Linear Logic

Lecture Notes for CIS 670

Steve Zdancewic
University of Pennsylvania

February 25, 2014

Contents

Chapter 1

Overview

1.1 Introduction

- Linear logic was introduced in 1987 by Jean-Yves Girard [62].
  - Spawned hundreds of subsequent papers, with many(!) variants (substructural logics)
  - Very influential paper: cited $>$ 4250 times according to Google Scholar.

  Why?

  - Linear logic is a generalization of "ordinary" logic in such a way that it becomes "resource conscious".
    * Key idea: a resource is a "hypothesis" that can be used only once (hence it is used up)
    * Weakening and Contraction rules of ordinary propositional logic
    * Also possible to ephasize the "communication" behavior: parts of the proof that can "interact"
    * Duality is central: DeMorgan's laws highlight many of the symmetries in logic
  - Linear logic is constructive and low level:
    * meaning that it has computational content and therefore has connections to programming languages
  - Fully classical propositional logic *and* intuitionistic logic are both encodeable in Linear Logic
  - Even in the intuitionistic fragment there are both call-by-value and call-by-name translations of lambda calculus into the logic
  - Connections to parallel computation (even emphasized by Girard)

  Many Uses, either directly, or inspirationally:

In Logic and Proof theory

- Proof theory: cut elimination, consistency
- Notion of duality, polarity, decomposition of (say) intuitionistic propositional logic.
- Proof terms: Proof Nets, μ calculus, linear lambda calculus, π calculus
- LL, LU, ILL, DILL, JILL, LNL, (LPC: Jennifer)

Semantics

- Coherence Spaces, Chu Spaces, Game Semantics
- Categorical Models:

Applications

- Manual memory management [84, 7, 69]
- Control of pointer aliasing [51, 69, 152, 132]
- Heap separation properties [118]
- Referential transparency (purity) [133, 33]
- Resource handles and capabilities [77, 37, 39]
- State-dependent program analysis (typestates) [46, 51, 50]
- Safe concurrent communication (session types) [72, 124, 130]
- Security policy enforcement [144, 123]
- Program optimization [146]
- Differential Privacy [117, 57]
- Implicit Computational Complexity [56]
- Concurrency: [94, 96]

TODO: Pottier's recent work on state. TODO: Wadler, Pfenning's recent work on session types

Chapter 2

Intuitionistic Linear Logic

These notes follow the judgmental presentation of intuitionistic linear logic in Chang, *et al.*'s paper [36]. They were also informed by Pfenning's lecture notes on linear logic. For more discussion about this approach to logic, see the notes on Martin-Löf's Sienna Lectures [93] and also Davies and Pfenning's paper on modal logic [111].

In the "judgmental" approach to defining logics, one makes a distinction between what it means for a piece of syntax to be a proposition and whether a proposition is true. Going even further, there may be different notions of "truth," each corresponding to a different means of justifying a particular proposition. Such modes of "truth" are *judgments* about propositions. The meaning of a judgment is determined by what is considered to count as "evidence" for it.

For example, consider the syntax "$1+1=0$". The judgment "$1+1=0$" *is a proposition* asserts that the syntax is a legal subject for manipulation using the logic. The designer of the logic decides what pieces of syntax are legal propositions by determining what counts as "evidence" that a proposition is legal. Not all syntax may be judged to be a legal proposition, for example, one might wish to disallow "$= +\mathrm{ELEPHANT} = 0$."

Different logics make different choices about what it means for a piece of syntax to be a legal propositions. For example, *propositional* logic usually assumes the existence of some (finite or countably infinite) set of *atomic propositions* $\{a_1, a_2, \ldots\}$ and then defines a *grammar* of propositions along the lines of:

$$p, q \quad ::= \quad a_i \mid p \wedge q \mid p \vee q \mid \neg p \mid p \Rightarrow q$$

For convenience, we abbreviate the judgment "*A is a proposition*" by A **prop**. A grammar like the one above is short hand for a collection of *inference rules* that collectively determine how to provide evidence for the judgment "*A is a proposition*." In this case, we would have the rules shown in Figure 2.1.

This approach means that a piece of syntax like "$1+1=0$" may (or may not) constitute a legal proposition. This could be indicated judgmentally by agreeing on what counts as evidence for the judgment ("$1+1=4$" is a proposition), or, more tersely ("$1+1=4$" **prop**).

$$\frac{a \in \{a_1, a_2, \ldots\}}{a\ \mathbf{prop}} \qquad \frac{p\ \mathbf{prop} \qquad q\ \mathbf{prop}}{p \wedge q\ \mathbf{prop}} \qquad \frac{p\ \mathbf{prop} \qquad q\ \mathbf{prop}}{p \vee q\ \mathbf{prop}}$$

$$\frac{p\ \mathbf{prop}}{\neg p\ \mathbf{prop}} \qquad \frac{p\ \mathbf{prop} \qquad q\ \mathbf{prop}}{p \Rightarrow q\ \mathbf{prop}}$$

Figure 2.1: Propositions of propositional logic.

The role of a "logic" is to characterize how judgments interact in a general way, independent of the details of how the evidence for those judgments is generated.

In *intuitionistic* (constructive) propositional logic, one is typically concerned with two or three different kinds of judgments, for example:

A **prop**	The syntax "A" is a proposition
A **true**	The proposition A is true (contingently provable)
A **valid**	The proposition A is valid (provable always)

In *classical* logic, besides A **true**, one might also consider the judgment A **false**, which asserts that the proposition A is false. The structure of classical logic would then determine how proofs, which give evidence for A **true**, interact with refutations, which give evidence for A **false**.

2.1 The Multiplicative Fragment

Judgment of the form A **lin** meaning "the proposition A is linearly true."

We will eventually add another judgment A **per** with the intended meaning that "A is persistently true" or "valid."

Following Martin-Löf, we must give the judgment A **lin** meaning by deciding what counts as evidence for it. Because we are thinking of linear propositions as some kind of resource (for now), it seems obvious that evidence for a resource would be the resource itself. If we think in terms of the money example, then the proposition q might mean "Steve has a quarter" and evidence of that proposition would be my posession of the quarter itself, which I could display as proof of the proposition.

Next we generalize to *hypothetical* judgments, which make some assumptions about the existence of evidence for judgments. In the context:

$$A_1\ \mathbf{lin}, A_2\ \mathbf{lin}, A_3\ \mathbf{lin} \vdash B\ \mathbf{lin}$$

We decide on what counts as evidence for a hypothetical judgment by specifying axioms and inference rules. In the case of linear logic, we want to interpret evidence for such a hypothetical judgment as a resource-efficient *plan* for producing evidence for the judgment B **lin** given evidence for the assumed hypotheses.

Plans of this kind should be resource-conscious in two senses:

- The plan should be "efficient." In particular, the plan should only mention *relevant* resources. That means that each hypothetical resource should be used at least once in the plan.
- The plan should be respect the our interpretation of A **lin** as a kind of resource. There are different ways one might possibly do this, but a very simple, and natural one is to say that the distinguishing characteristic of a resource is that it can't (easily!) be copied. Therefore, we should ensure that assuming A **lin** is not (necessarily) the same as assuming two instances of it, as in A **lin**, A **lin**.

These considerations will have to be taken in to account as we design the logic. The inference rules will thus have to respect our intended semantics of "plans" as evidence for hypothetical linear judgments.

$$\frac{}{A\ \mathbf{lin} \vdash A\ \mathbf{lin}}\ \textsc{Hyp}$$

An instance of this rule is:

$$q\ \mathbf{lin} \vdash q\ \mathbf{lin}$$

We could interpret it as: "Assuming that Steve has a quarter there is a trivial plan to demonstrate that Steve has a quarter."

A bad alternative for this hypothesis rule is:

$$\frac{}{\Delta, A\ \mathbf{lin} \vdash A\ \mathbf{lin}}\ \textsc{BadHyp}$$

It doesn't respect the "efficiency" criterion of our intended interpretation. This plan discards all of the resources in Δ.

Our interpretation of judgments as plans should validate the following substitution principle:

Principle 2.1 (Substitution)**.** *If* $\Delta_1 \vdash A\ \mathbf{lin}$ *and* $\Delta_2, A\ \mathbf{lin} \vdash B\ \mathbf{lin}$ *then* $\Delta_1, \Delta_2 \vdash B\ \mathbf{lin}$.

- The substitution principle is not a rule of the logic, but instead a structural invariant that justifies the use of a judgment like A **lin** as a hypothesis.
- Note how this takes into account the non-duplicability of resources. In particular, when we write Δ_1, Δ_2 we should interpret this as requiring the combination of resources hypothesized in both Δ_1 and Δ_2.

The following plausible substitution principle *does not* respect the intended interpretation of linear hypotheses.

Principle 2.2 (Bad Substitution)**.** *If* $\Delta \vdash A\ \mathbf{lin}$ *and* $\Delta, A\ \mathbf{lin} \vdash B\ \mathbf{lin}$ *then* $\Delta \vdash B\ \mathbf{lin}$.

To extend the logic with more structure, we consider different possible ways that generic, or hypothetical resources might be manipulated: as opposed to rules for working with particular resources (such as making change via coins), such rules describe plans for manipulating *any* resource. Such "general" plans correspond to logical connectives, and their meaning is justified (as suggested by Martin-Löf) by inference rules. In natural-deduction style proofs, the semantics can be given by the *introduction* rules, which say how to *produce* evidence from hypotheses, and *elimination* rules, which say how to *consume* such evidence.

When developing the inference rules for our logic, we can assess their correctness by ensuring that all of the rules we design are "in harmony" with the substitution principle. This amounts to showing two properties for each connective of the logic:

- *Local soundness:* This ensures that the elimination rules are not too strong, in the sense that all of the information produced by eliminating a judgment is available during its construction. (These will correspond to β-reductions justified by substitution.)
- *Local expansion:* This ensures that the elimination rules are not too weak, in the sense that by eliminating a judgment produces information sufficient to reconstitute it. (These will correspond to η-expansions.)

Note: *Because, for the moment, all of the judgments that we manipulate in the logic are of the form A **lin**, we write the hypothetical judgment*

$$\Delta_1, A_1\ \mathbf{lin}, A_2\ \mathbf{lin}, A_3\ \mathbf{lin}, \Delta_2 \vdash B\ \mathbf{lin}$$

as simply

$$\Delta_1, A_1, A_2, A_3, \Delta_2 \vdash B$$

This abbreviated form makes it less noisy to write inference rules, but we should keep in mind that this linear judgment is distinct from other judgments.

2.1.1 Multiplicative Conjunction

Suppose we have two resources A and B, what constitutes evidence that both are true? We write $(A \otimes B)$ for the *simultaneous* conjunction of A and B (also called *multiplicative* product, also called *tensor* product).

What constitutes a (general purpose, efficient) plan to produce an A and a B assuming some hypothetical resources?

$$\frac{\Delta_1 \vdash A_1 \quad \Delta_2 \vdash A_1}{\Delta_1, \Delta_2 \vdash A_1 \otimes A_2}\ \otimes\mathrm{I}$$

This rule (read top to bottom) says that given two plans, one building A_1 from Δ_1 and one building A_2 from Δ_2 we can build a composite plan that needs the resources of *both* constituent plans.

The corresponding elimination rule says that given a plan to produce $(A_1 \otimes A_2)$ we can assume them (as hypothetical judgments) simultaneously in some plan to construct B.

$$\dfrac{\Delta_1 \vdash A_1 \otimes A_2 \quad \Delta_2, A_1, A_2 \vdash B}{\Delta_1, \Delta_2 \vdash B}\ \otimes E$$

Local soundness:

$$\dfrac{\dfrac{\dfrac{\mathcal{D}_1}{\Delta_1 \vdash A_1} \qquad \dfrac{\mathcal{D}_2}{\Delta_2 \vdash A_2}}{\Delta_1, \Delta_2 \vdash A_1 \otimes A_2}\,\otimes I \qquad \dfrac{\mathcal{E}}{\Delta_3, A_1, A_2 \vdash B}}{\Delta_1, \Delta_2, \Delta_3 \vdash B}\,\otimes E \qquad \Longrightarrow^R \qquad \dfrac{\mathcal{E}'}{\Delta_1, \Delta_2, \Delta_3 \vdash B}$$

Local expansion:

$$\dfrac{\mathcal{D}}{\Delta \vdash A_1 \otimes A_2} \qquad \Longrightarrow^E \qquad \dfrac{\dfrac{\mathcal{D}}{\Delta \vdash A_1 \otimes A_2} \qquad \dfrac{\dfrac{}{A_1 \vdash A_1}\,\text{HYP} \quad \dfrac{}{A_2 \vdash A_2}\,\text{HYP}}{A_1, A_2 \vdash A_1 \otimes A_2}}{\Delta \vdash A_1 \otimes A_2}\,\otimes E$$

2.1.2 Multiplicative Conjunction Unit

The unit of multiplicative conjunction is the "trivial" resource 1. It has rules:

$$\dfrac{}{\cdot \vdash 1}\ 1I \qquad \dfrac{\Delta_1 \vdash 1 \quad \Delta_2 \vdash A}{\Delta_1, \Delta_2 \vdash A}\ 1E$$

Local soundness:

$$\dfrac{1I\ \dfrac{\mathcal{D}}{\cdot \vdash 1} \qquad \dfrac{\mathcal{E}}{\Delta \vdash A}}{\Delta \vdash A}\,1E \qquad \Longrightarrow^R \qquad \dfrac{\mathcal{E}}{\Delta \vdash A}$$

Local expansion:

$$\dfrac{\mathcal{D}}{\Delta \vdash 1} \qquad \Longrightarrow^E \qquad 1E\ \dfrac{\dfrac{\mathcal{D}}{\Delta \vdash 1} \qquad \dfrac{}{\cdot \vdash 1}\,1I}{\Delta \vdash 1}$$

2.1.3 Linear Implication

$$\dfrac{\Delta, A \vdash B}{\Delta \vdash A \multimap B}\ \multimap I \qquad \dfrac{\Delta_1 \vdash A \multimap B \quad \Delta_2 \vdash A}{\Delta_1, \Delta_2 \vdash B}\ \multimap E$$

Local soundness:

$$\frac{}{A \vdash A}\ \textsc{Hyp}$$

$$\frac{\Delta_1 \vdash A_1 \quad \Delta_2 \vdash A_1}{\Delta_1, \Delta_2 \vdash A_1 \otimes A_2}\ \otimes\mathrm{I}$$

$$\frac{\Delta_1 \vdash A_1 \otimes A_2 \quad \Delta_2, A_1, A_2 \vdash B}{\Delta_1, \Delta_2 \vdash B}\ \otimes\mathrm{E}$$

$$\frac{}{\cdot \vdash 1}\ 1\mathrm{I}$$

$$\frac{\Delta_1 \vdash 1 \quad \Delta_2 \vdash A}{\Delta_1, \Delta_2 \vdash A}\ 1\mathrm{E}$$

$$\frac{\Delta, A \vdash B}{\Delta \vdash A \multimap B}\ \multimap\mathrm{I}$$

$$\frac{\Delta_1 \vdash A \multimap B \quad \Delta_2 \vdash A}{\Delta_1, \Delta_2 \vdash B}\ \multimap\mathrm{E}$$

Figure 2.2: The multiplicative fragment of intuitionistic linear logic.

$$\multimap\mathrm{E}\ \dfrac{\multimap\mathrm{I}\ \dfrac{\dfrac{\mathcal{D}}{\Delta_1, A \vdash B}}{\Delta_1 \vdash A \multimap B} \qquad \dfrac{\mathcal{E}}{\Delta_2 \vdash A}}{\Delta_1, \Delta_2 \vdash B} \quad \Longrightarrow^R \quad \dfrac{\mathcal{D}'}{\Delta_1, \Delta_2 \vdash B}$$

Local Completeness:

$$\dfrac{\mathcal{D}}{\Delta \vdash A \multimap B} \quad \Longrightarrow^E \quad \dfrac{\dfrac{\dfrac{\mathcal{D}}{\Delta \vdash A \multimap B} \qquad \dfrac{}{A \vdash A}\ \textsc{Hyp}}{\Delta, A \vdash B}\ \multimap\mathrm{E}}{\Delta \vdash A \multimap B}\ \multimap\mathrm{I}$$

2.1.4 Summary

The multiplicative fragment of intutionistic linear logic is summarized in Figure Figure 2.2.

Investigate the relationships among the connectives:

$$\begin{array}{rcl}
A_1 \otimes (A_2 \otimes A_3) & \vdash & (A_1 \otimes A_2) \otimes A_3 \\
(A_1 \otimes A_2) \otimes A_3 & \vdash & A_1 \otimes (A_2 \otimes A_3) \\
1 \otimes A & \vdash & A \\
A & \vdash & 1 \otimes A \\
A \otimes 1 & \vdash & A \\
A & \vdash & A \otimes 1 \\
(A_1 \otimes A_2) \multimap B & \vdash & A_1 \multimap A_2 \multimap B \\
A_1 \multimap A_2 \multimap B & \vdash & (A_1 \otimes A_2) \multimap B
\end{array}$$

We can add axioms to the multiplicative fragment of linear logic. Such axioms are not justified by the judgmental construction, but should rather be justified by some "external" evidence.

For example, returning to the coin scenario, we might consider adding axioms like:

$$\frac{}{\cdot \vdash q \multimap d \otimes d \otimes n}\ \textsc{Change1}$$

$$\frac{}{\cdot \vdash q \multimap n \otimes n \otimes n \otimes n \otimes n}\ \textsc{Change2}$$

Axioms are subtley different than hypotheses because axioms are not "resources" in the sense that the are consumed. If we wanted to model the ability to make change for some (fixed) number of quarters, it would be better to represent the situation as the derivation of some hypothetical judgment:

$$\frac{\mathcal{D}}{q \multimap d \otimes d \otimes n \vdash B}$$

2.2 The Additive Fragment

2.2.1 Additive Products

Suppose we have two plans for processing a linear resource. For example, suppose that we have:

$$\frac{\mathcal{D}_1}{q \vdash d \otimes d \otimes n} \qquad \frac{\mathcal{D}_2}{q \vdash q}$$

Given a resource of only one quarter, we have seen that (in general) it is not possible to combine those two plans to obtain:

$$\frac{\dfrac{\mathcal{D}_1}{\dots} \quad \dfrac{\mathcal{D}_2}{\dots}}{q \vdash (d \otimes d \otimes n) \otimes q}\ \textsc{Bogus}$$

Such a plan would need to somehow copy the resource q.

However, we can easily imagine creating a plan that, given a quarter q follows one of the two plans. The resulting resource is "fungible" in that it presents two options for how to use the it. The new kind of resource is written $A \& B$ and pronounced A *with* B. For example, we have using the examples above:

$$\dfrac{\dfrac{\mathcal{D}_1}{q \vdash d \otimes d \otimes n} \qquad \dfrac{\mathcal{D}_2}{q \vdash q}}{q \vdash (d \otimes d \otimes n) \& q}$$

The introduction and elimination rules for $\&$ are:

$$\dfrac{\Delta \vdash A \quad \Delta \vdash B}{\Delta \vdash A \& B}\ \&\mathrm{I} \qquad \dfrac{\Delta \vdash A \& B}{\Delta \vdash A}\ \&\mathrm{E1} \quad \dfrac{\Delta \vdash A \& B}{\Delta \vdash B}\ \&\mathrm{E2}$$

As usual, we can check whether these rules make sense by checking local soundness and local expansion. However, because there are two elimination rules for $\&$, there are two possible local soundness reductions.

Local Soundness 1:

$$\&\mathrm{E1}\ \dfrac{\&\mathrm{I}\ \dfrac{\dfrac{\mathcal{D}_1}{\Delta \vdash A} \qquad \dfrac{\mathcal{D}_2}{\Delta \vdash B}}{\Delta \vdash A \& B}}{\Delta \vdash A} \quad \Longrightarrow^R \quad \dfrac{\mathcal{D}_1}{\Delta \vdash A}$$

Local Soundness 2:

$$\&\mathrm{E1}\ \dfrac{\&\mathrm{I}\ \dfrac{\dfrac{\mathcal{D}_1}{\Delta \vdash A} \qquad \dfrac{\mathcal{D}_2}{\Delta \vdash B}}{\Delta \vdash A \& B}}{\Delta \vdash B} \quad \Longrightarrow^R \quad \dfrac{\mathcal{D}_2}{\Delta \vdash B}$$

Local Expansion:

$$\dfrac{\mathcal{D}}{\Delta \vdash A \& B} \quad \Longrightarrow^E \quad \&\mathrm{I}\ \dfrac{\&\mathrm{E1}\ \dfrac{\dfrac{\mathcal{D}}{\Delta \vdash A \& B}}{\Delta \vdash A} \qquad \&\mathrm{E2}\ \dfrac{\dfrac{\mathcal{D}}{\Delta \vdash A \& B}}{\Delta \vdash B}}{\Delta \vdash A \& B}$$

Note that the semantics of $A \& B$ as a hypothetical resource means that the plan that uses the resources as an input gets to decide whether to treat it as an A or a B. This means that the "consumer" of the resource $A \& B$ gets to determine which way to go.

2.2.2 Additive Product Unit

The unit for $\&$ is a "0-ary" version of the product. This means that it uses its resources in 0 premises and there are no ways to eliminate it:

$$\frac{}{\Delta \vdash \top}\ \top\mathrm{I} \qquad\qquad \text{(no elimination form)}$$

However, we still have a reasonable notion of local soundness:

$$\frac{\mathcal{D}}{\Delta \vdash \top} \quad \Longrightarrow^{E} \quad \frac{}{\Delta \vdash \top}\ \top\mathrm{I}$$

An instance of this is:

$$\frac{}{\top \vdash \top}\ \textsc{Hyp} \quad \Longrightarrow^{E} \quad \frac{}{\top \vdash \top}\ \top\mathrm{I}$$

2.2.3 Additive Sums

The dual to "with" is a kind of resource that acts like a disjunction chosen by the provider of that resource, instead of the consumer. We call it the "additive sum" and write it as $A \oplus B$. It has two introduction rules and one elimination rule.

$$\frac{\Delta \vdash A}{\Delta \vdash A \oplus B}\ \oplus\mathrm{I1} \qquad \frac{\Delta \vdash B}{\Delta \vdash A \oplus B}\ \oplus\mathrm{I2} \qquad \frac{\Delta_1 \vdash A_1 \oplus A_2 \quad \Delta_2, A_1 \vdash B \quad \Delta_2, A_2 \vdash B}{\Delta_1, \Delta_2 \vdash B}\ \oplus\mathrm{E}$$

Local Soundness 1:

$$\frac{\oplus\mathrm{I1}\ \dfrac{\dfrac{\mathcal{D}}{\Delta_1 \vdash A_1}}{\Delta_1 \vdash A_1 \oplus A_2} \qquad \dfrac{\mathcal{E}_1}{\Delta_2, A_1 \vdash B} \qquad \dfrac{\mathcal{E}_2}{\Delta_2, A_2 \vdash B}}{\Delta_1, \Delta_2 \vdash B}\ \oplus\mathrm{E} \quad \Longrightarrow^{R} \quad \frac{\mathcal{E}_1'}{\Delta_1, \Delta_2 \vdash B}$$

Local Soundness 1:

$$\frac{\oplus\mathrm{I2}\ \dfrac{\dfrac{\mathcal{D}}{\Delta_1 \vdash A_2}}{\Delta_1 \vdash A_1 \oplus A_2} \qquad \dfrac{\mathcal{E}_1}{\Delta_2, A_1 \vdash B} \qquad \dfrac{\mathcal{E}_2}{\Delta_2, A_2 \vdash B}}{\Delta_1, \Delta_2 \vdash B}\ \oplus\mathrm{E} \quad \Longrightarrow^{R} \quad \frac{\mathcal{E}_2'}{\Delta_1, \Delta_2 \vdash B}$$

Local Expansion:

$$\frac{\mathcal{D}}{\Delta \vdash A \oplus B} \quad \Longrightarrow^{E} \quad \dfrac{\dfrac{\mathcal{D}}{\Delta \vdash A \oplus B} \qquad \dfrac{\overline{A \vdash A}\ \textsc{Hyp}}{\cdot, A \vdash A \oplus B}\ \oplus\text{I1} \qquad \dfrac{\overline{B \vdash B}\ \textsc{Hyp}}{\cdot, B \vdash A \oplus B}\ \oplus\text{I2}}{\Delta \vdash A \oplus B}\ \oplus\text{E}$$

2.2.4 Additive Sum Unit

This is a form of "False." The unit for $\oplus$ is a "0-ary" version of the sum. This means that its elimination rule requires 0 premises and there are no ways to introduce it:

$$\text{(no introduction form)} \qquad \frac{\Delta_1 \vdash 0}{\Delta_1, \Delta_2 \vdash A}\ 0\text{E}$$

There is still a reasonable notion of local expansion:

$$\frac{\mathcal{D}}{\Delta \vdash 0} \quad \Longrightarrow^{E} \quad \dfrac{\dfrac{\mathcal{D}}{\Delta \vdash 0}}{\Delta \vdash 0}\ 0\text{E}$$

2.3 Persistence and the Exponential Modality

So far, the fragment of intuitionistic linear logic that we have been working with is not expressive enough to encode ordinary propositional logic because there is no way that a hypothesis can be used multiple times (or even zero times). These properties of the multiplicative and additive fragments can be summarized by saying that the usual *weakening* and *contraction* rules of logic are not admissible.

Lemma 2.3 (Weakening **Wrong!**)**.** *If* $\Delta \vdash B$ **lin** *then* Δ, A **lin** $\vdash B$ **lin***.*

Lemma 2.4 (Contraction **Wrong!**)**.** *If* Δ, A **lin**, A **lin** $\vdash B$ **lin** *then* Δ, A **lin** $\vdash B$ **lin***.*

To recover the expressiveness of full propositional logic while retaining the interpretation of hypotheses as resources, we introduce a new kind of judgment, *persistence*, which indicates that a proposition is "persistently true." In contrast to the judgment A **lin**, A **per** means that the hypohesis A is not an exhaustible resource.

Again following Marin-Löf, we must decide what counts as evidence for the judgment A **per**. Besides axiomatic declarations of such persistent truths, it also seems reasonable to say that if we have a plan to construct a resource A from *no* other linear resources, then A is itself a persistent resource. We can always execute the plan to construct the A resource as many times as needed. This leads us to the following principle of persistence.

Principle 2.5 (Persistence)**.** *If* $\cdot \vdash A$ **lin** *then* A **per***.*

Pfenning calls this a *categorical* judgment. Persistence is the linear analogue to the usual notion of *validity* from propositional logic: a proposition is valid if it is true using no hypotheses.

We now need to reconsider the hypothetical judgments for linear propositions. In particular, it is now possible to have persistent hypotheses in addition to the linear hypotheses we've been using so far. Therefore we refine our hypothetical judgment to include a mix of linear and persistent hypotheses:

$$A_1 \ \mathbf{lin}, A_2 \ \mathbf{per}, A_3 \ \mathbf{per}, A_4 \ \mathbf{lin}, \ldots, A_n \ \mathbf{lin} \vdash B \ \mathbf{lin}$$

Because the order of hypotheses in the context doesn't matter, we can use the notationally more convenient (and by now standard) form of such hypothetical judgments that separates the persistent hypotheses from the linear ones. By convention we use Δ for the linear contexts and Γ for the persistent ones:

$$\begin{array}{rcl} \Delta & ::= & \cdot \mid A \ \mathbf{lin} \mid \Delta_1, \Delta_2 \\ \Gamma & ::= & \cdot \mid A \ \mathbf{per} \mid \Gamma_1, \Gamma_2 \end{array}$$

This means that the new form of the hypothetical judgments is:

$$\Gamma; \ \Delta \vdash A \ \mathbf{lin}$$

Note that the conclusion is still a judgment about the *linearity* of the proposition A. We could consider hypothetical judgments of the form $\Gamma; \ \Delta \vdash A \ \mathbf{per}$ that define the semantics of *persistence*. However, doing so will turn out to be unnecessary because we categorically defined persistence as $\cdot \vdash A \ \mathbf{lin}$; derivations about persistence are just a particular mode of derivations about linearity.

We do have to consider how the persistent hypotheses warrant new linear judgments. Observe that $A \ \mathbf{per}$ is just a mode of $A \ \mathbf{lin}$, we can use the following rule for persistent hypotheses:

$$\frac{}{\Gamma, A \ \mathbf{per}; \ \cdot \vdash A \ \mathbf{lin}} \ !\textsc{Hyp}$$

Note that it requires the linear context to be empty, but permits other persistent hypotheses to appear in Γ. The corresponding new formulation for linear hypotheses is:

$$\frac{}{\Gamma; \ A \ \mathbf{lin} \vdash A \ \mathbf{lin}} \ \textsc{Hyp}$$

It also permits persistent hypotheses to appear in Γ, but as usual, requires exactly the linear hypothesis being used. Together these rules will justify the weakening property (but only for the persistent context).

> ***Note:*** *What would be the consequences of adding the following rather than the* $!\textsc{Hyp}$ *rule above?*

$$\frac{\Gamma, A\ \mathbf{per};\ \Delta, A\ \mathbf{lin} \vdash B\ \mathbf{lin}}{\Gamma, A\ \mathbf{per};\ \Delta \vdash B\ \mathbf{lin}}$$

Why is the first rule preferable?

Together with the modifications to the hypothesis rules, we must reconsider the substitution principles. Now there are two kinds of substitution—one that applies to linear hypotheses and one that applies to persistent hypotheses. Again, the forms of the substitution principles capture the intended semantics:

Principle 2.6 (Substitution II)**.**

- *If* $\Gamma;\ \Delta_1 \vdash A\ \mathbf{lin}$ *and* $\Gamma;\ \Delta_2, A\ \mathbf{lin} \vdash B\ \mathbf{lin}$ *then* $\Gamma;\ \Delta_1, \Delta_2 \vdash B\ \mathbf{lin}$*.*
- *If* $\Gamma;\ \cdot \vdash A\ \mathbf{lin}$ *and* $\Gamma, A\ \mathbf{per};\ \Delta \vdash B\ \mathbf{lin}$ *then* $\Gamma;\ \Delta \vdash B\ \mathbf{lin}$*.*

We can also write down the weakening an contraction principles that we expect to hold of the resulting logic:

Principle 2.7 (Weakening)**.** *If* $\Gamma;\ \Delta \vdash B\ \mathbf{lin}$ *then* $\Gamma, A\ \mathbf{per};\ \Delta \vdash B\ \mathbf{lin}$*.*

Principle 2.8 (Contraction)**.** *If* $\Gamma, A\ \mathbf{per}, A\ \mathbf{per};\ \Delta \vdash B\ \mathbf{lin}$ *then* $\Gamma, A\ \mathbf{per};\ \Delta \vdash B\ \mathbf{lin}$*.*

We can *internalize* the persistence judgment as a modal operator !. The introduction rule makes the interpretation of the modality clear: the linear proposition $!A$ stands for a persistent judgment $A\ \mathbf{per}$:

$$\frac{\Gamma;\ \cdot \vdash A\ \mathbf{lin}}{\Gamma;\ \cdot \vdash\ !A\ \mathbf{lin}}\ !\mathrm{I}$$

The elimination rule says that a plan to create a persistent resource $!A$ justifies the use of A as a persistent hypothesis:

$$\frac{\Gamma;\ \Delta_1 \vdash\ !A\ \mathbf{lin} \qquad \Gamma, A\ \mathbf{per};\ \Delta_2 \vdash B}{\Gamma;\ \Delta_1, \Delta_2 \vdash B}\ !\mathrm{E}$$

As usual, we test our new substitution principles against these rules by checking for local soundness and local expansion:

Local Soundness:

$$\frac{\dfrac{\dfrac{\mathcal{D}}{\Gamma;\ \cdot \vdash A}}{\Gamma;\ \cdot \vdash\ !A}\ !\mathrm{I} \qquad \dfrac{\mathcal{E}}{\Gamma, A\ \mathbf{per};\ \Delta \vdash B}}{\Gamma;\ \Delta \vdash B}\ !\mathrm{E} \qquad \Longrightarrow^{R} \qquad \frac{\mathcal{E}'}{\Gamma;\ \Delta \vdash B}$$

Local Expansion:

$$\dfrac{\mathcal{D}}{\Gamma;\ \Delta \vdash\ !A} \quad \Longrightarrow^{E} \quad \dfrac{\dfrac{\mathcal{D}}{\Gamma;\ \Delta \vdash\ !A} \qquad \dfrac{\overline{\Gamma, A\ \mathbf{per};\ \cdot \vdash A}\ !\textsc{Hyp}}{\Gamma, A\ \mathbf{per};\ \cdot \vdash\ !A}\ !\mathrm{I}}{\Gamma;\ \Delta \vdash\ !A}\ !\mathrm{E}$$

As before, it is worthwhile to consider how the ! constructor interacts with the other connectives of the logic. In particular, we have that the following are all derivable in intuitionistic linear logic:

$$\begin{array}{rcl}
\cdot;\ !A & \vdash & !A \otimes !A \\
\cdot;\ !A & \vdash & 1 \\
\cdot;\ !A & \vdash & !!A \\
\cdot;\ !!A & \vdash & !A \\
\cdot;\ !(A \,\&\, B) & \vdash & !(A \otimes B) \\
\cdot;\ !A \otimes !B & \vdash & !(A \otimes B) \\
\cdot;\ 1 & \vdash & !\top \\
\cdot;\ !\top & \vdash & 1
\end{array}$$

On the other hand, the following are not possible to derive (though it will be easier to prove that this is the case once we look at the sequent calculus formulation of the logic):

$$\begin{array}{rcl}
\cdot;\ !(A \otimes B) & \nvdash & !(A \,\&\, B) \\
\cdot;\ !(A \otimes B) & \nvdash & !A \otimes !B
\end{array}$$

$$\frac{}{\Gamma;\ A\ \mathbf{lin} \vdash A\ \mathbf{lin}}\ \textsc{Hyp}$$

$$\frac{}{\Gamma, A\ \mathbf{per};\ \cdot \vdash A\ \mathbf{lin}}\ !\textsc{Hyp}$$

$$\frac{\Gamma;\ \Delta_1 \vdash 0}{\Gamma;\ \Delta_1, \Delta_2 \vdash A}\ 0\mathrm{E} \qquad \frac{}{\Gamma;\ \Delta \vdash \top}\ \top\mathrm{I}$$

$$\frac{\Gamma;\ \Delta \vdash A}{\Gamma;\ \Delta \vdash A \oplus B}\ \oplus\mathrm{I1} \qquad \frac{\Gamma;\ \Delta \vdash A \,\&\, B}{\Gamma;\ \Delta \vdash A}\ \&\mathrm{E1}$$

$$\frac{\Gamma;\ \Delta \vdash B}{\Gamma;\ \Delta \vdash A \oplus B}\ \oplus\mathrm{I2} \qquad \frac{\Gamma;\ \Delta \vdash A \,\&\, B}{\Gamma;\ \Delta \vdash B}\ \&\mathrm{E2}$$

$$\frac{\Gamma;\ \Delta_1 \vdash A_1 \oplus A_2 \quad \Gamma;\ \Delta_2, A_1 \vdash B \quad \Gamma;\ \Delta_2, A_2 \vdash B}{\Gamma;\ \Delta_1, \Delta_2 \vdash B}\ \oplus\mathrm{E} \qquad \frac{\Gamma;\ \Delta \vdash A \quad \Gamma;\ \Delta \vdash B}{\Gamma;\ \Delta \vdash A \,\&\, B}\ \&\mathrm{I}$$

$$\frac{}{\Gamma;\ \cdot \vdash 1}\ 1\mathrm{I}$$

$$\frac{\Gamma;\ \Delta_1 \vdash 1 \quad \Gamma;\ \Delta_2 \vdash A}{\Gamma;\ \Delta_1, \Delta_2 \vdash A}\ 1\mathrm{E}$$

$$\frac{\Gamma;\ \Delta_1 \vdash A_1 \quad \Gamma;\ \Delta_2 \vdash A_1}{\Gamma;\ \Delta_1, \Delta_2 \vdash A_1 \otimes A_2}\ \otimes\mathrm{I}$$

$$\frac{\Gamma;\ \Delta_1 \vdash A_1 \otimes A_2 \quad \Gamma;\ \Delta_2, A_1, A_2 \vdash B}{\Gamma;\ \Delta_1, \Delta_2 \vdash B}\ \otimes\mathrm{E}$$

$$\frac{\Gamma;\ \Delta, A \vdash B}{\Gamma;\ \Delta \vdash A \multimap B}\ \multimap\mathrm{I}$$

$$\frac{\Gamma;\ \Delta_1 \vdash A \multimap B \quad \Gamma;\ \Delta_2 \vdash A}{\Gamma;\ \Delta_1, \Delta_2 \vdash B}\ \multimap\mathrm{E}$$

$$\frac{\Gamma;\ \cdot \vdash A\ \mathbf{lin}}{\Gamma;\ \cdot \vdash\ !A\ \mathbf{lin}}\ !\mathrm{I}$$

$$\frac{\Gamma;\ \Delta_1 \vdash\ !A\ \mathbf{lin} \quad \Gamma, A\ \mathbf{per};\ \Delta_2 \vdash B}{\Gamma;\ \Delta_1, \Delta_2 \vdash B}\ !\mathrm{E}$$

Figure 2.3: Intuitionistic Linear Logic

Chapter 3

Terms for ILL

3.1 Evaluations

e	$::=$		expressions
	$\mid$	x	
	$\mid$	u	
	$\mid$	$\texttt{abort}\ e$	
	$\mid$	$\texttt{inj}_1\ e$	
	$\mid$	$\texttt{inj}_2\ e$	
	$\mid$	$\texttt{case}\ e\ \texttt{of}\ \texttt{inj}_1\ \ x.\ e_1 \mid \texttt{inj}_2\ \ y.\ e_2$	
	$\mid$	$[\,]$	
	$\mid$	$[\,e_1\ \&\ e_2\,]$	
	$\mid$	$\texttt{prj}_1\ e$	
	$\mid$	$\texttt{prj}_2\ e$	
	$\mid$	$\langle\rangle$	
	$\mid$	$\texttt{let}\ \langle\rangle = e_1\ \texttt{in}\ e_2$	
	$\mid$	$\langle e_1 \otimes e_2\rangle$	
	$\mid$	$\texttt{let}\ x \otimes y = e_1\ \texttt{in}\ e_2$	
	$\mid$	$\lambda x{:}A.\ e$	
	$\mid$	$e_1\ e_2$	
	$\mid$	$!e$	
	$\mid$	$\texttt{let}\ !u = e_1\ \texttt{in}\ e_2$	
	$\mid$	(e)	M
	$\mid$	$\{e_1/x\}e_2$	M
	$\mid$	$\{e_1/u\}e_2$	M
	$\mid$	$E[e]$	M

$$
\begin{array}{lll}
v & ::= & \text{values} \\
 & | & x \\
 & | & \mathtt{inj}_1\ v \\
 & | & \mathtt{inj}_2\ v \\
 & | & \mathtt{[}\, e_1 \mathbin{\&} e_2 \,\mathtt{]} \\
 & | & \langle\rangle \\
 & | & \langle v_1 \otimes v_2 \rangle \\
 & | & \lambda x{:}A.\ e \\
 & | & !e
\end{array}
$$

$\boxed{\Gamma;\ \Delta \vdash e : A}$ Intuitionistic Linear Logic Proof Terms

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

$$\frac{u{:}A \in \Gamma}{\Gamma;\ \cdot \vdash u : A}$$

$$\frac{\Gamma;\ \Delta_1 \vdash e_1 : A_1 \quad \Gamma;\ \Delta_2 \vdash e_2 : A_1}{\Gamma;\ \Delta_1, \Delta_2 \vdash \langle e_1 \otimes e_2 \rangle : A_1 \otimes A_2}$$

$$\frac{\Gamma;\ \Delta_1 \vdash e_1 : A_1 \otimes A_2 \quad \Gamma;\ \Delta_2, x_1{:}A_1, x_2{:}A_2 \vdash e_2 : B}{\Gamma;\ \Delta_1, \Delta_2 \vdash \mathtt{let}\ x_1 \otimes x_2 = e_1\ \mathtt{in}\ e_2 : B}$$

$$\frac{}{\Gamma;\ \cdot \vdash \langle\rangle : 1}$$

$$\frac{\Gamma;\ \Delta_1 \vdash e_1 : 1 \quad \Gamma;\ \Delta_2 \vdash e_2 : A}{\Gamma;\ \Delta_1, \Delta_2 \vdash \mathtt{let}\ \langle\rangle = e_1\ \mathtt{in}\ e_2 : A}$$

$$\frac{\Gamma;\ \Delta, x{:}A \vdash e : B}{\Gamma;\ \Delta \vdash \lambda x{:}A.\ e : A \multimap B}$$

$$\frac{\Gamma;\ \Delta_1 \vdash e_1 : A \multimap B \quad \Gamma;\ \Delta_2 \vdash e_2 : A}{\Gamma;\ \Delta_1, \Delta_2 \vdash (e_1\ e_2) : B}$$

$$\frac{\Gamma;\ \Delta \vdash e_1 : A \quad \Gamma;\ \Delta \vdash e_2 : B}{\Gamma;\ \Delta \vdash \mathtt{[}\, e_1 \mathbin{\&} e_2 \,\mathtt{]} : A \mathbin{\&} B}$$

$$\frac{\Gamma;\ \Delta \vdash e : A \mathbin{\&} B}{\Gamma;\ \Delta \vdash \mathtt{prj}_1\ e : A}$$

$$\frac{\Gamma;\ \Delta \vdash e : A \mathbin{\&} B}{\Gamma;\ \Delta \vdash \mathtt{prj}_2\ e : B}$$

$$\frac{}{\Gamma;\ \Delta \vdash \mathtt{[\,]} : \top}$$

$$\frac{\Gamma;\ \Delta \vdash e : A}{\Gamma;\ \Delta \vdash \mathtt{inj}_1\ e : A \oplus B}$$

$$\frac{\Gamma;\ \Delta \vdash e : B}{\Gamma;\ \Delta \vdash \mathtt{inj}_2\ e : A \oplus B}$$

$$\frac{\Gamma;\ \Delta_1 \vdash e_0 : A_1 \oplus A_2 \quad \Gamma;\ \Delta_2, x_1{:}A_1 \vdash e_1 : B \quad \Gamma;\ \Delta_2, x_2{:}A_2 \vdash e_2 : B}{\Gamma;\ \Delta_1, \Delta_2 \vdash \mathtt{case}\ e_0\ \mathtt{of}\ \mathtt{inj}_1\ \ x_1.\ e_1 \mid \mathtt{inj}_2\ \ x_2.\ e_2 : B}$$

$$\frac{\Gamma;\ \Delta_1 \vdash e : 0}{\Gamma;\ \Delta_1, \Delta_2 \vdash \mathtt{abort}\ e : A}$$

$$\frac{\Gamma;\ \cdot \vdash e : A}{\Gamma;\ \cdot \vdash\ !e : !A}$$

$$\frac{\Gamma;\ \Delta_1 \vdash e_1 : !A \quad \Gamma, u{:}A;\ \Delta_2 \vdash e_2 : B}{\Gamma;\ \Delta_1, \Delta_2 \vdash \mathtt{let}\ !u = e_1\ \mathtt{in}\ e_2 : B}$$

E	$::=$		evaluation contexts
	$\mid$	$\Box$	
	$\mid$	$\mathtt{inj}_1\ E$	
	$\mid$	$\mathtt{inj}_2\ E$	
	$\mid$	$\mathtt{case}\ E\ \mathtt{of}\ \mathtt{inj}_1\ \ x.\ e_1 \mid \mathtt{inj}_2\ \ y.\ e_2$	
	$\mid$	$\mathtt{prj}_1\ E$	
	$\mid$	$\mathtt{prj}_2\ E$	
	$\mid$	$\mathtt{let}\ \langle\rangle = E\ \mathtt{in}\ e$	
	$\mid$	$\langle E \otimes e \rangle$	
	$\mid$	$\langle v \otimes E \rangle$	
	$\mid$	$\mathtt{let}\ x \otimes y = E\ \mathtt{in}\ e$	
	$\mid$	$E\ e$	
	$\mid$	$v\ E$	
	$\mid$	$\mathtt{let}\ !u = E\ \mathtt{in}\ e$	
	$\mid$	(E)	M

$\boxed{E[e_1] = e_2}$ Context Filling

$$\frac{}{\Box[e_1] = e_1}$$

$$\frac{E[e_1] = e_2}{(\mathtt{inj}_1\ E)[e_1] = \mathtt{inj}_1\ e_2}$$

$$\frac{E[e_1] = e_2}{(\mathtt{inj}_2\ E)[e_1] = \mathtt{inj}_2\ e_2}$$

$$\frac{E[e_1] = e_2}{(\mathtt{case}\ E\ \mathtt{of}\ \mathtt{inj}_1\ \ x.\ e_1' \mid \mathtt{inj}_2\ \ y.\ e_2')[e_1] = \mathtt{case}\ e_2\ \mathtt{of}\ \mathtt{inj}_1\ \ x.\ e_1' \mid \mathtt{inj}_2\ \ y.\ e_2'}$$

$$\frac{E[e_1] = e_2}{(\texttt{prj}_1\ E)[e_1] = \texttt{prj}_1\ e_2}$$

$$\frac{E[e_1] = e_2}{(\texttt{prj}_2\ E)[e_1] = \texttt{prj}_2\ e_2}$$

$$\frac{E[e_1] = e_2}{(\texttt{let}\ \langle\rangle = E\ \texttt{in}\ e)[e_1] = \texttt{let}\ \langle\rangle = e_2\ \texttt{in}\ e}$$

$$\frac{E[e_1] = e_2}{\langle E \otimes e\rangle[e_1] = \langle e_2 \otimes e\rangle}$$

$$\frac{E[e_1] = e_2}{\langle v \otimes E\rangle[e_1] = \langle v \otimes e_2\rangle}$$

$$\frac{E[e_1] = e_2}{(E\ \ e)[e_1] = e_2\ e_1}$$

$$\frac{E[e_1] = e_2}{(v\ \ E)[e_1] = v\ \ e_2}$$

$$\frac{E[e_1] = e_2}{(\texttt{let}\ !u = E\ \texttt{in}\ e)[e_1] = \texttt{let}\ !u = e_2\ \texttt{in}\ e}$$

$\boxed{\{e/x\}e_1 = e_2}$ Substitution for Linear Hypotheses

$$\frac{}{\{e/x\}x = e}$$

$$\frac{\{e/x\}e = e'}{\{e/x\}(\texttt{inj}_1\ e) = \texttt{inj}_1\ e'}$$

$$\frac{\{e/x\}e_0 = e_0'}{\{e/x\}(\texttt{inj}_1\ e_0) = \texttt{inj}_1\ e_0'}$$

$$\frac{\{e/x\}e_0 = e_0'}{\{e/x\}(\texttt{case}\ e_0\ \texttt{of}\ \texttt{inj}_1\ \ y_1.\ e_1 \mid \texttt{inj}_2\ \ y_2.\ e_2) = \texttt{case}\ e_0'\ \texttt{of}\ \texttt{inj}_1\ \ y_1.\ e_1 \mid \texttt{inj}_2\ \ y_2.\ e_2}$$

$$\frac{\{e/x\}e_1 = e_1' \quad \{e/x\}e_2 = e_2' \quad x \neq y_1 \quad x \neq y_2}{\{e/x\}(\texttt{case}\ e_0\ \texttt{of}\ \texttt{inj}_1\ \ y_1.\ e_1 \mid \texttt{inj}_2\ \ y_2.\ e_2) = \texttt{case}\ e_0\ \texttt{of}\ \texttt{inj}_1\ \ y_1.\ e_1' \mid \texttt{inj}_2\ \ y_2.\ e_2'}$$

$$\frac{}{\{e/x\}\texttt{[]} = \texttt{[]}}$$

$$\frac{\{e/x\}e_0 = e_0'}{\{e/x\}(\texttt{prj}_1\ e_0) = \texttt{prj}_1\ e_0'}$$

$$\frac{\{e/x\}e_0 = e_0'}{\{e/x\}(\texttt{prj}_2\ e_0') = \texttt{prj}_2\ e_0'}$$

$$\frac{\{e/x\}e_1 = e_1' \quad \{e/x\}e_2 = e_2'}{\{e/x\}\,[\,e_1 \,\&\, e_2\,] \;=\; [\,e_1' \,\&\, e_2'\,]}$$

$$\frac{\{e/x\}e_1 = e_1'}{\{e/x\}(\texttt{let}\ \langle\rangle = e_1\ \texttt{in}\ e_2) = \texttt{let}\ \langle\rangle = e_1'\ \texttt{in}\ e_2}$$

$$\frac{\{e/x\}e_2 = e_2'}{\{e/x\}(\texttt{let}\ \langle\rangle = e_1\ \texttt{in}\ e_2) = \texttt{let}\ \langle\rangle = e_1\ \texttt{in}\ e_2'}$$

$$\frac{\{e/x\}e_1 = e_1'}{\{e/x\}\langle e_1 \otimes e_2\rangle = \langle e_1' \otimes e_2\rangle}$$

$$\frac{\{e/x\}e_2 = e_2'}{\{e/x\}\langle e_1 \otimes e_2\rangle = \langle e_1 \otimes e_2'\rangle}$$

$$\frac{\{e/x\}e_1 = e_1'}{\{e/x\}(\texttt{let}\ y_1 \otimes y_2 = e_1\ \texttt{in}\ e_2) = \texttt{let}\ y_1 \otimes y_2 = e_1'\ \texttt{in}\ e_2}$$

$$\frac{\{e/x\}e_2 = e_2'}{\{e/x\}(\texttt{let}\ y_1 \otimes y_2 = e_1\ \texttt{in}\ e_2) = \texttt{let}\ y_1 \otimes y_2 = e_1\ \texttt{in}\ e_2'}$$

$$\frac{\{e/x\}e_1 = e_1' \quad x \neq y}{\{e/x\}(\lambda y{:}A.\ e_1) = \lambda y{:}A.\ e_1'}$$

$$\frac{\{e/x\}e_1 = e_1'}{\{e/x\}(e_1\ e_2) = (e_1'\ e_2)}$$

$$\frac{\{e/x\}e_2 = e_2'}{\{e/x\}(e_1\ e_2) = (e_1\ e_2')}$$

$$\frac{\{e/x\}e_1 = e_1'}{\{e/x\}(\texttt{let}\ !u = e_1\ \texttt{in}\ e_2) = \texttt{let}\ !u = e_1'\ \texttt{in}\ e_2}$$

$$\frac{\{e/x\}e_2 = e_2'}{\{e/x\}(\texttt{let}\ !u = e_1\ \texttt{in}\ e_2) = \texttt{let}\ \langle\rangle = e_1\ \texttt{in}\ e_2'}$$

$\boxed{\{e/u\}e_1 = e_2}$ Substitution for Persistent Hypotheses

$$\frac{}{\{e/u\}u = e}$$

$$\frac{u' \neq u}{\{e/u\}u' = u'}$$

$$\frac{}{\{e/u\}x = x}$$

$$\frac{\{e/u\}e_0 = e_0'}{\{e/u\}\texttt{abort}\ e_0 = \texttt{abort}\ e_0'}$$

$$\frac{\{e/u\}e = e'}{\{e/u\}(\mathtt{inj}_1\ e) = \mathtt{inj}_1\ e'}$$

$$\frac{\{e/u\}e_0 = e_0'}{\{e/u\}(\mathtt{inj}_1\ e_0) = \mathtt{inj}_1\ e_0'}$$

$$\frac{\{e/u\}e_0 = e_0' \quad \{e/u\}e_1 = e_1' \quad \{e/u\}e_2 = e_2'}{\{e/u\}(\mathtt{case}\ e_0\ \mathtt{of}\ \mathtt{inj}_1\ \ y_1.\ e_1\ \mathtt{|}\ \mathtt{inj}_2\ \ y_2.\ e_2) = \mathtt{case}\ e_0\ \mathtt{of}\ \mathtt{inj}_1\ \ y_1.\ e_1'\ \mathtt{|}\ \mathtt{inj}_2\ \ y_2.\ e_2'}$$

$$\frac{}{\{e/u\}\mathtt{[\,]} = \mathtt{[\,]}}$$

$$\frac{\{e/u\}e_0 = e_0'}{\{e/u\}(\mathtt{prj}_1\ e_0) = \mathtt{prj}_1\ e_0'}$$

$$\frac{\{e/u\}e_0 = e_0'}{\{e/u\}(\mathtt{prj}_2\ e_0') = \mathtt{prj}_2\ e_0'}$$

$$\frac{\{e/u\}e_1 = e_1' \quad \{e/u\}e_2 = e_2'}{\{e/u\}\mathtt{[}\,e_1\ \&\ e_2\,\mathtt{]}\ =\ \mathtt{[}\,e_1'\ \&\ e_2'\,\mathtt{]}}$$

$$\frac{}{\{e/u\}\langle\rangle = \langle\rangle}$$

$$\frac{\{e/u\}e_1 = e_1' \quad \{e/u\}e_2 = e_2'}{\{e/u\}(\mathtt{let}\ \langle\rangle = e_1\ \mathtt{in}\ e_2) = \mathtt{let}\ \langle\rangle = e_1\ \mathtt{in}\ e_2'}$$

$$\frac{\{e/u\}e_1 = e_1' \quad \{e/u\}e_2 = e_2'}{\{e/u\}\langle e_1 \otimes e_2\rangle = \langle e_1 \otimes e_2'\rangle}$$

$$\frac{\{e/u\}e_1 = e_1' \quad \{e/u\}e_2 = e_2'}{\{e/u\}(\mathtt{let}\ y_1 \otimes y_2 = e_1\ \mathtt{in}\ e_2) = \mathtt{let}\ y_1 \otimes y_2 = e_1\ \mathtt{in}\ e_2'}$$

$$\frac{\{e/u\}e_1 = e_1' \quad x \neq y}{\{e/u\}(\lambda y{:}A.\ e_1) = \lambda y{:}A.\ e_1'}$$

$$\frac{\{e/u\}e_1 = e_1' \quad \{e/u\}e_2 = e_2'}{\{e/u\}(e_1\ e_2) = (e_1\ e_2')}$$

$$\frac{\{e/u\}e_0 = e_0'}{\{e/u\}!e_0 = !e_0'}$$

$$\frac{\{e/u\}e_1 = e_1' \quad \{e/u\}e_2 = e_2'}{\{e/u\}(\mathtt{let}\ !u = e_1\ \mathtt{in}\ e_2) = \mathtt{let}\ \langle\rangle = e_1'\ \mathtt{in}\ e_2'}$$

$\boxed{e_1 \ \mapsto\ e_2}$ Evaluation Relation

$$\frac{e_1 \ \mapsto\ e_2}{E[e_1] \ \mapsto\ E[e_2]}$$

$$\overline{\texttt{case inj}_1\ v\ \texttt{of inj}_1\ \ x.\ e_1 \mid \texttt{inj}_2\ \ y.\ e_2\ \mapsto\ \{v/x\}e_1}$$

$$\overline{\texttt{case inj}_2\ v\ \texttt{of inj}_1\ \ x.\ e_1 \mid \texttt{inj}_2\ \ y.\ e_2\ \mapsto\ \{v/y\}e_2}$$

$$\overline{\texttt{prj}_1\ [e_1 \,\&\, e_2]\ \mapsto\ e_1}$$

$$\overline{\texttt{prj}_2\ [e_1 \,\&\, e_2]\ \mapsto\ e_2}$$

$$\overline{\texttt{let}\ \langle\rangle = \langle\rangle\ \texttt{in}\ e\ \mapsto\ e}$$

$$\overline{\texttt{let}\ x \otimes y = \langle v_1 \otimes v_2\rangle\ \texttt{in}\ e\ \mapsto\ \{v_1/x\}\{v_2/y\}e}$$

$$\overline{(\lambda x{:}A.\ e)\ v\ \mapsto\ \{v/x\}e}$$

$$\overline{\texttt{let}\ !u = !e_1\ \texttt{in}\ e_2\ \mapsto\ \{e_1/u\}e_2}$$

3.2 Call-by-Name Translation of Intuitionistic Logic

The basis of the call-by-name translation of Intuitionistic Logic into Intuitionistic Linear Logic is given by extending the following type translation to terms in the "obvious" way:

$$\begin{array}{rcl}
True^\circ & = & 1 \\
(A \wedge B)^\circ & = & A^\circ \,\&\, B^\circ \\
(A \vee B)^\circ & = & !A^\circ \oplus !B^\circ \\
(A \Rightarrow B)^\circ & = & !A^\circ \multimap B^\circ
\end{array}$$

TODO: Typeset the term translation rules.

3.3 Call-by-Value Translation of Intuitionistic Logic

The basis of the call-by-value translation of Intuitionistic Logic into Intuitionstic Linear Logic is given by extending the following (mutually recursive) type translations to terms in the "obvious" way. Note that this translation makes a distinction between "values" and "computations." The main invariant of the translation is that the ! constructor is used in two ways: (1) to make *values* persistent (and hence duplicable), and (2) to make *computations* suspended.

$$
\begin{array}{rcl}
True^{\mathsf{v}} & = & 1 \\
(A \wedge B)^{\mathsf{v}} & = & !A^{\mathsf{v}} \mathbin{\&} !B^{\mathsf{v}} \\
(A \vee B)^{\mathsf{v}} & = & !A^{\mathsf{v}} \oplus !B^{\mathsf{v}} \\
(A \Rightarrow B)^{\mathsf{v}} & = & !A^{\mathsf{v}} \multimap B^{\mathsf{c}} \\
 & & \\
A^{\mathsf{c}} & = & !(A^{\mathsf{v}})
\end{array}
$$

TODO: Typeset the term translation rules.

Chapter 4

Proof Search and ILL

4.1 Normal Proofs

We can introduce two judgments that constrain the proofs to be in *normal form*—that is, containing no subterms that can be reduced by β-reduction (a.k.a. local soundness reduction rules).

$$
\begin{array}{lcll}
I & ::= & & \text{intro forms} \\
 & | & \mathtt{inj}_1\, I & \\
 & | & \mathtt{inj}_2\, I & \\
 & | & \mathtt{case}\ E\ \mathtt{of}\ \mathtt{inj}_1\ x.\, I_1 \mid \mathtt{inj}_2\ y.\, I_2 & \\
 & | & [\, I_1 \,\&\, I_2 \,] & \\
 & | & \langle\rangle & \\
 & | & \langle I_1 \otimes I_2 \rangle & \\
 & | & \lambda x{:}A.\, I & \\
 & | & !I & \\
 & | & \mathtt{let}\ \langle\rangle = E\ \mathtt{in}\ I & \\
 & | & \mathtt{let}\ x \otimes y = E\ \mathtt{in}\ I & \\
 & | & \mathtt{let}\ !u = E\ \mathtt{in}\ I & \\
 & | & [\,] & \\
 & | & \mathtt{abort}\ E & \\
 & | & E & \\
 & | & (I) & \mathsf{M}
\end{array}
$$

$$
\begin{array}{lcll}
E & ::= & & \text{elim forms} \\
 & | & x & \\
 & | & u & \\
 & | & \mathtt{prj}_1\, E & \\
 & | & \mathtt{prj}_2\, E & \\
 & | & E\ I &
\end{array}
$$

$|\quad (I : A)$
$|\quad (E) \qquad \mathsf{M}$

The corresponding inference rules are given in Figures 4.1 and 4.2.

We have the following soundness theorem for normal proofs:

Theorem 4.1 (Soundness for Normal Proofs)**.**

1. *If* $\Gamma; \Delta \vdash I : A \Uparrow$ *then* $\Gamma;\ \Delta \vdash A$.
2. *If* $\Gamma; \Delta \vdash E : A \downarrow$ *then* $\Gamma;\ \Delta \vdash A$.

It is much harder to prove completeness because there is no simple, local way to decorate a non-normal proof using the $A \Uparrow$ and $A \downarrow$ judgments.

Instead, we create an augmented version of the rules (marked with $\vdash^{+}$) and shown in Figures 4.3 and 4.4.

Theorem 4.2 (Soundness for Normal Proofs)**.**

1. *If* $\Gamma;\ \Delta \vdash^{+} A \Uparrow$ *then* $\Gamma;\ \Delta \vdash A$.
2. *If* $\Gamma;\ \Delta \vdash^{+} A \downarrow$ *then* $\Gamma;\ \Delta \vdash A$.

Theorem 4.3 (Completeness of Normal Proofs)**.**

1. *If* $\Gamma;\ \Delta \vdash A$ *then* $\Gamma;\ \Delta \vdash^{+} A \Uparrow$.
2. *If* $\Gamma;\ \Delta \vdash A$ *then* $\Gamma;\ \Delta \vdash^{+} A \downarrow$.

4.2 Sequent Calculus

$\boxed{\Gamma;\Delta\vdash I : J}$ Normal form Checking

$$\frac{\Gamma;\Delta_1\vdash I_1 : A_1\Uparrow \quad \Gamma;\Delta_2\vdash I_2 : A_1\Uparrow}{\Gamma;\Delta_1,\Delta_2\vdash \langle I_1\otimes I_2\rangle : A_1\otimes A_2\Uparrow}$$

$$\frac{\Gamma;\Delta_1\vdash E : A_1\otimes A_2\downarrow \quad \Gamma;\Delta_2,x_1{:}A_1,x_2{:}A_2\vdash I : B\Uparrow}{\Gamma;\Delta_1,\Delta_2\vdash \mathtt{let}\ x_1\otimes x_2 = E\ \mathtt{in}\ I : B\Uparrow}$$

$$\frac{}{\Gamma;\cdot\vdash\langle\rangle : 1\Uparrow}$$

$$\frac{\Gamma;\Delta_1\vdash E : 1\downarrow \quad \Gamma;\Delta_2\vdash I : A\Uparrow}{\Gamma;\Delta_1,\Delta_2\vdash \mathtt{let}\ \langle\rangle = E\ \mathtt{in}\ I : A\Uparrow}$$

$$\frac{\Gamma;\Delta,x{:}A\vdash I : B\Uparrow}{\Gamma;\Delta\vdash\lambda x{:}A.\ I : A\multimap B\Uparrow}$$

$$\frac{\Gamma;\Delta\vdash I_1 : A\Uparrow \quad \Gamma;\Delta\vdash I_2 : B\Uparrow}{\Gamma;\Delta\vdash \mathtt{[}I_1\mathbin{\&}I_2\mathtt{]} : A\mathbin{\&}B\Uparrow}$$

$$\frac{}{\Gamma;\Delta\vdash \mathtt{[\,]} : \top\Uparrow}$$

$$\frac{\Gamma;\Delta\vdash I : A\Uparrow}{\Gamma;\Delta\vdash \mathtt{inj}_1\ I : A\oplus B\Uparrow}$$

$$\frac{\Gamma;\Delta\vdash I : B\Uparrow}{\Gamma;\Delta\vdash \mathtt{inj}_2\ I : A\oplus B\Uparrow}$$

$$\frac{\Gamma;\Delta_1\vdash E : A_1\oplus A_2\downarrow \quad \Gamma;\Delta_2,x_1{:}A_1\vdash I_1 : B\Uparrow \quad \Gamma;\Delta_2,x_2{:}A_2\vdash I_2 : B\Uparrow}{\Gamma;\Delta_1,\Delta_2\vdash \mathtt{case}\ E\ \mathtt{of}\ \mathtt{inj}_1\ x_1.\ I_1 \mid \mathtt{inj}_2\ x_2.\ I_2 : B\Uparrow}$$

$$\frac{\Gamma;\Delta_1\vdash E : 0\downarrow}{\Gamma;\Delta_1,\Delta_2\vdash \mathtt{abort}\ E : A\Uparrow}$$

$$\frac{\Gamma;\cdot\vdash I : A\Uparrow}{\Gamma;\cdot\vdash !I : !A\Uparrow}$$

$$\frac{\Gamma;\Delta_1\vdash E : !A\downarrow \quad \Gamma,u{:}A;\Delta_2\vdash I : B\Uparrow}{\Gamma;\Delta_1,\Delta_2\vdash \mathtt{let}\ !u = E\ \mathtt{in}\ I : B\Uparrow}$$

$$\frac{\Gamma;\Delta\vdash E : A\downarrow}{\Gamma;\Delta\vdash E : A\Uparrow}$$

Figure 4.1: Normal form checking rules for ILL.

$\boxed{\Gamma; \Delta \vdash E : J}$ Normal Form Extraction

$$\frac{}{\Gamma; x{:}A \vdash x : A \downarrow}$$

$$\frac{u{:}A \in \Gamma}{\Gamma; \cdot \vdash u : A \downarrow}$$

$$\frac{\Gamma; \Delta_1 \vdash E : A \multimap B \downarrow \quad \Gamma; \Delta_2 \vdash I : A \Uparrow}{\Gamma; \Delta_1, \Delta_2 \vdash (E\ I) : B \downarrow}$$

$$\frac{\Gamma; \Delta \vdash E : A \,\&\, B \downarrow}{\Gamma; \Delta \vdash \mathtt{prj}_1\ E : A \downarrow}$$

$$\frac{\Gamma; \Delta \vdash E : A \,\&\, B \downarrow}{\Gamma; \Delta \vdash \mathtt{prj}_2\ E : B \downarrow}$$

Figure 4.2: Normal form hypothesis extraction rules for ILL.

$\boxed{\Gamma;\ \Delta \vdash^{+} J}$ Augmented Checking

$$\frac{\Gamma;\ \Delta_1 \vdash^{+} A_1 \Uparrow \qquad \Gamma;\ \Delta_2 \vdash^{+} A_1 \Uparrow}{\Gamma;\ \Delta_1, \Delta_2 \vdash^{+} A_1 \otimes A_2 \Uparrow}$$

$$\frac{\Gamma;\ \Delta_1 \vdash^{+} A_1 \otimes A_2 \downarrow \qquad \Gamma;\ \Delta_2, x_1{:}A_1, x_2{:}A_2 \vdash^{+} B \Uparrow}{\Gamma;\ \Delta_1, \Delta_2 \vdash^{+} B \Uparrow}$$

$$\frac{}{\Gamma;\ \cdot \vdash^{+} 1 \Uparrow}$$

$$\frac{\Gamma;\ \Delta_1 \vdash^{+} 1 \downarrow \qquad \Gamma;\ \Delta_2 \vdash^{+} A \Uparrow}{\Gamma;\ \Delta_1, \Delta_2 \vdash^{+} A \Uparrow}$$

$$\frac{\Gamma;\ \Delta, x{:}A \vdash^{+} B \Uparrow}{\Gamma;\ \Delta \vdash^{+} A \multimap B \Uparrow}$$

$$\frac{\Gamma;\ \Delta \vdash^{+} A \Uparrow \qquad \Gamma;\ \Delta \vdash^{+} B \Uparrow}{\Gamma;\ \Delta \vdash^{+} A \mathbin{\&} B \Uparrow}$$

$$\frac{}{\Gamma;\ \Delta \vdash^{+} \top \Uparrow}$$

$$\frac{\Gamma;\ \Delta \vdash^{+} A \Uparrow}{\Gamma;\ \Delta \vdash^{+} A \oplus B \Uparrow}$$

$$\frac{\Gamma;\ \Delta \vdash^{+} B \Uparrow}{\Gamma;\ \Delta \vdash^{+} A \oplus B \Uparrow}$$

$$\frac{\Gamma;\ \Delta_1 \vdash^{+} A_1 \oplus A_2 \downarrow \qquad \Gamma;\ \Delta_2, x_1{:}A_1 \vdash^{+} B \Uparrow \qquad \Gamma;\ \Delta_2, x_2{:}A_2 \vdash^{+} B \Uparrow}{\Gamma;\ \Delta_1, \Delta_2 \vdash^{+} B \Uparrow}$$

$$\frac{\Gamma;\ \Delta_1 \vdash^{+} 0 \downarrow}{\Gamma;\ \Delta_1, \Delta_2 \vdash^{+} A \Uparrow}$$

$$\frac{\Gamma;\ \cdot \vdash^{+} A \Uparrow}{\Gamma;\ \cdot \vdash^{+} !A \Uparrow}$$

$$\frac{\Gamma;\ \Delta_1 \vdash^{+} !A \downarrow \qquad \Gamma, u{:}A;\ \Delta_2 \vdash^{+} B \Uparrow}{\Gamma;\ \Delta_1, \Delta_2 \vdash^{+} B \Uparrow}$$

$$\frac{\Gamma;\ \Delta \vdash^{+} A \downarrow}{\Gamma;\ \Delta \vdash^{+} A \Uparrow}$$

Figure 4.3: Augmented rules for proving completeness.

$\boxed{\Gamma;\ \Delta \vdash^{+} J}$ Augmented Extraction

$$\frac{}{\Gamma;\ x{:}A \vdash^{+} A \downarrow}$$

$$\frac{u{:}A \in \Gamma}{\Gamma;\ \cdot \vdash^{+} A \downarrow}$$

$$\frac{\Gamma;\ \Delta_1 \vdash^{+} A \multimap B \downarrow \quad \Gamma;\ \Delta_2 \vdash^{+} A \Uparrow}{\Gamma;\ \Delta_1, \Delta_2 \vdash^{+} B \downarrow}$$

$$\frac{\Gamma;\ \Delta \vdash^{+} A \mathbin{\&} B \downarrow}{\Gamma;\ \Delta \vdash^{+} A \downarrow}$$

$$\frac{\Gamma;\ \Delta \vdash^{+} A \mathbin{\&} B \downarrow}{\Gamma;\ \Delta \vdash^{+} B \downarrow}$$

$$\frac{\Gamma;\ \Delta \vdash^{+} A \Uparrow}{\Gamma;\ \Delta \vdash^{+} A \downarrow}$$

Figure 4.4: Augmented extraction rules. Note the new coercion rule.

$\boxed{\Gamma;\ \Delta \stackrel{+}{\Longrightarrow} J}$ AugmentedILL Sequents

$$\frac{\Gamma;\ \Delta_1 \stackrel{+}{\Longrightarrow} A \quad \Gamma;\ \Delta_2, A \stackrel{+}{\Longrightarrow} B}{\Gamma;\ \Delta_1, \Delta_2 \stackrel{+}{\Longrightarrow} B}$$

Figure 4.5: Augmented sequent rules include all of the usual ones, plus cut.

$\boxed{\Gamma;\ \Delta \Longrightarrow J}$ ILL Sequents

$$\frac{}{\Gamma;\ A\uparrow \Longrightarrow A \Uparrow}$$

$$\frac{A \in \Gamma \quad \Gamma;\ A, \Delta_1 \Longrightarrow B}{\Gamma;\ \Delta_1 \Longrightarrow B}$$

$$\frac{\Gamma;\ \Delta_1 \Longrightarrow A \quad \Gamma;\ \Delta_2 \Longrightarrow B}{\Gamma;\ \Delta_1, \Delta_2 \Longrightarrow A \otimes B}$$

$$\frac{\Gamma;\ \Delta, A_1, A_2 \Longrightarrow B}{\Gamma;\ \Delta, A_1 \otimes A_2 \Longrightarrow B}$$

$$\frac{}{\Gamma;\ \cdot \Longrightarrow 1}$$

$$\frac{\Gamma;\ \Delta \Longrightarrow B}{\Gamma;\ 1, \Delta \Longrightarrow B}$$

$$\frac{\Gamma;\ \Delta, A \Longrightarrow B}{\Gamma;\ \Delta \Longrightarrow A \multimap B}$$

$$\frac{\Gamma;\ \Delta_1 \Longrightarrow A_1 \quad \Gamma;\ \Delta_2, A_2 \Longrightarrow B}{\Gamma;\ A_1 \multimap A_2, \Delta_1, \Delta_2 \Longrightarrow B}$$

$$\frac{\Gamma;\ \Delta \Longrightarrow A \quad \Gamma;\ \Delta \Longrightarrow B}{\Gamma;\ \Delta \Longrightarrow A \,\&\, B}$$

$$\frac{\Gamma;\ \Delta, A_1 \Longrightarrow B}{\Gamma;\ \Delta, A_1 \,\&\, A_2 \Longrightarrow B}$$

$$\frac{\Gamma;\ \Delta, A_2 \Longrightarrow B}{\Gamma;\ \Delta, A_1 \,\&\, A_2 \Longrightarrow B}$$

$$\frac{}{\Gamma;\ \Delta \Longrightarrow \top}$$

$$\frac{\Gamma;\ \Delta \Longrightarrow A}{\Gamma;\ \Delta \Longrightarrow A \oplus B}$$

$$\frac{\Gamma;\ \Delta \Longrightarrow B}{\Gamma;\ \Delta \Longrightarrow A \oplus B}$$

$$\frac{\Gamma;\ \Delta, A_1 \Longrightarrow B \quad \Gamma;\ \Delta, A_2 \Longrightarrow B}{\Gamma;\ \Delta, A_1 \oplus A_2 \Longrightarrow B}$$

$$\frac{}{\Gamma;\ 0, \Delta \Longrightarrow B}$$

$$\frac{\Gamma;\ \cdot \Longrightarrow A}{\Gamma;\ \cdot \Longrightarrow\ !A}$$

$$\frac{\Gamma, A;\ \Delta \Longrightarrow B}{\Gamma;\ !A, \Delta \Longrightarrow B}$$

Figure 4.6: Sequent Presentation of ILL (Augmented)

Bibliography

[1] Martín Abadi, Anindya Banerjee, Nevin Heintze, and Jon Riecke. A core calculus of dependency. In *POPL*. ACM, 1999.

[2] Samson Abramsky. Computational interpretations of linear logic. *Theoretical Computer Science*, 111:3–57, 1993.

[3] Samson Abramsky. Computational interpretations of linear logic. *Theoretical Computer Science*, 111:3–57, 1993.

[4] Amal Ahmed. Step-indexed syntactic logical relations for recursive and quantified types. In *ESOP*, pages 69–83, 2006.

[5] Amal Ahmed and Matthias Blume. Typed closure conversion preserves observational equivalence. *SIGPLAN Not.*, 43(9):157–168, 2008.

[6] Amal Ahmed, Matthew Fluet, and Greg Morrisett. L3: A linear language with locations. In *TLCA*, 2005.

[7] Amal Ahmed, Matthew Fluet, and Greg Morrisett. L3: A linear language with locations. *Fundam. Inf.*, 77(4):397–449, 2007.

[8] Andrew W Appel, Paul-André Mellies, Christopher D Richards, and Jérôme Vouillon. A very modal model of a modern, major, general type system. In *ACM SIGPLAN Notices*, volume 42, pages 109–122. ACM, 2007.

[9] Robert Atkey. Parameterised notions of computation. *JFP*, 19(3-4), 2009.

[10] Robert Atkey, Patricia Johann, and Andrew Kennedy. Abstraction and invariance for algebraically indexed types. In *POPL*. ACM, 2013.

[11] Patrick Baillot and Kazushige Terui. Light types for polynomial time computation in lambda-calculus. In *LICS*. IEEE, 2004.

[12] Andrew Barber. *Linear Type Theories, Semantics and Action Calculi*. PhD thesis, Edinburgh University, 1997.

[13] Emmanuel Beffara. A concurrent model for linear logic. *Electronic Notes in Theoretical Computer Science*, 155:147–168, 2006.

[14] G. Bellin and P. J. Scott. On the π-calculus and linear logic. *Theoretical Computer Science*, 135(1):11–65, 1994.

[15] Gianluigi Bellin, Martin Hyland, Edmund Robinson, and Christian Urban. Categorical proof theory of classical propositional calculus. *Theoretical Computer Science*, 364(2):146 – 165, 2006.

[16] Nick Benton, G. M. Bierman, J. Martin E. Hyland, and Valeria de Paiva. A term calculus for intuitionistic linear logic. In *Proceedings of the International Conference on Typed Lambda Calculi and Applications*, pages 75–90. Springer-Verlag LNCS 664, 1993.

[17] Nick Benton, Gavin M. Bierman, J. Martin E. Hyland, and Valeria de Paiva. Term assignment for intuitionistic linear logic. Technical Report 262, University of Cambridge, 1992.

[18] Nick Benton and Nicolas Tabareau. Compiling functional types to relational specifications for low level imperative code. In *TLDI*. ACM, 2009.

[19] Nick Benton and Philip Wadler. Linear logic, monads and the lambda calculus. In *Proceedings of the Eleventh Annual IEEE Symposium on Logic in Computer Science*, New Brunswick, New Jersey, July 1996.

[20] Nick Benton and Philip Wadler. Linear logic, monads and the lambda calculus. In *LICS*. IEEE, 1996.

[21] P. N. Benton. A mixed linear and non-linear logic: proofs, terms and models (preliminary report). Technical Report 352, Computer Laboratory, University of Cambridge, September 1994.

[22] P. N. Benton. A mixed linear and non-linear logic: proofs, terms and models. In *Proceedings of Computer Science Logic (CSL '94), Kazimierz, Poland.*, pages 121–135. Springer-Verlag, 1995.

[23] Josh Berdine. *Linear and Affine Typing of Continuation-Passing Style*. PhD thesis, Queen Mary, University of London, 2004.

[24] Josh Berdine, Peter W. O'Hearn, Uday S. Reddy, and Hayo Thielecke. Linearly used continuations. In *Proceedings of the Continuations Workshop*, 2001.

[25] G. M. Bierman, A. M. Pitts, and C. V. Russo. Operational properties of Lily, a polymorphic linear lambda calculus with recursion. In *Fourth International Workshop on Higher Order Operational Techniques in Semantics, Montral, volume 41 of Electronic Notes in Theoretical Computer Science*. Elsevier, 2000.

[26] Gavin Bierman. A classical linear lambda calculus. *Theoretical Computer Science*, 227(1–2):43–78, 1999.

[27] Gavin M. Bierman. Program equivalence in a linear functional language. *Journal of Functional Programming*, 10(2), 2000.

[28] Lars Birkedal, Rasmus Ejlers Møgelberg, and Rasmus Lerchedahl Petersen. Category-theoretic models of Linear Abadi & Plotkin Logic. *Theory and Applications in Categories*, 20(7), 2008.

[29] Andreas Blass. A game semantics for linear logic. *Annals of Pure and Applied logic*, 56(1):183–220, 1992.

[30] Aloïs Brunel. Quantitative classical realizability. *Inf. and Comp.*, 2013. To appear.

[31] Aloïs Brunel and Antoine Madet. Indexed realizability for bounded-time programming with references and type fixpoints. In *APLAS*. Springer, 2012.

[32] Aloïs Brunel and Kazushige Terui. Church => scott = ptime: an application of resource sensitive realizability. *EPTCS*, 23:31–46.

[33] T. Brus, M.C.J.D. van Eekelen, M. van Leer, M.J. Plasmeijer, and H.P. Barendregt. CLEAN - a language for functional graph rewriting. In *Proc. of Conference on Functional Programming Languages and Computer Architecture (FPCA)*, number 274 in LNCS, pages 364–384. Springer-Verlag, 1987.

[34] Luís Caires and Frank Pfenning. Session types as intuitionistic linear propositions. In *Proceedings of the 21st International Conference on Concurrency Theory (CONCUR 2010)*, Paris, France, August 2010. Springer LNCS.

[35] Luís Caires and Frank Pfenning. Session types as intuitionistic linear propositions. In *CONCUR*, 2010.

[36] Bor-Yuh Evan Chang, Kaustuv Chaudhuri, and Frank Pfenning. A judgmental analysis of linear logic. 2003.

[37] Arthur Charguéraud and François Pottier. Functional translation of a calculus of capabilities. In *ICFP '08: Proceeding of the 13th ACM SIGPLAN international conference on Functional programming*, pages 213–224, New York, NY, USA, 2008. ACM.

[38] J.R.B. Cockett and R.A.G. Seely. Weakly distributive categories. *Journal of Pure and Applied Algebra*, 114(2):133 – 173, 1997.

[39] Karl Crary, David Walker, and Greg Morrisett. Typed memory management in a calculus of capabilities. In *Proc. 26th ACM Symp. on Principles of Programming Languages (POPL)*, pages 262–275, San Antonio, Texas, January 1999.

[40] U. Dal Lago and M. Gaboardi. Linear dependent types and relative completeness. In LICS. IEEE, 2011.

[41] Ugo Dal Lago and Martin Hofmann. Bounded linear logic, revisited. In *TLCA*. 2009.

[42] Ugo Dal Lago and Martin Hofmann. A semantic proof of polytime soundness of light affine logic. *Theory of Computing Systems*, 46(4):673–689, 2010.

[43] Ugo Dal Lago and Martin Hofmann. Realizability models and implicit complexity. *TCS*, 412(20), 2011.

[44] Loris D'Antoni, Marco Gaboardi, Emilio Jesús Gallego Arias, Andreas Haeberlen, and Benjamin Pierce. Sensitivity analysis using type-based constraints. In *FPCDSL*. ACM, 2013.

[45] Valeria de Paiva and Eike Ritter. A parigot-style linear lambda-calculus for full intuitionistic linear logic. *Theory and Applications of Categories*, 17(3), 2006.

[46] Robert DeLine and Manuel Fähndrich. Enforcing high-level protocols in low-level software. In *Proc. of the SIGPLAN Conference on Programming Language Design*, pages 59–69, Snowbird, UT, June 2001.

[47] Derek Dreyer, Amal Ahmed, and Lars Birkedal. Logical step-indexed logical relations. In *LICS*. IEEE, 2009.

[48] David Easley and Jon Kleinberg. *Networks, crowds, and markets*, volume 8. Cambridge Univ Press, 2010.

[49] Thomas Ehrhard. Finiteness spaces. *Mathematical Structures in Computer Science*, 15(4):615–646, 2005.

[50] Manuel Fähndrich, Mark Aiken, Chris Hawblitzel, Orion Hodson, Galen Hunt, James R. Larus, and Steven Levi. Language support for fast and reliable message-based communication in singularity os. *SIGOPS Oper. Syst. Rev.*, 40(4):177–190, 2006.

[51] Manuel Fähndrich and Robert DeLine. Adoption and focus: Practical linear types for imperative programming. In *Proc. of the SIGPLAN Conference on Programming Language Design*, pages 13–24, Berlin, Germany, June 2002.

[52] M. Felleisen and R. Hieb. A revised report on the syntactic theories of sequential control and state. *Theoretical Computer Science*, 103(2):235–271, 1992.

[53] Andrzej Filinski. Linear continuations. In *Proc. 19th ACM Symp. on Principles of Programming Languages (POPL)*, pages 27–38, 1992.

[54] Andrzej Filinski. Representing layered monads. In *POPL*. ACM, 1999.

[55] Carsten Führmann and David Pym. Order-enriched categorical models of the classical sequent calculus. *Journal of Pure and Applied Algebra*, 204:21–78, 2006.

[56] Marco Gaboardi. *Linearity: an Analytic Tool in the study of Complexity and Semantics of Programming Languages*. PhD thesis, PhD thesis, Universita degli Studi di Torino-Institut National Polytechnique de Lorraine, 2007.

[57] Marco Gaboardi, Andreas Haeberlen, Justin Hsu, Arjun Narayan, and Benjamin C. Pierce. Linear dependent types for differential privacy. In *POPL*. ACM, 2013.

[58] Dan R. Ghica and Alex Smith. From bounded affine types to automatic timing analysis. *CoRR*, abs/1307.2473, 2013.

[59] J.-Y. Girard. Linear logic. *TCS*, 50(1):1–102, 1987.

[60] J.-Y. Girard, A. Scedrov, and P. Scott. Bounded linear logic. *TCS*, 97(1), 1992.

[61] Jean-Yves Girard. *Interprétation fonctionnelle et élimination des coupures de l'arith mé tique d'ordre supérieur*. Thèse d'état, University of Paris VII, 1972. Summary in *Scandinavian Logic Symposium*, pp. 63–92, North-Holland, 1971.

[62] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[63] Jean-Yves Girard. Proof-nets: the parallel syntax for proof-theory. *Lecture Notes in Pure and Applied Mathematics*, pages 97–124, 1996.

[64] Jean-Yves Girard. Locus solum: From the rules of logic to the logic of rules. *Mathematical. Structures in Comp. Sci.*, 11(3):301–506, 2001.

[65] Timothy G. Griffin. A formulae-as-types notion of control. In *Conference Record of the Seventeenth Annual ACM Symposium on Principles of Programming Languages*, pages 47–58. ACM Press, 1990.

[66] Nicolas Guenot. Focused proof search for linear logic in the calculus of structures. In *ICLP (Technical Communications)*, pages 84–93, 2010.

[67] Bertrand Guillou. Strictification of categories weakly enriched in symmetric monoidal categories. *Theory and Applications of Categories*, 24(20):564–579, 2010.

[68] Nevin Heintze and Jon G. Riecke. The SLam calculus: Programming with secrecy and integrity. In *POPL*. ACM, 1998.

[69] Michael Hicks, Greg Morrisett, Dan Grossman, and Trevor Jim. Experience with safe manual memory-management in Cyclone. In *ISMM '04: Proceedings of the 4th international symposium on Memory management*, pages 73–84, New York, NY, USA, 2004. ACM.

[70] Martin Hofmann and Steffen Jost. Static prediction of heap space usage for first-order functional programs. In *POPL*, 2003.

[71] Martin Hofmann and Philip J. Scott. Realizability models for BLL-like languages. *TCS.*, 318(1-2), 2004.

[72] Kohei Honda, Vasco T. Vasconcelos, and Makoto Kubo. Language primitives and type discipline for structured communication-based programming. In *ESOP98, volume 1381 of LNCS*, pages 122–138. Springer-Verlag, 1998.

[73] Kohei Honda, Vasco Thudichum Vasconcelos, and Makoto Kubo. Language primitives and type discipline for structured communication-based programming. In *ESOP*. Springer-Verlag, 1998.

[74] W. A. Howard. The formulae-as-types notion of contstruction. In *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus, and Formalism*. Academic Press, 1980.

[75] Dominic JD Hughes and Rob J Van Glabbeek. Proof nets for unit-free multiplicative-additive linear logic. *ACM Transactions on Computational Logic (TOCL)*, 6(4):784–842, 2005.

[76] Martin Hyland and Andrea Schalk. Glueing and orthogonality for models of linear logic. *Theoretical Computer Science*, 294(1):183–231, 2003.

[77] Atsushi Igarashi and Naoki Kobayashi. Resource usage analysis. *ACM Trans. Program. Lang. Syst.*, 27(2):264–313, 2005.

[78] Limin Jia, Jeffrey A. Vaughan, Karl Mazurak, Jianzhou Zhao, Luke Zarko, Joseph Schorr, and Steve Zdancewic. Aura: a programming language for authorization and audit. *SIGPLAN Not.*, 43(9):27–38, 2008.

[79] Oleg Kiselyov and Chung-chieh Shan. Lightweight monadic regions. In *Haskell '08: Proceedings of the first ACM SIGPLAN symposium on Haskell*, pages 1–12, New York, NY, USA, 2008. ACM.

[80] Naoki Kobayashi, Benjamin C. Pierce, and David N. Turner. Linearity and the Pi-Calculus. *Transactions on Programming Languages and Systems*, 21(5):914–947, 1999.

[81] Neelakantan R Krishnaswami, Nick Benton, and Jan Hoffmann. Higher-order functional reactive programming in bounded space. In *POPL*. ACM, 2012.

[82] Jean-Louis Krivine. A call-by-name lambda-calculus machine. *HOSC*, 20(3), 2007.

[83] Jean-Louis Krivine. Realizability in classical logic. *Panoramas et synthèses*, 27:197–229, 2009.

[84] Yves Lafont. The linear abstract machine. *Theoretical Computer Science*, 59:157–180, 1988. Corrections in vol. 62, pp. 327–328.

[85] Ugo Dal Lago and Ulrich Schöpp. Functional programming in sublinear space. In *ESOP*. Springer, 2010.

[86] J. Laird, G. Manzonetto, G. McCusker, and M. Pagani. Weighted relational models of typed lambda-calculi. In *LICS*. IEEE, 2013.

[87] Miguel Laplaza. Coherence for distributivity. *Lecture Notes in Math.*, 281, 1972.

[88] Chuck Liang and Dale Miller. Focusing and polarization in intuitionistic logic. In *Computer Science Logic*, pages 451–465. Springer, 2007.

[89] Chuck Liang and Dale Miller. Focusing and polarization in linear, intuitionistic, and classical logics. *Theoretical Computer Science*, 410(46):4747–4768, 2009.

[90] Patrick Lincoln and John Mitchell. Operational aspects of linear lambda calculus. In *7th Symposium on Logic in Computer Science, IEEE*, pages 235–246. IEEE Computer Society Press, 1992.

[91] John Maraist, Martin Odersky, David N. Turner, and Philip Wadler. Call-by-name, call-by-value, call-by-need, and the linear lambda calculus. In *11'th International Conference on the Mathematical Foundations of Programming Semantics*, New Orleans, Lousiana, – 1995.

[92] John Maraist, Martin Odersky, David N. Turner, and Philip Wadler. Call-by-name, call-by-value, call-by-need and the linear lambda calculus. *TCS*, 228(1-2), 1999.

[93] Per Martin-Löf. On the meanings of the logical constants and the justifications of the logical laws. *Nordic Journal of Philosophical Logic*, 1(1):11–60, 1996.

[94] Karl Mazurak and Steve Zdancewic. Lolliproc: to Concurrency from Classical Linear Logic via Curry-Howard and Control. In *Proc. of the 15th ACM SIGPLAN International Conference on Functional Programming (ICFP)*, 2010.

[95] Karl Mazurak and Steve Zdancewic. Lolliproc: to concurrency from classical linear logic via curry-howard and control. In *ICFP*, pages 39–50, 2010.

[96] Karl Mazurak, Jianzhou Zhao, and Steve Zdancewic. Lightweight linear types in System F°. In *ACM SIGPLAN International Workshop on Types in Languages Design and Implementation (TLDI)*, pages 77–88, 2010.

[97] Karl Mazurak, Jianzhou Zhao, and Steve Zdancewic. Lightweight linear types in System F°. In *TLDI '10: Proceedings of the 5th ACM SIGPLAN workshop on Types in language design and implementation*, pages 77–88, New York, NY, USA, 2010. ACM.

[98] Conor McBride. Faking it simulating dependent types in haskell. *J. Funct. Program.*, 12(5):375–392, 2002.

[99] James McKinna. Why dependent types matter. In *POPL '06: Conference record of the 33rd ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 1–1, New York, NY, USA, 2006. ACM.

[100] Paul-André Melliès. Categorical semantics of linear logic. *Panoramas et Syntheses*, 2009.

[101] Paul-André Melliès. Parametric monads and enriched adjunctions. Technical report, 2012. `http://www.pps.univ-paris-diderot.fr/~mellies/tensorial-logic/`.

[102] Paul-André Mellies and Jérôme Vouillon. Recursive polymorphic types and parametricity in an operational framework. In *Logic in Computer Science, 2005. LICS 2005. Proceedings. 20th Annual IEEE Symposium on*, pages 82–91. IEEE, 2005.

[103] Eugenio Moggi. Computational lambda-calculus and monads. In *LICS*. IEEE, 1989.

[104] Benoît Montagu and Didier Rémy. Modeling abstract types in modules with open existential types. In *POPL '09: Proceedings of the 36th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 354–365, New York, NY, USA, 2009. ACM.

[105] Aleksandar Nanevski, Greg Morrisett, and Lars Birkedal. Polymorphism and separation in Hoare type theory. In *ICFP '06: Proceedings of the eleventh ACM SIGPLAN International Conference on Functional programming*, pages 62–73, New York, NY, USA, 2006. ACM.

[106] F. Nielson, H. Riis Nielson, and C. L. Hankin. *Principles of Program Analysis*. Springer, 1999.

[107] Vivek Nigam and Dale Miller. Algorithmic specifications in linear logic with subexponentials. In *PPDP*. ACM, 2009.

[108] Dominic Orchard. Programming contextual computations, 2013. Cambridge University.

[109] Tomas Petricek, Dominic Orchard, and Alan Mycroft. Coeffects: Unified static analysis of context-dependence. In *ICALP*, 2013.

[110] Frank Pfenning. Structural cut elimination in linear logic. Technical report, DTIC Document, 1994.

[111] Frank Pfenning and Rowan Davies. A judgmental reconstruction of modal logic. *Mathematical structures in computer science*, 11(04):511–540, 2001.

[112] Andrew M Pitts. Parametric polymorphism and operational equivalence. *Mathematical Structures in Computer Science*, 10(3):321–359, 2000.

[113] Andrew M. Pitts. Step-indexed biorthogonality: a tutorial example. Dagstuhl, 2010.

[114] Gordon D. Plotkin. Second order type theory and recursion. Notes for a talk at the Scott Fest, February 1993.

[115] Norman Ramsey and Avi Pfeffer. Stochastic lambda calculus and monads of probability distributions. *ACM SIGPLAN Notices*, 37(1):154–165, 2002.

[116] Dana Randall. Efficient generation of random nonsingular matrices. *Random Structures & Algorithms*, 4(1):111–118, 1993.

[117] Jason Reed and Benjamin C. Pierce. Distance makes the types grow stronger: A calculus for differential privacy. In *ICFP*. ACM, 2010.

[118] John C. Reynolds. Separation logic: A logic for shared mutable data structures. In *Proceedings Seventeenth Annual IEEE Symposium on Logic in Computer Science*, pages 55–74, Los Alamitos, California, 2002. IEEE Computer Society.

[119] Eike Ritter, David J. Pym, and Lincoln A. Wallen. Proof-terms for classical and intuitionistic resolution. *Journal of Logic and Computation*, 10(2):173–207, 2000.

[120] Andrea Schalk. Whats is a categorical model of linear logic. Technical report, University of Manchester, 2004.

[121] Tim Sheard and Nathan Linger. Programming in omega. In *CEFP*, volume 5161 of *Lecture Notes in Computer Science*, pages 158–227. Springer, 2007.

[122] Kirsten Lackner Solberg. Annotated type systems for program analysis, 1995. Aarhus Univ.

[123] Nikhil Swamy and Michael Hicks. Verified enforcement of stateful information release policies. In *Proceedings of the ACM SIGPLAN Workshop on Programming Langauges and Analysis for Security (PLAS)*, 2008.

[124] Kaku Takeuchi, Kohei Honda, and Makoto Kubo. An interaction-based language and its typing system. In *Proceedings of PARLE'94*, pages 398–413. Springer-Verlag, 1994. Lecture Notes in Computer Science number 817.

[125] Jean-Pierre Talpin and Pierre Jouvelot. The type and effect discipline. In *LICS*. IEEE, 1992.

[126] Ross Tate. The sequential semantics of producer effect systems. In *POPL*, 2013.

[127] David N. Turner and Philip Wadler. Operational interpretations of linear logic. *Theoretical Computer Science*, 227(1-2):231–248, September 1999.

[128] Tarmo Uustalu and Varmo Vene. Signals and comonads. *J. UCS*, 11(7), 2005.

[129] Tarmo Uustalu and Varmo Vene. Comonadic notions of computation. *ENTCS*, 203, 2008.

[130] Vasco T. Vasconcelos, Simon J. Gay, and António Ravara. Type checking a multithreaded functional language with session types. *Theoretical Computer Science*, 368(1–2):64–87, 2006.

[131] Lionel Vaux. The algebraic lambda calculus. *Mathematical Structures in Computer Science; MSCS*, 19(5):1029, 2009.

[132] Edsko Vries, Rinus Plasmeijer, and David M. Abrahamson. Uniqueness typing simplified. In *Implementation and Application of Functional Languages: 19th International Workshop, IFL 2007, Freiburg, Germany, September 27-29, 2007. Revised Selected Papers*, pages 201–218, Berlin, Heidelberg, 2008. Springer-Verlag.

[133] Philip Wadler. Linear types can change the world! In M. Broy and C. Jones, editors, *Progarmming Concepts and Methods*, Sea of Galilee, Israel, April 1990. North Holland. IFIP TC 2 Working Conference.

[134] Philip Wadler. Linear types can change the world! In *IFIP TC 2*, 1990.

[135] Philip Wadler. The essence of functional programming. In *POPL*. ACM, 1992.

[136] Philip Wadler. There's no substitute for linear logic. In *8th International Workshop on the Mathematical Foundations of Programming Semantics*, 1992.

[137] Philip Wadler. A taste of linear logic. In *Mathematical Foundations of Computer Science*, volume 711 of *Lecture Notes in Computer Science*, pages 185–210. Springer-Verlag, 1993.

[138] Philip Wadler. The marriage of effects and monads. In *ICFP*. ACM, 1998.

[139] Philip Wadler. The girard-reynolds isomorphism. In *Theoretical Aspects of Computer Software*, pages 468–491. Springer, 2001.

[140] Philip Wadler. Call-by-value is dual to call-by-name. In *ICFP '03: Proceedings of the eighth ACM SIGPLAN international conference on Functional programming*, pages 189–201, New York, NY, USA, 2003. ACM.

[141] Philip Wadler. Down with the bureaucracy of syntax! Pattern matching for classical linear logic. unpublished manuscript, 2004.

[142] Philip Wadler. Call-by-value is dual to call-by-name–reloaded. In *Term Rewriting and Applications*, pages 185–203. Springer, 2005.

[143] Philip Wadler. Propositions as sessions. In *ICFP*. ACM, 2012.

[144] David Walker. A type system for expressive security policies. In *Proc. 27th ACM Symp. on Principles of Programming Languages (POPL)*, pages 254–267. ACM Press, Jan 2000.

[145] David Walker. *Advanced Topics in Types and Programming Languages*, chapter Substructural Type Systems. MIT Press, 2005.

[146] Keith Wansbrough and Simon Peyton Jones. Once upon a polymorphic type. In *POPL '99: Proceedings of the 26th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 15–28, New York, NY, USA, 1999. ACM.

[147] D.A. Wright and C.A. Baker-Finch. Usage Analysis with Natural Reduction Types. In *Proc. 3rd Workshop on Static Analysis*, 1993.

[148] Nobuko Yoshida, Kohei Honda, and Martin Berger. Linearity and bisimulation. *J. Log. Algebr. Program.*, 72(2):207–238, 2007.

[149] Steve Zdancewic and Andrew C. Myers. Secure information flow via linear continuations. *Higher Order and Symbolic Computation*, 15(2/3):209–234, 2002.

[150] Noam Zeilberger. On the unity of duality. *Annals of Pure and Applied Logic*, 153(1–3):66–96, 2006.

[151] Jianzhou Zhao, Qi Zhang, and Steve Zdancewic. Relational parametricity for polymorphic linear lambda calculus. In *Proceedings of the Eighth ASIAN Symposium on Programming Languages and Systems (APLAS)*, 2010.

[152] Dengping Zhu and Hongwei Xi. Safe Programming with Pointers through Stateful Views. In *Proceedings of the 7th International Symposium on Practical Aspects of Declarative Languages*, pages 83–97, Long Beach, CA, January 2005. Springer-Verlag LNCS vol. 3350.