Incompleteness and Undecidability

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.

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

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

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

Definition of Bounded Formula and Δ_0 Formula: A formula A in $\mathcal{L}_{A,\le}$ 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 $\mathcal{L}_{A,\le}$ using the translation for $t_1 \le t_2$ 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

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

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 $A(x)$:

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

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). The boolean operations $\wedge, \vee, \neg$, for example, are discussed in the context of primitive recursive relations on page 62 of the notes, and the operations of bounded quantification are discussed on page 63 of the notes.

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 primitive recursive.

Proof: Structural Induction on bounded formulas in the vocabulary $\mathcal{L}_{A,\leq}$. We use the fact that the primitive recursive relations (i.e. predicates) are closed under the boolean operations and bounded quantification, as discussed on pages 62 and 63 in the notes. $\square$

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 follows from the Ritchie-Cobham Theorem that all relations recognizable in space bounded by a primitive recursive function of the input length are primitive recursive. In particular, space $O(n^2)$ relations are primitive recursive, and 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. (defined page 75)

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 primitive recursive, and hence recursive, and therefore R is r.e., by the definition of r.e. $\square$

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 $\exists \leq$ on pages 62 and 63. 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

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

Closure Lemma: The $\exists\Delta_0$ relations are closed under $\exists$, $\wedge$ and $\vee$, and the bounded quantifiers $\exists \leq$ and $\forall \leq$. (See page 63 for the definitions of $\exists <$ and $\forall <$, from which $\exists \leq$ and $\forall \leq$ can be obtained.)

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 is primitive recursive, 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

$$R(\vec{x}) \Leftrightarrow (R(\vec{x}) = 0)$$

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 primitive recursive (see part iii) of the Theorem, page 77). We know that S is $\exists\Delta_0$ by the paragraph above, and thus R is $\exists\Delta_0$ by the Closure Lemma. $\square$

Proof of Main Lemma: Induction on primitive recursive functions. (Recall the definition of primitive recursive function on page 57.)

Base Case $Z, S, I_{n,i}$

Relation	$\exists\Delta_0$ formula
$(y = Z)$	$y = 0$
$(y = S(x))$	$y = sx$
$(y = I_{n,i}(\vec{x}))$	$y = x_i$

These are all Δ_0 formulas

Induction step
Case I: $f(\vec{x}) = g(h_1(\vec{x}), \ldots, h_k(\vec{x}))$, where $g, h_1, ..., h_k$ each has a $\exists\Delta_0$ graph. Then

$$(y = f(\vec{x})) = \exists y_1 \cdots \exists y_k (y_1 = h_1(\vec{x}) \wedge \cdots \wedge y_k = h_k(\vec{x}) \wedge y = g(y_1, \ldots, y_k))$$

The RHS is $\exists\Delta_0$ by the Closure Lemma.

Case II: Primitive Recursion. This is the hard case, and requires a new idea: the Gödel β Function. This 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 by no means immediately clear that the relation $(z = x^y)$ can be represented by a formula in the vocabulary $\mathcal{L} = [0, s, +, \cdot; =]$, which does not include exponentiation as a function symbol.

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

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

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

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

and

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

there exists r such that

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

Proof: of CRT is by counting: Distinct values of r, $0 \le r < \Pi m_i$, 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 $r, 0 \le r < \Pi m_i$. $\square$

Proof of Gödel Lemma: Let $d = (n + r_0 + \cdots + r_n + 1)!$
Let $m_i = d(i + 1) + 1$
Claim: $0 \le i < j \le n \Rightarrow \gcd(m_i, m_j) = 1$
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 $p \le j - i < n$, so $p \mid d$, a contradiction.

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

$$\beta(c, d, i) = rm(c, m_i) = r_i, \ 0 \le i \le n \ \ \square$$

Back to Case II of the Induction

f is defined by primitive recursion:
  $f(\vec{x}, 0) = g(\vec{x})$
  $f(\vec{x}, y + 1) = h(\vec{x}, y, f(\vec{x}, y))$
where graph(g) and graph(h) are $\exists\Delta_0$ relations.

Then $z = f(\vec{x}, y)$ iff $\exists r_0, \ldots, r_y$ such that

  (i) $r_0 = g(\vec{x})$
  (ii) $r_{i+1} = h(\vec{x}, i, r_i), \quad 0 \le i < y$
and (iii) $r_y = z$

Thus

$$(z = f(\vec{x}, y)) = \exists c \exists d[\beta(c,d,0) = g(\vec{x}) \wedge$$
$$\forall i < y(\beta(c,d,i+1) = h(\vec{x}, i, \beta(c,d,i)) \wedge \beta(c,d,y) = z]$$

The fact that the RHS is a $\exists\Delta_0$ relation follows from the Closure Lemma and the following two lemmas:

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:

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

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

Exercise 3 *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 84).

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

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

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 4 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 Initial Functions (Notes page 57) and 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).

(c) Show that if $R(\vec{x}, y)$ is a Δ_0 relation and $f(\vec{x}, y)$ is defined from R by bounded minimization (Notes page 64), then $f(\vec{x}, y)$ is a Δ_0 function. **Hint:** *This is easier than the argument for the primitive recursive case given on page 65.*

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 95.)

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 (see p73).

Exercise 5 *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 commands and programs in the section on computability (page 66). 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. $\Box$

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. $\Box$

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 6 *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 $A, B \in \Phi_0$, $\{A, \neg A\} \models B$.) Thus Σ is complete iff for all sentences A, exactly one of $\Sigma \vdash A$ and $\Sigma \vdash \neg A$ holds.

Exercise 7 *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 8 *Prove that $Th(\mathcal{M})$ is a complete theory, for every structure $\mathcal{M}$.*

For example, $\mathbf{TA} = Th(\underline{\mathbb{N}})$, so $\mathbf{TA}$ 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 $\mathbf{TA}$ 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 the Lemma page 76 $\hat{\Sigma} = \text{ran}(f)$, 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. $\Box$

Corollary 1: $VALID$ is r.e., where $VALID$ is the set of valid sentences (page 93).

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 **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 **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.)