Last time, we gave a model for first-order state based on Pitts and Stark.

— Metatheory

Let's work through the proof of the bind lemma. If we can prove it, then the proofs of the compatiblity lemmas from before will almost always “carry over” with little work. We'll then turn to adequacy, the example we looked at last week, and an example that proved “awkward” for such models.

Lemma (Closed bind lemma):
If $(W,e_1,e_2) \in E[\tau]\rho$,
and $(\forall v_1,v_2.\ \forall W' \sqsupseteq W.\ (W',v_1,v_2) \in V[\tau]\rho \Longrightarrow (W,K_1[v_1],K_2[v_2]) \in E[\tau']\rho')$
then $((W,K_1[e_1],K_2[e_2]) \in E[\tau']\rho'$.
Proof:
Suppose $(W,K'_1,K'_2) \in K[\tau']\rho'$.
TS: $(W,K'_1[K_1[e_1]], K'_2[K_2[e_2]]) \in O$.

By the first assumption,
STS: $(W',K'_1[K_1],K'_1[K_2]) \in K[\tau]\rho$

(Note: $K'_1[K_1]$ composes the two contexts, preserving K_1's hole.)

Suppose $W' \sqsupseteq W.\ (W',v_1,v_2) \in V[\tau]\rho$.
(1) STS: $(W',K'_1[K_1[v_1]],K'_2[K_2[v_2]]) \in O$.

By the second hypothesis,
WK: $(W',K_1[v_2],K_2[v_2]) \in E[\tau']\rho'$.

To conclude, we need to know that K'_1, K'_2 are related at W' by the definition of $E[\tau']\rho'$. We know they are related at W. By montonicity of the $K[-]$ relation,
WK: $(W,K'_1,K'_2) \in K[\tau']\rho'$
$\Longrightarrow$ (1).
Q.E.D.

We never opened the definition of O. That's reasonable. The proof shouldn't have to talk about heaps in any interesting way. Semantics should be compositional. The sequencing of different effectful operations does not itself care about the heaps. We've isolated operations that effect the heaps in this O thing. (We've glossed over compatibility cases involving pure operations. In the pure operations,

you do have to expand the O because you have something to show in those cases. But even then, you expand the O only in a boring way. You want to know, for example, that if two terms are related by O and they take some steps of expansion, then the terms are still related by O.
To show that, you quantify over heaps that are related by the world, and show after expansion that the so-obtained heaps satisfy the same world.)

Lemma (Compatibility for allocation):
    $\Delta; \Gamma \vdash e_1 \approx e_2 : \text{int}$
    —
    $\Delta; \Gamma \vdash \text{ref } e_1 \approx \text{ref } e_2 : \text{ref}$
Proof:
    By the bind lemma for open terms, this reduces to the following.
Q.E.D.

Lemma:
    If $(W,v_1,v_2) \in V[\text{int}]$,
    then $(W,\text{ref } v_1,\text{ref } v_2) \in E[\text{ref}]$.
Proof:
    Suppose $(W,K_1,K_2) \in K[\text{ref}]$.
    TS: $(W,K_1[\text{ref } v_1],K_2[\text{ref } v_2]) \in O$.

    Suppose $(h_1,h_2) : W$.
    TS: $h_1;K_1[\text{ref } v_1] \downarrow\downarrow W.j\ h_2;K_2[\text{ref } v_2]$.

    Let $\text{ell}_1 \notin \text{dom}(h_1)$ and $\text{ell}_2 \notin \text{dom}(h_2)$.
    STS: $h_1,\text{ell}_1 \mapsto v_1;K_1[\text{ell}_1] \downarrow\downarrow (W.j\text{-}1)\ h_2,\text{ell}_2 \mapsto v_2;K_2[\text{ell}_2]$.
    (Aside, we could prove this for W.j steps.
    In this particular proof, we don't need the stronger result.)

    Define $W' = \triangleright W +\!+ \{ (\text{ell}_1 \mapsto n, \text{ell}_2 \mapsto n) \mid n \in \mathbb{Z} \}$.
    (Read ++ as extended with. We bumped step indices with $\triangleright W$ to be consistent. We could have proved the stronger result.)
    TS: $W' \sqsupseteq W \wedge (h_1,\text{ell}_1 \mapsto v_1, h_2, \text{ell}_2 \mapsto v_2) : W'$
    $\Rightarrow$(Monotonicity)
        $(W',K_1,K_2) \in K[\text{ref}] \wedge (h_1,\text{ell}_1 \mapsto v_1, h_2, \text{ell}_2 \mapsto v_2) : W'$
    $\Rightarrow$ $(W',K_1[\text{ell}_1],K_2[\text{ell}_2]) \in O$.
    STS: $(W',\text{ell}_1,\text{ell}_2) \in V[\text{ref}]$.

    This is true since we added an island of the right form

to W to obtain W'.
Q.E.D.

Remarks:
• General procedure: For some world (with the right step index), show the heaps satisfy the world, show the continuations are related in the world, and show the values are related in the world.

• Regarding our not using the stronger step indices W.j and our using world W vs ▷W in the previous proof: In step-indexing with first-order state you do not have to count all of the steps. Steps do matter for recursion in that setting. With higher-order state that will change.

HW: Prove compatibility for ! and :=.

— Adequacy

Lemma (Adequacy):
If $\vdash e_1 \approx e_2 : \tau$,
then $\emptyset;e_1 \downarrow\downarrow \emptyset;e_2$.

(Aside: We could allow for arbitrary starting heaps. This is simpler.)
Proof:
We know e_1,e_2 are related in any world. We want to pick a world W such that the empty heaps satisfy W. (We'll pick the world with no islands.)

Let $n \in \mathbb{N}$. Set $W_0 := (n,\emptyset)$.

By assumption, $\forall n.\ (W_0,e_1,e_2) \in E[\tau]$.
WK: $\emptyset,\emptyset : W_0$.
TS: $(W_0,\bullet,\bullet) \in K[\tau]$ (very easy)
$\Rightarrow$ $\emptyset;e_1 \downarrow\downarrow_n \emptyset;e_2$.

We've shown co-termination for arbitrary n.
Thus $\emptyset;e_1 \downarrow\downarrow \emptyset;e_2$.
Q.E.D.

— Example

Recall our motivating example:

If

$\tau = (1 \longrightarrow 1) \times (1 \longrightarrow \text{int})$
$e_1 = \text{let x = ref 0 in } (\lambda\_.x := !x + 1, \lambda\_.!x)$
$e_2 = \text{let x = ref 0 in } (\lambda\_.x := !x - 1, \lambda\_.0 - !x)$,
then
$\vdash e_1 \approx e_2 : \tau$.

Proof:
TS: $\forall W.\ (W,e_1,e_2) \in E[\tau]$.
Suppose $(W,K_1,K_2) \in K[\tau]$ and $h_1,h_1 : W$.
TS: $h_1;K_1[e_1] \downarrow\downarrow W.j\ h_2;K_2[e_2]$.

Let's execute e_1 and e_2.
STS: $\forall l_1 \notin dom(h_1),\ l_2 \notin dom(h_2)$.
$h_1,l_1 \mapsto 0;\ K_1[v_1] \downarrow\downarrow W.j\ h_2,l_2 \mapsto 0;\ K_2[v_2]$
where
$v_1 = <\lambda\_.ell_1 := !ell_1 + 1, \lambda\_.!ell_1>$
$v_2 = <\lambda\_.ell_1 := !ell_1 - 1, \lambda\_.0 - !ell_1>$.

Define W' = W++(“a new island with invariant
$\{ (ell_1 \mapsto n, ell_2 \mapsto -n) \mid n \in \mathbb{N} \}$”).
We can show
$(h_1,l_1 \mapsto 0), (h_2,l_2 \mapsto 0) : W'$.
Like before, we left the separation making $h_1,h_2 : W$ alone and just added a new island.
By monotonicity, we know $(W',K_1,K_2) \in K[\tau]$.
STS: $(W',v_1,v_2) \in V[\tau]$.

> Aside: Notice we didn't do anything interesting with K_1, K_2 at all. The reason is in this language, we have a property of the operational semantics, sometimes called a “uniform reduction semantics”. The behavior of terms is independent of the evaluation contexts you put them in. (If $e_1 \mapsto_* e_2$, then $K[e_1] \mapsto_* K[e_2]$.) In proving this thing, we only needed to reason about these local reduction steps. It didn't matter what K_1 and K_2 were at all. Because of that, we can define The Pitts and Stark “principle of local invariants”. That gives us a way of writing this proof without ever mentioning K_1 and K_2. See Pitts and Stark, Proposition 5.1 (page 32). This idea only works with languages with a uniform reduction semantics. It wouldn't work with continuations, for example.

STS: $(W',\lambda\_.ell_1 := !ell_1 + 1,\lambda\_.ell_1 := !ell_1 - 1) \in V[1 \longrightarrow 1] \wedge$

$(W',\lambda\_.!ell_1,\lambda\_.0 - !ell_1) \in V[1 \to int]$.

For the first conjunct, let $W'' \sqsupseteq W'$. (We'll reuse some variables.)
TS: $(W'',ell_1 := !ell_1 + 1,ell_1 := !ell_1 - 1) \in E[1]$.
Suppose $(W'',K_1,K_2) \in K[1]$ and $h_1,h_2 : W''$.
TS: $h_1;K_1[ell_1 := !ell_1 + 1] \downarrow\downarrow W''.j\ h_2;K_2[ell_1 := !ell_1 - 1]$

Since h_1 and h_2 satisfy an extension of our world, we can pattern match, picking out our island.
We've preserved the invariant we set up: We can change the piece of the heaps concerning our island.

WK: $\exists n,h'_1,h'_2.$
$h_1 = [ell_1 \mapsto n] \uplus h'_1 \wedge h_2 = [ell_2 \mapsto n] \uplus h'_2 \wedge$
$\forall n'.\ [ell_1 \mapsto n'] \uplus h'_1 \wedge h_2 = [ell_2 \mapsto n'] \uplus h'_2 : W''$.

STS: $h_1[ell_1 := n+1];K_1[()] \downarrow\downarrow W''.j\ h_2[ell_2 := -n-1];K_2[()]$.
WK $h_1[ell_1 := n+1],h_2[ell_2 := -n-1] : W'' \wedge$
$(W'',K_1,K_2) \in K[1] \wedge$
$(W'',(),()) \in V[1]$.

The proof for the second conjunct is similar.
Q.E.D.

— Motiving example for transitional invariants

Here's the simplest example not easily handled by our fixed invariants.

Example (Awkard (Pitts and Stark)):
$\tau = (1 \to 1) \to int$
$v_1 = \lambda f.(f(); 1)$
$e_2 = let\ x = ref\ 0\ in\ \lambda f.(x := 1; f(); !x)$.

Our notion of invariant is too weak. The idea here is that once we set x to 1, we'll never change it back. Thus v_1 and e_2 are equivalent. We can't express an invariant satisfied by x starting out 0 and later becoming 1 (and staying 1).

What we want to do is say there are two states this module can be in. In the initial state, x points to zero. In the second state, x points to one. The x points to zero state is initial. The x points to one state, final. There's one transition:

$$x\hookrightarrow 0 \longrightarrow \quad x\hookrightarrow 1$$

We can then reason about Awkward. We set up an island with such a transitional invariant. Before calling f, we know we're in the $x\hookrightarrow 1$ state. We know that no context can move back to the $x\hookrightarrow 0$ state.