Comparing the recursion thms

- Sps Φ is effective operator, so $\Phi(\varphi_n) = \varphi_{F(n)}$ for some total recursive F
- By 2nd rec. thm $\exists n$ s.t. $\varphi_{F(n)} = \varphi_n$
- Thus $\Phi(\varphi_n) = \varphi_n$ for this n

- So: 2nd rec. theorem yields a fixed point for Φ, but not necessarily least
- 1st rec. theorem gives us least fixed point ... by constructing it!
- So 2nd rec. theorem $\Rightarrow$ weak version of the first.

- In other direction, sps h is a total rec. function s.t. if $\varphi_n = \varphi_m$ then $\varphi_{h(n)} = \varphi_{h(m)}$

  (i.e. h only depends "semantically" on the code n, i.e. on function coded by underlying TM)

  (e.g. if $h = F$ is coded version of an effective operator Φ, then h has this property, since if $\varphi_n = \varphi_m$, then $\varphi_{h(n)} = \Phi(\varphi_n)$

  $= \Phi(\varphi_m) = \varphi_{h(m)}$

- Can define an operator by
$\Phi(\varphi_n) = \varphi_{h(n)}$ well-defined by assumption on h.

- then Φ is effective on rec. functions (h witnesses effectivity)
so can be extended to globally cts. operator by Myhill-Shepherdson.
- by 1st rec. thm Φ has a fixed point, i.e. $\exists n$ s.t. $\Phi(\varphi_n) = \varphi_n$
i.e. $\varphi_{h(n)} = \varphi_n$
- so 1st rec. theorem implies weak version of second (only for "semantic" h's)

Solving functional equations

Q: what can we do w/ first rec. theorem?
A: find recursive sol'ns to "self-referential" functional eq'ns
↳ can think of a fixed point of Φ as sol'n to the eq'n
$$F(\vec{x}) = \Phi(F(\vec{x}))$$
↳ many familiar functions defined "recursively" by such eq'ns
↳ 2nd rec. theorem could sometimes show a sol'n exists
↳ 1st constructs a sol'n (and it's minimal)

Ex's we first take a look @ some simple operators + their least fixed points.

1. Sps $\Phi: \Omega_1 \to \Omega_1$ is identity
i.e. $\Phi(f) = f$ for all unary f.
↳ is continuous: if $\Phi(f)(x)\downarrow$, depends only on finitely many values of f (actually only one – $f(x)$ – since $\Phi(f)(x) = f(x)$)
↳ i.e. $\Phi(f)(x) = \Phi(\pi)(x)$ for any $\pi \subseteq f$ w/ $x \in dom(\pi)$ (when $\Phi(f)(x)\downarrow$)
↳ effective too: coded operator is $F(e) = e$, which is recursive.

- observe: every f a fixed pt. (by defn...)
- so ofc $f = \emptyset$ is least fixed pt. (and is recursive)
- we also get this from proof of 1st rec. thm since
$$\emptyset = \Phi(\emptyset) = \Phi(\Phi(\emptyset)) = \cdots$$
i.e. $f_0 = f_1 = f_2 = \cdots = \bigcup f_n = \emptyset = f_\Phi$
least fixed.

so iteration in this case is trivial
just looks like $\emptyset \to \emptyset \to \emptyset \to \cdots \to \emptyset$

(2) - To get an interesting iteration, need $\Phi(f)$ to depend on values of f in more interesting way.
- Let's start w/ a functional eq'n we want to solve, e.g.

$$F(x,y) = \begin{cases} y & \text{if } x = 0 \\ F(x-1,y)+1 & \text{if } x > 0 \end{cases}$$

- Can think of such an F as being a fixed point of the operator $\Phi : \Omega_2 \to \Omega_2$ defined by:

$$\Phi(f)(x,y) = \begin{cases} y & \text{if } x = 0 \\ f(x-1,y)+1 & \text{if } x > 0 \end{cases}$$

(i.e. $\Phi(f) = g$ where $g(x,y) = \}$)

- Φ is continuous: to compute $\Phi(f)(x,y)$ (when defined) only need to know (at most) $f(x-1,y)$

i.e. $\Phi(f)(x,y) = \Phi(\pi)(x,y)$ for any $\pi \subseteq f$ w/ $(x-1,y) \in \mathrm{dom}(\pi)$

when defined

- "clearly" effective too (ahem...)

↳ could write a TM that on input e a code for f, returns $F(e)$ a code for $\Phi(f)$

- a fixed point $\Phi(F) = F$ solves eq'n above
- there is (turns out) a unique fixed point in this case: $F(x,y) = x+y$
- we construct it from the iteration:

$f_0 = \phi$, i.e. $f_0(x,y) = \uparrow \quad \forall x,y$

$f_1 = \Phi(f_0)$

$$f_1(x,y) = \Phi(f_0)(x,y) = \begin{cases} y & x=0 \\ \uparrow & o/w \end{cases}$$

$f_2 = \Phi(f_1)$

$$f_2(x,y) = \Phi(f_1)(x,y) = \begin{cases} y & x=0 \\ f_1(0,y)+1 = y+1 & x=1 \\ \uparrow & x>1 \end{cases}$$

$f_3 = \Phi(f_2)$

$$f_3(x,y) = \Phi(f_2)(x,y) = f_2(x,y) \quad x=0,1$$

(monotonicity)

$$= f_2(1,y)+1 \quad x=2$$
$$= y+1+1$$
$$= y+2$$
$$\uparrow \quad \text{o.w.}$$

etc.

then $\phi \subseteq f_1 \subseteq f_2 \subseteq \dots \to F = \bigcup f_n$ is least fixed pt

ez to see $F(x,y) = x+y$ since total, unique fixed pt.

3. Now consider eq'n:

$$F(x) = \begin{cases} 1 & x = 0 \\ x\,F(x-1) & x > 0 \end{cases}$$

- this clearly defines $F(x) = x!$, and is unique sol'n.
- but if we couldn't see this, instead could find the fixed point of the (continuous - why?) (effective - why?) operator Φ defined by:

$$\Phi(f)(x) = \begin{cases} 1 & x = 0 \\ x\,f(x-1) & x > 0 \end{cases}$$

- iteration goes like:

$f_0 = \emptyset \qquad f_0(x) = \uparrow \quad \forall x$

$f_1 = \Phi(f_0)\,; \quad f_1(x) = \Phi(f_0)(x) = \begin{cases} 1 & x = 0 \\ \uparrow & o/w \end{cases}$

$f_2 = \Phi(f_1)\,; \quad f_2(x) = \Phi(f_1)(x) = \begin{cases} 1 & x = 0 \\ = f_1(0) \cdot 1 \overset{"}{=} 1 & x = 1 \\ = \uparrow & o/w \end{cases}$

$f_3 = \Phi(f_2)\,; \quad f_3(x) = \Phi(f_2)(x) = \begin{cases} 1 & x = 0 \\ 1 & x = 1 \\ f_2(1) \cdot 2 \overset{"}{=} 2 & x = 2 \\ \uparrow & o/w \end{cases}$

: etc.

$\emptyset \to f_1 \to f_2 \to \quad - \quad \to \cup f_n = F$
= factorial

4. Ackermann function $A(n,x)$ is (unique) fixed point of the operator $\Phi : \Omega_2 \to \Omega_2$ defined by

$$\Phi(f)(n,x) = \begin{cases} x+1 & n=0 \\ f(n-1,1) & n>0,\ x=0 \\ f(n-1, f(n,x-1)) & x,n>0 \end{cases}$$

... can construct A by iterating Φ get

$$A(n,x) = \begin{cases} x+1 & n=0 \\ A(n-1,1) & n>0,\ x=0 \\ A(n-1, A(n,x-1)) & x,n>0. \end{cases}$$

5. $F(x) = 1 \quad x=0,1$
$\quad = F(x-1)+F(x-2) \quad x>1$

defines Fibonacci
unique f.p. of $\Phi : \Omega_1 \to \Omega_1$

$$\Phi(f)(x) = \begin{cases} 1 & x=0,1 \\ f(x-1)+f(x-2) & x>1 \end{cases}$$

6. (an idempotent recursive fixed pt)
- Sps $h(x)$ a total recursive function
- $P(x)$ recursive relation

Consider eq'n: $F(x) = \begin{cases} x & \text{if } P(x) \\ F(F(h(x))) & \text{if } \neg P(x) \end{cases}$

- Fixed pt of the operator Φ:

$$\Phi(f)(x) = \begin{cases} x & \text{if } P(x) \\ f(f(h(x))) & \text{if } \neg P(x) \end{cases}$$

- instead of constructing F w/ an iteration, let's think abt how it must look.

- To compute $F(x)$

check: is $P(x)$? if yes: $F(x) = x$

if not $F(x) = F(F(h(x))) = F^2(h(x)) = $ what?

check: is $P(h(x))$? if yes $F(F(h(x))) = F(h(x)) = h(x)$

if not $F(h(x)) = F(F(h(h(x)))) = F^2(h^2(x))$

so $F(x) = F(F(h(x))) = F^3(h^2(x)) = ?$

check: is $P(h^2(x))$? if yes, $F(x) = F^3(h^2(x)) = h^2(x)$

if not: $F(h^2(x)) = F^2(h^3(x))$

so $F(x) = F^3(h^2(x)) = F^4(h^3(x))$

$\vdots$ etc.

Continuing: get:

$$F(x) = \begin{cases} x & \text{if } P(x) \\ h(x) & \text{if } \neg P(x) \wedge P(h(x)) \\ h^2(x) & \text{if } \neg P(x) \wedge \neg P(h(x)) \wedge P(h^2(x)) \\ h^3(x) & \text{if } \ldots \\ \vdots & \end{cases}$$

Claim $F^2(x) = F(x)$ (i.e. F is idempotent)

Pf. case out.

① if $P(x)$, $F^2(x) = x = F(x)$

② if $\neg P(x) \wedge P(h(x))$

$$F^2(x) = F(F(x)) = F(F^2(h(x)))$$
$$= h(x) = F^2(h(x)) = F(x)$$

③ if $\neg P(x) \wedge \neg P(h(x)) \wedge P(h^2(x))$

$$F^2(x) = F(F(x))$$
$$= F(h^2(x)) = h^2(x)$$
$$= F(x)$$

etc.. if undefined ✓ too.

⑦ Consider F s.t.

$$F(x) = \begin{cases} x/2 & x \text{ even} \\ F(F(3x+1)) & x \text{ odd} \end{cases}$$

$F(0) = 0$

$F(1) = F(F(4)) = F(2) = 1$

$F(2) = 1$

$F(3) = F(F(9+1)) = F(5) = F(F(15+1))$
$= F(8) = 4$

$F(4) = 2$

$F(5) = 4$

$F(6) = 3$

⋮

Exercise: prove F is total.

8. Sps we want F s.t.

$$F(x) = \begin{cases} 0 & x = 0, 1 \\ F(x/2) & x \text{ even} > 0 \\ F(3x+1) & x \text{ odd} > 1 \end{cases}$$

$F(0) = F(1) = 0$

$F(2) = F(1) = 0$

$F(3) = F(10) = F(5)$

$= F(16) = F(8) = F(4) = F(2) = F(1) = 1$

Is F total? Famous open problem...