— Proving the Støvring and Lassen Example

Let's start with the Støvring and Lassen (2007) example from several lectures ago. Recall:

$\tau := \mu\alpha.\neg\text{int} \longrightarrow \alpha$
$\varphi_1 := \text{fix f(x:}\tau\text{):int. callcc(k.f((unfold x)k))}$
$\varphi_2 := \lambda y{:}\tau.\text{callcc(k.(fix f(x:}\tau\text{):int. f((unfold x)k)) y).}$

Informally, φ_1 and φ_2 are similar, except that φ_1 grabs its continuation each time around the loop. These should be the same since the continuation each time around the loop should be the same (it's a tail recursive function).

Proposition: $\varphi_1 \equiv \varphi_2 : \tau \longrightarrow \text{int}$.

We can prove this in the step-indexed, $\top\top$-closed model.

Proof:

Let $n \in \mathbb{N}$.
TS: $(n,\varphi_1,\varphi_2) \in V[\tau \longrightarrow \text{int}]$.

Let j, v_1, v_2. Assume $j \leq n$ and $(j,v_1,v_2) \in V[\tau]$.
TS: $(F_1,F_2) \in E[\text{int}]$
where $F_1 :=$ callcc(k.f((unfold v_1)k))
and $F_2 :=$ callcc(k.(fix f(x:τ):int. f((unfold x)k)) v_2).

Let K_1, K_2. Assume $(j,K_1,K_2) \in K[\text{int}]$.
TS: $K_1[F_1] \downarrow\downarrow j\ K_2[F_2]$.

Aside: You can think of these kinds of proofs as being coinductive arguments (basically bisimulations) established by induction on steps.

WK:
$K_1[F_1] \longmapsto^* K_1[\varphi_1(\text{unfold } v_1)(\text{cont } K_1)] \wedge$
$K_2[F_2] \longmapsto^* K_2[F\_\{K_2\}(\text{unfold } v_2)(\text{cont } K_2)]$
where $F\_\{K_2\} :=$ fix f(x:τ):int. f((unfold x)(cont K_2)).

STS: $K_1[\varphi_1(\text{unfold } v_1)(\text{cont } K_1)] \downarrow\downarrow j$
$K_2[F\_\{K_2\}(\text{unfold } v_2)(\text{cont } K_2)]$.
This is basically where we got stuck when we last tried to prove this example.

This is our goal. Intuitively, it's our coinductive hypothesis. So how do we prove it? On Dec 11th, we tried to proceed without generalizing. Let's generalize over all values w_1, w_2 in these contexts and over all step indices $i \leq j$.

We'll prove the following lemma, instantiated with i := j and w_i := v_i to complete this proof.

Q.E.D.

Lemma:

$\forall i \leq j.\ (i, w_1, w_2) \in V[\tau] \Longrightarrow$
$K_1[\varphi_1(\text{unfold } w_1)(\text{cont } K_1)] \downarrow\downarrow i$
$K_2[F\_\{K_2\}(\text{unfold } w_2)(\text{cont } K_2)]$.

Proof:

By induction on i.
Let $i > 0$ and assume $(i, w_1, w_2) \in V[\tau]$.
WK: $(i, \text{unfold } w_1, \text{unfold } w_2) \in E[\neg\text{int} \longrightarrow \tau]$.

By downward closure of K[],
WK: $(i, \text{cont } K_1, \text{cont } K_2) \in V[\neg\text{int}]$.

$\therefore$ $(i, (\text{unfold } w_1)(\text{cont } K_1), (\text{unfold } w_2)(\text{cont } K_2)) \in E[\tau]$.

STS: $(i, K_1[\varphi_1\ \bullet], K_2[F\_\{K_2\}\ \bullet]) \in K[\tau]$.

Let $i' \leq i$, w'_1, w'_2. Assume $(i', w'_1, w'_2) \in V[\tau]$.
TS: $K_1[\varphi_1\ w'_1] \downarrow\downarrow i'\ K_2[F\_\{K_2\}\ w'_2]$.

WK:
† $K_1[\varphi_1\ w'_1] \mapsto+ K_1[\varphi_1((\text{unfold } w'_1)(\text{cont } K_1))]$
$K_2[F\_\{K_2\}\ w'_2] \mapsto+ K_2[F\_\{K_2\}(\text{unfold } w'_2)(\text{cont } K_2))]$.

It's important that the left-hand side takes a positive number of steps.

By downward closure,
$(i'-1, w'_1, w'_2) \in V[\tau]$.
By the IH,
$K_1[\varphi_1((\text{unfold } w'_1)(\text{cont } K_1))] \downarrow\downarrow (i'-1)$
$K_2[F\_\{K_2\}(\text{unfold } w'_2)(\text{cont } K_2))]$.
Since we took a positive numbe of steps in †, we have
$K_1[\varphi_1\ w'_1] \downarrow\downarrow i'\ K_2[F\_\{K_2\}\ w'_2]$.

Q.E.D.

There's a clear logical structure to how you reason about these things. It seems like you shouldn't need to fuss with the step indices. All statements “internalize” the notion of step-wise approximation.

Question from Deepak: Can we prove and then use a general fixed point rule? We did a similar induction in our last proof.

Answer:

> If you just have a rule for fixed points, then it only helps proving terms related. In this example, it wouldn't help. Our lemma worked with whole programs

— Logical step-indexed logical relations

[Aside from Dave: Derek threw up his paper. I couldn't really keep up.]

The idea: In doing these logical relations proofs, Derek quickly got bored. The step indices quickly got annoying and tedious. It seemed to him that it couldn't possibly matter (most of the time) how many steps you count.

The key ideas behind the paper:

• To use a relational logic for expressing parametricity proofs. That idea was old (if uncovered so far in the class). See Plotkin and Abadi's logic for parameteric polymorphism. The idea, basically, is you can express proofs using parametricity in a clean, abstract way. It's essentially a second-order logic with a primitve notion of typed relations. The nice thing, then, is you can express as a logical formula the statment saying when two things are related.

• Why do this? We want to hide the steps. But the steps pervade all the reasoning. We have to lift everything—most importantly, the statement when two things are logically related—into the logic. What is the logical way of accounting for these steps?

Two ideas: Appel and a bunch of other people (a very modal model of a modern major type system) plus Abadi-Plotkin. Combining these ideas seemed to be very useful for structuring step-indexed models. The idea: A later modality. It gives you way of saying φ is true “one step in the future”. This modality gives you a very easy way to express the limited references to steps you actually need in your proofs.

Derek threw up the paper.

The paper's language is quite close to the one Amal used (for comparison).

Figure 3 presents the syntax of the core logic.

There is a lot of standard stuff.

Interesting: A primitive notion of relations.

∀X.P is just first-order quantification.

If we want things to be values, we have a predicate Val(e).

∀R.P is quantification over relations.

There are constructors for building relations. Examples:
\bar{x}.P = (x_1,x_2).P := { (x_1,x_2) | P }.

The interesting new bit: recursive relations μr.R give you a way to write down what it means to be logically related at recursive type. There are restrictions on what these things can be. They have to be /contractive/. Basically, the meaning of R only depends on r one step later. Intuitively, r may only appear in R under a later modality.

An interesting new bit: ▷P. This is true at step-level n if P true at step-level n+1.

So the idea is we'll interpret the logic in Figure 3 over steps. The meaning of a formula depends on a step-index. Basically, all statements are step-indexed so you can hide most work with indices.

§3.2. On page 7, they give a model for the core logic.
We assume that all things are downward closed.
(Written as semantic truth values form a complete Heyting algebra in the paper.)

All propositions are true at n=0.

To read the rules for n>0:

> δ is a mapping from the relation variables in R to some interpretations of those relations. It's analogous to the ρ's in our work. Semantic, world-indexed, monotone relations.
>
> A is an atomic proposition. We rely on some (not-step indexed) interpretation function to make sense of those base facts.
>
> The interpretations of μr.R (gets smaller) and ▷(P₁,⋯,P_n) (gets larger) uses a trick. The whole thing is defined first by induction on n and then by induction on the size of the relation. The size function is defined so that ▷φ has size zero for any φ.
>
> The caligraphic P is just a list P₁,⋯,P_n of atomic propositions akin to a context.

Aside from Deepak/Derek: The whole point of the construction is to isolate the atomic things A from the step indices. In principle, you could try to push the step indices into A and their interpretation function I.

Figure 5 provides the core logic's inference rules.

The idea:

> Set up this core logic. (Prove it sound.)
>
> Then extend the logic with useful primitives for talking about programs.
>
> Then define the logical relation purely within the logic. (Do all the LR metatheory within the logic.)

The only thing we have to prove sound are the axioms in Figure 5. The judgement is interpreted:

> $X; R; P \vdash P := \forall n \geq 0.\ \forall \gamma \in [[X]].\ \forall \delta \in [[R]].\ [[\gamma P]]\delta n \implies [[\gamma P]]\delta n.$

Aside: It's not clear that these are the “right” axioms.

How did they generate these axioms? They wrote down what holds for the standard step-indexed models. Some of these axioms do not hold when

you add state. Derek would have to look it up; maybe existentials or universals.

Aside: Neel's recent work avoids the $\triangleright\Longrightarrow$ axiom.

> Distributivity axioms: $\triangleright$ distributes with all connectives.
>
> Some axioms that seemed convenient for replacement of syntactically equal terms ($e_1 = e_2$ is an atomic proposition) and semantically equal relations.

The most interesting rules are MONO and LÖB.

Löb is the coinduction princple. As long as you can show P is true now, given that its true one step later, then its true now. This is the idea we used in our proofs. The nice thing here: You don't have to talk about n. (Recall the logic defines all things true at n=0. This means you have to be careful at the n=1 case.)

Note the use of a syntactic equality $e_1 = e_2$ differs from Plotkin-Abadi logic.

Note the use of $R_1 \equiv R_2$ is relational equality (definable in the logic).

§4.2 covers the atomic relations they needed.

> These are all over closed terms.
>
> Interestingly: They only count unroll/roll reductions. This is slightly more precise than Amal who counts everything. See $I(e_1 \twoheadrightarrow_0 e_2)$ and $I(e_1 \twoheadrightarrow_1 e_2)$
>
> Interestingly, the model is made complete without $\top\top$-closure. If you're using a step-indexed model, you can make it complete as follows. Recall we started with
>
> $$E[\tau]\rho := \{ (n,e_1,e_2) \mid \forall j \le n.\ e_1 \downarrow j\ v_1 \Longrightarrow \exists v_2.\ e_2 \downarrow v_2 \wedge (n-j,v_1,v_2) \in V[\tau]\rho \}$$
>
> We can make it complete by replacing $\downarrow v_2$ with ciu-approximation:

$$E[\tau]\rho := \{ (n,e_1,e_2) \mid \forall j \le n.\ e_1 \downarrow j\ v_1 \Longrightarrow \exists v_2.\ v_2 \le u\ e_2 \wedge (n-j,v_1,v_2) \in V[\tau]\rho \}$$

Recall that to prove completeness, we needed a ciu-approximation closure property. We can bake it in.

Returning to the paper. You can state whatever rules you want about your atomic propositions so long as you can prove them “offline”.

§4.3 On page 12, they give a step-indexed logical relation that almost does not mention steps. It uses some notation given at the end of §4.2.

Note the defintion of the LR is a meta-level function. Given a type and a ρ, it gives you a formula in the logic. (The logic does not define $V[\tau]\rho$, the metalevel does.)

At a high-level, if you take this syntactic LR and merge it with the semantics of the logic, then you get back something quite close the step-indexed logical relations we've been working with. Derek et al factored out the work with step indices.

Note in the LR for $E[\tau]\rho$, you have a case distinction. You don't want to case analyze using proofs. They provided some lemmas to avoid most of that.

In Figure 7, they give some useful derivable (modulo the facts proved “outside” about our atomic formulas) rules for proving things in the logical relation. To actually prove these properties, they used Löb induction.

The Bind rule—more general than Bind2 and the bind we've used—lets us prove

$\exists v.\ x+y\downarrow v \wedge y+x\downarrow v$
$(v,v)\in V[\text{int}]$
—··· EXP and CIU ···
··· $\vdash (x+y,y+x) \in E[\text{int}]$
—CIU
$y, f_1 1 \downarrow y,\ y \le u\ f_2 1, \cdots, \vdash (x+y,y+f_2 0) \in E[\text{int}]$
—BIND2
$\cdots,x,f_1 0\downarrow x,\ x \le u\ f_2 0,\text{Val}(x) \vdash (x+f_1 1,f_2 1+f_2 0) \in E[\text{int}]$
=

$\cdots, x_1, x_2, f_1 0 \downarrow x_1, x_2 \leq_u f_2 0, (x_1, x_2) \in V[\text{int}] \vdash (x_1 + f_1 1, f_2 1 + f_2 0) \in V[\text{int}]$

—BIND

† $f_1, f_2, (f_1, f_2) \in V[\text{int} \to \text{int}] \vdash (f_1 0 + f_1 1, f_2 1 + f_2 0) \in E[\text{int}]$

—

$\vdash (\lambda f.f(1)+f(0), \lambda f.f(0)+f(1)) \in V[(\text{int} \to \text{int}) \to \text{int}]$

It's easy to show the first premise in the Bind rule. Under the context in †, $\vdash (f_1 0, f_2 0) \in E[\text{int}]$.
It's easy to show that

$x \leq_u f_2 0 \implies y + x \leq y + f_2 0$.

We couldn't prove this with our biorthogonal model. Since it's compatible with callcc and continuations ruin things.

The FUNEXT rule only works because they only count roll-unroll reductions as steps that matter. Compare it to the model we defined:

$V[\sigma \to \tau]\rho := \{ (n, \lambda x.e_1, \lambda x.e_2) \mid \forall j \leq n.\ \forall v_1, v_2.$
$(j, v_1, v_2) \in V[\sigma]\rho \implies (j, e_1[v_1/x], e_2[v_2/x]) \in E[\tau]\rho \}$

We could not prove the property in FUNEXT given this definition.

Note on the metatheory for this logical relation: Adequacy is the one thing that requires an appeal to the model.

Another nice thing this paper does: It gives (in the logic) a symmetric version of the LR. The direction d of logical approximation becomes a parameter of the judgement. If you can prove your judgement parametrically in d, then you've proven a symmetric version. Figure 9 provides symmetric versions of several derivable rules.