Soundness for DOT^ω (update 2023-04-12)

Cameron Wong

2023-04-12

1 Background

The key difficulty in introducing higher kinded types to DOT is, like with any DOT soundness proof, bad bounds. Particularly, in an open context, neither types nor terms satisfy subject reduction, making it unsound to perform evaluation in the presence of arbitrary subtyping lattices. This phenomenon is well-documented by Amin et al. [1], Rapoport et al. [2], Stucki and Giarrusso [3] and others besides.

The usual approach to dealing with this problem is to consider soundness only in closed terms, or in limited contexts where all bounds are known to be good. This notion was originally formalized by Amin et al. [1] as *tight typing*. Rapoport et al. [2] then demonstrated how tight typing can be used to dramatically simplify the proof of soundness for DOT. This was done by defining *inert typing contexts*, in which tight typing and general typing for DOT are equivalent. This allowed them to prove properties of the overall DOT type system by first dropping into tight typing, then using simple, intuitive reasoning over the tightly-typed system. In the words of Rapoport et al. [2], "To reconcile a subtyping lattice with a sound language, we only need to force the programmer to provide evidence that the custom lattice does not violate any familiar assumptions... This evidence takes the form of an argument value that must be passed to the lambda before the body that uses the custom type lattice can be allowed to execute.".

A major difficulty in extending this recipe to cover higher-kinded types is that, when types can contain computation themselves, some β-reduction *has* to be performed in the typing rules (e.g., to compare types during typechecking), which is similarly unsound in the presence of bad bounds. Thankfully, this, too, can be resolved by making the following observation:

Property 1. *If $\varnothing \vdash e : \tau$, then $\varnothing \vdash \tau : *$.*

That is, any closed expression e must be assigned a proper type (as opposed to a type lambda) if it has any type at all. This means that, even if τ is a complex type expression containing many type functions, each of those functions must ultimately be applied to another type expression witnessing the validity of any bounds introduced. We will formally define this property in Section 3.2.

2 DOT^ω, the declarative system

In lieu of a comprehensive overview of DOT^ω, we discuss only the details of the kinding system and how it interacts with vanilla DOT. The full declarative system can be found in Appendix A.

2.1 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**
$(\tau, \rho, S, U ::=)$	$\mid \top \mid \bot \mid (x : \tau) \to \rho \mid \tau \wedge \rho \mid \mu(x.\tau)$	(Proper types)
	$\mid \{\textbf{val}\ \ell : \tau\} \mid \{\textbf{type}\ M : K\}$	
$J, K ::=$	$S..U \mid \Pi(X : J).K$	**Kind**

Figure 1: Abstract syntax of DOT^ω

3 The Proof

3.1 Overview

Our general proof recipe will largely follow that of Rapoport et al. [2].

3.2 Tight Typing and Kinding

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

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

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

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

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

$$\frac{\Gamma \vdash A : S_1..U_1 \qquad \Gamma \vdash B : S_2..U_2}{\Gamma \vdash A \wedge B : S_1 \vee S_2..U_1 \wedge U_2}\ \textsc{K-Intersect} \qquad \frac{\Gamma \vdash A : S..U}{\Gamma \vdash \{\mathbf{val}\ \ell : A\}}\ \textsc{K-Field}$$

$$\frac{\Gamma \vdash K\ \texttt{kd}}{\Gamma \vdash \{\mathbf{type}\ M : K\}}\ \textsc{K-Typ} \qquad \frac{\Gamma \vdash x : \{\mathbf{type}\ M : K\}}{\Gamma \vdash x.M : K}\ \textsc{K-Typ-Mem}$$

$$\frac{\Gamma, x : \tau \vdash \tau : K}{\Gamma \vdash \mu(x.\tau) : K}\ \textsc{K-Rec} \qquad \frac{\Gamma \vdash A : J \qquad \Gamma \vdash J \leq K}{\Gamma \vdash A : K}\ \textsc{K-Sub}$$

Figure 5: Kind assignment

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

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

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

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

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

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

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

$$\frac{\Gamma \vdash X = \lambda(Z : J).A : \Pi(Z : J).K \qquad \Gamma \vdash Y : J}{\Gamma \vdash A[Y/Z] \leq X\ Y : K[Y/Z]}\ \textsc{ST-}\beta_2$$

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

$$\frac{\Gamma \vdash \tau : K}{\Gamma \vdash \{\mathbf{type}\ M = \tau\} : \{\mathbf{type}\ M : K\}}\ \textsc{Def-type}$$

Figure 8: Type assignment