Course of values Recursion

Q: what about recursion when $f(n+1)$ depends on all prev values $f(i)$?

- given $g: \mathbb{N}^{k+2} \to \mathbb{N}$
define $f: \mathbb{N}^{k+1} \to \mathbb{N}$ recursively by

code for empty string

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

think $n = 0$

$$f(n+1, \vec{x}) = g(\langle f(0,\vec{x}), f(1,\vec{x}), \ldots, f(n,\vec{x})\rangle, n, \vec{x})$$

so: $f(0, \vec{x}) = g(0, 0, \vec{x})$
$f(1, \vec{x}) = g(\langle f(0,\vec{x})\rangle, 0, \vec{x})$ starts clicking up when $n=2$
$f(2, \vec{x}) = g(\langle f(0,\vec{x}), f(1,\vec{x})\rangle, 1, \vec{x})$
⋮

g playing role of both g + h in usual recursion

WTS: if g is p.r. then so is f.

Pf. - define $\bar{f}(n, \vec{x}) = \langle f(0,\vec{x}), f(1,\vec{x}), \ldots, f(n,\vec{x})\rangle$
- so then $f(n, \vec{x}) = (\bar{f}(n, \vec{x}))_{n+1}$

- we show $\bar{f}$ is p.r., which gives f is pr by prev line (can you spell out?)

~~...~~ can define $\bar{f}$ by:

$$\bar{f}(0,\vec{x}) = \langle f(0,\vec{x})\rangle \swarrow G(\vec{x})$$

$$= \langle g(0,0,\vec{x})\rangle \quad (= F_1(g(0,0,\vec{x})))$$

in notn above)

p.r.

$$\bar{f}(n+1,\vec{x}) = \bar{f}(n,\vec{x}) \circledast \langle f(n+1,\vec{x})\rangle$$

$$= \underbrace{\bar{f}(n,\vec{x}) \circledast \langle g(\bar{f}(n,\vec{x}),n,\vec{x})\rangle}_{H(\bar{f}(n,\vec{x}),n,\vec{x})}$$

H pr since a p.r. combo

$\Rightarrow \bar{f}$ p.r. as claimed

$\Rightarrow f$ p.r.

R $f(n+1)$ depends on $f(0),\dots,f(n)$

<u>Equivalence of alphabets</u> ... Going back to A^* from $\mathbb{N}$

- next idea: working w/ numbers (i.e. when $A = \{\sigma_1\}$, $A^* \text{ "=" } \mathbb{N}$) just as good as working w/ words (i.e. when $A = \{\sigma_1,\dots,\sigma_p\}$) ... up to a primitive recursive coding.

now for words

- let A_p denote $\{\sigma_1,\dots,\sigma_p\}$
(so $A_1 = \{\sigma_1\}$, $A_2 = \{\sigma_1,\sigma_2\}$ etc)

- fix p and sps $w = \sigma_{i_0}\sigma_{i_1}\cdots\sigma_{i_n}$ is a A_p word (i.e. $i_k \le p \ \forall k$)

- define $C_p(w) = i_0 p^n + i_1 p^{n-1} + \cdots + i_{n-1} p + i_n$

$$= \sum_{k=0}^{n} i_k p^{n-k}$$

(and when $\omega = \emptyset$ define $c_p(\omega) = 0$)
= empty word

- We have $c_p : A_p^* \to \mathbb{N}$ $(= A_1^*)$
- actually: c_p a bijection of A_p^* w/ $\mathbb{N}$
  ↪ can think of a word $\sigma_{i_0}\sigma_{i_1}\cdots\sigma_{i_n} = \omega$ as "base p" rep'n of $c_p(\omega)$ ... but in the digits $\{1, 2, .., p\}$ instead of $\{0, 1, .., p-1\}$

- more explicitly:

$$\emptyset \overset{c_p}{\mapsto} 0$$

σ_1 $\quad$ $\sigma_1\sigma_1$ ... etc

$$\begin{array}{cccc}
\sigma_1 & \sigma_2 & \cdots & \sigma_p \\
\downarrow & \downarrow & & \downarrow \\
1 & 2 & & p
\end{array}$$
length 1 words $\to [1, p]$

$$\begin{array}{cccc}
\sigma_1\sigma_1 & \sigma_1\sigma_2 & \cdots & \sigma_1\sigma_p \\
\downarrow & \downarrow & & \downarrow \\
p+1 & p+2 & & p+p = 2p \\
\sigma_2\sigma_1 & & \cdots & \sigma_2\sigma_p \\
\downarrow & & & \downarrow \\
2p+1 & & & 3p \\
\vdots & & & \\
\sigma_p\sigma_1 & & \cdots & \sigma_p\sigma_p \\
\downarrow & & & \downarrow \\
p^2+1 & & & p^2+p
\end{array}$$
length 2 $\to [p+1, p^2+p]$

$$\begin{array}{ccc}
\sigma_1\sigma_1\sigma_1 & \cdots & \sigma_1\sigma_1\sigma_p \\
\downarrow & & \downarrow \\
p^2+p+1 & & p^2+2p \\
\vdots & & \\
\sigma_p\sigma_p\sigma_1 & \cdots & \sigma_p\sigma_p\sigma_p \\
\downarrow & & \downarrow \\
p^3+p^2+1 & & p^3+p^2+p
\end{array}$$
length 3 $\to [p^2+p+1, p^3+p^2+p]$

Claim C_p is pr (viewing C_p as from A_p^* to A_p^*)

pf: can define:

$C_p(\epsilon) = 0$

$C_p(w\sigma_j) = C_p(\overbrace{\sigma_{u_0} \cdots \sigma_{u_n}}^{w} \sigma_j)$

$= C_p(w) \cdot p + j$

- mixing not'n here, defined in $N = A_1^*$ not general A_p^*
- really mean (in orig. not'n)

$= C_p(w)^p \sigma_1^j$

$= \mathrm{Con}_2(C_p(w)^p, \sigma_1^j)$ ✓

everything p.r.

- Let $D_p : N \to A^*$ denote decoding function, i.e. $D_p = C_p^{-1}$

- WTS D_p is pr ... need to extend to all of A_p^* for this to be meaningful

- Define $\overline{D}_p : A_p^* \to A_p^*$ by $\overline{D}_p(w) = D_p(|w|)$

↳ here $|w|$ = length of w

= word obtained by repl. every letter in w w/ σ_1

e.g. $|\sigma_{c_0} \cdots \sigma_{c_n}| = \sigma_1\sigma_1 \cdots \sigma_1 = n+1$

- side note: $|\cdot|$ is pr since
  $|w\sigma_j| = |w|\sigma_1$

- notice: if $w \in N$ then $|w| = w$
  so that $\overline{D_p} \restriction N = D_p$

Fact $\overline{D_p}$ is pr

Pf: omitted (requires a bit of work)

↳ using C_p and D_p, can code functions f on A_p^* as functions $\hat{f}$ on $N = A_1^*$
↳ Sps $f: (A_p^*)^n \to A_p^*$, define $\hat{f}: N^n \to N$ by
$$\hat{f}(k_1, \ldots, k_n) = C_p(f(D_p(k_1), \ldots, D_p(k_n)))$$

Claim f is pr iff $\hat{f}$ is pr.

Pf: - first extend $\hat{f}$ to $\bar{\hat{f}}: (A_p^*)^n \to A_p^*$
$$\bar{\hat{f}}(w_1, \ldots, w_n) = \hat{f}(|w_1|, \ldots, |w_n|)$$
which $= C_p(f(\overline{D_p}(w_1), \ldots, \overline{D_p}(w_n)))$
which is pr since comp'n of pr
- note $\hat{f} = \bar{\hat{f}} \restriction N^n$
- hence claim follows from prop'n below.

↳ claim says: equiv to study computability of $N = A_1^*$ as on general A_p^*
(tho mb sometimes conceptually clearer to work on A_p^*)

$\hookrightarrow$ up to a pr coding / decoding can translate between $\mathbb{N}$ and A_p^*

Fact Sps $A \subseteq B$ are finite alphabets (think $A = A_p$ $B = A_q$ for $p \le q$)
Then a function $f : (A^*)^n \to A^*$ is pr (in A^*) iff there is a pr $g : (B^*)^n \to B^*$ s.t. $g \upharpoonright A^* = f$.

Pf: (Sketch): ($\Rightarrow$) if f p.r. on A induct on construction of f from initial functions to get g on B that restricts to f: every time use prim recurs'n, just erase (or do anything else) when adding a symbol $\sigma \in B \setminus A$.

($\Leftarrow$) if g pr on B and $g \upharpoonright A^* = f$ then whatever rules we used in constructing g from initial functions only outputs in A^* on inputs from A^* ... so can just run same construction to define f from initial functions (ignore rules when appending $\sigma \in B \setminus A$) ✓

$\hookrightarrow$ Fact says: a p.r. function f on A^* can be viewed as p.r. even if we add symbols to A

More general "bounded quantifiers"

- Given a pr rel'n $P(z, \vec{x})$ on A_p^* define a rel'n Q by:

$$Q(y, \vec{x}) \text{ iff } \exists z \, (|z| \leq |y| \wedge P(z, \vec{x}))$$

- WTS: Q also p.r.

- notice: if $y = \sigma_{i_0} \cdots \sigma_{i_n}$, so that $|y| = n+1$ then among words w of length $n+1$, $w = \sigma_p \sigma_p \cdots \sigma_p$ maximizes $C_p(w)$

have $C_p(\sigma_p^{n+1}) = p^{n+1} + p^n + \cdots + p$

- So: $\forall z$ we have $|z| \leq |y|$ iff $C_p(z) \leq C_p(\sigma_p^{n+1})$

$= [\cdots]$

- so: have $Q(y, \vec{x})$

iff $\exists z \, (|z| \leq |y| \; P(y, \vec{x}))$

iff $\exists t \leq C_p(\sigma_p^{n+1}) \; P(D(t), \vec{x})$

could write as $\leq$ too

- Let $rep_p : A^* \to A^*$ be function that replaces every symbol in a word w/ σ_p

i.e. $rep_p(\sigma_{i_0} \cdots \sigma_{i_n}) = \sigma_p \sigma_p \cdots \sigma_p$

- then rep_p is pr (why?)

- So: rewriting: $Q(y, \vec{x})$ iff

$$\exists t \leq C_p(rep_p(y)) \; P(D(t), \vec{x})$$

which is pr since combo of p.r. rel $\leq$, i.e. "t is initial seg of $C_p(rep_p(y))$"

Recursive Functions

- we introduce a more general notion of recursive function
- A = fixed alphabet
  A^* = words on A as before.

Def'n A partial n-ary function is a function $f: B \to A^*$ where B ($= \mathrm{dom}(f)$) is a subset of $(A^*)^n$.
f is total if $B = (A^*)^n$

- pr functions are total by def'n
- the empty function $\emptyset$ is a (legit) partial function
- if f, g partial funcs we write $f = g$ to mean $\mathrm{dom}(f) = \mathrm{dom}(g)$
  $\wedge$ $f(\vec{x}) = g(\vec{x})$ $\forall \vec{x} \in \mathrm{dom}(f)$
- we write $f(\vec{x})\downarrow$ to mean $\vec{x} \in \mathrm{dom}(f)$
  $f(\vec{x})\uparrow$ " $\vec{x} \notin$ "

Notation: we'll sometime write $f: (A^*)^n \to A^*$ even when f is only partial (i.e. $\mathrm{dom}(f) \subsetneq (A^*)^n$)

Composition rules for recursive functions

(i) substitution/composition: given partial functions $h_1, \ldots, h_m : (A^*)^n \to A^*$ and a partial $g: (A^*)^m \to A^*$

form $f : (A^*)^n \to A^*$ by:

$f(x_1, .., x_n) = g(h_1(x_1,..,x_n), .., h_m(x_1,..,x_n))$

(ii) Primitive recursion: given partial n-ary g and partial $(n+2)$-ary h_i (for each $\sigma_i \in A$) form f by:

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

$f(y\sigma_i, \vec{x}) = h_i(f(y, \vec{x}), y, \vec{x})$

then: $f(0, \vec{x})\downarrow$ iff $g(\vec{x})\downarrow$

$f(y\sigma_i, \vec{x})\downarrow$ iff $f(y, \vec{x})\downarrow$ and $h_i(f(y, \vec{x}), y, \vec{x})\downarrow$

(iii) (unbounded) minimalization: given $g : (A^*)^{n+1} \to A^*$ say that f is obtained by minimalization of g over $\{\sigma_j\}^*$ and write $f(\vec{x}) = \mu_j y\, [g(y, \vec{x}) = 0]$

if ① $f(\vec{x})\downarrow$ iff $\exists m$ s.t. $g(\sigma_j^m, \vec{x}) = 0$ and $\forall n \leq m$ have $g(\sigma_j^n, \vec{x})\downarrow$

② $f(\vec{x}) = \sigma_j^m$ where $g(\sigma_j^m, \vec{x}) = 0$ and $\forall n < m$ $g(\sigma_j^n, \vec{x})\downarrow$ but $g(\sigma_j^n, \vec{x}) \neq 0$.

note: in case when $A^* = \{\sigma_j\}^*$ we have $f(\vec{x}) = \mu y (g(y, \vec{x}) = 0)$

$=$ least n s.t. $g(n, \vec{x}) = 0$ and $g(m, \vec{x})\downarrow$ $\forall m < n$