The incompleteness of an incompleteness argument

Joachim Derichs

Abstract

Gödel’s argument for the First Incompleteness Theorem is, structurally, a proof by contradiction. This article intends to reframe the argument by, first, isolating an additional assumption the argument relies on, and then, second, arguing that the contradiction that emerges at the end should be redirected to refute this initial assumption rather than the completeness of number theory.
Section 1 – Introduction

It’s the signature move of Gödel’s argument to establish, based on the eponymous Gödel numbering, an ingenious mapping into number theory. But a mapping always has two sides, a preimage and an image, and we need to be as clear about the first as we are about the second. So what, exactly, is being mapped into number theory?

On one side the mapping is already wonderfully clear, as it lands in first-order number theory. On the side of the preimage, much murkier. What we typically find there is a semi-formalism, assembled from different elements that are not well integrated: usually some recursion theory, often some naïve string set theory, always some highly uncommon notations to deal with encodings. All in all, a setup that seems dangerously complicated, and confusing.

This article contends that we can do better, and wants to show it by restating the argument in a form that is more formally rigorous, and much more transparent.

Within the wider space of incompleteness arguments, Gödel’s version is already the strongest because, unlike the others, it mostly involves first-order logic, which is the best-tested and most rigorous of all the familiar systems of notation. In order to render the argument even stronger, our side of the aisle intends to go further down this road, by recasting it in a way that removes any remaining features not compatible with the predicate calculus. The end result will be a form of presentation that we are confident achieves much greater transparency as it uses only the most boringly safe elements of predicate logic.

What follows is a list of features typically found in other presentations that our side wants to avoid since we believe that resorting to any of these moves would weaken the argument, and distract from the aim of finding its strongest possible form. Rather than only Gödel’s original treatment the list addresses
the general area of incompleteness arguments. It is meant to give prominent examples, not to be exhaustive:

- **No semantics, i.e. no informal or semiformal acts of interpretation.** Whatever in the course of the argument may amount to an act of interpretation – typically, interpreting numeric strings as representing other strings – is to be realised purely formally, as a derivation inside a theory or a mapping between theories.

- **No mapping across formalisms.** While equivalence theorems that translate different formalisms into each other have their place, they should stay separate from incompleteness arguments. For simplicity’s sake, incompleteness arguments are best kept internal to one type of formalism only. Given that the relevant formalism for Gödel’s version of the argument is predicate logic – after all, the Incompleteness Theorem is explicitly stated as being about a first-order theory – it is most transparent for the preimage also to be presented as a theory. Cross-typed mappings offer too many wrinkles for assumptions to hide in, and are worth avoiding whenever possible.

- **No use of recursion theory.** The less important reason to avoid recursion theory and similar formalisms is their association with imagery of open-ended or ongoing processes (‘recursive enumeration’) that can be fatally ambiguous between the process extending as finitely far as one likes, or actually infinitely far. Predicates, which present as closed, completed infinities from the start, are much less distracting in this regard. The more important reason is to avoid cross-typing. Whereas recursive definition ranging over the natural numbers is straightforwardly interchangeable with predicate logic, recursion ranging over strings is murkier, with its reliance on concatenation embedded so deeply into the structure of the formalism that it becomes invisible. Nevertheless, even over strings, the reasonable expectation is for definability by recursion, with at most minor adjustments, to remain equivalent to definability as a predicate. Since equivalence frees us to choose either, we are better off choosing predicate logic.
• **No talk of (infinite) sets of strings.** The first reason is again to avoid cross-typing. To match the target, our side prefers predicates at the origin to sets. Avoiding sets has the added advantage of not having to make automatic assumptions that for arbitrary string predicates \( P \) an extension \( \{ x \mid P(x) \} \) exists. Everybody knows from history how dangerous this can be, and we would not want to prejudge whether string predicates are like the predicates of number theory, where assuming the existence of an extension is harmless, or like the predicates of set theory.

Writing down an expression that purports to define a set of strings is not enough to make it safe to assume the existence of the intended set. We will therefore be avoiding appeals to sets of strings.

• **As a consequence, also no talk of semiformal structures** whose domains are meant to be sets of strings.

• **No references to “truth”** which inevitably introduces incompletely articulated intuitive notions. Our side prefers fully articulated formal systems.

• **No talk of “numbers”,** except as a sloppy shorthand for syntactically grounded facts. Also, no “mapping by enumeration”, no object-level mappings of strings to “numbers” by “just counting”.

The Cantorian master concept, countable, is a misnomer. In terms of real effective ability, already the natural numbers are uncountable. We do not have the ability, are not (count)able, to set up individual 1-1 correspondences between two infinite rows of objects for the simple reason that we lack the ability to handle infinitely many objects simultaneously.

A never-finishing process of enumeration that at any point in time will have reached only finitely far is incapable of producing more than initial fragments of a mapping. Counting is not enough to establish even countably infinite mappings.
• **No unusual notations that occur only in metamathematics.** In particular, no quotation devices like corners, overlines, etc.

When one writes something like “P(\overline{ab})”, where P is a predicate in number theory, a and b are strings from a different theory, and the overline is meant to represent a gödel-numbering function that is part of neither theory, it quickly becomes hard to keep track of which string ‘is’ or ‘stands for’ which other string, and in which theory the argument is currently being conducted. The notation creates a hybrid sentence with disparate parts interlaced to form something that can seem to belong to no theory and several theories all at once. The general effect is one of blurring levels and contexts.

As if that were not enough the ‘ab’ notation introduces the function of concatenation by juxtaposition without further justification. And by treating a and b as variables one would in effect be able to quantify inside terms, which is highly irregular. These and other shenanigans are made impossible by staying within the notational means of predicate logic.

• **Under the same heading, no sleek notations of convenience.** Concatenation will be written throughout as a standard binary function \(c(x,y) = z\) (or equivalently, a standard ternary predicate \(C(x,y,z)\)). This is almost never done as it is painfully inconvenient. Inelegance, however, can be a virtue if it helps us stay away as far as possible from inadvertently introducing concatenation as a function given by juxtaposition. Clunky is good because it rattles whenever one would be tempted to write: *For all \(x,y, \ldots\) xy ...*

• **Finally, no infinitary constructions.** In particular, infinitary constructions arising from indexes created by enumeration. Our side does not want to make automatic assumptions that constructions based on countable enumeration succeed. Infinitary constructions like these do not prove existence, they merely insinuate that a postulate supporting the construction might make sense. Without such a postulate all the construction will have succeeded in producing is a new name claiming not to be empty.

In concrete terms, this means that we want to avoid endorsing the “\(P_i(x)\)” notation, where i stands for an infinite index of formulas. It does not form part of the standard repertoire of
predicate logic, and, due to the diagonal constructions it enables, is not conservative with respect to it.

Writing down the definition of a predicate in terms of an enumeration \( P \), is not enough to demonstrate its existence.

This concludes the list of constraints we propose to adopt voluntarily. Their common theme is a call for greater formal clarity, a reduction in the complexity of the mapping, and more caution in granting assumptions. All of which might be summarised as a request for more effective formalisation. Within the terms of reference for mathematics and logic there is no basis for refusing a request like this. A formal science is a formal science.

Once done clarifying, we shall have on the side of the preimage something that is as rigorous and formal as number theory standing on the image side. The mapping will be cleanly typed, moving from one theory to another. We shall be mapping predicates strings of a theory to their peers, the predicates of number theory.

With a theory on both sides, we shall proceed to set up clean syntactic mappings between them:

**Definition 1.1:** A modelling transformation is a function \( m \) from the strings of a theory \( T \) to the strings of a theory \( M \) such that

1. \( m \) respects the roles of strings, mapping

   terms of \( T \) to terms of \( M \),
   predicates of \( T \) to predicates (of the same arity) of \( M \),
   sentences of \( T \) to sentences of \( M \).

2. \( m \) commutes with logical connectors, quantification, and substitution, i.e.
for all sentences $\varphi$ \hspace{1cm} $m(\neg \varphi) = \neg m(\varphi)$

for all sentences $\varphi$, $\psi$ \hspace{1cm} $m(\varphi \lor \psi) = m(\varphi) \lor m(\psi)$

for all predicates $P$, $(x$ free) \hspace{1cm} $m(\exists x P(x)) = \exists x m(P(x))$

for all predicates $P$, terms $t$ \hspace{1cm} $m(\text{sub}(P,t)) = \text{sub}(m(P),m(t))$

3. $m$ preserves provability, if $T \vdash \varphi$ then $M \vdash m(\varphi)$

With an ironic nod to the semantic usage, we will call a consistent and complete $M$ that receives a $T$ under such a transformation a model for $T$.

The clarified preimage for Gödel’s mapping should not only be a theory, but a theory of a peculiar kind. It needs to be literal about the strings of number theory in this technically precise sense:

**Definition 1.2**: A theory $M$ is said to be meta to a theory $T$ if all strings belonging to $T$ are well-formed as primitive terms in $M$.

In a theory meta to number theory, the strings of number theory figure as closed terms. String predicates – e.g. the predicate of being a well-formed sentence – must first exist, unencoded, in a string theory before they can be projected, under a code, into number theory. The theory of number-theoretic strings is where string predicates live, in number theory they are only guests.

Interpretation, whether formal or informal, semantic or syntactic, is necessarily a binary relation, interpreting an $A$ as a $B$, the image as the preimage. Before we can even think of interpreting arithmetic relations as string relations, we must secure possession of the string relations first.

So before Gödel’s encoding argument can even begin, we need the following: At the origin of the mapping, a theory that is meta to number theory, and thus allows string predicates over the strings of
number theory to be defined. The most important relation for the argument being provability, all the other relations only serve to support it.

Let $\text{BASE}$ stand for any first-order successor-based theory that contains Robinson’s $Q$. $\text{BASE}$ could be number theory, or any suitably weaker subtheory. The exact nature of $\text{BASE}$ is irrelevant.

What we need, then, for Gödel’s argument to take off – or at least to take off in a clean, purely syntactic framework – is, minimally, this:

**Definition 1.3:** $\text{META}$ is a first-order theory that is meta to $\text{BASE}$ and contains a predicate $\text{PROV}(x)$ that directly defines provability for $\text{BASE}$.

A distinct concept calls for distinct terminology. We say *directly* defines in order to emphasise again that the strings of the base theory appear in $\text{META}$ nakedly, untouched by any encoding. Strings come just as they are. Literally literal.

For now, to maintain the symmetry between preimage and image, we shall limit the search for $\text{META}$ to first-order theories. We promise to circle back to this limitation before the end.

The plan for the remainder of the article is as follows:

(Section 2) Examine $\text{META}$, the distilled preimage

(Section 3) Examine concatenation, the crucial function in $\text{META}$

(Section 4) Build modelling transformations between $\text{META}$ and $\text{BASE}$

(Section 5) Recover a form of Gödel’s Theorem

(Section 6) Address the consequences of the restated Theorem
Section 2 – Building META

This section examines the anatomy of META, the theory that is meant to constitute the preimage. For the avoidance of any misunderstanding, META is not and does not pretend to be ‘the meta theory’ in which the argument is conducted. META is something much simpler: a fully formal string theory serving as the preimage for a mapping that the largely informal meta theory is used to describe. The language of META is first-order logic; the language of the background ‘theory’ is more or less lightly formalised English.

Definition 2.1: Let \( C \) be the first-order language (with identity) consisting of a single binary function \( c(x,y) \), and for constants, strings over a given finite alphabet \( \Sigma \).

The next definition refers, scrupulously, to ‘predicate expressions’ because ‘predicate’ can shade into assuming the existence of an extension, which our side wants to avoid prejudging. The choice of the word ‘class’ is similarly motivated.

Definition 2.2: Let \( \Delta C \) be the class of predicate expressions in \( C \) that use at most bounded quantification.

The natural model for the ordering relation in the quantifiers is string length.

Proposition 2.3: For META to exist, it is sufficient that there is a first-order theory \( T \) that

1. contains concatenation (and string identity)
2. gives \(< a finite definition: For any string \( s \) over \( \Sigma \),

\[
T \vdash x < s \iff (x = s_1 \lor x = s_2 \lor \ldots \lor x = s_m),
\]

where the \( s_i \) are all strings over \( \Sigma \) shorter than \( s \)
3. decides all predicates \( P \) in \( \Delta C \), so that for all strings \( s \) over \( \Sigma \),

\[
T \vdash P(s) \text{ or } T \vdash \neg P(s)
\]

\(^1\)No slip-up. Once membership is known to be decidable, it is legitimate to speak of predicates and their set extensions.
Condition (1) requires the meaning of predicate expressions to be anchored by the c-function. What precisely is entailed by capturing concatenation will be the topic of the next section. For the duration of this section we will follow the example of all the textbook arguments for Gödel’s Theorem and take native concatenation for granted. We will assume that the c-function behaves as expected, that it concatenates unencoded literals in just the way a naïve user would expect it to.

With the native c-function a given, the remaining content of the proposition covers well-trodden ground. We need only sketch a proof.

The intermediate step required is an expression PROOF\((x,y)\) that formalises the string relation ‘\(x\) is a proof sequence for the sentence \(y\)’. As first shown by Gödel himself, there exists a predicate PROOF*\((n,m)\) in number theory that, as far as concatenation extends, could serve as a homomorphic image for PROOF. Moreover, PROOF*\((n,m)\) would be primitive recursive, and readily decidable. So we could simply argue that if PROOF is definable indirectly as an arithmetic relation, then it must be even more easily definable directly, without the extra complication of an encoding.

If we were, somewhat redundantly, to start building PROOF anyway, three types of relations are necessary: String identity, pattern search and recursive construction.

Following the textbooks, we shall assume string identity alongside with concatenation.

Pattern search involves checking whether a string is composed of various independently defined components. A good example would be the substitution function. Pattern search is clearly bounded in the length of the inputs, and available in \(\Delta C\) as long as the definitions of the components are available. A typical example of a predicate required for resolving pattern searches would be \(\text{CONTAINS}(x,y)\), which can be seen to be in \(\Delta C\).

For the third type of relation, the recursive construction of formulas, we require:
**Definition [2.4]:** A string predicate $P$ is said to be definable by primitive recursion if can be
given by an axiom of this form:

$$P(x) \iff A(x) \lor ( \exists x_i < x \ B(x, x_i) \land P(x_1) \land \ldots \land P(x_n) ),$$

where $A(x)$ is a unary in $\Delta C$, and $B(x, x_i)$ is $n+1$-ary in $\Delta C$.

$A(x)$ is the seed clause, equivalent to the ground level of induction. $B$ is the construction clause that
describes how to get from one stage of iterative construction to the next.

**Lemma [2.5]:** Theories $T$, as defined in 2.3, decide primitive recursive string predicates
closed by a string $s$.

The finite definability of $<$ reduces primitive recursion to $\Delta C$. This would only be surprising if it were
not the case. Another sketch will do for a proof.

For illustration, we will use the example of NUMERAL. As a concession to readability it is supposed
that the base theory writes $0'' \cdots '''$ for successor, and not, as it should, $s(s( \ldots s(0) \ldots ))$.

With $A(x) := x = 0$, $B(x, x_1) := c(x_1, ^{*}) = x$, we have $\text{NUMERAL}(x) :=$

$$x = 0 \lor ( \exists x_i < x \ c(x_i, ^{*}) = x \land \text{NUMERAL}(x_i) )$$

By definition, $\text{NUMERAL}(s)$ unspools into:

$$s = 0 \lor ( \exists x_i < s \ c(x_i, ^{*}) = s \land ( x_1 = 0 \lor ( \exists x_2 < x_1 \ c(x_2, ^{*}) = x_1 \land \text{NUMERAL}(x_2) ) ))$$

Let’s say that we start with the list $x_1 < s \iff ( x_1 = s_1 \lor x_1 = s_2 \lor \ldots \lor x_1 = s_n )$. Since the inequality in
$\exists x_i < x_i$ is strict, with each unspooling the longest strings on the list get cut, so that the list shortens
progressively, and will eventually be empty. Once $\neg \exists x_i < x_i$ is reached we can strike the recursive
predicate letter. Of the last unspooling only the seed clause remains. We are left with a pure $\Delta C$ sentence, which by assumption is decidable.

For NUMERAL especially there does exist a much simpler, closed form definition, based on describing NUMERAL as ‘begins with 0, otherwise consists only of’. As a matter of fact, BEGINS_WITH($x,y$) and CONSISTS_ONLY_OF($x,y$) are both in $\Delta C$.

In the general case, however, we cannot dispense with definition by primitive recursion as closed form definitions directly from $\Delta C$ are more elusive.

With clauses A and B that are a little more complicated, but still either contained in $\Delta C$, or in turn definable by primitive recursion, we can proceed to define concepts like VARIABLE, TERM, PREDICATE, SENTENCE. The relation of being an immediate consequence requires another (elaborate but manageable) pattern search. Putting it all together, PROOF is built.

So assuming only that the meaning of predicate expressions from $C$ is anchored by concatenation, PROV($x$) := $\exists y$ PROOF($y,x$) will directly define provability.

To be clear, definability as used in the proposition is not intended to be definability according to this conventional definition:

**Definition 2.6:** A predicate $P$ from an arithmetic theory $AR$ defines a preformal string property $PROP$, if for all $a \in \Sigma^*$, $g$ an encoding function,

$\models PROP$ is true for $a \Rightarrow AR \vdash P(g(a))$

$\models PROP$ is false for $a \Rightarrow AR \vdash \neg P(g(a))$

First of all, our side does not want to encode. Instead, as already practised above, META means to formalise facts about naked literals. Hence our $g$ is absent / the identical function. (To avoid confusion,
we might choose to change the formatting for strings from the base theory, to **bold** or in some other uniform way, but that’s it. No character replacement, and certainly no numeric encoding.)

Second, the way in which the conventional definition includes unexamined properties, along with “structures” that are semiformal at best, is profoundly unsatisfactory. It would be irresponsible to put any trust into unformalised ideas about string properties unless and until they have been refined into the predicates of an actual theory. The conventional definition has it backwards. One should not be referring theories to informal notions as the standard from which to take directions. Our side wants theories to set the standard, and eliminate informal notions like **PROP**, as well as the accompanying truth talk.

Concatenation-based string theories represent much the clearest expression, the least slippery grip we have on the concept of a string property. String properties are so obviously at home there that it would in many ways be better to turn the definition talk around and say that the existence of a predicate in a string theory is what makes a well-defined string predicate. The predicates of string theories would then “define” by definition.

A concatenation-based theory like **META** does not have to prove that it defines string properties; canonically defining these properties is what it can be assumed to do by definition, or quasi-axiomatic convention.

With the direction of definition reversed, the arrow of implication reverses, too. Before we had, for sentences $\varphi$ of the base theory,

\[
\text{BASE} \vdash \text{PROV}^*(g(\varphi)) \iff \models \varphi \text{ is provable in BASE}
\]

\[
\text{BASE} \not\vdash \text{PROV}^*(g(\varphi)) \iff \models \varphi \text{ is not provable in BASE}
\]

Now **PROV** directly defines provability, so that

\[
\text{META} \vdash \text{PROV}(\varphi) \Rightarrow \text{BASE} \vdash \varphi
\]

\[
\text{META} \vdash \neg\text{PROV}(\varphi) \Rightarrow \text{BASE} \not\vdash \varphi
\]
No more g, and everything underpinned by thoroughly formal definitions. There are two clauses because if we can assume that c works at all, then it will also work for the negated predicate.

Later on, when ready to start mapping to BASE, we will be redefining \text{represents}_{\text{mult môni}}\ as:

\[ \text{META} \vdash \text{PROV}(\varphi) \Rightarrow \text{BASE} \vdash \text{PROV}^*(g(\varphi)) \]

So eventually, we will get BASE \vdash \text{PROV}^*(g(\varphi)) back, but will have eliminated all informal notions.

**Definition 2.7**: A first-order theory is said to be strictly axiomatisable if it is axiomatised by a finite number of sentences or schemes such that any scheme can be summarised by a single sentence in higher-order logic.

The single sentence from higher-order logic provides a pattern against which axioms can be checked. Clearly, Peano arithmetic is strictly axiomatisable – its scheme is summarised in second-order logic. For naturally occurring theories likely to become relevant to the argument, including of course Q, the definitions of ‘strictly axiomatisable’ and ‘recursively axiomatisable’ are coextensive. The adjusted definition is intended only to exclude artificial axiomatisations by pure enumeration or infinite indexing, which recursive axiomatisability might allow through.

**Corollary 2.8**: For any (consistent) strictly axiomatisable theory T, if META\textsubscript{T} is complete, then

\[ \text{META}\textsubscript{T} \vdash \text{PROV}_T(\varphi) \iff T \vdash \varphi \]

**Definition 2.9**: A theory that contains concatenation is said to *directly express* any string relation that it decides.

META, if it were complete, would directly express provability for BASE.
Note how from these two definitions

\[
\text{META} \vdash \text{PROV}(\varphi) \Rightarrow \text{BASE} \vdash \varphi \quad \text{(PROV directly defines)}
\]

\[
\text{META} \vdash \text{PROV}(\varphi) \Rightarrow \text{BASE} \vdash \text{PROV}^*(g(\varphi)) \quad \text{(PROV is represented new in BASE)}
\]

by cutting out META in the middle (essentially by assuming its completeness), one could obtain

\[
\text{BASE} \vdash \text{PROV}^*(g(\varphi)) \iff \text{BASE} \vdash \varphi
\]

which, by adding completeness for BASE, immediately yields

\[
\text{BASE} \vdash \text{PROV}^*(g(\varphi)) \leftrightarrow \text{BASE} \vdash \varphi
\]

This, in a nutshell, is the traditional argument: A mapping of BASE into itself, achieved by making inarticulate assumptions about a less than formal META. The way consistency and completeness assumptions for META entered into the argument was through extensional semantics, by taking for granted that the arbitrary sets of strings or informal string properties used as inputs to the mapping were “real” to begin with. In order to be available for mapping, the inputs must at least be possible and definite: Possible was presumed through the existence of an (intuited) model, which can be thought of as the semantic form of consistency; definiteness was given by excluded middle, which is the semantic form of completeness.

Section 3 – On concatenation

In the previous section we have shown that the existence of a meta string theory that directly defines provability reduces to the existence of a theory that contains concatenation.

Speaking of concatenation, we have to be clear which form of concatenation we mean. There are two related but essentially different forms of concatenation. Both have their origin in the naïve concept of concatenation, but they do address different aspects.

The first form of concatenation is implicit. It is well-known and well-understood. Concatenation is formalised as an implicit binary function not unlike the Peano successor function. A typical sentence of
such a theory would be the axiom of associativity: $c(x,c(y,z)) = c(c(x,y),z)$. An example of a weak theory of implicit concatenation, with most of the axioms going back to Tarski, is contained in Grzegorczyk [2005]. Stronger versions add an induction scheme, where the successor equivalent of a string is the string extended by one letter. The strongest version is defined, just as it is for arithmetic, by its second-order axiom of induction.

Implicit concatenation theories are characterised by the fact that they allow for only finitely many constants, one for each letter of their alphabet. Let’s say that there are only two letters, a and b. The string “ab” would then not be a term of the theory. Implicit concatenation theories are therefore unable to make, let alone decide, statements of the form “$c(a,b) = ab$”.

One could paraphrase this by saying that implicit concatenation theories express naïve concatenation only up to typographical permutation. Implicit concatenation only captures structural properties that are independent of an instantiation in concrete strings. It is not able to prove or refute that the concatenation of two given concrete strings is a third concrete string.

The second form of concatenation is much less well understood. We will call it extensional concatenation. It can be thought of as implicit concatenation made concrete, or fully interpreted. People, and mathematicians especially, use it fluently, and daily; but very rarely do they stop to examine it.

Suppose we start again with only two letters, and want a language that will prove all and only the ‘true’ atomic concatenation statements (as naïvely understood) about these letters. For letters a and b that would mean that the language should prove e.g. $c(a,b) = ab$ and $c(a,ab) = aab$, and disprove $c(a,a) = a$.

It should have become clear by now that extensional concatenation is a different and distinct concept from what is captured in theories of implicit concatenation. Extensional concatenation fixes the meaning of strings that implicit concatenation leaves to vary.
The point we have been building up to is that Gödel-style meta arguments absolutely require extensional concatenation. When the argument turns on a concrete string encoding another concrete string, representation up to typographical permutation is obviously not good enough.

Hence this first rough statement of an assumption:

**Gödel’s Assumption (rough) [3.1]:** Extensional concatenation is a well-defined function, and primitive recursive.

The standard argument assumes primitive recursiveness for plenty of string relations. PROOF and all the other relations involved in its construction can only be primitive recursive if native concatenation, which lies at the base of their definition, is primitive recursive, too.

Now as hinted in the introduction our side wants we want to avoid talk of recursion because recursive definitions for native string properties are almost inevitably circular, assuming concatenation by juxtaposition on the right-hand side\(^2\).

There is no way of avoiding the use of extensional concatenation in meta arguments, as there is no way of talking about the strings of an underlying theory without it. Given the use, we would want to know that the naïve concept of extensional concatenation can be trusted. The prime means of demonstrating a clear understanding, and inspire trust, would be to present a consistent formalisation.

Due to reservations about recursion theory, and more important, to fit underneath META, we expect extensional concatenation to be formalisable, and decidable, in predicate logic instead:

**Gödel’s Assumption (a little less rough) [3.2]:** Concatenation is formalisable. In other words, there exists a theory CONCAT containing a binary function c that correctly expresses

---

\(^2\)Recursive definitions of native concatenation typically look like this: \(c(x,y) = “f(xy)”\), where \(f\) relies on concatenation by juxtaposition (Note the absence of a comma of the right-hand side).
extensional concatenation for atomic sentences over a finite alphabet \( \Sigma \), so that for all \( s,t,u \in \Sigma^* \),

\[
\text{CONCAT} \vdash c(s,t) = u \iff u \text{ is the concatenation of } s \text{ and } t.
\]

As a ubiquitous meta function, it is extremely easy for informal uses of extensional concatenation to intrude into an argument. The best protection against such intrusions is what we are already planning to do: Restate the preimage as a theory, and the entire interpretation process as a mapping from theory to theory. Reconstruct the argument on a purely syntactic basis, as an interpretation free of semantics. To that end, we will now try to formalise extensional concatenation. This will feel awkward as it is not something the predicate calculus is designed to do. The very awkwardness will show that the protections are working.

Gödel’s Assumption, as stated, is infuriatingly vague. ‘Correctly expresses’ is just a pretentious way of saying ‘behaves as one would naïvely expect concatenation to behave’. The phrase ‘is the concatenation of’ assumes command of the very thing we want do more about than just assume. But the task we face is tricky: How does one articulate the inarticulate? How do we even know that there is something there to be expressed?

In order to get a firmer grip let’s try

**Definition 3.3:** \( \text{CONCAT}_n \) is a consistent, complete and strictly axiomatisable first-order theory that directly defines concatenation for strings up to length \( n \) over a finite alphabet \( \Sigma \).

Writing out examples of \( \text{CONCAT}_n \) is straightforward if tedious. To illustrate, we present \( \text{CONCAT}_2 \), a theory consisting of six constants, a single predicate letter \( C \), and the following axioms:

(As the assumption that concatenation should be a total function has become problematic, we switch to predicate-style notation to accommodate partial functions.)
Write \[ U_1(x) := x = 0 \lor x = ' \]
\[ U_2(x) := U_1(x) \lor x = '00' \lor x = '0' \lor x = '' \]

(A1) \( \forall x U_2(x) \)
There is an explicit, pre-declared finite domain. This in itself is nothing unusual, similar limitations are implicit in the workings of any real-world computer.

(A2) \( \forall x,y \left( \neg U_1(x) \lor \neg U_1(y) \right) \rightarrow \neg \exists z \ C(x,y,z) \)
Concatenation in CONCAT space is only a partial function, not defined beyond a given upper limit (again analogous to real-world computing). Limit setting could be made more sophisticated by appealing to string length rather than explicit listings.

(A3) \( C(0,0',0') \)
Within the strictures of the predicate calculus it is surprisingly difficult to give correct extensional values in any other way than by listing all instances individually.

(A4) \( C(0,0,00) \)

(A5) \( C(0',0,0') \)

(A6) \( C(0'',0'',0'') \)

(A7) \( \neg(0 = '0) \)
Necessary to prevent trivialising models where letters are equated to each other. Collapsed concatenation with all letter set to be equal is (nearly) successor.

The theory CONCAT space is evidently decidable. The first axiom, by limiting the domain to a finite number of explicit strings, ensures that quantifiers can be eliminated in favour of atomic statements, in a finite
number of steps. Axioms (A2) – (A6) together make sure that the list of atomic statements in $C$ to be checked against is exhaustive, and in turn quickly decided. Atomic statements in $=$ are resolvable into finite combinations of atomic statements in $C$ and equalities over letters only. Last, atomic (in)equalities over letters are decided with the help of (A7).

It seems reasonable to grant the intelligibility of the idea of constructing theories $\text{CONCAT}_n$ on roughly this pattern even for $n$ larger than the physical capacity of any human or computer. As categorical theories that directly define, the $\text{CONCAT}_n$ perfectly embody the concept they present. In their albeit limited realms, they are concatenation.

There is only one final complication: the self-reflexivity of extensional concatenation. Due to its circular nature, the construction of $\text{CONCAT}_2$ we gave does not, strictly speaking, prove the existence of languages $\text{CONCAT}_m$: The ability to construct $\text{CONCAT}_n$ presupposes string handling abilities tantamount to $\text{CONCAT}_m$ for some $m > n$. It therefore seems preferable to phrase the conclusion that these theories exist not as a proposition, but a postulate. Moreover, to avoid the implication that an infinite index $i: n \to \text{CONCAT}_n$ exists, we are going to phrase the postulate as a scheme with parameter $M$:

**Postulate Scheme [3.4]:** For $n < M$, $\text{CONCAT}_n$ exists.

Let $M$ be a massively large finite number. For $n < M$, we concede that $\text{CONCAT}_n$ exists. The point of $M$ is not make a specific number the limit, but to make it clear that there always is a limit. Every introduction of $\text{CONCAT}_n$ into a derivation comes with a side constraint that $n < M$. Let’s call $M$ the perimeter of concatenation. Inside the perimeter, everything works as (naïvely) expected. Most of the time, work goes on normally, and the perimeter can be ignored. Only when a contradiction arises is the perimeter constraint activated to absorb it. Sometimes, the argument can be repaired by shifting to a larger perimeter, $M+X$. So for example, a bounded diagonal argument that produces an element provably unlike any inside the perimeter can be repaired by shifting to a larger perimeter that includes the new element. At other times, the argument cannot be repaired. Diagonal arguments based on an
infinite index cannot be restated to be compatible with a perimeter constraint. Such arguments can only be made by appealing to a more powerful postulate.

With the theories $\text{CONCAT}_n$ in mind we are now able to clarify

**Definition [3.5]:** A first-order theory, styled $\text{CONCAT}$, is said to contain concatenation if it contains a function $c$ that satisfies the following definition scheme with parameter $M$:

For $n < M$, for all strings $s, t, u$ in $\Sigma^n$

$\text{CONCAT} |- c(s, t) = u$ iff $\text{CONCAT}_n |- C(s, t, u)$

We keep $c$ as a function on the left-hand side in order to ensure continued compatibility, and interchangeability of predicates, with theories of implicit concatenation.

**Gödel’s Assumption (clarified) [3.6]: $\text{CONCAT}$ exists.**

What began as something slippery and semantic – a claim about the legitimacy of using extensional concatenation in meta arguments – has now been turned into something purely syntactic, and thereby, unambiguous. Deciding atomic statements the same way as the theories $\text{CONCAT}_n$ is a pretty minimalist definition of what extensional concatenation should mean. Whatever one takes extensional concatenation to mean, however one tries to pin it down to a precise meaning, it is hard to see how any proposed formalisation of concatenation could be considered successful without meeting at least this lenient condition. As a formal statement of Gödel’s Assumption it seems more than fair.

Note that there is no important difference between primitive recursive and general recursive predicates when it comes to interpretability. Infinite extensional concatenation is already required to interpret primitive recursive predicates. Required, in fact, already for bounded predicates.

Let $P(x) := \forall y \leq x \ Q(y)$ be from $\Delta C$. Evaluating $P(s)$ for some string $s$ requires (at least) $\text{CONCAT}_{|s|}$. Evaluating $P(x)$ for all strings over $\Sigma$ requires greater than finite reach, which is to say: it would require
**CONCAT.** Even expressions from $\Delta C$ could only live up to their intended meaning if they are supported from below by full, unbounded concatenation.

Only finite predicates – those with at most a finite number of either positive or negative instances – can fit inside a perimeter, to be decided inside one **CONCAT**. General primitive recursive predicates cannot.

Thus only finite predicates are interpretable without Gödel’s Assumption. Without **CONCAT**, there would be nothing to interpret arithmetic predicates *as*, even bounded ones.

What is to stop us from disposing of Gödel’s Assumption by simply proving it, delivering a theory that contains concatenation? Can it really be so difficult to define and axiomatise extensional concatenation? Well, it’s not so simple, it turns out.

Concatenation is an unusual function. To appreciate just how unusual, observe that the candidate that first comes to mind when one thinks about compacting the lists of explicit extensional values into a single axiom – “$\forall x,y \, c(x,y) = xy$” – is certainly not well-formed as a first-order formula, or indeed in any conventional logic. Given how it allows quantification to reach inside terms, inside the “$xy$”, in a way that mashes meta and object level, it would not be surprising if the leading candidate for *Axiom of Concatenation*, once plausibly fleshed out into a non-standard theory, would turn out to be inconsistent. Is this the notion that the average logician unconsciously applies in appealing to their intuition of concatenation?

Without the trick of quantifying inside terms, **CONCAT** is in some trouble. The second idea that comes to mind – interpreting $xy$ as an (implicit) functional $x \times y$, or even more explicitly, $c^*(x,y)$ –, also fails. It leads to an immediate regress in the definition: $c(x,y) = c^*(x,y)$ is not helpful. Thanks to the stubborn rigour of the predicate calculus, the informal meta function of concatenation that we must inevitably

---

3 More than a suspicion. Inconsistency for a non-standard formalism built to contain the *Axiom of Concatenation* is indeed provable. Unpublished paper.
make use of to operate any theory is not so easily repurposed for undercover work on the inside of an axiom.

So it seems that in order to meet the demands of expressing the concept of extensional concatenation, **CONCAT** would have to fall back on the unloved, third-best idea of listing explicit values – which is not without complications, either. Without the trick, or an implicit functional, what stands on one side of the equation defining concatenation must be a primitive term: the $0'$ in $c(0,.) = 0'$ must be primitive. (This is perhaps even easier to see in predicate-style notation, $C(0,.,0')$. When functionals have been eliminated altogether, there is no doubt that $0'$ can only be a primitive term.) While manageable over finite domains, and perfectly workable for the definition of the theories $\text{CONCAT}_n$, when **CONCAT** tries to gather all the finite theories together, it has to start dealing with infinitely many primitive terms simultaneously. All of them must be handled, and fixed in their meanings by syntactic means only.

Formally, $0'$ is a constant, a primitive term not analysable by its own language. A single symbol, without interior divisions. At the same time, the definition of constants as strings over $\Sigma$ presupposes a shadow theory (the residual ‘meta theory’) that analyses them as non-primitive functional terms in concatenation. But we cannot forever rely on outsourcing work to the shadow theory, formalisation is not complete without knowing that we retain the capacity to internalise the work done by any actively involved third parties. The challenge for **CONCAT**, in a way, is to try and catch its own shadow. Put less poetically, in the mundane terms of computing: $0'$, as a single, undivided symbol, must have its own entry on the code table. This is a problem, as for all real-world computers the code table is finite.

Is the following operation computable? Challenged by three objects, decide whether one is the concatenation of the other two. The hard part is formatting the problem, recognising the objects as strings. The inputs have to be parsed, someone or something has to break through the opacity of strings presented as a single symbol, and delineate letters. The rest, once parsing is complete, is stupendously easy.
At first, it may be difficult to see what the problem is. The hard part may not seem hard at all. Parsing is something that everyone who has learned to read an alphabetic language does involuntarily, unprompted. One hardly notices the effort, and finds it much more difficult to stop with the parsing than to parse. But stop is what we must do in order to be able to examine the operation, and really understand what is involved in extensional concatenation.

Once letters have been delineated, answering the challenge is as simple as deciding sentences inside CONCAT\textsubscript{n}. Yet parsing is not equivalent to deciding sentences in CONCAT\textsubscript{n}, it is equivalent to setting up the CONCAT\textsubscript{n}.

Deciding sentences in CONCAT\textsubscript{n} is trivial. Setting up any one theory CONCAT\textsubscript{n} is still almost trivial. A general mechanism for setting up all CONCAT\textsubscript{n} is no longer trivial. It would, in fact, require infinitary powers.

The crux is to get from this
\[
\begin{array}{c|c|c|c}
| & 0 & \cdot & | \\
\end{array}
\] (on two consecutive squares)

to this
\[
\begin{array}{c|c|c}
| & 0' & | \\
\end{array}
\] (on one square)

and vice versa.

Evidently, a machine able to effect concatenation and its inverse for all strings over a finite $\Sigma$ would have to be able to accept and process infinitely many distinct string objects. Translated into the language of Turing machines, CONCAT presupposes the ability of accepting infinitely many distinct symbols on a single square of tape. According to the standard definition of computability CONCAT is thereby disqualified as a computable function.

A theorem that takes in an uncomputable function, ready-made, is not going to have any difficulty proving uncomputability: Showing that functions derived from CONCAT are uncomputable, while true, fails to show anything about the computability of unrelated functions.
Though a blow, this is not a knockout yet because under an alternative definition of computability broader than Turing’s, CONCAT might still be found to be a viable function. The decision whether CONCAT could exist as a non-standard function, provably diagonal to axiomatisations in standard predicate logic, is something we will have to return to.

Section 4 – Mappings between META and BASE

We are now able to put together META, the preimage: CONCAT is defined, and we already know that all string relations necessary for PROV are definable once CONCAT is given. Assuming only that CONCAT exists, then so does META.

**Proposition 4.1:** CONCAT, if it exists, is consistent.

Proof: By construction, CONCAT is consistent if the theories CONCAT$_n$ are. Their consistency in turn can be checked by checking a finite number of relevant combinations.

**Proposition 4.2:** CONCAT, if it exists, decides all predicates in $\Delta C$.

Proof: Let $P(x) \coloneqq \forall y \leq x \ Q(y)$ be a predicate from $\Delta C$, and $s$ a string over its alphabet. For some finite $m$, $0 \leq i \leq m$, let $s_i$ stand for the primitive string terms occurring in $P(s)$, $s$ included. Put $n = \max(|s_i|)$, the length of the longest string term from among the inputs and components of $P$. Then CONCAT$_n$ decides $P(s)$. And CONCAT$_n$ has a model in CONCAT, hence a subtheory of CONCAT decides $P(s)$.

For most intents and purposes, CONCAT already is META. It meets two of the three the requirements identified in Proposition 2.3: contains concatenation, and decides $\Delta C$ predicates. The only bit missing is the ordering relation $\leq$ that drives bounded quantification.
To complete one possible definition of META, we require the axiomatisation of an ordering relation that meshes with concatenation in that it contains the partial ordering by string inclusion, i.e. \( c(x,y) = z \rightarrow (x < z \land y < z) \), and is just strong enough to prove the finite definability of \( x \leq s \) (Condition (2) in Proposition 2.3). Let’s call this theory \( O \), for Order, and a family resemblance to \( Q \).

The purpose of \( O \) is to enable a primitive extensional form of induction that allows for primitive recursive predicates to be defined. \( O \), one might say, is poor man’s induction\(^4\).

There are various ways of giving \( O \), none particularly difficult\(^5\). The only thing that matters for our progress is that \( O \) exists, in some form. The exact nature of the theory is, once again, irrelevant. (Admittedly, as a set of axioms, \( O \) might make more sense as an extension package to concatenation than a stand-alone theory.)

One can think of META as either CONCAT \( \cup O \cup \) several axioms of primitive recursive definition (along the lines of 2.4), or in general as any theory that contains concatenation, and decides a predicate PROOF. (CONCAT extended by several axioms of the type (N1) – (N3) would also qualify.)

Having established the preimage, and knowing the image to be a first-order arithmetic theory, we can move on to mappings between them. All the assumptions are in place for us to be able to connect META with BASE by way of two lemmas.

\(^4\)Struggling for parsimony goes unrewarded here, as whether or not induction is introduced, concatenation is always assumed.

\(^5\)Grzegorczyk [2005] derives finite definability from axioms of implicit concatenation. Alternatively, it is also possible to define relations \(< \) and \( = \) so that \( = \) is designed to represent equality of string length, or distance from the origin. The equivalence classes under \( = \) then map to the natural numbers. Concatenation with a letter turns into a simple form of \( + 1 \). In this sense, \( O \) could provide a rudimentary form of arithmetic, not unlike \( Q \).
Embedding Lemma for Meta Theories [4.3]: For any strictly axiomatisable theory $T$, if $T$ is consistent, and $\text{META}$ is consistent and complete, then $\text{META}_T$ contains a model of $T$.

Proof: Let predicates $P(x)$ map to $\text{PROV}(\text{sub}(P,x))$, and terms identically. We show by induction on the composition of formulas that the mapping describes a modelling transformation. The only difficult case is $T \vdash \neg \varphi \Rightarrow \text{META}_T \vdash \neg \text{PROV}(\varphi)$.

Assume $T \vdash \neg \varphi$. By consistency of $T$, $T \not\vdash \varphi$.

From completeness of $\text{META}$, $\neg \text{PROV}$ directly expresses, so $\text{META} \vdash \neg \text{PROV}(\varphi)$.

The axiomatisability of $T$ is required for the definability of $\text{PROV}$ in $\text{META}_T$.

Gödel’s Lemma [4.4]: If $\text{BASE}$ is consistent, complete and strictly axiomatisable, $\text{META}$ consistent, then $\text{BASE}$ contains a model of $\text{META}$.

Proof: Technically speaking, what needs to be shown is that when concatenation-based relations from $\text{META}$ are associated with the right successor-based relations from $\text{BASE}$ by way of a Gödel numbering, then the resulting mapping is a modelling transformation.

This is a restatement of Gödel’s groundwork for the First Incompleteness Theorem, accepting the substance with only minor changes: Instead of trying to map, on the preimage side, from informal properties or recursive theory or (elements of) string sets, we are now mapping from the predicate strings of $\text{META}$. Instead of a largely unformalised meta-relation termed ‘representation’ we now have $m(\cdot)$, a provability-preserving homomorphism between first-order theories.

For all relations $R$ of $\text{META}$ it has to be shown that if $\text{META}$ proves a sentence $\varphi$ in $R$, then $\text{BASE}$ proves $m(\varphi)$. Among the relations, we are really interested only in the first and last, $c$ and $\text{PROV}$, but to reach the latter, all the relations in between have to be retraced.

First to be mapped is concatenation. As far as the $c$-function stretches, it can be shown that the targets envisaged in the standard argument are able to represent it.

Assume for simplicity that the target of $c$ is a function, $m(c) = c^*$. To even stake out the claim of interpretation for $c^*$, we would need:

\[
\text{CONCAT} \vdash c(s,t) = u \Rightarrow \text{BASE} \vdash c^*(m(s),m(t)) = m(u)
\]
while without the Assumption we have only:

For \( n < M \), for all strings \( s, t, u \) in \( \Sigma^n \)

\[
\text{CONCAT}_n \vdash c(s, t) = u \Rightarrow \text{BASE} \vdash c^*(m(s), m(t)) = m(u)
\]

Now it is true that far as finite claims of interpretation can be stated they can then also be shown to obtain. However, mapping atomic claims one by one is still not very promising, so on top of existing, we would want \text{CONCAT} to be axiomatisable. We would like axioms for \( c \), so that we could then prove their translates in \text{BASE}. Alas, we have none.

Failing axiomatisation, the other option is (second-order) meta-induction. For every letter \( a \) in \( \Sigma \), right-concatenation \( c_1 = c(x, a) \) and left-concatenation \( c_2 = c(a, x) \) have to be shown to preserve provability under \( m \), \text{META} \vdash c(s) = u \Rightarrow \text{BASE} \vdash c^*(m(s)) = m(u) \). Induction on the length of \( s \) and \( u \) would succeed, but does – obviously – require the assumption of \text{CONCAT}.

So there is hole in the lemma, and it is filled with assumptions about \text{CONCAT}.

Next, all primitive recursive string relations required for \text{PROOF}, as far as the \( c \)-function will carry them, are shown to be homomorphic to their targets among arithmetic relations. With completeness, the homomorphism would extend to general relations like \text{PROV}.

As before, the axiomatisability of \text{BASE} is required for the definability of \text{PROV} in \text{META}.

Let \( f \) and \( g \) be modelling transformations. Let \( g \) map \text{META} into \text{BASE}, let \( f \) map \text{BASE} into \text{META}, and for sentences \( \varphi \) put \( \varphi^{**} = f(g(\varphi)) \). (Choice of letters is not accidental: Gödel’s numbering \( g \) can be recovered as a mapping between terms induced by our \( g \).)

We now have the ingredients ready for proving that the predicates of \text{META} under \( f \circ g \) have a fixed point. This follows generically from, for instance, category theory. We will nevertheless give the proofs in long form.
**Round Trip Lemma [4.5]**: \( \text{META} \vdash \varphi \leftrightarrow \varphi^{**} \) (provided \( \text{BASE} \) and \( \text{META} \) are both consistent and complete, and \( \text{BASE} \) is strictly axiomatisable).

**Proof:** If both \( \text{BASE} \) and \( \text{META} \) are consistent and complete, and \( \text{BASE} \) axiomatisable, then the Embedding Lemma guarantees \( f \), and Gödel’s Lemma \( g \).

\( \text{META} \vdash \varphi \) implies \( \text{BASE} \vdash g(\varphi) \), which in turn implies \( \text{META} \vdash f(g(\varphi)) \). Therefore, relabelling only, \( \text{META} \vdash \varphi^{**} \).

For the converse suppose \( \text{META} \not\vdash \varphi \). By completeness, \( \text{META} \vdash \neg \varphi \). Then \( \text{BASE} \vdash g(\neg \varphi) \), and \( \text{META} \vdash f(g(\neg \varphi)) \). By modelling, this is equates to \( \text{META} \vdash \neg f(g(\varphi)) \). By completeness again, \( \text{META} \not\vdash \varphi^{**} \). Finally, \( \text{META} \vdash \varphi^{**} \Rightarrow \text{META} \vdash \varphi \) by modus tollens.

The cyclical image \( \varphi^{**} \) of \( \varphi \) in \( \text{META} \) must be if not equal then at least equivalent to \( \varphi \).

**Section 5 – The reconstructed Theorem.**

**Definition 5.1:** The predicate TRUE is a truth predicate for a meta theory \( \text{META}_T \)

\[ \text{if for all sentences } \varphi, \text{META}_T \vdash \text{TRUE}(g(\varphi)) \leftrightarrow \varphi \]

where \( g \) is a modelling transformation from \( \text{META} \) into \( T \).

What follows is a reconstruction or restatement of Gödel’s Theorem within the framework we have developed so far. Though evidently not identical, it aims to stay as close as possible to the spirit of the original.

**Gödel’s Theorem (according to Gödel) [5.2]:** If both \( \text{BASE} \) and \( \text{META} \) are consistent and complete, and \( \text{BASE} \) strictly axiomatisable, then \( \text{META} \) contains a truth predicate.

**Proof:** We show that PROV is a truth predicate.

\[ \text{META} \vdash \text{PROV}(g(\varphi)) \Rightarrow \text{META} \vdash \varphi. \]

As PROV expresses provability for \( \text{BASE} \), \( \text{META} \vdash \text{PROV}(g(\varphi)) \) implies

\[ \text{BASE} \vdash g(\varphi). \]
By embedding, BASE |- g(φ) implies META |- f(g(φ)) = φ*.

Hence META |- φ (Round tripping).

META |- PROV(g(φ)) ⇐ META |- φ

META |- φ implies BASE |- g(φ).

As PROV expresses provability for BASE, META |- PROV(g(φ))

Predictably next,

**Tarski’s Theorem [5.3]:** No consistent and complete theory can contain a truth predicate.

Proof: By applying to itself the predicate Fssb(P) ≡ ¬TRUE(g(sub(P,g(P)))), (F for false, ssb for self-substitution).

Gödel’s cleverly self-referential sentence can be recovered as g(Fssb(Fssb)), the image in BASE of the sentence Fssb(Fssb) that would break META.

**Corollary 5.4:** Either that

META exists, and is consistent and complete

or that

BASE is consistent, complete, and strictly axiomatisable

is false.

Proof: Taken together the two would prove the existence of a consistent and complete theory, META, with a truth predicate, contradicting Tarski’s Theorem.

At first sight there appears to be an honest choice. The question seems to be on which of the initial assumptions should we pin the blame for a contradiction that emerges at the end of a long derivation.

The string theory META is a *prima facie* favourite for carrying the blame – it is an odd, untested and hugely ambitious theory. This conclusion becomes inevitable once we note that essentially the same
argument can be made without involving arithmetic at all, simply by taking a second copy of META in place of BASE. (It goes without saying that the argument can not be made with two copies of an arithmetic theory.)

**Gödel’s Theorem (2nd approximation) [5.5]:** \(\text{META}^{\text{META}}\), if consistent, complete, and strictly axiomatisable, contains a truth predicate.

Proof: Let \(\text{META}^{\text{META}}\), a simple variation of \(\text{META}^{\text{BASE}}\), be the meta theory for a generic concatenation-based meta theory over a sufficiently large finite alphabet.

Take two copies of \(\text{META}^{\text{META}}\), labelled \(\text{META}_1\) and \(\text{META}_2\). Each theory is, without any circularity, a meta theory for the other.

If \(\text{META}_1\) is axiomatisable, then by the same argument we used for \(\text{BASE}\) a provability predicate \(\text{PROV}\) that applies to the underlying theory – \(\text{META}_2\) is this case – can be defined.

One way still takes the Embedding Lemma, the way back is much easier this time: the identical mapping. Round Trip then holds true, so the Theorem follows.

Although we are not yet able to pinpoint exactly what it is about \(\text{META}\) that caused this result, there is one thing that we can confidently say: everything of importance in Gödel’s argument happens on the preimage side. String theories are first in line to suffer incompleteness results. Only after they fall would these results also transfer to other kinds of theories. As the 2nd Approximation once again demonstrates, there can be Gödel-style incompleteness arguments without number theory, and any of its subtheories. But there can be no incompleteness arguments without CONCAT-based meta theories. (Equivalents of CONCAT are always used, even if not always honestly declared.) At least on the preimage side, extensional string theories must always occur. From here on, as a consequence, number theory will be largely irrelevant to the investigation.

\(\text{META}^{\text{Meta}}\) is in no way special. Any variation of \(\text{META}\) for different underlying theories consists of a version of CONCAT and varying definitions of meta predicates. The differing versions of CONCAT are contained in each other by a simple permutation of the alphabet; the meta predicates for one formal theory are definable by the same means as the predicates for any other. If we can successfully define a
meta string theory for one axiomatisable theory, then we should be able to define a meta string theory for any other axiomatisable theory. Without loss of generality, we can therefore continue to speak of META without a subscript.

**Corollary 5.6:** META, if it exists at all, and is consistent, cannot be both complete and strictly axiomatisable.

Proof: By Tarski’s Theorem, from the 2nd Approximation.

This is not quite the same as the trilemma that Gödel thought applied to number theory: From the three desirable qualities of consistency, completeness, and axiomatisability, pick any two. There is a forth option, that CONCAT, which was used to build META, does not exist. The contradiction that carries the proof is the same. But as a pay-off for the cleanly syntactic presentation we have worked to achieve, we have an additional assumption for it to refute.

**Definition 5.7:** Let prMETA be the subtheory of META consisting only of primitive recursive sentences, i.e. sentences that combine relations definable by primitive recursion with at most bounded quantification.

prMETA would decide PROOF(s,t), but cannot decide PROV(x).

**Corollary 5.8:** prMETA, if it exists at all, is essentially incomplete.

Proof: prMETA cannot have a consistent extension that is both strictly axiomatisable and complete because this extension would decide META.

**Corollary 5.9:** If CONCAT exists, then there exists a theory that is essentially incomplete.

Proof: CONCAT added to the other known axioms for META would decide prMETA.

(By Proposition 4.2 for \( \Delta C \) sentences, by finite definability for the ordering relation, by Lemma 2.5 for primitive recursive predicates.)
By the same argument, if CONCAT were strictly axiomatisable, then prMETA too.

Except for the ‘if CONCAT exists’ condition, this result – the finding of a weak, essentially incomplete theory – is the same as in traditional presentations. We could have reached an identical result by running the argument with CONCAT on the image side, instead of BASE6.

Section 6 – Consequences

We are now in possession of a contradiction that refutes the conjunction of three assumptions, and we know what it would mean to deploy the contradiction to refute completeness / axiomatisability. What would it mean to deploy the contradiction to refute the new assumption on the board, Gödel’s Assumption, the assumption of the right to use extensional concatenation in Gödel-style meta arguments, an assumption we have operationalised as the existence of CONCAT? It would mean rejecting the existence of a theory that successfully envelopes all the theories CONCATn. Approximating theories exist, but there would be, as it were, no limit to infinity.

In the end, there is a choice, though not an equal one. The situation is that there are two conflicting assumptions, neither of which appears to be provable outright. We have tried and failed to prove Gödel’s Assumption from weaker or less problematic assumptions. On the opposite side we find the

6 CONCAT, if it existed and were complete, should be able to model any convincing theory of implicit concatenation. In the end, theories of both extensional and implicit concatenation aim to express (aspects of) the same thing. Moreover, except for non-letter constants, they share the same language, and should be able to converge on the theorems they prove. This is a matter of some delicacy as it is known that adaptations of a Gödel-style incompleteness argument are available already for quite weak theories of implicit concatenation, e.g. Grzegorczyk [2005].

The point to note is that even though, confusingly, they bear the promising label ‘concatenation’, implicit theories of concatenation still have to be interpreted as extensional concatenation for any meta arguments to get underway. In this they are no different from number theory, and the weaker subtheories of number theory, that were (informally) interpreted as extensional concatenation for the sake of incompleteness arguments running in parallel. The results for implicit concatenation theories could be restated as we have restated the result for number theory, and would then also come back with an additional assumption about CONCAT.
assumption of completeness for theories equipollent to number theory, e.g. number theory itself and comparably strong concatenation theories. Because for as long as we are also unable to prove the assumption of completeness – and it would be unrealistic to expect that to change – no hard contradiction will emerge. While the choice stands open, the completeness assumption, innocent though it may be, remains vulnerable to blame shifting.

Let’s look first at the choice that makes less sense. Assuming a great deal of motivation it would remain just possible to continue insisting on the existence of CONCAT.

However, at this late stage it would effectively turn Gödel’s Assumption into a postulate, Gödel’s Axiom. The Axiom would assert the Assumption; or more generally, formalise assumptions that certain string predicate expressions\(^7\) successfully extensionalise into sets of strings. Through an axiom of this type, simply by taking descriptive expressions at face value, one could force into existence an infinitary string object, a theory in the extensional sense of a set of formal sentences.

Apart from being declared directly, the existence of CONCAT could also be introduced indirectly, by way of other assumptions turned axioms that imply Gödel’s. An indirect way could be to approve a method of infinitary construction, or endorse a special notation with the same effect. (The success of constructions, and the non-emptiness of descriptions in a new notation, can only be guaranteed by a postulate.)

The infinitary string object thus created would be custom-made to be unformalisable. A flat set of sentences, unaxiomatised and essentially unaxiomatisable. With the set pretending to contain the otherwise unreachable “truth” about extensional concatenation, we would be provably unable to

---

\(^7\)These expressions are not simply from \(C\). The language of the Assumption/Axiom necessarily contains non-standard elements. The clean variety uses CONCAT\(_n\) as a well-defined language element with perimeters applied. The dirtier varieties forget about perimeters, or even appeal to “true” CONCATENATION(\(x, y\)), a mystical function that, naturally, has to remain unformalised and underdefined. The dirty varieties thus run into a vicious regress of meta theories presupposing each other; the clean variety is unique in managing to halt the regress. (CONCAT\(_n\) regresses to CONCAT\(_{n+x}\), which is only a perimeter expansion away. For the finite theories, meta level is not fundamentally more powerful than starting level.)
summarise it by finite axioms, or a finite number of regular schemes. (True, the set might be “recursively enumerable”, but recursive enumerability for arbitrary sets of strings that do not follow the pattern of a regular axiom scheme is only another disguise for the Assumption.)

Although possible, we should be clear about what taking this course of action would mean. Because of mutual representability by Gödel numberings and similar techniques it would mean that undecidability comes roaring back for a wide range of theories. Once (re)introduced as CONCAT, it would immediately spread to other theories, including some about as weak as Q.

In the final analysis, to adopt Gödel’s Assumption as an axiom would mean postulating essential incompleteness, and all the weirdness it entails, because one considers it to be a desirable state of the world.

Be that as it may, we stand at a fork, with two paths ahead. One is known to exhaustion, the other temptingly unexplored. For what remains of the paper, the stroll home, we will treat the soft contradiction as a hard ⊥, and choose to consider Gödel’s Assumption refuted.

**Gödel’s Theorem (3rd approximation) [6.1]:** Extensional concatenation is not formalisable in first-order logic: For any theory T with strings over Σ for constants, there exists an n and an atomic sentence c(s,t) = u that is decided by T in a different way than it is by CONCAT\(_n\).

Proof: This is only a statement of the negation of Gödel’s Assumption aka ‘CONCAT exists’.

Recall that, as a working assumption, CONCAT was defined to be a first-order theory. An obvious line of inquiry would be to ask whether higher orders of the predicate calculus, or other logics, would make any difference. It appears not. Readers are invited to verify that moving up to the second order does not materially change the outcome. The most promising route to an axiomatisation, the trick of quantifying inside terms, violates the standards of any conventional logic, second order as much as first. The presumption becomes that no consistent formalisation exists, in any conceivable type of formalism.
To refute the Assumption is to conclude that there is no formalism that can contain concatenation, that
decides all c-sentences the expected way. No formalism that can consistently meet the low standard set
by the definition of CONCAT for deserving to be called ‘extensional concatenation’.

The new path starts by accepting that string relations are different. Unlike mainstream theories,
extensional string theories are required for their own definition. They are inherently self-referential,
must presuppose themselves. Complete enumeration of their formulas would close the circle, and
create a paradox.

To escape the paradox, we conclude that there is no such thing as concatenation extending infinitely far.
Infinitary computing is not feasible. The idea of transcending all perimeters is incoherent. The notion
makes no sense.

There is no total c-function, native string relations never were total, there is nothing on the side of the
preimage for arithmetic relations to represent. Interpretation fails for all non-finite relations.

The new path continues to understanding that the absence of total string functions is not caused by
theories failing to live up to some independent truth. It is caused by the fact that there is nothing to
these notions of truth, nothing to formalise in the first place. Fragmented formalisations perfectly
express that string reality itself is fragmented.

Although unformalisability will disappear, strange occurrences would not. Extensional string predicates
will continue to provoke phenomena that can look superficially similar to the traditional weirdness
created by essential incompleteness:

There are fragments that do not assemble into a whole. Fragmented predicates \(P_n\), based on CONCAT\(_n\),
are admitted to exist, but these fragments do not assemble into total predicates \(P\). Finite combinations
of \(P_n\) and \(Q_m\) are possible, but, in general, no countably infinite unions.
Definition [6.2]: For any $P(x)$ from $C$, let $P_n(x)$ be the expression from $\Delta C$ that restricts $P(x)$ to $\Sigma^n$, i.e. strings over $\Sigma$ of length $\leq n$.

There are approximating sequences that converge over initial ranges, up to the edge of their perimeter, but fail to reach infinity. Specifically, there are diagonal predicates $D$ where for finite $N < M$ we can prove:

$\text{CONCAT}_1 \vdash \forall x \; D_1(x)$

$\text{CONCAT}_2 \vdash \forall x \; D_2(x)$

$\text{CONCAT}_3 \vdash \forall x \; D_3(x)$

... 

$\text{CONCAT}_N \vdash \forall x \; D_N(x)$

but we still cannot prove $\forall x \; D(x)$. No $\text{CONCAT}_n$ can prove it for an unrestricted domain; nor would implicit theories of concatenation, where $D(x)$ also forms part of the language. On the contrary, the expectation is that a plausible, sufficiently powerful theory of implicit concatenation would reject $\forall x \; D(x)$. Extensional concatenation cannot interpret even theories of implicit concatenation.

The inference

"$\forall i \; \text{CONCAT}_i \vdash \forall x \; D_i(x) \Rightarrow \forall x \; D(x)$"

is intuitively compelling, but false. Not so much because it is wrong – after all, all strings would eventually get covered by the enumeration –, but because without $\text{CONCAT}$ we cannot even state it. With perimeters properly applied, all we can state is for arbitrarily large $M$

$\forall i, m \; \text{CONCAT}_i \vdash \forall x \; D_i(x) \Rightarrow \forall x \; D(x)$,
which is not compelling at all.

One might paraphrase Gödel’s assumption as the ability to quantify over perimeters, so that one can coherently speak of “\(\forall M\)”, or “\(\forall_i \text{ CONCAT}_{i}\)”. This is equivalent to being allowed to enter another line into the enumeration above, after \(\text{CONCAT}_n \vdash \forall x D_n(x)\), reading ‘...’

**Definition [6.3]**: \(C_n\) is the sublanguage of \(C\) consisting of all formulas that contain at most constants over \(\Sigma\) of length \(\leq n\).

Abbreviate ‘strictly axiomatisable, consistent, and complete’ to ‘stacc’.

**Proposition [6.4]**: For \(n < M\), if a stacc theory \(T\) decides \(C_i\), then it has a stacc extension \(T_n\) that decides \(C_n\).

Proof: (As our one concession to the readability of concatenation we allow that \(n\)-ary concatenation has been derived from binary: \(c[x,y,z] = c(x,c(y,z))\), etc.)

Let \(t\) be any non-letter constant from \(\Sigma^n \setminus \Sigma\). Since \(t\) is a string over \(\Sigma\), for some letters \(a_i\) from \(\Sigma\), not necessarily distinct, \(t = c[a_1,a_2,\ldots,a_i]\) is a theorem of \(\text{CONCAT}_n\). (Pedantically speaking, not the expression itself, but a somewhat different sentence corresponding to it.)

Over \(\text{CONCAT}_n\)’s finite domain thus exists a string function \(f\) from non-letter constants to composite functional terms in only letters. (The function’s existence seems safe to assume at least for as long as the domain is finite.)

Extend \(T\) to \(T_n\) by adding all atomic concatenation statements that \(\text{CONCAT}_n\) proves. Their number is finite, so the extension is still strictly axiomatisable.

To decide any \(C_n\) sentence in \(T_n\), translate it into a \(C_i\) sentence by replacing all its non-letter constants \(t\) by \(f(t)\). Equivalence between the formula and its replacement is provable by the rules of identity. Hence \(T_n\) is complete.
Clearly, the extension is relatively consistent, since all it does is to create additional labels for already existing function terms.

Let $\text{TIC}$ be a theory of implicit concatenation over $\Sigma$. $\text{TIC}$ being an implicit theory means that its language is $C_1$. Suppose $\text{TIC}$ were stacc. Write $\text{TIC}_a$ for its stacc extensions to $C_a$.

**Lemma [6.5]:** For $n < M$, $\text{TIC}_a$ contains a model of $\text{CONCAT}_a$.

Proof: Map $P$ from $\text{CONCAT}_a$ to $P_a(x)$ in $\text{TIC}_a$.

The big question is now: Does there exist an extension “$\text{TIC}_\infty$” ? $\text{TIC}_\infty$ would model all $\text{CONCAT}_a$, hence contain a model of $\text{CONCAT}$. $\text{TIC}_\infty$ is almost already META (missing at most the harmless addition that is $O$), therefore clearly undecidable. Moreover, if from any $\text{TIC}$ a $\text{TIC}_a$ were constructible, the undecidability of $\text{TIC}_a$ would immediately reflect back on $\text{TIC}$.

The answer is, only if Gödel’s Assumption holds. Otherwise, “extended by all the atomic concatenation statements provable in any $\text{CONCAT}_a$” is not a statement we can make. All we can state is, for $N < M$, “extended by all the atomic concatenation statements provable in $\text{CONCAT}_a \leq N$” – which is useless for getting to $\text{TIC}_\infty$, as it describes (at most) $\text{TIC}_N$.

As was the case for the $D_i(x)$, without the Assumption we cannot state strong antecedents for the existence of the infinitary limit object – nothing beyond $N$ –, let alone actually prove its existence.

Infinitary fusions of finite fragments that can be shown to imply Gödel’s Assumption are refuted in the same way that the Assumption was refuted. Similar to the attempts of forcing a limit for the $D_i(x)$ and the $\text{TIC}_a$, many other kinds of (countable) constructions – infinite unions, infinite sequences, infinite indexes, etc. – refutably fail.
Borrowing from model theory or topology, one might say extensional string predicates are ‘non-compact’, their finite properties do not transfer to infinity. (Only obliquely non-compact, because, unlike topology, the reason here is not the way finite parts and an infinite whole relate to each other, but rather that there is no infinite entity, full stop.) In another analogy, string predicates are like the proper classes of set theory, in that they fail to extensionalise as expected.

More important, while there is weirdness, it is localised. Unlike the traditional symptoms, these strange tactics are not infectious, they affect only extensional concatenation, and not any unrelated concepts, or theories. Number theory, in particular, would be totally immune.

We can enjoy a much more orderly world in which all well-defined problems are expected to be decidable, and belief in the existence of unformalisable mathematical truth is considered quaint.

**Gödel’s Theorem (final) [6.6]:** Extensional concatenation is not a well-defined function.

In some ways, this conclusion is less a theorem than an exercise in boundary policing, an argument for which functions should be accepted (or formally postulated) as existing. By showing how bizarre the consequences of admitting extensional concatenation would be it urges to conclude that the idea has no merit. The better choice of Gödel’s Axiom is to postulate not Gödel’s Assumption, but its negation.

Gödel’s incompleteness argument, though endlessly revealing about string theories, tells us very little about number theory. The original incompleteness result, the root cause of all the others, is the result for CONCAT. The contradiction that, when the existence of CONCAT is assumed, can be used to refute the completeness of number theory, and many other theories, is frankly imported from the preimage side. Incompleteness results are without exception the effect of projecting onto others theories the strange properties and failings of a hypothetically assumed CONCAT.

When we do conclude that CONCAT does not exist, the projections will instantly stop, and number theory, as well as all other theories able to model prMETA, will no longer be affected by any kind of
unformalisability. (This includes also implicit theories of concatenation.) Once we reject Gödel’s Assumption, no reason remains for believing in Gödel’s original conclusion. Although absence of counterevidence does not constitute evidence, it is only natural to revert to completeness as the default assumption.

The lesson of Gödel’s Theorem: Concatenation, used naïvely, can be just as treacherous as the membership relation $\in$. So in a way Gödel’s argument does for concatenation-based theories what Russell’s paradox does for set theory: For both types of theories, the naïve assumption that all predicate expressions ought to be able to have an extension turns out to be inconsistent despite its overwhelming intuitive appeal.
Gödel, Kurt [1931]: Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I, *Monatshefte für Mathematik und Physik*, 38, 173–198.

Grzegorczyk, Andrzej [2005]: Undecidability without arithmetization. *Studia Logica*, 79 (2), 163–230.