Coding Proofs

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 premise set Γ and $(\phi \rightarrow \psi)$ with premise set Δ, you may write ψ with premiss set $\Gamma \cup \Delta$.

Modus Tollens: If you've derived ϕ with premise set Γ and $(\sim \psi \rightarrow \sim \phi)$ with premise set Δ, you may write ψ with premise set $\Gamma \cup \Delta$.

Definitional Exchange: You may replace $(\phi \vee \psi)$ with $(\neg\phi \rightarrow \psi)$ or *vice versa*, keeping the same premise set. 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 $\mathfrak{L}_c$.

Either ψ is an element of Ω (so that ψ is derivable from Ω by rule PI) or ψ is derivable with premiss set Ω from one or more of the earlier members of s by one of the rules other than PI.

The last member of s has $\ulcorner\phi\urcorner$ 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

---

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.

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 $u < s$, $u \in z$ iff ($u \in x$ 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 $\{<s, \ulcorner\phi\urcorner>$: s is the code of a proof of ϕ from $\Gamma\}$[2] in Q, and hence in any consistent theory that includes 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 and in any other ω-consistent theory that includes 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; 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

---

2 In writing out the formula that strongly represents proofs in Γ, we'll use some Σ formula $\gamma(x)$ 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.

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.⊠