The expressive power of formal languages

MELAMPUS

The book paradox

Since formal languages may be recursively generated, it would *appear* that any wff in a formal language denotes (refers to) a structure that is *effectively computable*. Upon further reflection it transpires that this impression must be false, since on that basis it would be impossible to refer to any structure that was not "effective"; the whole question of what is effective and what not would be meaningless; it would be meaningless to discuss or refer to any function or structure that was not effective. We see immediately that the putative fact that a formal language is recursively generated does not strictly entail that the objects that expressions of that language refer to or denote are themselves recursive (effective).

This feature of formal languages, namely that, though recursively generated themselves, their denotations may or may not be effective computable entities, is illustrated by the "book paradox". This is the subject of the story by Borges[1] – the idea that one could make a program that would progressively and automatically generate every book whatsoever simply by combinatorics. Of these "books" the majority would be sheer nonsense but some would comprise whole texts. *Hamlet* would eventually be generated in this way – so would every theorem that has been and ever will be proven. And so on. What this shows is that the mere fact that meaningful expressions can be generated *at random* (or *systematically*, which amounts to the same thing, their appearance being random) does not mean that their denotations are effective. The solution must be that *the manner in which the expressions are generated must match the effective way in which their denotations are generated.* If an expression α is recursively generated from an expression β it does not follow that the relation R that α denotes can be recursively generated from the relation S that β denotes. For such a correspondence to hold it must be that the manner in which β is generated from α *matches* the manner in which S is generated from R. There must be a homomorphism or structure preserving relationship, and in the absence of such relationship, we cannot conclude from R is recursive that S is recursive.

Expressive power of language used in Gödel's theorem

A preliminary point is that in the proof of Gödel's theorem the introduction of Gödel numbering is *a red herring*! This may seem an extraordinary claim in view of the fact so much attention is given to Gödel numbering, which is seen as a crucial step in the proof. However, Gödel numbering is just *another cipher for a wff* and nothing more – a fact that is clear from

[1] Jorge Luis Borges: *The Library of Babel* in *Labyrinths*.

some interpretations of Gödel numbers as just ASCI code for the formulas. The device of Gödel numbering is merely used to convince one that certain formulae are recursively generated from other formulae – but that is something that might have been accepted merely by examining the formulae themselves. In my formal version of Gödel's theorem [Chapter 9] I provided a table of translations that includes the following...

Godel number, $\lceil \phi \rceil$ / "meta-language"	Formula, ϕ / "object-language"
$\lceil X \rceil$	$X = X(\xi)$
$\lceil \Gamma \rceil$	$\Gamma = \phi_1 \circ \phi_2 \circ \ldots \circ \phi_n$
$\mathrm{Pf}(\lceil \Gamma \rceil, \lceil X \rceil)$	$\Gamma \vdash X$
$\mathrm{Sub}(\lceil X \rceil, \lceil X \rceil, \lceil X \rceil)$	$X(X(\xi))$
... ...	... ...

I labelled the two sides of the translation, "meta-language" on the one hand, and "object-language" on the other. This is in accordance with the popular description of Gödel's theorem as a result of meta-mathematics. This distinction is misconceived. The Gödel numbers are merely alternative formal codes for the formulae, and we have not broken out of the object-language (of a sufficiently strong first order logic) at all in the above table, neither into the lattice that is its model – nor into the meta-language to discuss that model. It is indeed correct to say that the whole reasoning of Gödel's theorem proceeds in the formal language in which it is written. [See 9/1.5 for initial discussion of this point]

Nonetheless, in the proof of Gödel's theorem the expressive power of the formal language is shown to transcend the class of effective objects. The formal language contains names of objects that are not effective. My task is to show in detail how this happens. We start with the Gödel number $\lceil X \rceil$ of an effective entity that, for the sake of the argument, must be dependent on a parameter ξ, so we may write $X = X(\xi)$. Now ξ is a free variable in *X*, which endows $X(\xi)$ with the expressive potential of a function. To explain: in predicate logic, *X* is a predicate and $X(\xi)$ stands for *an arbitrary* predication of *X* of an individual. But here we are using ξ as a variable, or parameter. The situation is akin to that in Whitehead and Russell [1910][2] where predicates are actually functions. $X(\xi)$ is used both as a function and to denote the image (range) of the function *X*: $X(\xi) = \{y : y = X(\xi) \text{ for all } \xi \in L\}$ where *L* is the lattice. That is, implicit in expression $X(\xi)$ is the assumption that ξ ranges over lattice points. Thus *X* is a mapping of one lattice to another and copies the structure of the lattice over which ξ ranges into codomain (which is also a lattice). As ξ varies $X(\xi)$ over lattice points of the codomain. For fixed ξ, then $X(\xi)$ becomes a lattice point. Because ξ is a variable, then it can range over the whole lattice, and hence the range of $X(\xi)$ corresponds. Because $X(\xi)$ is a function, then, via the implicit function theorem, the indeterminate ξ may

[2] Gödel's original theorem was predicated of that system (Gödel [1931]).

itself be the image of another function $Y = Y(\zeta)$. So we may substitute Y for ξ in $X(\xi)$ to obtain $X(Y(\zeta))$. The expression $X(Y(\zeta))$ records these functions as mappings between three lattices; firstly the lattice over which ζ is defined, then its image $Y(\zeta)$, which is then mapped to $X(Y(\zeta))$. In the expression $X(X(\xi))$ the mapping has become an <u>automorphism</u> of the lattice: domain and image coincide.[3] As $X(X(\xi))$ is a mapping of the lattice onto itself, it may and does have a fixed point. Denote this fixed point by *b*. Then $b = X(b) = X(X(b))$. As X is an arbitrary function any fixed point could be anywhere in the lattice. So far, no "paradox" or "problem" has arisen for the expressive power of the language has not be stretched beyond that which is effective.

This gives another characterisation of the property of a logic of *sufficient strength*. A logic with sufficient strength enables one to define automorphisms of its models; it is from these automorphisms that the fixed point property arises. This in turn allows one to demonstrate the truth of Gödel's incompleteness theorem. Sufficient strength turns notional predicates of the compact region $CF(\omega)$ [5/5.8] into functions of the entire lattice; hence it conceptually embeds $CF(\omega)$ into its completion, $\mathbf{2}^{\omega}$, and thereby renders locally compact proof paths in $CF(\omega)$ essentially incomplete.

$\mathrm{Pf}(\ulcorner\Gamma\urcorner,\ulcorner X\urcorner)$ is the Gödel name of the formula $\Gamma \vdash X$ where Γ is a compact proof sequence. In this context it is more appropriate to use the equivalent notion $\Sigma \vdash X$ where Σ denotes a set from which X is derived, which includes the set of axioms for the lattice. Since $\mathrm{Pf}(\ulcorner\Gamma\urcorner,\ulcorner X\urcorner)$ is recursive the relation $\Sigma \vdash X$ denotes a compact proof path in the lattice. It also says that X lies in the compact filter defined by X. $\mathrm{Sub}(\ulcorner X\urcorner,\ulcorner \xi\urcorner,\ulcorner Y\urcorner) = \ulcorner X(Y(\zeta))\urcorner$ also does not take us out of the class of finite filters. $\mathrm{Pf}(\ulcorner\Sigma\urcorner,\mathrm{Sub}(\ulcorner X\urcorner,\ulcorner X\urcorner,\ulcorner X\urcorner))$ which expresses $\Sigma \vdash X(X(\xi))$ also does not take us out of the finite filter defined by Σ.

It is precisely at $\underline{\neg\mathrm{Pf}(\ulcorner\Sigma\urcorner,\mathrm{Sub}(\ulcorner X\urcorner,\ulcorner X\urcorner,\ulcorner X\urcorner))} \equiv \Sigma \nvdash X(X(\xi))$ that the expressive power of the language is extended by a recursive process in the formal language to denote an object that is <u>not effective</u> in the model. $\Sigma \nvdash X(X(\xi))$ denotes the whole part of the lattice that is <u>the complement of the lattice</u> $\Sigma \vdash X(X(\xi))$. It is <u>not a lattice point</u>. As the complement of an <u>open set</u> it includes both the boundary of $\Sigma \vdash X(X(\xi))$ and everything that is not contained in its interior. By this device we have named something that is not a lattice point or a relation between lattice points. The logic has been allowed to transcend in denotation whatever is recursive in the compact filter $\Sigma \vdash X(X(\xi))$.

---

[3] For a given individual *a* of the elementary domain, the expression $X(X(a))$ presents no more difficulties than it would for any function, $f(f(a))$.

At the next stage, where $P(\xi) \equiv (\forall \bar{x})\left(\neg \underline{\mathrm{Pf}\left(x, \mathrm{Sub}\left(\ulcorner X \urcorner, \ulcorner X \urcorner, \ulcorner X \urcorner\right)\right)}\right) \equiv \nvdash X\left(X(\xi)\right)$, $k = \ulcorner P \urcorner$ we have quantified over all compact paths (or their corresponding lattice points) in the lattice to obtain the expression of <u>that which cannot be proven by compact paths in the lattice</u>. When we substitute P for X we obtain $\mathrm{Pf}\left(\ulcorner \Sigma \urcorner, \mathrm{Sub}(k,k,k)\right)$, $\Sigma \vdash P\left(P(\xi)\right)$; this is something that is necessarily false of the lattice, though there is no finite proof in the lattice that this is so. By naming the fixed point of this function, $Q(\xi) \equiv (\forall \bar{x})\left(\neg \underline{\mathrm{Pf}\left(\bar{x}, \mathrm{Sub}(k,k,k)\right)}\right) \equiv \nvdash P\left(P(\xi)\right)$, $m = \ulcorner Q \urcorner$ we quickly obtain the formal expression of this: -

$$P\left(P(\xi)\right) \equiv \nvdash P\left(P(\xi)\right)$$
$$Q \equiv \nvdash Q$$

Whereas $\nvdash X\left(X(\xi)\right)$ places the fixed point on the boundary or above, $Q \equiv \nvdash Q$ places it <u>precisely on the boundary</u>. [6 / 2.3] So by this means we have achieved an expressive expansion of the formal language so that it can name and "define" lattice points that cannot effectively (that is, finitely) be reached from an arbitrary lattice point Σ.

It is possible to examine a version of Gödel's theorem that is based on anti-diagonalisation to explore the expansion of the expressive power of the language there as well. We start with some notion of the set of all recursive functions, and presume that these are represented in the formal language by an enumeration of wffs, $\phi_1, \phi_2, \phi_3, \ldots$. By anti-diagonalisation we <u>define</u> an expression by the rule $\phi(n) = \phi_n(n) + 1$. Then the assumption that this on the list of recursive functions leads to the contradiction, $\phi(k) = \phi_k(k) + 1$. So we see immediately that anti-diagonalisation leads to an expansion of the expressive powers of the language.

Expressive power of formal languages in general

Turning to the discussion of language in general.

1. Formal languages contain expressions can be recursively generated from an initial set of expressions.
2. Among these expressions some denote <u>effectively computable</u> entities and others do not.
3. There have always been a substantial number of examples of expressions that denote objects that are <u>not effective</u>. For example

   3.1 $\mathbf{P}(\omega) = \mathbf{2}^{\omega}$

   It is worth noting that the *power set operation* when applied to infinite sets is *non-effective*. If indeed we knew how to construct $\mathbf{P}(\omega) = \mathbf{2}^{\omega}$ there would be no problem of the continuum. The whole endeavour to solve the continuum hypothesis devolves around the desire to construct $\mathbf{P}(\omega) = \mathbf{2}^{\omega}$ from below.

3.2 The least definable ordinal.

3.3 As indicated, by anti-diagonalisation we can define sets that are not effective.

3.4 ω_1 is not effective but it can be named as "the least ordinal not equinumerous to $\omega_0 = \omega$".

3.5 Through forcing, all generic sets.

4. Negation, quantification, anti-diagonalisaton and forcing can lead out of the class of effective expressions to the class of those expressions that are non-effective.

To reiterate the conclusion: the expressions of all sufficiently strong formal languages are recursively generated, but their denotations are not.

In formal languages the building blocks of the language are defined at the first stage, and thereafter all expressions are simply combinations of those building blocks in accordance with formation rules. In natural languages this is not so. Natural languages allow for indeterminate expansions of the vocabulary and expressions by means of metaphor and concept stretching (See Lakatos [1979] Chap.1 Sec.8). When human beings encounter a difficulty in a problem they usually overcome that difficulty by inventing a new concept, one that is no mere definition by equivalence of existing concepts; often this invention is accompanied by the invention of a new word or expression. A natural language is a living language whose generation, life and decay mirrors the generation, life and decay of the culture that uses it.

The expressions "definable set / function / relation" are *highly ambiguous*. Some definitions of sets are definitions of objects that are effectively computable, whereas others are not. It therefore does not follow that merely to define a set is to define an object that is effective not even when that definition occurs within a first-order language. We have seen in this chapter that if we start with objects which are effective certain devices of our language can lead us out of the class of effective objects. Therefore, definition is a very tricky business indeed, *especially if we are obliged to keep track of what is and what is not effective* in it.

The Arithmetical hierarchy

Just as the expressive power of formal languages ranges "far beyond" that which is effective, the arithmetical hierarchy [Chap.2 / 2.4.2] is in general not effective, and "definable" set does not mean the same as effective. Negation, quantification, anti-diagonalisation and forcing can lead out of the class of effective expressions to the class of those expressions that are non-effective. Thus a definable set that is effective at one level can through these devices be turned into a definable set that is not effective. Similarly, the term "constructible" is not a synonym of "effective". Sets in the constructible hierarchy are not necessarily recursive. Evidence for this view: -

1. “It is worth noting that, in ZF or ZFC, all complements are relative. That is, if x is a set, then $\{z : z \notin x\}$ cannot be a set; it is always a proper class.” (Wolf [2005] p. 75.) What this illustrates is that complementation can take one out of the class of effective sets; the extraordinary fact is that, in the absence of any restriction, it always does. However, if there is a set model of set theory (which is doubtful) then that might make the complement of a set into a set – it certainly won’t make it into an effective set – so no gain for the view that “definable” is equivalent to “recursive”. Such a view would depend on the existence of a large cardinal – so upon an extraneous axiom. Thus we see that any construction or hierarchy of constructions that permits complements is potentially a source of non-effective sets, even if at some base level the sets are effective.
2. We have some specific “effective” results:
   - 2.1 Intersections, complements and unions of PR relations are PR.
   - 2.2 Intersections, complements and unions of recursive functions are recursive.
   - 2.3 Domain, range and graph of any recursive function must be R.E.

   These all lie on the “positive” side of the question of transitivity of effectiveness; but by implication they indicate the general result: effectiveness is not a transitive relation under “definition by formula”.
3. The cumulative hierarchy: $\{V_\alpha\}$ contains non-effective (recursive) sets. Hence, a set may be defined within $\{V_\alpha\}$ without being effective. All the proper axioms of ZFC except the axiom of infinity are true in the structure $(V_\omega, \in)$.

Definable and effective

The confusion between the use of “definable” as a synonym of “effective” is frequent. For example, we see the theorem: -

Result

The set of ordered pairs: $H = \{(w,x) : \varphi_\omega(x) \text{ is defined}\}$ is undefinable.

Proof

Let the characteristic function of this set be

$$\chi_H = \begin{cases} 1 & \text{if } (w,x) \text{ is defined} \\ 0 & \text{otherwise} \end{cases}$$

Suppose this is recursive, and let

$$f(x) = \begin{cases} 0 & \text{if } \chi_H(x,x) = 0 \\ \varphi_x(x) + 1 & \text{otherwise} \end{cases}$$

Since we suppose χ_H is recursive, then f must be, and $f = \varphi_k$ for some k. then $\varphi_k(k) = 0$ if $\chi_H(k,k) = 0$ if $\varphi_k(k)$ is undefined , and $\varphi_k(k) = \varphi_k(k) + 1$ otherwise. This is a contradiction. Hence H is undefinable.

The point here is that in $H = \{(w,x) : \varphi_\omega(x) \text{ is defined}\}$ the term "defined" means "effective" or "recursive"; yet in *some other sense* H has been defined, and to say it is "undefinable" means that it has been defined and yet is not effectively computable.

Representable

Consider the definition of a "representable" relation:

$$(a_1, \ldots, a_n) \in R \quad \text{iff} \quad K \vdash A(\bar{a}_1, \ldots, \bar{a}_n)$$

where A is a wff. At first glance the term "representable" might be taken to mean something more than just "recursive", but because of the relation of deduction required in the language, only recursive relations can be "representable" in this sense. The same applies to the definitions of "expressible function" and "definable set" (Mendelson [1979] p. 134) – they are *defined* in such a way that only effective sets can match them.[4] Consider the following statement of a corollary to Church's Theorem: -

> The set of all true sentences in the language of PA is not recursive, and, hence, <u>not representable</u>. (Wolf [2005] p. 137.)

My underlining. We see here the specific use of "representable" to *mean only a formula representing a recursive function.* So the general statement, "truth is not definable" is <u>not proven</u>. What is proven is that truth is not a recursive relation.

Berry's paradox

At the root of the problem of definability – is Grelling's paradox.

> Let Ω be <u>defined</u> (A) as the least <u>undefinable</u> (B) ordinal.

The paradox is that Ω is defined iff Ω is undefined.

Resolution of this paradox: The word "defined" is being used in different senses. One of these meanings is "named". In (A) "defined" means "named", and in (B) "defined" is relative to a specific relation, for example, a list; the term "the least" only has meaning in the context of a well-ordered list. It might be argued that there is no <u>least</u> undefinable ordinal. We have

[4] Assuming that the deductive relation $\vdash$ is compact.

Plato's, "How is it possible to think the thing that is not?"[5] We have the question of what does "the unnameable" mean? "The unmentionable." By anti-diagonalisation we construct an "undefinable" function, but we do construct it – i.e. define it. All these problems are resolved by realising that the term "definable" is being used in different senses whenever a version of Berry's paradox is suggested.

The arithmetical hierarchy revisited

The arithmetical hierarchy is introduced by the analysis of definable sets of structures. In this context "definable" means roughly what one would intuitively expect it to mean: that is, a definable structure is one that we can name and characterise in some way, irrespective of whether it is effective or not. Sets and functions "definable" in the standard model N[6] of arithmetic are said to be "arithmetical". Here "definable" does not mean "recursive".

1. Every natural number is defined by a term (numeral) and hence is $\varnothing$-definable.
2. Every set representable in PA is arithmetical. Every recursive set is arithmetical.
3. In fact, a subset of $\mathbb{N}^k$ is RE iff it is Σ_1.
4. A set is recursive iff it is Δ_1.This means that if a set is $\Delta_n, n \geq 2$ is recursive then it must be equivalent to a Δ_1 set. In that case $\Delta_n, n \geq 2$ shall be said to be *essentially ineffective* (my term) if there is no Δ_1 equivalence. The issues raised here are the same as those involved in quantifier elimination: if we can eliminate all quantifiers then the structure is recursive. In model theory definable $\neq$ recursive.

Preservation theorems

This also involves the question of *preservation theorems.*

> A first-order formula is said to be *positive* if it does not contain the connectives, $\neg, \supset$ and $\equiv$. So it may contain $\wedge$ and $\vee$ as well as quantifiers.

This is a tacit admission that complements lead out of the class of effective structures and so fail to preserve structures.

> **Lyndon's theorem**
> Let *T* be a consistent theory. Then *T* is preserved under homomorphic images iff *T* is equivalent to a set of positive sentences.

[5] This is a quotation from Plato's dialogue *The Theatetus.* Plato attacks the doctrine of Protagoras described as 167A of that dialogue: "for it is not possible either to think the thing that is not or to think anything but what one expreiences, and all experiences are true." See Cornford [1979] p. 110.

Examples

The axioms of group theory are positive.

The axioms of ring theory include $0 \neq 1$ which is not positive.

Preservation under submodels and intersections

Definition: Let T be a theory. T is said to be preserved under submodels if $B \models T$ and S$U \subseteq B$ imply $U \models T$.

Los-Tarski theorem

A theory is preserved under submodels iff it is equivalent to a set of Π_1 sentences.

These "positive" results indicate the "negative" conclusion: that a theory is not preserved otherwise. In other words, it is "easier" not to preserve a theory than to preserve it.[7]

In conclusion: the power of formal languages, including first-order logic, enables one to define and express notions that refer to objects that are not recursive. It is not sufficient for such an expression to be formally manipulated in a recursively generated language for the object that it denotes also to be effectively computable.

[6] See Boolos and Jerffrey [1980] Chatper 17.

[7] Preservation under direct products leads to the topic of Horn formulas. (See Monk [1976] p. 398)