## Contents

1 Overview
   1.1 Introduction ................................................. 5

2 Intuitionistic Linear Logic
   2.1 The Multiplicative Fragment ................................. 8
      2.1.1 Multiplicative Conjunction ............................. 10
      2.1.2 Multiplicative Conjunction Unit ......................... 11
      2.1.3 Linear Implication .................................. 11
      2.1.4 Summary ........................................... 12
   2.2 The Additive Fragment ...................................... 13
      2.2.1 Additive Products ................................... 13
      2.2.2 Additive Product Unit ................................ 15
      2.2.3 Additive Sums ....................................... 15
      2.2.4 Additive Sum Unit .................................. 16
   2.3 Persistence and the Exponential Modality ................. 16

3 Terms for ILL
   3.1 Evaluations ................................................. 21
   3.2 Call-by-Name Translation of Intuitionistic Logic .......... 27
   3.3 Call-by-Value Translation of Intuitionistic Logic ........ 27

4 Proof Search and ILL
   4.1 Normal Proofs .............................................. 29
   4.2 Sequent Calculus ........................................... 30
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 emphasize 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, $\mu$ calculus, linear lambda calculus, $\pi$ 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 “$= +ELEPHANT = 0$.”

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:

\[
p, q ::= a_i \mid p \land q \mid p \lor q \mid \neg p \mid p \Rightarrow q
\]

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).
a ∈ \{a_1, a_2, \ldots\} \quad p \text{ prop} \quad q \text{ prop} \quad p \text{ prop} \quad q \text{ prop}

a \text{ prop} \quad p \land q \text{ prop} \quad p \lor q \text{ prop}

\frac{p \text{ prop}}{\neg p \text{ prop}} \quad \frac{p \text{ prop} \quad q \text{ prop}}{p \Rightarrow q \text{ 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:

- $A \text{ prop}$ The syntax “$A$” is a proposition
- $A \text{ true}$ The proposition $A$ is true (contingently provable)
- $A \text{ valid}$ The proposition $A$ is valid (provable always)

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

## 2.1 The Multiplicative Fragment

Judgment of the form $A \text{ lin}$ meaning “the proposition $A$ is linearly true.”

We will eventually add another judgment $A \text{ per}$ with the intended meaning that “$A$ is persistently true” or “valid.”

Following Martin-Löf, we must give the judgment $A \text{ lin}$ 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 possession 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:

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

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 $B \text{ lin}$ 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 \( A \) 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 \( A \) is not (necessarily) the same as assuming two instances of it, as in \( A, A \).

These considerations will have to be taken into 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.

\[
\begin{align*}
A \vdash A & \quad \text{HYP} \\
\begin{array}{c}
q \vdash q \\
\end{array} & \quad \Delta \vdash \Delta \\
\end{align*}
\]

An instance of this rule is:

\[
\begin{align*}
q \vdash q & \\
\end{align*}
\]

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:

\[
\begin{align*}
\Delta, A \vdash A & \quad \text{BADHYP} \\
\end{align*}
\]

It doesn’t respect the “efficiency” criterion of our intended interpretation. This plan discards all of the resources in \( \Delta \).

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

**Principle 2.1** (Substitution). If \( \Delta_1 \vdash A \) and \( \Delta_2, A \vdash B \) then \( \Delta_1, \Delta_2 \vdash B \).

• The substitution principle is not a rule of the logic, but instead a structural invariant that justifies the use of a judgment like \( A \) as a hypothesis.

• Note how this takes into account the non-duplicability of resources. In particular, when we write \( \Delta_1, \Delta_2 \) we should interpret this as requiring the combination of resources hypothesized in both \( \Delta_1 \) and \( \Delta_2 \).

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

**Principle 2.2** (Bad Substitution). If \( \Delta \vdash A \) and \( \Delta, A \vdash B \) then \( \Delta \vdash B \).
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 work-
ing 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 con-
sume 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 $\beta$-reductions justified by substi-
tution.)

- **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 $\eta$-expansions.)

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

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

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?

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

This rule (read top to bottom) says that given two plans, one building $A_1$ from $\Delta_1$ and
one building $A_2$ from $\Delta_2$ 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 \(B\).

\[
\Delta_1 \vdash A_1 \otimes A_2 \quad \Delta_2, A_1, A_2 \vdash B
\]

\[\begin{array}{c}
\Delta_1, \Delta_2 \vdash B
\end{array}\]

\(\otimes E\)

Local soundness:

\[
\begin{array}{c}
\Delta_1 \vdash A_1 \\
\Delta_2 \vdash A_2 \\
\Delta_1, \Delta_2 \vdash A_1 \otimes A_2 \\
\Delta_3, A_1, A_2 \vdash B
\end{array}
\]

\[\begin{array}{c}
\Delta_1, \Delta_2, \Delta_3 \vdash B
\end{array}\]

\(\implies R\)

\[\Delta_1, \Delta_2, \Delta_3 \vdash B\]

Local expansion:

\[
\begin{array}{c}
\Delta \vdash A_1 \otimes A_2
\end{array}
\]

\[\implies E\]

\[
\begin{array}{c}
\Delta \vdash A_1 \otimes A_2
\end{array}
\]

\[\Delta \vdash A_1 \otimes A_2\]

\(\otimes E\)

2.1.2 Multiplicative Conjunction Unit

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

\[
\vdash 1
\]

\[
\Delta \vdash 1 \\
\Delta_1 \vdash 1 \\
\Delta_2 \vdash A
\]

\[\Delta_1, \Delta_2 \vdash A\]

\(1E\)

Local soundness:

\[
\begin{array}{c}
\vdash 1
\end{array}
\]

\[\Delta \vdash 1
\]

\[\Delta \vdash 1
\]

\[\Delta \vdash A
\]

\[\Delta \vdash A
\]

\(1E\)

\[\Delta_1 \vdash 1
\]

\[\Delta_2 \vdash A
\]

\[\Delta \vdash A
\]

\(1E\)

\[\Delta \vdash 1
\]

\(1E\)

\[\Delta \vdash 1
\]

Local expansion:

\[
\begin{array}{c}
\Delta \vdash 1
\end{array}
\]

\[\vdash 1
\]

\[\Delta \vdash 1
\]

\(1I\)

2.1.3 Linear Implication

\[
\Delta, A \vdash B
\]

\[\Delta \vdash A \implies B
\]

\(\implies I\)

\[
\Delta \vdash A
\]

\[\Delta_1 \vdash A \\
\Delta_2 \vdash A
\]

\[\Delta_1, \Delta_2 \vdash B
\]

\(\implies E\)
\[
A \vdash A \quad \text{HYP}
\]
\[
\Delta_1 \vdash A_1 \quad \Delta_2 \vdash A_1 \quad \otimes I
\]
\[
\Delta_1, \Delta_2 \vdash A_1 \otimes A_2 \quad \Delta_2, A_1, A_2 \vdash B \quad \otimes E
\]
\[
\Delta_1 \vdash A_1 \otimes A_2 \quad \Delta_2, A_1, A_2 \vdash B \quad \Delta_1, \Delta_2 \vdash B
\]
\[
\vdash \top \quad \top I
\]
\[
\Delta_1 \vdash 1 \quad \Delta_2 \vdash A \quad \Delta_1, \Delta_2 \vdash A \quad 1E
\]
\[
\Delta, A \vdash B \quad \Delta \vdash A \rightarrow B \quad \rightarrow I
\]
\[
\Delta_1 \vdash A \rightarrow B \quad \Delta_2 \vdash A \quad \Delta_1, \Delta_2 \vdash B \quad \rightarrow E
\]

Figure 2.2: The multiplicative fragment of intuitionistic linear logic.

\[
\begin{array}{c}
\mathcal{D} \\
\rightarrow^\mathcal{E} \\
\rightarrow^R \\
\rightarrow^\mathcal{D}
\end{array}
\]

Local Completeness:

\[
\begin{array}{c}
\Delta \vdash A \rightarrow B \\
\Delta \vdash A \rightarrow B \\
\Delta, A \vdash B
\end{array}
\]

\[
\begin{array}{c}
\Delta \vdash A \rightarrow B \\
\Delta, A \vdash B \\
\Delta \vdash A \rightarrow B
\end{array}
\]

2.1.4 Summary

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

Investigate the relationships among the connectives:
\[
\begin{align*}
A_1 \otimes (A_2 \otimes A_3) & \vdash (A_1 \otimes A_2) \otimes A_3 \\
(A_1 \otimes A_2) \otimes A_3 & \vdash A_1 \otimes (A_2 \otimes A_3) \\
1 \otimes A & \vdash A \\
A & \vdash 1 \otimes A \\
A \otimes 1 & \vdash A \\
A & \vdash A \otimes 1 \\
(A_1 \otimes A_2) \multimap B & \vdash A_1 \multimap A_2 \multimap B \\
A_1 \multimap A_2 \multimap B & \vdash (A_1 \otimes A_2) \multimap B
\end{align*}
\]

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:

\[
\frac{}{\vdash q \multimap d \otimes d \otimes n} \text{ \hspace{1em} \text{CHANGE1}}
\]

\[
\frac{}{\vdash q \multimap n \otimes n \otimes n \otimes n \otimes n} \text{ \hspace{1em} \text{CHANGE2}}
\]

Axioms are subtly different than hypotheses because axioms are not “resources” in the sense that they 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:

\[
\frac{D}{q \multimap d \otimes d \otimes n \vdash B}
\]

### 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:

\[
\frac{D_1}{q \vdash d \otimes d \otimes n} \quad \frac{D_2}{q \vdash q}
\]

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

\[
\begin{align*}
\frac{D_1 \quad D_2}{\vdash \cdots \cdots} \quad \frac{q \vdash (d \otimes d \otimes n) \otimes q}{\text{BOGUS}}
\end{align*}
\]
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:

\[
\begin{array}{c}
D_1 \\
\vdash d \otimes d \otimes n \\
D_2 \\
\vdash q
\end{array}
\]

\[
q \vdash (d \otimes d \otimes n) \& q
\]

The introduction and elimination rules for \( \& \) are:

\[
\begin{array}{c}
\Delta \vdash A \\
\Delta \vdash B \\
\end{array}
\&I

\begin{array}{c}
\Delta \vdash A \& B
\end{array}

\begin{array}{c}
\Delta \vdash A \\
\Delta \vdash B
\end{array}
\&E1

\begin{array}{c}
\Delta \vdash A \& B
\end{array}
\&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:

\[
\begin{array}{c}
D_1 \\
\vdash A \\
D_2 \\
\vdash B
\end{array}
\&I

\begin{array}{c}
\Delta \vdash A \& B
\end{array}
\Rightarrow^R

\begin{array}{c}
D_1 \\
\vdash A
\end{array}
\]

Local Soundness 2:

\[
\begin{array}{c}
D_1 \\
\vdash A \\
D_2 \\
\vdash B
\end{array}
\&I

\begin{array}{c}
\Delta \vdash A \& B
\end{array}
\Rightarrow^R

\begin{array}{c}
D_2 \\
\vdash B
\end{array}
\]

Local Expansion:

\[
\begin{array}{c}
\Delta \vdash A \& B
\end{array}
\Rightarrow^E

\begin{array}{c}
\Delta \vdash A
\end{array}
\&E1

\begin{array}{c}
\Delta \vdash A \& B
\end{array}
\]

\[
\begin{array}{c}
\Delta \vdash B
\end{array}
\&E2

\begin{array}{c}
\Delta \vdash A \& B
\end{array}
\]

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:

\[ \frac{}{\Delta \vdash \top} \text{TI} \] (no elimination form)

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

\[ \frac{\mathcal{D}}{\Delta \vdash \top} \implies E \quad \frac{\Delta \vdash \top}{\top \vdash \top} \text{TI} \]

An instance of this is:

\[ \frac{\top \vdash \top}{\text{HYP}} \implies E \quad \frac{\top \vdash \top}{\top \vdash \top} \text{TI} \]

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.

\[ \frac{\Delta \vdash A}{\Delta \vdash A \oplus B} \quad \text{⊕I1} \]

\[ \frac{\Delta \vdash B}{\Delta \vdash A \oplus B} \quad \text{⊕I2} \]

\[ \frac{\Delta_1 \vdash A_1, A_2 \quad \Delta_2 \vdash A_1 \quad \Delta_2 \vdash B \quad \Delta_1 \vdash A_2 \quad \Delta_2 \vdash B}{\Delta_1, \Delta_2 \vdash B} \quad \text{⊕E} \]

Local Soundness 1:

\[ \frac{\mathcal{D}}{\Delta_1 \vdash A_1} \quad \frac{\mathcal{E}_1}{\Delta_2, A_1 \vdash B} \quad \frac{\mathcal{E}_2}{\Delta_2, A_2 \vdash B} \quad \frac{\Delta_1, \Delta_2 \vdash B}{\Delta_1 \vdash A_1 \oplus A_2 \quad \Delta_2 \vdash A_1 \quad \Delta_2 \vdash B \quad \Delta_1 \vdash A_2 \quad \Delta_2 \vdash B} \quad \text{⊕E} \]

\[ \implies R \quad \frac{\mathcal{E}_1'}{\Delta_1 \vdash A_1 \oplus A_2} \quad \frac{\mathcal{E}_2'}{\Delta_2, A_1 \vdash B} \quad \frac{\Delta_2, A_2 \vdash B}{\Delta_1, \Delta_2 \vdash B} \quad \text{⊕E} \]

Local Soundness 1:

\[ \frac{\mathcal{D}}{\Delta_1 \vdash A_2} \quad \frac{\mathcal{E}_1}{\Delta_2, A_1 \vdash B} \quad \frac{\mathcal{E}_2}{\Delta_2, A_2 \vdash B} \quad \frac{\Delta_1, \Delta_2 \vdash B}{\Delta_1 \vdash A_1 \oplus A_2 \quad \Delta_2, A_1 \vdash B} \quad \text{⊕E} \]

\[ \implies R \quad \frac{\mathcal{E}_2'}{\Delta_1 \vdash A_1 \oplus A_2} \quad \frac{\mathcal{E}_1'}{\Delta_2, A_2 \vdash B} \quad \frac{\Delta_2, A_1 \vdash B}{\Delta_1, \Delta_2 \vdash B} \quad \text{⊕E} \]

Local Expansion:
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:

\[
\frac{D}{\Delta \vdash A \oplus B} \quad \Rightarrow_E
\]

\[
\frac{\Delta \vdash A \oplus B}{D\vdash \frac{\Delta \vdash A \oplus B \quad \frac{A \vdash A}{\Delta, A \vdash A \oplus B} \oplus \text{I} \quad \frac{B \vdash B}{\Delta, B \vdash A \oplus B} \oplus \text{I} \quad \frac{\Delta \vdash A \oplus B}{\Delta \vdash A \oplus B} \oplus \text{E}}\quad \text{HYP}
\]

There is still a reasonable notion of local expansion:

\[
\frac{D}{\Delta \vdash 0} \quad \Rightarrow_E \quad \frac{\Delta \vdash 0}{\Delta \vdash 0} \quad \text{0E}
\]

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 \( \Delta \vdash B \text{ lin} \) then \( \Delta, A \text{ lin} \vdash B \text{ lin} \).

**Lemma 2.4 (Contraction Wrong!).** If \( \Delta, A \text{ lin}, A \text{ lin} \vdash B \text{ lin} \) then \( \Delta, A \text{ lin} \vdash B \text{ lin} \).

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 \( A \text{ lin} \), \( A \text{ per} \) means that the hypothesis \( A \) is not an exhaustible resource.

Again following Marin-Löf, we must decide what counts as evidence for the judgment \( A \text{ per} \). 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 \( \cdot \vdash A \text{ lin} \) then \( A \text{ per} \).
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:

\[ A_1 \text{ lin}, A_2 \text{ per}, A_3 \text{ per}, A_4 \text{ lin}, \ldots, A_n \text{ lin} \vdash B \text{ lin} \]

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 \( \Delta \) for the linear contexts and \( \Gamma \) for the persistent ones:

\[
\Delta ::= \cdot \mid A \text{ lin} \mid \Delta_1, \Delta_2 \\
\Gamma ::= \cdot \mid A \text{ per} \mid \Gamma_1, \Gamma_2
\]

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

\[ \Gamma; \Delta \vdash A \text{ lin} \]

Note that the conclusion is still a judgment about the *linearity* of the proposition \( A \). We could consider hypothetical judgments of the form \( \Gamma; \Delta \vdash A \text{ per} \) that define the semantics of *persistence*. However, doing so will turn out to be unnecessary because we categorically defined persistence as \( \cdot \vdash A \text{ lin} \); 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 \( A \text{ per} \) is just a mode of \( A \text{ lin} \), we can use the following rule for persistent hypotheses:

\[
\frac{\Gamma; A \text{ per}; \cdot \vdash A \text{ lin}}{} \text{ Hyp}
\]

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

\[
\frac{\Gamma; A \text{ lin} \vdash A \text{ lin}}{} \text{ Hyp}
\]

It also permits persistent hypotheses to appear in \( \Gamma \), 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 Hyp rule above?*
\[
\frac{\Gamma, A \text{ per}; \Delta, A \text{ lin} \vdash B \text{ lin}}{\Gamma, A \text{ per}; \Delta \vdash B \text{ lin}}
\]

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 \( \Gamma; \Delta_1 \vdash A \text{ lin} \) and \( \Gamma; \Delta_2, A \text{ lin} \vdash B \text{ lin} \) then \( \Gamma; \Delta_1, \Delta_2 \vdash B \text{ lin} \).
- If \( \Gamma; \cdot \vdash A \text{ lin} \) and \( \Gamma, A \text{ per}; \Delta \vdash B \text{ lin} \) then \( \Gamma; \Delta \vdash B \text{ lin} \).

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

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

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

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 \( A \text{ per} \):

\[
\frac{\Gamma; \cdot \vdash A \text{ lin}}{\Gamma; \cdot \vdash !A \text{ lin}} \quad \text{!I}
\]

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

\[
\frac{\Gamma; \Delta_1 \vdash !A \text{ lin} \quad \Gamma, A \text{ per}; \Delta_2 \vdash B}{\Gamma; \Delta_1, \Delta_2 \vdash B} \quad \text{!E}
\]

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

**Local Soundness:**

\[
\frac{D}{\Gamma; \cdot \vdash A \quad \Gamma; \cdot \vdash !A \quad \Gamma, A \text{ per}; \Delta \vdash B \quad \Gamma; \Delta \vdash B}{\Rightarrow R}
\]

Linear Logic Lecture Notes  Draft of: February 25, 2014
Local Expansion:

\[
\frac{\mathcal{D}}{\Gamma; \Delta \vdash !A} \quad \quad \frac{\mathcal{D}}{\Gamma; \Delta \vdash !A} \quad \quad \frac{\mathcal{D}}{\Gamma; \Delta \vdash !A} \quad \quad \frac{\mathcal{D}}{\Gamma; \Delta \vdash !A} \\
\Gamma, A \text{ per}; \vdash !A \quad \quad \Gamma, A \text{ per}; \vdash !A \quad \quad \Gamma, A \text{ per}; \vdash !A
\]

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:

\[
\begin{align*}
\vdash !A & \vdash !A \otimes !A \\
\vdash !A & \vdash 1 \\
\vdash !A & \vdash !!A \\
\vdash !!A & \vdash !A \\
\vdash !(A \& B) & \vdash !(A \otimes B) \\
\vdash !A \otimes !B & \vdash !(A \otimes B) \\
\vdash 1 & \vdash !\top \\
\vdash !\top & \vdash 1
\end{align*}
\]

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):

\[
\begin{align*}
\vdash !(A \otimes B) & \not\vdash !(A \& B) \\
\vdash !(A \otimes B) & \not\vdash !A \otimes !B
\end{align*}
\]
\[
\Gamma; A \text{ lin} \vdash A \text{ lin} \quad \text{HYP}
\]

\[
\Gamma, A \text{ per}; \cdot \vdash A \text{ lin} \quad !\text{HYP}
\]

\[
\Gamma; \Delta_1 \vdash 0 \quad 0\text{E}
\]

\[
\Gamma; \Delta_1 \vdash A \quad \oplus\text{I1}
\]

\[
\Gamma; \Delta_1 \vdash B \quad \oplus\text{I2}
\]

\[
\Gamma; \Delta_1 \vdash A \implies A_2 \quad \Gamma; \Delta_2, A_1 \vdash B \quad \Gamma; \Delta_2, A_2 \vdash B \quad \otimes\text{E}
\]

\[
\Gamma; \cdot \vdash 1 \quad 1\text{I}
\]

\[
\Gamma; \Delta_1 \vdash 1 \quad \Gamma; \Delta_2 \vdash A \quad 1\text{E}
\]

\[
\Gamma; \Delta_1 \vdash A_1 \quad \Gamma; \Delta_2 \vdash A_1 \quad \otimes\text{I}
\]

\[
\Gamma; \Delta_1 \vdash A_1 \otimes A_2 \quad \Gamma; \Delta_2, A_1, A_2 \vdash B \quad \otimes\text{E}
\]

\[
\Gamma; \Delta, A \vdash B \quad \Gamma; \Delta \vdash A \implies B \quad \implies\text{I}
\]

\[
\Gamma; \Delta_1 \vdash A \implies B \quad \Gamma; \Delta_2 \vdash A \quad \implies\text{E}
\]

\[
\Gamma; \cdot \vdash A \text{ lin} \quad \text{II}
\]

\[
\Gamma; \cdot \vdash !A \text{ lin} \quad !\text{I}
\]

\[
\Gamma; \Delta_1 \vdash !A \text{ lin} \quad \Gamma, A \text{ per}; \Delta_2 \vdash B \quad !\text{E}
\]

Figure 2.3: Intuitionistic Linear Logic

Linear Logic Lecture Notes

Draft of: February 25, 2014
Chapter 3

Terms for ILL

3.1 Evaluations

\[ e ::= \text{expressions} \]
\[ \begin{align*}
  & \quad x \\
  & \quad u \\
  & \quad \text{abort } e \\
  & \quad \text{inj}_1 e \\
  & \quad \text{inj}_2 e \\
  & \quad \text{case } e \text{ of inj}_1 \ x. \ e_1 \mid \text{inj}_2 \ y. \ e_2 \\
  & \quad [] \\
  & \quad [e_1 \& e_2] \\
  & \quad \text{prj}_1 e \\
  & \quad \text{prj}_2 e \\
  & \quad \langle \rangle \\
  & \quad \text{let } \langle \rangle = e_1 \text{ in } e_2 \\
  & \quad \langle e_1 \otimes e_2 \rangle \\
  & \quad \text{let } x \otimes y = e_1 \text{ in } e_2 \\
  & \quad \lambda x:A. \ e \\
  & \quad e_1 \ e_2 \\
  & \quad !e \\
  & \quad \text{let } !u = e_1 \text{ in } e_2 \\
  & \quad (e) \\
  & \quad \{e_1/x\}e_2 \\
  & \quad \{e_1/u\}e_2 \\
  & \quad E[e] \\
\end{align*} \]
\( v ::= \) values
\[
\begin{align*}
& x \\
& \text{inj}_1 \: v \\
& \text{inj}_2 \: v \\
& [\: e_1 \& e_2 \:] \\
& \langle \rangle \\
& \langle v_1 \& v_2 \rangle \\
& \lambda x : A. \: e \\
& !e
\end{align*}
\]

\( \Gamma; \Delta \vdash e : A \) Intuitionistic Linear Logic Proof Terms

\[
\begin{align*}
\Gamma; x : A \vdash x : A \\
\Gamma; u : A \in \Gamma \\
\Gamma; \cdot \vdash u : A \\
\Gamma; \Delta_1 \vdash e_1 : A_1 \quad \Gamma; \Delta_2 \vdash e_2 : A_1 \\
\Gamma; \Delta_1, \Delta_2 \vdash (e_1 \& e_2) : A_1 \& A_2 \\
\Gamma; \Delta_1 \vdash e_1 : A_1 \& A_2 \quad \Gamma; \Delta_2, x_1 : A_1, x_2 : A_2 \vdash e_2 : B \\
\Gamma; \Delta_1, \Delta_2 \vdash \text{let } x_1 \& x_2 = e_1 \text{ in } e_2 : B \\
\Gamma; \cdot \vdash \langle \rangle : 1 \\
\Gamma; \Delta_1 \vdash e_1 : 1 \quad \Gamma; \Delta_2 \vdash e_2 : A \\
\Gamma; \Delta_1, \Delta_2 \vdash \text{let } \langle \rangle = e_1 \text{ in } e_2 : A \\
\Gamma; \Delta, x : A \vdash e : B \\
\Gamma; \Delta \vdash \lambda x : A. \: e : A \to B \\
\Gamma; \Delta_1 \vdash e_1 : A \to B \quad \Gamma; \Delta_2 \vdash e_2 : A \\
\Gamma; \Delta_1, \Delta_2 \vdash (e_1 \: e_2) : B \\
\Gamma; \Delta \vdash e_1 : A \quad \Gamma; \Delta \vdash e_2 : B \\
\Gamma; \Delta \vdash [\: e_1 \& e_2 \:] : A \& B \\
\Gamma; \Delta \vdash e : A \& B \\
\Gamma; \Delta \vdash \text{prj}_1 \: e : A \\
\Gamma; \Delta \vdash e : A \& B \\
\Gamma; \Delta \vdash \text{prj}_2 \: e : B \\
\Gamma; \Delta \vdash [] : \top
\end{align*}
\]
\[
\Gamma ; \Delta \vdash e : A \\
\Gamma ; \Delta \vdash \text{inj}_1 e : A \oplus B \\
\Gamma ; \Delta \vdash e : B \\
\Gamma ; \Delta \vdash \text{inj}_2 e : A \oplus B \\
\Gamma ; \Delta_1 \vdash e_0 : A_1 \oplus A_2 \\
\Gamma ; \Delta_2, x_1 : A_1 \vdash e_1 : B \\
\Gamma ; \Delta_2, x_2 : A_2 \vdash e_2 : B \\
\Gamma ; \Delta_1, \Delta_2 \vdash \text{case } e_0 \text{ of } \text{inj}_1 x_1 . e_1 \mid \text{inj}_2 x_2 . e_2 : B \\
\Gamma ; \Delta_1 \vdash e : 0 \\
\Gamma ; \Delta_1, \Delta_2 \vdash \text{abort } e : A \\
\Gamma ; \cdot \vdash e : A \\
\Gamma ; \cdot \vdash !e : !A \\
\Gamma ; \Delta_1 \vdash e_1 : !A \\
\Gamma ; \Delta_1, \Delta_2 \vdash \text{let } !u = e_1 \text{ in } e_2 : B \\
\Gamma ; \Delta_1, \Delta_2 \vdash \text{let } !u = e_1 \text{ in } e_2 : B
\]

\[
E ::= \\
\quad \square \\
\quad \text{inj}_1 E \\
\quad \text{inj}_2 E \\
\quad \text{case } E \text{ of } \text{inj}_1 x . e_1 \mid \text{inj}_2 y . e_2 \\
\quad \text{prj}_1 E \\
\quad \text{prj}_2 E \\
\quad \text{let } \langle \rangle = E \text{ in } e \\
\quad \langle E \otimes e \rangle \\
\quad \langle v \otimes E \rangle \\
\quad \text{let } x \otimes y = E \text{ in } e \\
\quad E e \\
\quad v E \\
\quad \text{let } !u = E \text{ in } e \\
\quad (E) \quad \text{M}
\]

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

\[
\square [e_1] = e_1 \\
E[e_1] = e_2 \\
\boxed{(\text{inj}_1 E)[e_1] = \text{inj}_1 e_2} \\
E[e_1] = e_2 \\
\boxed{(\text{inj}_2 E)[e_1] = \text{inj}_2 e_2} \\
E[e_1] = e_2 \\
\boxed{\text{case } E \text{ of } \text{inj}_1 x . e'_1 \mid \text{inj}_2 y . e'_2[e_1] = \text{case } e_2 \text{ of } \text{inj}_1 x . e'_1 \mid \text{inj}_2 y . e'_2}
\]

Linear Logic Lecture Notes
Draft of: February 25, 2014
\[
E[e_1] = e_2 \\
\text{(prj}_1 E)[e_1] = \text{prj}_1 e_2 \\
E[e_1] = e_2 \\
\text{(prj}_2 E)[e_1] = \text{prj}_2 e_2 \\
E[e_1] = e_2 \\
(\text{let } () = E \text{ in } e)[e_1] = \text{let } () = e_2 \text{ in } e \\
E[e_1] = e_2 \\
\langle E \otimes e \rangle[e_1] = \langle e_2 \otimes e \rangle \\
E[e_1] = e_2 \\
\langle v \otimes E \rangle[e_1] = \langle v \otimes e_2 \rangle \\
E[e_1] = e_2 \\
\langle E e \rangle[e_1] = e_2 e_1 \\
E[e_1] = e_2 \\
\langle v E \rangle[e_1] = v e_2 \\
E[e_1] = e_2 \\
(\text{let } !u = E \text{ in } e)[e_1] = \text{let } !u = e_2 \text{ in } e \\
\{ e/x \} e_1 = e_2 \\
\text{Substitution for Linear Hypotheses} \\
\{ e/x \} x = e \\
\{ e/x \} e = e' \\
\{ e/x \}(\text{inj}_1 e) = \text{inj}_1 e' \\
\{ e/x \} e_0 = e'_0 \\
\{ e/x \}(\text{inj}_1 e_0) = \text{inj}_1 e'_0 \\
\{ e/x \} e_0 = e'_0 \\
\{ e/x \}(\text{case } e_0 \text{ of } \text{inj}_1 y_1. e_1 \mid \text{inj}_2 y_2. e_2) = \text{case } e'_0 \text{ of } \text{inj}_1 y_1. e_1' \mid \text{inj}_2 y_2. e_2' \\
\{ e/x \} e_1 = e'_1 \\
\{ e/x \} e_2 = e'_2 \\
\{ e/x \} e_0 = e'_0 \\
\{ e/x \}(\text{case } e_0 \text{ of } \text{inj}_1 y_1. e_1 \mid \text{inj}_2 y_2. e_2) = \text{case } e_0 \text{ of } \text{inj}_1 y_1. e'_1 \mid \text{inj}_2 y_2. e'_2 \\
\{ e/x \} [ ] = [ ] \\
\{ e/x \} e_0 = e'_0 \\
\{ e/x \}(\text{prj}_1 e_0) = \text{prj}_1 e'_0 \\
\{ e/x \} e_0 = e'_0 \\
\{ e/x \}(\text{prj}_2 e'_0) = \text{prj}_2 e'_0 
\]
\[
\begin{align*}
\{e/x\} e_1 &= e'_1 & \{e/x\} e_2 &= e'_2 \\
\{e/x\} [e_1 \& e_2] &= [e'_1 \& e'_2] \\
\{e/x\} e_1 &= e'_1 \\
\{e/x\} (\text{let } \langle \rangle = e_1 \text{ in } e_2) &= \text{let } \langle \rangle = e'_1 \text{ in } e_2 \\
\{e/x\} e_2 &= e'_2 \\
\{e/x\} (\text{let } \langle \rangle = e_1 \text{ in } e_2) &= \text{let } \langle \rangle = e_1 \text{ in } e'_2 \\
\{e/x\} (e_1 \otimes e_2) &= (e'_1 \otimes e'_2) \\
\{e/x\} e_2 &= e'_2 \\
\{e/x\} (e_1 \otimes e_2) &= (e_1 \otimes e'_2) \\
\{e/x\} e_1 &= e'_1 \\
\{e/x\} (\text{let } y_1 \otimes y_2 = e_1 \text{ in } e_2) &= \text{let } y_1 \otimes y_2 = e'_1 \text{ in } e_2 \\
\{e/x\} e_2 &= e'_2 \\
\{e/x\} (\text{let } y_1 \otimes y_2 = e_1 \text{ in } e_2) &= \text{let } y_1 \otimes y_2 = e_1 \text{ in } e'_2 \\
\{e/x\} (\lambda y : A. e_1) &= \lambda y : A. e'_1 \\
\{e/x\} e_1 &= e'_1 \\
\{e/x\} (e_1 e_2) &= (e'_1 e_2) \\
\{e/x\} e_2 &= e'_2 \\
\{e/x\} (e_1 e_2) &= (e_1 e'_2) \\
\{e/x\} e_1 &= e'_1 \\
\{e/x\} (\text{let } !u = e_1 \text{ in } e_2) &= \text{let } !u = e'_1 \text{ in } e_2 \\
\{e/x\} e_2 &= e'_2 \\
\{e/x\} (\text{let } !u = e_1 \text{ in } e_2) &= \text{let } \langle \rangle = e_1 \text{ in } e'_2 \\
\{e/u\} e_1 &= e_2 & \text{Substitution for Persistent Hypotheses} \\
\{e/u\} u &= e \\
\text{if } u' \neq u \\
\{e/u\} u' &= u' \\
\{e/u\} x &= x \\
\{e/u\} e_0 &= e'_0 \\
\{e/u\} \text{abort } e_0 &= \text{abort } e'_0
\end{align*}
\]
\[
\{ e/u \} e = e' \\
\{ e/u \} (\text{inj}_1 e) = \text{inj}_1 e'
\]

\[
\{ e/u \} e_0 = e'_0 \\
\{ e/u \} (\text{inj}_1 e_0) = \text{inj}_1 e'_0
\]

\[
\{ e/u \} e_0 = e'_0 \\
\{ e/u \} e_1 = e'_1 \\
\{ e/u \} e_2 = e'_2
\]

\[
\{ e/u \} (\text{case } e_0 \text{ of } \text{inj}_1 \ y_1. \ e_1 \ | \ \text{inj}_2 \ y_2. \ e_2) = \text{case } e_0 \text{ of } \text{inj}_1 \ y_1. \ e'_1 \ | \ \text{inj}_2 \ y_2. \ e'_2
\]

\[
\{ e/u \}[\ ] = [] \\
\{ e/u \} e_0 = e'_0 \\
\{ e/u \}(\text{prj}_1 e_0) = \text{prj}_1 e'_0 \\
\{ e/u \} e_0 = e'_0 \\
\{ e/u \}(\text{prj}_2 e'_0) = \text{prj}_2 e'_0
\]

\[
\{ e/u \} e_1 = e'_1 \\
\{ e/u \} e_2 = e'_2
\]

\[
\{ e/u \}[e_1 \& e_2] = [e'_1 \& e'_2]
\]

\[
\{ e/u \}\langle \rangle = \langle \rangle \\
\{ e/u \} e_1 = e'_1 \quad \{ e/u \} e_2 = e'_2
\]

\[
\{ e/u \}(\text{let } \langle \rangle = e_1 \text{ in } e_2) = \text{let } \langle \rangle = e_1 \text{ in } e'_2
\]

\[
\{ e/u \} e_0 = e'_0 \\
\{ e/u \} e_1 = e'_1 \\
\{ e/u \} e_2 = e'_2
\]

\[
\{ e/u \}(\text{let } y_1 \otimes y_2 = e_1 \text{ in } e_2) = \text{let } y_1 \otimes y_2 = e_1 \text{ in } e'_2
\]

\[
\{ e/u \} e_1 = e'_1 \quad x \neq y
\]

\[
\{ e/u \}(\lambda y : A. \ e_1) = \lambda y : A. \ e'_1
\]

\[
\{ e/u \} e_1 = e'_1 \quad \{ e/u \} e_2 = e'_2
\]

\[
\{ e/u \}(e_1 \ e_2) = (e'_1 \ e'_2)
\]

\[
\{ e/u \} e_0 = e'_0 \\
\{ e/u \} !e_0 = !e'_0
\]

\[
\{ e/u \} e_1 = e'_1 \quad \{ e/u \} e_2 = e'_2
\]

\[
\{ e/u \}(\text{let } !u = e_1 \text{ in } e_2) = \text{let } \langle \rangle = e'_1 \text{ in } e'_2
\]

\[
\boxed{e_1 \mapsto e_2} \quad \text{Evaluation Relation}
\]

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

Linear Logic Lecture Notes Draft of: February 25, 2014
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:

\[ \begin{align*}
\text{True} & \implies 1 \\
\text{(A \land B)} & \implies \text{A} \implies \text{B} \\
\text{(A \lor B)} & \implies \text{!A} \lor \text{!B} \\
\text{(A \Rightarrow B)} & \implies \text{!A} \Rightarrow \text{B}
\end{align*} \]

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 Intuitionistic 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 \text{!} constructor is used in two ways: (1) to make \textit{values} persistent (and hence duplicable), and (2) to make \textit{computations} suspended.
\[\begin{align*}
True^v &= 1 \\
(A \land B)^v &= !A^v \& !B^v \\
(A \lor B)^v &= !A^v \oplus !B^v \\
(A \Rightarrow B)^v &= !A^v \rightarrow B^c \\
A^c &= !(A^v)
\end{align*}\]

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 $\beta$-reduction (a.k.a. local soundness reduction rules).

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

$$E ::= \begin{array}{l}
\text{elim forms} \\
| x \\
| u \\
| \text{prj}_1 E \\
| \text{prj}_2 E \\
| E I
\end{array}$$
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
Normal form Checking

\[\begin{align*}
\Gamma; \Delta \vdash I : J & \quad \text{Normal form Checking} \\
\Gamma; \Delta_1 \vdash I_1 : A_1 \uparrow & \quad \Gamma; \Delta_2 \vdash I_2 : A_1 \uparrow \\
\Gamma; \Delta_1, \Delta_2 \vdash (I_1 \otimes I_2) : A_1 \otimes A_2 \uparrow \\
\Gamma; \Delta_1 \vdash E : A_1 \otimes A_2 \downarrow & \quad \Gamma; \Delta_2, x_1 : A_1, x_2 : A_2 \vdash I : B \uparrow \\
\Gamma; \Delta_1, \Delta_2 \vdash \text{let } x_1 \otimes x_2 = E \text{ in } I : B \uparrow \\
\Gamma; \Delta \vdash \text{let } \langle \rangle = E \text{ in } I : A \uparrow \\
\Gamma; \Delta \vdash \text{let } x : A \vdash I : B \uparrow \\
\Gamma; \Delta \vdash \lambda x : A. I : A \multimap B \uparrow \\
\Gamma; \Delta \vdash I_1 : A \uparrow & \quad \Gamma; \Delta \vdash I_2 : B \uparrow \\
\Gamma; \Delta \vdash [I_1 \& I_2] : A \& B \uparrow \\
\Gamma; \Delta \vdash [] : \top \uparrow \\
\Gamma; \Delta \vdash I : A \uparrow \\
\Gamma; \Delta \vdash \text{inj}_1 I : A \oplus B \uparrow \\
\Gamma; \Delta \vdash I : B \uparrow \\
\Gamma; \Delta \vdash \text{inj}_2 I : A \oplus B \uparrow \\
\Gamma; \Delta \vdash E : A_1 \oplus A_2 \downarrow & \quad \Gamma; \Delta_2, x_1 : A_1 \vdash I_1 : B \uparrow & \quad \Gamma; \Delta_2, x_2 : A_2 \vdash I_2 : B \uparrow \\
\Gamma; \Delta_1, \Delta_2 \vdash \text{case } E \text{ of } \text{inj}_1 x_1. I_1 \mid \text{inj}_2 x_2. I_2 : B \uparrow \\
\Gamma; \Delta \vdash E : 0 \downarrow & \quad \Gamma; \Delta \vdash \text{abort } E : A \uparrow \\
\Gamma; \Delta_1, \Delta_2 \vdash E : !A \downarrow & \quad \Gamma; \Delta \vdash !I : !A \uparrow \\
\Gamma; \Delta_1 \vdash E : !A \downarrow & \quad \Gamma; \Delta \vdash \text{let } !u = E \text{ in } I : B \uparrow \\
\Gamma; \Delta_1, \Delta_2 \vdash \text{let } !u = E \text{ in } I : B \uparrow \\
\Gamma; \Delta \vdash E : A \downarrow \\
\Gamma; \Delta \vdash E : A \uparrow
\end{align*}\]

Figure 4.1: Normal form checking rules for ILL.
$$\Gamma; \Delta \vdash E : J$$  

Normal Form Extraction

$$\Gamma ; x : A \vdash x : A \downarrow$$

$$u : A \in \Gamma$$

$$\Gamma ; \cdot \vdash u : A \downarrow$$

$$\Gamma ; \Delta_1 \vdash E : A \rightarrow B \downarrow \quad \Gamma ; \Delta_2 \vdash I : A \uparrow$$

$$\Gamma ; \Delta_1, \Delta_2 \vdash (E I) : B \downarrow$$

$$\Gamma ; \Delta \vdash E : A \& B \downarrow$$

$$\Gamma ; \Delta \vdash \text{prj}_1 E : A \downarrow$$

$$\Gamma ; \Delta \vdash E : A \& B \downarrow$$

$$\Gamma ; \Delta \vdash \text{prj}_2 E : B \downarrow$$

Figure 4.2: Normal form hypothesis extraction rules for ILL.
Augmented Checking

\[
\Gamma; \Delta \vdash J \\
\frac{
\Gamma; \Delta_1 \vdash A_1 \uparrow \quad \Gamma; \Delta_2 \vdash A_1 \uparrow
}{
\Gamma; \Delta_1, \Delta_2 \vdash A_1 \otimes A_2 \uparrow
}
\frac{
\Gamma; \Delta_1 \vdash A_1 \otimes A_2 \downarrow \quad \Gamma; \Delta_2, x_1 : A_1, x_2 : A_2 \vdash B \uparrow
}{
\Gamma; \Delta_1, \Delta_2 \vdash B \uparrow
}
\]

\[
\frac{
\Gamma; \Delta_1 \vdash A \uparrow
}{
\bar{\Gamma}; \cdot \vdash 1 \uparrow
}
\frac{
\Gamma; \Delta_1 \vdash 1 \downarrow \quad \Gamma; \Delta_2 \vdash A \uparrow
}{
\Gamma; \Delta_1, \Delta_2 \vdash A \uparrow
}
\frac{
\Gamma; \Delta, x : A \vdash B \uparrow
}{
\Gamma; \Delta \vdash A \multimap B \uparrow
}
\]

\[
\frac{
\Gamma; \Delta \vdash B \uparrow
}{
\Gamma; \Delta \vdash A \& B \uparrow
}
\frac{
\Gamma; \Delta \vdash \top \uparrow
}{
\Gamma; \Delta \vdash A \uparrow
}
\frac{
\Gamma; \Delta \vdash A \otimes B \uparrow
}{
\Gamma; \Delta \vdash B \uparrow
}
\frac{
\Gamma; \Delta \vdash A \oplus B \uparrow
}{
\Gamma; \Delta \vdash A \downarrow \quad \Gamma; \Delta \vdash B \uparrow
}
\frac{
\Gamma; \Delta_1 \vdash A_1 \oplus A_2 \downarrow \quad \Gamma; \Delta_2, x_1 : A_1 \vdash B \uparrow \quad \Gamma; \Delta_2, x_2 : A_2 \vdash B \uparrow
}{
\Gamma; \Delta_1, \Delta_2 \vdash B \uparrow
}
\frac{
\Gamma; \Delta_1 \vdash 0 \downarrow
}{
\Gamma; \Delta_1, \Delta_2 \vdash A \uparrow
}
\frac{
\Gamma; \cdot \vdash A \uparrow
}{
\Gamma; \cdot \vdash !A \uparrow
}
\frac{
\Gamma; \Delta_1 \vdash !A \downarrow \quad \Gamma, u : A; \Delta_2 \vdash B \uparrow
}{
\Gamma; \Delta_1, \Delta_2 \vdash B \uparrow
}
\frac{
\Gamma; \Delta_1 \vdash A \downarrow
}{
\Gamma; \Delta \vdash A \uparrow
}
\]

Figure 4.3: Augmented rules for proving completeness.

Linear Logic Lecture Notes  Draft of: February 25, 2014
**Augmented Extraction**

\[
\Gamma; \Delta \vdash J \\
\frac{\Gamma; x : A \vdash A \downarrow}{\Gamma; \Delta \vdash A \downarrow} \\
\frac{u : A \in \Gamma}{\Gamma; \Delta \vdash A \downarrow} \\
\frac{\Gamma; \Delta_1 \vdash A \rightarrow B \downarrow \hspace{1cm} \Gamma; \Delta_2 \vdash A \uparrow}{\Gamma; \Delta_1, \Delta_2 \vdash B \downarrow} \\
\frac{\Gamma; \Delta \vdash A \& B \downarrow}{\Gamma; \Delta \vdash A \downarrow} \\
\frac{\Gamma; \Delta \vdash A \& B \downarrow}{\Gamma; \Delta \vdash A \downarrow} \\
\frac{\Gamma; \Delta \vdash A \uparrow}{\Gamma; \Delta \vdash A \uparrow} \\
\frac{\Gamma; \Delta \vdash A \downarrow}{\Gamma; \Delta \vdash A \downarrow}
\]

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

**AugmentedILL Sequents**

\[
\Gamma; \Delta \Rightarrow J \\
\frac{\Delta_1 \Rightarrow A \hspace{1cm} \Delta_2 \Rightarrow A \Rightarrow B}{\Delta_1, \Delta_2 \Rightarrow B}
\]

Figure 4.5: Augmented sequent rules include all of the usual ones, plus cut.
\( \Gamma; \Delta \implies J \) ILL Sequents

\[
\begin{array}{c}
\frac{\Gamma; A \uparrow \implies A \uparrow}{A \in \Gamma} \quad \frac{\Gamma; A, \Delta_1 \implies B}{\Gamma; \Delta_1 \implies B} \\
\frac{\Gamma; \Delta_1 \implies A \quad \Gamma; \Delta_2 \implies B}{\Gamma; \Delta_1, \Delta_2 \implies A \otimes B} \\
\frac{\Gamma; \Delta, A_1, A_2 \implies B}{\Gamma; \Delta, A_1 \otimes A_2 \implies B} \\
\frac{\Gamma; \cdot \implies 1}{\Gamma; \Delta \implies B} \\
\frac{\Gamma; 1, \Delta \implies B}{\Gamma; \Delta, A \implies B} \\
\frac{\Gamma; \Delta \implies A \multimap B}{\Gamma; \Delta \implies A \Darr B} \\
\frac{\Gamma; \Delta_1 \implies A_1 \quad \Gamma; \Delta_2, A_2 \implies B}{\Gamma; A_1 \multimap A_2, \Delta_1, \Delta_2 \implies B} \\
\frac{\Gamma; \Delta \implies A \quad \Gamma; \Delta \implies B}{\Gamma; \Delta \implies A \& B} \\
\frac{\Gamma; \Delta, A_1 \implies B}{\Gamma; \Delta, A_1 \& A_2 \implies B} \\
\frac{\Gamma; \Delta, A_2 \implies B}{\Gamma; \Delta, A_1 \& A_2 \implies B} \\
\frac{\Gamma; \Delta \implies \bot}{\Gamma; \Delta \implies B} \\
\frac{\Gamma; \Delta \implies A \multimap B}{\Gamma; \Delta \implies B} \\
\frac{\Gamma; \Delta \implies A \otimes B}{\Gamma; \Delta \implies A \Darr B} \\
\frac{\Gamma; \Delta, A_1 \implies B \quad \Gamma; \Delta, A_2 \implies B}{\Gamma; \Delta, A_1 \multimap A_2 \implies B} \\
\frac{\Gamma; 0, \Delta \implies B}{\Gamma; \cdot \implies A} \\
\frac{\Gamma; \cdot \implies !A}{\Gamma; A, \Delta \implies B} \\
\frac{\Gamma; A, \Delta \implies B}{\Gamma; !A, \Delta \implies B}
\end{array}
\]
Bibliography


Linear Logic Lecture Notes Draft of: February 25, 2014


[72] Kohei Honda, Vasco T. Vasconcelos, and Makoto Kubo. Language primitives and 

[73] Kohei Honda, Vasco Thudichum Vasconcelos, and Makoto Kubo. Language prim-
atives and type discipline for structured communication-based programming. In 

[74] W. A. Howard. The formulae-as-types notion of construction. In To H. B. Curry: 

[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 

[77] Atsushi Igarashi and Naoki Kobayashi. Resource usage analysis. ACM Trans. Pro-

[78] Limin Jia, Jeffrey A. Vaughan, Karl Mazurak, Jianzhou Zhao, Luke Zarko, Joseph 
Schorr, and Steve Zdancewic. Aura: a programming language for authorization 

[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-

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


229, 2009.


[85] Ugo Dal Lago and Ulrich Schöpp. Functional programming in sublinear space. In 

Linear Logic Lecture Notes Draft of: February 25, 2014


Linear Logic Lecture Notes Draft of: February 25, 2014


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


Linear Logic Lecture Notes Draft of: February 25, 2014


