Incompleteness

First part: Representing relations by formulas

Our goal now is to prove the Gödel Incompleteness Theorems, and associated undecidability results. Recall that **TA** (True Arithmetic) is the set of all sentences in the vocabulary $\mathcal{L}_A = [0, s, +, \cdot\ ; =]$ which are true in the standard model. We will prove that **TA** is not a recursive set, and not r.e., and in fact it has no recursive set of axioms. In view of this, we will study a standard subset of **TA** known as Peano Arithmetic (or **PA**), which is the set of sentences which are consequences of the Peano Postulates. Gödel's Second Incompleteness Theorem states that the consistency of **PA** cannot be proved in **PA**, and can be generalized to apply to any theory which can formalize a sufficient amount of number theory.

The undecidability and incompleteness results very much depend on the richness of the vocabulary $\mathcal{L}_A$; that is, both + and $\cdot$ must be present. As indicated on page 52 of the Notes, if just + is present, then the set of true sentences (Presburger Arithmetic) is decidable and has a nice axiomatization.

Notation: From now until the end of the course, the underlying vocabulary is $\mathcal{L} = \mathcal{L}_A = [0, s, +, \cdot\ ; =]$ (unless otherwise noted).

Recall that $\underline{\mathbb{N}}$ is the standard model or structure for $\mathcal{L}_A$. That is, the universe $M = \mathbb{N}$, and $0, s, +, \cdot$ get their standard meanings.

Representing relations by formulas

If $x_1, ..., x_n$ are distinct variables, and A is a formula, we will sometimes write $A(x_1, ..., x_n)$ (or $A(\vec{x})$) to indicate that we are thinking of A as representing a relation whose arguments are $x_1, ..., x_n$. In this case, if $t_1, ..., t_n$ are terms, then $A(t_1, ..., t_n)$ denotes A with the variables $x_1, ..., x_n$ simultaneously replaced by $t_1, ..., t_n$, respectively.

Numerals: We define $s_0 = 0$ and $s_{k+1} = ss_k$, $k = 0, 1, ...$

s_k is a term (or numeral) representing $k \in \mathbb{N}$. For example, s_3 stands for the term sss0. Numerals are syntactic objects. They represent numbers, which are semantic objects.

$A(s_{\vec{a}})$ means $A(s_{a_1}, \ldots, s_{a_n})$ where $a_1, \ldots, a_n \in \mathbb{N}$.

Definition: Suppose R is an n-ary relation, $A(\vec{x})$ is a formula such that all free variables in A are among $x_1, ..., x_n$. Then $A(\vec{x})$ *represents* R iff for all $\vec{a} \in \mathbb{N}^n$

$$R(\vec{a}) \Leftrightarrow \underline{\mathbb{N}} \models A(s_{\vec{a}})$$

($R(\vec{a})$ holds iff the sentence $A(s_{\vec{a}})$ is true in the standard model.)

This notion ties together a syntactic object (a formula A) and a semantic object (a relation R).

Definition: R is *arithmetical* iff R is representable by some formula (with vocabulary $\mathcal{L}_A$).

For example, the divisibility relation $x|y$ (x divides y) is representable by the formula $A(x,y) =_{syn} \exists z(x \cdot z = y)$. Therefore $x|y$ is an arithmetical relation.

We will show that many relations are arithmetical, including all recursive relations, all r.e. relations, and many more.

Bounded Quantifiers

Syntactic Definitions: Let t_1 and t_2 be terms.

$t_1 \leq t_2$ stands for $\exists z(t_1 + z = t_2)$, where z does not occur in t_1, t_2.

$\exists x \leq t\ A$ stands for $\exists x(x \leq t \wedge A)$, where x does not occur in t.

$\forall x \leq t\ A$ stands for $\forall x(x \leq t \supset A)$, where x does not occur in t.

These are *bounded quantifiers.* Note that these definitions apply to formulas in the vocabulary $\mathcal{L}_A$.

Notation: Let $\mathcal{L}_{A,\leq}$ be the vocabulary $\mathcal{L}_A$ expanded by the binary predicate symbol $\leq$. We define bounded quantifiers for this vocabulary as above, except now $x \leq t$ is not an abbreviation for $\exists z(x + z = t)$.

Definition of Bounded Formula and Δ_0 Formula: A formula A in $\mathcal{L}_{A,\leq}$ is a *bounded* formula iff all of its quantifiers are bounded. A formula A in $\mathcal{L}_A$ is *bounded* iff it is the translation of a bounded formula in $\mathcal{L}_{A,\leq}$ using the translation for $t_1 \leq t_2$ given above. A bounded formula of $\mathcal{L}_A$ is also called a Δ_0 formula.

Again "formula" always refers to a formula over $\mathcal{L}_A$, unless otherwise stated. Thus for example if we write a formula

$$\exists u \leq y(u \cdot x = y)$$

this stands for the $\mathcal{L}_A$ formula

$$\exists u((\exists z\ u + z = y) \wedge u \cdot x = y)$$

Definition: $R(\vec{x})$ is a Δ_0-relation iff some Δ_0 formula A represents R.

Note that all Δ_0 relations are arithmetical.

Example: The relation Prime(x) is represented by the following bounded formula $A(x)$:

$$s0 < x \wedge \forall z \leq x\ \forall y \leq x(x = z \cdot y \supset (z = 1 \vee z = x))$$

Thus Prime(x) is a Δ_0 relation.

Example: The relation $x|y$ is a Δ_0 relation.

Side Remark: All Δ_0 relations can be recognized in linear space on a Turing machine, when input numbers are represented in binary notation.

Lemma: The Δ_0 relations are closed under $\wedge, \vee, \neg$ and the bounded quantifiers $\forall \leq, \exists \leq$.

Proof: Notice that in this lemma, the operations in question are semantic operations, since they operate on relations (semantic objects).

However each of these semantic operations on relations corresponds to a syntactic operation on formulas. For example, suppose that R and S are n-ary Δ_0 relations. Then by definition of Δ_0, there are bounded formulas A and B which represent R and S, respectively. Then the formula $(A \wedge B)$ is a bounded formula which represents the relation $R \wedge S$. Therefore $R \wedge S$ is a Δ_0 relation. A similar argument applies to each of the other operations mentioned in the lemma.

Lemma: Every Δ_0 relation is recursive.

Proof: Structural Induction on bounded formulas in the vocabulary $\mathcal{L}_{A,\leq}$. We use the fact that the recursive relations (i.e. predicates) are closed under the boolean operations and bounded quantification. □

Remark: The converse of the above lemma is false, as can be shown by a diagonal argument. For those familiar with complexity theory, we can clarify things as follows. As noted in the Side Remark above, all Δ_0 relations can be recognized in linear space on a Turing machine. On the other hand, it is not hard to see that all , space $O(n^2)$ relations are primitive recursive. A straightforward diagonal argument shows that there are relations recognizable in n^2 space which are not recognizable in linear space, and hence are not Δ_0 relations.

Definition: A $\exists\Delta_0$ formula (also called a Σ_1 formula) is one of the form $\exists y A$, where A is a Δ_0 formula.

Definition: R is a $\exists\Delta_0$-relation iff R is represented by a $\exists\Delta_0$ formula.

Notice that we are applying the same adjective "$\exists\Delta_0$" to both relations and formulas. Of course all $\exists\Delta_0$ relations are arithmetical.

Theorem: Every $\exists\Delta_0$ relation is r.e.

Proof: Suppose that $R(\vec{x})$ is a $\exists\Delta_0$ relation. Then R is represented by a formula $\exists y A(\vec{x}, y)$, where $A(\vec{x}, y)$ is a bounded formula. Then A represents a Δ_0 relation $S(\vec{x}, y)$, such that $R(\vec{x}) = \exists y S(\vec{x}, y)$. By the previous lemma, S is recursive, and therefore R is r.e., by the definition of r.e. □

The converse is also true, so that in fact the $\exists\Delta_0$ relations coincide with the r.e. relations.

Exists Delta Theorem: Every r.e. relation is $\exists\Delta_0$.

The proof will take the next three pages. This is our easy analog of the much more difficult MRDP theorem stating that every r.e. relation is Diophantine (see page 81).

Unbounded quantifiers: We defined the Boolean operations $\wedge, \vee, \neg$ and the bounded quantifier operations $\forall \leq$ and $\exists \leq$. Now we defined the (unbounded) quantifier operations $\forall$ and $\exists$. Note that these are operations on relations as opposed to formulas, and hence they are semantic rather than syntactic operations.

Definition: The relation $S(\vec{x})$ is obtained from $R(\vec{x}, y)$ by the operation $\exists$ (existential quantification) if

$$S(\vec{x}) = \exists y R(\vec{x}, y), \text{ for all } \vec{x} \in \mathbb{N}^n$$

Similarly $S(\vec{x})$ is obtained from $R(\vec{x}, y)$ by the operation $\forall$ (universal quantification) if

$$S(\vec{x}) = \forall y R(\vec{x}, y), \text{ for all } \vec{x} \in \mathbb{N}^n$$

Note that the class of recursive relations is not closed under either of the operations $\exists, \forall$. For example, the Kleene T-predicate $T(z, x, y)$ is recursive, but K is not recursive, and yet

$$x \in K \Longleftrightarrow \exists y T(x, x, y),$$

where T is the following recursive predicate. $T(z, x, y)$ halts and outputs 1 if y codes the complete tableaux of the Turing machine encoded by z on input x, and the final configuration in y halts and outputs 1.

Closure Lemma: The $\exists\Delta_0$ relations are closed under $\exists$, $\wedge$ and $\vee$, and the bounded quantifiers $\exists \leq$ and $\forall \leq$.

Proof: Again note that these operations are semantic operations. Consider the operation $\exists$, for example. Suppose $R(\vec{x}, y)$ is represented by the formula $\exists z A(\vec{x}, y, z)$, where A is a bounded formula. Then $\exists y R(\vec{x}, y)$ is represented by the $\exists\Delta_0$ formula

$$\exists u(\exists y \leq u \exists z \leq u \; A(\vec{x}, y, z))$$

The argument in similar for the other operations. The case of $\forall \leq$ is interesting, but still quite similar.

Exercise 1 *Carry out the proof of the Closure Lemma for the other operations.*

Remark: We cannot extend the above Lemma to the operations $\forall$ and $\neg$. This is because the $\exists\Delta_0$ relations coincide with the r.e. relations (by the previous two theorems). We know that the r.e. relations are not closed under $\neg$, because K^c is not r.e.

Exercise 2 *Prove that the r.e. relations are not closed under* $\forall$.

Recall from page 79 that if f is an n-ary function, then graph(f) is the $n+1$-ary relation

$$R(\vec{x}, y) = (y = f(\vec{x}))$$

Main Lemma: If f a total computable function, then graph(f) is an $\exists\Delta_0$ relation.

Example: The relation $(y = 2^x)$ is $\exists\Delta_0$.

Proof of Exists Delta Theorem from Main Lemma: From this lemma it follows trivially that every primitive recursive relation is a $\exists\Delta_0$ relation, since

$$R(\vec{x}) \Leftrightarrow (R(\vec{x}) = 1)$$

where on the right, we view R as a 0-1 valued function.

Now we can show that every r.e. relation is $\exists\Delta_0$. Recall that one of our characterizations of r.e. relation was $R(\vec{x}) = \exists y S(\vec{x}, y)$, where S is recursive. We know that S is $\exists\Delta_0$ by the paragraph above, and thus R is $\exists\Delta_0$ by the Closure Lemma. □

Proof of Main Lemma:

We need a new idea: The Godel β function. This function provides us with a way of representing sequences of numbers by numbers, using $\exists\Delta_0$ formulas. Note that prime-power decomposition does not help us here, since it is not clear that the relation $z = x^y$ can be represented by a formula in the language of arithmetic, which does not include exponentiation as a built-in function.

Definition: (Gödel β function)

$$\beta(c, d, i) = rm(c, d(i+1)+1)$$

Recall $rm(x, y) = x \bmod y$.

Lemma: (Gödel) For any $n, r_0, \ldots, r_n$ there exists c, d such that

$$\beta(c, d, i) = r_i \quad 0 \leq i \leq n$$

Thus the pair (c, d) represents the sequence $r_0, r_1, ..., r_n$ using β.

For the proof, we need

Chinese Remainder Theorem (CRT): Given $r_0, \ldots, r_n$ and $m_0, \ldots, m_n$ such that

$$0 \leq r_i < m_i \quad 0 \leq i \leq n \tag{1}$$

and

$$\gcd(m_i, m_j) = 1 \quad 0 \leq i < j \leq n$$

there exists r such that

$$rm(r, m_i) = r_i \quad 0 \leq i \leq n$$

Proof: of CRT is by counting: Distinct values of r, $0 \leq r < \Pi m_i$, represent distinct sequences. But the total number of sequences $r_0, ..., r_n$ such that (1) holds is Πm_i. Hence every such sequence must be the sequence of remainders of some $r, 0 \leq r < \Pi m_i$. $\Box$

Proof of Gödel Lemma: Let $d = (n + r_0 + \cdots + r_n + 1)!$
Let $m_i = d(i+1) + 1$
Claim: $0 \leq i < j \leq n \Rightarrow \gcd(m_i, m_j) = 1$
For suppose p is prime, and $p \mid m_i$ and $p \mid m_j$
Then $p \mid d(i+1) + 1$ and $p \mid d(j+1) + 1$. Hence p divides their difference, i.e. $p \mid d(j-i)$. But p cannot divide d and $(d(i+1)+1)$ both, so $p \mid j - i$. But then $p \leq j - i < n$, so $p \mid d$, a contradiction.

By the CRT, there is a number $r = c$ so

$$\beta(c, d, i) = rm(c, m_i) = r_i, \ 0 \leq i \leq n \ \ \Box$$

Lemma 1: graph(β) is a Δ_0 relation.

Proof:

$$(y = \beta(c, d, i)) = [\exists q \leq c(c = q(d(i+1)+1) + y) \wedge y < d(i+1) + 1]$$

Lemma 2: If $R(\vec{x}, y)$ is a $\exists\Delta_0$ relation, graph(f) is a $\exists\Delta_0$ relation (where f is a total function), and $S(\vec{x}) = R(\vec{x}, f(\vec{x}))$, then S is a $\exists\Delta_0$ relation.

Proof:

$$S(\vec{x}) = \exists y(y = f(\vec{x}) \wedge R(\vec{x}, y))$$

We are now ready to prove the Main Lemma. We will assume that f is a unary total, computable function. (The function does not have to be unary, but this will slightly simplify our argument.) Recall that $Graph(f)(x, y) = 1$ if and only if $f(x) = y$. Since f is total computable, there exists a Turing machine, M_f that always halts and outputs $f(x) = y$ on input x, for all x. We will describe a $\exists\Delta_0$ relation, $R(x, y)$ at a high level. $R(x, y)$ says that there exists m, c, d such that four conditions hold. Intuitively m is the number of steps of the computation of M_f on x, and c, d describe the tableaux given by $r_1, \ldots, r_{m^2}$, via the Godel beta function. The first condition says that the first m numbers $r_1, \ldots, r_m$ encode the start configuration of M_f on x; The second condition says that the last m numbers $r_{(m-1)m}, \ldots, r_{m^2}$ encode the last configuration, which contains y in the first $|y|$ cells, and the state is the halt state q_2; The third condition states that for all configurations other than the last one, the state is not the halt state q_2; And finally, the last condition states that all 2-by-3 squares of cells, $(r_i, r_{i+1}, r_{i+2}, r_{i+m}, r_{i+2+m)})$ are consistent with the transition function of M_f.

All four conditions above are easily described by Δ_0 formula because all quantifiers are bounded (by m^2). They crucially rely on the Godel β function and Lemma 1. The last condition checks the computation locally, and it is not hard to prove that this sequence of local checks is satisfied if and only if $\vec{r}$ is a valid tableaux of M_f on input x.

This completes the proof (sketch) of the Main Lemma, that every primitive recursive function has a $\exists\Delta_0$ graph, and of the Exists Delta Theorem. $\Box$

Exercise 3 *Fill in the details of the above argument by describing the Δ_0 formulas for each of the four conditions described above.*

Exercise 4 *Give a formula $A(x, y)$ which represents the relation $(y = 2^x)$. Your presentation of $A(x, y)$ may use a formula $B(c, d, i, y)$ representing the graph of the Gödel β function $(y = \beta(c, d, i))$.*

Corollary to Exists Delta Theorem: Every r.e. relation is arithmetical (i.e. representable: see page 67).

Notice that *not* all arithmetical relations are r.e., since the arithmetical relations are closed under $\forall$ and $\neg$, unlike the r.e. relations. For example, K^c is arithmetical, but not r.e.

It follows from the Corollary that the set **TA** cannot be recursive or r.e. For example, K is r.e., so there is some formula $A(x)$ which represents K in **TA**. Thus

$$n \in K^c \Longleftrightarrow \neg A(s_n) \in \mathbf{TA}$$

If **TA** were r.e., it would follow that K^c is r.e., which yields a contradiction. In fact, this argument shows that even the set of $\exists\Delta_0$ sentences of **TA** is not recursive.

In the next section we prove Tarski's Theorem, which is a much stronger statement about the complexity of **TA**.

Exercise 5 Definition: *f is a Δ_0-function provided that f is a total n-ary function for some n, and*
(i) graph(f) is a Δ_0 relation, and
(ii) For some polynomial $p(\vec{x})$ with coefficients in $\mathbb{N}$,

$$f(\vec{x}) \leq p(\vec{x}) \text{ for all } \vec{x} \in \mathbb{N}^n$$

(a) Show that the Gödel β function $\beta(c, d, i)$ are Δ_0 functions.

(b) Show that the class of Δ_0 functions is closed under composition (as defined in the Notes, page 57).

TARSKI'S THEOREM

Tarski's theorem states that truth of sentences in the vocabulary $\mathcal{L}_\mathcal{A}$ cannot be expressed by any one formula $A(x)$ in $\mathcal{L}_\mathcal{A}$. This is made precise using the notion of arithmetical relation.

As a corollary to Tarski's Theorem we get a weak form of the Gödel Incompleteness Theorem: **TA** has no recursive set of axioms. (See Corollary 2, page 77.)

We have just shown that all r.e. relations are arithmetical. We now point out some easy closure properties of the set of arithmetical relations.

Lemma: The set of arithmetical relations is closed under the Boolean operations $\wedge, \vee, \neg$, and the quantifiers (bounded and unbounded) $\forall \leq, \exists \leq, \forall, \exists$.

Proof: The (easy) proof is essentially the same as for the corresponding lemma for the Δ_0 relations.

Exercise 6 *Show that the set of arithmetical relations is closed under substitution of total computable functions for variables.*

Assigning numbers to formulas: We assign a "Gödel" number $\#t$ to each term t and a Gödel number $\#A$ to each formula A in the same manner that we assigned numbers to Turing machines in the section on computability. The exact details of the assignment are not important, as long as there are algorithms which can go from terms and formulas to their numbers and from numbers to the terms and formulas that they represent.

Thus we can think of a set of sentences as a set of numbers:

Definition: If Γ is a set of sentences, then $\hat{\Gamma} = \{\#A \mid A \in \Gamma\}$.

We say that Γ is recursive, r.e., arithmetical, etc iff $\hat{\Gamma}$ is recursive, r.e., arithmetical, etc.

Theorem: (Tarski) **TA** is not arithmetical. More precisely, if we define the relation Truth by

$$\text{Truth}(m) \Leftrightarrow m = \#A, \text{ for some } A \in TA$$

Then Truth is not arithmetical.

Proof: We show that if Truth were arithmetical, then we could formulate the self-contradictory sentence "I am false". This idea is based on the liar paradox. The underlying technique is to get sentences in the vocabulary $\mathcal{L}_{\mathcal{A}}$ to refer to themselves. This idea is due to Gödel.

Gödel's method is to use the substitution function:

$$\text{sub}(m, n) = \begin{cases} \#A(s_n) \text{ if } \#A(x) = m \\ 0 \text{ if } m \text{ is not the number of any formula} \end{cases}$$

Lemma: The function sub is computable.

For the proof, we note that sub is clearly computable by an algorithm, so it is computable, by Church's Thesis. □

We define the "diagonal function" $d(n)$ by

$$d(n) = \text{sub}(n, n)$$

Thus $d(n) = \#A(s_n)$, where $\#A(x) = n$. Then d is a computable function.

Now suppose, contrary to Tarski's Theorem, that Truth is arithmetical. Define the Relation

$$R(x) = \neg\text{Truth}(d(x))$$

Then by the Lemma and Exercise above, R is an arithmetical relation. Say $A(x)$ represents $R(x)$, and $\#A(x) = e$. Then

$$d(e) = \#A(s_e)$$

Thus intuitively $A(s_e)$ says "I am false". In fact,
$A(s_e) \in TA \Leftrightarrow \neg\text{Truth}(d(e))$ (because A represents R)
$\Leftrightarrow A(s_e) \notin TA$ (def'n of Truth)
This is a contradiction, so Truth is not arithmetical. □

It follows from Tarski's theorem that the true sentences of arithmetic are not recursive, not r.e., not co-r.e., etc. In other words, they are wildly noncomputable.

Exercise 7 *Show using Church's Thesis that the set of true Δ_0 sentences is recursive, and therefore arithmetical. (Just give an informal algorithm.) Show the the set of true $\exists\Delta_0$ sentences is r.e., and therefore arithmetical.*

Arithmetic Hierarchy

For $k \geq 1$ we define a Σ_k formula to be one of the form

$$\exists y_1 \forall y_2 \exists y_3 \cdots Q y_k A(\vec{x}, y_1, \ldots, y_k)$$

where Q is $\exists$ if k is odd and Q is $\forall$ if k is even, and A is a Δ_0 formula.

Thus a Σ_1 formula is the same as an $\exists\Delta_0$ formula, and a Σ_2 formula has the form

$$\exists y \forall z A(\vec{x}, y, z)$$

We define Σ_k to be the set of relations $R(\vec{x})$ such that $R(\vec{x})$ is represented by a Σ_k formula.

Thus Σ_1 is the set of r.e. relations. It turns out that the sequence $\Sigma_1, \Sigma_2, \ldots$ forms a strict hierarchy of sets of relations:

$$\Sigma_1 \subsetneq \Sigma_2 \subsetneq \Sigma_3 \subsetneq \cdots$$

This is called the *arithmetic hierarchy.* Strictness can be proved by a diagonal argument, using the fact that for each $k \geq 1$, there is a binary relation $U_k(z, x)$ which is universal for all unary Σ_k relations. For example the r.e. relation

$$U_1(z, x) = \exists y T(z, x, y)$$

is universal for the set of unary r.e. relations.

The union $\bigcup_k \Sigma_k$ is the set of arithmetical relations.

$$\Sigma_1 \subset \Sigma_2 \subset \cdots$$

where Σ_1 is the set of r.e. sets and in general Σ_i is the set of all relations representable by $\exists\forall\cdots\Delta_0$ formulas; i.e. formulas which begin with i quantifiers starting with $\exists$ and alternating between $\exists$ and $\forall$, followed by a Δ_0 formula. Then the unions $\bigcup_i \Sigma_i$ is the set of all arithmetical relations.

Theories

Notation: Φ_0 denotes the set of $\mathcal{L}_A$-sentences (no free variables).

Thus **TA**$= \{A \in \Phi_0 \ : \ \underline{\mathbb{N}} \models A\}$. **TA** stands for *True Arithmetic*, the set of all true sentences in the language of arithmetic.

Definition: A *theory* is a set Σ of sentences closed under logical consequence. That is, if A is a sentence and $\Sigma \models A$ then $A \in \Sigma$.

Notation: If Σ is a theory, we often write $\Sigma \vdash A$ (read "Σ proves A") for $A \in \Sigma$. This is consistent with the notation $\Phi \vdash A$ introduced on page 47 in the context of LK proofs. It is perhaps more appropriate when the theory Σ is axiomatizable, but we will use this notation for any theory.

Since our underlying vocabulary is $\mathcal{L}_A$, we may assume (for this part of the Notes) that $\Sigma \subseteq \Phi_0$, for every theory Σ.

Definitions concerning a theory Σ

Σ is *consistent* iff $\Sigma \neq \Phi_0$

Σ is *complete* iff Σ is consistent, and for all sentences A either $\Sigma \vdash A$ or $\Sigma \vdash \neg A$.

Fact: Σ is consistent iff for all $A \in \Phi_0$, not both $A \in \Sigma$ and $\neg A \in \Sigma$. (Observe that for all $A, B \in \Phi_0$, $\quad \{A, \neg A\} \ \models \ B$.) Thus Σ is complete iff for all sentences A, exactly one of $\Sigma \vdash A$ and $\Sigma \vdash \neg A$ holds.

Exercise 8 *Prove that a theory Σ is consistent iff Σ has a model.*

Notation: If $\mathcal{M}$ is a structure over the language $\mathcal{L}$, then $Th(\mathcal{M})$ (the theory of $\mathcal{M}$) is the set of all sentences A such that $\mathcal{M} \models A$.

Exercise 9 *Prove that $Th(\mathcal{M})$ is a complete theory, for every structure $\mathcal{M}$.*

For example, **TA** $= Th(\underline{\mathbb{N}})$, so **TA** is a complete theory.

Definition: Σ is *sound* iff $\Sigma \subseteq \mathbf{TA}$.

In other words, Σ is sound iff all of its sentences are true in the standard model.

Thus **TA** is a theory which is complete, consistent and sound.

However a consistent theory need not be sound. For example the set of logical consequences of $\forall x \forall y(x = y)$ is consistent, because it has a model with a single-element universe, but it is not sound.

Notation: $VALID = \{A \in \Phi_0 : \ \models A\}$.

Thus $VALID$ is the set of valid sentences of $\mathcal{L}_A$. $VALID$ is a theory which is sound and consistent, but not complete. There are lots of sentences for which neither they nor their negation is valid. For example, $0 = 1 \notin VALID$ and $\neg 0 = 1 \notin VALID$.

$VALID$ is the *smallest theory.* That is, $VALID \subseteq \Sigma$ for all theories Σ.

Axiomatizable Theories

Definition: If Σ is a theory and $\Gamma \subseteq \Sigma$ then Γ is a set of *axioms* for Σ iff 1) Γ is recursive and 2) $\Gamma \models A$ for all $A \in \Sigma$. We say Σ is *axiomatizable* iff Σ has a set of axioms.

Theorem: A theory Σ is axiomatizable iff Σ is r.e.

Proof: $\Leftarrow$: The right-to-left direction is not so interesting, and is proved by a simple trick:

Suppose Σ is r.e. Then by a previous Lemma characterizing r.e. sets, $\hat{\Sigma} = \text{ran}(f)$, where f is a total, computable function of one variable. Thus $\hat{\Sigma} = \{f(0), f(1), \ldots\}$.

Let $A_n =$ be the sentence s.t. $\# A_n = f(n)$. Then $\Sigma = \{A_0, A_1, \ldots\}$, and this is an effective enumeration of Σ.

What is the set Γ of axioms? Let $B_n = A_0 \wedge A_1 \wedge \cdots \wedge A_n$ (with associativity to the left). Thus $B_n \in \Sigma$. (Why?) Let

$$\Gamma = \{B_0, B_1, B_2 \cdots\}$$

Claim: Γ is a set of axioms for Σ.

Condition 2) in the definition is obvious since $A_0 \wedge A_1 \wedge \cdots \wedge A_n \models A_n$

To demonstrate condition 1) (Γ is recursive) we need an algorithm to check whether a given formula C is in Γ. First C should be syntactically a conjunction of subformulas, say $C = C_0 \wedge C_1 \wedge \cdots \wedge C_m$ for some m. Now enumerate the first $m + 1$ formulas A_i, and check whether $A_i = C_i$, $i = 0, ..., m$.

$\Rightarrow$ The left-to-right direction of the Theorem is more interesting. Assume Σ is axiomatizable, and let Γ be a set of axioms for Σ. Then Γ is recursive, $\Gamma \subseteq \Sigma$, and $\Sigma = \{A \mid \Gamma \models A\}$. To show Σ is r.e. we show how to effectively enumerate it, i.e. we show how to enumerate the logical consequences of Γ.

For this we use the completeness theorem for LK (and compactness). The idea is that we enumerate all possible LK proofs for sentences in the vocabulary of arithmetic, and for each one check whether it is a proof of the form

$$B_1, ..., B_k \vdash A$$

where each B_i is a sentence in Γ. If so, then we output A.

This argument can be made more formal as follows: First define the (semantic) relation $P(a, b)$ by the condition

$$P(a, b) \Leftrightarrow b \text{ is the number of a } LK \text{ proof that } A \text{ is valid, where } \#A = a$$

Clearly there is an algorithm which, given a and b, checks whether $P(a, b)$ holds. Therefore P is recursive, by Church's Thesis.

Now define $Q(a, b)$ by

$$Q(a, b) \Leftrightarrow [b = \#(\neg B_1 \vee \cdots \vee \neg B_k \vee A) \text{ where } \#A = a \text{ and } B_1, ..., B_k \in \Gamma]$$

Again Q is recursive, by Church's thesis. (Recall that Γ is recursive.)

Note that
$A \in \Sigma$
$\Leftrightarrow \Gamma \models A$
$\Leftrightarrow \exists k \exists B_1 \cdots B_k \in \Gamma$ such that $(\neg B_1 \vee \cdots \vee \neg B_k \vee A)$ is valid.

(The last equivalence uses the Compactness Theorem.) Thus

$$a \in \hat{\Sigma} \Leftrightarrow \exists b \exists p \underbrace{[P(b, p) \wedge Q(a, b)]}_{\text{recursive}}$$

Thus $\hat{\Sigma}$ is r.e. □

Corollary 1: $VALID$ is r.e., where $VALID$ is the set of valid sentences (page 76).

Proof: $VALID$ can be axiomatized by the empty set of axioms, and the empty set is recursive.

Remark: Later we will show that $VALID$ is not recursive. It follows that the set of nonvalid sentences is not r.e. (why?). Hence the set of satisfiable sentences of $\mathcal{L}_A$ is not r.e., since A is nonvalid iff $\neg A$ is satisfiable, so the set of nonvalid sentences is many-one reducible to the set of satisfiable sentences. On the other hand, the set of unsatisfiable sentences is r.e. (why?).

Corollary 2: **TA** is not axiomatizable.

Proof: By Tarski's Theorem, **TA** is not arithmetical, so it is not r.e.

Corollary 3: Every sound axiomatizable theory is incomplete.

Proof: If Σ is sound then $\Sigma \subseteq \mathbf{TA}$, and if Σ axiomatizable, then $\Sigma \neq \mathbf{TA}$. So $\Sigma \subsetneq \mathbf{TA}$. Hence there is $A \in \mathbf{TA}$, (A is true) s.t. $A \notin \Sigma$. Also $\neg A \notin \Sigma$ because $\neg A$ is false. Hence Σ is incomplete.

These results are very robust. We just proved them for a specific vocabulary but let Σ' be any theory (not necessarily based on the vocabulary $[0, s, +, \cdot\ ; =]$). For example, Σ' could be Zermelo Fraenkel set theory with the axiom of choice (ZFC), which is strong enough to formalize all "ordinary" mathematics.

Assume that natural #'s can be defined in Σ'. In ZFC, this can be done as follows:

$$\emptyset = 0, \text{ and in general } n+1 = n \cup \{n\}$$

Assume we can define $0, s, +, \cdot$ on $\mathbb{N}$ in Σ' (we can in ZFC). Let $\mathbf{TA}'$ be the translation of $\mathbf{TA}$ to the new vocabulary. If we assume $\mathbf{TA}' \subseteq \Sigma'$, then Tarski's theorem still works. All notions of representable, arithmetical still apply. If Σ' is axiomatizable then the set of all theorems (i.e. Σ') is r.e. Also the set of number-theoretic theorems is r.e. Hence these theorems are a proper subset of $\mathbf{TA}'$.

In particular, there are sentences in $\mathbf{TA}$ whose translations into set theory are not theorems of ZFC.

Famous Conjectures:

Goldbach's conjecture: Every even integer is the sum of 2 primes

Riemann Hypothesis

$P \neq NP$

One can speculate that one of these might be true, but does not follow from the Zermelo-Fraenkel Axioms. (However it seems more likely that natural assertions like these will eventually either be proved or disproved in ZFC.)

Peano Arithmetic

Goals Now

1) We will introduce a standard set of axioms for the language $\mathcal{L}_A$. The theory generated by these axioms is denoted **PA** and called Peano Arithmetic. Since **PA** is a sound, axiomatizable theory, it follows by the corollaries to Tarski's Theorem that it is incomplete. Nevertheless, it appears to be strong enough to prove all of the standard results in the field of number theory (including such things as the prime number theorem, whose standard proofs use analysis). Even Andrew Wiles' proof of Fermat's Last Theorem has been claimed to be formalizable in **PA**.

2) We know that **PA** is sound and incomplete, so there are true sentences in the language $\mathcal{L}_A$ which are not theorems of **PA**. We will outline a proof of Gödel's Second Incompleteness Theorem, which states that a specific true sentence, asserting that **PA** is consistent, is not a theorem of **PA**. This theorem can be generalized to show that any consistent theory satisfying general conditions cannot prove its own consistency.

3) We will introduce a finitely axiomatized subtheory **RA** ("Robinson Arithmetic") of **PA** and prove that every consistent extension of **RA** (including **PA**) is "undecidable" (meaning not recursive). As corollaries, we get a stronger form of Gödel's first incompleteness theorem, as well as Church's Theorem: The set of valid sentences of $\mathcal{L}_A$ is not recursive.

The Theory PA (Peano Arithmetic)

The so-called Peano postulates for the natural numbers were introduced by Giuseppe Peano in 1889. In modern form they can be stated in the language of set theory as follows. Let $\mathbb{N}$ be a set containing an element 0, and let $S : \mathbb{N} \to \mathbb{N}$ be a function satisfying the following postulates:

GP1: $S(x) \neq 0$, for all $x \in \mathbb{N}$.
GP2: If $S(x) = S(y)$ then $x = y$, for all $x, y \in \mathbb{N}$.
GP3: Let A be any subset of $\mathbb{N}$ which contains 0 and which is closed under S (i.e. $S(x) \in A$ for all $x \in A$). Then $A = \mathbb{N}$.

Note that GP3 is a form of induction.

It is not hard to show that any two systems $\langle \mathbb{N}, S, 0\rangle$ and $\langle \mathbb{N}', S', 0'\rangle$ which both satisfy GP1,GP2, GP3 are isomorphic, in the sense that there is a bijection $\phi : \mathbb{N} \to \mathbb{N}'$ such that $\phi(0) = 0'$ and

$$\phi(S(x)) = S'(\phi(x)), \text{ for all } x \in \mathbb{N}$$

Thus the Peano postulates characterize $\mathbb{N}$ up to isomorphism.

However, when it comes to designing a formal theory in the predicate calculus based on these Peano postulates we cannot formulate GP3 except in the context of formal set theory. It turns out to be essentially impossible to formulate a completely satisfactory theory of sets.

One simple solution is to design a "first-order" theory of $\mathbb{N}$ in which the universe is supposed to be $\mathbb{N}$ and the underlying language is $[0, s; =]$. This was done on pages 49-50, and the result is a complete theory Th(s) which can be completely axiomatized. However this theory cannot formulate much of interest, because $+$ and $\cdot$ cannot be defined in this language.

Thus to formulate our theory **PA** we extend this simple language by adding $+$ and $\cdot$ to obtain the language $\mathcal{L}_A = [0, s, +, \cdot; =]$. In this language, postulates GP1 and GP2 are easily formulated. The best we can do to formulate GP3 is to represent sets by formulas $A(x)$ in the language $\mathcal{L}_A$, where $A(x)$ is supposed to represent the set $\{x \mid A(x)\}$. When this is done carefully, we come up with the Induction Scheme below.

In order to complete the axioms of **PA** we need recursive definitions of $+$ and $\cdot$. These are formulated below as P3, P4 for $+$ and P5,P6 for $\cdot$.

Axioms for PA

$$\begin{array}{ll}
\text{P1 } \forall x(sx \neq 0) & \\
\text{P2 } \forall x\forall y(sx = sy \supset x = y) & s \text{ is 1-1 function} \\
\left.\begin{array}{l}\text{P3 } \forall x(x + 0 = x) \\ \text{P4 } \forall x\forall y(x + sy = s(x + y))\end{array}\right\} & \text{define } + \\
\left.\begin{array}{l}\text{P5 } \forall x(x \cdot 0 = 0) \\ \text{P6 } \forall x\forall y(x \cdot sy = (x \cdot y) + x)\end{array}\right\} & \text{define } \cdot
\end{array}$$

Induction Scheme: Let $Ind(A(x))$ be the sentence

$$\forall y_1 \cdots \forall y_k[(A(0) \wedge \forall x(A(x) \supset A(sx))) \supset \forall x A(x)]$$

where A is any formula whose free variables are among $x, y_1, \cdots, y_k$. (The variables $y_1, \cdots, y_k$ are called parameters.) All such sentences $Ind(A)$ are axioms.

Let $\Gamma_{PA} = \{P_1, \ldots, P_6\} \cup \{\text{Induction axioms}\}$. Then Γ_{PA} is recursive. This is clear from Church's thesis.

Definition: **PA** $= \{A \in \Phi_0 \mid \Gamma_{PA} \models A\}$

Thus **PA** is an axiomatizable theory. It is a sound theory since all of its axioms (and hence all of its theorems) are true in the standard model $\underline{\mathbb{N}}$.

Terminology: We speak of sentences in **PA** as *theorems* of **PA**, because they can be proved (for example, by *LK* proofs), from the axioms of **PA**. We use the notation **PA** $\vdash A$ to mean that A is a theorem of **PA**.

Example 1:
We show that **PA** proves that all nonzero elements have predecessors. Let

$$A(x) \ = \ (x = 0 \vee \exists y(x = sy))$$

In order to prove this by induction there are two steps:

Basis: $x = 0$ **PA** $\vdash A(0)$

Induction Step: $z \leftarrow sz$ **PA** $\vdash \forall x(A(x) \supset A(sx))$

In fact, both $A(0)$ and $\forall x(A(x) \supset A(sx))$ are valid sentences, so no axioms of **PA** are needed to show that they are theorems of **PA**. It follows from the induction axiom $Ind(A(x))$ that

$$\textbf{PA} \vdash \forall x A(x)$$

Example 2:
We show that **PA** proves the associative law for $+$. Let

$$A(z) \ = \ (x + y) + z = x + (y + z)$$

We use the induction axiom $Ind(A(z))$.

Basis: $z = 0$

$$\begin{aligned} (x+y)+0 &= x+y && \text{P3} \\ &= x+(y+0) && \text{P3} \end{aligned}$$

Induction Step: $z \leftarrow sz$

$$\begin{aligned} (x+y)+sz &= s((x+y)+z) && \text{P4} \\ &= s(x+(y+z)) && \text{Induction Hypothesis} \\ &= x+s(y+z) && \text{P4} \\ &= x+(y+sz) && \text{P4} \end{aligned}$$

Thus by $Ind(A(z))$ it follows that

$$\mathbf{PA} \vdash \forall x \forall y \forall z A(z)$$

Exercise 10 *Show that* **PA** *proves the commutative law of addition, the associative and commutative laws of multiplication, and that multiplication distributes over addition, using the style of Example 2. In each case state carefully which induction axiom (or axioms) are needed, and which axioms P1,...,P6 are needed, (or which earlier results).*

Exercise 11 *Recall the theory of successor $Th(s)$ presented on pages 49-50. Show that all of the axioms S3, S4, S5, ... follow from S1 and S2 together with the Induction Scheme $Ind(A(x))$ for all formulas $A(x)$ in the language of successor* $[0, s; =]$.

PA is incomplete, because it is axiomatizable and sound (and has $\mathcal{L}_A$ as the underlying language): see Corollary 3, page 77. Later we will give explicit true sentences that are not theorems of **PA**, including the assertion that **PA** is consistent.

An apparent paradox is that the Peano postulates GP1, GP2, GP3 characterize the natural numbers in set theory (as explained above), and yet there are nonstandard models for **PA**. (We know there are nonstandard models both from the fact that **PA** is incomplete, and by the construction using compactness given on page 51.) However, the Peano Axioms only characterize the natural numbers under the assumption that we could do induction using an arbitrary set. In **PA**, we can only use induction on arithmetical sets.

Observed fact: All standard theorems of number theory are in **PA**. Even Wiles' 1995 proof of "Fermat's Last Theorem" apparently can be formalized in **PA**. So famous open problems, such as Goldbach's conjecture and the prime pair conjecture, can probably be either proved or disproved in **PA**. Goldbach's conjecture can certainly be disproved in **PA** if it is false: just present and verify a counter example. (Is the same true for the prime pair conjecture?)

RA: A finitely axiomatized subtheory of PA

Our main tool for showing that a theory such as **PA** is undecidable is showing that every r.e. relation (including the undecidable set K) is representable in the theory (see the definition below). This argument applies not only to **PA**, but to a weak subtheory of **PA** known as **RA**.

Recall the syntactic definition of $\leq$: $t_1 \leq t_2$ stands for $\exists z(t_1 + z = t_2)$, where z is a new variable.

We now extend P1,...,P6 with three new axioms.

P7 $\forall x(x \leq 0 \supset x = 0)$
P8 $\forall x \forall y(x \leq sy \supset (x \leq y \vee x = sy))$
P9 $\forall x \forall y(x \leq y \vee y \leq x)$

Definition: **RA** is the theory whose axioms are P1, $\cdots$, P6, P7, P8, P9.

Note that **RA** has no induction axioms. We note three important facts about **RA**:

1) **RA** $\subseteq$ **PA** (i.e. P7, P8, P9 are in **PA** because they can be proved by induction).

2) **RA** has only finitely many axioms.

2) The axioms of **RA** are $\forall$-sentences (over $\mathcal{L}_{A,\leq}$).

Later we will show that **RA** $\neq$ **PA**.

Definition: A theory Σ is *decidable* iff $\{\#A | A \in \Sigma\}$ is recursive.

Informally, Σ is decidable iff there is an algorithm which, given any sentence A, determines whether A is in Σ.

Definition: If Σ and Σ' are theories, then Σ' is an *extension* of Σ if $\Sigma \subseteq \Sigma'$.

We will show that **RA** is undecidable, and use this to prove that in fact every sound theory (over the language $\mathcal{L}_A$) is undecidable. Our main tool is the representation theorem below. Recall the definition (bottom of page 66) for a formula $A(\vec{x})$ to represent a relation $R(\vec{x})$. We now extend this definition to apply to a theory Σ.

Definition: A formula $A(\vec{x})$ *represents* a relation $R(\vec{x})$ in a theory Σ if for all $\vec{a} \in \mathbb{N}^n$

$$R(\vec{a}) \Leftrightarrow \Sigma \vdash A(s_{\vec{a}})$$

Note that according to our earlier definition, $A(\vec{x})$ represents $R(\vec{x})$ (with no theory mentioned) iff $A(\vec{x})$ represents $R(\vec{x})$ in **TA**.

Recall the definition (page 68) of a $\exists\Delta_0$ formula.

RA Representation Theorem: Every r.e. relation is representable in **RA** (and in every sound extension of **RA**) by an $\exists\Delta_0$ formula.

This is a major result and will take several pages to prove. Of course we already know from the Exists Delta Theorem (page 68) that every r.e. relation is representable in **TA**. The extra work now is showing that the true $\exists\Delta_0$ formulas are provable in **RA**.

Before giving the proof of the Theorem, we prove several consequences.

Corollary 1: Every sound extension of **RA** (including **PA**) is undecidable.

Proof: Let Σ be a sound extension of **RA**. It suffices to show $K \leq_m \Sigma$, or more precisely to show that $K \leq_m \hat{\Sigma}$, where $\hat{\Sigma}$ is the set of codes for theorems of Σ; that is $\hat{\Sigma} = \{\#A \mid \Sigma \vdash A\}$.

Since K is r.e., it follows from the theorem that K is represented in Σ by some $\exists\Delta_0$ formula $A(x)$. Thus for all $a \in \mathbb{N}$

$$a \in K \Leftrightarrow \Sigma \vdash A(s_a)$$

Define the total computable function $f : \mathbb{N} \to \mathbb{N}$ by

$$f(a) = \#A(s_a)$$

Then f is clearly computable by Church's thesis. Thus $a \in K \Leftrightarrow f(a) \in \hat{\Sigma}$. as required. □

Recall Corollary 1, page 77 states that the set $VALID$ of valid sentences of $\mathcal{L}_A$ is r.e. Now we can prove more:

Corollary 2: Church's Theorem: The set $VALID$ of valid sentences in the language $\mathcal{L}_A$ is undecidable.

Proof: We use the fact that **RA** is undecidable, and has only finitely many axioms, P1,...,P9. Let γ be the conjunction P1$\wedge$...$\wedge$P9 of these axioms. Then

$$A \in \mathbf{RA} \Longleftrightarrow (\gamma \supset A) \text{ is valid}$$

Hence we've reduced the problem of membership in **RA** to the validity problem, so validity is undecidable. (We've only given an informal argument for the reduction, so we need Church's thesis here.) □

Remark: In fact, the validity problem is undecidable for any language that contains a binary predicate symbol. This can proved directly by reduction of the halting problem for Turing machines to validity, as was shown in Turing's famous 1936 paper introducing Turing machines.

Decidability Theorem: Every complete axiomatizable theory is decidable.

Proof: We give an informal proof, using Church's thesis. If Σ is axiomatizable, then by the theorem on page 76, it is r.e. Here is an algorithm for determining whether a given formula A is in Σ, assuming that Σ is complete. Enumerate the members of Σ. Sooner or later, either A or $\neg A$ will appear in the enumeration. If A appears, then it is in Σ. If $\neg A$ appears, then A is not in Σ. □.

Now we can obtain an alternative proof of Corollary 3 to Tarski's Theorem, page 77:

Corollary: Every sound axiomatizable theory is incomplete.

Proof: Let Σ be a sound axiomatizable theory. If Σ is not an extension of **RA** it is certainly incomplete. If Σ is an extension of **RA**, then by Corollary 1 above Σ is undecidable, and hence by the Decidability Theorem Σ is incomplete. □

Exercise 12 *Prove that there is an $\exists\Delta_0$ sentence A such that $\neg A \in$ **TA** but **PA** $\nvdash \neg A$. (Compare this with Corollary 2, page 87.)*

In order to prove the **RA** Representation Theorem we need to recall the syntactic definitions involving $\leq$ given on page 67.

MAIN LEMMA: Every bounded sentence in **TA** is in **RA**. That is, every true bounded sentence can be proved from the axioms of **RA**. (Thus $\mathbf{TA} \cap \Delta_0 = \mathbf{RA} \cap \Delta_0$.)

Notation: When we write a specific number such as 4 in an example formula, this is an abbreviation for the corresponding numeral; s_4 (i.e. $ssss0$) in this case.

Example of a true bounded sentence:

$$\forall x \leq 1000 \exists y \leq 2 \cdot x[x = 0 \vee (x < y \wedge \mathrm{Prime}(y))]$$

Notice that since the quantifiers are bounded, and the assertion is being made for only finitely many pairs x, y. Each case can be proved separately by "brute force".

To prove the MAIN LEMMA it is easier to expand the language $\mathcal{L}_A$ to $\mathcal{L}_{A,\leq}$ by adding the binary connective $\leq$ as a primitive symbol . Then we expand the theory **RA** to the theory $\mathbf{RA}_{\leq}$ over the language $\mathcal{L}_{A,\leq}$ by interpreting $\leq$ in the axioms P7,P8,P9 as a primitive symbol, and by adding the new axiom

P0 $\forall x \forall y(x \leq y \leftrightarrow \exists z(x + z = y))$

Every formula A over $\mathcal{L}_{A,\leq}$ can be translated to a formula A' over $\mathcal{L}_A$ by replacing each atomic subformula of the form $t_1 \leq t_2$ in A by the formula $\exists z(t_1 + z = t_2)$, where z is a variable not occurring in t_1, t_2 (see page 67). Notice that if $\leq$ does not occur in A, then $A =_{syn} A'$.

Translation Lemma: For every formula A over $\mathcal{L}_{A,\leq}$,

$$\mathbf{RA}_{\leq} \vdash A \text{ iff } \mathbf{RA} \vdash A'$$

Proof: There is a natural one-one correspondence between models of $\mathbf{RA}_{\leq}$ and **RA**, namely for each model $\mathcal{M}$ of **RA** we associate the model $\hat{\mathcal{M}}$ of $\mathbf{RA}_{\leq}$ which is the same as $\mathcal{M}$ except we add the interpretation of $\leq$ in such a way that axiom P0 is satisfied. Then we claim that for every $\mathcal{L}_{A,\leq}$ formula A

$$\hat{\mathcal{M}} \models A \text{ iff } \mathcal{M} \models A'$$

The claim is easily proved by structural induction on A. The Translation Lemma follows easily from the claim. □

Proof of MAIN LEMMA: We prove the MAIN LEMMA for $\mathbf{RA}_{\leq}$. It follows for **RA** by the Translation Lemma.

Let A be a true bounded sentence. Move all $\neg$'s in A past other connectives so that they govern only atomic formulas $t = u$. Do this by using DeMorgan's Laws, and the equivalences

$$\neg\neg A \iff A, \quad \neg\forall x \leq t\ B \iff \exists x \leq t \neg B, \quad \neg\exists x \leq t\ B \iff \forall x \leq t \neg B$$

Exercise 13 *Show from the definitions of the bounded quantifiers $\exists x \leq t$ and $\forall x \leq t$ that for each of the three equivalences above the formulas on the left and right are logically equivalent (this is obvious for the first equivalence).*

The proof of the MAIN LEMMA proceeds by induction on the number of logical operators (other than $\neg$) in this modified A.

For the base case, A has one of the four forms $t = u$, $t \neq u$, $t \leq u$, $\neg t \leq u$.

Example: A is $s0 + s0 = ss0$. This can be proved in **RA** by the recursive definition of +:

$x + 0 = x$ (P3)
$x + sy = s(x + y)$ (P4)

More generally:

Lemma A1: For all $m, n \in \mathbb{N}$,

$$\mathbf{RA} \vdash s_m + s_n = s_{m+n} \text{ and}$$

$$\mathbf{RA} \vdash s_m \cdot s_n = s_{m \cdot n}$$

Proof: The first line is proved by induction (outside the system) on n using P3 and P4, as in the example. Then the second line is proved by induction on n using P5, P6, and the first line. □

If t is any closed term (i.e. with no variables), then $t^{\mathcal{M}} = n$ for some $n \in \mathbb{N}$, where $\mathcal{M}$ is the standard model. Thus $t = s_n \in \mathbf{TA}$.

Lemma A: If t is a closed term and $t = s_n$ is in **TA**, then $\mathbf{RA} \vdash t = s_n$.

Proof: Induction on the length of t, using Lemma A1.

Lemma B: If $m < n$, then $\mathbf{RA} \vdash s_n \neq s_m$.

Proof: Induction on m, using P1 and P2. □

For example, consider $ss0 \neq s0$. Recall that P2 is $\forall x(sx = sy \supset x = y)$. Thus $ss0 = s0 \supset s0 = 0$. But by P1, $s0 \neq 0$. Therefore $ss0 \neq s0$.

Remark: Arguments such as the one above could be formalized by an LK proof using the equality axioms. However the implications are clear without bothering to carry out such a formal proof, if we keep in mind the definition of logical consequence (page 23), and the Basic Semantic Definition (page 22), and in particular that = must be interpreted as equality in any structure.

The base case for the MAIN LEMMA for the sentences $t = u$ and $t \neq u$ follows easily from Lemma A and Lemma B. For the case $t \leq u$ we apply P0, so the problem reduces to the first case of Lemma A1. The case $\neg t \leq u$ follows from Lemma C below, together with Lemma B.

The induction step for the MAIN LEMMA follows from the following:

Lemma C: For all n, $\mathbf{RA}_{\leq}$ proves the sentence

$$\forall x(x \leq s_n \supset (x = 0 \vee x = s_1 \vee ... \vee x = s_n))$$

Proof: Induction on n. The base case is $x \leq 0 \supset x = 0$, which is P7. The induction step follows easily from P8. □ (Lemma C)

For the induction step in the proof of the MAIN LEMMA, let A be a true bounded sentence. We assume that $\neg$'s in A have been driven in as explained above, and A does not fit the base case, so the principle connective of A is one of $\wedge$, $\vee$, $\forall \leq$, $\exists \leq$. The cases of $\wedge$ and $\vee$ are trivial: just apply the induction hypothesis.

Now consider the case $\forall \leq$, say A is $\forall x \leq tB(x)$, and this is in **TA**. Since this is a sentence, and by definition of $\forall x \leq t$, x cannot occur in t, it follows that t is a closed term. Thus by Lemma A, **RA** can prove $t = s_n$ for some n.

For example, suppose $n = 23$. Then it suffices to show that $\forall x \leq 23\ B(x)$ is provable in $\mathbf{RA}_{\leq}$. By Lemma C, $\mathbf{RA}_{\leq}$ proves

$$x \leq 23 \supset (x = 0 \vee x = 1 \vee \cdots \vee x = 23)$$

By the Substitution Theorem (page 26) it follows in general, that for any closed term u,

$$\forall x(x = u \supset (B(u) \leftrightarrow B(x)))$$

is valid. Therefore it follows by reasoning in $\mathbf{RA}_{\leq}$ that $\forall x \leq tB(x)$ is implied by

$$B(0) \wedge B(1) \wedge \cdots \wedge B(23)$$

Since $\forall x \leq tB(x)$ is true, it follows that $B(0), B(1), ...$ are each true, so by the induction hypothesis each is in $\mathbf{RA}_{\leq}$. Hence their conjunction is in $\mathbf{RA}_{\leq}$, so $\forall x \leq tB$ is in $\mathbf{RA}_{\leq}$.

The case $\exists \leq$ is easier than the $\forall \leq$ case and does not require Lemma C. □ (MAIN LEMMA)

Exercise 14 *Prove the* $\exists \leq$ *case in the above proof.*

Corollaries to MAIN LEMMA

Corollary 1: The set of bounded sentences of **TA** is decidable.

Corollary 2: Every $\exists\Delta_0$ sentence of **TA** is provable in **RA**.

Corollary 3: The set of $\exists\Delta_0$ sentences of **TA** is r.e. (but not decidable).

Exercise 15 *Prove the above three corollaries.*

Exercise 16 *Let* $\exists y A(x,y)$ *be a* $\exists\Delta_0$ *formula which represents* $K(x)$ *in* **RA** *(where* $K(x) = (\{x\}_1(x) \neq \infty)$ *is the standard halting problem). Show that there is a consistent extension* Σ *of* **RA** *such that* $\exists y A(x,y)$ *does not represent* $K(x)$ *in* Σ. **Hint:** *Form* Σ *by adding a suitable false axiom to* **RA** *which retains consistency.*

Proof of RA Representation Theorem: (See page 82 for the statement.)

Proof: Suppose $R(\vec{x})$ is an r.e. relation. By the Exists Delta Theorem $R(\vec{x})$ is represented in **TA** by some $\exists\Delta_0$ formula $\exists y A(\vec{x}, y)$. Thus for all $\vec{a} \in \mathbb{N}^n$,

$$R(\vec{a}) \Leftrightarrow [\exists y A(s_{a_1}, ..., s_{a_n}, y) \in TA]$$

By Corollary 2 above and the soundness of **RA**, this is equivalent to

$$R(\vec{a}) \Leftrightarrow [\Sigma \vdash \exists y A(s_{a_1}, ..., s_{a_n}, y)]$$

where Σ is any sound extension of **RA**(i.e. **RA** $\subseteq \Sigma \subseteq$ **TA**). Thus by the definition $\exists y A(\vec{x}, y)$ represents $R(\vec{x})$ in Σ. □

The following is a generalization of Church's Theorem (page 83).

Theorem: Every sound theory is undecidable.

Exercise 17 *Prove the theorem.*

Results for consistent (possibly unsound) theories

Our goal now is to prove the following theorem:

Main Theorem: Every consistent extension of **RA** is undecidable.

Corollary: Every consistent axiomatizable extension of **RA** is incomplete.

Proof of Corollary: This follows from the Decidability Theorem (page 83). □

Notice that this strengthens the Corollary 3, page 77, to Tarski's Theorem, since we no longer need to assume soundness in order to conclude that an axiomatizable theory is incomplete (provided that the theory includes **RA**). Notice that soundness is a semantic notion, whereas consistency can be given a syntactic definition (there is no proof of 0=1). The proof of the Main Theorem can be made to avoid the complex semantic notion of truth of an arbitrary sentence of $\mathcal{L}_A$.

An example of an unsound consistent extension of **RA** is the theory $Th(\mathbb{Z}[X]^+)$ consisting of all sentences in the language $\mathcal{L}_A$ which are true in the structure $\mathbb{Z}[X]^+$, where the universe of $\mathbb{Z}[X]^+$ is the set of all polynomials $p(X)$ with integer coefficients such that either $p(X)$ is the zero polynomial, or the leading coefficient of $p(X)$ is positive. (Here $+$ and $\cdot$ are polynomial addition and multiplication, and the successor of $p(X)$ is $p(X)+1$.) The axioms P1,...,P9 are in the theory $Th(\mathbb{Z}[X]^+)$, but the theory is unsound, because the sentence

$$A = \quad \exists x \forall y (x \neq y + y \wedge x \neq y + y + s0) \tag{2}$$

is not in **TA** but is in $Th(\mathbb{Z}[X]^+)$. (To check the latter claim, let x be the polynomial X.)

Thus $Th(\mathbb{Z}[X]^+)$ is undecidable, by the Main Theorem.

Corollary: RA$\neq$ PA

Proof: Let A be the sentence in (2) above. Then $\neg A$ is a theorem of **PA** (it can be proved by induction on x), but $\neg A$ is not a theorem of **RA**, since the structure $\mathbb{Z}[X]^+$ just described is a model of **RA** which satisfies A.

Exercise 18 *Is $Th(\mathbb{Z}[X]^+)$ axiomatizable? Justify your answer.*

Notice that the structure $\mathbb{Z}[X]^+$ is a nonstandard model of **RA**. There are no such nice nonstandard models of **PA**. In fact one can prove that for any nonstandard model of **PA** with universe $\mathbb{N}$, the interpretations of $+$ and $\cdot$ are uncomputable functions.

In order to prove the Main Theorem we need a stronger notion of representability.

Recall the definition of *represents in a theory* Σ (page 82):

A represents R in Σ iff $\forall \vec{a} \in \mathbb{N}^n$ $(R(\vec{a}) \Leftrightarrow A(s_{\vec{a}}) \in \Sigma)$.

Definition: *A strongly represents* R in Σ iff $\forall \vec{a} \in \mathbb{N}^n$
$R(\vec{a}) \Rightarrow (A(s_{\vec{a}}) \in \Sigma)$, and $\neg R(\vec{a}) \Rightarrow (\neg A(s_{\vec{a}}) \in \Sigma)$

Notice that if Σ is a consistent theory, then if $A(\vec{x})$ strongly represents $R(\vec{x})$ in Σ it follows that $A(\vec{x})$ also represents $R(\vec{x})$ in Σ. The converse is not always true (unless Σ is complete).

In order to prove the Main Theorem, we will prove the following two results:

Undecidability Theorem: If every recursive relation is representable in a theory Σ then Σ is undecidable.

Strong RA Representation Theorem: Every recursive relation is strongly representable in **RA** by an $\exists\Delta_0$ formula.

Exercise 19 *Prove the converse of the above Theorem: If R is strongly representable in* **RA***, then R is recursive.*

Proof of the Main Theorem: This follows from the preceding two theorems by the following simple fact: If a relation is strongly representable in **RA** then it is strongly representable in every extension of **RA**, and hence it is representable (rather than strongly representable) in every consistent extension of **RA**. This is immediate from the definitions of representable and strongly representable (page 88). □

We now turn to the proof of the Undecidability Theorem. First note that if the hypothesis of this theorem is strengthened to assume that every r.e. (as opposed to recursive) relation is representable in Σ, then it would be very easy to prove that Σ is undecidable. (See the proof of Corollary 1 to the **RA** Representation Theorem, page 83). The reason the theorem is stated with the weaker hypothesis is to make the argument in the preceding paragraph work. See exercise 16 to see what goes wrong when using the alternative form of the Undecidability Theorem.

Proof of the Undecidability Theorem: (Like the proof of Tarski's Theorem)

Assume Σ is recursive. The idea is to formulate a sentence "I am not in Σ". This should be true, because Σ is consistent, but then it should be in Σ by representability, a contradiction.

Recall $d(x) = \text{sub}\ (x, x)$ from the proof of Tarski's theorem. Then d is a function (semantic notion) with the property that for all $a \in \mathbb{N}$, $d(a) = \#A(s_a)$ where $a = \#A(x)$. Note that d is a recursive function.

Define $R(x) \Leftrightarrow (x = \#A, \text{ for some } A \in \Sigma)$. Thus $R = \hat{\Sigma}$, and Σ is recursive iff R is recursive. In order to get a contradiction, assume R is recursive. Let

$$S(x) \Leftrightarrow \neg R(d(x))$$

Then S is recursive. Hence by hypothesis, $S(x)$ is represented in Σ by some formula $B(x)$.

By definition of representable
(1) $\quad \neg R(d(a)) \Leftrightarrow (B(s_a) \in \Sigma), \quad$ for all $a \in \mathbb{N}$

Let $e = \#B(x)$. Then $d(e) = \#B(s_e)$ by definition of $d(x)$. Then by (1),

$$\neg R(d(e)) \Leftrightarrow (B(s_e) \in \Sigma)$$

The LHS asserts $B(s_e) \notin \Sigma$, because R represents membership in Σ. This is a contradiction, hence Σ is not recursive. □

Proof of the Strong RA Representation Theorem: Suppose $R(\vec{x})$ is a recursive relation. Then both R and $\neg R$ are r.e., so by the Exists Delta Theorem, there are bounded formulas B_1 and B_2 such that $\exists y B_1(\vec{x}, y)$ represents $R(\vec{x})$ in **TA** and $\exists y B_2(\vec{x}, y)$ represents $\neg R(\vec{x})$ in **TA**. As pointed out in the previous proof, $\exists y B_1(\vec{x}, y)$ also represents $R(\vec{x})$ in **RA**, but in general it will not strongly represent $R(\vec{x})$ in **RA**. For strong representation we define a formula

$$A(\vec{x}) \equiv \exists y[B_1(\vec{x}, y) \wedge \forall z \leq y \neg B_2(\vec{x}, z)]$$

Claim: $A(\vec{x})$ strongly represents $R(\vec{x})$ in **RA**.

First we establish that for all $\vec{a} \in \mathbb{N}$,

$$R(\vec{a}) \Rightarrow \mathbf{RA} \vdash A(s_{\vec{a}}) \tag{3}$$

Since $\exists y B_1(\vec{x}, y)$ represents $R(\vec{x})$ in **RA**, we conclude from $R(\vec{a})$ that

$$\mathbf{RA} \vdash B_1(s_{\vec{a}}, s_b), \text{ for some } b \in \mathbb{N}$$

By the property of B_2 we know $\forall z \leq s_b \neg B_2(s_{\vec{a}}, z) \in \mathbf{TA}$, so by the MAIN LEMMA this sentence is in **RA**. This establishes (3) (take $y = b$).

It remains to establish

$$\neg R(\vec{a}) \Rightarrow \mathbf{RA} \vdash \neg A(s_{\vec{a}}) \tag{4}$$

Assume $\neg R(\vec{a})$. Note that $\neg A(s_{\vec{a}})$ is equivalent to

$$\forall y[\neg B_1(s_{\vec{a}}, y) \vee \exists z \leq y B_2(s_{\vec{a}}, z)] \tag{5}$$

Since $\exists z B_2(\vec{x}, z)$ represents $\neg R(\vec{x})$ in **RA** it follows that for some $c \in \mathbb{N}$

$$\mathbf{RA} \vdash B_2(s_{\vec{a}}, s_c) \tag{6}$$

By P9,

$$\mathbf{RA} \vdash \forall y(y \leq s_c \vee s_c \leq y)$$

(This is the only place that P9 is needed.) Thus to establish (5) in **RA** we consider two cases, depending on whether $y \leq s_c$ or $s_c \leq y$. For the first case, we note that

$$\forall y \leq s_c \neg B_1(s_a, y)$$

is a true bounded formula, and therefore by the MAIN LEMMA provable in **RA**, so (5) follows in **RA**

For the second case, by (6) we have

$$\mathbf{RA} \vdash \forall y(s_c \leq y \supset \exists z \leq y B_2(s_a, z))$$

so again (5) follows in **RA**. □

Exercise 20 *Let* $\neg\mathbf{RA} = \{A \mid \mathbf{RA} \vdash \neg A\}$*. Thus* $\neg\mathbf{RA}$ *is the set of sentences which* **RA** *proves false. Prove that* **RA** *and* $\neg\mathbf{RA}$ *are recursively inseparable. That is, prove that there is no recursive set* S *of sentences such that*

$$\mathbf{RA} \subseteq S \text{ and } \neg\mathbf{RA} \subseteq S^c$$

where $S^c = \{A \in \Phi_0 \mid A \notin S\}$. *(Note that* S *need not be a theory.)*

Hint: *Study the proof of Tarski's Theorem (page 91) and of the Undecidability Theorem (page 89). Assume that there is a recursive set* S *satisfying the indicated conditions. Formulate a sentence asserting "I am not in* S*", and obtain a contradiction.*