Incompleteness Undecidability & Undefinability

Here are some philosophical questions with mathematical answers:

○ (1) Is there a (recursive) algorithm for deciding whether an arbitrary sentence in the language of first-order arithmetic is true?

○ (2) Is there an algorithm for deciding whether an arbitrary sentence in the language of first-order arithmetic is a theorem of Peano or Robinson Arithmetic?

○ (3) Is there an algorithm for deciding whether an arbitrary sentence in the language of first-order arithmetic is a theorem of pure (first-order) logic?

○ (4) Is there a complete (even if not recursive) recursively axiomatizable theory in the language of first-order arithmetic?

○ (5) Is there a recursively axiomatizable sub-theory of Peano Arithmetic that proves the consistency of Peano Arithmetic (even if it leaves other questions undecided)?

○ (6) Is there a formula of arithmetic that defines arithmetic truth in the standard model, N (even if it does not represent it)?

○ (7) Is the (non-recursively enumerable) set of truths in the language of first-order arithmetic categorical? If not, is it $\omega$-categorical (i.e., categorical in models of cardinality $\omega$)?

Questions (1) -- (7) turn out to be linked. Their philosophical interest depends partly on the following philosophical thesis, of which we will make frequent, but inessential, use.

○ **Church-Turing Thesis:** A function is (intuitively) computable if/f it is recursive.

  ■ *Church:* “[T]he notion of an effectively calculable function of positive integers should be identified with that of recursive function (quoted in Epstein & Carnielli, 223).”

○ **Note:** Since a function is recursive if/f it is Turing computable, the Church-Turing Thesis also implies that a function is computable if/f it is Turing computable.

Robinson Arithmetic (Q)

Q is the following finitely-axiomatized first-order theory. It is of interest because it is among the weakest theories for which *recursiveness is equivalent to representability.* A function is recursive just in case it is representable (in a sense to be explained) in Q.
- (Q1) \( (\forall x)(\forall y)[(x' = y') \to (x=y)] \)
  - If the successors of two numbers are equal, then the numbers are equal.

- (Q2) \( (\forall x)[0 = x'] \)
  - Zero is not the successor of any number.

- (Q3) \((\forall x)[(\neg [x = 0]) \to ((\exists y)(x=y'))] \)
  - Every number other than zero is the successor of a number.
  - Note: (Q1) and (Q2) already imply that Q has only infinite models.

- (Q4) \((\forall x)[(x + 0) = x]\)
  - Any number plus zero is equal to that number.

- (Q5) \((\forall x)(\forall y)[(x + y') = (x + y)'] \)
  - The sum of a number and the successor of a number is equal to the successor of the sum of the two numbers.

- (Q6) \((\forall x)[(x * 0) = 0]\)
  - Any number times zero is zero.

- (Q7) \((\forall x)(\forall y)(x * y') = [(x * y) + x]\)
  - The product of some number and the successor of a number is equal to the product of the two numbers plus the first number.

- Note: We will write \( n \) to abbreviate the formal numeral \( 0'''...' \) (n primes)

**Basic Arithmetic in Q**

- Q is great with particulars. It is \( \Sigma_1 \)-complete (i.e., Q can prove all true \( \Sigma_1 \) sentences).

  - Recall: A \( \Sigma_1 \) formula is of the form \((\exists 1x)(\exists 2x)...(\exists nx)\Delta_0\), where \( \Delta_0 \) is a formula with only bounded quantifiers. Similarly, a \( \Pi_1 \) formula is of the form \((\forall 1x)(\forall 2x)...(\forall nx)\Delta_0\). Hence, the negation of a \( \Sigma_1 \) formula is \( \Pi_1 \), while the negation of a \( \Pi_1 \) formula is \( \Sigma_1 \).

  - Note: A theory can be \( \Sigma_1 \)-complete but not \( \Sigma_1 \)-sound! (That is, it could prove all true \( \Sigma_1 \) sentences as well as some false \( \Sigma_1 \) sentence.) We will construct just such a theory when we supplement Peano Arithmetic with the negation of its own consistency sentence.
- **Theorem 1**: For all natural numbers, i and j:
  - (a) If i = j, then Q |-- i = j
  - (b) If i ≠ j, then Q |-- i ≠ j

- **Theorem 2**: For all natural numbers, i, j, and k, such that i + j = k:
  - (a) Q |-- (i + j) = k
  - (b) Q |-- (∀ x)[((i + j) = x) → (x = k)]

- **Theorem 2.1**: For any natural numbers, i and j:
  - Q |-- (i + j) = (j + i)

- **Theorem 3**: For all natural numbers i, j, and k, such that i x j = k:
  - (a) Q |-- (i * j) = k
  - (b) Q |-- (∀ x)[((i * j) = x) → x = k]

- **Theorem 3.1**: For all natural numbers i and j:
  - Q |-- (i * j) = (j * i)

- **Caveat**: Just because Q proves that addition and multiplication are commutative for its numerals (by Theorems 2.1 and 3.1), it does not follow that Q proves that these operations are commutative for all natural numbers (i.e., universal generalizations). Indeed, we will see that Q proves neither of these general laws.

- **Proof**: Part (b) of **Theorem 2 & 3** follow from logic (of identity), given the other theorems. The other theorems are each of the form α = β or α ≠ β, where α and β are (perhaps complex) numerical terms. If such a term, θ, refers to n in the standard interpretation, N, then Q |-- θ = n. Consequently, if i = j, and α = β says so on N, then Q |-- α = β ←→ i = j, and Q |-- i = j, as i is the same numeral as j. Likewise, if i ≠ j, and α ≠ β says so on N, then, by the logic of identity, Q |-- i ≠ j, and again Q |-- α ≠ β.

- **Note**: These (meta)theorems suffice to significantly constrain models of Q. That part of any model which interprets Q’s numerals must look just like the standard model, N.

- **Upshot**: Any non-standard model of Q must look just like the standard model, N, but with some additional elements “tacked onto the end” (and additional values for ’, +, and *).
Representability

- We have really been showing that addition and multiplication are representable in $\mathbb{Q}$.

  ■ Caveat: The proof of this uses Induction, which $\mathbb{Q}$ itself does not have!

- Definition 1: A total function $f: \mathbb{N}^n \to \mathbb{N}$ is representable in theory, $T$, if/iff there is a formula, $\Phi$, in which $x_1, x_2, \ldots, x_n, x_{n+1}$ occur free, such that when $q_1, q_2, \ldots q_n, k$ are natural numbers with $f(q_1, q_2, \ldots q_n) = k$, we have:

  ■ (a) $T \vdash \Phi(q_1, q_2, \ldots q_n, k)$

  ■ (b) $T \vdash (\forall x)[\Phi((q_1, q_2, \ldots q_n, x) \rightarrow (x = k)]$

- In this case, the formula, $\Phi$, is said to represent function, $f$, in theory, $T$.

  ■ Note: Part (b) reflects the fact that $f$ is a function, giving a unique output.

    - Example: By Theorem 2, $(x_1 + x_2) = x_3$ represents the plus function in $\mathbb{Q}$, while, by Theorem 3, $(x_1 \times x_2) = x_3$ represents times.

    - Example: The zero, successor, and projections functions --i.e., the basic primitive recursive functions -- are representable in $\mathbb{Q}$.

- Theorem 4 (Representability Theorem): Every total recursive function is representable in any theory, $T$, which extends $\mathbb{Q}$ (where $T$ extends $\mathbb{Q}$ when $\mathbb{Q} \subseteq T$).

  ■ Note: $\mathbb{Q}$ explicitly represents +, × and *, but implicitly represents far more.

  ■ In fact, despite being so weak, $\mathbb{Q}$ is optimal, in the following sense:

- Theorem 5: If $T$ is any consistent theory with a recursive set of axioms, then every total function representable in $T$ is itself recursive.

  ■ Proof: Suppose that $f: \mathbb{N}^n \to \mathbb{N}$ is a total function represented in $T$ by $\Phi$. Then $\forall q_1 q_2 \ldots q_n, k \in \mathbb{N}, T \vdash \Phi(q_1 q_2 \ldots q_n, k)$ if/iff $f(q_1 q_2 \ldots q_n) = k$, and the theorems of $T$ are algorithmically enumerable. This gives an algorithm for computing $f$. Given $q_1 q_2 \ldots q_n$, wait for a theorem of the form $\Phi(q_1 q_2 \ldots q_n, k)$. (Since $T$ is consistent, there will be exactly one of this form.) By the Church-Turing Thesis, $f$ is therefore recursive.

- Theorem 6 (Equivalence): A total function is representable in $\mathbb{Q}$ if/iff it is recursive.

  ■ Proof: From Theorems 4 and 5 (assuming that $\mathbb{Q}$ is consistent).
Godel Numbering

- A formula in the language of arithmetic is a finite string of symbols. So, we can pair those symbols, and strings of them, with natural numbers, called their Godel numbers.

- There are many ways to do this. But the pairing must satisfy the following conditions.
  - (a) Different strings of symbols are assigned to different Godel numbers
  - (b) There is a (recursive) algorithm such that, given a natural number, it decides whether the number codes a string of basic symbols, and, if it does, what string it codes.
  - (c) There is a (recursive) algorithm such that, given a string of basic symbols, it decides what the Godel number of that string is.

- Here is one pairing, $\beta$, that will do the trick:
  
  | Symbol | ( | ) & v → ←→ ∼ ∀ ∃ 0 ' + * = x_i |
  | Number | 1 2 3 4 5 6 7 8 9 10 11 12 13 14 i+15 |

- This generates, corresponding to each string of symbols, $a_1a_2a_3a_4\ldots$, a string of Godel numbers, given by the above pairing, $\beta(a_1)\beta(a_2)\beta(a_3)\ldots$

- We can now assign the overall string $a_1a_2a_3a_4\ldots$ the Godel number:
  
  - $(p_1^{\beta(a_1)})(p_2^{\beta(a_2)})(p_3^{\beta(a_3)})\ldots$, where $p_1p_2p_3\ldots p_k$ are the first $k$ primes.

  - Note: This will generally be a very big number! But, given it, we can check after finitely-many steps that it is the Godel number of the string.

  - Note: We use here the Fundamental Theorem of Arithmetic, that every natural number has a unique decomposition into powers of primes.

- If $\Psi$ is a string of symbols, then we will write $\gamma(\Psi)$ for the Godel number of $\Psi$.

- It remains to code arbitrary formal proofs, $F$. If the following is such a proof,
  
  - $\Phi_1$
  - $\Phi_2$
  - $\Phi_3$
  - $\ldots$
  - $\Phi_k$
then we can code it with the function, $\Gamma(F)=(2^{\gamma(\Phi_1)})(3^{\gamma(\Phi_2)})\ldots(p_k^{\gamma(\Phi_k)})$.

- **Upshot:** We can now describe facts about our theory, $Q$, in terms of Gödel numbers. For instance, the second of our initial questions, (2), can now be phrased: is there an algorithm for deciding whether a given number is the Gödel number of a theorem of $Q$?

### The Diagonal Lemma
- **Theorem 6:** If $\Phi$ is a formula in which at most $x$ and $y$ occur free, then for all natural numbers $n$ and $k$:
  
  $\Phi(n, k), \forall y[\Phi(n, y) \rightarrow (y = k)] \quad \text{---} \quad \forall y[\Phi(n, y) \leftarrow (y = k)]$

  - **Note:** This is a theorem of first-order logic.
  - **Clarification:** Variables like $x$ and $y$ are doing double-duty, as meta-variables in our meta-language, and as variables in $Q$.

- **Theorem 7:** If $f: \mathbb{N} \rightarrow \mathbb{N}$ is a total function represented in theory, $T$, by formula, $\Phi$, then, for all natural numbers $n$ and $k$, if $f(n) = k$,

  $\quad T \vdash \forall y[\Phi(n, y) \leftrightarrow y = k]$

  - **Proof:** Suppose that $f(n) = k$. Then, by the **Definition 1**, we have that $T$ proves both of the antecedents of **Theorem 6**.

- **Notation:** If $\Phi$ is a formula with Gödel number, $n$, then $\langle \Phi \rangle$ is the term, $n$.

  - **Note:** $\langle \Phi \rangle$ is not $\gamma(\Phi)$. $\gamma(\Phi)$ is a natural number, while $\langle \Phi \rangle$ is the numeral in the language of arithmetic which, in the standard interpretation, $\mathbb{N}$, refers to $\gamma(\Phi)$.

  - **Example:** If $\Phi$ is ($\exists y)[x = (2 * y)] \land \neg(\exists y)[x = (4 * y)]$, and $m$ is $\gamma(\Phi)$, then $D\Phi$ is ($\exists x)(x = m \land (\exists y)[x = (2 * y)] \land \neg(\exists y)[x = (4 * y)]$.

  - **Definition 2 (Diagonalization):** If $\Phi$ is a formula in which at most some variable, $x$, occurs free, then the diagonalization of $\Phi$, written $D\Phi$, is the (closed) sentence:

    $\quad (\exists x)(x = \langle \Phi \rangle \land \Phi)$

    - **Example:** If $\Phi$ is ($\exists y)[x = (2 * y)] \land \neg(\exists y)[x = (4 * y)]$, and $m$ is $\gamma(\Phi)$, then $D\Phi$ is ($\exists x)(x = m \land (\exists y)[x = (2 * y)] \land \neg(\exists y)[x = (4 * y)]$.
• **Note:** If \( x \) is free in \( \Phi \), then the diagonalization of \( \Phi \) “says” that \( \gamma(\Phi) \) has the property, \( \Phi \), and \(( \exists x)(x = \gamma(\Phi) \land \Phi) \dashv \vdash \Phi(\gamma(\Phi))\).

  ■ **Note:** The diagonalization of a formula need not be true (under the standard interpretation), though it might be (and is in Example above).

• **Theorem 8:** If \( \Phi \) is a formula in which some variable, \( x \), occurs free, and \( n \) is a natural number:
  
  ○ \(-[\neg(\exists x)(x = n \land \Phi) \iff \Phi(n)]\)

  ○ **Note:** This is a logical truth.

• **Theorem 9:** If \( \Phi \) is a formula in which at most some variable, \( x \), occurs free:
  
  ○ \(-[D\Phi \iff \Phi(\gamma(\Phi))]\)

  ○ **Proof:** Immediate from Theorem 8.

• **Theorem 10:** There is a (primitive) recursive function, \( * \), such that, for all strings \( \alpha \) and \( \beta \) of basic symbols, if \( m = \gamma(\alpha) \) and \( n = \gamma(\beta) \), then \( m*n = \gamma(\alpha\beta) \).

• **Theorem 11:** There is a (primitive) recursive function, \( \text{diag}(x) \), with the following property. If \( n \) is the Godel number of a formula \( \Phi \) in which at most some variable, \( x \), occurs free, then \( \text{diag}(n) \) is the Godel number of the diagonalization of \( \Phi \), \( D\Phi \).

\[
\begin{array}{c}
\Phi \\
\downarrow \\
\text{has Godel number } n
\end{array}
\quad
\begin{array}{c}
(\exists x)(x = \gamma(\Phi) \land \Phi) \\
\downarrow \\
\text{has Godel number } m
\end{array}
\quad
\begin{array}{c}
n \\
\rightarrow \\
\text{diag}
\end{array}
\quad
\begin{array}{c}
m
\end{array}
\]

• **Note:** It is intuitively clear that the function, \( \text{diag}(x) \), is recursive since all we have to do to calculate it is check if \( n \) is the Godel number of a formula, and, if it is, add the code for “\((\exists x)(x = n \land \)” in front and add the code for “)” in the back.

• **Theorem 12 (Diagonal Lemma):** If \( \text{diag} \) is representable in a theory, \( T \), then, for each formula, \( \Phi \), in which at most some variable, \( y \), occurs free, there is a sentence, \( G\Phi \), such that:
Proof Sketch: Let $\Theta$ represent the function, $\text{diag}$, in theory $T$, with at most $x$ and $y$ free. Let $\psi$ be a formula with at most $y$ free, and take $\Omega$ to be the formula $(\exists y)(\Theta \& \psi)$, where at most $x$ may occur free. Finally, identify $G\Theta$ with the diagonalization of $\Omega$, $D\Omega$, to get $T \vdash [G\Phi \leftrightarrow \Phi(G\Phi^\gamma)]$.

- Example: If $\Phi$ is the formal analog of “is even”, then $G\Phi$ is a sentence which is true just in case the Godel number of $G\Phi$ is even.

- Example: If $\Phi$ is the formal analog of “codes a sentence which is provable”, then $G\Phi$ is a sentence which is true just in case the Godel number of $G\Phi$ codes a sentence which is, indeed, provable.

Note: The Diagonal Lemma means that every predicate in $T$ has a fixed point -- roughly, a point where you get the same thing out that you put in.

- Upshot: Every theory extending $Q$ has the capacity for self-reference.

Undecidable Theories

- Definition 3: A set, $A$, is representable in theory, $T$, if its characteristic function is (where the characteristic function, $f_A$, of set, $A$, is $f_A(n) = 1$ when $n \in A$ and $0$ otherwise.)

- Theorem 13: Every recursive set is representable in any theory, $T$, which extends $Q$.

  - Proof: Immediate from the Representability Theorem.

- Theorem 14: If $T$ is a theory extending $Q$, and $A$ is a set of natural numbers which is representable in $T$, then there is a formula, $\delta_A$, in which at most $x$ occurs free, such that, for each natural number, $n$:

  o (a) If $n \in A$, then $T \vdash \delta_A(n)$

  o (b) If $n \notin A$, then $T \vdash \neg \delta_A(n)$

  - Proof: Let $\Phi_A$ represent the characteristic function, $f_A$, of $A$ in $T$. Then $\forall n,k \in \mathbb{N}$, if $f_A(n) = k$, $T \vdash \Phi_A(n, k)$, and $T \vdash \forall y[\Phi_A(n, y) \rightarrow y=k]$. Define $\delta_A(x) = \Phi_A(x, 1)$. If $n \in A$, then $f_A(n) = 1$, so $T \vdash \Phi_A(n, 1)$ and, hence, $T \vdash \delta_A(n)$. Conversely, if $n \notin A$, then $f_A(n) = 0$, and, thus, $T \vdash \forall y[\Phi_A(n, y) \rightarrow y=0]$. But since $T$ extends $Q$, $T \models 1 \neq 0$. Hence, by logic in $T$, $T \vdash \neg \Phi_A(n, 1)$, and, therefore, $T \vdash \neg \delta_A(n)$.

  - In fact, the other direction holds as well.
• **Theorem 15**: If T extends Q, then the following conditions on a set, A, of natural numbers are equivalent:

○ (a) The set, A, is representable in T.

○ (b) There is a formula, $\delta_A$, in which at most some variable, x, occurs free, such that, for each natural number, n, if $n \in A$, then $T |- \delta_A(n)$, and, if $n \notin A$, then $T |-- \sim \delta_A(n)$.

• **Theorem 16 (Undecidability of Extensions of Q)**: If T is a consistent theory extending Q, then the set of Godel numbers of the theorems of T, $\text{GN}(T)$, is *not* representable in T.

○ *Proof*: Assume for reductio that $\text{GN}(T)$ is representable in T. Then by *Theorem 15*, there is a formula, $\delta(x)$, whose only free variable is x, such that, for any $n \in \mathbb{N}$,

- If $n \in \text{GN}(T)$, then $T |-- \delta(n)$
- If $n \notin \text{GN}(T)$, then $T |-- \sim \delta(n)$

○ Moreover, since $\text{GN}(T)$ is the set of Godel numbers of theorems of T, for each $\Phi$,

- If $T |-- \Phi$, then $\gamma(\Phi) \in \text{GN}(T)$
- If $\sim (T |-- \Phi)$, then $\gamma(\Phi) \notin \text{GN}(T)$

○ Hence, if $T |-- \Phi$, then $T |-- \delta(\gamma(\Phi))$, and if $\sim (T |-- \Phi)$, then $T |-- \sim \delta(\gamma(\Phi))$. So, applying the Diagonal Lemma to $\sim \delta$, we are ensured a sentence, $G_{\sim \delta}$ such that:

- $T |-- (G_{\sim \delta} \iff \sim \delta(\gamma(G_{\sim \delta})))$.

- *Note*: The sentence $\sim \delta(\gamma(G_{\sim \delta}))$ “says” that $G_{\sim \delta}$ is not a theorem of T.

○ However, both the assumption that $T |-- G_{\sim \delta}$ and $\sim (T |-- G_{\sim \delta})$ lead to contradiction.

- *Note*: *Theorem 16* allows that the set of theorems of T is representable in *extensions* of T. Moreover, the assumption that T is consistent is essential.

• **Definition 4**: True Arithmetic, TA, is the set of all sentences, S, such that $N \models S$.

• **Theorem 17**: The set of Godel numbers of sentences of True Arithmetic, $\text{GN}(TA)$, is not a recursive set.

○ *Proof*: Immediate from *Theorem 16*. 

9
- **Theorem 18 (Church-Turing Theorem)**: The set of theorems of first-order logic is not recursive.
  - **Proof**: If it were, then the set of theorems of Q would be too, contrary to Theorem 16. For an arbitrary Φ, check whether AxQ → Φ is a theorem of first-order logic.

**Undefinability of Truth**
- **Definition 5**: A relation R(x₁x₂,...,xₖ) of natural numbers is **definable** in the standard model, N, if/iff there is a formula ϕ(x₁x₂,...,xₖ) in the language of arithmetic such that for every n₁n₂,...,nₖ, R(n₁n₂,...,nₖ) if and only if N |= Φ(n₁,...,nₖ).

- **Theorem 19 (Godel-Tarski Theorem)**: The set of Godel numbers of sentences, S, true in N (i.e., {S: N|= S}) is not definable in arithmetic.
  - **Proof**: A set is definable in N if/iff it is representable in TA. But TA consistently extends Q. Hence, again, this set is not representable in TA, by Theorem 16.

- **Upshot**: Extensions of Q can define typical syntactic properties, but not semantic ones.

**Axiomatizability, Enumerability, and Completeness**
- **Definition 6**: A theory, T, is (recursively) **axiomatizable** if/iff there is a set, S, of axioms of T (i.e., a set S ⊆ T such that, for every Φ ∈ T, S |-- Φ) where the set of Godel numbers of S is recursive.
  - **Note**: There is a big difference between a theory’s having a recursive set of axioms and its having a recursive set of theorems. Q has a recursive -- indeed, finite -- set of axioms, but we have seen that its set of theorems is not recursive.

- **Definition 7**: A set, A, of natural numbers is recursively **enumerable** if/iff either A = ∅ or A = {f(0), f(1), f(2)...} where f is a total recursive function.
  - **Note**: If a set is recursive, then it is recursively enumerable.

- **Theorem 20**: If T is a recursively axiomatized theory, then the set of Godel numbers of theorems of T, GN(T), is recursively enumerable.
  - **Proof**: Since the set of theorems of T is not empty, call the Godel number of (∀ y)(y = y), n₀. Now, define a computable function, f, as follows. Given n ∈ N, first check whether n is the Godel number of a proof in T. If not, let f(n) = n₀. If so, we can compute from n what assumptions the proof relied on. Moreover, since the set of axioms of T is recursive, we can check whether each of these assumptions is a member of that set. If it is, let f(n) be the Godel number of the last line of the proof, and if not, let f(n) = n₀. Conversely, if Φ is a theorem, then some number, Γ(F), codes its proof, F, in T, meaning f(Γ(F)) = γ(Φ), as desired.
- **Upshot**: Though being recursive implies being recursively enumerable, the converse implication fails. But there is one circumstance when it does not.

- **Theorem 21**: If T is an axiomatizable theory which is complete (i.e., T |-- Φ or T |-- ~Φ, for all sentences), then the set of Godel numbers of theorems of T is recursive.

  - **Proof**: If T is inconsistent, it is trivially recursive. So, assume that T is consistent. By Theorem 20, there is a recursive function which enumerates its theorems. For any sentence, Φ, simply wait for Φ or ~Φ to appear after a finite number of steps.

- **Theorem 22 (Godel’s First Incompleteness Theorem)**: If T is a axiomatizable extension of Q, then either T is incomplete or inconsistent.

  - **Proof**: By Theorem 21, if T were complete, then it would have a recursive set of theorems, contradicting Theorem 16.

- **Theorem 23**: True Arithmetic is not axiomatizable.

  - **Proof**: Immediate from Theorem 22.

- **Theorem 24**: True Arithmetic is not recursively enumerable.

  - **Proof**: It is a complete and consistent extension of Q, so, if it were so enumerable, then, by Theorem 21, its theorems would be recursive, contrary to Theorem 16.

---

**Hilbert’s Program and Consistency**

- **Definition 7**: Peano Arithmetic (PA) is Q conjoined with all instances of:

  - **Induction Schema**: (Φ(0)) & ∀x(Φ(x) → Φ(x′)) → ∀xΦ(x)

    - **Note**: Φ may also contain parameters.

- As an axiomatizable (albeit not finitely axiomatizable) extension of Q, Theorem 20 implies that PA is incomplete, if consistent. But PA is much stronger than Q.

  - Although Q is good with particulars, it is bad with generalizations. Simple model-theoretic arguments show that it does not prove any of the following.

    - ∀x(x ≠ x′)
    - ∀x((x + (y + z)) = [(x + y) + z])
    - ∀x∀y((x + y) = (y + x])
    - ∀x[(0 + x) = x]
    - ∀y∀z[∃x((x′ + y) = z) & ∃x((x′ + z) = y]
    - ∀x∀y∀z((x * (y * z)) = ((x * y) * z])
    - ∀x∀y((x * y) = (y * x])
∀x ∀y ∀z[(x * (y + z)) = ((x * y) + (x * z)]

The problem is that *Q has no way of saying that the things picked out by its numerical terms are the only numbers.* This idea is regimented by Induction. And, indeed, in PA, all of the above are provable using the Induction Schema.

**Example:** In order to construct a model of (Q1) -- (Q7) + ~ ∃ x(x≠x'), add an element, ω, and interpret the successor function as before but with s(ω) = ω, interpret + as before but with n + ω = ω + n = ω, for all numbers, n (including ω), and let n * ω = ω * n = ω when n ≠ 0, and 0 otherwise.

- Hence, while PA must be incomplete, it might still be hoped that PA, or, even better, a “finitary” fragment of it, proves the consistency of PA. This was Hilbert’s ambition.

- In order to make this precise, we must formalize the sentence “PA is consistent” in the language of PA. Since PA |-- 0 = 1 if/ if PA is inconsistent, it suffices to formalize “PA does not prove 0 = 1”. The problem is that the set of (Godel numbers of) theorems of PA is not representable in PA, by *Theorem 16*. But there *is* a formula (actually, different formulas), Prov(x), with only x free, which *defines* the relation of provability in PA. So:

  - N |= Prov(⌜Φ⌝) if and only if PA |-- Φ

With this predicate in hand, our question can now be phrased: whether there is an axiomatizable theory, T, of which PA is an extension such that T |-- ~Prov(⌜0 = 1⌝).

- *Note:* ~Prov(⌜0 = 1⌝) is commonly written Con(PA).

**Definition 8:** If T extends Q, then the formula, Prov(x), of one free variable is a provability predicate for T if/ if, for all sentences and Ω and Ψ, the following hold:

- (a) If T |-- Ω , then T |-- Prov(⌜Ω⌝)

  - If T proves Ω , then it is a theorem of T that T proves Ω.

- (b) If T |-- Prov(⌜Ω → Ψ⌝) → [(Prov(⌜Ω⌝) → Prov(⌜Ψ⌝))]

  - It is a theorem of T that if the conditional Ω → Ψ, and its antecedent, Ω, are both provable in T, then so is the consequent, Ψ.

- (c) T |-- [Prov(⌜Ω⌝) → Prov(⌜Prov (⌜Ω⌝)⌝)]

  - It is a theorem of T that if Ω is provable in T, then the fact that Ω is provable in T is itself provable in T.

- *Note:* This condition can be questioned.
Theorem 25 (Lob’s Theorem): If T extends Q, and Prov(x) is a provability predicate for T, then, for each sentence, Ψ, T |-- Ψ if and only if T |-- Prov("Ψ") → Ψ.

Proof: Evidently, if T |-- Ψ, then T |-- [Prov("Ψ") → Ψ]. So, let us suppose that T |-- Prov("Ψ") → Ψ. Since T extends Q, diag(x) is representable in T, so letting δ(x) = [Prov("Ψ") → Ψ] with only x free, we get that T |-- (Gδ ↔ [Prov("Gδ") → Ψ]). A calculation using properties (a) and (b) and logic in T then yields that T |-- [Prov("Gδ") → Prov("Prov("Gδ")") → Prov("Ψ")], and by property (c) of Prov(x) we know that the antecedent of this conditional must be true, giving (by logic in T) that T |-- (Prov("Gδ") → Prov("Ψ")). And since we are assuming that T |-- Prov("Ψ") → Ψ, we have (by logic in T) that T |-- (Prov("Gδ") → Ψ). Finally, since T |-- [Gδ ↔ (Prov("Gδ") → Ψ)], we must have that T |-- Gδ. So, by (a), T |-- Prov("Gδ"), and, using modus ponens in T, T |-- Ψ, as desired.

Upshot: If T proves “if T proves something, it is true”, then T already proves it!

Theorem 26 (Godel’s Second Incompleteness Theorem): If T is a consistent extension of Q, and Prov(x) is a provability predicate for T, then ~(T |-- ~Prov("0 = 1")).

Proof: Let T be a consistent extension of Q and Prov(x) a provability predicate for it. Suppose T |-- ~Prov("0 = 1"). By logic in T, T |-- (Prov("0 = 1") → 0 = 1). So, by Lob’s Theorem, T |-- 0 = 1, contrary to the assumption that T is consistent.

Note: One can always choose ~Prov(x) to be a Π1 predicate, making ~Prov("0 = 1") a sentence of Goldbach Type. Once chosen we have a concrete example of a sentence that is not provable in PA, despite its being much more powerful than Q.

Nonstandard Models

Definition 9: A theory is categorical when all its models are isomorphic, and ω-categorical when all its models of cardinality κ are isomorphic.

Theorem 27: True Arithmetic is not ω-categorical.

Proof: Add a name, c, to the language of arithmetic, and define the following set, C, of formulas: \{x ≠ n : n is any numeral in the (non-augmented) language of arithmetic\}. Since every finite subset of the union of True Arithmetic (TA) and C is satisfiable, TA u C is too, and in a countable model, by the Compactness and Lowenheim-Skolem Theorems. But no such model can be isomorphic to N.

Example: The theory PA + ~Con(PA), which is consistent by Godel’s Second Incompleteness Theorem, has ω-models with a non-standard number, greater than the finite numbers, which witnesses a proof of 0 = 1.
Note: This means that, even if it were granted that there is a perfectly determinate body of arithmetic truths, it does not follow that they pick out a determinate model (even up to isomorphism).

Questions Revisited
- We are now in a position to answer all of the questions with which we began.
  - (1) There no algorithm for deciding whether a given sentence in the language of first-order arithmetic is true, by Theorem 17.
  - (2) There no algorithm for deciding whether a given sentence in the language of first-order arithmetic is a theorem of Robinson Arithmetic, by the Undecidability of Extensions of Q.
  - (3) There is no algorithm for determining whether a given sentence in the language of first-order arithmetic is a theorem of pure (first-order) logic, by the Church-Turing Theorem.
  - (4) There is no complete axiomatizable arithmetic theory in the language of first-order arithmetic, by Godel’s First Incompleteness Theorem.
  - (5) There is no axiomatizable theory of which Peano Arithmetic is an extension that proves the consistency of Peano Arithmetic, by Godel’s Second Incompleteness Theorem.
    - Godel: “For all formal systems for which the existence of undecidable arithmetical propositions was [demonstrated], the assertion of the consistency of the system in question itself belongs to the propositions undecidable in that system…..For a system in which all finitary...forms of proof are formalized, a finitary consistency proof, such as the formalists seek, would thus be...impossible (quoted in Epstein & Carnielli, 214).”
  - (6) There is no formula in the language of first-order arithmetic that defines arithmetic truth in the standard model, by the Godel-Tarski Theorem.
  - (7) The (non-recursively enumerable) set of truths in the language of first-order arithmetic, True Arithmetic, is not categorical or ω-categorical, by Theorem 27.

Bibliography


