Lecture Notes on Adjoint Logic

15-836: Substructural Logics
Frank Pfenning

Lecture 11
October 3, 2023

1 Introduction

In the last lecture we introduced LNL, a mixed linear/nonlinear logic that directly contains linear and nonlinear propositions instead of the exponential $!A$ (which could be defined as $\downarrow\uparrow A$). This eliminates some drawbacks of coding all nonlinear propositions via the exponential, but it has some of its own issues. For example, we have seen that there are two right rules for implication (one for $A \multimap B$ and one for $A \supset B$), and three left rules for implication (one for $A \multimap B$ and two for $A \supset B$).

One question is if we can streamline this so we would only have two rules implication (one right and one left rule) rather than five as in LNL. Another is if we can generalize the LNL approach to combine different logics, rather than just intuitionistic structural and linear logics. One answer to both questions is provided by *adjoint logic*, a general schema for combining certain classes of logics based on simple principles. The idea was first sketched by Reed [2009] and further developed by Pruiksma et al. [2018] and others (e.g., [Chargin, 2017, Licata and Shulman, 2016, Licata et al., 2017, Pruiksma and Pfenning, 2021]).

The generality of adjoint logic then yields a number of familiar logics such as *lax logic* Fairtlough and Mendler [1997] which is related to *computational monads* Moggi [1991], or the intuitionistic modal logic S4 [Pfenning and Davies, 2001] which is related to staged computation and metaprogramming [Davies and Pfenning, 2001].

In this and the following lecture we assume that exchange is always present as a structural property, although this is not necessary. We leave further discussion of ordered logic in the adjoint context either to a future lecture or a miniproject.

Mostly, it seems, we like to use weakening and contraction together. Occasionally, it is suitable to postulate just weakening or contraction in isolation. For example, if we want to allow failure and process cancelation, then weakening may be appropriate (not every process providing a channel may actually be used). The type system of Rust is also affine, that is, permits weakening but not contraction in certain ways that are not entirely captured here, but analogous. Another example is expressing *strictness analysis* in a language such as Haskell as a type system. When a function definitely uses its argument then it is strict, that is, it can (explicitly or implicitly) employ contraction but not weakening.

2 Adjoint Logic: The Basics

In adjoint logic every proposition has an intrinsic *mode of truth*, where each mode may or may not admit weakening and contraction. We define the meaning of the *connectives* uniformly at all modes by their right and left rules. So ultimately they are distinguished only by the structural properties their mode satisfies.

As an example, LNL as an instance of the adjoint logic framework would have two modes: S for structural propositions and L for linear propositions.

We also have a preorder $m \geq k$ on modes which expresses that the proof of a proposition A_k may depend on an antecedent B_m. Conversely, when $m \not\geq k$ then B_m may *not* be among the antecedents of a proof of m. We call this the *principle of independence*. A necessary use of this is in LNL, where a proof of a structural proposition A_S may not depend on a linear proposition A_L. So we have $\mathsf{S} > \mathsf{L}$ as our preorder (and, in particular $\mathsf{L} \not\geq \mathsf{S}$).

In addition to the usual connectives, adjoint logic also generalizes the shifts from LNL to go between two arbitrary modes. For ${\uparrow}_k^m A_k$ we require $m \geq k$ and for ${\downarrow}_m^\ell A_\ell$ we require $\ell \geq m$.

$$A_m ::= P_m \mid A_m \to B_m \mid A_m \times B_m \mid \mathbf{1} \mid A_m \mathbin{\&} B_m \mid \top \mid A_m + B_m \mid \mathbf{0} \mid {\uparrow}_k^m A_k \mid {\downarrow}_m^\ell A_\ell$$

We chose a new syntax, partially based on the reading of propositions as types. So $A \to B$ unifies $A \supset B$ and $A \multimap B$, $A \times B$ stands for $A \wedge B$ and $A \otimes B$, and $A + B$ stands for $A \vee B$ and $A \oplus B$. Even the logical constants $\mathbf{1}$, $\top$ and $\mathbf{0}$ should be thought of as having an intrinsic mode, even if we don't write them this way in the grammar.

We write $\sigma(m)$ for the set of structural properties satisfied by mode m, where $\sigma(m) \subseteq \{\mathsf{W}, \mathsf{C}\}$. As mentioned in the introduction, we always assume exchange. Based on the experience with cut elimination and validity (or the exponential), we require:

If $m \geq k$ then $\sigma(m) \supseteq \sigma(k)$.

And, indeed cut elimination fails if we omit this requirement. Furthermore, any dependence in a sequent must be allowed by the preorder among modes.

Whenever we write $\Delta \vdash A_m$ we require $\Delta \geq m$.

This *presupposition* means we can never ask a question $\Delta \vdash A_m$ unless for all $B_\ell \in \Delta$ we have $\ell \geq m$. We write Δ here for the antecedents because we treat the antecedents as a multiset. This means, weakening and contraction must be explicit rules for those modes that permit it.

Like Gentzen [1935], we read the inference rules of the sequent calculus bottom-up, as a means to construct a proof. When viewed in this direction, we need to ensure there are sufficient preconditions in the rule to ensure the premises satisfy our presupposition when the conclusion does.

As a start, write out the structural rules. Weakening applies to antecedents that permit it explicitly, and similarly for contraction.

$$\frac{\mathsf{W} \in \sigma(m) \quad \Delta \vdash C_r}{\Delta, A_m \vdash C_r}\,\mathsf{weaken} \qquad \frac{\mathsf{C} \in \sigma(m) \quad \Delta, A_m, A_m \vdash C_r}{\Delta, A_m \vdash C_r}\,\mathsf{contract}$$

Also generic are the rules of cut and identity. First, identity.

$$\frac{}{A_m \vdash A_m}\,\mathsf{id}$$

Cut requires a bit of thought. As usual, we start by writing down what we know directly, and then think about what else may be needed.

$$\frac{\Delta \vdash A_m \quad \Delta', A_m \vdash C_r}{\Delta, \Delta' \vdash C_r}\,\mathsf{cut?}$$

At first glance it might seem this should be it, but we remember that in LNL we needed three rules. For example, if $m = \mathsf{S}$ (that is, m is structural) then Δ may not contain any linear antecedents (that is, B_{L}). This issue surfaces here when we reason about our presupposition. Let's write in blue what we know and in red what we need to know for the premises.

$$\begin{array}{c} \Delta \geq m? \quad \Delta' \geq r, m \geq r? \\ \dfrac{\Delta \vdash A_m \qquad \Delta', A_m \vdash C_r}{\Delta, \Delta' \vdash C_r}\,\mathsf{cut?} \\ \Delta, \Delta' \geq r \end{array}$$

We see that we already know $\Delta' \geq r$ from the presupposition for the conclusion, but we know neither $\Delta \geq m$ nor $m \geq r$. These two conditions thus need to be explicitly enforced and we obtain:

$$\frac{\Delta \geq m \geq r \quad \Delta \vdash A_m \quad \Delta', A_m \vdash C_r}{\Delta, \Delta' \vdash C_r}\,\mathsf{cut}$$

You should convince yourself that in the case where we have just two modes, L and S, this gives rise exactly to the three forms of cut in LNL.

3 Logical Rules

Next we can define the logical rules, uniformly across the modes. We start with implication, which is almost a worst case for its complexity.

$$\frac{\Delta, A_m \vdash B_m}{\Delta \vdash A_m \to B_m}\ {\to}R$$

The presupposition tells us that $\Delta \geq m$, which is sufficient to ensure that $(\Delta, A_m) \geq m$. It is good there is no condition: since implication is negative, we would expect it to be right invertible and therefore not be subject to any conditions. For the left rule, matters are not quite as simple.

$$\begin{array}{c} \Delta \geq m? \quad \Delta' \geq r, m \geq r? \\ \dfrac{\Delta \vdash A_m \qquad \Delta', B_m \vdash C_r}{\Delta, \Delta', A_m \to B_m \vdash C_r}\ {\to}L? \\ (\Delta, \Delta') \geq r, m \geq r \end{array}$$

We see that $\Delta' \geq r$ and $m \geq r$ is already known, but $\Delta \geq m$ is not and must be added as a condition.

$$\frac{\Delta \geq m \quad \Delta \vdash A_m \quad \Delta', B_m \vdash C_r}{\Delta, \Delta', A_m \to B_m \vdash C_r}\ {\to}L$$

It turns out there isn't much of interest in the other rules. We only show the ones for tensor and unit.

$$\frac{\Delta \vdash A_m \quad \Delta' \vdash B_m}{\Delta, \Delta' \vdash A_m \times B_m}\ {\times}R \qquad \frac{\Delta, A_m, B_m \vdash C_r}{\Delta, A_m \times B_m \vdash C_r}\ {\times}L$$

$$\frac{}{\cdot \vdash \mathbf{1}}\ \mathbf{1}R \qquad \frac{\Delta \vdash C_r}{\Delta, 1 \vdash C_r}\ \mathbf{1}L$$

It is easy to see that for these (and the remaining rules except shifts) the presupposition for the conclusion immediately entails the presupposition for the premises and no additional conditions are needed. The rules are summarized in Figure 1.

4 The Shifts

The shifts generalize those from LNL, where $\uparrow$ would be replaced by $\uparrow_{\mathsf{L}}^{\mathsf{S}}$ and $\downarrow$ by $\downarrow_{\mathsf{L}}^{\mathsf{S}}$. Based on the polarity of the shifts in LNL we would expect the annotated shifts of adjoint logic to have the same polarities: $\uparrow$ should be negative and $\downarrow$ should be positive.

We can confirm this two ways: analyze the mode constraints in detail, and also derive the admissibility of the identity. Neither of these is proof, of course, which will be come back to in the next lecture.

We show first the rule with the known (blue) constraints and then the necessary constraints in red. The known constraint $m \geq k$ comes from the nature of the shift, since the upper mode must always be greater or equal to the lower mode.

$$\begin{array}{c} \Delta \geq k? \\ \dfrac{\Delta \vdash A_k}{\Delta \vdash {\uparrow}_k^m A_k}\,{\uparrow}R \\ \Delta \geq m, m \geq k \end{array} \qquad \dfrac{\Delta \vdash A_k}{\Delta \vdash {\uparrow}_k^m A_k}\,{\uparrow}R$$

As we might have predicted from the negative nature of the upshift, the presupposition of the premise follows from the presupposition of the conclusion.

In contrast, we expect some mode condition on the left rule for the upshift.

$$\begin{array}{c} \Delta \geq r, k \geq r? \\ \dfrac{\Delta, A_k \vdash C_r}{\Delta, {\uparrow}_k^m A_k \vdash C_r}\,{\uparrow}L \\ \Delta \geq r, m \geq k, m \geq r \end{array} \qquad \dfrac{k \geq r \quad \Delta, A_k \vdash C_r}{\Delta, {\uparrow}_k^m A_k \vdash C_r}\,{\uparrow}L$$

We show the rules for downshift in a similar form: collecting and checking dependence constraints and synthesizing the rule from them. If you try this yourself first, you will see that there is no leeway: the rules are uniquely determined.

$$\begin{array}{c} \Delta \geq \ell? \\ \dfrac{\Delta \vdash A_\ell}{\Delta \vdash {\downarrow}_m^\ell A_\ell}\,{\downarrow}R \\ \Delta \geq m, \ell \geq m \end{array} \qquad \dfrac{\Delta \geq \ell \quad \Delta \vdash A_\ell}{\Delta \vdash {\downarrow}_m^\ell A_\ell}\,{\downarrow}R$$

And finally the ${\downarrow}L$ rule, which requires no conditions.

$$\begin{array}{c} \Delta \geq r, \ell \geq r? \\ \dfrac{\Delta, A_\ell \vdash C_r}{\Delta, {\downarrow}_m^\ell A_\ell \vdash C_r}\,{\downarrow}L \\ \Delta \geq r, \ell \geq m, m \geq r \end{array} \qquad \dfrac{\Delta, A_\ell \vdash C_r}{\Delta, {\downarrow}_m^\ell A_\ell \vdash C_r}\,{\downarrow}L$$

5 Specific Logics as Instances of the Adjoint Schema

We obtain specific logics in the literature by specifying the modes, their dependence relation, and their structural properties.

Linear Logic. We obtain intuitionistic linear logic [Girard, 1987, Chang et al., 2003] with two modes, S and L where $\mathsf{S} > \mathsf{L}$ where $\sigma(\mathsf{S}) = \{\mathsf{W}, \mathsf{C}\}$ and $\sigma(\mathsf{L}) = \{\,\}$. Furthermore, we restrict the propositions at mode S as

$$A_{\mathsf{S}} ::= {\uparrow}_{\mathsf{L}}^{\mathsf{S}} A_{\mathsf{L}}$$

Then $!A_{\mathsf{L}} \triangleq {\downarrow}_{\mathsf{L}}^{\mathsf{S}} {\uparrow}_{\mathsf{L}}^{\mathsf{S}} A_{\mathsf{L}}$. We also rule out the shifts ${\uparrow}_m^m$ and ${\downarrow}_m^m$.

LNL. We obtain LNL [Benton, 1994] with the same modes, but the structural layer has a full set of connectives. We only rule out ${\uparrow}_m^m$ and ${\downarrow}_m^m$.

Intuitionistic S4. We obtain intuitionistic S4 [Pfenning and Davies, 2001] with two modes, V (validity) and T (truth), with $\sigma(\mathsf{V}) = \sigma(\mathsf{T}) = \{\mathsf{W}, \mathsf{C}\}$. The mode V is restricted analogously to S in linear logic:

$$A_{\mathsf{V}} ::= {\uparrow}_{\mathsf{T}}^{\mathsf{V}} A_{\mathsf{T}}$$

We also rule out ${\uparrow}_m^m$ and ${\downarrow}_m^m$. We define $\Box A_{\mathsf{T}} \triangleq {\downarrow}_{\mathsf{T}}^{\mathsf{V}} {\uparrow}_{\mathsf{T}}^{\mathsf{V}} A_{\mathsf{T}}$. As a composition of the two adjoint shift operators, $\Box$ is a *comonad*.

Perhaps somewhat surprisingly, we do not obtain the monad $\Diamond A_{\mathsf{T}}$ from a composition of shifts, although we can obtain $\bigcirc A_{\mathsf{T}}$, a strong monad and the basis for lax logic.

Lax Logic. We obtain *lax logic* [Fairtlough and Mendler, 1997] with two modes, T (truth) and X (lax truth), with $\mathsf{T} > \mathsf{X}$ and $\sigma(\mathsf{T}) = \sigma(\mathsf{X}) = \{\mathsf{W}, \mathsf{C}\}$. The mode X is restricted to

$$A_{\mathsf{X}} ::= {\downarrow}_{\mathsf{X}}^{\mathsf{T}} A_{\mathsf{T}}$$

and then $\bigcirc A_{\mathsf{T}} \triangleq {\uparrow}_{\mathsf{X}}^{\mathsf{T}} {\downarrow}_{\mathsf{X}}^{\mathsf{T}} A_{\mathsf{T}}$. The $\bigcirc$ modality is a (strong) monad.

It is now easy to extend and combine these. For examples, we can have a language for staged computation [Davies and Pfenning, 2001] where quoted expressions are drawn directly from the layer for validity. Or we can have a language that has both a monad and a comonad, with different structural properties.

6 Summary

The rules for adjoint logic are summarized in Figure 1.

Syntax with $\ell \geq m \geq k$ and $\sigma(m) \subseteq \{\mathsf{W}, \mathsf{C}\}$.

$$A_m ::= P_m \mid A_m \to B_m \mid A_m \times B_m \mid \mathbf{1} \mid A_m \mathbin{\&} B_m \mid \top \mid A_m + B_m \mid \mathbf{0} \mid {\uparrow}_k^m A_k \mid {\downarrow}_m^\ell A_\ell$$

Rules.

$$\frac{\mathsf{W} \in \sigma(m) \quad \Delta \vdash C_r}{\Delta, A_m \vdash C_r}\ \mathsf{weaken} \qquad \frac{\mathsf{C} \in \sigma(m) \quad \Delta, A_m, A_m \vdash C_r}{\Delta, A_m \vdash C_r}\ \mathsf{contract}$$

$$\frac{}{A_m \vdash A_m}\ \mathsf{id} \qquad \frac{\Delta \geq m \geq r \quad \Delta \vdash A_m \quad \Delta', A_m \vdash C_r}{\Delta, \Delta' \vdash C_r}\ \mathsf{cut}$$

$$\frac{\Delta \vdash A_k}{\Delta \vdash {\uparrow}_k^m A_k}\ {\uparrow}R \qquad \frac{k \geq r \quad \Delta, A_k \vdash C_r}{\Delta, {\uparrow}_k^m A_k \vdash C_r}\ {\uparrow}L$$

$$\frac{\Delta \geq \ell \quad \Delta \vdash A_\ell}{\Delta \vdash {\downarrow}_m^\ell A_\ell}\ {\downarrow}R \qquad \frac{\Delta, A_\ell \vdash C_r}{\Delta, {\downarrow}_m^\ell A_\ell \vdash C_r}\ {\downarrow}L$$

$$\frac{\Delta, A_m \vdash B_m}{\Delta \vdash A_m \to B_m}\ {\to}R \qquad \frac{\Delta \geq m \quad \Delta \vdash A_m \quad \Delta', B_m \vdash C_r}{\Delta, \Delta', A_m \to B_m \vdash C_r}\ {\to}L$$

$$\frac{\Delta \vdash A_m \quad \Delta' \vdash B_m}{\Delta, \Delta' \vdash A_m \times B_m}\ {\times}R \qquad \frac{\Delta, A_m, B_m \vdash C_r}{\Delta, A_m \times B_m \vdash C_r}\ {\times}L$$

$$\frac{}{\cdot \vdash \mathbf{1}}\ \mathbf{1}R \qquad \frac{\Delta \vdash C_r}{\Delta, 1 \vdash C_r}\ \mathbf{1}L$$

$$\frac{\Delta \vdash A_m \quad \Delta \vdash B_m}{\Delta \vdash A_m \mathbin{\&} B_m}\ \&R \qquad \frac{\Delta, A_m \vdash C_r}{\Delta, A_m \mathbin{\&} B_m \vdash C_r}\ \&L_1 \quad \frac{\Delta, B_m \vdash C_r}{\Delta, A_m \mathbin{\&} B_m \vdash C_r}\ \&L_2$$

$$\frac{}{\Delta \vdash \top}\ \top R \qquad \text{no } \top L \text{ rule}$$

$$\frac{\Delta \vdash A_m}{\Delta \vdash A_m + B_m}\ {+}R_1 \quad \frac{\Delta \vdash B_m}{\Delta \vdash A_m + B_m}\ {+}R_2 \qquad \frac{\Delta, A_m \vdash C_r \quad \Delta, B_m \vdash C_r}{\Delta, A_m + B_m \vdash C_r}\ {+}L$$

$$\text{no } \mathbf{0}R \text{ rule} \qquad \frac{}{\Delta, \mathbf{0} \vdash C_r}\ \mathbf{0}L$$

Figure 1: Adjoint Logic

References

Nick Benton. A mixed linear and non-linear logic: Proofs, terms and models. In Leszek Pacholski and Jerzy Tiuryn, editors, *Selected Papers from the 8th International Workshop on Computer Science Logic (CSL'94)*, pages 121–135, Kazimierz, Poland, September 1994. Springer LNCS 933. An extended version appears as Technical Report UCAM-CL-TR-352, University of Cambridge.

Bor-Yuh Evan Chang, Kaustuv Chaudhuri, and Frank Pfenning. A judgmental analysis of linear logic. Technical Report CMU-CS-03-131R, Carnegie Mellon University, Department of Computer Science, December 2003.

William Chargin. A general system of adjoint logic. Honors thesis, Carnegie Mellon University, December 2017.

Rowan Davies and Frank Pfenning. A modal analysis of staged computation. *Journal of the ACM*, 48(3):555–604, May 2001.

M. Fairtlough and M.V. Mendler. Propositional lax logic. *Information and Computation*, 137(1):1–33, August 1997.

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.

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

Daniel R. Licata and Michael Shulman. Adjoint logic with a 2-category of modes. In *International Symposium on Logical Foundations of Computer Science (LFCS)*, pages 219–235. Springer LNCS 9537, January 2016.

Daniel R. Licata, Michael Shulman, and Mitchell Riley. A fibrational framework for substructural and modal logics. In Dale Miller, editor, *Proceedings of the 2nd International Conference on Formal Structures for Computation and Deduction (FSCD'17)*, pages 25:1–25:22, Oxford, UK, September 2017. LIPIcs.

Eugenio Moggi. Notions of computation and monads. *Information and Computation*, 93(1):55–92, 1991.

Frank Pfenning and Rowan Davies. A judgmental reconstruction of modal logic. *Mathematical Structures in Computer Science*, 11:511–540, 2001. Notes to an invited talk at the *Workshop on Intuitionistic Modal Logics and Applications* (IMLA'99), Trento, Italy, July 1999.

Klaas Pruiksma and Frank Pfenning. A message-passing interpretation of adjoint logic. *Journal of Logical and Algebraic Methods in Programming*, 120(100637), 2021.

Klaas Pruiksma, William Chargin, Frank Pfenning, and Jason Reed. Adjoint logic. Unpublished manuscript, April 2018. URL `http://www.cs.cmu.edu/~fp/papers/adjoint18b.pdf`.

Jason Reed. A judgmental deconstruction of modal logic. Unpublished manuscript, May 2009. URL `http://www.cs.cmu.edu/~jcreed/papers/jdml2.pdf`.