Weak-head normalization of DOT^{ω} types

Soundness for DOT^{ω} update 2023-06-06

Cameron Wong

2023-06-06

1 Last time

Last time, we discussed the generalization of *tight typing* Rapoport et al. [2] to a higher-kinded setting. Tight typing is a re-formulation of the full DOT typing rules to have nicer properties, and showed how tight typing is equivalent to regular typing under a class of well-behaved (*inert*) contexts. Specifically, an inert context is one in which all abstract types (either introduced indirectly via type members or directly via type-level functions) have *singleton kinds* (fig. 1). Specifically, this also necessitated strengthening the premises of type-level beta-reduction to require that the body of an applied lambda still typechecks after substitution.

$$
\begin{aligned}
Sing_{S..U}(A) &= A..A \\
Sing_{\Pi(X:J).K}(A) &= \Pi(X : J).Sing_K(A\ X)
\end{aligned}
$$

Figure 1: Defining singleton kinds

Moving forwards, generalizing Rapoport et al. [2]'s proof of preservation to DOT^{ω} is straightforward, with no issues specific to the higher-kinded setting. Progress, however, requires canonical forms, which in turn requires at least some level of evaluation of type operators. We show only weak-head normalization, which is enough to give us back our canonical forms lemmas.

2 Inert contexts as variable stores

Typically, a logical relations-based proof requires tracking a variable store, with extra machinery ensuring that the store remains in sync with the typing context. Attempting to do so for type-level reductions in DOT^{ω}, however, is surprisingly fraught, owing to DOT^{ω}'s type language making use of two non-interchangeable classes of syntactic variables (namely, type variables and term variables), only one of which is ever substituted for. Splitting the context to assign to type and term variables separately is possible but does not actually solve the problem, as we are still unable to reduce applications of the form $x.F\ A$ for some termvar x.

A second problem is that, just by the nature of the high-level definition of interval kinds, any denotation $[\![S..U]\!]$ must necessarily reference the typing context Γ to even state that $\Gamma \vdash_{\#} S \leq \tau \leq U$.

Luckily, inert contexts provide a way out. We make use of the following property, which follows directly from the definition of inertness:

Property 1. For inert contexts Γ, typevars X and termvars x:

- if $\Gamma(X) = K$, or
- if $\Gamma(x) : \mu(\cdots \wedge \{\mathbf{type}\ M = K\} \wedge \dots)$,

then $K = Sing_{K'}(A)$ for some A and K'.

We also make use of the following lemmas:

Lemma 1. *If* $\Gamma \vdash_{\#} A : Sing_K(B)$*, then* $\Gamma \vdash_{\#} A == B : K$

Proof. By induction on the kind K. If $K = S..U$, then $Sing_K(B) = B..B$, so $\Gamma \vdash_{\#} B \leq A \leq B : S..U$ by ST-$\textsc{bnd}_1$ and ST-$\textsc{bnd}_2$. ☐

Lemma 2 (Type Substitution)**.** *For inert contexts* Γ*,*

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

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

Together, these mean that inert contexts can actually serve as (type) variable stores themselves. While this does require additional bookkeeping to ensure that the context remains inert while traversing the (tight) typing derivation, tight typing is designed precisely to ensure this. This has the additional side-effect of forcing us to reason entirely in terms of tight typing, but we wanted to do that anyway[1].

[1] Arbitrary reduction of open types is already known to be unsound in unrestricted DOT-style calculi. While we may be able to recover (weak-head) normalization anyway by starting from an empty context, any proof based on the unrestricted rules will need some mechanism to address abstract type members anyway, which was the original point of the tight typing rules to begin with.

3 The Proof

We define our logical relation as follows:

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

In a standard termination relation featuring a variable heap H, there is typically a secondary consistency premise $\Gamma \vDash H$ stating that $\Gamma(x) = T$ implies $H(x) \in [\![T]\!]$. Similarly, we need a side condition that, if $\Gamma(X) = Sing_K(A)$, then $\langle \Gamma, A\rangle \in \mathcal{E}[\![Sing_K(A)]\!]$, which we will write as Γ *store*.

We also make use of the following structural lemmas:

Lemma 3 (Relation Inversion)**.**

$$\frac{\langle \Gamma, A\rangle \in \mathcal{E}[\![K]\!]}{\Gamma \vdash_{\#} A : K}$$

Proof. Falls out of $\Gamma \vdash_{\#} A == \tau : K$. □

Lemma 4 (Relation Substitution)**.** *If* $\langle(\Gamma, X : Sing_J(B)), A\rangle \in \mathcal{E}[\![K]\!]$*, then* $\langle \Gamma, A[X/B] \in K[X/B]\rangle$.

Proof. Actually, this feels a little suspect. It seems like it must be true, but my attempts to prove it ends up being circular with the main theorem. I think there may be some termination measure I can use that is preserved by kind substitution. □

At last, the strong normalization theorem, then, is as follows:

Theorem 1 (Weak-head normalization of types)**.**

$$\frac{\Gamma \vdash_{\#} A : K \qquad K \; store}{\langle \Gamma, A\rangle \in \mathcal{E}[\![K]\!]}$$

Proof. By induction on the derivation:

- Case K-Var-#: By the store property.
- Case K-App-#:

  The relevant premises are $\Gamma \vdash_{\#} A : \Pi(X : J).K$ and $\Gamma \vdash_{\#} B : J$. By the inductive hypotheses, we have that there exist some J', A' such that $A = \lambda(X : J').A'$ and for all $B' \in \mathcal{E}[\![J]\!]$, $\langle(\Gamma, X : Sing_J(B')), A'\rangle \in \mathcal{E}[\![K]\!]$.

Instantiating with B gives $\langle(\Gamma, X : \mathit{Sing}_J(B)), A'\rangle \in \mathcal{E}[\![K]\!]$. By Lemma 3, we get that $\Gamma, X : \mathit{Sing}_J(B) \vdash_{\#} A' \in K$, providing the necessary premise for the ST-β rules, giving $\Gamma \vdash_{\#} A\ B == A'[X/B] : K[X/B]$. Finally, 4 and transitivity of type equality gives $\langle \Gamma, A\ B\rangle \in \mathcal{E}[\![K[X/B]]\!]$.

- Case K-Abs-#:

  The relevant premise is $\Gamma, X : J \vdash A : K$. Note that we cannot use this as an inductive hypothesis directly, as it is not tight. Instead, let B be such that $\langle \Gamma, B\rangle \in \mathcal{E}[\![J]\!]$. By Lemma 3, we have $\Gamma \vdash_{\#} B : J$ which can be weakened back to full DOT typing, giving $\Gamma \vdash B : J$. By narrowing on the variable X, we have $\Gamma, X : \mathit{Sing}_J(B) \vdash A : K$. Note that $\Gamma, X : \mathit{Sing}_J(B)$ is a store. Stores are inert, so we can convert back to tight typing to get $\Gamma, X : \mathit{Sing}_J(B) \vdash_{\#} A : K$. *Now* we cite the inductive hypothesis to get $\langle(\Gamma, X : \mathit{Sing}_J(B)), A\rangle \in \mathcal{E}[\![K]\!]$, as desired. This is well-founded because K is syntactically smaller than $\Pi(X : J).K$.

- Case K-Sing-#:

  The premise is $\Gamma \vdash_{\#} A : S..U$. By the inductive hypothesis, we get a whnf τ such that $\Gamma \vdash_{\#} A == \tau : S..U$. The definition of the relation requires showing $\Gamma \vdash_{\#} A \leq \tau \leq A : *$. This is not obviously true, but I think it is because proper types can not bind any "new" variables that aren't already reflected in Γ.

$\square$

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 2: 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 3: 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 4: 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 5: 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.

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

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

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

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

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

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

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

$$\frac{\Gamma Z : J \vdash A : K \qquad \Gamma \vdash B : J}{\Gamma \vdash A[Z/B] \leq (\lambda(Z : J).A)\ B : K[Z/B]}\ \textsc{ST-}\beta_2 \qquad \frac{\Gamma \vdash A : S..U}{\Gamma \vdash S \leq A : *}\ \textsc{ST-bnd}_1$$

$$\frac{\Gamma \vdash A : S..U}{\Gamma \vdash A \leq U : *}\ \textsc{ST-bnd}_2$$

Figure 6: Subtyping

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

Figure 7: Type equality

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

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

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

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

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

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

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

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

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

Figure 8: 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 9: 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 10: 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 11: Subkinding

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

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

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

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

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

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

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

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

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

Figure 12: Kind assignment

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

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

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

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

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

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

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

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

$$\frac{\Gamma \vdash_{\#} B : J \qquad \Gamma, Z : Sing_J(B) \vdash_{\#} A : K}{\Gamma \vdash_{\#} A[Z/B] \leq (\lambda(Z : J).A)\ B : K[Z/B]} \text{ ST-}\beta_2\text{-\#} \quad \frac{\Gamma \vdash_{\#} A : S..U}{\Gamma \vdash_{\#} S \leq A : *} \text{ ST-BND}_1\text{-\#}$$

$$\frac{\Gamma \vdash_{\#} A : S..U}{\Gamma \vdash_{\#} A \leq U : *} \text{ ST-BND}_2\text{-\#}$$

Figure 13: Subtyping

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

Figure 14: Type equality

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

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

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

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

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

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

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

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

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

$$\frac{\Gamma \vdash_{\#} \tau : K}{\Gamma \vdash_{\#} \{\mathbf{type}\ M = A\} : \{\mathbf{type}\ M : SingAK\}} \text{ Ty-Def-Typ-\#}$$

Figure 15: Type assignment

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

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

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

Figure 16: Precise value and variable typing

C Invertible Typing

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

$$\frac{\Gamma \vdash_{\#\#} x : \{\textbf{type}\ M : J\} \qquad \Gamma \vdash_{\#} J \leq K}{\Gamma \vdash_{\#\#} x : \{\textbf{type}\ M : K\}}\ \textsc{Type-\#\#}$$

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

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

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

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

Figure 17: Invertible value and variable typing

D Auxiliary Lemmas

Lemma 5 (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. $\square$

E DOT^{ω} Operational Semantics

$$\frac{t \mapsto t'}{E[t] \mapsto E[t']} \text{ TERM}$$

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

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

$$\frac{}{\text{let } x = y \text{ in } t \mapsto t[y/x]} \text{ LET-VAR}$$

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

Figure 18: DOT^ω Operational Semantics (Amin et al. [1])

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

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

$$\frac{}{E[\text{let } x = [y] \text{ in } t] \mapsto E[t[y/x]]} \text{ LET-VAR}$$

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

Figure 19: DOT^ω Operational Semantics with inlined TERM (Rapoport et al. [2])