Lecture Notes on
Linear Natural Deduction

15-836: Substructural Logics
Sophia Roshal

Lecture 23
December 7, 2023

1 Introduction

Throughout the course we have been focusing on the sequent calculus and its variations SAX [DeYoung et al., 2020] and SNAX [DeYoung and Pfenning, 2022] all of which follow a bottom up reasoning method. In this lecture we introduce *natural deduction* [Gentzen, 1935] which instead has 2 directions of reasoning, with introduction rules which follow bottom up reasoning, and elimination rules which follow top down reasoning. Using all the same tools as we have used so far, we will develop these rules, and prove some important properties as well as relate natural deduction back to the sequent calculus.

2 The Base Rules

As stated in the introduction, we have two forms of rules. Introduction rules (which we think of reasoning upwards) correspond directly to the sequent calculus right rules. The elimination rules (which we think of reasoning downwards) will need a bit more work. All the rules can be found in the appendix, but in this section we will focus on one positive ($\otimes$) and one negative ($\multimap$) connective. First, for completeness sake, we state the introduction rule for $\otimes$.

$$\frac{\Delta_1 \vdash M : A \quad \Delta_2 \vdash N : B}{\Delta_1, \Delta_2 \vdash (M, N) : (A \otimes B)} \otimes I$$

Now, we think about the elimination form. We should be starting, in our premise, with

$$\frac{\Delta_1 \vdash M : (A \otimes B)}{} \otimes E \text{ incomplete}$$

Since we are in a linear setting, we are required to somehow use both A and B, however, we aren't allowed to have multiple conclusions in our rules, so, similarly to the sequent calculus, we will break apart $A \otimes B$ into its parts, and use these parts to prove some new right hand side term C producing the following rule:

$$\frac{\Delta_1 \vdash M : A \otimes B \quad \Delta_2, x : A, y : B \vdash N : C}{\Delta_1, \Delta_2 \vdash \text{match } M \text{ with } (x, y) \text{ in } N : C} \otimes E$$

For $\multimap$, again the introduction rule is identical to the sequent calculus right rule.

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

For the elimination rule, we start with

$$\frac{\Delta_1 \vdash M : A \multimap B}{} \multimap E \text{ partial}$$

Thinking about linearity, we know we need to consume M, and approaching this from programming perspective, $\multimap$ is a the type of a function. To consume a function, we apply it, so this rule should correspond to function application.

$$\frac{\Delta_1 \vdash M : A \multimap B \quad \Delta_2 \vdash N : A}{\Delta_1, \Delta_2 \vdash M\ N : B} \multimap E$$

Other rules can be constructed in a similar fashion, with positive connectives corresponding almost directly to sequent calculus rules, and negative connectives requiring a bit more work.

3 Bidirectional Type Checking

In section 2, we presented some rules, and claimed that we read the introduction rules bottom up, as in the sequent calculus, and the elimination rules top down, but this isn't made in any way explicit. We can make this explicit via bidirectional type checking [Dunfield and Krishnaswami, 2022] by splitting the judgement $M : A$ into two judgements $M \Leftarrow C$ (read as M checks against C) and $M \Rightarrow A$ (read as M synthesizes A.) From an implementation perspective, we think of the checking judgement as type checking (where both M and C are provided as inputs) and the synthesis judgement as type inference (where only the term M is given, and we infer its type). The checking judgement is the upward reasoning direction, while the synthesis judgement is the downwards reasoning direction. We start with $\otimes$. The introduction rule is fairly straightforward. To check if the pair (M, N) has the type $A \otimes B$, we need to check the individual components.

$$\frac{\Delta_1 \vdash M \Leftarrow A \quad \Delta_2 \vdash N \Leftarrow B}{\Delta_1, \Delta_2 \vdash (M, N) \Leftarrow A \otimes B} \otimes I$$

The elimination rule is a bit trickier.

$$\frac{\Delta_1 \vdash M \Rightarrow (A \otimes B) \quad \Delta_2, x : A, y : B \vdash N \ ?\ C}{\Delta_1, \Delta_2 \vdash \text{match } M \text{ with } (x, y) \text{ in } N \ ?\ C} \otimes E \text{ incomplete}$$

The first premise we have labeled as a synthesis, this is to embody the downward direction reasoning. To label the other two, there are several ways to do so that would be "correct" in the sense that we would still have a complete system with respect to standard natural deduction. One way to decide, is to consider the sequent calculus rule, and label everything that appears on the left as a synthesis judgement, and everything that appears on the right as a checking judgement. Internally to natural deduction, we can come to the same conclusion for this rule, by thinking about the order that rules apply in. We first want to apply all the introduction rules we can before we begin apply elimination rules. Once we have reached the point of applying the $\otimes$ elimination rule, we are still in a checking mode, and from this rule we don't have information about what C (or N) is, so we might as well remain in a checking mode. This gives us the following rule

$$\frac{\Delta_1 \vdash M \Rightarrow (A \otimes B) \quad \Delta_2, x : A, y : B \vdash N \Leftarrow C}{\Delta_1, \Delta_2 \vdash \text{match } M \text{ with } (x, y) \text{ in } N \Leftarrow C} \otimes E$$

Now for $\multimap$. Again, the introduction rule is fairly straightforward.

$$\frac{\Delta, x : A \vdash M \Leftarrow B}{\Delta \vdash \lambda x.M \Leftarrow A \multimap B} \multimap I$$

For the elimination rule, we again think back to the sequent calculus, and which propositions appear on which side in the implication left rule. If we follow that, we are left with

$$\frac{\Delta_1 \vdash M \Rightarrow A \multimap B \quad \Delta_2 \vdash N \Leftarrow A}{\Delta_1, \Delta_2 \vdash M\ N \Rightarrow B} \multimap E$$

Internally to natural deduction, and thinking about the proof terms, we could come to the same conclusion. Starting from the top, we know $A \multimap B$ should be a synthesis as that's the term we are applying the elimination rule to. Once we have that type, we now have A as well so the second premise can be a checking judgement. Finally, for the conclusion, having it be a checking judgement would not be helpful to the proof. We'd still need to synthesize a type for M, so this should remain as a synthesis.

Lastly we also present the hypothesis rule.

$$\frac{}{x : A \vdash x \Rightarrow A} \mathsf{hyp}(x)$$

From the separation into upwards and downwards reasoning point of view, it doesn't make sense to make this an upwards rule as there is no upward direction to go. From a programming perspective, we want this rule to be a synthesis, as we don't want to have to annotate our variables with a type if it isn't necessary to do so.

While this looks like the identity rule of the sequent calculus, it has a different function and purpose (as we'll see later in this lecture), so we call it the *hypothesis* rule.

4 An example

Here, we prove one direction of currying in our bidirectional system. Through this derivation, we will see that on top of the base rules, we actually need one more.

$$
\dfrac{
\dfrac{}{ab : (A \otimes B) \vdash ab \Longrightarrow A \otimes B}\;\mathsf{hyp}(ab)
\qquad
\dfrac{
\dfrac{
\dfrac{
\dfrac{}{f : (A \multimap (B \multimap C)) \vdash f \Rightarrow (A \multimap (B \multimap C))}\;\mathsf{hyp}(f)
\quad
\dfrac{\dfrac{}{a : A \vdash a \Rightarrow A}\;\mathsf{hyp}(a)}{a : A \vdash a \Leftarrow A}\;?
}{f : (A \multimap (B \multimap C)), a : A \vdash f \Rightarrow (A \multimap (B \multimap C))}\;\multimap E
\quad
\dfrac{\dfrac{}{b : B \vdash b \Rightarrow B}\;\mathsf{hyp}(b)}{b : B \vdash b \Leftarrow B}\;?
}{f : (A \multimap (B \multimap C)), a : A, b : B \vdash ((f\ a)\ b) \Rightarrow C}\;\multimap E
}{f : (A \multimap (B \multimap C)), a : A, b : B \vdash ((f\ a)\ b) \Leftarrow C}\;?
}{
\dfrac{
\dfrac{f : (A \multimap (B \multimap C)), ab : A \otimes B \vdash \mathsf{match}\ ab\ \mathsf{with}\ (a,b)\ \mathsf{in}\ ((f\ a)\ b) \Leftarrow C}{f : (A \multimap (B \multimap C)) \vdash \lambda ab.\mathsf{match}\ ab\ \mathsf{with}\ (a,b)\ \mathsf{in}\ ((f\ a)\ b) \Leftarrow (A \otimes B) \multimap C}\;\multimap I
}{\cdot \vdash \lambda f.\lambda ab.\mathsf{match}\ ab\ \mathsf{with}\ (a,b)\ \mathsf{in}\ ((f\ a)\ b) \Leftarrow (A \multimap (B \multimap C)) \multimap ((A \otimes B) \multimap C)}\;\multimap I
}\;\otimes E
$$

There are a few places in this proof marked with a question mark, which leads us to one more rule:

$$\frac{\Delta \vdash M \Rightarrow A}{\Delta \vdash M \Leftarrow A} \Rightarrow/\Leftarrow^*$$

From an implementation perspective however, we need to modify this rule just a bit. Synthesis only takes the term as input so this rule doesn't entirely make sense as written, given that M synthesizes some type, and we don't necessarily know that it will be A specifically. This is something we need to check. We make a small modification.

$$\frac{\Delta \vdash M \Rightarrow A' \quad A = A'}{\Delta \vdash M \Leftarrow A} \Rightarrow/\Leftarrow$$

While we've written $A = A'$ for type equality here, we could also generalize this to be a sub-typing call, if our type system supports sub-typing. This one rule is the only place where we need to do type equality/sub-typing checks.

5 System Correctness

So far, we have completed an example proof in natural deduction that we knew should work. Unfortunately (or fortunately) proof by one example is not usually an accepted form of proof in most publication venues, and so we need to do a bit more work before being satisfied that the system we have developed is correct. There are several criteria we can consider when we want to decide whether a system we have is "correct". For the sequent calculus, we had an internal way to do so via admissibility of identity and cut which established harmony between the left and right rules. In natural deduction, we also have a similar internal notion via local soundness and completeness of the introduction and elimination rules. Another way we can establish correctness is by relating to some outside system. Throughout this course, we have established the sequent calculus as "the source of truth" and have proven many properties about it, so it would make sense to want to verify that natural deduction corresponds in some way to the sequent calculus.

5.1 Harmony

We have two properties to prove: local soundness (or proof normalization) which corresponds to cut elimination/admissibility in the sequent calculus and local completeness which corresponds to admissibility of identity in the sequent calculus. We start with local completeness. Here, we want to prove that we given an arbitrary proof of $\Delta \vdash M : A$, and applying the corresponding elimination rule, we should be able to reconstruct A again (of course, with a different proof term).

Proof: **Case:** $\otimes$

$$\begin{array}{c}\mathcal{D}\\ \Delta \vdash M : A \otimes B\end{array} \Longrightarrow \dfrac{\begin{array}{c}\mathcal{D}\\ \Delta \vdash M : A \otimes B\end{array} \quad \dfrac{\dfrac{}{x : A \vdash x : A}\,\mathsf{hyp} \quad \dfrac{}{y : B \vdash y : B}\,\mathsf{hyp}}{x : A, y : B \vdash (x, y) : (A \otimes B)}\,\otimes I}{\Delta \vdash \mathsf{match}\ M\ \mathsf{with}\ (x, y)\ \mathsf{in}\ (x, y) : (A \otimes B)}\,\otimes E$$

Case: $\multimap$

$$\begin{array}{c}\mathcal{D}\\ \Delta \vdash M : A \multimap B\end{array} \Longrightarrow \dfrac{\dfrac{\begin{array}{c}\mathcal{D}\\ \Delta \vdash M : A \multimap B\end{array} \quad \dfrac{}{x : A \vdash x : A}\,\mathsf{hyp}}{\Delta, x : A \vdash M\ x : B}\multimap E}{\Delta \vdash \lambda x.(M\ x) : A \multimap B}\multimap I$$

Other cases proceed similarly. □

Soundness requires a bit more work. We want to establish that if we apply an introduction rule then an elimination rule, we can actually simplify that proof. this corresponds to cut elimination (and more specifically the principal case of cut elimination with a right rule "introducing" the cut proposition and the left rule "eliminating" it). To do so, we need one more lemma: substitution. This corresponds to the more general cut reduction, where we may not be cutting together a right and left rule. We need to prove that given a derivation that relies on a variable, we can substitute that variable for a term of the same type.

Lemma 1 *The following rule is admissible in the system without it*

$$\dfrac{\Delta \vdash M : A \quad \Delta', x : A \vdash N(x) : C}{\Delta, \Delta' \vdash N(M) : C}\,subst$$

Proof: Proof proceeds by induction on the second given derivation. We provide an interesting case

Case: the second derivation ends in a $\multimap E$. This case actually splits into two possible cases, we show just the first (the second case is almost identical, with the difference of how the context is split).

$$\dfrac{\Delta \vdash M : A \quad \dfrac{\begin{array}{c}\mathcal{D}_1\\ \Delta_1', x : A \vdash N(x) : B \multimap C\end{array} \quad \begin{array}{c}\mathcal{D}_2\\ \Delta_2' \vdash N'(x) : B\end{array}}{\Delta_1', \Delta_2', x : A \vdash N(x)\ N'(x) : C}\multimap E}{\Delta, \Delta_1', \Delta_2' \vdash N(M)\ N'(M) : C}\,subst$$

First, we need to realize that x cannot appear in N' as it must appear in N due to linearity (and x is not present in Δ_2'), so $N'(x) = N'$ which in turn gives us $N'(M) = N'$. We can now push the substitution upwards, reducing this proof to the following:

$$\dfrac{\dfrac{\Delta \vdash M : A \quad \overset{\mathcal{D}_2}{\Delta_1', x : A \vdash N(x) : B \multimap C}}{\Delta, \Delta_1' \vdash N(M) : B \multimap C}\ subst \qquad \overset{\mathcal{D}_2}{\Delta_2' \vdash N' : B}}{\Delta, \Delta_1', \Delta_2' \vdash N(M)\ N' : C} \multimap E$$

Other cases proceed similarly. □

Moving on to proving local soundness.

Proof: Case: $\otimes$

$$\dfrac{\dfrac{\overset{\mathcal{D}_1}{\Delta_a \vdash M : A} \quad \overset{\mathcal{D}_2}{\Delta_b \vdash M' : B}}{\Delta_a, \Delta_b \vdash (M, M') : A \otimes B} \otimes I \qquad \overset{\mathcal{E}}{\Delta', x : A, y : B \vdash N : C}}{\Delta_a, \Delta_b, \Delta' \vdash \text{match } (M, M') \text{ with } (x, y) \text{ in } N : \text{C}} \otimes E$$

We know N relies on both x and y, and this is where substitution comes into play. We construct the following proof reduction via admissibility of substitution:

$$\dfrac{\overset{\mathcal{D}_1}{\Delta_a \vdash M : A} \qquad \dfrac{\overset{\mathcal{D}_2}{\Delta_b \vdash M' : B} \quad \overset{\mathcal{E}}{\Delta', x : A, y : B \vdash N(x, y) : C}}{\Delta_b, \Delta', x : A \vdash N(x, M') : C}\ subst}{\Delta_a, \Delta_b, \Delta' \vdash N(M, M') : C}\ subst$$

Case: $\multimap$

$$\dfrac{\dfrac{\overset{\mathcal{D}}{\Delta_1, x : A \vdash M : B}}{\Delta_1 \vdash \lambda x.M : A \multimap B} \multimap I \qquad \overset{\mathcal{E}}{\Delta_2 \vdash N : A}}{\Delta_1, \Delta_2 \vdash (\lambda x.M)N : B} \multimap E \implies$$

$$\dfrac{\overset{\mathcal{E}}{\Delta_2 \vdash N : A} \quad \overset{\mathcal{D}}{\Delta_1, x : A \vdash M(x) : B}}{\Delta_1, \Delta_2 \vdash M(N) : B}\ subst$$

other cases proceed similarly. □

5.2 Soundness/Completeness wrt Sequent Calculus

We have developed two natural deduction systems. We would like to be able to relate them to each other as well as to the sequent calculus, and to do so we can prove three theorems. To separate notation, we will use $\overset{nd}{\vdash}$ to represent derivations in standard natural deduction, $\overset{\uparrow\downarrow}{\vdash}$ to represent derivations in bidirectional natural deduction, and $\overset{seq}{\vdash}$ to represent derivations in the sequent calculus. The three theorems we need now are

Theorem 2 *If* $\Delta \overset{nd}{\vdash} M : C$ *then* $\Delta \overset{seq}{\vdash} C$

Theorem 3 *If* $\Delta \overset{seq}{\vdash} C$ *then* $\Delta' \overset{\uparrow\downarrow}{\vdash} M \Leftarrow C$ *for some* M *where* $\Delta' \overset{sub}{\vdash} \Delta$

Theorem 4 *If* $\Delta \overset{\uparrow\downarrow}{\vdash} M \Leftarrow C$ *then* $\Delta \overset{nd}{\vdash} M : C$

Notice that in the second theorem, we need a modification to what we initially might think of as the theorem. We will come back to that in the proof.

Proof: Theorem 1
We proceed by induction over the natural deduction derivation. We provide the cases for $\otimes$ and $\multimap$ as we have been throughout these notes. However, we leave out the cases for the introduction rules as they follow directly via application of the induction hypothesis.

Case: $\otimes E$

$$\dfrac{\overset{\mathcal{D}_1}{\Delta_1 \overset{nd}{\vdash} M : A \otimes B} \quad \overset{\mathcal{D}_2}{\Delta_2, x : A, y : B \overset{nd}{\vdash} N : C}}{\Delta_1, \Delta_2 \overset{nd}{\vdash} \text{match } M \text{ with } (x, y) \text{ in } N : C} \otimes E$$

From the inductive hypothesis on $\mathcal{D}_1$ and $\mathcal{D}_2$ we can conclude

$$\Delta_1 \overset{seq}{\vdash} A \otimes B$$

$$\Delta_2, A, B \overset{seq}{\vdash} C$$

We need to prove $\Delta_1, \Delta_2 \overset{seq}{\vdash} C$.

Looking at the statements we have from our inductive hypothesis, there doesn't seem to be a way to proceed directly. The only applicable rule is $\otimes R$ but that isn't particularly useful. However, we have one more tool that we can use. We have the admissability of cut in the sequent calculus. We proceed with the proof as follows.

$$\dfrac{\overset{IH(\mathcal{D}_1)}{\Delta_1 \overset{seq}{\vdash} A \otimes B} \quad \dfrac{\overset{IH(\mathcal{D}_2)}{\Delta_2, A, B \overset{seq}{\vdash} C}}{\Delta_2, A \otimes B \overset{seq}{\vdash} C} \otimes L}{\Delta_1, \Delta_2 \overset{seq}{\vdash} C} \mathit{cut}$$

Case: $\multimap E$

$$\dfrac{\overset{\mathcal{D}_1}{\Delta_1 \overset{nd}{\vdash} M : A \multimap B} \quad \overset{\mathcal{D}_2}{\Delta_2 \overset{nd}{\vdash} N : A}}{\Delta_1, \Delta_2 \overset{nd}{\vdash} M\ N : B} \multimap E$$

From the inductive hypothesis we conclude

$$\Delta_1 \vdash A \multimap B$$

$$\Delta_2 \vdash A$$

Again, there is no clear way forward unless we use the admissibility of cut (and in this case we also use the admissibility of identity). Once we do so, we obtain the following proof.

$$\dfrac{\overset{IH(\mathcal{D}_1)}{\Delta_1 \overset{seq}{\vdash} A \multimap B} \qquad \dfrac{\overset{IH(\mathcal{D}_2)}{\Delta_2 \overset{seq}{\vdash} A} \qquad \dfrac{}{B \overset{seq}{\vdash} B}\ \mathsf{id}}{\Delta_2, A \multimap B \overset{seq}{\vdash} B}\ {\multimap} L}{\Delta_1, \Delta_2 \overset{seq}{\vdash} B}\ cut$$

Cases for other connectives proceed in a similar fashion. $\square$

Proof: Theorem 2
We now work through why we need the modification in theorem 2. Assume first that we had not made the modification and instead had just

$$\text{If } \Delta \overset{seq}{\vdash} C \text{ then } \Delta \overset{\uparrow\downarrow}{\vdash} M \Leftarrow C$$

We try to proceed with a proof of $\otimes L$

$$\dfrac{\overset{\mathcal{D}'}{\Delta, A, B \vdash C}}{\Delta, A \otimes B \vdash C}\ \otimes L$$

If we were to apply the inductive hypothesis based on the incorrect theorem statement, we get

$$\Delta, x : A, y : B \overset{\uparrow\downarrow}{\vdash} M \Leftarrow C$$

and get stuck. There are no clear rules we can apply here, and we also arbitrarily labeled propositions in the context with some variables without any clear reasoning. So instead, we want some way of translating a sequent calculus context into a natural deduction one, that allows us to apply rules to what was in the sequent calculus context. To do so we define the following rules:

$$\dfrac{}{\cdot \overset{sub}{\vdash} \cdot} \qquad\qquad \dfrac{\Delta_1' \overset{sub}{\vdash} \Delta \qquad \Delta_2' \overset{\uparrow\downarrow}{\vdash} M \Rightarrow A}{\Delta_1', \Delta_2' \overset{sub}{\vdash} \Delta, A}$$

Δ_1' and Δ_2' are natural deduction contexts, while Δ is a sequent calculus context. We are able to go back to our corrected proof statement. Repeated here for convenience.

$$\text{If } \Delta \overset{seq}{\vdash} C \text{ then } \Delta' \overset{\uparrow\downarrow}{\vdash} M \Leftarrow C \text{ for some } M \text{ where } \Delta' \overset{sub}{\vdash} \Delta$$

Case: $\otimes L$

$$\dfrac{\begin{array}{c}\mathcal{D}' \\ \Delta, A, B \vdash C\end{array}}{\Delta, A \otimes B \vdash C}\ \otimes L$$

While we can no longer apply the induction hypothesis directly, we have some new assumptions to work with.

$$\Delta_{ab} \overset{\uparrow\downarrow}{\vdash} M \Rightarrow A \otimes B \tag{1}$$

$$\Delta' \overset{sub}{\vdash} \Delta \tag{2}$$

And we want to show

$$\Delta_{ab}, \Delta' \overset{\uparrow\downarrow}{\vdash} M' \Leftarrow C$$

We are now able to apply $\otimes E$ to (1) producing

$$\dfrac{\begin{array}{c}(1) \\ \Delta_{ab} \overset{\uparrow\downarrow}{\vdash} M \Rightarrow A \otimes B\end{array} \quad \begin{array}{c}? \\ \Delta', x : A, y : B \overset{\uparrow\downarrow}{\vdash} N \Leftarrow C\end{array}}{\Delta_{ab} \overset{\uparrow\downarrow}{\vdash} \text{match } M \text{ with } (x, y) \text{ in } N \Leftarrow C}\ \otimes E$$

We still need a proof of the second premise. Luckily, we can finally apply the inductive hypothesis! We can do so because from assumption we know $\Delta' \overset{sub}{\vdash} \Delta$, and from identity rules we have

$$x : A \overset{\uparrow\downarrow}{\vdash} x \Rightarrow A \tag{3}$$

$$y : B \overset{\uparrow\downarrow}{\vdash} y \Rightarrow B \tag{4}$$

We complete the proof

$$\dfrac{\begin{array}{c}(1) \\ \Delta_{ab} \overset{\uparrow\downarrow}{\vdash} M \Rightarrow A \otimes B\end{array} \quad \begin{array}{c}IH(\mathcal{D}', (2, 3, 4)) \\ \Delta', x : A, y : B \overset{\uparrow\downarrow}{\vdash} N \Leftarrow C\end{array}}{\Delta_{ab} \overset{\uparrow\downarrow}{\vdash} \text{match } M \text{ with } (x, y) \text{ in } N \Leftarrow C}\ \otimes E$$

Case: $\multimap L$

$$\dfrac{\begin{array}{c}\mathcal{D}_1 \\ \Delta_1 \overset{seq}{\vdash} A\end{array} \quad \begin{array}{c}\mathcal{D}_2 \\ \Delta_2, B \overset{seq}{\vdash} C\end{array}}{\Delta_1, \Delta_2, A \multimap B \overset{seq}{\vdash} C}\ \multimap L$$

Assumptions:

$$\Delta_1' \overset{sub}{\vdash} \Delta_1 \tag{5}$$

$$\Delta_2' \overset{sub}{\vdash} \Delta_2 \tag{6}$$

$$\Delta_{ab} \overset{\uparrow\downarrow}{\vdash} M \Rightarrow A \multimap B \tag{7}$$

We begin by applying $\multimap E$ on our last assumption.

$$\dfrac{\overset{(7)}{\Delta_{ab} \overset{\uparrow\downarrow}{\vdash} M \Rightarrow A \multimap B} \quad \overset{IH(\mathcal{D}_1,(5))}{\Delta_1' \overset{\uparrow\downarrow}{\vdash} N \Leftarrow A}}{\Delta_{ab}, \Delta_1' \overset{\uparrow\downarrow}{\vdash} M\ N \Rightarrow B} \multimap E \tag{8}$$

Now we can complete the proof by an application of the inductive hypothesis on $\mathcal{D}_2$ and the derivation (8) as well as assumption 6.

$$\overset{IH(\mathcal{D}_2,(6,8))}{\Delta_1', \Delta_2', \Delta_{ab} \overset{\uparrow\downarrow}{\vdash} M' \Leftarrow C}$$

While we don't know specifically what term we have, we know it is possible to construct such a term. Other cases proceed similarly. □

Proof: Theorem 3 To prove the final theorem, we need to simultaneously prove one more theorem, since $\Rightarrow$ and $\Leftarrow$ are defined with references to each other. So this proof proceeds by simultaneous induction on the following two statements, over the derivation.

$$\text{If } \Delta \overset{\uparrow\downarrow}{\vdash} M \Leftarrow A \text{ then } \Delta \overset{nd}{\vdash} M : A$$

$$\text{If } \Delta \overset{\uparrow\downarrow}{\vdash} M \Rightarrow A \text{ then } \Delta \overset{nd}{\vdash} M : A$$

□

This proof should be self evident as a bidirectional proof with the two directions collapsed back into the single judgement : should still be a valid natural deduction proof (with some possible collapsing of the change of direction rule). We omit this proof and leave it as an exercise.

6 Curry-Howard Correspondence

Howard [1969] discovered the correspondence between one of the most fundamental models of computation, the lambda calculus and the implication fragment of natural deduction. This goes beyond just type checking. The proof reductions we did to demonstrate local soundness correspond to computational rules in the lambda calculus.

$$A \rightarrow B \qquad\qquad (\lambda x.M)N \Longrightarrow [N/x]M$$

This correspondence of course extends beyond just the lambda calculus, and encompasses the full natural deduction system and functional programming.

7 Full Bidirectional Rules

$$\frac{\Delta \vdash M \Longrightarrow A' \quad A = A'}{\Delta \vdash M \Longleftarrow A} \Rightarrow/\Leftarrow \qquad \frac{\Delta \vdash M \Longleftarrow A}{\Delta \vdash (M : A) \Longrightarrow A} \Leftarrow/\Rightarrow$$

$$\frac{}{x : A \vdash x \Longrightarrow A} \mathsf{hyp}$$

$$\frac{\Delta, x : A \vdash e \Longleftarrow B}{\Delta \vdash \lambda x.\, M \Longleftarrow A \multimap B} \multimap I$$

$$\frac{\Delta \vdash M \Longrightarrow A \multimap B \quad \Delta' \vdash N \Longleftarrow A}{\Delta, \Delta' \vdash M\, N \Longrightarrow B} \multimap E$$

$$\frac{\Delta \vdash M_\ell \Longleftarrow A_\ell \quad (\forall \ell \in L)}{\Delta \vdash \{\ell \Rightarrow M_\ell\}_{\ell \in L} \Longleftarrow \&\{\ell : A_\ell\}_{\ell \in L}} \& I \qquad \frac{\Delta \vdash M \Longrightarrow \&\{\ell : A_\ell\}_{\ell \in L} \quad (\ell \in L)}{\Delta \vdash M.\ell \Longrightarrow A_\ell} \& E$$

$$\frac{\Delta \vdash e_1 \Longleftarrow M \quad \Delta' \vdash N \Longleftarrow B}{\Delta, \Delta' \vdash (M, N) \Longleftarrow A \otimes B} \otimes I$$

$$\frac{\Delta \vdash M \Longrightarrow A \otimes B \quad \Delta', x_1 : A, x_2 : B \vdash N \Longleftarrow C}{\Delta, \Delta' \vdash \mathbf{match}\ M\ ((x_1, x_2) \Rightarrow N) \Longleftarrow C} \otimes E$$

$$\frac{}{\cdot \vdash (\,) \Longleftarrow \mathbf{1}} \mathbf{1}I \qquad \frac{\Delta \vdash M \Longrightarrow \mathbf{1} \quad \Delta' \vdash N \Longleftarrow C}{\Delta, \Delta' \vdash \mathbf{match}\ M\ ((\,) \Rightarrow N) \Longleftarrow C} \mathbf{1}E$$

$$\frac{\Delta \vdash M \Longleftarrow A_\ell}{\Delta \vdash \ell(M) \Longleftarrow \oplus\{\ell : A_\ell\}_{\ell \in L}} \oplus I$$

$$\frac{\Delta \vdash M \Longrightarrow \oplus\{\ell : A_\ell\}_{\ell \in L} \quad \Delta', x : A_\ell \vdash N_\ell \Longleftarrow C \quad (\forall \ell \in L)}{\Delta, \Delta' \vdash \mathbf{match}\ M\ (\ell(x) \Rightarrow N_\ell)_{\ell \in L} \Longleftarrow C} \oplus E$$

References

Henry DeYoung and Frank Pfenning. Data layout from a type-theoretic perspective. In *38th Conference on the Mathematical Foundations of Programming Semantics (MFPS 2022)*. Electronic Notes in Theoretical Informatics and Computer Science 1, 2022. URL https://arxiv.org/abs/2212.06321v6. Invited paper. Extended version available at https://arxiv.org/abs/2212.06321v3.pdf.

Henry DeYoung, Frank Pfenning, and Klaas Pruiksma. Semi-axiomatic sequent calculus. In Z. Ariola, editor, *5th International Conference on Formal Structures for Computation and Deduction (FSCD 2020)*, pages 29:1–29:22, Paris, France, June 2020. LIPIcs 167.

Jana Dunfield and Neel Krishnaswami. Bidirectional typing. *ACM Computing Surveys*, 98 (5):1–38, 2022.

Gerhard Gentzen. Untersuchungen über das logische Schließen. *Mathematische Zeitschrift*, 39:176–210, 405–431, 1935. English translation in M. E. Szabo, editor, *The Collected Papers of Gerhard Gentzen*, pages 68–131, North-Holland, 1969.

W. A. Howard. The formulae-as-types notion of construction. Unpublished note. An annotated version appeared in: *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, 479–490, Academic Press (1980), 1969.