Lecture Notes on
Cut Elimination

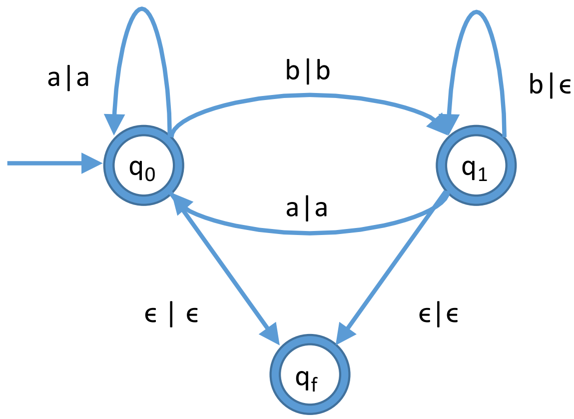
15-816: Substructural Logics
Frank Pfenning

Lecture 4
September 8, 2016

We first present some additional examples illustrating ordered inference that capture computations of finite state transducers and Turing machines. Then we return to cut elimination, also called Gentzen's Hauptsatz [Gen35], for the Lambek calculus given in its sequent formulation. Together with identity elimination, this justifies our program of viewing the inference rules as defining the meaning of the connectives even more clearly than the cut reduction and identity expansion by themselves. The idea behind these will emerge in the proof. Cut elimination also immediately yields a decidability proof for the Lambek calculus, something already observed by Lambek [Lam58].

1 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 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

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

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

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

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

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 (see Exercise 2).

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 $T_1 \; ; \; \ldots \; ; \; T_n$ 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 $\$ \; v^R$.

2 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.[2] 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}:

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

[2] In lecture we were looking to the left, but it is a bit unpleasant to define the initial state in that case.

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).

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

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.

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

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

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

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.

3 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 can be eliminated from any proof in the ordered sequent calculus. Together with identity elimination in Section 5, 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 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 $\Omega \Vdash x$ for a sequent in the cut-free sequent calculus. We write admissible rules using dashed lines and parenthesized justifications, as in

$$\frac{\Omega \Vdash x \quad \Omega_L \; x \; \Omega_r \Vdash z}{\Omega_L \; \Omega \; \Omega_R \Vdash z} \; (\mathsf{cut}_x)$$

Of course, we have not yet proved that cut is indeed admissible here!

Theorem 1 (Admissibility of Cut)
If $\Omega \Vdash x$ and $\Omega_L \; x \; \Omega_R \Vdash z$ then $\Omega_L \; \Omega \; \Omega_R \Vdash z$.

Proof: We assume we are given $\overset{\mathcal{D}}{\Omega \Vdash x}$ and $\overset{\mathcal{E}}{\Omega_L \; x \; \Omega_R \Vdash z}$ and we construct $\overset{\mathcal{F}}{\Omega_L \; \Omega \; \Omega_R \Vdash z}$.

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

1. either the cut formula x becomes smaller,
2. or x 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.

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

$$\mathcal{D} = \frac{}{x \Vdash x}\ \mathsf{id}_x \quad \text{and} \quad \begin{array}{c}\mathcal{E}\\ \Omega_L\ x\ \Omega_R \Vdash z\end{array} \quad \text{arbitrary}$$

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

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

$$\begin{array}{c}\mathcal{D}\\ \Omega \Vdash x\end{array} \quad \text{arbitrary, and} \quad \mathcal{E} = \frac{}{x \Vdash x}\ \mathsf{id}_x$$

We have to construct a proof of $\Omega_L\ \Omega\ \Omega_R \Vdash z$, but $\Omega_L = \Omega_R = (\cdot)$ and $z = x$, 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 proposition* of the inference.

Case: $/R \# /L$

$$\mathcal{D} = \frac{\begin{array}{c}\mathcal{D}_1\\ \Omega\ x_2 \Vdash x_1\end{array}}{\Omega \Vdash x_1 \,/\, x_2}\ /R \quad \text{and} \quad \mathcal{E} = \frac{\begin{array}{c}\mathcal{E}_2\\ \Omega'_R \Vdash x_2\end{array} \quad \begin{array}{c}\mathcal{E}_1\\ \Omega_L\ x_1\ \Omega''_R \Vdash z\end{array}}{\Omega_L\ (x_1 \,/\, x_2)\ \Omega'_R\ \Omega''_R \Vdash z}\ /L$$

Using the intuition gained from cut reduction, we can apply the induction hypothesis on x_2, $\mathcal{E}_2$, and $\mathcal{D}_1$ and we obtain

$$\begin{array}{c}\mathcal{D}'_1\\ \Omega\ \Omega'_R \Vdash x_1\end{array} \quad \text{by i.h. on } x_2, \mathcal{E}_2, \mathcal{D}_1$$

We can once again apply the induction hypothesis, this time on x_1, $\mathcal{D}'_1$, and $\mathcal{E}_1$:

$$\begin{array}{c}\mathcal{E}'_1\\ \Omega_L\ \Omega\ \Omega'_R\ \Omega''_R \Vdash z\end{array} \quad \text{by i.h. on } x_1, \mathcal{D}'_1, \mathcal{E}_1$$

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 x_1 is a subformula of $x_1 \,/\, x_2$.

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

$$\dfrac{\dfrac{\begin{array}{c}\mathcal{D}_1\\ \Omega\ x_2 \Vdash x_1\end{array}}{\Omega \Vdash x_1 \,/\, x_2}\ /R \qquad \dfrac{\begin{array}{c}\mathcal{E}_2\\ \Omega'_R \Vdash x_2\end{array} \quad \begin{array}{c}\mathcal{E}_1\\ \Omega_L\ x_1\ \Omega''_R \Vdash z\end{array}}{\Omega_L\ x_1 \,/\, x_2\ \Omega'_R\ \Omega''_R \Vdash z}\ /L}{\Omega_L\ \Omega\ \Omega'_R\ \Omega''_R \Vdash z}\ (\mathsf{cut}?)$$

construct

$$\dfrac{\dfrac{\begin{array}{c}\mathcal{E}_2\\ \Omega'_R \Vdash x_2\end{array} \quad \begin{array}{c}\mathcal{D}_1\\ \Omega\ x_2 \Vdash x_1\end{array}}{\Omega\ \Omega'_R \Vdash x_1}\ (\text{i.h. on } x_2) \qquad \begin{array}{c}\mathcal{E}_1\\ \Omega_L\ x_1\ \Omega''_R \Vdash z\end{array}}{\Omega_L\ \Omega\ \Omega'_R\ \Omega''_R \Vdash z}\ (\text{i.h. on } x_1)$$

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

$$\begin{array}{c}\mathcal{D}\\ \Omega \Vdash x\end{array} \quad \text{arbitrary} \quad \text{and} \quad \mathcal{E} = \dfrac{\begin{array}{c}\mathcal{E}_1\\ \Omega'_L \Vdash z_1\end{array} \quad \begin{array}{c}\mathcal{E}_2\\ \Omega''_L\ x\ \Omega_R \Vdash z\end{array}}{\Omega'_L\ \Omega''_L\ x\ \Omega_R \Vdash z_1 \bullet z_2}\ \bullet R$$

In this case we have the situation

$$\dfrac{\begin{array}{c}\mathcal{D}\\ \Omega \Vdash x\end{array} \quad \dfrac{\begin{array}{c}\mathcal{E}_1\\ \Omega'_L \Vdash z_1\end{array} \quad \begin{array}{c}\mathcal{E}_2\\ \Omega''_L\ x\ \Omega_R \Vdash z\end{array}}{\Omega'_L\ \Omega''_L\ x\ \Omega_R \Vdash z_1 \bullet z_2}\ \bullet R}{\Omega'_L\ \Omega''_L\ \Omega\ \Omega_R \Vdash z_1 \bullet z_2}\ (\mathsf{cut?})$$

and construct

$$\dfrac{\begin{array}{c}\mathcal{E}_1\\ \Omega'_L \Vdash z_1\end{array} \quad \dfrac{\begin{array}{c}\mathcal{D}\\ \Omega \Vdash x\end{array} \quad \begin{array}{c}\mathcal{E}_2\\ \Omega''_L\ x\ \Omega_R \Vdash z\end{array}}{\Omega''_L\ \Omega\ \Omega_R \Vdash z_2}\ \text{i.h. on } x, \mathcal{D}, \mathcal{E}_2}{\Omega'_L\ \Omega''_L\ \Omega\ \Omega_R \Vdash z_1 \bullet z_2}\ \bullet R$$

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.

4 Cut Elimination from Cut Admissibility

Because of its fundamental importance, there have been many different kinds of proofs of cut elimination for different logics. The first one, which also introduced the sequent calculus, was by Gentzen [Gen35]. We will develop a proof by *structural induction*, by far the most important method of proof in the study of proofs. This technique was developed in [Pfe94] for classical linear logic, adapted to linear logic by Chang et al. [CCP03]. A key insight is to use the admissibility of cut on cut-free proofs as a lemma.

Theorem 2 (Cut Elimination) *If* $\Omega \vdash x$ *then* $\Omega \Vdash x$.

Proof: We proceed by induction on the structure of

$$\begin{array}{c}\mathcal{D}\\ \Omega \vdash x\end{array}$$

Except for cut, all cases are straightforward. We show one such case.

Case: $\backslash L$

$$\mathcal{D} = \frac{\begin{array}{c}\mathcal{D}_1\\ \Omega' \vdash y\end{array} \quad \begin{array}{c}\mathcal{D}_2\\ \Omega_L\ z\ \Omega_R \vdash x\end{array}}{\Omega_L\ \Omega'\ (y \backslash z)\ \Omega_R \vdash x}\ \backslash L$$

Then construct

$$\mathcal{D}' = \frac{\begin{array}{c}\text{i.h.}(\mathcal{D}_1)\\ \Omega' \Vdash y\end{array} \quad \begin{array}{c}\text{i.h.}(\mathcal{D}_2)\\ \Omega_L\ z\ \Omega_R \Vdash x\end{array}}{\Omega_L\ \Omega'\ (y \backslash z)\ \Omega_R \Vdash x}\ \backslash L$$

$\mathcal{D}'$ is cut free since i.h.$(\mathcal{D}_1)$ and i.h.$(\mathcal{D}_2)$ are.

The remaining case is that of cut. Luckily, we can call on admissibility of cut to obtain a cut-free proof of the conclusion!

Case:

$$\mathcal{D} = \frac{\begin{array}{c}\mathcal{D}_1\\ \Omega' \vdash y\end{array} \quad \begin{array}{c}\mathcal{D}_2\\ \Omega_L\ y\ \Omega_2 \vdash x\end{array}}{\Omega_L\ \Omega'\ \Omega_R \vdash x}\ \mathsf{cut}_y$$

Then

$$\mathcal{D}' = \frac{\begin{array}{c}\text{i.h.}(\mathcal{D}_1)\\ \Omega' \Vdash y\end{array} \quad \begin{array}{c}\text{i.h.}(\mathcal{D}_2)\\ \Omega_L\ y\ \Omega_2 \Vdash x\end{array}}{\Omega_L\ \Omega'\ \Omega_R \Vdash x}\ (\mathsf{cut}_y)$$

$\square$

5 Identity Elimination

In this section[3] we show identity elimination. Fortunately, it is much easier than cut elimination. It is also not quite as important, since it's relationship to computation is less direct.

Theorem 3 (Admissibility of Identity) *In the sequent calculus where* id_a *is restricted to variables* a*, there are cut-free proofs of* $x \Vdash x$ *for any proposition* x*.*

Proof: By induction on the structure of x. We show only one case; all other cases are similar.

Case: $x = x_1 \mathbin{/} x_2$ Then

$$\dfrac{\dfrac{\dfrac{}{x_2 \Vdash x_2}\,\text{i.h.}(x_2) \quad \dfrac{}{x_1 \Vdash x_1}\,\text{i.h.}(x_1)}{x_1 \mathbin{/} x_2 \quad x_2 \Vdash x_1}\,/L}{x_1 \mathbin{/} x_2 \Vdash x_1 \mathbin{/} x_2}\,/R$$

□

The keys to this proof are the identity expansions, just like the cut reductions were the keys to the admissibility of cut.

Theorem 4 (Identity Elimination) *Whenever* $\Omega \vdash x$ *in the sequent calculus, then there exists a proof where identity is applied only to variables. If the given proof is cut-free, so will be the resulting one.*

Proof: By induction on the structure of the given proof, appealing to the admissibility of identity in the case of id_x. □

6 Consequences of Cut Elimination

There are many important consequences of cut elimination. One class of theorems are so-called *refutations*, showing that certain conjectures can not be proven. Here are a few.

Corollary 5 (Consistency) *It is not the case that* $\vdash x$ *for a variable* x*.*

[3] not covered in lecture

Proof: Assume $\vdash x$. By cut elimination, there must be a cut-free proof of x. But no rule could have this conclusion for a variable x. $\square$

Without cut elimination the above proof would not work, because the sequent in question might have been inferred by the cut rule.

Exercises

Exercise 1 The binary counter in Lecture 2, Section 2, is almost in the form of a subsequential finite-state transducer. If we think of eps as the end-marker $\$$, the only fly in the ointment is the early termination, before all the input is read.

1. Represent increment of a binary string as an FST that correctly reads all of its input. How many states do you need?
2. Use the construction in Section 1 to present this new version as ordered inference rules.
3. Represent the functions $2 * n$ and $2 * n + 1$ for input n both as FSTs and as ordered inference rules.
4. Any examples, conjectures, or theorems how fast the output of an FST may grow as a function of the input on the binary representation of natural numbers?

Exercise 2 Consider defining a new unary connective $\$\,x$ with the following left rule:

$$\frac{x\ \Omega \vdash z}{(\$\,x)\ \Omega \vdash z}\ \$L$$

which is intended provide us with x, but only if $\$\,x$ is at the left end of the context. Define matching right rule(s) and test identity expansion and cut reduction, or explain why it dos not seem to be possible.

Exercise 3 Proceed as in Exercise 2 for new unary connective $x\,\$$ (written in postfix form) defined by

$$\frac{\Omega\ x \vdash z}{\Omega\ (x\,\$) \vdash z}\ \$L$$

which is intended to provide us with x, but only if $x\,\$$ is at the right end of the context.

Exercise 4 Consider defining a new connective $y \Rightarrow x$ with the right rule

$$\frac{\Omega_L\ y\ \Omega_R \vdash x}{\Omega_L\ \Omega_R \vdash y \Rightarrow x}\ {\Rightarrow}R$$

which is intended to express that y implies x if we can prove x under the assumption y *somewhere* in the antecedent. Define matching left rule(s) and test identity expansion and cut reduction, or explain why it does not seem to be possible.

Exercise 5 Write out the following cases in the proof of cut admissibility.

1. Show the principal case for $\bullet R$ matched against $\bullet L$.
2. Show the principal case for $\& R$ matched against $\& L_2$.
3. Show the principal case for $\mathbf{1}R$ matched against $\mathbf{1}L$.
4. Show all commutative cases for arbitrary $\mathcal{D}$ and $\mathcal{E}$ being $/L$ applied to a side formula.
5. Show all commutative cases for $\mathcal{D}$ being $\backslash L$ and $\mathcal{E}$ being arbitrary.

Exercise 6 Reconsider the alternative rule $\backslash L^*$.

$$\frac{\Omega_L \; x \; \Omega_R \vdash z}{\Omega_L \; y \; (y \backslash x) \; \Omega_R \vdash z} \; \backslash L^*$$

from Lecture 2.

1. Show which cases in the proof of cut admissibility go awry.
2. Prove that cut elimination does *not* hold if $\backslash L$ is replaced by $\backslash L^*$.

Exercise 7 Among the following prove those are true and refute those that are not by taking advantage of cut elimination. We write $x \dashv\vdash y$ for $x \vdash y$ and $y \vdash x$.

1. $x \dashv\vdash x \;\&\; x$
2. $x \dashv\vdash x \bullet x$
3. $\mathbf{1} \dashv\vdash x \backslash x$

References

[CCP03] 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.

[Gen35] 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.

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

[Pfe94] Frank Pfenning. Structural cut elimination in linear logic. Technical Report CMU-CS-94-222, Department of Computer Science, Carnegie Mellon University, December 1994.

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