Boundary Types

Nico

January 5, 2015

1 Goals

The goal of this document is to motivate a three-world interpretation of two-staged languages. I'll be using some labels in a way that is completely opposite from previous formulations, so reader beware.

2 Two Independent Languages

As our baseline, we start with two independent languages, called L1 and L2. These languages can in general have any desired set of constructs. In this document, we'll assume that both languages have products, sums, functions, and some base types.

The valid types of L1 and L2 are given by the judgements A type @ 1 and A type @ 2, respectively. Likewise, the typing judgements are given by $e : A$ @ 1 and $e : A$ @ 2. As it stands, these languages are completely independent. That is, the rules for the @ 1 judgements and @ 2 judgements never depend on each other.

We call the thing after the @ sign a *world*. So far, this has only been either 1 or 2. We can save space in our presentation by abstracting judgements over worlds, conventionally using the metavariable w. Thus, the valid type judgement becomes A type @ w and the typing judgement becomes $e : A$ @ w. We can also define the rules parametrically over world, where appropriate.

3 Bridging the Languages: A Roadmap

Our goal in this project is to define some sort of linguistic superstructure that bridges L1 and L2. In particular, we'll be looking for a way to do this that admits a *temporal interpretation*, wherein the language L1 is thought to operate at one point in time, the language L2 is thought to operate at a later point in time, and information cannot pass from L2 to L1, lest there be a violation of causality. Strictly speaking, we should probably call this *directionality*, but we go with a temporal metaphor, because time is the most accessable example of a directional process.

We'll start by adding a new world $\mathbb{M}$ (for *mixed*) which encapsulates and coordiantes the other two languages. We'll then proceed by adding features, one at a time, to the $\mathbb{M}$-level language, starting with structure constructs, then moving on to products, functions, and sums.

All along the way, we'll be keeping track of how well the language admits temporal interpretation.

4 Structural Constructs

We start with a few structural constructs to get us off the ground, the most important being *encapsulation* rules, at the type and term level. These rules allow us to encapsulate computations of world 1 or 2, and pass around their result at the mixed level.

$$\frac{\Gamma \vdash e : A \mathbin{@} 1}{\Gamma \vdash e : \nabla A \mathbin{@} \mathbb{M}} \qquad\qquad \frac{\Gamma \vdash e : A \mathbin{@} 2}{\Gamma \vdash e : \bigcirc A \mathbin{@} \mathbb{M}}$$