Spring, 2004

Phi 432/532: Completeness & Decidability

We conclude our discussion of completeness by mentioning two simple corollaries of the completeness theorem, either or both of which are referred to as the **compactness theorem**:

Corollary 1: If $\Gamma \vDash \phi$, then there is some finite $\Sigma \subseteq \Gamma$ such that $\Sigma \vDash \phi$.

Proof: By definition, deducibility has this property. Hence, by soundness & completeness, so does logical consequence.

Corollary 2: Γ is semantically consistent iff every finite subset of Γ is semantically consistent.

Proof: Exercise.

Although we have proved completeness only for first order languages without identity, essentially the same method of proof can be used to establish completeness for languages with identity[1] and we'll take the latter result for granted in what follows.

AXIOMATIZATION

Earlier we introduced the notion of axiomatization, and said that a set Γ of L_E-*sentences* axiomatizes a class of L_E-structures $\mathcal{C}$ if:

$$\mathcal{A} \text{ is a model of } \Gamma \text{ iff } \mathcal{A} \in \mathcal{C}$$

We also gave some examples of classes of structures that can be successfully axiomatized.[2] We now want to refine that discussion and consider in more detail the particular problem of axiomatizing arithmetic, i.e. the structure $\mathbb{N}$ of the natural numbers. We have chosen the example of arithmetic because there is a clear sense in which it is fundamental—in mathematics at least. The trend in the nineteenth century was to explain every kind of mathematical entity as a construction out of natural numbers, with the result that the theory of the natural numbers was assigned a foundational role. Although we are less ready today to think in terms of a unique foundation for mathematics, it is still one of the most basic and best understood branches of mathematics, and is included as a part of many others.

So, let $\boldsymbol{E} = \langle \varnothing, \langle \mathbf{s}, \mathbf{+}, \boldsymbol{\times} \rangle, \mathbf{0} \rangle$ and $\mathbb{N} = \langle \boldsymbol{N}, \{\langle \mathbf{0}, 0\rangle, \langle \boldsymbol{s}, s\rangle, \langle \mathbf{+}, +\rangle, \langle \boldsymbol{\times}, \times\rangle\}\rangle$.[3] We remarked earlier that,

there is no set Γ of sentences in the language L_E, which axiomatizes $\{\mathbb{N}\}$.

---

[1]The only significant difference is in the proof of the satisfaction lemma. If the language of Γ contains identity, we take the domain of the model to be, not the terms of the language *per se*, but rather equivalence classes of these terms, where:

$$t \text{ and } u \text{ are members of the same equivalence class iff } \Gamma \vdash t \approx u.$$

[2]Equivalence relations, partial orderings, groups.

[3]We use boldface for non-logical expressions and ordinary type for their intepretations. So, for example, $\nu_{\mathbb{N}}(\mathbf{0}) = 0$.

There is one trivial way in which this may fail, and which we doesn't interest us. Consider, by way of example, the structure <$\boldsymbol{I}^+$, < $\boldsymbol{s}$, s>> comprising the set of positive integers and the successor function, and the structure <$\boldsymbol{I}^-$, < $\boldsymbol{s}$, p>> comprising the set of negative integers and the predecessor function. Although they have disjoint domains, in a sense these are the same structure. What does this mean? There is a 1-1 function f: $\boldsymbol{I}^+ \Rightarrow \boldsymbol{I}^-$ such that, for all i, $f(s(i)) = p(f(I))$. The two structures are said to be **isomorphic**: they are structurally indistinguishable. We can generalize this notion of isomorphism to arbitrary structures. (Zalabardo gives the definition on p.254.) No set of sentences can distinguish between isomorphic structures. So, in asking for an axiomatization of $\{\mathbb{N}\}$, we are asking for a set Γ of sentences such that every model of Γ is *isomorphic to* $\mathbb{N}$.

However, even with this proviso, there is no such set of first order sentences. An easy way to see this is to note that even $\boldsymbol{T}(\mathbb{N})$ fails to axiomatize $\mathbb{N}$ where, for any structure A, $\boldsymbol{T}(\mathsf{A})$ is the set of sentences (of the language of the structure) true in A.. [$\boldsymbol{T}(\mathsf{A})$ is called the **theory of** A.]

Definition: We define (by induction) a set of terms of L_E, the **numerals**, as follows:

1) **0** is a **numeral**
2) if $\boldsymbol{n}$ is a numeral, then so is $\boldsymbol{s(n)}$

Now, let $\boldsymbol{u}$ be a new individual constant and $\Gamma = \{\boldsymbol{u} \neq \boldsymbol{n} \mid \text{for all numerals } \boldsymbol{n}\}$. ($\Gamma$ is a set of sentences in the language obtained from L_E by adding a single new constant $\boldsymbol{u}$.) Clearly, for any finite subset Γ' of Γ, $\boldsymbol{T}(\mathbb{N}) \cup \Gamma'$ is satisfiable. (In fact, it is satisfiable in any expansion of $\mathbb{N}$ which interprets $\boldsymbol{u}$ as a number which is not mentioned by any sentence in Γ'.) It follows by the second formulation of the compactness theorem above that $\boldsymbol{T}(\mathbb{N}) \cup \Gamma$ is satisfiable. But any structure in which all the members of $\boldsymbol{T}(\mathbb{N}) \cup \Gamma$ hold is also a model of $\boldsymbol{T}(\mathbb{N})$.[4] Furthermore, such a structure cannot be isomorphic to $\mathbb{N}$, since whatever element of the universe is used to interpret the constant $\boldsymbol{u}$ cannot be mapped isomorphically onto a finite successor of 0.

This result reveals the expressive limitations of first order languages. Our next observation is that, at the end of the 19^{th} century, $\mathbb{N}$ was successfully axiomatized. (Credit for this is usually given to the Italian mathematician Giuseppi Peano, although Richard Dedekind has a prior claim.)

Definition:
Let $\boldsymbol{Q}$ (Robinson's arithmetic) consist of the following seven sentences:

1) $\forall x \forall y\ (\boldsymbol{s}(x) = \boldsymbol{s}(y) \rightarrow x = y)$
2) $\forall x\ \boldsymbol{0} \neq s(x)$
3) $\forall x\ (x \neq \boldsymbol{0} \rightarrow \exists y\ x = \boldsymbol{s}(y))$
4) $\forall x\ x + \boldsymbol{0} = x$
5) $\forall x \forall y\ x + \boldsymbol{s}(y) = \boldsymbol{s}(x + y)$
6) $\forall x\ x \times \boldsymbol{0} = \boldsymbol{0}$
7) $\forall x \forall y\ x \times \boldsymbol{s}(y) = (x \times y) + x$

The axioms of **Peano Arithmetic**, $\boldsymbol{PA}^2$ for short, are obtained from $\boldsymbol{Q}$ by adding the induction axiom:

$$\forall \boldsymbol{P}\ ((\boldsymbol{P}\boldsymbol{0} \wedge \forall y\ (Py \rightarrow P\boldsymbol{s}(y))) \rightarrow \forall y\ Py)$$

where $\boldsymbol{P}$ is a variable ranging over properties or sets.

---

[4]More precisely, if $\mathbb{N}'$ is a model of $\boldsymbol{T}(\mathbb{N}) \cup \Gamma$, we contract it to an interpretation $\mathbb{N}''$ of the language of $\boldsymbol{T}(\mathbb{N})$ by restricting the interpretation function so that $\boldsymbol{u}$ is no longer in its domain. Then, our remarks in the text above refer to $\mathbb{N}''$.

The predicate variable $\boldsymbol{P}$ is interpreted as ranging over the power set of the domain of a structure. (In the case of $\mathbb{N}$, this means that $\boldsymbol{P}$ ranges over all subsets of natural numbers.) With this semantic convention, it is quite easy to show that any L_E-structure C which is a model of $\boldsymbol{PA}^2$ is isomorphic to $\mathbb{N}$. This means, for all practical purposes, that there is only one structure, namely $\mathbb{N}$, which is a model of $\boldsymbol{PA}^2$.

It might seem that this solves the problem, but it really does not. We are interested in axiomatizing a structure (or class of structures) because we want to know which sentences are true in that structure. In the present case, if every model of $\boldsymbol{PA}^2$ is isomorphic to $\mathbb{N}$, then all and only members of $\boldsymbol{Cn}(\boldsymbol{PA}^2)$, i.e. $\{\phi \mid \boldsymbol{PA}^2 \vDash \phi\}$ are true in $\mathbb{N}$.[5] However, this is of no use to us unless we have a way of determining what sentences belongs to this set. In the case of first order languages, deducibility provides us with some information of this sort. Let's call a set Γ **decidable** if we have an effective method of determining, for any given sentence, whether or not it is a member of Γ. (So, for example, any finite set like $\boldsymbol{PA}^2$ is decidable: list its members and then check each candidate sentence against the list.) Then, if Γ is decidable and ϕ is indeed a member of $\boldsymbol{Cn}(\Gamma)$, we can enumerate derivations in a systematic way and eventually, but in a finite number of steps, arrive at a derivation of a sequent $\Gamma_0 \vdash \phi$, for some finite $\Gamma_0 \subseteq \Gamma$. Note that this does not mean that $\boldsymbol{Cn}(\Gamma)$ is decidable because enumerating derivations does not provide a method which will always yield an answer, whether affirmative or negative, to the question "is ϕ deducible from Γ?". That there can be no such method is a non-trivial result due to Alonzo Church that we will not have time to prove now. (Contrast this situation with the one for propositional logic, where truth tables provide just such a method.)

However, in the case of second order languages, we do not even have this much. In fact, we have no grasp at all of logical consequence for these languages. There is no calculus analogous to **the one given earlier** which captures second-order logical consequence, i.e. no complete set of rules of deduction for second order languages. That is why we deliberately excluded the device of quantifying over predicate variables and confined our attention to first order languages. So, it has become customary to restrict attention to first-order attempts to capture as much as possible of what is expressed by $\boldsymbol{PA}^2$. This means replacing the second-order induction axiom by a schema, i.e. an infinite number of first-order axioms. The result is $\boldsymbol{PA}$, the infinite set of sentences obtained by adding to $\boldsymbol{Q}$ every sentences of the form:

$$((\alpha)[\mathbf{0}/x] \wedge \forall y\ ((\alpha)[y/x] \rightarrow (\alpha)[\boldsymbol{s}(y)/x])) \rightarrow \forall y\ (\alpha)[\mathrm{y}/x]$$

where α is any formula of L_E whose only free variable is x. However, the compactness argument given above shows that the models of $\boldsymbol{PA}$ are not all isomorphic. (I am assuming here that $\boldsymbol{PA}$ is sound for $\mathbb{N}$, i.e. that every member of $\boldsymbol{PA}$ is true in $\mathbb{N}$.)

THE INCOMPLETENESS OF ARITHMETIC

Two structures are said to be **elementarily equivalent** if exactly the same sentences are true in each of them. (Elementary equivalence does not imply isomorphism, although the converse does hold.) Then, trivially, $\boldsymbol{T}(\mathbb{N})$ axiomatizes the class C of structures elementarily equivalent to $\mathbb{N}$. Unfortunately, that tells us nothing about $\mathbb{N}$ that we did not already know. The reason is that $\boldsymbol{T}(\mathbb{N})$ is not a decidable set: it is not even "one-way decidable" like the set $\boldsymbol{Cn}(\Gamma)$ discussed above. Hence, we don't count $\boldsymbol{T}(\mathbb{N})$ as an axiomatization of this class. So, the only remaining question for us to consider is whether we can in fact give a suitable axiomatization of this class C ; in other words, whether there is a decidable set

[5]Suppose that some ϕ which is not a member of $\boldsymbol{Cn}(\boldsymbol{PA}^2)$ is true in $\mathbb{N}$, then $\boldsymbol{PA}^2 \cup \{\neg\phi\}$ will be satisfiable; hence $\boldsymbol{PA}^2$ will have a model which is not isomorphic to $\mathbb{N}$.

Γ of L_E-sentences such that $\boldsymbol{Cn}(\Gamma) = \boldsymbol{T}(\mathbb{N})$. That there can be no such set follows from Gödel's First Incompleteness Theorem.

What the theorem states, in effect, is that for any reasonable extension of $\boldsymbol{Q}$—$\boldsymbol{PA}$ for example—we can construct a sentence ϕ such that neither $\boldsymbol{PA} \vdash \theta$ nor $\boldsymbol{PA} \vdash \neg\theta$. Since, for every L_E-sentence α, either α is true in $\mathbb{N}$ or $\neg\alpha$ is true in $\mathbb{N}$—this is a trivial consequence of the basic semantic definition—it follows that $\boldsymbol{Cn}(\boldsymbol{PA}) \neq \boldsymbol{T}(\mathbb{N})$. What do we mean by '**reasonable**'? That the set of axioms be decidable or, at least, that we can effectively enumerate is members so that, if ϕ say is one of them, we can establish this fact. What follows is a brief sketch of Gödel's argument:

For each m and $i \leq m$, the m-place projection function $p_{m,i} : \boldsymbol{N}^m \Rightarrow \boldsymbol{N}$ defined as follows:

$$p_{m,i}(n_1, \ldots, n_m) = n_i$$

Def: A **primitive recursive** function is a function on natural numbers which is definable from 0, successor and the projection functions using the following two schemata:

1) **substitution** $\quad f(\boldsymbol{x}) = g(h_1(\boldsymbol{x}), \ldots\ h_m(\boldsymbol{x}))$
2) **recursion** $\quad f(\boldsymbol{x}, 0) = g(\boldsymbol{x})$
$\quad f(\boldsymbol{x}, \boldsymbol{s}(y)) = h(\boldsymbol{x}, y, f(\boldsymbol{x}, y))$

(In 1,the value of f is being defined in terms of the values of previously defined functions. In 2, the value of f is being defined in terms of the values of previously defined functions and the value of f at smaller arguments. $\boldsymbol{x}$ is supposed to be an n-tuple of numbers and y a number.)

As we see from the previous section, both addition and multiplication are primitive recursive functions—as are many other familiar number-theoretic functions. Furthermore, it should be clear from the definition that primitive recursive functions are **computable**, whatever one's informal understanding of this term may be. (The converse, however, is not true: there are functions for which we can describe a computation procedure, but which are not primitive recursive.)

Let Φ be any set of sentences of the language L_E that includes $\boldsymbol{Q}$. The primitive recursive functions have an important property which is of particular interest from our point of view:

Let f be any n-place primitive recursive function, then there is a term $\boldsymbol{t}$ of L_E with free variables $x_1, \ldots, x_n$ such that, for all $m_1, \ldots, m_n, m_{n+1}$:

$$f(m_1, \ldots, m_n) = m_{n+1} \text{ iff } \Phi \vdash (\boldsymbol{t})[\boldsymbol{m}_1/x_1] \ldots [\boldsymbol{m}_n/x_n] \approx \boldsymbol{m}_{n+1}$$

and

$$f(m_1, \ldots, m_n) \neq m_{n+1} \text{ iff } \Phi \vdash \neg(\boldsymbol{t})[\boldsymbol{m}_1/x_1] \ldots [\boldsymbol{m}_n/x_n] \approx \boldsymbol{m}_{n+1}$$

(Our convention is that, if m is any number, then $\boldsymbol{m}$ is the numeral of that number.)

If we define a **primitive recursive relation** to be one whose **characteristic function**[6] is primitive recursive, then it follows that, for any n-place primitive recursive relation R, there is a formula α of $\mathfrak{L}$ with free variables $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ such that, for all $m_1, \ldots, m_n$:

$$R(m_1, \ldots, m_n) \text{ is true iff } \Phi \vdash (\alpha)[\boldsymbol{m}_1/x_1] \ldots [\boldsymbol{m}_n/x_n]$$

and

$$R(m_1, \ldots, m_n) \text{ is false iff } \Phi \vdash \neg(\alpha)[\boldsymbol{m}_1/x_1] \ldots [\boldsymbol{m}_n/x_n]$$

---

[6]If R is an n-place relation on a set $\boldsymbol{A}$, the characteristic function of R is the function $f_R: \boldsymbol{A}^n \Rightarrow \{0, 1\}$ whose value is 1 for an n-tuple of arguments which is in the extension of R, and whose value is 0 for an n-tuple of arguments which is not.

This is sometimes expressed by saying that α **strongly represents** (or **binumerates**) R in Φ. It turns out that every quantifier free formula of L_E binumerates a primitive recursive relation in Φ and, conversely, that every primitive recursive relation is binumerated in Φ by a quantifier free formula of L_E.

> **FACT:** There is a one-one function, call it the **coding function**, from formulas of L_E to natural numbers such that, under it, the usual syntactic operations correspond to primitive recursive functions. In this sense we can **arithmetize the syntax** of L_E.

This means, for example, that there is a primitive recursive "conditional function", i.e. a function which, when applied to the codes of α and β , yields as value the code of $(\alpha \rightarrow \beta)$. But it is not just such simple functions as that which are primitive recursive. We also have a "substitution function", i.e. a 2-place function which, when applied to the code of a term **t** and the code of a formula α, with free variable z, yields as value the code of $(\alpha)[t/z]$. Since it is primitive recursive, by the preceding, there is a term of L_E, say **sub**, involving two free variables (x and y, say) such that:

$$\Phi \vdash (\mathbf{sub})[\boldsymbol{m}/x][\boldsymbol{n}/y] \approx \boldsymbol{k}$$

whenever m is the code of a formula α, n is the code of a term t and k is the code of $(\alpha)[t/z]$. But there is no need to stop here. We also have a primitive recursive "conditional elimination function", i.e. one which, when applied to the code of a conditional and the code of its antecedent, yields the code of its consequent.

For ease of exposition, let's call a proof of the sequent $\Phi \vdash \alpha$ a deduction of α from Φ. Then, we can even define a primitive recursive relation:

"*m* is (the code of) a deduction from Φ of the formula (with code) *n*"

How? m is a deduction of n from Φ iff m is the code of a finite tree of ordered pairs, whose left term is a finite set of L_E formulas and whose right term is an L_E-formula, such that the every member of the left term of the ordered pair at the root of the tree is a member of Φ and its right term has code n; furthermore, each pair in the tree is either an axiom or follows from its immediate predecessor(s) by one of the rules of deduction.

The only dubious clause in this description is the one about being a member of Φ. We are assuming that there is a primitive recursive predicate P such that:

$P(m)$ iff m is the code of a member of Φ.

(In fact, we could weaken this condition a little to require only that we have an effective method of generating the codes of members of Φ, i.e. that the set of codes be "one-way decidable" in the sense explained earlier—without affecting the results below, but we shall not take up this issue.)

So, there is a formula, **ded** say, of L_E involving two free variables (x and y, say) which binumerates in Φ the relation "x is a deduction of y from Φ". In terms of it, we can define the property of being deducible from Φ, namely: $\exists x$ **ded**. Notice however that this last formula is not quantifier free, so that the property of being deducible from Φ is *not* primitive recursive. However, it does follow (from the properties of **ded**) that:

Lemma 1: If α is a formula with code n and α is deducible from Φ, then $\Phi \vdash \exists x(\mathbf{ded})[\boldsymbol{n}/y]$.

Proof: If α is deducible from Φ, then it has a deduction with code k, say. Because **ded** binumerates the "__is a deduction from Φ of__" relation in Φ, it follows that: $\Phi \vdash (\mathbf{ded})[\boldsymbol{k}/x][\boldsymbol{n}/y]$. From this, we conclude, by existential introduction, $\Phi \vdash \exists x(\mathbf{ded})[\boldsymbol{n}/y]$. ■

The converse of Lemma1 need not hold however. To go in the opposite direction we would need to infer from

1) $\Phi \vdash \exists x(\mathbf{ded})[\boldsymbol{n}/y]$

that, for some numeral $\boldsymbol{m}$,

2) $\Phi \vdash (\mathbf{ded})[\boldsymbol{m}/x][\boldsymbol{n}/y]$.

FACT: If α is a quantifier free formula involving only the free variable x, then for all n,

$$\Phi \vdash (\alpha)[\boldsymbol{n}/x] \text{ or } \Phi \vdash \neg(\alpha)[\boldsymbol{n}/x]$$

It follows that, to justify the inference from 1) to 2) for some $\boldsymbol{m}$, it is sufficient to verify:

It's not the case that $\Phi \vdash \neg(\mathbf{ded})[\boldsymbol{m}/x][\boldsymbol{n}/y]$.

Definition: Φ is said to be ω-**consistent** if, whenever $\Phi \vdash \exists z\alpha$, there is some n such that it's not the case that $\Phi \vdash \neg(\alpha)[\boldsymbol{n}/z]$

(Clearly, if we intend our quantifier to range over the natural numbers, ω-consistency is a desirable property.) Now, it follows from the fact just cited that, if Φ is ω-consistent, then 1) above implies 2) for some m. Hence, this m codes a derivation of the formula α whose code is n, and α is deducible from Φ. In other words: if if Φ is ω-consistent, then we can prove the converse of lemma 1 above.

Diagonalization Lemma: Let α be an L_E-formula with a single free variable z, then there is some n such that n is the code of a *sentence* β of L_E with the property that:

$$\Phi \vdash (\beta \leftrightarrow (\alpha)[\boldsymbol{n}/z])$$[7]

Proof: Let γ be $(\alpha)[t/z])$, where t is $(\mathbf{sub})[z/x][z/y]$, m be the code of γ and β be $\gamma[\boldsymbol{m}/z]$. We argue *in* Φ as follows:

β is $\gamma[\boldsymbol{m}/z]$, i.e. $(\alpha)[(\mathbf{sub})[m/x][m/y]/z]$. But m is the code of γ, so that $(\mathbf{sub})[\boldsymbol{m}/x][\boldsymbol{m}/y] = n$, where n is the code of β. (Just recall what the "substitution function" is supposed to do.) In other words, β is $(\alpha)[\boldsymbol{n}/z]$, where n is the code of β. ■

If we take the formula α in the statement of the diagonalization lemma to be $\neg\exists x(\mathbf{ded})[z/y]$, we obtain a sentence θ (with code n) such that:

$$\Phi \vdash (\theta \leftrightarrow \neg\exists x(\mathbf{ded})[\boldsymbol{n}/y])$$

It is sometimes said that the sentence θ expresses its own undeducibility from Φ. For this θ, we can establish the following:

Theorem: (*Gödel's First Incompleteness Theorem*)

1) If Φ is consistent, then it's not the case that $\Phi \vdash \theta$.

2) If Φ is ω-consistent, then it's not the case that $\Phi \vdash \neg\theta$.

Proof: Let n be the code of θ.

1) Suppose $\Phi \vdash \theta$, then by lemma 1 above, $\Phi \vdash \exists x(\mathbf{ded})[\boldsymbol{n}/y]$. But, by the diagonalization lemma: $\Phi \vdash (\theta \leftrightarrow \neg\exists x(\mathbf{ded})[\boldsymbol{n}/y])$, contradicting the consistency of Φ.

2) Suppose $\Phi \vdash \neg\theta$ then, by diagonalization again, $\Phi \vdash \neg\neg\exists x(\mathbf{ded})[\boldsymbol{n}/y]$, i.e. $\Phi \vdash \exists x(\mathbf{ded})[\boldsymbol{n}/y]$. Since Φ is assumed to be ω-consistent, we can appeal to the converse of lemma 1 above to conclude that $\Phi \vdash \theta$, contradicting the consistency of Φ (and *a fortiori* the ω-consistency of Φ). ■

---

[7] $(\beta \leftrightarrow (\alpha)[n/z])$ abbreviates $((\beta \rightarrow (\alpha)[n/z]) \wedge ((\alpha)[n/z] \rightarrow \beta))$.

APPENDIX

Zalabardo's substitution notation makes the proof of the diagonalization lemma less perspicuous than it might be. Here's how it would be written if we adopted the following notational convention:

If α is a formula with free variables x and y, say, we write α as $\alpha(x, y)$ and $(\alpha)[t/x][u/y]$ as $\alpha(t, u)$

If t is a term with free variables x and y, say, we write t as $t(x, y)$ and $(t)[s/x][u/y]$ as $t(s, u)$.

Diagonalization Lemma:
For any L_E-formula $\alpha(z)$ there is some n such that n is the code of a *sentence* β of L_E with the property that: $\Phi \vdash (\beta \leftrightarrow \alpha(\boldsymbol{n}))$

Proof: Let γ be $\alpha(t)$), where t is $\mathbf{sub}(z, z)$, m be the code of γ and β be $\gamma(\boldsymbol{m})$. We argue *in* Φ as follows: β is $\gamma(m)$, i.e. $\alpha(\mathbf{sub}(\boldsymbol{m}, \boldsymbol{m}))$. But m is the code of γ, so that $\mathbf{sub}(\boldsymbol{m}, \boldsymbol{m}) = n$, where n is the code of β. (Recall that $\mathbf{sub}(x, y)$ is the code of the formula obtained from the formula with code x by substituting the term with code y for the variable z.) In other words, β is $(\alpha)[\boldsymbol{n}/z]$, where n is the code of β. ■

In fact, we can make things a little clearer (or maybe not) if we adopt a further convention, this time about the notation for codes:

For any formula ϕ, we write the numeral which codes ϕ as $\ulcorner\phi\urcorner$ and, for any term t, we write the numeral which codes t as $\ulcorner t\urcorner$.

Diagonalization Lemma:
For any L_E-formula $\alpha(z)$ there is some *sentence* β of L_E such that: $\Phi \vdash (\beta \leftrightarrow \alpha(\ulcorner\beta\urcorner))$

Proof: Let γ be $\alpha(\ulcorner\mathbf{sub}(z, z)\urcorner)$, and β be $\gamma(\ulcorner\gamma\urcorner)$. We argue *in* Φ as follows: β is $\gamma(\ulcorner\gamma\urcorner)$, i.e. $\alpha(\mathbf{sub}(\ulcorner\gamma\urcorner, \ulcorner\gamma\urcorner))$. But $\mathbf{sub}(\ulcorner\gamma\urcorner, \ulcorner\gamma\urcorner) = \ulcorner\gamma(\ulcorner\gamma\urcorner)\urcorner$, i.e. the code of β. (Recall that $\mathbf{sub}(x, y)$ is the code of the formula obtained from the formula with code x by substituting the term with code y for the variable z.) In other words, β is $\alpha(\ulcorner\beta\urcorner)$. ■
