Structures for Structural Recursion

(Appendix)

Paul Downen   Philip Johnson-Freyd   Zena M. Ariola
University of Oregon, USA
{pdownen,philipjf,ariola}@cs.uoregon.edu

A. Strong Normalization

We now give a proof sketch for strong normalization of well-typed commands of the $\mu\tilde{\mu}_{\mathcal{S}}$-calculus (the full details of the proof follow afterward in expanded form). Our approach uses techniques based on both orthogonality [7] and a variant of Barbanera and Berardi's symmetric candidates [1]. The proof is given parametrically with respect to $\mathcal{S}$, so long as it satisfies some general conditions. First, $\mathcal{S}$ must be *stable*, meaning that (co-)values are closed under reduction and substitution, and non-(co-)values are closed under substitution and ς reduction. Second, $\mathcal{S}$ must be *focalizing*, meaning that (co-)variables, structures built from other (co-)values, and case abstractions must all be (co-)values. The latter criteria comes from focalization in logic [2, 7, 9]—each criterion comes from an inference rule for typing a (co-)value in focus. For the remainder of the section, we will assume $\mathcal{S}$ is stable and focalizing, but otherwise arbitrary.

Our main goal is now to come up with a suitable model of types that ensures strong normalization. Since types in $\mu\tilde{\mu}_{\mathcal{S}}$ classify both terms and co-terms, we build a model for types as a pair of sets of terms and co-terms, called *pre-types*. These pre-types relate to one another in two fundamental ways. On the one hand we have refinement: $\mathcal{A}$ refines $\mathcal{B}$ ($\mathcal{A} \sqsubseteq \mathcal{B}$) if all the (co-)terms of $\mathcal{A}$ are included in $\mathcal{B}$. On the other hand we have sub-typing: $\mathcal{A}$ is a subtype of $\mathcal{B}$ ($\mathcal{A} \leq \mathcal{B}$) if the terms of $\mathcal{A}$ are included in $\mathcal{B}$ and the co-terms of $\mathcal{B}$ are included in $\mathcal{A}$. That directions in sub-typing are reversed arises naturally from the view of co-terms as "experiments" or properties on values, and captures Liskov's substitution principle [6]: if $\mathcal{A} \leq \mathcal{B}$, then properties on the objects of $\mathcal{B}$ hold for $\mathcal{A}$ as well.

The fundamental operation on pre-types is orthogonality [7]: $\mathcal{A}^{\perp}$ contains all the strongly normalizing (co-)terms that form strongly normalizing commands with any (co-)term of $\mathcal{A}$. However, orthogonality alone is not enough to define the meaning of types in a classical setting [5]. For example, we need to be able to justify the Act and $CoAct$ rules for typing μ- and $\tilde{\mu}$- abstractions, which is not trivial [7], and requires us to know something more about reduction. Our main idea is that types need to be saturated under the reductions which happen at the top level (or "head") of the command. Furthermore, these head reductions can be classified into three groups based on their *charge*. In a positive head reduction, the term side of a command takes control. Dually, in a negative head reduction ($\mapsto_{-}$), the co-term side of a command takes control. More specifically, positive head reductions are given by the rules for μ_E, ς for data structures, and unfolding Ascend at the head of a command (ς head reduction occurs in commands of the form $\langle C^{\mathsf{K}}_{\varsigma}[v] \| E\rangle$ or $\langle C^{\mathsf{K}}_{\varsigma}[e] \| E\rangle$, and unfolding is similar). Dually, negative head reductions are given by the rules for $\tilde{\mu}_V$, ς for co-data structures and unfolding $\mathsf{Descend}$. Neutral head reductions ($\mapsto_0$) involve the term and co-term operating in concert, and include the remaining rules. We can then talk about the head of a pre-type $\mathcal{A}$, written $Head(\mathcal{A})$. A strongly normalizing term v is in $Head(\mathcal{A})$ whenever $\langle v \| E\rangle \mapsto_{+} c$ and c is strongly normalizing, for any co-value E in $\mathcal{A}$. The co-terms in $Head(\mathcal{A})$ are defined dually by negative head reductions.

With this machinery, we are finally ready to define the notion of reducibility candidate [4] that encompasses our intended meaning of types. A *reducibility candidate* is any pre-type $\mathcal{A}$ that is *orthogonally sound* ($\mathcal{A} \sqsubseteq \mathcal{A}^{\perp}$), *orthogonally complete* ($\mathcal{A}^{\perp} \sqsubseteq \mathcal{A}$) and includes its own head ($Head(\mathcal{A}) \sqsubseteq \mathcal{A}$). These three criteria include everything we need to know to ensure that typing implies strong normalization. Orthogonal soundness ensures us that Cut is sound. Orthogonal completeness ensures that open commands are still strongly normalizing. The inclusion of $Head$ ensures that more complex (co-)terms like the $\mu-$ and $\tilde{\mu}-$ abstractions or general (co-)data structures are included. Indeed, head reductions are used to characterize these complex (co-)terms by their behavior instead of by their syntax in a general and extensible way.

We use the fixed point construction of the symmetric candidates technique [1, 5] to simplify the process of ensuring the meaning of a type is a reducibility candidate: so long as we can fully describe a type by a pre-type of *simple* (co-)terms which can never take a positive or negative step, we can automatically generate a corresponding reducibility candidate for that type. The idea is to define the saturation of a pre-type that adds the missing (co-)terms. A strongly normalizing term v is in $Sat(\mathcal{A})$ when either $\langle v \| E\rangle$ is strongly normalizing or $\langle v \| E\rangle \mapsto_{+} c$ and c is strongly normalizing for every co-value E in $\mathcal{A}$, and dually for co-terms in $Sat(\mathcal{A})$. Sat preserves the sub-typing order of the lattice of pre-types, so we can take the fixed point (where $\sqcup$ unions the (co-)terms of pre-types).

Lemma 1. *For any $\mathcal{C}$, there is a solution to $\mathcal{A} = \mathcal{C} \sqcup Sat(\mathcal{A})$.*

Furthermore, as long as $\mathcal{C}$ is a forward closed, orthogonally sound pre-type of simple (co-)values, then this fixed point is a reducibility candidate. Thus, this fixed point construction gives us an operation, $\mathcal{R}(-)$, for generating saturated reducibility candidates from these simplified pre-types.

Types are interpreted by means of a generalized fold over the structure of their syntax. In general, we say $[\![\mathsf{F}(\overrightarrow{A})]\!]_{\mathcal{H}} = \mathcal{H}(\mathsf{F}(\overrightarrow{[\![A]\!]_{\mathcal{H}}}))$ where the map $\mathcal{H}$ determines the meaning of the constructed type from the meaning of its constituent parts. This $\mathcal{H}$ operates over *hybrid types* consisting of a *syntactic* type constructor with *semantic* sub-components. Our task is to come up with such an $\mathcal{H}$ that interprets all meaningful hybrid types. For each type constructor we give an axillary meaning function that describes the *core* meaning of hybrid types, $\langle\!\langle \mathsf{F}(\overrightarrow{A}) \rangle\!\rangle_{\mathcal{H}}$. Data types, like $\otimes$, are defined by their constructors:

$$v \in \langle\!\langle \mathcal{A} \otimes \mathcal{B} \rangle\!\rangle_{\mathcal{H}} \iff v \in (Cons_{\mathcal{H}}(\mathcal{A} \otimes \mathcal{B}), \emptyset)^{\perp s \perp s}$$
$$e \in \langle\!\langle \mathcal{A} \otimes \mathcal{B} \rangle\!\rangle_{\mathcal{H}} \iff e \in (Cons_{\mathcal{H}}(\mathcal{A} \otimes \mathcal{B}), \emptyset)^{\perp s}.$$

where $\mathcal{A}^{\perp s}$ is a refinement of the orthogonal of $\mathcal{A}$ including only simple (co-)terms. The operation $Cons_{\mathcal{H}}(\mathcal{A} \otimes \mathcal{B})$ gives a set:

$$Cons_{\mathcal{H}}(\mathcal{A} \otimes \mathcal{B}) \triangleq \{(V_1, V_2) | V_1 \in \mathcal{A}, V_2 \in \mathcal{B}\}.$$

We now give an $\mathcal{H}$ such that $\mathcal{R}(\langle\!\langle \mathsf{F}(\vec{A}) \rangle\!\rangle_{\mathcal{H}}) = \mathcal{H}(\mathsf{F}(\vec{A}))$, by well-founded recursion over hybrid types based on their dependencies. The syntactic well-formedness checks for (co-)data declarations ensure well-foundedness of the dependency order of hybrid types, such that changing $\mathcal{H}$ for any $h' > h$ does not change $\langle\!\langle h \rangle\!\rangle_{\mathcal{H}}$ (this condition is an example of "contractive functions" in the general framework of *Complete Ordered Families of Equivalence* [3]).

However, we must take care with size types. We use $\mathcal{H}$ to interpret type variables, as well as giving the meaning of 0, $M+1$, and ∞. We interpret sizes (both Ix and Ord) as countable ordinals, allowing for induction. It might seem that we could just set $\mathcal{H}(0) = 0$, but this does not work in general. The problem is that we might have a sequent, $c : \Gamma \vdash_{i<0} \Delta$, where we still want to prove that c is strongly normalizing. We therefore need to have *some* valid interpretation function $\mathcal{H}$ where $\mathcal{H}(i) < \mathcal{H}(0)$. *The meaning of zero might not be zero.* Luckily, we can come up with such an $\mathcal{H}$ by examining the length of Θ. Although case analysis may introduce new bindings like $i < j$, for which it is impossible to ensure the existence of a value less than $\mathcal{H}(j)$, this is not a problem since we only reduce inside case abstractions when all kinds are inhabited due to the reachability caveat. The meaning of a set of declared type constructors, $\mathcal{H} \in [\![\mathcal{F}]\!]$, includes partial functions from hybrid types to semantic types (reducibility candidates, countable ordinals, and functions between them) such that (1) given any $\mathsf{F} \in \mathcal{F}$ and appropriate arguments $\vec{\mathcal{A}}$ we have that $\langle\!\langle \mathsf{F}(\vec{\mathcal{A}}) \rangle\!\rangle_{\mathcal{H}} \sqsubseteq \mathcal{H}(\mathsf{F}(\vec{\mathcal{A}}))$, (2) $\mathcal{H}(0)$ and $\mathcal{H}(\infty)$ are ordinals with $\mathcal{H}(\infty)$ as a limit ordinal and $\mathcal{H}(0) < \mathcal{H}(\infty)$, and (3) $\mathcal{H}(+1(\mathcal{M})) > \mathcal{M}$ for any ordinal $\mathcal{M}$.

Lemma 2. *For any well-formed $\mathcal{F}$, there exists a $\mathcal{H} \in [\![\mathcal{F}]\!]$.*

Another source of complexity is that some syntactic types matter computationally, in the case of indices. Reduction rules (e.g for $\mathsf{Inflate}$) depend on the syntactic form of the index. We resolve this by interpreting kinds not just as sets of semantic objects, but also as logic relations between syntactic and semantic types [8]. Because kinds can depend on types, the meaning of semantic types and kinds (as sets) are intertwined.

$$[\![a]\!]_{\mathcal{H}} \triangleq \mathcal{H}(a) \qquad [\![\lambda a : k.B]\!]_{\mathcal{H}} \triangleq \lambda\mathcal{A} \in [\![k]\!].[\![B]\!]_{\mathcal{H}\{\mathcal{A}/a\}}$$
$$[\![\mathsf{F}(\vec{A})]\!]_{\mathcal{H}} \triangleq \mathcal{H}(\mathsf{F}(\overrightarrow{[\![A]\!]_{\mathcal{H}}})) \qquad [\![A\ B]\!]_{\mathcal{H}} \triangleq [\![A]\!]_{\mathcal{H}}([\![B]\!]_{\mathcal{H}})$$
$$[\![\star]\!]_{\mathcal{H}} \triangleq CR \qquad [\![\mathsf{Ix}]\!]_{\mathcal{H}} \triangleq \{\mathcal{H}(0)\} \cup \{\mathcal{H}(+1(\mathcal{M})) | \mathcal{M} \in [\![\mathsf{Ix}]\!]\}$$
$$[\![\mathsf{Ord}]\!]_{\mathcal{H}} \triangleq \mathbb{O} \qquad [\![< N]\!]_{\mathcal{H}} \triangleq \{\mathcal{M} \in \mathbb{O} | [\![N]\!]_{\mathcal{H}} \leq \mathcal{M}\}$$

$$\begin{aligned}
A\ [\![\star]\!]_{\mathcal{H}}\ \mathcal{A} &\iff true\\
A\ [\![k_1 \to k_2]\!]_{\mathcal{H}}\ \mathcal{A} &\iff \forall B [\![k_1]\!]_{\mathcal{H}} \mathcal{B}. A\ B\ [\![k_2]\!]_{\mathcal{H}}\ \mathcal{A}(\mathcal{B})\\
M\ [\![\mathsf{Ix}]\!]_{\mathcal{H}}\ \mathcal{M} &\iff [\![M]\!]_{\mathcal{H}} \sim_{\mathcal{H}} \mathcal{M}\\
M\ [\![\mathsf{Ord}]\!]_{\mathcal{H}}\ \mathcal{M} &\iff \exists M \twoheadrightarrow M'.[\![M']\!]_{\mathcal{H}} \leq \mathcal{M}\\
M\ [\![< N]\!]_{\mathcal{H}}\ \mathcal{M} &\iff \exists M \twoheadrightarrow M'.[\![M']\!]_{\mathcal{H}} \leq \mathcal{M} < [\![N]\!]_{\mathcal{H}}
\end{aligned}$$

Note that $[\![\blacksquare]\!]$ contains all these semantic kinds. Furthermore, $[\![\square]\!]$ adds the requirement that the relation between syntactic and semantic types is backward closed under syntactic β reduction of the λ-calculus: if $\kappa \in [\![\square]\!]$, $A \twoheadrightarrow_{\beta} B$, and $B\ \kappa\ \mathcal{B}$, then $A\ \kappa\ \mathcal{B}$.

We interpret typing environments as sets of substitutions.

$$\begin{aligned}
\gamma \in [\![\Gamma]\!]_{\mathcal{H}} &\iff \forall x : A \in \Gamma. x\{\gamma\} \in Val([\![A]\!]_{\mathcal{H}})\\
\delta \in [\![\Delta]\!]_{\mathcal{H}} &\iff \forall \alpha : A \in \Delta. \alpha\{\delta\} \in Val([\![A]\!]_{\mathcal{H}})
\end{aligned}$$

Kinding environments are interpreted as relations between syntactic substitutions and semantic interpretations.

$$\begin{aligned}
\theta [\![\epsilon]\!]_{\mathcal{H}} \mathcal{I} &\iff \forall h. \mathcal{H}(h) = \mathcal{I}(h)\\
\theta [\![\Theta', a : k]\!]_{\mathcal{H}} \mathcal{I} &\iff \exists \theta' [\![\Theta']\!]_{\mathcal{H}} \mathcal{I}'. a\{\theta\} [\![k]\!]_{\mathcal{I}} \mathcal{I}(a)\\
&\qquad \wedge (\forall b \neq a. b\{\theta\} = b\{\theta'\})\\
&\qquad \wedge (\forall h \neq a. \mathcal{I}(h) = \mathcal{I}'(h))
\end{aligned}$$

The main soundness property is that if $c : \Gamma \vdash_{\Theta} \Delta$ and $(\Gamma \vdash_{\Theta} \Delta)\,\mathbf{seq}$ are derivable, then for any $\mathcal{H} \in [\![\mathcal{F}]\!]$, where $\mathcal{H}$ assigns a big enough meaning to 0, then $[\![c : \Gamma \vdash_{\Theta} \Delta]\!]_{\mathcal{H}}$ as given in Figure 1. Further, because (co-)variables inhabit every reducibility candidate, $[\![c : (\Gamma \vdash_{\Theta} \Delta)]\!]_{\mathcal{H}}$ entails that c is strongly normalizing. Additionally, if $Erase(c) \to c'$ then there must be a command c'' such that $c \to c''$ and $c' = Erase(c'')$ since the type-erased reduction rules are more limited than the ones for non-erased commands. Thus, if a typed command c is strongly normalizing, then $Erase(c)$ must be as well.

Theorem 1. *If $c : \Gamma \vdash_{\Theta} \Delta$ and $(\Gamma \vdash_{\Theta} \Delta)\,\mathbf{seq}$, then c is strongly normalizing in the $\mu\tilde{\mu}_{\mathcal{S}}$-calculus. Furthermore, $Erase(c)$ is strongly normalizing in the type-erased $\mu\tilde{\mu}_{\mathcal{S}}$-calculus.*

B. Pre-types

A pre-type is a pair of a set of terms and a set of co-terms. If $\mathcal{A}$ is a pre-type, we write $v \in \mathcal{A}$ or $e \in \mathcal{A}$ to indicate that a given (co-)term is an element of that pre-type.

Since we are interested in strong normalization, we carve out the strongly normalizing commands and (co-)terms. The set $⫫$ of commands consists of all those commands which are strongly normalizing. $⫫$ is closed under reduction. The pre-type $\mathcal{W}$ of well-behaved (co-)terms is defined as containing all the strongly normalizing (co-)terms. Note that since all reducts of strongly normalizing commands and (co-)terms are themselves strongly normalizing, $⫫$ and $\mathcal{W}$ are forward closed: if $c \in ⫫$ and $c \to c'$ then $c' \in runs$, if $v \in \mathcal{W}$ and $v \to v'$ then $v' \in \mathcal{W}$, and if $e \in \mathcal{W}$ and $e \to e'$ then $e' \in \mathcal{W}$.

There are two fundamental orderings of pre-types. Given two pre-types $\mathcal{A}$ and $\mathcal{B}$, $\mathcal{A}$ *refines* $\mathcal{B}$, written $\mathcal{A} \sqsubseteq \mathcal{B}$, if and only if

$$\begin{aligned}
v \in \mathcal{A} &\implies v \in \mathcal{B}\\
e \in \mathcal{A} &\implies e \in \mathcal{B}
\end{aligned}$$

By contrast, $\mathcal{A}$ is a *subtype* of $\mathcal{B}$, written $\mathcal{A} \leq \mathcal{B}$, if and only if

$$\begin{aligned}
v \in \mathcal{A} &\implies v \in \mathcal{B}\\
e \in \mathcal{B} &\implies e \in \mathcal{A}
\end{aligned}$$

Note that these two orderings form a complete lattice on the set of strongly-normalizing pre-types. In terms of refinement, $(\emptyset, \emptyset)$ is the smallest element that refines every pre-type, and $\mathcal{W}$ serves as a largest element that every strongly-normalizing pre-type refines. In terms of subtyping, $(\emptyset, \{e \in \mathcal{W}\})$ is the smallest element that is a sub-type of every pre-type, and $(\{v \in \mathcal{W}\}, \emptyset)$ is the largest element that is a super-type of every pre-type. In general, we use square operations to refer to operations of the refinement lattice, and triangular operations to refer to the subtyping lattice. In particular given the semantic types $\mathcal{A} = (\mathcal{A}^+, \mathcal{A}^-)$ and $\mathcal{B} = (\mathcal{B}^+, \mathcal{B}^-)$, where $\mathcal{A}^+$ and $\mathcal{B}^+$ contain the terms of $\mathcal{A}$ and $\mathcal{B}$ and dually for $\mathcal{A}^-, \mathcal{B}^-$, we have the joins and meets of both orders:

$$\begin{aligned}
(\mathcal{A}^+, \mathcal{A}^-) \sqcup (\mathcal{B}^+, \mathcal{B}^-) &\triangleq (\mathcal{A}^+ \cup \mathcal{B}^+, \mathcal{A}^- \cup \mathcal{B}^-)\\
(\mathcal{A}^+, \mathcal{A}^-) \sqcap (\mathcal{B}^+, \mathcal{B}^-) &\triangleq (\mathcal{A}^+ \cap \mathcal{B}^+, \mathcal{A}^- \cap \mathcal{B}^-)\\
(\mathcal{A}^+, \mathcal{A}^-) \vee (\mathcal{B}^+, \mathcal{B}^-) &\triangleq (\mathcal{A}^+ \cup \mathcal{B}^+, \mathcal{A}^- \cap \mathcal{B}^-)\\
(\mathcal{A}^+, \mathcal{A}^-) \wedge (\mathcal{B}^+, \mathcal{B}^-) &\triangleq (\mathcal{A}^+ \cap \mathcal{B}^+, \mathcal{A}^- \cup \mathcal{B}^-)
\end{aligned}$$

$$
\begin{array}{ll}
[\![c : \Gamma \vdash_\Theta \Delta]\!]_{\mathcal{H}} \Leftrightarrow \forall \theta [\![\Theta]\!]_{\mathcal{H}} \mathcal{I}. \forall \gamma \in [\![\Gamma]\!]_{\mathcal{I}}, \delta \in [\![\Delta]\!]_{\mathcal{I}}. c\{\theta, \gamma, \delta\} \in \perp\!\!\!\perp & [\![\Gamma \vdash_\Theta v : A | \Delta]\!]_{\mathcal{H}} \Leftrightarrow \forall \theta [\![\Theta]\!]_{\mathcal{H}} \mathcal{I}. \forall \gamma \in [\![\Gamma]\!]_{\mathcal{I}}, \delta \in [\![\Delta]\!]_{\mathcal{I}}. v\{\theta, \gamma, \delta\} \in [\![A]\!]_{\mathcal{I}} \\
[\![\Theta \vdash A : k]\!]_{\mathcal{H}} \Leftrightarrow \forall \theta [\![\Theta]\!]_{\mathcal{H}} \mathcal{I}. A\{\theta\} [\![k]\!]_{\mathcal{I}} [\![A]\!]_{\mathcal{I}} & [\![\Theta \vdash A = B : k]\!]_{\mathcal{H}} \Leftrightarrow \forall \mathcal{I} \in [\![\Theta]\!]_{\mathcal{H}}. [\![A]\!]_{\mathcal{I}} = [\![B]\!]_{\mathcal{I}} \in [\![k]\!]_{\mathcal{I}}
\end{array}
$$

$$
[\![\Theta \vdash k : s]\!]_{\mathcal{H}} \Leftrightarrow \forall \mathcal{I} \in [\![\Theta]\!]_{\mathcal{H}}. [\![k]\!]_{\mathcal{I}} \in [\![s]\!] \qquad [\![(\vdash)\,\mathbf{seq}\,]\!]_{\mathcal{H}} \Leftrightarrow \mathit{true} \qquad [\![(\vdash_{\Theta, a:k})\,\mathbf{seq}\,]\!]_{\mathcal{H}} \Leftrightarrow [\![(\vdash_\Theta)\,\mathbf{seq}\,]\!] \wedge [\![\Theta \vdash k : \blacksquare]\!]_{\mathcal{H}}
$$

$$
[\![(\Gamma \vdash_\Theta \Delta)\,\mathbf{seq}\,]\!]_{\mathcal{H}} \Leftrightarrow [\![(\vdash_\Theta)\,\mathbf{seq}\,]\!]_{\mathcal{H}} \wedge (\forall x : A \in \Gamma. [\![\Theta \vdash A : \star]\!]_{\mathcal{H}}) \wedge (\forall \alpha : A \in \Delta. [\![\Theta \vdash A : \star]\!]_{\mathcal{H}})
$$

Figure 1. The semantic interpretation of sequents in the model.

Lemma 3 (Pre-type lattice)**.** *Each of the $\leq$ and $\sqsubseteq$ orderings on the set of pre-types form a complete lattice: that is, they have all joins and meets.*

Proof. The set of subsets of a set is a complete lattice ordered by $\subseteq$ with the usual $\bigcup$ and $\bigcap$ operations. Further, the dual of a complete lattice is itself a complete lattice, and the product of two complete lattices is a complete lattice. The case of $\sqsubseteq$ is the product of the two subset lattices; the case of $\leq$ is the product of the two subset lattices where one is dualized. $\square$

Lemma 4. *$\sqcup$ is monotonic (in both arguments) with respect to $\leq$ and is commutative and associative.*

Proof. That $\sqcup$ is commutative and associative follows immediately from its definition. The interesting fact is that, for any pre-types $\mathcal{A} = (\mathcal{A}^+, \mathcal{A}^-)$, $\mathcal{B} = (\mathcal{B}^+, \mathcal{B}^-)$, and $\mathcal{C} = (\mathcal{C}^+, \mathcal{C}^-)$ if $\mathcal{A} \leq \mathcal{B}$ then we have

$$
\begin{aligned}
\mathcal{A} \sqcup \mathcal{C} &= (\mathcal{A}^+ \cup \mathcal{C}^+, \mathcal{A}^- \cup \mathcal{C}^-) \\
&\leq (\mathcal{B}^+ \cup \mathcal{C}^+, \mathcal{B}^+ \cup \mathcal{C}^-) \\
&= \mathcal{B} \sqcup \mathcal{C}
\end{aligned}
$$

since $\mathcal{A}^+ \subseteq \mathcal{B}^+$ means $\mathcal{A}^+ \cup \mathcal{C}^+ \subseteq \mathcal{B}^+ \cup \mathcal{C}^+$ and $\mathcal{B}^- \subseteq \mathcal{A}^-$ means $\mathcal{B}^- \cup \mathcal{C}^- \subseteq \mathcal{A}^- \cup \mathcal{C}^-$. $\square$

The most basic operation on pre-types is the orthogonal $A^\perp$:

$$
\begin{aligned}
v \in \mathcal{A}^\perp &\iff v \in \mathcal{W} \wedge \forall e \in \mathcal{A}. \langle v \| e \rangle \in \perp\!\!\!\perp \\
e \in \mathcal{A}^\perp &\iff e \in \mathcal{W} \wedge \forall v \in \mathcal{A}. \langle v \| e \rangle \in \perp\!\!\!\perp
\end{aligned}
$$

We say that a pre-type $\mathcal{A}$ is *orthogonally sound* if and only if $\mathcal{A} \sqsubseteq \mathcal{A}^\perp$. In other words, for all $\mathcal{A} \sqsubseteq \mathcal{W}$ and for all $v, e \in \mathcal{A}$, $\langle v \| e \rangle \in \perp\!\!\!\perp$. Furthermore, a pre-type $\mathcal{A}$ is *orthogonally complete* if and only if $\mathcal{A}^\perp \sqsubseteq \mathcal{A}$. Therefore, an orthogonally sound and complete pre-type $\mathcal{A}$ is one such that $\mathcal{A} = \mathcal{A}^\perp$.

We can generalize the orthogonal operation to a general class of similar operations on pre-types that all share the same properties. We say that an operation on pre-types Op is a *negation operation inside* $\mathcal{D}$ if and only if $\mathcal{D}$ is a fixed pre-type and there exists predicates P and Q on term, co-term pairs such that:

$$
\begin{aligned}
v \in Op(\mathcal{A}) &\iff v \in \mathcal{D} \wedge \forall e \in \mathcal{A}. P(v, e) \\
e \in Op(\mathcal{A}) &\iff e \in \mathcal{D} \wedge \forall v \in \mathcal{A}. Q(v, e)
\end{aligned}
$$

Furthermore, Op is a *symmetric negation operation inside* $\mathcal{D}$ if and only if for all $v, e \in \mathcal{D}$, $P(v, e) \iff Q(v, e)$. Note that orthogonality is a symmetric negation operation inside $\mathcal{W}$. It follows that all such negation operations enjoy the standard basic properties of orthogonality.

Lemma 5 (Monotonicity)**.** *For any negation operation Op inside $\mathcal{D}$, $\mathcal{A} \leq \mathcal{B}$ implies $Op(\mathcal{A}) \leq Op(\mathcal{B})$.*

Proof. Suppose $v \in Op(\mathcal{A})$, so we know that $v \in \mathcal{D}$ and for all $e \in \mathcal{A}$, $P(v, e)$. Then, given any $e \in \mathcal{B}$, we know that $e \in \mathcal{A}$ because $\mathcal{A} \leq \mathcal{B}$, and so $P(v, e)$. Therefore, $v \in Op(\mathcal{B})$ as well.

Suppose $e \in Op(\mathcal{B})$, so we know that $e \in \mathcal{D}$ and for all $e \in \mathcal{B}$, $Q(v, e)$. Then, given any $v \in \mathcal{A}$, we know that $v \in \mathcal{B}$ because $\mathcal{A} \leq \mathcal{B}$, and so $Q(v, e)$. Therefore, $e \in Op(\mathcal{A})$ as well. $\square$

Lemma 6 (Contrapositive)**.** *For any negation operation Op inside $\mathcal{D}$, $\mathcal{A} \sqsubseteq \mathcal{B}$ implies $Op(\mathcal{B}) \sqsubseteq Op(\mathcal{A})$.*

Proof. Suppose $v \in Op(\mathcal{B})$, so we know that $v \in \mathcal{D}$ and for all $e \in \mathcal{B}$, $P(v, e)$. Then, given any $e \in \mathcal{A}$, we know that $e \in \mathcal{B}$ because $\mathcal{A} \sqsubseteq \mathcal{B}$, and so $P(v, e)$. Therefore, $v \in Op(\mathcal{B})$ as well.

Suppose $e \in Op(\mathcal{B})$, so we know that $e \in \mathcal{D}$ and for all $v \in \mathcal{B}$, $Q(v, e)$. Then, given any $v \in \mathcal{A}$, we know that $v \in \mathcal{B}$ because $\mathcal{A} \sqsubseteq \mathcal{B}$, and so $Q(v, e)$. Therefore, $e \in Op(\mathcal{B})$ as well. $\square$

Lemma 7 (Double Negation Introduction)**.** *For any symmetric negation operation Op inside $\mathcal{D}$, $\mathcal{A} \sqsubseteq \mathcal{D}$ implies $\mathcal{A} \sqsubseteq Op(Op(\mathcal{A}))$.*

Proof. For any $v \in \mathcal{A}$ and $e \in Op(\mathcal{A})$, $Q(v, e)$ by definition of $Op(\mathcal{A})$, so $P(v, e)$ as well since Op is symmetric. Therefore, $\mathcal{A} \sqsubseteq \mathcal{D}$ implies that $v \in \mathcal{D}$, so that $v \in Op(Op(\mathcal{A}))$. The case of $e \in \mathcal{A}$ implies $e \in Op(Op(\mathcal{A}))$ is dual. $\square$

Lemma 8 (Triple Negation Elimination)**.** *For any symmetric negation operation Op inside $\mathcal{D}$, $\mathcal{A} \sqsubseteq \mathcal{D}$ implies $Op(Op(Op(\mathcal{A}))) = Op(\mathcal{A})$.*

Proof. Note that $Op(\mathcal{A}) \sqsubseteq \mathcal{D}$ because Op is a negation operation inside $\mathcal{D}$. Therefore, by double negation introduction (Lemma 7), $Op(\mathcal{A}) \sqsubseteq Op(Op(Op(\mathcal{A})))$. Additionally, because $\mathcal{A} \sqsubseteq \mathcal{D}$, we know that $\mathcal{A} \sqsubseteq Op(Op(\mathcal{A}))$ by double negation introduction (Lemma 7), and so $Op(Op(Op(\mathcal{A}))) \sqsubseteq Op(\mathcal{A})$ by contrapositive (Lemma 6). Therefore, $Op(Op(Op(\mathcal{A}))) = Op(\mathcal{A})$. $\square$

We can rephrase the pre-type $\mathcal{W}$ of well-behaved (co-)terms solely in terms of orthogonality and (co-)variables. This ensures to us that any other pre-type $\mathcal{A} \sqsubseteq \mathcal{W}$ which is orthogonally complete must contain (co-)variables.

Lemma 9. *1. v is strongly normalizing iff $\langle v \| \alpha \rangle$ is.*
2. e is strongly normalizing iff $\langle x \| e \rangle$ is.

Proof. 1. Since v is a sub-term of $\langle v \| \alpha \rangle$, strong normalization of $\langle v \| \alpha \rangle$ implies strong normalization of v.

Going the other way, we show that every reduction of $\langle v \| \alpha \rangle$, except for possibly one top-level μ_E reduction, can be traced by v as well. We proceed to show that $\langle v \| \alpha \rangle$ is strongly normalizing because all of its reducts are by well-founded induction on $|v|$:

- Suppose $v = \mu\beta.c$, so that we have the top-level μ_E reduction:

  $$\langle \mu\beta.c \| \alpha \rangle \to_{\mu_E} c\{\alpha/\beta\}$$

  Furthermore, we know $\mu\alpha.c\{\alpha/\beta\}$ is strongly normalizing since it is α-equivalent to the strongly normalizing $\mu\beta.c$, which means that $c\{\alpha/\beta\}$ is also strongly normalizing since it is a sub-command of $\mu\alpha.c\{\alpha/\beta\}$.

- Suppose we have some other reduction internal to v, so that:

$$\langle v \| \alpha \rangle \to \langle v' \| \alpha \rangle$$

Then we know that $v \to v'$ so $|v'| < |v|$. Therefore, by the inductive hypothesis, we get that $\langle v' \| \alpha \rangle$ is strongly normalizing.
Since every reduct of $\langle v \| \alpha \rangle$ is strongly normalizing, then $\langle v \| \alpha \rangle$ is also strongly normalizing.

2. Analogous to the above by duality. □

Corollary 1. $\mathcal{W} = Var^{\perp}$.

Proof. Note that $Var^{\perp}$ is the semantic type:

$$v \in Var^{\perp} \iff v \in \mathcal{W} \wedge \forall \alpha \in Var. \langle v \| \alpha \rangle \in \perp\!\!\!\perp$$
$$e \in Var^{\perp} \iff e \in \mathcal{W} \wedge \forall x \in Var. \langle x \| e \rangle \in \perp\!\!\!\perp$$

And so $v, e \in \mathcal{W}$ if and only if $v, e \in Var^{\perp}$ by the above Lemma 9. □

Corollary 2. *If $\mathcal{A}^{\perp} \sqsubseteq \mathcal{A} \sqsubseteq \mathcal{W}$ then $Var \sqsubseteq \mathcal{A}$.*

Proof. Using the above Corollary 1, we conclude by double negation introduction (Lemma 7) and contrapositive (Lemma 8):

$$Var \sqsubseteq Var^{\perp\perp} = \mathcal{W}^{\perp} \sqsubseteq \mathcal{A}^{\perp} \sqsubseteq A$$ □

C. Head Reduction

There are two important properties for the chosen strategy $\mathcal{S}$ needed for the proof of strong normalization. First, we say a strategy *stable* if and only if:

1. (co-)values are closed under reduction and substitution, and
2. non-(co-)values are closed under substitution and ς reduction.

Second, we say a strategy is *focalizing* if and only if all:

1. (co-)variables,
2. structures built from (co-)values, and
3. case abstractions (recursive or non-recursive)

are considered (co-)values. The focalizing property of strategies corresponds to focalization in logic [7]—each criterion for a focalizing strategy comes from an inference rule for typing a (co-)value in focus. For the remainder of this proof, we assume that the chosen strategy $\mathcal{S}$ is both stable and focalizing.

The head reduction relation describes only those reductions that happen at the top of a command and is given in Figure 2. These reductions are *charged*. Neutrally charged reductions $\mapsto_0$ require cooperation of the term and co-term, like in the β rules. Positively charged reductions $\mapsto_+$ allow the term to take over the command in order to simplify itself. Negatively charged reductions $\mapsto_-$ are allow the co-term to take over the command. There are several useful facts that are immediately apparent about this definition of head reduction.

1. Every step of head reduction is simulated by several steps of the general reduction theory: $c \mapsto_{+,0,-} c'$ implies that $c \twoheadrightarrow c'$.
2. Head reduction only occurs when one side of the command is a (co-)value: if $\langle v \| e \rangle \mapsto_+$ then e is a co-value, if $\langle v \| e \rangle \mapsto_-$ then v is a value, and if $\langle v \| e \rangle \mapsto_0$ then both v and e are (co-)values. This last point about neutral head reductions implying both sides of the command is a (co-)value follows from the assumption that the chosen strategy $\mathcal{S}$ is focalizing.
3. When taken on their own, each of the charged head reduction relations, $\mapsto_0$, $\mapsto_+$, and $\mapsto_-$ are deterministic regardless of the chosen strategy, although the combined $\mapsto_{+/-}$ head reduction relation may be non-deterministic. Additionally, the combined $\mapsto_{+,0}$ and $\mapsto_{-,0}$ reduction relations are deterministic as well.

Furthermore, head reduction steps commute with the internal reductions inside either side of the command.

Lemma 10. *1. If $v \to v'$ and $\langle v \| e \rangle \mapsto_0 c$ then there is a c' such that $\langle v' \| e \rangle \mapsto_0 c'$ and $c \twoheadrightarrow c'$.*
2. If $e \to e'$ and $\langle v \| e \rangle \mapsto_0 c$ then there is a c' such that $\langle v \| e' \rangle \mapsto_0 c'$ and $c \twoheadrightarrow c'$.
3. If $v \to v'$ and $\langle v \| e \rangle \mapsto_- c$ then there is a c' such that $\langle v' \| e \rangle \mapsto_- c'$ and $c \twoheadrightarrow c'$.
4. If $e \to e'$ and $\langle v \| e \rangle \mapsto_+ c$ then there is a c' such that $\langle v \| e' \rangle \mapsto_- c$ and $c \twoheadrightarrow c'$.
5. If $v \to v'$ then either
(a) $\forall e, c$ such that $\langle v \| e \rangle \mapsto_+ c$ we have $c \twoheadrightarrow \langle v' \| e \rangle$, or
(b) $\forall e, c$ such that $\langle v \| e \rangle \mapsto_+ c$ there exists a c' such that $\langle v' \| e \rangle \mapsto_+ c'$ and $c \twoheadrightarrow c'$.
6. If $e \to e'$ then either
(a) $\forall v, c$ such that $\langle v \| e \rangle \mapsto_- c$ we have $c \twoheadrightarrow \langle v \| e' \rangle$ or
(b) $\forall v, c$ such that $\langle v \| e \rangle \mapsto_- c$ there exists a c' such that $\langle v \| e' \rangle \mapsto_- c'$ and $c \twoheadrightarrow c'$.

Proof. By cases on the possible reductions. Note that for statements 1-4, there are no critical pairs between neutral head reductions and general reductions, because (co-)values are closed under reduction by the stability of $\mathcal{S}$. Additionally, for the other statements, the fact that (co-)values are closed under reduction cuts out many critical pairs. For statement 5, we illustrate the remaining interesting critical pairs:

- $\mu\alpha.\langle v \| \alpha \rangle \to v$. Note that this is equivalent in the only possible head reduction in the command $\langle \mu\alpha.\langle v \| \alpha \rangle \| E \rangle \mapsto_+ \langle v \| E \rangle$, so we have case (a).
- $C^{\mathsf{K}}_{\varsigma}[v] \to \mu\alpha.\langle v \| \tilde{\mu}x.\langle C^{\mathsf{K}}_{\varsigma}[x] \| \alpha \rangle \rangle$. Given that
$$\langle C^{\mathsf{K}}_{\varsigma}[v] \| E \rangle \mapsto_+ \langle v \| \tilde{\mu}x.\langle C^{\mathsf{K}}_{\varsigma}[x] \| E \rangle \rangle$$
we have case (b) where c' is equal to the same $\langle v \| \tilde{\mu}x.\langle C^{\mathsf{K}}_{\varsigma}[x] \| E \rangle \rangle$ and
$$\langle \mu\alpha.\langle v \| \tilde{\mu}x.\langle C^{\mathsf{K}}_{\varsigma}[x] \| \alpha \rangle \rangle \| E \rangle \mapsto_+ \langle v \| \tilde{\mu}x.\langle C^{\mathsf{K}}_{\varsigma}[x] \| E \rangle \rangle$$
- $C^{\mathsf{K}}_{\varsigma}[e] \to \mu\alpha.\langle \mu\beta.\langle C^{\mathsf{K}}_{\varsigma}[\beta] \| \alpha \rangle \| e \rangle$. Analogous to a previous case by duality.
- $C^{\mathsf{K}}_{\varsigma}[v] \to C^{\mathsf{K}}_{\varsigma}[V]$. Given that
$$\langle C^{\mathsf{K}}_{\varsigma}[v] \| E \rangle \mapsto_+ \langle v \| \tilde{\mu}x.\langle C^{\mathsf{K}}_{\varsigma}[x] \| E \rangle \rangle$$
we have case (a) where
$$\langle v \| \tilde{\mu}x.\langle C^{\mathsf{K}}_{\varsigma}[x] \| E \rangle \rangle \to \langle V \| \tilde{\mu}x.\langle C^{\mathsf{K}}_{\varsigma}[x] \| E \rangle \rangle \to \langle C^{\mathsf{K}}_{\varsigma}[V] \| E \rangle$$
- $C^{\mathsf{K}}_{\varsigma}[e] \to C^{\mathsf{K}}_{\varsigma}[E]$. Analogous to a previous case by duality.
- $V\{N/i\} \to \mu(\mathsf{Rise}^{i<N}[\alpha](x).c\{M/j, V/x\})$ where $V = \mu(\mathsf{Rise}^{j<i}[\alpha](x).c)$. Note that this is equivalent to the only possible head reduction in the command $\langle V\{N/i\} \| E \rangle$, so we have case (a).

For statement 6, commutation follows analogously by duality. □

A term is *simple* if it never causes a positive head reduction. A co-term is simple if it never causes a negative head reduction. A pre-type is simple if all its (co-)terms are. Of note, observe that because our chosen strategy is assumed to be focalizing, all simple (co-)terms are (co-)values: a simple (co-)term can either take a neutral head reduction step, in which case it must be a (co-)value, or it is is a

$$\langle \mu\alpha.c \| E\rangle \mapsto_+ c\{E/\alpha\} \qquad \langle V \| \tilde{\mu}x.c\rangle \mapsto_- c\{V/x\}$$

$$\langle \mathsf{K}^{\vec{B}}(\vec{E},\vec{V}) \| \tilde{\mu}[\mathsf{K}^{\vec{b}}(\vec{\alpha},\vec{x}).c] \mid \ldots\rangle \mapsto_0 c\{\overrightarrow{B/b},\overrightarrow{E/\alpha},\overrightarrow{V/x}\} \qquad \langle \mu(\mathsf{H}^{\vec{b}}[\vec{x},\vec{\alpha}].c \mid \ldots) \| \mathsf{H}^{\mathsf{B}}[\vec{V},\vec{E}]\rangle \mapsto_0 c\{\overrightarrow{B/b},\overrightarrow{V/x},\overrightarrow{E/\alpha}\}$$

$$\langle C^{\mathsf{K}}_{\varsigma}[v] \| E\rangle \mapsto_+ \langle \mu\alpha.\langle v \| \tilde{\mu}y.\langle C^{\mathsf{K}}_{\varsigma} \| \alpha\rangle\rangle \| E\rangle \qquad \langle V \| C^{\mathsf{H}}_{\varsigma}[v]\rangle \mapsto_- \langle V \| \tilde{\mu}x.\langle v \| \tilde{\mu}y.\langle x \| C^{\mathsf{H}}_{\varsigma}[y]\rangle\rangle\rangle \quad \textbf{where}\, v \notin Value$$

$$\langle C^{\mathsf{K}}_{\varsigma}[e] \| E\rangle \mapsto_+ \langle \mu\alpha.\langle \mu\beta.\langle C^{\mathsf{K}}_{\varsigma}[\beta] \| \alpha\rangle \| e\rangle \| E\rangle \qquad \langle V \| C^{\mathsf{H}}_{\varsigma}[e]\rangle \mapsto_- \langle V \| \tilde{\mu}x.\langle \mu\beta.\langle x \| C^{\mathsf{H}}_{\varsigma}[\beta]\rangle \| e\rangle\rangle \quad \textbf{where}\, e \notin CoValue$$

$$\langle \mu(\mathsf{Rise}^{j<N}[\alpha](x).c) \| E\rangle \mapsto_+ \langle \mu(\mathsf{Rise}^{i<N}[\alpha].c\{i/j, \mu(\mathsf{Rise}^{j<i}[\alpha](x).c)/x\}) \| E\rangle$$

$$\langle V \| \tilde{\mu}[\mathsf{Fall}^{j<N}(x)[\alpha].c]\rangle \mapsto_- \langle V \| \tilde{\mu}[\mathsf{Fall}^{i<N}(x).c\{i/j, \tilde{\mu}[\mathsf{Fall}^{j<i}(x)[\alpha].c]\}]\rangle$$

$$\langle V \| \mathsf{Up}^0[E]\rangle \mapsto_0 c_0\{E/\alpha\} \qquad \langle V \| \mathsf{Up}^{M+1}[E]\rangle \mapsto_0 \langle \mu\beta.\langle V \| \mathsf{Up}^M[\beta]\rangle \| \tilde{\mu}x.c_1\{M/j, E/\alpha\}\rangle \qquad \textbf{where}\, V = \mu(\mathsf{Up}^0[\alpha].c_0 \mid \mathsf{Up}^{j+1}[\alpha](x).c_1)$$

$$\langle \mathsf{Down}^0(V) \| E\rangle \mapsto_0 c_0\{V/x\} \qquad \langle \mathsf{Down}^{M+1}(V) \| E\rangle \mapsto_0 \langle \mu\alpha.c_1\{M/j, V/x\} \| \tilde{\mu}y.\langle \mathsf{Down}^M(y) \| E\rangle\rangle \qquad \textbf{where}\, E = \tilde{\mu}[\mathsf{Down}^0(x).c_0 \mid \mathsf{Down}^{j+1}(x)[\alpha].c_1]$$

Figure 2. Head Reductions.

variable and doesn't actively participate in any reduction, so it must be a (co-)value by the assumption that the strategy is focalizing. Furthermore, the set of simple (co-)terms is closed under reduction because (co-)values are closed under reduction (since the strategy is stable). Also of note is the fact that all non-simple (co-)terms (that is, the ones which can cause a positive or negative head reduction) can never be part of a neutral reduction.

We define the operation $Simp(\mathcal{A})$ as containing all the (co-)terms of $\mathcal{A}$ which are simple. There is a simplified version of the orthogonality operation, $(-)^{\perp s}$, that operates in the simplified version of well-behaved (co-)terms $\mathcal{W}$. More specifically, the $(-)^{\perp s}$ is defined as:

$$\mathcal{A}^{\perp s} \triangleq Simp(\mathcal{A}^{\perp})$$

Note that $(-)^{\perp s}$ is a symmetric negation operation inside $Simp(\mathcal{W})$, so all the basic properties of monotonicity, contrapositive, double negation introduction, and triple negation elimination apply.

D. Reducibility Candidates

To define the the set of reducibility candidates, we have to determine the pre-types are sufficiently saturated so that they contain enough (co-)terms such that the general typing rules are sound, without invalidating the Cut rule. For example, we need to be sure that reducibility candidates contain the necessary μ- and $\tilde{\mu}$-abstractions according to Act and $CoAct$. We characterize this saturation in terms of head reduction, using the $Head(-)$ operation on pre-types defined as:

$$v \in Head(\mathcal{A}) \iff v \in \mathcal{W} \land \forall E \in \mathcal{A}. \langle v \| E\rangle \mapsto_+ c \in \perp\!\!\!\perp$$
$$e \in Head(\mathcal{A}) \iff e \in \mathcal{W} \land \forall V \in \mathcal{A}. \langle V \| e\rangle \mapsto_- c \in \perp\!\!\!\perp$$

Note that $Head(-)$ is a (non-symmetric) negation operation inside$\mathcal{W}$, so the monotonicity and contrapositive properties apply.

We now define reducibility candidates as all orthogonally sound and complete pre-types that contain their own $Head$:

Definition 1 (Reducibility candidates)**.** *A semantic type, $\mathcal{A}$, is a* reducibility candidate *iff $\mathcal{A} = \mathcal{A}^{\perp}$ and $Head(\mathcal{A}) \sqsubseteq \mathcal{A}$. The set of reducibility candidates is written CR.*

We can show that a reducibility candidate must contain all the μ- and $\tilde{\mu}-$ abstractions that are well-behaved when paired with any of its (co-)values. This follows from the fact that the μ_E and $\tilde{\mu}_V$ reductions are non-neutrally charged head reductions.

Lemma 11 (Strong activation)**.** *1. If $\mathcal{A}$ is a reducibility candidate and for all $E \in \mathcal{A}$, $c\{E/\alpha\} \in \perp\!\!\!\perp$, then $\mu\alpha.c \in \mathcal{A}$.*
2. If $\mathcal{A}$ is a reducibility candidate and for all $V \in \mathcal{A}$, $c\{V/x\} \in \perp\!\!\!\perp$, then $\tilde{\mu}x.c \in \mathcal{A}$.

Proof.
- Since $\mathcal{A}$ is a reducibility candidate, we know that $Head(\mathcal{A}) \sqsubseteq \mathcal{A}$. Observe that for all $E \in \mathcal{A}$,

  $$\langle \mu\alpha.c \| E\rangle \mapsto c\{E/\alpha\} \in \perp\!\!\!\perp$$

  by assumption. Therefore, $\mu\alpha.c \in Head(\mathcal{A}) \sqsubseteq \mathcal{A}$.
- Analogous to the previous statement by duality. □

Additionally, if a reducibility candidate contains all the structures built from (co-)values of other reducibility candidates, then it must contain the general constructs built from (co-)terms as well. This follows from the fact that all the lifting rules are implemented as non-neutrally charged head reductions.

Lemma 12 (Unfocalization)**.** *1. If $\mathcal{A}$,$\vec{\mathcal{B}}$,$\vec{\mathcal{C}}$ are reducibility candidates and for all $\overrightarrow{V \in \mathcal{B}}$,$\overrightarrow{E \in \mathcal{C}}$, $\mathsf{K}^{\vec{D}}(\vec{E},\vec{V}) \in \mathcal{A}$, then for all $\overrightarrow{v \in \mathcal{B}}$,$\overrightarrow{e \in \mathcal{C}}$, $\mathsf{K}^{\vec{D}}(\vec{e},\vec{v}) \in \mathcal{A}$.*
2. If $\mathcal{A}$,$\vec{\mathcal{B}}$,$\vec{\mathcal{C}}$ are reducibility candidates and for all $\overrightarrow{V \in \mathcal{B}}$,$\overrightarrow{E \in \mathcal{C}}$, $\mathsf{H}^{\vec{D}}[\vec{V},\vec{E}] \in \mathcal{A}$, then for all $\overrightarrow{v \in \mathcal{B}}$,$\overrightarrow{e \in \mathcal{C}}$, $\mathsf{H}^{\vec{D}}[\vec{v},\vec{e}] \in \mathcal{A}$.

Proof.
- We proceed by right-to-left induction on the immediate non-(co-)value sub-(co-)terms $\vec{e}$, $\vec{v}$. Note that because $\mathcal{A}$ is a reducibility candidate, $Head(\mathcal{A}) \sqsubseteq \mathcal{A}$, so it suffices to show that $\mathsf{K}^{\vec{D}}(\vec{e},\vec{v}) \in Head(\mathcal{A})$.
  - All of $\vec{e}$, $\vec{v}$ are (co-)values: $\mathsf{K}^{\vec{D}}(\vec{e},\vec{v}) \in \mathcal{A}$ by assumption.
  - All of $\vec{e}$ are co-values, and v_i is the right-most non-value in $\vec{v} = \vec{v'}, v_i, \vec{V'}$: Observe that for any $E \in \mathcal{A}$, we have the head lifting reduction

    $$\langle \mathsf{K}^{\vec{D}}(\vec{e},\vec{v}) \| E\rangle \mapsto \langle v_i \| \tilde{\mu}x.\langle \mathsf{K}^{\vec{D}}(\vec{e},\vec{v'},x,\vec{V'}) \| E\rangle\rangle \in \perp\!\!\!\perp$$

    because $\tilde{\mu}x.\langle \mathsf{K}^{\vec{D}}(\vec{e},\vec{v'},x,\vec{V'}) \| E\rangle \in \mathcal{B}_i$ by strong activation (Lemma 11) and the inductive hypothesis. Therefore, $\mathsf{K}^{\vec{D}}(\vec{e},\vec{v}) \in Head(\mathcal{A})$.
  - e_i is the right-most non-co-value in $\vec{e} = \vec{e'}, e_i, \vec{E'}$: Observe that for any $E \in \mathcal{A}$, we have the head lifting reduction

    $$\langle \mathsf{K}^{\vec{D}}(\vec{e},\vec{v}) \| E\rangle \mapsto \langle \mu\alpha.\langle \mathsf{K}^{\vec{D}}(\vec{E'},\alpha,\vec{e'},\vec{v}) \| E\rangle \| e_i\rangle \in \perp\!\!\!\perp$$

    because $\mu\alpha.\langle \mathsf{K}^{\vec{D}}(\vec{E'},\alpha,\vec{e'},\vec{v}) \| E\rangle$ by strong activation (Lemma 11) and the inductive hypothesis. Therefore, we have $\mathsf{K}^{\vec{D}}(\vec{e},\vec{v}) \in Head(\mathcal{A})$.
- Analogous to the previous statement by duality. □

We now show how to generate a full-fledged reducibility candidate from any simple core definition for a type using a variant of the symmetric candidates technique of Barbanera and Berardi [1]. We define the saturation function $Sat(\mathcal{A})$, which determines all the (co-)terms that are strongly normalizing with all the (co-)values of $\mathcal{A}$ either now, or one step by head reduction in the future:

$$\begin{aligned} v \in Sat(\mathcal{A}) &\iff v \in \mathcal{W} \\ &\qquad \wedge \forall E \in \mathcal{A}. \langle v \| E \rangle \in \perp\!\!\!\perp \vee \langle v \| E \rangle \mapsto_{+} c \in \perp\!\!\!\perp \\ e \in Sat(\mathcal{A}) &\iff e \in \mathcal{W} \\ &\qquad \wedge \forall V \in \mathcal{A}. \langle V \| e \rangle \in \perp\!\!\!\perp \vee \langle V \| e \rangle \mapsto_{-} c \in \perp\!\!\!\perp \end{aligned}$$

So starting with a seed $\mathcal{C}$, we can use $Sat(-)$, to grow a full-fledged reducibility candidate by iteratively saturating it, one $Step$ at a time:

$$Step_{\mathcal{C}}(\mathcal{A}) = \mathcal{C} \sqcup Sat(\mathcal{A})$$

This process eventually finishes, due to the fact that preserves the subtyping order of the lattice of pre-types (Lemma 3).

Corollary 3.

$$\mathcal{A} \leq \mathcal{B} \implies Step_{\mathcal{C}}(\mathcal{A}) \leq Step_{\mathcal{C}}(\mathcal{B})$$

Proof. By monotonicity (Lemma 5) because $Step_{\mathcal{C}}$ is a negation operation inside $\mathcal{W}$, for any pre-type $\mathcal{C}$. □

Corollary 4. *For any $\mathcal{C}$, there is a fixed point $\mathcal{A} = Step_{\mathcal{C}}(\mathcal{A})$.*

The fact that the fixed point to the $Step_{\mathcal{C}}$ function exists lets us define a function that saturates any pre-type $\mathcal{C}$.

Corollary 5. *There exists a function on pre-types $\mathcal{R}(-)$ such that $\mathcal{R}(\mathcal{C}) = Step_{\mathcal{C}}(\mathcal{R}(\mathcal{C}))$.*

Proof. By Corollary 4 there is a fixed to the function $Step_{\mathcal{C}}$, so we take $\mathcal{R}(\mathcal{C})$ to be the least one. □

Note that this $\mathcal{R}(-)$ operation gives us something that is *almost* a reducibility candidate. For any $Sat(\mathcal{A}) \sqsubseteq \mathcal{A} \sqsubseteq \mathcal{W}$ we have that:

$$Head(\mathcal{A}) \sqsubseteq Sat(\mathcal{A}) \sqsubseteq \mathcal{A}$$
$$\mathcal{A}^{\perp} \sqsubseteq Val(\mathcal{A})^{\perp} \sqsubseteq Sat(\mathcal{A}) \sqsubseteq \mathcal{A}$$

So all that is remaining is to show that, for certain well-behaved choices for $\mathcal{C}$, $\mathcal{R}(\mathcal{C}) \sqsubseteq \mathcal{R}(\mathcal{C})^{\perp}$. In particular, we will find that $\mathcal{R}(\mathcal{C})$ is guaranteed to be orthogonally sound, and thus a reducibility candidate, whenever $\mathcal{C}$ is in $TypeCore$.

Definition 2. *The set $TypeCore$ consists of pre-types $\mathcal{C}$ such that $\mathcal{C} \sqsubseteq \mathcal{C}^{\perp s}$ and $\mathcal{C}$ is forward closed.*

Note all pre-types $\mathcal{C}$ in $TypeCore$ are orthogonally sound because $\mathcal{C} \sqsubseteq \mathcal{C}^{\perp s} \sqsubseteq \mathcal{C}^{\perp}$. Furthermore, since $(-)^{\perp s}$ is itself a negation operation in $Simp(\mathcal{W})$, $\mathcal{C} \sqsubseteq \mathcal{C}^{\perp s} \sqsubseteq Simp(\mathcal{W})$ so it contains only simple (co-)terms, which must be (co-)values because the chosen strategy $\mathcal{S}$ is assumed to be focalizing.

Lemma 13 (Head orthogonality)**.** *Suppose that $\mathcal{A} \sqsubseteq \mathcal{W}$ is forward closed and for all $v, e \in \mathcal{A}$, $\langle v \| e \rangle \mapsto_{+,0,-} c$ implies $c \in \perp\!\!\!\perp$. Then $\mathcal{A} \sqsubseteq \mathcal{A}^{\perp}$.*

Proof. Note that a command c is in $\perp\!\!\!\perp$, meaning it is strongly normalizing, if and only if all reducts of c are in $\perp\!\!\!\perp$. Since $\mathcal{A} \sqsubseteq \mathcal{W}$, every (co-)term in $\mathcal{A}$ is strongly normalizing, so let $|v|$ and $|e|$ be the lengths of the longest reduction sequence from any $v, e \in \mathcal{A}$, respectively. We now proceed to show that for any $v, e \in \mathcal{A}$, $\langle v \| e \rangle \in \perp\!\!\!\perp$ because all of its reducts are in $\perp\!\!\!\perp$, by induction on $|v| + |e|$.

- $\langle v \| e \rangle \mapsto_{+,0,-} c$: we know $c \in \perp\!\!\!\perp$ by assumption.
- $\langle v \| e \rangle \to \langle v' \| e \rangle$ because $v \to v'$: then $v' \in \mathcal{A}$ because $\mathcal{A}$ is forward closed, and $|v'| < |v|$. Furthermore, we have that all head reducts of $\langle v' \| e \rangle$ are in $\perp\!\!\!\perp$ by Lemma 10 and the fact that $\perp\!\!\!\perp$ is closed under reduction. Therefore, $\langle v' \| e \rangle \in \perp\!\!\!\perp$ by the inductive hypothesis.
- $\langle v \| e \rangle \to \langle v \| e' \rangle$ because $e \to e'$: analogous to the previous case by duality. □

Lemma 14. *If $v, e \in \mathcal{W}$ and $\langle v \| e \rangle \mapsto_0 c \in runs$, then $\langle v \| e \rangle \in \perp\!\!\!\perp$.*

Proof. Consider the pre-type $\mathcal{A}$ defined as

$$v' \in \mathcal{A} \iff v \twoheadrightarrow v'$$
$$e' \in \mathcal{A} \iff e \twoheadrightarrow e'$$

Note that v and e are in $\mathcal{A}$. Further, given any $v', e' \in \mathcal{A}$, $\langle v' \| e' \rangle \mapsto_0 c'$ such that $c \twoheadrightarrow c'$ by Lemma 10 and induction on their reduction paths, so that $c' \in \perp\!\!\!\perp$ because $\perp\!\!\!\perp$ is closed under reduction. Observe that $\mathcal{A} \sqsubseteq \mathcal{W}$ and is forward closed by definition. Finally, because v and e are simple, and the set of simple (co-)terms is forward closed, $\mathcal{A}$ is simple. From this, we know that for all $v', e' \in \mathcal{A}$, $\langle v' \| e' \rangle \mapsto_{0,+,-} c'$ implies that $c' \in \perp\!\!\!\perp$. Thus, $\mathcal{A} \sqsubseteq \mathcal{A}^{\perp}$ by Lemma 13 and so $\langle v \| e \rangle \in \perp\!\!\!\perp$. □

Lemma 15. *If $\mathcal{C} \in TypeCore$ and $\mathcal{A} = Step_{\mathcal{C}}(\mathcal{A})$, then for all $v, e \in \mathcal{A}$, $\langle v \| e \rangle \mapsto_{+,0,-} c$ implies $c \in \perp\!\!\!\perp$.*

Proof. Let $v, e \in \mathcal{A} = \mathcal{C} \sqcup Sat(\mathcal{A})$. By cases, we have:

- $V, E \in \mathcal{C}$: $V \perp\!\!\!\perp E$ by the assumption that $\mathcal{C} \in TypeCore$, so $\mathcal{C}$ is orthogonally sound.
- $V \in \mathcal{C}$, $e \in Sat(\mathcal{A})$: by $e \in Sat(\mathcal{A})$ and the fact that V is a value, we know that either $\langle V \| e \rangle \in \perp\!\!\!\perp$ or $\langle V \| e \rangle \mapsto_{-} c \in \perp\!\!\!\perp$ for some c. In the first case, every reduct of $\langle V \| e \rangle$ is in $\perp\!\!\!\perp$, including any head reductions, since $\perp\!\!\!\perp$ is forward closed. In the second case, we know V is simple by the assumption that $\mathcal{C} \in TypeCore$ and e is non-simple so it cannot take part of a neutral head reduction. This means $\langle V \| e \rangle \not\mapsto_{+/0}$ and $\langle V \| e \rangle \mapsto_{-} c'$ implies $c' \in \perp\!\!\!\perp$ because $\mapsto_{-}$ is deterministic so $c' = c \in \perp\!\!\!\perp$. Therefore, $\langle V \| e \rangle \mapsto_{+,0,-} c$ implies $c \in \perp\!\!\!\perp$.
- $v \in Sat(\mathcal{A})$, $E \in \mathcal{C}$: analogous to the previous case by duality.
- $v, e \in Sat(\mathcal{A})$: Let us consider the possible head reductions:
  - $\langle v \| e \rangle \mapsto_0 c$: in this case, both v and e are simple (co-)values, and neither can take a positively charged head reduction step. Therefore, for $v, e \in Sat(\mathcal{A})$ to hold, it must be that $\langle v \| e \rangle \in \perp\!\!\!\perp$, and so $c \in \perp\!\!\!\perp$ because $\perp\!\!\!\perp$ is closed under reduction.
  - $\langle v \| e \rangle \mapsto_{+} c$: in this case, e must be a co-value, so for $v \in Sat(\mathcal{A})$ to hold, either $\langle v \| e \rangle \in \perp\!\!\!\perp$ already or there is a c' such that $\langle v \| e \rangle \mapsto_{+} c'$. In the former case, $c \in \perp\!\!\!\perp$ because $\perp\!\!\!\perp$ is closed under reduction, and in the latter case, $c = c' \in \perp\!\!\!\perp$ because $\mapsto_{+}$ is deterministic.
  - $\langle v \| e \rangle \mapsto_{-} c$: analogous to the previous case by duality. □

Lemma 16. *$Sat(\mathcal{A})$ is forward closed.*

Proof.
- Suppose $v \in Sat(\mathcal{A})$ and $v \to v'$. Let $E \in \mathcal{A}$, so that $v \in Sat(\mathcal{A})$ implies that either $v \perp\!\!\!\perp E$ or $\langle v \| E \rangle \mapsto_{+} c$.
  In the first case we know that $\langle v \| E \rangle \in \perp\!\!\!\perp$, so $\langle v \| E \rangle \to \langle v' \| E \rangle$ and $\langle v' \| E \rangle \in \perp\!\!\!\perp$ since $\perp\!\!\!\perp$ is forward closed.
  In the second case we know that $\langle v' \| E \rangle \leftarrow \langle v \| E \rangle \mapsto_{+} c \in \perp\!\!\!\perp$. Based on the commutation of internal and head reduction (Lemma 10), and the fact that $\perp\!\!\!\perp$ is forward closed, we have one of two cases:
  - $c \twoheadrightarrow \langle v' \| E \rangle \in \perp\!\!\!\perp$, or
  - $\langle v' \| E \rangle \mapsto_{+} c' \twoheadleftarrow c \in \perp\!\!\!\perp$

  Therefore, since in any case $\langle v' \| E \rangle \in \perp\!\!\!\perp$ or $\langle v' \| E \rangle \mapsto_{+} c' \in \perp\!\!\!\perp$, then $v' \in Sat(\mathcal{A})$.
- Suppose $e \in Sat(\mathcal{A})$ and $e \to e'$. Analogous to the previous case by duality. □

Corollary 6. *Given $\mathcal{A} = Step_{\mathcal{C}}(\mathcal{A})$, $\mathcal{A}$ is forward closed when $\mathcal{C} \in TypeCore$.*

Proof. Any $v, e \in \mathcal{A}$ is in either $\mathcal{C}$ or $Sat(\mathcal{A})$, so it follows by Lemma 16 and definition of $TypeCore$. □

Corollary 7. *For any $\mathcal{C} \in TypeCore$, $\mathcal{R}(\mathcal{C})$ is a reducibility candidate such that $\mathcal{C} \sqsubseteq \mathcal{R}(\mathcal{C})$.*

Proof. By Lemma 5, $\mathcal{R}(\mathcal{C}) = Step_{\mathcal{C}}(\mathcal{R}(\mathcal{C}))$, so by definition we know that $\mathcal{C} \sqsubseteq \mathcal{R}(\mathcal{C})$. Furthermore, we know that both $Head(\mathcal{R}(C))$ and $\mathcal{R}(\mathcal{C})^{\perp}$ refine $Sat(\mathcal{R}(\mathcal{C}))$, which in turn refines $\mathcal{R}(\mathcal{C})$. Furthermore, by definition of $TypeCore$ and Sat, we know that $\mathcal{R}(\mathcal{C}) \sqsubseteq \mathcal{W}$. Finally, by Corollary 6 and Lemma 15 we know that $\mathcal{R}(\mathcal{C})$ is forward closed and all head reducts are in $\perp\!\!\!\perp$, so $\mathcal{R}(\mathcal{C}) \sqsubseteq \mathcal{R}(\mathcal{C})^{\perp}$ by Lemma 13. Therefore, $\mathcal{R}(\mathcal{C})$ is a reducibility candidate. □

E. The Model

We use the notation $\mathbb{O}$ to denote a set of ordinals that is sufficiently large for our purposes: $\mathbb{O}$ contains all ordinals less than or equal to $\omega \times 2$. However, any larger set of ordinals would serve just as well.

Definition 3. *PK, or* pre-kinds *is the smallest set such that*

1. *$\mathbb{O} \in PK$,*
2. *$CR \in PK$, and*
3. *$\forall a, b \in PK, \{f : a \rightharpoonup b\} \in PK$.*

The universe $\mathcal{U}$ is defined as the union of PK

$$\mathcal{U} = \bigcup_{a \in PK} a$$

Definition 4. *A* semantic kind *$\kappa \in SKind$ is a pair (S, R) where $S \in \mathcal{P}(\mathcal{U})$ and $R \in \mathcal{P}(Type \times S)$. We say that S is the* domain *of the semantic kind and R is the* syntactic-semantic relation*.*

In order to define the meaning of types as a generic fold parameterized by an interpretation operation, we define an intermediate stage between syntactic and semantic types called hybrid types.

Definition 5. *The set of* hybrid types *($HType$) consists of*

1. *type variables,*
2. *$\mathsf{F}(\vec{\mathcal{A}})$ where $\overrightarrow{\mathcal{A} \in \mathcal{U}}$ and $\mathsf{F} \in \mathcal{F}$,*
3. *the constants 0 and ∞,*
4. *$+1(\mathcal{M})$ where $\mathcal{M} \in \mathcal{U}$.*

We define the interpretation functions on types and kinds using the same notation $[\![-]\!]$, which is disambiguated by context. Intuitively, this interpretation is parameterized by an operation, $\mathcal{H}$, which does the real hard work of assigning a semantic meaning to syntactic types. The purpose of abstracting out $\mathcal{H}$ is to break up the many recursions involved with interpreting recursive (co-)data types, so that the outermost interpretation is defined structurally over just the syntax of types, without concern on how the different types are related to one another.

$$[\![k \in Kind]\!] : (HType \rightharpoonup \mathcal{U}) \rightharpoonup SKind$$
$$[\![A \in Type]\!] : (HType \rightharpoonup \mathcal{U}) \rightharpoonup \mathcal{U}$$

The interpretation of types and kinds is given by structural induction over the syntax. Technically, we consider the interpretation giving the two components of semantic kinds, the domain and syntactic-semantic relation, as two separate functions. The interpretation of types and and the domain of kinds are defined by mutual induction over the structure of syntactic types and kinds, and the interpretation of the syntactic-semantic relation of kinds is defined by structural induction on kinds, and using the previous definition.

$$\begin{aligned}
[\![a]\!]_{\mathcal{H}} &\triangleq \mathcal{H}(a) \\
[\![0]\!]_{\mathcal{H}} &\triangleq \mathcal{H}(0) \\
[\![\infty]\!]_{\mathcal{H}} &\triangleq \mathcal{H}(\infty) \\
[\![N+1]\!]_{\mathcal{H}} &\triangleq \mathcal{H}(+1([\![N]\!]_{\mathcal{H}})) \\
[\![\mathsf{F}(\vec{A})]\!]_{\mathcal{H}} &\triangleq \mathcal{H}(\mathsf{F}(\overrightarrow{[\![A]\!]_{\mathcal{H}}})) \\
[\![\lambda a : k.B]\!]_{\mathcal{H}} &\triangleq \lambda \mathcal{A} \in \pi_1([\![k]\!]_{\mathcal{H}}).[\![B]\!]_{\mathcal{H}\{\mathcal{A}/a\}} \\
[\![A\ B]\!]_{\mathcal{H}} &\triangleq [\![A]\!]_{\mathcal{H}}([\![B]\!]_{\mathcal{H}})
\end{aligned}$$

$$\begin{aligned}
\pi_1([\![\star]\!]_{\mathcal{H}}) &\triangleq CR \\
\pi_1([\![k_1 \to k_2]\!]_{\mathcal{H}}) &\triangleq \pi_1([\![k_2]\!]_{\mathcal{H}})^{\pi_1([\![k_1]\!]_{\mathcal{H}})} \\
\pi_1([\![\mathsf{Ix}]\!]_{\mathcal{H}}) &\triangleq \{\mathcal{N} \in \mathbb{O} | \exists N \in Type, N \sim_{\mathcal{H}} \mathcal{N}\} \\
\pi_1([\![\mathsf{Ord}]\!]_{\mathcal{H}}) &\triangleq \mathbb{O} \\
\pi_1([\![< N]\!]_{\mathcal{H}}) &\triangleq \{\mathcal{M} \in \mathbb{O} \\
&\qquad | \exists N' \in Type. N \twoheadrightarrow_{\beta} N' \wedge \mathcal{M} < [\![N']\!]_{\mathcal{H}}\}
\end{aligned}$$

$$\begin{aligned}
\pi_2([\![\star]\!]_{\mathcal{H}}) &\triangleq Type \times CR \\
\pi_2([\![k_1 \to k_2]\!]_{\mathcal{H}}) &\triangleq \{(A, \mathcal{A}) | \forall (B, \mathcal{B}) \in \pi_2([\![k_1]\!]_{\mathcal{H}}). \\
&\qquad (A\ B, \mathcal{A}(\mathcal{B})) \in \pi_2([\![k_2]\!]_{\mathcal{H}})\} \\
\pi_2([\![\mathsf{Ix}]\!]_{\mathcal{H}}) &\triangleq \{(M, \mathcal{M}) | M \sim_{\mathcal{H}} \mathcal{M}\} \\
\pi_2([\![\mathsf{Ord}]\!]_{\mathcal{H}}) &\triangleq \{(M, \mathcal{M}) | \exists M' \in Type. M \twoheadrightarrow_{\beta} M' \\
&\qquad \wedge [\![M']\!]_{\mathcal{H}} \leq \mathcal{M}\} \\
\pi_2([\![< N]\!]_{\mathcal{H}}) &\triangleq \{(M, \mathcal{M}) | \exists M' \in Type. \\
&\qquad M \twoheadrightarrow_{\beta} M' \\
&\qquad \wedge [\![M']\!]_{\mathcal{H}} \leq \mathcal{M} < [\![N]\!]_{\mathcal{H}}\}
\end{aligned}$$

Note that the relation $\sim_{\mathcal{H}}$ used in $[\![\mathsf{Ix}]\!]_{\mathcal{H}}$ is defined as the smallest subset of Type $\times\ \mathbb{O}$ such that

1. $0 \sim_{\mathcal{H}} \mathcal{H}(0)$ when $\mathcal{H}(0) \in \mathbb{O}$,
2. for all $N \sim_{\mathcal{H}} \mathcal{N}$, $N + 1 \sim_{\mathcal{H}} \mathcal{H}(+1(\mathcal{N}))$ when $\mathcal{H}(+1(\mathcal{N})) \in \mathbb{O}$.

The application $[\![A]\!]_{\mathcal{H}}([\![B]\!]_{\mathcal{H}})$ is defined whenever there exists $\kappa_1, \kappa_2 \in PK$ such that $[\![A]\!]_{\mathcal{H}} \in \kappa_1 \rightharpoonup \kappa_2$ and $[\![B]\!]_{\mathcal{H}} \in \kappa_1$, otherwise it is undefined. We will use the shorthand notation $\mathcal{A} \in [\![k]\!]_{\mathcal{H}}$ when k is a kind to indicate that $\mathcal{A}$ is an element of $\pi_1([\![k]\!]_{\mathcal{H}})$, and $A\ [\![k]\!]_{\mathcal{H}}\ \mathcal{A}$ when $(A, \mathcal{A})$ is an element of $\pi_2([\![k]\!]_{\mathcal{H}})$.

Also note that we give an interpretation of sorts, as well. The sort of non-erasable kinds, $\blacksquare$, is interpreted as the whole set of all semantic kinds, and the sort of erasable kinds, $\square$, adds the restriction that the syntactic-semantic relation is backwards closed under β reduction.

$$\begin{aligned}
[\![\blacksquare]\!] &\triangleq SKind \\
[\![\square]\!] &\triangleq \{\kappa \in SKind \\
&\qquad | \forall (A, \mathcal{A}) \in \pi_2(\kappa). A' \twoheadrightarrow_{\beta} A \implies (A', \mathcal{A}) \in \pi_2(\kappa)\}
\end{aligned}$$

The main work of defining types is in giving the core interpretation of type constructors in $\mathcal{F}$:

$$\langle\!\langle A \in HType \rangle\!\rangle : (HType \rightharpoonup \mathcal{U}) \rightharpoonup TypeCore$$

First, we specify when an interpretation operation $\mathcal{H}$ assigns a plausible meaning to the numeric measures used in types.

Definition 6. *A map $\mathcal{H} : HType \rightharpoonup \mathcal{U}$ is plausible if*

1. $\mathcal{H}(0) \in \mathbb{O}$,
2. $\mathcal{H}(\infty) \in \mathbb{O}$,
3. for all $\mathcal{N} \in \mathbb{O}$, $\mathcal{H}(+1(\mathcal{N})) \in \mathbb{O}$,
4. $\mathcal{H}(0) < \mathcal{H}(\infty)$,
5. for all $\mathcal{N} \in \mathbb{O}$, $\mathcal{N} < \mathcal{H}(\infty)$ implies $\mathcal{H}(+1(\mathcal{N})) < \mathcal{H}(\infty)$, and
6. for all $\mathcal{M}, \mathcal{N} \in \mathbb{O}$, $\mathcal{M} < \mathcal{N}$ implies $\mathcal{M} < \mathcal{H}(+1(\mathcal{M})) \leq \mathcal{N}$.

We assume that $\mathcal{F}$ comes with some ordering among hybrid types that is both well-founded but also enough that the core interpretation of types is well-defined with respect to $\langle\!\langle - \rangle\!\rangle$. Note that we say two maps $\mathcal{H}, \mathcal{I} : HType \rightharpoonup \mathcal{U}$ are equivalent up to a hybrid type h (written $\mathcal{H} \equiv_{<h} \mathcal{I}$) if and only if for all $g < h \in HType$, $\mathcal{H}(g) = \mathcal{I}(g)$ and are defined.

Definition 7. *The set of type constructors $\mathcal{F}$ is well-founded if there is a partial well order $<$ on the set of hybrid types formable from $\mathcal{F}$ such that for any plausible $\mathcal{H}$:*

1. *Given any $\mathsf{F}(\overrightarrow{a : k_1}) : k_2 \in \mathcal{F}$ and $\overrightarrow{\mathcal{A} \in [\![k_1]\!]_{\mathcal{H}}}$, $\langle\!\langle \mathsf{F}(\vec{\mathcal{A}}) \rangle\!\rangle_{\mathcal{H}}$ is defined and in $[\![k_2]\!]_{\mathcal{H}}$ whenever for all $\mathsf{G}(\overrightarrow{b : k_3}) : k_4 \in \mathcal{F}$ and $\overrightarrow{\mathcal{B} \in [\![k_3]\!]_{\mathcal{H}}}$ such that $\mathsf{G}(\vec{\mathcal{B}}) < \mathsf{F}(\vec{\mathcal{A}})$, $\mathcal{H}(\mathsf{G}(\vec{\mathcal{B}}))$ is defined and in $[\![k_4]\!]_{\mathcal{H}}$.*
2. *For any other plausible $\mathcal{I}$ and for any $h \in HType$ such that $\mathcal{H} \equiv_{<h} \mathcal{I}$, $\langle\!\langle h \rangle\!\rangle_{\mathcal{H}} = \langle\!\langle h \rangle\!\rangle_{\mathcal{I}}$.*

Definition 8. *The set $[\![\mathcal{F}]\!]$ consists of partial functions $HType \rightharpoonup \mathcal{U}$ such that if $\mathcal{H} \in [\![\mathcal{F}]\!]$ then*

1. *for any $\mathsf{F}(\overrightarrow{a : k}) : k' \in \mathcal{F}$ and any $\overrightarrow{\mathcal{A} \in [\![k]\!]_{\mathcal{H}}}$, $\langle\!\langle \mathsf{F}(\vec{\mathcal{A}}) \rangle\!\rangle_{\mathcal{H}} \sqsubseteq \mathcal{H}(\mathsf{F}(\vec{\mathcal{A}})) \in [\![k']\!]_{\mathcal{H}}$, and*
2. *$\mathcal{H}$ is plausible.*

Lemma 17. *Given any well-founded set of type constructors $\mathcal{F}$ and $\mathcal{I} : HType \rightharpoonup \mathcal{U}$ there exists a $\mathcal{H} \in [\![\mathcal{F}]\!]$ such that for any $h = 0, \infty, +1(\mathcal{M}), a$ we have $\mathcal{H}(h) = \mathcal{I}(h)$.*

Proof. We define the function:

$$next : (HType \rightharpoonup \mathcal{U}) \rightarrow HType \rightharpoonup \mathcal{U}$$

$$next(\mathcal{H})(h) = \begin{cases} \mathcal{R}(\langle\!\langle \mathsf{F}(\vec{\mathcal{A}}) \rangle\!\rangle_{\mathcal{H}}) & \text{if } h = \mathsf{F}(\vec{\mathcal{A}}) \text{ and } \mathsf{F} \in \mathcal{F} \\ \mathcal{I}(h) & \text{otherwise} \end{cases}$$

The conditions we require for well-foundedness of $\mathcal{F}$ are exactly what we need to take a fixed point of this function and show that it is in $[\![\mathcal{F}]\!]$.

Now, by the second property of Definition 7 above, we know that for all h and $\mathcal{H} \equiv_{<h} \mathcal{H}'$, $next(\mathcal{H}) \equiv_{<h} next(\mathcal{H}')$ and so $next(\mathcal{H})(h) = next(\mathcal{H}')(h)$ as well. Using this fact, we show that $next$ has a (unique) fixed point. That is, we wish to find the $\mathcal{H}_\Box$ such that $next(\mathcal{H}_\Box) = \mathcal{H}_\Box$. The proof proceeds by induction on h by the well-founded order given for $\mathcal{F}$. For the inductive hypothesis, we get that there exists a map $\mathcal{H}'_\Box$ such that for all $g < h$, $\mathcal{H}'_\Box(g) = next(\mathcal{H}'_\Box)(g)$. In other words, $\mathcal{H}'_\Box \equiv_{<h} next(\mathcal{H}'_\Box)$ and so $next(\mathcal{H}'_\Box)(h) = next(next(\mathcal{H}'_\Box))(h)$. Thus, $\mathcal{H}_\Box = next(\mathcal{H}'_\Box)$ and $\mathcal{H}_\Box(h)$ is the unique possible value for h. That is, we know that any other $\mathcal{H}'' \equiv_{<h} \mathcal{H}_\Box$ such that $\mathcal{H}''(h) = next(\mathcal{H}'_\Box)(h)$ has the property that $\mathcal{H}'_\Box \equiv_{<h} \mathcal{H}'' \equiv_{<h} next(\mathcal{H}'_\Box) \equiv_{<h} \mathcal{H}_\Box$ and $next(\mathcal{H}'')(h) = next(\mathcal{H}_\Box)(h) = \mathcal{H}_\Box(h)$.

Thus, there is a unique $\mathcal{H}_\Box$ such that $next(\mathcal{H}_\Box) = \mathcal{H}_\Box$. We need only show that this $\mathcal{H}_\Box \in [\![\mathcal{F}]\!]$. That, is for any $\mathsf{F}(\overrightarrow{a : k}) : k \in \mathcal{F}$ and $\overrightarrow{\mathcal{A} \in [\![k]\!]_{\mathcal{H}_\Box}}$, $\langle\!\langle \mathsf{F}(\vec{\mathcal{A}}) \rangle\!\rangle_{\mathcal{H}_\Box} \sqsubseteq \mathcal{H}_\Box(\mathsf{F}(\vec{\mathcal{A}})) \in [\![k']\!]_{\mathcal{H}_\Box}$. Again, we proceed by well founded induction on the set of hybrid types, and what we need to show for the inductive step is exactly the definition of $next$, the fact that $\mathcal{C} \sqsubseteq \mathcal{R}(\mathcal{C})$, and the first property of Definition 7 above. □

We interpret the kinding environment $[\![\Theta]\!]_{\mathcal{H}}$ as a pair of $S \subset HType \rightharpoonup \mathcal{U}$ and a relation on substitutions and S

$$\begin{aligned}
[\![-]\!]_{\mathcal{H}} &= (\mathcal{H}, \{(\theta, \mathcal{H})\}) \\
[\![\Theta, a : k]\!]_{\mathcal{H}} &= (\{\mathcal{I} \mid \exists \mathcal{I}' \in \pi_1([\![\Theta]\!]_{\mathcal{H}}). \\
&\qquad \mathcal{I}(a) \in [\![k]\!]_{\mathcal{I}'} \\
&\qquad \wedge \forall h \in HType, h \neq a \Rightarrow \mathcal{I}(h) = \mathcal{I}'(h)\}. \\
&\quad \{(\theta, \mathcal{I}) \mid \exists (\theta', \mathcal{I}') \in \pi_2([\![\Theta]\!]_{\mathcal{H}}). \\
&\qquad \theta(a)\ [\![k]\!]_{\mathcal{I}}\ \mathcal{I} \wedge (\forall b \neq a. \theta(b) = \theta'(b)) \\
&\qquad \wedge (\forall h \in HType. h \neq a \Rightarrow \mathcal{I}(h) = \mathcal{I}'(h))\}
\end{aligned}$$

We use the notation $\mathcal{I} \in [\![\Theta]\!]_{\mathcal{H}}$ if $\mathcal{I} \in \pi_1([\![\Theta]\!]_{\mathcal{H}})$ and $\theta [\![\Theta]\!]_{\mathcal{H}} \mathcal{I}$ if $(\theta, \mathcal{I}) \in \pi_2([\![\Theta]\!]_{\mathcal{H}})$.

Lemma 18. *For any $\mathcal{I}$ in $[\![\Theta]\!]_{\mathcal{H}}$, $\mathcal{I}(h) = \mathcal{H}(h)$ whenever h is not a variable.*

Proof. By induction on Θ. □

Corollary 8. *For any $\mathcal{I} \in [\![\Theta]\!]_{\mathcal{H}}$, if $\mathcal{H} \in [\![\mathcal{F}]\!]$ then $\mathcal{I} \in [\![\mathcal{F}]\!]$.*

Lemma 19. *For any Θ, if $[\![\Theta]\!]_{\mathcal{H}}$ is defined and for every $a : k \in \Theta$ we know $k \neq \overrightarrow{k \rightarrow} < 0 \wedge k \neq \overrightarrow{k \rightarrow} < i$, then there exist θ and $\mathcal{I}$ such that $\theta\ [\![\Theta]\!]_{\mathcal{H}}\ \mathcal{I}$.*

Proof. All kinds except $\overrightarrow{k \rightarrow} < 0$ and $\overrightarrow{k \rightarrow} < i$ are inhabited. The kind $\star$ is inhabited by $(a, \mathcal{R}((\emptyset, \emptyset)))$, the kinds Ix, Ord and $< \infty$ are inhabited by $(0, \mathcal{H}(0))$, and $< N + 1$ is inhabited by $(N, [\![N]\!]_{\mathcal{H}})$. Function kinds are inhabited whenever their final result kind is, since we can always build the constant function. □

Lemma 20. *If θ is a substitution which only replaces type variables, and $c\{\theta\} \in \perp\!\!\!\perp$ then $c \in \perp\!\!\!\perp$.*

Proof. This follows from the fact that if $c \rightarrow c'$, then $c\{\theta\} \rightarrow c'\{\theta\}$, so strong normalization of $c\{\theta\}$ implies strong normalization of c. The main point is that filling in types for type variables can only allow for *more* reductions, either due to filling in the index of an $\mathsf{Inflate}$ or $\mathsf{Deflate}$ structure or by specifying that the upper bound for a quantified type variable in a case abstraction is inhabited. □

By contrast, we interpret $[\![\Gamma]\!]_{\mathcal{H}}$ and $[\![\Delta]\!]_{\mathcal{H}}$ as just sets of substitutions.

$$\begin{aligned}
\gamma \in [\![\Gamma]\!]_{\mathcal{H}} &\iff \forall x : A \in \Gamma. x\{\gamma\} \in Val([\![A]\!]_{\mathcal{H}}) \\
\delta \in [\![\Delta]\!]_{\mathcal{H}} &\iff \forall \alpha : A \in \Delta. \alpha\{\delta\} \in Val([\![A]\!]_{\mathcal{H}})
\end{aligned}$$

Note that Val is the function on pre-types that keeps only their (co-)values.

Lemma 21. 1. *If $[\![\Gamma]\!]_{\mathcal{H}}$ is defined then it is inhabited by the identity substitution.*
2. *If $[\![\Delta]\!]_{\mathcal{H}}$ is defined then it is inhabited by the identity substitution.*

Proof. If $[\![\Gamma]\!]_{\mathcal{H}}$ is defined then $\forall (x : A) \in \Gamma$, $[\![A]\!]_{\mathcal{H}}$ is defined and must yield a reducibility candidate since $Val([\![A]\!]_{\mathcal{H}})$ is defined. Since variables inhabit every reducibility candidate (Lemma 2), and are (co-)values (because we assume the chosen strategy is focalizing), the identity substitution inhabits $[\![\Gamma]\!]_{\mathcal{H}}$ (and by duality $[\![\Delta]\!]_{\mathcal{H}}$). □

Note that we denote the extension of an interpretation operation $\mathcal{H}$ with the interpretation $\mathcal{B}$ for a type variable b as $\mathcal{H}\{\mathcal{B}/b\}$, where

$$\mathcal{H}\{\mathcal{B}/b\}(h) \triangleq \begin{cases} \mathcal{B} & \text{if } h = b \\ \mathcal{H}(h) & \text{otherwise} \end{cases}$$

Lemma 22. *If* $[\![B]\!]_{\mathcal{H}}$ *is defined and* $[\![A]\!]_{\mathcal{H}\{[\![B]\!]_{\mathcal{H}}/b\}}$ *is defined, then* $[\![A]\!]_{\mathcal{H}\{[\![B]\!]_{\mathcal{H}}/b\}} = [\![A\{B/b\}]\!]_{\mathcal{H}}$.

Proof. By induction on the structure of A. □

F. Interpreting Specific Types

To define the core meaning following an orthogonality-based methodology, except that instead of attempting to build a reducibility candidate through the double-orthogonal closure, we build a simplified interpretation in $TypeCore$ using the $\perp s$ negation operation inside $Simp(\mathcal{W})$.

On the one hand, for the hybrid types of data type constructors, h, we build this core definition through some set of focalized constructions, $Cons_{\mathcal{H}}(h)$, from the $\perp s \perp s$ closure such that

$$v \in \langle\!\langle h \rangle\!\rangle_{\mathcal{H}} \iff v \in (Cons_{\mathcal{H}}(h), \emptyset)^{\perp s \perp s}$$
$$e \in \langle\!\langle h \rangle\!\rangle_{\mathcal{H}} \iff e \in (Cons_{\mathcal{H}}(h), \emptyset)^{\perp s}$$

On the other hand, for the hybrid types of co-data type constructors, g, we build this core definition through some set of focalized observations, $Obs_{\mathcal{H}}(h)$, from the $\perp s \perp s$ closure such that

$$v \in \langle\!\langle g \rangle\!\rangle_{\mathcal{H}} \iff v \in (\emptyset, Obs_{\mathcal{H}}(g))^{\perp s}$$
$$e \in \langle\!\langle g \rangle\!\rangle_{\mathcal{H}} \iff e \in (\emptyset, Obs_{\mathcal{H}}(g))^{\perp s \perp s}$$

Note that in both cases, $\langle\!\langle h \rangle\!\rangle_{\mathcal{H}} = \langle\!\langle h \rangle\!\rangle_{\mathcal{H}}{}^{\perp s}$. It follows that this definition is always in $TypeCore$. Furthermore by double negation introduction (Lemma 7), in the case of hybrid types for data type constructors h, we are ensured that $(Cons_{\mathcal{H}}(h), \emptyset) \sqsubseteq \langle\!\langle h \rangle\!\rangle_{\mathcal{H}}$ whenever $Cons_{\mathcal{H}}(h)$ is a set of strongly-normalizing simple values, and in the case of hybrid types for co-data type constructors g, $(\emptyset, Obs_{\mathcal{H}}(g)) \sqsubseteq \langle\!\langle g \rangle\!\rangle_{\mathcal{H}}$ whenever $Obs_{\mathcal{H}}(g)$ is a set of strongly-normalizing simple co-values.

To show that our set of type constructors, $\mathcal{F}$, is well-founded according to Definition 7, we suppose for simplicity that the declarations are given in a linear manner, so that declarations can only refer to the type constructors introduced by previous declarations, which is ensured by their checks for well-formedness. If a (co-)data type declaration refers to a type constructor that hasn't yet been declared, then the sequents for its constructors can't be well-formed by the rules we have available. This gives us the main basis for the well-founded ordering for $\mathcal{F}$: type constructors can only be greater than other type constructors earlier in the line. The rest of the relations among hybrid types of $\mathcal{F}$ are given by the possible recursive nature of the declaration itself.

F.1 Non-recursive (Co-)data Declarations

For any declared data type of the form

$$\mathbf{data}\, \mathsf{F}(\overrightarrow{a:k})\, \mathbf{where}$$
$$\overrightarrow{\mathsf{K} : \vec{B} \vdash_{\overrightarrow{d:k'}} \mathsf{F}(\vec{a}) | \vec{C}}$$

We require that for all $\mathsf{G}(\overrightarrow{d:k'}) : \star \in \mathcal{F}$, $\mathsf{G}(\vec{\mathcal{D}}) < \mathsf{F}(\mathcal{N}, \vec{\mathcal{A}})$. This allows us to give a *Cons*truction-oriented definition for $\langle\!\langle \mathsf{F}(\mathcal{N}, \vec{\mathcal{A}}) \rangle\!\rangle_{\mathcal{H}}$ such that the order of hybrid types satisfies Definition 7. In particular, $Cons_{\mathcal{H}}(\mathsf{F}(\mathcal{N}, \vec{\mathcal{A}}))$ is defined as:

$$\begin{aligned}
&Cons_{\mathcal{H}}(\mathsf{F}(\vec{\mathcal{A}}))\\
&\triangleq \{\mathsf{K}^{\overrightarrow{d:k'}}(\vec{\alpha}, \vec{x})\{\gamma, \delta, \theta\} |\\
&\quad \exists\, \mathsf{K} : \vec{B} \vdash_{\overrightarrow{d:k'}} \mathsf{F}(\vec{a}) | \vec{C} \in \overrightarrow{\mathsf{K} : \vec{B} \vdash_{\overrightarrow{d:k'}} \mathsf{F}(\vec{a}) | \vec{C}}.\\
&\quad \exists \theta\, [\![\overrightarrow{d:k'}]\!]_{\mathcal{H}\{\overrightarrow{\mathcal{A}/a}\}}\, \mathcal{I}. \exists \gamma \in [\![\overrightarrow{x:B}]\!]_{\mathcal{I}}, \delta \in [\![\overrightarrow{\alpha:C}]\!]_{\mathcal{I}}\}.
\end{aligned}$$

Lemma 23. *If* $\mathcal{F}$ *is well-founded,* $\mathcal{H} \in [\![\mathcal{F}]\!]$ *and the declaration of* F *is well-formed with respect to* $\mathcal{F}$*, then* $Cons_{\mathcal{H}}(\mathsf{F}(\vec{\mathcal{A}}))$ *is a defined pre-type for all* $\overrightarrow{\mathcal{A} \in [\![k]\!]_{\mathcal{H}}}$.

Proof. Because the declaration of F is well-formed, we know that the sequent for each of its constructors, $\mathsf{K} : \vec{B} \vdash_{\overrightarrow{d:k'}} \mathsf{F}(\vec{a}) | \vec{C}$, is also well-formed, $(\overrightarrow{x:B} \vdash_{\overrightarrow{a:k}, \overrightarrow{d:k'}} \overrightarrow{\alpha:C})\, \mathbf{seq}$. By the soundness of kinding rules (shown later in Theorem 35), we therefore know that each sequent is soundly represented in the model, $[\![(\overrightarrow{x:B} \vdash_{\overrightarrow{a:k}, \overrightarrow{d:k'}} \overrightarrow{\alpha:C})\, \mathbf{seq}\,]\!]_{\mathcal{H}}$. Therefore, $Cons_{\mathcal{H}}(\mathsf{F}(\vec{\mathcal{A}}))$ is a defined pre-type. □

As a corollary, we have that if $Cons_{\mathcal{H}}(\mathsf{F}(\vec{\mathcal{A}}))$ is defined then $\langle\!\langle \mathsf{F}(\vec{\mathcal{A}}) \rangle\!\rangle_{\mathcal{H}}$ is in $TypeCore$.

Lemma 24. *If* $Cons_{\mathcal{H}}(\mathsf{F}(\vec{\mathcal{A}}))$ *is defined, then it is a set of simple values in* $\mathcal{W}$.

Proof. Given any construction $v = \mathsf{K}^{\overrightarrow{d:k}}(\vec{\alpha}, \vec{x})\{\theta, \gamma, \delta\} \in Cons_{\mathcal{H}}(\mathsf{F}(\vec{\mathcal{A}}))$, we know that the substitutions γ and δ only ever replace the (co-)variables $\vec{\alpha}$ and $\vec{x}$ with strongly-normalizing (co-)values, since the substitutions only range over the (co-)values in reducibility candidates when defined (and all reducibility candidates refine $\mathcal{W}$). This means that v is simple, since it cannot take part in a positive head reduction (lifting never applies), and is therefore also a value because we assume our chosen strategy is focalizing. Furthermore, $v \in \mathcal{W}$ because all of its sub-(co-)-terms are strongly normalizing, and no other reductions apply to v. □

As a corollary, we have that if $Cons_{\mathcal{H}}(\mathsf{F}(\vec{\mathcal{A}}))$ is defined and contains a term v, then $v \in \langle\!\langle \mathsf{F}(\vec{\mathcal{A}}) \rangle\!\rangle_{\mathcal{H}}$.

Lemma 25. *If* $Cons_{\mathcal{H}}(\mathsf{F}(\vec{\mathcal{A}}))$ *is defined and* E *is the case abstraction* $\tilde{\mu}[\overrightarrow{\mathsf{K}^{\overrightarrow{d:k'}}(\vec{\alpha}, \vec{x}).c}]$ *such that*

$$\begin{aligned}
&\forall\, \mathsf{K} : \vec{B} \vdash_{\overrightarrow{d:k'}} \mathsf{F}(\vec{a}) | \vec{C} \in \overrightarrow{\mathsf{K} : \vec{B} \vdash_{\overrightarrow{d:k'}} \mathsf{F}(\vec{a}) | \vec{C}}.\\
&\exists \mathsf{K}^{\overrightarrow{d:k'}}(\vec{\alpha}, \vec{x}).c \in \overrightarrow{\mathsf{K}^{\overrightarrow{d:k'}}(\vec{\alpha}, \vec{x}).c}.\\
&\forall \theta\, [\![\overrightarrow{d:k'}]\!]_{\mathcal{H}\{\overrightarrow{\mathcal{A}/a}\}}\, \mathcal{I}. \forall \gamma \in [\![\overrightarrow{x:B}]\!]_{\mathcal{I}}, \delta \in [\![\overrightarrow{\alpha:C}]\!]_{\mathcal{I}}.\\
&\quad c\{\theta, \gamma, \delta\} \in \perp\!\!\!\perp
\end{aligned}$$

then $E \in \langle\!\langle \mathsf{F}(\vec{\mathcal{A}}) \rangle\!\rangle_{\mathcal{H}}$.

Proof. We must show that $E \in \mathcal{W}$ and $\langle V \| E \rangle \in \perp\!\!\!\perp$ for every $V \in Cons_{\mathcal{H}}(\mathsf{F}(\vec{\mathcal{A}}))$.

- $E \in \mathcal{W}$: for each sub-command c in E we have two cases, depending on whether or not any kind k' of the quantified type variables $\overrightarrow{d:k'}$ introduced by the pattern has the form $\overrightarrow{k''} \to < 0$ or $\overrightarrow{k''} \to < i$:
  - There is a k' such that $\overrightarrow{k''} \to < 0$ or $\overrightarrow{k''} \to < i$: then by the caveat on reduction inside a case abstraction, $c \not\mapsto$, so $c \in \perp\!\!\!\perp$.
  - There is no k' such that $\overrightarrow{k''} \to < 0$ or $\overrightarrow{k''} \to < i$: then by Lemma 19, there is a $\theta\, [\![\overrightarrow{d:k'}]\!]_{\mathcal{H}\{\overrightarrow{\mathcal{A}/a}\}}\, \mathcal{I}$, and by Lemma 21, the identity substitutions inhabit $[\![\overrightarrow{x:B}]\!]_{\mathcal{I}}$ and $[\![\overrightarrow{\alpha:C}]\!]_{\mathcal{I}}$. Therefore, we know that $c\{\theta\} \in \perp\!\!\!\perp$, and so $c \in \perp\!\!\!\perp$ by Lemma 20.

  In either case, all sub-commands of E are in $\perp\!\!\!\perp$, so $E \in \mathcal{W}$.
- $\langle V \| E \rangle \in \perp\!\!\!\perp$ for every $V \in Cons_{\mathcal{H}}(\mathsf{F}(\vec{\mathcal{A}}))$: note that for every such command, $\langle V \| E \rangle \mapsto_0 c$ and $c \in \perp\!\!\!\perp$ by definition of $Cons_{\mathcal{H}}(\mathsf{F}(\vec{\mathcal{A}}))$. Therefore, $\langle V \| E \rangle \in \perp\!\!\!\perp$ by Lemma 14. □

The case for non-recursive co-data type declarations follows analogously by duality.

F.2 (Co-)data Declarations by Noetherian Recursion

Declarations by noetherian recursion is largely analogous to the non-recursive declarations shown previously, and we highlight the difference here. For any declared data type of the form:

$$\mathbf{data}\,\mathsf{F}(i : \mathsf{Ord}, \overrightarrow{a : k})\ \mathbf{by}\ \text{noetherian recursion on } i\ \mathbf{where}$$
$$\overrightarrow{\mathsf{K} : \vec{B} \vdash_{\overrightarrow{d:k'}} \mathsf{F}(i, \vec{a}) | \vec{C}}$$

In addition to the normal ordering of hybrid types for non-recursive declarations, we also have $\mathsf{F}(\mathcal{N}, \vec{\mathcal{A}}) < \mathsf{F}(\mathcal{M}, \vec{\mathcal{B}})$ whenever $\mathcal{N} < \mathcal{M} \in \mathbb{O}$. This allows us to give a $Cons$truction-oriented definition for $\langle\!\langle \mathsf{F}(\mathcal{N}, \vec{\mathcal{A}}) \rangle\!\rangle_{\mathcal{H}}$ such that F can refer to itself for smaller choices of the ordinal index. In particular, $Cons_{\mathcal{H}}(\mathsf{F}(\mathcal{N}, \vec{\mathcal{A}}))$ is defined as:

$$\begin{aligned}
&Cons_{\mathcal{H}}(\mathsf{F}(\mathcal{N}, \vec{\mathcal{A}}))\\
&\triangleq \{\mathsf{K}^{\overrightarrow{d:k'}}(\vec{\alpha}, \vec{x})\{\gamma, \delta, \theta\} |\\
&\quad \exists\, \mathsf{K} : \vec{B} \vdash_{\overrightarrow{d:k'}} \mathsf{F}(i, \vec{a}) | \vec{C} \in \overrightarrow{\mathsf{K} : \vec{B} \vdash_{\overrightarrow{d:k'}} \mathsf{F}(i, \vec{a}) | \vec{C}}.\\
&\quad \exists \theta\ [\![\overrightarrow{d : k'}]\!]_{\mathcal{H}'}\ \mathcal{I}.\exists \gamma \in [\![\overrightarrow{x : B}]\!]_{\mathcal{I}}, \delta \in [\![\overrightarrow{\alpha : C}]\!]_{\mathcal{I}}\}.\\
&\mathbf{where}\ \mathcal{H}' = \mathcal{H}\{\mathcal{N}/i, \overrightarrow{\mathcal{A}/a}\}
\end{aligned}$$

Lemma 26. *If $\mathcal{F}$ is well-founded, $\mathcal{H} \in [\![\mathcal{F}]\!]$ and the declaration of F is well-formed with respect to $\mathcal{F}$, then $Cons_{\mathcal{H}}(\mathsf{F}(\mathcal{N}, \vec{\mathcal{A}}))$ is a defined pre-type for all $\mathcal{N} \in \mathbb{O}$ and $\overrightarrow{\mathcal{A} \in [\![k]\!]_{\mathcal{H}}}$.*

Proof. Because the well-formedness check for F can assume that F is already well-formed at smaller indices:

$$\frac{\Theta, i : \mathsf{Ord}, \Theta' \vdash M < i \qquad \overrightarrow{\Theta, i : \mathsf{Ord}, \Theta' \vdash A : k}}{\Theta, i : \mathsf{Ord}, \Theta' \vdash \mathsf{F}(M, \vec{A}) : \star}$$

we need to proceed by noetherian induction on the ordinal $\mathcal{N}$. Besides this difference, the proof follows analogously to Lemma 23. □

Lemma 27. *If $Cons_{\mathcal{H}}(\mathsf{F}(\mathcal{N}, \vec{\mathcal{A}}))$ is defined, then it is a set of simple values in $\mathcal{W}$.*

Proof. Analogous to the proof for Lemma 24. □

Lemma 28. *If $Cons_{\mathcal{H}}(\mathsf{F}(\mathcal{N}, \vec{\mathcal{A}}))$ is defined and E is the case abstraction $\tilde{\mu}[\overrightarrow{\mathsf{K}^{\overrightarrow{d:k'}}(\vec{\alpha}, \vec{x}).c}]$ such that*

$$\begin{aligned}
&\forall\, \mathsf{K} : \vec{B} \vdash_{\overrightarrow{d:k'}} \mathsf{F}(i, \vec{a}) | \vec{C} \in \overrightarrow{\mathsf{K} : \vec{B} \vdash_{\overrightarrow{d:k'}} \mathsf{F}(i, \vec{a}) | \vec{C}}.\\
&\exists \mathsf{K}^{\overrightarrow{d:k'}}(\vec{\alpha}, \vec{x}).c \in \overrightarrow{\mathsf{K}^{\overrightarrow{d:k'}}(\vec{\alpha}, \vec{x}).c}.\\
&\forall \theta\ [\![\overrightarrow{d : k'}]\!]_{\mathcal{H}\{\mathcal{N}/i, \overrightarrow{\mathcal{A}/a}\}}\ \mathcal{I}.\forall \gamma \in [\![\overrightarrow{x : B}]\!]_{\mathcal{I}}, \delta \in [\![\overrightarrow{\alpha : C}]\!]_{\mathcal{I}}.\\
&\quad c\{\theta, \gamma, \delta\} \in \perp\!\!\!\perp
\end{aligned}$$

then $E \in \langle\!\langle \mathsf{F}(\mathcal{N}, \vec{\mathcal{A}}) \rangle\!\rangle_{\mathcal{H}}$.

Proof. Analogous to the proof for Lemma 25. □

The case for a co-data type defined by noetherian recursion follows analogously by duality.

F.3 (Co-)data Declarations by Primitive Recursion

The constructions for a (co-)data type defined by primitive recursion are themselves defined by primitive recursion on the specified index: if the index is $\mathcal{H}(0)$ then the set of constructors used are those for for the 0 case; if the index equals $\mathcal{H}(+1(\mathcal{N}))$ for some $\mathcal{N} \in [\![Ix]\!]_{\mathcal{H}}$ then the successor case is chosen. For any declared data type of the form

$$\begin{aligned}
&\mathbf{data}\,\mathsf{F}(i : \mathsf{Ix}, \overrightarrow{a : k})\ \mathbf{by}\ \text{by primitive recursion on } i\\
&\mathbf{where}\ i = 0 \qquad \overrightarrow{\mathsf{K} : \vec{B} \vdash_{\overrightarrow{d:k'}} \mathsf{F}(0, \vec{a}) | \vec{C}}\\
&\mathbf{where}\ i = j + 1 \quad \overrightarrow{\mathsf{K}' : \vec{B'} \vdash_{\overrightarrow{d':k''}} \mathsf{F}(0, \vec{a}) | \vec{C'}}
\end{aligned}$$

In addition to the normal ordering of hybrid types for non-recursive declarations, we also have the additional orderings: $\mathsf{F}(\mathcal{N}, \vec{\mathcal{A}}) < \mathsf{F}(\mathcal{H}(+1(\mathcal{N})), \vec{\mathcal{B}})$ for all $\mathcal{N} \in [\![\mathsf{Ix}]\!]_{\mathcal{H}}$. We now define the $Cons$tructions for F by primitive recursion on the index $\mathcal{N}$:

$$\begin{aligned}
&Cons_{\mathcal{H}}(\mathsf{F}(\mathcal{H}(0), \vec{\mathcal{A}}))\\
&\triangleq \{\mathsf{K}^{\overrightarrow{d:k'}}(\vec{\alpha}, \vec{x})\{\gamma, \delta, \theta\} |\\
&\quad \exists\, \mathsf{K} : \vec{B} \vdash_{\overrightarrow{d:k'}} \mathsf{F}(0, \vec{a}) | \vec{C} \in \overrightarrow{\mathsf{K} : \vec{B} \vdash_{\overrightarrow{d:k'}} \mathsf{F}(0, \vec{a}) | \vec{C}}.\\
&\quad \exists \theta\ [\![\overrightarrow{d : k'}]\!]_{\mathcal{H}'}\ \mathcal{I}.\exists \gamma \in [\![\overrightarrow{x : B}]\!]_{\mathcal{I}}, \delta \in [\![\overrightarrow{\alpha : C}]\!]_{\mathcal{I}}\}.\\
&\mathbf{where}\ \mathcal{H}' = \mathcal{H}\{\overrightarrow{\mathcal{A}/a}\}\\
&Cons_{\mathcal{H}}(\mathsf{F}(\mathcal{H}(+1(\mathcal{N})), \vec{\mathcal{A}}))\\
&\triangleq \{\mathsf{K'}^{\overrightarrow{d':k''}}(\vec{\alpha}, \vec{x})\{\gamma, \delta, \theta\} |\\
&\quad \exists\, \mathsf{K}' : \vec{B'} \vdash_{\overrightarrow{d':k''}} \mathsf{F}(j+1, \vec{a}) | \vec{C'} \in\\
&\quad \overrightarrow{\mathsf{K}' : \vec{B'} \vdash_{\overrightarrow{d':k''}} \mathsf{F}(j+1, \vec{a}) | \vec{C'}}.\\
&\quad \exists \theta\ [\![\overrightarrow{d' : k''}]\!]_{\mathcal{H}'}\ \mathcal{I}.\exists \gamma \in [\![\overrightarrow{x : B'}]\!]_{\mathcal{I}}, \delta \in [\![\overrightarrow{\alpha : C'}]\!]_{\mathcal{I}}\}.\\
&\mathbf{where}\ \mathcal{H}' = \mathcal{H}\{\mathcal{N}/j, \overrightarrow{\mathcal{A}/a}\}
\end{aligned}$$

Lemma 29. *If $\mathcal{F}$ is well-founded, $\mathcal{H} \in [\![\mathcal{F}]\!]$ and the declaration of F is well-formed with respect to $\mathcal{F}$, then $Cons_{\mathcal{H}}(\mathsf{F}(\mathcal{N}, \vec{\mathcal{A}}))$ is a defined pre-type for all $\mathcal{N} \in \mathbb{O}$ and $\overrightarrow{\mathcal{A} \in [\![k]\!]_{\mathcal{H}}}$.*

Proof. Because the well-formedness checks for the successor constructors of F can assume that F is already well-formed the previous index j:

$$\frac{\overrightarrow{\Theta, j : \mathsf{Ix}, \Theta' \vdash A : k}}{\Theta, j : \mathsf{Ix}, \Theta' \vdash \mathsf{F}(j, \vec{A}) : \star}$$

we need to proceed by primitive induction on the ordinal $\mathcal{N}$. Besides this difference, the proof follows analogously to Lemma 23. □

Lemma 30. *If $Cons_{\mathcal{H}}(\mathsf{F}(\mathcal{N}, \vec{\mathcal{A}}))$ is defined, then it is a set of simple values in $\mathcal{W}$.*

Proof. Analogous to the proof for Lemma 24. □

Lemma 31. *1. If $Cons_{\mathcal{H}}(\mathsf{F}(\mathcal{H}(0), \vec{\mathcal{A}}))$ is defined and E is the case abstraction $\tilde{\mu}[\overrightarrow{\mathsf{K}^{\overrightarrow{d:k'}}(\vec{\alpha}, \vec{x}).c}]$ such that*

$$\begin{aligned}
&\forall\, \mathsf{K} : \vec{B} \vdash_{\overrightarrow{d:k'}} \mathsf{F}(0, \vec{a}) | \vec{C} \in \overrightarrow{\mathsf{K} : \vec{B} \vdash_{\overrightarrow{d:k'}} \mathsf{F}(0, \vec{a}) | \vec{C}}.\\
&\exists \mathsf{K}^{\overrightarrow{d:k'}}(\vec{\alpha}, \vec{x}).c \in \overrightarrow{\mathsf{K}^{\overrightarrow{d:k'}}(\vec{\alpha}, \vec{x}).c}.\\
&\forall \theta\ [\![\overrightarrow{d : k'}]\!]_{\mathcal{H}\{\overrightarrow{\mathcal{A}/a}\}}\ \mathcal{I}.\forall \gamma \in [\![\overrightarrow{x : B}]\!]_{\mathcal{I}}, \delta \in [\![\overrightarrow{\alpha : C}]\!]_{\mathcal{I}}.\\
&\quad c\{\theta, \gamma, \delta\} \in \perp\!\!\!\perp
\end{aligned}$$

then $E \in \langle\!\langle \mathsf{F}(\mathcal{H}(0), \vec{\mathcal{A}}) \rangle\!\rangle_{\mathcal{H}}$.

2. *If* $Cons_{\mathcal{H}}(\mathsf{F}(\mathcal{H}(\mathcal{N}), \overrightarrow{\mathcal{A}}))$ *is defined and* E *is the case abstraction* $\tilde{\mu}[\overrightarrow{\mathsf{K}'^{\overrightarrow{d':k''}}(\overrightarrow{\alpha}, \overrightarrow{x}).c}]$ *such that*

$$\forall \mathsf{K}' : \overrightarrow{B'} \vdash_{\overrightarrow{d':k''}} \mathsf{F}(j+1, \overrightarrow{a}) | \overrightarrow{C'}$$
$$\in \overrightarrow{\mathsf{K}' : \overrightarrow{B'} \vdash_{\overrightarrow{d':k''}} \mathsf{F}(j+1, \overrightarrow{a}) | \overrightarrow{C'}}.$$
$$\exists \mathsf{K}'^{\overrightarrow{d':k''}}(\overrightarrow{\alpha}, \overrightarrow{x}).c \in \overrightarrow{\mathsf{K}'^{\overrightarrow{d':k''}}(\overrightarrow{\alpha}, \overrightarrow{x}).c}.$$
$$\forall \theta \, [\![\overrightarrow{d' : k''}]\!]_{\mathcal{H}\{\mathcal{N}/j, \overrightarrow{\mathcal{A}/a}\}} \, \mathcal{I}. \forall \gamma \in [\![\overrightarrow{x : B'}]\!]_{\mathcal{I}}, \delta \in [\![\overrightarrow{\alpha : C'}]\!]_{\mathcal{I}}.$$
$$c\{\theta, \gamma, \delta\} \in \perp\!\!\!\perp$$

then $E \in \langle\!\langle \mathsf{F}(\mathcal{H}(\mathcal{N}), \overrightarrow{\mathcal{A}}) \rangle\!\rangle_{\mathcal{H}}$.

Proof. Analogous to the proof for Lemma 25 for both the zero and successor cases. □

F.4 Ascend and Descend

Interestingly, we do not need to include Ascend and Descend as part of the ordering relation on hybrid types. Taking the view that Descend and Ascend are just user defined types, we can compute their definitions as special cases of the general pattern:

$$Cons_{\mathcal{H}}(\mathsf{Descend}(\mathcal{N}, \mathcal{A}))$$
$$\triangleq \{\mathsf{Fall}^{M}(V) | M \, [\![< N]\!]_{\mathcal{H}} \, \mathcal{M}, V \in \mathcal{A}(\mathcal{N})\}$$

By Lemma 26, we know that for all well-founded $\mathcal{F}$ and $\mathcal{H} \in [\![\mathcal{F}]\!]$, then $Cons_{\mathcal{H}}(\mathsf{Descend}(\mathcal{N}, \mathcal{A}))$ is defined and $\langle\!\langle \mathsf{Descend}(\mathcal{N}, \mathcal{A}) \rangle\!\rangle_{\mathcal{H}}$ is in $TypeCore$ for any $\mathcal{N} \in \mathbb{O}$ and $\mathcal{A} \in [\![\mathsf{Ord} \to \star]\!]_{\mathcal{H}}$. By Lemma 27, we know that any defined $Cons_{\mathcal{H}}(\mathsf{Descend}(\mathcal{N}, \mathcal{A}))$ is included in $\langle\!\langle \mathsf{Descend}(\mathcal{N}, \mathcal{A}) \rangle\!\rangle_{\mathcal{H}}$. And by Lemma 28 we know that $E = \tilde{\mu}[\mathsf{Fall}^{j<N}(x).c] \in \langle\!\langle \mathsf{Descend}(\mathcal{N}, \mathcal{A}) \rangle\!\rangle_{\mathcal{H}}$ whenever

$$\forall M [\![< N]\!]_{\mathcal{H}} \mathcal{M}, V \in \mathcal{A}(\mathcal{M}). c\{V/x, M/j\} \in \perp\!\!\!\perp$$

For the special recursive form of case abstraction for Descend, we show that it is included in any family of reducibility candidates indexed by $\mathbb{O}$ which all include the non-recursive form.

Lemma 32 (Folding)**.** *Suppose that* $\mathcal{A}, \mathcal{B} : \mathbb{O} \to CR$ *and* $N \, [\![Ord]\!]_{\mathcal{H}} \, \mathcal{N}$ *such that*

$$\forall M \, [\![< N]\!]_{\mathcal{H}} \, \mathcal{M}, E \in \mathcal{B}(\mathcal{M}). c\{M/j, E/\alpha\} \in runs$$

implies that

$$\tilde{\mu}[\mathsf{Fall}^{j<N}[\alpha].c] \in \mathcal{A}(\mathcal{N})$$

Then

$$\forall M \, [\![< N]\!]_{\mathcal{H}} \, \mathcal{M}, E \in \mathcal{B}(\mathcal{M}), v \in \mathcal{A}(\mathcal{M}). c\{M/j, E/\alpha, V/x\} \in \perp\!\!\!\perp$$

implies that

$$\tilde{\mu}[\mathsf{Fall}^{j<N}[\alpha](x).c] \in \mathcal{A}(\mathcal{N})$$

Proof. By noetherian induction on $\mathcal{N} \in \mathbb{O}$.

Note that the inductive hypothesis is that given any $\mathcal{M} < \mathcal{N}$ and $M \, \mathcal{O}\nabla\lceil[\mathcal{H}] \, \mathcal{M}$,

$$\forall O \, [\![< M]\!]_{\mathcal{H}} \, \mathcal{O}, E \in \mathcal{B}(\mathcal{O}), V \in \mathcal{A}(\mathcal{O}). c\{O/j, E/\alpha, V/x\} \in \perp\!\!\!\perp$$

implies that

$$\tilde{\mu}[\mathsf{Fall}^{j<M}[\alpha](x).c] \in \mathcal{A}(\mathcal{M})$$

We will now use the fact that $\mathcal{A}(\mathcal{N}) \in CR$ to prove that it contains $\tilde{\mu}[\mathsf{Fall}^{j<N}[\alpha](x).c] \in \mathcal{A}(\mathcal{N})$. Observe that for any $E \in \mathcal{A}(\mathcal{N})$, we have the positive head reduction

$$\langle \tilde{\mu}[\mathsf{Fall}^{j<N}[\alpha](x).c] \| E \rangle$$
$$\mapsto_{+} \langle \tilde{\mu}[\mathsf{Fall}^{i<N}[\alpha].c\{i/j, \tilde{\mu}[\mathsf{Fall}^{j<i}[\alpha](x).c]/x\}] \| E \rangle$$

And since $Head(\mathcal{A}(\mathcal{N})) \sqsubseteq \mathcal{A}(\mathcal{N}) = \mathcal{A}(\mathcal{N})^{\perp}$, it suffices to show that

$$\tilde{\mu}[\mathsf{Fall}^{i<N}[\alpha].c\{i/j, \tilde{\mu}[\mathsf{Fall}^{j<i}[\alpha](x).c]/x\}] \in \mathcal{A}(\mathcal{N})$$

This follows from our assumption about $\mathcal{A}(\mathcal{N})$ so long as

$$\forall M \, [\![< N]\!]_{\mathcal{H}} \, \mathcal{M}, E \in \mathcal{B}(\mathcal{M}).$$
$$c\{M/j, E/\alpha, \tilde{\mu}[\mathsf{Fall}^{j<M}[\alpha](x).c]/x\} \in \perp\!\!\!\perp$$

which follows from our assumption about c so long as for any $M \, [\![< N]\!]_{\mathcal{H}} \, \mathcal{M}$,

$$\tilde{\mu}[\mathsf{Fall}^{j<M}[\alpha](x).c] \in \mathcal{A}(\mathcal{M})$$

which we show by the inductive hypothesis.

Now, suppose that $O \, [\![< M]\!]_{\mathcal{H}} \, \mathcal{O}, E \in \mathcal{B}(\mathcal{O})$, and $V \in \mathcal{A}(\mathcal{O})$. By unfolding the definitions of $[\![< M]\!]_{\mathcal{H}}$ and $[\![< N]\!]_{\mathcal{H}}$, we have:

- $M \, [\![< N]\!]_{\mathcal{H}} \mathcal{M} \equiv \exists M' \in Type. M \twoheadrightarrow_{\beta} M' \wedge [\![M']\!]_{\mathcal{H}} \leq \mathcal{M} < [\![N]\!]_{\mathcal{H}}$,
- $O \, [\![< M]\!]_{\mathcal{H}} \mathcal{O} \equiv \exists O' \in Type. O \twoheadrightarrow_{\beta} O' \wedge [\![O']\!]_{\mathcal{H}} \leq \mathcal{O} < [\![M]\!]_{\mathcal{H}}$,

Note that forward reduction of syntactic types cannot change the ordinal value of their interpretation.

Lemma 33. *If* $[\![M]\!]_{\mathcal{H}} \in \mathbb{O}$ *and* $M \twoheadrightarrow_{\beta} M'$ *then* $[\![M]\!]_{\mathcal{H}} = [\![M']\!]_{\mathcal{H}} \in \mathbb{O}$.

Proof. By induction on the reductions $M \twoheadrightarrow_{\beta} M'$ and induction on the source term to find the redex. □

Therefore, $[\![O']\!]_{\mathcal{H}} \leq \mathcal{O} < [\![M]\!]_{\mathcal{H}} \leq \mathcal{M} < [\![N]\!]_{\mathcal{H}}$, and so $O \, [\![< N]\!]_{\mathcal{H}} \mathcal{O}$ as well.

We can now use the assumption on the command c again to obtain $c\{O/j, E/\alpha, V/x\} \in \perp\!\!\!\perp$. Thus it follows that

$$\forall O [\![< M]\!]_{\mathcal{H}} \mathcal{O}, E \in \mathcal{B}(\mathcal{O}), V \in \mathcal{A}(\mathcal{O}). c\{O/j, E/\alpha, V/x\} \in \perp\!\!\!\perp$$

Applying the inductive hypothesis now grants us that

$$\tilde{\mu}[\mathsf{Fall}^{j<M}[\alpha](x).c] \in \mathcal{A}(\mathcal{M})$$

thus completing the proof. □

As a corollary, note that so long as $\mathcal{H}(\mathsf{Descend}(\mathcal{N}, \mathcal{B}))$ is a reducibility candidate for all $\mathcal{N} \in \mathbb{O}$ and $\mathcal{B} : \mathbb{O} \to CR$ where $\langle\!\langle \mathsf{Descend}(\mathcal{N}, \mathcal{B}) \rangle\!\rangle_{\mathcal{H}} \sqsubseteq \mathcal{H}(\mathsf{Descend}(\mathcal{N}, \mathcal{B}))$, then the function $\lambda \mathcal{N} \in \mathbb{O}. \mathcal{H}(\mathsf{Descend}(\mathcal{N}, \mathcal{B}))$ meets the criteria of Lemma 32, so it contains all the matching recursive case abstractions

$$\tilde{\mu}[\mathsf{Fall}^{j<N}(x)[\alpha].c] \in \mathsf{Descend}(\mathcal{N}, \mathcal{B})$$

meeting the specified criteria for every $N \, [\![\mathsf{Ord}]\!]_{\mathcal{H}} \, \mathcal{N}$.

The co-data type constructor Ascend follows analogously by duality.

F.5 Deflate and Inflate

Similarly, we can compute the definition of Deflate.

$$Cons_{\mathcal{H}}(\mathsf{Deflate}(\mathcal{A})) \triangleq \{\mathsf{Down}^{M}(V) | M \, [\![\mathsf{Ix}]\!]_{\mathcal{H}} \, \mathcal{M}, V \in \mathcal{A}(\mathcal{M})\}$$

All the usual properties for a data type like Deflate hold, and in addition we have the recursive form of case abstraction.

Lemma 34 (Looping)**.** *Suppose* $\mathcal{H}$ *is plausible and* $\mathcal{A} : \pi_1([\![\mathsf{Ix}]\!]_{\mathcal{H}}) \to CR$. *Given* c_0, c_1 *such that*

$$\forall V \in \mathcal{A}(\mathcal{H}(0)). c_0\{V/x\} \in \perp\!\!\!\perp$$
$$\forall N \, [\![\mathsf{Ix}]\!]_{\mathcal{H}} \, \mathcal{N}, V \in \mathcal{A}(\mathcal{H}(+1(\mathcal{N}))), E \in \mathcal{A}(\mathcal{N}).$$
$$c_1\{N/j, V/x, E/\alpha\} \in \perp\!\!\!\perp$$

then $E = \tilde{\mu}[\mathsf{Down}^{0:\mathsf{Ix}}(x).c | \mathsf{Down}^{j+1:\mathsf{Ix}}(x)[\alpha].c_1] \in \langle\!\langle \mathsf{Deflate}(\mathcal{A}) \rangle\!\rangle_{\mathcal{H}}$.

Proof. Note that $\langle\!\langle \mathsf{Deflate}(\mathcal{A})\rangle\!\rangle_{\mathcal{H}} = \langle\!\langle \mathsf{Deflate}(\mathcal{A})\rangle\!\rangle_{\mathcal{H}}{}^{\perp s}$. Since E is simple, it remains to show that $E \in \mathcal{W}$ and for all $V \in Cons_{\mathcal{H}}(\mathsf{Deflate}(\mathcal{A}))$, $\langle V \| E\rangle \in \perp\!\!\!\perp$.

- $E \in \mathcal{W}$: Note that since $\mathcal{H}$ is plausible, $\mathcal{H}(0), \mathcal{H}(+1(\mathcal{H}(0))) \in \mathbb{O}$. Additionally, $0 \sim_{\mathcal{H}} \mathcal{H}(0)$ and $0+1 \sim_{\mathcal{H}} \mathcal{H}(+1(\mathcal{H}(0)))$ by definition. Therefore, $\mathcal{A}(\mathcal{H}(0)), \mathcal{A}(\mathcal{H}(+1(\mathcal{H}(0)))) \in CR$, and so both contain the (co-)variables by Lemma 2. Therefore, $c_0 \in \perp\!\!\!\perp$ and $c_1\{0/j\} \in \perp\!\!\!\perp$, and so $c_1 \in \perp\!\!\!\perp$ by Lemma 20. Thus, E is strongly normalizing so $E \in \mathcal{W}$.
- For all $V \in Cons_{\mathcal{H}}(\mathsf{Deflate}(\mathcal{A}))$, $\langle V \| E\rangle \in \perp\!\!\!\perp$: Note that $V = \mathsf{Down}^{N}(V')$ for some $N \sim_{\mathcal{H}} \mathcal{N}$ and $V' \in \mathcal{A}(\mathcal{N})$. We proceed by induction on $N \sim_{\mathcal{H}} \mathcal{N}$.
  - $0 \sim_{\mathcal{H}} \mathcal{H}(0)$: Observe that
  $$\langle V \| E\rangle \mapsto_0 c_0\{V'/x\} \in \perp\!\!\!\perp$$
  so $\langle V \| E\rangle \in \perp\!\!\!\perp$ by Lemma 14.
  - $M+1 \sim_{\mathcal{H}} \mathcal{H}(+1(\mathcal{M}))$ for some $M \sim_{\mathcal{H}} \mathcal{M}$: Observe that
  $$\langle V \| E\rangle \mapsto_0 \langle \mu\alpha.c_1\{M/j, V/x\} \| \tilde{\mu}y.\langle \mathsf{Down}^{M}(y) \| E\rangle\rangle$$
  By the inductive hypothesis, $\langle \mathsf{Down}^{M}(V') \| E\rangle \in \perp\!\!\!\perp$ for all $V' \in \mathcal{A}(\mathcal{N})$, so $\tilde{\mu}y.\langle \mathsf{Down}^{M}(y)\|E\rangle \in \mathcal{A}(\mathcal{N})$ by strong activation (Lemma 11) because $\mathcal{A}(\mathcal{N})$ is a reducibility candidate. Furthermore, for all $E' \in \mathcal{A}(\mathcal{N})$, $c_1\{M/j, V/x, E'/\alpha\} \in \perp\!\!\!\perp$ by assumption on c_1, so we have $\mu\alpha.c_1\{M/j, V/x\}$ for the same reason. Therefore,
  $$\langle \mu\alpha.c_1\{M/j, V/x\} \| \tilde{\mu}y.\langle \mathsf{Down}^{M}(y) \| E\rangle\rangle \in \perp\!\!\!\perp$$
  and $\langle V \| E\rangle \in \perp\!\!\!\perp$ as well by Lemma 14. $\square$

The co-data type constructor $\mathsf{Inflate}$ follows analogously by duality.

G. Soundness

We give the following meaning to the sequents

$$\begin{aligned}
[\![\Theta \vdash A : k]\!]_{\mathcal{H}} &\iff (\forall \theta [\![\Theta]\!]_{\mathcal{H}} \mathcal{I}. A\{\theta\}\ [\![k]\!]_{\mathcal{I}}\ [\![A]\!]_{\mathcal{I}}) \\
&\quad \wedge \forall \mathcal{I} \in [\![\Theta]\!]_{\mathcal{H}}, [\![A]\!]_{\mathcal{I}} \in [\![k]\!]_{\mathcal{I}} \\
[\![\Theta \vdash k : s]\!]_{\mathcal{H}} &\iff \forall \mathcal{I} \in [\![\Theta]\!]_{\mathcal{H}}, [\![k]\!]_{\mathcal{I}} \in [\![s]\!] \\
[\![(\vdash_{\cdot})]\!]_{\mathcal{H}} &\iff true \\
[\![(\vdash_{\Theta, a:k})]\!]_{\mathcal{H}} &\iff [\![(\vdash_{\Theta})]\!]_{\mathcal{H}} \wedge [\![\Theta \vdash k : \blacksquare]\!]_{\mathcal{H}} \\
[\![(\Gamma \vdash_{\Theta} \Delta)\,\mathbf{seq}]\!]_{\mathcal{H}} &\iff [\![\vdash_{\Theta}]\!]_{\mathcal{H}} \\
&\quad \wedge (\forall x : A \in \Gamma. [\![\Theta \vdash A : \star]\!]_{\mathcal{H}}) \\
&\quad \wedge (\forall \alpha : A \in \Delta. [\![\Theta \vdash A : \star]\!]_{\mathcal{H}}) \\
[\![\Theta \vdash A = B : k]\!]_{\mathcal{H}} &\iff \forall \mathcal{I} \in [\![\Theta]\!]_{\mathcal{H}}. [\![A]\!]_{\mathcal{I}} = [\![B]\!]_{\mathcal{I}} \in [\![k]\!]_{\mathcal{I}} \\
[\![c : (\Gamma \vdash_{\Theta} \Delta)]\!]_{\mathcal{H}} &\iff \forall \theta\ [\![\Theta]\!]_{\mathcal{H}}\ \mathcal{I}. \forall \gamma \in [\![\Gamma]\!]_{\mathcal{I}}, \delta \in [\![\Delta]\!]_{\mathcal{I}}. \\
&\quad c\{\gamma, \delta, \theta\} \in \perp\!\!\!\perp \\
[\![\Gamma \vdash_{\Theta} v : T \mid \Delta]\!]_{\mathcal{H}} &\iff \forall \theta\ [\![\Theta]\!]_{\mathcal{H}}\ \mathcal{I}. \forall \gamma \in [\![\Gamma]\!]_{\mathcal{I}}, \delta \in [\![\Delta]\!]_{\mathcal{I}}. \\
&\quad v\{\gamma, \delta, \theta\} \in [\![T]\!]_{\mathcal{I}} \\
[\![\Gamma \mid e : T \vdash_{\Theta} \Delta]\!]_{\mathcal{H}} &\iff \forall \theta\ [\![\Theta]\!]_{\mathcal{H}}\ \mathcal{I}. \forall \gamma \in [\![\Gamma]\!]_{\mathcal{I}}, \delta \in [\![\Delta]\!]_{\mathcal{I}}. \\
&\quad e\{\gamma, \delta, \theta\} \in [\![T]\!]_{\mathcal{I}}
\end{aligned}$$

Lemma 35. *Given any $\mathcal{H} \in [\![\mathcal{F}]\!]$:*

1. *if $\Theta \vdash k : s$ and $(\vdash_{\Theta})\,\mathbf{seq}$ are derivable in $\mu\tilde{\mu}^{\mathcal{F}}_{\mathcal{S}}$ then $[\![\Theta \vdash k : s]\!]_{\mathcal{H}}$;*
2. *if $\Theta \vdash A : k$ and $(\vdash_{\Theta, a:k})\,\mathbf{seq}$ are derivable in $\mu\tilde{\mu}^{\mathcal{F}}_{\mathcal{S}}$ then $[\![\Theta \vdash A : k]\!]_{\mathcal{H}}$;*
3. *if $\Theta \vdash A = B : k$ and $(\vdash_{\Theta, a:k})\,\mathbf{seq}$ are derivable in $\mu\tilde{\mu}^{\mathcal{F}}_{\mathcal{S}}$ then $[\![\Theta \vdash A = B : k]\!]_{\mathcal{H}}$;*
4. *if $(\Gamma \vdash_{\Theta} \Delta)\,\mathbf{seq}$ is derivable in $\mu\tilde{\mu}^{\mathcal{F}}_{\mathcal{S}}$ then $[\![(\Gamma \vdash_{\Theta} \Delta)\,\mathbf{seq}]\!]_{\mathcal{H}}$.*

Proof. By mutual induction on the typing derivation. In the case where we have two assumption derivations, it is the first one which we take to be decreasing.

- $$\frac{}{\Theta \vdash 0 : \mathsf{Ix}}$$
  If $\mathcal{I} \in [\![\Theta]\!]_{\mathcal{H}}$ then $\mathcal{I} \in [\![\mathcal{F}]\!]$ by Lemma 8, so $[\![0]\!]_{\mathcal{I}} \in \mathbb{O}$. Further, $0\{\theta\} = 0 \sim_{\mathcal{I}} \mathcal{I}(0)$.
- $$\frac{\Theta \vdash M : \mathsf{Ix}}{\Theta \vdash M+1 : \mathsf{Ix}}$$
  If $\mathcal{I} \in [\![\Theta]\!]_{\mathcal{H}}$ then $\mathcal{I} \in [\![\mathcal{F}]\!]$ by Lemma 8 and since by the inductive hypothesis $[\![M]\!]_{\mathcal{I}} \in [\![\mathsf{Ix}]\!]_{\mathcal{I}}$, we know that $[\![M+1]\!]_{\mathcal{I}} = \mathcal{I}(+1([\![M]\!]_{\mathcal{I}})) \in [\![\mathsf{Ix}]\!]_{\mathcal{I}}$. Further, if $\theta\ [\![\Theta]\!]_{\mathcal{H}}\ \mathcal{I}$ then $M\{\theta\}\ [\![\mathsf{Ix}]\!]_{\mathcal{I}}\ [\![M]\!]_{\mathcal{I}}$ so $M\{\theta\} \sim_{\mathcal{I}} [\![M]\!]_{\mathcal{I}}$ and $(M+1)\{\theta\} \sim_{\mathcal{I}} \mathcal{I}(+1([\![M]\!]_{\mathcal{I}}))$.
- $$\frac{}{\Theta \vdash 0 < \infty}$$
  If $\mathcal{I} \in [\![\Theta]\!]_{\mathcal{H}}$ then $\mathcal{I} \in [\![\mathcal{F}]\!]$ by Lemma 8 so this holds by the requirement that $\mathcal{I}(0) < \mathcal{I}(\infty)$ from Definition 6.
- $$\frac{\Theta \vdash M < \infty}{\Theta \vdash M+1 < \infty}$$
  If $\mathcal{I} \in [\![\Theta]\!]_{\mathcal{H}}$ then $\mathcal{I} \in [\![\mathcal{F}]\!]$ by Lemma 8 so this holds by the requirement that $\mathcal{I}(+1(\mathcal{M})) < \mathcal{I}(\infty)$ whenever $\mathcal{M} < \mathcal{I}(\infty)$ from Definition 6.
- $$\frac{\Theta \vdash M < N}{\Theta \vdash M < M+1}$$
  If $\mathcal{I} \in [\![\Theta]\!]_{\mathcal{H}}$ then $\mathcal{I} \in [\![\mathcal{F}]\!]$ by Lemma 8 so this holds by the requirement that when ever $\mathcal{M} < \mathcal{N}$ we have $\mathcal{M} < \mathcal{I}(+1(\mathcal{M})) \leq \mathcal{N}$ from Definition 6.
- $$\frac{\Theta \vdash M_1 < M_2 \quad \Theta \vdash M_2 < M_3}{\Theta \vdash M_1 < M_3}$$
  This holds by the underlying transitivity of the ordering on $\mathbb{O}$.
- $$\frac{a : k' \notin \Theta'}{\Theta, a : k, \Theta' \vdash a : k}$$
  By induction on the length of Θ'. The proof of the base case is immediate. In the inductive case if $[\![\Theta, a : k, \Theta' \vdash a : k]\!]_{\mathcal{H}}$ then for any $\mathcal{I} \in [\![\Theta, a : k, \Theta', b : k']\!]_{\mathcal{H}}$ there must exist some $\mathcal{I}' \in [\![\Theta, a : k, \Theta']\!]_{\mathcal{H}}$ that agree on everything except b. Thus, we know that $[\![a]\!]_{\mathcal{I}} = [\![a]\!]_{\mathcal{I}'} \in [\![k]\!]_{\mathcal{I}'} = [\![k]\!]_{\mathcal{I}}$. Similarly, if $\theta\ [\![\Theta, a : k, \Theta', b : k']\!]_{\mathcal{H}}\ \mathcal{I}$ then there exists $\theta'\ [\![\Theta, a : k, \Theta']\!]_{\mathcal{H}}\ \mathcal{I}'$ such that $a\{\theta\} = a\{\theta'\}\ [\![k]\!]_{\mathcal{I}'}\ [\![a]\!]_{\mathcal{I}'} = [\![a]\!]_{\mathcal{I}}$ and so $a\{\theta\}\ [\![k]\!]_{\mathcal{I}}\ [\![a]\!]_{\mathcal{I}}$.
- $$\frac{\Theta, a : k_1 \vdash B : k_2 \quad \Theta \vdash k_2 : \square}{\Theta \vdash \lambda a : k_1.B : k_1 \to k_2}$$
  If $\mathcal{I} \in [\![\Theta]\!]_{\mathcal{H}}$ and $\mathcal{A}$ in $[\![k_1]\!]_{\mathcal{I}}$ then $\mathcal{I}\{\mathcal{A}/a\} \in [\![\Theta, a : k_1]\!]_{\mathcal{H}}$ so, by the inductive hypothesis, $[\![B]\!]_{\mathcal{I}\{\mathcal{A}/a\}} \in [\![k_2]\!]_{\mathcal{I}\{\mathcal{A}/a\}}$ and $[\![k_2]\!]_{\mathcal{I}} \in [\![\square]\!]$. Because $\Theta \vdash k_2 : \square$, $a \notin FV(k_2)$, so by extending $[\![k_2]\!]_{\mathcal{I}} = [\![k_2]\!]_{\mathcal{I}\{\mathcal{A}/a\}}$, we have $[\![B]\!]_{\mathcal{I}\{\mathcal{A}/a\}} \in [\![k_2]\!]_{\mathcal{I}}$ meaning $[\![\lambda a : k_1.B]\!]_{\mathcal{I}} \in [\![k_1 \to k_2]\!]_{\mathcal{I}}$. Now, suppose $\theta\ [\![\Theta]\!]_{\mathcal{H}}\ \mathcal{I}$, then for any $A\{\theta\}\ [\![k_1]\!]_{\mathcal{I}}\ \mathcal{A}$ we have that $\theta\{A\{\theta\}/a\}\ [\![\Theta, a : k_1]\!]_{\mathcal{I}}\ \mathcal{I}\{\mathcal{A}/a\}$ and so it follows that as well $B\{\theta, A\{\theta\}/a\}\ [\![k_2]\!]_{\mathcal{I}\{\mathcal{A}/a\}}\ [\![B]\!]_{\mathcal{I}\{\mathcal{A}/a\}}$. Again, we get $B\{\theta, A\{\theta\}/a\}\ [\![k_2]\!]_{\mathcal{I}}\ [\![B]\!]_{\mathcal{I}\{\mathcal{A}/a\}}$ by extending $[\![k_2]\!]_{\mathcal{I}} \in [\![\square]\!]$. Furthermore, since $[\![k_2]\!]_{\mathcal{I}} \in [\![\square]\!]$, we know that because $(\lambda a : k_1.B)\ A \to_{\beta} B\{A/a\}$, then we also have that

$(\lambda a : k_1.B\{\theta\})\ A\{\theta\}\ [\![k_2]\!]_\mathcal{I}\ [\![B]\!]_{\mathcal{I}\{\mathcal{A}/a\}}$ Therefore, we have $(\lambda a : k_1.B)\{\theta\}\ [\![k_1 \to k_2]\!]_\mathcal{I}\ [\![\lambda a : k.B]\!]_\mathcal{I}$.

- $$\frac{\Theta \vdash A : k_1 \to k_2 \quad \Theta \vdash B : k_1}{\Theta \vdash A\ B : k_2}$$
  If $\mathcal{I} \in [\![\Theta]\!]_\mathcal{H}$ then, by inductive hypothesis $[\![A]\!]_\mathcal{I} : [\![k_1]\!]_\mathcal{I} \to [\![k_2]\!]_\mathcal{I}$ and $[\![B]\!]_\mathcal{I} \in [\![k_1]\!]_\mathcal{I}$ so $[\![A\ B]\!]_\mathcal{I} \in [\![k_2]\!]_\mathcal{I}$. By the same token, if $\theta\ [\![\Theta]\!]_\mathcal{H}\ \mathcal{I}$ then $B\{\theta\}\ [\![k_1]\!]\ [\![B]\!]_\mathcal{I}$ so $(A\ B)\{\theta\}\ [\![k_2]\!]\ [\![A\ B]\!]_\mathcal{I}$.
- $$\frac{\Theta \vdash M < N \quad \Theta \vdash N : \mathsf{Ord}}{\Theta \vdash M : \mathsf{Ord}}$$
  If $\mathcal{I} \in [\![\Theta]\!]_\mathcal{H}$ then by inductive hypothesis $[\![M]\!]_\mathcal{I} \in [\![< N]\!]_\mathcal{I}$ which means that $[\![M]\!]_\mathcal{I} \in \mathbb{O}$ so $[\![M]\!]_\mathcal{I} \in [\![\mathsf{Ord}]\!]_\mathcal{I}$. If $\theta\ [\![\Theta]\!]_\mathcal{H}\ \mathcal{I}$ then by inductive hypothesis, $M\{\theta\}\ [\![< N]\!]_\mathcal{I}\ [\![M]\!]_\mathcal{I}$ meaning that there is a $M\{\theta\} \twoheadrightarrow_\beta M'$ such that $[\![M']\!]_\mathcal{I} \leq [\![M]\!]_\mathcal{I} < [\![N]\!]_\mathcal{I}$ thus $M\{\theta\}\ [\![\mathsf{Ord}]\!]_\mathcal{I}\ [\![M]\!]_\mathcal{I}$.
  Note that the use of the inductive hypothesis here is slightly non trivial and so requires explanation. We know that $(\vdash_{\Theta,a:\mathsf{Ord}})\ \mathbf{seq}$ is derivable and thus $(\vdash_\Theta)\ \mathbf{seq}$ is derivable. Further, since $\Theta \vdash N : \mathsf{Ord}$ is derivable so is $\Theta \vdash < N : \Box$ and so $(\vdash_{\Theta,a<N})\ \mathbf{seq}$.
- $$\frac{}{\Theta \vdash \infty : \mathsf{Ord}}$$
  By the requirement on $\mathcal{I} \in [\![\mathcal{F}]\!]$ that $\mathcal{I}(\infty) \in \mathbb{O}$.
- $$\frac{\overrightarrow{\Theta \vdash A : k}}{\Theta \vdash \mathsf{F}(\vec{A}) : \star}$$
  For any $\mathcal{I} \in [\![\Theta]\!]_\mathcal{H}$ we know that $\mathcal{I} \in [\![\mathcal{F}]\!]$ and since, by inductive hypothesis, $\overrightarrow{[\![A]\!]_\mathcal{I} \in [\![k]\!]_\mathcal{I}}$ that means $[\![\mathsf{F}(\vec{A})]\!]_\mathcal{I} = \mathcal{I}(\mathsf{F}(\overrightarrow{[\![A]\!]_\mathcal{I}})) \in CR$ so $[\![\mathsf{F}(\vec{A})]\!]_\mathcal{I} \in [\![\star]\!]_\mathcal{I}$. Now, if $\theta\ [\![\Theta]\!]\ \mathcal{I}$ then $\mathsf{F}(\vec{A})\{\theta\}\ [\![\star]\!]_\mathcal{I}\ [\![\mathsf{F}(\vec{A})]\!]_\mathcal{I}$ since $[\![\star]\!]$ is the total relation.
- $$\frac{\Theta \vdash k : \Box}{\Theta \vdash k : \blacksquare}$$
  For any $\mathcal{I} \in [\![\Theta]\!]_\mathcal{H}$, Observe that $[\![\blacksquare]\!]$ contains everything in $[\![\Box]\!]$, so the inductive hypothesis $[\![\Theta \vdash k : \Box]\!]_\mathcal{I}$ implies $[\![\Theta \vdash k : \blacksquare]\!]_\mathcal{I}$.
- $$\frac{\Theta \vdash k_1 : \blacksquare \quad \Theta \vdash k_2 : \Box}{\Theta \vdash k_1 \to k_2 : \Box}$$
  If $\mathcal{I} \in [\![\Theta]\!]_\mathcal{H}$ then, by the inductive hypothesis, $[\![k_1]\!]_\mathcal{I} \in [\![\blacksquare]\!]$ and $[\![k_2]\!]_\mathcal{I} \in [\![\Box]\!]$, so for any $A'\ B\ [\![k_2]\!]_\mathcal{I}\ \mathcal{A}(\mathcal{B})$, if $A \twoheadrightarrow_\beta A'$ then $A\ B\ [\![k_2]\!]_\mathcal{I}\ \mathcal{S}(\mathcal{B})$ as well. Therefore $[\![k_1 \to k_2]\!]_\mathcal{I} \in [\![\Box]\!]$ since $A'[\![k_1 \to k_2]\!]_\mathcal{I}\mathcal{A}$ and $A \twoheadrightarrow_\beta A'$ implies that $A[\![k_1 \to k_2]\!]_\mathcal{I}\mathcal{A}$.
- $$\frac{}{\Theta \vdash \star : \Box}$$
  Note that for any $\mathcal{I}$, syntactic type A, and $CR\ \mathcal{B}$, $A\ [\![\star]\!]_\mathcal{I}\ \mathcal{B}$, so $[\![\star]\!]_\mathcal{I} \in [\![\Box]\!]$.
- $$\frac{}{\Theta \vdash \mathsf{Ix} : \blacksquare}$$
  Immediate.
- $$\frac{}{\Theta \vdash \mathsf{Ord} : \Box}$$
  If $\mathcal{I} \in [\![\Theta]\!]$, then note that by definition of $[\![\mathsf{Ord}]\!]_\mathcal{I}$, $M'\ [\![\mathsf{Ord}]\!]_\mathcal{I}\ \mathcal{M}$ and $M \twoheadrightarrow_\beta M'$ implies that $M\ [\![\mathsf{Ord}]\!]_\mathcal{I}\ \mathcal{M}$ as well. Therefore, $[\![\mathsf{Ord}]\!]_\mathcal{I} \in [\![\Box]\!]$.
- $$\frac{\Theta \vdash N : \mathsf{Ord}}{\Theta \vdash (< N) : \Box}$$
  We have $\Theta \vdash \mathsf{Ord} : \blacksquare$ so $(\vdash_\Theta)\ \mathbf{seq}$ allows us to prove $(\vdash_{\Theta,a:\mathsf{Ord}})\ \mathbf{seq}$. Thus, by induction we know that $[\![\Theta \vdash N : \mathsf{Ord}]\!]_\mathcal{H}$. So, for any $\mathcal{I} \in [\![\Theta]\!]_\mathcal{H}$ we have $[\![N]\!]_\mathcal{I} \in \mathbb{O}$. Furthermore, for any $M'\ [\![< N]\!]_\mathcal{I}\ \mathcal{M}$ where $M \twoheadrightarrow_\beta M'$, then $M\ [\![< N]\!]_\mathcal{I}\ \mathcal{M}$ as well, so $[\![< N]\!]_\mathcal{I} \in [\![\Box]\!]$.
- $$\frac{\Theta, a : k_1 \vdash B : k_2 \quad \Theta \vdash A : k_1 \quad \Theta \vdash k_1 : \blacksquare}{\Theta \vdash B\{A/a\} = (\lambda a : k_1.B)\ A : k_2}$$
  If $\mathcal{I} \in [\![\Theta]\!]_\mathcal{H}$, by the inductive hypothesis, we know that $[\![A]\!]_\mathcal{I} \in [\![k_1]\!]_\mathcal{I}$ and $\mathcal{I}\{[\![A]\!]_\mathcal{I}/a\} \in [\![\Theta, a : k_1]\!]$ so $[\![B]\!]_{\mathcal{I}\{[\![A]\!]_\mathcal{I}/a\}} = [\![B\{A/a\}]\!]_\mathcal{I} \in [\![k_2]\!]\mathcal{I}$ by Lemma 22 as well. Furthermore, $[\![(\lambda a : k_1.B)\ A]\!]_\mathcal{I} = [\![B]\!]_{\mathcal{I}\{[\![A]\!]_\mathcal{I}/a\}} = [\![B\{A/a\}]\!]_\mathcal{I}$.
  Note that to invoke the inductive hypothesis on $\Theta, a : k_1 \vdash B : k_2$, we need $(\vdash_{\Theta,a:k_1,b:k_2})\ \mathbf{seq}$, which is derivable from the given $(\vdash_{\Theta,b:k_2})\ \mathbf{seq}$ and the premise $\Theta \vdash k_1 : \blacksquare$.
- $$\frac{\Theta \vdash B : k_1 \to k_2}{\Theta \vdash (\lambda a : k_1.B\ a) = B : k_1 \to k_2}$$
  If $\mathcal{I} \in [\![\Theta]\!]_\mathcal{H}$, then by the inductive hypothesis we have that $[\![B]\!]_\mathcal{I} \in [\![k_1 \to k_2]\!]_\mathcal{I}$. Furthermore, observe that we have $\lambda\mathcal{A} \in [\![k_1]\!]_\mathcal{I}.[\![B]\!]_{\mathcal{I}\{\mathcal{A}/a\}}(\mathcal{A}) = \lambda\mathcal{A} \in [\![k_1]\!]_\mathcal{I}.[\![B]\!]_\mathcal{I}(\mathcal{A})$ by Lemma 22 because $a \notin FV(B)$, so it denotes the same function as $[\![B]\!]_\mathcal{I}$.
- $$\frac{\Theta \vdash A : k}{\Theta \vdash A = A : k} \qquad \frac{\Theta \vdash B = A : k}{\Theta \vdash A = B : k}$$
  $$\frac{\Theta \vdash A = B : k \quad \Theta \vdash B = C : k}{\Theta \vdash A = C : k}$$
  $$\frac{\Theta \vdash A = A' : k_1 \to k_2 \quad \Theta \vdash B = B' : k_1 \quad \Theta \vdash k_1 : \blacksquare}{\Theta \vdash A\ B = A'\ B' : k_2}$$
  $$\frac{\Theta, a : k_1 \vdash B = B' : k_2}{\Theta \vdash (\lambda a : k_1.B) = (\lambda a : k_1.B') : k_1 \to k_2}$$
  Immediate by the inductive hypothesis. Note that to apply the inductive hypothesis for the equivalence $\Theta \vdash A\ B = A'\ B' : k_2$, we need the premise $\Theta \vdash k_1 : \blacksquare$ to ensure that $(\vdash_{\Theta,b:k_1})\ \mathbf{seq}$ is derivable.
  Also note that α-equivalence for type-level lambdas is derivable from the $\beta\eta$ equalities along with these rules.
- The rules for $\mathbf{seq}$ are immediate. $\Box$

Lemma 36. *If all (co-)data declarations in $\mathcal{F}$ are well-formed, then $\mathcal{F}$ is well-founded.*

Proof. By induction on the order of declarations in $\mathcal{F}$. For each declaration, we extend the order accordingly and use either Lemma 23, Lemma 26, or Lemma 23, depending on which kind of declaration it is, to ensure that the extended order satisfies the conditions in Definition 7. $\Box$

Theorem 2. *For any $\mathcal{H} \in [\![\mathcal{F}]\!]$,*

- *If $c : \Gamma \vdash_\Theta \Delta$ and $(\Gamma \vdash_\Theta \Delta)\ \mathbf{seq}$ are derivable in $\mu\tilde{\mu}^\mathcal{F}_\mathcal{S}$ then $[\![c : \Gamma \vdash_\Theta \Delta]\!]_\mathcal{H}$.*
- *If $\Gamma \vdash_\Theta v : A|\Delta$ and $(\Gamma \vdash_\Theta \alpha : A, \Delta)\ \mathbf{seq}$ are derivable in $\mu\tilde{\mu}^\mathcal{F}_\mathcal{S}$ then $[\![\Gamma \vdash_\Theta v : A|\Delta]\!]_\mathcal{H}$.*
- *If $\Gamma|e : A \vdash_\Theta \Delta$ and $(\Gamma, x : A \vdash_\Theta \Delta)\ \mathbf{seq}$ are derivable in $\mu\tilde{\mu}^\mathcal{F}_\mathcal{S}$ then $[\![\Gamma|e : A \vdash_\Theta \Delta]\!]_\mathcal{H}$.*

Proof. By induction on the first argument. In each case, suppose that, $\theta\ [\![\Theta]\!]_\mathcal{H}\ \mathcal{I}$ and $\gamma \in [\![\Gamma]\!]_\mathcal{I}$ and $\delta \in [\![\Delta]\!]_\mathcal{I}$, where Γ and Δ are the typing environments of the concluding sequent. Also, note that for each case, because $\mathcal{H} \in \llbracket \mathcal{F} \rrbracket$, we know that $\llbracket A \rrbracket_{\mathcal{I}}$ is a reducibility candidate whenever $\Theta \vdash A : \star$ and therefore whenever $x : A \in \Gamma$ or $\alpha : A \in \Delta$ due to the assumption that $(\Gamma \vdash_\Theta \Delta)\,\mathbf{seq}$. First, we have the structural rules that apply for any type.

- Var for type A: Note that $x : A \in \Gamma$ so by definition, $x\{\theta,\gamma,\delta\} \in Val(\llbracket A \rrbracket_{\mathcal{I}}) \sqsubseteq \llbracket A \rrbracket_{\mathcal{I}}$.
- $CoVar$ for type A: Analogous to the previous case by duality.
- Cut for type A: By the inductive hypothesis, $v\{\theta,\gamma,\delta\} \in \llbracket A \rrbracket_{\mathcal{I}}$ and $e\{\theta,\gamma,\delta\} \in \llbracket A \rrbracket_{\mathcal{I}}$. Furthermore, because $\llbracket A \rrbracket_{\mathcal{I}}$ is a reducibility candidate we know $\llbracket A \rrbracket_{\mathcal{I}} = \llbracket A \rrbracket_{\mathcal{I}}{}^{\perp\!\!\!\perp}$, so $\langle v\{\theta,\gamma,\delta\} \| e\{\theta,\gamma,\delta\} \rangle = \langle v \| e \rangle\{\theta,\gamma,\delta\} \in \perp\!\!\!\perp$.
- Act for type A: By the inductive hypothesis, we know that for all $E \in \llbracket A \rrbracket_{\mathcal{I}}$, $c\{\theta,\gamma,E/\alpha/\delta\} \in \perp\!\!\!\perp$. Therefore, $\mu\alpha.c\{\theta,\gamma,\delta\} = (\mu\alpha.c)\{\theta,\gamma,\delta\} \in \llbracket A \rrbracket_{\mathcal{I}}$ by strong activation (Lemma 11).
- $CoAct$ for type A: Analogous to the previous case by duality.
- Eq for types $A = B$: hold the soundness of type-level equality from Lemma 35, so that $\llbracket A \rrbracket_{\mathcal{I}} = \llbracket B \rrbracket_{\mathcal{I}}$.
- $CoEq$ for types $A = B$: Analogous to the previous case by duality.

Next, we have the type-specific left and right introduction rules. Remember that for the active types A in each case, because $\mathcal{H} \in \llbracket \mathcal{F} \rrbracket$ we know that $\langle\!\langle A \rangle\!\rangle_{\mathcal{I}} \sqsubseteq \llbracket A \rrbracket_{\mathcal{I}}$.

Suppose we have a data type

$$\mathbf{data}\,\mathsf{F}(\vec{a} : k)\,\mathbf{where}$$
$$\overrightarrow{\mathsf{K} : \vec{B} \vdash_{\overrightarrow{d:k'}} \mathsf{F}(\vec{X}) | \vec{C}}$$

Then if $\mathsf{K} : \vec{B} \vdash_{\overrightarrow{d:k'}} \mathsf{F}(\vec{a}) | \vec{C}$ is in $\overrightarrow{\mathsf{K} : \vec{B} \vdash_{\overrightarrow{d:k'}} \mathsf{F}(\vec{a}) | \vec{C}}$ We have the rule

$$\frac{\overrightarrow{\Theta \vdash D : k_i'\{\overrightarrow{A/a}\}} \quad \overrightarrow{\Gamma \vdash_\Theta v : B_i\{\overrightarrow{A/a}, \overrightarrow{D/d_i}\} | \Delta} \quad \overrightarrow{\Gamma | e : C_i\{\overrightarrow{A/a}, \overrightarrow{D/d_i}\} \vdash_\Theta \Delta}}{\Gamma \vdash_\Theta \mathsf{K}_i^{\vec{D}}(\vec{e}, \vec{v}) : \mathsf{F}\ \vec{A} | \Delta}$$

Now supposing that each of $\vec{e}$ and $\vec{v}$ are all (co-)values $\vec{E}$ and $\vec{V}$, by the inductive hypothesis, we know that

$$K_i^{\vec{D}}(\vec{E}, \vec{V})\{\theta,\gamma,\delta\} \in \langle\!\langle \mathsf{F}(\overrightarrow{\llbracket A \rrbracket_{\mathcal{I}}}) \rangle\!\rangle_{\mathcal{I}} \sqsubseteq \llbracket \mathsf{F}(\overrightarrow{\llbracket A \rrbracket_{\mathcal{I}}}) \rrbracket_{\mathcal{I}}$$

Furthermore, since

$$\overrightarrow{\llbracket B_i \rrbracket_{\mathcal{I}\{\overrightarrow{\llbracket A \rrbracket_{\mathcal{I}}/a}, \overrightarrow{\llbracket D \rrbracket_{\mathcal{I}}/d_i}\}}} = \overrightarrow{\llbracket B_i\{\overrightarrow{A/a}, \overrightarrow{D/d_i}\} \rrbracket_{\mathcal{I}}}$$
$$\overrightarrow{\llbracket C_i \rrbracket_{\mathcal{I}\{\overrightarrow{\llbracket A \rrbracket_{\mathcal{I}}/a}, \overrightarrow{\llbracket D \rrbracket_{\mathcal{I}}/d_i}\}}} = \overrightarrow{\llbracket C_i\{\overrightarrow{A/a}, \overrightarrow{D/d_i}\} \rrbracket_{\mathcal{I}}}$$

are all reducibility candidates, we must have that

$$K_i^{\vec{D}}(\vec{e}, \vec{v})\{\theta,\gamma,\delta\} \in \llbracket \mathsf{F}(\overrightarrow{\llbracket A \rrbracket_{\mathcal{I}}}) \rrbracket_{\mathcal{I}}$$

in general by unfocalization (Lemma 12).

For the left rule

$$\frac{c_1 : (\Gamma, \overrightarrow{x : B_1} \vdash_{\Theta, \overrightarrow{d_1 : l_1}} \Delta, \overrightarrow{\alpha : C_1})\{\overrightarrow{A/a}\} \quad \ldots}{\Gamma | \tilde{\mu}[\mathsf{K}_1^{\overrightarrow{d_1:l_1}}(\vec{\alpha}, \vec{x}).c_1 | \ldots]\{\overrightarrow{A/a}\} : \mathsf{F}\ \vec{A} \vdash_\Theta \Delta}$$

For each c_i, we have by inductive hypothesis that for any of $\theta'\ \llbracket \Theta, \overrightarrow{d_i : l_i} \rrbracket_{\mathcal{H}}\ \mathcal{I}'$, $\gamma' \in \llbracket \Gamma, \overrightarrow{x : B_i} \rrbracket_{\mathcal{I}'}$ and $\delta' \in \llbracket \Delta, \overrightarrow{\alpha : C_i} \rrbracket_{\mathcal{I}'}$, $c_i\{\theta',\gamma',\delta'\} \in \perp\!\!\!\perp$. Observe that given any choice of types $\overrightarrow{D\ \llbracket k_i' \rrbracket_{\mathcal{I}}\ \mathcal{D}}$, values $\overrightarrow{V \in \llbracket B_i \rrbracket_{\mathcal{I}\{\overrightarrow{\mathcal{D}/d_i}\}}}$ and co-values $\overrightarrow{E \in \llbracket C_i \rrbracket_{\mathcal{I}\{\overrightarrow{\mathcal{D}/d_i}\}}}$ then

$$\theta\{\overrightarrow{D/d_i}\}\ \llbracket \Theta, \overrightarrow{d_i : k_i'} \rrbracket_{\mathcal{H}}\ \mathcal{I}\{\overrightarrow{\mathcal{D}/d_i}\}$$
$$\gamma\{\overrightarrow{V/x}\} \in \llbracket \Gamma, \overrightarrow{x : B_i} \rrbracket_{\mathcal{I}\{\overrightarrow{\mathcal{D}/d_i}\}}$$
$$\delta\{\overrightarrow{E/\alpha}\} \in \llbracket \Delta, \overrightarrow{\alpha : C_i} \rrbracket_{\mathcal{I}\{\overrightarrow{\mathcal{D}/d_i}\}}$$

Thus,

$$\begin{aligned} &c_i\{\theta\{\overrightarrow{D/d_i}\}, \gamma\{\overrightarrow{V/x}\}, \delta\{\overrightarrow{E/\alpha}\}\} \\ &= c_i\{\theta,\gamma,\delta\}\{\overrightarrow{V/x}, \overrightarrow{E/\alpha}, \overrightarrow{D/d_i}\} \in \perp\!\!\!\perp \end{aligned}$$

From this we can see that

$$\tilde{\mu}[\mathsf{K}_1^{\overrightarrow{d_1:k_1'}}(\vec{\alpha}, \vec{x}).c_1\{\theta,\gamma,\delta\} | \ldots] \in \langle\!\langle \mathsf{F}(\vec{A}) \rangle\!\rangle_{\mathcal{I}} \sqsubseteq \llbracket \mathsf{F}(\vec{A}) \rrbracket_{\mathcal{I}}$$

by Lemma 25.

The rules for non-recursive co-data declarations are analogous by duality. Additionally, (co-)data type declarations defined by noetherian and primitive recursion also follow analogously.

The ordinary rules for Ascend and Descend are handled by viewing them as user defined co-data types. The remaining rule we need to look at is the recursive case abstraction.

$$\frac{c : \Gamma, x : A\ j \vdash_{\Theta, j<N} \alpha : \mathsf{Descend}(j, A), \Delta}{\Gamma | \tilde{\mu}[\mathsf{Fall}^{j<N}(x)[\alpha].c] : \mathsf{Descend}(N, A) \vdash_\Theta \Delta}$$

So that the inductive hypothesis fits the requirements of Lemma 32. Note that the recursive case abstraction for Ascend follows dually.

The recursive case abstraction rule for Deflate is

$$\frac{c_1 : \Gamma, x : A\ (j+1) \vdash_{\Theta, j:\mathsf{Ix}} \alpha : A\ j, \Delta \quad c_0 : \Gamma, x : A\ 0 \vdash_\Theta \Delta}{\Gamma | \tilde{\mu}[\mathsf{Down}^{0:\mathsf{Ix}}(x).c_0 | \mathsf{Down}^{j+1:\mathsf{Ix}}(x)[\alpha].c_1] : \mathsf{Deflate}(A) \vdash_\Theta \Delta}$$

and its soundness follows by Lemma 34 and the inductive hypothesis. Note that the recursive case abstraction for Inflate follows dually. □

Definition 9. *The set $\llparenthesis \Theta \rrparenthesis$ is the subset of the maps $HType \rightharpoonup \mathcal{U}$ which are "big enough" with respect to Θ. That is*

$$\begin{aligned} \llparenthesis \cdot \rrparenthesis &= \{\mathcal{H} : HType \rightharpoonup \mathcal{U}\} \\ \llparenthesis \Theta, i : k \rrparenthesis &= \{\mathcal{H} \in \llparenthesis \Theta \rrparenthesis \mid \mathcal{H}(i) < \mathcal{H}(j)\} \quad (k = \overrightarrow{k \to} < j) \\ \llparenthesis \Theta, i : k \rrparenthesis &= \llparenthesis \Theta \rrparenthesis \quad \text{(otherwise)} \end{aligned}$$

Lemma 37. *For all Θ, $\llparenthesis \Theta \rrparenthesis$ is inhabited.*

Proof. By induction. We can always set $\mathcal{H}(0)$ to be the length of Θ, which covers the worst case where we have $\Theta = i_{n-1} < 0, i_{n-2} < i_{n-1}, \ldots, i_0 < i_1$ by assigning 0 to $i_1, \ldots, n-2$ to i_{n-2}, and $n-1$ to i_{n-1}. More specifically, whenever we see a $i < M$ in Θ, we can assign i a value based on its position in Θ. □

Lemma 38. *For every Θ and well-founded $\mathcal{F}$, there exists a $\mathcal{H} \in \llparenthesis \Theta \rrparenthesis$ such that $\mathcal{H} \in \llbracket \mathcal{F} \rrbracket$*

Proof. By composition of the previous lemma with Lemma 17. □

Lemma 39. *If $\mathcal{H} \in \llparenthesis \Theta \rrparenthesis$ and $\mathcal{H} \in \llbracket \mathcal{F} \rrbracket$ then there exists $\theta\ \llbracket \Theta \rrbracket_{\mathcal{H}}\ \mathcal{I}$*

Proof. By induction on Θ. □

Corollary 9. *If $\mathcal{H} \in \llparenthesis \Theta \rrparenthesis$ and $\mathcal{H} \in \llbracket \mathcal{F} \rrbracket$ then $\llbracket c : (\Gamma \vdash_\Theta \Delta) \rrbracket_{\mathcal{H}}$ implies $c \in \perp\!\!\!\perp$. Similarly, $\llbracket \Gamma \vdash_\Theta v : A | \Delta \rrbracket_{\mathcal{H}}$ implies $v \in \mathcal{W}$ and $\llbracket \Gamma | e : A \vdash_\Theta \Delta \rrbracket_{\mathcal{H}}$ implies $e \in \mathcal{W}$.*

Theorem 3. *If $\mathcal{F}$ is well-founded, then*

1. *If $c : \Gamma \vdash_\Theta \Delta$ and $(\Gamma \vdash_\Theta \Delta)\,\mathbf{seq}$ are derivable in $\mu\tilde{\mu}_{\mathcal{S}}^{\mathcal{F}}$, then c is strongly normalizing in the $\mu\tilde{\mu}_{\mathcal{S}}^{\mathcal{F}}$-calculus.*
2. *If $\Gamma \vdash_\Theta v : A | \Delta$ and $(\Gamma \vdash_\Theta \alpha : A, \Delta)\,\mathbf{seq}$ are derivable in $\mu\tilde{\mu}_{\mathcal{S}}^{\mathcal{F}}$, then v is strongly normalizing in the $\mu\tilde{\mu}_{\mathcal{S}}^{\mathcal{F}}$-calculus.*
3. *If $\Gamma | e : A \vdash_\Theta \Delta$ and $(\Gamma \vdash_\Theta \Delta)\,\mathbf{seq}$ are derivable in $\mu\tilde{\mu}_{\mathcal{S}}^{\mathcal{F}}$, then e is strongly normalizing in the $\mu\tilde{\mu}_{\mathcal{S}}^{\mathcal{F}}$-calculus.*

Proof. Since there exists $\mathcal{H} \in (\!|\Theta|\!)$ and $\mathcal{H} \in [\![\mathcal{F}]\!]$ we have $[\![c : (\Gamma \vdash_\Theta \Delta)]\!]_{\mathcal{H}}$ by Theorem 2 and we know that $c : (\Gamma \vdash_\Theta \Delta)$ implies that $c \in\perp\!\!\!\perp$ by the previous corollary. The cases for (co-)terms follows analogously. $\square$

Note that type-erasure preserves strong normalization precisely because the type-erased rewriting theory is strictly weaker than rewriting the pre-erased programs.

Lemma 40. *If c is strongly normalizing in the $\mu\tilde{\mu}^{\mathcal{F}}_{\mathcal{S}}$-calculus, then $Erase(c)$ is strongly normalizing in the type-erased $\mu\tilde{\mu}^{\mathcal{F}}_{\mathcal{S}}$-calculus, and similarly for (co-)terms.*

Proof. The $Erase$ operation removes all erasable types, with kinds inhabiting $\square$, from commands and (co-)terms. In particular, we remove the type-level content of constructors $\mathsf{K}^{\vec{D}}(\vec{e}, \vec{v})$ and patterns $\mathsf{K}^{\overrightarrow{d:k}}(\vec{\alpha}, \vec{x})$, leaving only those types with the non-erasable kind Ix. Furthermore, because the caveat for reducing inside the branches of a case abstraction are strictly more limiting in the type-erased $\mu\tilde{\mu}^{\mathcal{F}}_{\mathcal{S}}$-calculus, we know that

- If $Erase(c) \to c'$ in the type-erased $\mu\tilde{\mu}^{\mathcal{F}}_{\mathcal{S}}$-calculus, there is a c'' such that $Erase(c'') = c'$ and $c \to c''$ in the $\mu\tilde{\mu}^{\mathcal{F}}_{\mathcal{S}}$-calculus.
- If $Erase(v) \to v'$ in the type-erased $\mu\tilde{\mu}^{\mathcal{F}}_{\mathcal{S}}$-calculus, there is a v'' such that $Erase(v'') = v'$ and $v \to v''$ in the $\mu\tilde{\mu}^{\mathcal{F}}_{\mathcal{S}}$-calculus.
- If $Erase(e) \to e'$ in the type-erased $\mu\tilde{\mu}^{\mathcal{F}}_{\mathcal{S}}$-calculus, there is a e'' such that $Erase(e'') = e'$ and $e \to e''$ in the $\mu\tilde{\mu}^{\mathcal{F}}_{\mathcal{S}}$-calculus.

Therefore, because the chosen c, v, e are strongly normalizing in the $\mu\tilde{\mu}^{\mathcal{F}}_{\mathcal{S}}$-calculus, their erasure $Erase(c), Erase(v), Erase(e)$ must also be strongly normalizing in the type-erased $\mu\tilde{\mu}^{\mathcal{F}}_{\mathcal{S}}$-calculus. $\square$

H. Natural Deduction Embedding

To show that the natural deduction calculus for effect-free functional programs is strongly normalizing, we demonstrate that:

1. the translation into the call-by-name $\mu\tilde{\mu}_{\mathcal{N}}$-calculus is type-preserving for well-typed terms,
2. each reduction of a natural deduction term corresponds to at least one reduction in $\mu\tilde{\mu}_{\mathcal{N}}$.

Lemma 41. *If $\Gamma \vdash_\Theta v : A$ is derivable then $\Gamma \vdash_\Theta v^\flat : A|$ is derivable.*

Proof. By induction on the structure of the derivation for $\Gamma \vdash_\Theta v : A$. $\square$

Note that $\twoheadrightarrow^+$ denotes the transitive, but *not* reflexive, closure of $\to$.

Lemma 42. *If $v \to v'$ in the natural deduction calculus then $v^\flat \twoheadrightarrow^+ v'^\flat$.*

Proof. By cases on the reductions in the natural deduction calculus. Note that $v^\flat\{v'^\flat/x\} = (v\{v'/x\})^\flat$ and $v^\flat\{B/b\} = (v\{B/b\})^\flat$.

- $\{\mathsf{H}^{\overrightarrow{b:k}}[\vec{x}] \Rightarrow v' | \ldots\}.\,\mathsf{H}^{\vec{B}}[\vec{v}] \to v'\{\overrightarrow{B/b}, \overrightarrow{v/x}\}$

$$\begin{aligned}
&\{\mathsf{H}^{\overrightarrow{b:k}}[\vec{x}] \Rightarrow v' | \ldots\}.\,\mathsf{H}^{\vec{B}}[\vec{v}]^\flat\\
&= \mu\alpha.\langle \mu(\mathsf{H}^{\overrightarrow{b:k}}[\vec{x}, \beta].\langle v'^\flat \| \beta\rangle | \ldots) \| \mathsf{H}^{\vec{B}}[\overrightarrow{v^\flat}, \alpha]\rangle\\
&\to \mu\alpha.\langle v'^\flat\{\overrightarrow{B/b}, \overrightarrow{v^\flat/x}\} \| \alpha\rangle\\
&\to v'^\flat\{\overrightarrow{B/b}, \overrightarrow{v^\flat/x}\}\\
&= (v'\{\overrightarrow{B/b}, \overrightarrow{v/x}\})^\flat
\end{aligned}$$

- **case** $\mathsf{K}^{\vec{B}}(\vec{v})$ **of** $\mathsf{K}^{\overrightarrow{b:k}}(\vec{x}) \Rightarrow v' | \ldots \to v'\{\overrightarrow{B/b}, \overrightarrow{v/x}\}$

$$\begin{aligned}
&\textbf{case } \mathsf{K}^{\vec{B}}(\vec{v}) \textbf{ of } \mathsf{K}^{\overrightarrow{b:k}}(\vec{x}) \Rightarrow v' | \ldots^\flat\\
&= \mu\alpha.\langle \mathsf{K}^{\vec{B}}(\overrightarrow{v^\flat}) \| \tilde{\mu}[\mathsf{K}^{\overrightarrow{b:k}}(\vec{x}).\langle v'^\flat \| \alpha\rangle | \ldots]\rangle\\
&\to \mu\alpha.\langle v'^\flat\{\overrightarrow{B/b}, \overrightarrow{v^\flat/x}\} \| \alpha\rangle\\
&\to v'^\flat\{\overrightarrow{B/b}, \overrightarrow{v^\flat/x}\}\\
&= (v'\{\overrightarrow{B/b}, \overrightarrow{v/x}\})^\flat
\end{aligned}$$

- $\{\mathsf{Rise}^{j<N}(x) \Rightarrow v\}$
 $\to \{\mathsf{Rise}^{i<N}(x) \Rightarrow v\{i/j, \{\mathsf{Rise}^{j<i}(x) \Rightarrow v\}/x\}\}$

$$\begin{aligned}
&\{\mathsf{Rise}^{j<N}(x) \Rightarrow v\}^\flat\\
&= \mu(\mathsf{Rise}^{j<N}[\alpha](x).\langle v^\flat \| \alpha\rangle)\\
&\to \mu(\mathsf{Rise}^{i<N}[\alpha].\langle v^\flat\{i/j, \mu(\mathsf{Rise}^{j<i}[\alpha](x).\langle v^\flat \| \alpha\rangle)/x\} \| \alpha\rangle)\\
&= \mu(\mathsf{Rise}^{i<N}[\alpha].\langle v^\flat\{i/j, \{\mathsf{Rise}^{j<i}(x) \Rightarrow v\}^\flat/x\} \| \alpha\rangle)\\
&= \mu(\mathsf{Rise}^{i<N}[\alpha].\langle (v\{i/j, \{\mathsf{Rise}^{j<i}(x) \Rightarrow v\}/x\})^\flat \| \alpha\rangle)\\
&= \{\mathsf{Rise}^{i<N} \Rightarrow v\{i/j, \{\mathsf{Rise}^{j<i}(x) \Rightarrow v\}/x\}\}^\flat
\end{aligned}$$

- $\{\mathsf{Up}^0 \Rightarrow v_0 | \mathsf{Up}^{j+1}(x) \Rightarrow v_1\}.\,\mathsf{Up}^0 \to v_0$

$$\begin{aligned}
&(\{\mathsf{Up}^0 \Rightarrow v_0 | \mathsf{Up}^{j+1}(x) \Rightarrow v_1\}.\,\mathsf{Up}^0)^\flat\\
&= \mu\alpha.\langle \mu(\mathsf{Up}^0[\beta].\langle v_0^\flat \| \beta\rangle | \mathsf{Up}^{j+1}[\beta](x).\langle v_1^\flat \| \beta\rangle) \| \mathsf{Up}^0[\alpha]\rangle\\
&\to \mu\alpha.\langle v_0^\flat \| \alpha\rangle\\
&\to v_0^\flat
\end{aligned}$$

- $\{\mathsf{Up}^0 \Rightarrow v_0 | \mathsf{Up}^{j+1}(x) \Rightarrow v_1\}.\mathsf{Up}^{M+1}$
 $\to v_1\{M/j, \{\mathsf{Up}^0 \Rightarrow v_0 | \mathsf{Up}^{j+1}(x) \Rightarrow v_1\}.\mathsf{Up}^M/x\}$
 Let V and $V^\flat$ be

$$\begin{aligned}
V &= \{\mathsf{Up}^0 \Rightarrow v_0 | \mathsf{Up}^{j+1}(x) \Rightarrow v_1\}\\
V^\flat &= \mu(\mathsf{Up}^0[\beta].\langle v_0^\flat \| \beta\rangle | \mathsf{Up}^{j+1}[\beta](x).\langle v_1^\flat \| \beta\rangle)
\end{aligned}$$

 in the following:

$$\begin{aligned}
&(\{\mathsf{Up}^0 \Rightarrow v_0 | \mathsf{Up}^{j+1}(x) \Rightarrow v_1\}.\,\mathsf{Up}^{M+1})^\flat\\
&= \mu\alpha.\langle V^\flat \| \mathsf{Up}^{M+1}[\alpha]\rangle\\
&\to \mu\alpha.\langle \mu\beta.\langle V^\flat \| \mathsf{Up}^M[\beta]\rangle \| \tilde{\mu}x.\langle v_1^\flat\{M/j\} \| \alpha\rangle\rangle\\
&\to \mu\alpha.\langle v_1^\flat\{M/j, \mu\beta.\langle V^\flat \| \mathsf{Up}^M[\beta]\rangle/x\} \| \alpha\rangle\\
&\to v_1^\flat\{M/j, \mu\beta.\langle V^\flat \| \mathsf{Up}^M[\beta]\rangle/x\}\\
&= v_1^\flat\{M/j, (V.\,\mathsf{Up}^M)^\flat/x\}\\
&= (v_1\{M/j, V.\,\mathsf{Up}^M\ /x\})^\flat
\end{aligned}$$

- **loop** $\mathsf{Down}^0(v)$ **of** $\mathsf{Down}^0(x) \Rightarrow v_0 | \mathsf{Down}^{j+1}(x) \Rightarrow v_1$
 $\to v_0$

$$\begin{aligned}
&(\textbf{loop } \mathsf{Down}^0(v) \textbf{ of } \mathsf{Down}^0(x) \Rightarrow v_0 | \mathsf{Down}^{j+1}(x) \Rightarrow v_1)^\flat\\
&= \mu\alpha.\langle \mathsf{Down}^0(v^\flat) \| \tilde{\mu}[\mathsf{Down}^0(x).\langle v_0^\flat \| \alpha\rangle | \mathsf{Down}^{j+1}(x)[\beta].\langle v_1^\flat \| \beta\rangle]\rangle\\
&\to \mu\alpha.\langle v_0^\flat\{v^\flat/x\} \| \alpha\rangle\\
&\to v_0^\flat\{v^\flat/x\}\\
&= (v_0\{v/x\})^\flat
\end{aligned}$$

- $\mathbf{loop}\ \mathsf{Down}^{M+1}(v)\ \mathbf{of}\ \mathsf{Down}^0(x) \Rightarrow v_0 |\ \mathsf{Down}^{j+1}(x) \Rightarrow v_1$

  $\to \mathbf{loop}\ \mathsf{Down}^{M}(v_1\{M/j, v/x\})\ \mathbf{of}$

  $\quad \mathsf{Down}^0(x) \Rightarrow v_0$

  $\quad |\ \mathsf{Down}^{j+1}(x) \Rightarrow v_1$

  Let E_α be

  $$\tilde{\mu}[\mathsf{Down}^0(x).\langle v_0^\flat \| \alpha\rangle | \mathsf{Down}^{j+1}(x)[\beta].\langle v_1^\flat \| \beta\rangle]$$

  in the following:

  $(\mathbf{loop}\ \mathsf{Down}^{M+1}(v)\ \mathbf{of}\ \mathsf{Down}^0(x) \Rightarrow v_0 |\ \mathsf{Down}^{j+1}(x) \Rightarrow v_1)^\flat$

  $= \mu\alpha.\langle \mathsf{Down}^{M+1}(v^\flat) \| E_\alpha\rangle$

  $\to \mu\alpha.\langle \mu\beta.\langle v_1^\flat\{M/j, v^\flat/x\} \| \beta\rangle \| \tilde{\mu}y.\langle \mathsf{Down}^M(x) \| E_\alpha\rangle\rangle$

  $\to \mu\alpha.\langle \mathsf{Down}^M(\mu\beta.\langle v_1^\flat\{M/j, v^\flat/x\} \| \beta\rangle) \| E_\alpha\rangle$

  $\to \mu\alpha.\langle \mathsf{Down}^M(v_1^\flat\{M/j, v^\flat/x\}) \| E_\alpha\rangle$

  $= \mu\alpha.\langle \mathsf{Down}^M(v_1\{M/j, v/x\})^\flat \| E_\alpha\rangle$

  $= (\mathbf{case}\ \mathsf{Down}^M(v_1\{M/j, v/x\})\ \mathbf{of}$

  $\quad \mathsf{Down}^0(x) \Rightarrow v_0$

  $\quad |\ \mathsf{Down}^{j+1}(x) \Rightarrow v_1)^\flat$

Cases where reduction occurs inside of a larger context follow from compositionality of the translation $(-)^\flat$. □

Theorem 4. *If* $\Gamma \vdash_\Theta v : A$ *and* $(\Gamma \vdash_\Theta \alpha : A)\ \mathbf{seq}$ *are derivable then* v *is strongly normalizing.*

Proof. By Lemma 41 we know that $\Gamma \vdash_\Theta v^\flat : A|$, and so by Theorem 3 $v^\flat$ is a strongly normalizing term of the $\mu\tilde{\mu}_{\mathcal{N}}$-calculus. Now, suppose that there is an infinite reduction path from v. By applying Lemma 42 over the steps of this infinite reduction path from v and composing them together, we obtain an infinite reduction path from $v^\flat$, which is a contradiction. Therefore, there is no infinite reduction path from v, so v is also strongly normalizing. □

References

[1] F. Barbanera and S. Berardi. A symmetric lambda calculus for "classical" program extraction. In *TACS '94*, pages 495–515, 1994.

[2] P.-L. Curien and G. Munch-Maccagnoni. The duality of computation under focus. *Theoretical Computer Science*, pages 165–181, 2010.

[3] P. Di Gianantonio and M. Miculan. A unifying approach to recursive and co-recursive definitions. In *TYPES*, volume 2646 of *LNCS*. 2003.

[4] J.-Y. Girard, P. Taylor, and Y. Lafont. *Proofs and types*. Cambridge University Press, 1989.

[5] S. Lengrand and A. Miquel. Classical Fω, orthogonality and symmetric candidates. *Annals of Pure and Applied Logic*, 153(1):3–20, 2008.

[6] B. Liskov. Keynote address - data abstraction and hierarchy. In *OOPSLA '87*, 1987.

[7] G. Munch-Maccagnoni. Focalisation and classical realisability. In *Computer Science Logic*, pages 409–423. Springer, 2009.

[8] A. Pitts. A note on logical relations between semantics and syntax. *Logic Journal of IGPL*, 5(4):589–601, 1997.

[9] N. Zeilberger. On the unity of duality. *Annals of Pure Applied Logic*, 153(1-3):66–96, 2008.