

Flow Typing: A New Lens on Linearity (Appendix)

SKY WILSHAW, University of Nottingham, United Kingdom

GRAHAM HUTTON, University of Nottingham, United Kingdom

This document provides a full list of typing rules for the paper *Flow Typing: A New Lens on Linearity*, as well as proofs for results not provided in the main text.

A Typing Rules

This section contains the full listing of typing rules for λ_F .

$$\begin{array}{c}
 \frac{\Gamma_1, x : A, y : B, \Gamma_2 \vdash t : C \vdash \Gamma_3}{\Gamma_1, y : B, x : A, \Gamma_2 \vdash t : C \vdash \Gamma_3} \text{EXCHL} \qquad \frac{\Gamma_1 \vdash t : C \vdash \Gamma_2, x : A, y : B, \Gamma_3}{\Gamma_1 \vdash t : C \vdash \Gamma_2, y : B, x : A, \Gamma_3} \text{EXCHR} \\
 \\
 \frac{}{\Gamma, x : A \vdash x : A \vdash \Gamma, x^\dagger} \text{VAR} \qquad \frac{}{\Gamma, x : A \vdash x : A/A' \vdash \Gamma, x : A'} \text{FVAR-LIVE} \\
 \\
 \frac{}{\Gamma, x^\dagger \vdash x : 1/A' \vdash \Gamma, x : A'} \text{FVAR-DEAD} \qquad \frac{\Gamma_1 \vdash t : B \vdash \Gamma_2 \quad \Gamma_2 \vdash u : A/B \vdash \Gamma_3}{\Gamma_1 \vdash (u := t) : A \vdash \Gamma_3} \text{ASSIGN} \\
 \\
 \frac{\Gamma_1 \vdash t_1 : A \vdash \Gamma_2 \quad \Gamma_2, x : A' \vdash t_2 : 1 \vdash \Gamma_3, x^\dagger}{\Gamma_1 \vdash (\text{lens } x \leftarrow t_1 \text{ in } t_2) : A/A' \vdash \Gamma_3} \text{LENS} \\
 \\
 \frac{\Gamma_2 \vdash u : A/B \vdash \Gamma_3 \quad \Gamma_3 \vdash t : C \vdash \Gamma_3^\dagger}{\Gamma_1, \Gamma_2 \vdash (\text{unlens } u \text{ in } t) : A \otimes (B \multimap C) \vdash \Gamma_1, \Gamma_2^\dagger} \text{UNLENS} \\
 \\
 \frac{\Gamma_1 \vdash t : A \vdash \Gamma_2 \quad \Gamma_2, x : A \vdash s : T \vdash \Gamma_3, x^\dagger}{\Gamma_1 \vdash (\text{let } x = t \text{ in } s) : T \vdash \Gamma_3} \text{LET} \\
 \\
 \frac{\Gamma_1 \vdash u : A/A' \vdash \Gamma_2 \quad \Gamma_3, x : A \vdash s : T \vdash \Gamma_4, x : A'}{\Gamma_1, \Gamma_3 \vdash (\text{let } x = \sim u \text{ in } s) : T \vdash \Gamma_2, \Gamma_4} \sim\text{LET} \\
 \\
 \frac{\Gamma_2, x : A \vdash s : T \vdash \Gamma_2^\dagger, x^\dagger}{\Gamma_1, \Gamma_2 \vdash (\lambda x : A. s) : (A \multimap T) \vdash \Gamma_1, \Gamma_2^\dagger} \multimap\text{-I} \qquad \frac{\Gamma_2, x : A \vdash s : T \vdash \Gamma_2^\dagger, x : A'}{\Gamma_1, \Gamma_2 \vdash (\lambda x : A/A'. s) : (A/A' \multimap T) \vdash \Gamma_1, \Gamma_2^\dagger} \sim\multimap\text{-I} \\
 \\
 \frac{\Gamma_1 \vdash t : (T \multimap T') \vdash \Gamma_2 \quad \Gamma_2 \vdash s : T \vdash \Gamma_3}{\Gamma_1 \vdash t s : T' \vdash \Gamma_3} \multimap\text{-E} \qquad \frac{\Gamma_1 \vdash t_1 : A \vdash \Gamma_2 \quad \Gamma_2 \vdash t_2 : B \vdash \Gamma_3}{\Gamma_1 \vdash (t_1, t_2) : A \otimes B \vdash \Gamma_3} \otimes\text{-I} \\
 \\
 \frac{\Gamma_1 \vdash u_1 : A/A' \vdash \Gamma_2 \quad \Gamma_3 \vdash u_2 : B/B' \vdash \Gamma_4}{\Gamma_1, \Gamma_3 \vdash (u_1, u_2) : (A \otimes B)/(A' \otimes B') \vdash \Gamma_2, \Gamma_4} \sim\otimes\text{-I}
 \end{array}$$

$$\begin{array}{c}
\frac{\Gamma_1 \vdash t : A \otimes B \vdash \Gamma_2 \quad \Gamma_2, x : A, y : B \vdash s : T \vdash \Gamma_3, x^\dagger, y^\dagger}{\Gamma_1 \vdash (\text{let } (x, y) = t \text{ in } s) : T \vdash \Gamma_3} \otimes\text{-E} \\
\\
\frac{\Gamma_1 \vdash u : (A \otimes B)/(A' \otimes B') \vdash \Gamma_2 \quad \Gamma_3, x : A, y : B \vdash s : T \vdash \Gamma_4, x : A', y : B'}{\Gamma_1, \Gamma_3 \vdash (\text{let } (x, y) = \sim u \text{ in } s) : T \vdash \Gamma_2, \Gamma_4} \sim\otimes\text{-E} \\
\\
\frac{}{\Gamma \vdash * : \mathbf{1} \vdash \Gamma} \mathbf{1}\text{-I} \quad \frac{}{\Gamma \vdash * : \mathbf{1}/\mathbf{1} \vdash \Gamma} \sim\mathbf{1}\text{-I} \quad \frac{\Gamma_1 \vdash t : \mathbf{1} \vdash \Gamma_2 \quad \Gamma_2 \vdash s : T \vdash \Gamma_3}{\Gamma_1 \vdash (\text{let } * = t \text{ in } s) : T \vdash \Gamma_3} \mathbf{1}\text{-E} \\
\\
\frac{\Gamma_1 \vdash u : \mathbf{1}/\mathbf{1} \vdash \Gamma_2 \quad \Gamma_3 \vdash s : T \vdash \Gamma_4}{\Gamma_1, \Gamma_3 \vdash (\text{let } * = \sim u \text{ in } s) : T \vdash \Gamma_2, \Gamma_4} \sim\mathbf{1}\text{-E} \quad \frac{\Gamma_1 \vdash t : A \vdash \Gamma_2}{\Gamma_1 \vdash \text{inl}_{A \oplus B}(t) : A \oplus B \vdash \Gamma_2} \oplus\text{-IL} \\
\\
\frac{\Gamma_1 \vdash t : B \vdash \Gamma_2}{\Gamma_1 \vdash \text{inr}_{A \oplus B}(t) : A \oplus B \vdash \Gamma_2} \oplus\text{-IR} \\
\\
\frac{\Gamma_1 \vdash t : A \oplus B \vdash \Gamma_2 \quad \Gamma_2, x : A \vdash s_1 : T \vdash \Gamma_3, x^\dagger \quad \Gamma_2, y : B \vdash s_2 : T \vdash \Gamma_3, y^\dagger}{\Gamma_1 \vdash (\text{case } t \text{ of } \text{inl}(x) \rightarrow s_1 \mid \text{inr}(y) \rightarrow s_2) : T \vdash \Gamma_3} \oplus\text{-E} \\
\\
\frac{\Gamma_1 \vdash u : (A \oplus B)/(A' \oplus B') \vdash \Gamma_2 \quad \Gamma_3, x : A \vdash s_1 : T \vdash \Gamma_4, x : A' \quad \Gamma_3, y : B \vdash s_2 : T \vdash \Gamma_4, y : B'}{\Gamma_1, \Gamma_3 \vdash (\text{case } \sim u \text{ of } \text{inl}(x) \rightarrow s_1 \mid \text{inr}(y) \rightarrow s_2) : T \vdash \Gamma_2, \Gamma_4} \sim\oplus\text{-E} \\
\\
\frac{\Gamma_1 \vdash t : \mathbf{0} \vdash \Gamma_2}{\Gamma_1 \vdash \text{false}_T(t) : T \vdash \Gamma_2} \mathbf{0}\text{-E} \quad \frac{\Gamma_1 \vdash u : \mathbf{0}/\mathbf{0} \vdash \Gamma_2}{\Gamma_1 \vdash \text{false}_T(\sim u) : T \vdash \Gamma_2} \sim\mathbf{0}\text{-E} \\
\\
\frac{\Gamma_2 \vdash t_1 : A \vdash \Gamma_2^\dagger \quad \Gamma_2 \vdash t_2 : B \vdash \Gamma_2^\dagger}{\Gamma_1, \Gamma_2 \vdash (t_1 \& t_2) : A \& B \vdash \Gamma_1, \Gamma_2^\dagger} \&\text{-I} \quad \frac{\Gamma_1 \vdash t : A \& B \vdash \Gamma_2}{\Gamma_1 \vdash \text{fst}(t) : A \vdash \Gamma_2} \&\text{-EL} \\
\\
\frac{\Gamma_1 \vdash t : A \& B \vdash \Gamma_2}{\Gamma_1 \vdash \text{snd}(t) : B \vdash \Gamma_2} \&\text{-ER} \quad \frac{\Gamma_1 \vdash t : A \vdash \Gamma_2}{\Gamma_1 \vdash \text{true}(t) : \top \vdash \Gamma_2} \top\text{-I} \\
\\
\frac{\Gamma_1 \vdash t_1 : !A_1 \vdash \Gamma_2 \quad \cdots \quad \Gamma_n \vdash t_n : !A_n \vdash \Gamma_{n+1} \quad x_1 : !A_1, \dots, x_n : !A_n \vdash t : B \vdash x_1^\dagger, \dots, x_n^\dagger}{\Gamma_1 \vdash \text{promote } t_1, \dots, t_n \text{ for } x_1, \dots, x_n \text{ in } t : !B \vdash \Gamma_{n+1}} \text{PROM} \\
\\
\frac{\Gamma_1 \vdash t : !A \vdash \Gamma_2}{\Gamma_1 \vdash \text{discard } t : \mathbf{1} \vdash \Gamma_2} \text{Disc} \quad \frac{\Gamma_1 \vdash t : !A \vdash \Gamma_2}{\Gamma_1 \vdash \text{copy } t : !A \otimes !A \vdash \Gamma_2} \text{COPY} \quad \frac{\Gamma_1 \vdash t : !A \vdash \Gamma_2}{\Gamma_1 \vdash \text{derelict } t : A \vdash \Gamma_2} \text{DER}
\end{array}$$

Note that $\sim\mathbf{1}\text{-E}$ could be made more general but this would break the pattern set out by the other algebraic data types.

B Typing Proofs

Throughout these appendices, ‘rule induction’ over flow judgements implicitly refers to ‘rule induction modulo FRAME’.

LEMMA B.1. *If $\Gamma_1 \vdash t : A \vdash \Gamma_2$ is a λ_L term and x appears dead in Γ_1 , then x appears dead in Γ_2 .*

PROOF. Rule induction: the linear lambda calculus cannot revive dead terms. $\square$

LEMMA B.2. *Define that a variable is used up in a typing derivation for a term if it is live on the left-hand side but dead on the right-hand side. Let $\Gamma_1 \vdash t : A \dashv \Gamma_2$ in λ_L with the flow type system. Then, x appears free in t if and only if it is used up by the derivation for t .*

PROOF. A simple induction on the typing derivation for t . In the base cases (VAR, 1-I, nullary PROM), the rule ensures that the free variables are exactly those that are used up. $\square$

COROLLARY B.3. *Let t be a λ_L term such that $\Gamma_1 \vdash t : A \dashv \Gamma_2$. Writing Γ for the sub-context of Γ_1 consisting of only the free variables of Γ_1 , we have $\Gamma \vdash t : A \dashv \Gamma^\dagger$.*

PROOF. By the previous lemma, all of the other variables do not appear free, so we can apply the frame rule to remove them from the derivation. $\square$

THEOREM B.4. *Let Γ and t be a context and term respectively in the language λ_L . Then*

$$\Gamma \vdash t : A \iff \Gamma \vdash t : A \dashv \Gamma^\dagger$$

PROOF. By rule induction, repeatedly applying the previous corollary to turn context threading into context splits. $\square$

THEOREM B.5. *A λ_L term t has at most one type and output context (up to permutation), given a fixed input context Γ_1 .*

$$(\Gamma_1 \vdash t : A \dashv \Gamma_2) \wedge (\Gamma_1 \vdash t : A' \dashv \Gamma'_2) \implies A = A' \wedge \Gamma_2 \text{ is a permutation of } \Gamma'_2$$

PROOF. By structural induction on expressions, generalising over types and contexts (which are understood to be identified up to permutation throughout this proof). Using the fact that typing is syntax-directed, most cases are trivial. The only nontrivial cases are when the context splits, which occurs only in $\multimap$ -I and $\&$ -I. Suppose that we are comparing the following two occurrences of $\multimap$ -I:

$$\frac{\Gamma_2, x : A \vdash t : B \dashv \Gamma_2^\dagger, x^\dagger}{\Gamma_1, \Gamma_2 \vdash (\lambda x : A. t) : A \multimap B \dashv \Gamma_1, \Gamma_2^\dagger} \quad \frac{\Gamma'_2, x : A \vdash t : B' \dashv \Gamma'_2^\dagger, x^\dagger}{\Gamma'_1, \Gamma'_2 \vdash (\lambda x : A. t) : A \multimap B' \dashv \Gamma'_1, \Gamma'_2^\dagger}$$

We apply the frame rule to the two assumptions to obtain:

$$\Gamma_1, \Gamma_2, x : A \vdash t : B \dashv \Gamma_1, \Gamma_2^\dagger \quad \Gamma'_1, \Gamma'_2, x : A \vdash t : B' \dashv \Gamma'_1, \Gamma'_2^\dagger$$

Since Γ_1, Γ_2 and Γ'_1, Γ'_2 are equal up to permutation, the inductive hypothesis now applies, giving $B = B'$ and $\Gamma_1, \Gamma_2^\dagger = \Gamma'_1, \Gamma'_2^\dagger$ up to permutation. The same trick can be used to prove the inductive step for $\&$ -I, thus giving the result. $\square$

C Translation Rules

In this section, we detail the translation rules from λ_F to λ_L . We will translate λ_F judgements to closed λ_L terms as follows:

$$\begin{aligned} (\Gamma_1 \vdash t : A \dashv \Gamma_2) &\mapsto (\vdash \bar{t} : (\Gamma_1^\circ, \Gamma_2^\circ) \multimap (A^\circ, \mathbf{1})) \\ (\Gamma_1 \vdash u : A/A' \dashv \Gamma_2) &\mapsto (\vdash \bar{u} : (\Gamma_1^\circ, \Gamma_2^\circ) \multimap (A^\circ, A'^\circ)) \end{aligned}$$

We will use conventional pattern-matching syntax to abbreviate terms, so for example $(\lambda * : \mathbf{1}. t)$ abbreviates $(\lambda x : \mathbf{1}. \text{let } * = x \text{ in } t)$. We use the following abbreviations for metatheoretically-defined combinators of closed λ_L terms:

$$\begin{aligned} (\circ) : (A \multimap B) \times (S \multimap A) &\rightarrow (S \multimap B) \\ g \circ f &= \lambda s : S. g (f s) \end{aligned}$$

$$\begin{aligned}
(\hat{\circ}) &: [(A, A') \leftrightarrow (B, B')] \times [(S, S') \leftrightarrow (A, A')] \rightarrow [(S, S') \leftrightarrow (B, B')] \\
g \hat{\circ} f &= \lambda s : S. \text{let } (a, r_a) = f \text{ s in let } (b, r_b) = g \text{ a in } (b, r_a \circ r_b) \\
(\gg) &: [(S, S') \leftrightarrow (A, 1)] \times [(S' \otimes A, S'') \leftrightarrow (B, B')] \rightarrow [(S, S'') \leftrightarrow (B, B')] \\
f \gg g &= \lambda s : S. \text{let } (a, r_a) = f \text{ s in } g (r_a *, a) \\
\text{id} &: S \multimap S \\
\text{id} &= \lambda s : S. s \\
\widehat{\text{id}} &: (S, S') \leftrightarrow (S, S') \\
\widehat{\text{id}} &= \lambda s : S. (s, \text{id}) \\
\widehat{\text{left}} &: (A \multimap B) \rightarrow [(A, C) \leftrightarrow (B, C)] \\
\widehat{\text{left}} &= \lambda f : (A \multimap B). \lambda a : A. (f \text{ a}, \text{id})
\end{aligned}$$

We call $\hat{\circ}$ *vertical* composition and $\gg$ *sequential* composition. Next, we define the following proof-relevant notion of context split:

Definition C.1. A *splitting* of a λ_F -context Γ is a list ψ of values $\{1, 2\}$ of the same length as Γ . We say that ψ *splits* Γ into Γ_1, Γ_2 if Γ_1 consists precisely of the entries of Γ for which ψ takes value 1, and likewise for Γ_2 . Given ψ , the values of Γ_1, Γ_2 are uniquely determined, and vice versa; we write $\psi : \Gamma \simeq (\Gamma_1 \mid \Gamma_2)$ when this is the case. Note that the same ψ can be used for contexts with different types but the same overall structure. We write $\text{split}(\psi)$ and $\text{merge}(\psi)$ for the following terms:

$$\begin{aligned}
&\vdash \text{split}(\psi) : \Gamma^\circ \multimap (\Gamma_1^\circ \otimes \Gamma_2^\circ) \\
&\text{split}(\emptyset) = \lambda *. (*, *) \\
&\text{split}((\psi, 1)) = \begin{cases} \lambda (\gamma, x). \text{let } (\gamma_1, \gamma_2) = \text{split}(\psi) \text{ } \gamma \text{ in } ((\gamma_1, x), \gamma_2) & \text{if } x \text{ live in } \Gamma \\ \text{split}(\psi) & \text{otherwise} \end{cases} \\
&\text{split}((\psi, 2)) = \begin{cases} \lambda (\gamma, x). \text{let } (\gamma_1, \gamma_2) = \text{split}(\psi) \text{ } \gamma \text{ in } (\gamma_1, (\gamma_2, x)) & \text{if } x \text{ live in } \Gamma \\ \text{split}(\psi) & \text{otherwise} \end{cases} \\
&\vdash \text{merge}(\psi) : (\Gamma_1^\circ \otimes \Gamma_2^\circ) \multimap \Gamma^\circ \\
&\text{merge}(\emptyset) = \lambda (*, *). * \\
&\text{merge}((\psi, 1)) = \begin{cases} \lambda ((\gamma_1, x), \gamma_2). (\text{merge}(\psi) (\gamma_1, \gamma_2), x) & \text{if } x \text{ live in } \Gamma \\ \text{merge}(\psi) & \text{otherwise} \end{cases} \\
&\text{merge}((\psi, 2)) = \begin{cases} \lambda (\gamma_1, (\gamma_2, x)). (\text{merge}(\psi) (\gamma_1, \gamma_2), x) & \text{if } x \text{ live in } \Gamma \\ \text{merge}(\psi) & \text{otherwise} \end{cases} \quad \diamond
\end{aligned}$$

The following set of rules is an exchange-free formulation of the flow typing rules. Note that in each of these definitions, the parameters occur exactly once, so they can be substituted strictly. Also, these rules make use of the fact that appearances of dead variables are ignored by $(-)^{\circ}$.

$$\begin{aligned}
&\frac{\psi : \Gamma_1 \simeq (\Gamma_2 \mid x : A)}{(\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow (A^\circ, 1)} \text{VAR} \\
&\quad \mapsto (\lambda (\gamma_2, (*, x)) : (\Gamma_2^\circ \otimes (x : A)^\circ). (x, \lambda * : 1. \gamma_2)) \circ \text{split}(\psi) \\
&\frac{\psi : \Gamma_1 \simeq (\Gamma_2 \mid x : A) \quad \psi' : \Gamma'_1 \simeq (\Gamma'_2 \mid x : A')}{(\Gamma_1^\circ, \Gamma_1'^{\circ}) \leftrightarrow (A^\circ, A'^{\circ})} \text{FVAR-LIVE}
\end{aligned}$$

$$\begin{aligned}
& \mapsto (\lambda(\gamma_2, (*, x)) : (\Gamma_2^\circ \otimes (x : A)^\circ). (x, \lambda x' : A'^\circ. \text{merge}(\psi) (\gamma_2, x')))) \circ \text{split}(\psi) \\
& \frac{\psi : \Gamma_2 \simeq (\Gamma_1 \mid x : A')}{(\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow (1, A'^\circ)} \text{FVAR-DEAD} \\
& \mapsto \lambda \gamma_1 : \Gamma_1^\circ. (*, (\lambda x : A'^\circ. \text{merge}(\psi) (\gamma_1, (*, x)))) \\
& \frac{f : (\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow (A^\circ, 1) \quad g : (\Gamma_2^\circ \otimes A'^\circ, \Gamma_3^\circ) \leftrightarrow (1, 1)}{(\Gamma_1^\circ, \Gamma_3^\circ) \leftrightarrow (A^\circ, A'^\circ)} \text{LENS} \\
& \mapsto \lambda \gamma_1 : \Gamma_1^\circ. \text{let } (a, r_a) = f \gamma_1 \text{ in } (a, \lambda a' : A'^\circ. \text{let } (*, r_g) = g (r_a *, a') \text{ in } r_g *) \\
& \frac{\psi : \Gamma \simeq (\Gamma_1 \mid \Gamma_2) \quad f : (\Gamma_2^\circ, \Gamma_3^\circ) \leftrightarrow (A^\circ, B^\circ) \quad g : (\Gamma_3^\circ, 1) \leftrightarrow (C, 1)}{(\Gamma^\circ, \Gamma_1^\circ) \leftrightarrow (A \otimes (B \multimap C), 1)} \text{UNLENS} \\
& \mapsto (\lambda(\gamma_1, \gamma_2) : (\Gamma_1^\circ \otimes \Gamma_2^\circ). \text{let } (a, r_a) = f \gamma_2 \text{ in} \\
& \quad ((a, \lambda b : B. \text{let } (c, r_c) = g (r_a b) \text{ in let } * = r_c * \text{ in } c), \gamma_1)) \circ \text{split}(\psi) \\
& \frac{f : (\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow (A^\circ, 1) \quad g : (\Gamma_2^\circ \otimes A^\circ, \Gamma_3^\circ) \leftrightarrow (B^\circ, B'^\circ)}{(\Gamma_1^\circ, \Gamma_3^\circ) \leftrightarrow (B^\circ, B'^\circ)} \text{LET} \\
& \mapsto f \gg g \\
& \frac{\psi : \Gamma \simeq (\Gamma_1 \mid \Gamma_3) \quad \psi : \Gamma' \simeq (\Gamma_2 \mid \Gamma_4) \quad f : (\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow (A^\circ, A'^\circ) \quad g : (\Gamma_3^\circ \otimes A^\circ, \Gamma_4^\circ \otimes A'^\circ) \leftrightarrow (B^\circ, B'^\circ)}{(\Gamma^\circ, \Gamma'^\circ) \leftrightarrow (B^\circ, B'^\circ)} \sim\text{LET} \\
& \mapsto (\lambda(\gamma_1, \gamma_3) : (\Gamma_1^\circ \otimes \Gamma_3^\circ). \\
& \quad \text{let } (a, r_a) = f \gamma_1 \text{ in let } (b, r_b) = g (\gamma_3, a) \text{ in} \\
& \quad (b, \lambda b' : B'^\circ. \text{let } (\gamma_4, a') = r_b b' \text{ in merge}(\psi) (r_a a', \gamma_4))) \\
& \quad \circ \text{split}(\psi) \\
& \frac{\psi : \Gamma \simeq (\Gamma_1 \mid \Gamma_2) \quad f : (\Gamma_2^\circ \otimes A^\circ, 1) \leftrightarrow (B^\circ, 1)}{(\Gamma^\circ, \Gamma_1^\circ) \leftrightarrow (A^\circ \multimap B^\circ, 1)} \multimap\text{-I} \\
& \mapsto (\lambda(\gamma_1, \gamma_2) : (\Gamma_1^\circ \otimes \Gamma_2^\circ). ((\lambda a : A^\circ. \text{let } (b, r_b) = f (\gamma_2, a) \text{ in let } * = r_b * \text{ in } b), (\lambda * : 1. \gamma_1))) \\
& \quad \circ \text{split}(\psi) \\
& \frac{\psi : \Gamma \simeq (\Gamma_1 \mid \Gamma_2) \quad f : (\Gamma_2^\circ \otimes A^\circ, 1) \leftrightarrow (B^\circ, B'^\circ)}{(\Gamma^\circ, \Gamma_1^\circ) \leftrightarrow (A^\circ \multimap (B^\circ \otimes (B'^\circ \multimap 1)), 1)} \multimap\sim\text{-I} \\
& \mapsto (\lambda(\gamma_1, \gamma_2) : (\Gamma_1^\circ \otimes \Gamma_2^\circ). ((\lambda a : A^\circ. f (\gamma_2, a)), (\lambda * : 1. \gamma_1))) \circ \text{split}(\psi) \\
& \frac{\psi : \Gamma \simeq (\Gamma_1 \mid \Gamma_2) \quad f : (\Gamma_2^\circ \otimes A^\circ, 1 \otimes A'^\circ) \leftrightarrow (B^\circ, 1)}{(\Gamma^\circ, \Gamma_1^\circ) \leftrightarrow (A^\circ \multimap (B^\circ \otimes A'^\circ), 1)} \sim\multimap\text{-I} \\
& \mapsto (\lambda(\gamma_1, \gamma_2) : (\Gamma_1^\circ \otimes \Gamma_2^\circ). \\
& \quad ((\lambda a : A^\circ. \text{let } (b, r_b) = f (\gamma_2, a) \text{ in let } (*, a') = r_b * \text{ in } (b, a')), (\lambda * : 1. \gamma_1))) \circ \text{split}(\psi) \\
& \frac{\psi : \Gamma \simeq (\Gamma_1 \mid \Gamma_2) \quad f : (\Gamma_2^\circ \otimes A^\circ, 1 \otimes A'^\circ) \leftrightarrow (B^\circ, B'^\circ)}{(\Gamma^\circ, \Gamma_1^\circ) \leftrightarrow ((A^\circ, A'^\circ) \leftrightarrow (B^\circ, B'^\circ), 1)} \sim\multimap\sim\text{-I}
\end{aligned}$$

$$\begin{aligned} &\mapsto (\lambda(\gamma_1, \gamma_2) : (\Gamma_1^\circ \otimes \Gamma_2^\circ). \\ &\quad ((\lambda a : A^\circ. \text{let } (b, r_b) = f(\gamma_2, a) \text{ in } (b, \lambda b' : B'^\circ. \text{let } (*, a') = r_b \ b' \text{ in } a')), (\lambda * : \mathbf{1}. \gamma_1))) \\ &\quad \circ \text{split}(\psi) \end{aligned}$$

$$\frac{f : (\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow (A^\circ \multimap B^\circ, \mathbf{1}) \quad g : (\Gamma_2^\circ, \Gamma_3^\circ) \leftrightarrow (A^\circ, \mathbf{1})}{(\Gamma_1^\circ, \Gamma_3^\circ) \leftrightarrow (B^\circ, \mathbf{1})} \multimap\text{-E}$$

$$\mapsto f \gg (\lambda(\gamma_2, h) : (\Gamma_2^\circ \otimes (A^\circ \multimap B^\circ)). (\widehat{\text{left}} \ h \ \hat{\circ} \ g) \ \gamma_2)$$

$$\frac{f : (\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow (A^\circ \multimap (B^\circ \otimes (B'^\circ \multimap \mathbf{1})), \mathbf{1}) \quad g : (\Gamma_2^\circ, \Gamma_3^\circ) \leftrightarrow (A^\circ, \mathbf{1})}{(\Gamma_1^\circ, \Gamma_3^\circ) \leftrightarrow (B^\circ, B'^\circ)} \multimap\sim\text{-E}$$

$$\mapsto f \gg (\lambda(\gamma_2, h) : (\Gamma_2^\circ \otimes (A^\circ \multimap (B^\circ \otimes (B'^\circ \multimap \mathbf{1}))))). (h \ \hat{\circ} \ g) \ \gamma_2)$$

$$\frac{f : (\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow (A^\circ \multimap (B^\circ \otimes A'^\circ), \mathbf{1}) \quad g : (\Gamma_2^\circ, \Gamma_3^\circ) \leftrightarrow (A^\circ, A'^\circ)}{(\Gamma_1^\circ, \Gamma_3^\circ) \leftrightarrow (B^\circ, \mathbf{1})} \sim\multimap\text{-E}$$

$$\begin{aligned} &\mapsto f \gg (\lambda(\gamma_2, h) : (\Gamma_2^\circ \otimes (A^\circ \multimap (B^\circ \otimes A'^\circ)))) \\ &\quad ((\lambda a : A^\circ. \text{let } (b, a') = h \ a \text{ in } (b, \lambda * : \mathbf{1}. a')) \ \hat{\circ} \ g) \ \gamma_2) \end{aligned}$$

$$\frac{f : (\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow ((A^\circ, A'^\circ) \leftrightarrow (B^\circ, B'^\circ), \mathbf{1}) \quad g : (\Gamma_2^\circ, \Gamma_3^\circ) \leftrightarrow (A^\circ, A'^\circ)}{(\Gamma_1^\circ, \Gamma_3^\circ) \leftrightarrow (B^\circ, B'^\circ)} \sim\multimap\sim\text{-E}$$

$$\mapsto f \gg (\lambda(\gamma_2, h) : (\Gamma_2^\circ \otimes ((A^\circ, A'^\circ) \leftrightarrow (B^\circ, B'^\circ))))). (h \ \hat{\circ} \ g) \ \gamma_2)$$

$$\frac{f : (\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow (A^\circ, \mathbf{1}) \quad g : (\Gamma_2^\circ, \Gamma_3^\circ) \leftrightarrow (B^\circ, \mathbf{1})}{(\Gamma_1^\circ, \Gamma_3^\circ) \leftrightarrow (A^\circ \otimes B^\circ, \mathbf{1})} \otimes\text{-I}$$

$$\mapsto f \gg (\lambda(\gamma_2, a) : (\Gamma_2^\circ \otimes A^\circ). \text{let } (b, r_b) = g \ \gamma_2 \text{ in } ((a, b), r_b))$$

$$\frac{\psi : \Gamma \simeq (\Gamma_1 \mid \Gamma_3) \quad f : (\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow (A^\circ, A'^\circ) \quad g : (\Gamma_3^\circ, \Gamma_4^\circ) \leftrightarrow (B^\circ, B'^\circ)}{(\Gamma^\circ, \Gamma_2^\circ \otimes \Gamma_4^\circ) \leftrightarrow (A^\circ \otimes B^\circ, A'^\circ \otimes B'^\circ)} \sim\otimes\text{-I}$$

$$\mapsto (\lambda(s, t) : (S \otimes T). \text{let } (a, r_a) = f \ s \text{ in let } (b, r_b) = g \ a \text{ in } \circ \text{split}(\psi))$$

$$\frac{f : (\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow (A^\circ \otimes B^\circ, \mathbf{1}) \quad g : ((\Gamma_2^\circ \otimes A^\circ) \otimes B^\circ, \Gamma_3^\circ) \leftrightarrow (C^\circ, C'^\circ)}{(\Gamma_1^\circ, \Gamma_3^\circ) \leftrightarrow (C^\circ, C'^\circ)} \otimes\text{-E}$$

$$\mapsto f \gg (\lambda(\gamma_2, (a, b)) : (\Gamma_2^\circ \otimes (A^\circ \otimes B^\circ)). g((\gamma_2, a), b))$$

$$\frac{\psi : \Gamma \simeq (\Gamma_1 \mid \Gamma_3) \quad \psi : \Gamma' \simeq (\Gamma_2 \mid \Gamma_4) \quad f : (\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow (A^\circ \otimes B^\circ, A'^\circ \otimes B'^\circ) \quad g : ((\Gamma_3^\circ \otimes A^\circ) \otimes B^\circ, (\Gamma_4^\circ \otimes A'^\circ) \otimes B'^\circ) \leftrightarrow (C^\circ, C'^\circ)}{(\Gamma^\circ, \Gamma'^\circ) \leftrightarrow (C^\circ, C'^\circ)} \sim\otimes\text{-E}$$

$$\begin{aligned} &\mapsto (\lambda(\gamma_1, \gamma_3) : (\Gamma_1^\circ, \Gamma_3^\circ). \text{let } ((a, b), r_f) = f \ \gamma_1 \text{ in let } (c, r_g) = g((\gamma_3, a), b) \text{ in} \\ &\quad (c, \lambda c' : C'^\circ. \text{let } ((\gamma_4, a'), b') = r_g \ c' \text{ in merge}(\psi)(r_f(a', b'), \gamma_4))) \circ \text{split}(\psi) \end{aligned}$$

$$\frac{}{(\Gamma^\circ, \Gamma^\circ) \leftrightarrow (\mathbf{1}, \mathbf{1})} \mathbf{1}\text{-I}$$

$$\mapsto \lambda\gamma : \Gamma^\circ. (*, \lambda * : \mathbf{1}. \gamma)$$

$$\frac{}{(\Gamma^\circ, \Gamma^\circ) \leftrightarrow (\mathbf{1}, \mathbf{1})} \sim\mathbf{1}\text{-I}$$

$$\mapsto \lambda \gamma : \Gamma^\circ. (*, \lambda * : \mathbf{1}. \gamma)$$

$$\frac{f : (\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow (\mathbf{1}, \mathbf{1}) \quad g : (\Gamma_2^\circ, \Gamma_3^\circ) \leftrightarrow (A^\circ, A'^\circ)}{(\Gamma_1^\circ, \Gamma_3^\circ) \leftrightarrow (A^\circ, A'^\circ)} \mathbf{1}\text{-E}$$

$$\mapsto f \gg (\lambda(\gamma_2, *) : (\Gamma_2^\circ \otimes \mathbf{1}). g \gamma_2)$$

$$\frac{\psi : \Gamma \simeq (\Gamma_1 \mid \Gamma_3) \quad \psi : \Gamma' \simeq (\Gamma_2 \mid \Gamma_4) \quad f : (\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow (\mathbf{1}, \mathbf{1}) \quad g : (\Gamma_3^\circ, \Gamma_4^\circ) \leftrightarrow (A^\circ, A'^\circ)}{(\Gamma_1^\circ \otimes \Gamma_3^\circ, \Gamma_2^\circ \otimes \Gamma_4^\circ) \leftrightarrow (A^\circ, A'^\circ)} \sim\mathbf{1}\text{-E}$$

$$\mapsto (\lambda(\gamma_1, \gamma_3) : (\Gamma_1^\circ \otimes \Gamma_3^\circ). \text{let } (*, r_f) = f \gamma_1 \text{ in let } (a, r_g) = g \gamma_3 \text{ in } (a, \lambda a' : A'^\circ. \text{merge}(\psi) (r_f *, r_g a')) \circ \text{split}(\psi))$$

$$\frac{f : (\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow (A^\circ, \mathbf{1})}{(\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow (A^\circ \oplus B^\circ, \mathbf{1})} \oplus\text{-IL}$$

$$\mapsto \widehat{\text{left inl}} \hat{\circ} f$$

$$\frac{f : (\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow (A^\circ \oplus B^\circ, \mathbf{1}) \quad g : (\Gamma_2^\circ \otimes A^\circ, \Gamma_3^\circ) \leftrightarrow (C^\circ, C'^\circ) \quad h : (\Gamma_2^\circ \otimes B^\circ, \Gamma_3^\circ) \leftrightarrow (C^\circ, C'^\circ)}{(\Gamma_1^\circ, \Gamma_3^\circ) \leftrightarrow (C^\circ, C'^\circ)} \oplus\text{-E}$$

$$\mapsto f \gg (\lambda(\gamma_2, x) : (\Gamma_2^\circ \otimes (A^\circ \oplus B^\circ)). \text{case } x \text{ of inl}(a) \rightarrow g(\gamma_2, a) \mid \text{inr}(b) \rightarrow h(\gamma_2, b))$$

$$\frac{\psi : \Gamma \simeq (\Gamma_1 \mid \Gamma_3) \quad \psi : \Gamma' \simeq (\Gamma_2 \mid \Gamma_4) \quad f : (\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow (A^\circ \oplus B^\circ, A'^\circ \oplus B'^\circ) \quad g : (\Gamma_3^\circ \otimes A^\circ, \Gamma_4^\circ \otimes A'^\circ) \leftrightarrow (C^\circ, C'^\circ) \quad h : (\Gamma_3^\circ \otimes B^\circ, \Gamma_4^\circ \otimes B'^\circ) \leftrightarrow (C^\circ, C'^\circ)}{(\Gamma_1^\circ \otimes \Gamma_3^\circ, \Gamma_2^\circ \otimes \Gamma_4^\circ) \leftrightarrow (C^\circ, C'^\circ)} \sim\oplus\text{-E}$$

$$\mapsto (\lambda(\gamma_1, \gamma_3) : (\Gamma_1^\circ \otimes \Gamma_3^\circ). \text{let } (x, r_f) = f \gamma_1 \text{ in case } x \text{ of } \mid \text{inl}(a) \rightarrow \text{let } (c, r_g) = g(\gamma_3, a) \text{ in } (c, \lambda c' : C'^\circ. \text{let } (\gamma_4, a') = r_g c' \text{ in merge}(\psi) (r_f (\text{inl}(a')), \gamma_4)) \mid \text{inr}(b) \rightarrow \text{let } (c, r_h) = h(\gamma_3, b) \text{ in } (c, \lambda c' : C'^\circ. \text{let } (\gamma_4, b') = r_h c' \text{ in merge}(\psi) (r_f (\text{inr}(b')), \gamma_4))) \circ \text{split}(\psi))$$

$$\frac{f : (\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow (\mathbf{0}, \mathbf{1})}{(\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow (A^\circ, A'^\circ)} \mathbf{0}\text{-E}$$

$$\mapsto (\lambda x : \mathbf{0}. \text{false}_{A^\circ \otimes (A'^\circ \multimap \mathbf{1})}(x)) \hat{\circ} f$$

$$\frac{f : (\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow (\mathbf{0}, \mathbf{0})}{(\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow (A^\circ, A'^\circ)} \sim\mathbf{0}\text{-E}$$

$$\mapsto (\lambda x : \mathbf{0}. \text{false}_{A^\circ \otimes (A'^\circ \multimap \mathbf{0})}(x)) \hat{\circ} f$$

$$\frac{f : (\Gamma_2^\circ, \mathbf{1}) \leftrightarrow (A^\circ, \mathbf{1}) \quad g : (\Gamma_2^\circ, \mathbf{1}) \leftrightarrow (B^\circ, \mathbf{1})}{(\Gamma_1^\circ \otimes \Gamma_2^\circ, \Gamma_1^\circ) \leftrightarrow (A^\circ \& B^\circ, \mathbf{1})} \&\text{-I}$$

$$\begin{aligned} &\mapsto \lambda(\gamma_1, \gamma_2) : (\Gamma_1^\circ \otimes \Gamma_2^\circ). \\ &\quad (((\text{let } (a, r_a) = f \ \gamma_2 \text{ in let } * = r_a * \text{ in } a) \ \& \ (\text{let } (b, r_b) = g \ \gamma_2 \text{ in let } * = r_b * \text{ in } b)), \\ &\quad \lambda * : \mathbf{1}. \gamma_1) \end{aligned}$$

$$\begin{aligned} &\frac{f : (\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow (A^\circ \ \& \ B^\circ, \mathbf{1})}{(\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow (A^\circ, \mathbf{1})} \text{ \&-EL} \\ &\mapsto \widehat{\text{left}} (\lambda x : A^\circ \ \& \ B^\circ. \text{fst}(x)) \hat{\circ} f \end{aligned}$$

$$\begin{aligned} &\frac{f : (\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow (A^\circ, \mathbf{1})}{(\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow (\top, \mathbf{1})} \top\text{-I} \\ &\mapsto \widehat{\text{left}} (\lambda x : A^\circ. \text{true}(x)) \hat{\circ} f \end{aligned}$$

$$\begin{aligned} &\frac{f_1 : (\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow (!A_1^\circ, \mathbf{1}) \quad \cdots \quad f_n : (\Gamma_n^\circ, \Gamma_{n+1}^\circ) \leftrightarrow (!A_n^\circ, \mathbf{1})}{(\Gamma_1^\circ, \Gamma_{n+1}^\circ) \leftrightarrow (!B^\circ, \mathbf{1})} \text{ PROM} \\ &\quad g : (!A_1^\circ \otimes \cdots \otimes !A_n^\circ, \mathbf{1}) \leftrightarrow (B^\circ, \mathbf{1}) \\ &\mapsto (\lambda \gamma_1 : (\mathbf{1} \multimap \Gamma_1^\circ). \text{let } (a_1, \gamma_2) = f_1 \ (\gamma_1 *) \text{ in} \\ &\quad (\lambda \gamma_2 : (\mathbf{1} \multimap \Gamma_2^\circ). \text{let } (a_2, \gamma_3) = f_2 \ (\gamma_2 *) \text{ in } (\cdots (\\ &\quad (\text{promote } a_1, \dots, a_n \text{ for } x_1, \dots, x_n \text{ in} \\ &\quad \text{let } (b, r_b) = g \ (x_1, \dots, x_n) \text{ in let } * = r_b * \text{ in } b), \gamma_{n+1}) \\ &\quad) \cdots) \\ &\quad (\lambda * : \mathbf{1}. \gamma_1) \end{aligned}$$

$$\begin{aligned} &\frac{f : (\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow (!A^\circ, \mathbf{1})}{(\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow (\mathbf{1}, \mathbf{1})} \text{ DISC} \\ &\mapsto \widehat{\text{left}} (\lambda x : !A^\circ. \text{discard } x) \hat{\circ} f \end{aligned}$$

$$\begin{aligned} &\frac{f : (\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow (!A^\circ, \mathbf{1})}{(\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow (!A^\circ \otimes !A^\circ, \mathbf{1})} \text{ COPY} \\ &\mapsto \widehat{\text{left}} (\lambda x : !A^\circ. \text{copy } x) \hat{\circ} f \end{aligned}$$

$$\begin{aligned} &\frac{f : (\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow (!A^\circ, \mathbf{1})}{(\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow (A^\circ, \mathbf{1})} \text{ DER} \\ &\mapsto \widehat{\text{left}} (\lambda x : !A^\circ. \text{derelict } x) \hat{\circ} f \end{aligned}$$

THEOREM C.2. *If $\Gamma_1 \vdash t : A \dashv \Gamma_2$ is a valid type derivation in λ_F , then $\vdash \bar{t} : (\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow (A^\circ, \mathbf{1})$ is a valid type derivation in λ_L . Similarly, if $\Gamma_1 \vdash u : A/A' \dashv \Gamma_2$ is a valid type derivation in λ_F , then $\vdash \bar{u} : (\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow (A^\circ, A'^\circ)$ is a valid type derivation in λ_L .*

PROOF. By construction, via rule induction. □

D Semantics

In this section we introduce an operational semantics for λ_L with its conventional type system.

D.1 Semantic Rules

Recall the following syntax for λ_L terms t , and define the notion of *value* v as follows:

$$\begin{aligned}
 t &:= x \mid (\text{let } x = t_1 \text{ in } t_2) \mid (\lambda x : A. t) \mid t_1 t_2 \\
 &\mid (t_1, t_2) \mid (\text{let } (x, y) = t_1 \text{ in } t_2) \mid * \mid (\text{let } * = t_1 \text{ in } t_2) \\
 &\mid \text{inl}_{A \oplus B}(t) \mid \text{inr}_{A \oplus B}(t) \mid (\text{case } t \text{ of } \text{inl}(x) \rightarrow t_1 \mid \text{inr}(y) \rightarrow t_2) \mid \text{false}_A(t) \\
 &\mid (t_1 \& t_2) \mid \text{fst}(t) \mid \text{snd}(t) \mid \text{true}(t) \\
 &\mid \text{promote } t_1, \dots, t_n \text{ for } x_1, \dots, x_n \text{ in } t \mid \text{discard } t \mid \text{copy } t \mid \text{derelict } t \\
 v &:= (\lambda x : A. t) \mid (v_1, v_2) \mid * \mid \text{inl}_{A \oplus B}(v) \mid \text{inr}_{A \oplus B}(v) \mid (t_1 \& t_2) \mid \text{true}(v) \mid \text{promote for in } t
 \end{aligned}$$

Abbreviate ‘promote for in t ’ as $!t$. Our semantic judgements are of the form $t \Downarrow v$, where t must be a closed term. This semantics is directly inspired by the operational semantics due to Abramsky [1, section 3].

$$\begin{aligned}
 &\frac{t_1 \Downarrow v_1 \quad t_2[v_1/x] \Downarrow v_2}{(\text{let } t_1 = x \text{ in } t_2) \Downarrow v_2} \text{LET} & \frac{}{(\lambda x : A. t) \Downarrow (\lambda x : A. t)} \multimap\text{-I} \\
 &\frac{t_1 \Downarrow (\lambda x : A. t) \quad t_2 \Downarrow v_1 \quad t[v_1/x] \Downarrow v_2}{(t_1 t_2) \Downarrow v_2} \multimap\text{-E} & \frac{t_1 \Downarrow v_1 \quad t_2 \Downarrow v_2}{(t_1, t_2) \Downarrow (v_1, v_2)} \otimes\text{-I} \\
 &\frac{t_1 \Downarrow (v_1, v_2) \quad t_2[v_1/x, v_2/y] \Downarrow v}{(\text{let } (x, y) = t_1 \text{ in } t_2) \Downarrow v} \otimes\text{-E} & \frac{}{* \Downarrow *} \text{1-I} & \frac{t_1 \Downarrow * \quad t_2 \Downarrow v}{(\text{let } * = t_1 \text{ in } t_2) \Downarrow v} \text{1-E} \\
 &\frac{t \Downarrow v}{\text{inl}_{A \oplus B}(t) \Downarrow \text{inl}_{A \oplus B}(v)} \oplus\text{-IL} & \frac{t_1 \Downarrow \text{inl}_{A \oplus B}(v_1) \quad t_2[v_1/x] \Downarrow v_2}{(\text{case } t_1 \text{ of } \text{inl}(x) \rightarrow t_2 \mid \text{inr}(y) \rightarrow t_3) \Downarrow v_2} \oplus\text{-EL} \\
 &\frac{}{(t_1 \& t_2) \Downarrow (t_1 \& t_2)} \&\text{-I} & \frac{t \Downarrow (t_1 \& t_2) \quad t_1 \Downarrow v}{\text{fst}(t) \Downarrow v} \&\text{-EL} & \frac{t \Downarrow v}{\text{true}(t) \Downarrow \text{true}(v)} \top\text{-I} \\
 &\frac{t_1 \Downarrow !t'_1 \quad \dots \quad t_n \Downarrow !t'_n}{\text{promote } t_1, \dots, t_n \text{ for } x_1, \dots, x_n \text{ in } t \Downarrow !(t[t'_1/x_1, \dots, t'_n/x_n])} \text{PROM} & \frac{t \Downarrow !t'}{(\text{discard } t) \Downarrow *} \text{DISC} \\
 &\frac{t \Downarrow !t'}{(\text{copy } t) \Downarrow (!t', !t')} \text{COPY} & \frac{t \Downarrow !t' \quad t' \Downarrow v}{(\text{derelict } t) \Downarrow v} \text{DER}
 \end{aligned}$$

PROPOSITION D.1 (DETERMINISM). *If $t \Downarrow v_1$ and $t \Downarrow v_2$ then $v_1 = v_2$.*

PROOF. Rule induction. □

PROPOSITION D.2. *All values are closed terms and reduce to themselves.*

PROOF. Trivial. □

D.2 Logical Relations

Define $\mathcal{W}\llbracket A \rrbracket$ for the collection of pairs of well-typed closed terms:

$$\mathcal{W}\llbracket A \rrbracket = \{(t_1, t_2) \mid (\vdash t_1 : A) \wedge (\vdash t_2 : A)\}$$

Define logical relations $\mathcal{V}[[A]] \subseteq \text{Value} \times \text{Value}$ and $\mathcal{T}[[A]] \subseteq \text{Term} \times \text{Term}$ as follows, by recursion on types.

$$\begin{aligned}
\mathcal{V}[[A \multimap B]] &:= \mathcal{W}[[A \multimap B]] \cap \\
&\quad \{((\lambda x : A. t_1), (\lambda x : A. t_2)) \mid \forall (v_1, v_2) \in \mathcal{V}[[A]], (t_1[v_1/x], t_2[v_2/x]) \in \mathcal{T}[[B]]\} \\
\mathcal{V}[[A \otimes B]] &:= \{((v_1, v_2), (v_3, v_4)) \mid (v_1, v_3) \in \mathcal{V}[[A]] \wedge (v_2, v_4) \in \mathcal{V}[[B]]\} \\
\mathcal{V}[[1]] &:= \{(*, *)\} \\
\mathcal{V}[[A \oplus B]] &:= \{(\text{inl}_{A \oplus B}(v_1), \text{inl}_{A \oplus B}(v_2)) \mid (v_1, v_2) \in \mathcal{V}[[A]]\} \\
&\quad \cup \{(\text{inr}_{A \oplus B}(v_1), \text{inr}_{A \oplus B}(v_2)) \mid (v_1, v_2) \in \mathcal{V}[[B]]\} \\
\mathcal{V}[[0]] &:= \emptyset \\
\mathcal{V}[[A \& B]] &:= \{((t_1 \& t_2), (t_3 \& t_4)) \mid (t_1, t_3) \in \mathcal{T}[[A]] \wedge (t_2, t_4) \in \mathcal{T}[[B]]\} \\
\mathcal{V}[[\top]] &:= \mathcal{W}[[\top]] \cap \{(\text{true}(v_1), \text{true}(v_2))\} \\
\mathcal{V}[[!A]] &:= \{(!t_1, !t_2) \mid (t_1, t_2) \in \mathcal{T}[[A]]\} \\
\mathcal{T}[[A]] &:= \mathcal{W}[[A]] \cap \{(t_1, t_2) \mid \exists (v_1, v_2) \in \mathcal{V}[[A]], (t_1 \Downarrow v_1) \wedge (t_2 \Downarrow v_2)\}
\end{aligned}$$

LEMMA D.3. *For all types A ,*

$$\mathcal{V}[[A]] \cup \mathcal{T}[[A]] \subseteq \mathcal{W}[[A]]$$

PROOF. Induction on A . □

We define the interpretations of contexts as follows:

$$C[[x_1 : A_1, \dots, x_n : A_n]] = \{((v_1, \dots, v_n), (v'_1, \dots, v'_n)) \mid \forall i, (v_i, v'_i) \in \mathcal{V}[[A_i]]\}$$

We write $t[\gamma]$ for the simultaneous substitution of the free variables of t with the entries in γ , where we assume that terms are carried with the variable names that appear in their contexts so that this is well-defined. Note that elements of $C[[\Gamma_1, \Gamma_2]]$ are in bijection with quadruples $((\gamma_1, \gamma_2), (\gamma'_1, \gamma'_2))$ where $(\gamma_1, \gamma'_1) \in C[[\Gamma_1]]$ and $(\gamma_2, \gamma'_2) \in C[[\Gamma_2]]$; we use this correspondence throughout our proofs. Finally, we define:

$$(\Gamma \models t_1 \approx t_2 : A) := \forall (\gamma_1, \gamma_2) \in C[[\Gamma]], (t_1[\gamma_1], t_2[\gamma_2]) \in \mathcal{T}[[A]]$$

LEMMA D.4.

$$(\models t_1 \approx t_2 : A) \iff (t_1, t_2) \in \mathcal{T}[[A]]$$

PROOF. Trivial. □

Following standard practice, we then prove *compatibility lemmas* in the shape of typing inference rules, demonstrating that $\approx$ is a congruence for all term constructors. The fundamental theorem of logical relations then follows by a simple rule induction.

THEOREM D.5 (FUNDAMENTAL THEOREM). *Suppose that $\vdash t : A$. Then $\models t \approx t : A$.*

LEMMA D.6 (SYMMETRY). *The logical relations $\mathcal{V}[[-]]$, $\mathcal{T}[[-]]$, $C[[-]]$ are all symmetric, so the relation $\Gamma \models (-) \approx (-) : A$ is symmetric.*

PROOF. A simple induction on types proves the result for $\mathcal{V}[[-]]$, $\mathcal{T}[[-]]$ simultaneously. More precisely, for a given type A , we assume both results hold for all smaller types, then prove the result for $\mathcal{V}[[A]]$, and conclude by proving the result for $\mathcal{T}[[A]]$. □

LEMMA D.7 (TRANSITIVITY). *The aforementioned relations are all transitive.*

PROOF. The same style of induction on types. $\square$

THEOREM D.8. $\Gamma \models (-) \approx (-) : A$ is an equivalence relation and a congruence on well-typed terms $\Gamma \vdash (-) : A$.

LEMMA D.9. If $t \Downarrow v$, then $\models t \approx v$.

This allows us to extend theorems about values to theorems about terms, by using congruence. We then prove function extensionality and standard β and η rules.

D.3 Categorical Semantics

Define a category $\mathcal{C}$ where the objects are λ_L types, and whose morphisms $A \rightarrow B$ are equivalence classes of well-typed closed terms $\vdash t : A \multimap B$ under the relation $\models (-) \approx (-) : A \multimap B$. The identity morphism $1_A : A \rightarrow A$ is the class of $\lambda x : A. x$, and the composition of $[t_1] : B \rightarrow C$ and $[t_2] : A \rightarrow B$ is the class of $t_3 := \lambda x : A. t_1 (t_2 x)$, which is well-defined since $\approx$ is a congruence for all well-typed term constructors.

PROPOSITION D.10. $\mathcal{C}$ is a category.

PROOF. Let $[t] : A \rightarrow B$. Then $[t] \circ 1_A$ is the class of:

$$(\lambda x : A. t ((\lambda y : A. y) x)) \approx (\lambda x : A. t x) \approx t$$

and $1_B \circ [t]$ is the class of:

$$(\lambda x : A. (\lambda y : B. y) (t x)) \approx (\lambda x : A. t x) \approx t$$

Next, if $[t_1] : C \rightarrow D$, $[t_2] : B \rightarrow C$, and $[t_1] : A \rightarrow B$, we compute $([t_1] \circ [t_2]) \circ [t_3]$ and $[t_1] \circ ([t_2] \circ [t_3])$ as:

$$\begin{aligned} ([t_1] \circ [t_2]) \circ [t_3] &\approx \lambda x : A. (\lambda y : B. t_1 (t_2 y)) (t_3 x) \\ &\approx \lambda x : A. t_1 (t_2 (t_3 x)) \\ &\approx \lambda x : A. t_1 ((\lambda y : A. t_2 (t_3 y)) x) \\ &\approx [t_1] \circ ([t_2] \circ [t_3]) \end{aligned}$$

as required. $\square$

The next propositions are proven in similar ways, appealing to the standard results indicated above such as function extensionality and β/η -laws.

PROPOSITION D.11. $\mathcal{C}$ is a symmetric monoidal category, where:

- the functor $\otimes$ is given on objects by the tensor product $\otimes$ of types, and on morphisms by

$$\frac{\vdash t_1 : A \multimap B \quad \vdash t_2 : C \multimap D}{\vdash (\lambda x : (A \otimes C). \text{let } (a, c) = x \text{ in } (t_1 a, t_2 c)) : (A \otimes C) \multimap (B \otimes D)}$$

- the monoidal unit is 1 ;
- the associator $\alpha : (A \otimes B) \otimes C \cong A \otimes (B \otimes C)$ is the class of $\lambda((a, b), c). (a, (b, c))$;
- the unitors $\lambda : 1 \otimes A \cong A$ and $\rho : A \otimes 1 \cong A$ are the classes of $\lambda(*, a). a$ and $\lambda(a, *). a$ respectively; and
- the symmetry $\gamma : A \otimes B \cong B \otimes A$ is the class of $\lambda(a, b). (b, a)$.

PROPOSITION D.12. $\mathcal{C}$ is closed monoidal, where the right adjoint to $A \otimes (-)$ is $A \multimap (-)$, and the evaluation morphism $\varepsilon : A \otimes (A \multimap B) \rightarrow B$ is the class of $\lambda(x, f). f x$.

PROPOSITION D.13. $A \oplus B$ is the categorical coproduct of A and B , with coprojections $\iota_1 := [\lambda x : A. \text{inl}_{A \oplus B}(x)]$ and $\iota_2 := [\lambda x : A. \text{inr}_{A \oplus B}(x)]$.

PROPOSITION D.14. $\mathbf{0}$ is the initial object, with initial morphism $\lambda x : \mathbf{0}. \text{false}_A(x)$.

PROPOSITION D.15. $A \& B$ is the categorical product of A and B , with legs $\pi_1 := [\lambda x : (A \& B). \text{fst}(x)]$ and $\pi_2 := [\lambda x : (A \& B). \text{snd}(x)]$.

PROPOSITION D.16. $\top$ is the terminal object, with terminal morphism $\lambda x : A. \text{true}(x)$.

PROPOSITION D.17. $\mathcal{C}$ has a comonad $\mathbb{G} = (!, \delta, \varepsilon)$ where

- the functorial action of $!$ on morphisms is

$$\frac{\vdash t : A \multimap B}{\vdash (\lambda x : !A. \text{promote } x \text{ for } y \text{ in } t \text{ (derelict } y)) : (!A) \multimap (!B)}$$

- the comultiplication $\delta_A : !A \rightarrow !(A)$ is

$$\delta_A := [\lambda x : !A. \text{promote } x \text{ for } y \text{ in } y]$$

- the counit $\varepsilon_A : !A \rightarrow A$ is

$$\varepsilon_A := [\lambda x : !A. \text{derelict } x]$$

Recall that an object A in a category is called *subterminal* if all parallel morphisms with codomain A are equal.

LEMMA D.18. A type is subterminal if and only if whenever $(v_1, v_1) \in \mathcal{V}[\![A]\!]$ and $(v_2, v_2) \in \mathcal{V}[\![A]\!]$, we have $(v_1, v_2) \in \mathcal{V}[\![A]\!]$.

PROOF. Suppose that A satisfies the given condition, and let $[t_1], [t_2] : B \rightarrow A$. To show $\vdash t_1 \approx t_2 : B \multimap A$, consider $(v_1, v_2) \in \mathcal{V}[\![B]\!]$ and prove $\vdash t_1 v_1 \approx t_2 v_2 : A$. By the fundamental theorem, $(t_1 v_1, t_1 v_1)$ and $(t_2 v_2, t_2 v_2)$ are elements of $\mathcal{V}[\![A]\!]$, then by the condition, $(t_1 v_1, t_2 v_2) \in \mathcal{V}[\![A]\!]$ as required.

Now suppose that A is subterminal, and consider $(v_1, v_1) \in \mathcal{V}[\![A]\!]$ and $(v_2, v_2) \in \mathcal{V}[\![A]\!]$. Since v_1 and v_2 are well-typed terms of type A , we know that the terms $(\lambda * : \mathbf{1}. v_1)$ and $(\lambda * : \mathbf{1}. v_2)$ are well-typed. As A is subterminal, the two terms give rise to equal morphisms. Applying extensionality, we conclude that $\vdash v_1 \approx v_2 : A$, giving $(v_1, v_2) \in \mathcal{V}[\![A]\!]$ as values reduce to themselves, as required. $\square$

LEMMA D.19. $!$ preserves subterminal objects.

PROOF. Suppose that A is subterminal; we show that $!A$ is subterminal. By the lemma above, let $(!t_1, !t_1), (!t_2, !t_2) \in \mathcal{V}[\![!A]\!]$, so $(t_1, t_1), (t_2, t_2) \in \mathcal{T}[\![A]\!]$. By definition of $\mathcal{T}[\![A]\!]$, both t_1 and t_2 reduce to values v_1, v_2 where $(v_1, v_1), (v_2, v_2) \in \mathcal{V}[\![A]\!]$. Since A is subterminal, we have $(v_1, v_2) \in \mathcal{V}[\![A]\!]$, from which we conclude that $(t_1, t_2) \in \mathcal{T}[\![A]\!]$, so $(!t_1, !t_2) \in \mathcal{V}[\![!A]\!]$. $\square$

PROPOSITION D.20 (TRANSMUTATION PRINCIPLE). There are natural isomorphisms:

$$\begin{aligned} m_{A,B}^2 &: !A \otimes !B \cong !(A \& B) \\ m_{A,B}^2 &= [\lambda(x, y) : !A. \text{promote } x, y \text{ for } z, w \text{ in (derelict } z \& \text{ derelict } w)] \\ (m_{A,B}^2)^{-1} &= [\lambda x : !(A \& B). \text{let } (y, z) = \text{copy } x \text{ in} \\ &\quad (\text{promote } y \text{ for } y' \text{ in fst(derelict } y'), \text{promote } z \text{ for } z' \text{ in snd(derelict } z')))] \\ m^0 &: \mathbf{1} \cong \top \\ m^0 &= [\lambda x : \mathbf{1}. !(\text{true}(x))] \\ (m^0)^{-1} &= [\lambda x : !\top. \text{discard } x] \end{aligned}$$

The result about subterminal objects gives an easy proof that m^0 is indeed an isomorphism.

COROLLARY D.21. *We have promotion morphisms:*

$$\begin{aligned}\text{join}_0 &: \mathbf{1} \rightarrow !\mathbf{1} \\ \text{join}_2 &: !A \otimes !B \rightarrow !(A \otimes B)\end{aligned}$$

defined by the composites

$$\begin{aligned}\text{join}_0 &= \mathbf{1} \xrightarrow{m^0} !\top \xrightarrow{\delta} !!\top \xrightarrow{!(m^0)^{-1}} !\mathbf{1} \\ \text{join}_2 &= !A \otimes !B \xrightarrow{m^2} !(A \& B) \xrightarrow{\delta} !! (A \& B) \xrightarrow{!(m^2)^{-1}} !(A \otimes B) \xrightarrow{!(\varepsilon \otimes \varepsilon)} !(A \otimes B)\end{aligned}$$

PROPOSITION D.22. *We have weakening and contraction morphisms:*

$$\begin{aligned}\text{wkg} &: !A \rightarrow \mathbf{1} \\ \text{wkg} &= [\lambda x : !A. \text{discard } x] \\ \text{ctr} &: !A \rightarrow !A \otimes !A \\ \text{ctr} &= [\lambda x : !A. \text{copy } x]\end{aligned}$$

In conclusion:

THEOREM D.23. $(\mathcal{C}, \otimes, \mathbf{1}, \multimap)$ is a symmetric monoidal closed category with finite products $(\&, \top)$ and coproducts $(\oplus, \mathbf{0})$, and endowed with a comonad ‘!’ with designated morphisms of type:

$$\text{wkg} : !A \rightarrow \mathbf{1}; \quad \text{ctr} : !A \rightarrow !A \otimes !A; \quad \text{join}_0 : \mathbf{1} \rightarrow !\mathbf{1}; \quad \text{join}_2 : !A \otimes !B \rightarrow !(A \otimes B)$$

Remark D.24. This is a weakening of the definition of a linear category (see e.g. [2, 6]). This contains enough data to interpret all of the connectives of λ_L , but we do not record any extra laws about how these connectives compute. $\diamond$

E Semantic Proofs

Let $(\mathcal{D}, \otimes, \mathbf{1}, \multimap)$ be a symmetric monoidal closed category (e.g. [5]). We assume without loss of generality that $\mathcal{D}$ is *strict* monoidal so that we can avoid explicit use of associators and unitors. Mac Lane [5] and Wilson et al. [7] describe ways to do this. We also assume some or all of the following:

- binary coproducts $A + B$ with coprojections ι_1, ι_2 ;
- an initial object $\mathbf{0}$ with morphisms $?_A : \mathbf{0} \rightarrow A$;
- binary products $A \times B$ with projections π_1, π_2 ;
- a terminal object $\top$ with morphisms $!_A : A \rightarrow \top$;
- a comonad $\mathbb{G} = (G, \varepsilon, \delta)$ where $\varepsilon : G \rightarrow \mathbf{1}_{\mathcal{D}}$ is the counit and $\delta : G \rightarrow G^2$ is the comultiplication, together with the following designated morphisms (subject to no laws):

$$\text{wkg} : GA \rightarrow \mathbf{1}; \quad \text{ctr} : GA \rightarrow GA \otimes GA; \quad \text{join}_0 : \mathbf{1} \rightarrow G\mathbf{1}; \quad \text{join}_2 : GA \otimes GB \rightarrow G(A \otimes B)$$

For convenience, we use string diagrams to visualise our semantic proofs. A comprehensive introduction to string diagrams can be found in [4], and an introduction to their use specifically for functional computation can be found in [3]. In order to do this, we will now establish notation and string-diagrammatic representations of the constructions used in this work; much of this notation is due to [3] and [7].

Due to strictness, contexts Γ can be identified with their packed representations Γ° . More precisely, a context can be thought of either as a bundle of adjacent wires or as a single wire, as desired; moreover, such a bundle can be freely reassociated. To echo this, we use a ‘double-struck’ style to

draw wires corresponding to contexts. Where we do not make use of strictness, we define operators for $\otimes$ and 1 as follows:

$$\begin{array}{ccc}
 A \otimes B & \text{---} \text{ } \begin{array}{c} \diagup \\ \diagdown \end{array} & \begin{array}{c} A \\ B \end{array} & 1 \longrightarrow \\
 A & \text{---} \text{ } \begin{array}{c} \diagdown \\ \diagup \end{array} & B & \longleftarrow 1 \\
 B & \text{---} & A \otimes B &
 \end{array}$$

The connectors for 1 , for example, allow us to freely create and delete objects of type 1 . For abstraction and application we introduce the following additional string diagram notation:

$$\begin{array}{ll}
 \text{abstraction} & \begin{array}{l} \tilde{f} : \Gamma \rightarrow (A \multimap B) \\ \text{where } f : \Gamma \otimes A \rightarrow B \end{array} \quad \mapsto \quad \begin{array}{c} \Gamma \text{---} \boxed{\begin{array}{c} \Gamma \quad B \\ \text{---} f \text{---} \\ A \end{array}} \text{---} A \multimap B \end{array} \\
 \text{application} & \begin{array}{l} \varepsilon : (A \multimap B) \otimes A \rightarrow B, \\ \text{the counit of } (-) \otimes A \dashv A \multimap (-) \end{array} \quad \mapsto \quad \begin{array}{c} A \multimap B \\ \text{---} \text{ } \begin{array}{c} \diagup \\ \diagdown \end{array} \text{---} B \\ A \end{array}
 \end{array}$$

For the comonad $\mathbb{G}$, we extend join_0 and join_2 to arbitrary n by iteration, assuming that (for concreteness) the applications of join are associated to the left:

$$\text{join}_n : GA_1 \otimes \cdots \otimes GA_n \rightarrow G(A_1 \otimes \cdots \otimes A_n)$$

We additionally use boxes to signify application of functors. If $f : A \rightarrow B$, we write $Gf : GA \rightarrow GB$ using the following notation:

$$GA \text{---} \boxed{\begin{array}{c} A \quad B \\ \text{---} f \text{---} \\ G \end{array}} \text{---} GB$$

We record here how the term syntax of λ_L , as well as the corresponding lens translations for the corresponding λ_F syntax, are interpreted in our string diagrams. First, note that

$$(\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow (A, 1) \cong \Gamma_1^\circ \multimap (\Gamma_2^\circ \otimes A)$$

and this isomorphism is natural in all parameters. In this section, all lenses have this form with final type $B = 1$, so to simplify notation, we transparently replace all such lenses with their images under this isomorphism. We use boxes with a downward-pointing wire to indicate this in our diagrams, so for example:

$$\Gamma_1 \text{---} \boxed{t} \text{---} \Gamma_2 \quad \text{abbreviates} \quad \Gamma_1 \text{---} \boxed{t} \text{---} \begin{array}{c} \Gamma_2 \\ 1 \multimap A \\ \longleftarrow 1 \end{array} \text{---} A$$

The remainder of this section describes the string diagrams that the term syntax of λ_L corresponds to, as well as the diagrams associated with each lens translation for such syntax.

identity/variable

$$\frac{}{x : A \vdash x : A}$$

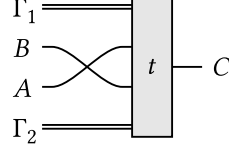
$$A \text{---} A$$

$$\frac{}{\Gamma, x : A \vdash x : A \dashv \Gamma, x^\dagger}$$

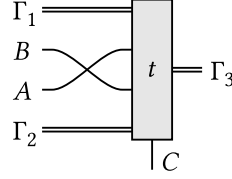
$$\frac{\Gamma = \Gamma}{A \searrow_A}$$

exchange

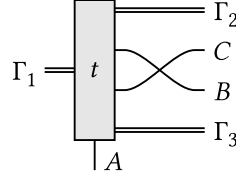
$$\frac{\Gamma_1, x : A, y : B, \Gamma_2 \vdash t : C}{\Gamma_1, y : B, x : A, \Gamma_2 \vdash t : C}$$



$$\frac{\Gamma_1, x : A, y : B, \Gamma_2 \vdash t : C \dashv \Gamma_3}{\Gamma_1, y : B, x : A, \Gamma_2 \vdash t : C \dashv \Gamma_3}$$

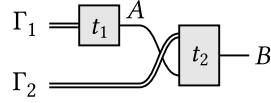


$$\frac{\Gamma_1 \vdash t : A \dashv \Gamma_2, x : B, y : C, \Gamma_3}{\Gamma_1 \vdash t : A \dashv \Gamma_2, y : C, x : B, \Gamma_3}$$

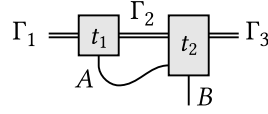


let

$$\frac{\Gamma_1 \vdash t_1 : A \quad \Gamma_2, x : A \vdash t_2 : B}{\Gamma_1, \Gamma_2 \vdash (\text{let } x = t_1 \text{ in } t_2) : B}$$

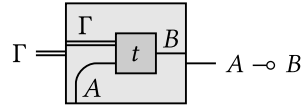


$$\frac{\Gamma_1 \vdash t_1 : A \dashv \Gamma_2 \quad \Gamma_2, x : A \vdash t_2 : B \dashv \Gamma_3, x^\dagger}{\Gamma_1 \vdash (\text{let } x = t_1 \text{ in } t_2) : B \dashv \Gamma_3}$$



function

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash (\lambda x : A. t) : A \multimap B}$$



$$\frac{\Gamma_2, x : A \vdash t : B \dashv \Gamma_2^\dagger, x^\dagger}{\Gamma_1, \Gamma_2 \vdash (\lambda x : A. t) : (A \multimap B) \dashv \Gamma_1, \Gamma_2^\dagger}$$

$$\frac{\Gamma_1 \vdash t_1 : A \multimap B \quad \Gamma_2 \vdash t_2 : A}{\Gamma_1, \Gamma_2 \vdash t_1 t_2 : B}$$

$$\frac{\Gamma_1 \vdash t_1 : A \multimap B \dashv \Gamma_2 \quad \Gamma_2 \vdash t_2 : A \dashv \Gamma_3}{\Gamma_1 \vdash t_1 t_2 : B \dashv \Gamma_3}$$

pair

$$\frac{\Gamma_1 \vdash t_1 : A \quad \Gamma_2 \vdash t_2 : B}{\Gamma_1, \Gamma_2 \vdash (t_1, t_2) : A \otimes B}$$

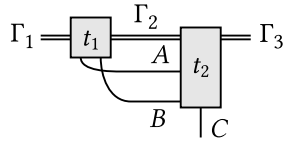
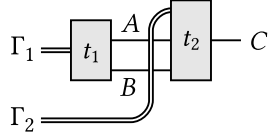
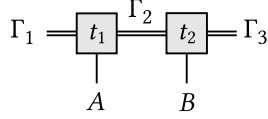
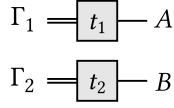
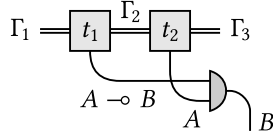
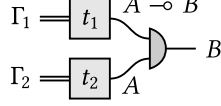
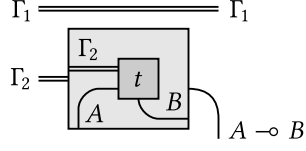
$$\frac{\Gamma_1 \vdash t_1 : A \dashv \Gamma_2 \quad \Gamma_2 \vdash t_2 : B \dashv \Gamma_3}{\Gamma_1 \vdash (t_1, t_2) : A \otimes B \dashv \Gamma_3}$$

$$\frac{\Gamma_1 \vdash t_1 : A \otimes B \quad \Gamma_2, x : A, y : B \vdash t_2 : C}{\Gamma_1, \Gamma_2 \vdash (\text{let } (x, y) = t_1 \text{ in } t_2) : C}$$

$$\frac{\Gamma_1 \vdash t_1 : A \otimes B \dashv \Gamma_2 \quad \Gamma_2, x : A, y : B \vdash t_2 : C \dashv \Gamma_3}{\Gamma_1 \vdash (\text{let } (x, y) = t_1 \text{ in } t_2) : C \dashv \Gamma_3}$$

unit

$$\frac{}{\vdash * : \mathbf{1}}$$



$$\frac{}{\Gamma \vdash * : \mathbf{1} \dashv \Gamma}$$

$$\Gamma \equiv \Gamma$$

$$\frac{\Gamma_1 \vdash t_1 : \mathbf{1} \quad \Gamma_2 \vdash t_2 : A}{\Gamma_1, \Gamma_2 \vdash (\text{let } * = t_1 \text{ in } t_2) : A}$$

$$\Gamma_1 \equiv \boxed{t_1} \xrightarrow{\mathbf{1}} \quad \Gamma_2 \equiv \boxed{t_2} \text{---} A$$

$$\frac{\Gamma_1 \vdash t_1 : \mathbf{1} \dashv \Gamma_2 \quad \Gamma_2 \vdash t_2 : A \dashv \Gamma_3}{\Gamma_1 \vdash (\text{let } * = t_1 \text{ in } t_2) : A \dashv \Gamma_3}$$

$$\Gamma_1 \equiv \boxed{t_1} \xrightarrow{\mathbf{1}} \quad \Gamma_2 \equiv \boxed{t_2} \text{---} A \text{---} \Gamma_3$$

sum

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash \text{inl}_{A \oplus B}(t) : A \oplus B}$$

$$\Gamma \equiv \boxed{t} \xrightarrow{A} \boxed{l_1} \text{---} A + B$$

$$\frac{\Gamma_1 \vdash t : A \dashv \Gamma_2}{\Gamma_1 \vdash \text{inl}_{A \oplus B}(t) : A \oplus B \dashv \Gamma_2}$$

$$\Gamma_1 \equiv \boxed{t} \xrightarrow{A} \boxed{l_1} \text{---} \Gamma_2 \text{---} A + B$$

$$\frac{\Gamma \vdash t : B}{\Gamma \vdash \text{inr}_{A \oplus B}(t) : A \oplus B}$$

$$\Gamma \equiv \boxed{t} \xrightarrow{B} \boxed{l_2} \text{---} A + B$$

$$\frac{\Gamma_1 \vdash t : B \dashv \Gamma_2}{\Gamma_1 \vdash \text{inr}_{A \oplus B}(t) : A \oplus B \dashv \Gamma_2}$$

$$\Gamma_1 \equiv \boxed{t} \xrightarrow{B} \boxed{l_2} \text{---} \Gamma_2 \text{---} A + B$$

$$\frac{\Gamma_1 \vdash t_1 : A \oplus B \quad \Gamma_2, x : A \vdash t_2 : C \quad \Gamma_2, y : B \vdash t_3 : C}{\Gamma_1, \Gamma_2 \vdash (\text{case } t_1 \text{ of } \text{inl}(x) \rightarrow t_2 \mid \text{inr}(y) \rightarrow t_3) : C}$$

$$\Gamma_1 \equiv \boxed{t_1} \xrightarrow{A+B} \boxed{t} \text{---} C \quad \text{where} \quad \begin{aligned} \Gamma_2 \equiv \boxed{t_1} \xrightarrow{A} \boxed{t} \text{---} C &= \Gamma_2 \equiv \boxed{t_2} \text{---} C \\ \Gamma_2 \equiv \boxed{t_2} \xrightarrow{B} \boxed{t} \text{---} C &= \Gamma_2 \equiv \boxed{t_3} \text{---} C \end{aligned} \quad \text{and}$$

$$\frac{\Gamma_1 \vdash t_1 : A \oplus B \dashv \Gamma_2 \quad \Gamma_2, x : A \vdash t_2 : C \dashv \Gamma_3, x^\dagger \quad \Gamma_2, y : B \vdash t_3 : C \dashv \Gamma_3, y^\dagger}{\Gamma_1 \vdash (\text{case } t_1 \text{ of } \text{inl}(x) \rightarrow t_2 \mid \text{inr}(y) \rightarrow t_3) : C \dashv \Gamma_3}$$

$$\begin{array}{c}
 \Gamma_1 = \boxed{t_1} = \Gamma_2 \quad \boxed{t} = \Gamma_3 \\
 A + B \quad \quad \quad C
 \end{array}
 \quad \text{where} \quad
 \begin{array}{c}
 \Gamma_2 = \boxed{t} = \Gamma_3 \\
 A \quad \boxed{t_1} \quad \quad \quad C
 \end{array}
 =
 \begin{array}{c}
 \Gamma_2 = \boxed{t_2} = \Gamma_3 \\
 A \quad \quad \quad C
 \end{array}
 \quad \text{and} \quad
 \begin{array}{c}
 \Gamma_2 = \boxed{t} = \Gamma_3 \\
 B \quad \boxed{t_2} \quad \quad \quad C
 \end{array}
 =
 \begin{array}{c}
 \Gamma_2 = \boxed{t_3} = \Gamma_3 \\
 B \quad \quad \quad C
 \end{array}$$

These definitions by universal property make use of the fact that $\Gamma_2 \otimes (A + B)$ is the coproduct of $\Gamma_2 \otimes A$ and $\Gamma_2 \otimes B$. This is true because $\otimes$ is a left adjoint and therefore commutes with coproducts.

false

$$\frac{\Gamma \vdash t : \mathbf{0}}{\Gamma \vdash \text{false}_A(t) : A}$$

$$\Gamma = \boxed{t} \xrightarrow{0} \boxed{?_A} \text{---} A$$

$$\frac{\Gamma_1 \vdash t : \mathbf{0} \dashv \Gamma_2}{\Gamma_1 \vdash \text{false}_A(t) : A \dashv \Gamma_2}$$

$$\Gamma_1 = \boxed{t} = \Gamma_2 \xrightarrow{0} \boxed{?_A} \text{---} A$$

additive pair

$$\frac{\Gamma \vdash t_1 : A \quad \Gamma \vdash t_2 : B}{\Gamma \vdash \langle t_1, t_2 \rangle : A \& B}$$

t such that

$$\begin{array}{c}
 \Gamma = \boxed{t} \xrightarrow{A \times B} \boxed{\pi_1} \text{---} A = \Gamma = \boxed{t_1} \text{---} A \quad \text{and} \\
 \Gamma = \boxed{t} \xrightarrow{A \times B} \boxed{\pi_2} \text{---} B = \Gamma = \boxed{t_2} \text{---} B
 \end{array}$$

$$\frac{\Gamma_2 \vdash t_1 : A \dashv \quad \Gamma_2 \vdash t_2 : B \dashv}{\Gamma_1, \Gamma_2 \vdash \langle t_1, t_2 \rangle : A \& B \dashv \Gamma_1}$$

$$\begin{array}{c}
 \Gamma_1 = \boxed{t} = \Gamma_1 \\
 \Gamma_2 = \boxed{t} \text{---} A \times B
 \end{array}$$

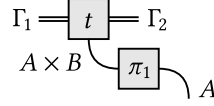
such that

$$\begin{array}{c}
 \Gamma_2 = \boxed{t} \xrightarrow{A \times B} \boxed{\pi_1} \text{---} A = \Gamma_2 = \boxed{t_1} \text{---} A \quad \text{and} \\
 \Gamma_2 = \boxed{t} \xrightarrow{A \times B} \boxed{\pi_2} \text{---} B = \Gamma_2 = \boxed{t_2} \text{---} B
 \end{array}$$

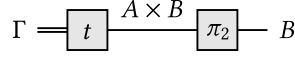
$$\frac{\Gamma \vdash t : A \& B}{\Gamma \vdash \text{fst}(t) : A}$$

$$\Gamma = \boxed{t} \xrightarrow{A \times B} \boxed{\pi_1} \text{---} A$$

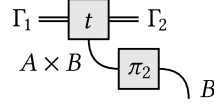
$$\frac{\Gamma_1 \vdash t : A \& B \vdash \Gamma_2}{\Gamma_1 \vdash \text{fst}(t) : A \vdash \Gamma_2}$$



$$\frac{\Gamma \vdash t : A \& B}{\Gamma \vdash \text{snd}(t) : B}$$

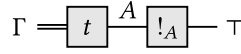


$$\frac{\Gamma_1 \vdash t : A \& B \vdash \Gamma_2}{\Gamma_1 \vdash \text{snd}(t) : B \vdash \Gamma_2}$$

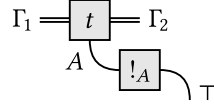


true

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash \text{true}(t) : \top}$$

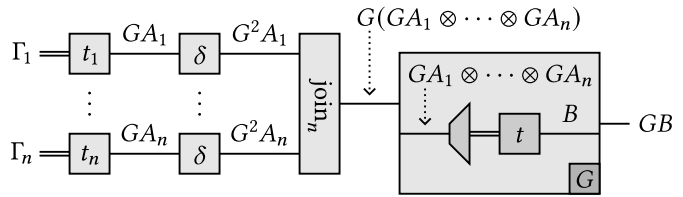


$$\frac{\Gamma_1 \vdash t : A \vdash \Gamma_2}{\Gamma_1 \vdash \text{true}(t) : \top \vdash \Gamma_2}$$

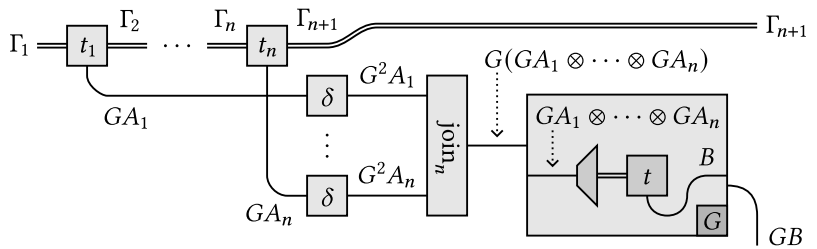


exponential types

$$\frac{\Gamma_1 \vdash t_1 : !A_1 \quad \dots \quad \Gamma_n \vdash t_n : !A_n \quad x_1 : !A_1, x_n : !A_n \vdash t : B}{\Gamma_1, \dots, \Gamma_n \vdash \text{promote } t_1, \dots, t_n \text{ for } x_1, \dots, x_n \text{ in } t : !B}$$



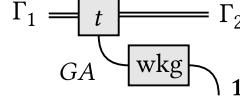
$$\frac{\Gamma_1 \vdash t_1 : !A_1 \vdash \Gamma_2 \quad \dots \quad \Gamma_n \vdash t_n : !A_n \vdash \Gamma_{n+1} \quad x_1 : !A_1, \dots, x_n : !A_n \vdash t : B \vdash}{\Gamma_1 \vdash \text{promote } t_1, \dots, t_n \text{ for } x_1, \dots, x_n \text{ in } t : !B \vdash \Gamma_{n+1}}$$



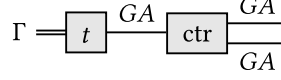
$$\frac{\Gamma \vdash t : !A}{\Gamma \vdash \text{discard } t}$$



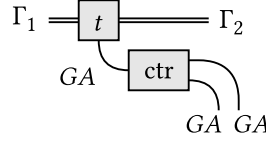
$$\frac{\Gamma_1 \vdash t_1 : !A \vdash \Gamma_2}{\Gamma_1 \vdash \text{discard } t : 1 \vdash \Gamma_2}$$



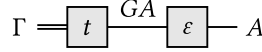
$$\frac{\Gamma \vdash t : !A}{\Gamma \vdash \text{copy } t : !A \otimes !A}$$



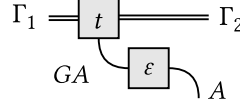
$$\frac{\Gamma_1 \vdash t : !A \vdash \Gamma_2}{\Gamma_1 \vdash \text{copy } t : !A \otimes !A \vdash \Gamma_2}$$



$$\frac{\Gamma \vdash t : !A}{\Gamma \vdash \text{derelict } t : A}$$



$$\frac{\Gamma_1 \vdash t : !A \vdash \Gamma_2}{\Gamma_1 \vdash \text{derelict } t : A \vdash \Gamma_2}$$



We may then prove the following lemma:

LEMMA E.1 (SEMANTICS OF FRAME RULE). *Let $\Gamma_1 \vdash t : A \vdash \Gamma_2$ be a linear term, using the syntax of λ_L . Then, writing $\Gamma_1, \Gamma : t/\Gamma \vdash \Gamma_2, \Gamma$ for the application of the frame rule to this term, we have the semantic equality:*

$$\begin{array}{c} \Gamma_1 \\ \hline \boxed{t/\Gamma} \\ \hline \Gamma \end{array} \vdash \Gamma_2 = \begin{array}{c} \Gamma_1 \\ \hline \boxed{t} \\ \hline \Gamma \end{array} \vdash \Gamma_2$$

PROOF. Straightforward rule induction. □

Using the simplifications described above, the correctness theorem becomes the following:

THEOREM E.2 (CORRECTNESS). *Let $\Gamma \vdash t : A$ be a λ_L term. Writing $\bar{t} : \Gamma \multimap 1 \otimes A$ for its lens translation λ_L :*

$$\Gamma \vdash \boxed{\bar{t}} \vdash A = \Gamma \vdash \boxed{t} \vdash A$$

PROOF. Straightforward rule induction. □

THEOREM E.3 (GENERALITY). *Let Γ_1, Γ_2 be λ_F contexts containing the same variables, and let A, B be λ_F types. Suppose that there is a closed λ_L lens of the following type:*

$$\vdash t : (\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow (A^\circ, B^\circ)$$

Then there is a lens term u in λ_F with type:

$$\Gamma_1 \vdash u : A/B \dashv \Gamma_2$$

such that $t \approx \bar{u}$. Moreover, if $B = \mathbf{1}$, there is a linear term t' with type:

$$\Gamma_1 \vdash t' : A \dashv \Gamma_2$$

such that $t \approx \bar{t}'$.

PROOF. This theorem does not admit a direct string-diagrammatic presentation as described above, since it concerns lens terms with non- $\mathbf{1}$ final type. For this proof, we make the following abbreviations:

$$\begin{aligned} (\circ) &: (A \multimap B) \times (S \multimap A) \rightarrow (S \multimap B) \\ g \circ f &= \lambda s : S. g (f s) \\ (\hat{\circ}) &: [(A, A') \leftrightarrow (B, B')] \times [(S, S') \leftrightarrow (A, A')] \rightarrow [(S, S') \leftrightarrow (B, B')] \\ g \hat{\circ} f &= \lambda s : S. \text{let } (a, r_a) = f s \text{ in let } (b, r_b) = g a \text{ in } (b, r_a \circ r_b) \\ (\gg) &: [(S, S') \leftrightarrow (A, \mathbf{1})] \times [(S' \otimes A, S'') \leftrightarrow (B, B')] \rightarrow [(S, S'') \leftrightarrow (B, B')] \\ f \gg g &= \lambda s : S. \text{let } (a, r_a) = f s \text{ in } g (r_a *, a) \\ \text{id} &: S \multimap S \\ \text{id} &= \lambda s : S. s \\ \widehat{\text{left}} &: (A \multimap B) \rightarrow [(A, C) \leftrightarrow (B, C)] \\ \widehat{\text{left}} &= \lambda f : (A \multimap B). \lambda a : A. (f a, \text{id}) \end{aligned}$$

Let

$$\vdash t : (\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow (A^\circ, B^\circ)$$

By the correctness theorem, t is a linear term with the following λ_F type:

$$\vdash t : (\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow (A^\circ, B^\circ) \dashv$$

and

$$\models \bar{t} \approx (\lambda * : \mathbf{1}. (t, \lambda x : \mathbf{1}. x)) : (\mathbf{1}, \mathbf{1}) \leftrightarrow ((\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow (A^\circ, B^\circ), \mathbf{1})$$

Let $x_1 : A_1, \dots, x_n : A_n$ be the names and types of the live variables in Γ_1 , and let $y_1 : B_1, \dots, y_m : B_m$ be the corresponding names and types in Γ_2 . Define u by:

$$u := (\text{let } (a, r) = t (*, x_1, \dots, x_n) \text{ in lens } b \leftarrow a \text{ in } (*, y_1, \dots, y_m) := r b)$$

Then

$$\Gamma_1 \vdash u : A/B \dashv \Gamma_2$$

We claim that $\bar{u} \approx t$. First,

$$\Gamma_1 \vdash t (*, x_1, \dots, x_n) : A^\circ \otimes (B^\circ \multimap \Gamma_2^\circ) \dashv \Gamma_1^\dagger$$

has translation:

$$\begin{aligned} \overline{t (*, x_1, \dots, x_n)} &\approx \bar{t} \gg (\lambda (\gamma, h) : (\Gamma_1^\circ \otimes ((\Gamma_1^\circ, \Gamma_2^\circ) \leftrightarrow (A^\circ, B^\circ))). (\widehat{\text{left}} h \hat{\circ} (*, x_1, \dots, x_n)) \gamma) \\ &\approx \bar{t} \gg (\lambda (\gamma, h). \text{let } (a, r_a) = \overline{(*, x_1, \dots, x_n)} \gamma \text{ in let } (b, r_b) = \widehat{\text{left}} h a \text{ in } (b, r_a \circ r_b)) \\ &\approx \bar{t} \gg (\lambda (\gamma, h). \text{let } (a, r_a) = (\gamma, \lambda x : \mathbf{1}. x) \text{ in let } (b, r_b) = \widehat{\text{left}} h a \text{ in } (b, r_a \circ r_b)) \\ &\approx \bar{t} \gg (\lambda (\gamma, h). \text{let } (b, r_b) = \widehat{\text{left}} h \gamma \text{ in } (b, r_b)) \\ &\approx \bar{t} \gg (\lambda (\gamma, h). \text{let } (b, r_b) = (h \gamma, \lambda x : \mathbf{1}. x) \text{ in } (b, r_b)) \end{aligned}$$

$$\begin{aligned}
&\approx \bar{t} \gg (\lambda(\gamma, h). (h \gamma, \lambda x : 1. x)) \\
&\approx (\lambda\gamma : \Gamma_1^\circ. (t, \lambda* : 1. \gamma)) \gg (\lambda(\gamma, h). (h \gamma, \lambda x : 1. x)) \\
&\approx \lambda\gamma : \Gamma_1^\circ. \text{let } (a, r_a) = (t, \lambda* : 1. \gamma) \text{ in } (a (r_a *), \lambda x : 1. x) \\
&\approx \lambda\gamma : \Gamma_1^\circ. (t \gamma, \lambda x : 1. x)
\end{aligned}$$

And

$$\Gamma_1^\dagger, a : A^\circ, r : (B^\circ \multimap \Gamma_2^\circ) \vdash (\text{lens } b \leftarrow a \text{ in } (*, y_1, \dots, y_m) := r \ b) \dashv \Gamma_2$$

has translation:

$$\begin{aligned}
&\overline{(\text{lens } b \leftarrow a \text{ in } (*, y_1, \dots, y_m) := r \ b)} \\
&\approx \lambda\gamma : ((1 \otimes A^\circ) \otimes (B^\circ \multimap \Gamma_2^\circ)). \text{let } (c, r_c) = \bar{a} \ \gamma \text{ in} \\
&\quad (c, \lambda b : B^\circ. \text{let } (*, r_g) = \overline{((*, y_1, \dots, y_m) := r \ b)} (r_c *, b) \text{ in } r_g *) \\
&\approx \lambda((*, a), r_a). \text{let } (c, r_c) = (a, (*, \lambda* : 1. (*, r_a))) \text{ in} \\
&\quad (c, \lambda b : B^\circ. \text{let } (*, r_g) = \overline{((*, y_1, \dots, y_m) := r \ b)} (r_c *, b) \text{ in } r_g *) \\
&\approx \lambda((*, a), r_a). (a, \lambda b : B^\circ. \text{let } (*, r_g) = \overline{((*, y_1, \dots, y_m) := r \ b)} ((*, r_a), b) \text{ in } r_g *) \\
&\approx \lambda((*, a), r_a). (a, \lambda b : B^\circ. r_a \ b) \\
&\approx \lambda((*, a), r_a). (a, r_a)
\end{aligned}$$

Combining, we obtain:

$$\begin{aligned}
\bar{u} &\approx \overline{(\text{let } (a, r) = t \ (*, x_1, \dots, x_n) \text{ in } \text{lens } b \leftarrow a \text{ in } (*, y_1, \dots, y_m) := r \ b)} \\
&\approx \overline{t \ (*, x_1, \dots, x_n)} \gg (\lambda(\gamma, (a, b)). \overline{(\text{lens } b \leftarrow a \text{ in } (*, y_1, \dots, y_m) := r \ b)} ((\gamma, a), b)) \\
&\approx (\lambda\gamma : \Gamma_1^\circ. (t \ \gamma, \lambda x : 1. x)) \gg (\lambda(\gamma, (a, b)). (\lambda(*, a), r_a). (a, r_a)) ((\gamma, a), b)) \\
&\approx (\lambda\gamma : \Gamma_1^\circ. (t \ \gamma, \lambda x : 1. x)) \gg (\lambda(*, (a, r_a)). (a, r_a)) \\
&\approx \lambda\gamma. \text{let } (c, r_c) = (t \ \gamma, \lambda x : 1. x) \text{ in } \text{let } (*, (a, r_a)) = (r_c *, c) \text{ in } (a, r_a) \\
&\approx \lambda\gamma. \text{let } (*, (a, r_a)) = (*, t \ \gamma) \text{ in } (a, r_a) \\
&\approx \lambda\gamma. \text{let } (a, r_a) = t \ \gamma \text{ in } (a, r_a) \\
&\approx \lambda\gamma. t \ \gamma \\
&\approx t
\end{aligned}$$

as required. A similar calculation can be used to show that the following linear term has translation logically equivalent to t when $B = 1$:

$$\Gamma_1 \vdash (\text{let } (a, r) = t \ (*, x_1, \dots, x_n) \text{ in } \text{let } * = ((*, y_1, \dots, y_m) := r \ *) \text{ in } a) \dashv \Gamma_2 \quad \square$$

References

- [1] Samson Abramsky. 1993. Computational Interpretations of Linear Logic. *Theoretical Computer Science* 111 (1993), 3–57. doi:10.1016/0304-3975(93)90181-R
- [2] G.M. Bierman. 1994. *On Intuitionistic Linear Logic*. Technical Report UCAM-CL-TR-346. University of Cambridge, Computer Laboratory. doi:10.48456/tr-346
- [3] Dan Ghica and Fabio Zanasi. 2024. String Diagrams for λ -calculi and Functional Computation. arXiv:2305.18945 [cs.LO] https://arxiv.org/abs/2305.18945
- [4] Ralf Hinze and Dan Marsden. 2023. *Introducing String Diagrams*. Cambridge University Press, Cambridge, England.
- [5] Saunders Mac Lane. 1978. *Categories for the Working Mathematician*. Graduate Texts in Mathematics, Vol. 5. Springer New York. doi:10.1007/978-1-4757-4721-8
- [6] Paul-André Mellies. 2009. Categorical semantics of linear logic. *Panoramas et synthèses* 27 (2009), 15–215.

- [7] Paul Wilson, Dan Ghica, and Fabio Zanasi. 2024. String Diagrams for Strictification and Coherence. *Logical Methods in Computer Science* Volume 20, Issue 4 (Oct. 2024), 13982. [doi:10.46298/lmcs-20\(4:8\)2024](https://doi.org/10.46298/lmcs-20(4:8)2024)