Gödel Numbering

{x: x is a horse} is a collection that has all the worlds horses as elements, and nothing else. Thus we have

For any y, y $\in$ {x: x is a horse} if and only if y is a horse.

Traveler, for example, is a horse, so Traveler $\in$ {x: x is a horse}. We would expect this pattern to hold generally, so that we have:

For any y, y $\in$ {x: x is _____} if and only if y is _____,

for any way of filling in the blank.

That's what we'd expect. But now try filling in the blank with "not an element of itself"; we get:

For any y, y $\in$ {x: x is not an element of itself} if and only if y is not an element of itself.

Substitute "{x: x is not an element of itself}" for "y," and we get a contradiction:

{x: x is not an element of itself} $\in$ {x: x is not an element of itself} if and only if {x: x is not an element of itself} is not an element of itself.

This is *Russell's paradox*, one of several set-theoretic paradoxes that shook the foundations of mathematics around the beginning of the twentieth century. The reason these paradoxes were so disturbing is that, during the last few decades of the nineteenth century, the idea of a set had played an increasing role in clarifying and securing the foundations of mathematics, particularly the calculus. By making set theory insecure, the paradoxes threatened to undo all these gains.

A strategy for responding to the paradoxes is to adopt, hopefully on a principled basis, axioms that restrict the way the blanks can be filled in enough to prevent contradictions but not so much as to prevent set theory from playing its useful mathematical roles. That's a good questions, but how do we know that our axioms still don't still produce contradictions? The axioms are chosen to block the path by which Russell obtained a contradiction, but how can we be sure we won't still arrive at a contradiction by some, more devious path?

David Hilbert proposed an innovative solution to this problem. Ordinarily, mathematicians study things like points, lines, planes, functions, and numbers, and they use mathematical proofs as a means of learning about these things. Hilbert proposed to treat mathematical proofs as themselves the objects of mathematical investigation. A new branch of mathematics, *metamathematics*, would study proofs the way geometers study points, lines, and planes. The hope would be that an investigation of proofs would enable us to prove that the axioms of set theory wouldn't lead to a contradiction. Graph theorists investigate when it is possible to find a path from one location to another within a complex network of points connected by curves. Proof theorists do something similar, looking to see whether there is a path leading from the axioms to a contradiction. Russell's paradox has taught us to be wary of proofs in set theory. Hilbert thinks we needn't be similarly chary of the proofs produced by proof theorists. The difference is that sets are frequently infinite, and so impossible to display concretely or survey fully. Proofs, on the other hand, are finite objects that we can actually write out on paper.

Metamathematics, as Hilbert envisaged, is separate from the rest of mathematics, because it studies a different kind of thing from the things ordinary mathematicians study. Gödel devised a method of assimilating proof theory into ordinary mathematics by assigning arithmetical codes to the different symbols, thereby turning the question whether a particular sentence is provable from a certain set of axioms into an arithmetical question.

The details of the coding are fairly arbitrary. What we'll see here is one possibility among many. We'll begin by encoding terms by ordered pairs and ordered triples. For any x, y, and z, Triple(x,y,z) = Pair(x,Pair(y,z)). If w = Triple(x,y,z),then 1stin3(w) = x, 2ndin3(w) = y, and 3rdin3(w) = z.

The code for "0" is the pair Pair(1,0), which we abbreviate $\ulcorner 0 \urcorner$.

The code for "x_n" is the pair Pair(2,n), which we abbreviate $\ulcorner x_n \urcorner$.

The code for $s\tau$ is the pair Pair(4,$\ulcorner \tau \urcorner$), which we abbreviate $\ulcorner s\tau \urcorner$.

The code for $(\tau+\rho)$ is the triple Triple(5,$\ulcorner\tau\urcorner$,$\ulcorner\rho\urcorner$), which we abbreviate $\ulcorner(\tau + \rho)\urcorner$.

The code for $(\tau\bullet\rho)$ is the triple Triple(6,$\ulcorner\tau\urcorner$,$\ulcorner\rho\urcorner$), which we abbreviate $\ulcorner(\tau \bullet \rho)\urcorner$.

The code for $(\tau\ E\ \rho)$ is the triple Triple(7,$\ulcorner\tau\urcorner$,$\ulcorner\rho\urcorner$), which we abbreviate $\ulcorner(\tau\ E\ \rho)\urcorner$.

Theorem. The set of codes of terms is a Δ set.

Proof. The set of codes of terms in Σ. x is the code of a term if and only if it's an element of a finite set s with the following properties:

If Pair(4,y) $\in$ s, y $\in$ s.

If Triple(5,y,z) $\in$ s, then y $\in$ s and z $\in$ s.

If Triple(6,y,z) $\in$ s, then y $\in$ s and z $\in$ s.

If Triple(7,y,z) $\in$ s, then y $\in$ s and z $\in$ s.

If y $\in$ s, then either y = Pair(1,0) or $(\exists n < y)y$ = Pair(2,n) or

$(\exists n < y)y$ = Pair(4,n) or $(\exists m < s)(\exists\ n < s)y$ = Triple(5,m,n) or

$(\exists m < s)(\exists\ n < s)y$ = Triple(6,m,n) or $(\exists m < s)(\exists\ n < s)y$ = Triple(7,m,n).

The properties the code number of s has to satisfy for these conditions to be met can be expressed by a bounded formula.

The complement of the set of codes of terms is Σ. It will be helpful to have the following definition on board:

Definition. $x \dot{-} y = x - y$ if $x \geq y$;

$= 0$ if $x < y$.

Note that $x \dot{-} y = z$ iff $((z + y) = x \vee (x < y \wedge z = 0))$; since this is a bounded formula, $\dot{-}$ is a Δ total function.

If x isn't the code of a closed term, then, if we try to form the structure tree for x, there will a branch that doesn't terminate either in "0" or a variable. Thus, x is not a code of a term if and only if there is a finite sequence s with the following properties:

$(s)_0 = x$.

If $n <$ length(s) and $(s)_n$ = Pair(4,y), then $n+1 <$ length(s) and $(s)_{n+1} = y$.

If $n < \text{length}(s)$ and $(s)_n = \text{Triple}(5,y,z)$, then $n+1 < \text{length}(s)$ and $(s)_{n+1}$ is equal to either y or z.

If $n < \text{length}(s)$ and $(s)_n = \text{Triple}(6,y,z)$, then $n+1 < \text{length}(s)$ and $(s)_{n+1}$ is equal to either y or z.

If $n < \text{length}(s)$ and $(s)_n = \text{Triple}(7,y,z)$, then $n+1 < \text{length}(s)$ and $(s)_{n+1}$ is equal to either y or z.

If $n+1 < \text{length}(s)$, then either $(s)_n = \text{Pair}(4,(s)_{n+1})$ or $(\exists z < s)(s)_n$ is equal to either $\text{Triple}(5,(s)_{n+1},z)$ or $\text{Triple}(5,z,(s)_{n+1})$ or $\text{Triple}(6,(s)_{n+1},z)$ or $\text{Triple}(6,z,(s)_{n+1})$ or $\text{Triple}(7,(s)_{n+1},z)$ or $\text{Triple}(7,z,(s)_{n+1})$.

$(s)_{\text{length}(s) \dot{-} 1} \neq \text{Pair}(1,0)$.

$\neg(\exists n < s)(s)_{\text{length}(s) \dot{-} 1} = \text{Pair}(2,n)$.⊠

The set of codes of closed formulas is Δ; just leave off the clause for variables.

A *finite tree* is a finite set of sequences with the property that any initial segment of a member of the set is a member of the set; the statement that s is the code of a fine tree can be formalized by a bounded formula. A *finite binary tree* is a finite tree consisting entirely of sequences of 0s and 1s. Where x is a code of a term, a *structure tree* for x is a pair Pair(s,f), where s is a code of a finite binary true and f is a function with domain the set of elements of the set coded by s that satisfies the following properties:

f assigns x to the trunk of the tree, < >.

If $y \in s$ and $f(y) = \text{Pair}(4,z)$, then $y \hat{} \langle 0 \rangle \in s$ and $f(y \hat{} \langle 0 \rangle) = z$, and $y \hat{} \langle 1 \rangle \notin s$.

If $y \in s$ and $f(y) = \text{Triple}(5,z,w)$, then $y \hat{} \langle 0 \rangle$ and $y \hat{} \langle 1 \rangle$ are both in s, and $f(y \hat{} \langle 0 \rangle) = z$ and $f(y \hat{} \langle 1 \rangle) = w$.

If $y \in s$ and $f(y) = \text{Triple}(6,z,w)$, then $y \hat{} \langle 0 \rangle$ and $y \hat{} \langle 1 \rangle$ are both in s, and $f(y \hat{} \langle 0 \rangle) = z$ and $f(y \hat{} \langle 1 \rangle) = w$.

If $y \in s$ and $f(y) = \text{Triple}(7,z,w)$, then $y \hat{} \langle 0 \rangle$ and $y \hat{} \langle 1 \rangle$ are both in s, and $f(y \hat{} \langle 0 \rangle) = z$ and $f(y \hat{} \langle 1 \rangle) = w$.

If y ^ <0> ∈ s and f(y ^ <0>) = z, then either f(y) = Pair(4,z) or

(∃w < s)(f(y ^ <1>) is defined and equal to w and f(y) = Triple(5,z,w)) or

(∃w < s)(f(y ^ <1>) is defined and equal to w and f(y) = Triple(6,z,w)) or

(∃w < s)(f(y ^ <1>) is defined and equal to w and f(y) = Triple(7,z,w)).

If y ^ <1> ∈ s and f(y ^ <1>) = w, then, for some z < s, f(y ^ <0> is defined and equal to z and either f(y) = Triple(5,z,w) or f(y) = Triple(6,z,w) or f(y) = Triple(7,z,w).

If y ∈ s and neither y ^ <0> nor y ^ <1> is in s, then either f(y) = Pair(1,0) or

(∃n < s)f(y) = Pair(2,n).

This can be formalized by a bounded formula.

Theorem. The function Den that takes a closed term to the number it denotes is Δ.

Proof: Den(x) = v iff x is a closed term and there exist s and f such that Pair(s,f) is a structure tree for x and there is a function g with domain s, with the following properties:

If f(y) = ⌜0⌝, g(y) = 0.

If f(y) = Pair(4,z), then g(y) = g(z) + 1.

If f(y) = Triple(5,z,w), then g(y) = g(z) + g(w).

If f(y) = Triple(6,z,w), g(y) = g(z) • g(w).

If f(y) = Triple(7,z,w), g(y) = g(z) E g(w).

g(< >) = v.

This can be formalized by a Σ formula. Since Den is a Σ partial function with a Δ domain, it's Δ.⊠

We encode formula the same way:

If τ and ρ are terms, the code for τ = ρ is Triple(8,⌜τ⌝,⌜ρ⌝), which we abbreviate ⌜τ = ρ⌝.

If τ and ρ are terms, the code for τ < ρ is Triple(9,⌜τ⌝,⌜ρ⌝), which we abbreviate

$\ulcorner \tau < \rho \urcorner$.

If ϕ is a formula, the code for $\neg\phi$ is Pair(10,$\ulcorner\phi\urcorner$), which we abbreviate $\ulcorner \neg \phi \urcorner$.

If ϕ and ψ are formulas, the code for $(\phi \vee \psi)$ is Triple(11,$\ulcorner\phi\urcorner$,$\ulcorner\psi\urcorner$), which we abbreviate $\ulcorner(\phi \vee \psi)\urcorner$.

If ϕ and ψ are formulas, the code for $(\phi \wedge \psi)$ is Triple(12,$\ulcorner\phi\urcorner$,$\ulcorner\psi\urcorner$), which we abbreviate $\ulcorner(\phi \wedge \psi)\urcorner$.

If ϕ and ψ are formulas, the code for $(\phi \rightarrow \psi)$ is Triple(13,$\ulcorner\phi\urcorner$,$\ulcorner\psi\urcorner$), which we abbreviate $\ulcorner(\phi \rightarrow \psi)\urcorner$.

If ϕ and ψ are formulas, the code for $(\phi \leftrightarrow \psi)$ is Triple(14,$\ulcorner\phi\urcorner$,$\ulcorner\psi\urcorner$), which we abbreviate $\ulcorner(\phi \leftrightarrow \psi)\urcorner$.

If ϕ is a formula, the code for $(\forall x_n)\phi$ is Triple(15,n,$\ulcorner\phi\urcorner$), which we abbreviate $\ulcorner(\forall x_n)\phi\urcorner$.

If ϕ is a formula, the code for $(\exists x_n)\phi$ is Triple(16,n,$\ulcorner\phi\urcorner$), which we abbreviate $\ulcorner(\exists x_n)\phi\urcorner$.

Theorem. The set of codes of formulas is Δ.

The proof is so close to the proof of the analogous proof for codes of terms that there's no real point in going through it.

Theorem. The function that, for θ a formula, τ a term, and n a natural number, takes $<\ulcorner\theta\urcorner,\ulcorner\tau\urcorner,n>$ to $\ulcorner\theta^{x_n}/_{\tau}\urcorner$ is Δ.

Proof: First note that the function that, given terms τ and a natural number n, takes $<\ulcorner\rho\urcorner,\ulcorner\tau\urcorner,n>$ to $\ulcorner\rho^{x_n}/_{\tau}\urcorner$ is Σ. That's so because the value the function takes with input $<\ulcorner\rho\urcorner,\ulcorner\tau\urcorner,n>$ is equal to z if and only if, where <s,f> is the structure tree for ρ, there is a function g, defined on s, with these properties:

If f(y) = Pair(1,0), g(y) = Pair(1,0).

If f(y) = Pair(2,n), g(y) = $\ulcorner\tau\urcorner$.

If f(y) = Pair(2,m), with $m \neq n$, g(y) = f(y)

If f(y) = Pair(4,v), g(y) = Pair(4,g(v)).

If f(v) = Triple(5,v,w), g(y) = Triple(5,g(v),g(w)).

If f(v) = Triple(6,v,w), g(y) = Triple(6,g(v),g(w)).

If f(v) = Triple(7,v,w), g(y) = Triple(5,g(v),g(w)).

g(< >) = z.

If θ is an atomic formula, τ a term, and n a number, $\ulcorner\theta^{X_n}/_{\tau}\urcorner$ is given by:

$\ulcorner\rho = \sigma^{X_n}/_{\tau}\urcorner$ = Triple(8,$\ulcorner\rho^{X_n}/_{\tau}\urcorner$,$\ulcorner\sigma^{X_n}/_{\tau}\urcorner$).

$\ulcorner\rho < \sigma^{X_n}/_{\tau}\urcorner$ = Triple(9,$\ulcorner\rho^{X_n}/_{\tau}\urcorner$,$\ulcorner\sigma^{X_n}/_{\tau}\urcorner$).

To complete the proof, we need to do the same thing with formulas that we just did with terms. Given a formula θ, we first find the structure tree for θ – a pair <s,f>, where s is a binary true, and f is a function that labels each node of s with a formula representing the structure of subformulas of θ, which appears at the trunk. Then we define a second function g so that, if f(x) = $\ulcorner\phi\urcorner$, g(x) = $\ulcorner\phi^{X_n}/_{\tau}\urcorner$. Then $\ulcorner\theta^{X_n}/_{\tau}\urcorner$ will be g(< >). I won't give the details, which are tedious and don't involve any new ideas.

The construction just outlined shows that the function taking $\ulcorner\theta\urcorner$ to $\ulcorner\theta^{X_n}/_{\tau}\urcorner$ is Σ. Being a Σ partial function with a Δ domain, it's Δ.⊠

What we'd like to do now is to see how to take proofs and code them arithmetically. The details are complicated, but the idea is simple. A proof is a sequence of expressions, and we know already how to code expressions as a numbers and how to code a sequence of numbers as a single number.

A couple of technical points require attention. The logical system we learned in Logic I required an infinite reservoir of infinite constants. It's not hard to give a system of rules that doesn't need the constants, but it's even easier to expand our system of Gödel numbering to accommodate the extra constants. Where $\mathfrak{L}$ is the language of arithmetic, let $\mathfrak{L}_c$ be the language obtained from $\mathfrak{L}$ by adding infinitely many new individual constants c_0, c_1, c_2, c_3,.... We can extend our system of Gödel numbering by letting $\ulcorner c_n\urcorner$ be Pair(3,n). That's why we skipped pairs

and triples beginning with 3 when we gave our earlier Gödel numbering for $\mathfrak{L}$; we were leaving room for the new constants.

Our deductive calculus from Logic I included bunch of simple rules and one very complicated rule, Tautological Consequence (TC), which permits you to write down any sentence that is either a tautology or a tautological consequence, taking as premiss set the union of the premiss sets of those earlier lines. TC is complex enough that it would be a lot of work to describe its operation arithmetically. Rather than doing so, we can replace TC with a bunch of simpler rules. There are many ways to do this. One method, which is particularly simple and which fits seamlessly with the system of rules we learned in Logic I, is to replace the rule TC with three new rules: *Modus Ponens* (If you've derived ϕ with premiss set Γ and $(\phi \rightarrow \psi)$ with premiss set Δ, you may write ψ with premiss set $\Gamma \cup \Delta$); *Modus Tollens* (If you've derived ϕ with premiss set Γ and $(\neg \psi \rightarrow \neg \phi)$ with premiss set Δ, you may write ψ with premiss set $\Gamma \cup \Delta$); and Definitional Exchange (You may replace $(\phi \vee \psi)$ with $(\neg\phi \rightarrow \psi)$ or *vice versa*; similarly for $(\phi \wedge \psi)$ and $\neg(\phi \rightarrow \neg\psi)$ and for $(\phi \leftrightarrow \psi)$ and $((\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi))$). For a proof that these new rules are a satisfactory replacement for TC, see Benson Mates, *Elementary Logic* (New York: Oxford University Press, 1972). It's not surprising that the Mates' system meshes nicely with the rules from Logic I, since the rules for Logic I were lifted from his book.

Where ϕ is a sentence of $\mathfrak{L}$ and Γ is a Δ set of sentences[1] of $\mathfrak{L}$, a number s is said to be a *proof* of ϕ from Γ just in case s is a sequence of ordered pairs <x,y> with the following properties:

x is a code of a finite set Ω of sentences of $\mathfrak{L}_c$.

y is a code of a sentence ψ of $\mathfrak{L}_c$.

---

1 What this really means is that the set of code numbers of members of Γ is Δ. In the future, we shall frequently efface the distinction between a sentence or set of sentences and its code number. I hope that no confusion results.

> Either ψ is an element of Γ or ψ is an element of Ω (and thus ψ is derived by rule PI) or the inscription of ψ with premiss set Ω is derived from one or more of the earlier members of s by one of the rules other than PI.
>
> The last member of s has ⌜ϕ⌝ as its second component and the code of a subset of Γ as its first.

To spell this out in detail, we would have to specify, rule by rule, what it takes for one line to be derived from an earlier line by a rule. For example <x,y> is derived from <z,w> by rule CP iff there is a v < y such that y = Triple(13,v,w) and, for any u < s, u ∈ z iff (u ∈ x or u = v). Going through the details helps inculcate the virtues of patience and endurance, but it doesn't inspire any intellectual virtues, so we won't do it here.

What we get is a Σ formula B_Γ that strongly represents the relation {<s,⌜ϕ⌝>: s is the code of a proof of ϕ from Γ}[2] in Q. If we define a Σ formula Bew_Γ (from the German "Beweis," for "proof") by:

$$Bew_\Gamma(x) =_{Def} (\exists s)s\ B_\Gamma\ x,$$

we get a formula that weakly represents {x: x is the code of a consequence of Γ} in Q.

In defining "Bew_Γ," we have supposed that Γ is a Δ system of axioms. This looks unnecessarily restrictive. In order to have a proof procedure for the set of consequences of a set of axioms, it's enough to have a proof procedure for the set of axioms; we don't need a decision procedure. To generate the consequences, we need to be reliably able to recognize the axioms;

---

2 In writing out the formula that strongly represents proofs in Γ, we'll use some Σ formula γ(x) to strongly represent to set of axioms of Γ. There are lots of different Σ formulas we could use to strongly represent Γ, and the each choice would give us a different formula to represent the proof-in-Γ relation. In some out-of-the-way corners of logic, this makes a difference, but it won't matter for us here. To be fully explicit, we ought to write "$B_{\gamma(x)}$," rather than "B_Γ," but the mildly ambiguous notation won't do us any harm.

we don't have to be able to recognize the nonaxioms. Thus it would appear that we would benefit from employing a more liberal notion of provability that allowed us to start with a Σ set of axioms, rather than a Δ set. It turns out that this appearance is illusory, because of the following theorem:

> **Craig's Theorem.** Let Γ be a Σ set of sentences. Then there is a Δ set of sentences that has the same consequences as Γ.

Proof: If Γ is the empty set, it's already Δ, and we're done. If Γ is nonempty, it is the range of some Δ total function, call it f. Let Ω = {Triple(15,n,f(n)): n a natural number}.

Ω is Δ. It's obviously Σ. To see that it's Π, note that its complement is {z: 1stin3(z) ≠ 15 or 3rdin3(z) ≠ f(2ndin3(z))}.

The members of Ω are all obtained from members of Γ by prefixing a vacuous universal quantifier. The members of Γ are obtained from members of Ω by deleting a vacuous initial universal quantifier. So Γ and Ω are logically equivalent.⊠