Recursive Rows in Rome

AH & JGM

ACM Reference Format:

AH & JGM. 2023. Recursive Rows in Rome. 1, 1 (November 2023), 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

1.1 The expression problem, in full

1.1.1 Seeking solutions sans encodings.

1.2 Recursion and rows

1.2.1 Row type systems with term- or type-level μ. There are none.

1.2.2 Structural typing of objects in recursive record calculi.

1.3 Challenges to practical extensibility

1.3.1 Polymorphic variants in OCaml.

1.3.2 Inheritance is not subtyping.

2 ROME: A ROW THEORETIC FOUNDATION FOR (CO)INDUCTIVE DATA TYPES

3 Rω—HIGHER ORDERED ROWS

Recursive types have a well known semantics as the least fixed-points of type-level operators. Rω is the only row calculus (to our knowledge) to include an (explicit) type-level λ operator. Like with $F\omega$, this necessarily sacrifices desirable metatheoretic properties, such as principality of types; hence, like $F\omega$, Rω may serve as a highly expressive intermediate or target language. Correspondingly, we perceive the addition of recursive terms and types to Rω to aid the adoption of recursion in surface languages.

We review the relevant syntax and typing of Rω now.

(Todo). It will be a challenge to trim this down, as [Hubers and Morris 2023] does with [Morris and McKinna 2019].

Author's address: AH & JGM.

XXXX-XXXX/2023/11-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

3.1 Syntax

The syntax of $\mathrm{R}\omega(\mathcal{T})$ is given in Figure 6.

$$
\begin{array}{llcl}
\multicolumn{4}{c}{\text{Term variables } x \qquad \text{Type variables } \alpha \qquad \text{Labels } \ell \qquad \text{Directions } d \in \{\mathsf{L}, \mathsf{R}\}} \\[1ex]
\text{Kinds} & \kappa & ::= & \star \mid \mathsf{L} \mid \mathsf{R}^{\kappa} \mid \kappa \to \kappa \\
\text{Predicates} & \pi, \psi & ::= & \rho \lesssim_d \rho \mid \rho \odot \rho \sim \rho \\
\text{Types} & \phi, \tau, \upsilon, \rho, \xi & ::= & \alpha \mid (\to) \mid \pi \Rightarrow \tau \mid \forall \alpha : \kappa . \tau \mid \lambda \alpha : \kappa . \tau \mid \tau\, \tau \\
& & \mid & \ell \mid \lfloor \xi \rfloor \mid \xi \triangleright \tau \mid \{\tau_1, \ldots, \tau_n\} \mid \Pi \rho \mid \Sigma \rho \\
\text{Terms} & H, M, N, P & ::= & x \mid \lambda x : \tau . M \mid M\, N \mid \Lambda \alpha : \kappa . M \mid M\, [\tau] \\
& & \mid & \ell \mid M \triangleright M \mid M / M \mid \mathsf{prj}_d\, M \mid M \mathbin{+\!\!+} M \mid \mathsf{inj}_d\, M \mid M \triangledown M \\
& & \mid & \mathsf{syn}_{\phi}\, M \mid \mathsf{ana}_{\phi}\, M \mid \mathsf{fold}\, M\, M\, M\, M \\
\text{Environments} & \Gamma & ::= & \varepsilon \mid \Gamma, \alpha : \kappa \mid \Gamma, x : \tau \mid \Gamma, \pi
\end{array}
$$

Fig. 1. Syntax

3.2 Types and Kinds

Figure 2 gives rules for context formation ($\vdash \Gamma$), kinding ($\Gamma \vdash \tau : \kappa$), and predicate formation ($\Gamma \vdash \pi$), parameterized by row theory $\mathcal{T}$.

$$\boxed{\vdash \Gamma}$$

$$
(\textsc{c-emp})\,\frac{}{\vdash \varepsilon}
\qquad
(\textsc{c-tvar})\,\frac{\vdash \Gamma}{\vdash \Gamma, \alpha : \kappa}
\qquad
(\textsc{c-var})\,\frac{\vdash \Gamma \quad \Gamma \vdash \tau : \star}{\vdash \Gamma, x : \tau}
\qquad
(\textsc{c-pred})\,\frac{\vdash \Gamma \quad \Gamma \vdash \pi}{\vdash \Gamma, \pi}
$$

$$\boxed{\Gamma \vdash \tau : \kappa}\ \boxed{\Gamma \vdash \pi}$$

$$
(\textsc{k-var})\,\frac{\vdash \Gamma \quad \alpha : \kappa \in \Gamma}{\Gamma \vdash \alpha : \kappa}
\qquad
(\textsc{k-}(\to))\,\frac{\vdash \Gamma}{\Gamma \vdash (\to) : \star \to \star \to \star}
\qquad
(\textsc{k-}{\Rightarrow})\,\frac{\Gamma \vdash \pi \quad \Gamma, \pi \vdash \tau : \star}{\Gamma \vdash \pi \Rightarrow \tau : \star}
$$

$$
(\textsc{k-}\forall)\,\frac{\Gamma, \alpha : \kappa \vdash \tau : \star}{\Gamma \vdash \forall \alpha : \kappa . \tau : \star}
\qquad
(\textsc{k-}{\to}I)\,\frac{\Gamma, \alpha : \kappa_1 \vdash \tau : \kappa_2}{\Gamma \vdash \lambda \alpha : \kappa_1 . \tau : \kappa_1 \to \kappa_2}
\qquad
(\textsc{k-}{\to}E)\,\frac{\Gamma \vdash \tau_1 : \kappa_1 \to \kappa_2 \quad \Gamma \vdash \tau_2 : \kappa_1}{\Gamma \vdash \tau_1\, \tau_2 : \kappa_2}
$$

$$
(\textsc{k-lab})\,\frac{\vdash \Gamma}{\Gamma \vdash \ell : \mathsf{L}}
\qquad
(\textsc{k-sing})\,\frac{\Gamma \vdash \xi : \mathsf{L}}{\Gamma \vdash \lfloor \xi \rfloor : \star}
\qquad
(\textsc{k-lty})\,\frac{\Gamma \vdash \xi : \mathsf{L} \quad \Gamma \vdash \tau : \kappa}{\Gamma \vdash \xi \triangleright \tau : \kappa}
\qquad
(\textsc{k-row})\,\frac{\Gamma \vdash_{\mathcal{T}} \{\overline{\xi \triangleright \tau}\} : \mathsf{R}^{\kappa}}{\Gamma \vdash \{\overline{\xi \triangleright \tau}\} : \mathsf{R}^{\kappa}}
$$

$$
(\textsc{k-}\Pi)\,\frac{\Gamma \vdash \rho : \mathsf{R}^{\kappa}}{\Gamma \vdash \Pi \rho : \kappa}
\qquad
(\textsc{k-}\Sigma)\,\frac{\Gamma \vdash \rho : \mathsf{R}^{\kappa}}{\Gamma \vdash \Sigma \rho : \kappa}
\qquad
(\textsc{k-lift}_1)\,\frac{\Gamma \vdash \rho : \mathsf{R}^{\kappa_1 \to \kappa_2} \quad \Gamma \vdash \tau : \kappa_1}{\Gamma \vdash \rho\, \tau : \mathsf{R}^{\kappa_2}}
$$

$$
(\textsc{k-lift}_2)\,\frac{\Gamma \vdash \phi : \kappa_1 \to \kappa_2 \quad \Gamma \vdash \rho : \mathsf{R}^{\kappa_1}}{\Gamma \vdash \phi\, \rho : \mathsf{R}^{\kappa_2}}
\qquad
(\textsc{k-}{\lesssim_d})\,\frac{\Gamma \vdash \rho_i : \mathsf{R}^{\kappa}}{\Gamma \vdash \rho_1 \lesssim_d \rho_2}
\qquad
(\textsc{k-}\odot)\,\frac{\Gamma \vdash \rho_i : \mathsf{R}^{\kappa}}{\Gamma \vdash \rho_1 \odot \rho_2 \sim \rho_3}
$$

Fig. 2. Contexts and kinding.

$$\boxed{\tau \equiv \tau} \quad \boxed{\pi \equiv \pi}$$

$$(\text{E-REFL})\ \frac{}{\tau \equiv \tau} \qquad (\text{E-SYM})\ \frac{\tau_1 \equiv \tau_2}{\tau_2 \equiv \tau_1} \qquad (\text{E-TRANS})\ \frac{\tau_1 \equiv \tau_2 \quad \tau_2 \equiv \tau_3}{\tau_1 \equiv \tau_3} \qquad (\text{E-}\beta)\ \frac{}{(\lambda\alpha : \kappa.\tau)\ \upsilon \equiv \tau[\upsilon/\alpha]}$$

$$(\text{E-}\xi_{\Rightarrow})\ \frac{\pi_1 \equiv \pi_2 \quad \tau_1 \equiv \tau_2}{\pi_1 \Rightarrow \tau_1 \equiv \pi_2 \Rightarrow \tau_2} \qquad (\text{E-}\xi_{\forall})\ \frac{\tau[\gamma/\alpha] \equiv \upsilon[\gamma/\beta]}{\forall\alpha : \kappa.\tau \equiv \forall\beta : \kappa.\upsilon}\ (\gamma \notin fv(\tau, \upsilon)) \qquad (\text{E-}\xi_{\text{APP}})\ \frac{\tau_i \equiv \upsilon_i}{\tau_1\ \tau_2 \equiv \upsilon_1\ \upsilon_2}$$

$$(\text{E-}\xi_{\triangleright})\ \frac{\xi_1 \equiv \xi_2 \quad \tau_1 \equiv \tau_2}{\xi_1 \triangleright \tau_1 \equiv \xi_2 \triangleright \tau_2} \qquad (\text{E-ROW})\ \frac{\{\overline{\xi_i \triangleright \tau_i}\} \equiv_{\mathcal{T}} \{\overline{\xi'_j \triangleright \tau'_j}\}}{\{\overline{\xi_i \triangleright \tau_i}\} \equiv \{\overline{\xi'_j \triangleright \tau'_j}\}} \qquad (\text{E-}\xi_{\lfloor\cdot\rfloor})\ \frac{\xi_1 \equiv \xi_2}{\lfloor\xi_1\rfloor \equiv \lfloor\xi_2\rfloor}$$

$$(\text{E-LIFT}_1)\ \frac{}{\{\xi \triangleright \phi\}\ \tau \equiv \{\xi \triangleright \phi\ \tau\}} \qquad (\text{E-LIFT}_2)\ \frac{}{\phi\ \{\xi \triangleright \tau\} \equiv \{\xi \triangleright \phi\ \tau\}}$$

$$(\text{E-}\xi_{\Pi\Sigma})\ \frac{\rho_1 \equiv \rho_2}{K\rho_1 \equiv K\rho_2} \qquad (\text{E-LIFT}_3)\ \frac{}{(K\rho)\ \tau \equiv K(\rho\ \tau)} \qquad (\text{E-SING})\ \frac{}{K\{\xi \triangleright \tau\} \equiv \xi \triangleright \tau} \qquad (K \in \{\Pi, \Sigma\})$$

$$(\text{E-}\xi_{\lesssim_d})\ \frac{\tau_i \equiv \upsilon_i}{\tau_1 \lesssim_d \tau_2 \equiv \upsilon_1 \lesssim_d \upsilon_2} \qquad (\text{E-}\xi_{\odot})\ \frac{\tau_i \equiv \upsilon_i}{\tau_1 \odot \tau_2 \sim \tau_3 \equiv \upsilon_1 \odot \upsilon_2 \sim \upsilon_3}$$

Fig. 3. Type and predicate equivalence

3.3 Terms

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

$$(\textsc{t-var})\ \frac{\vdash \Gamma \quad x : \tau \in \Gamma}{\Gamma \vdash x : \tau} \qquad (\textsc{t-}{\longrightarrow}I)\ \frac{\Gamma \vdash \tau_1 : \star \quad \Gamma, x : \tau_1 \vdash M : \tau_2}{\Gamma \vdash \lambda x : \tau_1 . M : \tau_1 \longrightarrow \tau_2} \qquad (\textsc{t-}{\longrightarrow}E)\ \frac{\Gamma \vdash M_1 : \tau_1 \longrightarrow \tau_2 \quad \Gamma \vdash M_2 : \tau_1}{\Gamma \vdash M_1\, M_2 : \tau_2}$$

$$(\textsc{t-}{\equiv})\ \frac{\Gamma \vdash M : \tau \quad \tau \equiv \upsilon}{\Gamma \vdash M : \upsilon} \qquad (\textsc{t-}{\Longrightarrow}I)\ \frac{\Gamma \vdash \pi \quad \Gamma, \pi \vdash M : \tau}{\Gamma \vdash M : \pi \Rightarrow \tau} \qquad (\textsc{t-}{\Longrightarrow}E)\ \frac{\Gamma \vdash M : \pi \Rightarrow \tau \quad \Gamma \Vdash_{\mathcal{T}} \pi}{\Gamma \vdash M : \tau}$$

$$(\textsc{t-}\forall I)\ \frac{\Gamma, \alpha : \kappa \vdash M : \tau}{\Gamma \vdash \Lambda \alpha : \kappa . M : \forall \alpha : \kappa . \tau} \qquad (\textsc{t-}\forall E)\ \frac{\Gamma \vdash M : \forall \alpha : \kappa . \tau \quad \Gamma \vdash \upsilon : \kappa}{\Gamma \vdash M\,[\upsilon] : \tau[\upsilon/\alpha]}$$

$$(\textsc{t-sing})\ \frac{\vdash \Gamma}{\Gamma \vdash \ell : \lfloor \ell \rfloor} \qquad (\textsc{t-}{\triangleright}I)\ \frac{\Gamma \vdash M_1 : \lfloor \ell \rfloor \quad \Gamma \vdash M_2 : \tau}{\Gamma \vdash M_1 \triangleright M_2 : \ell \triangleright \tau} \qquad (\textsc{t-}{\triangleright}E)\ \frac{\Gamma \vdash M_1 : \ell \triangleright \tau \quad \Gamma \vdash M_2 : \lfloor \ell \rfloor}{\Gamma \vdash M_1 / M_2 : \tau}$$

$$(\textsc{t-}\Pi E)\ \frac{\Gamma \vdash M : \Pi \rho_1 \quad \Gamma \Vdash_{\mathcal{T}} \rho_2 \lesssim_d \rho_1}{\Gamma \vdash \mathsf{prj}_d\, M : \Pi \rho_2} \qquad (\textsc{t-}\Pi I)\ \frac{\Gamma \vdash M_1 : \Pi \rho_1 \quad \Gamma \vdash M_2 : \Pi \rho_2 \quad \Gamma \Vdash_{\mathcal{T}} \rho_1 \odot \rho_2 \sim \rho_3}{\Gamma \vdash M_1 +\!\!+\, M_2 : \Pi \rho_3}$$

$$(\textsc{t-}\Sigma I)\ \frac{\Gamma \vdash M : \Sigma \rho_1 \quad \Gamma \Vdash_{\mathcal{T}} \rho_1 \lesssim \rho_2}{\Gamma \vdash \mathsf{inj}\, M : \Sigma \rho_2} \qquad (\textsc{t-}\Sigma E)\ \frac{\Gamma \vdash M_1 : \Sigma \rho_1 \longrightarrow \tau \quad \Gamma \vdash M_2 : \Sigma \rho_2 \longrightarrow \tau \quad \Gamma \Vdash_{\mathcal{T}} \rho_1 \odot \rho_2 \sim \rho_3}{\Gamma \vdash M_1 \triangledown M_2 : \Sigma \rho_3 \longrightarrow \tau}$$

$$(\textsc{t-ana})\ \frac{\begin{array}{c}\Gamma \vdash \rho : \mathsf{R}^\kappa \quad \Gamma \vdash \phi : \kappa \to \star \\ \Gamma \vdash M : \forall l : \mathsf{L}, u : \kappa, y_1, z, y_2 : \mathsf{R}^\kappa .\, (y_1 \odot \{l \triangleright u\} \sim z, z \odot y_2 \sim \rho) \Rightarrow \lfloor l \rfloor \to \phi\, u \to \tau\end{array}}{\Gamma \vdash \mathsf{ana}_\phi\, M : \Sigma(\phi\, \rho) \longrightarrow \tau}$$

$$(\textsc{t-syn})\ \frac{\begin{array}{c}\Gamma \vdash \rho : \mathsf{R}^\kappa \quad \Gamma \vdash \phi : \kappa \to \star \\ \Gamma \vdash M : \forall l : \mathsf{L}, u : \kappa, y_1, z, y_2 : \mathsf{R}^\kappa .\, (y_1 \odot \{l \triangleright u\} \sim z, z \odot y_2 \sim \rho) \Rightarrow \lfloor l \rfloor \to \phi\, u\end{array}}{\Gamma \vdash \mathsf{syn}_\phi\, M : \Pi(\phi\, \rho)}$$

$$(\textsc{t-fold})\ \frac{\begin{array}{c}M_1 : \forall l : \mathsf{L}, t : \star, y_1, z, y_2 : \mathsf{R}^\kappa .\, (y_1 \odot \{l \triangleright u\} \sim z, z \odot y_2 \sim \rho) \Rightarrow \lfloor l \rfloor \longrightarrow t \to \upsilon \\ \Gamma \vdash M_2 : \upsilon \to \upsilon \to \upsilon \quad \Gamma \vdash M_3 : \upsilon \quad \Gamma \vdash N : \Pi \rho\end{array}}{\Gamma \vdash \mathsf{fold}\, M_1\, M_2\, M_3\, N : \upsilon}$$

Fig. 4. Typing

Minimal Rows

Figure 5 gives the minimal row theory $\mathcal{M}$.

$$\boxed{\Gamma \vdash_{\mathsf{m}} \rho : \kappa} \quad \boxed{\rho \equiv_{\mathsf{m}} \rho}$$

$$(\textsc{k-mrow})\ \frac{\Gamma \vdash \xi : \mathsf{L} \qquad \Gamma \vdash \tau : \kappa}{\Gamma \vdash_{\mathsf{m}} \{\xi \triangleright \tau\} : \mathsf{R}^{\kappa}} \qquad (\textsc{e-mrow})\ \frac{\xi \equiv \xi' \qquad \tau \equiv \tau'}{\{\xi \triangleright \tau\} \equiv_{\mathsf{m}} \{\xi' \triangleright \tau'\}}$$

$$\boxed{\Gamma \Vdash_{\mathsf{m}} \pi}$$

$$(\textsc{n-ax})\ \frac{\pi \in \Gamma}{\Gamma \Vdash_{\mathsf{m}} \pi} \qquad (\textsc{n-refl})\ \frac{}{\Gamma \Vdash_{\mathsf{m}} \rho \lesssim_d \rho} \qquad (\textsc{n-trans})\ \frac{\Gamma \Vdash_{\mathsf{m}} \rho_1 \lesssim_d \rho_2 \qquad \Gamma \Vdash_{\mathsf{m}} \rho_2 \lesssim_d \rho_3}{\Gamma \Vdash_{\mathsf{m}} \rho_1 \lesssim_d \rho_3}$$

$$(\textsc{n-}\equiv)\ \frac{\Gamma \Vdash_{\mathsf{m}} \pi_1 \qquad \pi_1 \equiv \pi_2}{\Gamma \Vdash_{\mathsf{m}} \pi_2} \qquad (\textsc{n-}\lesssim\textsc{lift}_1)\ \frac{\Gamma \Vdash_{\mathsf{m}} \rho_1 \lesssim_d \rho_2}{\Gamma \Vdash_{\mathsf{m}} \phi\, \rho_1 \lesssim_d \phi\, \rho_2} \qquad (\textsc{n-}\lesssim\textsc{lift}_2)\ \frac{\Gamma \Vdash_{\mathsf{m}} \rho_1 \lesssim_d \rho_2}{\Gamma \Vdash_{\mathsf{m}} \rho_1\, \tau \lesssim_d \rho_2\, \tau}$$

$$(\textsc{n-}\odot\textsc{lift}_1)\ \frac{\Gamma \Vdash_{\mathsf{m}} \rho_1 \odot \rho_2 \sim \rho_3}{\Gamma \Vdash_{\mathsf{m}} \rho_1\, \tau \odot \rho_2\, \tau \sim \rho_3\, \tau} \qquad (\textsc{n-}\odot\textsc{lift}_2)\ \frac{\Gamma \Vdash_{\mathsf{m}} \rho_1 \odot \rho_2 \sim \rho_3}{\Gamma \Vdash_{\mathsf{m}} \phi\, \rho_1 \odot \phi\, \rho_2 \sim \phi\, \rho_3}$$

$$(\textsc{n-}\odot\lesssim_{\mathsf{L}})\ \frac{\Gamma \Vdash_{\mathsf{m}} \rho_1 \odot \rho_2 \sim \rho_3}{\Gamma \Vdash_{\mathsf{m}} \rho_1 \lesssim_{\mathsf{L}} \rho_3} \qquad (\textsc{n-}\odot\lesssim_{\mathsf{R}})\ \frac{\Gamma \Vdash_{\mathsf{m}} \rho_1 \odot \rho_2 \sim \rho_3}{\Gamma \Vdash_{\mathsf{m}} \rho_2 \lesssim_{\mathsf{R}} \rho_3}$$

Fig. 5. Minimal row theory $\mathcal{M} = \langle \vdash_{\mathsf{m}}, \equiv_{\mathsf{m}}, \Vdash_{\mathsf{m}} \rangle$

4 IX: THE INDEX CALCULUS

4.1 Syntax

$$\text{Term variables} \quad x\ \alpha$$

$$\begin{array}{llcl}
\text{Sorts} & \sigma & ::= & \star \mid \mathcal{U} \\
\text{Terms} & M, N, T & ::= & x \mid \star \mid \mathrm{Nat} \mid \mathrm{Zero} \mid \mathrm{Suc}\, M \mid \\
 & & & \mathrm{Ix}\, M \mid \mathrm{FZero} \mid \mathrm{FSuc}\, M \mid \\
 & & & \top \mid \mathtt{tt} \mid \\
 & & & \Pi\alpha : T.N \mid \lambda x : T.N \mid M\, N \mid \\
 & & & \Sigma\alpha : T.M \mid (\alpha : T, M) \mid M.1 \mid M.2 \\
 & & & M + N \mid \mathrm{left}\, M \mid \mathrm{right}\, M \mid \mathrm{case}\, M\, M\, M \\
 & & & M \equiv N \mid \mathrm{refl} \mid \ldots \\
\text{Environments} & \Gamma & ::= & \varepsilon \mid \Gamma, \alpha : T
\end{array}$$

Fig. 6. Syntax

Universes. Ix is stratified by two universes: $\star$, the type of types, and $\mathcal{U}$, the type of $\star$. This is analogous to e.g. Coq/CoC, in which both `Set` and `Prop` are impredicative and have type `Type`.

$$\begin{tikzcd}
 & \mathtt{Type} & \\
\mathtt{Set} \arrow[ur] & & \mathtt{Prop} \arrow[ul]
\end{tikzcd}$$

Ix is exactly the same modulo renaming and without a need for (the proof-irrelevant universe of) `Prop`.

4.2 Typing

Figure 7 gives the rules for three judgments: $\vdash \Gamma$, which states that typing environment Γ is well-formed; $\Gamma \vdash M : \sigma$, which states that M is a type with sort σ; and $\Gamma \vdash M : N$, which states that M has type N. You may observe that one could merge the judgments $\Gamma \vdash M : \sigma$ and $\Gamma \vdash M : N$ by incorporating the syntax of σ into the term language. (This is not a crazy thing to do in dependent type theory.) I find it is helpful to separate them for two reasons:

(1) The judgment $\Gamma \vdash M : \sigma$ is not actually analogous to a typing judgment, but rather to the judgment $\Gamma \vdash M$ type one would see in MLTT/CoC. We are asserting that M is a type with sort σ.

(2) Definitional equality is different for the two judgments. Definitional equality of types, e.g. $\Gamma \vdash \Pi M_1 N_1 = \Pi M_2 N_2 : \sigma$, for example, holds if the components are equal; that is to say, definitional equality of types is mostly congruence rules. Definitional equality of terms are *computational laws*, e.g., the computational law for (left) projection of dependent sums is given by $\Gamma \vdash (x : A, M).1 = x : A$.

(3) Separating the two in the mechanization allows me to index *terms* by *types*:

```
data Type : Context → Pre.Term → Set
data Term : (d : Context) → {t : Pre.Term} → Type d t  → Set
```

Defining terms and types as one AST (indexed by two `Pre.Terms`) means that *terms can not be intrinsically typed*, as we are forced to define

```
data Term : (d : Context) → Pre.Term → Pre.Term → Set
```

There is a lot of funny business one can do with induction-induction and induction-recursion, but to my knowledge you may not *index* a type by itself in Agda.

$$\boxed{\vdash \Gamma}$$

$$(\textsc{emp})\,\frac{}{\vdash \varepsilon} \qquad (\textsc{var})\,\frac{\vdash \Gamma \quad \Gamma \vdash M : \sigma}{\vdash \Gamma, x : M}$$

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

$$(\star)\,\frac{}{\Gamma \vdash \star : \mathcal{U}} \qquad (\top_\star)\,\frac{}{\Gamma \vdash \top : \star} \qquad (\top_{\mathcal{U}})\,\frac{}{\Gamma \vdash \top : \mathcal{U}}$$

$$(\mathit{Var})\,\frac{\alpha : \sigma \in \Gamma}{\Gamma \vdash \alpha : \sigma} \qquad (\textsc{Nat})\,\frac{}{\Gamma \vdash \mathrm{Nat} : \star} \qquad (\mathrm{Ix})\,\frac{\Gamma \vdash n : \mathrm{Nat}}{\Gamma \vdash \mathrm{Ix}\, n : \star}$$

$$(\Pi)\,\frac{\Gamma \vdash M : \sigma \quad \Gamma, \alpha : M \vdash N : \star}{\Gamma \vdash \Pi\alpha : M.N : \star} \qquad (\Sigma)\,\frac{\Gamma \vdash M : \sigma \quad \Gamma, \alpha : M \vdash N : \star}{\Gamma \vdash \Sigma\alpha : M.N : \star}$$

$$(+)\,\frac{\Gamma \vdash M : \star \quad \Gamma \vdash N : \star}{\Gamma \vdash M + N : \star} \qquad (\equiv)\,\frac{\Gamma \vdash M_1 : N_1 \quad \Gamma \vdash N_1 : \star \quad \Gamma \vdash M_2 : N_2 \quad \Gamma \vdash N_2 : \star}{\Gamma \vdash M_1 \equiv M_2 : \star}$$

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

$$(\mathit{Var})\,\frac{x : M \in \Gamma}{\Gamma \vdash x : M} \qquad (\mathtt{tt})\,\frac{}{\Gamma \vdash \mathtt{tt} : \top}$$

$$(\mathrm{Zero})\,\frac{}{\Gamma \vdash \mathrm{Zero} : \mathrm{Nat}} \qquad (\mathrm{Suc})\,\frac{\Gamma \vdash n : \mathrm{Nat}}{\Gamma \vdash \mathrm{Suc}\, n : \mathrm{Nat}}$$

$$(\mathrm{FZero})\,\frac{\Gamma \vdash n : \mathrm{Nat}}{\Gamma \vdash \mathrm{FZero} : \mathrm{Ix}\,(\mathrm{Suc}\, n)} \qquad (\mathrm{FSuc})\,\frac{\Gamma \vdash n : \mathrm{Nat} \quad \Gamma \vdash i : \mathrm{Ix}\, n}{\Gamma \vdash \mathrm{FSuc}\, i : \mathrm{Ix}\,(\mathrm{Suc}\, n)}$$

$$(\Pi I)\,\frac{\Gamma \vdash T : \sigma \quad \Gamma, x : T \vdash M : N}{\Gamma \vdash \lambda x : T.M : \Pi(x : T).N} \qquad (\Pi E)\,\frac{\Gamma \vdash M : \Pi(x : T_1).T_2 \quad \Gamma \vdash N : T_1}{\Gamma \vdash M\, N : T_2[N/x]}$$

$$(\Sigma I)\,\frac{\Gamma \vdash M : T_1 \quad \Gamma \vdash N : T_2[M/x]}{\Gamma \vdash (M : T_1, N) : \Sigma(x : T_1).T_2} \qquad (\Sigma E_1)\,\frac{\Gamma \vdash M : \Sigma(x : T_1).T_2}{\Gamma \vdash M.1 : T_1} \qquad (\Sigma E_2)\,\frac{\Gamma \vdash M : \Sigma(x : T_1).T_2}{\Gamma \vdash M.2 : T_2[M.1/x]}$$

$$(\equiv I)\,\frac{\Gamma \vdash M : \sigma}{\Gamma \vdash \mathrm{refl} : M \equiv M} \qquad (\textsc{conv})\,\frac{\Gamma \vdash M : T_1 \quad \Gamma \vdash T_1 = T_2 : \sigma}{\Gamma \vdash M : T_2}$$

$$(\equiv E)\,\frac{\begin{array}{c} \Gamma \vdash P : T_1 \equiv T_2 \\ \Gamma \vdash M : T_1 \\ \Gamma \vdash N : T_1 \\ \Gamma, x : T_1, y : T_1, p : x \equiv y \vdash T : \star \\ \Gamma, z : T_1 \vdash H : T[z/x, z/y, \mathrm{refl}/p] \end{array}}{\Gamma \vdash \mathcal{J}\, H\, M\, N\, P : T[M/x, N/y, P/p]}$$

Fig. 7. Context formation and typing rules for Ix terms

Let the meta-syntax τ denote both sort and term.

$$\boxed{\Gamma \vdash M = N : \sigma}$$

$$(\text{E-REFL})\,\frac{\Gamma \vdash M : \sigma}{\Gamma \vdash M = M : \sigma} \qquad (\text{E-SYM})\,\frac{\Gamma \vdash N = M : \sigma}{\Gamma \vdash M = N : \sigma} \qquad (\text{E-TRANS})\,\frac{\Gamma \vdash M = P : \sigma \quad \Gamma \vdash P = N : \sigma}{\Gamma \vdash M = N : \sigma}$$

$$\boxed{\Gamma \vdash M = N : T}$$

$$(\text{C-REFL})\,\frac{\Gamma \vdash M : T}{\Gamma \vdash M = M : T} \qquad (\text{C-SYM})\,\frac{\Gamma \vdash N = M : T}{\Gamma \vdash M = N : T} \qquad (\text{C-TRANS})\,\frac{\Gamma \vdash M = P : T \quad \Gamma \vdash P = N : T}{\Gamma \vdash M = N : T}$$

Fig. 8. Definitional equality & computational laws

5 TRANSLATION FROM Rω

5.1 Untyped Translation

We follow the approach of [Morris and McKinna 2019] and give both typed and untyped translations of Rω types. Figure 9 describe the untyped translation, which is used to show translational soundness of the typed translation.

$$\boxed{(\kappa)^\bullet}$$

$$\begin{aligned}
(\star)^\bullet &= \star \\
(\mathsf{L})^\bullet &= \top \\
(\kappa_1 \longrightarrow \kappa_2)^\bullet &= \Pi(\alpha : (\kappa_1)^\bullet).(\kappa_2)^\bullet \\
(\mathsf{R}^\kappa)^\bullet &= \Sigma(n : \mathrm{Nat}).\Pi(j : \mathrm{Ix}\ n).(\kappa)^\bullet
\end{aligned}$$

$$\boxed{(\sigma)^\bullet}$$

$$\begin{aligned}
(\alpha)^\bullet &= \alpha \\
(\tau_1 \longrightarrow \tau_2)^\bullet &= \Pi(\alpha : (\tau_1)^\bullet).(\tau_2)^\bullet \\
(\forall \alpha : \kappa.\tau)^\bullet &= \Pi(\alpha : (\kappa)^\bullet).(\tau)^\bullet \\
(\lambda \alpha : \kappa.\tau)^\bullet &= \Pi(\alpha : (\kappa)^\bullet).(\tau)^\bullet
\end{aligned}$$

$$\boxed{(M)^\bullet}$$

$$\ldots$$

$$\boxed{(\pi)^\bullet}$$

$$\ldots$$

Fig. 9. A compositional translation of Rω judgments to (untyped) Ix terms

5.2 Typed translation

$$\boxed{\Gamma \vdash \tau \rightsquigarrow \upsilon : \kappa}$$

$$(\textsc{c-foo})\ \frac{A}{B}$$

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

$$(\textsc{c-foo})\ \frac{A}{B}$$

$$\boxed{\Gamma \Vdash \pi \rightsquigarrow N}$$

$$(\textsc{c-foo})\ \frac{A}{B}$$

$$\boxed{\tau \equiv \upsilon \rightsquigarrow P}$$

$$(\textsc{c-foo})\ \frac{A}{B}$$

Fig. 10. Translation of Rω derivations to Ix derivations

5.3 Properties of Translation

Theorem 1 (Translational Soundness (Types)). *if* $\Gamma \vdash \tau : \kappa$ *such that* $\Gamma \vdash \tau \rightsquigarrow \upsilon : \kappa$ *then* $(\Gamma)^\bullet \vdash \upsilon : (\kappa)^\bullet$.

The following is bullshit w.r.t. definitional equality.

Theorem 2 (Translational Soundness (Type Equivalence)). *if* $\Gamma \vdash \tau_1 \rightsquigarrow \upsilon_1 : \kappa_1$ *and* $\Gamma \vdash \tau_2 \rightsquigarrow \upsilon_2 : \kappa_2$ *such that* $\tau_1 \equiv \tau_2$ *is derivable in Rω, then* $(\Gamma)^\bullet \vdash \upsilon_1 \equiv \upsilon_2$.

The next theorems presume an Rω instantiation of the simple row theory.

Theorem 3 (Translational Soundness (Row combination)). *if* $\Gamma \Vdash \rho_1 \cdot \rho_2 \sim \rho_3 \rightsquigarrow N$ *then* $(\Gamma)^\bullet \vdash N : \mathit{foobar}$.

Theorem 4 (Translational Soundness (Row containment)). *if* $\Gamma \Vdash \rho_1 \lesssim \rho_2 \rightsquigarrow N$ *then* $(\Gamma)^\bullet \vdash N : \mathit{foobar}$.

Finally,

Theorem 5 (Translational Soundness). *if* $\Gamma \vdash M : \tau$ *such that* $\Gamma \vdash M \rightsquigarrow N{:}\tau$ *then* $(\Gamma)^\bullet \vdash M : (\tau)^\bullet$.

6 OPERATIONAL SEMANTICS OF IX

7 RECURSION

This section will later be incorporated into earlier sections.

7.1 Rome, or, Rω with μ

$\boxed{\text{todo}}$

$$(\textsc{c-foo})\ \frac{A}{B}$$

Fig. 11. Additional Rω judgments for recursion

7.2 Mix, the recursive index calculus

$\boxed{\text{todo}}$

$$(\textsc{c-foo})\ \frac{A}{B}$$

Fig. 12. Additional Ix judgments for recursion

7.3 Translation and properties of translation

REFERENCES

Alex Hubers and J. Garrett Morris. 2023. Generic Programming with Extensible Data Types; Or, Making Ad Hoc Extensible Data Types Less Ad Hoc. *CoRR* abs/2307.08759 (2023). https://doi.org/10.48550/arXiv.2307.08759 arXiv:2307.08759

J. Garrett Morris and James McKinna. 2019. Abstracting extensible data types: or, rows by any other name. *Proc. ACM Program. Lang.* 3, POPL (2019), 12:1–12:28. https://doi.org/10.1145/3290325