

The Recursive Index Calculus and Its Translation From $R\omega$

AH & JGM

October 20, 2023

1 Design considerations of the formal index calculus

1.1 The necessity of Set-in-Set

Denote the translation of $R\omega$ type τ to the index calculus as $\llbracket \tau \rrbracket$. The big stars of systems $R\&c.$ are record and variants. Consider how we might translate the former. We first establish the translation of kinds so that we may assert that translated types are in the meaning of their translated kinds. (This is a sensible verification.) The following seems reasonable.

$$\begin{aligned}\llbracket \star \rrbracket_{\kappa} &= \star \\ \llbracket \mathbf{L} \rrbracket_{\kappa} &= \top \\ \llbracket \kappa_1 \rightarrow \kappa_2 \rrbracket_{\kappa} &= \llbracket \kappa_1 \rrbracket \rightarrow \llbracket \kappa_2 \rrbracket \\ \llbracket \mathbf{R}^{\kappa} \rrbracket_{\kappa} &= \exists i : \mathit{Nat}. (\mathit{Fin } i \rightarrow \llbracket \kappa \rrbracket)\end{aligned}$$

(Nevermind, for now, the complexity of existentially quantifying Nats within kinds.) Recall that records (in $R\omega$) live in kind \star but, as shown in our Agda denotation, should translate to functions which map finite indices to types. Below is a sensible type and translation for such an idea. (As $\llbracket \rho \rrbracket$ is a dependent product, let its first and second projections be defined as usual.)

$$\llbracket \Pi \rho \rrbracket = \forall i : \text{Nat}. \text{Fin } \llbracket \rho \rrbracket.1 \rightarrow \llbracket \rho \rrbracket.2 i$$

Yet, as $\Pi \rho : \star$ (in $\text{R}\omega$), we must conclude that the type $\forall i : \text{Nat}. \text{Fin } \llbracket \rho \rrbracket.1 \rightarrow \llbracket \rho \rrbracket.2 i$ has kind \star in μIx . I don't see a way around this. Further, it implies to me that we would like to flatten types and kinds and let types quantify over types. That is to say, we want a two-level stratification of terms and types only, omitting kinds. The next two subsections argue for this change.

1.2 Flattening kinds and types

Revisit now the translation of rows.

$$\llbracket \mathbf{R}^\kappa \rrbracket_\kappa = \exists i : \text{Nat}. (\text{Fin } i \rightarrow \llbracket \kappa \rrbracket)$$

The existential quantification of $i : \text{Nat}$ is ambiguous. We seem to need to either (i) permit the dependent, existential quantification over *type* Nat , or (ii) lift Nat indices to kinds and permit the existential quantification of indices in kinds. In the latter case, we would further need to represent the *application* of Fin to *index variable* i —and also, of course, need to represent existential quantification in kinds (and represent Fin !). This becomes cumbersome quickly, enough to beg the question: why distinguish types from kinds at all?

1.2.1 The inadequacy of Xi and Pfenning [1999]

Described above by (i) is the approach advocated by Xi and Pfenning [1999]. Xi and Pfenning [1999] only permits types to be indexed by a declared set of index objects, each with index sort—for example, vectors are indexed by index objects with index sort Nat . The syntax of (some simple) indices is given below.

$$\begin{array}{ll} \text{Index Sorts} & \gamma ::= \text{Nat} \mid \top \mid \dots \\ \text{Index Objects} & \iota ::= i \mid () \mid \dots \end{array}$$

Rather than types we have *families of types*, each indexed by *index objects*. For example, an *intlist* might be a family type family indexed by Nat , representing its length. Consider the typing of vector concatenation.

$$\text{concat} : \Pi m : \text{Nat}. \Pi n : \text{Nat}. \text{intlist } m \rightarrow \text{intlist } n \rightarrow \text{intlist } (m + n)$$

This machinery is not sufficient to type many $R\omega$ terms. To illustrate, presume a base set of index sorts γ and index objects ι to be as defined below. (For convenience, let the index objects of ι inhabit both Nat and $\text{Fin } \iota$.)

$$\begin{aligned} \gamma &::= \text{Nat} \mid \text{Fin } \iota \\ \iota &::= i \mid 0 \mid \text{Suc } \iota \end{aligned}$$

Now consider an example translation of record concatenation from $R\omega$ to μIx . In $R\omega$, we have:

$$\text{concat} : \forall (z_1 z_2 z_3 : \mathbf{R}^*). z_1 \cdot z_2 \sim z_3 \Rightarrow \Pi z_1 \rightarrow \Pi z_2 \rightarrow \Pi z_3$$

This (hypothetically) translates to (the sketch I have in my head of) the μIx type:

$$\begin{aligned} \text{concat} : & \forall^i mnl : \text{Nat}. \\ & \forall (z_1 : \text{Fin } m \rightarrow \star) (z_2 : \text{Fin } n \rightarrow \star) (z_3 : \text{Fin } l \rightarrow \star). \\ & \llbracket z_1 \cdot z_2 \cdot z_3 \rrbracket \rightarrow \\ & (\forall (i : \text{Fin } m) \rightarrow z_1 i) \\ & (\forall (i : \text{Fin } n) \rightarrow z_2 i) \\ & \forall (i : \text{Fin } l). z_3 i \end{aligned}$$

Let \forall^i denote the quantification of indices in types and $\llbracket z_1 \cdot z_2 \sim z_3 \rrbracket$ denote the (yet-undefined) translation of $R\omega$ predicates to μIx types. The glaring incompatibility of Xi and Pfenning [1999] is that $R\omega$ is higher-order. So, it is not clear what the quantification of z_1 over $(\text{Fin } m \rightarrow \star)$ means. Is $(\text{Fin } m \rightarrow \star)$ a kind? Xi and Pfenning [1999] permits only the quantification over indices, and the use of those indices in types. The authors do not permit $F\omega$ (or just System F)-style quantification over type variables. And it is not clear how to lift their calculus to higher-order; I suspect, also, nontrivial.

A lot of our problems go away when committing to an impredicative, dependent, term-and-type-stratified type theory. So, this is what I suggest we do.

Firstly, the higher-order nature of $F\omega$ is given for free, as there are no more kinds and thus all type-level quantification is over types. W.r.t. mechanizational ease, we substantially reduce the overlap in ASTs.

What we are left with is more less MLTT with (built-in) finite naturals. Call it $\lambda^{\Pi, \Sigma}$ for now.

1.3 Consistency in impredicative MLTT (and workarounds thereof)

Martin L of Type Theory is well known to be inconsistent without a predicative universe hierarchy Hurkens [1995], Monnier and Bos [2019], Coquand [1992].

2 An informal translation of $R\omega$ to μIx

For the purposes of investigation, consider a translation of the term-language of $R\omega$ into μIx . (μIx is, of course, yet undefined—this translation will inform such a definition.)

One observation to make of $R\omega$ is that much of what you may see in a type does not have kind \star . In particular, rows do not. Further, rows effectively behave like terms in the type language. So it might be necessary to stratify rows and types apart, so that rows may behave as they wish in the type language and terms may do similarly in the term language, which (I postulate) should avoid the necessity of \star having type \star , which is known to be unsound in (most? all?) dependent type theories.¹

¹There must be, of course, ways around this—e.g., **Pred** is impredicative in **Coq**—but I think it’s best to use this sort of machinery as a last resort.

$\Gamma \vdash \tau : \kappa$	$\llbracket \Gamma \vdash \tau : \kappa \rrbracket$	$\mu\text{Ix Type}$	$\mu\text{Ix Kind}$
$\llbracket \Gamma \vdash \ell : [\ell] \rrbracket$	tt	\top	\star
$\llbracket \Gamma \vdash M_1 \triangleright M_2 : \ell \triangleright \tau \rrbracket$	$\llbracket M_2 \rrbracket$	$\llbracket \tau \rrbracket$	\star
$\llbracket \Gamma \vdash M_1 / M_2 : \tau \rrbracket$	$\llbracket M_1 \rrbracket$	$\llbracket \tau \rrbracket$	\star
$\llbracket \Gamma \vdash \text{prj } M : \Pi \rho_2 \rrbracket$		$\Pi(i : \text{Ix } \llbracket \rho_2 \rrbracket.1). \text{Ix } \llbracket \rho_2 \rrbracket.2 \ i$	
$\llbracket \Gamma \vdash M_1 ++ M_2 : \Pi \rho_3 \rrbracket$			
$\llbracket \Gamma \vdash \text{inj } M : \Sigma \rho_2 \rrbracket$			
$\llbracket \Gamma \vdash M_1 \nabla M_2 : \Sigma \rho_3 \rightarrow \tau \rrbracket$			
$\llbracket \Gamma \vdash \text{ana}_\phi M : \Sigma(\phi \rho) \rightarrow \tau \rrbracket$			
$\llbracket \Gamma \vdash \text{syn}_\phi M : \Pi(\phi \rho) \rrbracket$			
$\llbracket \Gamma \vdash \text{fold } M_1 M_2 M_3 N : v \rrbracket$			
$\llbracket \Gamma \vdash \lambda x : \tau_1. M : \tau_1 \rightarrow \tau_2 \rrbracket$			
$\llbracket \Gamma \vdash M_1 M_2 : \tau_2 \rrbracket$			
$\llbracket \Gamma \vdash M : \pi \Rightarrow \tau \rrbracket$			
$\llbracket \Gamma \vdash M : \tau \rrbracket$			
$\llbracket \Gamma \vdash \Lambda \alpha : \kappa. M : \forall \alpha : \kappa. \tau \rrbracket$			
$\llbracket \Gamma \vdash M[v] : \tau[v/\alpha] \rrbracket$			

3 The formalized index calculus

3.1 Syntax

	Term variables x	Type variables $\alpha \iota$
Index Sorts	$\gamma ::= \text{Nat} \mid \text{Ix } \iota$	
Index Objects	$\iota ::= i \mid \text{Zero} \mid \text{Suc } \iota$	
Types	$\tau, v ::= \gamma \mid \iota \mid \star \mid \alpha \mid \top \mid \Pi i : \tau. v \mid \Sigma i : \tau. v$	
	$\tau v \mid \tau \sim v$	
Terms	$M, N ::= x \mid () \mid \lambda x : \tau. M \mid M N \mid (M, N) \mid M.1 \mid M.2$	
Environments	$\Gamma ::= \varepsilon \mid \Gamma, x : \tau$	

Figure 1: Syntax

3.2 Typing

$$\begin{array}{c}
\boxed{\Gamma \vdash \iota : \gamma} \\
\frac{\vdash \Gamma \quad i : \gamma \in \Gamma}{\Gamma \vdash i : \gamma} \\
\frac{\vdash \Gamma}{\Gamma \vdash \text{Zero} : \text{Nat}} \quad \frac{\Gamma \vdash \iota : \text{Nat}}{\Gamma \vdash \text{Suc } \iota : \text{Nat}} \\
\frac{\vdash \Gamma}{\Gamma \vdash \text{Zero} : \text{Ix } 1} \quad \frac{\Gamma \vdash n : \text{Nat} \quad \Gamma \vdash \iota : \text{Ix } n}{\Gamma \vdash \text{Suc } \iota : \text{Ix } (\text{Suc } n)} \quad \frac{\Gamma \vdash n : \text{Nat} \quad \Gamma \vdash \iota : \text{Ix } n}{\Gamma \vdash \iota : \text{Ix } (\text{Suc } n)} \\
\boxed{\vdash \Gamma} \\
(\text{C-EMP}) \frac{}{\vdash \varepsilon} \quad (\text{C-VAR}) \frac{\vdash \Gamma \quad \Gamma \vdash \tau \text{Type}}{\vdash \Gamma, x : \tau} \\
\boxed{\Gamma \vdash \tau : v} \\
(\text{K-}\star\text{-IN-}\star) \frac{\vdash \Gamma}{\Gamma \vdash \star : \star} \quad (\text{K-}\top) \frac{\vdash \Gamma}{\Gamma \vdash \top : \star} \quad (\text{K-VAR}) \frac{\vdash \Gamma \quad x : \tau \in \Gamma}{\Gamma \vdash x : \tau} \\
(\text{K-}\forall) \frac{\Gamma, \alpha : \kappa \vdash \tau : \star}{\Gamma \vdash \forall \alpha : \kappa. \tau : \star} \quad (\text{K-}\exists) \frac{\Gamma, \alpha : \tau \vdash v : \star}{\Gamma \vdash \exists \alpha : \tau. v : \star} \\
(\text{K-}\Pi) \frac{\Gamma, \alpha : \kappa \vdash \tau : \star}{\Gamma \vdash \forall \alpha : \kappa. \tau : \star} \quad (\text{K-}\Sigma) \frac{\Gamma, \alpha : \kappa \vdash \tau : \star}{\Gamma \vdash \exists \alpha : \kappa. \tau : \star} \\
(\text{K-}\rightarrow I) \frac{\Gamma, \alpha : \kappa_1 \vdash \tau : \kappa_2}{\Gamma \vdash \lambda \alpha : \kappa_1. \tau : \kappa_1 \rightarrow \kappa_2} \quad (\text{K-}\rightarrow E) \frac{\Gamma \vdash \tau_1 : \kappa_1 \rightarrow \kappa_2 \quad \Gamma \vdash \tau_2 : \kappa_1}{\Gamma \vdash \tau_1 \tau_2 : \kappa_2} \\
(\text{K-}\sim) \frac{\Gamma \vdash \tau_1 : \star \quad \Gamma \vdash \tau_2 : \star}{\Gamma \vdash \tau_1 \sim \tau_2 : \star} \quad (\text{K-IX}) \frac{\Gamma \vdash \tau : \text{Nat}}{\Gamma \vdash \text{Ix } \tau : \star} \\
(\text{K-}\mu) \frac{\Gamma \vdash \tau : \star \rightarrow \star}{\Gamma \vdash \mu \tau : \star} \quad (\text{K-}\nu) \frac{\Gamma \vdash \tau : \star \rightarrow \star}{\Gamma \vdash \nu \tau : \star}
\end{array}$$

Figure 2: Contexts and kinding.

3.3 Typing

4 Adding Recursion to R_ω

The static semantics of R_ω , as defined in Hubers and Morris [2023], are given in Appendix A for reference. We define only the syntax and rules necessary for least- and greatest-fixed points with general term-level recursion, which are routine.

$$\begin{array}{c}
 \boxed{\Gamma \vdash \tau : \kappa} \\
 \frac{\Gamma \vdash \tau : \star \rightarrow \star}{\Gamma \vdash \mu\tau : \star} \quad \frac{\Gamma \vdash \tau : \star \rightarrow \star}{\Gamma \vdash \nu\tau : \star} \\
 \boxed{\Gamma \vdash M : \tau} \\
 \dots
 \end{array}$$

5 Translating μIx from R_ω

Rules for the System F_ω fragment of R_ω have a trivial correspondence to the F_ω fragment of μIx and are omitted. The syntax and typing judgments on the left are that of R_ω (see Appendix A); on the right are μIx .

$$\begin{array}{c}
\boxed{\llbracket \kappa \rrbracket} \\
\llbracket \star \rrbracket = \star \\
\llbracket \mathbf{L} \rrbracket = \star \\
\llbracket \kappa_1 \rightarrow \kappa_2 \rrbracket = \llbracket \kappa_1 \rrbracket \rightarrow \llbracket \kappa_2 \rrbracket \\
\llbracket \mathbf{R}^\kappa \rrbracket = \text{Nat} \rightarrow \llbracket \kappa \rrbracket \\
\\
\boxed{\llbracket \Gamma \vdash \tau : \kappa \rrbracket} \\
\llbracket \frac{\Gamma \vdash \rho : \mathbf{R}^\kappa}{\Gamma \vdash \Pi \rho : \kappa} \rrbracket = \forall n : \text{Nat}. \text{Ix } n \rightarrow \llbracket \Gamma \vdash \rho : \mathbf{R}^\kappa \rrbracket n \\
\llbracket \frac{\Gamma \vdash \rho : \mathbf{R}^\kappa}{\Gamma \vdash \Sigma \rho : \kappa} \rrbracket = \exists n : \text{Nat}. \llbracket \Gamma \vdash \rho : \mathbf{R}^\kappa \rrbracket n \\
\llbracket \frac{\vdash \Gamma}{\Gamma \vdash \ell : \mathbf{L}} \rrbracket = \top \\
\llbracket \frac{\Gamma \vdash \xi : \mathbf{L}}{\Gamma \vdash \lfloor \xi \rfloor : \star} \rrbracket = \top \\
\llbracket \frac{\Gamma \vdash \xi : \mathbf{L} \quad \Gamma \vdash \tau : \kappa}{\Gamma \vdash \xi \triangleright \tau : \kappa} \rrbracket = \llbracket \Gamma \vdash \tau : \kappa \rrbracket \\
\llbracket \frac{\Gamma \vdash \rho : \mathbf{R}^{\kappa_1 \rightarrow \kappa_2} \quad \Gamma \vdash \tau : \kappa_1}{\Gamma \vdash \rho \tau : \mathbf{R}^{\kappa_2}} \rrbracket = \lambda n : \text{Nat}. \llbracket \Gamma \vdash \rho : \mathbf{R}^{\kappa_1 \rightarrow \kappa_2} \rrbracket n \llbracket \Gamma \vdash \tau : \kappa_1 \rrbracket \\
\llbracket \frac{\Gamma \vdash \phi : \kappa_1 \rightarrow \kappa_2 \quad \Gamma \vdash \rho : \mathbf{R}^{\kappa_1}}{\Gamma \vdash \phi \rho : \mathbf{R}^{\kappa_2}} \rrbracket = \lambda n : \text{Nat}. \llbracket \Gamma \vdash \phi : \kappa_1 \rightarrow \kappa_2 \rrbracket n \llbracket \Gamma \vdash \rho : \mathbf{R}^{\kappa_1} \rrbracket \\
\llbracket \frac{\Gamma \vdash_{\mathcal{T}} \{\overline{\xi \triangleright \tau}\} : \mathbf{R}^\kappa}{\Gamma \vdash \{\overline{\xi \triangleright \tau}\} : \mathbf{R}^\kappa} \rrbracket = \llbracket \Gamma \vdash_{\mathcal{T}} \{\overline{\xi \triangleright \tau}\} : \mathbf{R}^\kappa \rrbracket \\
\llbracket \frac{\Gamma \vdash \pi \quad \Gamma, \pi \vdash \tau : \star}{\Gamma \vdash \pi \Rightarrow \tau : \star} \rrbracket = \llbracket \pi \rrbracket \rightarrow \llbracket \tau \rrbracket \\
\\
\boxed{\llbracket \Gamma \vdash \pi \rrbracket} \\
\dots
\end{array}$$

Figure 3: A compositional translation of $\mathcal{R}\omega$ to μIx

References

- Thierry Coquand. The paradox of trees in type theory. *BIT*, 32(1):10–14, 1992. doi: 10.1007/BF01995104. URL <https://doi.org/10.1007/BF01995104>.
- Alex Hubers and J. Garrett Morris. Generic programming with extensible data types; or, making ad hoc extensible data types less ad hoc. *CoRR*, abs/2307.08759, 2023. doi: 10.48550/arXiv.2307.08759. URL <https://doi.org/10.48550/arXiv.2307.08759>.
- Antonius J. C. Hurkens. A simplification of girard’s paradox. In *Typed Lambda Calculi and Applications, Second International Conference on Typed Lambda Calculi and Applications, TLCA ’95, Edinburgh, UK, April 10-12, 1995, Proceedings*, pages 266–278, 1995. doi: 10.1007/BFb0014058. URL <https://doi.org/10.1007/BFb0014058>.
- Stefan Monnier and Nathaniel Bos. Is impredicativity implicitly implicit? In Marc Bezem and Assia Mahboubi, editors, *25th International Conference on Types for Proofs and Programs, TYPES 2019, June 11-14, 2019, Oslo, Norway*, volume 175 of *LIPIcs*, pages 9:1–9:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. ISBN 978-3-95977-158-0. doi: 10.4230/LIPIcs.TYPES.2019.9. URL <https://doi.org/10.4230/LIPIcs.TYPES.2019.9>.
- Hongwei Xi and Frank Pfenning. Dependent types in practical programming. In *POPL ’99, Proceedings of the 26th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, San Antonio, TX, USA, January 20-22, 1999*, pages 214–227, 1999. doi: 10.1145/292540.292560. URL <https://doi.org/10.1145/292540.292560>.

A The static semantics of R_ω

A.1 Syntax

The syntax of $R_\omega(\mathcal{T})$ is given in Figure 4.

Term variables x	Type variables α	Labels ℓ	Directions $d \in \{\mathbf{L}, \mathbf{R}\}$
Kinds	$\kappa ::= \star \mid \mathbf{L} \mid \mathbf{R}^\kappa \mid \kappa \rightarrow \kappa$		
Predicates	$\pi, \psi ::= \rho \lesssim_d \rho \mid \rho \odot \rho \sim \rho$		
Types	$\phi, \tau, v, \rho, \xi ::= \alpha \mid (\rightarrow) \mid \pi \Rightarrow \tau \mid \forall \alpha : \kappa. \tau \mid \lambda \alpha : \kappa. \tau \mid \tau \tau$ $\mid \ell \mid [\xi] \mid \xi \triangleright \tau \mid \{\tau_1, \dots, \tau_n\} \mid \Pi \rho \mid \Sigma \rho$		
Terms	$M, N ::= x \mid \lambda x : \tau. M \mid M N \mid \Lambda \alpha : \kappa. M \mid M [\tau]$ $\mid \ell \mid M \triangleright M \mid M / M \mid \mathbf{prj}_d M \mid M ++ M \mid \mathbf{inj}_d M \mid M \nabla M$ $\mid \mathbf{syn}_\phi M \mid \mathbf{ana}_\phi M \mid \mathbf{fold} M M M M$		
Environments	$\Gamma ::= \varepsilon \mid \Gamma, \alpha : \kappa \mid \Gamma, x : \tau \mid \Gamma, \pi$		

Figure 4: Syntax

A.2 Types and Kinds

Figure 5 gives rules for context formation ($\vdash \Gamma$), kinding ($\Gamma \vdash \tau : \kappa$), and predicate formation ($\Gamma \vdash \pi$), parameterized by row theory \mathcal{T} .

$$\begin{array}{c}
\boxed{\vdash \Gamma} \\
\\
\text{(C-EMP)} \frac{}{\vdash \varepsilon} \quad \text{(C-TVAR)} \frac{\vdash \Gamma}{\vdash \Gamma, \alpha : \kappa} \quad \text{(C-VAR)} \frac{\vdash \Gamma \quad \Gamma \vdash \tau : \star}{\vdash \Gamma, x : \tau} \quad \text{(C-PRED)} \frac{\vdash \Gamma \quad \Gamma \vdash \pi}{\vdash \Gamma, \pi} \\
\\
\boxed{\Gamma \vdash \tau : \kappa} \quad \boxed{\Gamma \vdash \pi} \\
\\
\text{(K-VAR)} \frac{\vdash \Gamma \quad \alpha : \kappa \in \Gamma}{\Gamma \vdash \alpha : \kappa} \quad \text{(K-}(\rightarrow)\text{)} \frac{\vdash \Gamma}{\Gamma \vdash (\rightarrow) : \star \rightarrow \star \rightarrow \star} \quad \text{(K-}\Rightarrow\text{)} \frac{\Gamma \vdash \pi \quad \Gamma, \pi \vdash \tau : \star}{\Gamma \vdash \pi \Rightarrow \tau : \star} \\
\\
\text{(K-}\forall\text{)} \frac{\Gamma, \alpha : \kappa \vdash \tau : \star}{\Gamma \vdash \forall \alpha : \kappa. \tau : \star} \quad \text{(K-}\rightarrow I\text{)} \frac{\Gamma, \alpha : \kappa_1 \vdash \tau : \kappa_2}{\Gamma \vdash \lambda \alpha : \kappa_1. \tau : \kappa_1 \rightarrow \kappa_2} \quad \text{(K-}\rightarrow E\text{)} \frac{\Gamma \vdash \tau_1 : \kappa_1 \rightarrow \kappa_2 \quad \Gamma \vdash \tau_2 : \kappa_1}{\Gamma \vdash \tau_1 \tau_2 : \kappa_2} \\
\\
\text{(K-LAB)} \frac{\vdash \Gamma}{\Gamma \vdash \ell : \mathbf{L}} \quad \text{(K-SING)} \frac{\Gamma \vdash \xi : \mathbf{L}}{\Gamma \vdash \lfloor \xi \rfloor : \star} \quad \text{(K-LTY)} \frac{\Gamma \vdash \xi : \mathbf{L} \quad \Gamma \vdash \tau : \kappa}{\Gamma \vdash \xi \triangleright \tau : \kappa} \quad \text{(K-ROW)} \frac{\Gamma \vdash_{\mathcal{T}} \{\overline{\xi \triangleright \tau}\} : \mathbf{R}^\kappa}{\Gamma \vdash \{\xi \triangleright \tau\} : \mathbf{R}^\kappa} \\
\\
\text{(K-II)} \frac{\Gamma \vdash \rho : \mathbf{R}^\kappa}{\Gamma \vdash \Pi \rho : \kappa} \quad \text{(K-}\Sigma\text{)} \frac{\Gamma \vdash \rho : \mathbf{R}^\kappa}{\Gamma \vdash \Sigma \rho : \kappa} \quad \text{(K-LIFT}_1\text{)} \frac{\Gamma \vdash \rho : \mathbf{R}^{\kappa_1 \rightarrow \kappa_2} \quad \Gamma \vdash \tau : \kappa_1}{\Gamma \vdash \rho \tau : \mathbf{R}^{\kappa_2}} \\
\\
\text{(K-LIFT}_2\text{)} \frac{\Gamma \vdash \phi : \kappa_1 \rightarrow \kappa_2 \quad \Gamma \vdash \rho : \mathbf{R}^{\kappa_1}}{\Gamma \vdash \phi \rho : \mathbf{R}^{\kappa_2}} \quad \text{(K-}\lesssim_d\text{)} \frac{\Gamma \vdash \rho_i : \mathbf{R}^\kappa}{\Gamma \vdash \rho_1 \lesssim_d \rho_2} \quad \text{(K-}\odot\text{)} \frac{\Gamma \vdash \rho_i : \mathbf{R}^\kappa}{\Gamma \vdash \rho_1 \odot \rho_2 \sim \rho_3}
\end{array}$$

Figure 5: Contexts and kinding.

$$\begin{array}{c}
\boxed{\tau \equiv \tau} \quad \boxed{\pi \equiv \pi} \\
\\
(\text{E-REFL}) \frac{}{\tau \equiv \tau} \quad (\text{E-SYM}) \frac{\tau_1 \equiv \tau_2}{\tau_2 \equiv \tau_1} \quad (\text{E-TRANS}) \frac{\tau_1 \equiv \tau_2 \quad \tau_2 \equiv \tau_3}{\tau_1 \equiv \tau_3} \quad (\text{E-}\beta) \frac{}{(\lambda \alpha : \kappa. \tau) v \equiv \tau[v/\alpha]} \\
\\
(\text{E-}\xi_{\Rightarrow}) \frac{\pi_1 \equiv \pi_2 \quad \tau_1 \equiv \tau_2}{\pi_1 \Rightarrow \tau_1 \equiv \pi_2 \Rightarrow \tau_2} \quad (\text{E-}\xi_{\forall}) \frac{\tau[\gamma/\alpha] \equiv v[\gamma/\beta]}{\forall \alpha : \kappa. \tau \equiv \forall \beta : \kappa. v} (\gamma \notin fv(\tau, v)) \quad (\text{E-}\xi_{\text{APP}}) \frac{\tau_i \equiv v_i}{\tau_1 \tau_2 \equiv v_1 v_2} \\
\\
(\text{E-}\xi_{\triangleright}) \frac{\xi_1 \equiv \xi_2 \quad \tau_1 \equiv \tau_2}{\xi_1 \triangleright \tau_1 \equiv \xi_2 \triangleright \tau_2} \quad (\text{E-ROW}) \frac{\overline{\{\xi_i \triangleright \tau_i\}} \equiv_{\mathcal{T}} \overline{\{\xi'_j \triangleright \tau'_j\}}}{\overline{\{\xi_i \triangleright \tau_i\}} \equiv \overline{\{\xi'_j \triangleright \tau'_j\}}} \quad (\text{E-}\xi_{[\cdot]}) \frac{\xi_1 \equiv \xi_2}{[\xi_1] \equiv [\xi_2]} \\
\\
(\text{E-LIFT}_1) \frac{}{\{\xi \triangleright \phi\} \tau \equiv \{\xi \triangleright \phi \tau\}} \quad (\text{E-LIFT}_2) \frac{}{\phi \{\xi \triangleright \tau\} \equiv \{\xi \triangleright \phi \tau\}} \\
\\
(\text{E-}\xi_{\Pi\Sigma}) \frac{\rho_1 \equiv \rho_2}{K \rho_1 \equiv K \rho_2} \quad (\text{E-LIFT}_3) \frac{}{(K \rho) \tau \equiv K(\rho \tau)} \quad (\text{E-SING}) \frac{}{K \{\xi \triangleright \tau\} \equiv \xi \triangleright \tau} \quad (K \in \{\Pi, \Sigma\}) \\
\\
(\text{E-}\xi_{\lesssim_d}) \frac{\tau_i \equiv v_i}{\tau_1 \lesssim_d \tau_2 \equiv v_1 \lesssim_d v_2} \quad (\text{E-}\xi_{\odot}) \frac{\tau_i \equiv v_i}{\tau_1 \odot \tau_2 \sim \tau_3 \equiv v_1 \odot v_2 \sim v_3}
\end{array}$$

Figure 6: Type and predicate equivalence

A.3 Terms

$$\boxed{\Gamma \vdash M : \tau}$$

$$\begin{array}{c}
(\text{T-VAR}) \frac{\vdash \Gamma \quad x : \tau \in \Gamma}{\Gamma \vdash x : \tau} \quad (\text{T-}\rightarrow I) \frac{\Gamma \vdash \tau_1 : \star \quad \Gamma, x : \tau_1 \vdash M : \tau_2}{\Gamma \vdash \lambda x : \tau_1. M : \tau_1 \rightarrow \tau_2} \quad (\text{T-}\rightarrow E) \frac{\Gamma \vdash M_1 : \tau_1 \rightarrow \tau_2 \quad \Gamma \vdash M_2}{\Gamma \vdash M_1 M_2 : \tau_2} \\
(\text{T-}\equiv) \frac{\Gamma \vdash M : \tau \quad \tau \equiv v}{\Gamma \vdash M : v} \quad (\text{T-}\Rightarrow I) \frac{\Gamma \vdash \pi \quad \Gamma, \pi \vdash M : \tau}{\Gamma \vdash M : \pi \Rightarrow \tau} \quad (\text{T-}\Rightarrow E) \frac{\Gamma \vdash M : \pi \Rightarrow \tau \quad \Gamma \Vdash_{\mathcal{T}} \pi}{\Gamma \vdash M : \tau} \\
(\text{T-}\forall I) \frac{\Gamma, \alpha : \kappa \vdash M : \tau}{\Gamma \vdash \Lambda \alpha : \kappa. M : \forall \alpha : \kappa. \tau} \quad (\text{T-}\forall E) \frac{\Gamma \vdash M : \forall \alpha : \kappa. \tau \quad \Gamma \vdash v : \kappa}{\Gamma \vdash M[v] : \tau[v/\alpha]} \\
(\text{T-SING}) \frac{\vdash \Gamma}{\Gamma \vdash \ell : [\ell]} \quad (\text{T-}\triangleright I) \frac{\Gamma \vdash M_1 : [\ell] \quad \Gamma \vdash M_2 : \tau}{\Gamma \vdash M_1 \triangleright M_2 : \ell \triangleright \tau} \quad (\text{T-}\triangleright E) \frac{\Gamma \vdash M_1 : \ell \triangleright \tau \quad \Gamma \vdash M_2 : [\ell]}{\Gamma \vdash M_1 / M_2 : \tau} \\
(\text{T-II}E) \frac{\Gamma \vdash M : \Pi \rho_1 \quad \Gamma \Vdash_{\mathcal{T}} \rho_2 \lesssim_d \rho_1}{\Gamma \vdash \text{prj}_d M : \Pi \rho_2} \quad (\text{T-II}I) \frac{\Gamma \vdash M_1 : \Pi \rho_1 \quad \Gamma \vdash M_2 : \Pi \rho_2 \quad \Gamma \Vdash_{\mathcal{T}} \rho_1 \odot \rho_2 \sim \rho_3}{\Gamma \vdash M_1 ++ M_2 : \Pi \rho_3} \\
(\text{T-}\Sigma I) \frac{\Gamma \vdash M : \Sigma \rho_1 \quad \Gamma \Vdash_{\mathcal{T}} \rho_1 \lesssim \rho_2}{\Gamma \vdash \text{inj} M : \Sigma \rho_2} \quad (\text{T-}\Sigma E) \frac{\Gamma \vdash M_1 : \Sigma \rho_1 \rightarrow \tau \quad \Gamma \vdash M_2 : \Sigma \rho_2 \rightarrow \tau \quad \Gamma \Vdash_{\mathcal{T}} \rho_1 \odot \rho_2 \sim}{\Gamma \vdash M_1 \nabla M_2 : \Sigma \rho_3 \rightarrow \tau} \\
(\text{T-ana}) \frac{\Gamma \vdash \rho : \mathbf{R}^\kappa \quad \Gamma \vdash \phi : \kappa \rightarrow \star \quad \Gamma \vdash M : \forall l : \mathbf{L}, u : \kappa, y_1, z, y_2 : \mathbf{R}^\kappa. (y_1 \odot \{l \triangleright u\} \sim z, z \odot y_2 \sim \rho) \Rightarrow [l] \rightarrow \phi u \rightarrow \tau}{\Gamma \vdash \text{ana}_\phi M : \Sigma(\phi \rho) \rightarrow \tau} \\
(\text{T-syn}) \frac{\Gamma \vdash \rho : \mathbf{R}^\kappa \quad \Gamma \vdash \phi : \kappa \rightarrow \star \quad \Gamma \vdash M : \forall l : \mathbf{L}, u : \kappa, y_1, z, y_2 : \mathbf{R}^\kappa. (y_1 \odot \{l \triangleright u\} \sim z, z \odot y_2 \sim \rho) \Rightarrow [l] \rightarrow \phi u}{\Gamma \vdash \text{syn}_\phi M : \Pi(\phi \rho)} \\
(\text{T-fold}) \frac{M_1 : \forall l : \mathbf{L}, t : \star, y_1, z, y_2 : \mathbf{R}^\kappa. (y_1 \odot \{l \triangleright u\} \sim z, z \odot y_2 \sim \rho) \Rightarrow [l] \rightarrow t \rightarrow v \quad \Gamma \vdash M_2 : v \rightarrow v \rightarrow v \quad \Gamma \vdash M_3 : v \quad \Gamma \vdash N : \Pi \rho}{\Gamma \vdash \text{fold} M_1 M_2 M_3 N : v}
\end{array}$$

Figure 7: Typing

Minimal Rows

Figure 8 gives the minimal row theory \mathcal{M} .

$$\begin{array}{c}
\boxed{\Gamma \vdash_{\mathbf{m}} \rho : \kappa} \quad \boxed{\rho \equiv_{\mathbf{m}} \rho} \\
\\
(\text{K-MROW}) \frac{\Gamma \vdash \xi : \mathbf{L} \quad \Gamma \vdash \tau : \kappa}{\Gamma \vdash_{\mathbf{m}} \{\xi \triangleright \tau\} : \mathbf{R}^\kappa} \quad (\text{E-MROW}) \frac{\xi \equiv \xi' \quad \tau \equiv \tau'}{\{\xi \triangleright \tau\} \equiv_{\mathbf{m}} \{\xi' \triangleright \tau'\}} \\
\\
\boxed{\Gamma \Vdash_{\mathbf{m}} \pi} \\
\\
(\text{N-AX}) \frac{\pi \in \Gamma}{\Gamma \Vdash_{\mathbf{m}} \pi} \quad (\text{N-REFL}) \frac{}{\Gamma \Vdash_{\mathbf{m}} \rho \lesssim_d \rho} \quad (\text{N-TRANS}) \frac{\Gamma \Vdash_{\mathbf{m}} \rho_1 \lesssim_d \rho_2 \quad \Gamma \Vdash_{\mathbf{m}} \rho_2 \lesssim_d \rho_3}{\Gamma \Vdash_{\mathbf{m}} \rho_1 \lesssim_d \rho_3} \\
\\
(\text{N-}\equiv) \frac{\Gamma \Vdash_{\mathbf{m}} \pi_1 \quad \pi_1 \equiv \pi_2}{\Gamma \Vdash_{\mathbf{m}} \pi_2} \quad (\text{N-}\lesssim_{\text{LIFT}_1}) \frac{\Gamma \Vdash_{\mathbf{m}} \rho_1 \lesssim_d \rho_2}{\Gamma \Vdash_{\mathbf{m}} \phi \rho_1 \lesssim_d \phi \rho_2} \quad (\text{N-}\lesssim_{\text{LIFT}_2}) \frac{\Gamma \Vdash_{\mathbf{m}} \rho_1 \lesssim_d \rho_2}{\Gamma \Vdash_{\mathbf{m}} \rho_1 \tau \lesssim_d \rho_2 \tau} \\
\\
(\text{N-}\odot_{\text{LIFT}_1}) \frac{\Gamma \Vdash_{\mathbf{m}} \rho_1 \odot \rho_2 \sim \rho_3}{\Gamma \Vdash_{\mathbf{m}} \rho_1 \tau \odot \rho_2 \tau \sim \rho_3 \tau} \quad (\text{N-}\odot_{\text{LIFT}_2}) \frac{\Gamma \Vdash_{\mathbf{m}} \rho_1 \odot \rho_2 \sim \rho_3}{\Gamma \Vdash_{\mathbf{m}} \phi \rho_1 \odot \phi \rho_2 \sim \phi \rho_3} \\
\\
(\text{N-}\odot_{\lesssim_{\mathbf{L}}}) \frac{\Gamma \Vdash_{\mathbf{m}} \rho_1 \odot \rho_2 \sim \rho_3}{\Gamma \Vdash_{\mathbf{m}} \rho_1 \lesssim_{\mathbf{L}} \rho_3} \quad (\text{N-}\odot_{\lesssim_{\mathbf{R}}}) \frac{\Gamma \Vdash_{\mathbf{m}} \rho_1 \odot \rho_2 \sim \rho_3}{\Gamma \Vdash_{\mathbf{m}} \rho_2 \lesssim_{\mathbf{R}} \rho_3}
\end{array}$$

Figure 8: Minimal row theory $\mathcal{M} = \langle \vdash_{\mathbf{m}}, \equiv_{\mathbf{m}}, \Vdash_{\mathbf{m}} \rangle$