A Hybrid Formulation of the Ordered Logical Framework

Chris Martens

May 2, 2008

Abstract

The logical framework LF is a powerful tool for encoding and carrying out the metatheory of logics and programming languages in a mechanized way. However, current work on LF has yielded little support for the metatheory of certain kinds of logic that are useful for reasoning about state. One fruitful approach (for the case of linear logic) has been to use hybrid logic, inspired by Kripke modal logic and temporal logic, to give the metareasoning tool access to how the object language context is being manipulated. The goal of this thesis is to apply the same approach to ordered logic, an setting capable of expressing even more constraints.

Contents

1 Introduction

The logical framework LF [8] can be used to encode and reason within and about logical systems defined by the user. It and similar tools are widely used among the programming language research community for carrying out the metatheory of their systems. But as the systems people wish to express become more expressive, so too must the metalogic in order to provide the most natural encodings.

One would like to believe, however, that "more expressive" does not necessarily mean "more complicated". If something seems complex to express, it would be nice to try to generalize the problem, understand why it is so complex, and build an equally general, clear, and powerful solution that not only solves the problem under scrutiny but also many similar problems.

State and resources are examples of such notoriously difficult-to-reason-about concepts. We know of systems that model them well, but LF in its original formulation does not provide an elegant way of reasoning about such systems.

We explore one particular attempt to extend LF with such a general reasoning mechanism. This mechanism is termed *hybrid logic*, inspired by Kripke modal logic and temporal logic. It allows the framework direct access on the metalanguage level to the resources being manipulated in the object language. So far it has been primarily explored for the case of *linear logic* [14].

The goal of this thesis is to apply the same approach to *ordered logic*, an setting capable of expressing even more constraints. In some sense this is just pushing the envelope a step further to see just to what extent in the substructural hierarchy the approach can be applied. It involves redefining only a small fragment of the type theory; namely the simple algebraic calculus of worlds (or labels). We investigate the faithfulness of the embedding of an existing ordered logical gramework into a fragment of this system. We also explore some of the potential practical applications of such a framework.

1.1 Acknowledgments

I would like to thank William Lovas for insightful discussion and LF-related explanations, Jason Reed for helping me get through the grittier technical details, and Frank Pfenning for his excellent guidance and advisorship.

2 Background

2.1 Logical Frameworks

Logical frameworks have many purposes. They provide a way of formalizing the definitions of logics and programming languages that we would otherwise write by hand; moreover, they allow us to prove things about these definitions, such as type safety, in a machine-checkable way. The practice of writing down proofs about formal systems and having a computer check them is called *mechanizing metatheory* [11]. They are sometimes also used as theorem provers, logic programming languages, and educational proof assistants.

The particular logical framework this work extends is LF, which has a particular niche in programming language research, mainly due to the unique way it handles variable binding (that is, as part of the metalanguage) [10]. LF follows the methodology of "judgments-as-types", meaning the judgments of the object language one writes down correspond to *types* in LF, and an object-language derivation of a judgment is an LF term with the type corresponding to that judgment.

2.2 Substructural Logics

Linear and ordered logics are called "substructural" because they lack certain structrual properties a logic is generally assumed to have.

We typically use the metavariables Γ an A, B to denote, respectively, sets of assumptions, and propositions; we use a judgment like $\Gamma \vdash A$ to mean "the set of assumptions in Γ *entail* A" (permit A to be proved). The structural rules, then, are as follows:

- Weakening - If a context Γ proves a proposition A, adding an extra assumption B to Γ still allows us to prove A
- Contraction - If Γ contains two copies of A and proves B, then Γ with only one copy of A can still prove B
- Exchange - If Γ proves A, we can arbitrarily reorder the elements of Γ and still be able to prove A.

What we will call *unrestricted logic* has all three structural properties, and it is what most people would recognize as "ordinary" propositional logic.

Linear logic [6] lacks the first two of these; thus, every assumption in its context must be used exactly once. For this reason it has been called the "logic of resources" – linear logic sequents model exactly what resources must go in for the consequent to come out. Note that in the absense of contraction, having many copies of an assumption around is a different thing from having only one copy around – our context now resembles not a set but a multiset.

Because linear logic has a different notion of truth (that treats it more like a resource), it has a different notion of logical consequence. Given a linear context Δ, $\Delta \vdash_{linear} A$ does not mean the same thing as if the context and the entailment were unrestricted. Linear entailment is captured in the object language with the linear implication connective, $\multimap$. $A \multimap B$ can be thought of as meaning "the resources A produce exactly B" in the same way that $A \rightarrow B$ (where $\rightarrow$ is unrestricted implication) can be thought of as "A being true yields that B is true" (but is not necessarily needed for its proof, and may be used more than once!).

To argue by example that eliminating weakening and contraction models resources, consider a vending machine modeled by the rule “quarter AND quarter ⊸ can of soda”. If instead we had “quarter AND quarter → can of soda”, we could go on to prove “quarter AND quarter → can of soda AND can of soda AND can of soda ...”. We could also notice that we have redundant assumptions (“quarter AND quarter”) and use contraction to get “quarter → can of soda”, or we could weaken to get “quarter AND a million dollars → can of soda”.

Ordered logic, the system actually in the spotlight of this thesis, lacks all three structural rules. Not only must each resource in the context be consumed, but it must be done in order. This yields two implication connectives, ↠ and ↣, read “right implication” and “left implication”. The context more closely resembles a *list* of assumptions rather than a set (as in the unrestricted case) or multiset (as in the linear case). The order in which the pieces of the context are put together matters; for left implication, we make the assumption on the left of the current context, and analogously for right implication. Note that implication associates to the right (by syntactic convention), so $A \rightarrow B \rightarrow C$ actually means $A \rightarrow (B \rightarrow C)$, for any form of implication discussed in this document.

For some intuition, consider propositions as resources again, but imagine an ATM instead of a vending machine: there’s a sequence of steps (insert credit card, enter PIN, select options) that must be performed in order to yield the withdrawal of money. The proposition “insert credit card ↠ enter PIN ↠ select options ↠ cash” enforces this ordering. One could also express it in left implications: “select options ↣ enter pin ↣ insert credit card ↣ cash”, or in some combination of left and right: “enter PIN ↠ insert credit card ↣ select options ↠ cash”. [1]

The two forms of ordered implication are defined as follows in natural deduction style (similar to the formulation presented in Polakow’s thesis):

$$\frac{\Omega, A \vdash B}{\Omega \vdash A \twoheadrightarrow B} \twoheadrightarrow I$$

To introduce a right implication, prove the consequent from the context along with the assumption attached to the right side.

$$\frac{\Omega \vdash A \twoheadrightarrow B \quad \Omega_A \vdash A}{\Omega, \Omega_A \vdash B} \twoheadrightarrow E$$

To eliminate a right implication, prove the antecedent under some context Ω_A, and plug in those assumptions on the right hand side of the context that yields the implication to get the consequent.

$$\frac{A, \Omega \vdash B}{\Omega \vdash A \rightarrowtail B} \rightarrowtail I$$

[1] Note, however, that these different formulations are only equivalent in the sense that they entail each other linearly or unrestrictedly – they do not entail one another in the ordered system.

To introduce a left implication, now we assume the antecedent on the left-hand side.

$$\frac{\Omega \vdash A \rightarrowtail B \quad \Omega_A \vdash A}{\Omega_A, \Omega \vdash B} \rightarrowtail E$$

To eliminate a left implication, plug in the resources used to produce it on the left hand side of the context with the implication to derive the consequent.

It is important to note that although we say ordered logic “lacks” more structural rules, we are actually working toward a *more* expressive system. What we ultimately want is the ability to talk about many different kinds of provability (ordered, linear, unrestricted) and a way to move between them. Once we have “hit bottom”, so to speak, in terms of a restricted structure, we can introduce modalities to mobilize or unconstrain hypotheses, moving them between levels of restrictedness.

As an example of why one might wish for the constraints present in purely ordered logic, consider parsing natural language sentences. (This approach could be used for formal languages as well.) We can write down a rule for sentence formation:

$$\mathsf{np} \twoheadrightarrow \mathsf{vp} \twoheadrightarrow \mathsf{snt}$$

That is, a sentence is a verb phrase to the right of a noun phrase.

We can add a few more rules, using tv to denote a verb with an object (e.g. “eat”) and det to denote a determiner like “a” or “the” :

$$\begin{aligned}
\mathsf{tv} \twoheadrightarrow \mathsf{np} &\twoheadrightarrow \mathsf{vp} \\
\mathsf{det} \twoheadrightarrow \mathsf{np} &\twoheadrightarrow \mathsf{np} \\
\mathsf{dog} &\twoheadrightarrow \mathsf{np} \\
\mathsf{cat} &\twoheadrightarrow \mathsf{np} \\
\mathsf{the} &\twoheadrightarrow \mathsf{det} \\
\mathsf{chased} &\twoheadrightarrow \mathsf{tv}
\end{aligned}$$

The sentence “the dog chased the cat” indeed produces a snt under these rules; the order that words must come in is restricted by the right implication. If one were to think of these as types and write expressions to inhabit them, it should be clear that the unrestricted notion of $\rightarrow$ would allow you to arbitrarily permute, duplicate or discard the argument terms such that all notion of grammar is lost.

One can then define a logic that combines all three reasoning approaches (unrestricted, linear, and ordered) to engender powerful programming or reasoning abilities. Consider the rules for linear implication, written with a $\multimap$:

$$\frac{\Delta, A, \Delta' \vdash B}{\Delta, \Delta' \vdash A \multimap B}$$

$$\frac{\Delta \vdash A \multimap B \quad \Delta_A \vdash A}{\Delta \bowtie \Delta_A \vdash B}$$

The $\bowtie$ here indicates "nondeterministic merge", meaning that the contexts are put together in an order-agnostic manner. Our placement of the hypothesis A in the introduction rule is arbitrary, so we can assume it to be in the middle of some (possibly empty) contexts.

In a combined reasoning system, we have unrestricted hypothesis in Γ, linear ones in Δ, and ordered ones in Ω, so the basic typing judgment turns into something like $\Gamma; \Delta; \Omega \vdash M : A$. The exact specifications of such a system, however, are not important for getting a feel for how combined reasoning works.

If we revisit our natural language parsing example, we can consider relative clauses rel, such as "whom the dog chased" – the word "whom" (for example) precedes what is essentially a sentence missing a noun phrase. We can express this with

$$(\mathsf{np} \multimap \mathsf{snt}) \twoheadrightarrow \mathsf{whom} \rightarrowtail \mathsf{rel}$$

The $\multimap$ indicates that the noun phrase can be missing from anywhere in the sentence; the $\rightarrowtail$ indicates that the relativizer ("whom") should go on the left of the incomplete sentence.

2.3 Ordered LF

Jeff Polakow's thesis work [12] was a conservative extension of LF with the combined reasoning described above. He introduced many examples of applications of such a framework, wherein one can leverage the combined reasoning power of the metalanguage to do unprecedented things with object languages.

One example is simply to use the framework as a logic programming tool in which one could easily write the parsing example above. Other examples of the logic programming tool (called Olli) leverage the fact that the ordered OLF context can behave very naturally as a stack or a queue, including

- translating between λ-calculus terms and deBruijn-indexed terms
- constructing an abstract machine for evaluating Mini-ML
- mergesort

A more compelling example for using OLF as a metatheoretic tool is a CPS (continuation passing style) language analysis. CPS-translated terms reflect the evaluation order of the source language, and in a left-to-right, call-by-value direct style, the CPS terms are such that continuation variables are used linearly, and local parameters to continuations form a stack. Polakow gives an encoding of CPS terms in OLF in which these occurrence invariants are captured implicitly in the ordered types. For a more in-depth explanation of this example, see [13].

2.4 The Problem

OLF is a suitable framework for carrying out the metatheory of other logics and languages using the power of ordered logic. However, it has the unfortunate limitation of being unable to express the statements of *ordered* (and other substructural) metatheory itself. Consider, for example, the cut elimination theorem for ordered logic:

Theorem 1 (Ordered Cut Admissibility)**.** *If* $\Omega_A \vdash A$ *and* $\Omega_1, A, \Omega_2 \vdash C$*, then* $\Omega_1, \Omega_A, \Omega_2 \vdash C$*.*

Encoding this theorem in OLF turns out to be difficult because we have no way to explicitly capture the structure of the context.

It is the eventual hope that the Hybrid LF formulation we are about to present can solve this problem. Even without that, however, the hybrid formulation gives us access to a lot of power not present in OLF. We will visit examples of such possibilities at the end of this document.

3 Language

The language of LF is essentially the simply typed λ-calculus with dependent function types and kinds. Its syntax is given as

$$\begin{array}{rcl} \textit{terms } M, N & ::= & \lambda x.N \mid M N \mid c \mid x \\ \textit{types } A & ::= & \Pi x : A.A' \mid a\ M_1 \ldots M_n \\ \textit{kinds } K & ::= & \Pi x : A.K \mid \text{type} \end{array}$$

where kinds classify type families in the way that types classify terms, and $\Pi x : A.\_$ is the dependent form of types and kinds.

We will use → as an abbreviation for the degenerate case of Π types and kinds where the body does not use the bound variable.

Most of this syntax is not needed to understand the new concepts introduced in hybrid LF; it is here mainly for completeness.

3.1 Hybrid LF

Hybrid logic [3, 4, 2] was inspired by a combination of temporal logic and Kripke modal logic.

The key notion of Hybrid LF is the notion of a *world* or *label* (this document will use these terms interchangeably). The basic typing judgment, rather than $\Gamma \vdash M : A$, becomes $\Gamma \vdash M : A[p]$, where p is a world. This judgment can be read "M has type A under Γ and uses the resources described by p".

Consider the following example: if we have

$$x{:}A, y{:}B, z{:}C \vdash c\ x\ z\ z : D$$

we may wish to express that the term uses the hypothesis x once, z twice, and y no times – indeed, we may wish to express that it uses x and the two zs in the order that it does. Now the idea is to attach a label to the type (D in this case) that does this. The initial thought might be just to say

$$x{:}A, y{:}B, z{:}C \vdash c\ x\ z\ z : D[x, z, z]$$

but this has the problem that the label's well formedness (since it depends on terms) could depend on the current context, and the whole problem in the beginning was the inability to quantify over contexts. So we create a new syntactic class of *worlds*:

$$p, q ::= \alpha \mid p * q \mid \epsilon$$

where ϵ is the empty world and α is a world variable. For now we leave it vague what $*$ should be other than joining two worlds together (this is not the final definition of worlds for the ordered case; this is a pedagogical example of what worlds can look like).

If we want to carry through with our example above, we need some way of attaching worlds to resources. First of all, we need to allow world variable assumptions into our context:

$$\Gamma ::= \ldots \mid \Gamma, \alpha : \mathsf{world}$$

Next, we need a type constructor that internalizes the notion of $A[p]$, called @, defined as

$$\frac{\Gamma \vdash M : A[p]}{\Gamma \vdash M : (A@p)[q]}$$

$$\frac{\Gamma \vdash M : (A@p)[q]}{\Gamma \vdash M : A[p]}$$

Now we can express the notion of a variable in the context being attached to a world by adding

$$\alpha : \mathsf{world}, x : A@\alpha$$

to the context.

Going back to our example, we can now say something like

$$\alpha : \mathsf{world}, \beta : \mathsf{world}, \gamma : \mathsf{world}, x{:}A@\alpha, y{:}B@\beta, z{:}C@\gamma \vdash c\ x\ z\ z : D[\alpha * \gamma * \gamma]$$

to show that the term consumes x, z, and z.

3.2 Ordered Worlds

The missing piece now is what these worlds need to look like, and how they need to behave, to be expressive enough for ordered (and in particular, combined) metareasoning. At the very least we would like a system with the *same* expressiveness as OLF.

In the formulation of HLF for linear logic, the $*$ operator is defined to form a commutative monoid with ϵ; that is, it is associative and commutative, and p, $\epsilon * p$, and $p * \epsilon$ are all equivalent.

Our first attempt at adapting this system to combined reasoning with ordered logic was to introduce a new world connective, $\bullet$, the same as $*$ but noncommutative. This turns out to work for expressing the orderedness of the context, but it does not work well in conjunction with commutative $*$, at least for the purposes of encoding OLF. Instead, our only binary operator is $\bullet$, and we have a unary "mobility" operator on worlds, ¡, which allows the world it's applied to to move around freely in the otherwise ordered structure. Formally, our world grammar is now

$$p, q ::= \alpha \mid p \bullet q \mid \text{¡}p \mid \epsilon$$

and we have the following equivalence axioms:

$$\begin{array}{rcl}
(p \bullet q) \bullet r & \equiv & p \bullet (q \bullet r) \\
p \bullet \epsilon & \equiv & p \\
\epsilon \bullet p & \equiv & p \\
\text{¡}p \bullet q & \equiv & q \bullet \text{¡}p \\
\text{¡}\epsilon & \equiv & \epsilon \\
\text{¡¡}p & \equiv & \text{¡}p \\
\text{¡}(\text{¡}p \bullet \text{¡}q) & \equiv & \text{¡}p \bullet \text{¡}q
\end{array}$$

The first three axioms give associativity of $\bullet$ and unity of ϵ; the rest of the axioms detail how the ¡ operator works. Its defining characteristic is that any world under a ¡ can commute with any world adjacent to it. Additionally, any world under a ¡ that consists of only other worlds under ¡s is equivalent to the same world without the outer ¡.

Now we can encode right, left, linear, and unrestricted implication in our system, in natural deduction style, as follows:

$$\frac{\Gamma, \alpha : \mathsf{world}, x : A@\alpha \vdash M : B[p \bullet \alpha]}{\Gamma \vdash \lambda x.AM : A \twoheadrightarrow B[p]} \twoheadrightarrow I$$

$$\frac{\Gamma, \alpha : \mathsf{world}, x : A@\alpha \vdash M : B[\alpha \bullet p]}{\Gamma \vdash \lambda x.AM : A \rightarrowtail B[p]} \rightarrowtail I$$

$$\frac{\Gamma, \alpha : \mathsf{world}, x : A@(\text{¡}\alpha) \vdash M : B[p \bullet \text{¡}\alpha]}{\Gamma \vdash \lambda x.AM : A \multimap B[p]} \multimap I$$

$$\frac{\Gamma, x : A@\epsilon \vdash M : B[p]}{\Gamma \vdash \lambda x.AM : A \rightarrow B[p]} \rightarrow I$$

$$\frac{\Gamma \vdash M_1 : A \twoheadrightarrow B[p] \quad \Gamma \vdash M_2 : A[q]}{\Gamma \vdash M_1 M_2 : B[p \bullet q]} \twoheadrightarrow E$$

$$\frac{\Gamma \vdash M_1 : A \rightarrowtail B[p] \quad \Gamma \vdash M_2 : A[q]}{\Gamma \vdash M_1 M_2 : B[q \bullet p]} \rightarrowtail E$$

$$\frac{\Gamma \vdash M_1 : A \multimap B[p] \quad \Gamma \vdash M_2 : A[\text{¡}q]}{\Gamma \vdash M_1 M_2 : B[p \bullet \text{¡}q]} \multimap E$$

$$\frac{\Gamma \vdash M_1 : A \rightarrow B[p] \quad \Gamma \vdash M_2 : A[\epsilon]}{\Gamma \vdash M_1 M_2 : B[p]} \rightarrow E$$

These rules are presented here initially in natural deduction style for pedagogical reasons. Later in the document we will present the real formal system in spine form.

In addition to the addition of labels and the @ type operator, we add two more type operators:

$$A ::= \ldots \mid \forall p.A \mid \downarrow p.A$$

Intuitively, the $\forall$ operator quantifies over all worlds, and the $\downarrow$ operator binds the "current" world. They are defined formally as follows.

$$\frac{\Gamma, \alpha : \mathsf{world} \vdash M : A[p]}{\Gamma \vdash M : \forall \alpha.A[p]}$$

$$\frac{\Gamma \vdash M : \forall \alpha.A[p] \quad \Gamma \vdash q : \mathsf{world}}{\Gamma \vdash M : [q/\alpha]A[p]}$$

$$\frac{\Gamma \vdash M : ([p/\alpha]A)[p]}{\Gamma \vdash M : \downarrow \alpha.A[p]}$$

$$\frac{\Gamma \vdash M : \downarrow \alpha.A[p]}{\Gamma \vdash M : [p/\alpha]A[p]}$$

Additionally, we add the universal quantification to kinds:

$$K ::= \ldots \mid \forall p.K$$

defined analogously to the $\forall$ type operator.

This completes the definition of the extension to the logical framework.

It should be noted that the inclusion of the $\forall$ and $\downarrow$ type operators makes the inclusion of $\multimap$, $\twoheadrightarrow$, and $\rightarrowtail$ redundant. We can define them in the following way:

$$A \twoheadrightarrow B \equiv_{def} \forall\alpha \downarrow \beta . A@\alpha \rightarrow B@\beta \bullet \alpha$$

$$A \rightarrowtail B \equiv_{def} \forall\alpha \downarrow \beta . A@\alpha \rightarrow B@\alpha \bullet \beta$$

$$A \multimap B \equiv_{def} \forall\alpha \downarrow \beta . A@¡\alpha \rightarrow B@\beta \bullet ¡\alpha$$

3.3 Spine form

To give a formal account of the system, I am using a piece of logical frameworks machinery known as *spine form* [5]. This is a different way of formulating LF terms wherein all arguments to a function (and projections from pairs, if we had them) are contrained to appear in a *spine*, and all expressions of function applications take the form of a head (variable or constant) applied to a spine. As an example, the term $\lambda x.\lambda y.(c\ x)\ y$ would turn into $\lambda x.\lambda y.c \cdot (x; y)$. The motivation for using spine form draws from the idea of *focussing* in theorem proving ([1], [7]) and allows us to perform eliminations all at once.

3.3.1 Spine form HLF

Our complete type theory with spines is as follows.

Syntax:

$$\begin{array}{lrcl}
\text{worlds} & p, q, r & ::= & \alpha \mid p \bullet q \mid ¡p \mid \epsilon \\
\text{kinds} & K & ::= & \Pi x : A.K \mid \forall\alpha.K \mid \text{type} \\
\text{types} & A & ::= & \Pi x : A.A' \mid a \cdot S \mid \forall\alpha.B \mid \downarrow\alpha.B \mid A@p \\
\text{terms} & M, N & ::= & \lambda x.N \mid MN \mid c \cdot S \mid x \cdot S \\
\text{spines} & S & ::= & () \mid (M; S) \\
\text{contexts} & \Gamma & ::= & \cdot \mid \Gamma, x : A \mid \Gamma, \alpha : \mathsf{world}
\end{array}$$

World formation rules:

$$\frac{\alpha : \mathsf{world} \in \Gamma}{\Gamma \vdash \alpha : \mathsf{world}}$$

$$\frac{\Gamma \vdash p \Leftarrow \mathsf{world}}{\Gamma \vdash ¡p : \mathsf{world}}$$

$$\frac{\Gamma \vdash p \Leftarrow \mathsf{world} \quad \Gamma \vdash q \Leftarrow \mathsf{world}}{\Gamma \vdash p \bullet q : \mathsf{world}}$$

The rest of the type theory is identical to (a subset of) that in [14], reproduced here for posterity.

There are four judgments:

$$\begin{aligned}\Gamma &\vdash p : \mathsf{world}\\ \Gamma &\vdash M : A[p]\\ \Gamma &\vdash A : \text{type}\\ \Gamma &\vdash S : A[p] > C[r]\end{aligned}$$

The first and third check well-formedness of worlds and types, respectively. The second is a type checking judgment parameterized by a world. The fourth is a spine typing judgment that says S is a spine which, if a head of type $A@p$ is applied to it, will produce a term C that uses resources r.

3.3.2 Type checking

We write R to stand for either $x \cdot S$ or $c \cdot S$.

$$\frac{\Gamma \vdash R : a \cdot S[p] \quad S =_\alpha S' \quad p \equiv_E q}{\Gamma \vdash R : a \cdot S'[q]}$$

$$\frac{\Gamma, x : A \vdash M : B[p]}{\Gamma \vdash \lambda x.M : \Pi x{:}A.B[p]}$$

$$\frac{\Gamma, \alpha{:}\mathsf{world} \vdash M : B[p]}{\Gamma \vdash M : \forall \alpha.B[p]}$$

$$\frac{\Gamma \vdash M : [p/a]B[p]}{\Gamma \vdash M : {\downarrow}\alpha.B[p]}$$

$$\frac{\Gamma \vdash M : A[q]}{\Gamma \vdash M : A@q[p]}$$

$$\frac{x{:}A \in \Gamma \quad \Gamma \vdash S : A[\epsilon] > C[r]}{\Gamma \vdash x \cdot S : C[r]}$$

$$\frac{c{:}A \in \Sigma \quad \Gamma \vdash S : A[\epsilon] > C[r]}{\Gamma \vdash c \cdot S : C[r]}$$

3.3.3 Spine Typing

$$\frac{}{\Gamma \vdash () : a \cdot S[p] > a \cdot S[p]}$$

$$\frac{\Gamma \vdash M : A[\epsilon] \quad \Gamma \vdash S : [M/x]B[p] > C[r]}{\Gamma \vdash (M; S) : \Pi x{:}A.B[p] > C[r]}$$

$$\frac{\Gamma \vdash q : \mathsf{world} \quad \Gamma \vdash S : [q/\alpha]B[p] > C[r]}{\Gamma \vdash S : \forall \alpha.B[p] > C[r]}$$

$$\frac{\Gamma \vdash S : [p/\alpha]B[p] > C[r]}{\Gamma \vdash S : \downarrow\alpha.B[p] > C[r]}$$

$$\frac{\Gamma \vdash S : A[q] > C[r]}{\Gamma \vdash S : A@q[p] > C[r]}$$

3.3.4 Spine form OLF

Jeff Polakow's formulation of OLF does not use spines, so here we write down formulation in order to illustrate how spines should work in an ordered framework and also for the sake of the embedding in the next section.

Basic typing judgment for spines in OLF:

$$\Gamma; \Delta; \Omega_1 - \Omega_2 \vdash_{OLF} S : A > B$$

$\Delta_1 \bowtie \Delta_2$ is the nondeterministic merge operation from before.
Rules:
Nil

$$\frac{}{\Gamma; \cdot; \cdot - \cdot \vdash_{OLF} () : a \cdot S > a \cdot S}$$

Ordered right spine cons

$$\frac{\Gamma; \Delta_1; \Omega_{12} \vdash_{OLF} M : A \quad \Gamma; \Delta_2; \Omega_1 - \Omega_2 \vdash_{OLF} S : B > C}{\Gamma; \Delta_1 \bowtie \Delta_2; \Omega_1 - \Omega_{12}\Omega_2 \vdash_{OLF} (M; S) : A \twoheadrightarrow B > C}$$

Ordered left spine cons

$$\frac{\Gamma; \Delta_1; \Omega_{12} \vdash_{OLF} M : A \quad \Gamma; \Delta_2; \Omega_1 - \Omega_2 \vdash_{OLF} S : B > C}{\Gamma; \Delta_1 \bowtie \Delta_2; \Omega_1\Omega_{12} - \Omega_2 \vdash_{OLF} (M; S) : A \rightarrowtail B > C}$$

Linear spine cons

$$\frac{\Gamma; \Delta_1; \cdot \vdash_{OLF} M : A \quad \Gamma; \Delta_2; \Omega_1 - \Omega_2 \vdash_{OLF} S : B > C}{\Gamma; \Delta_1 \bowtie \Delta_2; \Omega_1 - \Omega_2 \vdash_{OLF} (M; S) : A \multimap B > C}$$

Unrestricted spine cons

$$\frac{\Gamma; \cdot; \cdot \vdash_{OLF} M : A \quad \Gamma; \Delta; \Omega_1 - \Omega_2 \vdash_{OLF} S : B > C}{\Gamma; \Delta; \Omega_1 - \Omega_2 \vdash_{OLF} (M; S) : A \to B > C}$$

Ordered variable

$$\frac{\Gamma; \Delta; \Omega_1 - \Omega_2 \vdash_{OLF} S : A > C}{\Gamma; \Delta; \Omega_1 x : A \Omega_2 \vdash_{OLF} x \cdot S \Leftarrow C}$$

Linear variable

$$\frac{\Gamma; \Delta; \Omega_1 - \Omega_2 \vdash_{OLF} S : A > C}{\Gamma; \Delta, x : A; \Omega_1 \Omega_2 \vdash_{OLF} x \cdot S \Leftarrow C}$$

4 Metatheory of Hybrid OLF

This section addresses properties we would like to prove of the framework we've just defined.

4.1 A conservative extension

We would like to show that Hybrid OLF is a *conservative extension* of OLF. To do this, we define an encoding of each OLF connective, and prove this encoding "correct" with respect to OLF. "Correct" means two things: first, that everything derivable in the image of the translation into Hybrid OLF is similarly derivable in the domain (soundness); second, that everything derivable in OLF is similarly derivable in the image of its translation (completeness).

4.1.1 Soundness

We have a translation (not defined here) from any of a set of OLF contexts (Γ, Δ, Ω) to an HLF context (Γ) that defines a world for each variable in the appropriate way, and similarly a translation from HLF contexts to eac OLF context, relative to some world, that takes the relevant world variables and pulls out from the context the term variables that are attached to them. Given these, we can say the following:

Theorem 2 (Soundness). *If a term M translated from OLF has type $A[p]$ in HLF, then in OLF it has type A under a context related to p.*

A formal statement and proof of soundness follows.

First, we need several auxilliary definitions and lemmas.

4.2 Definitions and lemmas for soundness/completeness

4.2.1 Translating contexts

Given OLF contexts Δ and Ω, $\Omega^{@}$ is defined as

$$(x_1 : A_1, \ldots, x_n : A_n)^{@} = (\alpha_1 : \mathsf{world}, x_1 : A_1@\alpha_1, \ldots, \alpha_n : \mathsf{world}, x_n : A_n@\alpha_n)$$

$\Delta^{\text{¡}}$ is defined as

$$(x_1 : A_1, \ldots, x_n : A_n)^{\text{¡}} = (\alpha_1 : \mathsf{world}, x_1 : A_1@\text{¡}\alpha_1, \ldots, \alpha_n : \mathsf{world}, x_n : A_n@\text{¡}\alpha_n)$$

Given $p \rightsquigarrow (\mathbf{o}[p], \mathbf{u}[p])$, the OLF context

$$\Omega\downharpoonleft_{\mathbf{o}[p]} = (x_1 : A_1, \ldots, x_m : A_m)$$

such that $\mathbf{o}[p] \equiv \alpha_1 \ldots \alpha_m$ for distinct i such that $x_i : A_i@\alpha_i \in \Omega$ for every $i \in 1 \ldots m$,
and the OLF context

$$\Delta\downharpoonleft_{\mathbf{u}[p]} = (x_1 : A_1, \ldots, x_m : A_m)$$

such that $\mathbf{u}[p] \equiv \text{¡}\alpha_1 \ldots \text{¡}\alpha_m$ for distinct i such that $x_i : A_i@\text{¡}\alpha_i \in \Delta$ for every $i \in 1 \ldots m$.

4.2.2 Lemmas about worlds

Separation
In the embedding of OLF, we will make use of a canonical separation of worlds into ordered parts and unordered parts. The details of this separation will not be filled in here; we will demand certain axioms of this operation for the proof, with the belief that such an operation can be constructed (although this was not fully fleshed out, as it was not the primary focus of the work).

The basic idea: every world in the codomain of the OLF translation can be "flattened" into $(\mathbf{o}[p], \mathbf{u}[p])$ (an ordered part and an unordered part) where the former can be treated as a list of atoms and the latter as a multiset of mobile atoms. Formally, this means we will require (where $p \rightsquigarrow (\mathbf{o}[p], \mathbf{u}[p])$ means "p separates as ordered part $\mathbf{o}[p]$ and $\mathbf{u}[p]$")

- $p \bullet q \rightsquigarrow (\mathbf{o}[p]\mathbf{o}[q], \mathbf{u}[p]\mathbf{u}[q])$
- $\text{¡}p \rightsquigarrow (\epsilon, \mathbf{o}[p]\mathbf{u}[p])$
- If $p \equiv p'$, then $\mathbf{o}[p] \equiv \mathbf{o}[p']$ up to associativity and unity of nil, and $\mathbf{u}[p] \equiv \mathbf{u}[p']$ up to multiset equality (ACU).
- If $p \rightsquigarrow (o, u)$ and $p' \rightsquigarrow (o', u')$, where $o' \equiv_{AU} o$ and $u \equiv_{ACU} u'$, then $p \equiv p'$.

We will treat $\mathbf{o}[p]$ and $\mathbf{u}[p]$ as opaque functions that project out the appropriate component of the pair that results from separation. The image of the o and u "functions" is opaque but for the fact that we know we can list-append os and multiset union us in the expected way (both represented by adjacency in my notation).

Factorization

If A is an OLF type, and $\Gamma, \Delta^{@}, \Omega^{¡} \vdash S : A[p] > C[r]$, then there exist worlds q_1 and q_2 such that $r \equiv_E q_1 \bullet p \bullet q_2$.

Proof proceeds by induction over the formation of worlds. □

Statement (general):

If

$$\Gamma, \Delta^{¡}, \Omega^{@} \vdash M : A[p]$$

then there exists a world p' such that $p \equiv p'$ and

$$\Gamma; \Delta\lfloor_{\mathbf{u}[p']}; \Omega\lfloor_{\mathbf{o}[p']} \vdash_{OLF} M : A$$

Statement (spines):

If

$$\Gamma, \Delta^{¡}, \Omega^{@} \vdash S : A[p] > B[r]$$

then there exist q_1 and q_2 such that

$$\begin{array}{rcl} \mathbf{o}[r] & \equiv & \mathbf{o}[q_1]\mathbf{o}[p]\mathbf{o}[q_2] \\ \mathbf{u}[r] & \equiv & \mathbf{u}[q_2]\mathbf{u}[p]\mathbf{u}[q_2] \end{array}$$

and

$$\Gamma; \Delta\lfloor_{\mathbf{u}[q_1]\mathbf{u}[q_2]}; \Omega\lfloor_{\mathbf{o}[q_1]} - \Omega\lfloor_{\mathbf{o}[q_2]} \vdash_{OLF} S : A > B$$

Proof by induction on the typing derivation.

Case: Nil

Assume

$$\Gamma, \Delta^{¡}, \Omega^{@} \vdash () : a \cdot S[p] > a \cdot S[p]$$

$$\begin{array}{rcl} \mathbf{o}[p] & \equiv_{AU} & \mathbf{o}[\epsilon]\mathbf{o}[p]\mathbf{o}[\epsilon] \\ \mathbf{u}[p] & \equiv_{ACU} & \mathbf{u}[\epsilon]\mathbf{u}[p]\mathbf{u}[\epsilon] \end{array}$$

and by the OLF nil spine rule,

$$\Gamma; \cdot; \cdot - \cdot \vdash_{OLF} () : a \cdot S > a \cdot S$$

By definition, $\Delta\lfloor_{\epsilon}$ and $\Omega\lfloor_{\epsilon}$ are $\cdot$, so by letting q_1 and q_2 be ϵ, we have

$$\Gamma; \Delta\lfloor_{\mathbf{u}[q_1]\mathbf{u}[q_2]}; \Omega\lfloor_{\mathbf{o}[q_1]} - \Omega\lfloor_{\mathbf{o}[q_2]} \vdash_{OLF} () : a \cdot S > a \cdot S$$

as needed.

Case: Ordered-right spine cons
Assume

$$\Gamma, \Delta^{\mathsf{i}}, \Omega^{@} \vdash (M; S) : \forall\alpha.{\downarrow}\beta.A@\alpha \to B@(\beta \bullet \alpha)[p] > C[r]$$

By inversion,

$$\dfrac{\dfrac{\dfrac{\Gamma, \Delta^{\mathsf{i}}, \Omega^{@} \vdash M \Leftarrow A[s] \quad \Gamma, \Delta^{\mathsf{i}}, \Omega^{@} \vdash S : B[p \bullet s] > C[r]}{\Gamma, \Delta^{\mathsf{i}}, \Omega^{@} \vdash (M; S) : A@s \to B@(p \bullet s)[p] > C[r]}}{\Gamma, \Delta^{\mathsf{i}}, \Omega^{@} \vdash (M; S) : {\downarrow}\beta.A@s \to B@(\beta \bullet s)[p] > C[r]}}{\Gamma, \Delta^{\mathsf{i}}, \Omega^{@} \vdash (M; S) : \forall\alpha.{\downarrow}\beta.A@\alpha \to B@(\beta \bullet \alpha)[p] > C[r]}$$

By inductive hypothesis on the top left sequent, there exists a world s' such that $s \equiv s'$ and

$$\Gamma; \Delta{\downharpoonleft}_{\mathbf{u}[s]}; \Omega{\downharpoonleft}_{\mathbf{o}[s']} \vdash_{OLF} M \Leftarrow A$$

By rule,

$$\begin{array}{rcl} \mathbf{o}[s] & \equiv & \mathbf{o}[s'] \\ \mathbf{u}[s] & \equiv & \mathbf{u}[s'] \end{array}$$

By induction hypothesis on the top right sequent, there exist some q_1' and q_2' such that

$$\Gamma; \Delta{\downharpoonleft}_{\mathbf{u}[q_1']\mathbf{u}[q_2']}; \Omega{\downharpoonleft}_{\mathbf{o}[q_1']} - \Omega{\downharpoonleft}_{\mathbf{o}[q_2']} \vdash_{OLF} S : B > C$$

and

$$\begin{array}{rcl} \mathbf{o}[r] & \equiv & \mathbf{o}[q_1']\mathbf{o}[p \bullet s]\mathbf{o}[q_2'] \\ \mathbf{u}[r] & \equiv & \mathbf{u}[q_1']\mathbf{u}[p \bullet s]\mathbf{u}[q_2'] \end{array}$$

By OLF spine typing rule,

$$\dfrac{\Gamma; \Delta{\downharpoonleft}_{\mathbf{u}[s]}; \Omega{\downharpoonleft}_{\mathbf{o}[s]} \vdash_{OLF} M \Leftarrow A \quad \Gamma; \Delta{\downharpoonleft}_{\mathbf{u}[q_1']\mathbf{u}[q_2']}; \Omega{\downharpoonleft}_{\mathbf{o}[q_1']} - \Omega{\downharpoonleft}_{\mathbf{o}[q_2']} \vdash_{OLF} S : B > C}{\Gamma; \Delta{\downharpoonleft}_{\mathbf{u}[q_1']\mathbf{u}[s]\mathbf{u}[q_2']}; \Omega{\downharpoonleft}_{\mathbf{o}[q_1']} - \Omega{\downharpoonleft}_{\mathbf{o}[s]\mathbf{o}[q_2']} \vdash_{OLF} (M; S) : A \twoheadrightarrow B > C}$$

By rule,

$$\mathbf{u}[s]\mathbf{u}[q_2'] \equiv \mathbf{u}[s \bullet q_2']$$

and

$$\mathbf{o}[s]\mathbf{o}[q_2'] \equiv \mathbf{o}[s \bullet q_2']$$

so by letting $q_1 = q_1'$ and $q_2 = s \bullet q_2'$, we have

$$\mathbf{o}[r] \equiv \mathbf{o}[q_1]\mathbf{o}[p]\mathbf{o}[q_2]$$

and

$$\Gamma; \Delta\downharpoonleft_{\mathbf{u}[q_1]\mathbf{u}[p]\mathbf{u}[q_2]}; \Omega\downharpoonleft_{q_1} - \Omega\downharpoonleft_{q_2} \vdash_{OLF} (M; S) : A \twoheadrightarrow B > C$$

as needed.

Case: Ordered-left spine cons
Assume

$$\Gamma, \Delta^{\mathsf{i}}, \Omega^{@} \vdash (M; S) : \forall\alpha.{\downarrow}\beta.A@\alpha \rightarrow B@(\alpha \bullet \beta)[p] > C[r]$$

By inversion,

$$\cfrac{\cfrac{\cfrac{\Gamma, \Delta^{\mathsf{i}}, \Omega^{@} \vdash M \Leftarrow A[s] \quad \Gamma, \Delta^{\mathsf{i}}, \Omega^{@} \vdash S : B[s \bullet p] > C[r]}{\Gamma, \Delta^{\mathsf{i}}, \Omega^{@} \vdash (M; S) : A@s \rightarrow B@(s \bullet p)[p] > C[r]}}{\Gamma, \Delta^{\mathsf{i}}, \Omega^{@} \vdash (M; S) : {\downarrow}\beta.A@s \rightarrow B@(s \bullet \beta)[p] > C[r]}}{\Gamma, \Delta^{\mathsf{i}}, \Omega^{@} \vdash (M; S) : \forall\alpha.{\downarrow}\beta.A@\alpha \rightarrow B@(\alpha \bullet \beta)[p] > C[r]}$$

By inductive hypothesis on the top left sequent,

$$\Gamma; \Delta\downharpoonleft_{\mathbf{u}[s]}; \Omega\downharpoonleft_{\mathbf{o}[s]} \vdash_{OLF} M \Leftarrow A$$

By induction hypothesis on the top right sequent, there exist some q_1' and q_2' such that

$$\Gamma; \Delta\downharpoonleft_{\mathbf{u}[q_1']\mathbf{u}[q_2']}; \Omega\downharpoonleft_{\mathbf{o}[q_1']} - \Omega\downharpoonleft_{\mathbf{o}[q_2']} \vdash_{OLF} S : B > C$$

and

$$\begin{array}{rcl} \mathbf{o}[r] & \equiv & \mathbf{o}[q_1']\mathbf{o}[s \bullet p]\mathbf{o}[q_2'] \\ \mathbf{u}[r] & \equiv & \mathbf{u}[q_1']\mathbf{u}[s \bullet p]\mathbf{u}[q_2'] \end{array}$$

By OLF spine typing rule,

$$\frac{\Gamma; \Delta\downharpoonleft_{\mathbf{u}[s]}; \Omega\downharpoonleft_{\mathbf{o}[s]} \vdash_{OLF} M \Leftarrow A \quad \Gamma; \Delta\downharpoonleft_{\mathbf{u}[q_1']\mathbf{u}[q_2']}; \Omega\downharpoonleft_{\mathbf{o}[q_1']} - \Omega\downharpoonleft_{\mathbf{o}[q_2']} \vdash_{OLF} S : B > C}{\Gamma; \Delta\downharpoonleft_{\mathbf{u}[q_1']\mathbf{u}[s]\mathbf{u}[q_2']}; \Omega\downharpoonleft_{\mathbf{o}[q_1']\mathbf{o}[s]} - \Omega\downharpoonleft_{\mathbf{o}[q_2']} \vdash_{OLF} (M; S) : A \twoheadrightarrow B > C}$$

By rule,

$$\mathbf{u}[q_1']\mathbf{u}[s] \equiv \mathbf{u}[q_1' \bullet s]$$

and

$$\mathbf{o}[q_1']\mathbf{o}[s] \equiv \mathbf{o}[q_1' \bullet s]$$

so by letting $q_1 = q_1' \bullet s$ and $q_2 = q_2'$, we have

$$\mathbf{o}[r] \equiv \mathbf{o}[q_1]\mathbf{o}[p]\mathbf{o}[q_2]$$

and

$$\Gamma; \Delta\lfloor_{\mathbf{u}[q_1]\mathbf{u}[p]\mathbf{u}[q_2]}; \Omega\lfloor_{q_1} - \Omega\lfloor_{q_2} \vdash_{OLF} (M;S) : A \twoheadrightarrow B > C$$

as needed.

Case: Linear spine cons
Assume

$$\Gamma, \Delta^{¡}, \Omega^{@} \vdash (M;S) : \forall\alpha.{\downarrow}\beta.A@¡\alpha \to B@(\beta \bullet ¡\alpha)[p] > C[r]$$

By inversion

$$\dfrac{\dfrac{\dfrac{\Gamma, \Delta^{¡}, \Omega^{@} \vdash M \Leftarrow A[¡s] \quad \Gamma, \Delta^{¡}, \Omega^{@} \vdash S : B[p \bullet ¡s] > C[r]}{\Gamma, \Delta^{¡}, \Omega^{@} \vdash (M;S) : A@¡s \to B@(p \bullet ¡s)[p] > C[r]}}{\Gamma, \Delta^{¡}, \Omega^{@} \vdash (M;S) : {\downarrow}\beta.A@¡s \to B@(\beta \bullet ¡s)[p] > C[r]}}{\Gamma, \Delta^{¡}, \Omega^{@} \vdash (M;S) : \forall\alpha.{\downarrow}\beta.A@¡\alpha \to B@(\beta \bullet ¡\alpha)[p] > C[r]}$$

By inductive hypothesis on the top left sequent,

$$\Gamma; \Delta\lfloor_{\mathbf{u}[¡s]}; \Omega\lfloor_{\mathbf{o}[¡s]} \vdash_{OLF} M \Leftarrow A$$

$\mathbf{o}[¡s]$ is empty, so

$$\Gamma; \Delta\lfloor_{\mathbf{u}[¡s]}; \cdot \vdash_{OLF} M \Leftarrow A$$

By IH on the top right sequent, there exist worlds q_1' and q_2' such that

$$\Gamma; \Delta\lfloor_{\mathbf{u}[q_1']\mathbf{u}[q_2']}; \Omega\lfloor_{q_1'} - \Omega\lfloor_{q_2'} \vdash_{OLF} S : B > C$$

and

$$\begin{array}{rcl}
\mathbf{o}[r] & \equiv & \mathbf{o}[q_1']\mathbf{o}[p \bullet ¡s]\mathbf{o}[q_2'] \\
 & \equiv & \mathbf{o}[q_1']\mathbf{o}[p]\mathbf{o}[q_2'] \\
 & \equiv & \mathbf{o}[q_1']\mathbf{o}[p]\mathbf{o}[¡s]\mathbf{o}[q_2'] \\
 & \equiv & \mathbf{o}[q_1']\mathbf{o}[p]\mathbf{o}[¡s \bullet q_2'] \\
\mathbf{u}[r] & \equiv & \mathbf{u}[q_1']\mathbf{u}[p \bullet ¡s]\mathbf{u}[q_2'] \\
 & \equiv & \mathbf{u}[q_1']\mathbf{u}[p]\mathbf{u}[¡s]\mathbf{u}[q_2'] \\
 & \equiv & \mathbf{u}[q_1']\mathbf{u}[p]\mathbf{u}[¡s \bullet q_2']
\end{array}$$

By OLF linear spine cons rule,

$$\frac{\Gamma; \Delta\!\downharpoonright_{\mathbf{u}[¡s]}; \cdot \vdash_{OLF} M \Leftarrow A \quad \Gamma; \Delta\!\downharpoonright_{\mathbf{u}[q_1']\mathbf{u}[q_2']}; \Omega\!\downharpoonright_{q_1'} - \Omega\!\downharpoonright_{q_2'} \vdash_{OLF} S : B > C}{\Gamma; \Delta\!\downharpoonright_{\mathbf{u}[¡s]} \bowtie \Delta\!\downharpoonright_{\mathbf{u}[q_1']\mathbf{u}[q_2']}; \Omega\!\downharpoonright_{\mathbf{o}[q_1']} - \Omega\!\downharpoonright_{\mathbf{o}[q_2']} \vdash_{OLF} (M; S) : A \multimap B > C}$$

By the definition of $\Delta\!\downharpoonright$,

$$\Delta\!\downharpoonright_{\mathbf{u}[¡s]} \bowtie \Delta\!\downharpoonright_{\mathbf{u}[q_1']\mathbf{u}[q_2']} = \Delta\!\downharpoonright_{\mathbf{u}[¡s]\mathbf{u}[q_1']\mathbf{u}[q_2']}$$

and

$$\mathbf{u}[¡s]\mathbf{u}[q_1']\mathbf{u}[q_2'] \equiv \mathbf{u}[q_1']\mathbf{u}[¡s]\mathbf{u}[q]_2' \equiv \mathbf{u}[q_1']\mathbf{u}[¡s \bullet q_2']$$

so we have

$$\Gamma; \Delta\!\downharpoonright_{\mathbf{u}[q_1']\mathbf{u}[¡s\bullet q_2']}; \Omega\!\downharpoonright_{\mathbf{o}[q_1']} - \Omega\!\downharpoonright_{\mathbf{o}[¡s\bullet q_2']} \vdash_{OLF} (M; S) : A \multimap B > C$$

Set $q_1 = q_1'$ and $q_2 = ¡s \bullet q_2'$:

$$\Gamma; \Delta\!\downharpoonright_{\mathbf{u}[q_1]\mathbf{u}[q_2]}; \Omega\!\downharpoonright_{\mathbf{o}[q_1]} - \Omega\!\downharpoonright_{\mathbf{o}[q_2]} \vdash_{OLF} (M; S) : A \multimap B > C$$

$$\begin{array}{rcl} \mathbf{o}[r] & \equiv & \mathbf{o}[q_1]\mathbf{o}[p]\mathbf{o}[q_2] \\ \mathbf{u}[r] & \equiv & \mathbf{u}[q_1]\mathbf{u}[p]\mathbf{u}[q_2] \end{array}$$

as needed.

Case: Ordinary spine cons

Assume

$$\Gamma, \Delta^{¡}, \Omega^{@} \vdash (M; S) : A \to B[p] > C[r]$$

By inversion,

$$\frac{\Gamma, \Delta^{¡}, \Omega^{@} \vdash M \Leftarrow A[\epsilon] \quad \Gamma, \Delta^{¡}, \Omega^{@} \vdash S : B[\epsilon] > C[r]}{\Gamma, \Delta^{¡}, \Omega^{@} \vdash (M; S) : A \to B[p] > C[r]}$$

By IH on the top left sequent, since $\mathbf{o}[\epsilon]$ and $\mathbf{u}[\epsilon]$ are nil and $\Delta\!\downharpoonright_{\epsilon}$ and $\Omega\!\downharpoonright_{\epsilon}$ are both $\cdot$,

$$\Gamma; \cdot; \cdot \vdash_{OLF} M \Leftarrow A$$

By IH on the top right sequent, there exist q_1' and q_2' such that

$$\begin{array}{rcl} \mathbf{o}[r] & \equiv & \mathbf{o}[q_1']\mathbf{o}[\epsilon]\mathbf{o}[q_2'] \\ & \equiv & \mathbf{o}[q_1']\mathbf{o}[q_2'] \\ \mathbf{u}[r] & \equiv & \mathbf{u}[q_1']\mathbf{u}[\epsilon]\mathbf{u}[q_2'] \\ & \equiv & \mathbf{u}[q_1']\mathbf{u}[q_2'] \end{array}$$

and

$$\Gamma; \Delta\downharpoonleft_{\mathbf{u}[q_1']\mathbf{u}[q_2']}; \Omega\downharpoonleft_{q_1'} - \Omega\downharpoonleft_{q_2'} \vdash_{OLF} S : B > C$$

By OLF ordinary spine cons rule, and by setting $q_1 = q_1'$ and $q_2 = q_2'$, we get

$$\Gamma; \Delta\downharpoonleft_{\mathbf{u}[q_1]\mathbf{u}[q_2]}; \Omega\downharpoonleft_{q_1} - \Omega\downharpoonleft_{q_2} \vdash_{OLF} (M; S) : A \to B > C$$

as needed.

Case: Ordered lambda right

$$\frac{\Gamma, \Delta^{\text{¡}}, \Omega^{@}, \alpha_x : \mathsf{world}, x : A@\alpha_x \vdash M \Leftarrow B[p \bullet \alpha_x]}{\Gamma, \Delta^{\text{¡}}, \Omega^{@} \vdash \lambda x.M \Leftarrow A \twoheadrightarrow B[p]}$$

By inductive hypothesis,

$$\Gamma; \Delta\downharpoonleft_{\mathbf{u}[p\bullet\alpha_x]}; \Omega\downharpoonleft_{\mathbf{o}[p\bullet\alpha_x]} \vdash_{OLF} M \Leftarrow B$$

Since $\mathbf{u}[\alpha_x]$ is nil and $\Omega\downharpoonleft_{\mathbf{o}[p\bullet\alpha_x]}$ is $\Omega\downharpoonleft_{\mathbf{o}[p]}, x : A$, we have

$$\frac{\Gamma; \Delta\downharpoonleft_{\mathbf{u}[p]}; \Omega\downharpoonleft_{\mathbf{o}[p]}, x : A \vdash_{OLF} M \Leftarrow B}{\Gamma; \Delta\downharpoonleft_{\mathbf{u}[p]}; \Omega\downharpoonleft_{\mathbf{o}[p]} \vdash_{OLF} \lambda x.M \Leftarrow A \twoheadrightarrow B}$$

as needed.

Case: Ordered lambda left

$$\frac{\Gamma, \Delta^{\text{¡}}, \Omega^{@}, \alpha_x : \mathsf{world}, x : A@\alpha_x \vdash M \Leftarrow B[\alpha_x \bullet p]}{\Gamma, \Delta^{\text{¡}}, \Omega^{@} \vdash \lambda x.M \Leftarrow A \rightarrowtail B[p]}$$

By inductive hypothesis,

$$\Gamma; \Delta\downharpoonleft_{\mathbf{u}[\alpha_x\bullet p]}; \Omega\downharpoonleft_{\mathbf{o}[\alpha_x\bullet p]} \vdash_{OLF} M \Leftarrow B$$

Since $\mathbf{u}[\alpha_x]$ is nil and $\Omega\downharpoonleft_{\mathbf{o}[\alpha_x\bullet p]}$ is $x : A, \Omega\downharpoonleft_{\mathbf{o}[p]}$, we have

$$\frac{\Gamma; \Delta\downharpoonleft_{\mathbf{u}[p]}; \Omega\downharpoonleft_{x:A,\mathbf{o}[p]} \vdash_{OLF} M \Leftarrow B}{\Gamma; \Delta\downharpoonleft_{\mathbf{u}[p]}; \Omega\downharpoonleft_{\mathbf{o}[p]} \vdash_{OLF} \lambda x.M \Leftarrow A \rightarrowtail B}$$

as needed.

Case: Linear lambda

$$\frac{\Gamma, \Delta^{\mathsf{i}}, \alpha_x : \mathsf{world}, x : A@\text{¡}\alpha_x, \Omega^{@} \vdash M \Leftarrow B[p \bullet \text{¡}\alpha_x]}{\Gamma, \Delta^{\mathsf{i}}, \Omega^{@} \vdash \lambda x.M \Leftarrow A \multimap B[p]}$$

By inductive hypothesis,

$$\Gamma; \Delta{\downharpoonright}_{\mathbf{u}[p\bullet\alpha_x]}; \Omega{\downharpoonright}_{\mathbf{o}[p\bullet\text{¡}\alpha_x]} \vdash_{OLF} M \Leftarrow B$$

Since $\mathbf{o}[\text{¡}\alpha_x]$ is nil and $\Delta{\downharpoonright}_{\mathbf{u}[p\bullet\text{¡}\alpha_x]}$ is $\Delta{\downharpoonright}_{\mathbf{u}[p]}, x : A$, we have

$$\frac{\Gamma; \Delta{\downharpoonright}_{\mathbf{u}[p]}, x : A; \Omega{\downharpoonright}_{\mathbf{o}[p]} \vdash_{OLF} M \Leftarrow B}{\Gamma; \Delta{\downharpoonright}_{\mathbf{u}[p]}; \Omega{\downharpoonright}_{\mathbf{o}[p]} \vdash_{OLF} \lambda x.M \Leftarrow A \multimap B}$$

as needed.

Case: Ordinary lambda

$$\frac{\Gamma, x : A, \Delta^{\mathsf{i}}, \Omega^{@} \vdash M \Leftarrow B[p]}{\Gamma, \Delta^{\mathsf{i}}, \Omega^{@} \vdash \lambda x.M \Leftarrow A \to B[p]}$$

By inductive hypothesis,

$$\frac{\Gamma, x : A; \Delta{\downharpoonright}_{\mathbf{u}[p]}; \Omega{\downharpoonright}_{\mathbf{o}[p]} \vdash_{OLF} M \Leftarrow B}{\Gamma; \Delta{\downharpoonright}_{\mathbf{u}[p]}; \Omega{\downharpoonright}_{\mathbf{o}[p]} \vdash_{OLF} \lambda x.M \Leftarrow A \to B}$$

as needed.

Case: Ordered variable

$$\frac{x : A@\alpha_x \in \emptyset^{@} \qquad \dfrac{\Gamma, \Delta^{\mathsf{i}}, \Omega^{@} \vdash S : A[\alpha_x] > C[r]}{\Gamma, \Delta^{\mathsf{i}}, \Omega^{@} \vdash S : A@\alpha_x[\epsilon] > C[r]}}{\Gamma, \Delta^{\mathsf{i}}, \Omega^{@} \vdash x \cdot S \Rightarrow C[r]}$$

By IH, there exist q_1 and q_2 such that

$$\begin{array}{rcl} \mathbf{o}[r] & \equiv & \mathbf{o}[q_1]\mathbf{o}[\alpha_x]\mathbf{o}[\bullet q_2] \\ \mathbf{u}[r] & \equiv & \mathbf{u}[q_1]\mathbf{u}[\alpha_x]\mathbf{u}[q_2] \end{array}$$

and

$$\frac{\Gamma; \Delta{\downharpoonright}_{\mathbf{u}[q_1]\mathbf{u}[q_2]}; \Omega{\downharpoonright}_{\mathbf{o}[q_1]} - \Omega{\downharpoonright}_{\mathbf{o}[q_2]} \vdash_{OLF} S : A > C}{\Gamma; \Delta{\downharpoonright}_{\mathbf{u}[q_1]\mathbf{u}[q_2]}; \Omega{\downharpoonright}_{\mathbf{o}[q_1]} x : A \Omega{\downharpoonright}_{\mathbf{o}[q_2]} \vdash_{OLF} x \cdot S \Rightarrow C}$$

$\Omega\downharpoonright_{\mathbf{o}[q_1]} x : A \Omega\downharpoonright_{\mathbf{o}[q_2]}$ is $\Omega\downharpoonright_{\mathbf{o}[r]}$, and $\Delta\downharpoonright_{\mathbf{u}[q_1]\mathbf{u}[q_2]}$ is $\Delta\downharpoonright_{\mathbf{u}[r]}$ because $\mathbf{u}[\alpha_x]$ is nil, so this is as needed.

Case: Linear variable

$$\frac{x : A@\text{¡}\alpha_x \in \Delta^{\text{¡}} \quad \dfrac{\Gamma, \Delta^{\text{¡}}, \Omega^{@} \vdash S : A[\text{¡}\alpha_x] > C[r]}{\Gamma, \Delta^{\text{¡}}, \Omega^{@} \vdash S : A@\text{¡}\alpha_x[\epsilon] > C[r]}}{\Gamma, \Delta^{\text{¡}}, \Omega^{@} \vdash x \cdot S \Rightarrow C[r]}$$

By IH, there exist q_1 and q_2 such that

$$\begin{array}{rcl}
\mathbf{o}[r] & \equiv & \mathbf{o}[q_1]\mathbf{o}[\text{¡}\alpha_x]\mathbf{o}[q_2] \\
 & \equiv & \mathbf{o}[q_1]\mathbf{o}[q_2] \\
 & \equiv & \mathbf{o}[q_1]\mathbf{o}[q_2 \bullet \text{¡}\alpha_x] \\
\mathbf{u}[r] & \equiv & \mathbf{u}[q_1]\mathbf{u}[\text{¡}\alpha_x]\mathbf{u}[q_2] \\
 & \equiv & \mathbf{u}[q_1]\mathbf{u}[q_2]\mathbf{u}[\text{¡}\alpha_x] \\
 & \equiv & \mathbf{u}[q_1]\mathbf{u}[q_2 \bullet \text{¡}\alpha_x]
\end{array}$$

and

$$\frac{\Gamma; \Delta\downharpoonright_{\mathbf{u}[q_1]\mathbf{u}[q_2]}; \Omega\downharpoonright_{\mathbf{o}[q_1]} - \Omega\downharpoonright_{\mathbf{o}[q_2]} \vdash_{OLF} S : A > C}{\Gamma; \Delta\downharpoonright_{\mathbf{u}[q_1]\mathbf{u}[q_2]}, x : A; \Omega\downharpoonright_{\mathbf{o}[q_1]}\Omega\downharpoonright_{\mathbf{o}[q_2]} \vdash_{OLF} x \cdot S \Rightarrow C}$$

Since $\Delta\downharpoonright_{\mathbf{u}[q_1]\mathbf{u}[q_2]}, x : A$ is $\Delta\downharpoonright_{r}$, this is as needed.

Case: Ordinary variable

$$\frac{x : A \in \Gamma \quad \Gamma, \Delta^{\text{¡}}, \Omega^{@} \vdash S : A[\epsilon] > C[r]}{\Gamma, \Delta^{\text{¡}}, \Omega^{@} \vdash x \cdot S \Rightarrow C[r]}$$

By IH, there exist worlds q_1 and q_2 such that

$$\begin{array}{rcl}
\mathbf{o}[r] & \equiv & \mathbf{o}[q_1]\mathbf{o}[\epsilon]\mathbf{o}[q_2] \\
 & \equiv & \mathbf{o}[q_1]\mathbf{o}[q_2] \\
 & \equiv & \mathbf{o}[q_1 \bullet q_2] \\
\mathbf{u}[r] & \equiv & \mathbf{u}[q_1]\mathbf{u}[\epsilon]\mathbf{u}[q_2] \\
 & \equiv & \mathbf{u}[q_1]\mathbf{u}[q_2] \\
 & \equiv & \mathbf{u}[q_1 \bullet q_2]
\end{array}$$

and

$$\frac{\Gamma; \Delta\downharpoonleft_{\mathbf{u}[q_1]\mathbf{u}[q_2]}; \Omega\downharpoonleft_{\mathbf{o}[q_1]} - \Omega\downharpoonleft_{\mathbf{o}[q_2]} \vdash_{OLF} S : A > C}{\Gamma; \Delta\downharpoonleft_{\mathbf{u}[q_1]\mathbf{u}[q_2]}; \Omega\downharpoonleft_{\mathbf{o}[q_1]\mathbf{o}[q_2]} \vdash_{OLF} x \cdot S \Rightarrow C}$$

Since

$$\begin{array}{rcl} \mathbf{o}[r] & \equiv & \mathbf{o}[q_1 \bullet q_2] \\ \mathbf{u}[r] & \equiv & \mathbf{u}[q_1 \bullet q_2] \end{array}$$

by rule, $r \equiv q_1 \bullet q_2$, so this is as needed.

This concludes the proof of soundness. □

4.2.3 Completeness

Given similar context translations and similar caveats about generality as above:

Theorem 3 (Completeness)**.** *Given a term M well-typed with A in OLF, the translation of M is well-typed as $A[p]$ in HLF, for all p correctly constructed from the OLF context.*

Statement (general):

If

$$\Gamma; \Delta; \Omega \vdash_{OLF} M : A$$

then

$$\Gamma, \Delta^{\text{¡}}, \Omega^{@} \vdash M : A[p]$$

where $\mathbf{o}[p] = \alpha_\Omega$ and $\mathbf{u}[p] = \alpha_\Delta$.

Statement (spines):

If

$$\Gamma; \Delta; \Omega_1 - \Omega_2 \vdash_{OLF} S : A > C$$

then for all Δ', Ω_1', Ω_2' extending $\Delta^{\text{¡}}$, $\Omega_1^{@}$, $\Omega_2^{@}$ respectively, and for any world p such that $\Delta', \Omega_1', \Omega_2' \vdash p\ \mathsf{world}$, there exsists a q such that

$$\mathbf{o}[q] \equiv \alpha_{\Omega_1}\mathbf{o}[p]\alpha_{\Omega_2}$$

and

$$\mathbf{u}[q] \equiv \mathbf{u}[p]\mathbf{u}[\alpha_Delta]$$

such that

$$\Gamma, \Delta', \Omega' \vdash S : A[p] > C[q]$$

Proof by induction on the typing derivation.

The proof of completeness is unfinished; a later version of this document will contain it in full.

4.3 Other Theorems

We have conjectured decidability of typechecking, which is to say that the typechecking process will always terminate. It is clear that this holds for at least the fragment of our system in the image of the translation from OLF, since typechecking of OLF has been proven decidable. The remainder is conjectured to be decidable, given that a similar problem, namely the word problem for semigroups, is decidable, but whether this works in the expected way is not yet known.

5 Conclusion

5.1 Contributions

I have presented a formulation of an ordered logical framework within the hybrid logical framework, constructing evidence for the feasibility of the extension of the hybrid logic approach to more expressive settings. I have worked out the necessary algebra on worlds for this extension and proven that they provide a faithful embedding of OLF into the framework. I have worked on several examples to show the potential applications of such a framework.

5.2 Future Work

We have shown that our hybrid logical framework is a conservative extension of OLF, and thus it has the full power to express the OLF examples described in the introduction. However, we would also like to know that we have gained something over OLF. In particular, there are objects of this framework outside of the image of the translation of OLF, given the ¡ operator on worlds, the $\downarrow$ operator on types and then $\forall$ operators on types and kinds.

In particular, I am hoping to explore how this framework can express something akin to the memory layout system described in [9]. This system uses a mobility operator, similar to ¡ but on types, to express pointers into the heap – data whose address in memory is arbitrary – whereas the adjacency properties given of the ordered fuse ($\bullet$) allow representation of structs and ordered pairs whose adjacency in memory is critical.

Finally, the main projected goal for future work is to describe the metatheory of ordered logic in this system.

References

[1] J. M. Andreoli. Logic programming with focusing proofs in linear logic. *Journal of Logic and Computation*, 2(3):297–347, 1992.

[2] C. Areces, P. Blackburn, and M. Marx. Hybrid logics: Characterization, interpolation and complexity. *Journal of Symbolic Logic*, 66(3):977–1010, 2001.

[3] T. Braüner and V. de Paiva. Towards constructive hybrid logic. *Elec. Proc. of Methods for Modalities*, 3, 2003.

[4] Torben Braüner and Valeria de Paiva. Intuitionistic hybrid logic. To appear., 2006.

[5] Iliano Cervesato and Frank Pfenning. A linear spine calculus. Technical Report CMU-CS-97-125, Department of Computer Science, Carnegie Mellon University, April 1997.

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

[7] J.Y. Girard. Locus Solum: From the rules of logic to the logic of rules. *Mathematical Structures in Computer Science*, 11(03):301–506, 2001.

[8] Robert Harper, Furio Honsell, and Gordon Plotkin. A framework for defining logics. *Journal of the Association for Computing Machinery*, 40(1):143–184, January 1993.

[9] Leaf Petersen, Robert Harper, Karl Crary, and Frank Pfenning. A type theory for memory allocation and data layout. In G. Morrisett, editor, *Conference Record of the 30th Annual Symposium on Principles of Programming Languages (POPL'03)*, pages 172–184, New Orleans, Louisiana, January 2003. ACM Press. Extended version available as Technical Report CMU-CS-02-171, December 2002.

[10] Frank Pfenning and Conal Elliott. Higher-order abstract syntax. In *Proceedings of the ACM SIGPLAN '88 Symposium on Language Design and Implementation*, pages 199–208, Atlanta, Georgia, June 1989.

[11] Frank Pfenning and Carsten Schürmann. System description: Twelf — a meta-logical framework for deductive systems. In H. Ganzinger, editor, *Proceedings of the 16th International Conference on Automated Deduction (CADE-16)*, pages 202–206, Trento, Italy, July 1999. Springer-Verlag LNAI 1632.

[12] Jeff Polakow. *Ordered Linear Logic and Applications.* PhD thesis, School of Computer Science, Carnegie Mellon University, May 2001. Available as Technical Report CMU-CS-01-152.

[13] Jeff Polakow and Frank Pfenning. Properties of terms in continuation-passing style in an ordered logical framework. In Joëlle Despeyroux, editor, *2nd Workshop on Logical Frameworks and Meta-languages (LFM'00)*, Santa Barbara, California, June 2000. Proceedings available as INRIA Technical Report.

[14] Jason Reed. A hybrid metalogical framework. Thesis Proposal Working Draft, 2007.