Some Restrictions in Higher-Kinded Systems

Lemon

April 19, 2025

Contents

This note came to be after noticing some details when reading Chapter 30, Higher-Order Polymorphism of TAPL (Pierce, Benjamin C., 2002).

1 Background

One of the kinding rule for F_ω is as follow:

$$\frac{\Gamma, X :: K_1 \vdash T_2 :: \star}{\Gamma \vdash \forall X :: K_1 \,.\, T_2 :: \star}$$

We pose the question:

> In the presumption, why does T_2 need to have the kind $\star$?

2 Intuition

Intuitively, the kind "$\star$" are:

Types that can be inhabited with actual value.

We illustrate this idea with some examples:

- The type `Int` has kind "$\star$", and we can construct values such as `10 : Int`
- The type `Maybe Char` has kind "$\star$", and we can construct values such as `Just 'A' : Maybe Char`
- The type `Maybe` has kind "$\star \to \star$", and there is no value with the type just `Maybe`

3 How Restrictions Came to Be?

Consider some standard typing rules of F_ω (Something similar to the one in TAPL), with the judgment:

$$\Gamma \vdash t : T$$

In context Γ, the term t has type T

Because we are assigning a term t with a type T, T must have kind "$\star$" (Non-$\star$ type cannot be inhabited)

When developing the metatheory, we will likely want to prove the theorem:

$$\Gamma \vdash t : T \Rightarrow \Gamma \vdash T : \star$$

What is the main challenge when proving this theorem? Let us consider the typing rule cases:

3.1 The Harmless Ones

3.1.1 T-App

$$\frac{\Gamma \vdash t_1 : T_{11} \to T_{12} \qquad \Gamma \vdash t_2 : T_{11}}{\Gamma \vdash t_1\ t_2 : T_{12}}$$

This is no trouble, IH will guarantee desired result

3.1.2 T-TAbs

$$\frac{\Gamma, X :: K_1 \vdash t_2 :: T_2}{\Gamma \vdash \lambda X :: K_1 \ . \ t_2 : \forall X :: K_1 \ . \ T_2}$$

This is no trouble, IH will guarantee $T_2 : \star$ in extended context, and quantifying over the free variable will make $\forall X :: K_1 \ . \ T_2$ of kind $\star$ again.

3.1.3 T-TApp

$$\frac{\Gamma \vdash t_1 : \forall X :: K_{11} \ . \ T_{12} \qquad \Gamma \vdash T_2 :: K_{11}}{\Gamma \vdash t_1 \ [T_2] \ : [T_2/X]T_{12}}$$

This is no trouble, IH will guarantee $\forall X :: K_{11} \ . \ T_{12}$ is of kind $\star$, and assuming some substitution lemma, $[T_2/X]T_{12}$ will be of kind $\star$ too.

3.2 The Trouble Makers

3.2.1 T-Var

$$\frac{x : T \in \Gamma}{\Gamma \vdash x : T}$$

This will definitely cause issue if T is freely allowed to be anything (e.g. `Maybe`), then we will output a non-$\star$ kinded type.

There are a few ways to fix this issue:

1. Add a precondition checking if Γ is "well-formed". Check whether all the type assigned to variables are actually of kind $\star$

2. "Check the input", make sure when context gets extended, the extension does not introduce a variable binding to non-$\star$ type

In a sense, option 2 will preserve the well-formedness defined in option 1.

If we choose option 2, it will lead us to the next section

3.2.2 T-Abs

$$\frac{\Gamma \vdash T_1 :: \star \qquad \Gamma, x : T_1 \vdash t_2 : T_2}{\Gamma \vdash \lambda x : T_1 \ . \ t_2 : T_1 \to T_2}$$

The first presumption prevents non-$\star$ kinded types from being introduced to the context

4 Back to $\Gamma, X :: K_1 \vdash T_2 :: \star$

Back to why T_2 has to be of kind $\star$. A different way of phrasing this question might be:

Why can't T_2 be higher-kind?

This has to do with how "$\forall$" can be introduced. "$\forall$" can only be introduced via a kind abstraction. In TAPL's case, this will be $\lambda X :: K \ . \ t$, though in other text it might be $\Lambda X :: K \ . \ t$.

Regardless of the representation, the term t should have the type T_2. And because T_2 can be given a value, it must be of kind $\star$

5 Thanks

To Ningning and Ethan for the discussion