19

(13) $Rm(m,n)$:= remainder r of m when divided by n

:= $m \pmod n$ when $n > 0$

= 0 $n = 0$

Pf: $Rm(m,n) = \mu r \leq n-1 \, (\exists q \leq m \, (m = nq + r))$

everything p.r.

(14) $Qt(m,n)$:= quotient q of m when divided by n

Pf: you try.

- Sps $A = \{\sigma_1\}$ so $A^* = \mathbb{N}$
- recall from last time:
  $n+m$, $n \cdot m$, n^m all p.r. functions on $\mathbb{N}$
  $m|n$ p.r. relation
- $Rm(m,n)$:= rem of m when divided by n
  $Qt(m,n)$:= quotient of " "
  p.r. too

- Let's show for $Rm(m,n)$
  More explicitly, we define
  $Rm(m,n)$:= remainder of m/n when $n \neq 0$
  = 0 o/w

i.e. $Rm(m,n) := \begin{cases} \mu r \le (n-1)(\exists q \le m\,(m = nq + r)) & \text{if } n \neq 0 \\ 0 & \text{if } n = 0 \end{cases}$

$\Rightarrow$ pr since everything p.r. and piecewise conditions are p.r.

- Define unary relation Prime(n) on $\mathbb{N}$ by: Prime(n) iff n is prime

<u>Claim</u>. Prime(n) is p.r.

Pf: notice: Prime(n) iff $n \geq 2 \wedge \forall x \leq n\,(x | n \Rightarrow (x = 1 \vee x = n))$

so Prime(n) is Boolean combo of p.r. rel'ns, hence p.r.

(Note: for rel'ns P, Q
$P \Rightarrow Q$ shorthand for $\neg P \vee Q$
... so "$P \Rightarrow Q$" is p.r. if P, Q are)

- Define p_n = nth prime
($p_0 = 2$, $p_1 = 3$, ...)

- WTS: p_n is p.r. (i.e. function $p: \mathbb{N} \to \mathbb{N}$, $p(n) = p_n$ is p.r.)

- need some prelims

<u>Claim</u> $p_n \leq 2^{2^n}$

<u>Pf</u>: - $p_0 = 2 \leq 2^{2^0}$
- Spse true for n
- Euclid's arg: none of $p_0, p_1, \ldots, p_n$ divide $1 + \prod_{i=0}^{n} p_i$

$$\Rightarrow p_{n+1} \leq 1 + \prod_{i=0}^{n} p_i \qquad \text{(why?)}$$

$$\overset{\text{induct'n}}{\leq} 1 + \prod_{0}^{n} 2^{2^i} = 1 + 2^{\sum 2^i} \leftarrow \text{geom. series}$$

$$= 1 + 2^{(2^{n+1}-1)}$$

$$\leq 2^{2^{n+1}} \checkmark$$

→ will allow us to do bdd min'n when searching for primes

<u>note</u>: $p_{n+1} = \mu t \leq 2^{2^{n+1}} \left(\text{Prime}(t) \wedge p_n < t\right)$

this is a recursive def'n of p_{n+1} in terms of p_n -- let's see if we can turn into official p.r. def'n.

- want to define:

$$p_0 = 2$$

$$p_{n+1} = f(p_n, n) \qquad \text{where } f \text{ p.r.}$$

... so what is f?

shorthand: $x+1 \le t$

- define $g(x,y) := \mu t \le y \, (\text{Prime}(t) \wedge x<t)$
- g is p.r. (bdd min'n of pr rel'n)
- define $f(x,n) := g(x, 2^{2^n})$
  $= \mu t \le 2^{2^n} \, (\text{Prime}(t) \wedge x<t)$
- f is p.r. (2^{2^n} is pr since exp'n p.r. then comp'n)
- note: $f(p_n, n) = \mu t \le 2^{2^n} \, (\text{Prime}(t) \wedge p_n < t)$
  ↳ now above def'n really does define p_n as p.r. function!

Coding

- idea we'll use a lot: encoding a sequence as a natural number
- given a sequence $n_0, n_1, \ldots, n_k$ of nat'l nums (can think of as tuple $(n_0, \ldots, n_k)$) define the code $\langle n_0, n_1, \ldots, n_k \rangle$ as follows:

$$\langle \emptyset \rangle = 0$$

← notation: 0 is the code for empty seq not $\langle 0 \rangle$

$$\langle n_0, \ldots, n_k \rangle = p_0^{n_0+1} p_1^{n_1+1} \cdots p_k^{n_k+1} = \prod_{i=0}^{k} p_i^{n_i+1}$$

- eg $\langle 5,0,1 \rangle = 2^{5+1} 3^{0+1} 5^{1+1} = 2^6 \cdot 3 \cdot 5^2 = 4800$
  ↳ we add +1's in exponents in case 0 appears in sequence

- note: for a fixed k, function F_k defined by $F_k(n_0,..,n_k) = \langle n_0,..,n_k\rangle$ is p.r. (why?) (you try!)
- doesn't make sense to talk abt function that takes a sequence $(n_0,..,n_k)$ of arbitrary length k to $\langle n_0,..,n_k\rangle$ as being pr... pr functions take as inputs tuples of a fixed length.

- Can also consider decoding function

$D(n,i) = (n)_i :=$ ith coord of sequence coded by n (when n codes a seq)

will use both notations

$:= -1 + ($exponent of p_i in prime factorization of $n)$ if $n>1$

truncated subtraction

$:= 0$ if $n=0,1$

Claim $D(n,i)$ is p.r.

Pf. $D(n,i) = [\underbrace{\mu k \leq n\,(p_i^k \mid n)}_{\text{bdd max'n of pr rel'n}}] - 1$

- note. $D(n,i) = 0$ in both case when exponent of p_i is 1 and 0.
- point is: when n is code $\langle n_0,..,n_k\rangle$ for a seq of length $k \geq i$, $D(n,i)$ outputs ith entry of this sequence correctly.

on $\mathbb{N}$

- Define unary rel'n: $Seg(n)$ iff n is a code
  i.e. $Seg(n)$ iff $(n=0 \vee \forall i \le n \, \forall j \le i \, (p_i | n \Rightarrow p_j | n))$
  p.r. bdd p.r.

- note Seg is p.r.
- e.g. $Seg(2^3 3^1 5^2)$ holds
  $\langle 2, 0, 1 \rangle$
  but $\neg Seg(2^3 \cdot 5^2)$
  not a code for a sequence

- let $lh(n) :=$ length of sequence coded by n (when n a code)
  0 (o/w)

- i.e. $lh(n) = 1 + (\mu k \le n \, (p_k | n \wedge \neg (p_{k+1} | n)))$ if $Seg(n) \wedge n \neq 0$
  0 o/w

- observe lh is p.r. function (piecewise p.r. def'n)
- point is $lh(\langle n_0, \ldots, n_k \rangle) = k+1$
- there is some ambiguity: $lh(n) = 0$ both when n is code for empty sequence (i.e. $n=0$) and when not even a code.

- next w.t.s operation that "concatenates codes" is p.r.
- need following general result

<u>Prop'n</u> Sps $f(y, \vec{x})$ is p.r. (on $\mathbb{N}$)
Then $g(n, \vec{x}) := \prod_{i=0}^{n} f(i, \vec{x}) = f(0,\vec{x}) \cdot f(1,\vec{x}) \cdots f(n,\vec{x})$

and $h(n, \vec{x}) := \sum_{i=0}^{n} f(i, \vec{x}) = f(0,\vec{x}) + \cdots + f(n,\vec{x})$

are both p.r.

<u>pf.</u> e.g. can define g by
$$g(0,\vec{x}) = f(0,\vec{x})$$
$$g(n+1,\vec{x}) = g(n,\vec{x}) \cdot f(n+1,\vec{x})$$
which is p.r. def'n ($\cdot$ is p.r.)

similar for h. ✓

<u>Prop'n</u> (concatenating codes) define a binary function $\circledast$ on $\mathbb{N}$ by:
$$s \circledast t = \langle n_0, \ldots, n_k, m_0, \ldots, m_l \rangle \text{ when } s = \langle n_0, \ldots, n_k \rangle \quad t = \langle m_0, \ldots, m_l \rangle \text{ are codes}$$
$$= 0 \text{ else}$$

<u>then</u> $\circledast$ is p.r.

<u>pf.</u> if $s = \langle n_0, \ldots, n_k \rangle$ and $t = \langle m_0, \ldots, m_l \rangle$

$(s)_i$ $\qquad\qquad\qquad\qquad$ $(t)_i$

$$= \prod_{0}^{k} p_i^{n_i+1} \qquad = \prod_{0}^{l} p_i^{m_i+1}$$
$$= \prod_{0}^{lh(s)-1} p_i^{n_i+1} \qquad = \prod_{0}^{lh(t)-1} p_i^{m_i+1}$$

then $s \circledast t = \langle n_0, \ldots, n_k, m_0, \ldots, m_l \rangle$
$$= \left(\prod_{0}^{lh(s)-1} p_i^{n_i+1} \right) \left(\prod_{lh(s)}^{lh(s)+lh(t)-1} p_i^{m_i+1} \right)$$

$$= s \cdot \prod_{0}^{\text{len}(t)-1} p_{\text{len}(s)+i}^{(t)_i + 1}$$

hence can define in general:

$$s * t = \begin{cases} s \cdot \prod_{0}^{\text{len}(t)-1} p_{\text{len}(s)+i}^{(t)_i+1} & \text{if } Seq(s) \text{ and } Seq(t) \\ 0 & \text{else} \end{cases}$$

which is p.r. ✓

Simultaneous recursion

Motivating Q: how to set recursion s.t. $f(n+1)$ depends not only on $f(n)$ but multiple prev. outputs (e.g. $f(n), f(n-1)$)

- Given p.r. functions g_1, g_2, h_1, h_2
Sps we simultaneously define f_1, f_2:

$$f_1(0, \vec{x}) = g_1(\vec{x})$$
$$f_2(0, \vec{x}) = g_2(\vec{x})$$

$$f_1(n+1, \vec{x}) = h_1(f_1(n, \vec{x}), f_2(n, \vec{x}), n, \vec{x})$$
$$f_2(n+1, \vec{x}) = h_2(f_1(n, \vec{x}), f_2(n, \vec{x}), n, \vec{x})$$

Claim: f_1, f_2 also p.r.

Pf: to see first define
$$f(n, \vec{x}) = \langle f_1(n, \vec{x}), f_2(n, \vec{x}) \rangle$$

– Suff to show f is p.r. since then:

$f_1(n,\vec{x}) = (f(n,\vec{x}))_0$ — decoding is p.r.

$f_2(n,\vec{x}) = (f(n,\vec{x}))_1$ — p.r.

– can define f by prim rec'n:

$f(0,\vec{x}) = \langle g_1(\vec{x}), g_2(\vec{x}) \rangle$ $(= G(\vec{x})$ where G p.r.$)$

$f(n+1,\vec{x}) = \langle h_1((f(n,\vec{x}))_0, (f(n,\vec{x}))_1, n, \vec{x}),$
$h_2(\text{same}) \rangle$

$(= H(f(n,\vec{x}), n, \vec{x})$ where H p.r.$)$

✓

<u>Ex</u>: The Fibonacci sequence is p.r.

i.e. define

$f(0) = 0$
$g(0) = 1$

} f is f_1, g is f_2 in above; g_1 is 0, g_2 is 1

$f(n+1) = g(n)$
$g(n+1) = f(n) + g(n)$

} $h_1(f(n), g(n), n) = g(n)$; $h_2(f(n), g(n), n) = f(n)+g(n)$

✓

notice $g(n) = a_n$ = nth term of Fib.

this is a simultaneous recursion!