15-819K: Logic Programming

Lecture 14

Cut Elimination

Frank Pfenning

October 12, 2006

In this lecture we consider how to prove that connectives are asynchronous as goals and then consider cut elimination, one of the most important and fundamental properties of logical systems. We then revisit residuation to restore some of the connectives not present in the asynchronous fragment of linear logic. For each synchronous assumption we find a corresponding synchronous goal connective.

14.1 Proving Connectives Asynchronous

We have claimed in the last lecture that certain connectives of linear logic are asynchronous as goals in order to justify their inclusion in our linear logic programming language. I know of essentially two ways to prove that such operators are indeed asynchronous. The first is by simple inductions, one for each asynchronous connectives. The following theorem provides an example. In todays lecture we generally omit the judgment form for propositions such as true on the right-hand side, and res or ures on the left-hand side, since this can be infered from the placement of the proposition.

Theorem 14.1 *If $\Delta \Vdash A \mathbin{\&} B$ then $\Delta \Vdash A$ and $\Delta \Vdash B$.*

Proof: By induction on the structure of $\mathcal{D}$, the deduction of $\Delta \Vdash A \mathbin{\&} B$. We show only two cases; others are similar.

Case: $\mathcal{D} = \dfrac{\mathcal{D}_1 \quad \mathcal{D}_2}{\Delta \Vdash A \quad \Delta \Vdash B} \mathbin{\&}R.$

$\Delta \Vdash A$	Subderivation $\mathcal{D}_1$
$\Delta \Vdash B$	Subderivation $\mathcal{D}_2$

Case: $\mathcal{D} = \dfrac{\begin{array}{c}\mathcal{D}_1 \\ \Delta', C_1 \Vdash A \mathbin{\&} B\end{array}}{\Delta', C_1 \mathbin{\&} C_2 \Vdash A \mathbin{\&} B}\ \&L_1$ where $\Delta = (\Delta', C_1 \mathbin{\&} C_2)$.

$\Delta', C_1 \Vdash A$ and	
$\Delta', C_1 \Vdash B$	By i.h. on $\mathcal{D}_1$
$\Delta', C_1 \mathbin{\&} C_2 \Vdash A$	By rule $\&L_1$
$\Delta', C_1 \mathbin{\&} C_2 \Vdash B$	By rule $\&L_1$

□

There is a second way to proceed, using the admissibility of cut from the next section directly, without appeal to induction.

14.2 Admissibility of Cut

So far we have not justified that the right and left rules for the connectives actually match up in an expected way. What we would like to show is that the judgment of a being resource and the judgment of truth, when combined in a linear hypothetical judgment, coincide. There are two parts to this. First, we show that with the resource A we can achieve the goal A, for arbitrary A (not just atomic predicates).

Theorem 14.2 (Identity Principle) *A res $\Vdash$ A true for any proposition A.*

Proof: See Exercise 12.1. □

Second, we show that if we can achieve A as a goal, it is legitimate to assume A as a resource. This completes the theorems which show our sequent calculus is properly designed.

Theorem 14.3 (Admissibility of Cut)
If $\Delta_A \Vdash A$ true and Δ_C, A res $\Vdash C$ true then $\Delta_C, \Delta_A \Vdash C$ true.

Proof: We prove this here only for the purely linear fragment, without the operators involving unrestricted resources (!A, $A \supset B$). The proof proceeds by nested induction, first on the structure of A, the so-called *cut formula*, then simultaneously on the structure of $\mathcal{D}$, the derivation of $\Delta_A \Vdash A$ *true* and $\mathcal{E}$, the derivation of Δ_C, A *res* $\Vdash C$ *true*. This form of induction means we can appeal to the induction hypothesis

1. either on a smaller cut formula with arbitrary derivations, or

2. on the same cut formula A and same $\mathcal{D}$, but a subderivation of $\mathcal{E}$, or

3. on the same cut formula A and same $\mathcal{E}$, but a subderivation of $\mathcal{D}$.

There are many cases to distinguish; we show only three which illustrate the reasons behind the form of the induction above.

Case: $\mathcal{D} = \dfrac{\begin{array}{cc}\mathcal{D}_1 & \mathcal{D}_2 \\ \Delta_A \Vdash A_1 & \Delta_A \Vdash A_2\end{array}}{\Delta_A \Vdash A_1 \mathbin{\&} A_2}\ \&R$ and $\mathcal{E} = \dfrac{\begin{array}{c}\mathcal{E}_1 \\ \Delta_C, A_1 \Vdash C\end{array}}{\Delta_C, A_1 \mathbin{\&} A_2 \Vdash C}\ \&L_1$.

$\Delta_C, \Delta_A \Vdash C$ By i.h. on A_1, $\mathcal{D}_1$ and $\mathcal{E}_1$

Case: $\mathcal{D} = \dfrac{\begin{array}{cc}\mathcal{D}_1 & \mathcal{D}_2 \\ \Delta_1 \Vdash A_1 & \Delta_2 \Vdash A_2\end{array}}{\Delta_1, \Delta_2 \Vdash A_1 \otimes A_2}\ \otimes R$ and $\mathcal{E} = \dfrac{\begin{array}{c}\mathcal{E}' \\ \Delta_C, A_1, A_2 \Vdash C\end{array}}{\Delta_C, A_1 \otimes A_2 \Vdash C}\ \otimes L$.

$\Delta_C, A_1, \Delta_2 \Vdash C$ By i.h. on A_2, $\mathcal{D}_2$, and $\mathcal{E}'$
$\Delta_C, \Delta_1, \Delta_2 \Vdash C$ By i.h. on A_1, $\mathcal{D}_1$ and the previous line
$\Delta_C, \Delta_A \Vdash C$ Since $\Delta_A = (\Delta_1, \Delta_2)$

Case: $\begin{array}{c}\mathcal{D} \\ \Delta_A \Vdash A\end{array}$ is arbitrary and $\mathcal{E} = \dfrac{\begin{array}{c}\mathcal{E}_1 \\ \Delta', D_1, A \Vdash C\end{array}}{\Delta', D_1 \mathbin{\&} D_2, A \Vdash C}\ \&L_1$.

$\Delta', D_1, \Delta_A \Vdash C$ By i.h. on A, $\mathcal{D}$ and $\mathcal{E}_1$
$\Delta', D_1 \mathbin{\&} D_2, \Delta_A \Vdash C$ By rule $\&L_1$

□

The shown cases are typical in that if the cut formulas were just introduced on both the right and the left, then we can appeal to the induction hypothesis on its subformulas. Otherwise, we can keep the cut formula and either $\mathcal{D}$ or $\mathcal{E}$ the same and appeal to the induction hypothesis on the subderivations on the other side.

The case for $\otimes$ above shows why we cannot simply use an induction over the derivations $\mathcal{D}$ and $\mathcal{E}$, because the second time we appeal to the induction hypothesis, one of the derivations come from a previous appeal to the induction hypothesis and could be much larger than $\mathcal{E}$.

The proof is not too difficult to extend to the case with unrestricted assumptions (see Exercise 14.2).

14.3 Cut Elimination

Cut elimination is the property that if we take cut as a new inference rule, it can be eliminated from any proof. Actually, here we would have two cut rules.

$$\frac{\Gamma; \Delta_A \Vdash A\ true \quad \Gamma; \Delta_C, A\ res \Vdash C\ true}{\Gamma; \Delta_C, \Delta_A \Vdash C\ true}\ \text{cut}$$

$$\frac{\Gamma; \cdot \Vdash A\ true \quad (\Gamma, A\ ures); \Delta \Vdash C\ true}{\Gamma; \Delta \Vdash C\ true}\ \text{cut!}$$

Showing that cut can be eliminated is an entirely straightforward induction over the structure of the deduction with cuts. In each case we just appeal to the induction hypothesis on each premiss and re-apply the rule to the get the same conclusion. The only exception are the cut rules, in which case we obtain cut-free derivations of the premisses by induction hypothesis and then appeal to the admissibility of cut to get a cut-free derivation of the conclusion.

Cut as a new rule, however, is unfortunate from the perspective of proof search. When read bottom-up, we have to invent a new proposition A, which we then prove. When this proof succeeds we would be allowed to assume it into our overall proof. While mathematically inventing such lemmas A is critical, in a logic programming language it destroys the goal-directed character of search.

14.4 Asynchronous Connectives, Revisited

Using cut elimination we can give alternate proofs that connectives are asynchronous. We show only one example, for conjunction.

Theorem 14.4 *If $\Delta \Vdash A \,\&\, B$ then $\Delta \Vdash A$ and $\Delta \Vdash B$.*

Proof: (Alternate) Direct, using admissibility of cut.

$\Delta \Vdash A \,\&\, B$	Given
$A \Vdash A$	Identity principle
$A \,\&\, B \Vdash A$	By rule $\&L_1$
$\Delta \Vdash A$	By admissibility of cut
$B \Vdash B$	Identify principle
$A \,\&\, B \Vdash B$	By rule $\&L_2$
$\Delta \Vdash B$	By admissibility of cut

□

14.5 Residuation and Synchronous Goal Connectives

In the focusing calculus from the previous lecture, all connectives are asynchronous as goals and synchronous when in focus as assumptions. In our little programming example of peg solitaire, we extensively used simultaneous conjunction ($\otimes$) and and also disjunction ($\oplus$). One question is how to extend our language to include these connectives. So far, we have, for both programs and goals:

$$A \quad ::= \quad P \mid A_1 \multimap A_2 \mid A_1 \mathbin{\&} A_2 \mid \top \mid \forall x.\, A \mid A_1 \supset A_2$$

A principled way to approach this question is to return to residuation. Given a program clause this constructs a goal whose search behavior is equivalent to the behavior of the clause. Since we have already seen residuation in detail for the non-linear case, we just present the rules here.

$$\frac{}{P' \Vdash P > P' \doteq P} \qquad \frac{D_1 \Vdash P > G_1}{G_2 \multimap D_1 \Vdash P > G_1 \otimes G_2}$$

$$\frac{D_1 \Vdash P > G_1 \quad D_2 \Vdash P > G_2}{D_1 \mathbin{\&} D_2 \Vdash P > G_1 \oplus G_2} \qquad \frac{}{\top \Vdash P > \mathbf{0}}$$

$$\frac{D \Vdash P > G \quad x \notin \mathrm{FV}(P)}{\forall x.\, D \Vdash P > \exists x.\, G} \qquad \frac{D_1 \Vdash P > G_1}{G_2 \supset D_1 \Vdash P > G_1 \otimes\!!\, G_2}$$

There are a few connectives we have not seen in their full generality in linear logic, namely equality, existential quantification, and a curious asymmetric connective $G_1 \otimes\!!\, G_2$. We concentrate here on their behavior as goals (see Exercise 14.5). Because these connectives mirror the synchronous behavior of the assumption in focus, proving one of these is now a focusing judgment, except that we focus on a goal. We write this as $\Gamma; \Delta \gg G$.

First, in our proof search judgment we replace the focus and copy rules by appeals to residuation.

$$\frac{D \in \Gamma \quad D \Vdash P > G \quad \Gamma; \Delta \gg G}{\Gamma; \Delta \Vdash P} \text{ resid!}$$

$$\frac{D \Vdash P > G \quad \Gamma; \Delta \gg G}{\Gamma; \Delta, D \Vdash P} \text{ resid}$$

Next the rules for focusing on the right.

$$\frac{\Delta = (\cdot)}{\Delta \gg P \doteq P}\ \text{id} \qquad \frac{\Delta = (\Delta_1, \Delta_2) \quad \Delta_1 \gg G_1 \quad \Delta_2 \gg G_2}{\Delta \gg G_1 \otimes G_2}\ \otimes R$$

$$\frac{\Delta \gg G_1}{\Delta \gg G_1 \oplus G_2}\ \oplus R_1 \qquad \frac{\Delta \gg G_2}{\Delta \gg G_1 \oplus G_2}\ \oplus R_2 \qquad \text{no } \mathbf{0}R \text{ rule for } \Delta \gg \mathbf{0}$$

$$\frac{\Delta \gg G(t/x)}{\Delta \gg \exists x.\, G}\ \exists R \qquad \frac{\Gamma; \Delta \gg G_1 \quad \Gamma; \cdot \gg G_2}{\Gamma; \Delta \gg G_1 \otimes! \, G_2}\ \otimes! R$$

Furthermore, we transition back to asynchronous decomposition when we encounter an asynchronous connective. We call this *blurring* the focus. Conversely, we focus on the right when encountering a synchronous connective.

$$\frac{\Delta \Vdash G \quad G \text{ asynch.}}{\Delta \gg G}\ \text{blur} \qquad \frac{\Delta \gg G \quad G \text{ synch.}}{\Delta \Vdash G}\ \text{rfocus}$$

For completeness, we give the remaining rules for asynchronous goals (the atomic case is above in the resid and resid! rules).

$$\frac{\Delta, D_1 \Vdash G_2}{\Delta \Vdash D_1 \multimap G_2}\ \multimap R \qquad \frac{\Delta \Vdash G_1 \quad \Delta \Vdash G_2}{\Delta \Vdash G_1 \,\&\, G_1}\ \& R \qquad \frac{}{\Delta \Vdash \top}\ \top R$$

$$\frac{\Delta \Vdash G \quad x \notin \mathrm{FV}(\Gamma; \Delta)}{\Delta \Vdash \forall x.\, G}\ \forall R \qquad \frac{(\Gamma, D_1); \Delta \Vdash G_2}{\Gamma; \Delta \Vdash D_1 \supset G_2}\ \supset R$$

This yields the following grammar of so-called *linear hereditary Harrop formulas* which form the basis of the Lolli language. The fragment without $\multimap$ and $\otimes$, replacing $\wedge/\&, \vee/\oplus, \bot/\mathbf{0}, \wedge/\otimes!$, is called *hereditary Harrop formulas* and forms the basis for λProlog.

$$\begin{array}{llcl}
\text{Goals} & G & ::= & \\
\quad \text{Asynch.} & & P & \mid D_1 \multimap G_2 \mid G_1 \,\&\, G_2 \mid \top \mid \forall x.\, G \mid D_1 \supset G_2 \\
\quad \text{Synch.} & \mid & P' \doteq P \mid G_1 \otimes G_2 \mid G_1 \oplus G_2 \mid \mathbf{0} \mid \exists x.\, A \mid G_1 \otimes! \, G_2 \\
\text{Programs} & D & ::= & P \mid G_2 \multimap D_1 \mid D_1 \,\&\, D_2 \mid \top \mid \forall x.\, D \mid G_2 \supset D_1
\end{array}$$

We have lined up the synchronous goals with their counterparts as synchronous programs just below, as explained via residuation.

Strictly speaking, going back and forth between the $\Delta \Vdash G$ and $\Delta \gg G$ is unnecessary: we could coalesce the two into one because programs are

always fully synchronous. However, it highlights the difference between the synchronous and asynchronous right rules: Asynchronous decomposition in $\Delta \Vdash G$ is automatic and involves no choice, synchronous decomposition $\Delta \gg G$ involves a significant choice and may fail. Moreover, in just about any logical extension of focusing beyond this fragment, we need to pause when the goal becomes synchronous during the asynchronous decomposition phase and consider whether to focus on an assumption instead. Here, this would always fail.

In practice, it is convenient to admit an even slightly richer set of goals, whose meaning can be explained either via a transformation to the connectives already shown above or directly via synchronous or asynchronous rules for them (see Exercise 14.3).

14.6 Completeness of Focusing

Soundness of the focusing system is easy to see, since each rule is a restriction of the corresponding left and right rules for the non-focused sequent calculus. Completeness is somewhat more difficult. We can continue the path mapped out in the proof that various connectives are asynchronous as goals, proving that the same connectives are indeed synchronous as programs. Alternatively, we can prove a generalization of the cut elimination results for focused derivations and use that in an overall completeness proof. The references below give some pointers to the two different styles of proof in the literature.

14.7 Historical Notes

Cut elimination, one of the most fundamental theorems in logic, is due to Gentzen [3], who introduced the sequent calculus and natural deduction for both classical and intuitionistic logic and showed cut elimination. His formulation of the sequent calculus had explicit rules for exchange, weaking, and contraction, which make the proof somewhat more tedious than the one we presented here. I first provided proofs by simple nested structural induction, formalized in a logical framework, for intuitionistic and classical [5, 6] as well as linear logic [4].

Andreoli introduced focusing for classical linear logic [1] and proved its completeness through a number of inversion and admissibility properties. An alternative proof using cut elimination as a central lemma, applied to intuitionistic linear logic was given by Chaudhuri [2].

14.8 Exercises

Exercise 14.1 *Prove $A \multimap B$ to be asynchronous on the right in two ways:*

i. directly by induction, and

ii. by appealing to the admissibility of cut.

Exercise 14.2 *In order to prove the cut elimination theorem in the presence of unrestricted assumptions, we generalize to the following:*

1. *(Cut) If* $\Gamma; \Delta_A \Vdash A\ true$ *and* $\Gamma; \Delta_C, A\ res \Vdash C\ true$ *then* $\Gamma; \Delta_C, \Delta_A \Vdash C\ true$.
2. *(Cut!) If* $\Gamma; \cdot \Vdash A\ true$ *and* $(\Gamma, A\ ures); \Delta_C \Vdash C\ true$ *then* $\Gamma; \Delta_C \Vdash C\ true$

The second form of cut expresses that if we can prove A without using resources, it is legitimate to assume it as an unrestricted resource, essentially because we can generate as many copies of A as we need (it requires no resources).

The nested induction now proceeds first on the structure of the cut formula A, then on the form of cut where cut $<$ cut!, then simultaneously on the structures of the two given derivations $\mathcal{D}$ and $\mathcal{E}$. This means we can appeal to the induction hypothesis

1. *either on a subformula of A with arbitrary derivations, or*
2. *on the same formula A where cut! appeals to cut, or*
3. *on the same cut formula and same form of cut and same $\mathcal{D}$, but a subderivation of $\mathcal{E}$, or*
4. *on the same cut formula and same form of cut and same $\mathcal{E}$, but a subderivation of $\mathcal{D}$.*

Show the cases explicitly involving $!A$, $A \supset B$, and copy in this proof. You may assume that weakening the unrestricted assumptions by adding more is legitimate and does not change the structure of the given deduction. Note carefully appeals to the induction hypothesis and explain why they are legal.

Exercise 14.3 *We consider even larger set of goals to squeeze the last bit of convenience out of our language without actually affecting its properties.*

i. Give the rule(s) to allow $!G$ as a goal.

ii. Give the rule(s) to allow $\mathbf{1}$ *as a goal.*

iii. We could allow simultaneous conjunction on the left-hand side of linear implication goals, because $(D_1 \otimes D_2) \multimap G$ *is equivalent to* $D_1 \multimap (D_2 \multimap G)$*, which lies within the permitted fragment. Explore which formulas* R *could be allowed in goals of the form* $R \multimap G$ *because they can be eliminated by a local equivalence-preserving transformation such as the one for* $\otimes$*.*

iv. Now explore which formulas S *could be allowed in goals of the form* $S \supset G$ *without affecting the essence of the language.*

Exercise 14.4 *Prove that the focusing system with left and right rules is equivalent to the system with only right rules and residuation for atomic goals.*

Exercise 14.5 *Through residuation, we have introduced two new connectives to linear logic,* $A \otimes\!! B$ *and* $P' \doteq P$*, but we have only considered their right rules.*

Give corresponding left rules for them in the sequent calculus and prove cut elimination and identity for your rules.

14.9 References

[1] Jean-Marc Andreoli. Logic programming with focusing proofs in linear logic. *Journal of Logic and Computation*, 2(3):297–347, 1992.

[2] Kaustuv Chaudhuri. *The Focused Inverse Method for Linear Logic*. PhD thesis, Carnegie Mellon University, 2006. To appear.

[3] 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.

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

[5] Frank Pfenning. A structural proof of cut elimination and its representation in a logical framework. Technical Report CMU-CS-94-218, Department of Computer Science, Carnegie Mellon University, November 1994.

[6] Frank Pfenning. Structural cut elimination I. intuitionistic and classical logic. *Information and Computation*, 157(1/2):84–141, March 2000.