Lecture 11
Peano Arithmetic and Primitive Recursion

Michael Beeson

Primitive recursive functions are representable

- This theorem is more difficult than it may appear.
- The main point of confusion for students is the appearance that the proof is more complicated than necessary.
- The difficulties have to do with the coding of sequences.

What has to be proved

We review the definition of representability. We need to construct, for each primitive recursive f, a formula $A(\mathbf{x}, m)$ such that

(i) for all $\mathbf{x}$, $\vdash A(\bar{\mathbf{x}}, \bar{y})$ if $y = f(\mathbf{x})$, and

(ii) for all $\mathbf{x}$, $\vdash \exists! y\, A(\bar{\mathbf{x}}, y)$.

An obvious approach that does not work

- We know that the graph of each primitive recursive function is definable by a bounded arithmetic formula.
- We know that predicates defined by bounded arithmetic formulas are representable.
- Therefore the graph of each primitive recursive function is representable.
- That gives us the formula A we need such that
  (i) for all $\mathbf{x}$, $\vdash A(\bar{\mathbf{x}}, \bar{y})$ if $y = f(\mathbf{x})$.
- But it does not show:
  (ii) for all $\mathbf{x}$, $\vdash \exists! y\, A(\bar{\mathbf{x}}, y)$.
- Showing f is representable is harder than showing its graph is representable.

The correct approach

We proceed by induction on primitive recursive functions.

- We must show that the initial primitive recursive functions (constants and projection functions) are representable.
- Then we must show that if G and $F_1, \ldots, F_k$ are primitive recursive, then the generalized composition $G(F_1(\mathbf{x}), \ldots, F_k(\mathbf{x}))$ is representable.
- Finally we must show that if F is defined by primitive recursion from G and H, and G and H are representable, then so is F.
- The first two tasks are straightforward, and are left as exercises.
- But the third is surprisingly difficult.

The difficult case

Suppose that f is defined by primitive recursion:

$$f(\mathbf{x}, 0) = g(\mathbf{x})$$

$$f(\mathbf{x}, n') = h(\mathbf{x}, n, f(\mathbf{x}, n)).$$

The key observation is this, first expressed informally:

$$\begin{aligned} f(x,n) \quad = \quad & y \leftrightarrow \exists v_0, \ldots, v_n (v_0 = g(x) \\ & \wedge \forall i < n (v_{i+1} = h(x, n, v_i)) \wedge y = v_n) \end{aligned}$$

This formulation makes it clear that what we need is a way to code sequences of numbers as integers. We have already discussed two solutions of this problem, but what we need here is a *representable* way of coding sequences as integers.

Powers-of-primes sequence coding doesn't work here

Suppose we try to do this using the encoding by powers of primes, where for example the sequence $2, 5, 3$ is encoded as $2^{2+1} \cdot 3^{5+1} \cdot 5^{3+1}$. Recall that $(v)_i$ is the power of prime p_i in c, less 1 if that power is positive. Then $(v)_i$ is the i-th member of the sequence coded by v. It seems that we are on the right track:

$$\begin{aligned} f(x,n) &= y \leftrightarrow \exists v((v)_0 = g(x) \\ &\quad \wedge \forall i < n((v)_{i+1} = h(x,n,(v)_i)) \wedge y = v_n) \end{aligned}$$

But for this formula to represent f, we would have to show that the function $g(v,i) = (v)_i$ is representable. How can we do that? It seems to be as difficult as the general problem of showing any primitive recursive function is representable.

The Chinese Remainder Theorem

- Gödel found the solution using the Chinese remainder theorem!
- That is a theorem of elementary number theory, which I am assuming you do not know. Therefore we will learn it today.
- The Chinese remainder theorem says that given any set of relatively prime numbers $m_1, \ldots, m_n$, and and numbers $c_1, \ldots c_n$, we can simultaneously solve the n congruences $x \equiv c_i \mod m_i$.
- For example, given $5, 3, 9$ as the c_i and $2, 5, 17$ as the relatively prime numbers, the Chinese remainder theorem says we can find an x such that $x \equiv 5 \mod 2$, $x \equiv 3 \mod 5$, and $x \equiv 9 \mod 17$.
- $x = 43$ works in this example.
- That leaves us with two tasks: prove the Chinese Remainder Theorem, and show how Gödel used it.
- We'll go first to its use; the proof of the Chinese Remainder Theorem is given on the last slide of this lecture.

How *not* to use the Chinese Remainder Theorem

- It won't do just to take the m_i to be the first n primes, and code $(c_1, \ldots, c_n)$ as the solution x of the congruences $x \equiv c_i \mod m_i$.
- Then the decoding function $(v)_i$ would be $v \mod p_i$.
- But how can we prove that is representable, without first proving that the n-th prime function p_i is representable?
- We are back to square one.
- The Chinese remainder theorem itself is not quite the *entire* trick to this proof.

The β function

To get over this difficulty, we need to define the numbers m_i to use in the Chinese remainder theorem by a formula.

- Gödel defined the m_i, which he called $\delta(d, i)$, by

$$\delta(d, i) := 1 + (i + 1)d$$

- He also defined

$$\beta(c, d, i) = c \mod \delta(d, i)$$

- The sequence $a_0, \ldots, a_n$ is to be coded by the *two* integers c and d such that

$$\beta(c, d, i) = a_i \qquad \text{for } i = 0, 1, \ldots, n$$

- We hope every sequence $a_0, \ldots, a_n$ is encoded by some (c, d).
- The Chinese remainder theorem implies that, provided the numbers $\delta(d, i)$ are relatively prime, for $i = 0, \ldots, n$.

Two things left to prove

- Given $a_0, \ldots, a_n$, we can choose d so that the $\delta(d, i)$ are relatively prime for $i = 0, \ldots, n$.
- β is representable

Choosing d so the $\delta(d, i)$ are relatively prime

- Following Kleene, page 241:
- Let s be the greatest of $n, a_0, \ldots, a_n$, and take $d = s!$.
- Let $d_i = \delta(d, i)$ for $i = 0, 1, \ldots, n$.
- Suppose d_j and d_{j+k} have a factor in common. Then let p be a prime dividing both of them. Then p divides the difference

$$d_{j+k} - d_j \quad = \quad (1 + (j + k + 1)s!) - (1 + (j + 1)s!) \quad = \quad ks!$$

- But p cannot divide $s!$ since it divides $1 + (j + 1)s!$.
- Also p cannot divide k, since $k \leq n \leq s$ and every number less than s divides $s!$; then if p divides k also p divides $s!$, which it does not.
- But p is prime and divides $ks!$, so it must divide either k or $s!$, contradiction.
- That proves that the d_i are indeed relatively prime.

Goal: to show β is representable

- This is proved at the end of § 41 of Kleene, where it may seem a bit mysterious, coming long before it is motivated.
- Recall

$$\begin{aligned} \beta(c,d,i) &= c \mod \delta(d,i) \\ &= \mathrm{rm}(c,(i'\cdot d)') \end{aligned}$$

- So we have to begin by showing $\mathrm{rm}(x,y)$, the remainder of x on division by y, i.e. $x \mod y$, is representable.
- That in turn requires the use of some simpler things, such as $x < y$ and cutoff division.
- We can't just quote the theorem we're trying to prove, namely that every primitive recursive function is representable.

$x < y$ is representable

- $z < y$ is an abbreviation for $\exists v\,(v' + z = y)$.
- It would also work to define it as $\exists v\,(z + v' = y)$, but we follow Kleene.
- We will show that this formula represents $x < y$.
- Later at home, compare our proof to Kleene, p. 196, Example 2.

$x < y$ is representable, first part

- Suppose $a < b$. We must show two things, the first of which is

$$\vdash \exists c(c' + \bar{a} = \bar{b}).$$

- It will suffice to show $\vdash (\bar{k}' + \bar{a} = \bar{b})$ for some k, as the formula with $\exists$ can be derived from the one with $\bar{k}$ by the inference rule known as "$\exists$-introduction."
- The obvious k is $k = b - a - 1$, which is a natural number since $a < b$.
- Now the value of the term $\bar{k}'$ is $k + 1$, which is $b - a$; so the value of the term $\bar{k}' + \bar{a}$ is b, which is also the value of $\bar{b}$.
- Hence the provability of $(\bar{k}' + \bar{a} = \bar{b})$ is a special case of the fact that terms are provably equal to the numerals for their value, which we proved in the last lecture.

$x < y$ is representable, second part

- We have to show that if not $a < b$ (which is the same as $a \geq b$)
$$\vdash \neg\exists c(c' + \bar{a} = \bar{b}).$$
- Suffices to show
$$\vdash c' + \bar{a} \neq \bar{b}.$$
- Note c is a variable, not a numeral.
- We will proceed (on the next slide) by (informal) induction on a.
- We will only use induction informally, we will not use the induction axioms of $\mathbf{PA}$.
- That doesn't matter now, but several weeks from now, it will matter, so we will just take care of it, by paying attention to not using formal induction.
- It's also good to pay attention to the difference between using induction informally (at the meta-level) and using it formally (in $\mathbf{PA}$).

$\mathbf{PA} \vdash c' + \bar{a} \neq \bar{b}$ for $b \leq a$

- We prove by informal induction on a that for all $b \leq a$

$$\vdash c' + \bar{a} \neq \bar{b}.$$

- Note that c is a free variable, not a numeral.
- Base case: The only $b \leq 0$ is $b = 0$. So we have to prove $c' + 0 \neq 0$. But $c' + 0 = c'$ and $c' \neq 0$, done.
- Induction step: Suppose $\vdash c' + \bar{a} \neq \bar{m}$ for all $m \leq a$. We have to prove $\vdash c' + \bar{a}' \neq \bar{k}$ for all $k \leq a + 1$.
- If $k \neq 0$ then $k = m + 1$ for some $m \leq a$ so (provably) $c' + \bar{a} \neq \bar{m}$ and hence $c' + \bar{a}' = (c' + \bar{a})' \neq \bar{m}'$, since if we did have equality then we would have $\vdash c' + \bar{a} \neq m$ by the axiom that successor is one to one, but that is contrary to the induction hypothesis.
- if $k = 0$ then we have to prove $\vdash c' + \bar{a}' \neq 0$.
- But $\vdash c' + \bar{a}' = (c' + \bar{a})' \neq 0$ since 0 is not a successor of anything.
- Done! Notice we did not use any formal induction (within $\mathbf{PA}$); we only used the non-induction axioms of $\mathbf{PA}$.

Cutoff division is representable, first part

By x/y we mean $\lfloor x/y \rfloor$, i.e. truncated integer division, but we specify $x/0 = 0$.

- We will show it is represented by the formula $\exists v, z\,(v \leq x \wedge x = y \cdot v + z \wedge z < y \vee y = 0 \wedge z = x \wedge v = 0)$.
- We have to show

$$\vdash \exists v, z\,(v \leq \bar{x} \wedge \bar{x} = \bar{y} \cdot v + z \wedge z < \bar{y} \vee \bar{y} = 0 \wedge z = \bar{x} \wedge v = 0).$$

- If $y = 0$, take $z = x$ and $v = 0$. Then $\vdash \bar{y} = 0 \wedge \bar{z} = x \wedge \bar{v} = 0$. Done!
- If $y > 0$, let $v = x/y$ and $z = x \mod y$. It suffices to show

$$\vdash \bar{x} = \bar{y} \cdot \bar{v} + \bar{z} \wedge \bar{z} < \bar{y} \wedge \bar{v} \leq \bar{x}.$$

- We have $\vdash \bar{z} < \bar{y}$ since $<$ is representable and $z < y$. Similarly $\vdash \bar{v} \leq \bar{x}$. So it suffices to show $\vdash \bar{x} = \bar{y} \cdot \bar{v} + \bar{z}$. That follows because the terms on both sides have the same value, namely x, and we showed last time that each closed term is provably equal to its value.

Cutoff division is representable, second part

Note, Kleene states this p. 202 but doesn't actually prove it! We have to show

$$\vdash \exists! v\,(\exists z\,(v \leq x \wedge \bar{x} = \bar{y} \cdot v + z \wedge z < \bar{y}) \vee \bar{y} = 0 \wedge z = \bar{x} \wedge v = 0).$$

- The case $y = 0$ is easy (that's why we needed to specify $v = 0$)
- When $y > 0$: As before let $v = x/y$ and $z = x \mod y$; we already showed $\vdash \bar{x} = \bar{y} \cdot \bar{v} + \bar{z} \wedge \bar{z} < \bar{y} \wedge \bar{v} \leq \bar{x}$, so now it suffices to show

$$\vdash (\exists z\,(u \leq \bar{x} \wedge \bar{x} = \bar{y} \cdot u + z \wedge z < \bar{y})) \supset u = \bar{v}.$$

- For that it suffices to show

$$\vdash u \leq \bar{x} \wedge \bar{x} = \bar{y} \cdot u + z \wedge z < \bar{y} \supset u = \bar{v}.$$

Cutoff division continued

We have to prove

$$\vdash u \leq \bar{x} \wedge \bar{x} = \bar{y} \cdot u + z \wedge z < \bar{y} \supset u = \bar{v}.$$

Here again the key is that $u \leq \bar{x}$ is provably equivalent to a disjunction

$$u = 0 \vee \ldots \vee u = \bar{x}$$

and $z < \bar{y}$ is provably equivalent to a disjunction

$$z = 0 \vee \ldots \vee u = \overline{y - 1}.$$

Thus the antecedent of what we have to show provable is provably equivalent to a disjunction of all the cases of putting a particular numeral $\bar{j}$ in for u and a particular numeral $\bar{\ell}$ in for z. So it suffices to prove for each $j \leq x$ and $\ell < y$ that (with $z = x \mod y$ and $v = x/y$)

$$\vdash \bar{x} = \bar{y} \cdot \bar{j} + \bar{\ell} \supset \bar{j} = \bar{v}$$

Cutoff division, continued

We are trying to show that for each $j \leq x$ and $\ell < y$ that (with $z = x \mod y$ and $v = x/y$)

$$\vdash \bar{x} = \bar{y} \cdot \bar{j} + \bar{\ell} \supset \bar{j} = \bar{v}$$

Now if $j = v$, we have $\vdash \bar{j} = \bar{v}$, since $\bar{j}$ and $\bar{v}$ are identical. And if $j \neq v$, the antecedent is refutable, i.e.

$$\vdash \bar{x} \neq \bar{y} \cdot \bar{j} + \bar{z}$$

since the two sides are terms with unequal values. The reason why the values are unequal is that the true quotient v and remainder z are the *only* solution $(j, \ell) = (v, z)$ of the equation, given the constraints $\ell < y$ and $j < x$.

Remainder is representable, first part

- We are following Kleene, p. 203, more or less.
- Define $R(x, y, z)$ to be $\exists w, u, v\,(u + w = x \wedge x = y \cdot w + z \wedge v' + z = y)$.
- With $z < y$ an abbreviation for $\exists v\,(v' + z = y)$, we have $R(x, y, z)$ equivalent to $\exists w\,(w \leq x \wedge x = y \cdot w + z \wedge z < y)$, as shown Kleene p. 203.
- Suppose $z = x \mod y$. We have to show $\vdash R(\bar{x}, \bar{y}, \bar{z})$.
- Since $z < y$ we have $\vdash \bar{z} < \bar{y}$ as shown previously.
- Suffices to show $\vdash \bar{x} = \bar{y} \cdot \bar{w} + \bar{z}$ for $w = \lfloor x/y \rfloor$.
- that follows from the fact that closed terms are provably equal to their values, since the value of the term on the right is x, because $x = y \cdot w + z$.

Remainder is representable, second part

- We have to show that for each x, y,

$$\vdash \exists! z \exists w (w \leq \bar{x} \wedge \bar{x} = \bar{y} \cdot w + z \wedge z < \bar{y})$$

- In view of the last slide, it suffices to show that with $z = x \mod y$,

$$\vdash (\exists v \, (v \leq \bar{x} \wedge \bar{x} = \bar{y} \cdot v + \bar{z} \wedge u < \bar{y})) \supset u = \bar{z}.$$

- Since the $\exists$ is in the antecedent of an implication, it suffices to show

$$\vdash (v \leq \bar{x} \wedge \bar{x} = \bar{y} \cdot v + u \wedge u < \bar{y}) \supset u = \bar{z}.$$

- Here we use again the fact that

$$\vdash u < \bar{y} \equiv u = 0 \vee u = \bar{1} \ldots \vee u = \overline{y - 1}.$$

Remainder is representable, continued

- Recall we're trying to prove, with $z = x \mod y$, that

$$\vdash (v \leq \bar{x} \wedge \bar{x} = \bar{y} \cdot v + u \wedge u < \bar{y}) \supset u = \bar{z}.$$

- It suffices to prove

$$\vdash (\bar{x} = \bar{y} \cdot v + u \wedge (u = 0 \vee \ldots \vee u = \overline{y-1})$$

$$\wedge (v = 0 \vee \ldots \vee v = \bar{x})) \supset u = \bar{z}.$$

- It suffices to prove that for each $j = 0, 1, \ldots y - 1$ and each $v \leq x$, we have

$$\vdash \bar{x} = \bar{y} \cdot \bar{v} + \bar{j} \supset \bar{j} = \bar{z}.$$

- if $j = z$ that is clear.
- if $j \neq z$ it suffices to show

$$\vdash \bar{x} \neq \bar{y} \cdot \bar{v} + \bar{j}$$

But that follows, because the two sides are terms with unequal values (since $j \neq z = x \mod y$).

β is representable

Recall

$$\begin{aligned} \beta(c,d,i) &= c \mod \delta(d,i) \\ &= \mathrm{rm}(c,(i'\cdot d)') \end{aligned}$$

So β is a composition of representable functions, and therefore representable. But since we left that lemma as an exercise, let us be a bit more explicit:

- To prove β is representable, we need a formula $B(c,d,i,w)$ such that

  (i) if $w=\beta(c,d,i)$ then $\vdash B(\bar{c},\bar{d},\bar{i},\bar{w})$, and

  (ii) $\vdash \exists! w\,(B(\bar{c},\bar{,i},w))$
- We take $B(c,d,i,w)$ to be $R(c,(i'\cdot d)',w)$ where R represents $\mathrm{rm}(x,y)$.
- We do not check every detail; see Kleene p. 204 if you want to see them.

Comparison to Kleene

- In Kleene, the proof that every primitive recursive function is representable is preceded by the somewhat easier proof that it is definable by a bounded arithmetic predicate (Kleene does not state "bounded", but the proof proves it).
- We have already derived that theorem another way, by showing that every primitive recursive function is Turing computable, and hence given by a normal form using the $\mathbf{T}$-predicate, and the $\mathbf{T}$-predicate is definable by a bounded arithmetical formula.
- For that we used another coding of sequences, a supposedly more "modern" one using bits and binary expansions.
- But if we were to try to use this approach to prove that every primitive recursive function is representable, we would still run into the same difficulty, of needing to encode a quantifier over sequences as a single quantifier over numbers in a representable way.
- Conclusion: the β function still needs to be separately proved representable.

Every primitive recursive function and relation is representable in $\mathbf{PA}$

That just restates the theorem that we started out to prove, and finally succeeded to prove.

Every primitive recursive function and relation is definable in $\mathbf{PA}$

- We proved this already, by showing that the primitive recursive functions are Turing computable, and the Turing computable functions are definable by bounded formulas.
- But now that we have Gödel's β-function, we could prove it again, without mentioning Turing machines, directly by induction on the definition of primitive recursive functions.
- The same formula that was constructed in the previous proof to represent f also defines it.
- It is easier to prove definability than representability, as we don't need to check that facts about division and remainder are provable.

Every μ-recursive function is representable in $\mathbf{PA}$

Technically we defined "representable" only for total functions. But changing $=$ to $\cong$ in the definition, we can speak about representability of partial functions.

Every μ-recursive function has the form

$$f(\mathbf{x}) \cong \mu y\, R(\mathbf{x}, y)$$

where R is a primitive recursive relation. Let $A(\mathbf{x}, y)$ represent R. Then

$$B(\mathbf{x}, y) := A(\mathbf{x}, y) \wedge \forall z < y\, \neg A(\mathbf{x}, z)$$

represents f.

- If $f(\mathbf{x}) = y$, then $R(\mathbf{x}, y)$ and for all $z < y$, not $R(\mathbf{x}, y)$. Hence $\vdash A(\bar{x}, \bar{y})$.
- Since $z < \bar{y}$ is equivalent to the disjunction of the $\bar{z} < \bar{y}$ for $z < y$, and the formulas $A(\bar{z}, \bar{y})$ are all refutable for $z < y$, we have $\vdash \neg\exists z < \bar{y} A(z, \bar{y})$.
- Hence $\vdash B(\mathbf{x}, y)$.

The representable functions are all computable

Let f (which takes n variables $\mathbf{x}$) be represented by a formula $A(\mathbf{x}, y)$.

- To compute $f(\mathbf{x})$, we search for a y and a proof of $A(\bar{x}, \bar{y})$.
- Explicitly, for $k = 1, 2, \ldots$, we examine all proofs of length at most k to see if one of them ends in $A(\bar{x}, \bar{y})$ for some y.
- To make this into a proof that representable functions are μ-recursive, we only need to assign numbers to proofs somehow.
- We will do that next time.

Summary

The same class of partial functions is defined by

- The μ-recursive functions
- The functions representable in $\mathbf{PA}$
- The Turing-computable partial functions

The Chinese Remainder Theorem

Let $m_1, \ldots m_n$ and $c_1, \ldots c_n$ be given. We want to solve $x = c_j \mod m_j$ simultaneously for $j = 1, 2, \ldots, n$.

- Let N the product of all the m_i and $X = \{1, 2, \ldots, N-1\}$.
- Let Y be the set of n-tuples $\langle a_1, \ldots a_n \rangle$ with $a_i < m_i$.
- Define $F : X \to Y$ by $F(x) = \langle x \mod m_1, \ldots, x \mod m_n \rangle$.
- There are N elements of X and N elements of Y.
- F is one-to-one, since if $F(x) = F(y)$ with $y < x$, that means $x = y \mod m_i$ for each $i \leq n$, so $x - y$ is a multiple of each m_i. Since the m_i are relatively prime, $x - y$ is a multiple of their product N; but that contradicts $x < N$ and $y < N$.
- By the pigeon-hole principle, F is onto, i.e. every member of Y has the form $F(x)$.
- But that is exactly the conclusion of the Chinese remainder theorem.

Remarks about the Chinese Remainder Theorem

- The algorithm implicit in that proof is just a brute-force search.
- It is possible to do much better.
- For the purposes of this course, we don't care.
- You can find out more at the Wikipedia or Math World articles.
- The Chinese remainder theorem can be stated in $\mathbf{PA}$, using sequence numbers to code the sequences.
- Even the pigeonhole principle can be stated and proved in $\mathbf{PA}$, using sequences to code finite functions.
- In the exercise, you are not asked to do this, but you are asked whether we need to do it.