Recall recursion:

Syntax:

$e ::= \cdots \mid \text{fix } f(x).e$
$v ::= \cdots \mid \text{fix } f(x).e$

$\lambda x.e \approx \text{fix } f(x).e \quad \text{where } f \notin fv(e).$

Statics:

$\Delta; \Gamma, f : \sigma \longrightarrow \tau, x{:}\sigma \vdash e : \tau$
—
$\Delta; \Gamma \vdash \text{fix } f(x).e : \sigma \longrightarrow \tau$

Dynamics:

$(\text{fix } f(x).e)\ v \longmapsto e[\text{fix } f(x).e/f][v/x]$

Alternative Dynamics:

$\text{fix } x.e \longmapsto e[\text{fix } x.e/x]$

— We cannot prove compatibility for fix

Here's the compatibility lemma for fix:

$\Delta; \Gamma, f : \sigma \longrightarrow \tau, x{:}\sigma \vdash e_1 \approx e_2 : \tau$
—
$\Delta; \Gamma \vdash \text{fix } f(x)e_1 \approx \text{fix } f(x).e_2 : \sigma \longrightarrow \tau$

The straight-forward proof breaks down.

Let $\rho \in D[\Delta]$, $(\gamma_1,\gamma_2) \in G[\Gamma]\rho$, $\delta_1,\delta_2 \in \Delta \longrightarrow \text{CType}$.
TS: $(\delta_1\gamma_1(\text{fix } f(x)e_1), \delta_2\gamma_2(\text{fix } f(x)e_2)) \in E[\sigma{\longrightarrow}\tau]\rho$.
$\Leftarrow$ $(\delta_1\gamma_1(\text{fix } f(x)e_1), \delta_2\gamma_2(\text{fix } f(x)e_2)) \in V[\sigma{\longrightarrow}\tau]\rho$.

Let $(v_1,v_2) \in V[\sigma]\rho$.
TS: $(\delta_1\gamma_1(\text{fix } f(x)e_1)\ v_1, \delta_2\gamma_2(\text{fix } f(x)e_2)\ v_2) \in V[\tau]\rho$.

By closure under expansion,
STS: $(\delta_1\gamma_1 e_1[F_1/f][v_1/x]), \delta_2\gamma_2 e_2[F_2/f][v_2/x]) \in E[\tau]\rho$
where
$F_i := \delta_i\gamma_i\text{fix } f(x).e_i$.

To make progress, we must show

$(F_1,F_2) \in E[\sigma\rightarrow\tau]\rho$.

But that's exactly our goal.

— Admissibility in domain theory

Pitts uses something called ⊤⊤-closure. We won't start with that general concept. We'll start with the property we need to make the previous proof go through.

We'll refer to domain theory. We need the concept of /Admissibility/ to make such recursive proofs go through.

We could encode admissibility directly as a property of syntax. (We'll state a property of syntax that gives us what we need in this case.) That turns out not to be the cleanest, most direct way for our purpose. Instead, we'll use Pitts' more general notion of ⊤⊤-closure.

Question: Does admissibility scale, like ⊤⊤-closure, to more interesting models.

Answer (Derek/Neel): Syntactically and semantically, admissibility is uglier. Admissibility doesn't work very well semantically. Problems come up with admissibility when you try to express properties about disjunctions and existentials. ⊤⊤-closure doesn't so suck.

In the following, we refer to Pitts' lecture notes (see the paramore web page).

Scott's Fixed Point Induction Principle [Slide 40, p46]

Let $f : D \rightarrow D$ be a continuous function on a domain D. For any /admissible/ subset (aka property) $S \subseteq D$, to prove that the least fixed point of f is in S:

$$\begin{aligned} & fix(f) \in S \\ \Leftarrow \quad & \forall d{\in}D\ (d{\in}S \Rightarrow f(d) \in S). \end{aligned}$$

The (admissible) property S we care about is "things are in the logical relation".

Here's the analogy:

Think of d as the variable f in our failed proof.
Think of the function f as λf.λx.e in our failed proof.
Think of S as the LR.

We'll return to this slide and prove the principle once we've defined all the relevant terms.

[Slide 15, p14]:
A /partially ordered set/ (D,⊑) is reflexive, transitive, and antisymmetric.

[Slide 22, p21]:
A /chain complete poset/, or cpo for short, is a poset (D,⊑) in which all countable increasing chains
$d_0 \sqsubseteq d_1 \sqsubseteq d_1 \cdots$
have least upper bounds ⊔(n≥0) d_n.

Idea: Think of the chain d_i as an ordering on definedness.
d_0 gives you no answers
d_1 gives you some answers before diverging.
etc.
d_n gives you answers for n unrollings, then diverges.
The fixed point never diverges. It will always give you the answer you need.

You want least upper bounds so you can talk about limits of these chains.

[Slide 22, p21]:
A /domain/ is a cpo that possesses a least element ⊥.

Idea: Think of ⊥ as the function that diverges immediately.

Continuity and strictness [Slide 18, p15; Slide 28, p27]:
If D and E are cpo's, the function f : D ⟶ E is /continuous/ iff it is monotone and preserves least upper bounds of chains:

- ∀d,d' ∈ D. d ⊑ d' ⇒ f(d) ⊑ f(d').
- For all chains $d_0 \sqsubseteq d_1 \sqsubseteq \cdots$ in D, we have
  f(⊔(n≥0)d_n) = ⊔(n≥0)f(d_n) in E.

Idea: Somehow the behavior of the fixed point is approximated correctly by its finite approximations.

Kleene's fixed point theorem (not Tarski's) [Slide 29, p29]:
Let f : D ⟶ D be a continuous function on a domain D. Then f possesses a least fixed point given by

$\text{fix}(f) = \sqcup(n \geq 0)\ f^n(\bot)$.

Idea: Think of D as the partial function space σ⟶τ.
This theorem says the fixed point is the limit of this chain
of finite approximations.

Chain-closed and admissible subsets [Slide 39, p45]:

Let D be a cpo. A subset S ⊆ D is called /chain-closed/ iff
for all chains $d_0 \sqsubseteq d_1 \sqsubseteq \cdots$ in D

$(\forall n \geq 0.\ d_n \in S) \Rightarrow (\sqcup(n \geq 0) d_n) \in S$.

If D is a domain, S ⊆ D is called /admissible/ iff it is
a chain-closed subset of D and ⊥ ∈ S.

Idea: For us, S will be the logically related elements.
We'll say the subset is admissible if it contains ⊥ and it's
chain-closed.
If the property of interest (ie, logical approximation) holds for all
finite approximations, then it holds at the limit.

We have two things to show:

Our LR is admissible (so it has this property).
Use admissibility to prove an analog of the Scott induction.

Now that we've defined all the terms, we can return to Scott's Fixed
Point Induction Principle [Slide 40, p.46]

5-Slides 7–10 Proof of the Scott Induction Principle.

[Derek flipped through some other slides by Pitts offering a
proof of the princple.]

— Admissibility in our setting

Back to our setting: How do we set things up so we get exactly the
same kind of argument Pitts used in Slides 7–10 to prove the Scott
Induction Principle?

(We'll end up using Scott induction as a reasoning principle. We are
NOT using the domain theory directly, we're being inspired by it.)

We want to define a notion of admissibility in our operational
setting. To do that, we have to define how to translate this idea of ⊥
and all the elements of the chain f^n.

Definition:

For i∈1..2, we'll define F^n_i and F_i.

F^0_i = fix f(x).f(x)   (= "⊥")
F^{n+1}_i = λx.hat{e_i}[F^n_i/f]

F_i = fix f(x).hat(e_i).

Where hat{e_i} will either be our original e_i or just closed e_i (we want to deal with the δ_i's and γ_i's but it's not obvious at a glance how).

The idea is that F^n_i, when applied, unrolls F_i at most n times, then diverges.

Idea: Our [σ⇀τ] = "the domain D".

The property we will seek to prove is:

Admissibility:

If $\forall n.(t_1[F_1{}^\wedge n/f], t_2[F_2{}^\wedge n/f]) \in E[\tau]\rho$,
then $(t_1[F_1/f], t_2[F_2/f]) \in E[\tau]\rho$.

We'll have to adjust our model to make the proof of admissibility go through.

But suppose we had admissibility. Then we can make progress in our stalled compatibility proof.

For now, we can see how our failed compatibility proof might go through. Think of it as the Scott Induction case, since that's what we'll follow. What we want to do is instantiate the admissibility theorem where t_1, t_2 are both just the variable f.

Lemma (Compatibility for fix, assuming admissibility holds)

$\Delta; \Gamma, f : \sigma \longrightarrow \tau, x{:}\sigma \vdash e_1 \approx e_2 : \tau$
—
$\Delta; \Gamma \vdash \text{fix } f(x)e_1 \approx \text{fix } f(x).e_2 : \sigma \longrightarrow \tau$

Note that since we're presenting these ideas incrementally, we'll have to come back and fix this proof when we change the model. (The new model will have a different definition of the term relation.)

handwave: Proof:

Let $\rho \in D[\Delta]$, $(\gamma_1,\gamma_2) \in G[\Gamma]\rho$, $\delta_1,\delta_2 \in \Delta \longrightarrow$ CType.

TS: (fix $f(x).\delta_1\gamma_1 e_1$, fix $f(x).\delta_2\gamma_2 e_2) \in E[\sigma\longrightarrow\tau]\rho$.

Instantiating admissibility with t_1=f=t_2

and hat{e_i} = δ_iγ_ie_i,

STS: $\forall n.(F_1$^n, $F_2^n) \in E[\sigma\longrightarrow\tau]\rho$.

(Aside: We've reduced the problem to reasoning only about finite approximations of the chain.)

Proof by induction on n.

Case n = 0:

STS: (fix f(x).f(x), fix f(x).f(x)) $\in E[\sigma\longrightarrow\tau]\rho$

$\Leftarrow$ (fix f(x).f(x), fix f(x).f(x)) $\in V[\sigma\longrightarrow\tau]\rho$

$\Leftarrow$ $\forall(v_1,v_2) \in V[\sigma]\rho$.

$(F_1^0\ v_1, F_2^0\ v_2) \in E[\tau]\rho$.

Idea: Both $F_1^0\ v_1\uparrow$ and $F_2^0\ v_2\uparrow$.

So our new E[] relation must relate diverging terms.

Case n+1:

TS: (F^{n+1}$_1$, F^{n+1}$_2$) $\in E[\sigma\longrightarrow\tau]\rho$

$\Leftarrow$ (F^{n+1}$_1$, F^{n+1}$_2$) $\in V[\sigma\longrightarrow\tau]\rho$

$\Leftarrow$ $\forall(v_1,v_2) \in V[\sigma]\rho$.

(F_1^{n+1} v_1, F_2{n+1} v_2) $\in E[\tau]\rho$

$\Leftarrow$ (Closure under expansion.)

(hat{e_1}[F_1^n/f][v_1/x], hat{e_2}[F_2^n][v_2/x]) $\in E[\tau]\rho$.

Observe: By IH (F_1^n, F_2^n) $\in E[\sigma\longrightarrow\tau]\rho$.

Define γ'_i := γ_i[x↦v_i][f↦F_i^n].

Then $(\gamma'_1,\gamma'_2) \in G[\Gamma,f{:}\sigma\longrightarrow\tau,x{:}\sigma]\rho$.

By our premise $(\Delta; \Gamma, f : \sigma \longrightarrow \tau, x{:}\sigma \vdash e_1 \approx e_2 : \tau)$,

we have

$(\delta_1\gamma'_1 e_1,\delta_2\gamma'_2 e_2) \in E[\tau]\rho$

$\Rightarrow$ $((\delta_1\gamma_1 e_1)[v_1/x][F_1$^n/f], $(\delta_2\gamma_2 e_2)[v_2/x][F_2$^n/f]) $\in E[\tau]\rho$.

handwave: Q.E.D.

You get the picture.

— Modifying the logical relation

We won't have time today to both give the new model and complete these proofs. We'll write down the new term relation, discuss intuitions, and sketch future work.

To make this proof work, we can consider some alternatives.

First, an alternative approach.

One blindingly obvious (but klutzy) way to define E[τ]ρ such that it relates divergent computations and is admissible: Just extend E[τ]ρ to relate divergent computations.

$$
\begin{aligned}
E[\tau]\rho = \{ (e_1,e_2) \mid e_1\uparrow \wedge e_2\uparrow \} \cup \\
\{ (e_1,e_2) \mid \exists v_1,v_2.\ e_1 \downarrow v_1 \wedge e_2 \downarrow v_2 \wedge (v_1,v_2) \in V[\tau]\rho \}
\end{aligned}
$$

This leads to a difficulty with admissibility:

> When we go to prove admissibility, a property of the logical relation, we want to use induction on τ.
>
> When we get to the base case V[α]ρ, we're trying to show the admissibility property holds for an arbitrary R ∈ Cand.
>
> So we have to bake admissibility into our notion of candidate relation.

(Aside: Derek thinks we can simplify the notion of admissibility used in the preceding hand-waving proof. We only use it in a degenerate way. Derek will discuss next time.)

Derek and Georg worked this out a few years ago and Derek concluded it was rather painful (but doable). There are a couple reasons. First, it's painful to prove admissibility at arbitrary τ. Second, when you want to use the LR and you pick some relation R to represent some type α, you have to show R admissible. (Even the statement of (our syntactic variant of) admissibility is nasty. It's not always obvious how to show it.)

Instead, we'll use biorthogonality. It's completely different. It's a really cool idea, but very non-obvious.

— ⊤⊤-closure aka ⊥⊥-closure aka biorthogonality

Introduced in an accessible fashion in (Pitts and Stark, 1998). (We'll get to this paper.)

It handles a number of things. The original paper lacks a very clear explanation of the power of the technique.

Here's the definition (ideas will follow):

> Terms are related if they behave (well and) in related ways under related evaluation contexts.
>
> $E[\tau]\rho = \{ (e_1,e_2) \mid \forall(K_1,K_2)\in K[\tau]\rho.\ K_1[e_1]\downarrow \Leftrightarrow K_2[e_2]\downarrow \}$
>
> For all observations of interest, termination suffices. We can always fiddle continuations to get programs, say, evaluating to the same values.
>
> $K[\tau]\rho = \{ (K_1,K_2) \mid \forall(v_1,v_2)\in V[\tau]\rho.\ K_1[v_1]\downarrow \Leftrightarrow K_2[v_2]\downarrow \}$.

Next time we'll discuss ideas and begin to work out the new metatheory. (We can stick with our old Cand.)