Lecture Notes on Ordered Logic

15-317: Constructive Logic
Frank Pfenning

Lecture 21
November 16, 2017

1 Introduction

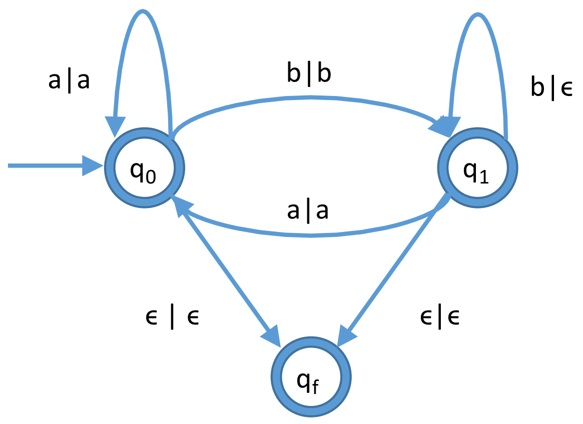
In this lecture we first review ordered inference with two new examples: a finite state transducer (which indicates how to represent finite state transducers in general), and Turing machines.

After that, we follow the playbook of Part II of this course and develop a sequent calculus so that all inference will proceed in one direction, namely bottom-up. We briefly consider some cases in the proof of cut admissibility for the system and carry out some sample proofs. For more on ordered logic, we recommend a course on Substructural Logics from Fall 2016 at Carnegie Mellon University.

Finally, we consider *linear logic* [Gir87] as an intermediate point between ordered and intuitionistic logic via a simple change to ordered logic.

2 Example: Finite State Transducers

A *subsequential finite-state transducer* [Sch77] (FST) consists of a finite number of states, an input alphabet and an output alphabet, and a transition function δ that takes a state and an input symbol to a new state and an output string. We also distinguish an initial state and a final state, from which no further transitions are possible. Finite state transducers have a number of important closure properties and are closely related to deterministic finite automata (DFAs). They are often depicted with transition diagrams. As an example we show a FST which transforms an input string consisting of a and b symbols by compressing all runs of b into a single b. Each transition is labeled as $x \mid w$ where x is either an input symbol or ϵ (when the input string is empty) and w is a word over the output alphabet.



In order to represent this in the Lambek calculus so that ordered inference corresponds to computation, we introduce propositions a and b to represents the symbols, here shared between the input and output alphabets. We also have a proposition $\$$ representing an endmarker and *reverse* the word[1]. For example, the string $bbaba$ will be represented as the ordered antecedents $\$\ a\ b\ a\ b\ b$. Furthermore, we have a new proposition for every state in the FST, here q_0, q_1, and q_f. Initially, our antecedents will be populated by the representation of the input string followed by the initial state. In this example, we start with

$$\$\ a\ b\ a\ b\ b\ q_0$$

We now present inference rules so that each ordered inference corresponds to a transition of the finite state transducer. In the premise we have the input (represented as a proposition) followed by the state; in the conclusion we have the new state followed by the output. The empty input string is represented by $\$$, which we need to write when we transition into the final

[1]for reasons that may nor may not become clear in a future lecture

state.

$$\frac{a\ q_0}{q_0\ a} \qquad \frac{b\ q_0}{q_1\ b} \qquad \frac{\$\ q_0}{q_f\ \$}$$

$$\frac{a\ q_1}{q_0\ a} \qquad \frac{b\ q_1}{q_1} \qquad \frac{\$\ q_1}{q_f\ \$}$$

Since it is convenient, we add one more inference rule

$$\frac{q_f}{\cdot}$$

so that the overall computation with input word w, and initial state q_0 to output v in final state q_f is modeled by inference

$$\begin{array}{c} \$\ w^R\ q_0 \\ \vdots \\ \dfrac{q_f\ \$\ v^R}{\$\ v^R} \end{array}$$

where s^R represents the reversal of a string s. We could also fold the last step into the rules producing q_f, replacing q_f by the empty context.

You can see why we used an endmarker \$: unlike the usual assumption for finite-state transducers, ordered inference cannot depend on whether it takes place at the end of the context. This is because any ordered inference, by its very definition, applies to any consecutive part of the state. In the sequent calculus this is explicit in all the left rules that have arbitrary Ω_L and Ω_R surrounding the principal proposition of the inference. Trying to restrict this would lead to a breakdown in the sequent calculus.

We can use this construction to represent any subsequential finite-state transducer, with one inference rule for every transition. We will not develop the formal details, which are somewhat tedious but straightforward.

We can compose transducers the way we could compose functions. If transducer T_1 transforms input w_0 into w_1 and T_2 transforms w_1 to w_2, then T_1 ; T_2 transforms w_0 to w_2. There is a construction on the automata-theoretic descriptions of transducers to show that T_1 ; T_2 is indeed another finite-state subsequential transducer if T_1 and T_2 are.

Here, in the setting of ordered inference, we can easily represent the composition of transducers T_1 ; ... ; T_n just by renaming the sets of states apart and then creating the initial state as

$$\$\ w^R\ q_0^1 \ldots q_0^n$$

where q_0^i is the initial state of FST T_i. As T_1 starts to produce output, the configuration will have the form

$$\$\ w_0^R\ q_k^1\ w_1^R\ q_0^2 \ldots q_0^n$$

At this point, T_2 (represented by q_0^2) can start to consume some of its input and produce its output, and so on. Effectively, we have a chain of transducers operating concurrently as long as enough input is available to each of them. Eventually, all of them will end up in their final state and we will end up with the final configuration $\$\ v^R$.

3 Example: Turing Machines

In this section we generalize the construction from the previous section to represent Turing machines. We represent the contents of the unbounded tape of the Turing machine as a *finite* context

$$\$\ \ldots\ a_{-1}\ q\ a_0\ a_1\ \ldots \$$$

with two endmarkers \$. The proposition q represents the current state of the machine, and we imagine it "looks to its right" so that the contents of the current cell would be a_0. The initial context for the initial state q_0 is just

$$\$\ q_0\ a_0\ \ldots\ a_n\ \$$$

where $a_0 \ldots a_n$ is the input word written on the tape. Returning to the general case

$$\$\ \ldots\ a_{-1}\ q\ a_0\ a_1\ \ldots \$$$

if the transition function for state q and symbol a_0 specifies to write symbol a_0', transition to state q', and move to the *right*, then the next configuration would be

$$\$\ \ldots\ a_{-1}\ a_0'\ q'\ a_1\ \ldots \$$$

This can easily be represented, in general, by the rule

$$\frac{q\ a}{a'\ q'}\ \mathsf{MR}$$

which we call MR for *move right*.

To see how to represent moving to the left, reconsider

$$\$\ \ldots\ a_{-1}\ q\ a_0\ a_1\ \ldots \$$$

If we are supposed to write a_0', transition to q', and move to the left, the next state should be

$$\$\ \ldots\ q'\ a_{-1}\ a_0'\ a_1\ \ldots \$$$

The corresponding rule would, using b for a_{-1}:

$$\frac{b\ q\ a}{q'\ b\ a'}\ \mathsf{ML}_b$$

We would have such a rule for each b in the (fortunately finite) tape alphabet (which excludes the endmarker), or we could represent it schematically

$$\frac{x\ q\ a}{q'\ x\ a'}\ \mathsf{ML}^*$$

except we would have a side condition that $x \neq \$$. We should also have rules that allow us to extend the tape by the designated blank symbol '␣' (which is part of the usual definition of Turing machines).

$$\frac{\$\ q\ a}{\$\ q'\ ␣\ a'}\ \mathsf{ML}_\$ \qquad \frac{q\ \$}{q\ ␣\ \$}\ \mathsf{ER}_q$$

Finally, if we are in a final state q_f from which no further transitions are possible, we can simply eliminate it from the configuration.

$$\frac{q_f}{\cdot}\ \mathsf{F}$$

A somewhat more symmetric and elegant solution allows the tape head in state q (represented by the proposition q) to be looking either right or left, represented by $q\ \triangleright$ and $\triangleleft\ q$. When we look right and have to move left or vice versa, we just change the direction in which we are looking to implement the move. Then we get the following elegant set of rules, two for each possible transition, two extra ones for extending the tape, and two (if we like) for erasing the final state.

$$\frac{q\ \triangleright\ a}{a'\ q'\ \triangleright}\ \mathsf{LRMR} \qquad \frac{q\ \triangleright\ a}{\triangleleft\ q'\ a'}\ \mathsf{LRML}$$

$$\frac{a\ \triangleleft\ q}{a'\ q'\ \triangleright}\ \mathsf{LLMR} \qquad \frac{a\ \triangleleft\ q}{\triangleleft\ q'\ a'}\ \mathsf{LLML}$$

$$\frac{\triangleright\ \$}{\triangleright\ ␣\ \$}\ \mathsf{ER} \qquad \frac{\$\ \triangleleft}{\$\ ␣\ \triangleleft}\ \mathsf{EL} \qquad \frac{\triangleleft\ q_f}{\cdot}\ \mathsf{FL} \qquad \frac{q_f\ \triangleright}{\cdot}\ \mathsf{FR}$$

The initial configuration represented by the context

$$\$\ q_0\ \triangleright\ a_1\ \ldots\ a_n\ \$$$

and the final configuration as

$$\$\ b_1\ \ldots\ b_k\ \$$$

and we go from the first to the last by a process of ordered inference.

Of course, a Turing machine may not halt, in which case inference would proceed indefinitely, never arriving at a quiescent state in which no inference is possible.

Our modeling of the Turing machine is here faithful in the sense that each step of the Turing machine corresponds to one inference. There is a small caveat in that we have to extent the tape with an explicit inference, while Turing machines are usually preloaded with a two-way infinite tape with blank symbols on them. But except for those little stutter-steps, the correspondence is exact.

Composition of Turing machines in this representation is unfortunately not as simple as for FSTs since the output is not produced piecemeal, going in one direction, but will be on the tape when the final state is reached. We would have to return the tape head (presumably in the final state) to the left end of the tape and then transition to the starting state of the second machine.

Both for finite-state transducers and Turing machines, nondeterminism is easy to add: we just add multiple rules if there are multiple possible transitions from a state. This works, because the inference process is naturally nondeterministic: any applicable rule can be applied.

We will return to automata and Turing machines in a future lecture when we will look at the problem again from a different perspective.

4 Ordered Hypothetical Judgments

The notion of grammatical inference represents parsing as the process of constructing a proof. For example, if we have a phrase (= sequence of words) $w_1 \ldots w_n$ we find their syntactical types $x_1 \ldots x_n$ (guessing if necessary if they are ambiguous) and then set up the problem

$$\begin{array}{c}(w_1 : x_1) \cdots (w_n : x_n)\\ \vdots\\ ? : s\end{array}$$

where "?" will represent the parse tree as a properly parenthesized expression (assuming, of course, we can find a proof).

So far, we can only represent the inference itself, but not the *goal* of parsing a whole sentence. In order to express that we introduce *hypothetical judgments* as a new primitive concept. The situation above is represented as

$$(w_1 : x_1) \cdots (w_n : x_n) \Longrightarrow ? : s$$

or, more generally, as

$$(p_1 : x_1) \cdots (p_n : x_n) \Longrightarrow r : z$$

The turnstile symboxl "$\Longrightarrow$" here separates the *succedent* $r : z$ from the *antecedents* $p_i : x_i$. We sometimes call the left-hand side the *context* or the *hypotheses* and the right-hand side the *conclusion*. Calling the succedent a conclusion is accurate in the sense that it is the conclusion of a hypothetical deduction, but it can also be confusing since we also used "conclusions" to describe what is below the line in a rule of inference. We hope it will always be clear from the situation which of these we mean.

Since we are studying ordered inference right now, the antecedents that form the context are intrinsically ordered. When we want to refer to a sequence of such antecedents we write Ω where "Omega" is intended to suggest "Order". When we capture other forms of inference like linear inference we will revisit this assumption.

5 Inference with Sequents: Looking Left

Now that we have identified hypothetical judgments, written as sequents $\Omega \Longrightarrow r : z$, we should examine what this means for our logical rules of inference. Fortunately, we have had only two connectives, *over* and *under*, first shown here without the proof terms (that is, without the parse trees):

$$\frac{B \,/\, A \quad A}{B}\ /E \qquad\qquad \frac{A \quad A \setminus B}{B}\ \setminus E$$

Now that the propositions we know appear as antecedents, the direction of the rules appears to be reversed when considered on sequents. Let us remember, by analogy, the elimination rule of natural deduction and the left rules for the sequent calculus for ordinary implication (without consideration of order)

$$\frac{A \supset B \quad A}{B} \supset\! E \qquad\qquad \frac{\Gamma, A \supset B \Longrightarrow A \quad \Gamma, A \supset B, B \Longrightarrow C}{\Gamma, A \supset B \Longrightarrow C} \supset\! L$$

In the ordered case, say $A \setminus B$ the antecedents will have form

$$\Omega_1 \; (A \setminus B) \; \Omega_2$$

because we are allowed to apply the $\setminus L$ rule to any antecedent. In the intuitionistic case, we just says $\Gamma, A \supset B$ because we could silently reorder the antecedents.

The second consideration is that the proof of A must be from a block of antecedents that are immediately to the left of $A \setminus B$. This is because the premises of the $\setminus E$ must be consecutive among the current hypothesis.

With these considerations, we arrive at the following rule

$$\frac{\Omega \Longrightarrow A \quad \Omega_L \; B \; \Omega_R \Longrightarrow C}{\Omega_L \; \Omega \; (A \setminus B) \; \Omega_R \Longrightarrow C} \setminus L$$

We have written Ω_L and Ω_R to indicate the rest of the context, which remains unaffected by the inference. The left rule for "over" ($B \,/\, A$) is symmetric:

$$\frac{\Omega_L \; B \; \Omega_R \Longrightarrow C \quad \Omega \Longrightarrow A}{\Omega_L \; (B \,/\, A) \; \Omega \; \Omega_R \Longrightarrow C} \;/L$$

Our inferences, now taking place on the antecedent, take us upward in the tree. This means we need at lease one more rule to complete the proof and signal the success of a hypothetical proof. Both forms with and without the proof terms should be self-explanatory. We use id (for *identity*) to label this inference.

$$\frac{}{A \Longrightarrow A} \; \mathsf{id}_A$$

Unlike the usual intuitionistic sequent calculus, A must be the only antecedent. This is akin to saying that we do not allow *weakening*, neither implicitly nor explicitly.

6 Inference with Sequents: Looking Right

Thinking about the parsing problem $A \setminus (B \,/\, C)$ should be somehow equivalent to $(A \setminus B) \,/\, C$ since both yield a B when given an A to the left and C to the right. Setting this equivalence up as two hypothetical judgments

$$A \setminus (B \,/\, C) \Longrightarrow (A \setminus B) \,/\, C$$

and

$$(A \setminus B) \,/\, C \Longrightarrow A \setminus (B \,/\, C)$$

that we are trying to prove however fails. No inference is possible. We are lacking the ability to express when we can deduce a *succedent* with a logical connective. Lambek [Lam58] states that we should be able to deduce

$$\frac{C}{B \,/\, A} \qquad \text{if} \qquad \frac{C \quad A}{B}$$

So $B \,/\, A$ should follow from C if we get B if we put A to the right of C. With pure inference, as practiced in the last lecture, we had no way to turn this "*if*" into form of inference rule. However, armed with hypothetical judgments it is not difficult to express precisely this:

$$\frac{C\ A \Longrightarrow B}{C \Longrightarrow B \,/\, A}$$

Instead of a single proposition z we allow a context, so we write this

$$\frac{\Omega\ A \Longrightarrow B}{\Omega \Longrightarrow B \,/\, A}\ /R$$

This is an example of a *right rule*, because it analyzes the structure of a proposition in the succedent and we pronounce it as *over right*. The $\setminus R$ (*under right*) rule can be derived analogously.

$$\frac{A\ \Omega \Longrightarrow B}{\Omega \Longrightarrow A \setminus B}\ \setminus R$$

In the next section we will look at the question how we know that these rules are correct. For example, we might have accidentally swapped these two rules, in which case our logic would somehow be flawed.

Let's come back to the motivating example and try to construct a proof of

$$A \setminus (B \,/\, C) \Longrightarrow (A \setminus B) \,/\, C$$

Remember, all the rules work bottom-up, either on some antecedent (a left rule) or on the succedent (a right rule). No left rule applies here (there is no x to the left of $x \setminus (\ldots)$) but fortunately the $/R$ rule does.

$$\frac{(A \setminus (B \,/\, C))\ C \Longrightarrow A \setminus B}{A \setminus (B \,/\, C) \Longrightarrow (A \setminus B) \,/\, C}\ /R$$

Again, no left rule applies (the parentheses are in the wrong place) but a right rule does.

$$\dfrac{\dfrac{A \quad A \backslash (B \,/\, C) \quad C \Longrightarrow B}{A \backslash (B \,/\, C) \quad C \Longrightarrow A \backslash B}\,\backslash R}{A \backslash (B \,/\, C) \Longrightarrow (A \backslash B) \,/\, C}\,/R$$

Finally, now a left rule applies.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A \Longrightarrow A}\,\mathsf{id} \qquad (B \,/\, C) \quad C \Longrightarrow B}{A \quad (A \backslash (B \,/\, C)) \quad C \Longrightarrow B}\,\backslash L}{(A \backslash (B \,/\, C)) \quad C \Longrightarrow A \backslash B}\,\backslash R}{A \backslash (B \,/\, C) \Longrightarrow (A \backslash B) \,/\, C}\,/R$$

One more left rule, and then we can apply identity.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A \Longrightarrow A}\,\mathsf{id} \qquad \dfrac{\dfrac{}{B \Longrightarrow B}\,\mathsf{id}_B \quad \dfrac{}{C \Longrightarrow C}\,\mathsf{id}_C}{(B \,/\, C) \quad C \Longrightarrow B}\,/L}{A \quad (A \backslash (B \,/\, C)) \quad C \Longrightarrow B}\,\backslash L}{(A \backslash (B \,/\, C)) \quad C \Longrightarrow A \backslash B}\,\backslash R}{A \backslash (B \,/\, C) \Longrightarrow (A \backslash B) \,/\, C}\,/R$$

The proof in the other direction is similar and left as exercise.

7 Alternative Conjunction

As already mentioned in the last lecture, some words have more than one syntactic type. For example, *and* has type $s \backslash s \,/\, s$ (omitting parentheses now since the two forms are equivalent by the reasoning the previous section) and also type $n \backslash n^* \,/\, n$, constructing a plural noun from two singular ones. We can combine this into a single type $x \,\&\, y$, pronounced x *with* y:

$$\textit{and} : (s \backslash s \,/\, s) \,\&\, (n \backslash n^* \,/\, n)$$

Then, in a deduction, we are confronted with a choice between the two for every occurrence of *and*. For example, in typing *Alice and Bob work and Eve* *likes Alice*, we choose $n \setminus n^* \,/\, n$ for the first occurence of *and*, and $s \setminus s \,/\, s$ for the second.

Lambek did not explicitly define this connective, but it would be defined by the rules

$$\frac{A \mathbin{\&} B}{A}\ \&E_1 \qquad \frac{A \mathbin{\&} B}{B}\ \&E_2$$

As before, these rules turn into left rules in the sequent calculus, shown here only without the proof terms.

$$\frac{\Omega_L\ A\ \Omega_R \Longrightarrow C}{\Omega_L\ (A \mathbin{\&} B)\ \Omega_R \Longrightarrow C}\ \&L_1 \qquad \frac{\Omega_L\ B\ \Omega_R \Longrightarrow C}{\Omega_L\ (A \mathbin{\&} B)\ \Omega_R \Longrightarrow C}\ \&L_2$$

To derive the right rule we must ask ourselves under which circumstances we could use a proposition both as an x and as a y. That's true, if we can show both, from the some antecedents.

$$\frac{\Omega \Longrightarrow A \quad \Omega \Longrightarrow B}{\Omega \Longrightarrow A \mathbin{\&} B}\ \&R$$

8 Concatenation

In a sequent, there are multiple antecedents (in order!) but only one succedent. If we need to consider multiple propositions in the succedent we need to define a new connective that expresses adjacency as a new proposition. We write $x \bullet y$ (read: x *fuse* y). In the Lambek calculus, we would simply write

$$\frac{A \bullet B}{A\ B}\ \bullet E$$

As a left rule, this is simple turned upside down and becomes

$$\frac{\Omega_L\ x\ y\ \Omega_R \Longrightarrow z}{\Omega_L\ x \bullet y\ \Omega_R \Longrightarrow z}\ \bullet L$$

As a right rule for $x \bullet y$, we have to divide the context into to segments, one proving x and the other proving y.

$$\frac{\Omega_1 \Longrightarrow x \quad \Omega_2 \Longrightarrow y}{\Omega_1\ \Omega_2 \Longrightarrow x \bullet y}\ \bullet R$$

Note that there is some nondeterminism in this rule if we decide to use it to prove a sequent, because we have to decide *where* to split the context

$\Omega = (\Omega_1\ \Omega_2)$. For a context with n propositions there are $n+1$ possibilities. For example, if we want to express that a phrase represented by Ω is parsed into *two sentences* we can prove the hypothetical judgment

$$\Omega \Longrightarrow s \bullet s$$

We can then prove

$$\begin{array}{ccccccc} \textit{Alice} & \textit{works} & \textit{Bob} & \textit{sleeps} & & ? \\ : & : & : & : & & : \\ n & n \setminus s & n & n \setminus s & \Longrightarrow & s \bullet s \end{array}$$

but we have to split the phrase exactly between *works* and *Bob* so that both premises can be proved. Assuming a notation of $p \cdot q : x \bullet y$ if $p : x$ and $q : y$, the proof term for $s \bullet s$ in this example would be $(\textit{Alice works}) \cdot (\textit{Bob sleeps})$.

9 Emptiness[2]

In this section we consider **1**, the unit of concatenation, which corresponds to the empty context. The left and right rules are nullary versions of the binary concatenation. In particular, there must be no antecedents in the right rule for **1**.

$$\frac{}{\Longrightarrow \mathbf{1}}\ \mathbf{1}R \qquad \frac{\Omega_L\ \Omega_R \Longrightarrow z}{\Omega_L\ \mathbf{1}\ \Omega_R \Longrightarrow z}\ \mathbf{1}L$$

10 Admissiblity of Cut

We return from the examples to metatheoretic considerations. Our goal in this section and the next is to show that the cut rule is admissible in the ordered sequent calculus. Together with identity elimination this gives us a global version of harmony for our logic and a good argument for thinking of the right and left rules in the sequent calculus as defining the meaning of the connectives.

A key step on the way will be the *admissibility of cut* in the cut-free sequent calculus. We say that a rule of inference is *admissible* if there is a proof of the conclusion whenever there are proofs of all the premises. This is a somewhat weaker requirement that saying that a rule is *derivable*, which means we have a closed-form hypothetical proof of the conclusion given all

[2]not covered in lecture

the premises. Derivable rules remain derivable even if we extend our logic by new propositions and inference rules (once a proof, always a proof), but admissible rules may no longer remain admissible and have to be reconsidered.

Since the cut-free sequent calculus will play an important role in this course, we write $\Omega \Longrightarrow x$ for a sequent in the cut-free sequent calculus. We write admissible rules using dashed lines and parenthesized justifications, as in

$$\frac{\Omega \Longrightarrow A \quad \Omega_L\ A\ \Omega_r \Longrightarrow C}{\Omega_L\ \Omega\ \Omega_R \Longrightarrow C}\ (\mathsf{cut}_A)$$

Of course, we have not yet proved that cut is indeed admissible here! It turns out that the proof remains essentially the same as we have seen for intuitionistic logic.

Theorem 1 (Admissibility of Cut)
If $\Omega \Longrightarrow A$ and $\Omega_L\ A\ \Omega_R \Longrightarrow C$ then $\Omega_L\ \Omega\ \Omega_R \Longrightarrow C$.

Proof: We assume we are given $\begin{array}{c}\mathcal{D}\\ \Omega \Longrightarrow A\end{array}$ and $\begin{array}{c}\mathcal{E}\\ \Omega_L\ A\ \Omega_R \Longrightarrow C\end{array}$ and we construct $\begin{array}{c}\mathcal{F}\\ \Omega_L\ \Omega\ \Omega_R \Longrightarrow C\end{array}$.

The proof proceeds by a so-called *nested induction*, first on A and then the proofs $\mathcal{D}$ and $\mathcal{E}$. This means we can appeal to the induction hypothesis when

1. either the cut formula A becomes smaller,
2. or A remains the same, and
   (a) $\mathcal{D}$ becomes smaller and $\mathcal{E}$ stays the same,
   (b) or $\mathcal{D}$ stays the same and $\mathcal{E}$ becomes smaller.

This is also called *lexicographic induction* since it is an induction over a lexicographic order, first considering x and then $\mathcal{D}$ and $\mathcal{E}$.

The idea for this kind of induction can be synthesized from the proof if we observe what constructions take place in each case. We will see that the ideas of the cut reductions in the last lecture will be embodied in the proof cases. We distinguish three kinds of cases based on $\mathcal{D}$ and $\mathcal{E}$.

Identity cases. When one premise or the other is an instance of the identity rule we can eliminate the cut outright. This should be expected since identity ("*if we can use x we may prove x*") and cut ("*if we can prove x we may use x*") are direct inverses of each other.

Principal cases. When the cut formula x is introduced by the last inference in both premises we can reduce the cut to (potentially several) cuts on strict subformulas of A. We have demonstrated this by cut reductions in the last lecture.

Commutative cases. When the cut formula is a side formula of the last inference in either premise, we can appeal to the induction hypothesis on this premise and then re-apply the last inference. These constitute valid appeals to the induction hypothesis because the cut formula and one of the deductions in the premises remain the same while the other becomes smaller.

We now go through representative samples of these cases. First, the two identity cases.[3]

Case: $\mathsf{id} \mathbin{\#} \mathcal{E}$

$$\mathcal{D} = \frac{}{A \Longrightarrow A}\ \mathsf{id}_A \quad \text{and} \quad \begin{array}{c}\mathcal{E}\\ \Omega_L\ A\ \Omega_R \Longrightarrow C\end{array} \quad \text{arbitrary}$$

We have to construct a proof of $\Omega_L\ \Omega\ \Omega_R \Longrightarrow C$, but $\Omega = A$, so we can let $\mathcal{F} = \mathcal{E}$.

Case: $\mathcal{D} \mathbin{\#} \mathsf{id}$

$$\begin{array}{c}\mathcal{D}\\ \Omega \Longrightarrow A\end{array} \quad \text{arbitrary, and} \quad \mathcal{E} = \frac{}{A \Longrightarrow A}\ \mathsf{id}_A$$

We have to construct a proof of $\Omega_L\ \Omega\ \Omega_R \Longrightarrow C$, but $\Omega_L = \Omega_R = (\cdot)$ and $C = A$, so we can let $\mathcal{F} = \mathcal{D}$.

Next we look at a principal case, where the cut proposition x (here $x_1\ /\ x_2$) was introduced in the last inference in both premises, in which case we say x is the *principal formula* of the inference.

Case: $/R \mathbin{\#} /L$

$$\mathcal{D} = \frac{\begin{array}{c}\mathcal{D}_1\\ \Omega\ A_2 \Longrightarrow A_1\end{array}}{\Omega \Longrightarrow A_1\ /\ A_2}\ /R \quad \text{and} \quad \mathcal{E} = \frac{\begin{array}{c}\mathcal{E}_2\\ \Omega'_R \Longrightarrow A_2\end{array} \quad \begin{array}{c}\mathcal{E}_1\\ \Omega_L\ A_1\ \Omega''_R \Longrightarrow z\end{array}}{\Omega_L\ (A_1\ /\ A_2)\ \Omega'_R\ \Omega''_R \Longrightarrow z}\ /L$$

[3] in lecture, we only showed a couple of principal cases

Using the intuition gained from cut reduction, we can apply the induction hypothesis on A_2, $\mathcal{E}_2$, and $\mathcal{D}_1$ and we obtain

$$\begin{array}{c}\mathcal{D}_1' \\ \Omega\ \Omega_R' \Longrightarrow A_1\end{array} \quad \text{by i.h. on } A_2, \mathcal{E}_2, \mathcal{D}_1$$

We can once again apply the induction hypothesis, this time on A_1, $\mathcal{D}_1'$, and $\mathcal{E}_1$:

$$\begin{array}{c}\mathcal{E}_1' \\ \Omega_L\ \Omega\ \Omega_R'\ \Omega_R'' \Longrightarrow z\end{array} \quad \text{by i.h. on } A_1, \mathcal{D}_1', \mathcal{E}_1$$

Note that $\mathcal{D}_1'$ is the result of the previous appeal to the induction hypothesis and therefore not known to be smaller than $\mathcal{D}_1$, but the appeal to the induction hypothesis is justified since A_1 is a subformula of $A_1 \mathbin{/} A_2$.

Now we can let $\mathcal{F} = \mathcal{E}_1'$ since $\Omega_R = \Omega_R'\ \Omega_R''$ in this case, so we already have the right endsequent.

A more concise way to write down the same argument is in the form of a tree, where rules that are admissible (by induction hypothesis!) are justified in this manner.

Given

$$\dfrac{\dfrac{\begin{array}{c}\mathcal{D}_1 \\ \Omega\ A_2 \Longrightarrow A_1\end{array}}{\Omega \Longrightarrow A_1 \mathbin{/} A_2}\ /R \qquad \dfrac{\begin{array}{c}\mathcal{E}_2 \\ \Omega_R' \Longrightarrow A_2\end{array} \quad \begin{array}{c}\mathcal{E}_1 \\ \Omega_L\ A_1\ \Omega_R'' \Longrightarrow z\end{array}}{\Omega_L\ A_1 \mathbin{/} A_2\ \Omega_R'\ \Omega_R'' \Longrightarrow z}\ /L}{\Omega_L\ \Omega\ \Omega_R'\ \Omega_R'' \Longrightarrow z}\ (\mathsf{cut?})$$

construct

$$\dfrac{\dfrac{\begin{array}{c}\mathcal{E}_2 \\ \Omega_R' \Longrightarrow A_2\end{array} \quad \begin{array}{c}\mathcal{D}_1 \\ \Omega\ A_2 \Longrightarrow A_1\end{array}}{\Omega\ \Omega_R' \Longrightarrow A_1}\ (\text{i.h. on } A_2) \qquad \begin{array}{c}\mathcal{E}_1 \\ \Omega_L\ A_1\ \Omega_R'' \Longrightarrow z\end{array}}{\Omega_L\ \Omega\ \Omega_R'\ \Omega_R'' \Longrightarrow z}\ (\text{i.h. on } A_1)$$

This is of course the local reduction, revisited as part of an inductive proof.

Finally we look at a *commutative case*, where the last inference rule applied in the first or second premise of the cut must have been different from the cut formula. We call this a *side formula*. We organize the cases around which rule was applied to which premise. Fortunately, they all go the same way: we "push" up the cut past the inference that was applied to the side formula. We show only one example.

Case: $\mathcal{D} \ \# \ \bullet R$

$$\overset{\mathcal{D}}{\Omega \Longrightarrow A} \quad \text{arbitrary} \quad \text{and} \quad \mathcal{E} = \dfrac{\overset{\mathcal{E}_1}{\Omega'_L \Longrightarrow C_1} \quad \overset{\mathcal{E}_2}{\Omega''_L \; A \; \Omega_R \Longrightarrow C}}{\Omega'_L \; \Omega''_L \; A \; \Omega_R \Longrightarrow C_1 \bullet C_2} \; \bullet R$$

In this case we have the situation

$$\dfrac{\overset{\mathcal{D}}{\Omega \Longrightarrow A} \quad \dfrac{\overset{\mathcal{E}_1}{\Omega'_L \Longrightarrow C_1} \quad \overset{\mathcal{E}_2}{\Omega''_L \; A \; \Omega_R \Longrightarrow C}}{\Omega'_L \; \Omega''_L \; A \; \Omega_R \Longrightarrow C_1 \bullet C_2} \; \bullet R}{\Omega'_L \; \Omega''_L \; \Omega \; \Omega_R \Longrightarrow C_1 \bullet C_2} \; (\mathsf{cut?})$$

and construct

$$\dfrac{\overset{\mathcal{E}_1}{\Omega'_L \Longrightarrow C_1} \quad \dfrac{\overset{\mathcal{D}}{\Omega \Longrightarrow A} \quad \overset{\mathcal{E}_2}{\Omega''_L \; A \; \Omega_R \Longrightarrow C}}{\Omega''_L \; \Omega \; \Omega_R \Longrightarrow C_2} \; \text{i.h. on } A, \mathcal{D}, \mathcal{E}_2}{\Omega'_L \; \Omega''_L \; \Omega \; \Omega_R \Longrightarrow C_1 \bullet C_2} \; \bullet R$$

Effectively, we have commuted the cut upward, past the $\bullet R$ inference.

□

Our proof was *constructive*: it presents an effective method for constructing a cut-free proof of the conclusion, given cut-free proofs of the premises. The algorithm that can be extracted from the proof is nondeterministic, since some of the commuting cases overlap when the principal formula is a side formula in both premises. For most logics (although usually classical logic) the result is unique up to further permuting conversions between inference rules, a characterization we will have occasion to discuss later.

11 Linear Logic

In ordered logic, we use *ordered antecedents* which we have previously already used to make the inversion phase of proof search deterministic. We can define

$$\Omega ::= \epsilon \mid A \mid \Omega_1 \cdot \Omega_2$$

where '$\cdot$' is an associative operator with unit ϵ. In the lecture so far, we have omitted $\cdot$ and just used juxtaposition, and replace ϵ by the empty context.

Now we obtain *linear logic* by using antecedents Δ of the form

$$\Delta ::= \epsilon \mid A \mid \Delta_1, \Delta_2$$

where ',' is an associate and *commutative* operator with unit ϵ. Otherwise, the rules stay exactly the same as those for ordered logic!

Now we notice some interesting phenomena. For example, in ordered logic we have two implications, $A \setminus B$ and $B \mathbin{/} A$. But if order does not matter, we cannot tell which side of the antecedents A will end up at so the two become logically equivalent. We write $A \multimap B$ (pronounced A *lolli* B).

Conjunction is even more interesting. In ordered logic we have three forms of conjunction $A \bullet B$ (A *fuse* B), and $A \circ B$ (A *twist* B, see Problem 2, Assignment 9) and $A \mathbin{\&} B$ (A *with* B). In linear logic, fuse and twist collapse to $A \otimes B$ (A *tensor* B) because their only difference is the order of the components. On the other hand $A \& B$ remains the same and is available in linear as well as ordered logic. Finally, in intuitionistic logic there is only one conjunction, $A \wedge B$, when one considers provability. But, if one considers the structure of proofs as well, we actually have to: a negative one (corresponding to $A \mathbin{\&} B$) and a positive one ($A \otimes B$).

We will discuss linear logic and it operational interpretation further in the upcoming lectures.

References

[Gir87] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[Lam58] Joachim Lambek. The mathematics of sentence structure. *The American Mathematical Monthly*, 65(3):154–170, 1958.

[Sch77] Marcel Paul Schützenberger. Sur une variante des fonctions sequentielles. *Theoretical Computer Science*, 4(1):47–57, 1977.