Soundness for DOT^{ω} (update 2023-04-13)

Cameron Wong

2023-04-13

1 DOT^{ω} Overview

1.1 Syntax

$x, y, z...$	**Term Variable**	$X, Y, Z...$	**Type Variable**
ℓ	**Value Label**	M	**Type Label**
$e, t ::=$	$x \mid v \mid x.\ell \mid x\ y \mid \text{let } x = e \text{ in } t$		**Term**
$v ::=$	$\lambda(x : \tau).e \mid \nu(x : \tau).d$		**Value**
$d ::=$	$\{\textbf{val } \ell = e\} \mid \{\textbf{type } M = A\} \mid d_1 \wedge d_2$		**Definition**
$A, B, C ::=$	$X \mid \lambda(X : K).A \mid x.M \mid A\ B$		**Type**
$(\tau, \rho, S, U, T ::=)$	$\mid \top \mid \bot \mid (x : \tau) \to \rho \mid \tau \wedge \rho \mid \mu(x.\tau)$ $\mid \{\textbf{val } \ell : \tau\} \mid \{\textbf{type } M : K\}$		(Proper types)
$J, K ::=$	$S..U \mid \Pi(X : J).K$		**Kind**

Figure 1: Abstract syntax of DOT^{ω}

We also use $*$ as shorthand for $\bot..\top$.

Following Stucki [4], we represent type bounds at the kind level, with $\Gamma \vdash \tau : S..U$ meaning $S \leq \tau$ and $\tau \leq U$. Type members are then assigned a kind, rather than marking bounds syntactically.

1.2 Typing and Kinding

The full declarative typing system can be found in Appendix A. Most rules are lifted directly from Stucki [4], adding back recursive types, type members and intersection types. We also do not feature an η rule, but may re-add it if necessary.

Notice that, instead of the rules <:-Sel and Sel-<: relating a type member to its bounds, we instead have one rule K-Typ-Mem giving each type member its kind.

Kind Assignment

$$\frac{\Gamma \vdash A : S..U}{\Gamma \vdash A : A..A}\ \text{K-Sing}$$

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

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

$$\frac{\Gamma \vdash x : \{\textbf{type}\ M : K\}}{\Gamma \vdash x.M : K}\ \text{K-Typ-Mem}$$

Subtyping

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

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

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

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

Figure 2: Kinding and subtyping in DOT^ω (selected rules)

2 The Proof

2.1 Outline

Our general proof recipe will largely follow that of Rapoport et al. [2]. Unlike the first-order system, merely knowing that $\Gamma \vdash v : A$ does not allow us to deduce the canonical form of the value v, as A may be a complex type expression containing β-redexes. Nor can we simply normalize the type expression A, as we may be in a crazy context in which subtyping cannot be trusted and subject reduction does not hold. If we *do* know that the context is inert, however, then reduction is sound and all is well.

The overall plan of attack, then, will proceed as follows:

1. Extend DOT's tight typing rules with corresponding tight *kinding* and reduction rules, alongside extending the notion of inert contexts to restrict type variables as well.

2. Show that general typing and kinding implies tight typing and kinding in inert contexts.

3. Prove that, under tight subtyping, reduction of type expressions is sound and normalizing.

4. Finally, use the above to prove the relevant canonical forms lemmas necessary to show progress and preservation of DOT^{ω}.

None of these are new ideas; the contribution is in applying Rapoport et al. [2]'s simple proof recipe to address issues raised by Stucki [4]. Items 1 and 2 have proof-of-concept mechanizations in Agda.

2.2 Inert Higher-kinded Contexts

Rapoport et al. [2] defines inert contexts to be contexts assigning inert types to each variable x contained within, where an inert type τ is:

- A function $(x : \tau_1) \to \tau_2$, or
- A recursive type $\mu(x : \tau)$, where τ is an intersection of field declarations and tight type declarations $\{\textbf{type}\ M : S..S\}$, where all type labels are distinct.

To generalize this to higher kinds, then, we must account for kinds assigned to type variables and kinds of type members $\{\textbf{type}\ M : K\}$. Fortunately, this work has already been done for us.

Stucki and Giarrusso [5] define *higher-order type intervals* as:

$$A..{}_{A'..B'}B ::= A..B$$
$$A..{}_{\Pi(X:J).K}B ::= \Pi(X : J).A\ X..{}_K B\ X$$

From here, we can recover Stone and Harper [3]'s *generalized singletons* simply by setting the bounds A and B to be equal, e.g.

$$S(\tau : S..U) ::= \tau..\tau$$
$$S(A : \Pi(X : J).K) ::= \Pi(X : J).S(A\ X : K)$$

This gives the full definition of inert contexts as contexts containing only:

- Dependent term functions $(x : \tau_1) \to \tau_2$,
- Recursive types built as the intersection of record fields and type declarations $\{\textbf{type}\ M : S(A : K)\}$ for some A and K, and
- Singleton kinds $S(A : K)$.

Note that we exclude from our definition of singletons intervals that may be $\beta\eta$ equivalent, such as $(\lambda(\_ : *).\top)\ \top..\top$ This is to avoid needing to perform

reduction (which, remember, may be unsound!) when determining whether a kind is indeed a singleton.

2.3 Tight Reduction

As nice as singleton kinds are, they are not yet sufficient to tame the wild world of arbitrary subtyping. For example, the singleton kind $\Pi(X : \top..\bot).\{\textbf{type } M : \top..\bot\}..\{\textbf{type } M : \top..\bot\}$ producing the absurd type $\{\textbf{type } M : \top..\bot\}$ is inhabited (by a function that assigns $M = X$), so merely restricting type variables to having singleton kinds is not enough.

Instead, we must use the fact that such a type-level function cannot be called at all (or, dually, that calling such a function requires witnessing the goodness of any bounds). Tight typing (Amin et al. [1]) provides just this functionality for term-level functions, but what about at the type level?

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

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

Figure 3: Tight Type Reduction

To recover subject reduction, we must amend the β-reduction rules to ensure that it does not make use of any strange subtyping relationships that would not be present otherwise. This is done by type checking the lambda body under the restricted context $\Gamma, Z : S(B : J)$ rather than $\Gamma, Z : J$ (fig. 2.3).

Precise Variable Typing

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

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

Tight Kinding (selected rule)

$$\frac{\Gamma \vdash_{!} x : \{\textbf{type } M : S(A : K)\}}{\Gamma \vdash_{\#} x.M : S(A : K)}\ \textsc{K-Typ-Mem}\text{-}\#$$

Figure 4: Adapting precise typing (Amin et al. [1]) to a higher-kinded setting

We also amend K-Typ-Mem, generalizing Amin et al. [1]'s tight type member rule to act on higher-order singletons (fig. 2.3). The full proposed tight typing

rules can be found in Appendix B.

We now show that this restricted setting is equivalent to the general typing/kinding rules. We do this by making use of two lemmas:

Lemma 1 (β-# Premise)**.** *If* Γ *is inert, then if* $\Gamma, X : J \vdash A : K$ *and* $\Gamma \vdash_{\#} B : J$*, then* $\Gamma, X : S(B : J) \vdash_{\#} A : K$

Lemma 2 (K-Typ-Mem-# Premise)**.** *If* Γ *is inert, then if* $\Gamma \vdash_{\#} x : \{\textbf{\textit{type}}\ M : K\}$*, then there exists some* A *such that* $\Gamma \vdash_{\#} A : K$ *and* $\Gamma \vdash_{!} x : \{\textbf{\textit{type}}\ M : S(A : K)\}$

The proof of Lemma 2 follows largely the same as Rapoport et al. [2] (the details of defining invertible tight typing for DOT^{ω} are straightforward and therefore elided).

Unfortunately, Lemma 1 cannot be stated entirely in terms of tight typing. The culprit is the $\Gamma, X : J \vdash A : K$ in the premise of the ST-β rules, which cannot be converted to a tightly-typed equivalent (because J may introduce bad bounds). Instead, we must prove this lemma mutuall alongside the full equivalence theorem:

Theorem 1. *If* Γ *is an inert context such that* $\Gamma \vdash e : \tau$*, then* $\Gamma \vdash_{\#} e : \tau$*.*

Proof. Following Rapoport et al. [2], we focus on showing equivalence of individual rules rather than the typing/kinding relations as a whole.

- K-Typ-Mem discharges to K-Typ-Mem-# via Lemma 2
- ST-β_1 and ST-β_2 similarly invoke the corresponding tight-typing rules, with premises obtained from Lemma 1.

□

Proof of Lemma 1. We elide a proof that $\Gamma \vdash S(B : J) \leq J$.

By narrowing, $\Gamma, X : S(B : J) \vdash A : K$. By Theorem 1, $\Gamma, X : S(B : J) \vdash_{\#} A : K$ as desired. □

Note that, for this mutual proof to be well-founded, the proof of the narrowing theorem must ensure that the proof of $\Gamma, X : S(B : J) \vdash A : K$ is at most as large as the proof of $\Gamma, X : J \vdash A : K$, which thankfully, it does.

3 What Remains

The next step is, as marked on the outline, to show that tight subtyping still respects subject reduction and normalization. From there, we can use the same

steps as Rapoport et al. [2] to show progress and preservation.

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 5: 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 6: Kind formation

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

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

Figure 7: Subkinding

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

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

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

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

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

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

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

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

Figure 8: 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 9: Subtyping

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

Figure 10: Type equality

B Full 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.

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

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

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

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

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

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

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

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

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

Figure 11: Type assignment

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

Figure 12: Context formation

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

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

Figure 13: Kind formation

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

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

Figure 14: Subkinding

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

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

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

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

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

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

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

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

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

Figure 15: 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 16: Subtyping

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

Figure 17: 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 18: 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 19: Precise Value and Variable Typing