Lecture Notes on Proof Terms

15-836: Substructural Logics
Frank Pfenning

Lecture 4
September 7, 2023

1 Introduction

In the last lecture we have seen the fundamental properties of cut and identity elimination. They guarantee the *harmony* of the right and left rules for the connectives provides us with a *proof-theoretic semantics*: the meaning of a proposition is given by its cut-free proofs. This is a valid semantic point of view since the left and right rules only decompose propositions into their constituents so we don't have to look "outside" for their meaning. To put it another way: the proof-theoretic semantics is compositional.

In intuitionistic logics, therefore, proofs are the primary carriers of meaning. We therefore should think of them as being "first-class", which is not usually the case in classical mathematics: proofs are carried out, of course, but the study of their *formal structure* is not so important. For example, you are unlikely to see a notation for mathematical proofs as objects.

Besides the fact that proofs fundamentally provide meaning to the propositions, they also have a central *computational role*. We will explore this in the next lecture. So we need notations so we can write out proofs, reason about them, execute them, etc. In this lecture we take a neutral point of view: all we want to do is to compactly record the structure of proofs in the form of terms. These terms should have enough information to unwind them into the two-dimensional proofs we are used to, and vice versa.

These desiderata don't change from structural to substructural logics, although the process of checking them may change substantially.

2 Annotating the Sequent

As in the last lecture, we will focus here on ordered logic but the approach itself is quite general. Our goal is to endow the inference rules with additional information

so they operate on sequents of the form

$$\Omega \vdash M : A$$

where M is a proof (term) of A. In order to write out proof terms we should be able to refer to particular antecedents in Ω. For example, if we have a rule $\backslash L$ and we have multiple antecedents for the form $A \setminus B$, which antecedent is the rule applied to? One solution is to *count*. For example, the rule might be applied to the fifth antecedent. This works to an extent in ordered logic because the order of antecedents never changes. However, it is complicated by the fact that antecedents are split in some rules. For example, in a concrete rule application

$$\frac{A_0\ A_1 \vdash B \quad A_2\ A_3 \vdash C}{A_0\ A_1\ A_2\ A_3 \vdash B \bullet C}\ \bullet R$$

the numbering of antecedents is simple in the conclusion, but in the second premise we suddenly start counting at 2 instead of 0. It is possible to account for that, but proofs are very difficult to read. Also, since antecedents in *linear* logic are subject to exchange, the numbering might change in complicated ways, as in

$$\frac{A_0\ A_2 \vdash B \quad A_1\ A_3 \vdash C}{A_0\ A_1\ A_2\ A_3 \vdash B \otimes C}\ \otimes R$$

Again, this can be dealt with, but there is a more abstract alternative. We label all antecedents with distinct variables that we can refer to in proof terms. A sequent then has the form

$$(x_1 : A_1)\ \ldots\ (x_n : A_n) \vdash M : A$$

where all the x_i are distinct and may be mentioned in M. We show the proof terms (but not the variables in the antecedents) in blue.

We start with $A \setminus B$. Starting out, here is rule we want to annotate on the left, and a partial annotation on the right. The Ω's now stand for antecedents annotated with variables.

$$\frac{A\ \Omega \vdash B}{\Omega \vdash A \setminus B}\ \backslash R \qquad \frac{(x : A)\ \Omega \vdash ? : B}{\Omega \vdash ? : A \setminus B}\ \backslash R$$

A first thing we can say is that x must be chosen so it is fresh and doesn't occur already in Ω. This is so pervasive that it may often not be explicitly stated, relying on the presupposition that all variables declared in the antecedent are distinct.

Continuing, we see that somehow there will be proof $M : B$ once we annotate the premise. We then just need to fill in the slot in the conclusion with a term using it. We uniformly use the name of the rule as a proof constructor.

$$\frac{(x : A)\ \Omega \vdash M : B}{\Omega \vdash ? : A \setminus B}\ \backslash R \qquad \frac{(x : A)\ \Omega \vdash M : B}{\Omega \vdash (\backslash R\ (x.\ M)) : A \setminus B}\ \backslash R$$

We also need to indicate the variable x and track, somehow, that it will be fresh in the premise. We use the notation $(x.\, M)$ for a *bound* occurrence of x with scope M. Because the concrete names of bound variables do not matter, we can always silently rename it in case the particular name x is already among the antecedents. Many rules will take advantage of this notation and convention.

What about the left rule? It is applied to a particular antecedent, so this needs to be explicit.

$$\frac{\Omega_A \vdash A \quad \Omega_L\ B\ \Omega_R \vdash C}{\Omega_L\ \Omega_A\ (A \setminus B)\ \Omega_R \vdash C}\ \setminus L \qquad \frac{\Omega_A \vdash M : A \quad \Omega_L\ (y : B)\ \Omega_R \vdash P : C}{\Omega_L\ \Omega_A\ (x : A \setminus B)\ \Omega_R \vdash (\setminus L\ x\ ?\ ?) : C}\ \setminus L$$

We have already filled in the proof terms for the two premises, and also a name (y) for the antecedent B in the second premise. Now M is carried down without change, because all variables in Ω_A already exists in the conclusion, but we need to abstract P over y because it must be fresh.

$$\frac{\Omega_A \vdash M : A \quad \Omega_L\ (y : B)\ \Omega_R \vdash P : C}{\Omega_L\ \Omega_A\ (x : A \setminus B)\ \Omega_R \vdash (\setminus L\ x\ M\ (y.\, P)) : C}\ \setminus L$$

The rules for right implication can be developed entirely analogously.

$$\frac{\Omega\ A \vdash M : B}{\Omega \vdash (/R\ (x.\, M)) : B\ /\ A}\ /R \qquad \frac{\Omega_A \vdash M : A \quad \Omega_L\ (y : B)\ \Omega_R \vdash P : C}{\Omega_L\ (x : B\ /\ A)\ \Omega_A\ \Omega_R \vdash (/L\ x\ M\ (y.\, P)) : C}\ /L$$

At this point we are almost ready for an example, except for the identity. In certain places, like the first argument to left rules, only variables x are allowed. For the succedent we have an arbitrary term M. This means we could either include variables as a special case of a term, or we could use an explicit term construction like Id. We use the latter approach, so that every inference rules is turned into a corresponding constructor without exception.

$$\frac{}{(x : A) \vdash (\mathsf{Id}\ x) : A}\ \mathsf{id}$$

As an example, let's look at Lambek's associativity law from last lecture, using identity as a full rule.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A \vdash A}\ \mathsf{id} \quad \dfrac{\dfrac{}{B \vdash B}\ \mathsf{id} \quad \dfrac{}{C \vdash C}\ \mathsf{id}}{(C\ /\ B)\ B \vdash C}\ /L}{A\ (A \setminus (C\ /\ B))\ B \vdash C}\ \setminus L}{(A \setminus (C\ /\ B))\ B \vdash A \setminus C}\ \setminus R}{A \setminus (C\ /\ B) \vdash (A \setminus C)\ /\ B}\ /R$$

To annotate this proof with a proof term, we start bottom up, labeling the antecedent with a variable and writing question marks where we have not yet filled in the information.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{? : A \vdash ? : A}\,\mathsf{id} \quad \dfrac{\dfrac{}{(? : B) \vdash ? : B}\,\mathsf{id} \quad \dfrac{}{(? : C) \vdash ? : C}\,\mathsf{id}}{(? : C \,/\, B)\ (? : B) \vdash ? : C}\,/L}{(? : A)\ (x : A \setminus (C \,/\, B))\ (? : B) \vdash ? : C}\,\setminus L}{(x : A \setminus (C \,/\, B))\ (? : B) \vdash ? : A \setminus C}\,\setminus R}{x : A \setminus (C \,/\, B) \vdash ? : (A \setminus C) \,/\, B}\,/R$$

The $/R$ rule introduces a new variable. In order to keep things straight, let's give it the name b.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{? : A \vdash ? : A}\,\mathsf{id} \quad \dfrac{\dfrac{}{(b : B) \vdash ? : B}\,\mathsf{id} \quad \dfrac{}{(? : C) \vdash ? : C}\,\mathsf{id}}{(? : C \,/\, B)\ (b : B) \vdash ? : C}\,/L}{(? : A)\ (x : A \setminus (C \,/\, B))\ (b : B) \vdash ? : C}\,\setminus L}{(x : A \setminus (C \,/\, B))\ (b : B) \vdash ? : A \setminus C}\,\setminus R}{x : A \setminus (C \,/\, B) \vdash ? : (A \setminus C) \,/\, B}\,/R$$

Actually, we now have some information about the proof term in the conclusion, but let's hold off filling that in until we have propagated more information upward. The second inference for $\setminus R$ works in a symmetric way. We call the new variable a.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{a : A \vdash ? : A}\,\mathsf{id} \quad \dfrac{\dfrac{}{(b : B) \vdash ? : B}\,\mathsf{id} \quad \dfrac{}{(? : C) \vdash ? : C}\,\mathsf{id}}{(? : C \,/\, B)\ (b : B) \vdash ? : C}\,/L}{(a : A)\ (x : A \setminus (C \,/\, B))\ (b : B) \vdash ? : C}\,\setminus L}{(x : A \setminus (C \,/\, B))\ (b : B) \vdash ? : A \setminus C}\,\setminus R}{x : A \setminus (C \,/\, B) \vdash ? : (A \setminus C) \,/\, B}\,/R$$

Now the $\setminus L$ rule introduces a new variable, as does the following $/L$. We write those in, naming sure to choose fresh names.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{a : A \vdash ? : A}\,\mathsf{id} \quad \dfrac{\dfrac{}{(b : B) \vdash ? : B}\,\mathsf{id} \quad \dfrac{}{(c : C) \vdash ? : C}\,\mathsf{id}}{(z : C \,/\, B)\ (b : B) \vdash ? : C}\,/L}{(a : A)\ (x : A \setminus (C \,/\, B))\ (b : B) \vdash ? : C}\,\setminus L}{(x : A \setminus (C \,/\, B))\ (b : B) \vdash ? : A \setminus C}\,\setminus R}{x : A \setminus (C \,/\, B) \vdash ? : (A \setminus C) \,/\, B}\,/R$$

Now that we have named all antecedents in all sequents, we can fill in the proof terms according to our proof term language, starting at the top and moving towards the bottom. We combine these into two steps, starting with the identities.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{a : A \vdash (\mathsf{Id}\ a) : A}\ \mathsf{id} \quad \dfrac{\dfrac{}{(b : B) \vdash (\mathsf{Id}\ b) : B}\ \mathsf{id} \quad \dfrac{}{(c : C) \vdash (\mathsf{Id}\ c) : C}\ \mathsf{id}}{(z : C \,/\, B)\ (b : B) \vdash ? : C}\ /L}{(a : A)\ (x : A \setminus (C \,/\, B))\ (b : B) \vdash ? : C}\ \setminus L}{(x : A \setminus (C \,/\, B))\ (b : B) \vdash ? : A \setminus C}\ \setminus R}{x : A \setminus (C \,/\, B) \vdash ? : (A \setminus C) \,/\, B}\ /R$$

And then we complete the terms for the remaining rule applications, working downwards.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{a : A \vdash (\mathsf{Id}\ a) : A}\ \mathsf{id} \quad \dfrac{\dfrac{}{(b : B) \vdash (\mathsf{Id}\ b) : B}\ \mathsf{id} \quad \dfrac{}{(c : C) \vdash (\mathsf{Id}\ c) : C}\ \mathsf{id}}{(z : C \,/\, B)\ (b : B) \vdash (/L\ z\ (\mathsf{Id}\ b)\ (c.\,\mathsf{Id}\ c)) : C}\ /L}{(a : A)\ (x : A \setminus (C \,/\, B))\ (b : B) \vdash (\setminus L\ x\ (\mathsf{Id}\ a)\ (/L\ z\ (\mathsf{Id}\ b)\ (c.\,\mathsf{Id}\ c))) : C}\ \setminus L}{(x : A \setminus (C \,/\, B))\ (b : B) \vdash (\setminus R\ (a.\setminus L\ x\ (\mathsf{Id}\ a)\ (/L\ z\ (\mathsf{Id}\ b)\ (c.\,\mathsf{Id}\ c)))) : A \setminus C}\ \setminus R}{x : A \setminus (C \,/\, B) \vdash (/R\ (b.\setminus R\ (a.\setminus L\ x\ (\mathsf{Id}\ a)\ (/L\ z\ (\mathsf{Id}\ b)\ (c.\,\mathsf{Id}\ c))))) : (A \setminus C) \,/\, B}\ /R$$

There is a lot of redundant information in this derivation. In fact, starting with

$$x : A \setminus (C \,/\, B) \vdash (/R\ (b.\setminus R\ (a.\setminus L\ x\ (\mathsf{Id}\ a)\ (/L\ z\ (\mathsf{Id}\ b)\ (c.\,\mathsf{Id}\ c))))) : (A \setminus C) \,/\, B$$

we can reconstruct the whole derivation in a unique way.

This correspondence can be stated formally as two theorems (for antecedents Ω that are labelled with unique variables).

(i) Given Ω, M, and A, either there is a unique derivation

$$\begin{array}{c}\mathcal{D}\\ \Omega \vdash M : A\end{array}$$

or there is no such derivation.

(ii) Given a derivation $\Omega \vdash A$ where the applications of left rules are marked with their corresponding variable, then there is a unique term M such that $\Omega \vdash M : A$.

The process s of constructing a derivation from a term is not entirely straightforward because of the necessary splits of the hypotheses in rules with multiple premises. We could either look ahead to see which variables occur, or we can propagate all antecedents to one of the premises and then pass on the ones that were

not used to the other premises. In this algorithm, we need to make sure that uses of variables in a derivation are consecutive so that order is suitably respected.

A general algorithm for the input/output interpretation of antecedents that works for structural, linear, and ordered antecedents and even a mix is described by Polakow [2000], with more proof details in his Ph.D. thesis [Polakow, 2001]. This system work *during proof search* when a full proof term isn't even available for checking, so it solves a somewhat more difficult problem than is needed here. Still, it provides an elegant algorithmic solution.

We show one more example, for $A \bullet B$. There is no variable binding in the right rule.

$$\frac{\Omega_1 \vdash A \quad \Omega_2 \vdash B}{\Omega_1\ \Omega_2 \vdash A \bullet B}\ \bullet R \qquad \frac{\Omega_1 \vdash M : A \quad \Omega_2 \vdash N : B}{\Omega_1\ \Omega_2 \vdash (\bullet R\ M\ N) : A \bullet B}\ \bullet R$$

The left rule is a kind of pattern matching and therefore has to bind *two* fresh variables.

$$\frac{\Omega_L\ A\ B\ \Omega_R \vdash C}{\Omega_L\ (A \bullet B)\ \Omega_R \vdash C}\ \bullet L \qquad \frac{\Omega_L\ (y : A)\ (z : B)\ \Omega_R \vdash P : C}{\Omega_L\ (x : A \bullet B)\ \Omega_R \vdash (\bullet L\ x\ (y.\, z.\, P)) : C}\ \bullet L$$

Since the proof terms are constructed quite systematically, we don't show the remaining rules. The language of proof terms is summarized in Figure 1.

3 Cut Reductions on Proof Terms

We an express the cut reductions between two proofs on the proof terms themselves. We show only one example (the principal reduction for fuse), but others are similar. We first introduce a proof term for cut, taking it here as a given rules of inference rather than just admissible.

$$\frac{\Omega \vdash M : A \quad \Omega_L\ (x : A)\ \Omega_R \vdash P : C}{\Omega_L\ \Omega\ \Omega_R \vdash (\mathsf{Cut}_A\ M\ (x.\, P)) : C}\ \mathsf{cut}_A$$

Now to the particular case. Before the reduction, we have

$$\frac{\dfrac{\Omega_1 \vdash M : A \quad \Omega_2 \vdash N : B}{\Omega_1\ \Omega_2 \vdash (\bullet R\ M\ N) : A \bullet B}\ \bullet R \qquad \dfrac{\Omega_L\ (y : A)\ (z : B)\ \Omega_R \vdash P : C}{\Omega_L\ (x : A \bullet B)\ \Omega_R \vdash (\bullet L\ x\ (y.\, z.\, P)) : C}\ \bullet L}{\Omega_L\ \Omega_1\ \Omega_2\ \Omega_R \vdash (\mathsf{Cut}_{A \bullet B}\ (\bullet R\ M\ N)\ (x.\, \bullet L\ x\ (y.\, z.\, P))) : C}\ \mathsf{cut}_{A \bullet B}$$

and after the reduction (writing in proof terms afresh):

$$\longrightarrow_R \quad \frac{\Omega_2 \vdash N : B \qquad \dfrac{\Omega_1 \vdash M : A \quad \Omega_L\ (y : A)\ (z : B)\ \Omega_R \vdash P : C}{\Omega_L\ \Omega_1\ (z : B)\ \Omega_R \vdash (\mathsf{Cut}_A\ M\ (y.\, P)) : C}\ \mathsf{cut}_A}{\Omega_L\ \Omega_1\ \Omega_2\ \Omega_R \vdash (\mathsf{Cut}_B\ N\ (z.\, \mathsf{Cut}_A\ M\ (y.\, P))) : C}\ \mathsf{cut}_B$$

Expressing this purely on the proof term, we can recognize it as a kind of pattern matching reduction, except that we don't substitute the way we would usually think of it in the definition of functional languages.

$$\mathsf{Cut}_{A \bullet B}\ (\bullet R\ M\ N)\ (\bullet L\ x\ (y.\, z.\, P)) \longrightarrow_R \mathsf{Cut}_B\ N\ (z.\, \mathsf{Cut}_A\ M\ (y.\, P))$$

In the next lecture similar cut reductions play a much more significant role because we will make the computational intuition more precise.

4 Invertibility and Polarity

When constructing proofs, bottom-up, a priori we have many possible choices. Any left rule might apply to any matching antecedent, or a right rule to a matching succedent. Applying a rule is a small step, breaking down just one connective. Then we are again faced with a similar choice. Reducing this nondeterminism is critical in proof search procedures, although it may not mean much regarding the question of decidability.

For example, it is easy to see that the pure ordered logic we have seen is decidable once we know cut elimination, because the premises of all the rules are smaller than the conclusion in the sense of having fewer connectives in them. Therefore, any way we can try to construct a proof, bottom-up, will have to terminate, either in success or in failure. If we try all of them, we will either find a proof or there cannot be any.

Fortunately, we don't need to search that blindly while remaining complete. For each connective, either the left rule or the right rule in the sequent calculus is *invertible* in the sense that the premises are provable if and only if the conclusion is. So we can use such a rule, bottom-up, without having to consider any other choices because we have preserved provability exactly.

The question is which rules are invertible. There is an easy test: whichever rule is applied first (again, reading bottom-up) in the identity expansion is the invertible rule while the counterpart on the other side is not. Here is a tiny example:

$$\dfrac{\dfrac{\dfrac{}{A \vdash A}\ \mathsf{id}_A \quad \dfrac{}{B \vdash B}\ \mathsf{id}_B}{A\ B \vdash A \bullet B}\ \bullet R}{A \bullet B \vdash A \bullet B}\ \bullet L$$

While it is now plausible that $\bullet L$ rule is invertible, we can see that $\bullet R$ is not because we cannot (yet) break up the antecedents appropriately.

We can *prove* invertibility of $\bullet L$ in pure ordered logic using the admissibility of cut and identity. You may want to try this yourself before peeking at the solution on the next page.

$$\dfrac{\dfrac{\cdots\cdots\ \mathsf{id}_A \quad \cdots\cdots\ \mathsf{id}_B}{A \vdash A \qquad B \vdash B}\ \bullet R \qquad \Omega_L\ (A \bullet B)\ \Omega_R \vdash C}{\Omega_L\ A\ B\ \Omega_R \vdash C}\ \mathsf{cut}_{A \bullet B}$$

You can see if you read this from the unproved premise to the conclusion, it is just the inverted $\bullet L$. Given any concrete proof of the second premise we can apply cut elimination to obtain a direct proof of the conclusion.

But we have to be a bit careful that this notion of *invertibility* may not completely coincide with the inference rule being reversible. For example, the rule

$$\frac{}{\cdot \vdash \mathbf{1}}\ \mathbf{1}R$$

is technically invertible in the sense that whenever the conclusion is, so are all the premises (namely none). However, we cannot always apply this rule when we see $\mathbf{1}$ in a succedent because the antecedents may not be empty. If we had formulated the rule slightly differently:

$$\frac{\Delta = (\cdot)}{\Delta \vdash \mathbf{1}}\ \mathbf{1}R$$

then it would not no longer be invertible.

Therefore, instead of talking about the right or left invertibility of a *rule*, we talk about the *right or left invertibility of a connective*. If we can *always* apply its right or left rule without losing provability when a connective appears at the top level of a proposition, we call the connective invertible on the right or on the left, respectively.

The distinction of whether left or right rules are invertible is of fundamental importance in studying proof theory, and its connection to computation. We call the right invertible connectives *negative*, while left invertible connectives are *positive*. For ordered logic, we get the following classification:

Negative (right invertible): $A \setminus B$, $A \,/\, B$, $A \mathbin{\&} B$, $\top$

Positive (left invertible): $A \bullet B$, $A \circ B$, $\mathbf{1}$, $A \oplus B$, $\mathbf{0}$

5 A Zoo of Connectives

In linear and ordered logic, the polarity of each connective is uniquely determined. Somewhat surprisingly, though, in structural logic conjunction has both invertible left and right rules. This is because it actually unifies two different connectives we know from linear logic: truth in the same state $A \otimes B$ (positive) and external choice $A \& B$ (negative). It turns out that it is highly beneficial to make this distinction even

for intuitionistic logic, but this is rarely done—its significance was not recognized until the discovery of call-by-push-value (really: a polarized type system) [Levy, 2006]. We will see that positive types (including positive pairs) are *eager* while negative types (including externa choice) are *lazy*.

Below is the table of connective in the various logics, where we see that certain connectives further on the right becomes indistinguishable when we move to the left. For example, if the order of antecedents is irrelevant (e.g., in linear logic) then left and right implication ($A \backslash B$ and B / A) become indistinguishable and are written as $A \multimap B$.

structural	linear	ordered	polarity	pronunciation
$A \supset B$	$A \multimap B$	$A \setminus B$	negative	A under B
		$B \,/\, A$	negative	B over A
$A \wedge B$	$A \otimes B$	$A \bullet B$	positive	A fuse B
		$A \circ B$	positive	A twist B
	$A \mathbin{\&} B$	$A \mathbin{\&} B$	negative	A with B
$A \vee B$	$A \oplus B$	$A \oplus B$	positive	A plus B
$\top$	1	1	positive	one
	$\top$	$\top$	negative	top
$\bot$	0	0	positive	zero

The ambiguous nature of general structural conjunction $A \wedge B$ and $\top$ is resolved at the linear level because these connectives split into two each: one positive and one negative.

6 Summary

The language of proof terms is in Figure 1. Since the constructors are named after the inference rules we don't bother showing the inference rules. You should be able to easily write them out.

Valid proof terms are in one-to-one correspondence with proofs, so they merely serve as a compact notation here. We can then express operations such as cut reduction on these terms, rather than showing complex derivations.

If we think of cut and identity as being admissible, then Id_A (at types other than atoms P) and Cut_A would be meta-level operations to compute a cut-free proof from the arguments. But we need to keep in mind that cut reduction is highly nondeterministic, so perhaps $\mathsf{Cut}_A\ M\ (x.\, P) \sim N$ it is best thought of a 4-place relation between A, M, $x.\, P$, and N (all of the proofs being cut-free).

$$
\begin{array}{rcll}
M, N, P & ::= & \mathsf{Cut}_A\ M\ (x.\, P) & \mid\ \mathsf{Id}\ x \\
& \mid & \backslash R\ (x.\, M) & \mid\ \backslash L\ x\ M\ (y.\, P) \\
& \mid & /R\ (x.\, M) & \mid\ /L\ x\ M\ (y.\, P) \\
& \mid & \bullet R\ M\ N & \mid\ \bullet L\ x\ (y.\, z.\, P) \\
& \mid & \circ R\ M\ N & \mid\ \circ L\ x\ (z.\, y.\, P) \\
& \mid & \mathbin{\&} R\ M\ N & \mid\ \mathbin{\&} L_1\ x\ (y.\, P) \mid \mathbin{\&} L_2\ x\ (z.\, P) \\
& \mid & \mathbf{1} R & \mid\ \mathbf{1} L\ x\ M \\
& \mid & \oplus R_1\ M \mid \oplus R_2\ N & \mid\ \oplus L\ x\ (y.\, N)\ (z.\, P) \\
& \mid & \top R & \mid \\
& \mid & & \mid\ \mathbf{0} L\ x
\end{array}
$$

Figure 1: Proof terms for ordered logic

References

Paul Blain Levy. Call-by-push-value: Decomposing call-by-value and call-by-name. *Higher-Order and Symbolic Computation*, 19(4):377–414, 2006.

Jeff Polakow. Linear logic programming with an ordered context. In M. Gabbrielli and F. Pfenning, editors, *Conference on Principles and Practice of Declarative Programming (PPDP 2000)*, pages 68–79, Montreal, Canada, September 2000. ACM.

Jeff Polakow. *Ordered Linear Logic and Applications*. PhD thesis, Department of Computer Science, Carnegie Mellon University, August 2001.