

Chapter udf

Theories and Computability

tcp.1 Introduction

inc:tcp:int:
sec

We have the following:

1. A definition of what it means for a function to be representable in \mathbf{Q} (??)
2. a definition of what it means for a relation to be representable in \mathbf{Q} (??)
3. a theorem asserting that the representable functions of \mathbf{Q} are exactly the computable ones (??)
4. a theorem asserting that the representable relations of \mathbf{Q} are exactly the computable ones ??)

A *theory* is a set of sentences that is deductively closed, that is, with the property that whenever T proves φ then φ is in T . It is probably best to think of a theory as being a collection of sentences, together with all the things that these sentences imply. From now on, I will use \mathbf{Q} to refer to the *theory* consisting of the set of sentences derivable from the eight axioms in ???. Remember that we can code formula of \mathbf{Q} as numbers; if φ is such a formula, let $\# \varphi$ denote the number coding φ . Modulo this coding, we can now ask whether various sets of formulas are computable or not.

tcp.2 \mathbf{Q} is C.e.-Complete

inc:tcp:qce:
sec

Theorem tcp.1. \mathbf{Q} is *c.e.* but not decidable. In fact, it is a complete *c.e.* set.

Proof. It is not hard to see that \mathbf{Q} is *c.e.*, since it is the set of (codes for) sentences y such that there is a proof x of y in \mathbf{Q} :

$$Q = \{y : \exists x \text{Prf}_{\mathbf{Q}}(x, y)\}.$$

But we know that $\text{Prf}_{\mathbf{Q}}(x, y)$ is computable (in fact, primitive recursive), and any set that can be written in the above form is c.e.

Saying that it is a complete c.e. set is equivalent to saying that $K \leq_m Q$, where $K = \{x : \varphi_x(x) \downarrow\}$. So let us show that K is reducible to \mathbf{Q} . Since Kleene's predicate $T(e, x, s)$ is primitive recursive, it is representable in \mathbf{Q} , say, by φ_T . Then for every x , we have

$$\begin{aligned} x \in K &\rightarrow \exists s T(x, x, s) \\ &\rightarrow \exists s (\mathbf{Q} \vdash \varphi_T(\bar{x}, \bar{x}, \bar{s})) \\ &\rightarrow \mathbf{Q} \vdash \exists s \varphi_T(\bar{x}, \bar{x}, s). \end{aligned}$$

Conversely, if $\mathbf{Q} \vdash \exists s \varphi_T(\bar{x}, \bar{x}, s)$, then, in fact, for some natural number n the formula $\varphi_T(\bar{x}, \bar{x}, \bar{n})$ must be true. Now, if $T(x, x, n)$ were false, \mathbf{Q} would prove $\neg \varphi_T(\bar{x}, \bar{x}, \bar{n})$, since φ_T represents T . But then \mathbf{Q} proves a false formula, which is a contradiction. So $T(x, x, n)$ must be true, which implies $\varphi_x(x) \downarrow$.

In short, we have that for every x , x is in K if and only if \mathbf{Q} proves $\exists s T(\bar{x}, \bar{x}, s)$. So the function f which takes x to (a code for) the sentence $\exists s T(\bar{x}, \bar{x}, s)$ is a reduction of K to \mathbf{Q} . \square

tcp.3 ω -Consistent Extensions of \mathbf{Q} are Undecidable

explanation

The proof that \mathbf{Q} is c.e.-complete relied on the fact that any sentence provable in \mathbf{Q} is “true” of the natural numbers. The next definition and theorem strengthen this theorem, by pinpointing just those aspects of “truth” that were needed in the proof above. Don't dwell on this theorem too long, though, because we will soon strengthen it even further. We include it mainly for historical purposes: Gödel's original paper used the notion of ω -consistency, but his result was strengthened by replacing ω -consistency with ordinary consistency soon after.

inc:tcp:qcn:
sec

Definition tcp.2. A theory \mathbf{T} is ω -consistent if the following holds: if $\exists x \varphi(x)$ is any sentence and \mathbf{T} proves $\neg \varphi(\bar{0}), \neg \varphi(\bar{1}), \neg \varphi(\bar{2}), \dots$ then \mathbf{T} does not prove $\exists x \varphi(x)$.

inc:tcp:qcn:
thm:oconsis-q

Theorem tcp.3. Let \mathbf{T} be any ω -consistent theory that includes \mathbf{Q} . Then \mathbf{T} is not decidable.

Proof. If \mathbf{T} includes \mathbf{Q} , then \mathbf{T} represents the computable functions and relations. We need only modify the previous proof. As above, if $x \in K$, then \mathbf{T} proves $\exists s \varphi_T(\bar{x}, \bar{x}, s)$. Conversely, suppose \mathbf{T} proves $\exists s \varphi_T(\bar{x}, \bar{x}, s)$. Then x must be in K : otherwise, there is no halting computation of machine x on input x ; since φ_T represents Kleene's T relation, \mathbf{T} proves $\neg \varphi_T(\bar{x}, \bar{x}, \bar{0}), \neg \varphi_T(\bar{x}, \bar{x}, \bar{1}), \dots$, making \mathbf{T} ω -inconsistent. \square

tcp.4 Consistent Extensions of \mathbf{Q} are Undecidable

inc:tcp:cqn:
sec Remember that a theory is *consistent* if it does not prove both φ and $\neg\varphi$ explanation for any formula φ . Since anything follows from a contradiction, an inconsistent theory is trivial: every sentence is provable. Clearly, if a theory is ω -consistent, then it is consistent. But being consistent is a weaker requirement (i.e., there are theories that are consistent but not ω -consistent.). We can weaken the assumption in [Definition tcp.2](#) to simple consistency to obtain a stronger theorem.

Lemma tcp.4. *There is no “universal computable relation.” That is, there is no binary computable relation $R(x, y)$, with the following property: whenever $S(y)$ is a unary computable relation, there is some k such that for every y , $S(y)$ is true if and only if $R(k, y)$ is true.*

Proof. Suppose $R(x, y)$ is a universal computable relation. Let $S(y)$ be the relation $\neg R(y, y)$. Since $S(y)$ is computable, for some k , $S(y)$ is equivalent to $R(k, y)$. But then we have that $S(k)$ is equivalent to both $R(k, k)$ and $\neg R(k, k)$, which is a contradiction. \square

Theorem tcp.5. *Let \mathbf{T} be any consistent theory that includes \mathbf{Q} . Then \mathbf{T} is not decidable.*

Proof. Suppose \mathbf{T} is a consistent, decidable extension of \mathbf{Q} . We will obtain a contradiction by using \mathbf{T} to define a universal computable relation.

Let $R(x, y)$ hold if and only if

$$x \text{ codes a formula } \theta(u), \text{ and } \mathbf{T} \text{ proves } \theta(\bar{y}).$$

Since we are assuming that \mathbf{T} is decidable, R is computable. Let us show that R is universal. If $S(y)$ is any computable relation, then it is representable in \mathbf{Q} (and hence \mathbf{T}) by a formula $\theta_S(u)$. Then for every n , we have

$$\begin{aligned} S(\bar{n}) &\rightarrow T \vdash \theta_S(\bar{n}) \\ &\rightarrow R(\# \theta_S(u)^\#, n) \end{aligned}$$

and

$$\begin{aligned} \neg S(\bar{n}) &\rightarrow T \vdash \neg \theta_S(\bar{n}) \\ &\rightarrow T \not\vdash \theta_S(\bar{n}) \quad (\text{since } \mathbf{T} \text{ is consistent}) \\ &\rightarrow \neg R(\# \theta_S(u)^\#, n). \end{aligned}$$

That is, for every y , $S(y)$ is true if and only if $R(\# \theta_S(u)^\#, y)$ is. So R is universal, and we have the contradiction we were looking for. \square

Let “true arithmetic” be the theory $\{\varphi : \mathbb{N} \models \varphi\}$, that is, the set of sentences in the language of arithmetic that are true in the standard interpretation.

Corollary tcp.6. *True arithmetic is not decidable.*

tcp.5 **Axiomatizable Theories**

A theory \mathbf{T} is said to be *axiomatizable* if it has a computable set of axioms A . (Saying that A is a set of axioms for \mathbf{T} means $T = \{\varphi : A \vdash \varphi\}$.) Any “reasonable” axiomatization of the natural numbers will have this property. In particular, any theory with a finite set of axioms is *axiomatizable*. inc:tcp:cax:
sec

Lemma tcp.7. *Suppose \mathbf{T} is axiomatizable. Then \mathbf{T} is computably enumerable.*

Proof. Suppose A is a computable set of axioms for \mathbf{T} . To determine if $\varphi \in T$, just search for a proof of φ from the axioms.

Put slightly differently, φ is in \mathbf{T} if and only if there is a finite list of axioms ψ_1, \dots, ψ_k in A and a proof of $(\psi_1 \wedge \dots \wedge \psi_k) \rightarrow \varphi$ in first-order logic. But we already know that any set with a definition of the form “there exists ... such that ...” is *c.e.*, provided the second “...” is computable. \square

tcp.6 **Axiomatizable Complete Theories are Decidable**

A theory is said to be *complete* if for every sentence φ , either φ or $\neg\varphi$ is provable. inc:tcp:cdc:
sec

Lemma tcp.8. *Suppose a theory \mathbf{T} is complete and axiomatizable. Then \mathbf{T} is decidable.*

Proof. Suppose \mathbf{T} is complete and A is a computable set of axioms. If \mathbf{T} is inconsistent, it is clearly computable. (Algorithm: “just say yes.”) So we can assume that \mathbf{T} is also consistent.

To decide whether or not a sentence φ is in \mathbf{T} , simultaneously search for a proof of φ from A and a proof of $\neg\varphi$. Since \mathbf{T} is complete, you are bound to find one or another; and since \mathbf{T} is consistent, if you find a proof of $\neg\varphi$, there is no proof of φ .

Put in different terms, we already know that \mathbf{T} is *c.e.*; so by a theorem we proved before, it suffices to show that the complement of \mathbf{T} is *c.e.* also. But a formula φ is in $\bar{\mathbf{T}}$ if and only if $\neg\varphi$ is in \mathbf{T} ; so $\bar{\mathbf{T}} \leq_m \mathbf{T}$. \square

tcp.7 **Q has no Complete, Consistent, Axiomatizable Extensions**

Theorem tcp.9. *There is no complete, consistent, axiomatizable extension of \mathbf{Q} .* inc:tcp:inc:
sec

Proof. We already know that there is no consistent, decidable extension of \mathbf{Q} . But if \mathbf{T} is complete and *axiomatized*, then it is decidable. \square inc:tcp:inc:
thm:first-incompleteness

This theorem is not that far from Gödel's original 1931 formulation of the [First Incompleteness Theorem](#). Aside from the more modern terminology, the key differences are this: Gödel has “ ω -consistent” instead of “consistent”; and he could not say “[axiomatizable](#)” in full generality, since the formal notion of computability was not in place yet. (The formal models of computability were developed over the following decade, including by Gödel, and in large part to be able to characterize the kinds of theories that are susceptible to the Gödel phenomenon.)

The theorem says you can't have it all, namely, completeness, consistency, and [axiomatizability](#). If you give up any one of these, though, you can have the other two: \mathbf{Q} is consistent and computably axiomatized, but not complete; the inconsistent theory is complete, and computably axiomatized (say, by $\{0 \neq 0\}$), but not consistent; and the set of true sentences of arithmetic is complete and consistent, but it is not computably axiomatized.

tcp.8 Sentences Provable and Refutable in \mathbf{Q} are Computably Inseparable

[inc:tcp:ins:](#) Let $\bar{\mathbf{Q}}$ be the set of sentences whose *negations* are provable in \mathbf{Q} , i.e., $\bar{\mathbf{Q}} = \{\varphi : \mathbf{Q} \vdash \neg\varphi\}$. Remember that disjoint sets A and B are said to be computably inseparable if there is no computable set C such that $A \subseteq C$ and $B \subseteq \bar{C}$. [sec](#)

Lemma tcp.10. \mathbf{Q} and $\bar{\mathbf{Q}}$ are computably inseparable.

Proof. Suppose C is a computable set such that $\mathbf{Q} \subseteq C$ and $\bar{\mathbf{Q}} \subseteq \bar{C}$. Let $R(x, y)$ be the relation

x codes a formula $\theta(u)$ and $\theta(\bar{y})$ is in C .

We will show that $R(x, y)$ is a universal computable relation, yielding a contradiction.

Suppose $S(y)$ is computable, represented by $\theta_S(u)$ in \mathbf{Q} . Then

$$\begin{aligned} S(\bar{n}) &\rightarrow \mathbf{Q} \vdash \theta_S(\bar{n}) \\ &\rightarrow \theta_S(\bar{n}) \in C \end{aligned}$$

and

$$\begin{aligned} \neg S(\bar{n}) &\rightarrow \mathbf{Q} \vdash \neg\theta_S(\bar{n}) \\ &\rightarrow \theta_S(\bar{n}) \in \bar{\mathbf{Q}} \\ &\rightarrow \theta_S(\bar{n}) \notin C \end{aligned}$$

So $S(y)$ is equivalent to $R(\#(\theta_S(\bar{u})), y)$. □

tcp.9 Theories Consistent with \mathbf{Q} are Undecidable

The following theorem says that not only is \mathbf{Q} undecidable, but, in fact, [inc:tcp:con:sec](#) any theory that does not disagree with \mathbf{Q} is undecidable.

Theorem tcp.11. *Let \mathbf{T} be any theory in the language of arithmetic that is consistent with \mathbf{Q} (i.e., $\mathbf{T} \cup \mathbf{Q}$ is consistent). Then \mathbf{T} is undecidable.*

Proof. Remember that \mathbf{Q} has a finite set of axioms, Q_1, \dots, Q_8 . We can even replace these by a single axiom, $\alpha = Q_1 \wedge \dots \wedge Q_8$.

Suppose \mathbf{T} is a decidable theory consistent with \mathbf{Q} . Let

$$C = \{\varphi : \mathbf{T} \vdash \alpha \rightarrow \varphi\}.$$

We show that C would be a computable separation of \mathbf{Q} and $\bar{\mathbf{Q}}$, a contradiction. First, if φ is in \mathbf{Q} , then φ is provable from the axioms of \mathbf{Q} ; by the deduction theorem, there is a proof of $\alpha \rightarrow \varphi$ in first-order logic. So φ is in C .

On the other hand, if φ is in $\bar{\mathbf{Q}}$, then there is a proof of $\alpha \rightarrow \neg\varphi$ in first-order logic. If \mathbf{T} also proves $\alpha \rightarrow \varphi$, then \mathbf{T} proves $\neg\alpha$, in which case $\mathbf{T} \cup \mathbf{Q}$ is inconsistent. But we are assuming $\mathbf{T} \cup \mathbf{Q}$ is consistent, so \mathbf{T} does not prove $\alpha \rightarrow \varphi$, and so φ is not in C .

We've shown that if φ is in \mathbf{Q} , then it is in C , and if φ is in $\bar{\mathbf{Q}}$, then it is in \bar{C} . So C is a computable separation, which is the contradiction we were looking for. \square

This theorem is very powerful. For example, it implies:

Corollary tcp.12. *First-order logic for the language of arithmetic (that is, the set $\{\varphi : \varphi \text{ is provable in first-order logic}\}$) is undecidable.*

Proof. First-order logic is the set of consequences of \emptyset , which is consistent with \mathbf{Q} . \square

tcp.10 Theories in which \mathbf{Q} is Intepretable are Undecidable

We can strengthen these results even more. Informally, an interpretation of a language \mathcal{L}_1 in another language \mathcal{L}_2 involves defining the universe, relation symbols, and function symbols of \mathcal{L}_1 with [formulas](#) in \mathcal{L}_2 . Though we won't take the time to do this, one can make this definition precise. [inc:tcp:itp:sec](#)

Theorem tcp.13. *Suppose \mathbf{T} is a theory in a language in which one can interpret the language of arithmetic, in such a way that \mathbf{T} is consistent with the interpretation of \mathbf{Q} . Then \mathbf{T} is undecidable. If \mathbf{T} proves the interpretation of the axioms of \mathbf{Q} , then no consistent extension of \mathbf{T} is decidable.*

The proof is just a small modification of the proof of the last theorem; one could use a counterexample to get a separation of \mathbf{Q} and $\bar{\mathbf{Q}}$. One can take **ZFC**, Zermelo-Fraenkel set theory with the axiom of choice, to be an axiomatic foundation that is powerful enough to carry out a good deal of ordinary mathematics. In **ZFC** one can define the natural numbers, and via this interpretation, the axioms of \mathbf{Q} are true. So we have

Corollary tcp.14. *There is no decidable extension of **ZFC**.*

Corollary tcp.15. *There is no complete, consistent, computably *axiomatizable* extension of **ZFC**.*

The language of **ZFC** has only a single binary relation, \in . (In fact, you don't even need equality.) So we have

Corollary tcp.16. *First-order logic for any language with a binary relation symbol is undecidable.*

This result extends to any language with two unary function symbols, since one can use these to simulate a binary relation symbol. The results just cited are tight: it turns out that first-order logic for a language with only *unary* relation symbols and at most one *unary* function symbol is decidable.

One more bit of trivia. We know that the set of sentences in the language $0, ', +, \times, <$ true in the standard model is undecidable. In fact, one can define $<$ in terms of the other symbols, and then one can define $+$ in terms of \times and $'$. So the set of true sentences in the language $0, ', \times$ is undecidable. On the other hand, Presburger has shown that the set of sentences in the language $0, ', +$ true in the language of arithmetic is decidable. The procedure is computationally infeasible, however.

Photo Credits

Bibliography