Lecture #7 —Continued

0.0.0.1 Proposition. *If $R(\vec{x}, y, \vec{z}) \in \mathcal{PR}_*$ and $\lambda\vec{w}.f(\vec{w}) \in \mathcal{PR}$, then $R(\vec{x}, f(\vec{w}), \vec{z})$ is in $\mathcal{PR}_*$.*

Proof. By lemma, let $g \in \mathcal{PR}$ such that

$$R(\vec{x}, y, \vec{z}) \equiv g(\vec{x}, y, \vec{z}) = 0, \text{ for all } \vec{x}, y, \vec{z}$$

Then

$$R(\vec{x}, f(\vec{w}), \vec{z}) \equiv g(\vec{x}, f(\vec{w}), \vec{z}) = 0, \text{ for all } \vec{x}, \vec{w}, \vec{z}$$

By the lemma, and since $\lambda\vec{x}\vec{w}\vec{z}.g(\vec{x}.f(\vec{w}), \vec{z}) \in \mathcal{PR}$ by Grzegorczyk Ops, we have that $R(\vec{x}, f(\vec{w}), \vec{z}) \in \mathcal{PR}_*$. $\square$

0.0.0.2 Proposition. *If* $R(\vec{x}, y, \vec{z}) \in \mathcal{R}_*$ *and* $\lambda\vec{w}.f(\vec{w}) \in \mathcal{R}$*, then* $R(\vec{x}, f(\vec{w}), \vec{z})$ *is in* $\mathcal{R}_*$*.*

Proof. Similar to that of 0.0.0.1. $\square$

0.0.0.3 Corollary. *If $f \in \mathcal{PR}$ (respectively, in $\mathcal{R}$), then its* graph, $z = f(\vec{x})$ *is in $\mathcal{PR}_*$ (respectively, in $\mathcal{R}_*$).*

Proof. Using the relation $z = y$ and 0.0.0.1. $\square$

0.0.0.4 Exercise. Using unbounded search, prove that if $z = f(\vec{x})$ is in $\mathcal{R}_*$ and f is total, then $f \in \mathcal{R}$. $\square$

0.0.0.5 Definition. (Bounded Quantifiers) The abbreviations

$(\forall y)_{<z} R(z, \vec{x})$

$(\forall y)_{y<z} R(z, \vec{x})$

$(\forall y < z) R(z, \vec{x})$

all stand for

$(\forall y)\big(y < z \to R(z, \vec{x})\big)$

while correspondingly,

$(\exists y)_{<z} R(z, \vec{x})$

$(\exists y)_{y<z} R(z, \vec{x})$

$(\exists y < z) R(z, \vec{x})$

all stand for

$(\exists y)\big(y < z \wedge R(z, \vec{x})\big)$

Similarly for the non strict inequality "$\leq$". $\square$

0.0.0.6 Theorem. *$\mathcal{PR}_*$ is closed under bounded quantification.*

Proof. By logic it suffices to look at the case of $(\exists y)_{<z}$ since $(\forall y)_{<z} R(y, \vec{x}) \equiv \neg(\exists y)_{<z} \neg R(y, \vec{x})$.

Let then $R(y, \vec{x}) \in \mathcal{PR}_*$ and *let us give the name $Q(z, \vec{x})$ to*

$(\exists y)_{<z} R(y, \vec{x})$ for convenience.

We note that $Q(0, \vec{x})$ is false (why?) and logic says:

$Q(z + 1, \vec{x}) \equiv Q(z, \vec{x}) \vee R(z, \vec{x})$.

Thus, as the following prim. rec. shows, $c_Q \in \mathcal{PR}$.

$$\begin{aligned} c_Q(0, \vec{x}) &= 1 \\ c_Q(z + 1, \vec{x}) &= c_Q(z, \vec{x}) c_R(z, \vec{x}) \end{aligned}$$

□

0.0.0.7 Corollary. *$\mathcal{R}_*$ is closed under bounded quantification.*

Lecture #8 —Oct.5

0.0.0.8 Definition. (Bounded Search) Let f be a <u>**total**</u> number-theoretic function of $n+1$ variables.

The symbol $(\mu y)_{<z} f(y, \vec{x})$, for all $z, \vec{x}$, stands for

$$\begin{cases} \min\{y : y < z \wedge f(y, \vec{x}) = 0\} & \text{if } (\exists y)_{<z} f(y, \vec{x}) = 0 \\ z & \text{otherwise} \end{cases}$$

So, unsuccessful search returns the first number to the right of the search-range.

We define "$(\mu y)_{\leq z}$" to mean "$(\mu y)_{<z+1}$". □

0.0.0.9 Theorem. *$\mathcal{PR}$ is closed under the bounded search operation $(\mu y)_{<z}$. That is, if $\lambda y\vec{x}.f(y,\vec{x}) \in \mathcal{PR}$, then $\lambda z\vec{x}.(\mu y)_{<z}f(y,\vec{x}) \in \mathcal{PR}$.*

Proof. Set $g = \lambda z\vec{x}.(\mu y)_{<z}f(y,\vec{x})$ for convenience.

Then the following primitive recursion settles it:

*Recall that "**if** $R(\vec{z})$ **then** y **else** w" <u>means</u> "**if** $c_R(\vec{z}) = 0$ **then** y **else** w".*

$$0, 1, 2, \ldots, z-1, z = \overbrace{0, 1, 2, \ldots, z-1}, z$$

So

$$g(0, \vec{x}) = 0$$

Why 0 above?

$$\begin{aligned} g(z+1, \vec{x}) = \text{ if } & (\exists y)_{<z}\Big(f(y,\vec{x}) = 0\Big) \text{ then } g(z,\vec{x}) \\ & \text{else if } f(z,\vec{x}) = 0 \text{ then } z \\ & \text{else } z+1 \end{aligned}$$

$\square$

0.0.0.10 Corollary. *$\mathcal{PR}$ is closed under the bounded search operation $(\mu y)_{\leq z}$.*

0.0.0.11 Exercise. *Prove the corollary.* □

0.0.0.12 Corollary. *$\mathcal{R}$ is closed under the bounded search operations $(\mu y)_{<z}$ and $(\mu y)_{\leq z}$.*

Consider now a set of *mutually exclusive* relations $R_i(\vec{x})$, $i = 1, \ldots, n$, that is, $R_i(\vec{x}) \wedge R_j(\vec{x})$ *is* false*, for each* $\vec{x}$ *as long as* $i \neq j$.

Then we can define a function f *by cases* R_i from given functions f_j by the requirement (for all $\vec{x}$) given below:

$$f(\vec{x}) = \begin{cases} f_1(\vec{x}) & \text{if } R_1(\vec{x}) \\ f_2(\vec{x}) & \text{if } R_2(\vec{x}) \\ \ldots & \ldots \\ f_n(\vec{x}) & \text{if } R_n(\vec{x}) \\ f_{n+1}(\vec{x}) & \text{otherwise} \end{cases}$$

where, as is usual in mathematics, "if $R_j(\vec{x})$" is short for "if $R_j(\vec{x})$ is true"

and the *"otherwise" is the condition* $\neg(R_1(\vec{x}) \vee \cdots \vee R_n(\vec{x}))$.

We have the following result:

0.0.0.13 Theorem. (Definition by Cases) *If the functions f_i, $i = 1, \ldots, n+1$ and the relations $R_i(\vec{x})$, $i = 1, \ldots, n$ are in $\mathcal{PR}$ and $\mathcal{PR}_*$, respectively, then so is f above.*

Proof. By repeated use (Grz Ops) of if-then-else. So,

$$
\begin{aligned}
f(\vec{x}) = \text{if} \quad & R_1(\vec{x}) \text{ then } f_1(\vec{x}) \\
\text{else if } & R_2(\vec{x}) \text{ then } f_2(\vec{x}) \\
\vdots \\
\text{else if } & R_n(\vec{x}) \text{ then } f_n(\vec{x}) \\
\text{else } & \qquad\qquad\quad f_{n+1}(\vec{x})
\end{aligned}
$$

$\square$

0.0.0.14 Corollary. Same statement as above, replacing $\mathcal{PR}$ and $\mathcal{PR}_*$ by $\mathcal{R}$ and $\mathcal{R}_*$, respectively.

The tools we now have at our disposal allow easy certification of the primitive recursiveness of some very useful functions and relations. But first a definition:

0.0.0.15 Definition. $(\mu y)_{<z} R(y, \vec{x})$ means $(\mu y)_{<z} c_R(y, \vec{x})$. $\square$

Thus, if $R(y, \vec{x}) \in \mathcal{PR}_*$ (resp. $\in \mathcal{R}_*$), then $\lambda z \vec{x}.(\mu y)_{<z} R(y, \vec{x}) \in \mathcal{PR}$ (resp. $\in \mathcal{R}$), since $c_R \in \mathcal{PR}$ (resp. $\in \mathcal{R}$).

0.0.0.16 Example. *The following are in $\mathcal{PR}$ or $\mathcal{PR}_*$ as appropriate*:

(1) $\lambda xy.\lfloor x/y \rfloor$[1] *(the quotient of the division x/y).*

This is another example of a nontotal function with an "obvious" way to remove the points where it is undefined (recall $\lambda xy.x^y$).

Thus the symbol "$\lfloor x/y \rfloor$"

is *extended* to *mean*

$$(\mu z)_{\le x}\big((z+1)y > x\big) \qquad (*)$$

for all x, y.

▶ Pause. **Why** is the above expression correct?

Because setting $z = \lfloor x/y \rfloor$ we have

[1]For any real number x, the symbol "$\lfloor x \rfloor$" is called the *floor* of x. It succeeds in the literature (with the same definition) the so-called "greatest integer function, $[x]$", i.e., the *integer part* of the real number x. Thus, **by definition**, $\lfloor x \rfloor \le x < \lfloor x \rfloor + 1$.

$$z \le \frac{x}{y} < z + 1$$

by the definition of "$\lfloor \ldots \rfloor$".

Thus, z is *smallest* such that $x/y < z + 1$, or such that $x < y(z + 1)$. ◀

It follows that, for $y > 0$, the search in $(*)$ yields the "normal math" value for $\lfloor x/y \rfloor$, while it re-defines $\lfloor x/0 \rfloor$ as $= x + 1$.

(2) *$\lambda xy.rem(x, y)$ (the remainder of the division x/y).*

$$rem(x, y) = x \dot{-} y\lfloor x/y \rfloor.$$

(3) *$\lambda xy.x|y$ (x divides y).*

$x|y \equiv rem(y,x) = 0.$

Note that if $y > 0$, we cannot have $0|y$ —*a good thing!*— since $rem(y,0) = y > 0$.

Our redefinition of $\lfloor x/y \rfloor$ yields, however, $0|0$, but we can live with this in practice.

(4) *$Pr(x)$ (x is a prime).*

$$Pr(x) \equiv x > 1 \wedge (\forall y)_{\leq x}(y|x \rightarrow y = 1 \vee y = x).$$

(5) $\pi(x)$ *(the number of primes $\leq x$).*[2]

The following primitive recursion certifies the claim:

$\pi(0) = 0,$

and

$\pi(x + 1) =$ if $Pr(x + 1)$ then $\pi(x) + 1$ else $\pi(x)$.

[2]The π-function plays a central role in number theory, figuring in the so-called *prime number theorem*. See, for example, [LeV56].

(6) *$\lambda n.p_n$ (the nth prime).*

First note that the graph $y = p_n$ is primitive recursive:

$y = p_n \equiv Pr(y) \wedge \pi(y) = n + 1.$

Next note that, for all n,

$p_n \leq 2^{2^n}$ (see Exercise 0.0.0.18 below),

thus $p_n = (\mu y)_{\leq 2^{2^n}}(y = p_n)$,

which settles the claim.

(7) $\lambda nx.\,\exp(n,x)$ *(*the *exponent of p_n in the prime factorization of x).*

$$\exp(n,x) = (\mu y)_{\leq x} \neg (p_n^{y+1} | x).$$

► Is x a good bound? **Yes!** $x = \ldots p_n^y \ldots \geq p_n^y \geq 2^y > y$.

(8) *$Seq(x)$ (x's prime number factorization contains at least one prime, but no gaps).*

$Seq(x) \equiv x > 1 \wedge (\forall y)_{\leq x}(\forall z)_{\leq x}(Pr(y) \wedge Pr(z) \wedge y < z \wedge z|x \to y|x)$. $\square$

0.0.0.17 Remark. *What makes* $\exp(n,x)$ "the *exponent of* p_n *in the prime factorization of* x", *rather than* an *exponent, is Euclid's prime number factorization theorem:* Every number $x>1$ has a unique factorization —within permutation of factors— as a product of primes. □

0.0.0.18 Exercise. Prove by induction on n, that for all n we have $p_n \leq 2^{2^n}$.

Hint. Consider, as Euclid did,[3] $p_0 p_1 \cdots p_n + 1$. If this number is prime, then it is greater than or equal to p_{n+1} (why?). If it is composite, then none of the primes up to p_n divide it. So any prime factor of it is greater than or equal to p_{n+1} (why?). $\square$

[3] In his proof that there are infinitely many primes.

Lecture #9, Oct. 7

0.1 CODING Sequences

0.1.0.1 Definition. (Coding Sequences) Any sequence of numbers, $a_0, \ldots, a_n$, $n \geq 0$, is *coded* by the number denoted by the symbol

$$\langle a_0, \ldots, a_n \rangle$$

and defined as $\prod_{i \leq n} p_i^{a_i+1}$ □

Example. Code 1, 0, 3. I get $2^{1+1}3^{0+1}5^{3+1}$

For *coding* to be useful, we need a simple *decoding* scheme.

By Remark 0.0.0.17 there is no way to have $z = \langle a_0, \ldots, a_n \rangle = \langle b_0, \ldots, b_m \rangle$, *unless*

(i) $n = m$

and

(ii) For $i = 0, \ldots, n$, $a_i = b_i$.

Thus, it makes sense to correspondingly define the decoding expressions:

(*i*) *$lh(z)$ (pronounced "length of z") as shorthand for* $(\mu y)_{\leq z} \neg(p_y | z)$

▶ *A **comment** and a **question***:

- **The comment**: If p_y is the first prime NOT in the decomposition of z, and $Seq(z)$ holds, then since numbering of primes starts at 0, *the length of the coded sequence z is indeed y.*

- **Question**: Is the bound z sufficient? **Yes!**

$$z = 2^{a+1} 3^{b+1} \ldots p_{y \dot{-} 1}^{exp(y \dot{-} 1, z)} \geq \underbrace{2 \cdot 2 \cdots 2}_{y \text{ times}} = 2^y > y$$

(*ii*) *$(z)_i$ is shorthand for* $\exp(i, z) \dot{-} 1$

Note that

(a) *$\lambda iz.(z)_i$ and $\lambda z.lh(z)$ are in $\mathcal{PR}$.*

(b) If $Seq(z)$, then $z = \langle a_0, \ldots, a_n \rangle$ for some $a_0, \ldots, a_n$. In this case, $lh(z)$ equals the number of distinct primes in the decomposition of z, that is, the length $n+1$ of the coded sequence. Then $(z)_i$, for $i < lh(z)$, equals a_i. For larger i, $(z)_i = 0$. Note that if $\neg Seq(z)$ then $lh(z)$ need not equal the number of distinct primes in the decomposition of z. For example, 10 has 2 primes, but $lh(10) = 1$.

The tools lh, $Seq(z)$, and $\lambda iz.(z)_i$ are sufficient to perform *decoding*, primitive recursively, once the truth of $Seq(z)$ is established. This coding/decoding scheme is essentially that of [Göd31], and will be the one we use throughout these notes.

0.1.1 Simultaneous Primitive Recursion

Start with total h_i, g_i for $i = 0, 1, \ldots, k$. Consider the new functions f_i defined by the following "*simultaneous primitive recursion schema*" for all $x, \vec{y}$.

$$\begin{cases} f_0(0, \vec{y}) & = h_1(\vec{y}) \\ \vdots & \\ f_k(0, \vec{y}) & = h_k(\vec{y}) \\ f_0(x+1, \vec{y}) & = g_0(x, \vec{y}, f_0(x, \vec{y}), \ldots, f_k(x, \vec{y})) \\ \vdots & \\ f_k(x+1, \vec{y}) & = g_k(x, \vec{y}, f_0(x, \vec{y}), \ldots, f_k(x, \vec{y})) \end{cases} \tag{2}$$

Hilbert and Bernays proved the following:

0.1.1.1 Theorem. If all the h_i and g_i are in $\mathcal{PR}$ (resp. $\mathcal{R}$), then so are all the f_i obtained by the schema (2) of simultaneous recursion.

Proof. Define, for all $x, \vec{y}$,

$$F(x, \vec{y}) \stackrel{\text{Def}}{=} \langle f_0(x, \vec{y}), \ldots, f_k(x, \vec{y}) \rangle$$

$$H(\vec{y}) \stackrel{\text{Def}}{=} \langle h_0(\vec{y}), \ldots, h_k(\vec{y}) \rangle$$

$$G(x, \vec{y}, z) \stackrel{\text{Def}}{=} \langle g_0(x, \vec{y}, (z)_0, \ldots, (z)_k), \ldots, g_k(x, \vec{y}, (z)_0, \ldots, (z)_k) \rangle$$

We readily have that $H \in \mathcal{PR}$ (resp. $\in \mathcal{R}$) and $G \in \mathcal{PR}$ (resp. $\in \mathcal{R}$) depending on where we assumed the h_i and g_i to be. We can now rewrite schema (2) (p.28) as

$$\begin{cases} F(0, \vec{y}) & = H(\vec{y}) \\ F(x+1, \vec{y}) & = G\Big(x, \vec{y}, F\big(x, \vec{y}\big)\Big) \end{cases} \tag{3}$$

► The 2nd line of (3) is obtained from

$$\begin{aligned} F(x+1, \vec{y}) &= \langle f_0(x+1, \vec{y}), \ldots, f_k(x+1, \vec{y}) \rangle \\ &= \Big\langle g_0\Big(x, \vec{y}, f_0(x, \vec{y}), \ldots, f_k(x, \vec{y})\Big), \ldots, g_k\Big(\text{same as } g_0\Big) \Big\rangle \\ &= \Big\langle g_0\Big(x, \vec{y}, \big(F(x, \vec{y})\big)_0, \ldots, \big(F(x, \vec{y})\big)_k\Big), \ldots, g_k\Big(\text{same as } g_0\Big) \Big\rangle \end{aligned}$$

By the above remarks, $F \in \mathcal{PR}$ (resp. $\in \mathcal{R}$) depending on where we assumed the h_i and g_i to be. In particular, this holds for each f_i since, for all $x, \vec{y}$, $f_i(x, \vec{y}) = \big(F(x, \vec{y})\big)_i$. $\square$

0.1.1.2 Example. We saw one way to justify that $\lambda x.rem(x, 2) \in \mathcal{PR}$ in 0.0.0.16. A direct way is the following. Setting $f(x) = rem(x, 2)$, for all x, we notice that the sequence of outputs (for $x = 0, 1, 2, \ldots$) of f is

$$0, 1, 0, 1, 0, 1 \ldots$$

Thus, the following primitive recursion shows that $f \in \mathcal{PR}$:

$$\begin{cases} f(0) & = 0 \\ f(x+1) & = 1 \dot{-} f(x) \end{cases}$$

Here is a way, via simultaneous recursion, to obtain a proof that $f \in \mathcal{PR}$, without using any arithmetic! Notice the infinite "matrix"

$$\begin{matrix} 0 & 1 & 0 & 1 & 0 & 1 & \ldots \\ 1 & 0 & 1 & 0 & 1 & 0 & \ldots \end{matrix}$$

Let us call g the function that has as its sequence outputs the entries of the second row—obtained by shifting the first row by one position to the left. The first rowstill represents our f. Now

$$\begin{cases} f(0) & = 0 \\ g(0) & = 1 \\ f(x+1) & = g(x) \\ g(x+1) & = f(x) \end{cases} \tag{1}$$

□

0.1.1.3 Example. We saw one way to justify that $\lambda x.\lfloor x/2 \rfloor \in \mathcal{PR}$ in 0.0.0.16. A direct way is the following.

$$\begin{cases} \left\lfloor \dfrac{0}{2} \right\rfloor & = 0 \\ \left\lfloor \dfrac{x+1}{2} \right\rfloor & = \left\lfloor \dfrac{x}{2} \right\rfloor + rem(x,2) \end{cases}$$

where rem is in $\mathcal{PR}$ by 0.1.1.2.

Alternatively, here is a way that can do it —via simultaneous recursion— and with only the knowledge of how to add 1. Consider the matrix

$$\begin{matrix} 0 & 0 & 1 & 1 & 2 & 2 & 3 & 3 & \ldots \\ 0 & 1 & 1 & 2 & 2 & 3 & 3 & 4 & \ldots \end{matrix}$$

The top row represents $\lambda x.\lfloor x/2 \rfloor$, let us call it "$f$". The bottom row we call g and is, again, the result of shifting row one to the left by one position. Thus, we have a simultaneous recursion

$$\begin{cases} f(0) & = 0 \\ g(0) & = 0 \\ f(x+1) & = g(x) \\ g(x+1) & = f(x) + 1 \end{cases} \tag{2}$$

$\square$

Bibliography

[Dav65] M. Davis, *The undecidable*, Raven Press, Hewlett, N. Y., 1965.

[Göd31] K. Gödel, *Über formal unentsceidbare sätze der pricipia mathematica und verwandter systeme i*, Monatshefte für Math. und Physic **38** (1931), 173–198, (Also in English in Davis [Dav65, 5–38]).

[LeV56] William J. LeVeque, *Topics in number theory*, vol. I, Addison-Wesley, Reading, Massachusetts, 1956.