Soundness for DOT^{ω} (update 2023-05-26)

Cameron Wong

2023-05-26

1 Last time

The key to this proof of soundness is in the equivalence of DOT's regular typing rules to a restricted set of *tight typing* rules which are far better behaved. Previously, we demonstrated how tight typing can be generalized to a higher-kinded setting by adjusting the beta-reduction rules to enforce that the lambda body is still well-kinded with the concrete argument, avoiding the impossible reductions that are usually permitted by bad bounds.

2 Normalization of types

The interesting change in showing type soundness in a higher-kinded system is in the proof of progress, which requires deducing the form of a value from its type. Specifically, the type of a term may be itself a complex beta redex that is equivalent to one with a canonical form. To resolve this, we must first show that every well-kinded type can indeed be reduced to such a form.

An important observation is that, in inert contexts, all abstract types (type members and type variables) must be singleton kinds. This means that an inert context can also serve as a variable store for type-level evaluation (via Lemma 1), meaning we can use a logical-relation based approach using inert contexts Γ as evaluation heaps.

A straightforward attempt may look like so:

$$\begin{aligned}
[\![S..U]\!] &= \{\langle \Gamma, \tau\rangle : \Gamma \text{ inert} \wedge \Gamma \vdash S \le \tau \le U : *\} \\
[\![\Pi(X : J).K]\!] &= \{\langle \Gamma, \lambda(X : J).A\rangle : \forall B \in [\![J]\!].\Gamma \text{ inert} \wedge \langle (\Gamma, X : S(B : K)), A\rangle \in \mathcal{E}[\![K]\!]\} \\
\mathcal{E}[\![K]\!] &= \{\langle \Gamma, A\rangle : \exists V.V \text{ normal} \wedge \Gamma \vdash A == V : K\}
\end{aligned}$$

However, with this definition, it is not true that $\Gamma \vdash \lambda(X : J).A : \Pi(X : J).K$ implies that $\langle \Gamma, \lambda(X : J).A\rangle \in \mathcal{E}[\![\Pi(X : J).K]\!]$ when Γ is inert. The culprit is the possibility of bad bounds in the parameter kind J, leading to the evaluation of the function body getting stuck with a type operator of the wrong form.

I'm 95% sure that this doesn't actually break the statement – in order to reach kind $*$, we *must* have some witness that the kind J is good, so it's just a matter

of designing the relation to pass this evidence through properly, but I wasn't able to come up with a good one.

We could attempt to use the same method as Tiark (indexing the relation with upper and lower bounds), but it would likely require lots of reworking – in System D-sub, the bounds are a different syntactic sort (types) as the thing being reduced (terms), but here the bounds are actually the same sort (types vs kinds). It also is not obvious to me how to generalize it to function kinds.

3 The Rest

If types can be strongly normalized, then we can recover Rapoport et al. [2]'s proofs of progress and preservation by simply normalizing types before proceeding with the same induction. Importantly, as progress and preservation are expressed from an empty context, this can always be done by using the concrete good-bounds witnesses to construct the necessary inert contexts.

If it can't be shown that types are strongly normalizing, we may be able to get away with weak-head normalization, as that's all that's really necessary for canonical forms. This will likely require a more involved progress and preservation proof that interleaves weak-head steps with progress.

A DOT^{ω} Full rules

$$\frac{}{\varnothing\ \texttt{ctx}} \qquad \frac{\Gamma\ \texttt{ctx} \qquad \Gamma \vdash K\ \texttt{kd}}{\Gamma, X : K\ \texttt{ctx}} \qquad \frac{\Gamma\ \texttt{ctx} \qquad \Gamma \vdash A : *}{\Gamma, x : A\ \texttt{ctx}}$$

Figure 1: Context formation

$$\frac{\Gamma \vdash S : * \qquad \Gamma \vdash U : *}{\Gamma \vdash S..U\ \texttt{kd}}\ \textsc{Wf-Intv} \qquad \frac{\Gamma \vdash J\ \texttt{kd} \qquad \Gamma, X : J \vdash K\ \texttt{kd}}{\Gamma \vdash \Pi(X : J).K\ \texttt{kd}}\ \textsc{Wf-DArr}$$

Figure 2: Kind formation

$$\frac{\Gamma \vdash S_2 \leq S_1 : * \qquad \Gamma \vdash U_1 \leq U_2 : *}{\Gamma \vdash S_1..U_1 \leq S_2..U_2}\ \text{SK-}\textsc{Intv}$$

$$\frac{\Gamma \vdash \Pi(X : J_1).K_1\ \texttt{kd} \qquad \Gamma \vdash J_2 \leq J_1 \qquad \Gamma, X : J_2 \vdash K_1 \leq K_2}{\Gamma \vdash \Pi(X : J_1).K_1 \leq \Pi(X : J_2).K_2}\ \text{SK-}\textsc{DArr}$$

Figure 3: Subkinding

$$\frac{\Gamma, X : K\ \texttt{ctx}}{\Gamma, X : K \vdash X : K}\ \textsc{K-Var} \qquad \frac{}{\Gamma \vdash \top : *}\ \textsc{K-Top} \qquad \frac{}{\Gamma \vdash \bot : *}\ \textsc{K-Bot}$$

$$\frac{\Gamma \vdash A : S..U}{\Gamma \vdash A : A..A}\ \textsc{K-Sing} \qquad \frac{\Gamma \vdash A : * \qquad \Gamma, x : A \vdash B : *}{\Gamma \vdash (x : A) \to B : *}\ \textsc{K-Arr}$$

$$\frac{\Gamma \vdash J\ \texttt{kd} \qquad \Gamma, X : J \vdash A : K \qquad \Gamma, X : J \vdash K\ \texttt{kd}}{\Gamma \vdash \lambda(X : J).A : \Pi(X : J).K}\ \textsc{K-Abs}$$

$$\frac{\Gamma \vdash A : \Pi(X : J).K \qquad \Gamma \vdash B : J \qquad \Gamma, X : J \vdash K\ \texttt{kd} \qquad \Gamma \vdash K[B/X]\ \texttt{kd}}{\Gamma \vdash A\ B : K[B/X]}\ \textsc{K-App}$$

$$\frac{\Gamma \vdash A : S_1..U_1 \qquad \Gamma \vdash B : S_2..U_2}{\Gamma \vdash A \wedge B : S_1 \vee S_2..U_1 \wedge U_2}\ \textsc{K-Intersect}$$

$$\frac{\Gamma \vdash A : S..U}{\Gamma \vdash \{\mathbf{val}\ \ell : A\} : *}\ \textsc{K-Field} \qquad \frac{\Gamma \vdash K\ \texttt{kd}}{\Gamma \vdash \{\mathbf{type}\ M : K\} : *}\ \textsc{K-Typ}$$

$$\frac{\Gamma \vdash x : \{\mathbf{type}\ M : K\}}{\Gamma \vdash x.M : K}\ \textsc{K-Typ-Mem} \qquad \frac{\Gamma, x : \tau \vdash \tau : K}{\Gamma \vdash \mu(x.\tau) : K}\ \textsc{K-Rec}$$

$$\frac{\Gamma \vdash A : J \qquad \Gamma \vdash J \leq K}{\Gamma \vdash A : K}\ \textsc{K-Sub}$$

Figure 4: Kind assignment

Note that K-Intersect rules refers to the union type $S_1 \vee S_2$, despite no such construct being present in the language as a whole. I am currently investigating whether the explicit addition of this construct is necessary.

$$\frac{\Gamma \vdash A : K}{\Gamma \vdash A \leq A : K}\ \text{ST-refl} \qquad \frac{\Gamma \vdash A \leq B : K \qquad \Gamma \vdash B \leq C : K}{\Gamma \vdash A \leq C : K}\ \text{ST-trans}$$

$$\frac{\Gamma \vdash A : S..U}{\Gamma \vdash A \leq \top : *}\ \text{ST-top} \qquad \frac{\Gamma \vdash A : S..U}{\Gamma \vdash \bot \leq A : *}\ \text{ST-bot}$$

$$\frac{\Gamma \vdash A \wedge B : K}{\Gamma \vdash A \wedge B \leq A : K}\ \text{ST-and-}\ell_1 \qquad \frac{\Gamma \vdash A \wedge B : K}{\Gamma \vdash A \wedge B \leq B : K}\ \text{ST-and-}\ell_2$$

$$\frac{\Gamma \vdash S \leq A : K \qquad \Gamma \vdash S \leq B : K}{\Gamma \vdash S \leq A \wedge B : K}\ \text{ST-and-r}$$

$$\frac{\Gamma \vdash A \leq B : *}{\Gamma \vdash \{\mathbf{val}\ \ell : A\} \leq \{\mathbf{val}\ \ell : B\} : *}\ \text{ST-field}$$

$$\frac{\Gamma \vdash J \leq K}{\Gamma \vdash \{\mathbf{type}\ M : J\} \leq \{\mathbf{type}\ M : K\} : *}\ \text{ST-typ}$$

$$\frac{\Gamma Z : J \vdash A : K \qquad \Gamma \vdash B : J}{\Gamma \vdash (\lambda(Z : J).A)\ B \leq A[Z/B] : K[Z/B]}\ \text{ST-}\beta_1$$

$$\frac{\Gamma Z : J \vdash A : K \qquad \Gamma \vdash B : J}{\Gamma \vdash A[Z/B] \leq (\lambda(Z : J).A)\ B : K[Z/B]}\ \text{ST-}\beta_2$$

Figure 5: Subtyping

$$\frac{\Gamma \vdash A \leq B : K \qquad \Gamma \vdash B \leq A : K}{\Gamma \vdash A = B : K}\ \text{Eq}$$

Figure 6: Type equality

$$\frac{\Gamma, x : \tau \ \texttt{ctx}}{\Gamma, x : \tau \vdash x : \tau} \ \textsc{Ty-Var}$$

$$\frac{\Gamma \vdash e_1 : \tau \qquad \Gamma, x : \tau \vdash e_2 : \rho \qquad x \notin fv(\rho)}{\Gamma \vdash \text{let } x = e_1 \text{ in } e_2 : \rho} \ \textsc{Ty-Let}$$

$$\frac{\Gamma, x : \tau \vdash e : \rho}{\Gamma \vdash \lambda(x : \tau).e : (x : \tau) \to \rho} \ \textsc{Ty-Fun-I}$$

$$\frac{\Gamma \vdash x : (z : \tau) \to \rho \qquad \Gamma \vdash y : \tau}{\Gamma \vdash x \ y : \rho[z/y]} \ \textsc{Ty-Fun-E} \qquad \frac{\Gamma \vdash x : \tau}{\Gamma \vdash x : \mu(x : \tau)} \ \textsc{Ty-}\mu\textsc{-I}$$

$$\frac{\Gamma \vdash x : \mu(z : \tau)}{\Gamma \vdash x : \tau[z/x]} \ \textsc{Ty-}\mu\textsc{-E} \qquad \frac{\Gamma, x : \tau \vdash d : \tau}{\Gamma \vdash \nu(x : \tau)d : \mu(x : \tau)} \ \textsc{Ty-Rec-I}$$

$$\frac{\Gamma, x : \{\textbf{val } \ell : \tau\}}{\Gamma \vdash x.\ell : \tau} \ \textsc{Ty-Rec-E} \qquad \frac{\Gamma \vdash x : \tau_1 \qquad \Gamma \vdash x : \tau_2}{\Gamma \vdash x : \tau_1 \wedge \tau_2} \ \textsc{Ty-And-I}$$

$$\frac{\Gamma \vdash e : \tau_1 \qquad \Gamma \vdash \tau_1 \leq \tau_2 : *}{\Gamma \vdash e : \tau_2} \ \textsc{Ty-Sub}$$

$$\frac{\Gamma \vdash e : \rho}{\Gamma \vdash \{\textbf{val } \ell = e\} : \{\textbf{val } \ell : \rho\}} \ \textsc{Ty-Def-Trm}$$

$$\frac{\Gamma \vdash \tau : K}{\Gamma \vdash \{\textbf{type } M = A\} : \{\textbf{type } M : S(A : K)\}} \ \textsc{Ty-Def-Typ}$$

Figure 7: Type assignment

B Tight typing rules

In most cases, tight typing is merely forwarded to the premises. In any rule that extends the context with possibly-untrusted bounds, tight typing reverts to general typing.

$$\frac{}{\varnothing\ \texttt{ctx}_{\#}} \qquad \frac{\Gamma\ \texttt{ctx}_{\#} \qquad \Gamma \vdash_{\#} K\ \texttt{kd}}{\Gamma, X : K\ \texttt{ctx}} \qquad \frac{\Gamma\ \texttt{ctx}_{\#} \qquad \Gamma \vdash_{\#} A : *}{\Gamma, x : A\ \texttt{ctx}}$$

Figure 8: Context formation

$$\frac{\Gamma \vdash_{\#} S : * \qquad \Gamma \vdash_{\#} U : *}{\Gamma \vdash_{\#} S..U\ \texttt{kd}}\ \textsc{Wf-Int-}\#$$

$$\frac{\Gamma \vdash_{\#} J\ \texttt{kd} \qquad \Gamma, X : J \vdash_{\#} K\ \texttt{kd}}{\Gamma \vdash_{\#} \Pi(X : J).K\ \texttt{kd}}\ \textsc{Wf-DArr-}\#$$

Figure 9: Kind formation

$$\frac{\Gamma \vdash_{\#} S_2 \leq S_1 : * \qquad \Gamma \vdash_{\#} U_1 \leq U_2 : *}{\Gamma \vdash_{\#} S_1..U_1 \leq S_2..U_2}\ \textsc{SK-Intv-}\#$$

$$\frac{\Gamma \vdash_{\#} \Pi(X : J_1).K_1\ \texttt{kd} \qquad \Gamma \vdash_{\#} J_2 \leq J_1 \qquad \Gamma, X : J_2 \vdash K_1 \leq K_2}{\Gamma \vdash_{\#} \Pi(X : J_1).K_1 \leq \Pi(X : J_2).K_2}\ \textsc{SK-DArr-}\#$$

Figure 10: Subkinding

$$\frac{\Gamma, X : K\ \mathtt{ctx}_{\#}}{\Gamma, X : K \vdash_{\#} X : K}\ \textsc{K-Var-\#} \qquad \frac{}{\Gamma \vdash_{\#} \top : *}\ \textsc{K-Top-\#}$$

$$\frac{}{\Gamma \vdash_{\#} \bot : *}\ \textsc{K-Bot-\#} \qquad \frac{\Gamma \vdash_{\#} A : S..U}{\Gamma \vdash_{\#} A : A..A}\ \textsc{K-Sing-\#}$$

$$\frac{\Gamma \vdash_{\#} A : * \qquad \Gamma, x : A \vdash B : *}{\Gamma \vdash_{\#} (x : A) \to B : *}\ \textsc{K-Arr-\#}$$

$$\frac{\Gamma \vdash_{\#} J\ \mathtt{kd} \qquad \Gamma, X : J \vdash A : K \qquad \Gamma, X : J \vdash_{\#} K\ \mathtt{kd}}{\Gamma \vdash_{\#} \lambda(X : J).A : \Pi(X : J).K}\ \textsc{K-Abs-\#}$$

$$\frac{\Gamma \vdash_{\#} A : \Pi(X : J).K \qquad \Gamma \vdash_{\#} B : J \qquad \Gamma, X : J \vdash K\ \mathtt{kd} \qquad \Gamma \vdash_{\#} K[B/X]\ \mathtt{kd}}{\Gamma \vdash_{\#} A\ B : K[B/X]}\ \textsc{K-App-\#}$$

$$\frac{\Gamma \vdash_{\#} A : S_1..U_1 \qquad \Gamma \vdash_{\#} B : S_2..U_2}{\Gamma \vdash_{\#} A \wedge B : S_1 \vee S_2..U_1 \wedge U_2}\ \textsc{K-Intersect-\#}$$

$$\frac{\Gamma \vdash_{\#} A : S..U}{\Gamma \vdash_{\#} \{\mathbf{val}\ \ell : A\} : *}\ \textsc{K-Field-\#} \qquad \frac{\Gamma \vdash_{\#} K\ \mathtt{kd}}{\Gamma \vdash_{\#} \{\mathbf{type}\ M : K\} : *}\ \textsc{K-Typ-\#}$$

$$\frac{\Gamma \vdash_{!} x : \{\mathbf{type}\ M : S(A : K)\}}{\Gamma \vdash_{\#} x.M : S(A : K)}\ \textsc{K-Typ-Mem-\#} \qquad \frac{\Gamma, x : \tau \vdash \tau : K}{\Gamma \vdash_{\#} \mu(x.\tau) : K}\ \textsc{K-Rec-\#}$$

$$\frac{\Gamma \vdash_{\#} A : J \qquad \Gamma \vdash_{\#} J \leq K}{\Gamma \vdash_{\#} A : K}\ \textsc{K-Sub-\#}$$

Figure 11: Kind assignment

$$\frac{\Gamma \vdash_{\#} A : K}{\Gamma \vdash_{\#} A \leq A : K}\ \text{ST-REFL-}\#$$

$$\frac{\Gamma \vdash_{\#} A \leq B : K \qquad \Gamma \vdash_{\#} B \leq C : K}{\Gamma \vdash_{\#} A \leq C : K}\ \text{ST-TRANS-}\#$$

$$\frac{\Gamma \vdash_{\#} A : S..U}{\Gamma \vdash_{\#} A \leq \top : *}\ \text{ST-TOP-}\# \qquad \frac{\Gamma \vdash_{\#} A : S..U}{\Gamma \vdash_{\#} \bot \leq A : *}\ \text{ST-BOT-}\#$$

$$\frac{\Gamma \vdash_{\#} A \wedge B : K}{\Gamma \vdash_{\#} A \wedge B \leq A : K}\ \text{ST-AND-}\ell_1\text{-}\# \qquad \frac{\Gamma \vdash_{\#} A \wedge B : K}{\Gamma \vdash_{\#} A \wedge B \leq B : K}\ \text{ST-AND-}\ell_2\text{-}\#$$

$$\frac{\Gamma \vdash_{\#} S \leq A : K \qquad \Gamma \vdash_{\#} S \leq B : K}{\Gamma \vdash_{\#} S \leq A \wedge B : K}\ \text{ST-AND-R-}\#$$

$$\frac{\Gamma \vdash_{\#} A \leq B : *}{\Gamma \vdash_{\#} \{\mathbf{val}\ \ell : A\} \leq \{\mathbf{val}\ \ell : B\} : *}\ \text{ST-FIELD-}\#$$

$$\frac{\Gamma \vdash_{\#} J \leq K}{\Gamma \vdash_{\#} \{\mathbf{type}\ M : J\} \leq \{\mathbf{type}\ M : K\} : *}\ \text{ST-TYP-}\#$$

$$\frac{\Gamma \vdash_{\#} B : J \qquad \Gamma, Z : S(B : J) \vdash_{\#} A : K}{\Gamma \vdash_{\#} (\lambda(Z : J).A)\ B \leq A[Z/B] : K[Z/B]}\ \text{ST-}\beta_1\text{-}\#$$

$$\frac{\Gamma \vdash_{\#} B : J \qquad \Gamma, Z : S(B : J) \vdash_{\#} A : K}{\Gamma \vdash_{\#} A[Z/B] \leq (\lambda(Z : J).A)\ B : K[Z/B]}\ \text{ST-}\beta_2\text{-}\#$$

Figure 12: Subtyping

$$\frac{\Gamma \vdash_{\#} A \leq B : K \qquad \Gamma \vdash_{\#} B \leq A : K}{\Gamma \vdash_{\#} A = B : K}\ \text{EQ-}\#$$

Figure 13: Type equality

$$\frac{\Gamma, x : \tau \ \mathtt{ctx}_{\#}}{\Gamma, x : \tau \vdash_{\#} x : \tau} \textsc{Ty-Var-}\#$$

$$\frac{\Gamma \vdash_{\#} e_1 : \tau \qquad \Gamma, x : \tau \vdash e_2 : \rho \qquad x \notin fv(\rho)}{\Gamma \vdash_{\#} \text{let } x = e_1 \text{ in } e_2 : \rho} \textsc{Ty-Let-}\#$$

$$\frac{\Gamma, x : \tau \vdash e : \rho}{\Gamma \vdash_{\#} \lambda(x : \tau).e : (x : \tau) \to \rho} \textsc{Ty-Fun-I-}\#$$

$$\frac{\Gamma \vdash_{\#} x : (z : \tau) \to \rho \qquad \Gamma \vdash_{\#} y : \tau}{\Gamma \vdash_{\#} x\ y : \rho[z/y]} \textsc{Ty-Fun-E-}\#$$

$$\frac{\Gamma \vdash_{\#} x : \tau}{\Gamma \vdash_{\#} x : \mu(x : \tau)} \textsc{Ty-}\mu\textsc{-I-}\# \qquad \frac{\Gamma \vdash_{\#} x : \mu(z : \tau)}{\Gamma \vdash_{\#} x : \tau[z/x]} \textsc{Ty-}\mu\textsc{-E-}\#$$

$$\frac{\Gamma, x : \tau \vdash d : \tau}{\Gamma \vdash_{\#} \nu(x : \tau)d : \mu(x : \tau)} \textsc{Ty-Rec-I-}\# \qquad \frac{\Gamma, x : \{\mathbf{val}\ \ell : \tau\}}{\Gamma \vdash_{\#} x.\ell : \tau} \textsc{Ty-Rec-E-}\#$$

$$\frac{\Gamma \vdash_{\#} x : \tau_1 \qquad \Gamma \vdash_{\#} x : \tau_2}{\Gamma \vdash_{\#} x : \tau_1 \wedge \tau_2} \textsc{Ty-And-I-}\#$$

$$\frac{\Gamma \vdash_{\#} e : \tau_1 \qquad \Gamma \vdash_{\#} \tau_1 \leq \tau_2 : *}{\Gamma \vdash_{\#} e : \tau_2} \textsc{Ty-Sub-}\#$$

$$\frac{\Gamma \vdash_{\#} e : \rho}{\Gamma \vdash_{\#} \{\mathbf{val}\ \ell = e\} : \{\mathbf{val}\ \ell : \rho\}} \textsc{Ty-Def-Trm-}\#$$

$$\frac{\Gamma \vdash_{\#} \tau : K}{\Gamma \vdash_{\#} \{\mathbf{type}\ M = A\} : \{\mathbf{type}\ M : S(A : K)\}} \textsc{Ty-Def-Typ-}\#$$

Figure 14: Type assignment

$$\frac{}{\Gamma, x : \tau \vdash_{!} x : \tau} \textsc{Var-}! \qquad \frac{\Gamma \vdash_{!} x : \mu(z : \tau)}{\Gamma \vdash_{!} x : \tau[z/x]} \textsc{Rec-E-}!$$

$$\frac{\Gamma \vdash_{!} x : \tau_1 \wedge \tau_2}{\Gamma \vdash_{!} x : \tau_1} \textsc{And}_1\textsc{-E-}! \qquad \frac{\Gamma \vdash_{!} x : \tau_1 \wedge \tau_2}{\Gamma \vdash_{!} x : \tau_2} \textsc{And}_2\textsc{-E-}!$$

$$\frac{\Gamma, x : \tau \vdash e : \rho \qquad x \notin fv(\tau)}{\Gamma \vdash_{!} \lambda(x : \tau).e : (x : \tau) \to \rho} \textsc{Fun-I-}! \qquad \frac{\Gamma, x : \tau \vdash d : \tau}{\Gamma \vdash_{!} \nu(x : \tau)d : \mu(x : \tau)} \textsc{Record-I-}!$$

Figure 15: Precise value and variable typing

C Invertible Typing

$$\frac{\Gamma \vdash_! x : \tau}{\Gamma \vdash_{\#\#} x : \tau}\ \textsc{Var}\text{-}\#\# \qquad \frac{\Gamma \vdash_{\#\#} x : \{\mathbf{val}\ \ell : \tau\} \qquad \Gamma \vdash_\# \tau \le \rho : *}{\Gamma \vdash_{\#\#} x : \{\mathbf{val}\ \ell : \rho\}}\ \textsc{Val}\text{-}\#\#$$

$$\frac{\Gamma \vdash_{\#\#} x : \{\mathbf{type}\ M : J\} \qquad \Gamma \vdash_\# J \le K}{\Gamma \vdash_{\#\#} x : \{\mathbf{type}\ M : K\}}\ \textsc{Type}\text{-}\#\#$$

$$\frac{\Gamma \vdash_{\#\#} x : (z : S) \to T \qquad \Gamma \vdash_\# S' \le S : J \qquad \Gamma, z : S' \vdash_\# T \le T' : K}{\Gamma \vdash_{\#\#} x : (z : S') \to T'}\ \textsc{Fun}\text{-}\#\#$$

$$\frac{\Gamma \vdash_{\#\#} x : A \qquad \Gamma \vdash_{\#\#} x : B}{\Gamma \vdash_{\#\#} x : A \wedge B}\ \textsc{Intersect}\text{-}\#\#$$

$$\frac{\Gamma \vdash_{\#\#} x : A \qquad \Gamma \vdash_! z : \{\mathbf{type}\ M : A..A\}}{\Gamma \vdash_{\#\#} x : z.M}\ \textsc{Sel}\text{-}\#\#$$

$$\frac{\Gamma \vdash_{\#\#} x : \tau}{\Gamma \vdash_{\#\#} x : \mu(x : \tau)}\ \textsc{Rec-I}\text{-}\#\# \qquad \frac{\Gamma \vdash_{\#\#} x : \tau}{\Gamma \vdash_{\#\#} x : \top}\ \textsc{Top}\text{-}\#\#$$

Figure 16: Invertible value and variable typing

D Auxiliary Lemmas

Lemma 1 (Type Substitution)**.** *For inert contexts* Γ*,*

- $\Gamma, X : S(A : J) \vdash T : K$ *implies* $\Gamma \vdash T[X/A] : K[X/A]$
- $\Gamma, X : S(A : J) \vdash T_1 \le T_2 : K$ *implies* $\Gamma \vdash T_1[X/A] \le T_2[X/A] : K[X/A]$

Proof. Convert to tight typing, then induct on the tight typing and tight subtyping judgments. □

Lemma 2 (Tight to invertible typing)**.** *For inert contexts* Γ*,* $\Gamma \vdash_\# x : \tau$ *implies* $\Gamma \vdash_{\#\#} x : \tau$*, and for all values* v*,* $\Gamma \vdash_\# v : \tau$ *implies* $\Gamma \vdash_{\#\#} v : \tau$*.*

Proof. By straightforward induction on $\Gamma \vdash_\# x : \tau$ and $\Gamma \vdash_\# v : \tau$. This is formalized in Agda. □

E DOT^ω Operational Semantics

$$\frac{t \mapsto t'}{E[t] \mapsto E[t']}\ \textsc{Term}$$

$$\frac{v = \lambda(z : \tau).t}{\text{let } x = v \text{ in } E[x\ y] \mapsto \text{let } x = v \text{ in } E[t[y/z]]}\ \textsc{Apply}$$

$$\frac{v = \nu(x : \tau)...\{\textbf{val } \ell = t\}}{\text{let } x = v \text{ in } E[x.\ell] \mapsto \text{let } x = v \text{ in } E[t]}\ \textsc{Project}$$

$$\frac{}{\text{let } x = y \text{ in } t \mapsto t[y/x]}\ \textsc{Let-Var}$$

$$\frac{}{\text{let } x = (\text{let } y = e \text{ in } t') \text{ in } t \mapsto \text{let } y = e \text{ in let } x = t' \text{ in } t}\ \textsc{Let-Let}$$

Figure 17: DOT^{ω} Operational Semantics (Amin et al. [1])

$$\frac{E \text{ contains the binding let } x = \lambda(z : \tau).t}{E[x\ y] \mapsto E[t[y/z]]}\ \textsc{Apply}$$

$$\frac{E \text{ contains the binding let } x = \nu(x : \tau)...\{\textbf{val } \ell = t\}}{E[x.\ell] \mapsto E[t]}\ \textsc{Project}$$

$$\frac{}{E[\text{let } x = [y] \text{ in } t] \mapsto E[t[y/x]]}\ \textsc{Let-Var}$$

$$\frac{}{E[\text{let } x = [\text{let } y = e \text{ in } t'] \text{ in } t] \mapsto E[\text{let } y = e \text{ in let } x = t' \text{ in } t]}\ \textsc{Let-Let}$$

Figure 18: DOT^{ω} Operational Semantics with inlined Term (Rapoport et al. [2])