

# Recursive Rows in Rome

AH & JGM

## ACM Reference Format:

AH & JGM. 2023. Recursive Rows in Rome. 1, 1 (December 2023), 8 pages. <https://doi.org/10.1145/nnnnnnnn>.  
nnnnnnnn

## 1 IX: THE INDEX CALCULUS

### 1.1 Syntax

Sorts	$\sigma ::= \star \mid \square$
Terms	$M, N, T ::= \star \mid x \mid$ $\mathbb{N} \mid Z \mid S M \mid$ $\text{case}_{\mathbb{N}} M N T \mid$ $\text{I}_x M \mid \text{I}_0 \mid \text{I}_S M \mid$ $\text{case}_{\text{I}_x} M N T \mid \{M_1, \dots, M_n\} \mid \lambda() \mid$ $\tau \mid \text{tt} \mid$ $\forall \alpha : T.N \mid \lambda x : T.N \mid M N \mid$ $\exists \alpha : T.M \mid \langle \alpha : T, M \rangle \mid \text{case}_{\exists} M N \mid$ $M + N \mid \text{left } M \mid \text{right } M \mid$ $\text{case}_{+} M N T \mid$ $M \equiv N \mid \text{refl } T M N \mid$
Environments	$\Gamma ::= \varepsilon \mid \Gamma, \alpha : T$

Fig. 1. Syntax

#### 1.1.1 Meta-syntax & syntactic sugar. Let

- (1)  $\tau \rightarrow v$  denote the non-dependent universal quantification  $\forall(\_ : \tau).v$ ;
- (2)  $\tau \times v$  denote the non-dependent existential quantification  $\exists(\_ : \tau).v$ ;
- (3)  $0, 1, 2, \dots$  denote object-level natural numbers in the intuitive fashion;
- (4)  $i_n$  denote the index obtained by  $n$  applications of  $\text{I}_S$  to  $\text{I}_0$ ; and

---

Author's address: AH & JGM.

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2023 Association for Computing Machinery.

XXXX-XXXX/2023/12-ART \$15.00

<https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

(5) the syntax

$$\{\!\{M_1, \dots, M_n\}\!\}$$

denote the large elimination of a known, finite quantity of indices to types  $M_1, \dots, M_n$ , elaborated by the equations:

$$\begin{aligned}\{\!\{M_1\}\!\} &:= \lambda(i : \text{Ix } 1).\text{case}_{\text{Ix}} i M_1 \lambda() \\ \{\!\{M_1, \dots, M_n\}\!\} &:= \lambda(i : \text{Ix } n).\text{case}_{\text{Ix}} i M_1 \{\!\{M_2, \dots, M_n\}\!\}\end{aligned}$$

## 1.2 Typing

$$\begin{array}{c} \boxed{\vdash \Gamma} \\ \text{(EMP)} \frac{}{\vdash \varepsilon} \quad \text{(VAR)} \frac{\vdash \Gamma \quad \Gamma \vdash M : \sigma}{\vdash \Gamma, x : M} \\ \boxed{\Gamma \vdash M : \sigma} \\ \begin{array}{llll} (\star) \frac{\vdash \Gamma}{\Gamma \vdash \star : \square} & (\top) \frac{\vdash \Gamma}{\Gamma \vdash \top : \sigma} & (\text{NAT}) \frac{\vdash \Gamma}{\Gamma \vdash \mathbb{N} : \star} & (\text{Ix}) \frac{\Gamma \vdash n : \mathbb{N}}{\Gamma \vdash \text{Ix } n : \star} \\ (\forall) \frac{\Gamma \vdash M : \sigma_1 \quad \Gamma, \alpha : M \vdash N : \sigma_2}{\Gamma \vdash \forall \alpha : M.N : \sigma_2} & (\exists) \frac{\Gamma \vdash M : \sigma_1 \quad \Gamma, \alpha : M \vdash N : \sigma_2}{\Gamma \vdash \exists \alpha : M.N : \sigma_2} & & \\ (+) \frac{\Gamma \vdash M : \sigma \quad \Gamma \vdash N : \sigma}{\Gamma \vdash M + N : \sigma} & (\equiv) \frac{\Gamma \vdash M : \sigma \quad \Gamma \vdash N : \sigma}{\Gamma \vdash M \equiv N : \sigma} & & \end{array} \end{array}$$

Fig. 2. Context and type well-formedness

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

$$\begin{array}{c}
(\text{VAR}) \frac{\vdash \Gamma \quad x : M \in \Gamma}{\Gamma \vdash x : M} \quad (\text{tt}) \frac{\vdash \Gamma}{\Gamma \vdash \text{tt} : \top} \\
(Z) \frac{\vdash \Gamma}{\Gamma \vdash Z : \mathbb{N}} \quad (S) \frac{\Gamma \vdash n : \mathbb{N}}{\Gamma \vdash S n : \mathbb{N}} \quad (\mathbb{N}E) \frac{\Gamma \vdash A : \sigma \quad \Gamma \vdash M : \mathbb{N} \quad \Gamma \vdash N : A \quad \Gamma \vdash P : \mathbb{N} \rightarrow A}{\Gamma \vdash \text{case}_{\mathbb{N}} M N P : A} \\
(l_0) \frac{\Gamma \vdash n : \mathbb{N}}{\Gamma \vdash l_0 : \text{Ix}(S n)} \quad (l_S) \frac{\Gamma \vdash n : \mathbb{N} \quad \Gamma \vdash i : \text{Ix } n}{\Gamma \vdash l_S i : \text{Ix}(S n)} \quad (\lambda()) \frac{\Gamma \vdash A : \sigma}{\Gamma \vdash \lambda() : \text{Ix } 0 \rightarrow A} \\
(l_S E) \frac{\Gamma \vdash A : \sigma \quad \Gamma \vdash M : \text{Ix}(S n) \quad \Gamma \vdash N : A \quad \Gamma \vdash P : \text{Ix } n \rightarrow A}{\Gamma \vdash \text{case}_{\text{Ix}} M N P : A} \\
(\forall I) \frac{\Gamma \vdash T : \sigma \quad \Gamma, x : T \vdash M : N}{\Gamma \vdash \lambda x : T. M : \forall(x : T). N} \quad (\forall E) \frac{\Gamma \vdash M : \forall(x : T_1). T_2 \quad \Gamma \vdash N : T_1}{\Gamma \vdash M N : T_2[N/x]} \\
(\exists I) \frac{\Gamma \vdash T_1 : \sigma \quad \Gamma \vdash M : T_1 \quad \Gamma \vdash N : T_2[M/x]}{\Gamma \vdash \langle\langle M : T_1, N \rangle\rangle : \exists(x : T_1). T_2} \\
(\exists E) \frac{\Gamma \vdash A : \sigma \quad \Gamma \vdash M : \exists(x : T_1). T_2 \quad \Gamma \vdash N : \forall(x : T_1). T_2 \rightarrow A}{\Gamma \vdash \text{case}_{\exists} M N : A} \\
(+_1 I) \frac{\Gamma \vdash M : A}{\Gamma \vdash \text{left } M : A + B} \quad (+_2 I) \frac{\Gamma \vdash N : B}{\Gamma \vdash \text{right } N : A + B} \\
(+E) \frac{\Gamma \vdash C : \sigma \quad \Gamma \vdash M : A + B \quad \Gamma \vdash N : A \rightarrow C \quad \Gamma \vdash P : B \rightarrow C}{\Gamma \vdash \text{case}_+ M N P : C} \\
(\equiv I) \frac{\Gamma \vdash M : \sigma}{\Gamma \vdash \text{refl} : M \equiv M} \quad (\text{CONV}) \frac{\Gamma \vdash M : T_1 \quad \Gamma \vdash T_1 = T_2 : \sigma}{\Gamma \vdash M : T_2} \\
\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, \text{refl}/p] \\
(\equiv E) \frac{}{\Gamma \vdash \mathcal{J} H M N P : T[M/x, N/y, P/p]}
\end{array}
\end{array}$$

Fig. 3. Typing lx terms

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

$$\begin{array}{c}
(\text{E-REFL}) \frac{\Gamma \vdash M : \sigma}{\Gamma \vdash M = M : \sigma} \quad (\text{E-SYM}) \frac{\Gamma \vdash N = M : \sigma}{\Gamma \vdash M = N : \sigma} \quad (\text{E-TRANS}) \frac{\Gamma \vdash M = P : \sigma \quad \Gamma \vdash P = N : \sigma}{\Gamma \vdash M = N : \sigma} \\
\boxed{\Gamma \vdash M = N : T} \\
(\text{C-REFL}) \frac{\Gamma \vdash M : T}{\Gamma \vdash M = M : T} \quad (\text{C-SYM}) \frac{\Gamma \vdash N = M : T}{\Gamma \vdash M = N : T} \quad (\text{C-TRANS}) \frac{\Gamma \vdash M = P : T \quad \Gamma \vdash P = N : T}{\Gamma \vdash M = N : T}
\end{array}$$

Fig. 4. Definitional equality &amp; computational laws

### 1.3 Properties

**THEOREM 1 (WELL-SORTEDNESS).** *if  $\Gamma \vdash M : N$  then  $\vdash \Gamma$  and there exists  $\sigma$  such that  $\Gamma \vdash N : \sigma$ .*

### 1.4 Elaborating Ix to the CoC + Fin

Following the above, I believe Ix to a smaller calculus with the syntax below. This is effectively the calculus of constructions with primitive naturals and finite indices.

Sorts	$\sigma ::= \star \mid \square$
Terms	$M, N, T ::= \star \mid x \mid$ $\mathbb{N} \mid Z \mid S M \mid$ $\text{case}_{\mathbb{N}} M N T \mid$ $\text{Ix } M \mid l_0 \mid l_S M \mid$ $\text{case}_{\text{Ix}} M N T \mid$ $\forall \alpha : T. N \mid \lambda x : T. N \mid M N \mid$ $M \equiv N \mid \text{refl } T M N \mid$
Environments	$\Gamma ::= \varepsilon \mid \Gamma, \alpha : T$

Fig. 5. Syntax

Elaboration is given below. One would also expect a rule for bot elimination—or you could simply encode Ix 0 as  $\forall X : \star. X$ .

$$\begin{aligned}
\top &\rightsquigarrow \text{Ix } 1 \\
\text{tt} &\rightsquigarrow l_0 \\
\perp &\rightsquigarrow \text{Ix } 0 \\
A \rightarrow B &\rightsquigarrow \forall (x : A). B \\
\exists (x : A). B &\rightsquigarrow \forall (C : \square). (\forall (x : A). B \rightarrow C) \rightarrow C \\
\langle\langle M : T, N \rangle\rangle &\rightsquigarrow \lambda (C : \square). \lambda (f : (\forall (x : A). B \rightarrow C)). f M N \\
A + B &\rightsquigarrow \exists (i : \text{Ix } 2). \langle\langle A, B \rangle\rangle i \\
\text{left } M &\langle\langle i_0 : \text{Ix } 2, M \rangle\rangle \\
\text{right } N &\langle\langle i_1 : \text{Ix } 2, N \rangle\rangle \\
A \times B &\rightsquigarrow \exists (x : A). B
\end{aligned}$$

One could also translate naturals away using your favorite functional encoding. (I imagine there are encodings for Fin, too.)

## 2 TRANSLATION FROM $R\omega$

### 2.1 Untyped Translation

We follow the approach of [Morris and McKinna 2019] and give both typed and untyped translations of  $R\omega$  types. Figure 7 describe the untyped translation, which is used to show translational soundness of the typed translation (Figure 7).

$$\boxed{\llbracket \kappa \rrbracket}$$

$$\llbracket \star \rrbracket = \star$$

$$\llbracket \mathbf{L} \rrbracket = \top$$

$$\llbracket \kappa_1 \rightarrow \kappa_2 \rrbracket = \llbracket \kappa_1 \rrbracket \rightarrow \llbracket \kappa_2 \rrbracket$$

$$\llbracket \mathbf{R}^\kappa \rrbracket = \exists (n : \mathbb{N}). \text{Ix } n \rightarrow \llbracket \kappa \rrbracket$$

$$\boxed{\llbracket \Delta \vdash \tau : \kappa \rrbracket}$$

$$\llbracket \Delta \vdash \alpha : \kappa \rrbracket = \alpha$$

$$\llbracket \Delta \vdash \tau_1 \rightarrow \tau_2 : \star \rrbracket = \llbracket \tau_1 \rrbracket \rightarrow \llbracket \tau_2 \rrbracket$$

$$\llbracket \Delta \vdash \forall \alpha : \kappa. \tau : \star \rrbracket = \forall (\alpha : \llbracket \kappa \rrbracket). \llbracket \tau \rrbracket$$

$$\llbracket \Delta \vdash \lambda \alpha : \kappa. \tau : \kappa \rightarrow \kappa' \rrbracket = \forall (\alpha : \llbracket \kappa \rrbracket). \llbracket \tau \rrbracket$$

$$\llbracket \Delta \vdash \pi \Rightarrow \tau : \kappa \rrbracket = \llbracket \pi \rrbracket \rightarrow \llbracket \tau \rrbracket$$

$$\llbracket \Delta \vdash \tau v : \kappa \rrbracket = \llbracket \tau \rrbracket \llbracket v \rrbracket$$

$$\llbracket \Delta \vdash \ell : \mathbf{L} \rrbracket = \top$$

$$\llbracket \Delta \vdash \lfloor \xi \rrbracket : \star \rrbracket = \top$$

$$\llbracket \Delta \vdash (\xi \triangleright \tau) : \kappa \rrbracket = \llbracket \tau \rrbracket$$

$$\llbracket \Delta \vdash \Pi \rho : \star \rrbracket = \text{case}_{\exists} \llbracket \rho \rrbracket (\lambda n : \mathbb{N}. \lambda P : \text{Ix } n \rightarrow \star. \forall (i : \text{Ix } n). P i)$$

$$\llbracket \Delta \vdash \Sigma \rho : \star \rrbracket = \text{case}_{\exists} \llbracket \rho \rrbracket (\lambda n : \mathbb{N}. \lambda P : \text{Ix } n \rightarrow \star. \exists (i : \text{Ix } n). P i)$$

$$\llbracket \Delta \vdash \epsilon : \mathbf{R}^\kappa \rrbracket = \langle \langle 0 : \mathbb{N}, \lambda () \rangle \rangle$$

$$\llbracket \Delta \vdash \rho \llbracket v \rrbracket : \mathbf{R}^{\kappa_2} \rrbracket = \text{case}_{\exists} \llbracket \rho \rrbracket (\lambda n : \mathbb{N}. \lambda (P : \text{Ix } n \rightarrow \llbracket \kappa_1 \rrbracket \rightarrow \llbracket \kappa_2 \rrbracket). \langle \langle n : \mathbb{N}, \lambda (j : \text{Ix } n). (P j) \llbracket \tau \rrbracket \rangle \rangle)$$

$$\llbracket \Delta \vdash \llbracket \tau \rrbracket \rho : \mathbf{R}^{\kappa_2} \rrbracket = \text{case}_{\exists} \llbracket \rho \rrbracket (\lambda n : \mathbb{N}. \lambda (P : \text{Ix } n \rightarrow \llbracket \kappa_1 \rrbracket). \langle \langle n : \mathbb{N}, \lambda (j : \text{Ix } n). \llbracket \tau \rrbracket (P j) \rangle \rangle)$$

$$\llbracket \Delta \vdash (\xi \triangleright_{\mathbf{R}} \tau) : \mathbf{R}^\kappa \rrbracket = \langle \langle 1 : \mathbb{N}, \llbracket \tau \rrbracket \rangle \rangle$$

Fig. 6. Translating kinding derivations to untyped  $\text{Ix}$  terms

$$\begin{aligned}
& \boxed{\llbracket \Delta \vdash \pi : \kappa \rrbracket} \\
& \text{case } \llbracket \rho_1 \rrbracket (\lambda n : \mathbb{N}. \lambda (P : \text{Ix } n \rightarrow \llbracket \kappa \rrbracket)). \\
\llbracket \Delta \vdash \rho_1 \lesssim \rho_2 : \kappa \rrbracket &= \text{case } \llbracket \rho_2 \rrbracket (\lambda m : \mathbb{N}. \lambda (Q : \text{Ix } m \rightarrow \llbracket \kappa \rrbracket)). \\
& \forall (i : \text{Ix } n). \exists (j : \text{Ix } m). P \, i \equiv Q \, j)) \\
& \text{case } \llbracket \rho_1 \rrbracket (\lambda n : \mathbb{N}. \lambda (P : \text{Ix } n \rightarrow \llbracket \kappa \rrbracket)). \\
& \text{case } \llbracket \rho_2 \rrbracket (\lambda m : \mathbb{N}. \lambda (Q : \text{Ix } m \rightarrow \llbracket \kappa \rrbracket)). \\
& \text{case } \llbracket \rho_3 \rrbracket (\lambda l : \mathbb{N}. \lambda (R : \text{Ix } l \rightarrow \llbracket \kappa \rrbracket)). \\
& (\forall (i : \text{Ix } n). \exists (k : \text{Ix } l). P \, i \equiv R \, k) \\
\llbracket \Delta \vdash \rho_1 \odot \rho_2 \sim \rho_3 : \kappa \rrbracket &= \times (\forall (j : \text{Ix } m). \exists (k : \text{Ix } l). Q \, j \equiv R \, k) \\
& \times (\forall (k : \text{Ix } l). \\
& (\exists (i : \text{Ix } n). P \, i \equiv R \, k) \\
& + (\exists (j : \text{Ix } m). Q \, j \equiv R \, k))))
\end{aligned}$$

Fig. 7. Translating predicate well-formedness judgments

## 2.2 Typed translation

There is some subtlety in mechanizing environments. Environments in  $R\omega$  store kinds, *typing derivations*, and *predicate well-formedness derivations*. If we are to simply translate derivations to untyped syntax, we are losing a bit of information. I am not sure, however, it is possible to translate derivations (in  $R\omega$ ) to derivations (in  $\text{Ix}$ ) without a de facto type checker for  $\text{Ix}$ . I think we will have to perform the former: let derivations in  $R\omega$  environments translate to untyped types and sorts in  $\text{Ix}$  environments. Then, argue as metatheory that  $\vdash \Delta \rightsquigarrow \Gamma$  implies  $\vdash \Gamma$ .

$$\boxed{\vdash \Delta \rightsquigarrow \Gamma}$$

$$\begin{array}{c}
\text{(C-}\epsilon\text{)} \frac{}{\vdash \epsilon \rightsquigarrow \epsilon} \quad \text{(C-TVAR)} \frac{\vdash \Delta \rightsquigarrow \Gamma}{\vdash \Delta, \alpha : \kappa \rightsquigarrow \Gamma, \alpha : \llbracket \kappa \rrbracket} \\
\text{(C-VAR)} \frac{\vdash \Delta \rightsquigarrow \Gamma}{\vdash \Delta, x : \tau \rightsquigarrow \Gamma, x : \llbracket \tau \rrbracket} \quad \text{(C-PRED)} \frac{\vdash \Delta \rightsquigarrow \Gamma}{\vdash \Delta, \pi : \kappa \rightsquigarrow \Gamma, p : \llbracket \pi \rrbracket} \text{ (} p \text{ fresh)}
\end{array}$$

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

$$\begin{array}{c}
\text{(T-VAR)} \frac{x : \tau \in \Delta}{\Delta \vdash x \rightsquigarrow x : \tau} \\
\text{(T-}\rightarrow\text{I)} \frac{\Delta, x : \tau \vdash M \rightsquigarrow N : v}{\Delta \vdash \lambda x : \tau. M \rightsquigarrow \lambda x : \llbracket \tau \rrbracket. N : \tau \rightarrow v} \quad \text{(T-}\rightarrow\text{I)} \frac{\Delta \vdash M \rightsquigarrow F : \tau \rightarrow v \quad \Delta \vdash N \rightsquigarrow E : \tau}{\Delta \vdash M N \rightsquigarrow F E : v} \\
\text{(T-}\Rightarrow\text{I)} \frac{\Delta, \pi \vdash M \rightsquigarrow N : \tau}{\Delta \vdash M \rightsquigarrow \lambda(p : \llbracket \pi \rrbracket). N : \pi \Rightarrow \tau} \quad \text{(T-}\Rightarrow\text{E)} \frac{\Delta \vdash M \rightsquigarrow F : \pi \Rightarrow \tau \quad \Delta \Vdash \pi \rightsquigarrow E}{\Delta \vdash M \rightsquigarrow F E : \tau} \\
\text{(T-}\forall\text{I)} \frac{\Delta \vdash M \rightsquigarrow N : \tau}{\Delta \vdash \Lambda \alpha : \kappa. M \rightsquigarrow \lambda(\alpha : \llbracket \kappa \rrbracket). N : \forall \alpha : \kappa. \tau} \quad \text{(T-}\forall\text{E)} \frac{\Delta \vdash M \rightsquigarrow N : \forall \alpha : \kappa. \tau}{\Delta \vdash M[v] \rightsquigarrow N \llbracket v \rrbracket : \tau[v/\alpha]} \\
\text{(T-SING)} \frac{}{\Delta \vdash \ell \rightsquigarrow \text{tt} : \llbracket \ell \rrbracket} \quad \text{(T-}\triangleright\text{I)} \frac{\Delta \vdash N \rightsquigarrow E : \tau}{\Delta \vdash M \triangleright N \rightsquigarrow E : \ell \triangleright \tau} \quad \text{(T-}\triangleright\text{E)} \frac{\Delta \vdash M \rightsquigarrow E : \tau \quad \Delta \vdash N \rightsquigarrow \text{tt} : \llbracket \ell \rrbracket}{\Delta \vdash M/N \rightsquigarrow E : \ell \triangleright \tau} \\
\text{(C-FOO)} \frac{A}{B} \quad \text{(C-FOO)} \frac{A}{B} \quad \text{(C-FOO)} \frac{A}{B}
\end{array}$$

Fig. 8. Translation of  $R\omega$  environments and typing derivations

$$\begin{array}{c}
\text{(C-FOO)} \frac{A}{B} \\
\boxed{\Delta \Vdash \pi \rightsquigarrow N} \\
\text{(C-FOO)} \frac{A}{B} \\
\boxed{\tau \equiv v \rightsquigarrow P} \\
\text{(C-FOO)} \frac{A}{B}
\end{array}$$

Fig. 9. Translation of  $R\omega$  derivations to  $Ix$  derivations

### 2.3 Properties of Translation

**THEOREM 2 (TRANSLATIONAL SOUNDNESS (ENVIRONMENTS)).** *if  $\vdash \Delta \rightsquigarrow \Gamma$  then  $\vdash \Gamma$ .*

**THEOREM 3 (TRANSLATIONAL SOUNDNESS (TYPES)).** *if  $\Delta \vdash \tau : \kappa$  and  $\vdash \Delta \rightsquigarrow \Gamma$  then  $\Gamma \vdash \llbracket \tau \rrbracket : \llbracket \kappa \rrbracket$ .*

THEOREM 4 (TRANSLATIONAL SOUNDNESS (TYPE EQUIVALENCE)). *if*

- (1)  $\Gamma \vdash \tau_1 \rightsquigarrow v_1 : \kappa_1$ ;
- (2)  $\Gamma \vdash \tau_2 \rightsquigarrow v_2 : \kappa_2$ ; *and*
- (3)  $\tau_1 \equiv \tau_2 \rightsquigarrow P$ ,

*then*  $\llbracket \Gamma \rrbracket \vdash P : v_1 \equiv v_2$ .

THEOREM 5 (TRANSLATIONAL SOUNDNESS (PREDICATES)). *if*  $\Gamma \Vdash \pi$  *such that*  $\Gamma \Vdash \pi \rightsquigarrow N$  *then*  $\llbracket \Gamma \rrbracket \vdash N : \llbracket \pi \rrbracket$ .

Finally,

THEOREM 6 (TRANSLATIONAL SOUNDNESS). *if*  $\Gamma \vdash M : \tau$  *such that*  $\Gamma \vdash M \rightsquigarrow N : \tau$  *then*  $\llbracket \Gamma \rrbracket \vdash N : \llbracket \tau \rrbracket$ .

### 3 OPERATIONAL SEMANTICS

#### REFERENCES

- 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>