

Lecture Notes on Combining Logics

15-816: Substructural Logics
Frank Pfenning

Lecture 12
October 6, 2016

In this lecture we will introduce a general approach to combining logics, using the special case of ordered and linear logic. Other examples of this construction are *monads* in the sense used in functional programming to integrate effects [Mog89] and LNL [Ben94] and the modal logic S4 which gives rise a type system for quotation [DP01, PD01]. A systematic logical study was initiated by Reed [Ree09] and subsequently extended and applied to concurrency [PG15].

1 Connecting Two Layers with Shifts

We start by writing down ordered and linear logic in a way that exhibits the correspondence between the propositions. We use a subscript O to indicate an *ordered mode* and L to indicate *linear mode*. In a later lecture we will introduce other modes. p_o and p_l stand for ordered and linear propositional variables, respectively.

Linear $A_l, B_l ::= p_l \mid A_l \oplus B_l \mid A_l \& B_l \mid \mathbf{1} \mid A_l \multimap B_l \mid A_l \otimes B_l$
Ordered $A_o, B_o ::= p_o \mid A_o \oplus B_o \mid A_o \& B_o \mid \mathbf{1} \mid A_o \setminus B_o \mid B_o / A_o \mid A_o \bullet B_o \mid A_o \circ B_o$

Strictly speaking, there are also two versions of internal and external choice and the unit $\mathbf{1}$, but since their rules and behavior are independent of whether we have ordered or unordered antecedents, so we write them the same way. At this point there is no overlap, so the two logics are not connected in any way. To allow ordered proposition to mention linear ones and vice versa, we add two *shift* operators. Since we view linear propositions as

more powerful antecedents (they can move around, after all, while the ordered ones are locked into place) we think of L as stronger than O (written $L > O$) and we use an uparrow to go from ordered to linear and a downarrow to go from linear to ordered. Officially, this arrow is annotated with the two modes. We will omit them in today's lecture since O and L will be the only modes.

Linear $A_L, B_L ::= p_L \mid A_L \oplus B_L \mid A_L \& B_L \mid \mathbf{1} \mid A_L \multimap B_L \mid A_L \otimes B_L \mid \uparrow_O^L A_O$
 Ordered $A_O, B_O ::= p_O \mid A_O \oplus B_O \mid A_O \& B_O \mid \mathbf{1} \mid A_O \setminus B_O \mid B_O / A_O \mid A_O \bullet B_O \mid A_O \circ B_O \mid \downarrow_O^L A_L$

We read $\uparrow_O^L A_O$ as *from ordered A_O up to linear* or just *up A_O* , and $\downarrow_O^L A_L$ as *from linear A_L down to ordered* or just *down A_L* .

2 Counterexample: Swap

We refer to unary logical connectives that mediate between different modes of truth as *modal operators* or *modalities*. In traditional philosophical logic such modes could be *knowledge*, *belief*, *necessity*, or *possibility*, among many others that have been investigated. Here the mode determines whether we are considering ordered or linear logic.

We need some principled understanding of the shift modalities before we can write correct left and right rules to define their meaning. I hesitate to write down wrong rules, lest they burn themselves into your memory, but it is worth playing the *what if* game for a little to understand the restrictions we will eventually impose. Consider the judgment

$$\Delta_L ; \Omega_O \vdash C_m$$

where Δ_L collects all *linear* antecedents, Ω_O collects all *ordered* antecedents, and C_m is the succedent with the mode m either O or L. The first attempt at the rules, **which will be shown to be wrong**, would be to just strip the modality in all four cases. We take care to place each proposition in its appropriate antecedent zone.

$$\begin{array}{cc} \frac{\Delta ; \Omega \vdash A_O}{\Delta ; \Omega \vdash \uparrow A_O} \uparrow R?? & \frac{\Delta ; \Omega_1 A_O \Omega_2 \vdash C_m}{\Delta, \uparrow A_O ; \Omega_1 \Omega_2 \vdash C_m} \uparrow L?? \\ \frac{\Delta ; \Omega \vdash A_L}{\Delta ; \Omega \vdash \downarrow A_L} \downarrow R?? & \frac{\Delta, A_L ; \Omega_1 \Omega_2 \vdash C_m}{\Delta ; \Omega_1 (\downarrow A_L) \Omega_2 \vdash C_m} \downarrow L?? \end{array}$$

Next comes the counterexample which shows that this combined logic does not make sense in the way we intended.

$$\begin{array}{c}
 \frac{}{\cdot ; A_0 \vdash A_0} \text{id} \quad \frac{}{\cdot ; B_0 \vdash B_0} \text{id} \quad \frac{}{\cdot ; A_0 \vdash A_0} \text{id} \\
 \frac{}{\cdot ; A_0 \vdash A_0} \text{id} \quad \frac{}{\cdot ; B_0 A_0 \vdash B_0 \bullet A_0} \bullet R \\
 \frac{}{\cdot ; A_0 \vdash \uparrow A_0} \uparrow R?? \quad \frac{}{\uparrow A_0 ; B_0 \vdash B_0 \bullet A_0} \uparrow L?? \\
 \frac{}{\cdot ; A_0 B_0 \vdash B_0 \bullet A_0} \text{cut}?? \\
 \frac{}{\cdot ; A_0 \bullet B_0 \vdash B_0 \bullet A_0} \bullet L
 \end{array}$$

Why is this troublesome? The conclusion proves that fuse is commutative, which means that the ordered connectives no longer have the meaning we expect. Moreover, something must fail in cut elimination: our conclusion contains only ordered propositions, and in ordered logic it is certainly *not* the case that fuse is commutative. There can be no cut-free proof of the endsequent in the combined logic, so cut elimination must fail.

The issue is that $\uparrow A_0$ is a *linear* proposition, which is therefore “mobile” in the sense that it can move around arbitrarily among the antecedents. If we could indeed prove $\cdot ; A_0 \vdash \uparrow A_0$ this means we could make any ordered proposition mobile, place it wherever we like (the use of $\uparrow L??$ in the counterexample) when we make it ordered again. This “trick” was exploited to illegally move A_0 from the left of B_0 to its right.

Our analysis is that the proof of *linear* proposition $\uparrow A_0$ should not be allowed to depend on the *ordered* proposition A_0 , because the linear succedent can be used in ways not justified by the ordered antecedent. This would rule out the very judgment $\cdot ; A_0 \vdash \uparrow A_0$ so the cut rule cannot be correct, nor can the application of $\uparrow R??$. On the other hand, $\uparrow L??$ looks defensible.

We can also take a look at what the result of a hypothetical cut reduction would be. It is a principal case, since the cut formula $\uparrow A_0$ is the principal formula of both inferences. We obtain:

$$\begin{array}{c}
 \frac{}{\cdot ; A_0 \vdash A_0} \text{id} \quad \frac{}{\cdot ; B_0 \vdash B_0} \text{id} \quad \frac{}{\cdot ; A_0 \vdash A_0} \text{id} \\
 \frac{}{\cdot ; A_0 \vdash A_0} \text{id} \quad \frac{}{\cdot ; B_0 A_0 \vdash B_0 \bullet A_0} \bullet R \\
 \frac{}{\cdot ; A_0 B_0 \vdash B_0 \bullet A_0} \text{cut}?? \\
 \frac{}{\cdot ; A_0 \bullet B_0 \vdash B_0 \bullet A_0} \bullet L
 \end{array}$$

We see that the conclusion of the cut would have antecedents $B_o A_o$ when before the cut reduction it was $A_o B_o$. Thus, the cut reduction fails.

3 The Declaration of Independence

The considerations in the previous section lead us to the following *principle of independence*.

Independence Principle: The proof of a linear succedent cannot depend on ordered antecedents.

The general form of this principle in later lectures will be: *The proof of a succedent with mode m can only depend on antecedents with modes equal to or stronger than m .* Here we consider a mode stronger if it supports more structural properties such as exchange, weakening, or contraction.

This means we really have only the following two judgment forms.

$$\begin{array}{c} \Delta_L ; \Omega_o \vdash A_o \\ \Delta_L \vdash A_L \end{array}$$

Ordered succedents A_o can depend on linear and ordered antecedents, while linear succedents A_L can only depend on linear antecedents. We will later unify these with others into a single judgment form with some pre-suppositions to ensure they are meaningful.

Now we can go through our rules and subject them to the independence principle to arrive at the following collection of right and left rules.

$$\begin{array}{cc} \frac{\Delta ; \cdot \vdash A_o}{\Delta \vdash \uparrow A_o} \uparrow R & \frac{\Delta ; \Omega_1 A_o \Omega_2 \vdash C_o}{\Delta, \uparrow A_o ; \Omega_1 \Omega_2 \vdash C_o} \uparrow L \\ \frac{\Delta \vdash A_L}{\Delta ; \cdot \vdash \downarrow A_L} \downarrow R & \frac{\Delta, A_L ; \Omega_1 \Omega_2 \vdash C_o}{\Delta ; \Omega_1 (\downarrow A_L) \Omega_2 \vdash C_o} \downarrow L \end{array}$$

Please make sure you go through these and understand how they arise from the previous dubious rules simply by heeding the independence principle. As an example, with consider $\downarrow R$. We had proposed

$$\frac{\Delta ; \Omega \vdash A_L}{\Delta ; \Omega \vdash \downarrow A_L} \downarrow R??$$

In the premise, there cannot be any ordered antecedents, leading to a better approximation:

$$\frac{\Delta \vdash A_L}{\Delta ; \Omega \vdash \downarrow A_L} \downarrow R??$$

But every antecedent in Ω must be used—we cannot simply drop them at this rule which would be tantamount to *weakening*. So we must require there to be no ordered antecedents and we arrive at the correct rule.

$$\frac{\Delta \vdash A_L}{\Delta ; \cdot \vdash \downarrow A_L} \downarrow R$$

4 Cut and Identity

The rules for cut and identity can now be derived from the same considerations of independence. Identity is particularly straightforward.

$$\frac{}{\cdot ; A_o \vdash A_o} \text{id}_o \quad \frac{}{A_L \vdash A_L} \text{id}_L$$

For cut, it turns out there are three rules because independence rules out one of the four combinations of the judgment forms.

$$\frac{\Delta' ; \Omega' \vdash A_o \quad \Delta ; \Omega_1 A_o \Omega_2 \vdash C_o}{\Delta, \Delta' ; \Omega_1 \Omega' \Omega_2 \vdash C_o} \text{cut}_{oo}$$

$$\frac{\Delta' \vdash A_L \quad \Delta, A_L ; \Omega \vdash C_o}{\Delta, \Delta' ; \Omega \vdash C_o} \text{cut}_{Lo} \quad \frac{\Delta' \vdash A_L \quad \Delta, A_L \vdash C_L}{\Delta, \Delta' \vdash C_L} \text{cut}_{LL}$$

5 Other Propositional Rules

The other propositional rules are carried over or straightforwardly adapted from ordered or linear logic. We show only two examples: the rules for $A_o \setminus B_o$ and $A_L \multimap B_L$. The $\multimap L$ rules split into two, for similar reasons why there are two version of the cut rule for linear propositions.

$$\frac{\Delta ; A_o \Omega \vdash B_o}{\Delta ; \Omega \vdash A_o \setminus B_o} \setminus R \quad \frac{\Delta' ; \Omega' \vdash A_o \quad \Delta ; \Omega_1 B_o \Omega_2 \vdash C_o}{\Delta, \Delta' ; \Omega_1 \Omega' (A_o \setminus B_o) \Omega_2 \vdash C_o} \setminus L$$

$$\frac{\Delta, A_L \vdash B_L}{\Delta \vdash A_L \multimap B_L} \multimap R \quad \frac{\Delta' \vdash A_L \quad \Delta, B_L \vdash C_L}{\Delta, \Delta', A_L \multimap B_L \vdash C_L} \multimap L_L \quad \frac{\Delta' \vdash A_L \quad \Delta, B_L ; \Omega \vdash C_o}{\Delta, \Delta', A_L \multimap B_L ; \Omega \vdash C_o} \multimap L_o$$

6 Roundtrips

Including shifts in the language of propositions allows us to consider *round trips* from ordered to linear logic and back, and from ordered to linear and back. How does $\downarrow\uparrow A_o$ compare to A_o ? Conversely, how does $\uparrow\downarrow A_l$ compare to A_l ? We just try construct a cut-free derivation, bottom up, for each of these proposition to see which hold and which do not.

$$\begin{array}{c}
 \text{no rule applicable} \\
 \cdot ; A_o \vdash \downarrow\uparrow A_o
 \end{array}
 \qquad
 \begin{array}{c}
 \frac{}{\cdot ; A_o \vdash A_o} \text{id}_o \\
 \frac{}{\uparrow A_o ; \cdot \vdash A_o} \uparrow R \\
 \frac{}{\cdot ; \downarrow\uparrow A_o \vdash A_o} \downarrow L
 \end{array}$$

$$\begin{array}{c}
 \frac{}{A_l \vdash A_l} \text{id}_l \\
 \frac{}{A_l ; \cdot \vdash \downarrow A_l} \downarrow R \\
 \frac{}{A_l \vdash \uparrow\downarrow A_l} \uparrow R
 \end{array}
 \qquad
 \begin{array}{c}
 \text{no rule applicable} \\
 \uparrow\downarrow A_l \vdash A_l
 \end{array}$$

In two sequents we get stuck immediately due to the restrictions imposed by the independence principle.

So if we know A_l we can conclude $\uparrow\downarrow A_l$ via a roundtrip to an ordered proposition but not the other way around. Conversely, if we are trying to prove A_o it is sufficient to prove $\downarrow\uparrow A_o$.

The observations

$$\begin{array}{l}
 A_o \not\vdash \downarrow\uparrow A_o \\
 \downarrow\uparrow A_o \vdash A_o \\
 A_l \vdash \uparrow\downarrow A_l \\
 \uparrow\downarrow A_l \not\vdash A_l
 \end{array}$$

suggest that $\downarrow\uparrow A_o \simeq \Box A_o$ behaves like the modality of *necessity*, which can be characterized categorically as a *comonad*, while $\uparrow\downarrow A_o \simeq \Diamond A_l$ behaves like a form of *possibility*, which can be characterized categorically as a *strong monad*.

We pursue this conjecture a little further to see if the shifts distribute over implications. They do. While we distribute the shift, linear implications will turn into over/under and vice versa. We prefer $A \setminus B$ here since the order of its arguments is consistent with linear implication, but this

choice is arbitrary. We start

$$\frac{\frac{\cdot ; (\downarrow A) (\downarrow(A \multimap B)) \vdash \downarrow B}{\cdot ; \cdot \vdash \downarrow(A \multimap B) \setminus \downarrow A \setminus \downarrow B} \text{?}}{\cdot ; \cdot \vdash \downarrow(A \multimap B) \setminus \downarrow A \setminus \downarrow B} \setminus R \times 2$$

We pause here to note that the $\downarrow R$ rule is not applicable since the ordered context is not empty. So we apply the $\downarrow L$ rules twice, to empty it out. The remainder is straightforward.

$$\frac{\frac{\frac{\frac{\frac{}{A_L \vdash A_L} \text{id}}{A_L, A_L \multimap B_L \vdash B_L} \multimap L_L}{A_L, A_L \multimap B_L ; \cdot \vdash \downarrow B_L} \downarrow R}{\cdot ; (\downarrow A_L) (\downarrow(A_L \multimap B_L)) \vdash \downarrow B_L} \downarrow L \times 2}{\cdot ; \cdot \vdash \downarrow(A_L \multimap B_L) \setminus \downarrow A_L \setminus \downarrow B_L} \setminus R \times 2$$

Similarly, \uparrow distributes over ordered implication $A \setminus B$. We pause after two steps.

$$\frac{\frac{\frac{}{\uparrow(A_O \setminus B_O), \uparrow A_O \vdash \uparrow B_O} \text{?}}{\cdot \vdash \uparrow(A_O \setminus B_O) \multimap \uparrow A_O \multimap \uparrow B_O} \multimap R \times 2}{\cdot \vdash \uparrow(A_O \setminus B_O) \multimap \uparrow A_O \multimap \uparrow B_O} \multimap R \times 2$$

Now we need to apply $\uparrow R$ so we can then apply $\uparrow L$ twice to place the two antecedents in the ordered zone in the correct order.

$$\frac{\frac{\frac{\frac{\frac{}{A_O \vdash A_O} \text{id}_O}{\cdot ; A_O (A_O \setminus B_O) \vdash B_O} \setminus L}{\uparrow(A_O \setminus B_O), \uparrow A_O ; \cdot \vdash B_O} \uparrow L \times 2}{\uparrow(A_O \setminus B_O), \uparrow A_O \vdash \uparrow B_O} \uparrow R}{\cdot \vdash \uparrow(A_O \setminus B_O) \multimap \uparrow A_O \multimap \uparrow B_O} \multimap R \times 2$$

The observations

$$\begin{aligned} & \cdot ; \cdot \vdash \downarrow(A_L \multimap B_L) \setminus \downarrow A_L \setminus \downarrow B_L \\ & \cdot \vdash \uparrow(A_O \setminus B_O) \multimap \uparrow A_O \multimap \uparrow B_O \end{aligned}$$

provide further evidence for our conjectures of the modal nature of $\downarrow\uparrow$ and $\uparrow\downarrow$ and necessity and strong possibility (see Exercises 1 and 2).

7 Operational Interpretation of Shifts

When we assign processes as proof terms we mark channels ordered or linear, written as x_o or x_L . This doesn't affect the identity of a channel, only how it is used.

$$\begin{aligned} (x_{1L}:A_{1L}, \dots, x_{nL}:A_{nL}) ; (y_{1O}:B_{1O} \dots y_{kO}:B_{kO}) \vdash P :: (z_o : C_o) \\ (x_{1L}:A_{1L}, \dots, x_{nL}:A_{nL}) \vdash P :: (z_L : C_L) \end{aligned}$$

As we remarked several times, the operational interpretation of ordered and their corresponding linear connectives is exactly the same. Order only imposes a restriction on the use of ordered channels. Therefore we could assume that the shifts have no essential *operational* purpose. Indeed, looking at the rules it seems that a message of shift corresponds to communicating a mutual agreement that a channel changes status, from ordered to linear or vice versa. We have to determine which of the rules carries information and which does not.

$$\frac{\Delta ; \cdot \vdash A_o}{\Delta \vdash \uparrow A_o} \uparrow R \qquad \frac{\Delta ; \Omega_1 A_o \Omega_2 \vdash C_o}{\Delta, \uparrow A_o ; \Omega_1 \Omega_2 \vdash C_o} \uparrow L$$

We see that $\uparrow R$ can always be applied and thus carries no information, while an antecedent $\uparrow A_o$ may have to wait until the succedent is ordered. It also implicitly carries the information on *where* in the context to insert A_o . This means $\uparrow R$ will receive and $\uparrow L$ will send.

$$\begin{aligned} \frac{\Delta ; \cdot \vdash P :: (x_o : A_o)}{\Delta \vdash \text{shift} \leftarrow \text{recv } x_L ; P :: (x_L : \uparrow A_o)} \uparrow R \\ \frac{\Delta ; \Omega_1 (x_o:A_o) \Omega_2 \vdash Q :: (z_o : C_o)}{\Delta, x_L:\uparrow A_o ; \Omega_1 \Omega_2 \vdash \text{send } x_L \text{ shift} ; Q :: (z_o : C_o)} \uparrow L \end{aligned}$$

For the down shift modality, the roles are reversed. $\downarrow R$ does not always apply since there may be ordered antecedents. On the other hand, $\downarrow L$ always applies (read bottom-up, of course) and therefore it carries no information and will receive.

$$\begin{aligned} \frac{\Delta \vdash P :: (x_L : A_L)}{\Delta ; \cdot \vdash \text{send } x_o \text{ shift} ; P :: (x_o : \downarrow A_L)} \downarrow R \\ \frac{\Delta, x_L:A_L ; \Omega_1 \Omega_2 \vdash Q :: (z_o : C_o)}{\Delta ; \Omega_1 (x_o:\downarrow A_L) \Omega_2 \vdash \text{shift} \leftarrow \text{recv } x_o ; Q :: (z_o : C_o)} \downarrow L \end{aligned}$$

As with internal/external choice and ordered implication/fuse, the up and down shifts form dual pairs and therefore use the same program construct. We can always tell from the situation which of the two is meant.

The operational rules are straightforward.

$$\frac{\text{proc}(x_L, \text{shift} \leftarrow \text{recv } x_L ; P) \quad \text{proc}(z_O, \text{send } x_L \text{ shift} ; Q)}{\text{proc}(x_O, P) \quad \text{proc}(z_O, Q)} \uparrow C$$

$$\frac{\text{proc}(x_O, \text{send } x_O \text{ shift} ; P) \quad \text{proc}(z_O, \text{shift} \leftarrow \text{recv } x_O ; Q)}{\text{proc}(x_L, P) \quad \text{proc}(z_O, Q)} \downarrow C$$

8 Example: A Linear Client Using an Ordered Service

By the independence principle, a process that provides along a linear channel cannot use an ordered channel. But if we have an ordered process that uses linear channels internally, we can use an ordered storage server (say, a stack) to store the linear channel $x_L : A_L$. In order to allow this we have to coerce the linear channel to be ordered, of type $\downarrow A_L$, so we define

$x_L : A_L ; \cdot \vdash \text{down} :: (y_O : \downarrow A_L)$
 $y_O \leftarrow \text{down} \leftarrow x_L =$
 send y_O shift ;
 $y_L \leftarrow x_L$

The we assume we have ordered type

$\text{stack}_A = \&\{\text{ins} : A \setminus \text{stack}_A$
 $\text{del} : \oplus\{\text{none} : \mathbf{1}, \text{some} : A \bullet \text{stack}\}\}$

with the usual ordered implementation. Now assume we are in a situation where we own a channel of type A_L and also a stack with elements of type $\downarrow A_L$

$$(x_L : A_L) ; (s : \text{stack}_{\downarrow A_L}) \vdash (z_O : C_O)$$

In order to push x_L onto the stack we first lower it to be ordered. Upon retrieval, we do the opposite. In the types we omit the succedent since it plays no role in this code.

$w_O \leftarrow \text{down} \leftarrow x_L ;$ % $(x_L : A_L) ; (s : \text{stack}_{\downarrow A_L}) \vdash \dots$
 $s.\text{ins} ; \text{send } s \ w_O ;$ % $\cdot ; (w_O : \downarrow A_L) (s : \text{stack}_{\downarrow A_L}) \vdash \dots$
 $\dots \text{ other operations } \dots$ % $\cdot ; (s : \text{stack}_{\downarrow A_L}) \vdash \dots$

```

s.del ;                               %  · ; s : ⊕ {none : 1, some : ↓AL • stack↓AL} ⊢ ...
case s (none ⇒ wait s ; ...
      | some ⇒ uo ← recv s ;         %  · ; (uo:↓AL) (s:stack↓AL) ⊢ ...
      shift ← recv uo ; %  (uL:AL) ; (s:stack↓AL) ⊢ ...
      ...)

```

9 Identity Expansion

To verify harmony we should check cut reduction and identity expansion. We begin with the simpler identity expansion.

$$\begin{array}{c}
\frac{}{\cdot ; A_o \vdash A_o} \text{id}_o \quad \frac{}{A_L \vdash A_L} \text{id}_L \\
\frac{}{\cdot ; A_o \vdash A_o} \uparrow L \quad \frac{}{A_L \vdash A_L} \downarrow R \\
\frac{\uparrow A_o ; \cdot \vdash A_o}{\uparrow A_o \vdash \uparrow A_o} \uparrow R \quad \frac{A_L ; \cdot \vdash \downarrow A_L}{\cdot ; \downarrow A_L \vdash \downarrow A_L} \downarrow L
\end{array}$$

10 Cut Reduction

Cut reduction usually has the key insight in the proof of admissibility of cut and therefore ultimately cut elimination. We only show one case of cut reduction (for $\uparrow A_o$) and elide other cases as well as admissibility of cut (see also Exercise 3).

Assume the first premise of a cut ends in $\uparrow R$ rule.

$$\begin{array}{c}
\frac{\frac{\mathcal{D}'}{\Delta' ; \cdot \vdash A_o} \uparrow R \quad \frac{\mathcal{E}}{\Delta, \uparrow A_o ; \Omega \vdash C_o}}{\Delta, \Delta' ; \Omega \vdash C_o} \text{cut}_{Lo} \quad \frac{\frac{\mathcal{D}'}{\Delta' ; \cdot \vdash A_o} \uparrow R \quad \frac{\mathcal{E}}{\Delta, \uparrow A_o \vdash C_L}}{\Delta, \Delta' \vdash C_L} \text{cut}_{LL}
\end{array}$$

The $\uparrow L$ rule has the form

$$\frac{\Delta ; \Omega_1 A_o \Omega_2 \vdash C_o}{\Delta, \uparrow A_o ; \Omega_1 \Omega_2 \vdash C_o} \uparrow L$$

so only the first of these two cases will result in a cut reduction. The second case above will lead to some commuting conversions, if A_o is a side formula

of inferences in \mathcal{E} . The only case for reduction then has the form

$$\frac{\frac{\mathcal{D}'}{\Delta' ; \cdot \vdash A_o} \uparrow R \quad \frac{\mathcal{E}'}{\Delta ; \Omega_1 A_o \Omega_2 \vdash C_o} \uparrow L}{\Delta, \Delta' ; \Omega_1 \Omega_2 \vdash C_o} \text{cut}_{\text{Lo}}$$

which reduces to

$$\frac{\mathcal{D}' \quad \mathcal{E}'}{\Delta, \Delta' ; \Omega_1 \Omega_2 \vdash C_o} \text{cut}_{\text{oo}}$$

Exercises

Exercise 1 Define $\Box A_o = \downarrow \uparrow A_o$. Taking the small liberty of omitting the linear zone when it is empty, we have already seen

1. $\Box A_o \vdash A_o$
2. $A_o \not\vdash \Box A_o$

Prove or refute each of the following:

1. $\Box A_o \vdash \Box \Box A_o$
2. $\Box(A_o \setminus B_o) (\Box A_o) \vdash \Box B_o$

Exercise 2 Define $\Diamond A_l = \uparrow \downarrow A_l$. We have already seen

1. $A_l \vdash \Diamond A_l$
2. $\Diamond A_l \not\vdash A_l$

Proof or refute each of the following

1. $\Diamond \Diamond A_l \vdash \Diamond A_l$
2. $\Diamond(A_l \multimap B_l), \Diamond A_l \vdash \Diamond B_l$

Exercise 3 Show all cases of cut reduction for $\downarrow A_l$.

Exercise 4 Recall

$$x_L : A_L ; \cdot \vdash \text{down} :: (y_O : \downarrow A_L)$$

from [Section 8](#). A priori, there are four interesting possibilities for such coercions. For each, show that it cannot exist or write a process that implements it.

References

- [Ben94] 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 (CLS'94)*, Kazimierz, Poland, September 1994. Springer LNCS 933. An extended version appears as Technical Report UCAM-CL-TR-352, University of Cambridge.
- [DP01] Rowan Davies and Frank Pfenning. A modal analysis of staged computation. *Journal of the ACM*, 48(3):555–604, May 2001.
- [Mog89] Eugenio Moggi. Computational lambda calculus and monads. In *Proceedings of the Fourth Symposium on Logic in Computer Science*, pages 14–23, Asilomar, California, June 1989. IEEE Computer Society Press.
- [PD01] 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.
- [PG15] Frank Pfenning and Dennis Griffith. Polarized substructural session types. In A. Pitts, editor, *Proceedings of the 18th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS 2015)*, pages 3–22, London, England, April 2015. Springer LNCS 9034. Invited talk.
- [Ree09] Jason Reed. A judgmental deconstruction of modal logic. Unpublished manuscript, 2009.