

Introductory Material

1. Logical Metatheory

Definition: *Logical metatheory* is the use of logical and mathematical methods to study logic or logical systems themselves.

In logical metatheory, rather using a logical system to construct a proof about another subject matter (or an unknown one), we prove things about logic itself. This might include proving things about proving things.

For example, in this course, we will prove that, in first-order predicate logic, we can construct a derivation or proof for every valid argument, and also that every argument we can construct a proof for is valid. These results are called, respectively, *completeness* and *soundness*.

Definition: The *object language* is the language being studied, or the language under discussion.

Definition: The *metalanguage* is the language used when studying the object language.

In this course, the symbolic languages you learned in previous logic courses are our *object languages*.

Our *metalanguage* is plain English, or actually, plain English supplemented by some special symbols and devices.

If the metalanguage, we'll use words like "if", "and", "or", etc., and treat their logic as equivalent to the classical rules for " $\rightarrow$ ", " $\wedge$ ", " $\vee$ ", etc. We assume "if...then..." obeys *modus ponens* and *modus tollens*, for example.

When working in the metalanguage, we will not **use** symbols like " $\rightarrow$ ", " $\wedge$ ", " $\vee$ ", but will **mention** them; typically with quotation marks (though often we get lazy and omit them). The main purpose of the metalanguage is to discuss the object language, so we will do this quite a lot.

We also need variables in the metalanguage where the values of the variables are object-language expressions.

Definition: A *schematic letter* is a metalinguistic variable for object-language expressions, which, when placed inside a representation of an object language expression, is used as part of a device to mention all object-language expressions formed by substituting its possible value in the position in which it appears.

Whether you know it or not, you've seen and used these already. In your Intro to Logic book's sheet of deduction rules, *modus ponens* looked like:

$$\frac{\mathcal{A} \rightarrow \mathcal{B} \quad \mathcal{A}}{\mathcal{B}} \quad \text{or} \quad \frac{\varphi \rightarrow \psi \quad \varphi}{\psi}$$

The cursive or Greek letters are not in the actual object language; they are metalinguistic schematic letters, and this rule schema tells you that *modus ponens* covers not just the case $Fa \rightarrow Ga, Fa \vdash Ga$, but any inference of the right form.

The Open Logic Text uses Greek letters for schematic letters whose values are object-language formulas; we will be following this convention, and adding other kinds of schematic letters and metalanguage variables as we go.

It is important, however, not to mix up object-language expressions and metalanguage ones. We cannot write " $\forall \varphi \dots$ ", because " $\forall$ " is an object language quantifier, and φ a metalanguage variable. Instead, we might write "For every formula φ , ...".

Something like " $\varphi \rightarrow \psi$ " is a hybrid, belonging in neither language cleanly. Some use "Quine corners", e.g., " $\ulcorner \varphi \rightarrow \psi \urcorner$ " to mention not the hybrid, but the resulting expression after assigning values to φ and ψ and inserting them appropriately. We will not, to reserve this notation for something else. Hopefully context will make it clear what is meant.

2. Set Theory

With a very brief exception later in the semester, we will not be studying the logic of set theory as an object language. (Take Philosophy 514, Mathematical Logic II for that!) We will be *using* set theory in the *metalanguage*, however. Consider all the mathematical notation below to be a supplement to English!

Sets

Definition: A **set** is a collection of entities for which it is determined, for every entity of a given type, that the entity either is or is not included in the set.

Definition: An **urelement** is a thing that is not a set.

Definition: An entity A is a **member** of set X iff it is included in that set.

We write this as: " $A \in X$ ". We write " $A \notin X$ " to mean that A is not a member of X .

Sets are **extensional**, or determined entirely by their members: for sets X and Y , $X = Y$ iff for all A , $A \in X$ iff $A \in Y$.

Definition: A **singleton** or **unit set** is a set containing exactly one member.

" $\{A\}$ " means the set containing A alone. Generally, " $\{A_1, \dots, A_n\}$ " means the set containing $A_1, \dots, A_n$, but nothing else.

The members of sets are not ordered, so from $\{A, B\} = \{C, D\}$ one cannot infer that $A = C$, only that either $A = C$ or $A = D$.

Sometimes $\{A : \dots A \dots\}$ is used to name the set of all objects A such that $\dots A \dots$ is true of them, but this notation is potentially dangerous, for reasons we will see later.

Definition: If X and Y are sets, X is said to be a **subset** of Y , written " $X \subseteq Y$ ", iff all members of X are members of Y ; and X is said to be a **proper subset** of Y , written " $X \subset Y$ ", iff all members of X are members of Y , but not all members of Y are members of X .

Definition: If X and Y are sets, the **union** of X and Y , written " $X \cup Y$ ", is the set that contains everything that is a member of either X or Y .

Definition: The **intersection** of X and Y , written " $X \cap Y$ ", is the set that contains everything that is a member of both X and Y .

Definition: The **relative complement** or **difference** of X and Y , written " $X \setminus Y$ ", is the set containing all members of X that are not members of Y .

Definition: The **empty set** or **null set**, written " $\emptyset$ ", " Λ " or " $\{ \}$ ", is the set with no members.

Definition: If X and Y are sets, then they are **disjoint** iff they have no members in common, i.e., iff $X \cap Y = \emptyset$.

Ordered n -tuples and relations

Definition: An **ordered n -tuple**, written " $\langle A_1, \dots, A_n \rangle$ ", is something somewhat like a set, except that the elements are given a fixed order, so that $\langle A_1, \dots, A_n \rangle = \langle B_1, \dots, B_n \rangle$ iff $A_i = B_i$ for all i such that $1 \leq i \leq n$.

An ordered 2-tuple, e.g., $\langle A, B \rangle$ is also called an **ordered pair**. An entity is identified with its 1-tuple.

Definition: If X and Y are sets, then the **Cartesian product** of X and Y , written " $X \times Y$ ", is the set of all ordered pairs $\langle A, B \rangle$ such that $A \in X$ and $B \in Y$.

Generally, " X^n " is used to represent all ordered n -tuples consisting entirely of members of X . Notice that $Y^2 = X \times X$.

The following definition is philosophically problematic, but a common way of speaking in mathematics.

Definition: An **n -place relation (in extension)** on set X is any subset of X^n .

A 2-place relation is also called a **binary relation**. Binary relations are taken to be of sets of ordered pairs. A 1-place relation is also called (the extension of) a **property**.

Definition: If R is a binary relation, then the **domain** of R is the set of all A for which there is an B such that $\langle A, B \rangle \in R$.

Definition: If R is a binary relation, the **range** of R is the set of all B for which there is an A such that $\langle A, B \rangle \in R$.

Definition: The **field** of R is the union of the domain and range of R .

Definition: If R is a binary relation, R is **reflexive** iff $\langle A, A \rangle \in R$ for all A in the field of R .

Definition: If R is a binary relation, R is **symmetric** iff for all A and B in the field of R , $\langle A, B \rangle \in R$ only if $\langle B, A \rangle \in R$.

Definition: If R is a binary relation, R is **transitive** iff for all A, B and C in the field of R , if $\langle A, B \rangle \in R$ and $\langle B, C \rangle \in R$ then $\langle A, C \rangle \in R$.

Definition: A binary relation R is an **equivalence relation** iff R is symmetric, transitive and reflexive.

Definition: If R is an equivalence relation then, the **R -equivalence class on A** , written " $[A]^R$ ", is the set of all B such that $\langle A, B \rangle \in R$.

We sometimes write " $R(A, B)$ " to mean $\langle A, B \rangle \in R$, though this can be dangerously object-languagey.

Functions

Definition: A **function (in extension)** is a binary relation which, for all A, B and C , if it includes $\langle A, B \rangle$ then it does not also contain $\langle A, C \rangle$ unless $B = C$.

So if F is a function and A is in its domain, then there is a unique B such that $\langle A, B \rangle \in F$; this unique B is denoted by " $F(A)$ ".

Definition: An **n -place function** is a function whose domain consists of n -tuples. Then " $F(A_1, \dots, A_n)$ " abbreviates " $F(\langle A_1, \dots, A_n \rangle)$ ".

Definition: An **n -place operation on X** is a function whose domain is X^n and whose range is a subset of X .

Definition: If F is a function, then F is **one-one** iff for all A and B in the domain of F , $F(A) = F(B)$ only if $A = B$.

Cardinal numbers

Definition: If X and Y are sets, then they are **equinumerous**, written " $X \approx Y$ ", iff there is a one-one function whose domain is X and whose range is Y .

Definition: Sets X and Y have the **same cardinality** or **cardinal number** if and only if they are equinumerous.

Definition: If X and Y are sets, then the cardinal number of X is said to be **smaller** than the cardinal number of Y iff there is a set Z such that $Z \subseteq Y$ and $X \approx Z$ but there is no set W such that $W \subseteq X$ and $W \approx Y$.

Definition: If X is a set, then X is **denumerable** iff X is equinumerous with the set of natural numbers $\{0, 1, 2, 3, 4, \dots\}$, (and so on ad inf.).

Definition: **Aleph null/naught/zero**, written " $\aleph_0$ ", is the cardinal number of any denumerable set.

Definition: If X is a set, then X is **finite** iff either $X = \emptyset$ or there is some positive integer n such that X is equinumerous with the set $\{1, \dots, n\}$.

Definition: A set is **infinite** iff it is not finite.

Definition: A set is **countable** iff it is either finite or denumerable.

HOMEWORK 1

Assuming that X, Y and Z are sets, R is a relation, and A and B are any entities, informally verify the following:

- (1) if $X \subseteq Y$ and $Y \subseteq Z$ then $X \subseteq Z$
- (2) $Y \cap \emptyset = \emptyset$ and $Y \cup \emptyset = Y$
- (3) $Y^1 = Y$
- (4) If R is an equivalence relation, then $([A]^R = [B]^R$ iff $\langle A, B \rangle \in R$) and (if $[A]^R \neq [B]^R$ then $[A]^R$ and $[B]^R$ are disjoint).
- (5) $Y \approx Y$
- (6) The set of even non-negative integers is denumerable.

3. Mathematical Induction

This is another tool we'll need in the metalanguage.

Definition: *The principle of mathematical induction states the following:*

If (Φ is true of 0), then if (for all natural numbers n , if Φ is true of n , then Φ is true of $n + 1$), then Φ is true of all natural numbers.

To use the principle mathematical induction to establish that something is true of all natural numbers, one needs to prove the two antecedents, i.e.:

Base step. Φ is true of 0

Induction step. for all natural numbers n , if Φ is true of n , then Φ is true of $n + 1$

Typically, the induction step is proven by means of a conditional proof in which it is assumed that Φ is true of n , and from this assumption it is shown that Φ must be true of $n + 1$. In the context of this conditional proof, the assumption that Φ is true of n is called the **inductive hypothesis**.

A related (and equivalent) principle:

Definition: *The principle of complete (or strong) induction states that:*

If (for all natural numbers n , whenever Φ is true of all numbers less than n , Φ is also true of n) then Φ is true of all natural numbers.

In this class, we often use corollaries specific to the study of logical systems. (My names.)

Definition: *The principle of formula induction states that:*

For a given logical language, if Φ holds of the simplest well-formed formulas (wffs) of that language, and Φ holds of any complex wff provided that Φ holds of those simpler wffs out of which it is constructed, then Φ holds of all wffs.

This is just a version of the above. Let Φ' be the property a number has if and only if all wffs having that number of logical operators have Φ . If Φ is true of the simplest well-formed formulas, then 0 has Φ' . Similarly, if Φ holds of any wffs that are

constructed out of simpler wffs provided that those simpler wffs have Φ , then whenever a given natural number n has Φ' then $n + 1$ also has Φ' . By mathematical induction, all natural numbers have Φ' , i.e., no matter how many operators a wff contains, it has Φ .

This is invoked by proving:

Base step. Φ is true of the simplest well-formed formulas (wffs) of that language; and

Induction step. Φ holds of any wffs that are constructed out of simpler wffs provided that those simpler wffs have Φ .

Definition: *The principle of proof induction:*

In a logical system that contains derivations or proofs, if Φ is true of a given step of the proof whenever Φ is true of all previous steps of the proof, then Φ is true of all steps of the proof.

The principle of proof induction is an obvious corollary of the principle of complete induction applied to line numbers.

HOMEWORK 2

Answer all of these we don't get to in class:

- (1) Let Φ be the property a number x has just in case the sum of all numbers leading up to and including x is $\frac{x(x+1)}{2}$. Use the principle of mathematical induction to show that Φ is true of all natural numbers.
- (2) Let Φ be the property a number x has just in case it is either 0 or 1 or it is evenly divisible by a prime number greater than 1. Use the principle of complete induction to show that Φ is true of all natural numbers.
- (3) Let Φ be the property a wff ψ of propositional logic has if and only if has a even number of parentheses. Use the principle of formula induction to show that ψ holds of all wffs of propositional logic.
- (4) Consider a logical system for propositional logic that has only one inference rule: *modus ponens*. Use the principle of proof induction to show that every line of a proof in this system is true if the premises are true.

4. Propositional Logic and Truth Functions

Past semesters I did an entire unit on this; this year, we're going to skip to metatheory for first-order predicate logic, with this brief exception.

The book uses these object language symbols:

$\neg$	negation	(alternatives $\sim, -$)
$\wedge$	conjunction	(alternatives $\cdot, \&$)
$\vee$	(inclusive) disjunction	
$\rightarrow$	(material) implication	(alternatives $\supset, \Rightarrow$)
$\leftrightarrow$	(material) equivalence	(alternatives $\equiv, \Leftrightarrow$)
$\perp$	absurd/falsehood	(alternatives $\times, \zeta$)
$\top$	triviality/truth	

I assume you are familiar with these.

The language also contains atomic propositional letters $P_1, P_2, \dots, P_n, \dots, Q_1, Q_2, \dots, Q_n, \dots$.

$\perp$ and $\top$ are not connectives but replacements for statement letters that have a fixed interpretation as something false, and true, respectively.

The connectives can be thought of as representing truth-functions. (Negation is one-place; the others two-place.)

Definition: An *n*-place truth function is a function whose domain is $(\{T(ruth), F(alsity)\})^n$ and whose range is a subset of $\{T, F\}$.

For example, we can think of $\vee$ as representing:

$\{\langle\langle T, T \rangle, T \rangle, \langle\langle T, F \rangle, T \rangle, \langle\langle F, T \rangle, T \rangle, \langle\langle F, F \rangle, F \rangle\}$

φ	ψ	$\varphi \vee \psi$
T	T	T

Matching its truth table:

T	F	T
F	T	T
F	F	F

Definition: A formula φ with propositional letters $P_1, \dots, P_n$ represents the *n*-place truth function G iff for each $\langle\langle V_1, \dots, V_n \rangle, U \rangle$ in G if V_i is the truth value P_i for all $i, 1 \leq i \leq n$, then the formula φ has truth value U .

With this definition you can see that $P_1 \vee P_2$ represents the function above.

Because we haven't formulated the syntax/semantics precisely, the following metatheoretic results are handled a bit informally.

Result: Every truth function can be represented by a propositional formula using $\vee, \wedge$ and $\neg$ as its only connectives.

Proof: Let G be an n -place truth function. For each $\langle V_1, \dots, V_n \rangle$ such that $\langle\langle V_1, \dots, V_n \rangle, T \rangle \in G$, consider the conjunction $[\neg]P_1 \wedge [\neg]P_2 \wedge \dots [\neg]P_n$, where a negation is attached to P_i just in case V_i is F; if V_i is T, use P_i without a negation. Let φ be the disjunction of all such conjunctions. (If there are no such conjunctions, this can only be because G always yields falsity, and we can use the self-contradiction $P_1 \wedge \neg P_2$.) It follows that φ represents G . For any given $\langle\langle V_1, \dots, V_n \rangle, U \rangle$, if U is truth, then the corresponding conjunction is true, making the disjunction true. If U is falsity, then every conjunction in φ is false, making φ false. *

To put this back into "English", take any arbitrary final column of a truth table for, say, a formula with three propositional letters.

P_1	P_2	P_3	"Output"
T	T	T	T
T	T	F	F
T	F	T	F
T	F	F	T
F	T	T	T
F	T	F	F
F	F	T	F
F	F	F	F

We represent this with a formula, φ , by disjoining the conjunction corresponding to the "true rows":

$$(P_1 \wedge P_2 \wedge P_3) \vee (P_1 \wedge \neg P_2 \wedge \neg P_3) \vee (\neg P_1 \wedge P_2 \wedge P_3)$$

This formula is true on those rows, and no others. We could do the same for any other truth function.

Corollary: A set of connectives/propositional constants sufficient to define forms equivalent to $\neg, \vee$ and $\wedge$ is adequate to form formulas representing any truth function.

Proof: The equivalent forms may be used in place of $\neg$, $\vee$ and $\wedge$ in the above proof. *

Let $|$ be the Sheffer stroke, meaning “not both”, and $\downarrow$ be the Peirce/Sheffer dagger, meaning “neither...nor...”. The truth tables for these operators are then:

φ	ψ	$\varphi \psi$	φ	ψ	$\varphi \downarrow \psi$
T	T	F	T	T	F
T	F	T	T	F	F
F	T	T	F	T	F
F	F	T	F	F	T

Corollary: *The following sets are each individually adequate to represent all truth functions:*
 $\{\neg, \wedge\}$, $\{\neg, \vee\}$, $\{\neg, \rightarrow\}$, $\{\rightarrow, \perp\}$, $\{\downarrow\}$, $\{\downarrow\}$.

Proof: For $\{\neg, \wedge\}$, note that $\neg(\neg\varphi \wedge \neg\psi)$ is equivalent to $\varphi \vee \psi$, so all of $\neg$, $\vee$ and $\wedge$ can be represented from this set.

Similarly, for $\{\neg, \vee\}$, $\neg(\neg\varphi \vee \neg\psi)$ is equivalent to $\varphi \wedge \psi$.

For $\{\rightarrow, \perp\}$, note that $\varphi \rightarrow \perp$ is equivalent to $\neg\varphi$, and $(\varphi \rightarrow \perp) \rightarrow \psi$ is equivalent with $\varphi \vee \psi$.

For the “not both” Sheffer stroke, $|$, we have:

$\varphi \varphi$	is equivalent to	$\neg\varphi$
$(\varphi \varphi) (\psi \psi)$	is equivalent to	$\varphi \vee \psi$
$(\varphi \psi) (\varphi \psi)$	is equivalent to	$\varphi \wedge \psi$

The others are left as homework. *

Result: *The Sheffer stroke and Peirce/Sheffer dagger are the only binary operators capable of representing all truth functions by themselves, or more exactly, any such binary operator must represent the same truth function as one or the other of them.*

Proof: (1) Suppose $*$ were such a binary operator.

(2) It cannot be that $\varphi * \psi$ is true when φ and ψ are both true; or else a statement using only true propositional letters formed using it alone can only be true, and negations and self-contradictions will be undefinable.

(3) For similar reasons it cannot be that $\varphi * \psi$ is false when φ and ψ are false, or any statement using only false propositional letters formed using it alone can only be false; and negations, tautologies, etc., will be undefinable.

(4) This means we know at least this much about its truth table:

φ	ψ	$\varphi * \psi$
T	T	F
T	F	?
F	T	?
F	F	T

This leaves only four possibilities for the two unknowns (?’s).

(5) Filling it in T and then F makes $\varphi * \psi$ equivalent with $\neg\psi$, but negation alone can only define two truth-functions: negation and double negation; iterating will just alternate between them.

(6) Filling it in F and then T makes $\varphi * \psi$ equivalent with $\neg\varphi$, which is inadequate for the same reason.

(7) Filling it in with T T gives the same function as $|$ and filling it in with F F gives $\downarrow$; therefore, $*$ must represent the same function that one of them does. *

HOMEWORK 3

(1) Prove that $\{\neg, \rightarrow\}$ and $\{\downarrow\}$ are adequate to represent all truth functions.

(2) Give an example of a 3-place truth functional operator that by itself is adequate to represent all truth functions.

It would be possible to take a smaller set of propositional connectives as “primitive” as regard the others as “defined” from them. This would even simplify the deductive systems built on top of them. The Open Logic Text does not do this, however, and uses $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\perp$ and $\top$ all as equally basic parts of its symbolism.

Unit 1

Metatheory for First-Order Predicate Logic

1. Syntax

To do metatheory properly, the exact grammar of your language needs to be spelled out precisely. I allow myself some use/mention sloppiness.

Basic building blocks

Definition: A *logical connective* or *logical operator* is one of the symbols $\neg, \vee, \wedge, \rightarrow, \leftrightarrow, \forall, \exists$.

Definition: A *propositional constant* is either of the symbols $\perp$ or $\top$.

Definition: A *variable* is one of the lowercase letters near the end of the alphabet, v, w, x, y , or z , written with or without a numerical subscript:

Examples: $x, x_1, x_{12}, y, y_2, z, z_{13}$, etc.

Important convention: I use the unitalicized letters “ x ”, “ y ”, etc., as object language variables, and italicized letters in the same range—“ x ”, “ y ”, “ z ”—as metalinguistic schematic letters for *any* object-language variables. Thus, e.g., the **schema**:

$$\forall x(Fx \rightarrow Gx)$$

Represents all of:

$$\forall x(F^1x \rightarrow G^1x)$$

$$\forall y(D^1y \rightarrow E^1y)$$

$$\forall x_3(L_2^1x_3 \rightarrow M_3^1x_3), \text{ etc.}$$

The difference is subtle, and usually not so important to keep straight. After all, object language variables tend to be interchangeable.

The book’s convention on this is not entirely clear, but it seems to use sans serif v_n for object language variables, and italic letters schematically, but this is not as clear as one would like.

Definition: An *constant* is one of the lowercase letters early in the alphabet a to d , written with or without a numerical subscript.

Examples: a, a_2, b, c_{124} , etc.

Again, italicized a – d are schematic.

Definition: A *predicate* is one of the uppercase letters from A to T , written with a numerical superscript ≥ 1 , and with or without a numerical subscript, or the identity predicate $=^2$.

Examples: $A^1, R^2, H^4, F_2^1, G_4^3$, etc.

The superscript indicates how many terms the predicate takes to form a statement. A predicate with a superscript 1 is called a **monadic** predicate. A predicate with a superscript 2 is called a **binary** or **dyadic** predicate.

It is customary to leave these superscripts off when it is obvious from context what they must be. E.g., “ $R^2(a, b)$ ” can be written simply “ $R(a, b)$ ”.

Definition: A *function letter* is one of the lowercase letters from f to l , written with a numerical superscript ≥ 1 , and with or without a numerical subscript.

Examples: f^1, g^2, h_3^3 , etc.

The numerical superscript indicates how many **argument places** the function letter has. Again, obvious superscripts can be left off.

Constructed expressions

Definition: A *term* of the language is defined recursively as follows:

- (i) all variables are terms;
- (ii) all constants are terms;
- (iii) if f is a function letter with superscript n , and $t_1, \dots, t_n$ are terms, then $f(t_1, \dots, t_n)$ is a term;
- (iv) nothing that cannot be constructed by repeated applications of the above is a term.

Examples: $a, x, f(a), g(x, f(y))$, etc.

I use italicized lowercase letters t, t_1, t_2, u, u_1 , etc., schematically for any terms, simple or complex. Unitalicized t or u is never used.

Definition: A *ground term* is a term that does not contain variables (only constants and function letters).

Definition: An *atomic formula* is a propositional constant or any expression of the form $F(t_1, \dots, t_n)$ where F is a predicate with superscript n , and $t_1, \dots, t_n$ are all terms.

Examples: $\perp, \top, F^1(a), R(y, b), F^1(f(x)), R_4^3(a, b, c), H^4(x, b, y, g(a, x))$, etc.

Atomic formulas are those that do not have other formulas as parts. We can leave out parentheses after predicates and commas between their terms if no function letters are used, e.g. “ Fa ” for “ $F(a)$ ” and “ Rxy ” for “ $R(x, y)$ ”.

Definition: A (*well-formed*) *formula (wff)* is recursively defined as follows:

- (i) any atomic formula is a wff;
- (ii) if φ is a wff, then $\neg\varphi$ is a wff;
- (iii) if φ and ψ are wffs, then $(\varphi \vee \psi), (\varphi \wedge \psi), (\varphi \rightarrow \psi)$ and $(\varphi \leftrightarrow \psi)$ are wffs;
- (iv) if φ is a wff and x is a variable, then $\forall x\varphi$ and $\exists x\varphi$ are wffs;
- (v) nothing that cannot be constructed by repeated applications of the above is a wff.

If you prefer $(\forall x)$ or even (x) to $\forall x$, that’s fine; I might not notice.

As a shorthand convenience, we allow ourselves to omit parentheses with the convention that if things are otherwise ambiguous, the operators earlier on this list are to have wider scopes than those later on the list:

$$\leftrightarrow, \rightarrow, \vee, \wedge, \forall x, \exists x, \neg$$

Hence:

$$\forall x(Fx \wedge Gx \rightarrow \neg \exists y Fy \vee Gb)$$

Should be interpreted to mean:

$$\forall x((Fx \wedge Gx) \rightarrow ((\neg \exists y Fy) \vee Gb))$$

Repeated operators of the same type associate to the left, so if we write:

$$Fa \wedge Fb \wedge Gc \vee Ga \vee Rab$$

This means:

$$(((Fa \wedge Gb) \wedge Gc) \vee Ga) \vee Rab$$

We also allow other styles $[\dots] \{ \dots \}$ to be used if it helps find the matching parentheses.

Definition: A *first-order language* $\mathcal{L}$ is any language that makes use of the above definition of a formula, or modifies it at most by restricting which constants, function letters and predicates are utilized (provided that it uses at least one predicate letter).

E.g., a language that does not use function letters still counts as a first-order language. So would a language that used no constants and only a single predicate, say, a set theory whose only predicate was for the membership relation.

Free and bound variables

Definition: When a quantifier $\forall x$ or $\exists x$ occurs as part of a wff φ , the *scope* of the quantifier is defined as the smallest part $\forall x\psi$ or $\exists x\psi$ of φ such that $\forall x\psi$ or $\exists x\psi$ is itself a wff.

Definition: If x is a variable that occurs within a wff φ , then an occurrence of x in φ is said to be a **bound occurrence** iff it occurs in the scope of a quantifier $\forall x$ or $\exists x$ (using x) within φ ; otherwise, the occurrence of x is said to be a **free occurrence**.

Examples: 1. All three occurrences of “ x ” in “ $\forall x(Fx \rightarrow Fx)$ ” are bound.
2. The (solitary) occurrence of “ x ” in “ $Fx \rightarrow \forall y Gy$ ” is free.

Definition: A variable x that occurs within a wff φ is said to be a **bound variable**, or simply **bound**, iff there is at least one bound occurrence of x in φ .

Definition: A variable x that occurs within a wff φ is said to be a **free variable**, or simply **free**, iff there is at least one free occurrence of x in φ .

Notice that “ x ” is both bound and free in “ $Fx \rightarrow \forall x Gx$ ”, because some occurrences are bound and one is free.

Definition: A wff φ is said to be **closed** iff φ contains no free variables; otherwise φ is said to be **open**.

Definition: A **sentence** is a closed wff.

Definition: If φ is a wff, t is a term and x is a variable, then t is **free for x** in φ iff no free occurrence of x in φ lies within the scope of some quantifier $\forall y$ or $\exists y$ where y is a variable occurring in t .

Basically, this means that if you substitute t for all the free occurrences of x within φ , you won’t end up inadvertently “binding” any of the variables that happen to occur in t .

Examples:

1. “ a ” is free for “ x ” in “ $\forall y Gy \rightarrow Gx$ ”.
2. “ $f^2(x, z)$ ” is free for “ x ” in “ $\forall y Gy \rightarrow Gx$ ”.
3. “ z ” is *not* free for “ x ” in “ $\forall y Gy \rightarrow \forall z Rxz$ ”.
4. “ $f^2(a, z)$ ” is not free for “ x ” in “ $\exists y Gy \rightarrow \forall z Rxz$ ”.
5. “ $f^2(a, z)$ ” is free for “ x ” in “ $\exists y Gy \rightarrow \forall z \forall x Rxz$ ”.
6. All terms are free for “ x ” in “ $\forall y Gy$ ”.
7. All terms are free for “ y ” in “ $\exists y Gy$ ”.

If “ φ ” is being used in the metalanguage schematically for a formula, and t for a term free for variable x in φ , then the notation “ $\varphi[t/x]$ ” means the formula that results when all free occurrences of x are replaced by the term t ; this notation is part of the metalanguage only.

Although less precise, sometimes we shall use “ $\varphi(x)$ ” schematically for a formula that is expected to have x free in it; and if we also use “ $\varphi(t)$ ” in that context, then what is meant is also what results by replacing all free occurrences of x with t . Of course, however, “ φ ” here should not be thought of as predicate, or even a schematic letter for a predicate.

Notice in each case that only free variables are replaced, so if $\varphi(x)$ is “ $Gx \rightarrow \forall x Fx$ ”, then $\varphi(y)$ is “ $Gy \rightarrow \forall x Fx$ ”, not “ $Gy \rightarrow \forall y Fy$ ”.

The book uses a symbol $\equiv$ as the metalinguistic relation of syntactic identity; I avoid this, to reduce possible confusion for those accustomed to using this symbol for material equivalence in the object language.

HOMEWORK 4

- (1) Determine which occurrences of variables are bound and which are free in the following wffs:
 $\forall x \exists y (Rxy \rightarrow Syz)$
 $\forall x (\exists y Ryx \leftrightarrow Ryx)$
 $\forall x \neg \exists y Fyx \rightarrow \exists z (Fz \wedge Rxz)$
- (2) For each of the formulas φ on the left, determine, for each term t on the right, which are and which are not free for “ x ” in φ :

formulas	terms
Fx	y
Fa	$f(x)$
$\forall y Ryx$	$f(x, y)$
$\forall y \exists x Ryx$	$f(x, a)$
$\exists y (Fx \wedge Fy)$	$f(a, y, g(x))$

2. Semantics

Structures

Definition: A **structure** or **interpretation** $\mathfrak{M}$ for a first order language $\mathcal{L}$ consists of the following four things:

- (1) *The specification of some non-empty set, written $|\mathfrak{M}|$ to serve as the **domain of quantification** for the language.*

This set is the sum total of entities the quantifiers are interpreted to “range over”.

- (2) *A function assigning, for each constant in the language, some fixed member of $|\mathfrak{M}|$ for which it is taken to stand.*

For a given constant c , this member is denoted in the metalanguage by $c^{\mathfrak{M}}$.

- (3) *A function assigning, for each predicate F with superscript n in the language, some subset of $|\mathfrak{M}|^n$, written $F^{\mathfrak{M}}$.*

That is, the structure assigns to each predicate a set of n -tuples from $|\mathfrak{M}|$, its “extension”.

- (4) *An assignment, for each function letter f with superscript n in the language, some n -place operation on $|\mathfrak{M}|$, written $f^{\mathfrak{M}}$.*

You can take “interpretation” here pretty literally; a structure is basically an assignment of a universe of discourse to the quantifiers, and an appropriate semantic value to each non-logical constant of the language.

Variable assignments and satisfaction

Definition: A **variable assignment**, s , for interpretation $\mathfrak{M}$, is a function that maps every variable of the language of $\mathfrak{M}$ to a member of $|\mathfrak{M}|$.

There can be (and unless $\mathfrak{M}$ has only a single member always are) many different variable assignments for any given structure.

E.g., if $|\mathfrak{M}|$ is {John, Paul, George, Ringo}, then there is one that maps every variable to John, one that maps every variable to Ringo, one that maps “x” to John and the rest to Ringo, etc.

Suppose we have an open formula, “Gx”, meaning, e.g., “x is a guitarist”. A given interpretation (structure)—such as the one that makes $G^{\mathfrak{M}}$ the set {John, George}—does not say that this is true or false, because it depends on “x”. But we say the formula is *satisfied* when “x” is assigned to George.

Technically, we make satisfaction a relationship between a variable assignment and a formula (open or closed). A variable assignment mapping “x” to

George satisfies “Gx” on this interpretation, but one mapping “x” to Ringo will not. The same variable assignment that satisfies “Gx” might not satisfy “Gy” if it maps “y” to Ringo.

Definition: The **value** of term t for variable assignment s for structure $\mathfrak{M}$, written $\text{Val}_s^{\mathfrak{M}}(t)$, is defined recursively as follows:

- (1) If t is a constant c , then $\text{Val}_s^{\mathfrak{M}}(t)$ is $c^{\mathfrak{M}}$;
- (2) if t is a variable x , then $\text{Val}_s^{\mathfrak{M}}(t)$ is $s(x)$;
- (3) if t is a function term $f(t_1, \dots, t_n)$, then $\text{Val}_s^{\mathfrak{M}}(t)$ is $f^{\mathfrak{M}}(\text{Val}_s^{\mathfrak{M}}(t_1), \dots, \text{Val}_s^{\mathfrak{M}}(t_n))$, i.e., the value of the function assigned to f in the structure for the values of the argument terms as arguments.

Definition: The notion of a variable assignment s in structure $\mathfrak{M}$ **satisfying** a given formula φ , written $\mathfrak{M}, s \models \varphi$, is defined recursively. For a given variable assignment s in structure $\mathfrak{M}$:

- (i) $\mathfrak{M}, s \models \top$, always holds; $\mathfrak{M}, s \models \perp$ never holds. (We can write this as $\mathfrak{M}, s \models \perp$.)

I.e., $\top$ and $\perp$ are, respectively, satisfied by every assignment for every structure, and satisfied by no assignment for any structure.

- (ii) If φ is an atomic wff of the form $F(t_1, \dots, t_n)$, then $\mathfrak{M}, s \models \varphi$ iff $\langle \text{Val}_s^{\mathfrak{M}}(t_1), \dots, \text{Val}_s^{\mathfrak{M}}(t_n) \rangle \in F^{\mathfrak{M}}$.

I.e., atomic wffs are satisfied when the values of their terms on the assignment are members of the extension the structure assigns to the predicate.

- (iii) If φ is of the form $\neg\psi$, then $\mathfrak{M}, s \models \varphi$ iff $\mathfrak{M}, s \not\models \psi$.

I.e., an assignment satisfies a negation just in case it does not satisfy what it negates.

- (iv) If φ is of the form $\psi \vee \theta$, then $\mathfrak{M}, s \models \varphi$ iff either $\mathfrak{M}, s \models \psi$ or $\mathfrak{M}, s \models \theta$.

I.e., an assignment satisfies a disjunction iff it satisfies either of the two disjuncts.

- (v) If φ is of the form $\psi \wedge \theta$, then $\mathfrak{M}, s \models \varphi$ iff both $\mathfrak{M}, s \models \psi$ and $\mathfrak{M}, s \models \theta$.

I.e., an assignment satisfies a conjunction iff it satisfies both conjuncts.

- (vi) If φ is of the form $\psi \rightarrow \theta$, then $\mathfrak{M}, s \models \varphi$ iff either $\mathfrak{M}, s \not\models \psi$ or $\mathfrak{M}, s \models \theta$.

I.e., an assignment satisfies a conditional iff it does not satisfy the antecedent, or does satisfy the consequent.

- (vii) If φ is of the form $\psi \leftrightarrow \theta$, then $\mathfrak{M}, s \models \varphi$ iff either both $\mathfrak{M}, s \models \psi$ and $\mathfrak{M}, s \models \theta$ or both $\mathfrak{M}, s \not\models \psi$ and $\mathfrak{M}, s \not\models \theta$.

I.e., an assignment satisfies a biconditional iff it either satisfies both sides, or it satisfies neither side.

- (viii) If φ is of the form $\forall x \psi$, then $\mathfrak{M}, s \models \varphi$ iff every variable assignment s' differing from s by at most its value for variable x (including s itself) satisfies ψ .

- (ix) If φ is of the form $\exists x \psi$, then $\mathfrak{M}, s \models \varphi$ iff at least one variable assignment s' differing from s by at most its value for variable x (including s itself) satisfies ψ .

To understand the last two clauses, let $|\mathfrak{M}|$ again be {John, Paul, George, Ringo} and let $G^{\mathfrak{M}}$ be {John, George}. Consider variable assignment s that assigns every variable to John. There are three other variable assignments that differ from it by at most what it assigns to “ x ”:

variable	s	s'	s''	s'''
v	John	John	John	John
w	John	John	John	John
x	John	Paul	George	Ringo
y	John	John	John	John
z	John	John	John	John
v_1	John	John	John	John
w_1	John	John	John	John
x_1	John	John	John	John
y_1	John	John	John	John
z_1	John	John	John	John
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

s satisfies “ Gx ” because s ’s value for “ x ”, viz., John, is a guitarist. But for s to satisfy “ $\forall x Gx$ ”, this means that not only s , but also s' , s'' and s''' all would have to satisfy “ Gx ”, but s' and s''' do not. Therefore not even s satisfies “ $\forall x Gx$ ”. However, for s to satisfy “ $\exists x Gx$ ”, only one or more of these has to satisfy “ Gx ”; since s itself satisfies “ Gx ”, s also satisfies “ $\exists x Gx$ ”.

HOMEWORK 5

- (1) Does s satisfy “ $Gx \rightarrow \forall x Gx$ ”? Explain your answer.
 (2) What about s' , s'' and s''' ? Again, explain your answers.

Some important results are noted here. We are not going to bother with proofs for all of them, as hopefully they are intuitively obvious.

Result:

Modus ponens preserves satisfaction, i.e., for all φ , ψ , $\mathfrak{M}$ and s for $\mathfrak{M}$, if $\mathfrak{M}, s \models \varphi$ and $\mathfrak{M}, s \models \varphi \rightarrow \psi$ then $\mathfrak{M}, s \models \psi$.

Proof: Assume both $\mathfrak{M}, s \models \varphi$ and $\mathfrak{M}, s \models \varphi \rightarrow \psi$. Because of the definition of satisfaction for formulas of the form $\varphi \rightarrow \psi$, the second assumption means that either $\mathfrak{M}, s \not\models \varphi$ or $\mathfrak{M}, s \models \psi$; the first option is ruled out by the first assumption, so $\mathfrak{M}, s \models \psi$. *

Result: If s and s' are two variable assignments for structure $\mathfrak{M}$, φ is a formula, and $s(x) = s'(x)$ for every variable x that occurs free in φ , then $\mathfrak{M}, s \models \varphi$ iff $\mathfrak{M}, s' \models \varphi$.

Result: If s and s' are two variable assignments for structure $\mathfrak{M}$, φ is a formula, and $s(x) = s'(x)$ for every variable x that occurs free in φ , then $\mathfrak{M}, s \models \varphi$ iff $\mathfrak{M}, s' \models \varphi$.

Corollary: If s and s' are two variable assignments for structure $\mathfrak{M}$, φ is a closed formula/sentence, then $\mathfrak{M}, s \models \varphi$ iff $\mathfrak{M}, s' \models \varphi$ always.

Truth, falsity and models

Definition: A wff φ is **true** in structure $\mathfrak{M}$ iff for every variable assignment s for $\mathfrak{M}$ it holds that $\mathfrak{M}, s \models \varphi$; we write this simply as $\mathfrak{M} \models \varphi$.

Definition: A wff φ is **false** in structure $\mathfrak{M}$ iff $\mathfrak{M} \models \neg\varphi$. (I.e., falsity is truth of the negation.)

HOMEWORK 6

- (1) Give an example of a formula φ and a structure $\mathfrak{M}$ where φ is neither true nor false in $\mathfrak{M}$, and explain why.
- (2) Prove that every sentence φ is either true or false in every structure $\mathfrak{M}$. (You may use the corollary above.)

Definition: If $\mathfrak{M}$ is a structure and Γ a set of formulas, $\mathfrak{M}$ is a **model of** Γ , iff, for all $\varphi \in \Gamma$, $\mathfrak{M} \models \varphi$; we abbreviate this by writing $\mathfrak{M} \models \Gamma$.

By now, we are using the sign $\models$ in a number of different ways, and there are more to come! Note that all uses of this sign are metalinguistic. It cannot be used in the object language, and certainly not embedded inside an object language statement such as “ $Fa \rightarrow (Ga \models Fa)$ ” or something like that.

Notice that every structure will be a model for some sets of wffs. In fact, the study of formal semantics for artificial languages is sometimes called “model theory”. I tend to use the words “structure”, “model” and “interpretation” interchangeably.

Logical validity and related notions

Definition: A wff φ is said to be **logically true** or **(logically) valid** iff φ is true in every structure $\mathfrak{M}$. This can be abbreviated by writing simply $\models \varphi$.

Just like we can leave off the s before the $\models$ when the formula is satisfied by every assignment in a given structure, we can leave off both the s and the $\mathfrak{M}$ when the formula is satisfied by every assignment in every structure.

Definition: A set of sentences Γ **entails** formula φ iff φ is true in every structure that is a model of Γ , i.e., for every $\mathfrak{M}$, if $\mathfrak{M} \models \Gamma$ then $\mathfrak{M} \models \varphi$. We also write this as $\Gamma \models \varphi$.

We also say that φ is a **logical consequence** of Γ iff Γ entails φ .

If Γ has a single member, ψ , we may write simply $\psi \models \varphi$ and say that ψ **logically implies** φ ; or for any finite number of members list them as $\psi_1, \dots, \psi_n \models \varphi$.

Definition: A set of sentences Γ is **satisfiable** iff there is at least one structure $\mathfrak{M}$ such that $\mathfrak{M} \models \Gamma$ and **unsatisfiable** otherwise.

Result: If φ is a sentence, $\models \varphi$ iff $\{\neg\varphi\}$ is unsatisfiable.

Proof: (1) First suppose that $\models \varphi$ but for *reductio* suppose that $\{\neg\varphi\}$ is satisfiable. There is then some $\mathfrak{M}$ such that $\mathfrak{M} \models \neg\varphi$. For every s in $\mathfrak{M}$, $\mathfrak{M}, s \models \neg\varphi$ and by the definition of satisfaction, $\mathfrak{M}, s \not\models \varphi$. But this contradicts the fact that $\mathfrak{M} \models \varphi$ for every $\mathfrak{M}$.

(2) For the other direction, suppose that $\{\neg\varphi\}$ is unsatisfiable. Then there is no $\mathfrak{M}$ such that $\mathfrak{M} \models \neg\varphi$. I.e., for every $\mathfrak{M}$ there is an s for which $\mathfrak{M}, s \not\models \neg\varphi$, and so $\mathfrak{M}, s \models \varphi$. Since φ is a sentence, the same holds for every other assignment s in $\mathfrak{M}$, and hence $\mathfrak{M} \models \varphi$. Since this is true for every $\mathfrak{M}$, it follows that $\models \varphi$. *

Result: For any sets of sentences Γ and Δ , and every sentence φ , if $\Gamma \subseteq \Delta$ and $\Gamma \models \varphi$ then $\Delta \models \varphi$.

Proof: Assume that $\Gamma \subseteq \Delta$ and $\Gamma \models \varphi$. Let $\mathfrak{M}$ be any interpretation that makes every member of Δ true; since Γ is a subset, this means that $\mathfrak{M}$ makes every member of Γ true as well. Because $\Gamma \models \varphi$ this means that $\mathfrak{M} \models \varphi$. Since $\mathfrak{M}$ is arbitrary, this means that $\Delta \models \varphi$. *

HOMEWORK 7

- (1) Prove that $\{\perp\}$ is unsatisfiable.
- (2) Prove that for every set of sentences Γ , $\Gamma \models \perp$ iff Γ is unsatisfiable.

Definition: Two formulas φ and ψ are **logically equivalent** iff in every structure $\mathfrak{M}$, φ and ψ are satisfied by the same variable assignments.

Result: If φ and ψ are sentences, they are logically equivalent if and only if they are true in the same structures.

- Proof:* (1) Suppose φ and ψ are equivalent, and let φ be true in a structure $\mathfrak{M}$. Hence every variable assignment satisfies φ , and because ψ is equivalent, every variable assignment satisfies ψ , and so ψ is true in $\mathfrak{M}$. The same is true in reverse, and so φ and ψ are true in the same structures.
- (2) For the reverse, suppose φ and ψ are true in the same structures. Let s be any variable assignment in any structure $\mathfrak{M}$. If s satisfies φ , all other assignments do as well, because φ is a sentence. Hence, φ is true, and so is ψ , and s satisfies ψ as well. If s does not satisfy φ , then φ is not true, and neither is ψ . Hence some variable assignment s' does not satisfy ψ in $\mathfrak{M}$; because ψ is closed, neither does s . Therefore, φ and ψ are satisfied by the same variable assignments in all structures, and are logically equivalent. *

Tableaux and semantics

Strictly speaking, tableaux or “logic trees” can be considered a form of deductive system or derivation system. Indeed, the Open Logic Text distinguishes four kinds of derivation systems:

- Natural deduction systems (There are probably what you learned in your introductory classes; lots of rules with techniques such as indirect and conditional proofs.)
- Sequent calculi (Kind of weird; rules are stated in terms of left and right sequences where each step means that the truth of all the left formulas requires the truth of at least one right formula.)
- Tableaux (Tree like structures with branching rules tracking the semantic rules of the language.)
- Axiomatic systems (We’ll study these in a few days; they are the focus of most metatheory.)

If we were doing propositional logic, we could add “truth tables” to this list. Are truth tables a deductive system? Technically, yes, but they are designed to track the semantic rules for truth functions. Because of this they can be used to demonstrate the *invalidity* of an argument, or non-tautologyhood of a formula.

Tableaux are similar, and the closest we come to a “decision procedure” for predicate logic. Unlike natural deduction and axiomatic derivation systems, they are at least sometimes effective at helping us find a structure that demonstrates the *invalidity* of a sentence or argument.

The method:

- (1) A tableau takes the form of a downward growing tree representing a hypothetical structure. At the base are a number of “assumptions”, which are $\mathbb{T}$ or $\mathbb{F}$ written before a sentence, representing the truth or falsity, respectively, of the sentence in question in the hypothetical structure.
- (2) For each sentence on the branch, the tableau rule for it is applied, and can be “checked off” when the rule is applied. (In the case of true universals and false existentials, this checking off is provisional, more on this below.)
- (3) Some rules create branches, both of which need to be “explored”. The branches represent different possible ways the hypothetical structure could be.
- (4) A branch closes if it makes the same sentence both true and false. We mark it closed with $\otimes$.
- (5) If the rules are correctly applied to each statement on the branch, no rule reset a true universal or false existential checkmark, and it remains open, the open branch describes a structure consistent with the opening assumptions.

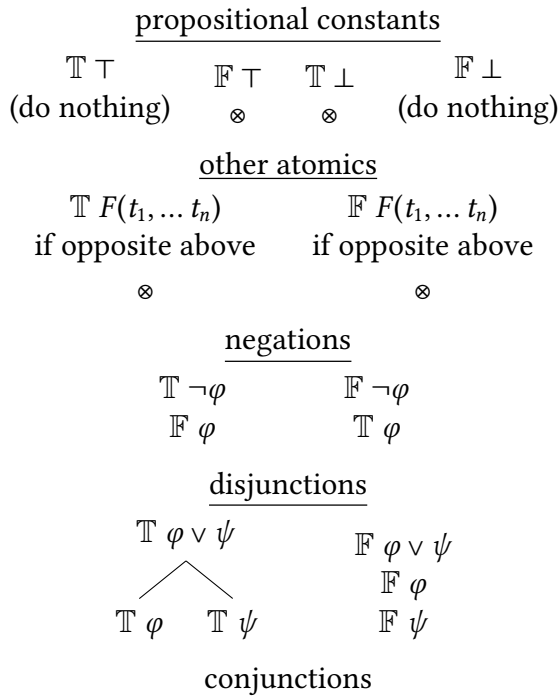
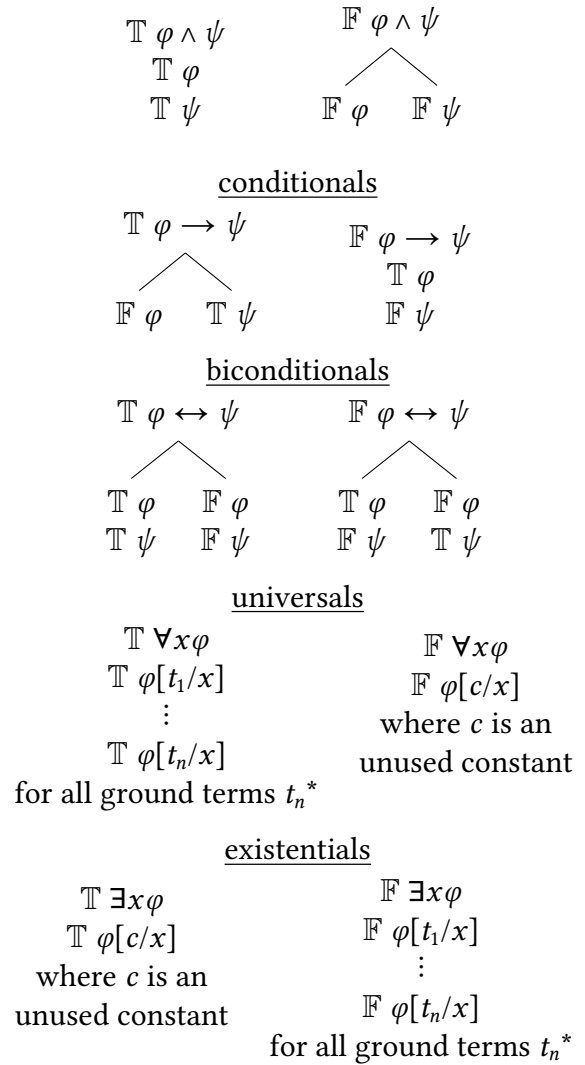
- (6) To avoid loops of steps preventing progress, do true existentials and false universals as soon as possible, but delay re-doing true universals or false existentials afterwards until they are all that is left.
- (7) It is still possible, however, to get stuck in an infinite loop of steps, which is what prevents this from being a full decision procedure for predicate logic.

Uses for tableaux:

- Testing if a formula is valid by looking for a structure in which it is false.
- Testing if a set of formulas is satisfiable by looking for a structure in which they are all true.
- Testing if a formula is entailed by a set of formulas by looking for a structure in which the members of the set are true, but the alleged entailment is false. (I.e., testing if an argument is valid.)

Tableaux rules

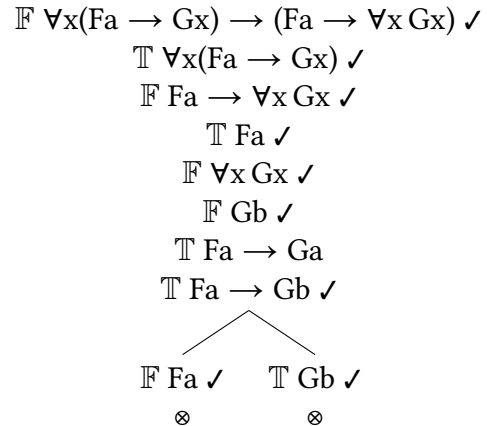
For each category of wff, there is $\mathbb{T}$ rule and an $\mathbb{F}$ rule. What you have is the first line; what you get is below that. Rules must be applied to all open branches below.



* = Do at least one, using the term “a” if there are no ground terms yet. Since new ground terms may appear later on the branch, these rules may need to be reapplied later, and so no checkmark is “final”.

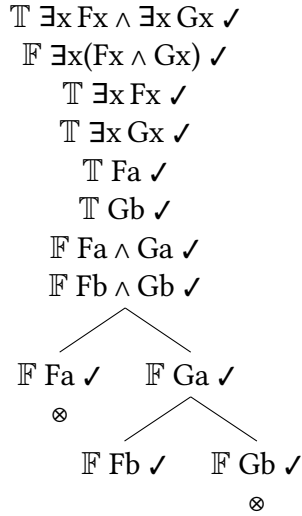
Example (1):

Validity of $\forall x(Fa \rightarrow Gx) \rightarrow (Fa \rightarrow \forall x Gx)$.



No possible structure can make good on our initial assumption, and so all must make this sentence true; it is therefore logically valid.

Example (2): $\exists x Fx \wedge \exists x Gx$ does not entail $\exists x(Fx \wedge Gx)$.



Here we have applied to rule to each line in the tree, and there are no new ground terms to apply to the false existential.

Describing a structure from an open tableau branch.

- (1) Let $|\mathfrak{M}|$ have as many elements as there are ground terms on the branch.

In the example, we have two constants, “a” and “b”, so let $|\mathfrak{M}|$ be {Adele, Britney}.

- (2) Assign an item in the domain to each ground term; it could be a number, or even the term itself. Sometimes it’s more fun to pick other objects easy to remember from the constants. This fixes what we need to interpret the constants and function letters.

E.g., let “a” $^{\mathfrak{M}}$ = Adele and “b” $^{\mathfrak{M}}$ = Britney.

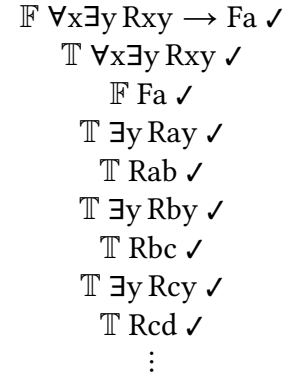
- (3) Include a given object or tuple in the extension of a predicate just in case the corresponding atomic statement appears with $\mathbb{T}$ on the branch. So here we have “F” $^{\mathfrak{M}}$ = {Adele} and “G” $^{\mathfrak{M}}$ = {Britney}, since “Fa” and “Gb” are true, but “Fb” and “Ga” are false.

- (4) The resulting structure should make all the sentences with $\mathbb{T}$ in front of them on the branch true, and those with $\mathbb{F}$ in front of them false.

In the book you can find a proof of the last claim, and from it, the overall metatheory of tableaux proving the method sound and complete. In this course, we will reserve such metatheoretic focus for axiomatic deductions.

The book also annotates tableaux with line numbers and rule markers; I do not bother.

Example (3): An inconclusive tree.



Each time we add a new constant, it triggers us to redo the true universal, yielding another true existential, and another new constant, and on it goes. In a case like this, the loop is easy to see, and it is intuitively clear that the branch will remain open. In fact, obviously, it is possible to make this formula false. Let $|\mathfrak{M}|$ be the set of natural numbers, let “a” $^{\mathfrak{M}}$ be zero, let “R” $^{\mathfrak{M}}$ be the less-than relation and let “F” $^{\mathfrak{M}}$ be the set of odd numbers.

But there is no generic “test” that would allow us to know that branches in general must terminate in any finite number of steps if they are going to close at all. If I have done 10 million steps, is that always enough? Not in general, so this does not always provide an effective test.

In fact later in the semester, we’ll prove that there is no effective decision procedure for testing validity in first-order predicate logic.

HOMEWORK 8

Construct tableaux to test the following; for those that are invalid, describe a counterexample structure. Is it true that...

- (1) $\models (Fa \leftrightarrow Ga) \rightarrow (\neg Fa \vee \neg Ga)?$
- (2) $\models \exists x(Fx \rightarrow \forall y Gy) \rightarrow \exists x(Fx \wedge Gx)?$
- (3) $\forall x \forall y Rxy \models \forall x Rxx?$
- (4) $\neg \exists x \forall y Ryx \models \exists x \forall y \neg Rxy?$

3. Axiomatic Deductions

As mentioned above, there are different kinds of derivations or deduction systems for logic. Axiomatic deductions were the first to be rigorously used or studied, and the easiest to do metatheory for, although they can be cumbersome to use.

The chief/defining features are:

- Each deduction can be characterized as a single ordered series of formulas: $\psi_1, \psi_2, \dots, \psi_n$, etc. The conclusion is always the last of the series.
- All deductions are direct; there are no special types of deductions such as “conditional” or “indirect”.
- The number of inference rules is kept small, using axioms in their place where possible. (An “axiom” is a line of a deduction that does not need to be “justified” by another line.)

Definition: An *axiomatic derivation* or *proof* of a result φ from a set of formulas Γ is a finite ordered series of formulas $\psi_1, \psi_2, \dots, \psi_n$, where the last item ψ_n is φ , and each step ψ_i either (a) is an axiom, (b) is a member of Γ , or (c) follows from previous members of the series by an inference rule.

Definition: A formula φ is *derivable* from a set Γ iff there exists at least one axiomatic derivation of φ from Γ ; this is written $\Gamma \vdash \varphi$.

If Γ 's members can be listed, we may write simply $\psi_1, \dots, \psi_n \vdash \varphi$.

Definition: A formula φ is a *theorem* of an axiomatic system iff $\emptyset \vdash \varphi$, i.e. iff there is a derivation of it using only axioms and inference rules. We may write this simply $\vdash \varphi$.

Propositional axioms and MP

Classical truth-functional propositional logic can be “axiomatized” by taking every formula of one of the following forms (—there are infinitely many of each—) as an axiom:

$(\varphi \wedge \psi) \rightarrow \varphi$	(SimpL Ax)
$(\varphi \wedge \psi) \rightarrow \psi$	(SimpR Ax)
$\varphi \rightarrow [\psi \rightarrow (\varphi \wedge \psi)]$	(Conj Ax)
$\varphi \rightarrow (\varphi \vee \psi)$	(AddR Ax)
$\varphi \rightarrow (\psi \vee \varphi)$	(AddL Ax)
$(\varphi \rightarrow \theta) \rightarrow \{(\psi \rightarrow \theta) \rightarrow [(\varphi \vee \psi) \rightarrow \theta]\}$	(Cases Ax)
$\varphi \rightarrow (\psi \rightarrow \varphi)$	(TCAx)
$[\varphi \rightarrow (\psi \rightarrow \theta)] \rightarrow [(\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \theta)]$	(CMP Ax)
$(\varphi \rightarrow \psi) \rightarrow [(\varphi \rightarrow \neg \psi) \rightarrow \neg \varphi]$	(CMT Ax)
$\neg \varphi \rightarrow (\varphi \rightarrow \psi)$	(FA Ax)
$\top$	(Verum Ax)
$\perp \rightarrow \varphi$	(Falsum Ax)
$(\varphi \rightarrow \perp) \rightarrow \neg \varphi$	(Red Ax)
$\neg \neg \varphi \rightarrow \varphi$	(DNE Ax)

Propositional logic then requires a single inference rule: *modus ponens* (MP): if $\varphi_j \rightarrow \varphi_i$ and φ_j already occur in a derivation, φ_i may follow.

Instances of these forms are axioms, but the forms themselves can be called **axiom schemata**.

$\vdash \text{Fa} \rightarrow \text{Fa}$	
1. $\text{Fa} \rightarrow [(\text{Fa} \rightarrow \text{Fa}) \rightarrow \text{Fa}]$	TCAx
2. $\{\text{Fa} \rightarrow [(\text{Fa} \rightarrow \text{Fa}) \rightarrow \text{Fa}]\} \rightarrow \{[\text{Fa} \rightarrow (\text{Fa} \rightarrow \text{Fa})] \rightarrow (\text{Fa} \rightarrow \text{Fa})\}$	CMP Ax
3. $[\text{Fa} \rightarrow (\text{Fa} \rightarrow \text{Fa})] \rightarrow (\text{Fa} \rightarrow \text{Fa})$	1, 2 MP
4. $\text{Fa} \rightarrow (\text{Fa} \rightarrow \text{Fa})$	TCAx
5. $\text{Fa} \rightarrow \text{Fa}$	3, 4 MP

Usually, in this class we are not interested in particular theorems in the object language, but *forms*; we can make the above proof general by replacing “Fa” with φ :

$\vdash \varphi \rightarrow \varphi$	(“Ref $\rightarrow$ ”)
1. $\varphi \rightarrow [(\varphi \rightarrow \varphi) \rightarrow \varphi]$	TCAx
2. $\{\varphi \rightarrow [(\varphi \rightarrow \varphi) \rightarrow \varphi]\} \rightarrow \{[\varphi \rightarrow (\varphi \rightarrow \varphi)] \rightarrow (\varphi \rightarrow \varphi)\}$	CMP Ax
3. $[\varphi \rightarrow (\varphi \rightarrow \varphi)] \rightarrow (\varphi \rightarrow \varphi)$	1, 2 MP
4. $\varphi \rightarrow (\varphi \rightarrow \varphi)$	TCAx
5. $\varphi \rightarrow \varphi$	3, 4 MP

This is a **derivation schema** that shows that *every* instance of $\varphi \rightarrow \varphi$ is a theorem. We can say that $\varphi \rightarrow \varphi$ is a **theorem schema**.

Once we have established a theorem schema, we can treat it like an axiom schema in subsequent proofs, since it would merely be a matter of replacing that line of the proof with the corresponding steps. We'll cite it as "Ref $\rightarrow$ ".

$\varphi \vdash \neg\neg\varphi$ ("DNI")

- | | |
|--|-------------------|
| 1. φ | Premise |
| 2. $\varphi \rightarrow (\neg\varphi \rightarrow \varphi)$ | TCAX |
| 3. $\neg\varphi \rightarrow \varphi$ | 1, 2 MP |
| 4. $(\neg\varphi \rightarrow \varphi) \rightarrow ((\neg\varphi \rightarrow \neg\varphi) \rightarrow \neg\neg\varphi)$ | CMTAX |
| 5. $(\neg\varphi \rightarrow \neg\varphi) \rightarrow \neg\neg\varphi$ | 3, 4 MP |
| 6. $\neg\varphi \rightarrow \neg\varphi$ | Ref $\rightarrow$ |
| 7. $\neg\neg\varphi$ | 5, 6 MP |

Strictly speaking the above does not meet the definition of an axiomatic derivation, or even a schema for one, but it would if we replaced line 6 with the steps from our previous derivation schema, with $\neg\varphi$ everywhere for φ .

Result: If $\Gamma \vdash \varphi$ and $\varphi \vdash \psi$, then $\Gamma \vdash \psi$.

Proof: After the derivation of φ from Γ , concatenate the remaining steps of the derivation $\varphi \vdash \psi$, using the φ from final line of the first derivation instead of taking φ as a premise. The result will be a derivation of ψ from Γ . *

This, with the above result $\varphi \vdash \neg\neg\varphi$, mean that if we ever have a line φ , we can get its double negation. We could just insert the steps of the derivation schema above. So from now on, we will not bother, and simply treat "DNI" as if it were an inference rule. We call this a **derived rule**.

Corollary: If $\Gamma \vdash \varphi_1$ and...and $\Gamma \vdash \varphi_n$ and $\varphi_1, \dots, \varphi_n \vdash \psi$, then $\Gamma \vdash \psi$.

$\varphi \rightarrow \psi, \neg\psi \vdash \neg\varphi$ ("MT")

- | | |
|--|---------|
| 1. $\varphi \rightarrow \psi$ | Premise |
| 2. $\neg\psi$ | Premise |
| 3. $\neg\psi \rightarrow (\varphi \rightarrow \neg\psi)$ | TCAX |
| 4. $\varphi \rightarrow \neg\psi$ | 2, 3 MP |
| 5. $(\varphi \rightarrow \psi) \rightarrow ((\varphi \rightarrow \neg\psi) \rightarrow \neg\varphi)$ | CMTAX |
| 6. $(\varphi \rightarrow \neg\psi) \rightarrow \neg\varphi$ | 1, 5 MP |
| 7. $\neg\varphi$ | 4, 6 MP |

Again, from now on, we'll just pretend MT is an inference rule just like MP.

$\varphi \rightarrow \psi, \psi \rightarrow \theta \vdash \varphi \rightarrow \theta$ ("HS")

- | | |
|--|---------|
| 1. $\varphi \rightarrow \psi$ | Premise |
| 2. $\psi \rightarrow \theta$ | Premise |
| 3. $(\psi \rightarrow \theta) \rightarrow (\varphi \rightarrow (\psi \rightarrow \theta))$ | TCAX |
| 4. $\varphi \rightarrow (\psi \rightarrow \theta)$ | 2, 3 MP |
| 5. $[\varphi \rightarrow (\psi \rightarrow \theta)] \rightarrow [(\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \theta)]$ | CMPAX |
| 6. $(\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \theta)$ | 4, 5 MP |
| 7. $\varphi \rightarrow \theta$ | 1, 6 MP |

$\varphi \rightarrow (\psi \rightarrow \theta) \vdash \psi \rightarrow (\varphi \rightarrow \theta)$ ("Int")

- | | |
|--|---------|
| 1. $\varphi \rightarrow (\psi \rightarrow \theta)$ | Premise |
| 2. $[\varphi \rightarrow (\psi \rightarrow \theta)] \rightarrow [(\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \theta)]$ | |
| | CMPAX |
| 3. $(\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \theta)$ | 1, 2 MP |
| 4. $\psi \rightarrow (\varphi \rightarrow \psi)$ | TCAX |
| 5. $\psi \rightarrow (\varphi \rightarrow \theta)$ | 3, 4 HS |

HOMEWORK 9

Give proof schemata for the derived rules for disjunctive syllogism ("DS"). Hint: use (CasesAX), (FAAX) and (Ref $\rightarrow$).

- (1) $\varphi \vee \psi, \neg\varphi \vdash \psi$
- (2) $\varphi \vee \psi, \neg\psi \vdash \varphi$

We've already used this simple one, but we might as well name it: "TC" for "true consequent".

$\varphi \vdash \psi \rightarrow \varphi$ (TC)

- | | |
|---|---------|
| 1. φ | Premise |
| 2. $\varphi \rightarrow (\psi \rightarrow \varphi)$ | (TCAX) |
| 3. $\psi \rightarrow \varphi$ | 1, 2 MP |

$$\begin{array}{l} \varphi \rightarrow (\psi \rightarrow \theta) \vdash (\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \theta) \\ \varphi \rightarrow (\psi \rightarrow \theta), \varphi \rightarrow \psi \vdash \varphi \rightarrow \theta \end{array} \quad (\text{CMP})$$

(Direct from CMPAx and MP once or twice.)

$$\begin{array}{l} (\varphi \wedge \psi) \rightarrow \theta \vdash \varphi \rightarrow (\psi \rightarrow \theta) \quad (\text{Exp}) \\ 1. (\varphi \wedge \psi) \rightarrow \theta \quad \text{Premise} \\ 2. \psi \rightarrow [(\varphi \wedge \psi) \rightarrow \theta] \quad 1 \text{ TC} \\ 3. [\psi \rightarrow (\varphi \wedge \psi)] \rightarrow (\psi \rightarrow \theta) \quad 2 \text{ CMP} \\ 4. \varphi \rightarrow \{[\psi \rightarrow (\varphi \wedge \psi)] \rightarrow (\psi \rightarrow \theta)\} \quad 3 \text{ TC} \\ 5. \varphi \rightarrow [\psi \rightarrow (\varphi \wedge \psi)] \quad \text{ConjAx} \\ 6. \varphi \rightarrow (\psi \rightarrow \theta) \quad 4, 5 \text{ CMP} \end{array}$$

$$\begin{array}{l} \varphi \rightarrow (\psi \rightarrow \theta) \vdash (\varphi \wedge \psi) \rightarrow \theta \quad (\text{Imp}) \\ 1. \varphi \rightarrow (\psi \rightarrow \theta) \quad \text{Premise} \\ 2. (\varphi \wedge \psi) \rightarrow \varphi \quad \text{SimpLax} \\ 3. (\varphi \wedge \psi) \rightarrow (\psi \rightarrow \theta) \quad 1, 2 \text{ HS} \\ 4. (\varphi \wedge \psi) \rightarrow \psi \quad \text{SimpRAx} \\ 5. (\varphi \wedge \psi) \rightarrow \theta \quad 3, 4 \text{ CMP} \end{array}$$

Other consequences (some easy, some not):

$$\begin{array}{ll} \varphi \wedge \psi \vdash \varphi & (\text{Simp}) \\ \varphi \wedge \psi \vdash \psi & (\text{Simp}) \\ \varphi, \psi \vdash \varphi \wedge \psi & (\text{Conj}) \\ \varphi \vdash \varphi \vee \psi & (\text{Add}) \\ \varphi \vdash \psi \vee \varphi & (\text{Add}) \\ \neg \varphi \vdash \varphi \rightarrow \psi & (\text{FA}) \\ \neg \neg \varphi \vdash \varphi & (\text{DNE}) \\ \varphi \vee \varphi \vdash \varphi & (\text{Taut}) \\ \varphi \vdash \varphi \wedge \varphi & (\text{Taut}) \\ \varphi \wedge \psi \vdash \psi \wedge \varphi & (\text{Com}) \\ \varphi \vee \psi \vdash \psi \vee \varphi & (\text{Com}) \\ (\varphi \vee \psi) \vee \theta \vdash \varphi \vee (\psi \vee \theta) & (\text{Assoc}) \\ (\varphi \wedge \psi) \wedge \theta \vdash \varphi \wedge (\psi \wedge \theta) & (\text{Assoc}) \\ \varphi \wedge (\psi \vee \theta) \vdash (\varphi \wedge \psi) \vee (\varphi \wedge \theta) & (\text{Dist}) \\ \varphi \vee (\psi \wedge \theta) \vdash (\varphi \vee \psi) \wedge (\varphi \vee \theta) & (\text{Dist}) \\ \varphi \rightarrow \theta, \psi \rightarrow \xi, \varphi \vee \psi \vdash \theta \vee \xi & (\text{CD}) \\ \varphi \rightarrow \psi, \neg \varphi \rightarrow \psi \vdash \psi & (\text{Inev}) \\ \varphi \rightarrow \psi, \theta \rightarrow \psi \vdash (\varphi \vee \theta) \rightarrow \psi & (\text{Cases}) \\ \varphi, \neg \varphi \vdash \perp & (\text{Contra}) \\ \perp \vdash \varphi & (\text{Explode}) \\ \text{etc.} & \end{array}$$

In short, we “recapture” all the rules one is used to having in a natural deduction system.

Reducing the axiom schemata

Unfortunately, the book seems to have forgotten to add any axioms for the biconditional operator $\leftrightarrow$. I think this was probably an honest oversight, but I’m glad for it, as it allows me to introduce the possibility of using a more austere syntax. If we added axiom schemata, most likely we would need:

$$\begin{array}{ll} (\varphi \leftrightarrow \psi) \rightarrow (\varphi \rightarrow \psi) & (\text{EquivLR}) \\ (\varphi \leftrightarrow \psi) \rightarrow (\psi \rightarrow \varphi) & (\text{EquivRL}) \\ (\varphi \rightarrow \psi) \rightarrow ((\psi \rightarrow \varphi) \rightarrow (\varphi \leftrightarrow \psi)) & (\text{EquivIn}) \end{array}$$

As an alternative to adding these to our axiom schemata, however, we could simply eliminate $\leftrightarrow$ as a **primitive** part of the notation, and use it only as an **abbreviation**:

$$\varphi \leftrightarrow \psi \text{ is shorthand for } (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$$

In that case, instances of (EquivLR), (EquivRL), and (EquivIn) are just *notational variants* of instances of (SimpLax), (SimpRAx) and (ConjAx).

In fact, we could go further; we could use only a minimum number of primitive signs, say $\{\rightarrow, \neg\}$ or even just $\{|\}$, and eliminate the axioms governing the defined ones. For example, if we took $\psi \vee \varphi$ to abbreviate $\neg \psi \rightarrow \varphi$, then we wouldn’t need an axiom schema for $\varphi \rightarrow (\psi \vee \varphi)$, as this would just be an instance of (TCAx), and $\psi \vee \varphi, \neg \psi \vdash \varphi$ would just be *modus ponens*, etc. With only $\{\rightarrow, \neg\}$ as primitive, we can dramatically reduce the axiomatic schemata. E.g., here’s Frege’s own original axiomatization:

$$\begin{array}{l} \varphi \rightarrow (\psi \rightarrow \varphi) \\ [\varphi \rightarrow (\psi \rightarrow \theta)] \rightarrow [(\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \theta)] \\ (\varphi \rightarrow \psi) \rightarrow (\neg \psi \rightarrow \neg \varphi) \\ \varphi \rightarrow \neg \neg \varphi \\ \neg \neg \varphi \rightarrow \varphi \end{array}$$

It’s possible to replace the last three with only:

$$(\neg \varphi \rightarrow \neg \psi) \rightarrow [(\neg \varphi \rightarrow \psi) \rightarrow \varphi]$$

And we can go further still, though there are diminishing returns. In 1917, Jean Nicod discovered a system for propositional logic using only the Sheffer stroke $|$ as primitive, with the single schema:

$$(\varphi | (\psi | \theta)) | ((\xi | (\xi | \xi)) | ((\zeta | \psi) | ((\varphi | \zeta) | (\varphi | \zeta))))$$

The only inference rule is: From $\varphi | (\theta | \psi)$ and φ infer ψ . This suffices for all of propositional logic.

There is something nice about austere systems, but they can require very alien and complicated derivations for what seem like obvious results for “non-primitive” but familiar operators.

I shall assume we have (EquivLR), (EquivRL), (EquivIn) and their easy derived rules:

$$\begin{array}{ll} \varphi \leftrightarrow \psi \vdash \varphi \rightarrow \psi & (\text{BE}) \\ \varphi \leftrightarrow \psi \vdash \psi \rightarrow \varphi & (\text{BE}) \\ \varphi \rightarrow \psi, \psi \rightarrow \varphi \vdash \varphi \leftrightarrow \psi & (\text{BI}) \\ \varphi \leftrightarrow \psi, \varphi \vdash \psi & (\text{BMP}) \\ \varphi \leftrightarrow \psi, \psi \vdash \varphi & (\text{BMP}) \\ \varphi \leftrightarrow \psi, \neg\varphi \vdash \neg\psi & (\text{BMT}) \\ \varphi \leftrightarrow \psi, \neg\psi \vdash \neg\varphi & (\text{BMT}) \end{array}$$

It’s not really important whether we consider these as results of new axioms or just consequences of introducing non-primitive notation.

Quantifier axioms and rules

For predicate logic, we also need axioms/rules for the quantifiers. (Here too, we could have took $\exists x$ to abbreviate $\neg\forall x\neg$ instead...)

Where t is a ground term (closed term), any instance of the following schemata are considered axioms:

$$\begin{array}{ll} \forall x\varphi \rightarrow \varphi[t/x] & (\text{UIAx}) \\ \varphi[t/x] \rightarrow \exists x\varphi & (\text{EGAx}) \end{array}$$

Instances of (UIAx) include, e.g., “ $\forall x Fx \rightarrow Fa$ ” and $\forall x(Fx \rightarrow Gx) \rightarrow (Fb \rightarrow Gb)$. With MP it allows us to “instantiate” variables to constants or function terms. Instances of (EGAx) include “ $Fa \rightarrow \exists x Fx$ ”

and the like. Hence we get the derived rules (Hardegree’s $\forall O$ and $\exists I$):

$$\forall x\varphi(x) \vdash \varphi(t) \quad (\text{UI})$$

$$\varphi(t) \vdash \exists x\varphi(x) \quad (\text{EI})$$

We also need rules for “using” existentials and “getting” universals. Different versions of axiomatic systems have historically done this different ways. The “old way” involves allowing free variables in formulas in proofs, and allowing one to “generalize” on open formulas to arrive at a version with an initial quantifier. This way has ceded ground to a “new way” that uses “dummy constants”, which is what the Open Logic Text does. This involves the quantifier rules (QR):

QR (two forms):

- If constant a does not occur in a premise or ψ then from $\psi \rightarrow \varphi(a)$ one may infer $\psi \rightarrow \forall x\varphi(x)$.
- If constant a does not occur in a premise or ψ then from $\varphi(a) \rightarrow \psi$ infer $\exists x\varphi(x) \rightarrow \psi$.

HOMEWORK 10

Show that $\vdash \forall x(\varphi(x) \rightarrow \perp) \rightarrow \neg\exists x\varphi(x)$. You may use derived rules from propositional logic only.

The first form of QR allows us to do the work of universal generalization (or Hardegree’s “universal derivation (UD)”), and the second the work of existential instantiation (Hardegree’s “ $\exists O$ ”). The latter must wait till after we tackle the “deduction theorem”, but for the former, note the following:

Result (UG): *If one obtains a line of the form $\varphi(a)$, where a is a constant that does not occur in a premise, then one can derive $\forall x\varphi(x)$.*

- Proof:* (1) Suppose one obtains $\varphi(a)$ where a is not in a premise.
(2) By TC and MP, we get $\top \rightarrow \varphi(a)$.
(3) By QR, we get $\top \rightarrow \forall x\varphi(x)$.
(4) By VerumAx and MP we get $\forall x\varphi(x)$. *

From now on, we'll just call the above steps "UG", and use it as if it were a derived rule. Note that $\varphi(a) \vdash \forall x\varphi(x)$ is generally not the case, however, as a cannot occur in a premise.

The deduction theorem

The main thing now keeping this method from being as easy to use as a natural deduction system is the lack of conditional and indirect (or "reductio") proofs. We'll prove that these techniques are not needed, as any *apparent* derivation using them could be transformed into one that doesn't.

Result (DT): *If $\Gamma \cup \{\theta\} \vdash \varphi$ then $\Gamma \vdash \theta \rightarrow \varphi$.*

Proof: Assume that $\Gamma \cup \{\theta\} \vdash \varphi$ and that the derivation of ψ from $\Gamma \cup \{\theta\}$ is $\psi_1, \dots, \psi_n$. (We assume the derivation is written out in full, without derived rules or cited theorems.) We shall use proof induction to show that every member of this series ψ_i has the property that $\Gamma \vdash \theta \rightarrow \psi_i$. As inductive hypothesis, we are allowed to assume that all members prior to an arbitrary ψ_i have this property. We must show that given this assumption, ψ_i has it too. There are six cases to consider:

- (1) ψ_i is an axiom; then $\Gamma \vdash \psi_i$ and by TC, $\Gamma \vdash \theta \rightarrow \psi_i$
- (2) ψ_i is a member of Γ ; then $\Gamma \vdash \psi_i$ and by TC, $\Gamma \vdash \theta \rightarrow \psi_i$.
- (3) ψ_i is a member of $\{\theta\}$, i.e., it is θ ; in that case, $\Gamma \vdash \theta \rightarrow \psi_i$ by Ref $\rightarrow$.
- (4) ψ_i follows from previous members of the series, ψ_j and $\psi_j \rightarrow \psi_i$ by MP. By the inductive hypothesis, $\Gamma \vdash \theta \rightarrow \psi_j$ and $\Gamma \vdash \theta \rightarrow (\psi_j \rightarrow \psi_i)$. By CMP, $\Gamma \vdash \theta \rightarrow \psi_i$.
- (5) ψ_i follows from a previous member of the series by the first form of QR. Hence ψ_i takes the form $\xi \rightarrow \forall x\zeta(x)$, and there is a previous member of the ψ -series $\xi \rightarrow \zeta(a)$, where a does not occur in $\Gamma \cup \{\theta\}$, nor in ξ . By the inductive hypothesis $\Gamma \vdash \theta \rightarrow (\xi \rightarrow \zeta(a))$. By Imp, $\Gamma \vdash (\theta \wedge \xi) \rightarrow \zeta(a)$. Because a does not occur in any member of $\Gamma \cup \{\theta\}$, it doesn't occur in θ ;

hence it also does not occur in $\theta \wedge \xi$. Hence, by QR, $\Gamma \vdash (\theta \wedge \xi) \rightarrow \forall x\zeta(x)$. By Exp, $\Gamma \vdash \theta \rightarrow (\xi \rightarrow \forall x\zeta(x))$, i.e., $\Gamma \vdash \theta \rightarrow \psi_i$.

- (6) ψ_i follows a previous member of the series by the second form of QR. Hence, ψ_i takes the form $\exists x\xi(x) \rightarrow \zeta$, and there is a previous member of the ψ -series $\xi(a) \rightarrow \zeta$, where a does not occur in ζ nor in $\Gamma \cup \{\theta\}$. By the inductive hypothesis $\Gamma \vdash \theta \rightarrow (\xi(a) \rightarrow \zeta)$. By Int, $\Gamma \vdash \xi(a) \rightarrow (\theta \rightarrow \zeta)$. a does not occur in either θ or ζ , so by QR, $\Gamma \vdash \exists x\xi(x) \rightarrow (\theta \rightarrow \zeta)$. By Int again, $\Gamma \vdash \theta \rightarrow (\exists x\xi(x) \rightarrow \zeta)$, i.e., $\Gamma \vdash \theta \rightarrow \psi_i$.

This covers all possible cases, so by proof induction, it holds for every member of the ψ -series, including the last one, which is φ . Therefore, $\Gamma \vdash \theta \rightarrow \varphi$. *

I am very fond of this proof. Essentially, it provides a *recipe* for *transforming* a proof in which one makes an additional formula as an assumption/premise/hypothesis and turning it into a proof of the conditional from that extra hypothesis to the result the original proof reaches with its help. Consider:

$$\forall x(\varphi(x) \rightarrow \psi(x)), \forall x\varphi(x) \vdash \forall x\psi(x)$$

(The proof below is not ideal, because I must resist using derived rules in order to apply the method.)

- | | |
|---|-----------|
| 1. $\forall x(\varphi(x) \rightarrow \psi(x))$ | Premise |
| 2. $\forall x\varphi(x)$ | "Hyp" |
| 3. $\forall x(\varphi(x) \rightarrow \psi(x)) \rightarrow (\varphi(a) \rightarrow \psi(a))$ | UIAx |
| 4. $\varphi(a) \rightarrow \psi(a)$ | 1, 3 MP |
| 5. $\forall x\varphi(x) \rightarrow \varphi(a)$ | UIAx |
| 6. $\varphi(a)$ | 2, 5 MP |
| 7. $\psi(a)$ | 4, 6 MP |
| 8. $\psi(a) \rightarrow (\top \rightarrow \psi(a))$ | TCAx |
| 9. $\top \rightarrow \psi(a)$ | 7, 8 MP |
| 10. $\top \rightarrow \forall x\psi(x)$ | 9 QR |
| 11. $\top$ | VerumAx |
| 12. $\forall x\psi(x)$ | 10, 11 MP |

Line 2 here is a premise, but I've labeled it "Hyp" to indicate that we can think of it as an extra assumption, as if we wanted to do a "conditional" proof for $\forall x\varphi(x) \rightarrow \forall x\psi(x)$. The recipe provided by the deduction theorem turns it into a proof of that. Here Γ is $\{\forall x(\varphi(x) \rightarrow \psi(x))\}$ and the θ is $\forall x\varphi(x)$. We apply steps to the original proofs to get $\forall x\varphi(x) \rightarrow \dots$, filling in the dots with the original line.

Start with line 1, which is case (2).

- 1a. $\forall x(\varphi(x) \rightarrow \psi(x))$ Premise
 1b. $\forall x\varphi(x) \rightarrow [\forall x(\varphi(x) \rightarrow \psi(x))]$ 1a TC

Notice that 1b here is the original line 1 with $\forall x\varphi(x) \rightarrow \dots$ before it. We want that for all steps. Original line 2 is case (3):

- 2a. $\forall x\varphi(x) \rightarrow \forall x\varphi(x)$ Ref $\rightarrow$

Then case (1):

- 3a. $\forall x(\varphi(x) \rightarrow \psi(x)) \rightarrow (\varphi(a) \rightarrow \psi(a))$ UIAx
 3b. $\forall x\varphi(x) \rightarrow [\forall x(\varphi(x) \rightarrow \psi(x)) \rightarrow (\varphi(a) \rightarrow \psi(a))]$ 3a TC

For case (4) we need to rely the inductive hypothesis.

- 4a. $\forall x\varphi(x) \rightarrow (\varphi(a) \rightarrow \psi(a))$ 1b, 3b CMP

Another case (1):

- 5a. $\forall x\varphi(x) \rightarrow \varphi(a)$ UIAx
 5b. $\forall x\varphi(x) \rightarrow (\forall x\varphi(x) \rightarrow \varphi(a))$ 5a TC

Then two more case (4)'s:

- 6a. $\forall x\varphi(x) \rightarrow \varphi(a)$ 2a, 5b CMP
 7a. $\forall x\varphi(x) \rightarrow \psi(a)$ 4a, 6a CMP

Obviously, this is not the most efficient way to get these results. 6a is the *same* as a line we already have, and an axiom to boot! Continuing:

- 8a. $\psi(a) \rightarrow (\top \rightarrow \psi(a))$ TCAx
 8b. $\forall x\varphi(x) \rightarrow [\psi(a) \rightarrow (\top \rightarrow \psi(a))]$ 8a TC
 9a. $\forall x\varphi(x) \rightarrow (\top \rightarrow \psi(a))$ 7a, 8b CMP

Line 10 falls into case (5), which is new.

- 10a. $(\forall x\varphi(x) \wedge \top) \rightarrow \psi(a)$ 9a Imp
 10b. $(\forall x\varphi(x) \wedge \top) \rightarrow \forall x\psi(x)$ 10a QR
 10c. $\forall x\varphi(x) \rightarrow (\top \rightarrow \forall x\psi(x))$ 10b Exp

And to finish:

- 11a. $\top$ VerumAx
 11b. $\forall x\varphi(x) \rightarrow \top$ 11a TC
 12a. $\forall x\varphi(x) \rightarrow \forall x\psi(x)$ 10c, 11b CMP

Notice the only premise used in the “transformed” proof was at 1a, so this establishes:

$$\forall x(\varphi(x) \rightarrow \psi(x)) \vdash \forall x\varphi(x) \rightarrow \forall x\psi(x)$$

We could even go further and transform the transformed proof (though this would require expanding all the derived rules first) into one for:

$$\vdash \forall x(\varphi(x) \rightarrow \psi(x)) \rightarrow (\forall x\varphi(x) \rightarrow \forall x\psi(x))$$

HOMEWORK 11

On page 17, there is a proof for the derived rule of *modus tollens* (MT). Use the recipe provided by the proof of the deduction theorem to transform it, step by step, into a proof transposition in the form:

$$\varphi \rightarrow \psi \vdash \neg\psi \rightarrow \neg\varphi \quad (\text{Trans})$$

We have proven that conditional proof is *not needed*. Anything you could prove with conditional proof we can prove without it by transforming the alleged conditional proof into a direct one.

From now on (—well, at least after the above homework assignment—) we won’t actually bother with the transformations. If we want to prove a conditional $\theta \rightarrow \varphi$ from Γ , We’ll just do the proof of φ with an additional “Hyp”, θ , and then “cite” the deduction theorem (“DT”) as thereby establishing that $\Gamma \vdash \theta \rightarrow \varphi$ is true as well.

Sometimes it helps to write proofs using the sign $\vdash$ to keep track of what premises/hypotheses a line depends on, e.g.:

$$\vdash \forall x\neg\varphi(x) \rightarrow \neg\exists x\varphi(x) \quad (\text{CQ})$$

1. $\forall x\neg\varphi(x) \vdash \forall x\neg\varphi(x)$ Hyp
2. $\forall x\neg\varphi(x) \vdash \neg\varphi(a)$ 1 UI
3. $\forall x\neg\varphi(x) \vdash \varphi(a) \rightarrow \perp$ 2 FA
4. $\forall x\neg\varphi(x) \vdash \exists x\varphi(x) \rightarrow \perp$ 3 QR
5. $\vdash (\exists x\varphi(x) \rightarrow \perp) \rightarrow \neg\exists x\varphi(x)$ RedAx
6. $\forall x\neg\varphi(x) \vdash \neg\exists x\varphi(x)$ 4, 5 MP
7. $\vdash \forall x\neg\varphi(x) \rightarrow \neg\exists x\varphi(x)$ 6 DT

It is important for QR and rules/methods based on it, like UG, that a certain constant not appear in the premises/hypotheses. This applies to those that have not *already* been discharged. Consider:

$$\forall x(\varphi(x) \rightarrow \psi(x)) \vdash \forall x(\neg\psi(x) \rightarrow \neg\varphi(x))$$

1. $\forall x(\varphi(x) \rightarrow \psi(x)) \vdash \forall x(\varphi(x) \rightarrow \psi(x))$ Premise
2. $\neg\psi(a) \vdash \neg\psi(a)$ Hyp

3. $\forall x(\varphi(x) \rightarrow \psi(x)) \vdash \varphi(a) \rightarrow \psi(a)$ 1 UI
4. $\forall x(\varphi(x) \rightarrow \psi(x)), \neg\psi(a) \vdash \neg\varphi(a)$ 2, 3 MT
5. $\forall x(\varphi(x) \rightarrow \psi(x)) \vdash \neg\psi(a) \rightarrow \neg\varphi(a)$ 4 DT
6. $\forall x(\varphi(x) \rightarrow \psi(x)) \vdash \forall x(\neg\psi(x) \rightarrow \neg\varphi(x))$ 5 UG

Here we could not have applied UG at line 4 to get $\forall x\neg\varphi(x)$, because a occurs in one of the premises/hypotheses that line depends on. However, it is OK to apply it to line 5 to get line 6, because we have “discharged” that hypotheses with DT. It is important to be careful about assumptions/premises used for DT and QR/UG.

Corollaries of the deduction theorem

The deduction theorem also underwrites the indirect proof or reductio-style proof technique.

Result (RAA): If $\Gamma \cup \{\varphi\} \vdash \perp$, then $\Gamma \vdash \neg\varphi$.

Proof: Assume $\Gamma \cup \{\varphi\} \vdash \perp$. By the deduction theorem, $\Gamma \vdash \varphi \rightarrow \perp$. By RedAx and MP, $\Gamma \vdash \neg\varphi$. *

From now on we allow ourselves to use the reductio technique; notice moreover that $\varphi, \neg\varphi \vdash \perp$ follows easily from FAx.

Finally the deduction theorem demonstrates the relationship between the second form of QR and the “natural deduction” rule that allows you to instantiate an existential to a “new” or “unused” constant (“EI” or “EO/∃-Elim”). Note that $\exists x\varphi(x) \vdash \varphi(a)$ is not in general the case, but what is the case is this:

Result (EI): If $\Gamma \vdash \exists x\varphi(x)$ and $\Gamma \cup \{\varphi(a)\} \vdash \psi$, where a does not occur in Γ , or ψ , then $\Gamma \vdash \psi$.

Proof: Suppose $\Gamma \vdash \exists x\varphi(x)$ and $\Gamma \cup \{\varphi(a)\} \vdash \psi$ and that a does not occur in Γ or ψ . By the deduction theorem, $\Gamma \vdash \varphi(a) \rightarrow \psi$. By QR, $\Gamma \vdash \exists x\varphi(x) \rightarrow \psi$, and by MP, $\Gamma \vdash \psi$. *

For example:

$$\forall x(\varphi(x) \rightarrow \psi(x)), \exists x\varphi(x) \vdash \exists x\psi(x)$$

1. $\forall x(\varphi(x) \rightarrow \psi(x)) \vdash \forall x(\varphi(x) \rightarrow \psi(x))$ Premise
2. $\exists x\varphi(x) \vdash \exists x\varphi(x)$ Premise
3. $\varphi(a) \vdash \varphi(a)$ Hyp
4. $\forall x(\varphi(x) \rightarrow \psi(x)) \vdash \varphi(a) \rightarrow \psi(a)$ 1 UI
5. $\forall x(\varphi(x) \rightarrow \psi(x)), \varphi(a) \vdash \psi(a)$ 3, 4 MP
6. $\forall x(\varphi(x) \rightarrow \psi(x)), \varphi(a) \vdash \exists x\psi(x)$ 5 EG
7. $\forall x(\varphi(x) \rightarrow \psi(x)) \vdash \varphi(a) \rightarrow \exists x\psi(x)$ 6 DT
8. $\forall x(\varphi(x) \rightarrow \psi(x)) \vdash \exists x\varphi(x) \rightarrow \exists x\psi(x)$ 7 QR
9. $\forall x(\varphi(x) \rightarrow \psi(x)), \exists x\varphi(x) \vdash \exists x\psi(x)$ 2, 8 MP

Written up like this, we never actually use the existential line (2) until the end, and we simply take $\varphi(a)$, naturally thought of as the “instance” of (2), as a new hypothesis at line (3). It is harmless, though not technically *correct*, to write this as follows:

1. $\forall x(\varphi(x) \rightarrow \psi(x)) \vdash \forall x(\varphi(x) \rightarrow \psi(x))$ Premise
2. $\exists x\varphi(x) \vdash \exists x\varphi(x)$ Premise
3. $\exists x\varphi(x) \vdash^* \varphi(a)$ 2 EI
4. $\forall x(\varphi(x) \rightarrow \psi(x)) \vdash \varphi(a) \rightarrow \psi(a)$ 1 UI
5. $\forall x(\varphi(x) \rightarrow \psi(x)), \exists x\varphi(x) \vdash^* \psi(a)$ 3, 4 MP
6. $\forall x(\varphi(x) \rightarrow \psi(x)), \exists x\varphi(x) \vdash \exists x\psi(x)$ 5 EG

Here I have marked the turnstiles at lines 3 and 5 with asterisks, since the $\vdash$ relation doesn’t actually hold here. However, once the “dummy constant” disappears, the trick applies and the result does hold. (Even in your natural deduction systems, you were not allowed to end a proof with the dummy constant as a part of the conclusion.) Hence, I will allow you from here to pretend as if there is a rule “EI” (or “∃-Elim/∃-Out”), since derivations using them can always be transformed into one using the “instance” as a hypothesis, with QR, instead.

By this point we have probably recaptured pretty much all the logical rules you’re used to using, or could easily get the rest. But we want something stronger: to prove that *every* logically valid argument has a corresponding axiomatic derivation. We also want the reverse. This brings us to our main metatheoretic results.

4. Bridging $\vdash$ and $\models$

Soundness and consistency

Result: *The axioms of propositional and quantifier logic are all logically valid.*

- Proof:* (1) The propositional axioms are all truth-table tautologies; we here skip over the actual tables. The semantic rules track the truth table rules, so these are all valid.
- (2) Every instance of $\forall x\varphi(x) \rightarrow \varphi(t)$, where t is a closed term is logically valid. Suppose for *reductio* that this is not the case for some formula of this form. Then there is a structure $\mathfrak{M}$ and variable assignment s such that $\mathfrak{M}, s \models \forall x\varphi(x)$ but $\mathfrak{M}, s \not\models \varphi(t)$. Let s' be the variable assignment just like s except $\text{Val}_{s'}^{\mathfrak{M}}(x) = t^{\mathfrak{M}}$. Since $\mathfrak{M}, s \models \forall x\varphi(x)$ it must be that $\mathfrak{M}, s' \models \varphi(x)$. But s' differs from s by at most its value for x , and since $\mathfrak{M}, s \models \varphi(t)$ it must be that $\mathfrak{M}, s' \models \varphi(t)$. This is a contradiction, and therefore every instance of $\forall x\varphi(x) \rightarrow \varphi(t)$ must be valid after all.
- (3) Every instance of $\varphi(t) \rightarrow \exists x\varphi(x)$ is logically valid. Again suppose for *reductio* that some formula of this form were not valid. Then there is some structure $\mathfrak{M}$ and variable assignment s where $\mathfrak{M}, s \models \varphi(t)$ but $\mathfrak{M}, s \not\models \exists x\varphi(x)$. By the definition of satisfaction, $\mathfrak{M}, s \models \varphi(t)$ and $\mathfrak{M}, s \models \exists x\varphi(x)$. Let s' be the variable assignment just like s except $\text{Val}_{s'}^{\mathfrak{M}}(x) = t^{\mathfrak{M}}$. Since $\mathfrak{M}, s \models \varphi(t)$ it follows that $\mathfrak{M}, s' \models \varphi(x)$. But then there is an assignment different from s at most its value for x that satisfies $\varphi(x)$, and hence $\mathfrak{M}, s \models \exists x\varphi(x)$, which is a contradiction. Hence every instance of $\varphi(t) \rightarrow \exists x\varphi(x)$ is valid after all.
- (4) Cases (1)–(3) exhaust the axioms in question; hence, all logical axioms are logically valid. *

Result (Soundness): *If $\Gamma \vdash \varphi$ then $\Gamma \models \varphi$.*

Proof: Assume $\Gamma \vdash \varphi$. We need to show that every structure that is a model of Γ also makes φ true.

Let the derivation of φ from Γ be the series $\psi_1, \dots, \psi_n$. We will show by proof induction that every ψ_i in this series is such that every structure that is a model of Γ is also a model of it. We are entitled to assume that we have already shown this to be true for every member of the ψ -series prior to ψ_i .

There are five cases to consider.

- (1) ψ_i is an axiom. By our earlier result, ψ_i is logically valid, and hence true in *every* structure, including all those making the members of Γ true.
- (2) ψ_i is a member of Γ ; it is therefore made true by all interpretations making every member of Γ true, since it is one.
- (3) ψ_i follows from previous members of the series by MP. Then there are previous members of the series ψ_j and $\psi_j \rightarrow \psi_i$. By the inductive hypothesis, every structure making the members of Γ true make these true. Let $\mathfrak{M}$ be such a structure. Since it makes ψ_j and $\psi_j \rightarrow \psi_i$ true, it must also make ψ_i true by the semantics of $\rightarrow$. The same is true for any model of Γ .
- (4) ψ_i follows from a previous member of the series by the first form of QR. Then it takes the form $\xi \rightarrow \forall x\zeta(x)$ and there is a previous member $\xi \rightarrow \zeta(a)$, where a does not occur in Γ . By the inductive hypothesis, every model of Γ makes $\xi \rightarrow \zeta(a)$ true. Suppose for *reductio* that some model $\mathfrak{M}$ of Γ does not make $\xi \rightarrow \forall x\zeta(x)$ true. Then for some s , $\mathfrak{M}, s \models \xi$ and $\mathfrak{M}, s \not\models \forall x\zeta(x)$. This means that for some s' differing from s only for its value for x is such that $\mathfrak{M}, s' \models \zeta(x)$. Now consider the structure $\mathfrak{M}'$ differing from $\mathfrak{M}$ only in that $a^{\mathfrak{M}'} = \text{Val}_{s'}^{\mathfrak{M}}(x)$. $\mathfrak{M}$ is a model of Γ , and so $\mathfrak{M}'$ must be as well, since a does not occur in Γ . Since $|\mathfrak{M}| = |\mathfrak{M}'|$, the variable assignment is also an assignment for $\mathfrak{M}'$. Since $\mathfrak{M}'$ is a model of Γ , by the inductive hypothesis,

it makes $\xi \rightarrow \zeta(a)$ true, and therefore $\mathfrak{M}', s \models \xi \rightarrow \zeta(a)$. But ξ does not contain a , and so since $\mathfrak{M}, s \models \xi$ it is also true that $\mathfrak{M}', s \models \xi$ and hence $\mathfrak{M}', s \models \zeta(a)$. But since s' differs from s only in assigning to x the same thing as $\mathfrak{M}'$ generally assigns to a , and $\zeta(a)$ differs from $\zeta(x)$ by containing a where $\zeta(x)$ contains x free, and $\mathfrak{M}, s' \models \zeta(x)$ it follows that $\mathfrak{M}', s \models \zeta(a)$. This is a contradiction. Hence, $\mathfrak{M}$ must make $\xi \rightarrow \forall x \zeta(x)$ (i.e., ψ_i) true after all, and the same follows for any other model of Γ . Hence ψ_i is true for every model of Γ .

- (5) ψ_i follows from a previous member of the series by the second form of QR. ψ_i takes the form $\exists x \zeta(x) \rightarrow \xi$, and there is a previous wff in the ψ -series of the form $\zeta(a) \rightarrow \xi$, where a is in neither Γ nor ξ . By the inductive hypothesis, $\zeta(a) \rightarrow \xi$ is true in every model of Γ . Suppose for reductio that $\exists x \zeta(x) \rightarrow \xi$ is not true in some model $\mathfrak{M}$ of Γ . Then there is some s such that $\mathfrak{M}, s \models \exists x \zeta(x)$ and $\mathfrak{M}, s \not\models \xi$. There is some s' differing from s only in what it assigns to x such that $\mathfrak{M}, s' \models \zeta(x)$. Again consider structure $\mathfrak{M}'$ just like $\mathfrak{M}$ except $a^{\mathfrak{M}'} = \text{Val}_s^{\mathfrak{M}}(x)$. It is a model of Γ as well, since Γ does not contain a . So it makes $\zeta(a) \rightarrow \xi$ true, and $\mathfrak{M}', s \models \zeta(a) \rightarrow \xi$. But ξ does not contain a , and since $\mathfrak{M}, s \not\models \xi$ it must also be that $\mathfrak{M}', s \not\models \xi$. Then it must be that $\mathfrak{M}', s \not\models \zeta(a)$. But $\zeta(a)$ is just like $\zeta(x)$ except containing a where $\zeta(x)$ contains x free, and since $\mathfrak{M}, s' \models \zeta(x)$ it follows that $\mathfrak{M}', s \models \zeta(a)$, which contradicts our earlier result. Hence, the assumption that $\exists x \zeta(x) \rightarrow \xi$ is not true in some model $\mathfrak{M}$ of Γ must be mistaken, and $\exists x \zeta(x) \rightarrow \xi$, i.e., ψ_i , must be true in all such models.

These five cases exhaust the possible steps of the derivation, and so it is true for all steps ψ_i in the ψ -series that $\Gamma \models \psi_i$. φ is the last member of this series, and so $\Gamma \models \varphi$. *

Corollary: All theorems of pure logic are logically valid, i.e., if $\vdash \varphi$ then $\models \varphi$.

Proof: If $\vdash \varphi$ then $\emptyset \vdash \varphi$, and so by the above $\emptyset \models \varphi$. But every structure is a model of $\emptyset$ trivially, and so every structure makes φ true, i.e., $\models \varphi$. *

Definition: A set of sentences Γ is **inconsistent** iff $\Gamma \vdash \perp$.

Corollary (Consistency): (a) If Γ is inconsistent, then Γ is unsatisfiable, and (b) $\nvdash \perp$.

Proof: Suppose $\Gamma \vdash \perp$; it follows that $\Gamma \models \perp$, and then every model of Γ makes $\perp$ true. If Γ were satisfiable, some structure would make $\perp$ true, which is impossible. Similarly, if $\vdash \perp$, then $\perp$ would be logically valid, and true in every structure, which is impossible. *

HOMEWORK 12

Prove (a) $\Gamma \vdash \neg \varphi$ iff $\Gamma \cup \{\varphi\}$ is inconsistent; and (b) $\Gamma \cup \{\varphi\}$ and $\Gamma \cup \{\neg \varphi\}$ are both inconsistent iff Γ is inconsistent.

Lemmas for completeness

The opposite direction from soundness is completeness; this is the more difficult direction, and we'll need some definitions and lemmas first. Completeness for first-order predicate logic was first shown by Kurt Gödel in 1930, but the strategy we're using derives from Leon Henkin.

Definition: If $\langle k_1, \dots, k_n \rangle$ is an n -tuple of natural numbers, its **code** is the product of the first n primes raised respectively to the powers of $(k_1+1), \dots, (k_n+1)$.

Example: The code for $\langle 4, 10, 3, 0 \rangle$ is $2^5 \cdot 3^{11} \cdot 5^4 \cdot 7^1$, or 24,800,580,000.

The reason we add one to the exponents is to ensure different results when the sequence has 0's at the end, since, e.g., $2^4 \cdot 3^{10} \cdot 5^3 \cdot 7^0 = 2^4 \cdot 3^{10} \cdot 5^3$ but $2^5 \cdot 3^{11} \cdot 5^4 \cdot 7^1 \neq 2^5 \cdot 3^{11} \cdot 5^4$. No two sequences will have the same code this way, as numbers with different prime factorizations are always different.

Result: For any first-order language $\mathcal{L}$, the set of its sentences is denumerable, i.e., we can generate a one-one correspondence between the set of natural numbers and the set of its sentences.

sentence that is the one with the n th lowest is a 1-1 function between natural numbers and sentences of $\mathcal{L}$. Hence, the set of sentences of $\mathcal{L}$ is denumerable. *

We'll be doing a lot more with Gödel numbers in our next unit, unsurprisingly. In this unit, it plays only a very minor role in our next result.

Definition: A set of sentences Γ is **complete** iff for every closed wff φ , either $\varphi \in \Gamma$ or $\neg\varphi \in \Gamma$.

The above is not to be confused with “complete” in the sense of capturing every semantically valid sentence.

Definition: A set of sentences Γ is **universal** iff for every wff $\varphi(x)$ that contains at most x free, if it is the case for all ground terms t that $\varphi(t) \in \Gamma$, then $\forall x\varphi(x) \in \Gamma$.

In its treatment of the completeness proofs, the book talks about “saturated” sets instead of universal sets, which is a distinct but related notion. My proof is subtly different, and somewhat more streamlined than theirs, in the interest of saving time.

Result (Lindenbaum Extension Lemma):

If Γ is a consistent set of sentences, then there is a set of sentences Δ such that: (a) $\Gamma \subseteq \Delta$, (b) Δ is consistent, (c) Δ is complete, and (d) Δ is universal.

Proof: (1) Assume that Γ is a consistent set of sentences.

(2) We need an infinite list of constants that do not occur in Γ , adding new constants to the language if need be. I will use $e, e_1, \dots, e_n, \dots$. (You will note that “e” was not included in the definition of a constant on p. 7.)

(3) By the denumerability of the set of sentences of the language, we can arrange them in an infinite sequence:

$$\varphi_1, \varphi_2, \varphi_3, \dots, \text{ etc.}$$

Proof: (1) We begin by assigning a different n -tuple to each simple sign in the language as follows:

- The logical operators, constants and punctuation marks on the list $\perp \top \neg \vee \wedge \rightarrow \leftrightarrow \forall \exists = ()$, are respectively assigned $\langle 0, 1 \rangle$ through $\langle 0, 13 \rangle$.
- Each variable is assigned $\langle 1, i, l \rangle$, where i is its subscript if it has one (0 otherwise) and l is 1, 2, 3, 4, 5, or 6 depending on whether the variable uses letter “u”, “v”, “w”, “x”, “y” or “z”.
- Each constant is assigned $\langle 2, i, l \rangle$ where i is its subscript if it has one (0 otherwise), and l is 1, 2, 3, 4, or 5 depending on whether the letter used is “a”, “b”, “c”, “d” or “e”.
- Each function letter is assigned $\langle 3, n, i, l \rangle$, where n is the number of its superscript, i the number of its subscript (0 if none), and l is 1–7 depending on the letter “f”–“l”.
- Each predicate is assigned $\langle 4, n, i, l \rangle$, where n is the number of its superscript, i the number of its subscript (0 if none), and l is 1–20 depending on the letter “A”–“T”.

(2) Each simple sign then has a “symbol code”, i.e., the code for the n -tuple we have assigned it. E.g., the “symbol code” for the predicate A^1 is $2^5 \cdot 3^2 \cdot 5^1 \cdot 7^2$.

(3) For each *string* of simple symbols, we assign it a unique number, called its **Gödel number**, which is the code for the sequence of symbol codes for the symbols making it up in order. E.g., for the wff “ $A^1(a)$ ” the Gödel number is $2^{(2^5 \cdot 3^2 \cdot 5^1 \cdot 7^2)+1} \cdot 3^{(2^1 \cdot 3^{12})+1} \cdot 5^{(2^3 \cdot 3^1 \cdot 5^2)+1} \cdot 7^{(2^1 \cdot 3^{13})+1}$, which is some stupidly huge number for such a simple sentence.

(4) For each natural number n , there is an n th lowest number that is the Gödel number of a sentence of $\mathcal{L}$. The function mapping each n to the

Making use of this sequence, let us recursively define an infinite sequence of sets:

$$\Gamma_0, \Gamma_1, \Gamma_2, \dots, \text{ etc.}$$

As follows:

- a) Let $\Gamma_0 = \Gamma$.
- b) We define Γ_{n+1} in terms of Γ_n in one of the following three ways:
 - (i) if $\Gamma_n \cup \{\varphi_{n+1}\}$ is consistent, then let $\Gamma_{n+1} = \Gamma_n \cup \{\varphi_{n+1}\}$;
 - (ii) if $\Gamma_n \cup \{\varphi_{n+1}\}$ is inconsistent and φ_{n+1} does not take the form $\forall x \psi(x)$, then let $\Gamma_{n+1} = \Gamma_n \cup \{\neg \varphi_{n+1}\}$;
 - (iii) if $\Gamma_n \cup \{\varphi_{n+1}\}$ is inconsistent and φ_{n+1} does take the form $\forall x \psi(x)$, then let $\Gamma_{n+1} = \Gamma_n \cup \{\neg \varphi_{n+1}\} \cup \{\neg \psi(e_m)\}$, where “ e_m ” is the first e -constant that does not occur in Γ_n .

- (4) Let Δ be the union of all of the members of the Γ -sequence (i.e., $\Gamma_0 \cup \Gamma_1 \cup \Gamma_2 \cup \dots$ etc.)
- (5) Obviously, $\Gamma \subseteq \Delta$. This establishes part (a) of the consequent of the Lemma.

- (6) Every member of the Γ -sequence is consistent. We prove this by mathematical induction.

Base step: Γ_0 is Γ , and it is consistent *ex hypothesi*.

Induction step: Suppose Γ_n is consistent. It follows that Γ_{n+1} is consistent by a proof by cases:

Case (i): $\Gamma_{n+1} = \Gamma_n \cup \{\varphi_{n+1}\}$ and $\Gamma_n \cup \{\varphi_{n+1}\}$ is consistent, so Γ_{n+1} is consistent.

Case (ii): $\Gamma_{n+1} = \Gamma_n \cup \{\neg \varphi_{n+1}\}$, and $\Gamma_n \cup \{\varphi_{n+1}\}$ is inconsistent.

- Hence $\Gamma_n \cup \{\varphi_{n+1}\} \vdash \perp$.
- By the deduction theorem, $\Gamma_n \vdash \varphi_{n+1} \rightarrow \perp$.
- By RedAx and MP, $\Gamma_n \vdash \neg \varphi_{n+1}$.
- Suppose for *reductio* that Γ_{n+1} is inconsistent.
- This means that $\Gamma_n \cup \{\neg \varphi_{n+1}\} \vdash \perp$.
- By reasoning parallel to the above, and DNE, $\Gamma_n \vdash \varphi_{n+1}$.
- But from $\Gamma_n \vdash \neg \varphi_{n+1}$ and $\Gamma_n \vdash \varphi_{n+1}$ we get $\Gamma_n \vdash \perp$, so Γ_n is inconsistent.
- This contradicts the inductive hypothesis. Hence Γ_{n+1} is consistent.

tent.

Case (iii): $\Gamma_{n+1} = \Gamma_n \cup \{\neg \varphi_{n+1}\} \cup \{\neg \psi(e_m)\}$, $\Gamma_n \cup \{\varphi_{n+1}\}$ is inconsistent and φ_{n+1} takes the form $\forall x \psi(x)$.

- By the same reasoning as in the previous case, $\Gamma_n \vdash \neg \varphi_{n+1}$.
- Suppose for *reductio* that Γ_{n+1} is inconsistent.
- So $\Gamma_n \cup \{\neg \varphi_{n+1}\} \cup \{\neg \psi(e_m)\} \vdash \perp$.
- By DT, $\Gamma_n \cup \{\neg \psi(e_m)\} \vdash \neg \varphi_{n+1} \rightarrow \perp$.
- By MP, $\Gamma_n \cup \{\neg \psi(e_m)\} \vdash \perp$.
- By DT, $\Gamma_n \vdash \neg \psi(e_m) \rightarrow \perp$.
- By RedAx, and DNE, $\Gamma_n \vdash \psi(e_m)$.
- e_m is not included in Γ_n . Hence, by UG, $\Gamma_n \vdash \forall x \psi(x)$, or in other words, $\Gamma_n \vdash \varphi_{n+1}$.
- We already have $\Gamma_n \vdash \neg \varphi_{n+1}$, so $\Gamma_n \vdash \perp$.
- So Γ_n is inconsistent, which contradicts the inductive hypothesis.
- Hence Γ_{n+1} is consistent.

- (7) It follows from (6) that Δ is consistent.

- a) Note that the Γ -sequence is constantly expanding: For all j and k such that $j < k$, $\Gamma_j \subseteq \Gamma_k$. Crudely, Δ can be thought of as the upper limit of the expansion.
- b) So every finite subset of Δ is a subset of some Γ_i for some suitably large i .
- c) However, every derivation from Δ has only a finite number of steps, and hence only makes use of a finite subset of Δ .
- d) If $\Delta \vdash \perp$, for some suitably large i , it would have to be that $\Gamma_i \vdash \perp$.
- e) This is impossible because all the members of the Γ -sequence are consistent by (6).
- f) Hence, Δ is consistent.
- g) This establishes part (b) of the consequent of the Lemma.

- (8) Δ is obviously complete as well.

- a) All sentences are members of the sequence $\varphi_1, \varphi_2, \dots$, etc.
- b) For each φ_i , either it or its negation is a member of Γ_i , and $\Gamma_i \subseteq \Delta$.
- c) So for all closed wffs $\varphi_1, \varphi_2, \dots$, etc., either it or its negation is included in Δ .
- d) This establishes part (c) of the consequent

of the Lemma.

- (9) Finally, Δ is also universal.
- We show this by *reductio*. Suppose otherwise, i.e., suppose that there is a wff $\psi(x)$ that contains at most x free, such that for all ground terms t , $\psi(t) \in \Delta$, but $\forall x \psi(x) \notin \Delta$.
 - $\forall x \psi(x)$ is closed, so because Δ is complete, it must be that $\neg \forall x \psi(x) \in \Delta$.
 - Because $\forall x \psi(x)$ is closed, it also follows that $\forall x \psi(x)$ is a member of the φ -sequence, i.e., $\forall x \psi(x)$ is φ_{n+1} for some number n .
 - Obviously, however, since $\forall x \psi(x) \notin \Delta$, it follows that Γ_{n+1} is not obtained from Γ_n using case (i).
 - Nor was it obtained using case (ii), since φ_{n+1} is of the form $\forall x \psi(x)$.
 - This leaves case (iii), so Γ_{n+1} is $\Gamma_n \cup \{\neg \varphi_{n+1}\} \cup \{\neg \psi(e_m)\}$ for some m .
 - Hence, $\neg \psi(e_m) \in \Gamma_{n+1}$ and so $\neg \psi(e_m) \in \Delta$.
 - But by our assumption, it holds for all ground terms t that $\psi(t) \in \Delta$.
 - The constant “ e_m ” is a ground term, so $\psi(e_m) \in \Delta$.
 - Hence, both $\Delta \vdash \neg \psi(e_m)$ and $\Delta \vdash \psi(e_m)$, and so $\Delta \vdash \perp$.
 - However, this is impossible, because we have already shown Δ to be consistent.
 - Our supposition has been shown to be impossible, hence Δ is universal.
 - This establishes part (d) of the consequent of the Lemma. *
- (10) By suitably defining Δ , we have shown each of parts (a)–(d) of the consequent of the Lemma on the basis of the assumption of its antecedent. Hence, the Lemma is established. *

The Lindenbaum extension lemma is another one of my favorites: any consistent story can be filled in to be a complete story of the way things are.

We now show that if you have a complete story about the way things are, you can turn it into a model.

Definition: A structure $\mathfrak{M}$ is a **denumerable structure** iff its domain of quantification $|\mathfrak{M}|$ is denumerable.

Result: If Δ is a consistent and complete set of sentences, then Δ is closed under entailment, i.e., for every sentence φ , if $\Delta \vdash \varphi$ then $\varphi \in \Delta$.

Proof: Assume otherwise, i.e., $\Delta \vdash \varphi$ but $\varphi \notin \Delta$. Then, because φ is complete, $\neg \varphi \in \Delta$, and so $\Delta \vdash \neg \varphi$. But then $\Delta \vdash \perp$, which is impossible, as Δ is consistent. *

Result (The Term Model Lemma): If Δ is a consistent, complete, and universal set of sentences, then there is at least one denumerable structure that is a model for Δ .

- Proof:* (1) Assume that Δ is a consistent, complete, and universal set of closed wffs. (It is thus closed under entailment.) We can then describe a denumerable structure $\mathfrak{M}$, called the “term model for Δ ” using the following procedure.
- Essentially, we’ll let all the ground terms of the language stand for *themselves*. (Another possible way of constructing a model would be to let each ground term stand for its Gödel number. However, let us proceed using the former method.)
 - Let the domain of quantification $|\mathfrak{M}|$ of $\mathfrak{M}$ be the set of ground terms of the language. Note that there are denumerably many ground terms, so $\mathfrak{M}$ is a denumerable structure.
 - For each constant c , let $c^{\mathfrak{M}}$ be c itself.
 - For each function letter f^n , let $f^{n\mathfrak{M}}$ be that n -place operation on $|\mathfrak{M}|$ which includes all ordered pairs of the form

$$\langle \langle t_1, \dots, t_n \rangle, f^n(t_1, \dots, t_n) \rangle$$

- i.e., the operation that has the ground term $f^n(t_1, \dots, t_n)$ as value for $\langle t_1, \dots, t_n \rangle$ as argument.
- For each predicate F^n , let $F^{n\mathfrak{M}}$ be that subset of $|\mathfrak{M}|^n$ that includes the n -tuple $\langle t_1, \dots, t_n \rangle$ iff the atomic wff $F^n(t_1, \dots, t_n)$ is included in Δ .

- (7) We must now prove that $\mathfrak{M}$ is a model for Δ , i.e., that for all sentences φ , if $\varphi \in \Delta$, then $\mathfrak{M} \models \varphi$. We will actually prove something stronger, i.e., that for all sentences φ , $\varphi \in \Delta$ iff $\mathfrak{M} \models \varphi$. We prove this by formula induction.

Base step: φ is a atomic.

- The cases of $\perp$ and $\top$ are trivial. Because Δ is consistent, $\perp \notin \Delta$ and $\mathfrak{M} \not\models \perp$, and similarly $\top \in \Delta$ and $\mathfrak{M} \models \top$.
- Focus on the cases where φ takes the form $F^n(t_1, \dots, t_n)$ where $t_1, \dots, t_n$ are ground terms.
- Because ground terms contain no variables, all sequences in $\mathfrak{M}$ will associate each t_i with itself.
- So by the definition of satisfaction, all sequences in $\mathfrak{M}$ will satisfy φ iff $\langle t_1, \dots, t_n \rangle \in F^n \mathfrak{M}$.
- By the definition of truth, $\mathfrak{M} \models \varphi$ iff $\langle t_1, \dots, t_n \rangle \in (F)^{\mathfrak{M}}$.
- By our characterization of $\mathfrak{M}$ under (6) above, $\langle t_1, \dots, t_n \rangle \in F^n \mathfrak{M}$ iff $F(t_1, \dots, t_n) \in \Delta$.
- So $F(t_1, \dots, t_n) \in \Delta$ iff $\mathfrak{M} \models \varphi$, i.e., $\varphi \in \Delta$ iff $\mathfrak{M} \models \varphi$.

Induction step: Assume as inductive hypothesis that it holds for all sentences ψ with fewer connectives than φ , that $\psi \in \Delta$ iff $\mathfrak{M} \models \psi$. We will then show that it holds for φ that $\varphi \in \Delta$ iff $\mathfrak{M} \models \varphi$ as well.

This proceeds by a proof by cases on the make-up of φ .

Case (a): φ takes the form $\neg\psi$, where ψ has one fewer connective than φ .

- By the inductive hypothesis, $\psi \in \Delta$ iff $\mathfrak{M} \models \psi$.
- Because Δ is consistent, if $\varphi \in \Delta$, then $\psi \notin \Delta$.
- Because Δ is complete, if $\psi \notin \Delta$, then $\varphi \in \Delta$.
- So $\psi \notin \Delta$ iff $\varphi \in \Delta$.
- Hence $\varphi \in \Delta$ iff $\mathfrak{M} \not\models \psi$.
- Since ψ is closed, $\mathfrak{M} \models \neg\psi$ iff $\mathfrak{M} \not\models \psi$.
- Hence, $\varphi \in \Delta$ iff $\mathfrak{M} \models \neg\psi$, i.e., $\varphi \in \Delta$ iff $\mathfrak{M} \models \varphi$.

Case (b): φ takes the form $\psi \vee \theta$, where ψ and θ have fewer connectives. First, if $\varphi \in \Delta$ then $\mathfrak{M} \models \varphi$:

- Assume $\varphi \in \Delta$, i.e., $\psi \vee \theta \in \Delta$.
- It must be that either $\psi \in \Delta$ or $\theta \in \Delta$. Otherwise, because Δ is complete, we would have both $\neg\psi \in \Delta$ and $\neg\theta \in \Delta$. But then by DS, $\Delta \vdash \theta$ and then $\Delta \vdash \perp$, which is impossible.
- By the inductive hypothesis, either $\mathfrak{M} \models \psi$ or $\mathfrak{M} \models \theta$. Either way, $\mathfrak{M} \models \psi \vee \theta$, i.e., $\mathfrak{M} \models \varphi$.

Likewise if $\mathfrak{M} \models \varphi$ then $\varphi \in \Delta$.

- Assume $\mathfrak{M} \models \varphi$, i.e., $\mathfrak{M} \models \psi \vee \theta$.
- By the semantics for $\vee$, Either $\mathfrak{M} \models \psi$ or $\mathfrak{M} \models \theta$.
- Hence, by the inductive hypothesis, either $\psi \in \Delta$ or $\theta \in \Delta$. Either way, $\Delta \vdash \psi \vee \theta$ by Add.
- By closure, $\psi \vee \theta \in \Delta$, i.e., $\varphi \in \Delta$.

Putting these two results together, we get that $\varphi \in \Delta$ iff $\mathfrak{M} \models \varphi$.

Case (c): φ takes the form $\psi \wedge \theta$, where ψ and θ have fewer connectives. We can show that $\varphi \in \Delta$ iff $\mathfrak{M} \models \varphi$. (This is homework.)

Case (d): φ takes the form $\psi \rightarrow \theta$, where ψ and θ are sentences with fewer connectives. First we prove that if $\varphi \in \Delta$ then $\mathfrak{M} \models \varphi$.

- Suppose $\varphi \in \Delta$.
- Since Δ is complete and consistent, $\psi \in \Delta$ or $\neg\psi \in \Delta$, but not both, and likewise with θ .
- However, because $\psi \rightarrow \theta \in \Delta$ and Δ is consistent, it cannot be that both $\psi \in \Delta$ and $\neg\theta \in \Delta$, or else MP would give us $\Delta \vdash \theta$ and $\Delta \vdash \perp$.
- It follows that either $\psi \notin \Delta$ or $\theta \in \Delta$.
- By the inductive hypothesis, either $\mathfrak{M} \not\models \psi$ or $\mathfrak{M} \models \theta$.
- By the semantics for conditionals, it follows either way that $\mathfrak{M} \models \psi \rightarrow \theta$, i.e., $\mathfrak{M} \models \varphi$.

Now we prove that if $\mathfrak{M} \models \varphi$ then

$\varphi \in \Delta$.

- Suppose $\mathfrak{M} \models \varphi$, i.e., $\mathfrak{M} \models \psi \rightarrow \theta$.
- Because ψ and θ are closed, either $\mathfrak{M} \models \psi$ or $\mathfrak{M} \models \theta$.
- By the inductive hypothesis, $\psi \notin \Delta$ or $\theta \in \Delta$.
- If $\psi \notin \Delta$, then, because Δ is complete, $\neg\psi \in \Delta$. Then $\Delta \vdash \psi \rightarrow \theta$ by FA, and by closure, $\psi \rightarrow \theta \in \Delta$.
- If $\theta \in \Delta$, then by TC, $\Delta \vdash \psi \rightarrow \theta$, and by closure, $\psi \rightarrow \theta \in \Delta$.
- So either way, $\psi \rightarrow \theta \in \Delta$, i.e., $\varphi \in \Delta$.

Case (e): φ takes the form $\psi \leftrightarrow \theta$, where ψ and θ have fewer connectives. We can show that $\varphi \in \Delta$ iff $\mathfrak{M} \models \varphi$. (More homework.)

Case (f): φ takes the form $\forall x\psi(x)$, where $\psi(x)$ contains fewer connectives.

First we prove that if $\varphi \in \Delta$ then $\mathfrak{M} \models \varphi$.

- Suppose $\varphi \in \Delta$, i.e., $\forall x\psi(x) \in \Delta$.
- Because $\psi(x)$ contains at most x free, for all ground terms t , $\psi(t)$ is a sentence.
- Because Δ is complete, for all ground terms t , either $\psi(t) \in \Delta$ or $\neg\psi(t) \in \Delta$.
- However, since Δ is consistent, it must be that for all ground terms t , $\psi(t) \in \Delta$.
- By the inductive hypothesis, for all ground terms t , $\mathfrak{M} \models \psi(t)$.
- Because the domain of quantification for $\mathfrak{M}$ is the set of ground terms, and every ground term is interpreted as standing for itself, a variable assignment s will satisfy $\psi(x)$ iff $\psi(t)$ is true where t is $\text{Val}_s^{\mathfrak{M}}(x)$.
- Because $\psi(t)$ is true in $\mathfrak{M}$ for all ground terms t , all variable assignments of $\mathfrak{M}$ will satisfy $\psi(x)$, and will also satisfy $\forall x\psi(x)$.

• Hence, $\mathfrak{M} \models \forall x\psi(x)$, i.e., $\mathfrak{M} \models \varphi$.

We now prove that if $\mathfrak{M} \models \varphi$ then $\varphi \in \Delta$.

- Suppose $\mathfrak{M} \models \varphi$, i.e., all variable assignments satisfy $\forall x\psi(x)$.
- Hence, all variable assignments satisfy $\psi(x)$, regardless of what entity in the domain gets assigned to x .
- Because the domain of quantification for $\mathfrak{M}$ is the set of ground terms, and every ground term is interpreted as standing for itself, variable assignment s will satisfy $\psi(x)$ iff $\psi(t)$ is true for that ground term t which is $\text{Val}_s^{\mathfrak{M}}(x)$.
- So, for all ground terms t , $\mathfrak{M} \models \psi(t)$.
- By the inductive hypothesis, it follows that, for all ground terms t , $\psi(t) \in \Delta$.
- Because Δ is universal, it follows that $\forall x\psi(x) \in \Delta$, i.e., $\varphi \in \Delta$.

Case (g): φ takes the form $\exists x\psi(x)$, where $\psi(x)$ contains fewer connectives.

First we prove that if $\varphi \in \Delta$ then $\mathfrak{M} \models \varphi$.

- Suppose $\varphi \in \Delta$, i.e., $\exists x\psi(x) \in \Delta$.
- It cannot be that $\neg\psi(t) \in \Delta$ for all ground terms t . For if this were true, then, because Δ is universal, $\forall x\neg\psi(x) \in \Delta$, hence $\Delta \vdash \neg\exists x\psi(x)$ by CQ and MP, and then $\Delta \vdash \perp$, which is impossible.
- Hence for some closed term t , $\neg\psi(t) \notin \Delta$ and therefore $\psi(t) \in \Delta$.
- By the inductive hypothesis, $\mathfrak{M} \models \psi(t)$. Hence for any variable assignment s , $\mathfrak{M}, s \models \exists x\psi(x)$, since for the s' differing from s in assigning t to x , $\mathfrak{M}, s' \models \psi(x)$.
- Therefore $\mathfrak{M} \models \exists x\psi(x)$, i.e., $\mathfrak{M} \models \varphi$.

Now we prove that if $\mathfrak{M} \models \varphi$ then $\varphi \in \Delta$.

- Suppose $\mathfrak{M} \models \varphi$, i.e., $\mathfrak{M} \models$

$\exists x\psi(x)$.

- For any assignment s , there is one differing from it s' at its assignment to x that satisfies $\psi(x)$. Because the domain of quantification is the set of ground terms, $\text{Val}_s^{\mathfrak{M}}(x) = t$ for some ground term. Clearly, $\mathfrak{M}, s' \vdash \psi(t)$ as well, and because $\psi(t)$ is closed, $\mathfrak{M} \models \psi(t)$.
- By the inductive hypothesis, $\psi(t) \in \Delta$.
- By EG, $\Delta \vdash \exists x\psi(x)$, and by closure, $\exists x\psi(x) \in \Delta$, i.e., $\varphi \in \Delta$

(8) By induction, regardless of φ 's length, $\varphi \in \Delta$ iff $\mathfrak{M} \models \varphi$. So $\mathfrak{M}$ is a model for Δ . This establishes the Lemma. $\ast$

HOMEWORK 13

Fill in the argument for cases (c) and (e) above; compare cases (a), (b) and (d) for the general lines of argument. For case (e), you may cite the $\leftrightarrow$ “axioms” given on p. 18, or the derived rules that follow from them (p. 19) as needed.

Completeness and compactness

We now get a result key for completeness. In fact, some books label this result itself “completeness”.

Result (Satisfiability): *If Γ is a consistent set of sentences, then there is a denumerable structure that is a model of Γ .*

Proof: Assume Γ is consistent. By the Lindenbaum Extension Lemma, there is a consistent, complete and universal set Δ , of which it is a subset. By the Term Model lemma, there is a denumerable structure $\mathfrak{M}$ which is a model for Δ . This means $\mathfrak{M}$ makes every sentence in Δ true, and therefore also makes every sentence in Γ true. $\ast$

Corollary (Löwenheim-Skolem Theorem): *If a set Γ of sentences is satisfiable, then it has a denumerable structure as a model.*

Proof: Suppose Γ is satisfiable. By our Consistency result earlier, it must be consistent. Therefore, by Satisfiability, it has a denumerable structure as a model. $\ast$

The suprising thing about this result is that it shows that no matter what kind of interpretation a theory (“collection of sentences”) is intended to have, if it has any kind of interpretation making it true, it has one of denumerable size. This is very counterintuitive if the “intended” models quantify over larger infinities, such as a theory for the mathematics of real numbers.

Result (Completeness): *If $\Gamma \models \varphi$ then $\Gamma \vdash \varphi$.*

- Proof:* (1) Suppose $\Gamma \models \varphi$.
 (2) Suppose for reductio that $\Gamma \cup \{\neg\varphi\}$ is consistent.
 (3) Then, by Satisfiability, there is model $\mathfrak{M}$ for $\Gamma \cup \{\neg\varphi\}$, and so $\mathfrak{M} \models \neg\varphi$.
 (4) But since $\mathfrak{M}$ is a model for Γ and $\Gamma \models \varphi$, it follows that $\mathfrak{M} \models \varphi$.
 (5) Every variable assignment s for $\mathfrak{M}$ both satisfies and does not satisfy φ , which is impossible.
 (6) Hence, $\Gamma \cup \{\neg\varphi\}$ is inconsistent.
 (7) This means that $\Gamma \cup \{\neg\varphi\} \vdash \perp$.
 (8) By the deduction theorem $\Gamma \vdash \neg\varphi \rightarrow \perp$.
 (9) By RedAx and DNE, $\Gamma \vdash \varphi$. $\ast$

Corollary ($\vdash/\models$ Bridge): $\Gamma \vdash \varphi$ iff $\Gamma \models \varphi$

Proof: Follows directly from Soundness and Completeness. $\ast$

Corollary: $\vdash \varphi \text{ iff } \models \varphi$

Proof: By the above, $\emptyset \vdash \varphi \text{ iff } \emptyset \models \varphi$. By definition, $\vdash \varphi \text{ iff } \emptyset \vdash \varphi$, and clearly $\models \varphi \text{ iff } \emptyset \models \varphi$, as all structures are models of $\emptyset$ trivially. *

Definition: A set of sentences Γ is **finitely satisfiable** iff every finite subset Δ of Γ is satisfiable.

Corollary (Compactness): (a) $\Gamma \models \varphi$ iff there is a finite subset Δ of Γ such that $\Delta \models \varphi$, and (b) Γ is satisfiable iff it is finitely satisfiable.

Proof: For the right-to-left part of (a), notice that every structure making every member of Γ true also makes every member of Δ true, and so must make φ true.

The left-to-right direction is more interesting:

- (1) Suppose $\Gamma \models \varphi$.
- (2) By completeness, $\Gamma \vdash \varphi$.
- (3) This means there is an axiomatic derivation of φ from Γ . This derivation is of finite length, and can use only finitely many members from Γ .
- (4) Let Δ be the members used in the derivation. For reasons given above, Δ is finite.
- (5) The members of Δ were taken from Γ , so $\Delta \subseteq \Gamma$.
- (6) Clearly, $\Delta \vdash \varphi$, and so, by Soundness, $\Delta \models \varphi$.

For (b), left-to-right, note if Γ is satisfiable, then some structure $\mathfrak{M}$ makes every member of Γ true; this structure also makes every member of any subset Δ of Γ true, and so each such set Δ is satisfiable.

For (b), right-to-left:

- (1) Suppose Γ is finitely satisfiable.
- (2) Suppose for reductio that Γ is inconsistent.
- (3) Then $\Gamma \vdash \perp$, and by soundness, $\Gamma \models \perp$.
- (4) But then, by (a), there is a finite subset Δ of Γ , $\Delta \models \perp$.
- (5) This means Δ is unsatisfiable, which is impossible, as Γ is finitely satisfiable.
- (6) Hence Γ is consistent, and by Satisfiability, Γ is satisfiable. *

Compactness is also seen as a surprising result. The book also gives direct proofs of compactness that do not go “through” completeness.

A related but relatively obvious result I’m throwing in just because it will be useful later:

Result (Logical Conditionalization): $\Gamma \vdash \varphi$ iff there is a finite number of members of Γ , $\psi_1, \dots, \psi_n$ such that $\vdash (\psi_1 \wedge \dots \wedge \psi_n) \rightarrow \varphi$.

Proof: (1) Suppose $\Gamma \vdash \varphi$; only a finite number of members of Γ can be used in the proof; let them be $\psi_1, \dots, \psi_n$.

(2) Hence $\psi_1, \dots, \psi_n \vdash \varphi$

(3) By successive applications of the deduction theorem, $\vdash \psi_1 \rightarrow (\psi_2 \rightarrow \dots (\psi_n \rightarrow \varphi))$, and by Imp, $\vdash (\psi_1 \wedge \dots \wedge \psi_n) \rightarrow \varphi$.

(4) For the other direction, assume $\vdash (\psi_1 \wedge \dots \wedge \psi_n) \rightarrow \varphi$, where each $\psi_i \in \Gamma$.

(5) By Conj and MP, $\Gamma \vdash \varphi$. *

HOMEWORK 14

Suppose that φ is a contingent sentence without quantifiers. For example, φ might be “ $Fa \rightarrow Ga$ ”, or “ $(R(a, f(a)) \wedge T(b, a)) \leftrightarrow (T(b, a) \leftrightarrow R(a, f(a)))$ ”. Now suppose we form a schema by replacing each *distinct* atomic formula in φ with a distinct schematic letter. For our examples, we’d get, respectively, “ $\psi_1 \rightarrow \psi_2$ ” and “ $(\psi_1 \wedge \psi_2) \leftrightarrow (\psi_2 \leftrightarrow \psi_1)$ ”. Let Γ be the set of all instances of the resulting schema. (E.g., if φ is “ $Fa \rightarrow Ga$ ”, then Γ is the set of all conditionals.) Assuming only that φ is contingent (i.e., not a tautology or self-contradiction, truth-table-wise), show that Γ is inconsistent.

5. Identity Logic

Syntax and semantics

To capture the laws of identity logic, we just need to treat “ $=$ ”, which we included in our definition of predicates, in a special way. (So far, it’s just been a

predicate that a structure could interpret any way it sees fit.)

We write $t_1 = t_2$ as an alternative to $=^2(t_1, t_2)$, and $t_1 \neq t_2$ as an alternative to $\neg =^2(t_1, t_2)$.

Definition: A structure $\mathfrak{M}$ is **identity-normal** iff $=^2_{\mathfrak{M}}$ is the identity relation for the domain of $\mathfrak{M}$, i.e., the set of ordered pairs $\langle A, A \rangle$ for every A where $A \in |\mathfrak{M}|$.

Definition: A sentence φ is **identity-valid** iff for every identity-normal structure $\mathfrak{M}$, $\mathfrak{M} \models \varphi$. I abbreviate this $\models^= \varphi$.

Definition: A sentence φ is an **identity-logic-consequence** of Γ iff for every identity-normal-structure $\mathfrak{M}$, if $\mathfrak{M} \models \Gamma$ then $\mathfrak{M} \models \varphi$. I abbreviate this $\Gamma \models^= \varphi$.

The book usually means $\models^=$ by $\models$; we'll switch to that later too.

Identity axioms and derived rules

The book also axiomatizes identity logic slightly differently, but this way makes it slightly easier to apply our earlier results here.

Definition: A sentence is an **identity axiom** iff it is $\text{Ref}=\text{Ax}$ or has the form LLAx below, where φ does not contain any constants:

$$\forall x \, x = x \quad (\text{Ref}=\text{Ax})$$

$$\forall z_1 \dots \forall z_n \forall x \forall y (x = y \rightarrow (\varphi \rightarrow \varphi[y/x])) \quad (\text{LLAx})$$

(Here $z_1 \dots z_n$ are all the other free variables in φ besides x and y ; if there are none, then the $\forall z_1 \dots \forall z_n$ part is omitted.)

Definition: Let $I_{\mathcal{L}}$ be the set of identity axioms for first-order language $\mathcal{L}$.

Definition: A sentence φ is **identity-derivable** from Γ iff $\Gamma \cup I_{\mathcal{L}} \vdash \varphi$. We can abbreviate this as $\Gamma \vdash^= \varphi$.

$$\vdash^= t = t \quad (\text{Ref}=\text{=})$$

$$1. \forall x \, x = x$$

$$2. t = t$$

$$\text{Ref}=\text{Ax}$$

$$1 \text{ UI}$$

$$\vdash^= \forall x \forall y (x = y \rightarrow y = x) \quad (\text{Sym}=\text{T})$$

$$1. \forall z \forall x \forall y (x = y \rightarrow (x = z \rightarrow y = z)) \quad \text{LLAx}$$

$$2. a = b \rightarrow (a = a \rightarrow b = a) \quad 1 \text{ UI} \times 3$$

$$3. a = a \rightarrow (a = b \rightarrow b = a) \quad 2 \text{ Int}$$

$$4. a = a \quad \text{Ref}=\text{=}$$

$$5. a = b \rightarrow b = a \quad 4, 5 \text{ MP}$$

$$6. \forall x \forall y (x = y \rightarrow y = x) \quad 6 \text{ UG} \times 2$$

$$t_1 = t_2 \vdash^= t_2 = t_1 \quad (\text{Sym}=\text{=})$$

By $\text{UI} \times 2$ on $\text{Sym}=\text{T}$ and MP .

$$\vdash^= \forall x \forall y \forall z (x = y \rightarrow (y = z \rightarrow x = z)) \quad (\text{Trans}=\text{T})$$

$$1. \forall z \forall x \forall y (x = y \rightarrow (x \neq z \rightarrow y \neq z)) \quad \text{LLAx}$$

$$2. a = b \rightarrow (a \neq c \rightarrow b \neq c) \quad 1 \text{ UI} \times 3$$

$$3. a = b \rightarrow (b = c \rightarrow a = c) \quad 2 \text{ SL}$$

$$4. \forall x \forall y \forall z (x = y \rightarrow (y = z \rightarrow x = z)) \quad 3 \text{ UG} \times 3$$

$$t_1 = t_2, t_2 = t_3 \vdash^= t_1 = t_3 \quad (\text{Trans}=\text{=})$$

Follows from $\text{Trans}=\text{T}$, $\text{UI} \times 3$ and $\text{MP} \times 2$.

$$t_1 = t_2, \varphi(t_1) \vdash^= \varphi(t_2) \quad (\text{LL})$$

$$t_1 = t_2, \varphi(t_2) \vdash^= \varphi(t_1)$$

The first follows by LLAx , UI as needed and $\text{MP} \times 2$. Note that if $\varphi(x)$ contains constants we wish to remain, we can use the instance of LLAx with free variables $z_1, \dots, z_n$ in place of them, and start by instantiating to those constants. The second follows from the first and $\text{Sym}=\text{=}$.

In identity logic we can define numerically definite quantifiers: here y is the first variable not occurring in $\varphi(x)$.

$$\exists_0 x \varphi(x) \text{ is shorthand for } \neg \exists x \varphi(x)$$

$$\exists_{n+1} x \varphi(x) \text{ is shorthand for}$$

$$\exists x [\varphi(x) \wedge \exists_n y (y \neq x \wedge \varphi(y))]$$

So in particular:

$$\exists_1 x \varphi(x) \text{ abbreviates } \exists x [\varphi(x) \wedge \neg \exists y (y \neq x \wedge \varphi(y))]$$

HOMEWORK 15

Show that $\vdash^= \exists_1 x \varphi(x) \leftrightarrow \exists x \forall y (\varphi(y) \leftrightarrow y = x)$.

Identity logic metatheory

Result: Every member of $I_{\mathcal{L}}$ is identity-valid.

- Proof:* (1) Let $\mathfrak{M}$ be an identity-normal structure.
 (2) For $\text{Ref}=Ax$, note for any given assignment s , whatever $\text{Val}_s^{\mathfrak{M}}(x)$ is, $\langle \text{Val}_s^{\mathfrak{M}}(x), \text{Val}_s^{\mathfrak{M}}(x) \rangle \in =^{\mathfrak{M}}$, and so $\mathfrak{M}, s \models x = x$, and therefore $\mathfrak{M} \models \forall x x = x$.
 (3) Suppose for reductio that for some instance of LLAx , $\mathfrak{M} \not\models \forall z_1 \dots \forall z_n \forall x \forall y (x = y \rightarrow (\varphi \rightarrow \varphi[y/x]))$. Then there would have to be an variable assignment s where $\mathfrak{M}, s \not\models x = y \rightarrow (\varphi \rightarrow \varphi[y/x])$. Then $\mathfrak{M}, s \models x = y$ and $\mathfrak{M}, s \models \varphi$ but $\mathfrak{M}, s \not\models \varphi[y/x]$. However, if $\mathfrak{M}, s \models x = y$, then $\langle \text{Val}_s^{\mathfrak{M}}(x), \text{Val}_s^{\mathfrak{M}}(y) \rangle \in =^{\mathfrak{M}}$ and so, because $\mathfrak{M}$ is identity normal, $\text{Val}_s^{\mathfrak{M}}(x) = \text{Val}_s^{\mathfrak{M}}(y)$. But then since $\mathfrak{M}, s \models \varphi$ it must be that $\mathfrak{M}, s \models \varphi[y/x]$, which contradicts our earlier result.
 (4) Hence, for any identity-normal structure, $\mathfrak{M}$, all identity axioms are true, which means they are all identity-valid. *

Result (Identity soundness): If $\Gamma \vdash^= \varphi$ then $\Gamma \models^= \varphi$.

- Proof:* (1) Assume $\Gamma \vdash^= \varphi$, i.e., $\Gamma \cup I_{\mathcal{L}} \vdash \varphi$.
 (2) By our more generic Soundness result for first-order logic, $\Gamma \cup I_{\mathcal{L}} \models \varphi$.
 (3) Let $\mathfrak{M}$ be an arbitrary identity-normal model. Because all members of $I_{\mathcal{L}}$ are identity-valid, $\mathfrak{M} \models I_{\mathcal{L}}$. If $\mathfrak{M} \models \Gamma$, then, $\mathfrak{M} \models \Gamma \cup I_{\mathcal{L}}$, and thus $\mathfrak{M} \models \varphi$.
 (4) In other words, φ is true in all identity-normal models of Γ , and hence $\Gamma \models^= \varphi$. *

Corollary: If $\vdash^= \varphi$ then $\models^= \varphi$.

Result (Factoring lemma): For any structure (identity-normal or otherwise) $\mathfrak{M}$, if $\mathfrak{M} \models I_{\mathcal{L}}$, then there is a structure $\mathfrak{M}'$ which is identity-normal such that, for all φ , $\mathfrak{M} \models \varphi$ iff $\mathfrak{M}' \models \varphi$.

Proof: Let $\mathfrak{M}$ be a model of $I_{\mathcal{L}}$. By our more general soundness result, all φ such that $I_{\mathcal{L}} \vdash \varphi$ are true in $\mathfrak{M}$, so all φ such that $\vdash^= \varphi$ are true in $\mathfrak{M}$.

Let E be the relation $\mathfrak{M}$ assigns to $=^2$ (also known as $=^{\mathfrak{M}}$). E is not guaranteed to be the identity relation, but it must have these properties:

- E is reflexive, since $\mathfrak{M} \models \forall x x = x$.
- E is symmetric, since $\mathfrak{M} \models \forall x \forall y (x = y \rightarrow y = x)$.
- E is transitive, since $\mathfrak{M} \models \forall x \forall y \forall z (x = y \rightarrow (y = z \rightarrow x = z))$.
- Therefore, E is an equivalence relation.
- Moreover, if $\langle A_1, A_2 \rangle \in E$ for any two members of the domain A_1 and A_2 , then A_1 and A_2 are “indiscernible” according to $\mathfrak{M}$, that is, they are members of precisely the same extensions of precisely the same predicates in precisely the same positions, since, e.g.:

$$\mathfrak{M} \models \forall x \forall y (x = y \rightarrow (F^1 x \rightarrow F^1 y))$$

$$\mathfrak{M} \models \forall z \forall x \forall y (x = y \rightarrow (R^2 xz \rightarrow R^2 yz))$$

$$\mathfrak{M} \models \forall z \forall x \forall y (x = y \rightarrow (R^2 zx \rightarrow R^2 zy))$$

and so on.

Our goal is to construct an identity-normal structure which renders all and only the same sentences true. We do this by swapping out the elements of the original structure with their E -equivalence classes.

More precisely let $\mathfrak{M}'$ be the structure characterized as follows:

- $|\mathfrak{M}'|$ is set of E -equivalence classes formed on members of $|\mathfrak{M}|$, i.e., the set of all $[A]^E$ where $A \in |\mathfrak{M}|$.
- For each constant c , $c^{\mathfrak{M}'} = [c^{\mathfrak{M}}]^E$.
- For each function letter f^n , $f^{n\mathfrak{M}'}$ is the set of all pairs $\langle \langle [A_1]^E, \dots, [A_n]^E \rangle, [B]^E \rangle$ where $\langle \langle A_1, \dots, A_n \rangle, B \rangle \in f^{n\mathfrak{M}}$.

- For each predicate F^n , $F^{n\mathfrak{M}'}$ is the set of all n -tuples $\langle [A_1]^E, \dots, [A_n]^E \rangle$ where $\langle A_1, \dots, A_n \rangle \in F^{n\mathfrak{M}}$.

First, note that $\mathfrak{M}'$ is identity-normal. The predicate $=^2$ is assigned all ordered pairs $\langle [A_1]^E, [A_2]^E \rangle$ such that $\langle A_1, A_2 \rangle \in E$. Since E is an equivalence relation, this means that $[A_1]^E$ and $[A_2]^E$ must be the same classes (per your first homework).

We now must show that for all sentences $\mathfrak{M} \models \varphi$ iff $\mathfrak{M}' \models \varphi$. For each variable assignment s of $\mathfrak{M}$, there is a corresponding one $s^\#$ such that for any variable x , $\text{Val}_{s^\#}^{\mathfrak{M}'}(x) = [\text{Val}_s^{\mathfrak{M}}(x)]^E$. It follows inductively from our characterization of $\mathfrak{M}'$'s assignment to constants and function letters that $\text{Val}_{s^\#}^{\mathfrak{M}'}(t) = [\text{Val}_s^{\mathfrak{M}}(t)]^E$ for all terms t .

We will show by formula induction that it holds of every formula φ , whether open or closed, $\mathfrak{M}, s \models \varphi$ iff $\mathfrak{M}', s^\# \models \varphi$ for all such pairs s and $s^\#$.

Base step. Suppose φ is atomic. If φ is $\top$ then $\mathfrak{M}, s \models \varphi$ and $\mathfrak{M}', s^\# \models \varphi$. If φ is $\perp$, then neither of these is true. Otherwise, φ takes the form $F^n(t_1, \dots, t_n)$. If $\mathfrak{M}, s \models \varphi$, then $\langle \text{Val}_s^{\mathfrak{M}}(t_1), \dots, \text{Val}_s^{\mathfrak{M}}(t_n) \rangle \in F^{n\mathfrak{M}}$ and so $\langle \text{Val}_{s^\#}^{\mathfrak{M}'}(t_1), \dots, \text{Val}_{s^\#}^{\mathfrak{M}'}(t_n) \rangle \in F^{n\mathfrak{M}'}$ and so $\mathfrak{M}', s^\# \models \varphi$. If not, then not.

Induction step. Suppose for all formulas ψ with fewer connectives than φ , that for every pair of assignments s and $s^\#$ it holds that $\mathfrak{M}, s \models \psi$ iff $\mathfrak{M}', s^\# \models \psi$. We must show the same is true for φ . There are three kinds of cases to consider.

- (1) φ is a molecular statement, and takes one of the forms $\neg\psi$, $\psi \vee \theta$, $\psi \wedge \theta$, $\psi \rightarrow \theta$, or $\psi \leftrightarrow \theta$. By the inductive hypothesis, $\mathfrak{M}, s \models \psi$ iff $\mathfrak{M}', s^\# \models \psi$ and $\mathfrak{M}, s \models \theta$ iff $\mathfrak{M}', s^\# \models \theta$ for any $s, s^\#$. But the satisfaction or not of a molecular statement depends only on the satisfaction (or not) of its parts. Hence, $\mathfrak{M}, s \models \varphi$ iff $\mathfrak{M}', s^\# \models \varphi$, for any pair $s, s^\#$.
- (2) φ takes the form $\forall x\psi$. For some s , suppose $\mathfrak{M}, s \models \varphi$, which means that for every s' different from s at what is assigned to x is such that $\mathfrak{M}, s' \models \psi$. Every assignment $s'^\#$ different from $s^\#$ in $\mathfrak{M}'$ by what gets assigned to x is paired up with such an s' in $\mathfrak{M}$ and so by the inductive hypothesis, $\mathfrak{M}', s'^\# \models \psi$, and therefore $\mathfrak{M}', s^\# \models \forall x\psi$, i.e., $\mathfrak{M}', s^\# \models \varphi$. Suppose instead

that $\mathfrak{M}, s \not\models \varphi$, then for some s' , $\mathfrak{M}, s' \not\models \psi$ and for the corresponding $s'^\#$ we have $\mathfrak{M}', s'^\# \not\models \psi$ and thus $\mathfrak{M}', s^\# \not\models \varphi$. So $\mathfrak{M}, s \models \varphi$ iff $\mathfrak{M}', s^\# \models \varphi$.

- (3) φ takes the form $\exists x\psi$. For some s , suppose $\mathfrak{M}, s \models \varphi$, which means that for some s' different from s at what is assigned to x is such that $\mathfrak{M}, s' \models \psi$. s' is paired up with some $s'^\#$ in $\mathfrak{M}'$, and by the inductive hypothesis, $\mathfrak{M}', s'^\# \models \psi$, and therefore $\mathfrak{M}', s^\# \models \exists x\psi$, i.e., $\mathfrak{M}', s^\# \models \varphi$. Suppose instead that $\mathfrak{M}, s \not\models \varphi$, then for every s' differing from s by what it assigns to x is such that $\mathfrak{M}, s' \not\models \psi$ and, by the inductive hypothesis, for each $s'^\#$ differing from $s^\#$ in this way we have $\mathfrak{M}', s'^\# \not\models \psi$ and thus $\mathfrak{M}', s^\# \not\models \varphi$. So again $\mathfrak{M}, s \models \varphi$ iff $\mathfrak{M}', s^\# \models \varphi$.

Hence regardless of the complexity of φ , for each pair of assignments s and $s^\#$, $\mathfrak{M}, s \models \varphi$ iff $\mathfrak{M}', s^\# \models \varphi$. Such pairs exhaust the variable assignments in $\mathfrak{M}$ and $\mathfrak{M}'$, so $\mathfrak{M} \models \varphi$ iff $\mathfrak{M}' \models \varphi$. *

Result (Identity completeness): If $\Gamma \models \varphi$ then $\Gamma \vdash \varphi$.

The proof of identity completeness is left as an exam question. But as a hint, the proof is similar to the more generic completeness proof on p. 30, except with $\Gamma \cup I_L \cup \{\neg\varphi\}$ playing the role of $\Gamma \cup \{\neg\varphi\}$, and that at some point the Factoring lemma is needed.

Corollary: $\Gamma \vdash \varphi$ iff $\Gamma \models \varphi$

Proof: Immediate from identity soundness and identity completeness. *

From here on out, we are mainly interested in logical systems built on identity logic, and so we will begin using $\vdash$ to mean $\vdash^=$ and $\models$ to mean $\models^=$, unless otherwise noted.

Unit 2

Metamathematics and the Gödel Results

1. Theories in General

In this unit, we take logical systems for mathematics as our object languages.

Definition: A set of sentences Γ is **closed** (or more specifically “closed under entailment”) iff for all sentences φ in a given first-order language, if $\Gamma \vdash \varphi$, then $\varphi \in \Gamma$.

Definition: A **theory** is a closed set of sentences.

Definition: The **closure** of a set of sentences Γ is the set of all sentences that can be derived from Γ , i.e., $\{\varphi : \Gamma \vdash \varphi\}$.

A closed set or theory is one containing all its logical consequences. Any set that isn’t a theory has a closure which is a theory.

There are two ways that theories are typically specified: either by providing a specification of a set of “proper” or “non-logical” axioms of which the theory is the “closure”, or by providing a structure where the theory is everything true in that structure. One thing we might be interested in is connecting the two: can the same theory be equivalently specified both semantically and deductively?

By convention, we shall start using boldface abbreviations as constant names of theories, especially where the same abbreviation, non-bold face, is a set of sentences of which it is the closure. For example, with might use **A** as the name of the closure of some set we’re calling *A*. (We’ll continue to use uppercase Greek letters such as Γ and Δ as variables in the metalanguage for sets of sentences, including theories, but also **T** as a variable for an arbitrary theory.)

Definition: A theory Γ is **axiomatized** by a set Δ iff Γ is the closure of Δ .

Definition: A theory Γ is an **extension** of Δ iff $\Delta \subseteq \Gamma$.

Note that because sets have extensional identity conditions, the same theory might be axiomatized by different sets of axioms, if those axioms allow you to derive all and only the same results.

Result: If Γ is the closure of Δ (or axiomatized by Δ), then for all φ , $\Gamma \vdash \varphi$ iff $\Delta \vdash \varphi$.

HOMEWORK 16

Prove the result above.

This result allows us to talk more or less interchangeably about a set of axioms and the “theory” which is its closure.

2. Theories for Arithmetic

We are concerned here only with theories where the intended language and semantics deal only with natural numbers 0, 1, 2, 3, ..., their properties and relations. The set of natural numbers is (in mathematics) often called $\mathbb{N}$.

The language of arithmetic

Definition: The **language of arithmetic**, $\mathcal{L}_A$, is the first-order language making use of the following vocabulary:

- (1) *There is only one constant, a , but instead of “ a ”, by convention we write “ 0 ”.*
- (2) *There are two predicate, both dyadic, A^2 and $=^2$, but by convention, rather than writing “ $A^2(t_1, t_2)$ ” and “ $=^2(t_1, t_2)$ ” we write $t_1 < t_2$ and $t_1 = t_2$.*
- (3) *There are three function letters, one monadic and two dyadic, f^1 , f^2 and f_1^2 , but by convention, rather than writing $f^1(t)$, $f^2(t_1, t_2)$ and $f_1^2(t_1, t_2)$, we write $(t)'$, $(t_1 + t_2)$, and $(t_1 \times t_2)$, and omit the parentheses when there is no ambiguity.*

We shall for the moment consider logical systems using this more austere language; one in which there are no other predicates, constants or function signs (at least taken as primitive). We still use the full range of variables. “Dummy” constants are also allowed in UG and EI proofs, though not axioms.

Semantics for $\mathcal{L}_A$

These vocabulary items in $\mathcal{L}_A$ have obvious intended interpretations.

Definition: *The **standard interpretation** or **standard model** of arithmetic, $\mathfrak{N}$, is the structure defined as follows.*

- (1) *The domain of quantification $|\mathfrak{N}|$ for $\mathfrak{N}$ is $\mathbb{N}$.*
- (2) *The constant “ 0 ” is interpreted as standing for zero, i.e., $0^{\mathfrak{N}} = \text{zero}$.*
- (3) *The predicate “ $=^2$ ” is interpreted as standing for identity, i.e., $=^{\mathfrak{N}}$ is $\{\langle n, n \rangle : n \in \mathbb{N}\}$. (This makes $\mathfrak{N}$ identity-normal.)*
- (4) *The predicate “ $<$ ” is interpreted as standing for the less-than relation on $\mathbb{N}$, i.e., $<^{\mathfrak{N}}$ is $\{\langle m, n \rangle : m \in \mathbb{N}, n \in \mathbb{N} \text{ and } m \text{ is less than } n\}$.*
- (5) *The functor “ $'$ ” is interpreted as standing for the successor function on $\mathbb{N}$, i.e., $'^{\mathfrak{N}}$ is $\{\langle m, n \rangle : m \in \mathbb{N}, n \in \mathbb{N} \text{ and } n \text{ is one more than } m\}$.*
- (6) *The functor “ $+$ ” is interpreted as standing for the addition operation on $\mathbb{N}$, i.e., $+^{\mathfrak{N}}$ is $\{\langle \langle m, n \rangle, s \rangle : m, n, s \in \mathbb{N} \text{ and } s \text{ is the sum of } m \text{ and } n\}$.*
- (7) *The functor “ $\times$ ” is interpreted as standing for multiplication, i.e., $\times^{\mathfrak{N}}$ is $\{\langle \langle m, n \rangle, s \rangle : m, n, s \in \mathbb{N} \text{ and } s \text{ is the product of } m \text{ and } n\}$.*

Definition: Let **true arithmetic** or **TA** be the set of sentences that are true in $\mathfrak{N}$, i.e., $\{\varphi : \mathfrak{N} \models \varphi\}$.

Result: **TA** is a theory.

Proof: (1) Let φ be an arbitrary sentence such that $\mathbf{TA} \vdash \varphi$.
 (2) By the soundness of first-order logic with identity, $\mathbf{TA} \models \varphi$. This means that φ is true in every model of **TA**.
 (3) By the definition of **TA**, $\mathfrak{N}$ makes every member of **TA** true, and so $\mathfrak{N}$ is a model of **TA**.
 (4) Hence, φ is true in $\mathfrak{N}$.
 (5) Thus, $\varphi \in \mathbf{TA}$ by the definition of **TA**.
 (6) Since φ was arbitrary, this shows that **TA** is closed under entailment, and therefore, is a theory. *

The question then arises if there is a different way of specifying **TA**, one that appeals to a limited set of axioms. Obviously, **TA** is trivially axiomatized by itself, but this is not very useful. If the axioms one is allowed to use are “everything true”, proofs using this set of axioms do not serve very well as an independent means for establishing what is true and what isn’t. Is there a finite, or at least easily and independently determinable, list of axioms from which all other truths of arithmetic follow, i.e., of which **TA** is the closure?

Peano and Robinson arithmetics

At the end of the 19th and beginning of the 20th century, prior to Gödel’s results, it was widely thought that all of arithmetic could be derived from the following assumptions:

Definition: *Stated in English, the **Peano-Dedekind axioms** are the following principles:*

- (P1) *Zero is a natural number.*
- (P2) *Every natural number has a successor which is also a natural number.*
- (P3) *Zero is not the successor of any natural number.*
- (P4) *No two natural numbers have the same successor.*
- (P5) *If something is both true of zero, and true of the successor of a number whenever it is true of that*

number, then it is true of all natural numbers (i.e., the principle of mathematical induction).

These were first considered in the context in which it was presupposed that one could make use of set-theoretic definitions of things like multiplication, addition, less-than, etc. Dropping this assumption, and refactoring for modern symbolic logic, we can generate the following set of axioms more or less doing the work of (P1)–(P4):

$$\forall x \forall y (x' = y' \rightarrow x = y) \quad (Q_1)$$

$$\forall x 0 \neq x' \quad (Q_2)$$

$$\forall x (x \neq 0 \rightarrow \exists y x = y') \quad (Q_3)$$

$$\forall x x + 0 = x \quad (Q_4)$$

$$\forall x \forall y x + y' = (x + y)' \quad (Q_5)$$

$$\forall x x \times 0 = 0 \quad (Q_6)$$

$$\forall x \forall y x \times y' = (x \times y) + x \quad (Q_7)$$

$$\forall x \forall y (x < y \leftrightarrow \exists z z' + x = y) \quad (Q_8)$$

We might gloss these in English this way:

- Q₁. No two numbers have the same successor (P4).
- Q₂. Zero is not the successor of any number (P3).
- Q₃. Every number other than zero is the successor of some number. (This partly captures our intention only to quantify over natural numbers.)
- Q₄. Any number plus zero is itself.
- Q₅. Any number plus the successor of another number is the successor of the sum of the first number and the second. (Together, this and the previous axiom provide an implicit definition of multiplication, which is needed when taking it as primitive.)
- Q₆. Any number multiplied by zero is zero.
- Q₇. The product of one number and the successor of another is equal to the sum of their product and the first. (Again, this and the previous one implicitly define multiplication.)
- Q₈. One natural number is less than another if and only if there is a non-zero number which sums with the first to yield the second. (This basically provides a complete definition of less-than for natural numbers; indeed, one might have taken $t_1 < t_2$ to simply abbreviate $\exists z z' + t_1 = t_2$ instead.

The work of (P1) is basically captured by having “0” as a constant assigned to a member of the domain of the quantification, and needn’t be expressed by a separate axiom. Similarly, the work of (P2) is done more or less just by having “’” in the language.

Definition: Let the set Q be the set containing Q_1 through Q_8 .

Definition: *Robinson arithmetic*, or Q , is the theory which is the closure of Q .

Robinson arithmetic does not have any way to capture mathematical induction as an object-language notion, as thus is weaker than the system intended by Peano and Dedekind. It grew to be something that was studied only later, as an object of comparison with Peano arithmetic. As we shall see as we proceed, however, is it is already very powerful and interesting on its own.

To capture (P5), we can consider this schema—(PA₉)—for mathematical induction, where $y_1, \dots, y_n$ are the free variables of $\varphi(x)$ apart from x :

$$\forall y_1 \dots \forall y_n ([\varphi(0) \wedge \forall x (\varphi(x) \rightarrow \varphi(x'))] \rightarrow \forall x \varphi(x))$$

Definition: Let PA be the set containing Q_1 through Q_8 and every instance of the induction schema PA_9 .

Definition: (First-order) *Peano arithmetic*, or PA , is the theory which is the closure of set PA .

Technically, the set PA of axioms for PA has infinitely many members, as PA_9 is not a single axiom, but a schema which has infinitely many instances. In a *higher-order* logic, the schematic letter φ might be replaced by an object-language variable, and PA might be axiomatized in a finite way.

Since Q is a subset of PA , clearly Q is a subset of PA , and so PA is an extension of Q .

Result: Theories Q , PA and TA are all consistent.

Proof: The members of all three theories are true in the standard interpretation of arithmetic. For **TA** this is true by definition. For **Q** and **PA**, note that the members of **Q** and **PA** are all true, and by soundness, so are their entailments. Hence, these theories are satisfiable, and therefore consistent.*

Most proofs in **PA** of universally quantified sentences make use of MI.

HOMEWORK 17

Prove: $\mathbf{PA} \vdash \forall x \forall y (x \times y = 0 \rightarrow x = 0 \vee y = 0)$. You can use (0+) above.

$$\mathbf{PA} \vdash \forall x \forall y (x + y = 0 \rightarrow x = 0 \wedge y = 0) \quad (0+)$$

1. $b + 0 = 0$ Hyp
2. $b + 0 = b$ Q_4, UI
3. $b = 0$ 1, 2 LL
4. $0 = 0$ Ref=
5. $b = 0 \wedge 0 = 0$ 3, 4 Conj
6. $b + 0 = 0 \rightarrow b = 0 \wedge 0 = 0$ 1–5 CP/DT
7. $b + c = 0 \rightarrow b = 0 \wedge c = 0$ Hyp
8. $b + c' = 0$ Hyp
9. $b + c' = (b + c)'$ Q_5, UI
10. $(b + c)' = 0$ 8, 9 LL
11. $(b + c)' \neq 0$ $Q_2, \text{Sym}=T, UI, MT$
12. $\perp$ 10, 11 Contra
13. $b = 0 \wedge c' = 0$ 12 Explode
14. $b + c' = 0 \rightarrow b = 0 \wedge c' = 0$ 8–13 CP/DT
15. $(b + c = 0 \rightarrow b = 0 \wedge c = 0) \rightarrow$
 $(b + c' = 0 \rightarrow b = 0 \wedge c' = 0)$ 7–14 CP/DT
16. $\forall y [(b + y = 0 \rightarrow b = 0 \wedge y = 0) \rightarrow$
 $(b + y' = 0 \rightarrow b = 0 \wedge y' = 0)]$ 15 UG
17. $\forall x [(x + 0 = 0 \rightarrow x = 0 \wedge 0 = 0) \wedge$
 $\forall y [(x + y = 0 \rightarrow x = 0 \wedge y = 0) \rightarrow$
 $(x + y' = 0 \rightarrow x = 0 \wedge y' = 0)]] \rightarrow$
 $\forall y (x + y = 0 \rightarrow x = 0 \wedge y = 0)$ PA_9
18. $([b + 0 = 0 \rightarrow b = 0 \wedge 0 = 0] \wedge$
 $\forall y [(b + y = 0 \rightarrow b = 0 \wedge y = 0) \rightarrow$
 $(b + y' = 0 \rightarrow b = 0 \wedge y' = 0)]) \rightarrow$
 $\forall y (b + y = 0 \rightarrow b = 0 \wedge y = 0)$ 17 UI
19. $\forall y (b + y = 0 \rightarrow b = 0 \wedge y = 0)$ 6, 16, 18 MP $\times 2$
20. $\forall x \forall y (x + y = 0 \rightarrow x = 0 \wedge y = 0)$ 19 UG

So formulated, we can think of lines 1–6 as the base step and lines 7–17 as the induction step, where line 7 is the inductive hypothesis. Actually, this example is a bit unusual in that line 7 is never actually used, and so we might have shortened things by getting line 14 with FA and line 15 by TC, but the structure of the induction would be less clear then.

We might as well make this a derived rule:

$$\mathbf{PA}, \varphi(0), \forall x (\varphi(x) \rightarrow \varphi(x')) \vdash \forall x \varphi(x) \quad (MI)$$

$$\mathbf{PA} \vdash \forall x 0 + x = x \quad (+0)$$

1. $0 + 0 = 0$ Q_4, UI
2. $0 + b = b$ Hyp
3. $0 + b' = (0 + b)'$ Q_5, UI
4. $0 + b' = b'$ 2, 3 LL
5. $0 + b = b \rightarrow 0 + b' = b'$ 2–4 DT
6. $\forall x (0 + x = x \rightarrow 0 + x' = x')$ 4 UG
7. $\forall x 0 + x = x$ 1, 6 MI

$$\mathbf{PA} \vdash \forall x \forall y x' + y = (x + y)' \quad (\text{Alt}Q_5)$$

1. $b' + 0 = b'$ Q_4, UI
2. $b + 0 = b$ Q_4, UI
3. $b' = b'$ Ref=
4. $(b + 0)' = b'$ 2, 3 LL
5. $b' + 0 = (b + 0)'$ 1, 4 LL
6. $b' + c = (b + c)'$ Hyp
7. $b' + c' = (b' + c)'$ Q_5, UI
8. $b' + c' = (b + c)''$ 6, 7 LL
9. $b + c' = (b + c)'$ Q_5, UI
10. $b' + c' = (b + c')'$ 8, 9 LL
11. $b' + c = (b + c)' \rightarrow b' + c' = (b + c')'$ 6–10 DT
12. $\forall y [b' + y = (b + y)' \rightarrow b' + y' = (b + y')']$ 11 UG
13. $\forall y b' + y = (b + y)'$ 5, 12 MI
14. $\forall x \forall y x' + y = (x + y)'$ 13 UG

$$\mathbf{PA} \vdash \forall x \forall y x + y = y + x \quad (\text{Com}+)$$

1. $b + 0 = b$ Q_4, UI
2. $0 + b = b$ +0, UI
3. $b + 0 = 0 + b$ 1, 2 LL
4. $b + c = c + b$ Hyp
5. $b + c' = (b + c)'$ Q_5, UI
6. $b + c' = (c + b)'$ 3, 4 LL
7. $c' + b = (c + b)'$ Alt Q_5, UI
8. $b + c' = c' + b$ 6, 7 LL
9. $b + c = c + b \rightarrow b + c' = c' + b$ 4–8 DT
10. $\forall y (b + y = y + b \rightarrow b + y' = y' + b)$ 9 UG
11. $\forall y b + y = y + b$ 3, 10 MI
12. $\forall x \forall y x + y = y + x$ 10 UG

3. A Naïve Attempt to Go More Primitive: System F

Do we really need to take all of $0, ', +, \times, <$ as primitive? Cardinal numbers would seem to be what sets or classes have in common when they are the “same size”: can we define a number as the class of all classes of a certain size, perhaps, in set theory. Here is a tempting try.

Set theory can be formulated using only the predicate $\in$, but it is often more convenient to introduce the notation $\{x : \varphi(x)\}$ as well:

Syntax

1. We add to the syntax of predicate logic the following subnecutive, which yields a term for any variable x and wff $\varphi(x)$.

$$\{x : \varphi(x)\}$$

All occurrences of x here are bound.

2. We also choose a two-place predicate letter E^2 to use for the membership relation. An expression of the form $(t \in u)$ is shorthand for $E^2(t, u)$, and $(t \notin u)$ is shorthand for $\neg E^2(t, u)$; we also use the identity predicate.

Axioms

System F is the theory which is the closure of the set containing all specified instances of the following schemata, (NC) and (Ext) (and their universal closures, if φ and ψ contain additional free variables):

$$\forall x(\varphi(x) \leftrightarrow x \in \{y : \varphi(y)\}), \quad (\text{NC})$$

for all cases in which the variable y is free for x in $\varphi(x)$.

$$\forall x(x \in \{y : \varphi(y)\} \leftrightarrow x \in \{z : \psi(z)\}) \rightarrow \{y : \varphi(y)\} = \{z : \psi(z)\} \quad (\text{Ext})$$

where $\{y : \varphi(y)\}$ and $\{z : \psi(z)\}$ do not contain x free.

Some intuitive definitions

(Let x, y and z below be the first variables not occurring in t and u .)

SET THEORETIC DEFINITIONS

$$(t \cap u) \text{ for } \{x : x \in t \wedge x \in u\}$$

$$(t \cup u) \text{ for } \{x : x \in t \vee x \in u\}$$

$$(t \setminus u) \text{ for } \{x : x \in t \wedge x \notin u\}$$

$$\bar{t} \text{ for } \{x : x \notin t\}$$

$$(t \subseteq u) \text{ for } \forall x(x \in t \rightarrow x \in u)$$

$$\forall \text{ for } \{x : x = x\}$$

$$\emptyset \text{ for } \{x : x \neq x\}$$

$$\{t\} \text{ for } \{x : x = t\}$$

$$\{t, u\} \text{ for } \{x : x = t \vee x = u\}$$

$$\langle t, u \rangle \text{ for } \{\{t\}, \{t, u\}\}$$

$$(t \times u) \text{ for } \{x : \exists y \exists z(x = \langle y, z \rangle \wedge y \in t \wedge z \in u)\}$$

$$\text{Dom}(t) \text{ for } \{x : \exists y \langle x, y \rangle \in t\}$$

$$\text{Rng}(t) \text{ for } \{x : \exists y \langle y, x \rangle \in t\}$$

$$\text{Fld}(t) \text{ for } (\text{Dom}(t) \cup \text{Rng}(t))$$

$$\text{Inv}(t) \text{ for } \{x : \exists y \exists z(x = \langle y, z \rangle \wedge \langle z, y \rangle \in t)\}$$

$$\text{Fnct}(t) \text{ for } \forall x \forall y \forall z(\langle x, y \rangle \in t \wedge \langle x, z \rangle \in t \rightarrow y = z)$$

$$\text{One-one}(t) \text{ for } (\text{Fnct}(t) \wedge \text{Fnct}(\text{Inv}(t)))$$

MATHEMATICAL DEFINITIONS

$$(t \approx u) \text{ for } \exists x(\text{One-one}(x) \wedge \text{Dom}(x) = t \wedge \text{Rng}(x) = u)$$

$$\text{Card}(t) \text{ for } \{x : x \approx t\}$$

$$0 \text{ for } \text{Card}(\emptyset)$$

$$t' \text{ for } \{x : \exists y(y \in x \wedge x \setminus \{y\} \in t)\}$$

$$1 \text{ for } 0'$$

$$2 \text{ for } 1'$$

$$3 \text{ for } 2' \quad \dots \text{ and so on for other numerals}$$

$$\mathbb{N} \text{ for } \{x : \forall y(0 \in y \wedge \forall z(z \in y \rightarrow z' \in y) \rightarrow x \in y)\}$$

$$\text{Fin}(t) \text{ for } \exists x(x \in \mathbb{N} \wedge t \in x)$$

$$\text{Infin}(t) \text{ for } \neg \text{Fin}(t)$$

$$\text{Denum}(t) \text{ for } (t \approx \mathbb{N})$$

$$\text{Ctbl}(t) \text{ for } (\text{Fin}(t) \vee \text{Denum}(t))$$

$$(t \leq u) \text{ for } \exists x \exists y \exists z(x \in t \wedge y \in u \wedge z \subseteq y \wedge x \approx z)$$

$$(t < u) \text{ for } ((t \leq u) \wedge \neg(u \leq t))$$

$$\S(t) \text{ for } \{x : x \in \mathbb{N} \wedge x < t\}$$

$$(t + u) \text{ for } \text{Card}((\S(t) \times \{0\}) \cup (\S(u) \times \{1\}))$$

$$(t \cdot u) \text{ for } \text{Card}(\S(t) \times \S(u))$$

Results

With these definitions in place, one can derive the Peano-Dedekind axioms as theorems in the following forms:

- (P1) $\mathbf{F} \vdash 0 \in \mathbb{N}$
(P2) $\mathbf{F} \vdash \forall x(x \in \mathbb{N} \rightarrow x' \in \mathbb{N})$
(P3) $\mathbf{F} \vdash \forall x(x \in \mathbb{N} \rightarrow 0 \neq x')$
(P4) $\mathbf{F} \vdash \forall x(x \in \mathbb{N} \wedge y \in \mathbb{N} \rightarrow (x' = y' \rightarrow x = y))$
(P5) $\mathbf{F} \vdash \varphi[0] \wedge \forall x(\varphi(x) \rightarrow \varphi(x')) \rightarrow \forall x(x \in \mathbb{N} \rightarrow \varphi(x))$

As well as analogues of **PA**'s other axioms:

- $\mathbf{F} \vdash \forall x(x \in \mathbb{N} \rightarrow (x \neq 0 \rightarrow \exists y x = y'))$
 $\mathbf{F} \vdash \forall x(x \in \mathbb{N} \rightarrow x + 0 = x)$
 $\mathbf{F} \vdash \forall x \forall y(x \in \mathbb{N} \wedge y \in \mathbb{N} \rightarrow x + y' = (x + y)')$
 $\mathbf{F} \vdash \forall x x \cdot 0 = 0$
 $\mathbf{F} \vdash \forall x \forall y(x \in \mathbb{N} \wedge y \in \mathbb{N} \rightarrow (x \cdot y') = ((x \cdot y) + x))$
 $\mathbf{F} \vdash \forall x \forall y(x \in \mathbb{N} \wedge y \in \mathbb{N} \rightarrow (x < y \leftrightarrow \exists z(z \in \mathbb{N} \wedge x' + z = y)))$

Also, we have, e.g.:

- $\mathbf{F} \vdash (\exists_1 x \varphi(x)) \leftrightarrow (\{x : \varphi(x)\} \in 1)$
 $\mathbf{F} \vdash (\exists_2 x \varphi(x)) \leftrightarrow (\{x : \varphi(x)\} \in 2)$
 $\mathbf{F} \vdash (\exists_3 x \varphi(x)) \leftrightarrow (\{x : \varphi(x)\} \in 3)$

And so on.

Disaster

The system **F**, unfortunately, is inconsistent due to Russell's paradox:

$$\mathbf{F} \vdash \{y : y \notin y\} \notin \{y : y \notin y\} \leftrightarrow \{y : y \notin y\} \in \{y : y \notin y\}$$

Proof: Direct from (NC) and universal instantiation. Whence both $\mathbf{F} \vdash \{y : y \notin y\} \in \{y : y \notin y\}$, and $\mathbf{F} \vdash \{y : y \notin y\} \notin \{y : y \notin y\}$.

Hence $\mathbf{F} \vdash \varphi$ for all wffs φ , making the system entirely unsuitable for mathematics. We have both $\mathbf{F} \vdash 1 + 1 = 2$ and $\mathbf{F} \vdash 1 + 1 = 3$!

There have been many modifications in various set theories and higher-order logics which do generate consistent theories that can "interpret" Peano arithmetic, but none have the same intuitive appeal. When I teach Mathematical Logic II, this is a main theme.

HOMEWORK 18

Without using Russell's paradox or other contradiction, prove $\mathbf{F} \vdash \forall x \forall y (\{x\} = \{y\} \rightarrow x = y)$.

Let us return to our consideration of **Q** and **PA** as presented in the Open Logic Text, however.

4. Numerals and Q versus PA

Numerals are the canonical signs for numbers in a given language or communication system. In the decimal system used in mathematics and everyday life, a numeral would be a series of digits like "112" or "504", without function signs, etc.

In language $\mathcal{L}_A$ we can give this definition instead:

Definition: In $\mathcal{L}_A$, the **numeral** for number n is the constant "0" followed by n successor function signs.

The numeral for zero (0) is "0".

The numeral for one (1) is "0'".

The numeral for two (2) is "0''".

The numeral for three (3) is "0'''", and so on.

Convention: Let $\bar{n}$ be an abbreviation for the numeral for number n in language $\mathcal{L}_A$.

This notation $\bar{n}$ can be a bit confusing: what occurs under the bar is a *metalinguage* expression for a number; we can use any mathematical notation of regular mathematics there. But what the whole notation $\bar{n}$ including the overbar represents is always *object-language* numeral of the form "0''...(n times)...".

(And obviously, this is not meant as the set-complement notation we used in **F**.)

Generally, expected results involving numerals only (and not quantified variables) are obtainable even in the weaker system **Q**, e.g.:

$$\mathbf{Q} \vdash \bar{1} + \bar{1} = \bar{2}$$

- | | |
|----------------------------------|------------------|
| 1. $0' + 0' = (0' + 0)'$ | Q_4 UI |
| 2. $0' + 0 = 0'$ | Q_3 UI |
| 3. $0' + 0' = 0''$ | 1, 2 LL |
| 4. $\bar{1} + \bar{1} = \bar{2}$ | 3 def. $\bar{n}$ |

Indeed, we can be more general:

Result (Num+): For any natural numbers n and m , $\mathbf{Q} \vdash \bar{n} + \bar{m} = \overline{n+m}$.

Proof: Notice the result does not follow by Ref= or anything similar. On the left side of the above, we have the object-language “+” sign; on the right, we have the one from the metalanguage, and the resulting numeral is the numeral for the sum.

Although $\mathbf{Q}$ does not have object-language induction principles, we can still do induction in the metalanguage on the numbers used in $\mathbf{Q}$ ’s numerals. We will prove this by induction on m .

- (1) Let n be an arbitrary natural number.
- (2) When m is zero, the numeral for $n + m$ ($n + 0$) is just the numeral for n , and we have $\mathbf{Q} \vdash \bar{n} + 0 = \overline{n+0}$ by $\mathbf{Q}_3$.
- (3) As inductive hypothesis suppose for a given m that we have $\mathbf{Q} \vdash \bar{n} + \bar{m} = \overline{n+m}$. We must now show that $\mathbf{Q} \vdash \bar{n} + \overline{m+1} = \overline{n+(m+1)}$.
- (4) By $\mathbf{Q}_5$ and UI, $\mathbf{Q} \vdash \bar{n} + \bar{m}' = (\bar{n} + \bar{m})'$.
- (5) The numeral for $m + 1$ is 0 followed by $m + 1$ many successor signs, which is the same as $\bar{m}'$, and so the above result is also $\mathbf{Q} \vdash \bar{n} + \overline{m+1} = (\bar{n} + \bar{m})'$.
- (6) By the inductive hypothesis and LL, $\mathbf{Q} \vdash \bar{n} + \overline{m+1} = \overline{n+m'}$.
- (7) By the mathematical fact of associativity of addition we accept in the metalanguage, $n+(m+1)$ is $(n+m)+1$, the numeral for which is $\overline{n+m'}$.
- (8) Hence our earlier result is the same as $\mathbf{Q} \vdash \bar{n} + \overline{m+1} = \overline{n+(m+1)}$.
- (9) Therefore, by mathematical induction, $\mathbf{Q} \vdash \bar{n} + \bar{m} = \overline{n+m}$ is true for any m , and also for any n , since n was arbitrary. *

Result (NumCom+): For any natural numbers n and m , $\mathbf{Q} \vdash \bar{n} + \bar{m} = \overline{m+n}$.

Proof: (1) By Num+ above, we have both:

$$\mathbf{Q} \vdash \bar{n} + \bar{m} = \overline{n+m}$$

$$\mathbf{Q} \vdash \bar{m} + \bar{n} = \overline{m+n}$$

- (2) By the fact of commutativity of addition we accept in the metalanguage, the numeral for $n + m$ is the same numeral as that for $m + n$, and so by symmetry and transitivity of identity, $\mathbf{Q} \vdash \bar{n} + \bar{m} = \overline{m+n}$. *

Of course, all theorems of $\mathbf{Q}$ are also theorems of $\mathbf{PA}$, and so the results above also hold for $\mathbf{PA}$.

It is interesting to contrast NumCom+ with our earlier result of Com+ for $\mathbf{PA}$. NumCom+ holds for both $\mathbf{Q}$ and $\mathbf{PA}$, but Com+ is not a theorem of $\mathbf{Q}$.

$$\mathbf{PA} \vdash \forall x \forall y \ x + y = y + x \quad (\text{Com+})$$

The difference of course is that one uses object-language variables, and one uses metalanguage variable for object-language numerals, which are not variables but closed terms.

This difference is typical of the difference between $\mathbf{PA}$ and $\mathbf{Q}$. In $\mathbf{Q}$, one can typically prove all the *instances* of the universally quantified theorems of $\mathbf{PA}$ obtained by induction by doing a “forced march” through all the instances leading up to a given instance. But one cannot obtain that general result itself. Quite often, however, getting all the instances is enough to be useful.

There are a few universal statements that can be proven in $\mathbf{Q}$ alone, but only those that do not require induction in the object language, e.g.:

$$\mathbf{Q} \vdash \forall x \neg x < 0 \quad (\neq 0)$$

- | | |
|-----------------------------|-----------------------------|
| 1. $b < 0$ | Hyp |
| 2. $\exists z \ z' + b = 0$ | 1, $\mathbf{Q}_8$ UI, BMP |
| 3. $c' + b = 0$ | 2 EI |
| 4. $b = 0$ | Hyp |
| 5. $c' + 0 = 0$ | 3, 4 LL |
| 6. $c' + 0 = c'$ | $\mathbf{Q}_4$, UI |
| 7. $0 = c'$ | 5, 6 LL |
| 8. $0 \neq c'$ | $\mathbf{Q}_2$, UI |
| 9. $\perp$ | 7, 8 Contra |
| 10. $b \neq 0$ | 4–9 RAA |
| 11. $\exists y \ b = y'$ | 10, $\mathbf{Q}_3$, UI, MP |

- | | | |
|----------------------------|---------------------|---|
| 12. $b = d'$ | 11 EI | (5) But $\mathbf{Q} \vdash 0 \neq 0'$...however many.../ by $\mathbf{Q}_2$ and UI. |
| 13. $c' + d' = 0$ | 3, 12 LL | (6) Hence we get $\perp$, and so by RAA, $\mathbf{Q} \vdash \bar{n} \neq \bar{m}$. |
| 14. $c' + d' = (c' + d)'$ | $\mathbf{Q}_5$, UI | For part (b): |
| 15. $0 = (c' + d)'$ | 13, 14 LL | (1) Suppose that $\mathbf{Q} \vdash \bar{n} = \bar{m}$ but for reductio suppose that n and m are different. |
| 16. $0 \neq (c' + d)'$ | $\mathbf{Q}_2$, UI | (2) By part (a), $\mathbf{Q} \vdash \bar{n} \neq \bar{m}$, and so $\mathbf{Q} \vdash \perp$. |
| 17. $\perp$ | 15, 16 Contra | (3) This impossible, since $\mathbf{Q}$ is consistent. |
| 18. $\neg b < 0$ | 1–17 RAA | (4) Therefore it must be n and m are the same. * |
| 19. $\forall x \neg x < 0$ | 8 UG | |

Result (Num $\times$): For any natural numbers n and m , $\mathbf{Q} \vdash \bar{n} \times \bar{m} = \overline{n \times m}$.

The above also holds for **PA**, with the same proof.

Result (NumAlt $\mathbf{Q}_5$): For every natural number n , $\mathbf{Q} \vdash \forall x x' + \bar{n} = (x + \bar{n})'$.

HOMEWORK 19

- (a) Prove Num $\times$. You may use induction in the metalanguage (only) as well as appeal to Num $+$.
- (b) Using induction (again, in the metalanguage) on the number of function signs they contain, prove that for all ground terms t of the language of arithmetic, $\mathcal{L}_A$, there is a numeral $\bar{n}$ such that $\mathbf{Q} \vdash t = \bar{n}$.

Result (Num=): For any natural numbers n and m , (a) if n is different from m , then $\mathbf{Q} \vdash \bar{n} \neq \bar{m}$; and (b) if $\mathbf{Q} \vdash \bar{n} = \bar{m}$ then n and m are in fact the same number.

Proof: For part (a):

- (1) Suppose that n and m are different; then one is larger than the other, and $\bar{n}$ and $\bar{m}$ contain a different number of “’” signs.
- (2) In the object language, assume for RAA that $\bar{n} = \bar{m}$.
- (3) The above means that we have $0' \dots n \text{ times} \dots' = 0' \dots m \text{ times} \dots'$, where one side has more “’”s than the other.
- (4) By either n or m instances of $\mathbf{Q}_1$ and MP, depending on which is smaller, we get either $0' \dots (n-m \text{ times}) \dots' = 0$ or $\vdash 0 = 0' \dots (m-n \text{ times}) \dots'$.

Proof: We prove this by induction on n in the metalanguage. For the base step, we show that $\mathbf{Q} \vdash \forall x x' + 0 = (x + 0)'$:

- | | |
|----------------------------------|---------------------|
| 1. $b + 0 = b$ | $\mathbf{Q}_4$, UI |
| 2. $b' + 0 = b'$ | $\mathbf{Q}_4$, UI |
| 3. $b' = b'$ | Ref= |
| 4. $(b + 0)' = b'$ | 1, 3 LL |
| 5. $b' + 0 = (b + 0)'$ | 2, 4 LL |
| 6. $\forall x x' + 0 = (x + 0)'$ | 5 UG |

For the induction step, we can assume that $\mathbf{Q} \vdash \forall x x' + \bar{n} = (x + \bar{n})'$, and must show that $\mathbf{Q} \vdash \forall x x' + \overline{n+1} = (x + \overline{n+1})'$:

- | | |
|---|-----------------------|
| 1. $\forall x x' + \bar{n} = (x + \bar{n})'$ | Inductive hypothesis |
| 2. $b' + \bar{n} = (b + \bar{n})'$ | 1 UI |
| 3. $b + \bar{n}' = (b + \bar{n})'$ | $\mathbf{Q}_5$, UI |
| 4. $b' + \bar{n} = b + \bar{n}'$ | 2, 3 LL |
| 5. $(b' + \bar{n})' = (b' + \bar{n})'$ | Ref= |
| 6. $(b' + \bar{n})' = (b + \bar{n}')'$ | 4, 5 LL |
| 7. $b' + \bar{n}' = (b' + \bar{n})'$ | $\mathbf{Q}_5$, UI |
| 8. $b' + \bar{n}' = (b + \bar{n}')'$ | 6, 7 LL |
| 9. $b' + \overline{n+1} = (b + \overline{n+1})'$ | Def. $\overline{n+1}$ |
| 10. $\forall x x' + \overline{n+1} = (x + \overline{n+1})'$ | 9 UG |

Result (Spread): For every natural number n , $\mathbf{Q} \vdash \forall x (x < \bar{n}' \leftrightarrow x = 0 \vee \dots \vee x = \bar{n})$.

	19. $d' = 0'$	18, Ref=, LL
Hyp	20. $b = 0'$	10, 19 LL
1, Q ₈ , UI, BMP	21. $b = 0 \vee b = 0' \vee \dots \vee b = \bar{n}'$	20 Add
2 EI	22. $d = 0 \rightarrow b = 0 \vee \dots \vee b = \bar{n}'$	18–21 CP
Hyp	23. $d = 0'$	Hyp
Q ₃ , UI, MP	24. $d' = 0''$	18, Ref=, LL
4 EI	25. $b = 0''$	10, 24 LL
3, 6 LL	26. $b = 0 \vee b = 0' \vee b = 0'' \vee \dots \vee b = \bar{n}'$	25 Add
Q ₅ , UI	27. $d = 0' \rightarrow b = 0 \vee \dots \vee b = \bar{n}'$	23–26 CP
7, 8 LL	:	
9, Q ₁ , UI, MP	28. $d = \bar{n}$	Hyp
10 EG	29. $d' = \bar{n}'$	28, Ref=, LL
11, Q ₈ , UI, BMP	30. $b = \bar{n}'$	10, 29 LL
$\neq 0$, UI	31. $b = 0 \vee \dots \vee b = \bar{n}'$	30 Add
12, 13 Contra	32. $d = \bar{n} \rightarrow b = 0 \vee \dots \vee b = \bar{n}'$	28–31 CP
4–14 RAA	33. $b = 0 \vee \dots \vee b = \bar{n}'$	17, 22, 27, ..., 32 Cases
1–15 CP	34. $b \neq 0 \rightarrow b = 0 \vee \dots \vee b = \bar{n}'$	8–33 CP
Hyp	35. $b = 0 \vee \dots \vee b = \bar{n}'$	7, 34 Inev
Q ₄ , UI	36. $b < \bar{n}'' \rightarrow b = 0 \vee \dots \vee b = \bar{n}'$	2–35 CP
18 EG	37. $b = 0 \vee \dots \vee b = \bar{n}'$	Hyp
19, Q ₈ , UI, BMP	38. $b = 0$	Hyp
17, 20 LL	39. $\bar{n}'' + 0 = \bar{n}''$	Q ₄ , UI
17–21 CP	40. $\exists z z' + 0 = \bar{n}''$	39 EG
16, 22 BI	41. $\exists z z' + b = \bar{n}''$	38, 40 LL
23 UG	42. $b < \bar{n}''$	41, Q ₈ , UI, BMP
e $Q \vdash \forall x(x <$	43. $b = 0 \rightarrow b < \bar{n}''$	38–42 CP
d to show that	44. $b = 0'$	Hyp
$= \overline{n+1}$).	45. $\bar{n}' + 0' = \bar{n}''$	Num+
Ind. hyp.	46. $\exists z z' + 0' = \bar{n}''$	45 EG
Hyp	47. $\exists z z' + b = \bar{n}''$	44, 46 LL
2, Q ₈ , UI, BMP	48. $b < \bar{n}''$	47, Q ₈ , UI, BMP
3 EI	49. $b = 0' \rightarrow b < \bar{n}''$	44–48 CP
Hyp	:	
5 Add	50. $b = \bar{n}'$	Hyp
5–6 CP	51. $0' + \bar{n}' = \bar{n}''$	Num+
Hyp	52. $\exists z z' + \bar{n}' = \bar{n}''$	51 EG
3, Q ₃ , UI, BMP	53. $\exists z z' + b = \bar{n}''$	50, 52 LL
9 EI	54. $b < \bar{n}''$	53, Q ₈ , UI, BMP
4, 10 LL	55. $b = \bar{n}' \rightarrow b < \bar{n}''$	50–54 CP
Q ₅ , UI	56. $b < \bar{n}''$	37, 43, 49, ..., 56 Cases
11, 12 LL	57. $b = 0 \vee \dots \vee b = \bar{n}' \rightarrow b < \bar{n}''$	37–56 CP
13, Q ₁ , UI, MP	58. $b < \bar{n}'' \leftrightarrow b = 0 \vee \dots \vee b = \bar{n}'$	36, 57 BI
14 EG	59. $\forall x(x < \bar{n}'' \leftrightarrow x = 0 \vee \dots \vee x = \bar{n}')$	58 UG
15, Q ₈ , UI, BMP	60. $\forall x(x < \overline{n+1}' \leftrightarrow x = 0 \vee \dots \vee x = \overline{n+1})$	59 Def. $\overline{n+1}$
1, 16, UI, BMP		
Hyp	(Spread) also holds for PA , with the same proof.	

43

Result (Trichotomy): For every number n ,
 $\mathbf{Q} \vdash \forall y (y < \bar{n} \vee \bar{n} < y \vee y = \bar{n})$.

Proof: Induction in the metalanguage again. For the base:

- | | |
|--|-----------------------------|
| 1. $b = 0$ | Hyp |
| 2. $b < 0 \vee 0 < b \vee b = 0$ | 1 Add |
| 3. $b = 0 \rightarrow b < 0 \vee 0 < b \vee b = 0$ | 1–2 CP |
| 4. $b \neq 0$ | Hyp |
| 5. $\exists y \, b = y'$ | 4, Q ₃ , UI, MP |
| 6. $b = c'$ | 5 EI |
| 7. $c' + 0 = c'$ | Q ₄ , UI |
| 8. $c' + 0 = b$ | 6, 7 LL |
| 9. $\exists z \, z' + 0 = b$ | 8 EG |
| 10. $0 < b$ | 9, Q ₈ , UI, BMP |
| 11. $b < 0 \vee 0 < b \vee b = 0$ | 10 Add |
| 12. $b \neq 0 \rightarrow b < 0 \vee 0 < b \vee b = 0$ | 4–11 CP |
| 13. $b < 0 \vee 0 < b \vee b = 0$ | 3, 12 Inev |
| 14. $\forall y (y < 0 \vee 0 < y \vee y = 0)$ | 13 UG |

For the induction step, we assume $\mathbf{Q} \vdash \forall y (y < \bar{n} \vee \bar{n} < y \vee y = \bar{n})$ and proceed to show $\mathbf{Q} \vdash \forall y (y < \overline{n+1} \vee \overline{n+1} < y \vee y = \overline{n+1})$

- | | |
|---|------------------------------|
| 1. $b < \bar{n} \vee \bar{n} < b \vee b = \bar{n}$ | Ind. hyp., UI |
| 2. $b = 0 \vee \dots \vee b = \bar{n} \rightarrow b < \bar{n}'$ | Spread, UI, BE |
| 3. $b < \bar{n} \rightarrow b = 0 \vee \dots \vee b = \overline{n-1}$ | Spread, UI, BE |
| 4. $b < \bar{n}$ | Hyp |
| 5. $b = 0 \vee \dots \vee b = \overline{n-1}$ | 3, 4 MP |
| 6. $b = 0 \vee \dots \vee b = \overline{n-1} \vee b = \bar{n}$ | 5 Add |
| 7. $b < \bar{n}'$ | 2, 6 MP |
| 8. $b < \bar{n}' \vee \bar{n}' < b \vee b = \bar{n}'$ | 7 Add |
| 9. $b < \bar{n} \rightarrow b < \bar{n}' \vee \bar{n}' < b \vee b = \bar{n}'$ | 4–8 CP |
| 10. $\bar{n} < b$ | Hyp |
| 11. $\exists z \, z' + \bar{n} = b$ | 10, Q ₈ , UI, BMP |
| 12. $c' + \bar{n} = b$ | 11 EI |
| 13. $c = 0$ | Hyp |
| 14. $0' + \bar{n} = \bar{n}'$ | Num+ |
| 15. $c' + \bar{n} = \bar{n}'$ | 13, 14 LL |
| 16. $b = \bar{n}'$ | 12, 15 LL |
| 17. $b < \bar{n}' \vee \bar{n}' < b \vee b = \bar{n}'$ | 16 Add |
| 18. $c = 0 \rightarrow b < \bar{n}' \vee \bar{n}' < b \vee b = \bar{n}'$ | 13–17 CP |
| 19. $c \neq 0$ | Hyp |
| 20. $\exists y \, c = y'$ | 19, Q ₃ , UI, MP |

- | | |
|--|-------------------------------|
| 21. $c = d'$ | 20 EI |
| 22. $d'' + \bar{n} = b$ | 12, 21 LL |
| 23. $d'' + \bar{n} = (d' + \bar{n})'$ | NumAltQ ₅ , UI |
| 24. $(d' + \bar{n})' = b$ | 22, 23 LL |
| 25. $d' + \bar{n}' = (d' + \bar{n})'$ | Q ₅ , UI |
| 26. $d' + \bar{n}' = b$ | 24, 25 LL |
| 27. $\exists z \, z' + \bar{n}' = b$ | 26 EG |
| 28. $\bar{n}' < b$ | 27, Q ₈ , UI, BMP |
| 29. $b < \bar{n}' \vee \bar{n}' < b \vee b = \bar{n}'$ | 28 Add |
| 30. $c \neq 0 \rightarrow b < \bar{n}' \vee \bar{n}' < b \vee b = \bar{n}'$ | 19–29 CP |
| 31. $b < \bar{n}' \vee \bar{n}' < b \vee b = \bar{n}'$ | 18, 30 Inev |
| 32. $\bar{n} < b \rightarrow b < \bar{n}' \vee \bar{n}' < b \vee b = \bar{n}'$ | 10–31 CP |
| 33. $b = \bar{n}$ | Hyp |
| 34. $b = 0 \vee \dots \vee b = \bar{n}$ | 33 Add |
| 35. $b < \bar{n}'$ | 2, 34 MP |
| 36. $b < \bar{n}' \vee \bar{n}' < b \vee b = \bar{n}'$ | 35 Add |
| 37. $b = \bar{n} \rightarrow b < \bar{n}' \vee \bar{n}' < b \vee b = \bar{n}'$ | 33–36 CP |
| 38. $b < \bar{n}' \vee \bar{n}' < b \vee b = \bar{n}'$ | 1, 9, 32, 37 Cases |
| 39. $\forall y (y < \overline{n+1} \vee \overline{n+1} < y \vee y = \overline{n+1})$ | 38, UG, Def. $\overline{n+1}$ |

In **PA**, we get not only (Trichotomy) but the object-language quantified version:

$$\mathbf{PA} \vdash \forall x \forall y (y < x \vee x < y \vee y = x)$$

by doing the induction in the object language instead.

5. Segue: Remaining Questions

- **Q** obviously is not, but is **PA** sufficient to establish everything true in the standard interpretation, i.e., is **PA** the same as **TA**?
- Related to this, for every (closed) sentence φ in $\mathcal{L}_A$, is it the case that either $\mathbf{PA} \vdash \varphi$ or $\mathbf{PA} \vdash \neg\varphi$, as it is for **TA**?
- As noted before, **TA** “cheats” as—at least as we have described it—there is no independently effective way to decide whether or not something is allowed as an axiom.
- If the set **PA** doesn’t do it, is there *any* set that “axiomatizes” **TA**, where there is an effective procedure for deciding what is included and what isn’t?

- Is there an effective procedure for deciding what is a member of **TA**, or for that matter the (closed) theories **PA** or **Q** (or for that matter, the closure of $\emptyset$ under logical consequence, i.e., all logical truths)?
- What are the limits to which mathematical properties, relations and functions can be represented in a system such as **Q** or **PA**? What does it even mean for it to “represent” them anyway?

Providing answers to these questions requires getting serious about what we *mean* by an “effective way” of deciding something. We now segue into talking about at least one attempt to make this notion more precise: the theory of recursive functions.

6. Recursive Functions

Background

Informally, we could say that a property or relation is “effectively decidable” if there is a purely mechanical or rote way of determining whether or not it holds for a given thing or given relation, something that could be calculated or computed without any special ingenuity or creativity or room for debate, using a process guaranteed to terminate after a finite number of steps.

Similarly, we could say that a function is “effectively computable” if there is such a means for determining its value for a given argument or arguments. To bridge these:

Definition: The *characteristic function* of n -place relation R , written χ_R , is the n -place function that yields 1, if R holds between its arguments, and yields 0, otherwise.

$$\chi_R(x_0, \dots, x_{n-1}) = \begin{cases} 1, & \text{if } R(x_0, \dots, x_{n-1}) \text{ holds,} \\ 0, & \text{otherwise.} \end{cases}$$

It seems natural to think that a relation is effectively decidable iff its characteristic function is effectively computable.

For the moment we are going to restrict our focus to functions on natural numbers: functions

that take natural numbers as arguments and return natural numbers as values.

Concepts applicable to numbers can often be extended to other things if there are ways of attaching numbers to them, such as attaching a Gödel numbering of formulas.

For the moment, however, we are not considering any particular object language, and the notation in this section should just be taken as that of regular mathematics, as employable in the metalanguage.

The following three sets of functions have been proven to be the same set:

- The set of (general) recursive functions on natural numbers.
- The set of functions on natural numbers representable in the λ -calculus.
- The set of functions on natural numbers computable by a Turing machine.

Membership in any three of these is a tempting analysis of what it means for a function to be “effectively decidable”. The fact that these are all provably the same arguably lends support to the following:

Definition: *The Church-Turing thesis is the thesis that a function is effectively computable if and only if it is (general) recursive/ λ -definable/Turing computable.*

Here we focus on the first notion.

The initial functions

Definition: The *zero function* is the constant function whose value for any natural number is 0, written:

$$\text{zero}(x) = 0.$$

Definition: The *successor function* is the function whose value for any natural number is the next natural number, written:

$$\text{succ}(x) = \text{one more than } x.$$

Note we do not write “ x' ” here, as we are working in the regular mathematics of the metalanguage, not the object language of **Q** or **PA**, and we don’t want to confuse them.

Definition: The n -place **projection function** for position i (where $0 \leq i \leq (n - 1)$) is the function that takes n arguments, and returns the i^{th} argument (counting from 0), unchanged, written:

$$P_i^n(x_0, \dots, x_{n-1}) = x_i.$$

For example $P_1^3(3, 6, 12) = 6$, $P_0^1(55) = 55$, etc. There are different project functions for different values of n and i .

Together, these functions are called the *initial functions*. Clearly, there seems to be an “effective procedure” for calculating their values for any given argument(s).

Methods of defining functions

What ways of defining new functions in terms of others preserve “effective calculability”? The most obvious involves feeding the values of one function into another function as arguments. This is called *composition* or *substitution*:

Definition: If f is an m -place function, and $g_0, \dots, g_{m-1}$ are each n -place functions, then the **composition** of f with $g_0, \dots, g_{m-1}$ is the n -place function h such that:

$$h(x_0, \dots, x_{n-1}) = f(g_0(x_0, \dots, x_{n-1}), \dots, g_{m-1}(x_0, \dots, x_{n-1})).$$

In the simplest case ($n = m = 1$) we have $h(x) = f(g(x))$, which is sometimes also written $(f \circ g)(x)$.

If we know how to calculate f ’s and the g -functions’ values, then to calculate h , we merely calculate the g ’s for the same arguments, and then use those values as arguments to f .

Definition: If f is an n -place function, and g is an $(n + 2)$ -place function, then the function defined by **primitive recursion** from f and g is the $(n + 1)$ -place function h such that:

$$\begin{aligned} h(x_0, \dots, x_{n-1}, 0) &= f(x_0, \dots, x_{n-1}), \text{ and} \\ h(x_0, \dots, x_{n-1}, y + 1) &= g(x_0, \dots, x_{n-1}, y, h(x_0, \dots, x_{n-1}, y)), \end{aligned}$$

Or, if k is a constant natural number, and g is a two-place function, then the function defined by primitive recursion from k and g is the one-place function h such that:

$$\begin{aligned} h(0) &= k \\ h(y + 1) &= g(y, h(y)). \end{aligned}$$

Recursive definitions are to definitions what induction is to proving things. We define its value for 0 as the (last) argument, then this gives us a means for determining the next value, and so on, where we can “work our way up” to the value for any given natural number.

Definition: If f is $(n + 1)$ -place function which is such that for every set of natural numbers $x_0, \dots, x_{n-1}$, there is always at least one number y where $f(x_0, \dots, x_{n-1}, y) = 0$, then the function defined by **minimization** on f is the n -place function h such that:

$$\begin{aligned} h(x_0, \dots, x_{n-1}) &= \text{the least } y \text{ such that} \\ f(x_0, \dots, x_{n-1}, y) &= 0 \end{aligned}$$

This method is also called “unbounded search” or the “choice of least rule”.

This one is a bit more involved. If we know how to calculate f , we can calculate h for arguments $x_0, \dots, x_{n-1}$ by calculating $f(x_0, \dots, x_{n-1}, 0)$ and then $f(x_0, \dots, x_{n-1}, 1)$ and so on until we find the first one that returns 0, which is then the value of h for those arguments.

The set of (primitive) recursive functions

Definition: The set of **primitive recursive functions** is the smallest set of functions containing the zero function, the successor function, the projection functions, and is closed under composition and primitive recursion.

Or put another way, a function is primitive recursive iff:

- (1) it is one of the initial functions (zero, succ, P_i^n);
- or

- (2) it can be defined by a finite series of steps of recursion and/or composition from functions already shown to be primitive recursive.

Definition: The set of **(general) recursive functions** is the smallest set of functions containing the zero function, the successor function, the projection functions and is closed under composition, primitive recursion and minimization.

A function is general recursive if it can be defined from other recursive functions, starting with the initial functions, using a finite series of steps of the methods of composition, primitive recursion and minimization.

All primitive recursive functions are also general recursive; the reverse is not true, but we shall not bother with the proofs of this in this course.

Definition: Relation R is **(primitive) recursive relation** iff its characteristic function is (primitive) recursive. (A property can be considered a 1-place relation, this also defines the notion of “(primitive) recursive property.”)

Some convenience results

Result (dummy arguments): If the n -place function f is (primitive) recursive, then so is the $(n + 1)$ -place function h , whose value, $h(x_0, \dots, x_{n-1}, x_n)$, is always simply $f(x_0, \dots, x_{n-1})$, so that the last argument to h is always simply ignored.

Proof: Function h can be defined by composition using f and the projection functions:

$$h(x_0, \dots, x_{n-1}, x_n) = f(P_0^{n+1}(x_0, \dots, x_n), \dots, P_{n-1}^{n+1}(x_0, \dots, x_n)). \quad *$$

Result (permuting arguments): If n -place function f is (primitive) recursive, then so is the n -place function h whose value, $h(\dots, x_i, \dots, x_j, \dots)$, is always $f(\dots, x_j, \dots, x_i, \dots)$.

Proof: Again, using composition and projection:

$$h(\dots, x_i, \dots, x_j, \dots) = f(\dots, P_j^n(x_0, \dots, x_{n-1}), \dots, P_i^n(x_0, \dots, x_{n-1}), \dots). \quad *$$

Result (identifying arguments): If the $(n + 1)$ -place function f is (primitive) recursive, then so is n -place function h whose value, $h(x_0, \dots, x_{n-1})$ is always $f(x_0, x_0, \dots, x_{n-1})$.

Proof: Our method is similar to the above:

$$h(x_0, \dots, x_{n-1}) = f(P_0^n(x_0, \dots, x_{n-1}), P_0^n(x_0, \dots, x_{n-1}), \dots, P_{n-1}^n(x_0, \dots, x_{n-1})). \quad *$$

Result (all polyadic zero functions): For any n , the n -place zero function zero^n is primitive recursive.

Proof: This follows by composition, since:

$$\text{zero}^n(x_0, \dots, x_{n-1}) = \text{zero}(P_0^n(x_0, \dots, x_{n-1})) \quad *$$

Result (all constant functions): For any n and k , the n -place constant function const_k^n , the value of which for any n -arguments, is always k regardless of what the arguments are, is primitive recursive.

Proof: This can be proven by induction on k . For $k = 0$, the n -place constant function is the same as the n -place zero function. For the rest, the n -place constant function whose value is always $k + 1$ can be defined by composition since:

$$\text{const}_{k+1}^n(x_0, \dots, x_{n-1}) = \text{succ}(\text{const}_k^n(x_0, \dots, x_{n-1})).$$

*

The upshot of the above results is that we need not be fussy when giving definitions using composition, recursion, etc., about the order and number of arguments, and about whether or not we use a constant value where a function is technically needed.

Examples

Result: *The functions below are primitive recursive.*

- (a) Addition: $x + y$. Definable by recursion:
 $x + 0 = P_0^1(x) = x$
 $x + (y + 1) = \text{succ}(x + y)$
- (b) Multiplication: $x \times y$. Recursion again:
 $x \times 0 = \text{zero}(x) = 0$
 $x \times (y + 1) = (x \times y) + x$
- (c) x to the power of y : x^y . Recursion:
 $x^0 = \text{const}_1^1(x) = 1$
 $x^{y+1} = (x^y) \times x$
- (d) Predecessor: $\text{pred}(x)$. Recursion:
 $\text{pred}(0) = 0$
 $\text{pred}(y + 1) = P_0^2(y, \text{pred}(y)) = y$
- (e) Subtract-as-much-as-you-can: $x \div y$. Recursion:
 $x \div 0 = x$
 $x \div (y + 1) = \text{pred}(x \div y)$
- (f) Absolute difference: $|x - y|$. Composition:
 $|x - y| = (x \div y) + (y \div x)$
- (g) Factorial: $x!$ Recursion:
 $0! = 1$
 $(y + 1)! = y! \times (y + 1)$
- (h) Minimum of 2 arguments: $\min(x, y)$. Composition:
 $\min(x, y) = x \div (x \div y)$

- (i) For any $n > 2$, the minimum of n arguments, because each such function can be defined by substitution using the previous one:
 $\min(x_0, \dots, x_{n-1}, x_n) = \min(\min(x_0, \dots, x_{n-1}), x_n)$
- (j) Maximum of 2 (or more) arguments:
 $\max(x, y) = y + (x \div y)$
 $\max(x_0, \dots, x_n) = \max(\max(x_0, \dots, x_{n-1}), x_n)$
- (k) Characteristic function of not being zero:
 $\chi_{\text{NotZero}}(x) = x \div \text{pred}(x)$
- (l) Characteristic function of being zero:
 $\chi_{\text{IsZero}}(x) = 1 \div \text{NotZero}(x)$
- (m) Remainder upon division of y by x : $\text{rem}(x, y)$. Recursion:
 $\text{rem}(x, 0) = 0$
 $\text{rem}(x, y + 1) = \text{succ}(\text{rem}(x, y)) \times \chi_{\text{NotZero}}(|x - \text{succ}(\text{rem}(x, y))|)$
- (n) Quotient upon division of y by x : $d(x, y)$. (Rounded down.) Recursion:
 $d(x, 0) = 0$
 $d(x, y + 1) = d(x, y) + \chi_{\text{IsZero}}(|x - \text{succ}(\text{rem}(x, y))|)$

HOMEWORK 20

Define the **graph** of n -place function f , written γ_f , to be the $(n + 1)$ -place relation that holds between $x_0, \dots, x_{n-1}, x_n$ iff $f(x_0, \dots, x_{n-1}) = x_n$. Show that γ_f is a general recursive relation iff f is a general recursive function.

Bounded sums and products

The following notation:

$$\sum_{z < y} f(x_0, \dots, x_{n-1}, z)$$

stands for the $(n + 1)$ -place **bounded sum** function h , whose value for $x_0, \dots, x_{n-1}, y$ as arguments is the sum of all the values of f for $x_0, \dots, x_{n-1}, 0$ through $x_0, \dots, x_{n-1}, y$.

Result: *If f is (primitive) recursive, then so is the bounded sum function h , as explained above.*

Proof: The function h can be defined by recursion as follows:

$$\begin{aligned} h(x_0, \dots, x_{n-1}, 0) &= 0 \\ h(x_0, \dots, x_{n-1}, y+1) &= h(x_0, \dots, x_{n-1}, y) + f(x_0, \dots, x_{n-1}, y) \end{aligned} \quad *$$

Similar results follow for the form:

$$\sum_{z \leq y} f(x_0, \dots, x_{n-1}, z)$$

This is definable by composition, since:

$$\sum_{z \leq y} f(x_0, \dots, x_{n-1}, z) = \sum_{z < y+1} f(x_0, \dots, x_{n-1}, z)$$

Similarly for doubly bounded sums:

$$\sum_{y < z < v} f(x_0, \dots, x_{n-1}, z) = \sum_{z < \text{pred}(v \div y)} f(x_0, \dots, x_{n-1}, z + y + 1)$$

The following notation:

$$\prod_{z < y} f(x_0, \dots, x_{n-1}, z)$$

stands for the $(n+1)$ -place **bounded product** function h , whose value for $x_0, \dots, x_{n-1}, y$ as arguments is the product of all the values of f for $x_0, \dots, x_{n-1}, 0$ through $x_0, \dots, x_{n-1}, y-1$.

Result: If f is (primitive) recursive, so is the bounded product function h , as defined above.

HOMEWORK 21

Prove the result above.

Similar results follow for bounded products for $z \leq y$, as well as doubly bounded products ($y < z < v$).

Recursive properties and relations

Result: The equality, less-than-or-equal-to and evenly-dividing-into relations are primitive recursive.

Proof:

$$\begin{aligned} \chi_{=}(x, y) &= \chi_{\text{IsZero}}(|x - y|), \\ \chi_{\leq}(x, y) &= \chi_{\text{IsZero}}(x \div y), \\ \chi_{|}(x, y) &= \chi_{\text{IsZero}}(\text{rem}(x, y)). \end{aligned} \quad *$$

*

Definition: The **negation of a relation R** , viz., “not- R ”, is the relation that holds of $x_0, \dots, x_{n-1}$ iff R does not hold of $x_0, \dots, x_{n-1}$.

Definition: The **conjunction of relations R and S** , written “ R -and- S ”, is the relation that holds of $x_0, \dots, x_{n-1}$ iff R and S both hold of $x_0, \dots, x_{n-1}$.

Definition: The **disjunction of two relations R and S** , written “ R -or- S ”, is the relation that holds for $x_0, \dots, x_{n-1}$ iff either R or S hold of $x_0, \dots, x_{n-1}$.

(And so on for other propositional connectives.)

Result (propositional connectives):

If R and S are (primitive) recursive relations, then so are their negations, conjunctions, disjunctions, and so on.

Proof: By definition, if R and S are (primitive) recursive, their characteristic functions χ_R and χ_S are (primitive) recursive, in which case the characteristic functions of their negations and disjunctions can be defined as follows:

$$\begin{aligned} \chi_{\text{not-}R}(x_0, \dots, x_{n-1}) &= 1 \div \chi_R(x_0, \dots, x_{n-1}) \\ \chi_{R\text{-and-}S}(x_0, \dots, x_{n-1}) &= \chi_R(x_0, \dots, x_{n-1}) \times \chi_S(x_0, \dots, x_{n-1}) \end{aligned}$$

Other propositional operations on relations can be defined in terms of conjunction and negation. *

Example: $\chi_{<}(x, y) = \chi_{\leq}(x, y) \times (1 \div \chi_{=}(x, y))$.

I use the notation:

$$\bigvee_{z < y}^z R(x_0, \dots, x_{n-1}, z)$$

in the metalanguage(!) to stand for the relation Q that holds for $x_0, \dots, x_{n-1}, y$ iff the relation R

holds for all arguments $x_0, \dots, x_{n-1}, z$ for every $z < y$. (The Open Logic Text writes instead “ $(\forall z < y)R(x_0, \dots, x_{n-1}, z)$ ”, but I find this too close to the notation used in the object language, and potentially confusing.)

Result (bounded quantification): *If R is a (primitive) recursive number theoretic relation, so is the universally quantified relation Q as annotated above.*

Proof: The characteristic function for Q can be defined in terms of the characteristic function for R by substitution as follows:

$$\chi_Q(x_0, \dots, x_{n-1}, y) = \prod_{z < y} \chi_R(x_0, \dots, x_{n-1}, z). \quad *$$

HOMEWORK 22

Prove that

$$\exists_{z < y} R(x_0, \dots, x_{n-1}, z)$$

(where this means what you expect) is (primitive) recursive if R is.

Similar results follow for bounded quantifiers using $\leq$, doubly bounded quantifiers, etc.

The notation:

$$(\mu y)R(x_0, \dots, x_{n-1}, y)$$

represents the n -place function whose value for $x_0, \dots, x_{n-1}$ as arguments returns the *least* y such that the $(n + 1)$ -place relation R holds between $x_0, \dots, x_{n-1}$ and y .

Result (unbounded search operator): *If R is a recursive relation such that for every set of numbers $x_0, \dots, x_{n-1}$, there is y such that $x_0, \dots, x_{n-1}, y$ holds for some y , then $(\mu y)R(x_0, \dots, x_{n-1}, y)$ is a recursive function.*

Proof: Suppose R is such a relation. By our earlier result, so is $\text{not-}R$, and its characteristic function $\chi_{\text{not-}R}$ will return 0 for some y for every $x_0, \dots, x_{n-1}$. Hence, we can define $(\mu y)R(x_0, \dots, x_{n-1}, y)$ using minimization as yielding the least y such that $\chi_{\text{not-}R}(x_0, \dots, x_{n-1}, y) = 0$. $*$

Note that if we use the “unbounded” search operator “ (μx) ” when defining functions, the results are only guaranteed to be recursive, not primitive recursive, even if the relation used in the definition is primitive recursive.

However, if we can define the function using a *bounded* search operator instead, the result will be primitive recursive if the relation is.

Result (bounded search operator): *If R is a (primitive) recursive relation, then the function:*

$$(\mu y < z)R(x_0, \dots, x_{n-1}, y)$$

whose value for $x_0, \dots, x_{n-1}, z$ as arguments, is the least y less than z such that $R(x_0, \dots, x_{n-1}, y)$ holds, and whose value is y if there is no such z , is a (primitive) recursive function

Proof: The function can be defined using the characteristic function of $\text{not-}R$ by composition as follows:

$$(\mu y < z)R(x_0, \dots, x_{n-1}, y) =$$

$$\sum_{y < z} \left(\prod_{w \leq y} \chi_{\text{not-}R}(x_0, \dots, x_{n-1}, w) \right). \quad *$$

As z increases, the bounded product will keep adding 1 to the bounded sum so long as R does not hold for any $x_0, \dots, x_{n-1}, w$ where $w \leq z$. As soon as a z is reached for which R does hold for some $x_0, \dots, x_{n-1}, w$ where $w \leq z$, the bounded product will stop adding to the bounded sum, and so the result will be identical to the least such z .

Notice the above proof does not make use of minimization, and so something defined in this way will be primitive recursive assuming R is.

Result (conditional definitions):

If $R_0, \dots, R_{m-1}$ are exclusive (primitive) recursive relations, and $f_0, \dots, f_m$ are (primitive) recursive functions, and function h is such that

$$h(x_0, \dots, x_{n-1}) = \begin{cases} f_0(x_0, \dots, x_{n-1}), & \text{if } R_0(x_0, \dots, x_{n-1}), \\ \vdots \\ f_m(x_0, \dots, x_{n-1}), & \text{otherwise.} \end{cases}$$

then h is also (primitive) recursive.

Proof:

$$h(x_0, \dots, x_{n-1}) = (f_0(x_0, \dots, x_{n-1}) \times \chi_{R_0}(x_0, \dots, x_{n-1})) + \dots + (f_m(x_0, \dots, x_{n-1}) \times \chi_{\text{not-}(R_0\text{-or-}\dots\text{-or-}R_{m-1})}(x_0, \dots, x_{n-1})).$$

✱

From here on out, we often simply give direct definitions of properties and relations in terms of notions already shown to be (primitive) recursive, rather than defining their characteristic functions. The transitions are usually not difficult.

Coding functions

Recall that the “code” for a given sequence of numbers $k_0, \dots, k_{n-1}$ is the product of the first n primes raised to the power one greater than each of these numbers:

$$2^{k_0+1} \times 3^{k_1+1} \times \dots \times (p_{n-1})^{k_{n-1}+1}.$$

Result: *The functions, properties and relations below are primitive recursive.*

(a) Being prime:

$$\text{prime}(x) = (2 \leq x)\text{-and-}$$

$$\bigvee_{y \leq x}^y (\text{if } y|x \text{ then } y = 1 \text{ or } y = x)$$

(b) Lowest prime above x :

$$\text{nextPrime}(x) =$$

$$(\mu y \leq x! + 1)((x < y)\text{-and-}\text{prime}(y))$$

(c) The x^{th} prime, starting from 0. (Recursion.)

$$p_0 = 2$$

$$p_{y+1} = \text{nextPrime}(p_y)$$

(d) The length of the sequence coded by x :

$$\text{len}(x) = \begin{cases} 0, & \text{if } x = 0 \text{ or } x = 1 \\ 1 + (\mu y < x)(p_y|x \text{ and not-}(p_{y+1}|x)), & \text{otherwise.} \end{cases}$$

(e) The code of the sequence appending number y to the sequence encoded by x :

$$\text{append}(x, y) = \begin{cases} 2^{y+1}, & \text{if } x = 0 \text{ or } x = 1, \\ x \times (p_{\text{len}(x)})^{y+1}, & \text{otherwise.} \end{cases}$$

(f) For any $n \geq 1$, the code of $x_1, \dots, x_n$:

$$\text{code}^1(x_1) = \text{append}(0, x_1)$$

$$\text{code}^{n+1}(x_1, \dots, x_n, x_{n+1}) = \text{append}(\text{code}^n(x_1, \dots, x_n), x_{n+1})$$

(g) The y^{th} member of the series encoded by x :

$$(x)_y = \begin{cases} 0, & \text{if } \text{len}(x) < y, \\ (\mu z < x) \text{ not-}(p_y^{z+2}|x), & \text{otherwise.} \end{cases}$$

(h) The code of the sequence resulting from concatenating the first z elements of the sequence encoded by y to the sequence encoded by x :

$$\text{hconcat}(x, y, 0) = x$$

$$\text{hconcat}(x, y, z + 1) =$$

$$\text{append}(\text{hconcat}(x, y, z), (y)_z)$$

(i) The code of the sequence which concatenates the full sequence encoded by y to the end of sequence encoded by x :

$$x \frown y = \text{hconcat}(x, y, \text{len}(y))$$

Course-of-values recursion

Because a single number can be used to encode a finite sequence of numbers, it is possible to define a function whose value for x as argument encodes the sequence (“course”) of values of another function for all arguments leading up to and including x .

If f is a $(n + 1)$ -place operation on natural numbers, then the notation “ $f\#$ ” is used for the $(n + 1)$ -place number-theoretic function whose value for $x_0, \dots, x_{n-1}, y$ is the number that encodes the series of values for f for all sets of arguments starting

from $x_0, \dots, x_{n-1}, 0$ and ending with $x_0, \dots, x_{n-1}, y - 1$.

Result: A function f is (primitive) recursive iff $f^\#$ is (primitive) recursive.

Proof: We prove this in both directions.

- (a) Suppose that f has already been shown to be (primitive) recursive. One can then obtain $f^\#$ by composition as follows:

$$f^\#(x_0, \dots, x_{n-1}, y) = \prod_{z < y} (p_z)^{f(x_0, \dots, x_{n-1}, z) + 1}$$

- (b) On the other hand, suppose that $f^\#$ has already been shown to be (primitive) recursive. One can then obtain f as follows:

$$f(x_0, \dots, x_{n-1}, y) = (f^\#(x_1, \dots, x_n, y + 1))_y \quad *$$

Sometimes it is easier to define $f^\#$ recursively than it is to define f , especially, a function whose value for a given number depends not only upon its value for the previous number, but upon more than one or even all of its prior values. Such functions are said to be obtained by **course-of-values recursion**, rather than simple recursion.

Example: Consider $\text{fib}(x)$ whose value for any x is the x th item in the **Fibonacci sequence**, which adds the previous two members to get the next (starting with 1, 1):

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ..., and so on.

This function cannot be defined by the recursion rule in a simple way, because its value for $y + 1$ depends not only on its value for y but also on its value for $y - 1$.

However, $\text{fib}^\#$ can be defined by recursion as follows:

$$\begin{aligned} \text{fib}^\#(0) &= 0 \\ \text{fib}^\#(y + 1) &= \begin{cases} 2^2, & \text{if } y = 0, \\ 2^2 \cdot 3^2, & \text{if } y = 1, \\ \text{append}(\text{fib}^\#(y), (\text{fib}^\#(y))_{y-2} + (\text{fib}^\#(y))_{y-1}), & \text{otherwise.} \end{cases} \end{aligned}$$

Since $\text{fib}^\#$ is primitive recursive, and fib can be obtained from it by composition:

$$\text{fib}(x) = (\text{fib}^\#(x + 1))_x$$

The function fib is also primitive recursive.

Similar results follow for other functions one would want to define in a similar sort of way. Generally, we can say that if a function f is obtained from (primitive) recursive functions by course-of-values recursion, then f is (primitive) recursive itself.

Course-of-values recursion is to simple recursion what strong induction is to weak induction.

HOMEWORK 23

Show that the function g , described below, is primitive recursive, by giving an explicit definition of $g^\#$:

$$g(0) = 2 \text{ and } g(1) = 4 \text{ and}$$

$$g(x + 2) = (3 \times g(x + 1)) - ((2 \times g(x)) + 1)$$

From now on, we will not fuss about recursive definitions always only appealing to the value for the previous argument, but also allow such definitions that appeal to the value for any lesser argument.

7. Representing Recursive Functions in Q/PA

Definitions

We now return to our discussion of formal systems. Our eventual goal to prove that the functions that can be represented in systems **Q** and **PA** are precisely the recursive functions.

First we need to make the notion of *representability* precise.

Definition: If **T** is a theory, and f is an n -place operation on the natural numbers, then f is **representable** in **T** iff there an open formula with $n+1$ free variables, $\phi(x_1, \dots, x_n, y)$, such that, for any set of natural numbers $k_1, \dots, k_n, m$, where $f(k_1, \dots, k_n) = m$, we have:

- (a) $\mathbf{T} \vdash \varphi(\overline{k_1}, \dots, \overline{k_n}, \overline{m})$; and
(b) $\mathbf{T} \vdash \forall y(\varphi(\overline{k_1}, \dots, \overline{k_n}, y) \rightarrow \overline{m} = y)$.

Note $\varphi(x_1, \dots, x_n, y)$ need not take the form $y = f(x_1, \dots, x_n)$ for some object-language function letter f (though it could).

Definition: If $\mathbf{T}$ is a theory, and R is an n -place relation between natural numbers, then R is **representable** in $\mathbf{T}$ iff there an open formula with n free variables, $\varphi(x_1, \dots, x_n)$, such that, for any set of natural numbers $k_1, \dots, k_n, \dots$:

- (a) If $R(k_1, \dots, k_n)$ then $\mathbf{T} \vdash \varphi(\overline{k_1}, \dots, \overline{k_n})$; and
(b) If not- $R(k_1, \dots, k_n)$, then $\mathbf{T} \vdash \neg\varphi(\overline{k_1}, \dots, \overline{k_n})$.

Result: Every function and relation representable in $\mathbf{Q}$ is also representable in $\mathbf{PA}$.

Proof: Since every theorem of $\mathbf{Q}$ is also a theorem of $\mathbf{PA}$, the results needed to obtain that a given function or relation is represented in $\mathbf{Q}$ by a given formula are also obtainable in $\mathbf{PA}$ for the same formula. *

The initial functions

Result: The zero function is representable in $\mathbf{Q}$.

Proof: Let $\varphi(x, y)$ be the formula $x = x \wedge 0 = y$. Suppose $\text{zero}(k) = m$. Then m is zero, and $\overline{m}$ is “0”, and so $\mathbf{Q} \vdash \overline{k} = \overline{k} \wedge 0 = \overline{m}$ by two appeals to $\text{Ref} =$ and conjoining. Moreover, $\mathbf{Q} \vdash \forall y(\overline{k} = \overline{k} \wedge 0 = y \rightarrow \overline{m} = y)$, i.e., $\mathbf{Q} \vdash \forall y(\overline{k} = \overline{k} \wedge 0 = y \rightarrow 0 = y)$, by generalizing on the tautology $\overline{k} = \overline{k} \wedge 0 = b \rightarrow 0 = b$. *

Result: The successor function is representable in $\mathbf{Q}$.

Proof: Let $\varphi(x, y)$ be $y = x'$. Suppose that $\text{succ}(k) = m$. This means that m is one more than k and the numeral for $\overline{m}$ is $\overline{k'}$. Hence $\mathbf{Q} \vdash \overline{m} = \overline{k'}$ by $\text{Ref} =$, and $\mathbf{Q} \vdash \forall y(y = \overline{k'} \rightarrow y = \overline{m})$ by generalizing on a tautology. *

HOMEWORK 24

Prove that addition and multiplication are representable in $\mathbf{Q}$.

Result: For any n and i , the projection function P_i^n is representable in $\mathbf{Q}$.

Proof: Let φ be the formula $x = x \wedge x_1 = x_1 \wedge \dots \wedge x_{n-1} = x_{n-1} \wedge x_i = y$. Now suppose $P_i^n(k_0, \dots, k_{n-1}) = m$. This means that $m = k_i$, and so they have the same numeral. Hence we have $\mathbf{Q} \vdash \overline{k_0} = \overline{k_0} \wedge \dots \wedge \overline{k_{n-1}} = \overline{k_{n-1}} \wedge \overline{k_i} = \overline{m}$ by $\text{Ref} =$ and conjoining. Similarly, $\mathbf{Q} \vdash \forall y(\overline{k_0} = \overline{k_0} \wedge \dots \wedge \overline{k_{n-1}} = \overline{k_{n-1}} \wedge \overline{k_i} = y \rightarrow \overline{m} = y)$ by generalizing on a tautology. *

Representable relations

Result: The equality relation is representable in $\mathbf{Q}$.

Proof: Consider the wff $x_1 = x_2$. If k_1 is equal to k_2 , they have the same numeral, and so $\mathbf{Q} \vdash \overline{k_1} = \overline{k_2}$ by $\text{Ref} =$. If k_1 is not k_2 , then by $\text{Num} =$, $\mathbf{Q} \vdash \overline{k_1} \neq \overline{k_2}$. *

Result: The less-than relation is representable in $\mathbf{Q}$.

Proof: Consider the wff $x_1 < x_2$. Suppose k_1 is less than k_2 . Then there is some non-zero number j such that $j + k_1 = k_2$. By $\text{Num} +$, $\mathbf{Q} \vdash \overline{j} + \overline{k_1} = \overline{k_2}$.

Since j is not zero, the numeral for j is of the form $\overline{j-1}$, and so by EG, Q_8 and BMP, $\mathbf{Q} \vdash \overline{k_1} < \overline{k_2}$. Suppose that k_1 is not less than k_2 . Either k_2 is 0, or it isn't. If it is, then we have $Q \vdash \neg \overline{k_1} < \overline{k_2}$ from ($\neq$ 0). Suppose k_2 is greater than 0. We know that k_1 is distinct from every number less than k_2 , so, by Num=:

$$\begin{aligned} \mathbf{Q} &\vdash \overline{k_1} \neq 0 \\ \mathbf{Q} &\vdash \overline{k_1} \neq \overline{1} \\ &\vdots \\ \mathbf{Q} &\vdash \overline{k_1} \neq \overline{k_2 - 1} \end{aligned}$$

Therefore, $\mathbf{Q} \vdash \neg(\overline{k_1} = 0 \vee \overline{k_1} = \overline{1} \vee \dots \vee \overline{k_1} = \overline{k_2 - 1})$, and so by (Spread) and BMT, $\mathbf{Q} \vdash \neg \overline{k_1} < \overline{k_2}$. *

Result: If relation R is representable in $\mathbf{Q}$, then so is its characteristic function χ_R .

- Proof:* (1) Suppose R is representable in $\mathbf{Q}$ by $\psi(x_1, \dots, x_n)$.
(2) Let $\varphi(x_1, \dots, x_n)$ be $(\psi(x_1, \dots, x_n) \wedge y = \overline{1}) \vee (\neg \psi(x_1, \dots, x_n) \wedge y = 0)$.
(3) Suppose $\chi_R(k_1, \dots, k_n) = m$. Then either $R(k_1, \dots, k_n)$ and $m = 1$ or not- $R(k_1, \dots, k_n)$ and $m = 0$.
(4) Consider the first case. Then because $\psi(x_1, \dots, x_n)$ represents R , we have $\mathbf{Q} \vdash \psi(\overline{k_1}, \dots, \overline{k_n})$; and we have $\mathbf{Q} \vdash \overline{m} = \overline{1}$ by Ref=. From these by Conj and Add we get $\mathbf{Q} \vdash (\psi(\overline{k_1}, \dots, \overline{k_n}) \wedge \overline{m} = \overline{1}) \vee (\neg \psi(\overline{k_1}, \dots, \overline{k_n}) \wedge \overline{m} = 0)$.
(5) Assume in the object language that $(\psi(\overline{k_1}, \dots, \overline{k_n}) \wedge b = \overline{1}) \vee (\neg \psi(\overline{k_1}, \dots, \overline{k_n}) \wedge b = 0)$. We already have $\mathbf{Q} \vdash \psi(\overline{k_1}, \dots, \overline{k_n})$, and so we can eliminate the second disjunct. Thus we get $b = \overline{1}$, i.e., $b = \overline{m}$, and so by symmetry of identity $\overline{m} = b$. Discharging and generalizing $\mathbf{Q} \vdash \forall y((\psi(\overline{k_1}, \dots, \overline{k_n}) \wedge y = \overline{1}) \vee (\neg \psi(\overline{k_1}, \dots, \overline{k_n}) \wedge y = 0) \rightarrow \overline{m} = y)$.
(6) The proof for the second case is just like the first, with the roles of the two conjuncts of $\varphi(x_1, \dots, x_n)$ reversed. *

Result: If the characteristic function χ_R of relation R is representable in $\mathbf{Q}$, then so is R .

HOMEWORK 25

Prove the result above. Use the fact (from Q_2) that $\mathbf{Q} \vdash 0 \neq \overline{1}$.

Composition and minimization

Result: The set of functions representable in $\mathbf{Q}$ (or any other theory built on identity logic) is closed under composition, i.e., if $f, g_0, \dots, g_{m-1}$ are each representable in $\mathbf{Q}$, then the function h such that:

$$h(x_0, \dots, x_{n-1}) = f(g_0(x_0, \dots, x_{n-1}), \dots, g_{m-1}(x_0, \dots, x_{n-1})).$$

is also representable in $\mathbf{Q}$.

- Proof:* (1) Suppose that $f, g_0, \dots, g_{m-1}$ are all representable in $\mathbf{Q}$. Let the wffs that represent them be, respectively $\xi(x_1, \dots, x_m, y)$ and $\psi_0(x_1, \dots, x_n, y), \dots, \psi_{m-1}(x_1, \dots, x_n, y)$.
(2) Now consider the wff below, which I'll abbreviate $\varphi(x_1, \dots, x_n, y)$:

$$\begin{aligned} \exists z \exists z_1 \dots \exists z_{m-1} [&\psi_0(x_1, \dots, x_n, z) \wedge \\ &\psi_1(x_1, \dots, x_n, z_1) \wedge \dots \wedge \psi_{m-1}(x_1, \dots, x_n, z_{m-1}) \wedge \\ &\xi(z, z_1, \dots, z_{m-1}, y)]. \end{aligned}$$

- (3) Suppose now that $h(k_1, \dots, k_n) = l$, and let the values of $g_0, \dots, g_{m-1}$ for $k_1, \dots, k_n$ as arguments be $j_0, \dots, j_{m-1}$ respectively; this means that $f(j_0, \dots, j_{m-1}) = l$.

- (4) Because these formulas represent these functions, we have:

$$\begin{aligned} \mathbf{Q} &\vdash \psi_0(\bar{k}_1, \dots, \bar{k}_n, \bar{j}_0) \\ \mathbf{Q} &\vdash \psi_1(\bar{k}_1, \dots, \bar{k}_n, \bar{j}_1) \\ &\vdots \\ \mathbf{Q} &\vdash \psi_{m-1}(\bar{k}_1, \dots, \bar{k}_n, \bar{j}_{m-1}) \\ \mathbf{Q} &\vdash \xi(\bar{j}_0, \dots, \bar{j}_{m-1}, \bar{l}) \end{aligned}$$

- (5) By Conj, and EG, $\mathbf{Q} \vdash \varphi(\bar{k}_1, \dots, \bar{k}_n, \bar{l})$.
(6) In the object language, assume $\varphi(\bar{k}_1, \dots, \bar{k}_n, b)$ for arbitrary b .
(7) By EI and Simp, for some constants $c, c_1, \dots, c_{m-1}$, we get:

$$\begin{aligned} \psi_0(\bar{k}_1, \dots, \bar{k}_n, c) \\ \psi_1(\bar{k}_1, \dots, \bar{k}_n, c_1) \\ \vdots \\ \psi_{m-1}(\bar{k}_1, \dots, \bar{k}_n, c_{m-1}) \\ \xi(c, c_1, \dots, c_{m-1}, b) \end{aligned}$$

- (8) By the definition of representation, we also have:

$$\begin{aligned} \psi_0(\bar{k}_1, \dots, \bar{k}_n, c) &\rightarrow \bar{j}_0 = c \\ \psi_1(\bar{k}_1, \dots, \bar{k}_n, c_1) &\rightarrow \bar{j}_1 = c_1 \\ &\vdots \\ \psi_{m-1}(\bar{k}_1, \dots, \bar{k}_n, c_{m-1}) &\rightarrow \bar{j}_{m-1} = c_{m-1} \end{aligned}$$

- (9) Putting the results above together, by LL, we get $\xi(\bar{j}_0, \dots, \bar{j}_{m-1}, b)$.
(10) We have $\forall y(\xi(\bar{j}_0, \dots, \bar{j}_{m-1}, y) \rightarrow \bar{l} = y)$, and so we get $\bar{l} = b$.
(11) Hence $\mathbf{Q} \vdash \varphi(\bar{k}_1, \dots, \bar{k}_n, b) \rightarrow \bar{l} = b$ by DT.
(12) Hence $\mathbf{Q} \vdash \forall y(\varphi(\bar{k}_1, \dots, \bar{k}_n, y) \rightarrow \bar{l} = y)$ by UG.
(13) Together (5) and (12) mean that h is representable by $\varphi(x_1, \dots, x_n, y)$. $\ast$

Result: The set of functions representable in $\mathbf{Q}$ is closed under minimization, i.e., if f is a function for which there is always a y such that

$f(x_0, \dots, x_{n-1}, y) = 0$ for any given natural numbers $x_0, \dots, x_{n-1}$, and f is representable in $\mathbf{Q}$, then so is the function h where:

$$h(x_0, \dots, x_{n-1}) = \text{the least } y \text{ such that } f(x_0, \dots, x_{n-1}, y) = 0$$

is also representable in $\mathbf{Q}$.

Proof: (1) Suppose f is such a function, and suppose f is represented by the wff $\psi(x_1, \dots, x_n, x_{n+1}, y)$.

- (2) By definition, if $f(k_1, \dots, k_n, k_{n+1}) = m$, then

$$\begin{aligned} \mathbf{Q} &\vdash \psi(\bar{k}_1, \dots, \bar{k}_n, \bar{k}_{n+1}, \bar{m}) \\ \mathbf{Q} &\vdash \forall y(\psi(\bar{k}_1, \dots, \bar{k}_n, \bar{k}_{n+1}, y) \rightarrow \bar{m} = y). \end{aligned}$$

- (3) Let $\varphi(x_1, \dots, x_n, y)$ be the wff:

$$\psi(x_1, \dots, x_n, y, 0) \wedge \forall z(z < y \rightarrow \neg \psi(x_1, \dots, x_n, z, 0))$$

We will show that this wff represents h .

- (4) Suppose that $h(k_1, \dots, k_n) = m$; we need to show that:

$$\mathbf{Q} \vdash \psi(\bar{k}_1, \dots, \bar{k}_n, \bar{m}, 0) \wedge \forall z(z < \bar{m} \rightarrow \neg \psi(\bar{k}_1, \dots, \bar{k}_n, z, 0)) \quad (4a)$$

$$\mathbf{Q} \vdash \forall y[\psi(\bar{k}_1, \dots, \bar{k}_n, y, 0) \wedge \forall z(z < y \rightarrow \neg \psi(\bar{k}_1, \dots, \bar{k}_n, z, 0)) \rightarrow \bar{m} = y] \quad (4b)$$

- (5) Since $h(k_1, \dots, k_n) = m$, we know that:

(5a) $f(k_1, \dots, k_n, m) = 0$ and

(5b) For all $j < m$, $f(k_1, \dots, k_n, j) \neq 0$.

- (6) By (2) and (5a), we have the first conjunct of (4a).

- (7) Either m is 0 or it is not. Suppose it 0. Then by ($\neq 0$), the antecedent of the second conjunct of (4a) is provably false for every z , and so the conditional is always true, and we can prove the conjunct in $\mathbf{Q}$.

- (8) Suppose instead that m is not 0. Then $\bar{m}$ is $m - 1'$.

- (9) By (2) and (5b), for all numbers $j < m$ we have a result of the form below, where r is some number other than 0.

$$\mathbf{Q} \vdash \forall y(\psi(\bar{k}_1, \dots, \bar{k}_n, \bar{j}, y) \rightarrow \bar{r} = y)$$

- (10) Instantiating y to 0, and using the fact that $\mathbf{Q} \vdash \bar{r} \neq 0$ whenever r is not 0 (Num=), by MT, we get a result of the form below for every $j < m$:

$$\mathbf{Q} \vdash \neg\psi(\bar{k}_1, \dots, \bar{k}_n, \bar{j}, 0).$$

- (11) In the object language, suppose $b < \overline{m-1}'$. By (Spread), we get:

$$b = 0 \vee b = \bar{1} \vee \dots \vee b = \overline{m-1} \quad (2.1)$$

- (12) By (10) and (11) and a big proof by cases, we get $\neg\psi(\bar{k}_1, \dots, \bar{k}_n, b, 0)$.
 (13) Discharging and generalizing, we have:

$$\mathbf{Q} \vdash \forall z(z < \overline{m-1}' \rightarrow \neg\psi(\bar{k}_1, \dots, \bar{k}_n, z, 0))$$

which by (8) is the second conjunct of (4a).

- (14) Thus whether m is or is not 0, (4a) is true.
 (15) To show (4b), assume in the object language for arbitrary b that:

$$\psi(\bar{k}_1, \dots, \bar{k}_n, b, 0) \wedge \forall z(z < b \rightarrow \neg\psi(\bar{k}_1, \dots, \bar{k}_n, z, 0))$$

- (16) By (Trichotomy), we have:

$$b < \overline{m} \vee \overline{m} < b \vee b = \overline{m}$$

- (17) But by (4a), $b < \overline{m} \rightarrow \neg\psi(\bar{k}_1, \dots, \bar{k}_n, b, 0)$, and so by MT with the first conjunct of (15), $\neg b < \overline{m}$.
 (18) By a similar argument using the first conjunct of (4a) and the second conjunct of (15), we get $\neg \overline{m} < b$.
 (19) By DS on (16), we get $b = \overline{m}$ and by Sym=, $\overline{m} = b$.
 (20) Discharging the assumption at (15), and applying UG, we get (4b). $\ast$

Gödel's β -function

Consider the function defined as follows:

$$\beta(x_1, x_2, x_3) = \text{rem}(1 + ((x_3 + 1) \times x_2), x_1)$$

Surprisingly, for any series of natural numbers with $n + 1$ members

$$k_0, k_1, k_2, \dots, k_n$$

one can find two fixed natural numbers b and c such that, for any $i \leq n$, $\beta(b, c, i) = k_i$.

To see this, first, let c be

$$(\max(n, k_0, k_1, k_2, \dots, k_n))!$$

Next, consider the following sequence:

$$u_0, u_1, u_2, \dots, u_n$$

where each $u_i = 1 + ((i + 1) \times c)$, for all $i \leq n$.

No two members of the u -series have a factor in common other than 1. (Tedious arithmetic.)

It follows from this and a known principle of modular arithmetic, the *Chinese remainder theorem*, that there is at least one number b such that the remainder upon division of b by u_i is always k_i for every i from 0 to n . (More tedious arithmetic.)

Therefore, because each u_i is $1 + ((i + 1) \times c)$, it follows that $\text{rem}(1 + ((i + 1) \times c), b) = k_i$, which is to say that $\beta(b, c, i) = k_i$.

Example: Suppose that our k -sequence is simply:

$$1, 2, 1$$

Then $n = 2$ and $\max(2, 1, 2, 1)$ is also 2, and finally, $c = 2!$, which is also 2.

Then, the u -series is:

$$3, 5, 7$$

(These numbers share no common factor.)

It follows by the Chinese remainder theorem that there at least one number b such that $\text{rem}(3, b) = 1$, and $\text{rem}(5, b) = 2$, and $\text{rem}(7, b) = 1$. (In this case, b could be 22, or 127, etc.)

For this sequence, for all $0 \leq i \leq 2$,

$$\beta(22, 2, i) = k_i$$

The upshot of all this is that it provides yet another method of talking indirectly about sequences of numbers. Each sequence corresponds to a b and c , and its elements can be “retrieved” by β .

Result: The function β is representable in $\mathbf{Q}$.

Sketch of proof: We use the wff, which we shall later abbreviate as $\varphi_\beta(x_1, x_2, x_3, y)$:

$$\exists z[x_1 = ((1 + ((x_3 + 1) \times x_2)) \times z) + y \wedge y < 1 + ((x_3 + 1) \times x_2)]$$

This says that y is the remainder when x_1 is divided by $1 + ((x_3) + x_2)$ because it says that when y is added to a multiple of $1 + ((x_3) + x_2)$ you get x_1 , and that y is less than that number, so the multiple in question must be the greatest possible less than or equal to it.

We shall not bother giving the full proof that this meets the conditions for representability, but we have already shown that all the functions and relations involved ($+$, $\times$, $=$, $<$) meet those conditions.

We use the power of the β function to represent sequences of numbers to capture recursive definitions in $\mathbf{Q}$ and $\mathbf{PA}$. Note that b and c are numbers representing a given sequence, then $\varphi_\beta(\bar{b}, \bar{c}, \bar{i}, \bar{k})$ says that k is the i^{th} member of this sequence, e.g.:

$$\begin{aligned}\mathbf{Q} &\vdash \varphi_\beta(\overline{22}, \overline{2}, \overline{0}, \overline{1}) \\ \mathbf{Q} &\vdash \varphi_\beta(\overline{22}, \overline{2}, \overline{1}, \overline{2}) \\ \mathbf{Q} &\vdash \varphi_\beta(\overline{22}, \overline{2}, \overline{2}, \overline{1})\end{aligned}$$

And a consequence of this, such as:

$$\exists x \exists y [\varphi_\beta(x, y, \overline{0}, \overline{1}) \wedge \varphi_\beta(x, y, \overline{1}, \overline{2}) \wedge \varphi_\beta(x, y, \overline{2}, \overline{1})]$$

is true just in case *there exists* a sequence whose first three elements are 1, 2 and 1.

Representing recursion in $\mathbf{Q}$

Result: The set of functions representable in $\mathbf{Q}$ is closed under primitive recursion, i.e., if f and g are representable in $\mathbf{Q}$, then so is the function h , where either:

$$\begin{aligned}h(x_0, \dots, x_{n-1}, 0) &= f(x_0, \dots, x_{n-1}), \text{ and} \\ h(x_0, \dots, x_{n-1}, y + 1) &= \\ &\quad g(x_0, \dots, x_{n-1}, y, h(x_0, \dots, x_{n-1}, y)), \\ &\quad \text{or} \\ h(0) &= k \\ h(y + 1) &= g(y, h(y)).\end{aligned}$$

Sketch of proof: A fuller proof is given in the text; we shall content ourselves with an rough explanation and an example, as the full proof is rather intricate.

Suppose h is definable recursively from f and g (or k and g) in the way indicated, and suppose that $h(k_1, \dots, k_n, j) = m$. There is then a sequence of values $v_0, \dots, v_j$ of h for the same first n arguments and all the final arguments leading up to and including j :

$$\begin{aligned}v_0 &= h(k_1, \dots, k_n, 0) &&= f(k_1, \dots, k_n) \text{ (or } k) \\ v_1 &= h(k_1, \dots, k_n, 1) &&= g(k_1, \dots, k_n, 0, v_0) \\ &\vdots &&\vdots \\ v_j &= h(k_1, \dots, k_n, j) &&= m = g(k_1, \dots, k_n, j - 1, v_{j-1})\end{aligned}$$

For example, suppose $h(x, y) = x^y$. In the case $h(2, 3) = 8$, we have:

$$\begin{aligned}v_0 &= h(2, 0) &&= 1 \\ v_1 &= h(2, 1) &&= v_0 \times 2 \\ v_2 &= h(2, 2) &&= v_1 \times 2 \\ v_3 &= h(2, 3) &&= 8 = v_2 \times 2\end{aligned}$$

We will use the object-language representation of Gödel’s β -function in effect to claim that there is a sequence

$$v_0, \dots, v_j$$

having the appropriate characteristics.

Suppose that f is represented in the object language by $\varphi_f(x_1, x_2, \dots, x_n, y)$ (or in the simpler

case, we use $y = \bar{k}$), and suppose g is represented by $\varphi_g(x_1, \dots, x_n, x_{n+1}, x_{n+2}, y)$. Then the wff $\varphi_h(x_1, \dots, x_n, x_{n+1}, y)$ we can use to represent h is:

$$\begin{aligned} \exists v \exists w \Big(& \exists y_2 [\varphi_\beta(v, w, 0, y_2) \wedge \varphi_f(x_1, \dots, x_n, y_2)] \wedge \\ & \varphi_\beta(v, w, x_{n+1}, y) \wedge \\ & \forall z [z < x_{n+1} \rightarrow \exists y_3 \exists y_4 [\varphi_\beta(v, w, z, y_3) \wedge \\ & \varphi_\beta(v, w, z', y_4) \wedge \varphi_g(x_1, \dots, x_n, z, y_3, y_4)]] \Big) \end{aligned}$$

Ugly! What on Earth does this say?!

- What we want it to say is that y is the value of the recursive function h for $x_1, \dots, x_n, x_{n+1}$ as arguments. Does it?
- Remember that the β function is used to talk indirectly about finite sequences. Because each finite sequence corresponds to a fixed b and c such that $\beta(b, c, i)$ is always the i^{th} member of the sequence, quantification over sequences can in effect be done by quantifying over two numbers.
- The existential quantification over v and w at the start of this wff in effect says “there is a sequence such that ...”.
- Given that $\varphi_\beta(\dots)$ represents the β function, and $\varphi_f(\dots)$ represents f , the first conjunct on the inside says that there is a y_2 at the start (0-spot) of the sequence, and it’s the value of f for $x_1, \dots, x_n$. This basically says how the sequence of values begins.
- Next, it says that y is at the x_{n+1} -spot of the sequence of values, which is to be expected if y is the value of h when h ’s last argument is x_{n+1} .
- Lastly, given that $\varphi_g(\dots)$ represents g , it says that for each previous spot in the sequence (each z -spot, where z is below the argument we’re currently considering), the member of the sequence at the next spot (y_4) is obtained from the member at the z -spot (y_3) in the appropriate way from the g function.

Let us consider the example of $x_1^{x_2}$.

Its definition uses $k = 1$ (this plays the role of f) and multiplication (this plays the role of g).

Making some minor simplifications, these are represented in S by the wffs $y = \bar{1}$ and $y = x_1 \times x_2$.

According to the above recipe, the function $x_1^{x_2}$

is represented by the following wff:

$$\begin{aligned} \exists v \exists w \Big(& \exists y_2 [\varphi_\beta(v, w, 0, y_2) \wedge y_2 = \bar{1}] \wedge \\ & \varphi_\beta(v, w, x_2, y) \wedge \\ & \forall z [z < x_2 \rightarrow \exists y_3 \exists y_4 [\varphi_\beta(v, w, z, y_3) \wedge \\ & \varphi_\beta(v, w, z', y_4) \wedge y_4 = y_3 \times x_1]] \Big) \end{aligned}$$

This says that there is a sequence of natural numbers with $x_2 + 1$ members, the first of which is 1, the last of which is y , and each one relates to the previous one by being its product when multiplied by x_1 . With some thought, it is clear that this is the case if and only if $y = x_1^{x_2}$.

Completing the full proof of the meta-theorem would involve establishing that if $h(k_1, \dots, k_n, k_{n+1}) = m$, then:

$$\begin{aligned} \mathbf{Q} \vdash & \varphi_h(\bar{k}_1, \dots, \bar{k}_n, \bar{k}_{n+1}, \bar{m}) \\ \mathbf{Q} \vdash & \forall y [\varphi_h(\bar{k}_1, \dots, \bar{k}_n, \bar{k}_{n+1}, y) \rightarrow \bar{m} = y] \end{aligned}$$

This is where the proof gets very involved for the general case.

However, consider the example that $1^3 = 1$. In that case, the v -sequence is $1^0, 1^1, 1^2, 1^3$, which is 1, 1, 1, 1. The β -function encoding for this sequence uses $b = 1$ and $c = 6$. We have:

$$\begin{aligned} \mathbf{Q} \vdash & \varphi_\beta(\bar{1}, \bar{6}, 0, \bar{1}) \\ \mathbf{Q} \vdash & \varphi_\beta(\bar{1}, \bar{6}, \bar{1}, \bar{1}) \\ \mathbf{Q} \vdash & \varphi_\beta(\bar{1}, \bar{6}, \bar{2}, \bar{1}) \\ \mathbf{Q} \vdash & \varphi_\beta(\bar{1}, \bar{6}, \bar{3}, \bar{1}) \\ \mathbf{Q} \vdash & \bar{1} = \bar{1} \\ \mathbf{Q} \vdash & \bar{1} = \bar{1} \times \bar{1} \end{aligned}$$

from the fact that $\varphi_\beta(x_1, x_2, x_3, y)$ represents β and other results we have established.

From the first and fifth, we get:

$$\mathbf{Q} \vdash \exists y_2 [\varphi_\beta(\bar{1}, \bar{6}, 0, y_2) \wedge y_2 = \bar{1}]$$

The pairs of first and second, second and third, and third and fourth, along with the last, give us:

$$\begin{aligned}
\mathbf{Q} &\vdash \exists y_3 \exists y_4 [\varphi_\beta(\bar{1}, \bar{6}, 0, y_3) \wedge \\
&\quad \varphi_\beta(\bar{1}, \bar{6}, 0', y_4) \wedge y_4 = y_3 \times \bar{1}] \\
\mathbf{Q} &\vdash \exists y_3 \exists y_4 [\varphi_\beta(\bar{1}, \bar{6}, \bar{1}, y_3) \wedge \\
&\quad \varphi_\beta(\bar{1}, \bar{6}, \bar{1}', y_4) \wedge y_4 = y_3 \times \bar{1}] \\
\mathbf{Q} &\vdash \exists y_3 \exists y_4 [\varphi_\beta(\bar{1}, \bar{6}, \bar{2}, y_3) \wedge \\
&\quad \varphi_\beta(\bar{1}, \bar{6}, \bar{2}', y_4) \wedge y_4 = y_3 \times \bar{1}]
\end{aligned}$$

The numerals $0, \bar{1}, \bar{2}$ exhaust the numerals for $z < 3$, and so, by (Spread) and a proof by cases:

$$\begin{aligned}
\forall z [z < \bar{3} \rightarrow \exists y_3 \exists y_4 [\varphi_\beta(\bar{1}, \bar{6}, z, y_3) \wedge \\
\quad \varphi_\beta(\bar{1}, \bar{6}, z', y_4) \wedge y_4 = y_3 \times \bar{1}]]
\end{aligned}$$

Putting some of these results together, and existentially generalizing from $\bar{1}$ and $\bar{6}$, we get:

$$\begin{aligned}
\mathbf{Q} &\vdash \exists v \exists w (\exists y_2 [\varphi_\beta(v, w, 0, y_2) \wedge y_2 = \bar{1}] \wedge \\
&\quad \varphi_\beta(v, w, \bar{3}, \bar{1}) \wedge \\
\forall z [z < \bar{3} \rightarrow \exists y_3 \exists y_4 [\varphi_\beta(v, w, z, y_3) \wedge \\
&\quad \varphi_\beta(v, w, z', y_4) \wedge y_4 = y_3 \times \bar{1}]])
\end{aligned}$$

Which is our way of saying that $1^3 = 1$. The pattern of this example generalizes to all our functions we could define by the primitive recursion method.

Result: *All (primitive and general) recursive functions are representable in $\mathbf{Q}$ and $\mathbf{PA}$.*

Proof: By results achieved so far, we know that the set of functions representable in $\mathbf{Q}$ contains the initial functions (the zero function, the successor function, and projection functions) and is closed under composition, primitive recursion and minimization. Since the set of (general) recursive functions is the smallest such set, all its members are contained within. This includes the primitive recursive functions, which are a subset of the general recursive functions. Moreover, all those representable in $\mathbf{Q}$ are also representable in $\mathbf{PA}$. *

Corollary: *All (primitive and general) recursive relations are representable in $\mathbf{Q}$ and $\mathbf{PA}$.*

Proof: Suppose R is a recursive relation. This means that its characteristic function χ_R is a recursive function. By the result above, χ_R can be represented in $\mathbf{Q}$. By what you proved in homework 25, it follows that R can be represented in $\mathbf{Q}$ (and thus $\mathbf{PA}$). *

We shall later prove that the converse is also true: all functions/relations representable in $\mathbf{Q}$ (and even $\mathbf{PA}$) are recursive.

HOMEWORK 26

Show that if f is a recursive function, then there is a wff $\varphi(x_1, \dots, x_n, y)$ in the language $\mathcal{L}_A$ such that, for all natural numbers $k_1, \dots, k_n$ and m , it holds that $\mathfrak{N} \models \varphi(\bar{k}_1, \dots, \bar{k}_n, \bar{m})$ iff $f(k_1, \dots, k_n) = m$. (Hint: appeal to the fact that $\mathbf{Q} \subseteq \mathbf{TA}$.)

8. Arithmetization of Syntax

System $\odot$

Thought of as broadly as possible, a *syntax* is just a series of rules for what counts as an “allowed” string of basic symbols, and a “deduction system” just a set of rules for which allowed strings, and manipulations between strings, are “favored” or “endorsed”. This would even include:

Definition: The *basic symbols* of $\odot$ are “ $\square$ ” and “ $\circ$ ”.

Definition: An *allowed string* of $\odot$ is any finite string of the basic symbols that begins with “ $\square$ ”.

Examples: “ $\square$ ”, “ $\square\square$ ”, “ $\square\square\square$ ”, are allowed strings, but “ $\circ\square\square\square$ ” is not.

Definition: An *allowed string* follows by another allowed string by the rule **add-circle** iff it is just like the other string, except with “ $\circ$ ” appended at the end.

Example: “□○○” follows by add-circle from “□○”.

Definition: The set of *avored strings* of $\odot$ is the smallest set containing “□” and closed under add-circle.

We can think of $\odot$ as a “deduction system” whose “theorems” or “avored strings” include exactly:

□
 □○
 □○○
 □○○○
 □○○○○, etc.

We could do “metatheory” for $\odot$, and it would be pretty easy to do. In fact, its metatheory can be reduced to simple mathematics if we *arithmetize its syntax*, or convert our talk of strings to talk of numbers.

Definition: The *Schmödel number* of an allowed string of $\odot$ is the number which is represented in binary notation when “□” is interpreted as the digit “1” and “○” is interpreted as “0”.

Examples: The Schmödel number of “□○○” is 4 (100 in binary), and the Schmödel number of “□□○○□” is 26 (11010 in binary).

Result: An allowed string of $\odot$ is favored iff its Schmödel number is a power of 2.

Proof: Each favored string is derived from □ (Schmödel number 1) by adding some number of circles at the end, which is the same as multiplying its Schmödel number by 2 (just like adding a 0 at the end in decimal notation is the same as multiplying by 10). *

Indeed, the “arithmetization” of system $\odot$ means that we can do its metatheory in systems like **Q** and **PA**. Arguably:

$$\overline{32} = \overline{1} \vee \forall x [\exists y \ x \times y = \overline{32} \rightarrow (x = 1 \vee \exists y \ 2 \times y = x)]$$

says that □○○○○ is favored, since its Schmödel number (32) has the property of being a power of 2 iff it is either 1, or is only divisible by 1 and even numbers.

Of course, by means of Gödel numbering, systems such as **Q** and **PA** can even, at least, partially, act as *their own metatheories*, and by so doing, even “say” things about themselves in a sense.

Gödel numbering

We begin by assigning a different number to every simple sign in the language of first-order logic; see page 25 for details.

Definition: The *Gödel number of a string* of symbols of the language of first-order logic is the number that encodes (using the first however-many primes raised to one more than each number) the series of the numbers for the simple symbols in order.

Example: The Gödel number of string “A¹(a)” is $2^{(2^5 \cdot 3^2 \cdot 5^1 \cdot 7^2)+1} \cdot 3^{(2^1 \cdot 3^{12})+1} \cdot 5^{(2^3 \cdot 3^1 \cdot 5^2)+1} \cdot 7^{(2^1 \cdot 3^{13})+1}$, since the number of simple symbol A¹ is $2^5 \cdot 3^2 \cdot 5^1 \cdot 7^2$, the number of “(” is “ $2^1 \cdot 3^{12}$ ”, and so on.

Warning: Be sure to differentiate the number for the simple symbol, “(”, which is $2^1 \cdot 3^{12}$, from the Gödel number of the single-character-long string “(”, which is $2^{(2^1 \cdot 3^{12})+1}$.

We shall use the notation:

$$\# \varphi \#$$

in the metalanguage for the Gödel number of string φ , so that, e.g.:

$$\# A^1(a) \# = 2^{(2^5 \cdot 3^2 \cdot 5^1 \cdot 7^2)+1} \cdot 3^{(2^1 \cdot 3^{12})+1} \cdot 5^{(2^3 \cdot 3^1 \cdot 5^2)+1} \cdot 7^{(2^1 \cdot 3^{13})+1}$$

Since whatever string φ is, $\# \varphi \#$ is just a particular constant number, we can make use of this notation in recursive definitions.

Constructed expressions such as (complex) terms and formulas are just kinds of strings, and so we may use this notation for their Gödel numbers.

We can extend the notion as such:

Definition: The *Gödel number of a sequence* of strings $\varphi_0, \dots, \varphi_{n-1}$ is the number that encodes the sequence of Gödel numbers of the strings in the series, in order.

Example: Our first example derivation:

$Fa \rightarrow [(Fa \rightarrow Fa) \rightarrow Fa]$
 $\{Fa \rightarrow [(Fa \rightarrow Fa) \rightarrow Fa]\}$
 $\rightarrow \{[Fa \rightarrow (Fa \rightarrow Fa)] \rightarrow (Fa \rightarrow Fa)\}$
 $[Fa \rightarrow (Fa \rightarrow Fa)] \rightarrow (Fa \rightarrow Fa)$
 $Fa \rightarrow (Fa \rightarrow Fa)$
 $Fa \rightarrow Fa$

Is a sequence of strings having the Gödel number:

$$\begin{aligned}
 &2^{\#Fa \rightarrow [(Fa \rightarrow Fa) \rightarrow Fa]^{\#} + 1} \times \\
 &3^{\#\{Fa \rightarrow [(Fa \rightarrow Fa) \rightarrow Fa]\} \rightarrow \{[Fa \rightarrow (Fa \rightarrow Fa)] \rightarrow (Fa \rightarrow Fa)\}^{\#} + 1} \times \\
 &5^{\#[Fa \rightarrow (Fa \rightarrow Fa)] \rightarrow (Fa \rightarrow Fa)^{\#} + 1} \times \\
 &7^{\#Fa \rightarrow (Fa \rightarrow Fa)^{\#} + 1} \times \\
 &11^{\#Fa \rightarrow Fa^{\#} + 1}
 \end{aligned}$$

A single finite number, albeit an enormous one.

Warning: Just as we must differentiate between the number of a simple symbol and the Gödel number of a single-symbol string of just it, we must differentiate between the Gödel number of a string from that of a single-string-long sequence of strings containing just it, i.e., the difference between $\#Fa \rightarrow [(Fa \rightarrow Fa) \rightarrow Fa]^{\#}$ and $2^{\#Fa \rightarrow [(Fa \rightarrow Fa) \rightarrow Fa]^{\#} + 1}$.

Recursive syntax

Result: The following functions, properties and relations relating to the syntax of first-order logic are primitive recursive.

We sometimes bother giving the formal definitions showing them to be primitive recursive, sometimes won't bother, as it gets tedious. More details can be found in the book.

(a) x is the Gödel number of a string of just a variable:

$$\text{Var}(x) \text{ iff } \exists_{y < x}^y \exists_{1 \leq z \leq 6}^z x = \text{code}(\text{code}(1, y, z))$$

(b) x is the Gödel number of a string of just a constant:

$$\text{Const}(x) \text{ iff } \exists_{y < x}^y \exists_{1 \leq z \leq 5}^z x = \text{code}(\text{code}(2, y, z))$$

Or for $\mathcal{L}_A$:

$$\text{Const}_{\mathcal{L}_A}(x) \text{ iff } x = \#0^{\#}$$

(c) x is the Gödel number a y -place function letter

$$\text{FL}(x, y) \text{ iff } \exists_{z < x}^z \exists_{1 \leq w \leq 7}^w x = \text{code}(\text{code}(3, y, z, w))$$

$$\text{FL}_{\mathcal{L}_A}(x, y) \text{ iff } (y = 1 \text{ and } x = \#/'^{\#}) \text{ or } (y = 2 \text{ and } (x = \#+^{\#} \text{ or } x = \#_x^{\#}))$$

(d) x is the Gödel number of a y -place predicate:

$$\text{Pred}(x, y) \text{ iff } (y = 2 \text{ and } x = \#_=\#) \text{ or } \exists_{z < x}^z \exists_{1 \leq w \leq 10}^w x = \text{code}(\text{code}(4, y, z, w))$$

$$\text{Pred}_{\mathcal{L}_A}(x, y) \text{ iff } y = 2 \text{ and } (x = \#_=\# \text{ or } x = \#_<^{\#})$$

(e) $\text{flatten}(x)$ = the Gödel number for the string obtained by concatenating each item in the sequence of strings having Gödel number x with $\#,\#$ in between each item.

(f) $\text{sequenceBound}(x, y)$ = the greatest possible number a code for a sequence of length y whose elements are no greater than x could be

(g) x is the Gödel number of a term:

$\text{Term}(x)$ iff $\text{Var}(x)$ or $\text{Const}(x)$ or

$$\exists_{y < x}^y \exists_{z < x}^z \exists_{w < \text{sequenceBound}(x, z)}^w [\text{FL}(y, z) \text{ and}$$

$$x = y \frown \#(\# \frown \text{flatten}(w) \frown \#)^{\#} \text{ and } \text{len}(w) = z \text{ and}$$

$$\bigvee_{0 \leq v < z}^v \bigvee_{x_1 < x}^{x_1} (\text{if } x_1 = (w)_v \text{ then } \text{Term}(x_1))]$$

(This appears to define $\text{Term}(x)$ in terms of itself, but notice x_1 is bound below x , and so this is allowed by course of values recursion.) We can define $\text{Term}_{\mathcal{L}_A}$ similarly, with $\text{Const}_{\mathcal{L}_A}$ and $\text{FL}_{\mathcal{L}_A}$, etc.

(h) $\text{num}(x)$: the Gödel number of the numeral for x , defined recursively:

$$\text{num}(0) = \#0^{\#}$$

$$\text{num}(x + 1) = \#f^1(\# \frown \text{num}(x) \frown \#)^{\#}$$

(Recall that t' is shorthand for $f^1(t)$.)

(i) x is the Gödel number of an atomic formula.

$\text{Atom}(x)$ iff $x = \#_ \perp^{\#}$ or $x = \#_ \top^{\#}$ or

$$\exists_{y < x}^y \exists_{z < x}^z \exists_{w < \text{sequenceBound}(x, z)}^w [\text{Pred}(y, z) \text{ and}$$

$x = y \frown \#(\# \frown \text{flatten}(w) \frown \#)^{\#}$ and
 $\text{len}(w) = z$ and
 $\bigvee_{0 \leq v < z} \bigvee_{x_1 < x} (\text{if } x_1 = (w)_v \text{ then Term}(x_1))$
 (Atom_{L_A}(x) similarly.)

- (j) x is the the Gödel number of a well-formed formula:

Frm(x) iff Atom(x) or

$$\begin{aligned}
 & \bigvee_{y < x} (\text{Frm}(y) \text{ and } x = \# \neg^{\#} \frown y) \text{ or} \\
 & \bigvee_{y < x} \bigvee_{z < x} \bigvee_{w < x} \left(\text{Frm}(y) \text{ and Frm}(z) \text{ and} \right. \\
 & \quad (w = \# \rightarrow^{\#} \text{ or } w = \# \leftrightarrow^{\#} \text{ or} \\
 & \quad w = \# \vee^{\#} \text{ or } w = \# \wedge^{\#}) \text{ and} \\
 & \quad x = \#(\# \frown y \frown w \frown z \frown \#)^{\#} \text{ or} \\
 & \bigvee_{y < x} \bigvee_{z < x} \bigvee_{w < x} \left(\text{Frm}(y) \text{ and Var}(z) \text{ and} \right. \\
 & \quad (w = \# \forall^{\#} \text{ or } w = \# \exists^{\#}) \text{ and} \\
 & \quad x = \# w^{\#} \frown \# z^{\#} \frown \# y^{\#} \left. \right)
 \end{aligned}$$

We can get Frm_{L_A} similarly.

- (k) FreeOcc(x, y, z): the formula with Gödel number x contains a free occurrence of the variable with Gödel number y in position z .
- (l) Sent(x): x is the Gödel number of a sentence (i.e., formula without free variables)
 (And Sent_{L_A}(x) similarly from Frm_{L_A}.)
- (m) Subst(x, y, z): the Gödel number of the formula resulting by replacing all free occurrences of the variable with Gödel number z with the term with Gödel number y .
- (n) UnivClosure(x): the Gödel number of the formula resulting from prepending a universal quantifier of the form $\forall y$ to the formula with Gödel number x for every variable it contains free.
- (o) Neg(x): the Gödel number of the negation of the formula with Gödel number x
- (p) Cond(x, y): the Gödel number of the conditional whose antecedent is the conjunction of all wffs whose Gödel numbers are in the sequence encoded by x and whose consequent is the wff whose Gödel number is y .
- (q) The (evil) diagonalization function: the Gödel number of the result of replacing the numeral for y for all free occurrences of the (object-

language) variable x in the formula with Gödel number y :
 $\text{diag}(y) = \text{Subst}(y, \text{num}(y), \#x^{\#})$

Derivations and recursion

Result: *The following properties and relations are recursive.*

- (a) MP(x, y): x is the Gödel number of a sequence where the y^{th} element follows by *modus ponens* from previous elements of the sequence.

$$\text{MP}(x, y) \text{ iff } \bigvee_{z < y} \bigvee_{w < y} \left((x)_z = \#(\# \frown (x)_w \frown \# \rightarrow^{\#} \frown (x)_y \frown \#)^{\#} \right)$$

- (b) QR(x, y): x is the Gödel number of a sequence where the y^{th} element follows by the *quantifier rules* (QR) from a previous element.

- (c) x is an instance of the axiom schema (SimplAx):

SimplAx(x) iff

$$\bigvee_{y < x} \bigvee_{z < x} (\text{Sent}(y) \text{ and Sent}(z) \text{ and } x = \#((\# \frown y \frown \# \wedge^{\#} \frown z \frown \#) \rightarrow^{\#} \frown y \frown \#)^{\#})$$

(And so on for all the other logical axiom schemata.)

- (d) IsAx(x): x is the Gödel number of an instance of any of the axiom schemata of first-order logic with identity (including, e.g., Ref=A_x, LLAx).

- (e) Deriv(x): x is the Gödel number of a sequence constituting a purely logical derivation (i.e., one without premises or non-logical axioms):

$$\text{Deriv}(x) \text{ iff } \bigvee_{y < \text{len}(x)} (\text{IsAx}((x)_y) \text{ or}$$

$$\text{MP}(x, y) \text{ or QR}(x, y))$$

Definition: Γ is a **(primitive) recursive set of sentences** iff the property of being the Gödel number of a member of Γ is (primitive) recursive.

Definition: A theory is **recursively axiomatizable** (or sometimes, simply, **axiomatizable**) iff it

is the closure of a recursive set (or equivalently, axiomatized by a recursive set).

Result: $\mathbf{Q}$ is recursively axiomatizable.

Proof: $\mathbf{Q}$ is the closure of the set of axioms $\mathbf{Q}_1$ – $\mathbf{Q}_8$, and the property of being the Gödel number of one of these is simply the disjunction of being each one of them, which is primitive recursive:

$$\begin{aligned} \text{IsQA}_x(x) \text{ iff} \\ x = \# \forall x \forall y (x' = y' \rightarrow x = y)^\# \text{ or } \dots \\ \text{or } x = \# \forall y (x < y \leftrightarrow \exists z z' + x = y)^\# \end{aligned} \quad *$$

Result: $\mathbf{PA}$ is recursively axiomatizable.

Proof: $\mathbf{PA}$ is the closure of the set containing $\mathbf{Q}_1$ – $\mathbf{Q}_8$ and every instance of $\mathbf{PA}_9$. Hence x is the Gödel number of a member of this set iff $\text{IsQA}_x(x)$ or:

$$\begin{aligned} \exists_{y \leq x}^y \exists_{z < y}^z \exists_{v < x}^v (\text{Frm}(z) \text{ and } \text{Var}(v) \text{ and} \\ y = \#((\# \frown \text{Subst}(z, \#0^\#, v) \frown \# \wedge \forall^\# \frown v \frown \\ \#(\# \frown z \frown \# \rightarrow \# \frown \text{Subst}(z, \#f^1(\# \frown v \frown \#)^\#, v) \\ \frown \#)) \rightarrow \forall^\# \frown v \frown z \frown \#)^\# \text{ and} \\ x = \text{UnivClosure}(y)) \end{aligned}$$

and so is primitive recursive. *

Result: If Γ is a recursive set of sentences, so is any set $\Gamma \cup \{\varphi_1, \dots, \varphi_n\}$ obtained by adding a finite number of sentences to Γ .

HOMEWORK 27

Prove the result above.

Result: If Γ is a (primitive) recursive set of sentences, then the relation $\text{Prf}_\Gamma(x, y)$ that holds between x and y when x is the Gödel number of purely logical derivation of a formula of the form $(\psi_1 \wedge \dots \wedge \psi_n) \rightarrow \varphi$, where each of $\psi_i \in \Gamma$ and y is the Gödel number of φ , is also (primitive) recursive.

Proof: Let R_Γ be the (primitive) recursive property a number has iff it is the Gödel number of a member of Γ . We can define Prf_Γ as follows:

$$\text{Prf}_\Gamma(x, y) \text{ iff } \text{Deriv}(x) \text{ and } \exists_{z < \text{sequenceBound}(x, x)}^z \left[(x)_{\text{len}(x)-1} = \text{Cond}(z, y) \text{ and } \bigwedge_{w < \text{len}(z)}^w R_\Gamma((z)_w) \right]$$

This says that x is the Gödel number of a purely logical derivation, the last line of which is a conditional with an antecedent that is a conjunction of formulas all of which come from Γ , and the consequent of which has Gödel number y . *

Corollary: $\text{Prf}_\mathbf{Q}$ and $\text{Prf}_{\mathbf{PA}}$ are primitive recursive relations.

Result: $\text{Prf}_\Gamma(x, \# \varphi^\#)$ holds for some x iff $\Gamma \vdash \varphi$.

Proof: Follows directly from the definition of Prf_Γ and the Logical Conditionalization corollary of Compactness listed on page 31. *

Representable functions are recursive

Result: Every function representable in $\mathbf{Q}$ (or $\mathbf{PA}$, or any other recursively axiomatizable theory of which $\text{Num} =$ holds) is recursive.

Proof: (1) Let f be a function representable in $\mathbf{Q}$. By definition, there is a wff $\varphi_f(x_1, \dots, x_n, y)$ such that if $f(k_1, \dots, k_n) = m$, then:

$$\begin{aligned}\mathbf{Q} &\vdash \varphi_f(\overline{k_1}, \dots, \overline{k_n}, \overline{m}) \\ \mathbf{Q} &\vdash \forall y (\varphi_f(\overline{k_1}, \dots, \overline{k_n}, y) \rightarrow \overline{m} = y)\end{aligned}$$

(2) Let A be the $(n + 1)$ -place function which yields as value the Gödel number of the formula that results by replacing the free variables $x_1, \dots, x_n, y$ in $\varphi_f(x_1, \dots, x_n, y)$ by the numerals for its arguments, i.e.:

$$A(k_1, \dots, k_n, k_{n+1}) = \# \varphi_f(\overline{k_1}, \dots, \overline{k_n}, \overline{k_{n+1}})^\#$$

(3) A is primitive recursive, as it can be defined as follows:

$$\begin{aligned}A(k_1, \dots, k_n, k_{n+1}) &= \text{Subst}(\text{Subst}(\dots \\ &(\text{Subst}(\# \varphi_f(x_1, \dots, x_n, y)^\#, \text{num}(k_1), \# x_1^\#), \dots), \\ &\text{num}(k_n), \# x_n^\#), \text{num}(k_{n+1}), \# y^\#)\end{aligned}$$

(4) Now consider the relation R such that:
 $R(k_1, \dots, k_n, s)$ is true iff s encodes a pair of numbers j_1 and j_2 , where $\text{Prf}_{\mathbf{Q}}(j_1, \# \varphi_f(\overline{k_1}, \dots, \overline{k_n}, \overline{j_2})^\#)$, or roughly, where j_1 is the Gödel number of an object language proof that $f(k_1, \dots, k_n) = j_2$.

(5) R is primitive recursive, as:

$$\begin{aligned}R(k_1, \dots, k_n, s) &\text{ iff } \exists_{z_1 < s} \exists_{z_2 < s} [s = \text{code}(z_1, z_2) \text{ and} \\ &\text{Prf}_{\mathbf{Q}}(z_1, A(k_1, \dots, k_n, z_2))]\end{aligned}$$

(6) We can now give a recursive definition of function g so:

$$g(k_1, \dots, k_n) = ((\mu s)(k_1, \dots, k_n, s))_1$$

(7) g is none other than f .

- To see this, suppose $f(k_1, \dots, k_n) = m$ and $g(k_1, \dots, k_n) = j$. This means that j is some number for which there is a proof establishing:

$$\mathbf{Q} \vdash \varphi_f(\overline{k_1}, \dots, \overline{k_n}, \overline{j})$$

- By this and (1), we get $\mathbf{Q} \vdash \overline{m} = \overline{j}$, and so, by Num=, m and j are in fact the same number.

- The same holds for any other arguments to f and g , and so f and g are the same function.

(8) Hence f can be recursively defined with the definition given for g . *

Corollary: *In any consistent, recursively axiomatizable extension of $\mathbf{Q}$, every representable function is recursive.*

HOMEWORK 28

Prove the corollary above, with the crucial piece being that any extension of $\mathbf{Q}$ in which Num= did not hold would be inconsistent. (Num= is stated on page 42.)

Corollary: *The set of functions representable in $\mathbf{Q}$ (or $\mathbf{PA}$) is precisely the set of (general) recursive functions.*

Corollary: *The set of relations representable in $\mathbf{Q}$ (or $\mathbf{PA}$) is precisely the set of (general) recursive relations.*

9. Diagonalization

Preliminaries

Some notation:

$\ulcorner \varphi \urcorner$ is shorthand for $\# \varphi^\#$.

I.e., $\ulcorner \varphi \urcorner$ is the object language numeral for the Gödel number of φ .

Example: $\ulcorner A^1(a) \urcorner$ is the string consisting of $2^{(2^5 \cdot 3^2 \cdot 5^1 \cdot 7^2)+1} \cdot 3^{(2^1 \cdot 3^{12})+1} \cdot 5^{(2^3 \cdot 3^1 \cdot 5^2)+1} \cdot 7^{(2^1 \cdot 3^{13})+1}$ successor function signs applied to “0”. Don’t try writing that out in full.

For our next result, consider an (infinitely large) chart with each row representing an open formula with one free variable, x , and each column representing the Gödel number of an open formula, arranged in the same order.

Where they meet, we put the sentence in which the open formula of the row is asserted of the numeral for the number for the column.

Suppose $\varphi_1(x)$ means “ x is even”. Then as we go across the first row, each value is the formula claiming that the number at the top is even.

The numbers along the “diagonal” of the chart are special. These are underlined. The first one is the Gödel number of the claim that the Gödel number of “ x is even” is even. The “evil” *diagonalization function*, diag , introduced on p. 62, is the function from the numbers at the top to the number of the underlined formula in their column.

But diag is a recursive function, and is itself representable, by $\theta_{\text{diag}}(x, y)$, say.

For each open formula $\varphi_i(x)$, there is, somewhere else on the list, the open sentence asserting that every value of diagonalization applied to x satisfies $\varphi_i(x)$, i.e., $\forall y(\theta_{\text{diag}}(x, y) \rightarrow \varphi_i(y))$. A formula in its row claims that the underlined number in *its own column* satisfies $\varphi_i(x)$, e.g., in the first cell there is the (true!) claim that $\ulcorner \varphi_1(\ulcorner \varphi_1(x) \urcorner) \urcorner^\#$ is even.

Since this second open formula has its own row, there will eventually come some column where the diagonal reaches this row. The sentence in that square asserts that the Gödel number of the sentence in the underlined square in that column is even: i.e., that *its own Gödel number* is even!

This special square is called a “fixed point”. Fixed points make self-reference possible, and notice that there will be one not only for $\varphi_1(x)$, but for any other open formula.

...	...	...	...	...
$\varphi_1(x)$:	$\varphi_1(\ulcorner \varphi_1(x) \urcorner)$	$\varphi_1(\ulcorner \varphi_2(x) \urcorner)$	$\varphi_1(\ulcorner \varphi_3(x) \urcorner)$	$\ulcorner \forall y(\theta_{\text{diag}}(x, y) \rightarrow \varphi_1(y)) \urcorner^\#$
$\varphi_2(x)$:	$\varphi_2(\ulcorner \varphi_1(x) \urcorner)$	$\varphi_2(\ulcorner \varphi_2(x) \urcorner)$	$\varphi_2(\ulcorner \varphi_3(x) \urcorner)$	$\varphi_2(\ulcorner \forall y(\theta_{\text{diag}}(x, y) \rightarrow \varphi_2(y)) \urcorner)$
$\varphi_3(x)$:	$\varphi_3(\ulcorner \varphi_1(x) \urcorner)$	$\varphi_3(\ulcorner \varphi_2(x) \urcorner)$	$\varphi_3(\ulcorner \varphi_3(x) \urcorner)$	$\varphi_3(\ulcorner \forall y(\theta_{\text{diag}}(x, y) \rightarrow \varphi_3(y)) \urcorner)$
...	...	...	...	...

The fixed point theorem

Result (fixed point theorem): For any theory $\mathbf{T}$ in which all recursive functions are representable, it holds that for every formula $\varphi(x)$ with one free variable, there is a wff ψ such that:

$$\mathbf{T} \vdash \psi \leftrightarrow \varphi(\ulcorner \psi \urcorner)$$

Proof: (1) Suppose $\mathbf{T}$ is such a theory, and $\varphi(x)$ an arbitrary open formula.

(2) $\mathbf{T}$ represents the recursive function diag ; hence there is a wff $\theta_{\text{diag}}(x, y)$, where if $\text{diag}(k) = m$, then:

$$\mathbf{T} \vdash \theta_{\text{diag}}(\bar{k}, \bar{m})$$

$$\mathbf{T} \vdash \forall y(\theta_{\text{diag}}(\bar{k}, y) \rightarrow \bar{m} = y)$$

(3) Consider the wff $\forall y(\theta_{\text{diag}}(x, y) \rightarrow \varphi(y))$, and abbreviate it as $\alpha(x)$.

(4) Let ψ be the sentence $\alpha(\ulcorner \alpha(x) \urcorner)$. (Note that this does not contain x free, as $\ulcorner \alpha(x) \urcorner$ is just 0 followed by many “”s.)

(5) Given the way that ψ is constructed:

$$\text{diag}(\ulcorner \alpha(x) \urcorner) = \ulcorner \psi \urcorner.$$

(6) By (2) and (5) we have:

$$\mathbf{T} \vdash \theta_{\text{diag}}(\ulcorner \alpha(x) \urcorner, \ulcorner \psi \urcorner) \quad (6a)$$

$$\mathbf{T} \vdash \forall y(\theta_{\text{diag}}(\ulcorner \alpha(x) \urcorner, y) \rightarrow \ulcorner \psi \urcorner = y) \quad (6b)$$

(7) First we show $\mathbf{T} \vdash \psi \rightarrow \varphi(\ulcorner \psi \urcorner)$.

1. ψ Hyp
2. $\alpha(\ulcorner \alpha(x) \urcorner)$ 1 Def. ψ
3. $\forall y(\theta_{\text{diag}}(\ulcorner \alpha(x) \urcorner, y) \rightarrow \varphi(y))$ 2 Def. $\alpha(x)$
4. $\theta_{\text{diag}}(\ulcorner \alpha(x) \urcorner, \ulcorner \psi \urcorner) \rightarrow \varphi(\ulcorner \psi \urcorner)$ 3 UI
5. $\varphi(\ulcorner \psi \urcorner)$ 4, (6a) above, MP
6. $\psi \rightarrow \varphi(\ulcorner \psi \urcorner)$ 1–5 DT

(8) Next we show $\mathbf{T} \vdash \varphi(\ulcorner \psi \urcorner) \rightarrow \psi$:

1. $\varphi(\ulcorner \psi \urcorner)$ Hyp
2. $\theta_{\text{diag}}(\ulcorner \alpha(x) \urcorner, b)$ Hyp
3. $\theta_{\text{diag}}(\ulcorner \alpha(x) \urcorner, b) \rightarrow \ulcorner \psi \urcorner = b$ (6b), UI
4. $\ulcorner \psi \urcorner = b$ 2, 4 MP

5. $\varphi(b)$ 1, 4 LL
6. $\theta_{\text{diag}}(\ulcorner \alpha(x) \urcorner, b) \rightarrow \varphi(b)$ 2–5 DT
7. $\forall y(\theta_{\text{diag}}(\ulcorner \alpha(x) \urcorner, y) \rightarrow \varphi(y))$ 6 UG
8. $\alpha(\ulcorner \alpha(x) \urcorner)$ 7 Def. $\alpha(x)$
9. ψ 8 Def. ψ
10. $\varphi(\ulcorner \psi \urcorner) \rightarrow \psi$ 1–9 DT

(9) By (7) and (8) and BI, $\mathbf{T} \vdash \psi \leftrightarrow \varphi(\psi)$.

✱

Corollary: The fixed point theorem applies to $\mathbf{Q}$, $\mathbf{PA}$, $\mathbf{TA}$, and any other theory extending $\mathbf{Q}$.

Proof: All recursive functions are represented in $\mathbf{Q}$, and therefore would also be in any theory containing it, $\mathbf{PA}$ and $\mathbf{TA}$ included. ✱

10. Gödel’s Results and Corollaries

Expressing Prf in the object language

Result (Prf $\vdash$): In any theory $\mathbf{T}$, which is a recursively axiomatizable extension of $\mathbf{Q}$, and Γ is the (or a) recursive set of which $\mathbf{T}$ is the closure, there is a formula $\text{Prf}_{\Gamma}(x, y)$ such that, if $\text{Prf}_{\Gamma}(k, m)$, then:

$$\mathbf{Q} \text{ (and } \mathbf{T}) \vdash \text{Prf}_{\Gamma}(\bar{k}, \bar{m})$$

And if not- $\text{Prf}_{\Gamma}(k, m)$

$$\mathbf{Q} \text{ (and } \mathbf{T}) \vdash \neg \text{Prf}_{\Gamma}(\bar{k}, \bar{m})$$

(Here we use sans serif Prf to differentiate the object-language expression.)

Proof: Since Γ is a recursive set, Prf_{Γ} is a recursive relation. Since $\mathbf{Q}$ (and also $\mathbf{T}$ itself) represents all recursive relations, there must be some formula meeting the conditions above. ✱

Where $\mathbf{T}$ is a recursively axiomatized theory, we shall conventionally use $\text{Prf}_{\mathbf{T}}$ interchangeably with some one of the formulas Prf_{Γ} , where Γ is recursive and axiomatizes $\mathbf{T}$. E.g., $\text{Prf}_{\mathbf{Q}}$ (bold $\mathbf{Q}$) for $\text{Prf}_{\mathbf{Q}}$ (regular $\mathbf{Q}$) and $\text{Prf}_{\mathbf{PA}}$ for $\text{Prf}_{\mathbf{PA}}$. Strictly speaking, Prf_{Γ} and Prf_{Δ} are different relations even when both Γ and Δ axiomatize the same theory $\mathbf{T}$, and so Prf_{Γ} and Prf_{Δ} are different wffs, but in general it is unimportant which we pick.

Let us also use the notation:

$\text{Prov}_{\mathbf{T}}(x)$ abbreviates $\exists y \text{Prf}_{\mathbf{T}}(y, x)$

We can think of this as a “provability” predicate: there exists some proof of the formula with Gödel number x in (one of the axiomatizations of) $\mathbf{T}$.

ω -consistency

Definition: A set of formulas Γ is ω -consistent iff for every formula $\varphi(y)$ with one free variable, if $\Gamma \vdash \neg\varphi(\bar{n})$ for every number n , then $\Gamma \nvdash \exists y\varphi(y)$.

Result: If a theory $\mathbf{T}$ is ω -consistent, then it is consistent.

Proof: Suppose $\mathbf{T}$ is ω -consistent but suppose for *reductio* that it is inconsistent. Then every sentence can be proven in $\mathbf{T}$, and so for every open sentence it would be true that $\mathbf{T} \vdash \neg\varphi(\bar{n})$ for every number n , and also that $\mathbf{T} \vdash \exists y\varphi(y)$, contradicting its ω -consistency. *

Result: If $\mathbf{T} \subseteq \mathbf{TA}$, then $\mathbf{T}$ is ω -consistent.

Proof: Suppose that $\mathbf{T} \subseteq \mathbf{TA}$ but suppose for *reductio* that $\mathbf{T}$ is not ω -consistent. There is then some formula such that $\mathbf{T} \vdash \neg\varphi(\bar{n})$ for every number n , and also that $\mathbf{T} \vdash \exists y\varphi(y)$, and therefore that $\mathbf{TA} \vdash \neg\varphi(\bar{n})$ for every number n , and also that $\mathbf{TA} \vdash \exists y\varphi(y)$. From the latter and simple logic,

$\mathbf{TA} \vdash \neg\forall y\neg\varphi(y)$. But then for every n , $\mathfrak{N} \models \neg\varphi(\bar{n})$. Since the numerals exhaust the domain of quantification for $\mathfrak{N}$, it would hold that $\mathfrak{N} \models \forall y\neg\varphi(y)$, and thus $\mathbf{TA} \vdash \forall y\neg\varphi(y)$. But this would make $\mathbf{TA}$ inconsistent, which we know not to be true. *

Corollary: $\mathbf{Q}$, $\mathbf{PA}$, and $\mathbf{TA}$ are ω -consistent.

Two senses of completeness

On page 25, we gave this definition:

Definition: A set of sentences Γ is **complete** iff for every closed wff φ , either $\varphi \in \Gamma$ or $\neg\varphi \in \Gamma$.

We can apply this to theories, in which case...

Definition: A closed wff is an **undecidable sentence** of $\mathbf{T}$ iff both $\mathbf{T} \nvdash \varphi$ and $\mathbf{T} \nvdash \neg\varphi$.

Result: If $\mathbf{T}$ is a theory, then it is complete iff it has no undecidable sentences.

Proof: By the definition of a theory, $\mathbf{T}$ is closed under entailment, and so $\varphi \in \mathbf{T}$ iff $\mathbf{T} \vdash \varphi$. *

This notion of completeness is due to Emil Post, and for me it is tempting to write it as completeness^{Post}, since there is another sense of completeness used in logic: the sense of *capturing everything valid according to the semantics* at play, sometimes called “semantic completeness”, or if we like, “ $\models$ -completeness”.

These aren’t the same notion. In some contexts, they’re barely related. When we proved that first-order logic with identity is complete in the sense that if φ is identity-valid, then $\vdash^= \varphi$, we certainly weren’t proving, or attempting to prove that either $\models^= \varphi$ or $\models^= \neg\varphi$ for every φ . If φ is contingent, neither is, nor should be, true.

However, if the semantics at play is *truth in just one intended structure*, such as truth-in- $\mathfrak{N}$, there is more of a connection.

Result: *If for all φ such that $\mathfrak{N} \models \varphi$, it holds that $\mathbf{T} \models \varphi$, then $\mathbf{T}$ is complete. (Or if φ is $\mathfrak{N} \models$ -complete then it is complete^{Post}.)*

Proof: Suppose $\mathbf{T}$ proves all sentences true in the standard interpretation. For every sentence φ , either it or its negation is true in the standard interpretation. Hence, either $\mathbf{T} \vdash \varphi$ or $\mathbf{T} \vdash \neg\varphi$. *

Corollary: *$\mathbf{TA}$ is complete.*

Proof: $\mathbf{TA}$ meets the condition in the previous result by definition. *

In the other direction, if a theory is complete^{Post}, and $\mathfrak{N}$ is a model for it, then it is $\mathfrak{N} \models$ -complete as well.

Result: *If a theory $\mathbf{T} \subseteq \mathbf{TA}$ is complete, then for all φ such that $\mathfrak{N} \models \varphi$, it holds that $\mathbf{T} \vdash \varphi$.*

Proof: Suppose $\mathbf{T}$ is such a theory and suppose $\mathfrak{N} \models \varphi$. Since $\mathbf{T}$ is complete, either $\mathbf{T} \vdash \varphi$ or $\mathbf{T} \vdash \neg\varphi$. But suppose $\mathbf{T} \vdash \neg\varphi$ for *reductio*. Since $\mathbf{T} \subseteq \mathbf{TA}$, it follows that $\mathbf{TA} \vdash \neg\varphi$ and therefore $\mathfrak{N} \models \neg\varphi$. However, it cannot be that both $\mathfrak{N} \models \varphi$ and $\mathfrak{N} \models \neg\varphi$. Hence it must be that $\mathbf{T} \vdash \varphi$. *

So in this context, there is a kind of convergence between the two senses. But this is not the case in every context, because in other contexts, we may be interested in more than just a single structure.

In effect, Gödel showed that $\mathbf{PA}$ is incomplete in both senses. How? Well, speak of the devil, it's finally time for ... (drumroll please) ...

Gödel's first incompleteness theorem

Suppose a sentence $\mathcal{G}$ says of itself that it's not provable. If we can prove it, we thereby disprove it. If we can't prove it, there is something true we cannot prove. Frustrating.

Result: *If theory $\mathbf{T}$ is any ω -consistent, recursively axiomatizable extension of $\mathbf{Q}$, then $\mathbf{T}$ has at least one undecidable sentence, $\mathcal{G}_{\mathbf{T}}$, called “the Gödel sentence for $\mathbf{T}$ ”.*

Proof: (1) Suppose $\mathbf{T}$ is such a theory.
 (2) ($\text{Prf} \vdash$) applies, and so we have a wff $\text{Prf}_{\mathbf{T}}(x, y)$, in terms of which we can define $\text{Prov}_{\mathbf{T}}(x)$.
 (3) Consider the wff $\neg\text{Prov}_{\mathbf{T}}(x)$. This is a wff with only one free variable. By the fixed point theorem, there is a wff $\mathcal{G}_{\mathbf{T}}$ such that:

$$\mathbf{T} \vdash \mathcal{G}_{\mathbf{T}} \leftrightarrow \neg\text{Prov}_{\mathbf{T}}(\ulcorner \mathcal{G}_{\mathbf{T}} \urcorner)$$

($\mathcal{G}_{\mathbf{T}}$ says of itself, “I’m not provable in $\mathbf{T}$!”)

(4) First, we show that $\mathbf{T} \not\vdash \mathcal{G}_{\mathbf{T}}$.

- Suppose for *reductio* that $\mathbf{T} \vdash \mathcal{G}_{\mathbf{T}}$.
- Then $\text{Prf}_{\mathbf{T}}(k, \ulcorner \mathcal{G}_{\mathbf{T}} \urcorner)$ holds for some k .
- By ($\text{Prf} \vdash$), $\mathbf{T} \vdash \text{Prf}_{\mathbf{T}}(\bar{k}, \ulcorner \mathcal{G}_{\mathbf{T}} \urcorner)$.
- By EG, $\mathbf{T} \vdash \exists y \text{Prf}_{\mathbf{T}}(y, \ulcorner \mathcal{G}_{\mathbf{T}} \urcorner)$.
- This is the same as $\mathbf{T} \vdash \text{Prov}_{\mathbf{T}}(\ulcorner \mathcal{G}_{\mathbf{T}} \urcorner)$.
- But by (3) and our assumption, $\mathbf{T} \vdash \neg\text{Prov}_{\mathbf{T}}(\ulcorner \mathcal{G}_{\mathbf{T}} \urcorner)$.
- Hence $\mathbf{T}$ is inconsistent.
- However, $\mathbf{T}$ must be consistent, as it is ω -consistent.
- Hence our assumption that $\mathbf{T} \vdash \mathcal{G}_{\mathbf{T}}$ is impossible.

(5) By (4), we know that there is no number n for which $\text{Prf}_{\mathbf{T}}(n, \ulcorner \mathcal{G}_{\mathbf{T}} \urcorner)$ holds. Hence for every number n , $\neg\text{Prf}_{\mathbf{T}}(n, \ulcorner \mathcal{G}_{\mathbf{T}} \urcorner)$ instead. By ($\text{Prf} \vdash$), we have, for every number n :

$$\mathbf{T} \vdash \neg\text{Prf}_{\mathbf{T}}(\bar{n}, \ulcorner \mathcal{G}_{\mathbf{T}} \urcorner)$$

(6) Since $\mathbf{T}$ is ω -consistent, from (5), it follows that $\mathbf{T} \not\vdash \exists y \text{Prf}_{\mathbf{T}}(y, \ulcorner \mathcal{G}_{\mathbf{T}} \urcorner)$

(7) But this means that $\mathbf{T} \not\models \neg \mathcal{G}_{\mathbf{T}}$.

- Suppose otherwise, i.e., that $\mathbf{T} \vdash \neg \mathcal{G}_{\mathbf{T}}$.
- By (3), $\mathbf{T} \vdash \text{Prov}_{\mathbf{T}}(\ulcorner \mathcal{G}_{\mathbf{T}} \urcorner)$.
- But this is $\mathbf{T} \vdash \exists y \text{Prf}_{\mathbf{T}}(y, \ulcorner \mathcal{G}_{\mathbf{T}} \urcorner)$, which we have already shown impossible. *

(8) By (4) and (7), we get that $\mathcal{G}_{\mathbf{T}}$ is an undecidable sentence of $\mathbf{T}$.

Corollary: *Every recursively axiomatizable, ω -consistent extension of $\mathbf{Q}$ is incomplete.*

Proof: By the first incompleteness theorem, every such theory has at least one undecidable sentence.*

Corollary: *$\mathbf{PA}$ is incomplete.*

Proof: $\mathbf{PA}$ is a recursively axiomatizable, ω -consistent extension of $\mathbf{Q}$, and so the previous corollary applies. *

Corollary: *There are sentences true in the standard interpretation $\mathfrak{N}$ not provable in $\mathbf{PA}$.*

Proof: This follows from our earlier result that all $\mathfrak{N} \models$ -complete theories are complete^{Post}. Indeed, it is clear that $\mathcal{G}_{\mathbf{PA}}$ is itself such a sentence, as it asserts of itself that it is not provable in $\mathbf{PA}$, and is correct about that. *

Corollary: *Every ω -consistent, recursively axiomatizable extension of $\mathbf{PA}$ is incomplete.*

Proof: Every extension of $\mathbf{PA}$ is also an extension of $\mathbf{Q}$, so the first corollary above applies. *

Corollary: *$\mathbf{TA}$ is not recursively axiomatizable.*

Proof: $\mathbf{TA}$ is an ω -consistent extension of $\mathbf{Q}$. If it were recursively axiomatizable, it would be incomplete. However, it is complete. *

Corollary: *$\mathbf{PA}$ is not $\mathbf{TA}$.*

Proof: $\mathbf{PA}$ is recursively axiomatizable whereas $\mathbf{TA}$ is not. (One is complete, the other is not, and so on...)*

HOMEWORK 29

Let $\text{Theorem}_{\mathbf{PA}}$ be the property a number has just in case it is the Gödel number of a wff φ such that $\mathbf{PA} \vdash \varphi$. Prove that despite what one may assume, $\text{Theorem}_{\mathbf{PA}}$ is not representable in $\mathbf{PA}$ by the wff $\text{Prov}_{\mathbf{PA}}(x)$ as defined earlier, at least using the technical definition of “representable”.

The Gödel-Rosser theorem

The idea of a consistent but non- ω -consistent arithmetical theory seems weird. It’s sort of like that friend who says they want to get together, only not on Monday, and not on Tuesday, and not on Wednesday, and not next Monday either, and so on.

Still, one might be tempted to endorse such a theory if it were a route to completeness. But J. B. Rosser is a spoilsport.

Result: *If theory $\mathbf{T}$ is a consistent, recursively axiomatizable extension of $\mathbf{Q}$, then $\mathbf{T}$ has at least one undecidable sentence, $\mathcal{R}_{\mathbf{T}}$, called “the Rosser sentence for $\mathbf{T}$ ”.*

- Proof:* (1) Suppose $\mathbf{T}$ is such a theory.
- (2) ($\text{Prf} \vdash$) applies, so we can again use object language formula $\text{Prf}_{\mathbf{T}}(x, y)$.
- (3) The function $\text{neg}(k)$, whose value for k as argument, is the Gödel number of the negation of the formula with Gödel number k , is primitive recursive, and thus representable in $\mathbf{T}$. Let $\text{neg}(x, y)$ represent it, so, for every wff φ :

$$\begin{aligned}\mathbf{T} &\vdash \text{neg}(\ulcorner \varphi \urcorner, \ulcorner \neg \varphi \urcorner) \\ \mathbf{T} &\vdash \forall y (\text{neg}(\ulcorner \varphi \urcorner, y) \rightarrow \ulcorner \neg \varphi \urcorner = y)\end{aligned}$$

- (4) Consider the wff, hereafter abbreviated $\rho(x)$:

$$\forall y [\text{Prf}_{\mathbf{T}}(y, x) \rightarrow \forall z (\text{neg}(x, z) \rightarrow \exists w (w < y \wedge \text{Prf}_{\mathbf{T}}(w, z)))]$$

- (5) The fixed point theorem applies, and so there is a wff $\mathcal{R}_{\mathbf{T}}$ such that:

$$\mathbf{T} \vdash \mathcal{R}_{\mathbf{T}} \leftrightarrow \rho(\ulcorner \mathcal{R}_{\mathbf{T}} \urcorner)$$

Given how we've defined $\rho(x)$, $\mathcal{R}_{\mathbf{T}}$ says of itself that for every proof of it, there's a proof with a smaller Gödel number of its negation (or "all" its negations). Or even more crudely, $\mathcal{R}_{\mathbf{T}}$ says "if I'm provable, so is my negation!"

- (6) By applying (3) to $\mathcal{R}_{\mathbf{T}}$ itself, we have:

$$\begin{aligned}\mathbf{T} &\vdash \text{neg}(\ulcorner \mathcal{R}_{\mathbf{T}} \urcorner, \ulcorner \neg \mathcal{R}_{\mathbf{T}} \urcorner) \\ \mathbf{T} &\vdash \forall y (\text{neg}(\ulcorner \mathcal{R}_{\mathbf{T}} \urcorner, y) \rightarrow \ulcorner \neg \mathcal{R}_{\mathbf{T}} \urcorner = y)\end{aligned}$$

- (7) Let's show that $\mathbf{T} \not\vdash \mathcal{R}_{\mathbf{T}}$.

- Suppose for *reductio* that $\mathbf{T} \vdash \mathcal{R}_{\mathbf{T}}$.
- From (5), we get $\mathbf{T} \vdash \rho(\ulcorner \mathcal{R}_{\mathbf{T}} \urcorner)$, i.e., $\mathbf{T} \vdash \forall y [\text{Prf}_{\mathbf{T}}(y, \ulcorner \mathcal{R}_{\mathbf{T}} \urcorner) \rightarrow \forall z (\text{neg}(\ulcorner \mathcal{R}_{\mathbf{T}} \urcorner, z) \rightarrow \exists w (w < y \wedge \text{Prf}_{\mathbf{T}}(w, z)))]$
- $\text{Prf}_{\mathbf{T}}(k, \ulcorner \mathcal{R}_{\mathbf{T}} \urcorner)$ holds for some number k , and by ($\text{Prf} \vdash$), $\mathbf{T} \vdash \text{Prf}_{\mathbf{T}}(\bar{k}, \ulcorner \mathcal{R}_{\mathbf{T}} \urcorner)$.
- Together, we get $\mathbf{T} \vdash \forall z (\text{neg}(\ulcorner \mathcal{R}_{\mathbf{T}} \urcorner, z) \rightarrow \exists w (w < \bar{k} \wedge \text{Prf}_{\mathbf{T}}(w, z)))$.
- With (6), we then have $\mathbf{T} \vdash \exists w (w < \bar{k} \wedge \text{Prf}_{\mathbf{T}}(w, \ulcorner \neg \mathcal{R}_{\mathbf{T}} \urcorner))$; by simple logic, $\mathbf{T} \vdash \neg \forall w (w < \bar{k} \rightarrow \neg \text{Prf}_{\mathbf{T}}(w, \ulcorner \neg \mathcal{R}_{\mathbf{T}} \urcorner))$.
- $\mathbf{T}$ consistent and so it must be that $\mathbf{T} \not\vdash \neg \mathcal{R}_{\mathbf{T}}$.
- But then for every number n , and *a fortiori*, for every number less than k , by ($\text{Prf} \vdash$):

$$\mathbf{T} \vdash \neg \text{Prf}_{\mathbf{T}}(\bar{n}, \ulcorner \neg \mathcal{R}_{\mathbf{T}} \urcorner)$$

- In the object language, assume that $b < \bar{k}$. By (Spread), $b = 0 \vee \dots \vee b = \bar{k} - 1$. By a big proof by cases $\neg \text{Prf}_{\mathbf{T}}(b, \ulcorner \neg \mathcal{R}_{\mathbf{T}} \urcorner)$. Discharging and generalizing, $\mathbf{T} \vdash \forall w (w < \bar{k} \rightarrow \neg \text{Prf}_{\mathbf{T}}(w, \ulcorner \neg \mathcal{R}_{\mathbf{T}} \urcorner))$.
 - But then $\mathbf{T} \vdash \perp$, which is impossible.
- (8) Step (7) ensures us that for every number n , $\neg \text{Prf}_{\mathbf{T}}(n, \ulcorner \mathcal{R}_{\mathbf{T}} \urcorner)$, and so, $\mathbf{T} \vdash \neg \text{Prf}_{\mathbf{T}}(\bar{n}, \ulcorner \mathcal{R}_{\mathbf{T}} \urcorner)$ by ($\text{Prf} \vdash$).
- (9) Now we show $\mathbf{T} \not\vdash \neg \mathcal{R}_{\mathbf{T}}$.
- Suppose for *reductio* that $\mathbf{T} \vdash \neg \mathcal{R}_{\mathbf{T}}$.
 - Then for some j , $\text{Prf}_{\mathbf{T}}(j, \ulcorner \neg \mathcal{R}_{\mathbf{T}} \urcorner)$, and so, by ($\text{Prf} \vdash$), $\mathbf{T} \vdash \text{Prf}_{\mathbf{T}}(\bar{j}, \ulcorner \neg \mathcal{R}_{\mathbf{T}} \urcorner)$.
 - Suppose in the object language that $\text{Prf}_{\mathbf{T}}(b, \ulcorner \mathcal{R}_{\mathbf{T}} \urcorner)$.
 - By (Trichotomy), we have $b < \bar{j} \vee \bar{j} < b \vee b = \bar{j}$.
 - We can rule out $b = \bar{j}$, because, by (8), we have $\mathbf{T} \vdash \neg \text{Prf}_{\mathbf{T}}(\bar{j}, \ulcorner \mathcal{R}_{\mathbf{T}} \urcorner)$, which contradicts our supposition.
 - We can also rule out $b < \bar{j}$. For suppose $b < \bar{j}$. Then, by (Spread), we have $b = 0 \vee \dots \vee b = \bar{j} - 1$, and each disjunct, with (8), leads to a contradiction with our supposition.
 - Hence $\bar{j} < b$. So $\bar{j} < b \wedge \text{Prf}_{\mathbf{T}}(\bar{j}, \ulcorner \neg \mathcal{R}_{\mathbf{T}} \urcorner)$, and then, by EG, $\exists w (w < b \wedge \text{Prf}_{\mathbf{T}}(w, \ulcorner \neg \mathcal{R}_{\mathbf{T}} \urcorner))$.
 - Make another supposition in the object language: $\text{neg}(\ulcorner \mathcal{R}_{\mathbf{T}} \urcorner, c)$. Then, by (6), $\ulcorner \neg \mathcal{R}_{\mathbf{T}} \urcorner = c$, and we get $\exists w (w < b \wedge \text{Prf}_{\mathbf{T}}(w, c))$ by LL. Discharging and generalizing, we get: $\forall z (\text{neg}(\ulcorner \mathcal{R}_{\mathbf{T}} \urcorner, z) \rightarrow \exists w (w < b \wedge \text{Prf}_{\mathbf{T}}(w, z)))$.
 - Discharging and generalizing the earlier assumption $\text{Prf}_{\mathbf{T}}(b, \ulcorner \mathcal{R}_{\mathbf{T}} \urcorner)$, we get $\mathbf{T} \vdash \rho(\ulcorner \mathcal{R}_{\mathbf{T}} \urcorner)$.
 - But then, by (5), we get $\mathbf{T} \vdash \mathcal{R}_{\mathbf{T}}$, which would make $\mathbf{T}$ inconsistent, which is impossible.
- (10) By (7) and (9), $\mathcal{R}_{\mathbf{T}}$ is an undecidable sentence of $\mathbf{T}$. *

Corollary: *Every consistent, recursively axiomatizable extension of $\mathbf{PA}$ is incomplete.*

Proof: Every extension of **PA** is also an extension of **Q**, so the Gödel-Rosser theorem applies, and it has at least one undecidable sentence. *

The Hilbert-Bernays derivability conditions

Definition: The formula $\text{Prov}_T(x)$ for theory **T** meets the **Hilbert-Bernays derivability conditions** iff the following are true for all sentences φ and ψ :

- (HB1) If $\mathbf{T} \vdash \varphi$ then $\mathbf{T} \vdash \text{Prov}_T(\ulcorner \varphi \urcorner)$.
- (HB2) $\mathbf{T} \vdash \text{Prov}_T(\ulcorner \varphi \rightarrow \psi \urcorner) \rightarrow (\text{Prov}_T(\ulcorner \varphi \urcorner) \rightarrow \text{Prov}_T(\ulcorner \psi \urcorner))$
- (HB3) $\mathbf{T} \vdash \text{Prov}_T(\ulcorner \varphi \urcorner) \rightarrow \text{Prov}_T(\ulcorner \text{Prov}_T(\ulcorner \varphi \urcorner) \urcorner)$

If you are familiar with modal logic, you may notice a strong similarity between these and the principles/rules of modal system S4.

Result: The derivability conditions hold for $\text{Prov}_{\mathbf{PA}}(x)$ for **PA**, i.e., for all sentences φ and ψ :

- (HB1) If $\mathbf{PA} \vdash \varphi$ then $\mathbf{PA} \vdash \text{Prov}_{\mathbf{PA}}(\ulcorner \varphi \urcorner)$.
- (HB2) $\mathbf{PA} \vdash \text{Prov}_{\mathbf{PA}}(\ulcorner \varphi \rightarrow \psi \urcorner) \rightarrow (\text{Prov}_{\mathbf{PA}}(\ulcorner \varphi \urcorner) \rightarrow \text{Prov}_{\mathbf{PA}}(\ulcorner \psi \urcorner))$
- (HB3) $\mathbf{PA} \vdash \text{Prov}_{\mathbf{PA}}(\ulcorner \varphi \urcorner) \rightarrow \text{Prov}_{\mathbf{PA}}(\ulcorner \text{Prov}_{\mathbf{PA}}(\ulcorner \varphi \urcorner) \urcorner)$

Sad excuse: While proving (HB1) for $\text{Prov}_{\mathbf{PA}}(x)$ can be done using results we have studied so far, establishing (HB2) and (HB3) are too complicated to be worth our time. They involve digging in to the exact definition of $\text{Prf}_{\mathbf{PA}}(x, y)$, which comes from the recursive definition of the relation $\text{Prf}_{\mathbf{PA}}$ from the initial functions, and how this translates into the object language with many nested uses of φ_β , which is a nightmare. Moreover, our main interest is using these to help demonstrate what **PA** cannot prove, and clearly, if we were worried that the above did not hold, this would not improve our estimation of what **PA** can do.

HOMework 30

Prove that (HB1) holds for $\text{Prov}_{\mathbf{PA}}(x)$. Hint: use $(\text{Prf} \vdash)$.

Gödel's second incompleteness theorem

PA is consistent, but **PA** cannot prove its own consistency.

For any theory **T** for which we have a provability predicate $\text{Prov}_T(x)$, let Con_T be the wff $\neg \text{Prov}_T(\ulcorner \perp \urcorner)$.

Result: $\mathbf{PA} \not\vdash \text{Con}_{\mathbf{PA}}$

- Proof:* (1) Recall the Gödel sentence for **PA**; we have, $\mathbf{PA} \vdash \mathcal{G}_{\mathbf{PA}} \leftrightarrow \neg \text{Prov}_{\mathbf{PA}}(\ulcorner \mathcal{G}_{\mathbf{PA}} \urcorner)$.
- (2) $\mathbf{PA} \vdash \mathcal{G}_{\mathbf{PA}} \rightarrow \neg \text{Prov}_{\mathbf{PA}}(\ulcorner \mathcal{G}_{\mathbf{PA}} \urcorner)$ by (1) and BE.
- (3) $\mathbf{PA} \vdash \mathcal{G}_{\mathbf{PA}} \rightarrow (\text{Prov}_{\mathbf{PA}}(\ulcorner \mathcal{G}_{\mathbf{PA}} \urcorner) \rightarrow \perp)$ by (2) and some easy logic.
- (4) $\mathbf{PA} \vdash \text{Prov}_{\mathbf{PA}}(\ulcorner \mathcal{G}_{\mathbf{PA}} \rightarrow (\text{Prov}_{\mathbf{PA}}(\ulcorner \mathcal{G}_{\mathbf{PA}} \urcorner) \rightarrow \perp) \urcorner) \rightarrow$
by (3) and (HB1).
- (5) $\mathbf{PA} \vdash \text{Prov}_{\mathbf{PA}}(\ulcorner \mathcal{G}_{\mathbf{PA}} \urcorner) \rightarrow$
 $\text{Prov}_{\mathbf{PA}}(\ulcorner \text{Prov}_{\mathbf{PA}}(\ulcorner \mathcal{G}_{\mathbf{PA}} \urcorner) \rightarrow \perp \urcorner)$
by (4), (HB2) and MP.
- (6) $\mathbf{PA} \vdash \text{Prov}_{\mathbf{PA}}(\ulcorner \text{Prov}_{\mathbf{PA}}(\ulcorner \mathcal{G}_{\mathbf{PA}} \urcorner) \rightarrow \perp \urcorner) \rightarrow$
 $[\text{Prov}_{\mathbf{PA}}(\ulcorner \text{Prov}_{\mathbf{PA}}(\ulcorner \mathcal{G}_{\mathbf{PA}} \urcorner) \urcorner) \rightarrow \text{Prov}_{\mathbf{PA}}(\ulcorner \perp \urcorner)]$
by (HB2).
- (7) $\mathbf{PA} \vdash \text{Prov}_{\mathbf{PA}}(\ulcorner \mathcal{G}_{\mathbf{PA}} \urcorner) \rightarrow$
 $[\text{Prov}_{\mathbf{PA}}(\ulcorner \text{Prov}_{\mathbf{PA}}(\ulcorner \mathcal{G}_{\mathbf{PA}} \urcorner) \urcorner) \rightarrow \text{Prov}_{\mathbf{PA}}(\ulcorner \perp \urcorner)]$
by (5) and (6) and HS.
- (8) $\mathbf{PA} \vdash \text{Prov}_{\mathbf{PA}}(\ulcorner \mathcal{G}_{\mathbf{PA}} \urcorner) \rightarrow$
 $\text{Prov}_{\mathbf{PA}}(\ulcorner \text{Prov}_{\mathbf{PA}}(\ulcorner \mathcal{G}_{\mathbf{PA}} \urcorner) \urcorner)$ by (HB3).
- (9) $\mathbf{PA} \vdash \text{Prov}_{\mathbf{PA}}(\ulcorner \mathcal{G}_{\mathbf{PA}} \urcorner) \rightarrow \text{Prov}_{\mathbf{PA}}(\ulcorner \perp \urcorner)$
by (7) and (8) and CMP.
- (10) $\mathbf{PA} \vdash \neg \text{Prov}_{\mathbf{PA}}(\ulcorner \perp \urcorner) \rightarrow \neg \text{Prov}_{\mathbf{PA}}(\ulcorner \mathcal{G}_{\mathbf{PA}} \urcorner)$
by (9) and Trans.
- (11) $\mathbf{PA} \vdash \neg \text{Prov}_{\mathbf{PA}}(\ulcorner \mathcal{G}_{\mathbf{PA}} \urcorner) \rightarrow \mathcal{G}_{\mathbf{PA}}$ by (1) and BE.
- (12) $\mathbf{PA} \vdash \neg \text{Prov}_{\mathbf{PA}}(\ulcorner \perp \urcorner) \rightarrow \mathcal{G}_{\mathbf{PA}}$
by (10) and (11) and HS.
- (13) $\neg \text{Prov}_{\mathbf{PA}}(\ulcorner \perp \urcorner)$ is $\text{Con}_{\mathbf{PA}}$, and so if it were true that $\mathbf{PA} \vdash \text{Con}_{\mathbf{PA}}$, by MP, it would be true that $\mathbf{PA} \vdash \mathcal{G}_{\mathbf{PA}}$, but we know it is not.
- (14) Hence it must be that $\mathbf{PA} \not\vdash \text{Con}_{\mathbf{PA}}$. *

Actually, it was discovered later that not only can **PA** not prove of itself that it cannot prove a contradiction, it can *never* prove that it cannot prove something, even if it cannot prove it.

Löb's theorem

You may be familiar with the paradox variously called **Curry's paradox** or **Löb's paradox** or the **Santa Claus paradox**, which is another consequence of self-reference run amok.

Let S be the self-referential sentence:

"If S is true, Santa Claus exists."

We can use this to "prove" that Santa Claus exists.

1. S is true. (Assumption)
2. If S is true, Santa Claus exists. (1, Meaning of S)
3. Santa Claus exists. (1, 2 MP)
4. If S is true, Santa Claus exists. (1–3 CP)
5. S is true. (4, Meaning of S)
6. Santa Claus exists. (4, 5 MP)

Of course, the choice of "Santa Claus exists" here is arbitrary. We could have used any other sentence instead.

As we know, the fixed point theorem makes self-reference possible in **Q** and **PA**, etc. We cannot quite recreate Curry's paradox, because we don't have a truth predicate. (We'll prove we don't soon.) But if we have a provability predicate with certain characteristics, we can wreak havoc in a similar way, by forming sentences that say of themselves if they're provable, then something else.

Result: If $\mathbf{T}$ is an extension of **Q** for which $\text{Prov}_{\mathbf{T}}(x)$ meets the Hilbert-Bernays derivability conditions, then for any sentence φ if $\mathbf{T} \vdash \text{Prov}_{\mathbf{T}}(\ulcorner \varphi \urcorner) \rightarrow \varphi$ then $\mathbf{T} \vdash \varphi$.

Proof: (1) Let $\mathbf{T}$ be such a theory, and φ any wff.
 (2) Suppose that $\mathbf{T} \vdash \text{Prov}_{\mathbf{T}}(\ulcorner \varphi \urcorner) \rightarrow \varphi$.
 (3) Consider the open sentence $\text{Prov}_{\mathbf{T}}(x) \rightarrow \varphi$. The fixed point theorem tells us that there is a wff ψ such that, $\mathbf{T} \vdash \psi \leftrightarrow (\text{Prov}_{\mathbf{T}}(\ulcorner \psi \urcorner) \rightarrow \varphi)$.

- (4) $\mathbf{T} \vdash \psi \rightarrow (\text{Prov}_{\mathbf{T}}(\ulcorner \psi \urcorner) \rightarrow \varphi)$ by (3) and BE.
- (5) $\mathbf{T} \vdash \text{Prov}_{\mathbf{T}}(\ulcorner \psi \rightarrow (\text{Prov}_{\mathbf{T}}(\ulcorner \psi \urcorner) \rightarrow \varphi) \urcorner)$
by (4) and (HB1).
- (6) $\mathbf{T} \vdash \text{Prov}_{\mathbf{T}}(\ulcorner \psi \urcorner) \rightarrow$
 $\text{Prov}_{\mathbf{T}}(\ulcorner \text{Prov}_{\mathbf{T}}(\ulcorner \psi \urcorner) \rightarrow \varphi \urcorner)$
by (5), (HB2) and MP.
- (7) $\mathbf{T} \vdash \text{Prov}_{\mathbf{T}}(\ulcorner \text{Prov}_{\mathbf{T}}(\ulcorner \psi \urcorner) \rightarrow \varphi \urcorner) \rightarrow$
 $(\text{Prov}_{\mathbf{T}}(\ulcorner \text{Prov}_{\mathbf{T}}(\ulcorner \psi \urcorner) \urcorner) \rightarrow \text{Prov}_{\mathbf{T}}(\ulcorner \varphi \urcorner))$
by (HB2).
- (8) $\mathbf{T} \vdash \text{Prov}_{\mathbf{T}}(\ulcorner \psi \urcorner) \rightarrow$
 $(\text{Prov}_{\mathbf{T}}(\ulcorner \text{Prov}_{\mathbf{T}}(\ulcorner \psi \urcorner) \urcorner) \rightarrow \text{Prov}_{\mathbf{T}}(\ulcorner \varphi \urcorner))$
by (6) and (7) and HS.
- (9) $\mathbf{T} \vdash \text{Prov}_{\mathbf{T}}(\ulcorner \psi \urcorner) \rightarrow \text{Prov}_{\mathbf{T}}(\ulcorner \text{Prov}_{\mathbf{T}}(\ulcorner \psi \urcorner) \urcorner)$
by (HB3).
- (10) $\mathbf{T} \vdash \text{Prov}_{\mathbf{T}}(\ulcorner \psi \urcorner) \rightarrow \text{Prov}_{\mathbf{T}}(\ulcorner \varphi \urcorner)$
by (8) and (9) and CMP.
- (11) $\mathbf{T} \vdash \text{Prov}_{\mathbf{T}}(\ulcorner \psi \urcorner) \rightarrow \varphi$ by (2) and (10) and HS.
- (12) $\mathbf{T} \vdash \psi$ by (3) and (11) and BMP.
- (13) $\mathbf{T} \vdash \text{Prov}_{\mathbf{T}}(\ulcorner \psi \urcorner)$ by (12) and (HB1).
- (14) $\mathbf{T} \vdash \varphi$ by (11) and (13) and MP. *

Löb's theorem is extremely weird. Shouldn't it be true for *every* sentence, even the a false one, that *if* it were provable, *then* it'd be true? How could this enough to show that it is *in fact* true?

Corollary: For any sentence φ of **PA**, if $\mathbf{PA} \vdash \text{Prov}_{\mathbf{PA}}(\ulcorner \varphi \urcorner) \rightarrow \varphi$ then $\mathbf{PA} \vdash \varphi$.

Proof: **PA** meets the conditions for Löb's theorem to apply. *

Corollary: There is no sentence φ such that $\mathbf{PA} \vdash \neg \text{Prov}_{\mathbf{PA}}(\ulcorner \varphi \urcorner)$.

Proof: (1) Suppose $\mathbf{PA} \vdash \neg \text{Prov}_{\mathbf{PA}}(\ulcorner \varphi \urcorner)$ for reductio.
 (2) By FA, $\mathbf{PA} \vdash \text{Prov}_{\mathbf{PA}}(\ulcorner \varphi \urcorner) \rightarrow \varphi$.
 (3) By the previous corollary, $\mathbf{PA} \vdash \varphi$.
 (4) By (HB1), $\mathbf{PA} \vdash \text{Prov}_{\mathbf{PA}}(\ulcorner \varphi \urcorner)$.

- (5) Hence $\mathbf{PA} \vdash \perp$, which we know to be impossible. $\ast$

Note that we did not use the second incompleteness theorem in proving Löb's theorem, and so we might have got it instead directly from this corollary.

In a sense, these results show that there are limits to the extent that $\mathbf{PA}$ captures its own metatheory, even though its language can often express metatheoretic results about itself that it cannot prove.

HOMEWORK 31

Consider the *Henkin sentence*, $H_{\mathbf{PA}}$ for $\mathbf{PA}$, which is like the “opposite” of the Gödel sentence: rather than asserting its own unprovability, it asserts its own provability. Answer these questions:

- What instance of the fixed point theorem gives us $H_{\mathbf{PA}}$?
- Is $H_{\mathbf{PA}}$ an undecidable sentence of $\mathbf{PA}$ like $G_{\mathbf{PA}}$?
- If not, $\vdash H_{\mathbf{PA}}$ or $\vdash \neg H_{\mathbf{PA}}$?
- Is $H_{\mathbf{PA}}$ true in the standard interpretation? Is there any way to tell?

11. Arithmetic Definability and Truth

Definition: A n -place relation R between natural numbers is **arithmetically definable** iff there is a wff $\varphi(x_1, \dots, x_n)$ of $\mathcal{L}_A$ such that $R(k_1, \dots, k_n)$ iff $\mathfrak{N} \models \varphi(\overline{k_1}, \dots, \overline{k_n})$.

Recall that we can consider properties to be 1-place relations, so a property of numbers is arithmetically definable iff there is a $\varphi(x)$ where k has the property $\varphi(\overline{k})$ is true in $\mathfrak{N}$. We restrict ourselves to $\mathcal{L}_A$ here, so $\varphi(x)$ must use only $=$, $<$, $+$, $\times$, $'$, 0 , and logical symbols.

Result: If R is representable in any theory $\mathbf{T}$ where $\mathbf{T} \subseteq \mathbf{TA}$, then R is arithmetically definable.

Proof: (1) Suppose R is representable in such a theory $\mathbf{T}$.

- Then there is a wff $\varphi(x_1, \dots, x_n)$, where
 - if $R(k_1, \dots, k_n)$, then $\mathbf{T} \vdash \varphi(\overline{k_1}, \dots, \overline{k_n})$; b) if not- $R(k_1, \dots, k_n)$ then $\mathbf{T} \vdash \neg\varphi(\overline{k_1}, \dots, \overline{k_n})$
- Suppose R holds. Then $\mathbf{T} \vdash \varphi(\overline{k_1}, \dots, \overline{k_n})$, and since $\mathbf{T} \subseteq \mathbf{TA}$, $\mathbf{TA} \vdash \varphi(\overline{k_1}, \dots, \overline{k_n})$, and so $\mathfrak{N} \models \varphi(\overline{k_1}, \dots, \overline{k_n})$.
- Suppose $\mathfrak{N} \models \varphi(\overline{k_1}, \dots, \overline{k_n})$ and for *reductio* suppose not- $R(k_1, \dots, k_n)$. Then $\mathbf{T} \vdash \neg\varphi(\overline{k_1}, \dots, \overline{k_n})$. But by reasoning similar to the previous direction, this means that $\mathfrak{N} \models \neg\varphi(\overline{k_1}, \dots, \overline{k_n})$, which is impossible. $\ast$

Corollary: Every recursive relation is arithmetically definable.

Proof: Every recursive relation is representable in $\mathbf{Q}$ and $\mathbf{Q} \subseteq \mathbf{TA}$. $\ast$

However, not all arithmetically definable relations are representable in $\mathbf{Q}$ and $\mathbf{PA}$. For example, the property a number has just in case it is the Gödel number of a theorem of $\mathbf{PA}$ is arithmetically definable by $\text{Prov}_{\mathbf{PA}}(x)$, but it is not representable in $\mathbf{PA}$ by that wff, nor any other. (We'll prove that in the next section.)

Definition: By extension, we say that property Φ of sentences of a first-order language is **arithmetically definable** iff the property of being the Gödel number of a sentence having the property is arithmetically definable in the above sense, i.e., if there is a wff of $\mathcal{L}_A$ such that ψ has Φ iff $\mathfrak{N} \models \varphi(\ulcorner \psi \urcorner)$.

Result (Tarski's theorem): The property of truth in $\mathfrak{N}$ is not arithmetically definable.

Proof: (1) Suppose for *reductio* that truth in $\mathfrak{N}$ is arithmetically definable.

- (2) By the definition of arithmetical definability, there is a wff which we could write as $\text{true}(x)$ such that for all wffs ψ , $\mathfrak{N} \models \psi$ iff $\mathfrak{N} \models \text{true}(\ulcorner \psi \urcorner)$.
- (3) If we apply the fixed point theorem to $\neg \text{true}(x)$, we get:

$$\mathbf{TA} \vdash \mathcal{L} \leftrightarrow \neg \text{true}(\ulcorner \mathcal{L} \urcorner)$$

(An arithmetized version of the liar sentence of the Epimenides/Liar paradox.)

- (4) It cannot be that $\mathfrak{N} \models \mathcal{L}$. For suppose otherwise. Then, by (2), $\mathfrak{N} \models \text{true}(\ulcorner \mathcal{L} \urcorner)$, which would mean that $\mathbf{TA} \vdash \text{true}$. Then, by (3), $\mathbf{TA} \vdash \neg \mathcal{L}$, and hence both $\mathfrak{N} \models \mathcal{L}$ and $\mathfrak{N} \models \neg \mathcal{L}$, which is impossible.
- (5) Since $\mathcal{L}$ is closed, from (4), $\mathfrak{N} \models \neg \mathcal{L}$. But then, $\mathbf{TA} \vdash \neg \mathcal{L}$, and by (3), $\mathbf{TA} \vdash \text{true}(\ulcorner \mathcal{L} \urcorner)$. This would mean $\mathfrak{N} \models \text{true}(\ulcorner \mathcal{L} \urcorner)$, and by (2), $\mathfrak{N} \models \mathcal{L}$, and again, $\mathcal{L}$ and its negation are both true in $\mathfrak{N}$. This is impossible.
- (6) Hence, our assumption at (1) must be mistaken, and truth in $\mathfrak{N}$ is not arithmetically definable. *

Note that truth in $\mathfrak{N}$ is, set-theoretically, the same “property” as being provable in $\mathbf{TA}$. We have seen that, e.g., $\text{Prov}_{\mathbf{PA}}(x)$ doesn’t “represent” provability in $\mathbf{PA}$, even though it semantically *means* that property. For $\mathbf{TA}$, not only is there not an open wff $\text{Prov}_{\mathbf{TA}}$ that “represents” provability in $\mathbf{TA}$ (= truth) there isn’t even a wff in the language of $\mathcal{L}_A$ that means that at all.

A question to ponder: what are the consequences for the philosophy of mathematics that arithmetical truth is not itself something that can be studied in a purely arithmetical way?

Corollary: *Truth in $\mathfrak{N}$ is not a recursive property.*

Proof: All recursive properties are arithmetically definable, but it is not. *

(Actually, we were in a position to prove this earlier. The set of true sentences of arithmetic is the theory $\mathbf{TA}$. We proved earlier that $\mathbf{TA}$ is not recursively axiomatizable. But trivially, $\mathbf{TA}$ axiomatizes itself. Hence, $\mathbf{TA}$ cannot be a recursive set.)

12. Undecidability

Another corollary of Tarski’s theorem

Corollary: *If the Church-Turing thesis is true, there is no effective decision procedure for deciding whether or not a sentence of the language of arithmetic is true, as standardly interpreted.*

Proof: According to the Church-Turing thesis, a property is effectively decidable iff it is recursive. Since the property of being (the Gödel number of) a truth of arithmetic (being such that $\mathfrak{N} \models \varphi$ for a sentence φ of $\mathcal{L}_A$) is not recursive, it is not effectively decidable. *

But it gets worse. Not only is there no recursive decision procedure for deciding truth, there is not even a recursive procedure for deciding what is provable or not in a given consistent theory compatible with arithmetic, whether it be $\mathbf{TA}$, or even $\mathbf{PA}$, $\mathbf{Q}$, or pure predicate logic!

Definition: A theory $\mathbf{T}$ is **recursively decidable** (or just **decidable**) iff it is a recursive set of sentences. Otherwise, it is **recursively undecidable**.

Recall that theories are sets closed under entailment. To say that a theory is undecidable is to say that *there is no recursive decision procedure for determining what is entailed by a certain set of axioms.*

A couple warnings:

- Don’t confuse recursive decidability with recursive axiomatizability. A theory may be the *closure* of a recursive axiom set, without itself being a recursive set.

- Don't confuse the issue of whether a theory is recursively decidable with the issue of whether or not it has undecidable sentences. The use of the same word here is accidental. **TA** for example, is recursively undecidable even though it does not have any undecidable sentences.

Deciding proofs versus deciding theorems

- Question 1: Is there an recursive method for determining what is, and what is not, provable in a theory like **PA**?
- Deriv , and Prf_{PA} are recursive relations: if I give you a alleged proof, there is an effective procedure to check whether the proof is correct or incorrect.
- Because of this, Prf_{PA} is representable as $\text{Prf}_{\text{PA}}(x, y)$ in **PA** itself.
- Using $\text{Prf}_{\text{PA}}(x, y)$ we can define the wff $\text{Prov}_{\text{PA}}(x)$.
- In a homework exercise, you showed that the property of being the Gödel number of a member of **PA** is not representable by this.
- Question 2: Can this property be represented in **PA** at all?
- Since **PA** represents all and only recursive relations, these questions stand or fall together, and the answers to both are no.
- The same questions can be asked about **Q** or even barebones logic and the pattern of answers are the same.

The undecidability of arithmetical theories

Result (recursive undecidability): *If theory **T** is a consistent extension of **Q** then **T** is recursively undecidable.*

Proof: (1) Suppose that **T** is such a theory, and suppose for *reductio* that **T** is recursively decidable.

- (2) This means the property of being the Gödel number of a member of **T** is a recursive property.
- (3) Since **T** is a theory, $\varphi \in \mathbf{T}$ iff $\mathbf{T} \vdash \varphi$.
- (4) **T** is an extension of **Q**, and so it represents all recursive properties.
- (5) There is therefore a wff $\tau(x)$, such that, for any wff φ :
 - (a) If $\mathbf{T} \vdash \varphi$ then $\mathbf{T} \vdash \tau(\ulcorner \varphi \urcorner)$; and
 - (b) If $\mathbf{T} \not\vdash \varphi$ then $\mathbf{T} \vdash \neg \tau(\ulcorner \varphi \urcorner)$.
- (6) Apply the fixed point theorem to $\neg \tau(x)$ to get a Gödel-like sentence:

$$\mathbf{T} \vdash \mathcal{A} \leftrightarrow \neg \tau(\ulcorner \mathcal{A} \urcorner)$$

- (7) It cannot be that $\mathbf{T} \vdash \mathcal{A}$, since then, by (5)-(a) $\mathbf{T} \vdash \tau(\ulcorner \mathcal{A} \urcorner)$, and by (6), $\mathbf{T} \vdash \neg \mathcal{A}$, which would make **T** inconsistent, which we have assumed is not the case.
- (8) But (7) means that $\mathbf{T} \not\vdash \mathcal{A}$, and so, by (5)-(b), $\mathbf{T} \vdash \neg \tau(\ulcorner \mathcal{A} \urcorner)$. But then by (6), $\mathbf{T} \vdash \mathcal{A}$. This we have already shown to be impossible.
- (9) Hence our assumption at (1) that **T** is recursively decidable must be mistaken. *

Corollary: ***Q** and **PA** are recursively undecidable.*

Proof: **PA** is an extension of **Q**, and **Q** is trivially an extension of itself, and both are consistent. *

Finite extensions and decidability

Result (finite extension lemma): *If theory **T** is axiomatized by set Γ , and theory **T**^{*} is axiomatized by the set $\Gamma \cup \{\varphi_1, \dots, \varphi_n\}$, adding a finite number of sentences to Γ , then if **T** is recursively decidable, so is **T**^{*}.*

Proof: (1) Suppose **T** and **T**^{*} are such theories.

- (2) Since $\mathbf{T}$ is axiomatized by Γ , we have that for all ψ , $\mathbf{T} \vdash \psi$ iff $\Gamma \vdash \psi$.
- (3) Similarly, for all ψ , $\mathbf{T}^* \vdash \psi$ iff $\Gamma \cup \{\varphi_1, \dots, \varphi_n\} \vdash \psi$.
- (4) By DT and Imp in one direction, and Conj and MP in the other, it is clear that $\Gamma \cup \{\varphi_1, \dots, \varphi_n\} \vdash \psi$ iff $\Gamma \vdash (\varphi_1 \wedge \dots \wedge \varphi_n) \rightarrow \psi$.
- (5) Hence, for all ψ , $\mathbf{T}^* \vdash \psi$ iff $\mathbf{T} \vdash (\varphi_1 \wedge \dots \wedge \varphi_n) \rightarrow \psi$.
- (6) Let $R_{\mathbf{T}}$ be the property of being the Gödel number of a member of $\mathbf{T}$.
- (7) We can define the property $R_{\mathbf{T}^*}$ of being the Gödel number of a member of $\mathbf{T}^*$ this way:

$$R_{\mathbf{T}^*}(x) \text{ iff } R_{\mathbf{T}}(\ulcorner (\varphi_1 \wedge \dots \wedge \varphi_n) \rightarrow \urcorner \wedge x \wedge \ulcorner \urcorner)$$
- (8) If $R_{\mathbf{T}}$ is a recursive property, then this is a recursive definition and $R_{\mathbf{T}^*}$ is recursive as well.
- (9) In other words, if $\mathbf{T}$ is recursively decidable, so is $\mathbf{T}^*$.

Proof: Suppose for *reductio* that $\mathbf{FOL}$ is recursively decidable. Then the property $R_{\mathbf{FOL}}$ of being the Gödel number of a member of $\mathbf{FOL}$ is as well. The property of being the Gödel number of a sentence of the language of arithmetic, $\text{Sent}_{\mathcal{L}_A}$, is primitive recursive. The conjunction of any two recursive properties is recursive. But the conjunction of $R_{\mathbf{FOL}}$ and $\text{Sent}_{\mathcal{L}_A}$ is equivalent to the property of being the Gödel number of a member of $\mathbf{FOL}_{\mathcal{L}_A}$, which we have just shown, is not recursive. Hence, $\mathbf{FOL}$ is recursively undecidable. $\ast$

Corollary: *The set of logically valid sentences is not a recursive set of sentences.*

Proof: By the soundness and completeness results of our first unit, this set is precisely $\mathbf{FOL}$. $\ast$

Logic itself is undecidable

These results are due to Alonzo Church.

Definition: $\mathbf{FOL}_{\mathcal{L}_A}$ is the theory which is the closure of $\emptyset$ in the language of arithmetic $\mathcal{L}_A$.

Definition: $\mathbf{FOL}$ is the theory which is the closure of $\emptyset$ in a language using the full range of constants, function letters and predicates.

Result: $\mathbf{FOL}_{\mathcal{L}_A}$ is recursively undecidable.

Proof: $\mathbf{Q}$ is a finite extension of $\mathbf{FOL}_{\mathcal{L}_A}$, as it is axiomatized by the set $\mathbf{Q} = \emptyset \cup \{\mathbf{Q}_1, \dots, \mathbf{Q}_8\}$. Hence, if $\mathbf{FOL}_{\mathcal{L}_A}$ were recursively decidable, $\mathbf{Q}$ would be as well, but we know $\mathbf{Q}$ is not. $\ast$

Corollary: $\mathbf{FOL}$ is recursively undecidable.

Corollary: *If the Church-Turing thesis is true, there is no effective decision procedure for determining whether or not a formula is logically valid.*

Proof: If the Church-Turing thesis were true, then if there were such a procedure, it would have to be recursive, but we know the set of logically valid formulas is not a recursive set. $\ast$

Corollary: *If the Church-Turing thesis is true, there is no effective decision procedure for determining in general whether or not an argument is logically valid.*

Proof: If there were such a procedure for determining whether or not $\Gamma \models \varphi$ in general, it would apply also to the case $\emptyset \models \varphi$, but this is the same as determining whether or not φ is valid. $\ast$

HOMEWORK 32

Prove that if the Church-Turing thesis is true, there is no effective procedure for determining whether or not two sentences φ and ψ are logically equivalent. (This is the bane of my attempt to have a computer check my students' Phil 110 translation exercises.)