Note: - this says fixing operator $\Phi^{1,1}_f$ is unif. closed modulo ψ

- in English: "given a code e for a 2-ary rec. function $\psi_e(x,y)$ and a fixed $x \in \mathbb{N}$, $S(e,x)$ = code for 1-ary rec. function $\psi_{S(e,x)}(y)$ obtained from $\psi_e(x,y)$ by fixing first coord x."
- even briefer: "we can recursively fix the first coordinates of 2-ary recursive functions"

Theorem If ψ is an acceptable enumeration of the recursive functions, then for each $m, n \geq 1$, there is a total recursive function S^m_n (an *S-m-n function*) s.t.

$$\psi_e(x_1, \ldots, x_m, y_1, \ldots, y_n) = \psi_{S^m_n(e,\vec{x})}(\vec{y})$$

("if we can recursively fix the first coordinates of 2-ary functions, then we can recursively fix m coordinates of $(m+n)$-ary functions")

We'll actually show: there is a total recursive $\overline{S}(e,t)$ s.t. $S^m_n(e,\vec{x}) = \overline{S}(e, \langle \vec{x} \rangle)$

Pf: recall: $s*t$ is function s.t. $\langle \vec{x} \rangle * \langle \vec{y} \rangle = \langle \vec{x}, \vec{y} \rangle$; is p.r.

- Let S be total rec. funct. witnessing acceptability: so $\varphi(S(e,x),y) = \varphi(e,x,y)$ $(:= \varphi(e, \langle x,y \rangle))$
- Consider function

$$d(p,z) = \varphi((p)_0, (p), *z)$$
$$(:= \varphi_{(p)_0}((p), *z))$$

(think: d decodes p as $\langle e, \langle \vec{x} \rangle \rangle$, and when $z = \langle \vec{y} \rangle$, computes $\varphi_e(\langle \vec{x} \rangle * \langle \vec{y} \rangle)$)

- observe d is recursive, let e_0 be code for d: $d = \varphi_{e_0}$
- so we have: $\varphi_{e_0}(\langle e, \langle \vec{x} \rangle \rangle, \langle \vec{y} \rangle) = \varphi_e(\langle \vec{x} \rangle * \langle \vec{y} \rangle)$

- define $\overline{S}(e,t) = S(e_0, \langle e, t \rangle)$
- let's check this works:

$$\varphi(\overline{S}(e, \langle \vec{x} \rangle), \langle \vec{y} \rangle) := \varphi(S(e_0, \langle e, \langle \vec{x} \rangle \rangle), \langle \vec{y} \rangle)$$
$$= \varphi(e_0, \langle e, \langle \vec{x} \rangle \rangle, \langle \vec{y} \rangle) \quad (S \text{ witnessing acceptable})$$
$$= \varphi(e, \langle \vec{x} \rangle * \langle \vec{y} \rangle)$$
$$= \varphi_e(\langle \vec{x}, \vec{y} \rangle)$$
$$= \varphi_{S^m_n(e,\vec{x})}(\vec{y}) \quad \checkmark$$

Theorem (the s-m-n theorem)
There is an acceptable enumeration, in fact, our canonical enumeration $\varphi(e,x) = \varphi_e(x) = U(\mu c T(e,x,c))$ is acceptable.

Pf. (sketch): define $S(e,x)$ as follows:

- if e is not code of TM on $\{1,,\}$, $S(e,x) = e$
- if e is such a code, say $e = \langle M \rangle$, define $S(e,x) = \langle M_{e,x} \rangle$ = code for a TM $M_{e,x}$ defined as follows:
  - ↪ on input y, $M_{e,x}$ prints $\langle x,y \rangle$ on its tape and then operates like M.

(an operation a TM can handle)

- this works: when e codes a (coded 2-ary) machine M_e

$$\begin{aligned}\varphi(S(e,x),y) \; (= \varphi_{S(e,x)}(y)) &= \varphi_{\langle M_{e,x} \rangle}(y) \\ &= \text{output of } M_{e,x} \text{ on input } y \\ &= \text{output of } M_e \text{ on input } \langle x,y \rangle \\ &= \varphi_e(\langle x,y \rangle) \\ &= \varphi_e(x,y) = \varphi(e,x,y) \checkmark\end{aligned}$$

Comparing effective enumerations

Q: Is every effective enumeration ψ_e of the recursive functions acceptable?
A: Turns out no

↪ to study Q's like this, we define a notion of comparison between effective enumerations.

Def'n Sps φ and ψ are effective enumerations of the recursive functions. We write $\psi \leq \varphi$ if there is a total rec. func. $t: \mathbb{N} \to \mathbb{N}$ s.t. $\psi(e) = \overline{\varphi}_{t(e)}$ $\forall e \in \mathbb{N}$.

Idea is: if $\psi \leq \varphi$, then once we know φ (and t), we know ψ: given a # ψ code e, can determine function it codes namely $\varphi_{t(e)}$.

Def'n Write $\psi \approx \varphi$ if $\psi \leq \varphi$ and $\varphi \leq \psi$.

Following prop'n says acceptable enum's are top of the food chain among all effective enum's.

Prop'n If ψ is an effective enum. and φ is an acceptable effective enum., then $\psi \leq \varphi$

In particular, if both ψ, φ acceptable then $\psi \approx \varphi$

Pf: - Sps φ is effective (i.e. recursive) + acceptable.
- Consider function $f(k) = \psi((k)_0, (k)_1)$
- point is: $f(\langle e,x\rangle) = \psi(e,x) = \psi_e(x)$
- f is recursive since ψ is
- Let e_0 be φ-code for f
- i.e. $\varphi_{e_0} = f$ so that $\varphi_{e_0}(\langle e,x\rangle) = \psi_e(x)$; i.e. $\varphi(e_0, e, x) = \psi_e(x)$
- by acceptability we have a recursive S s.t. $\varphi(e_0, e, x) = \varphi(S(e_0,e), x)$
- i.e. $\psi_e(x) = \varphi_{S(e_0,e)}(x)$
- so define $t: \mathbb{N} \to \mathbb{N}$ by $t(e) = S(e_0, e)$ ✓

- turns out: if $\varphi \approx \psi$ both acceptable, can even find a recursive bijection $t: \mathbb{N} \to \mathbb{N}$ reducing one to other (i.e. φ, ψ are "isomorphic enumerations")
- called Roger's isomorphism theorem (we might prove later, since permuting)

—

Effective operators

Def'n An operator $\Phi(\vec{x}, \vec{f})$ is called an effective operator if for some (equiv: any – see below) acceptable enum φ, there is a total recursive F s.t.

$$\Phi(\vec{x}, \varphi_{e_1}, \ldots, \varphi_{e_m}) = \varphi_{F(\vec{x}, e_1, \ldots, e_m)}$$

(same as def'n of "uniformly closed" but w/ added hypothesis that φ is acceptable)

Observe if ψ another acceptable enum then Φ also effective wrt ψ, since we can convert.

$$\begin{aligned}\Phi(\vec{x}, \psi_{e_1}, \ldots, \psi_{e_m}) &= \Phi(\vec{x}, \varphi_{t(e_1)}, \ldots, \varphi_{t(e_m)}) \\ &= \varphi_{F(t(e_1), \ldots, t(e_m))} \\ &\quad \uparrow \vec{x} \\ &= \psi_{s(F(\vec{x}, t(e_1), \ldots, t(e_m)))}\end{aligned}$$

where t, s functions we get from $\psi \leq \varphi$ and $\varphi \leq \psi$.

↳ so this turns out to be right notion of "recursive operator" since it doesn't depend on which (acceptable) enum ya use to encode the recursive functions.

- Given an operator $\Phi(\vec{x}, \vec{F})$, can define associated functional Φ^*
- Φ^* defined by
  if $\Phi(\vec{x}, \vec{F}) = h$ $\quad (h: N^n \to N)$
  then $\Phi^*(\vec{x}, \vec{F}, \vec{z}) = h(\vec{z})$

- using an enumeration φ_e of the rec. functions can associate to a functional Φ^* a function F_Φ defined by:

$$F_\Phi(\vec{x}, \vec{e}, \vec{z}) = \Phi^*(\vec{x}, \varphi_{e_1}, \ldots, \varphi_{e_m}, \vec{z})$$
$$\vec{e} = e_1, \ldots, e_m = \Phi(\vec{x}, \varphi_{e_1}, \ldots, \varphi_{e_m})(\vec{z})$$

Claim Φ is an effective operator if and only if F_Φ is recursive

Pf. ($\Rightarrow$) Sps Φ is effective and φ is an acceptable enumeration (F_I defined wrt φ)
- then $F_\Phi(\vec{x}, \vec{e}, \vec{z}) = \Phi(\vec{x}, \varphi_{e_1}, \ldots, \varphi_{e_m})(\vec{z})$
  defn of Φ $= \varphi_{F(\vec{x}, e_1, \ldots, e_m)}(\vec{z})$
  effective $= \varphi(F(\vec{x}, e_1, \ldots, e_m), \vec{z})$
  recursive ✓

($\Leftarrow$) Now sps F_Φ is recursive
- $\Phi(\vec{x}, \varphi_{e_1}, \ldots, \varphi_{e_m})(\vec{z}) = F_\Phi(\vec{x}, \vec{e}, \vec{z})$
  F_Φ recursive, so let e_0 be code
  $= \varphi(e_0, \vec{x}, \vec{e}, \vec{z})$
  s-m-n func $= \varphi(s(e_0, \vec{x}, \vec{e}), \vec{z})$
  $= \varphi_{s(e_0, \vec{x}, \vec{e})}(\vec{z})$

- so define $F(\vec{x}, \vec{e}) = S(e_0, \vec{x}, \vec{e})$
- then F witnesses Φ is effective ✓

Ex Sps Φ is primitive recursive operator, we show Φ is effective.

- Φ defined by $\Phi(g_1, g_2) = h$
  where $h(0, \vec{x}) = g_1(\vec{x})$
  $h(y+1, \vec{x}) = g_2(h(y, \vec{x}), y, \vec{x})$
- so F_Φ defined by
  $F_\Phi(e_1, e_2, 0, \vec{x}) = \varphi_{e_1}(\vec{x})$
  $F_\Phi(e_1, e_2, y+1, \vec{x}) = \varphi_{e_2}(F_\Phi(e_1, e_2, y, \vec{x}), y, \vec{x})$

- notice this actually a def'n of F_Φ by prim recursion, hence F_Φ is recursive (and p.r. if g_1, g_2 are)
- hence Φ effective ✓

⇒ thm above shows why this def'n of "effective operator" is correct one: Φ effective iff its coded version F_Φ is recursive.

Rice's theorem

- Let φ_e be an acceptable enum of the rec. functions (can always use the std one: $\varphi(e,x) = U(\mu z T(e,x,z))$)