Chapter udf

Representability in Q

req.1 Introduction

inc:req:int:sec We will describe a very minimal such theory called “**Q**” (or, sometimes, “Robinson’s Q,” after Raphael Robinson). We will say what it means for a function to be *representable* in **Q**, and then we will prove the following:

> A function is representable in **Q** if and only if it is computable.

For one thing, this provides us with another model of computability. But we will also use it to show that the set $\{\varphi : \mathbf{Q} \vdash \varphi\}$ is not decidable, by reducing the halting problem to it. By the time we are done, we will have proved much stronger things than this.

The language of **Q** is the language of arithmetic; **Q** consists of the following axioms (to be used in conjunction with the other axioms and rules of first-order logic with identity predicate):

$$\forall x\, \forall y\, (x' = y' \rightarrow x = y) \tag{Q_1}$$

$$\forall x\, \mathsf{o} \neq x' \tag{Q_2}$$

$$\forall x\, (x \neq \mathsf{o} \rightarrow \exists y\, x = y') \tag{Q_3}$$

$$\forall x\, (x + \mathsf{o}) = x \tag{Q_4}$$

$$\forall x\, \forall y\, (x + y') = (x + y)' \tag{Q_5}$$

$$\forall x\, (x \times \mathsf{o}) = \mathsf{o} \tag{Q_6}$$

$$\forall x\, \forall y\, (x \times y') = ((x \times y) + x) \tag{Q_7}$$

$$\forall x\, \forall y\, (x < y \leftrightarrow \exists z\, (z' + x) = y) \tag{Q_8}$$

For each natural number n, define the numeral $\overline{n}$ to be the term $0^{\prime\prime\cdots\prime}$ where there are n tick marks in all. So, $\overline{0}$ is the constant symbol o by itself, $\overline{1}$ is o', $\overline{2}$ is o'', etc.

As a theory of arithmetic, **Q** is *extremely* weak; for example, you can’t even prove very simple facts like $\forall x\, x \neq x'$ or $\forall x\, \forall y\, (x + y) = (y + x)$. But we will see that much of the reason that **Q** is so interesting is *because* it is so weak.

In fact, it is just barely strong enough for the incompleteness theorem to hold. Another reason **Q** is interesting is because it has a *finite* set of axioms.

A stronger theory than **Q** (called *Peano arithmetic* **PA**) is obtained by adding a schema of induction to **Q**:

$$(\varphi(\mathsf{0}) \land \forall x\,(\varphi(x) \rightarrow \varphi(x'))) \rightarrow \forall x\,\varphi(x)$$

where $\varphi(x)$ is any formula. If $\varphi(x)$ contains free variables other than x, we add universal quantifiers to the front to bind all of them (so that the corresponding instance of the induction schema is a sentence). For instance, if $\varphi(x, y)$ also contains the variable y free, the corresponding instance is

$$\forall y\,((\varphi(\mathsf{0}) \land \forall x\,(\varphi(x) \rightarrow \varphi(x'))) \rightarrow \forall x\,\varphi(x))$$

Using instances of the induction schema, one can prove much more from the axioms of **PA** than from those of **Q**. In fact, it takes a good deal of work to find "natural" statements about the natural numbers that can't be proved in Peano arithmetic!

Definition req.1. A function $f(x_0, \dots, x_k)$ from the natural numbers to the natural numbers is said to be *representable in* **Q** if there is a formula $\varphi_f(x_0, \dots, x_k, y)$ such that whenever $f(n_0, \dots, n_k) = m$, **Q** proves

inc:req:int: defn:representable-fn

1. $\varphi_f(\overline{n_0}, \dots, \overline{n_k}, \overline{m})$
2. $\forall y\,(\varphi_f(\overline{n_0}, \dots, \overline{n_k}, y) \rightarrow \overline{m} = y)$.

There are other ways of stating the definition; for example, we could equivalently require that **Q** proves $\forall y\,(\varphi_f(\overline{n_0}, \dots, \overline{n_k}, y) \leftrightarrow y = \overline{m})$.

Theorem req.2. *A function is representable in* **Q** *if and only if it is computable.*

inc:req:int: thm:representable-iff-comp

There are two directions to proving the theorem. The left-to-right direction is fairly straightforward once arithmetization of syntax is in place. The other direction requires more work. Here is the basic idea: we pick "general recursive" as a way of making "computable" precise, and show that every general recursive function is representable in **Q**. Recall that a function is general recursive if it can be defined from zero, the successor function succ, and the projection functions P^n_i, using composition, primitive recursion, and regular minimization. So one way of showing that every general recursive function is representable in **Q** is to show that the basic functions are representable, and whenever some functions are representable, then so are the functions defined from them using composition, primitive recursion, and regular minimization. In other words, we might show that the basic functions are representable, and that the representable functions are "closed under" composition, primitive recursion, and regular minimization. This guarantees that every general recursive function is representable.

It turns out that the step where we would show that representable functions are closed under primitive recursion is hard. In order to avoid this step, we show first that in fact we can do without primitive recursion. That is, we show that every general recursive function can be defined from basic functions using composition and regular minimization alone. To do this, we show that primitive recursion can actually be done by a specific regular minimization. However, for this to work, we have to add some additional basic functions: addition, multiplication, and the characteristic function of the identity relation $\chi_=$. Then, we can prove the theorem by showing that all of *these* basic functions are representable in $\mathbf{Q}$, and the representable functions are closed under composition and regular minimization.

req.2 Functions Representable in Q are Computable

inc:req:rpc: sec

Lemma req.3. *Every function that is representable in* $\mathbf{Q}$ *is computable.*

Proof. Let's first give the intuitive idea for why this is true. If $f(x_0, \dots, x_k)$ is representable in $\mathbf{Q}$, there is a formula $\varphi(x_0, \dots, x_k, y)$ such that

$$\mathbf{Q} \vdash \varphi_f(\overline{n_0}, \dots, \overline{n_k}, \overline{m}) \quad \text{iff} \quad m = f(n_0, \dots, n_k).$$

To compute f, we do the following. List all the possible derivations δ in the language of arithmetic. This is possible to do mechanically. For each one, check if it is a derivation of a formula of the form $\varphi_f(\overline{n_0}, \dots, \overline{n_k}, \overline{m})$. If it is, m must be $= f(n_0, \dots, n_k)$ and we've found the value of f. The search terminates because $\mathbf{Q} \vdash \varphi_f(\overline{n_0}, \dots, \overline{n_k}, \overline{f(n_0, \dots, n_k)})$, so eventually we find a δ of the right sort.

This is not quite precise because our procedure operates on derivations and formulas instead of just on numbers, and we haven't explained exactly why "listing all possible derivations" is mechanically possible. But as we've seen, it is possible to code terms, formulas, and derivations by Gödel numbers. We've also introduced a precise model of computation, the general recursive functions. And we've seen that the relation $\mathrm{Prf}_{\mathbf{Q}}(d, y)$, which holds iff d is the Gödel number of a derivation of the formula with Gödel number x from the axioms of $\mathbf{Q}$, is (primitive) recursive. Other primitive recursive functions we'll need are num (**??**) and Subst (**??**). From these, it is possible to define f by minimization; thus, f is recursive.

First, define

$$\begin{aligned} &A(n_0, \dots, n_k, m) = \\ &\qquad \mathrm{Subst}(\mathrm{Subst}(\dots \mathrm{Subst}({}^{\#}\varphi_f{}^{\#}, \mathrm{num}(n_0), {}^{\#}x_0{}^{\#}), \\ &\qquad\qquad \dots), \mathrm{num}(n_k), {}^{\#}x_k{}^{\#}), \mathrm{num}(m), {}^{\#}y^{\#}) \end{aligned}$$

This looks complicated, but it's just the function $A(n_0, \dots, n_k, m) = {}^{\#}\varphi_f(\overline{n_0}, \dots, \overline{n_k}, \overline{m})^{\#}$.

Now, consider the relation $R(n_0, \dots, n_k, s)$ which holds if $(s)_0$ is the Gödel number of a derivation from $\mathbf{Q}$ of $\varphi_f(\overline{n_0}, \dots, \overline{n_k}, \overline{(s)_1})$:

$$R(n_0, \dots, n_k, s) \quad \text{iff} \quad \mathrm{Prf}_{\mathbf{Q}}((s)_0, A(n_0, \dots, n_k, (s)_1)$$

If we can find an s such that $R(n_0, \dots, n_k, s)$ hold, we have found a pair of numbers—$(s)_0$ and (s_1)—such that $(s)_0$ is the Gödel number of a derivation of $A_f(\overline{n_0}, \dots, \overline{n_k}, (s)_1)$. So looking for s is like looking for the pair d and m in the informal proof. And a computable function that "looks for" such an s can be defined by regular minimization. Note that R is regular: for every $n_0, \dots, n_k$, there is a derivation δ of $\mathbf{Q} \vdash \varphi_f(\overline{n_0}, \dots, \overline{n_k}, \overline{f(n_0, \dots, n_k)})$, so $R(n_0, \dots, n_k, s)$ holds for $s = \langle {}^{\#}\delta^{\#}, f(n_0, \dots, n_k)\rangle$. So, we can write f as

$$f(n_0, \dots, n_k) = (\mu s\; R(n_0, \dots, n_k, s))_1.$$

□

req.3 The Beta Function Lemma

In order to show that we can carry out primitive recursion if addition, multiplication, and $\chi_=$ are available, we need to develop functions that handle sequences. (If we had exponentiation as well, our task would be easier.) When we had primitive recursion, we could define things like the "n-th prime," and pick a fairly straightforward coding. But here we do not have primitive recursion—in fact we want to show that we can do primitive recursion using minimization—so we need to be more clever. inc:req:bet:sec

Lemma req.4. *There is a function $\beta(d, i)$ such that for every sequence a_0, …, a_n there is a number d, such that for every $i \leq n$, $\beta(d, i) = a_i$. Moreover, β can be defined from the basic functions using just composition and regular minimization.* *inc:req:bet:lem:beta*

Think of d as coding the sequence $\langle a_0, \dots, a_n\rangle$, and $\beta(d, i)$ returning the i-th element. (Note that this "coding" does *not* use the prower-of-primes coding we're already familiar with!). The lemma is fairly minimal; it doesn't say we can concatenate sequences or append elements, or even that we can *compute* d from a_0, …, a_n using functions definable by composition and regular minimization. All it says is that there is a "decoding" function such that every sequence is "coded."

The use of the notation β is Gödel's. To repeat, the hard part of proving the lemma is defining a suitable β using the seemingly restricted resources, i.e., using just composition and minimization—however, we're allowed to use addition, multiplication, and $\chi_=$. There are various ways to prove this lemma, but one of the cleanest is still Gödel's original method, which used a number-theoretic fact called the Chinese Remainder theorem.

Definition req.5. Two natural numbers a and b are *relatively prime* if their greatest common divisor is 1; in other words, they have no other divisors in common.

Definition req.6. $a \equiv b \mod c$ means $c \mid (a-b)$, i.e., a and b have the same remainder when divided by c.

Here is the *Chinese Remainder theorem*:

Theorem req.7. *Suppose x_0, …, x_n are (pairwise) relatively prime. Let y_0, …, y_n be any numbers. Then there is a number z such that*

$$\begin{aligned} z &\equiv y_0 \mod x_0 \\ z &\equiv y_1 \mod x_1 \\ &\vdots \\ z &\equiv y_n \mod x_n. \end{aligned}$$

Here is how we will use the Chinese Remainder theorem: if x_0, …, x_n are bigger than y_0, …, y_n respectively, then we can take z to code the sequence $\langle y_0, \dots, y_n \rangle$. To recover y_i, we need only divide z by x_i and take the remainder. To use this coding, we will need to find suitable values for x_0, …, x_n.

A couple of observations will help us in this regard. Given y_0, …, y_n, let

$$j = \max(n, y_0, \dots, y_n) + 1,$$

and let

$$\begin{aligned} x_0 &= 1 + j! \\ x_1 &= 1 + 2 \cdot j! \\ x_2 &= 1 + 3 \cdot j! \\ &\vdots \\ x_n &= 1 + (n+1) \cdot j! \end{aligned}$$

Then two things are true:

1. x_0, …, x_n are relatively prime.
2. For each i, $y_i < x_i$.

To see that (1) is true, note that if p is a prime number and $p \mid x_i$ and $p \mid x_k$, then $p \mid 1 + (i+1)j!$ and $p \mid 1 + (k+1)j!$. But then p divides their difference,

$$(1 + (i+1)j!) - (1 + (k+1)j!) = (i-k)j!.$$

Since p divides $1+(i+1)j!$, it can't divide $j!$ as well (otherwise, the first division would leave a remainder of 1). So p divides $i-k$, since p divides $(i-k)j!$. But $|i-k|$ is at most n, and we have chosen $j > n$, so this implies that $p \mid j!$, again a contradiction. So there is no prime number dividing both x_i and x_k. Clause (2) is easy: we have $y_i < j < j! < x_i$.

Now let us prove the β function lemma. Remember that we can use 0, successor, plus, times, $\chi_=$, projections, and any function defined from them using composition and minimization applied to regular functions. We can also use a relation if its characteristic function is so definable. As before we can show that these relations are closed under boolean combinations and bounded quantification; for example:

1. $\mathrm{not}(x) = \chi_=(x, 0)$
2. $(\min x \leq z)\, R(x, y) = \mu x\ (R(x, y) \lor x = z)$
3. $(\exists x \leq z)\ R(x, y) \Leftrightarrow R((\min x \leq z)\, R(x, y), y)$

We can then show that all of the following are also definable without primitive recursion:

1. The pairing function, $J(x, y) = \frac{1}{2}[(x + y)(x + y + 1)] + x$
2. Projections
$$K(z) = (\min x \leq q)\, (\exists y \leq z\, [z = J(x, y)])$$
and
$$L(z) = (\min y \leq q)\, (\exists x \leq z\, [z = J(x, y)]).$$
3. $x < y$
4. $x \mid y$
5. The function $\mathrm{rem}(x, y)$ which returns the remainder when y is divided by x

Now define
$$\beta^*(d_0, d_1, i) = \mathrm{rem}(1 + (i + 1)d_1, d_0)$$
and
$$\beta(d, i) = \beta^*(K(d), L(d), i).$$
This is the function we need. Given $a_0, \dots, a_n$, as above, let
$$j = \max(n, a_0, \dots, a_n) + 1,$$
and let $d_1 = j!$. By the observations above, we know that $1+d_1, 1+2d_1, \dots, 1+(n+1)d_1$ are relatively prime and all are bigger than $a_0, \dots, a_n$. By the Chinese Remainder theorem there is a value d_0 such that for each i,
$$d_0 \equiv a_i \mod (1 + (i + 1)d_1)$$
and so (because d_1 is greater than a_i),
$$a_i = \mathrm{rem}(1 + (i + 1)d_1, d_0).$$
Let $d = J(d_0, d_1)$. Then for each $i \leq n$, we have
$$\begin{aligned} \beta(d, i) &= \beta^*(d_0, d_1, i) \\ &= \mathrm{rem}(1 + (i + 1)d_1, d_0) \\ &= a_i \end{aligned}$$
which is what we need. This completes the proof of the β-function lemma.

req.4 Simulating Primitive Recursion

inc:req:pri:sec

Now we can show that definition by primitive recursion can be "simulated" by regular minimization using the beta function. Suppose we have $f(\vec{z})$ and $g(u, v, \vec{z})$. Then the function $h(x, \vec{z})$ defined from f and g by primitive recursion is

$$\begin{array}{rcl} h(0, \vec{z}) & = & f(\vec{z}) \\ h(x+1, \vec{z}) & = & g(x, h(x, \vec{z}), \vec{z}). \end{array}$$

We need to show that h can be defined from f and g using just composition and regular minimization, using the basic functions and functions defined from them using composition and regular minimization (such as β).

inc:req:pri:lem:prim-rec

Lemma req.8. *If h can be defined from f and g using primitive recursion, it can be defined from f, g, the functions* zero, succ, P^n_i, add, mult, $\chi_=$, *using composition and regular minimization.*

Proof. First, define an auxiliary function $\hat{h}(x, \vec{z})$ which returns the least number d such that d codes a sequence which satisfies

1. $(d)_0 = f(\vec{z})$, and
2. for each $i < x$, $(d)_{i+1} = g(i, (d)_i, \vec{z})$,

where now $(d)_i$ is short for $\beta(d, i)$. In other words, $\hat{h}$ returns the sequence $\langle h(0, \vec{z}), h(1, \vec{z}), \dots, h(x, \vec{z})\rangle$. We can write $\hat{h}$ as

$$\hat{h}(x, z) = \mu d\, (\beta(d, 0) = f(\vec{z}) \land \forall i < x\, \beta(d, i+1) = g(i, \beta(d, i), \vec{z})).$$

Note: no primitive recursion is needed here, just minimization. The function we minimize is regular because of the beta function lemma Lemma req.4.

But now we have

$$h(x, \vec{z}) = \beta(\hat{h}(x, \vec{z}), x),$$

so h can be defined from the basic functions using just composition and regular minimization. □

req.5 Basic Functions are Representable in Q

inc:req:bre:sec

First we have to show that all the basic functions are representable in **Q**. In the end, we need to show how to assign to each k-ary basic function $f(x_0, \dots, x_{k-1})$ a formula $\varphi_f(x_0, \dots, x_{k-1}, y)$ that represents it.

We will be able to represent zero, successor, plus, times, the characteristic function for equality, and projections. In each case, the appropriate representing function is entirely straightforward; for example, zero is represented by the formula $y = \mathsf{0}$, successor is represented by the formula $x_0' = y$, and addition is represented by the formula $(x_0 + x_1) = y$. The work involves showing that

Q can prove the relevant sentences; for example, saying that addition is represented by the formula above involves showing that for every pair of natural numbers m and n, **Q** proves

$$\begin{gathered}\overline{n} + \overline{m} = \overline{n+m} \text{ and} \\ \forall y\, ((\overline{n} + \overline{m}) = y \rightarrow y = \overline{n+m}).\end{gathered}$$

Proposition req.9. *The zero function* $\mathrm{zero}(x) = 0$ *is represented in* **Q** *by* $y = \mathsf{o}$.

inc:req:bre: prop:rep-zero

Proposition req.10. *The successor function* $\mathrm{succ}(x) = x + 1$ *is represented in* **Q** *by* $y = x'$.

inc:req:bre: prop:rep-succ

Proposition req.11. *The projection function* $P^n_i(x_0, \dots, x_{n-1}) = x_i$ *is represented in* **Q** *by* $y = x_i$.

inc:req:bre: prop:rep-proj

Problem req.1. Prove that $y = \mathsf{o}$, $y = x'$, and $y = x_i$ represent zero, succ, and P^n_i, respectively.

Proposition req.12. *The characteristic function of* $=$,

inc:req:bre: prop:rep-id

$$\chi_=(x_0, x_1) = \begin{cases} 1 & \text{if } x_0 = x_1 \\ 0 & \text{otherwise} \end{cases}$$

is represented in **Q** *by*

$$(x_0 = x_1 \land y = \overline{1}) \lor (x_0 \neq x_1 \land y = \overline{0}).$$

The proof requires the following lemma.

Lemma req.13. *Given natural numbers n and m, if $n \neq m$, then* $\mathbf{Q} \vdash \overline{n} \neq \overline{m}$.

inc:req:bre: lem:q-proves-neq

Proof. Use induction on n to show that for every m, if $n \neq m$, then $Q \vdash \overline{n} \neq \overline{m}$.

In the base case, $n = 0$. If m is not equal to 0, then $m = k + 1$ for some natural number k. We have an axiom that says $\forall x\, 0 \neq x'$. By a quantifier axiom, replacing x by $\overline{k}$, we can conclude $0 \neq \overline{k}'$. But $\overline{k}'$ is just $\overline{m}$.

In the induction step, we can assume the claim is true for n, and consider $n+1$. Let m be any natural number. There are two possibilities: either $m = 0$ or for some k we have $m = k + 1$. The first case is handled as above. In the second case, suppose $n + 1 \neq k + 1$. Then $n \neq k$. By the induction hypothesis for n we have $\mathbf{Q} \vdash \overline{n} \neq \overline{k}$. We have an axiom that says $\forall x\, \forall y\, x' = y' \rightarrow x = y$. Using a quantifier axiom, we have $\overline{n}' = \overline{k}' \rightarrow \overline{n} = \overline{k}$. Using propositional logic, we can conclude, in **Q**, $\overline{n} \neq \overline{k} \rightarrow \overline{n}' \neq \overline{k}'$. Using modus ponens, we can conclude $\overline{n}' \neq \overline{k}'$, which is what we want, since $\overline{k}'$ is $\overline{m}$. □

explanation

Note that the lemma does not say much: in essence it says that **Q** can prove that different numerals denote different objects. For example, **Q** proves $0'' \neq 0'''$. But showing that this holds in general requires some care. Note also that although we are using induction, it is induction *outside* of **Q**.

Proof of Proposition req.12. If $n = m$, then $\overline{n}$ and $\overline{m}$ are the same term, and $\chi_=(n, m) = 1$. But $\mathbf{Q} \vdash (\overline{n} = \overline{m} \land \overline{1} = \overline{1})$, so it proves $\varphi_=(\overline{n}, \overline{m}, \overline{1})$. If $n \neq m$, then $\chi_=(n, m) = 0$. By Lemma req.13, $\mathbf{Q} \vdash \overline{n} \neq \overline{m}$ and so also $(\overline{n} \neq \overline{m} \land \mathsf{o} = \mathsf{o})$. Thus $\mathbf{Q} \vdash \varphi_=(\overline{n}, \overline{m}, \overline{0})$.

For the second part, we also have two cases. If $n = m$, we have to show that that $\mathbf{Q} \vdash \forall(\varphi_=(\overline{n}, \overline{m}, y) \rightarrow y = \overline{1})$. Arguing informally, suppose $\varphi_=(\overline{n}, \overline{m}, y)$, i.e.,

$$(\overline{n} = \overline{n} \land y = \overline{1}) \lor (\overline{n} \neq \overline{n} \land y = \overline{0})$$

The left disjunct implies $y = \overline{1}$ by logic; the right contradicts $\overline{n} = \overline{n}$ which is provable by logic.

Suppose, on the other hand, that $n \neq m$. Then $\varphi_=(\overline{n}, \overline{m}, y)$ is

$$(\overline{n} = \overline{m} \land y = \overline{1}) \lor (\overline{n} \neq \overline{m} \land y = \overline{0})$$

Here, the left disjunct contradicts $\overline{n} \neq \overline{m}$, which is provable in $\mathbf{Q}$ by Lemma req.13; the right disjunct entails $y = \overline{0}$. □

inc:req:bre: prop:rep-add **Proposition req.14.** *The addition function* $\mathrm{add}(x_0, x_1) = x_0 + x_1$ *is is represented in* $\mathbf{Q}$ *by*

$$y = (x_0 + x_1).$$

inc:req:bre: lem:q-proves-add **Lemma req.15.** $\mathbf{Q} \vdash (\overline{n} + \overline{m}) = \overline{n + m}$

Proof. We prove this by induction on m. If $m = 0$, the claim is that $\mathbf{Q} \vdash (\overline{n} + \mathsf{o}) = \overline{n}$. This follows by axiom Q_4. Now suppose the claim for m; let's prove the claim for $m + 1$, i.e., prove that $\mathbf{Q} \vdash (\overline{n} + \overline{m + 1}) = \overline{n + m + 1}$. Note that $\overline{m + 1}$ is just $\overline{m}'$, and $\overline{n + m + 1}$ is just $\overline{n + m}'$. By axiom Q_5, $\mathbf{Q} \vdash (\overline{n} + \overline{m}') = (\overline{n} + \overline{m})'$. By induction hypothesis, $\mathbf{Q} \vdash (\overline{n} + \overline{m}) = \overline{n + m}$. So $\mathbf{Q} \vdash (\overline{n} + \overline{m}') = \overline{n + m}'$. □

Proof of Proposition req.14. The formula $\varphi_{\mathrm{add}}(x_0, x_1, y)$ representing add is $y = (x_0 + x_1)$. First we show that if $\mathrm{add}(n, m) = k$, then $\mathbf{Q} \vdash \varphi_{\mathrm{add}}(\overline{n}, \overline{m}, \overline{k})$, i.e., $\mathbf{Q} \vdash \overline{k} = (\overline{n} + \overline{m})$. But since $k = n + m$, $\overline{k}$ just is $\overline{n + m}$, and we've shown in Lemma req.15 that $\mathbf{Q} \vdash (\overline{n} + \overline{m}) = \overline{n + m}$.

We also have to show that if $\mathrm{add}(n, m) = k$, then

$$\mathbf{Q} \vdash \forall y\, (\varphi_{\mathrm{add}}(\overline{n}, \overline{m}, y) \rightarrow y = \overline{k}).$$

Suppose we have $\overline{n} + \overline{m} = y$. Since

$$\mathbf{Q} \vdash (\overline{n} + \overline{m}) = \overline{n + m},$$

we can replace the left side with $\overline{n + m}$ and get $\overline{n + m} = y$, for arbitrary y. □

inc:req:bre: prop:rep-mult **Proposition req.16.** *The multiplication function* $\mathrm{mult}(x_0, x_1) = x_0 \cdot x_1$ *is represented in* $\mathbf{Q}$ *by*

$$y = (x_0 \times x_1).$$

Proof. Exercise. □

Lemma req.17. $\mathbf{Q} \vdash (\overline{n} \times \overline{m}) = \overline{n \cdot m}$ *inc:req:bre: lem:q-proves-mult*

Proof. Exercise. □

Problem req.2. Prove Lemma req.17.

Problem req.3. Use Lemma req.17 to prove Proposition req.16.

req.6 Composition is Representable in Q

Suppose h is defined by *inc:req:cmp: sec*

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

where we have already found formulas $\varphi_f, \varphi_{g_0}, \dots, \varphi_{g_{k-1}}$ representing the functions f, and g_0, $\dots$, g_{k-1}, respectively. We have to find a formula φ_h representing h.

Let's start with a simple case, where all functions are 1-place, i.e., consider $h(x) = f(g(x))$. If $\varphi_f(y, z)$ represents f, and $\varphi_g(x, y)$ represents g, we need a formula $\varphi_h(x, z)$ that represents h. Note that $h(x) = z$ iff there is a y such that both $z = f(y)$ and $y = g(x)$. (If $h(x) = z$, then $g(x)$ is such a y; if such a y exists, then since $y = g(x)$ and $z = f(y)$, $z = f(g(x))$.) This suggests that $\exists y\, (\varphi_g(x, y) \wedge \varphi_f(y, z))$ is a good candidate for $\varphi_h(x, z)$. We just have to verify that $\mathbf{Q}$ proves the relevant formulas.

Proposition req.18. *If $h(n) = m$, then $\mathbf{Q} \vdash \varphi_h(\overline{n}, \overline{m})$.* *inc:req:cmp: prop:rep1*

Proof. Suppose $h(n) = m$, i.e., $f(g(n)) = m$. Let $k = g(n)$. Then

$$\mathbf{Q} \vdash \varphi_g(\overline{n}, \overline{k})$$

since φ_g represents g, and

$$\mathbf{Q} \vdash \varphi_f(\overline{k}, \overline{m})$$

since φ_f represents f. Thus,

$$\mathbf{Q} \vdash \varphi_g(\overline{n}, \overline{k}) \wedge \varphi_f(\overline{k}, \overline{m})$$

and consequently also

$$\mathbf{Q} \vdash \exists y\, (\varphi_g(\overline{n}, y) \wedge \varphi_f(y, \overline{m})),$$

i.e., $\mathbf{Q} \vdash \varphi_h(n, m)$. □

Proposition req.19. *If $h(n) = m$, then $\mathbf{Q} \vdash \forall z\, (\varphi_h(\overline{n}, z) \rightarrow z = \overline{m})$.* *inc:req:cmp: prop:rep2*

Proof. Suppose $h(n) = m$, i.e., $f(g(n)) = m$. Let $k = g(n)$. Then

$$\mathbf{Q} \vdash \forall y\,(\varphi_g(\overline{n}, y) \to y = \overline{k})$$

since φ_g represents g, and

$$\mathbf{Q} \vdash \forall z\,(\varphi_f(\overline{k}, z) \to z = \overline{m})$$

since φ_f represents f. Using just a little bit of logic, we can show that also

$$\mathbf{Q} \vdash \forall z\,(\exists y\,(\varphi_g(\overline{n}, y) \land \varphi_f(y, z)) \to z = \overline{m}).$$

i.e., $\mathbf{Q} \vdash \forall y\,(\varphi_h(\overline{n}, y) \to y = \overline{m})$. □

The same idea works in the more complex case where f and g_i have arity greater than 1.

inc:req:cmp: prop:rep-composition

Proposition req.20. *If $\varphi_f(y_0, \dots, y_{k-1}, z)$ represents $f(y_0, \dots, y_{k-1})$ in $\mathbf{Q}$, and $\varphi_{g_i}(x_0, \dots, x_{l-1}, y)$ represents $g_i(x_0, \dots, x_{l-1})$ in $\mathbf{Q}$, then*

$$\exists y_0, \dots \exists y_{k-1}\,(\varphi_{g_0}(x_0, \dots, x_{l-1}, y_0) \land \dots \land \\ \varphi_{g_{k-1}}(x_0, \dots, x_{l-1}, y_{k-1}) \land \varphi_f(y_0, \dots, y_{k-1}, z))$$

represents

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

Proof. Exercise. □

Problem req.4. Using the proofs of Proposition req.19 and Proposition req.19 as a guide, carry out the proof of Proposition req.20 in detail.

req.7 Regular Minimization is Representable in Q

inc:req:min: sec

Let's consider unbounded search. Suppose $g(x, z)$ is regular and representable in $\mathbf{Q}$, say by the formula $\varphi_g(x, z, y)$. Let f be defined by $f(z) = \mu x\,[g(x, z) = 0]$. We would like to find a formula $\varphi_f(z, y)$ representing f. The value of $f(z)$ is that number x which (a) satisfies $g(x, z) = 0$ and (b) is the least such, i.e., for any $w < x$, $g(w, z) \neq 0$. So the following is a natural choice:

$$\varphi_f(z, y) \equiv \varphi_g(y, z, 0) \land \forall w\,(w < y \to \lnot\varphi_g(w, z, 0)).$$

In the general case, of course, we would have to replace z with z_0, …, z_k.

The proof, again, will involve some lemmas about things $\mathbf{Q}$ is strong enough to prove.

inc:req:min: lem:succ

Lemma req.21. *For every variable x and every natural number n,*

$$\mathbf{Q} \vdash (x' + \overline{n}) = (x + \overline{n})'.$$

Proof. The proof is, as usual, by induction on n. In the base case, $n = 0$, we need to show that $\mathbf{Q}$ proves $(x' + 0) = (x + 0)'$. But we have:

$$\begin{aligned}
&\mathbf{Q} \vdash (x' + 0) = x' \quad \text{by axiom } Q_4 && \text{(req.1)}\\
&\mathbf{Q} \vdash (x + 0) = x \quad \text{by axiom } Q_4 && \text{(req.2)}\\
&\mathbf{Q} \vdash (x + 0)' = x' \quad \text{by eq. (req.2)} && \text{(req.3)}\\
&\mathbf{Q} \vdash (x' + 0) = (x + 0)' \quad \text{by eq. (req.1) and eq. (req.3)}
\end{aligned}$$

inc:req:min:step1 inc:req:min:step2 inc:req:min:step3

In the induction step, we can assume that we have shown that $\mathbf{Q} \vdash (x' + \overline{n}) = (x + \overline{n})'$. Since $\overline{n+1}$ is $\overline{n}'$, we need to show that $\mathbf{Q}$ proves $(x' + \overline{n}') = (x + \overline{n}')'$. We have:

$$\begin{aligned}
&\mathbf{Q} \vdash (x' + \overline{n}') = (x' + \overline{n})' \quad \text{by axiom } Q_5 && \text{(req.4)}\\
&\mathbf{Q} \vdash (x' + \overline{n}') = (x + \overline{n}')' \quad \text{inductive hypothesis} && \text{(req.5)}\\
&\mathbf{Q} \vdash (x' + \overline{n})' = (x + \overline{n}')' \quad \text{by eq. (req.4) and eq. (req.5).}
\end{aligned}$$

inc:req:min:step5 inc:req:min:step6

□

It is again worth mentioning that this is weaker than saying that $\mathbf{Q}$ proves $\forall x\, \forall y\, (x' + y) = (x + y)'$. Although this sentence is true in $\mathfrak{N}$, $\mathbf{Q}$ does not prove it.

Lemma req.22. *inc:req:min:lem:less*

1. $\mathbf{Q} \vdash \forall x\, \neg x < \mathsf{0}$.
2. *For every natural number n,*

$$\mathbf{Q} \vdash \forall x\, (x < \overline{n+1} \rightarrow (x = \mathsf{0} \lor \cdots \lor x = \overline{n})).$$

Proof. Let us do 1 and part of 2, informally (i.e., only giving hints as to how to construct the formal derivation).

For part 1, by the definition of $<$, we need to prove $\neg\exists y\, (y' + x) = \mathsf{0}$ in $\mathbf{Q}$, which is equivalent (using the axioms and rules of first-order logic) to $\forall y\, (y' + x) \neq 0$. Here is the idea: suppose $(y' + x) = \mathsf{0}$. If $x = \mathsf{0}$, we have $(y' + \mathsf{0}) = \mathsf{0}$. But by axiom Q_4 of $\mathbf{Q}$, we have $(y' + \mathsf{0}) = y'$, and by axiom Q_2 we have $y' \neq \mathsf{0}$, a contradiction. So $\forall y\, (y' + x) \neq \mathsf{0}$. If $x \neq \mathsf{0}$, by axiom Q_3, there is a z such that $x = z'$. But then we have $(y' + z') = 0$. By axiom Q_5, we have $(y' + z)' = \mathsf{0}$, again contradicting axiom Q_2.

For part 2, use induction on n. Let us consider the base case, when $n = 0$. In that case, we need to show $x < \overline{1} \rightarrow x = \mathsf{0}$. Suppose $x < \overline{1}$. Then by the defining axiom for $<$, we have $\exists y\, (y' + x) = \mathsf{0}'$. Suppose y has that property; so we have $y' + x = \mathsf{0}'$.

We need to show $x = \mathsf{0}$. By axiom Q_3, if $x \neq \mathsf{0}$, we get $x = z'$ for some z. Then we have $(y' + z') = \mathsf{0}'$. By axiom Q_5 of $\mathbf{Q}$, we have $(y' + z)' = \mathsf{0}'$. By axiom Q_1, we have $(y' + z) = \mathsf{0}$. But this means, by definition, $z < \mathsf{0}$, contradicting part 1. □

inc:req:min: lem:trichotomy

Lemma req.23. *For every $m \in \mathbb{N}$,*

$$\mathbf{Q} \vdash \forall y\, ((y < \overline{m} \lor \overline{m} < y) \lor y = \overline{m}).$$

Proof. By induction on m. First, consider the case $m = 0$. $\mathbf{Q} \vdash \forall y\, (y \neq \mathsf{o} \rightarrow \exists z\, y = z')$ by Q_3. But if $y = z'$, then $(z' + \mathsf{o}) = (y + \mathsf{o})$ by the logic of $=$. By Q_4, $(y + \mathsf{o}) = y$, so we have $(z' + \mathsf{o}) = y$, and hence $\exists z\, (z' + \mathsf{o}) = y$. By the definition of $<$ in Q_8, $\mathsf{o} < y$. If $\mathsf{o} < y$, then also $\mathsf{o} < y \lor y < \mathsf{o}$. We obtain: $y \neq \mathsf{o} \rightarrow (\mathsf{o} < y \lor y < \mathsf{o})$, which is equivalent to $(\mathsf{o} < y \lor y < \mathsf{o}) \lor y = \mathsf{o}$.

Now suppose we have

$$\mathbf{Q} \vdash \forall y\, ((y < \overline{m} \lor \overline{m} < y) \lor y = \overline{m})$$

and we want to show

$$\mathbf{Q} \vdash \forall y\, ((y < \overline{m+1} \lor \overline{m+1} < y) \lor y = \overline{m+1})$$

The first disjunct $y < \overline{m}$ is equivalent (by Q_8) to $\exists z\, (z' + y) = \overline{m}$. If $(z' + y) = \overline{m}$, then also $(z' + y)' = \overline{m}'$. By Q_4, $(z' + y)' = (z'' + y)$. Hence, $(z'' + y) = \overline{m}'$. We get $\exists u\, (u' + y) = \overline{m+1}$ by existentially generalizing on z' and keeping in mind that $\overline{m}'$ is $\overline{m+1}$. Hence, if $y < \overline{m}$ then $y < \overline{m+1}$.

Now suppose $\overline{m} < y$, i.e., $\exists z\, (z' + \overline{m}) = y$. By Q_3 and some logic, we have $z = \mathsf{o} \lor \exists u\, z = u'$. If $z = \mathsf{o}$, we have $(\mathsf{o}' + \overline{m}) = y$. Since $\mathbf{Q} \vdash (\mathsf{o}' + \overline{m}) = \overline{m+1}$, we have $y = \overline{m+1}$. Now suppose $\exists u\, z = u'$. Then:

$$\begin{aligned}
y &= (z' + \overline{m}) && \text{by assumption}\\
(z' + \overline{m}) &= (u'' + \overline{m}) && \text{from } z = u'\\
(u'' + \overline{m}) &= (u' + \overline{m})' && \text{by Lemma req.21}\\
(u' + \overline{m})' &= (u' + \overline{m}') && \text{by } Q_5\text{, so}\\
y &= (u' + \overline{m+1})
\end{aligned}$$

By existential generalization, $\exists u\, (u' + \overline{m+1}) = y$, i.e., $\overline{m+1} < y$. So, if $\overline{m} < y$, then $\overline{m+1} < y \lor y = \overline{m+1}$.

Finally, assume $y = \overline{m}$. Then, since $\mathbf{Q} \vdash (\mathsf{o}' + \overline{m}) = \overline{m+1}$, $(\mathsf{o}' + y) = \overline{m+1}$. From this we get $\exists z\, (z' + y) = \overline{m+1}$, or $y < \overline{m+1}$.

Hence, from each disjunct of the case for m, we can obtain the case for $m+1$. □

inc:req:min: prop:rep-minimization

Proposition req.24. *If $\varphi_g(x, z, y)$ represents $g(x, y)$ in $\mathbf{Q}$, then*

$$\varphi_f(z, y) \equiv \varphi_g(y, z, \mathsf{o}) \land \forall w\, (w < y \rightarrow \lnot\varphi_g(w, z, \mathsf{o})).$$

represents $f(z) = \mu x\, [g(x, z) = 0]$.

Proof. First we show that if $f(n) = m$, then $\mathbf{Q} \vdash \varphi_f(\overline{n}, \overline{m})$, i.e.,

$$\mathbf{Q} \vdash \varphi_g(\overline{m}, \overline{n}, \mathsf{o}) \land \forall w\, (w < \overline{m} \rightarrow \lnot\varphi_g(w, \overline{n}, \mathsf{o})).$$

Since $\varphi_g(x, z, y)$ represents $g(x, z)$ and $g(m, n) = 0$ if $f(n) = m$, we have

$$\mathbf{Q} \vdash \varphi_g(\overline{m}, \overline{n}, \mathsf{o}).$$

If $f(n) = m$, then for every $k < m$, $g(k, n) \neq 0$. So

$$\mathbf{Q} \vdash \neg\varphi_g(\overline{k}, \overline{n}, \mathsf{o}).$$

We get that

$$\mathbf{Q} \vdash \forall w\, (w < \overline{m} \rightarrow \neg\varphi_g(w, \overline{n}, \mathsf{o})). \qquad \text{(req.6)}$$

inc:req:min:rep-less

by Lemma req.22 (by (1) in case $m = 0$ and by (2) otherwise).

Now let's show that if $f(n) = m$, then $\mathbf{Q} \vdash \forall y\, (\varphi_f(\overline{n}, y) \rightarrow y = \overline{m})$. We again sketch the argument informally, leaving the formalization to the reader.

Suppose $\varphi_f(\overline{n}, y)$. From this we get (a) $\varphi_g(y, \overline{n}, \mathsf{o})$ and (b) $\forall w\, (w < y \rightarrow \neg\varphi_g(w, \overline{n}, \mathsf{o}))$. By Lemma req.23, $(y < \overline{m} \lor \overline{m} < y) \lor y = \overline{m}$. We'll show that both $y < \overline{m}$ and $\overline{m} < y$ leads to a contradiction.

If $\overline{m} < y$, then $\neg\varphi_g(\overline{m}, \overline{n}, \mathsf{o})$ from (b). But $m = f(n)$, so $g(m, n) = 0$, and so $\mathbf{Q} \vdash \varphi_g(\overline{m}, \overline{n}, \mathsf{o})$ since φ_g represents g. So we have a contradiction.

Now suppose $y < \overline{m}$. Then since $\mathbf{Q} \vdash \forall w\, (w < \overline{m} \rightarrow \neg\varphi_g(w, \overline{n}, \mathsf{o}))$ by eq. (req.6), we get $\neg\varphi_g(y, \overline{n}, \mathsf{o})$. This again contradicts (a). □

req.8 Computable Functions are Representable in Q

inc:req:crq:sec

Theorem req.25. *Every computable function is representable in* **Q**.

Proof. For definiteness, and using the Church-Turing Thesis, let's say that a function is computable iff it is general recursive. The general recursive functions are those which can be defined from the zero function zero, the successor function succ, and the projection function P^n_i using composition, primitive recursion, and regular minimization. By Lemma req.8, any function h that can be defined from f and g can also be defined using composition and regular minimization from f, g, and zero, succ, P^n_i, add, mult, $\chi_=$. Consequently, a function is general recursive iff it can be defined from zero, succ, P^n_i, add, mult, $\chi_=$ using composition and regular minimization.

We've furthermore shown that the basic functions in question are representable in **Q** (Propositions req.9 to req.12, req.14 and req.16), and that any function defined from representable functions by composition or regular minimization (Proposition req.20, Proposition req.24) is also representable. Thus every general recursive function is representable in **Q**. □

explanation

We have shown that the set of computable functions can be characterized as the set of functions representable in **Q**. In fact, the proof is more general.

From the definition of representability, it is not hard to see that any theory extending **Q** (or in which one can interpret **Q**) can represent the computable functions. But, conversely, in any proof system in which the notion of proof is computable, every representable function is computable. So, for example, the set of computable functions can be characterized as the set of functions representable in Peano arithmetic, or even Zermelo-Fraenkel set theory. As Gödel noted, this is somewhat surprising. We will see that when it comes to provability, questions are very sensitive to which theory you consider; roughly, the stronger the axioms, the more you can prove. But across a wide range of axiomatic theories, the representable functions are exactly the computable ones; stronger theories do not represent more functions as long as they are axiomatizable.

req.9 Representing Relations

inc:req:rel: sec

Let us say what it means for a *relation* to be representable.

inc:req:rel: defn:representing-relations

Definition req.26. A relation $R(x_0, \dots, x_k)$ on the natural numbers is *representable in* **Q** if there is a formula $\varphi_R(x_0, \dots, x_k)$ such that whenever $R(n_0, \dots, n_k)$ is true, **Q** proves $\varphi_R(\overline{n_0}, \dots, \overline{n_k})$, and whenever $R(n_0, \dots, n_k)$ is false, **Q** proves $\lnot\varphi_R(\overline{n_0}, \dots, \overline{n_k})$.

inc:req:rel: thm:representing-rels

Theorem req.27. *A relation is representable in* **Q** *if and only if it is computable.*

Proof. For the forwards direction, suppose $R(x_0, \dots, x_k)$ is represented by the formula $\varphi_R(x_0, \dots, x_k)$. Here is an algorithm for computing R: on input n_0, …, n_k, simultaneously search for a proof of $\varphi_R(\overline{n_0}, \dots, \overline{n_k})$ and a proof of $\lnot\varphi_R(\overline{n_0}, \dots, \overline{n_k})$. By our hypothesis, the search is bound to find one or the other; if it is the first, report "yes," and otherwise, report "no."

In the other direction, suppose $R(x_0, \dots, x_k)$ is computable. By definition, this means that the function $\chi_R(x_0, \dots, x_k)$ is computable. By Theorem req.2, χ_R is represented by a formula, say $\varphi_{\chi_R}(x_0, \dots, x_k, y)$. Let $\varphi_R(x_0, \dots, x_k)$ be the formula $\varphi_{\chi_R}(x_0, \dots, x_k, \overline{1})$. Then for any n_0, …, n_k, if $R(n_0, \dots, n_k)$ is true, then $\chi_R(n_0, \dots, n_k) = 1$, in which case **Q** proves $\varphi_{\chi_R}(\overline{n_0}, \dots, \overline{n_k}, \overline{1})$, and so **Q** proves $\varphi_R(\overline{n_0}, \dots, \overline{n_k})$. On the other hand, if $R(n_0, \dots, n_k)$ is false, then $\chi_R(n_0, \dots, n_k) = 0$. This means that **Q** proves

$$\forall y\,(\varphi_{\chi_R}(\overline{n_0}, \dots, \overline{n_k}, y) \rightarrow y = \overline{0}).$$

Since **Q** proves $\overline{0} \neq \overline{1}$, **Q** proves $\lnot\varphi_{\chi_R}(\overline{n_0}, \dots, \overline{n_k}, \overline{1})$, and so it proves $\lnot\varphi_R(\overline{n_0}, \dots, \overline{n_k})$. □

Problem req.5. Show that if R is representable in **Q**, so is χ_R.

req.10 Undecidability

inc:req:und:sec

We call a theory **T** *undecidable* if there is no computational procedure which, after finitely many steps and unfailingly, provides a correct answer to the question "does **T** prove φ?" for any sentence φ in the language of **T**. So **Q** would be decidable iff there were a computational procedure which decides, given a sentence φ in the language of arithmetic, whether $\mathbf{Q} \vdash \varphi$ or not. We can make this more precise by asking: Is the relation $\mathrm{Prov}_{\mathbf{Q}}(y)$, which holds of y iff y is the Gödel number of a sentence provable in **Q**, recursive? The answer is: no.

Theorem req.28. **Q** *is undecidable, i.e., the relation*

$$\mathrm{Prov}_{\mathbf{Q}}(y) \Leftrightarrow \mathrm{Sent}(y) \land \exists x\, \mathrm{Prf}_{\mathbf{Q}}(x, y)$$

is not recursive.

Proof. Suppose it were. Then we could solve the halting problem as follows: Given e and n, we know that $\varphi_e(n) \downarrow$ iff there is an s such that $T(e, n, s)$, where T is Kleene's predicate from **??**. Since T is primitive recursive it is representable in **Q** by a formula ψ_T, that is, $\mathbf{Q} \vdash \psi_T(\overline{e}, \overline{n}, \overline{s})$ iff $T(e, n, s)$. If $\mathbf{Q} \vdash \psi_T(\overline{e}, \overline{n}, \overline{s})$ then also $\mathbf{Q} \vdash \exists y\, \psi_T(\overline{e}, \overline{n}, y)$. If no such s exists, then $\mathbf{Q} \vdash \lnot\psi_T(\overline{e}, \overline{n}, \overline{s})$ for every s. But **Q** is ω-consistent, i.e., if $\mathbf{Q} \vdash \lnot\varphi(\overline{n})$ for every $n \in \mathbb{N}$, then $\mathbf{Q} \nvdash \exists y\, \varphi(y)$. We know this because the axioms of **Q** are true in the standard model $\mathfrak{N}$. So, $\mathbf{Q} \nvdash \exists y\, \psi_T(\overline{e}, \overline{n}, y)$. In other words, $\mathbf{Q} \vdash \exists y\, \psi_T(\overline{e}, \overline{n}, y)$ iff there is an s such that $T(e, n, s)$, i.e., iff $\varphi_e(n) \downarrow$. From e and n we can compute ${}^{\#}\exists y\, \psi_T(\overline{e}, \overline{n}, y)^{\#}$, let $g(e, n)$ be the primitive recursive function which does that. So

$$h(e, n) = \begin{cases} 1 & \text{if } \mathrm{Prov}_{\mathbf{Q}}(g(e, n)) \\ 0 & \text{otherwise.} \end{cases}$$

This would show that h is recursive if $\mathrm{Prov}_{\mathbf{Q}}$ is. But h is not recursive, by **??**, so $\mathrm{Prov}_{\mathbf{Q}}$ cannot be either. □

Corollary req.29. *First-order logic is undecidable.*

Proof. If first-order logic were decidable, provability in **Q** would be as well, since $\mathbf{Q} \vdash \varphi$ iff $\vdash \omega \to \varphi$, where ω is the conjunction of the axioms of **Q**. □

Photo Credits

Bibliography