the proto-syntax and understand more standard implementations, such as pre, as derivative. Interestingly, choosing a different definition of the proto-syntax will have an influence on the notion of reasonable numberings developed in the upcoming sections. This is to be expected. If a reasonable numbering encodes the syntax, different notions of the syntax will yield different notions of reasonable numberings. I shall not investigate these options in this article and leave the discussion of the correct conception of the proto-syntax, as well as a complete defence of the algebraic conception of the syntax, to future work (see also [5, 29]).

example of this construction.

The proto-syntax conception may be viewed as a special case of the data structures conception, in which the only relevant data structure, common by definition to all numberings, is the proto-syntax (see also [29, Remark 4.12]). The key advantage of this restriction is that it frees the analysis of numberings from what Halbach and Visser [38, p. 711] call the various "matter[s] of implementation". Which implementation and background universe are reasonable? What is the "essence of conjunction"? Are brackets essential or not? (See, e.g., [38, p. 711], [29, §5], and [30, §§6–7].) In the present framework, such distracting questions are set aside in favour of questions concerning the numbering itself and its representation of the proto-syntax.

This abstraction is justified by the fact that the induced syntax does not retain information about the implementation on which it was induced. Although the construction of the α-syntax refers to the particular implementation on which the numbering α is defined, the resulting structure is purely arithmetical and represents the syntax independently of that implementation. Indeed, one cannot recover from the α-syntax the particular implementation or data structure used in its construction. Furthermore, given any implementation $\imath$, one can define a numbering β on $\imath$ whose induced β-syntax is identical to the α-syntax. Thus, the specific implementation plays no essential role once the induced syntax has been obtained. There is therefore no need to further generalise over implementations, data structures, or background universes: the representation of the proto-syntax by a numbering is independent of such notions.

A further advantage of this conception is its uniformity. Since every numbering α induces an α-syntax that is isomorphic to the structure of the proto-syntax $\mathbf{S}$, all numberings can be studied within a single and uniform framework. Conditions imposed on the notion of induced syntax apply uniformly across all numberings, thereby defining classes of numberings that are not relative to any particular implementation or data structure.

To illustrate, suppose that the domains and fundamental operations of each induced syntax are required to be decidable and recursive, respectively. Mal'cev's theorem then yields a unique class of $\mathbb{RELC}$-equivalent numberings relative to their induced syntax. Of course, numberings in this class need not themselves be $\mathbb{RELC}$-equivalent, since they may not even be defined on the same domain. Nevertheless, since the basic syntactic relations are solely defined on the syntactic expressions, the recursivity of any such syntactic relation is invariant throughout this class. Together with an analysis of the proofs in Grabmayr [29, §7], this observation shows that our standard metamathematical results generalise to this class of numberings.

§6. Towards Reasonable Numberings. In the previous section, I argued in favour of the second conception of the arithmetisation of syntax, developed with the notion of proto-syntax. The key observation was that every numbering α induces a particular representation of the structure of the proto-syntax — the α-syntax. I therefore suggest answering the question of which numberings are reasonable by asking whether their induced structures are themselves reasonable representations of the proto-syntax.

In §4, I briefly argued that, to answer questions about reasonable representations in metamathematics, one should use both the resemblance and conditions criteria. This section develops this approach, by investigating the conditions under which the induced α-syntax of a numbering α resembles the proto-syntax. I motivate some resemblance conditions for numberings in §6.1 and argue for several standard necessary properties of reasonable numberings. This requires, first, defining and motivating the key properties of the proto-syntax, and, second, to specify how such properties are represented by some arithmetical properties of numberings. Then, in §6.2, I discuss the relation between the choice of numbering and the base theory, and suggest a conditions criterion for numberings. Together, the resemblance and conditions criteria for numberings provide a general framework for addressing questions about numberings, and, in

particular, for motivating some of their usual properties.[25]

§6.1. Resemblance Criterion for Numberings. Which properties of numberings can be motivated by the resemblance criterion for numberings? On the present analysis, the key properties of the proto-syntax will provide, via resemblance, some necessary conditions on reasonable numberings. I first discuss structural properties, followed by computable properties, which, as argued in §4.1, also enter the resemblance criterion.

Structural Properties. The definition of numberings imposed two structural conditions on them: functionality and injectivity. Both conditions are standard in metamathematics and can be motivated within this framework by a resemblance condition.

First, numberings are standardly assumed to be functions. This assumption greatly simplifies the arithmetisation. However, some authors have challenged it (see, in particular, [4, p. 507] and [78]). They argue that there may be no harm in allowing multiple numbers to code a single object (as, e.g., with Gödel's β-function and codings of expressions based on it). One could understand this by analogy with the fact that recursive functions have many indices in computability theory. These considerations are, of course, technically correct (it requires some extra care, since one should not be able to prove that the encoded objects have contradictory properties with respect to its different codes.) It is nonetheless difficult to philosophically assess and motivate whether or not reasonable numberings should be functional.

Second, numberings are also standardly assumed to be injective. This requirement is obvious. For suppose a function α assigns a number n to two distinct terms s and t, and that only s has a given syntactic property. Since proofs carried out in terms of the code n would not discriminate between s and t, we may incorrectly attribute the property to both terms.

In our framework, the following resemblance condition for numberings gives us reasons to take functionality and injectivity to be necessary properties for a numbering to be reasonable. The main structural properties of the proto-syntax are that its domains are inductively and freely generated by its fundamental operations. These properties are essential because they allow us to prove recursion theorems, thereby ensuring that our basic syntactic definitions are well-defined. As we previously noted, this amounts to taking $\mathbf{S}$ as the structure of the proto-syntax. If isomorphism is taken as the relevant notion of resemblance, or "sameness", among structures, a well-defined notion of structural resemblance for numberings can be stated as follows: For a numbering α, the α-syntax structurally resembles the proto-syntax iff the α-syntax is isomorphic to $\mathbf{S}$. It is then easy to see that any numbering α (understood as an injective function) induces an α-syntax, which is isomorphic to $\mathbf{S}$. By contrast, non-functional or non-injective coding relations $R \subseteq \mathrm{Exp}_\iota \times \omega$ do not, without further quotienting or disambiguation, induce such an isomorphic copy. Hence, under this analysis, the structural condition of injectivity and functionality are justified as necessary conditions for reasonable numberings. Note that since these two requirements, and more generally our approach, rely on the resemblance between the α-syntax and the structure of the proto-syntax $\mathbf{S}$, different accounts of what is the structure of the proto-syntax might motivate different structure requirements.[26]

[25] Note that I deal in this section with the notion of reasonable numberings. However, this notion could also be relativised to a given metamathematical context. Take, for example, the numbering gn_4 (with X as the PA-provable sentences). The set of gn_4-codes of the terms is decidable, and all the gn_4-tracking functions for the terms are also recursive: gn_4 being essentially equivalent to the standard gn_1 with respect to the terms. We could then understand gn_4 as being partially reasonable (for the terms). Our analysis based on the resemblance and conditions criteria could then be relativised to metamathematical context, by considering only the relevant subparts of the proto-syntax.

[26] For simplicity in this article, Gödel numberings are injective functions whose domain includes the set of expressions, i.e., the set of all elements that are either a term or a formula together with the assumption that no

Are there other structural properties that could be motivated within this framework by some resemblance condition? Two common candidates are the properties of monotonicity and domination, that most standard numberings in the literature satisfy. Nonetheless, it is difficult to see how such properties could be motivated within our framework. Let us briefly consider them.

First, the property of monotonicity, in its weaker version, amounts to requiring the codes of the sub-expressions of an expression to be less than the code of the expression itself. It is tempting to understand monotonicity as reflecting some structural property of the syntax. For instance, one could argue that the elements of a free algebra are generated from "less complex" and previously generated elements, and thereby motivate monotonicity via resemblance. This line, however, does not fit well within our framework. The relevant notion of complexity at play in an algebra is the number of generative steps, not the intrinsic complexity of elements so generated. The fact that an element a is generated in fewer steps than an element b does not imply that a is, in the relevant sense, less complex than b. One could, for example, argue that the number 7, being prime, is less complex than the number 6, although it is generated in more steps by the successor operation. In the absence of other conditions, it is unclear how this would motivate monotonicity. Grabmayr [27, p. 63] points out another issue: the less-than order on numbers is total, whereas the sub-expression relation is only partial. Again, it is unclear what the correct notion of resemblance would be between such different types of ordering (see [27, §3.4] for a more detailed discussion of monotonicity and related criticisms).

Second, the property of domination requires the evaluation in the standard model of a term to be less than its code (see [32, §8]). The appeal to the evaluation of a term is external to the notion of the proto-syntax, and therefore lacks any motivation within our framework.

Given that most standard numberings do possess these properties, they could well be consequences of other structural requirements which are yet to be found. I leave open whether further structural properties should enter the resemblance criterion for numberings.

Computable Properties. Typically, numberings are assumed to be efficient (see, e.g., [31, 52]). Although there are several accounts on how to make this requirement mathematically precise (see [27, 40, 50, 70, 71]), the literature largely lacks an explanation of why, and how, such a condition should be imposed on reasonable numberings (Grabmayr [27, §3.2] is a noteworthy exception which I discuss below).

As an example, let us consider the numbering gn_4 (with X as the PA-provable sentences) which is usually regarded as unreasonable by appealing to computable properties. Such a numbering encodes the notion of PA-provability, and in doing so, some of the gn_4-tracking functions of the pre-constructor functions are not recursive, while making the set of PA-provable sentences decidable.[27] These functions are usually assumed to be at least recursive whereas the set of PA-provable sentences is not. This creates a mismatch between the expected complexity of these notions and the complexity of their sets of gn_4-codes For this reason, gn_4 is taken as an unreasonable choice. This analysis is then captured by the resemblance condition between the proto-syntax and the arithmetical structures induced by each numbering.

term is itself a formula. But since the needed recursion theorems concern the sets of terms and formulas, rather than the set of expressions, one could then understand a notion of numbering α as a pair of injective functions $\alpha_{\mathrm{Ter}_\iota}$ and $\alpha_{\mathrm{Fml}_\iota}$ on the terms and the formulas. In this conception, the requirement of injectivity would be on $\alpha_{\mathrm{Ter}_\iota}$ and $\alpha_{\mathrm{Fml}_\iota}$, but a number n could be both the $\alpha_{\mathrm{Ter}_\iota}$-code of a term and the $\alpha_{\mathrm{Fml}_\iota}$-code of a formula. (Alternatively, α could be taken as a function on Exp_ι for which $\alpha{\restriction}\mathrm{Ter}_\iota$ and $\alpha{\restriction}\mathrm{Fml}_\iota$ are injective.) In this case, as a reviewer pointed out, the present argument is really a case for output-sort-injectivity.

[27] See [27, Appendix] for an interesting study on, given a standard implementation $\imath$, how many α-tracking functions of the $\imath$-constructor functions can be recursive while keeping the set of PA-provable sentences decidable.

As Quine [65, p. 291] puts it, the syntax should enjoy "systematic recognizability", or in other words, it exhibits certain computable properties. Typically, the properties of being a term or being a formula should be at least computable.[28] For example, Hájek and Pudlák [34, pp. 21–22] work with computable presentations of the syntax (with background universe of numbers, they require the relevant sets and functions to be Δ_1. This assumption reflects the common view that $\mathsf{I}\Sigma_1$ is sufficiently strong to arithmetise the syntax. It is because the set of the codes of formulas is Δ_1 (and, furthermore, can be made Δ_1 in $\mathsf{I}\Sigma_1$) that $\mathsf{I}\Sigma_1$, with its induction on Σ_1-formulas, is adequate, whereas, e.g., Q is not. Alternatively, one could require the syntax to be of a much lower complexity, such as primitive recursive, or even p-time computable (see, e.g., [76, p. 543]).

I shall not argue here for the specific complexity of the proto-syntax, and simply assume that it is at least computable and, more importantly, fixed. Accordingly, the domains and fundamental operations of the proto-syntax are taken to have this complexity. Such properties must then be represented by a numbering, via some resemblance condition. Here again, there is a natural notion of resemblance between computable (or weaker) properties and, via the relevant Church-Turing thesis and arithmetical hierarchies, arithmetical properties on sets of numbers and arithmetical functions. It follows that, for a numbering α to be reasonable, its α-syntax should also possess the corresponding arithmetical properties: the sets of codes of variables, terms, and formulas, as well as all α-tracking functions of the fundamental operations of $\mathbf{S}$ must match the complexity of the proto-syntax. In particular, the α-syntax should at least have decidable domains and recursive fundamental operations.

Grabmayr [27, §3.2] rejects this view and holds that the complexity of reasonable numberings is "sensitive to the considered metamathematical context", and, in particular, to the choice of the base theory and its "resources." Since, say, Tarski's undefinability of truth is about the definability of truth, some definable properties are taken to be necessary conditions for reasonableness, whereas, for $\mathsf{G2}$, recursive properties are necessary. Interestingly, when it comes to generalisations, we can prove the invariance of Tarski's result under definable numberings and of $\mathsf{G2}$ under recursive numberings [29, §7]. However, as Visser [77, p. 68] reminds us, there is "no good reason that these questions [i.e., mathematical generalisations and philosophical questions about reasonable choices] should be in sync." Grabmayr's position directly conflicts with our analysis of numberings through their representations of the proto-syntax. If one believes that our results about syntax are absolute, then there is no motivation to relativise the complexity of reasonable numberings with respect to the metamathematical context. In other words, the complexity of the syntax, and hence the reasonable numberings, is orthogonal to the choice of the base theory or the metamathematical notion under investigation. Nonetheless, Grabmayr is correct in stressing that the base theory, or its "resources", plays a central role in the choice of numberings. This is the concern of the next section.

Let us conclude this subsection by noting that the question of which properties characterise the reasonable numberings is not settled. In particular, if self-referential numberings, such as gn_5, are taken as unreasonable, then our necessary properties are insufficient. Indeed, there are self-referential numberings of elementary complexity, and it is currently unknown whether such numberings can be of p-time complexity (see [32, §7.]). If the proto-syntax is of higher complexity, the properties considered in this section are then insufficient to exclude self-referential numberings. I have also dismissed the properties of monotonicity and domination, often invoked to rule out such numberings. Therefore, further investigation would be needed to exclude self-

[28] This informal claim expresses an intuitive constraint on any implementation of syntax. Making it precise raises familiar difficulties, as different background universes have different notions of computability, which are not all extensionally equivalent. See, e.g., [30].

referential numberings from the reasonable numberings.[29]

§6.2. Conditions Criterion for Numberings. Let us turn now to the question of how the choice of the base theory interacts with the choice of numbering. If the preceding resemblance condition provides some necessary conditions for a numbering to be reasonable, by itself, it does not settle the question of what it means for a numbering to be reasonable *for* a given theory.

The intuitive distinction between a numbering being reasonable simpliciter and its being reasonable for a given theory T can be illustrated as follows. Presumably, the standard numbering gn_1 is both reasonable as well as reasonable for PA. However, in §3, I showed that $\mathsf{I}\Delta_0$ fails to prove most basic syntactic results with gn_1. In this sense, despite being reasonable, gn_1 fails to be reasonable for $\mathsf{I}\Delta_0$. The numbering gn_4, with X as the set of PA-provable sentences, is an obvious candidate for a numbering that is neither reasonable nor reasonable for PA since PA even fails to binumerate some of the gn_4-tracking functions of the pre-constructor functions. Note, however, that gn_4 may be reasonable for stronger (not necessarily c.e.) base theories, provided that they can handle PA-provability, while still being an unreasonable numbering. A similar phenomenon occurs when we consider $\mathsf{I}\Sigma_1$ and PA with the numbering gn_4 with X as the set of $\mathsf{I}\Sigma_1$-provable sentences. Finally, self-referential numberings such as gn_5 could well be taken as unreasonable, while still being reasonable for PA. Indeed, gn_5 can be made $\mathbb{REC}$-equivalent to gn_1.

In §4.3, I briefly argued that the conditions criterion gives us a way to specify when a representation is reasonable for a given theory T. Given our previous analysis, for a numbering α to be reasonable for T, T should recognise that the structure induced by α is a representation of the proto-syntax. In other words, it should prove sufficiently many basic features of the α-syntax. The aim of this subsection is not to offer a complete or decisive analysis of the conditions criterion for numberings. Rather, starting from the previous necessary structural and computable properties for reasonable numberings, I outline a conditions criterion for numberings which relativise the class of reasonable numberings to the choice of the base theory.

Binumeration. Structural resemblance requires a numbering to induce an isomorphic copy of the structure of the proto-syntax. For a numbering α to be reasonable for a base theory T, it is therefore natural to ask T to "recognise" the α-syntax, i.e., which numbers are codes for variables, terms and formulas, as well as the generation of all these codes via the α-tracking functions of the fundamental operations of the proto-syntax. A minimal constraint is thus to ask T to binumerate the domains and fundamental operations of the α-syntax. Since any c.e. theory extending Q binumerates the same sets and functions (i.e., the decidable and recursive ones, respectively), this condition makes no difference between the reasonable numberings and reasonable numberings for any such theory. Nonetheless, if we consider stronger (not c.e.) base theories, some unreasonable numberings might become reasonable for some of these theories.

Note that the condition of binumeration alone does not seem to be sufficient to define our notion of reasonableness for a theory (see, e.g., [35, pp. 34–35]). A more restrictive condition would be to require T to prove the injectivity and functionality of the numbering on the domains of syntactic expressions.

[29] Three strategies appear viable in this framework: (i) to motivate an additional structural property that rules them out; (ii) to show that there are no self-referential numberings of p-time complexity and argue that the proto-syntax should have this complexity; or (iii) to extend the analysis to operations on the proto-syntax such as substitution understood as homomorphisms and, via some resemblance condition, to impose corresponding constraints on numberings. I am sceptical that the proto-syntax possesses other *kinds* of properties than the structural and computable ones.

Provably Recursive Functions. In addition to structural properties, our resemblance condition requires numberings to reflect the computability aspects of the proto-syntax. Accordingly, T should also recognise the correct complexity of the α-syntax for α to be reasonable for T. I suggest capturing this by requiring T to binumerate the components of the α-syntax with formulas of adequate complexity in T. If the proto-syntax is taken to be computable, then T should binumerate the components of the α-syntax not only with the usual Σ_1-formulas, but also with Δ_1-formulas, thereby recognising their correct complexity (via the relevant Church-Turing thesis and arithmetical hierarchy). Similar requirements would apply if the proto-syntax is taken to be of lower complexity.

This condition follows from the more general and common requirement that the α-tracking functions of the fundamental operations of $\mathbf{S}$ be provably recursive in T (see, e.g., [35, p. 33]). For instance, let $\mathtt{neg}^{\alpha}_{\mathbf{S}}$ be the α-tracking function for negation. If such a function is provably recursive in T, it then follows that the theory T binumerates $\mathtt{neg}^{\alpha}_{\mathbf{S}}$ with a Δ_1-formula in T (see [34, p. 48]).[30] Such a condition would explain why gn_1 on pre is not reasonable for $\mathsf{I}\Delta_0$. Since gn_1 relies on exponentiation, there is no polynomial in x that bounds $\neg^{\mathsf{gn}_1}_{\mathsf{pre}}$. Hence, $\mathsf{I}\Delta_0$ fails to prove the totality of $\neg^{\mathsf{gn}_1}_{\mathsf{pre}}$, thereby failing to recognise it as computable.

I leave open whether further well-motivated conditions should be imposed on numberings that are reasonable for a base theory.

§7. Conclusion. In this article, I have examined numberings from a philosophical perspective, with a particular focus on how a precise definition of reasonable numberings can be formulated and motivated. After introducing the debate and the relevant data and definitions in §§2 and 3, I proposed in §4 approaching the question of reasonable numberings through two criteria already used in the literature: the resemblance and conditions criteria. In §5, I argued against understanding arithmetisation primarily as the arithmetisation of data structures, and proposed instead an algebraic conception based on the notion of proto-syntax. This conception enabled the formulation and defence of resemblance and conditions criteria for numberings in §6:

Given a signature $\mathcal{L}$, an implementation $\imath$ of the syntax of $\mathcal{L}$, and a base theory T, I proposed that a numbering α on $\imath$ should count as:

- *reasonable* iff the α-syntax resembles the proto-syntax of $\mathcal{L}$,
- *reasonable for* T iff T proves the relevant conditions for the α-syntax.

For both criteria, I defended several common requirements on numberings, while rejecting others.

Overall, the analysis offers a unified framework for formulating and assessing questions about numberings and their role in metamathematics, within which further technical and philosophical constraints for defining reasonable numberings may be explored.

§A. Appendix. The aim of this appendix is to outline the structure of the proto-syntax as a free algebra and to show how, for any numbering α, one can define its representation of the proto-syntax: the α-syntax.

[30] Note that the converse does not hold. Furthermore, this condition is trivial for some *un*sound theories. The issue lies in the usual definition of recursive provability (see [32, Rem. 7.5.]). However, this may not be too worrisome since having an unsound base theory seems to be a nonstarter for interpreting metamathematical results.

Structure of the Proto-Syntax. We work in the setting of *ordered many-sorted algebra*: these are many-sorted algebras in which an inclusion ordering on sorts is allowed. Let us write $s_1 \subseteq s_2$ to mean that s_1 is a sub-sort of s_2, i.e., that the domain of s_1 is a subset of the domain of s_2. See [29, Appendix A], [26], and [58] for additional details.

The abstract signature $\Omega(\mathcal{L})$ of the basic language of arithmetic $\mathcal{L}$ is defined as a $\{\mathtt{var}, \mathtt{ter}, \mathtt{fml}\}$-sorted language with the subsort relation $\mathtt{var} \subseteq \mathtt{ter}$. It comprises the following $\Omega(\mathcal{L})$-symbols:

1. `v` is a function symbol of type $\mathtt{var}^0 \to \mathtt{var}$
2. `nvar` is a function symbol of type $\mathtt{var} \to \mathtt{var}$
3. `zero` is a function symbol of type $\mathtt{ter}^0 \to \mathtt{ter}$
4. `succ` is a function symbol of type $\mathtt{ter} \to \mathtt{ter}$
5. `add` and `mul` are function symbols of type $\mathtt{ter}^2 \to \mathtt{ter}$
6. `id` is a function symbol of type $\mathtt{ter}^2 \to \mathtt{fml}$
7. `bot` is a function symbol of type $\mathtt{fml}^0 \to \mathtt{fml}$
8. `cond` is a function symbol of type $\mathtt{fml}^2 \to \mathtt{fml}$
9. `univ` is a function symbol of type $\mathtt{var} \times \mathtt{fml} \to \mathtt{fml}$

A $\Omega(\mathcal{L})$-algebra $\mathbf{A}$ is the following structure:

$$\langle\langle |\mathtt{var}|, |\mathtt{ter}|, |\mathtt{fml}|\rangle, \langle \sigma_{\mathbf{A}}\rangle_{\sigma\in\Omega(\mathcal{L})}\rangle$$

with $|\mathtt{var}| \subseteq |\mathtt{ter}|$, where $|s|$ denotes the domain of the sort s and $\sigma_{\mathbf{A}}$ is a fundamental operation of $\mathbf{A}$ given by the symbol $\sigma \in \Omega(\mathcal{L})$. A *free* $\Omega(\mathcal{L})$-algebra is an algebra $\mathbf{A}$ with the universal property: for every $\Omega(\mathcal{L})$-algebra $\mathbf{B}$ there exists a unique $\Omega(\mathcal{L})$-homomorphism $f\colon \mathbf{A} \to \mathbf{B}$. The term algebra $\mathbf{T}_{\Omega(\mathcal{L})}$ is a canonical example of a free $\Omega(\mathcal{L})$-algebra.

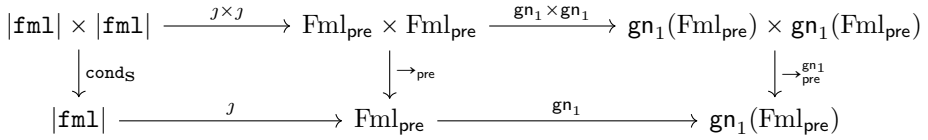
For any implementation $\imath$ under consideration, the sets of variables, terms, and formulas are inductively and freely generated. Moreover, $\mathrm{Var}_\imath \subseteq \mathrm{Ter}_\imath$ and $\mathrm{Ter}_\imath \cap \mathrm{Fml}_\imath = \varnothing$. Hence, there is a corresponding free $\Omega(\mathcal{L})$-algebra $\mathbf{S}_\imath$ whose domains are $\mathrm{Var}_\imath$, $\mathrm{Ter}_\imath$, and $\mathrm{Fml}_\imath$, and whose fundamental operations are the $\imath$-constructor functions. It follows that any such $\mathbf{S}_\imath$ is isomorphic to the free $\Omega(\mathcal{L})$-algebra $\mathbf{S}$ – the structure of the proto-syntax of $\mathcal{L}$.

The α-Syntax. For an implementation $\imath$, let $\jmath$ be the isomorphism between $\mathbf{S}$ and $\mathbf{S}_\imath$. Given a numbering α on $\imath$, the *α-syntax* is the free $\Omega(\mathcal{L})$-algebra induced by a push-through construction from $\mathbf{S}$ via $\alpha \circ \jmath$ (see, e.g., [10, p. 37]). In particular, the α-syntax is the following $\Omega(\mathcal{L})$-algebra:

$$\langle\langle \alpha(\mathrm{Var}_\imath), \alpha(\mathrm{Ter}_\imath), \alpha(\mathrm{Fml}_\imath)\rangle, \langle \sigma_\imath^\alpha\rangle_{\sigma\in A}\rangle$$

where the domains of the α-syntax are the sets of α-codes of variables, terms, and formulas, and its fundamental operations are the α-tracking functions of the corresponding $\imath$-constructor functions, restricted to the relevant domain. This algebra is a well-defined free $\Omega(\mathcal{L})$-algebra, and hence is isomorphic to $\mathbf{S}$.

To illustrate this construction, let us consider the structure induced by the standard numbering gn_1 on pre. The gn_1-syntax is then defined as the free $\Omega(\mathcal{L})$-algebra whose domains are the sets of codes $\mathsf{gn}_1(\mathrm{Var}_{\mathsf{pre}})$, $\mathsf{gn}_1(\mathrm{Ter}_{\mathsf{pre}})$, and $\mathsf{gn}_1(\mathrm{Fml}_{\mathsf{pre}})$ and whose fundamental operations are gn_1-tracking functions of the pre-constructor functions (again restricted to their relevant domains). Then the following diagram commutes:



where $\jmath\colon \mathbf{S} \to \mathbf{S}_{\mathsf{pre}}$ and $\mathsf{gn}_1 \circ \jmath\colon \mathbf{S} \to \mathsf{gn}_1$-syntax are isomorphisms, and all the functions on the diagram are restricted to the relevant domains.

Acknowledgement. I am indebted to Alessandro Giordani, Bathasar Grabmayr, Volker Halbach, Byron Simmons, Albert Visser, Daniel Waxman and two anonymous referees for helpful comments and suggestions. I would also like to thank Carlo Nicolai, Lavinia Picollo, and Lorenzo Rossi for various discussions on related topics. Finally, I presented versions of this material in various places and thank the audiences for constructive feedback and discussion.

Funding. This paper was supported by the Swiss National Science Foundation, the SNSF Doc.CH project "Intensionality in Metamathematics and Gödel's Second Incompleteness Theorem", grant number 227030.

References

[1] J. Avigad. *Mathematical Logic and Computation*. Cambridge University Press, 2022.

[2] L. D. Beklemishev and F. N. Pakhomov. *Automatic structures and the problem of natural well-orderings*. 2024. arXiv: 2407.10198 [math.LO].

[3] P. Bernays. 'Review of "On the consistency of Quine's new foundations for mathematical logic", by B. Rosser'. In: *Journal of Symbolic Logic* 5.1 (1940), pp. 32–33.

[4] A. Bezboruah and J. C. Shepherdson. 'Gödel's second incompleteness theorem for Q'. In: *The Journal of Symbolic Logic* 41.2 (1976), pp. 503–512.

[5] J.-Y. Béziau. 'The mathematical structure of logical syntax'. In: *Advances in Contemporary Logic and Computer Science*. Ed. by W. A. Carnielli and I. M. L. D'Ottaviano. Vol. 235. Contemporary Mathematics. American Mathematical Society, 1999, pp. 3–15.

[6] G. Boolos, J. Burgess and R. Jeffrey. *Computability and Logic*. 5th ed. Cambridge University Press, 2007.

[7] E. Brauer. 'The dependence of computability on numerical notations'. In: *Synthese* 198.11 (2021), pp. 10485–10511.

[8] S. R. Buss. *Bounded Arithmetic*. Bibliopolis, 1986.

[9] S. R. Buss. 'Chapter II: First-order proof theory of arithmetic'. In: *Handbook of Proof Theory*. Ed. by S. R. Buss. Vol. 137. Studies in Logic and the Foundations of Mathematics. Elsevier, 1998, pp. 79–147.

[10] T. Button and S. Walsh. *Philosophy and Model Theory*. Oxford University Press, 2018.

[11] P. Cegielski and D. Richard. 'On arithmetical first-order theories allowing encoding and decoding of lists'. In: *Theoretical Computer Science* 222.1 (1999), pp. 55–75.

[12] Y. Cheng. 'Current research on Gödel's incompleteness theorems'. In: *The Bulletin of Symbolic Logic* 27.2 (2021), pp. 113–167.

[13] A. Church. 'An unsolvable problem of elementary number theory'. In: *American Journal of Mathematics* 58.2 (1936), pp. 345–363.

[14] S. Cook and P. Nguyen. *Logical Foundations of Proof Complexity*. Cambridge University Press, 2010.

[15] M. Detlefsen. 'What does Gödel's second theorem say?' In: *Philosophia Mathematica* 9.3 (2001), pp. 37–71.

[16] H. B. Enderton. *A Mathematical Introduction to Logic*. New York: Academic Press, 1972.

[17] S. Feferman. 'Finitary inductively presented logics'. In: *Logic Colloquium '88*. Ed. by A. Ferro, C. Bonotto, S. Valentini and A. Zanardo. Studies in Logic and the Foundations of Mathematics 127. Amsterdam New York Oxford Tokyo: North-Holland, 1989. Pp. 191–220.

[18] S. Feferman. 'Arithmetization of metamathematics in a general setting'. In: *Fundamenta Mathematicae* 49 (1960), pp. 35–92.

[19] S. Feferman. 'Toward useful type-free theories. I'. In: *The Journal of Symbolic Logic* 49.1 (1984), pp. 75–111.

[20] S. Feferman. 'Three conceptual problems that bug me'. Unpublished lecture notes presented at the 7th Scandinavian Logic Symposium, Uppsala, Sweden, August 18–20, 1996. 1996. URL: `https://math.stanford.edu/~feferman/papers/conceptualprobs.pdf`.

[21] F. Ferreira and G. Ferreira. 'Interpretability in Robinson's Q'. In: *Bulletin of Symbolic Logic* 19.3 (2013), pp. 289–317.

[22] M. Fitting. *Incompleteness in the Land of Sets*. London: College Publications, 2007.

[23] C. Franks. *The Autonomy of Mathematical Knowledge: Hilbert's Program Revisited*. Cambridge University Press, 2009.

[24] K. Gödel. 'Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I'. In: *Monatshefte für Mathematik* 38.1 (1931), pp. 173–198.

[25] K. Gödel. 'On undecidable propositions of formal mathematical systems'. In: *Kurt Gödel Collected Works Volume I: Publications 1929–1936*. Ed. by S. Feferman, J. W. Dawson, S. C. Kleene, G. H. Moore and J. v. Heijenoort. Oxford University Press, 1986, pp. 346–371.

[26] J. A. Goguen, J. W. Thatcher, E. G. Wagner and J. B. Wright. 'Initial algebra semantics and continuous algebras'. In: *Journal of the ACM* 24.1 (1977), pp. 68–95.

[27] B. Grabmayr. 'On the invariance of Gödel's second theorem with regard to numberings'. In: *The Review of Symbolic Logic* 14.1 (2021), pp. 51–84.

[28] B. Grabmayr. 'On the Philosophical Interpretation of Metamathematical Results'. PhD thesis. Humboldt-Universität, 2021.

[29] B. Grabmayr. 'A step towards absolute versions of metamathematical results'. In: *Journal of Philosophical Logic* 53 (2024), pp. 247–291.

[30] B. Grabmayr. 'Montague's problem'. In: *Journal of Philosophy* (forthcoming).

[31] B. Grabmayr, V. Halbach and L. Ye. 'Varieties of self-reference in metamathematics'. In: *Journal of Philosophical Logic* 52.4 (2023), pp. 1005–1052.

[32] B. Grabmayr and A. Visser. 'Self-reference upfront: A study of self-referential Gödel numberings'. In: *Review of Symbolic Logic* 16.2 (2023), pp. 385–424.

[33] A. Grzegorczyk. 'Undecidability without arithmetization'. In: *Studia Logica* 79.2 (2005), pp. 163–230.

[34] P. Hájek and P. Pudlák. *Metamathematics of First-Order Arithmetic*. Cambridge University Press, 2016.

[35] V. Halbach. *Axiomatic Theories of Truth*. 2nd ed. Cambridge University Press, 2014.

[36] V. Halbach and G. Leigh. *The Road to Paradox: A Guide to Syntax, Truth, and Modality*. Cambridge University Press, 2024.

[37] V. Halbach and A. Visser. 'Self-reference in arithmetic I'. In: *The Review of Symbolic Logic* 7.4 (2014), pp. 671–691.

[38] V. Halbach and A. Visser. 'Self-reference in arithmetic II'. In: *The Review of Symbolic Logic* 7.4 (2014), pp. 692–712.

[39] R. K. Heck. 'Self-reference and the languages of arithmetic'. In: *Philosophia Mathematica* 15.1 (2007), pp. 1–29.

[40] R. K. Heck. 'Self-reference: The meta-mathematics of the liar paradox'. In: *The liar paradox*. Ed. by L. Rossi. Cambridge University Press, 2026.

[41] D. Hilbert and P. Bernays. *Grundlagen der Mathematik II*. 2nd ed. Grundlehren der mathematischen Wissenschaften. Springer, 1970.

[42] L. Humberstone. *The Connectives*. MIT Press, 2011.

[43] E. Jeřábek. *Did Edward Nelson accept the incompleteness theorems?* MathOverflow. Available from: `https://mathoverflow.net/q/378777` (last accessed on 19.01.2025). 2020.

[44] S. C. Kleene. 'Review of *Grundlagen der Mathematik*, by D. Hilbert P. Bernays'. In: *Journal of Symbolic Logic* 5.1 (1940), pp. 16–20.

[45] S. C. Kleene. *Introduction to Metamathematics*. 1st ed. North Holland, 1952.

[46] S. Kripke. 'Outline of a theory of truth'. In: *Journal of Philosophy* 72.19 (1975), pp. 690–716.

[47] S. Kripke. *Gödel's theorem and direct self-reference*. 2021. arXiv: 2010.11979 [math.LO].

[48] T. Kurahashi and A. Visser. *Certified Σ_1-sentences*. 2024. arXiv: 2306.13049 [math.LO].

[49] P. Lindström. *Aspects of Incompleteness*. Cambridge University Press, 2016.

[50] Y. Manin. *A Course in Mathematical Logic for Mathematicians*. 2nd Edition. Springer, 2009.

[51] A. A. Markov Jr. *The Theory of Algorithms*. Vol. 42. Proceedings of the Steklov Institute of Mathematics. Springer, 1954.

[52] A. Montalbán. *Computable Structure Theory: Within the Arithmetic*. Perspectives in Logic. Cambridge University Press, 2021.

[53] J. Murwanashyaka. 'Weak essentially undecidable theories of concatenation, part II'. In: *Archive for Mathematical Logic* 63.3–4 (2023), pp. 353–390.

[54] E. Nelson. *Predicative Arithmetic*. Princeton University Press, 1986.

[55] K.-G. Niebergall. 'On the limits of Gödel's second incompleteness theorem'. In: *Argument und Analyse*. Ed. by C. U. Moulines and K.-G. Niebergall. Leiden, The Netherlands: Brill Mentis, 2002, pp. 109–136.

[56] K.-G. Niebergall. '"Natural" representations and extensions of Gödel's second theorem'. In: *Logic Colloquium '01: Lecture Notes In Logic, 20*. Ed. by M. Baaz, S. Friedman and J. Krajícek. 2005, pp. 350–368.

[57] K.-G. Niebergall. 'Intensionality in philosophy and metamathematics'. In: *Intensionality*. Ed. by R. Kahle. Lecture Notes in Logic, 22. CRC Press, 2005, pp. 123–159.

[58] A. Oberschelp. 'Order sorted predicate logic'. In: *Sorts and Types in Artificial Intelligence*. Springer, 1990, pp. 7–17.

[59] P. Odifreddi. *Classical Recursion Theory: The Theory of Functions and Sets of Natural Numbers*. Elsevier Science Pub. Co., 1989.

[60] F. Pakhomov. *How to escape Tennenbaum's theorem*. 2022. arXiv: 2209.00967 [math.LO].

[61] A. C. Paseau. 'What are the formulas of a logic?' In: *Erkenntnis* 91.3 (2026), pp. 1371–1386.

[62] L. Picollo and D. Waxman. 'Arithmetical pluralism and the objectivity of syntax'. In: *Noûs* 59.2 (2025), pp. 372–391.

[63] P. Pudlák. 'On the lengths of proofs of consistency'. In: *Collegium Logicum*. Springer, 1996, pp. 65–86.

[64] W. V. O. Quine. 'Concatenation as a basis for arithmetic'. In: *The Journal of Symbolic Logic* 11.4 (1946), pp. 105–114.

[65] W. V. O. Quine. *Mathematical Logic*. Revised Edition. Harvard University Press, 1981.

[66] M. Rathjen. 'The art of ordinal analysis'. In: *Proceedings of the International Congress of Mathematicians Madrid, August 22–30, 2006*. Vol. II. EMS Press, 2007, pp. 45–69.

[67] W. Rautenberg. *A Concise Introduction to Mathematical Logic*. 3rd ed. Springer, 2010.

[68] B. Rosser. 'Extensions of some theorems of Gödel and Church'. In: *The Journal of Symbolic Logic* 1.3 (1936), pp. 87–91.

[69] T. Schindler. 'Type-Free Truth'. PhD thesis. Ludwig Maximilians Universität München, 2015.

[70] P. Smith. *An Introduction to Gödel's Theorems*. Logic Matters, 2020.

[71] R. M. Smullyan. *Theory of Formal Systems*. Princeton University Press, 1961.

[72] S. Świerczkowski. 'Finite sets and Gödel's incompleteness theorems'. In: *Dissertationes Mathematicae* 422 (2003), pp. 1–58.

[73] A. Tarski. 'The concept of truth in formalized language'. In: *Logic, Semantics, Metamathematics*. Ed. by J. Corcoran and J. H. Woodger. 2nd ed. Hackett Publishing, 1983.

[74] D. Van Dalen. *Logic and Structure*. 5th ed. Springer, 2012.

[75] A. Visser. 'Semantics and the liar paradox'. In: *Handbook of Philosophical Logic*. Ed. by D. Gabbay and F. Guenthner. 2nd ed. Vol. 11. Springer, 2004, pp. 149–240.

[76] A. Visser. 'Can we make the second incompleteness theorem coordinate free?' In: *Journal of Logic and Computation* 21.4 (2011), pp. 543–560.

[77] A. Visser. 'The second incompleteness theorem: Reflections and ruminations'. In: *Gödel's Disjunction: The Scope and Limits of Mathematical Knowledge*. Ed. by L. Horsten and P. Welch. Oxford University Press, 2016, pp. 69–90.

[78] A. Visser. *The art of arithmetisation*. Manuscript. 2025.

[79] A. Visser. *From numbers to ur-strings*. 2025. arXiv: 2411.15873 [math.LO].

[80] A. Visser. *On a theorem by Bezboruah & Shepherdson*. 2026. arXiv: 2603.06068 [math.LO].

[81] A. J. Wilkie and J. B. Paris. 'On the scheme of induction for bounded arithmetic formulas'. In: *Annals of Pure and Applied Logic* 35.C (1987), pp. 261–302.