A Structural Proof of Cut Elimination and Its Representation in a Logical Framework

Frank Pfenning

November 1994
CMU-CS-94-218

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

We present new proofs of cut elimination for intuitionistic and classical sequent calculi. In both cases the proofs proceed by three nested structural inductions, avoiding the explicit use of multi-sets and termination measures on sequent derivations. This makes them amenable to elegant and concise representations in LF, which are given in full detail.

This work was supported by NSF Grant CCR-9303383
The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of NSF or the U.S. government.

Keywords: Logic, Cut Elimination, Logical Framework

Contents

1 Introduction **1**

2 Intuitionistic Sequent Calculus **2**

3 Proof Terms for the Sequent Calculus **8**

4 Representing Sequent Derivations in LF **10**

5 Admissibility of Cut **13**

6 Extension to Classical Logic **18**

7 Conclusion **24**

A Detailed Admissibility Proofs for Cut **26**
A.1 Intuitionistic Calculus 26
A.2 Classical Calculus 38

B Cut Elimination **55**
B.1 Intuitionistic Calculus 56
B.2 Classical Calculus 59

Acknowledgments **62**

References **62**

1 Introduction

Gentzen's sequent calculi [Gen35] for intuitionistic and classical logic have been the central tool in many proof-theoretical investigations and applications of logic in computer science such as logic programming (e.g. [MNPS91]) or automated theorem proving (e.g. [Wal90]). The central property of sequent calculi is *cut elimination* (Gentzen's *Hauptsatz*) which yields consistency of the logic as a corollary. The algorithm for cut elimination may be interpreted computationally, similar to the way that normalization for natural deduction may be viewed as functional computation. For the case of linear logic, this point was made by Girard [Gir87] and later elaborated by Abramsky [Abr93]; see also [Gal93] for a tutorial introduction.

Many proofs of cut elimination have been given in the literature yet, to our knowledge, none of them have been formalized even though this is clearly possible (see, for example, Matthews [Mat94] pencil-and-paper analysis of cut elimination for the $(\vee, \neg)$ fragment of classical propositional logic in FS_0). They are difficult to mechanize for a number of reasons which in combination are quite intimidating. Most proofs require tedious data structures (such as multi-sets) and use complex termination measures. They also involve global conditions on occurrences of parameters in sequent derivations. In this paper we present new proofs of cut elimination for intuitionistic and classical sequent calculi and give their representations in the logical framework LF [HHP93] as implemented in the Elf system [Pfe91]. Multi-sets are avoided altogether in these proofs, and termination measures are replaced by three nested structural inductions. Parameters are treated as variables bound in derivations, thus naturally capturing occurrence conditions. Starting point for the proofs is Kleene's sequent system G_3 [Kle52], which we derive systematically from the point of view that a sequent calculus should be a calculus of proof search for natural deductions. It can easily be related to Gentzen's original and other sequent calculi.

The reader interested in structural cut elimination for intuitionistic or classical logic, but not its formalization, should be able to follow this paper by ignoring the material regarding its implementation. In order to understand and appreciate the representation of the sequent calculus and the proof of cut elimination the reader should have a basic knowledge of the representation methodology of LF and the Elf meta-language; the interested reader is referred to [HHP93] and [MP91, Pfe91].

The remainder of the paper is organized as follows. In Section 2 we introduce a formulation of the intuitionistic sequent calculus motivated from natural deduction. In Section 3 we give a notation for proof terms that record the structure of the sequent derivation. This is an important intermediate step towards the representation of sequents in LF shown in Section 4. The proof of admissibility of cut in the intuitionistic sequent calculus and its implementation are the subject of Section 5. In Section 6 we extend these results to the classical case. We conclude with an assessment and some remarks about future work in Section 7. In Appendices A.1 (intuitionistic) and A.2 (classical) we give the complete implementations of admissibility of cut in Elf together with an automatically generated informal version of each case in the proof. Appendix B gives a formulation of cut elimination as a translation from a sequent calculus with cut to a sequent calculus without cut. For both, intuitionistic and classical logic, this is a direct corollary of the admissibility of cut in the corresponding cut-free system.

2 Intuitionistic Sequent Calculus

Logical frameworks such as hereditary Harrop formulas [MNPS91] and LF [HHP93] are inherently biased towards natural deduction because of the strong correspondence between natural deductions and the typed λ-expressions used for their representation. Finding an elegant encoding of sequents and sequent derivations in a logical framework is therefore the first critical issue in an implementation of a proof of cut elimination. Felty's representation [Fel89] in λProlog, for example, uses lists of hypotheses which is advantageous for search but makes a formal meta-theory prohibitively complex. Frameworks based on sequent calculi such as LU [Gir93] or Forum [Mil94] allow direct encodings, but they lack a notation for the proof terms that are required to describe cut elimination.

In this section we develop a formulation of the sequent calculus for intuitionistic logic by transcribing the process of searching for a natural deduction into an inference system. The proximity to natural deduction then allows a high-level encoding of sequent derivations in LF. The resulting sequent calculus is basically Kleene's system G_3 [Kle52] which he introduced to obtain a simple decidability proof for its propositional fragment. We assume familiarity with natural deduction.

We consider a complete set of logical connectives and quantifiers so that we do not miss any important issues. Atomic formulas $p(t_1, \ldots, t_n)$ for first-order terms $t_1, \ldots, t_n$ are denoted by P.

$$\text{Formulas} \quad A \quad ::= \quad P \mid A_1 \wedge A_2 \mid A_1 \supset A_2 \mid A_1 \vee A_2 \mid \neg A \mid \top \mid \bot \mid \forall x.\, A \mid \exists x.\, A$$

The notions of free and bound variable are defined as usual. We identify formulas that differ only in the names of their bound variables and write $[t/x]A$ for capture-avoiding substitution of t for x in A. We use A, B, and C to range over formulas.

The main judgment of natural deduction is derivability of a formula A, written as $\vdash A$, but we follow custom and mostly omit the turnstile in the presentation. In natural deduction the meaning of each logical connective or quantifier is given by introduction and elimination rules. The introduction rule specifies how to infer a formula with a given principal connective. The elimination rule specifies how we may use an assumption with a given principal connective. During search for a natural deduction our goal is to deduce C from hypotheses $A_1, \ldots, A_n$. We may take four kinds of actions.

1. We may solve the goal immediately when a hypothesis A_i is equal to C.

2. We may use an introduction rule to infer C. Each premise yields a new subgoal.

3. We may apply an elimination rule to a hypothesis A_i. Typically, this yields a new subgoal with an additional hypothesis.

4. We may introduce a lemma into the derivation.

We also observe from the nature of hypothetical reasoning:

1. The order in which hypotheses are assumed is irrelevant.

2. Hypotheses may be used arbitrarily often.

3. Hypothesis need not be used.

We abbreviate hypotheses $A_1, \ldots, A_n$ by Γ. Since the order of hypothesis is irrelevant we write $\Gamma = \Gamma', A$ if A occurs in Γ and Γ' consists of the remaining hypotheses. Note that the same hypothesis may occur more than once.

A *sequent* $\Gamma \longrightarrow C$ is a judgment representing the goal of deriving C from Γ. A derivation of $\Gamma \longrightarrow C$ represents a trace of a particular successful search, although in this paper we do not show the routine extraction of a natural deduction C from a sequent derivation. The proof search actions listed above give rise to various inference rules for the sequent calculus. Using our obervations about natural deduction we eliminate all structural rules from Gentzen's system by building them into each rule. Intuitively, weakening is incorporated into initial sequents and contraction is built into each left rule. Exchange remains implicit in the notation Γ, A.

Initial Sequents. The goal may be solved immediately when a hypothesis A matches the conclusion. In sequent form:

$$\frac{}{\Gamma, A \longrightarrow A}\,I$$

Introduction rules are used to reason backwards from the conclusion during search for a natural deduction. Consequently, they apply to the formula on the right-hand side of the sequent arrow. Dually, elimination rules are used to reason forward from hypotheses and thus apply to a formula on the left-hand side of the sequent arrow. Therefore, the sequent rules for each connective can be divided into *right* and *left* rules. We examine each of the connectives and quantifiers, showing the introduction and elimination and corresponding right and left sequent rules.

Conjunction. The introduction/right rules are straightforward.

$$\frac{A \qquad B}{A \wedge B}\,\wedge\mathrm{I} \qquad\qquad \frac{\Gamma \longrightarrow A \qquad \Gamma \longrightarrow B}{\Gamma \longrightarrow A \wedge B}\,\wedge R$$

For the elimination/left rules we have to remember to keep the hypothesis $A \wedge B$, since hypotheses may be used arbitrarily often in a natural deduction.

$$\frac{A \wedge B}{A}\,\wedge\mathrm{E_L} \qquad\qquad \frac{\Gamma, A \wedge B, A \longrightarrow C}{\Gamma, A \wedge B \longrightarrow C}\,\wedge L_1$$

$$\frac{A \wedge B}{B}\,\wedge\mathrm{E_R} \qquad\qquad \frac{\Gamma, A \wedge B, B \longrightarrow C}{\Gamma, A \wedge B \longrightarrow C}\,\wedge L_2$$

Implication. The premise of the introduction rule for an implication $A \supset B$ is a judgment hypothetical in A labelled u. The hypothesis is discharged at this inference which we indicate by a superscript on the instance of the inference rule. In the sequent formulation on the right, we add A to Γ: we have reduced the goal of deriving $A \supset B$ from Γ to the goal of deriving B from Γ and

A.

$$
\dfrac{\begin{array}{c}\overline{A}^{\,u}\\ \vdots \\ B\end{array}}{A \supset B}\supset\mathrm{I}^u
\qquad\qquad
\dfrac{\Gamma, A \longrightarrow B}{\Gamma \longrightarrow A \supset B}\supset R
$$

If we have an implication $A \supset B$ as a hypothesis while deriving C we use it by proving A (from the same hypotheses) and then assuming B as additional hypothesis for proving C.

$$
\dfrac{A \supset B \qquad A}{B}\supset\mathrm{E}
\qquad\qquad
\dfrac{\Gamma, A \supset B \longrightarrow A \qquad \Gamma, A \supset B, B \longrightarrow C}{\Gamma, A \supset B \longrightarrow C}\supset L
$$

In order to maintain the correspondence to natural deduction it is important to copy the implicational assumption $A \supset B$ to both premises, even though it is redundant in the right premise.

Disjunction. There are two introduction rules for disjunction in natural deduction and consequently two right rules for disjunction in the sequent calculus.

$$
\dfrac{A}{A \vee B}\vee\mathrm{I_L}
\qquad\qquad
\dfrac{\Gamma \longrightarrow A}{\Gamma \longrightarrow A \vee B}\vee R_1
$$

$$
\dfrac{B}{A \vee B}\vee\mathrm{I_R}
\qquad\qquad
\dfrac{\Gamma \longrightarrow B}{\Gamma \longrightarrow A \vee B}\vee R_2
$$

The elimination rule for disjunction explicitly refers to a conclusion C and is thus already closer to a sequent rule.

$$
\dfrac{A \vee B \qquad \begin{array}{c}\overline{A}^{\,u_1}\\ \vdots \\ C\end{array} \qquad \begin{array}{c}\overline{B}^{\,u_2}\\ \vdots \\ C\end{array}}{C}\vee\mathrm{E}^{u_1,u_2}
\qquad\qquad
\dfrac{\Gamma, A \vee B, A \longrightarrow C \qquad \Gamma, A \vee B, B \longrightarrow C}{\Gamma, A \vee B \longrightarrow C}\vee L
$$

Negation. Negation in intuitionistic natural deduction is usually explained by considering $\neg A$ as an abbreviation of $A \supset \bot$. In sequent calculi, on the other hand, it is modeled by an empty right-hand side. These do not correspond so we need to find another formulation for negation. The goal is to find an introduction rule for $\neg A$ that does not require another logical symbol (such as $\bot$). We use the idea of a judgment parametric in a propositional variable p to achieve this. For the sequent calculus this means that p may not occur in Γ or A.

$$
\dfrac{\begin{array}{c}\overline{A}^{\,u}\\ \vdots \\ p\end{array}}{\neg A}\neg\mathrm{I}^{p,u}
\qquad\qquad
\dfrac{\Gamma, A \longrightarrow p}{\Gamma \longrightarrow \neg A}\neg R^p
$$

The elimination rule is simpler.

$$\frac{\neg A \qquad A}{C}\,\neg\mathrm{E} \qquad\qquad \frac{\Gamma, \neg A \longrightarrow A}{\Gamma, \neg A \longrightarrow C}\,\neg L$$

It may not be obvious at first, but the introduction and elimination rule (and also the left and right rules) match up precisely. We will see this in the proof of cut-elimination.

Truth. There is only an introduction rule for $\top$ in natural deduction. Correspondingly, we only have a right rule in the sequent calculus.

$$\frac{}{\top}\,\top\mathrm{I} \qquad\qquad \frac{}{\Gamma \longrightarrow \top}\,\top R$$

Falsehood. Dually, there is only an elimination and corresponding left rule for $\bot$.

$$\frac{\bot}{C}\,\bot\mathrm{E} \qquad\qquad \frac{}{\Gamma, \bot \longrightarrow C}\,\bot L$$

Universal Quantification. Universal quantification employs an individual parameter. In the sequent calculus, this means that the parameter a must be new, that is, it may not appear in Γ or $\forall x.\ A$.

$$\frac{[a/x]A}{\forall x.\ A}\,\forall\mathrm{I}^a \qquad\qquad \frac{\Gamma \longrightarrow [a/x]A}{\Gamma \longrightarrow \forall x.\ A}\,\forall R^a$$

In the elimination rule we substitute an arbitrary term t for a universally quantified variable x. This substitution may need to rename bound variables so that no variable free in t is captured by a quantifier in A.

$$\frac{\forall x.\ A}{[t/x]A}\,\forall\mathrm{E} \qquad\qquad \frac{\Gamma, \forall x.\ A, [t/x]A \longrightarrow C}{\Gamma, \forall x.\ A \longrightarrow C}\,\forall L$$

In the customary notation for this elimination and right rules, the term t is not uniquely determined if x does not occur free in A. In the proof term calculus in Section 3 we make sure that t occurs explicitly in order to avoid potential ambiguities.

Existential Quantification. The introduction/right rules are straightforward.

$$\frac{[t/x]A}{\exists x.\ A}\,\exists\mathrm{I} \qquad\qquad \frac{\Gamma \longrightarrow [t/x]A}{\Gamma \longrightarrow \exists x.\ A}\,\exists R$$

The apparent complexity of the elimination rule vanishes when viewed in the sequent calculus. Once again, a must be a new parameter, that is, it may not occur in Γ, $\exists x.\ A$, or C.

$$\frac{\exists x.\ A \qquad \begin{array}{c}\overline{[a/x]A}^{\,u}\\ \vdots \\ C\end{array}}{C}\,\exists\mathrm{E}^{a,u} \qquad\qquad \frac{\Gamma, \exists x.\ A, [a/x]A \longrightarrow C}{\Gamma, \exists x.\ A \longrightarrow C}\,\exists L^a$$

Lemma Introduction. Introducing a lemma A during the search for a natural deduction corresponds directly to the cut rule in the sequent calculus: in order to derive C from Γ we derive A and show that with the additional hypothesis A we can derive C.

$$\frac{\Gamma \longrightarrow A \qquad \Gamma, A \longrightarrow C}{\Gamma \longrightarrow C}\,\mathrm{Cut}$$

The theorem of *cut elimination* states that every sequent $\Gamma \longrightarrow C$ that is derivable in the system with cut, can also be derived in the system without cut. An equivalent, but slightly more convenient way of stating this is that cut is *admissible* in the system without cut, that is, whenever we can derive the premisses of this rule without using cut, we can also derive the conclusion without using cut. We concentrate our development on admissibility of cut and relegate cut elimination in the sense of Gentzen to Appendix B.

We summarize the rules for the cut-free calculus G_3. They are sound and complete in the usual sense, which can easily be shown by relating them to Gentzen's sequent calculus or to natural deduction (see Theorem 2).

$$\frac{}{\Gamma, A \longrightarrow A}\,I$$

$$\frac{\Gamma \longrightarrow A \qquad \Gamma \longrightarrow B}{\Gamma \longrightarrow A \wedge B}\,{\wedge}R \qquad \frac{\Gamma, A \wedge B, A \longrightarrow C}{\Gamma, A \wedge B \longrightarrow C}\,{\wedge}L_1 \qquad \frac{\Gamma, A \wedge B, B \longrightarrow C}{\Gamma, A \wedge B \longrightarrow C}\,{\wedge}L_2$$

$$\frac{\Gamma, A \longrightarrow B}{\Gamma \longrightarrow A \supset B}\,{\supset}R \qquad \frac{\Gamma, A \supset B \longrightarrow A \qquad \Gamma, A \supset B, B \longrightarrow C}{\Gamma, A \supset B \longrightarrow C}\,{\supset}L$$

$$\frac{\Gamma \longrightarrow A}{\Gamma \longrightarrow A \vee B}\,{\vee}R_1 \qquad \frac{\Gamma \longrightarrow B}{\Gamma \longrightarrow A \vee B}\,{\vee}R_2 \qquad \frac{\Gamma, A \vee B, A \longrightarrow C \qquad \Gamma, A \vee B, B \longrightarrow C}{\Gamma, A \vee B \longrightarrow C}\,{\vee}L$$

$$\frac{\Gamma, A \longrightarrow p}{\Gamma \longrightarrow \neg A}\,\neg R^p \qquad \frac{\Gamma, \neg A \longrightarrow A}{\Gamma, \neg A \longrightarrow C}\,\neg L$$

$$\frac{}{\Gamma \longrightarrow \top}\top R$$

$$\frac{}{\Gamma, \bot \longrightarrow C}\bot L$$

$$\frac{\Gamma \longrightarrow [a/x]A}{\Gamma \longrightarrow \forall x.\ A}\forall R^a \qquad \frac{\Gamma, \forall x.\ A, [t/x]A \longrightarrow C}{\Gamma, \forall x.\ A \longrightarrow C}\forall L$$

$$\frac{\Gamma \longrightarrow [t/x]A}{\Gamma \longrightarrow \exists x.\ A}\exists R \qquad \frac{\Gamma, \exists x.\ A, [a/x]A \longrightarrow C}{\Gamma, \exists x.\ A \longrightarrow C}\exists L^a$$

The *principal formula* of an inference is either the formula being introduced on the left or the right, or the formula occurring on the left and the right in an initial sequent. All other formulas are *side formulas* of the last inference. These notions also apply to individual formula occurrences.

The system without cut is easily seen to be consistent, since there is no rule with which one could infer the sequent $\longrightarrow \bot$. In terms of natural deduction this means that introducing a lemma during search is never necessary: If there is a deduction of C from hypotheses Γ we can find it by using only introduction rules reasoning backwards from C and using only elimination rules reasoning forward from the hypothesis Γ. This yields consistency of natural deduction as an easy corollary, since there is no introduction rule for $\bot$.

Our formulation of the sequent calculus has the following elementary properties. It is not important for our main development, but these properties also hold for the system with the cut rule.

Lemma 1 (Elementary Properties of Sequent Calculus)

1. *(Weakening) If* $\Gamma \longrightarrow C$ *then* $\Gamma, A \longrightarrow C$.
2. *(Contraction) If* $\Gamma, A, A \longrightarrow C$ *then* $\Gamma, A \longrightarrow C$.
3. *(Term Substitution) If* $\Gamma \longrightarrow C$ *with free individual parameter* a *then* $[t/a]\Gamma \longrightarrow [t/a]C$ *for any term* t.
4. *(Formula Substitution) If* $\Gamma \longrightarrow C$ *with free propositional parameter* p *then* $[A/p]\Gamma \longrightarrow [A/p]C$ *for any formula* A.

Proof: All are immediate by induction over the structure of the derivation of the assumption. In all cases the structure of the derivation is not changed—a property made explicit in Lemma 3. □

Now let $\Gamma \xrightarrow{LJ^-} C$ stand for derivability in Gentzen's sequent calculus LJ without cut (LJ$^-$), but augmented with rules for truth and falsehood. We can easily translate between derivations in LJ$^-$ and G$_3$ by removing or inserting instances of the structural rules in LJ$^-$. Empty right-hand sides as permitted in LJ present only a small complication.

Theorem 2 (Equivalence of G_3 and LJ^-)

1. *$\Gamma \longrightarrow C$ iff $\Gamma \stackrel{LJ^-}{\longrightarrow} C$, and*

2. *$\Gamma \longrightarrow p$ for a parameter p that does not occur in Γ iff $\Gamma \stackrel{LJ^-}{\longrightarrow}$.*

Proof: The proof in both directions proceeds by induction over the structure of the given derivation. We require weakening and contraction lemmas for G_3 (Lemma 1) to model the structural rules of weakening and contraction in LJ. □

At this point we could define the size of a formula A as the number of its connectives and quantifiers, the length of a derivation as the number of inference rules it contains, and then prove the admissibility of cut in the cut-free system by three nested inductions over the size of the cut formula and the lengths of the derivations of $\Gamma \longrightarrow A$ and $\Gamma, A \longrightarrow C$. However, such a proof is not well-suited for implementation. The first difficulty is the implementation of the sequent calculus itself and the notions of multi-set it requires. The second difficulty is that most proof checkers or theorem provers use structural induction more effectively than proofs with termination measures. We will return to both points in the next section.

3 Proof Terms for the Sequent Calculus

The sequent rules as given so far do not preserve all the information present in a natural deduction. For example, the two different natural deductions of $A \supset (A \supset A)$ below are mapped to same sequent derivation.

$$
\dfrac{\dfrac{\overline{A}\,u}{A \supset A}\supset\mathrm{I}^w}{A \supset (A \supset A)}\supset\mathrm{I}^u
\qquad
\dfrac{\dfrac{\overline{A}\,w}{A \supset A}\supset\mathrm{I}^w}{A \supset (A \supset A)}\supset\mathrm{I}^u
\qquad
\dfrac{\dfrac{\overline{A, A \longrightarrow A}\,I}{A \longrightarrow A \supset A}\supset R}{\longrightarrow A \supset (A \supset A)}\supset R
$$

In the sequent notation we cannot tell which of the two identical hypotheses was used in the initial sequent. If we are only interested in derivability (or truth), then this is tolerable. However, if we are interested in the *structure* of derivations such ambiguities should be resolved. Clearly, for many applications in computer science and, of course, also for the proof of cut elimination the structure of derivations is of central importance. We therefore endow sequent derivations with proof terms that resolve this kind of ambiguity. This is also an important intermediate step toward the representation of the rules in LF.

There are at least three distinct roles that proof terms may play for a sequent calculus, an issue recognized by Gallier [Gal93] and Breazu-Tannen et al. [BTKP93]. The most immediate perhaps is to annotate sequent derivations with λ-terms that represent the natural deductions they correspond to. The second is to think of proof terms as expressions in a programming language and view a sequent derivation as a typing derivation. The third is to view proof terms as a compact notation for sequent derivations from which they may essentially be reconstructed. This view is particularly useful for our endeavor, since the representation in a logical framework should also have this property.

The first step is to label hypotheses. The second is to record a proof term d on the right of the sequent arrow. A sequent then has the form $\Gamma \longrightarrow d : A$ where Γ has the form $h_1{:}A_1, \ldots, h_n{:}A_n$. We assume that all hypothesis labels in a context are distinct. In order to avoid confusion with similar, but subtly different proof term notations in the literature, we systematically introduce precisely one new proof term constructor for each inference rule of the sequent calculus and give each a descriptive name. Rules that introduce parameters or hypotheses bind variables at the level of proof terms—a phenomenon which should be familiar from the Curry-Howard isomorphism. The idea of *higher-order abstract syntax* (here applied to a syntax for proof terms) is to reduce all binding operators to one, namely λ. This makes it immediately syntactically apparent which variables are bound and where. We also indicate the "type" of bound variables: they may bind individuals $(x{:}i)$, formulas $(p{:}o)$ or hypotheses $(h{:}A)$.

$$\frac{}{\Gamma, h{:}A \longrightarrow \mathrm{axiom}\; h : A}\, I$$

$$\frac{\Gamma \longrightarrow d_1 : A \qquad \Gamma \longrightarrow d_2 : B}{\Gamma \longrightarrow \mathrm{andr}\; d_1\, d_2 : A \wedge B}\, \wedge R$$

$$\frac{\Gamma, h{:}A \wedge B, h_1{:}A \longrightarrow d : C}{\Gamma, h{:}A \wedge B \longrightarrow \mathrm{andl}_1\, (\lambda h_1{:}A.\; d)\; h : C}\, \wedge L_1$$

$$\frac{\Gamma, h{:}A \wedge B, h_2{:}B \longrightarrow d : C}{\Gamma, h{:}A \wedge B \longrightarrow \mathrm{andl}_2\, (\lambda h_2{:}B.\; d)\; h : C}\, \wedge L_2$$

$$\frac{\Gamma, h{:}A \longrightarrow d : B}{\Gamma \longrightarrow \mathrm{impr}\, (\lambda h{:}A.\; d) : A \supset B}\, \supset R \qquad \frac{\Gamma, h{:}A \supset B \longrightarrow d_1 : A \qquad \Gamma, h{:}A \supset B, h_2{:}B \longrightarrow d_2 : C}{\Gamma, h{:}A \supset B \longrightarrow \mathrm{impl}\; d_1\, (\lambda h_2{:}B.\; d_2)\; h : C}\, \supset L$$

$$\frac{\Gamma \longrightarrow d : A}{\Gamma \longrightarrow \mathrm{orr}_1^B\; d : A \vee B}\, \vee R_1$$

$$\frac{\Gamma \longrightarrow d : B}{\Gamma \longrightarrow \mathrm{orr}_2^A\; d : A \vee B}\, \vee R_2$$

$$\frac{\Gamma, h{:}A \vee B, h_1{:}A \longrightarrow d_1 : C \qquad \Gamma, h{:}A \vee B, h_2{:}B \longrightarrow d_2 : C}{\Gamma, h{:}A \vee B \longrightarrow \mathrm{orl}\, (\lambda h_1{:}A.\; d_1)\, (\lambda h_2{:}B.\; d_2)\; h : C}\, \vee L$$

$$\frac{\Gamma, h{:}A \longrightarrow d : p}{\Gamma \longrightarrow \mathrm{notr}\, (\lambda p{:}o.\; \lambda h{:}A.\; d) : \neg A}\, \neg R^p \qquad \frac{\Gamma, h{:}\neg A \longrightarrow d : A}{\Gamma, h{:}\neg A \longrightarrow \mathrm{notl}^C\; d\; h : C}\, \neg L$$

$$\frac{}{\Gamma \longrightarrow \mathrm{truer} : \top}\, \top R$$

$$\frac{}{\Gamma, h{:}\bot \longrightarrow \mathrm{falsel}^C\; h : C}\, \bot L$$

$$\frac{\Gamma \longrightarrow d : [a/x]A}{\Gamma \longrightarrow \mathrm{forallr}\ (\lambda a{:}i.\ d) : \forall x.\ A}\forall R^a \qquad \frac{\Gamma, h{:}\forall x.\ A, h_1{:}[t/x]A \longrightarrow d : C}{\Gamma, h{:}\forall x.\ A \longrightarrow \mathrm{foralll}\, t\ (\lambda h_1{:}[t/x]A.\ d)\ h : C}\forall L$$

$$\frac{\Gamma \longrightarrow d : [t/x]A}{\Gamma \longrightarrow \mathrm{existsr}\, t\, d : \exists x.\ A}\exists R \qquad \frac{\Gamma, h{:}\exists x.\ A, h_1{:}[a/x]A \longrightarrow d : C}{\Gamma, h{:}\exists x.\ A \longrightarrow \mathrm{existsl}\ (\lambda a{:}i.\ \lambda h_1{:}[a/x]A.\ d)\ h : C}\exists L^a$$

Cut is not included as a primitive rule of inference, but its proof term (see Appendix B.1) would once again only reflect the structure of the derivation.

$$\frac{\Gamma \longrightarrow d : A \qquad \Gamma, h{:}A \longrightarrow e : C}{\Gamma \longrightarrow \mathrm{cut}\, d\ (\lambda h{:}A.\ e) : C}\mathrm{Cut}$$

Erasure of the proof terms from a sequent derivation in this calculus yields derivations from the rules given in the previous section. The proofs of the following properties are all immediate structural inductions. For typographical reasons we often write $\mathcal{D} :: (J)$ if $\mathcal{D}$ is a derivation of judgment J. The notion of substitution into a derivation should be self-explanatory, perhaps with the exception of $[h_1/h_2]\mathcal{D}$, where $h_1{:}A$ and $h_2{:}A$ are hypotheses. Here we mean the result of erasing hypothesis h_2 on the left-hand side of every sequent occurring in $\mathcal{D}$ and substituting h_1 in every place where h_2 occurs on the right-hand side of a sequent. This may require renaming some locally bound hypotheses to avoid capture of h_1. We write $(\mathcal{D}, h{:}A)$ for the result of adding hypothesis $h{:}A$ to every sequent in $\mathcal{D}$, possibly renaming parameters introduced in $\mathcal{D}$ so as not to conflict with parameters in A.

Lemma 3 (Basic Properties of Sequent Calculus with Proof Terms) *The intuitionistic sequent calculus with proof terms satisfies the following properties.*

1. *(Weakening) If* $\mathcal{D} :: (\Gamma \longrightarrow d : C)$ *then* $(\mathcal{D}, h{:}A) :: (\Gamma, h{:}A \longrightarrow d : C)$ *where* h *is a new label.*
2. *(Contraction or Hypothesis Substitution) If* $\mathcal{D} :: (\Gamma, h_1{:}A, h_2{:}A \longrightarrow d : C)$ *then* $[h_1/h_2]\mathcal{D} :: (\Gamma, h_1{:}A \longrightarrow [h_1/h_2]d : C)$.
3. *(Term Substitution) If* $\mathcal{D} :: (\Gamma \longrightarrow d : C)$ *is a derivation with free individual parameter* a *then* $[t/a]\mathcal{D} :: ([t/a]\Gamma \longrightarrow [t/a]d : [t/a]C)$.
4. *(Formula Substitution) If* $\mathcal{D} :: (\Gamma \longrightarrow d : C)$ *is a derivation with free formula parameter* p *then* $[A/p]\mathcal{D} :: ([A/p]\Gamma \longrightarrow [A/p]d : [A/p]C)$.
5. *(Uniqueness) If* $\mathcal{D} :: (\Gamma \longrightarrow d : C)$ *and* $\mathcal{D}' :: (\Gamma \longrightarrow d : C')$ *then* $\mathcal{D} = \mathcal{D}'$ *and* $C = C'$ *(modulo variable renaming).*

4 Representing Sequent Derivations in LF

In this section we briefly summarize the representation of formulas in LF using the idea of higher-order abstract syntax and show how the proof terms of the previous section can be converted to an adequate encoding of the sequent calculus. Readers interested primarily in the proof of cut elimination itself may safely skip this section.

For the sake of brevity we show the actual code in Elf [Pfe91], an implementation of LF which permits type declarations with implicit quantifiers. Elf also gives an operational interpretation to signatures as logic programs which will be of interest later in the implementation of cut elimination. First, the representation of formulas. The obvious representation function $\ulcorner\cdot\urcorner$ is a compositional bijection between canonical (= long $\beta\eta$) LF objects of type o (in an appropriate context) and formulas (see [HHP93]). An important characteristic of this encoding (and the others we give below) is that variables of the object language are mapped to variables of the meta-language. Consequently, variables that are bound in the object language must be bound with corresponding scope in the meta-language.

```
i : type.  % individuals
o : type.  % formulas

and    : o -> o -> o.              true   : o.
imp    : o -> o -> o.              false  : o.
or     : o -> o -> o.              forall : (i -> o) -> o.
not    : o -> o.                   exists : (i -> o) -> o.
```

As an example, consider the formula

$$(\forall x.\ (Ax \supset B)) \supset ((\exists x.\ Ax) \supset B).$$

Here, A and B are meta-variables, and Ax indicates that A may contain free occurrences of x while B may not. In the LF meta-language, this is implemented by an explicit abstraction. Using infix notation (which is supported in Elf) the formula above is represented by

```
((forall [x:i] (A x imp B)) imp ((exists [x:i] A x) imp B).
```

in a context with `A:i -> o` and `B:o`. The concrete syntax `[x:U] M` stands for $\lambda x{:}U.\ M$ in the logical framework.

Before giving the signature for the sequent calculus we state the adequacy theorem since it is a useful guide in interpreting the declarations. We use $\vdash^{LF}$ for derivability in LF under the signature consisting of the declarations yet to come. Assume we have a derivation

$$\begin{array}{c}\mathcal{D}\\ h_1{:}A_1, \ldots, h_n{:}A_n \longrightarrow d : C\end{array}$$

with free individual parameters among $a_1, \ldots, a_k$ and propositional parameters among $p_1, \ldots, p_m$. Its representation $\ulcorner\mathcal{D}\urcorner$ is a canonical object M such that

$$a_1{:}i, \ldots, a_k{:}i, p_1{:}o, \ldots p_m{:}o, h_1{:}\mathrm{hyp}\,\ulcorner A_1\urcorner, \ldots, h_n{:}\mathrm{hyp}\,\ulcorner A_n\urcorner \vdash^{LF} M : \mathrm{conc}\,\ulcorner C\urcorner,$$

where hyp and conc are type families indexed by formulas. We call the representation *adequate* if $\ulcorner\cdot\urcorner$ is a bijection between cut-free sequent derivations and such well-typed canonical objects and if it is also *compositional* in sense that

$$\begin{array}{rl}\ulcorner[t/a]\mathcal{D}\urcorner = [\ulcorner t\urcorner/a]\ulcorner\mathcal{D}\urcorner, & \\ \ulcorner[C/p]\mathcal{D}\urcorner = [\ulcorner C\urcorner/p]\ulcorner\mathcal{D}\urcorner, & \qquad\text{and}\\ \ulcorner[h_1/h_2]\mathcal{D}\urcorner = [h_1/h_2]\ulcorner\mathcal{D}\urcorner. & \end{array}$$

One observes a strong similarity between the proof terms d and the representing LF objects M. In transcribing the proof terms into LF, we mainly have to take care to distinguish between hypotheses and conclusions via the type families hyp and conc. We do not give an explicit definition of $\ulcorner\mathcal{D}\urcorner$—the declarations below and their correspondence to proof terms are suggestive so that the diligent reader should be able to write it out without any problems. Note that `%` begins a comment that extends to the end of the line, that `{x:U}V` is Elf's concrete syntax for $\Pi x{:}U.\ V$, and that `[x:U]M` stands for $\lambda x{:}U.\ M$. Most Π-quantifiers are left implicit and are reconstructed by Elf's front end in proper dependency order and with their most general types.

```
hyp  : o -> type.  % Hypotheses (left)
conc : o -> type.  % Conclusion (right)

axiom : (hyp A -> conc A).

andr  : conc A
         -> conc B
         -> conc (A and B).

andl1 : (hyp A -> conc C)
         -> (hyp (A and B) -> conc C).

andl2 : (hyp B -> conc C)
         -> (hyp (A and B) -> conc C).

impr  : (hyp A -> conc B)
         -> conc (A imp B).

impl  : conc A
         -> (hyp B -> conc C)
         -> (hyp (A imp B) -> conc C).

orr1  : conc A
         -> conc (A or B).

orr2  : conc B
         -> conc (A or B).

orl   : (hyp A -> conc C)
         -> (hyp B -> conc C)
         -> (hyp (A or B) -> conc C).

notr  : ({p:o} hyp A -> conc p)
         -> conc (not A).

notl  : conc A
         -> (hyp (not A) -> conc C).

truer : conc (true).

falsel : (hyp (false) -> conc C).

forallr : ({a:i} conc (A a))
           -> conc (forall A).

foralll : {T:i} (hyp (A T) -> conc C)
           -> (hyp (forall A) -> conc C).

existsr : {T:i} conc (A T)
           -> conc (exists A).

existsl : ({a:i} hyp (A a) -> conc C)
           -> (hyp (exists A) -> conc C).
```

The encoding satisfies the representation theorem as outlined above. It circumvents many of the problems that ordinarily arise in representations of the sequent calculus. Multi-sets are avoided, since hypotheses on the left-hand side of the sequent arrow are transported into the LF context. Variable naming conditions are encoded through the usual functional representation of parametric judgments.

Theorem 4 (Adequacy of Sequent Representation) *The representation of sequent derivations in LF is adequate.*

Proof: By inductions over the structure of sequent derivations and canonical forms in LF. The proof requires Lemma 3. □

Since the representation is adequate, checking the validity of sequent derivations can be accomplished by type-checking their representations in LF. As an example, consider the following cut-free sequent derivation.

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{(\forall x.\,(Ax \supset B)), (\exists x.\,Ax), Aa, (Aa \supset B) \longrightarrow Aa}\,I \qquad \dfrac{}{(\forall x.\,(Ax \supset B)), (\exists x.\,Ax), Aa, (Aa \supset B), B \longrightarrow B}\,I}{(\forall x.\,(Ax \supset B)), (\exists x.\,Ax), Aa, (Aa \supset B) \longrightarrow B}\,{\supset}L}{(\forall x.\,(Ax \supset B)), (\exists x.\,Ax), Aa \longrightarrow B}\,\forall L}{(\forall x.\,(Ax \supset B)), (\exists x.\,Ax) \longrightarrow B}\,\exists L^a}{(\forall x.\,(Ax \supset B)) \longrightarrow ((\exists x.\,Ax) \supset B)}\,{\supset}R}{\longrightarrow ((\forall x.\,(Ax \supset B)) \supset ((\exists x.\,Ax) \supset B))}\,{\supset}R}{}
$$

Its representation in Elf is the following term.

```
[A:i -> o] [B:o]

(impr [h1:hyp (forall [x:i] A x imp B)]
   (impr [h2:hyp (exists [x:i] A x)]
      (existsl ([a:i] [h3:hyp (A a)]
                foralll a ([h4:hyp (A a imp B)]
                           impl (axiom h3)
                           ([h5:hyp B] axiom h5)
                           h4)
                h1)
         h2)))
```

Note that the values of variables that were implicitly quantified in the constant declarations are not explicitly supplied here, but reconstructed by Elf's front end.

5 Admissibility of Cut

The proof of cut elimination uses one principal lemma: the admissibility of cut in the cut-free system. From this, cut elimination follows by a simple structural induction (see Appendix B).

Theorem 5 (Admissibility of Cut) *Let $\mathcal{D} :: (\Gamma \longrightarrow d : A)$ and $\mathcal{E} :: (\Gamma, h{:}A \longrightarrow e : C)$ be cut-free sequent derivations. Then there exists a proof term f and a cut-free sequent derivation $\mathcal{F} :: (\Gamma \longrightarrow f : C)$.*

Proof: The proof proceeds by three nested structural inductions on A, d, and e. In other words, we may use the induction hypothesis for (immediate) subformulas of A and arbitrary d and e, or for A, a subterm of d and arbitrary e, and for A, d, and a subterm of e. We distinguish cases for $\mathcal{D}$ and $\mathcal{E}$, which is the same as distinguishing cases for the proof terms d and e, since they determine the derivation (Lemma 3(5)). The proof is constructive so that it describes an algorithm that computes a derivation $\mathcal{F}$ given the derivations $\mathcal{D}$ and $\mathcal{E}$.

The cases can be divided into four categories: (1) Either $\mathcal{D}$ or $\mathcal{E}$ is initial with A as its principal formula, (2) A is the principal formula of the last inference in both $\mathcal{D}$ and $\mathcal{E}$, (3) A is a side formula of the last inference in $\mathcal{D}$, and (4) A is a side formula of the last inference in $\mathcal{E}$. These classes are not mutually exclusive, so the algorithm induced by our proof is non-deterministic. We capture this non-determinism as a *relation* between $\ulcorner A \urcorner$, $\ulcorner \mathcal{D} \urcorner$, $\ulcorner \mathcal{E} \urcorner$, and $\ulcorner \mathcal{F} \urcorner$, which is implemented by a type family

```
ca : {A:o} conc A -> (hyp A -> conc C) -> conc C -> type.
```

Note that $\ulcorner \mathcal{E} \urcorner$ may use the hypothesis A in addition to the ambient hypotheses Γ which are implicit. We show how each case in the proof contributes a declaration to `ce`. First, the two cases where either $\mathcal{D}$ or $\mathcal{E}$ is an initial sequent with principal formula A.

Case:

$$\mathcal{D} = \frac{}{\Gamma', H{:}A \longrightarrow \text{axiom}\, H : A} I$$

and $\mathcal{E} :: (\Gamma', H{:}A, h{:}A \longrightarrow e : C)$ is arbitrary. Here $\Gamma = \Gamma', H{:}A$ and $d = \text{axiom}\, H$. Then we let $f = [H/h]e$ and

$$\mathcal{F} = \begin{array}{c} [H/h]\mathcal{E} \\ \Gamma', H{:}A \longrightarrow [H/h]e : C \end{array}.$$

The substitution of H for h in $\mathcal{E}$ is represented by applying the function that represents $\mathcal{E}$ to the representation of H. This gives the correct representation of the result by compositionality of $\ulcorner\cdot\urcorner$ and Lemma 3(2).

```
ca_axiom_l : ca A (axiom H) E (E H).
```

Case:

$$\mathcal{E} = \frac{}{\Gamma, h{:}A \longrightarrow \text{axiom}\, h : A} I$$

and $\mathcal{D} :: (\Gamma \longrightarrow d : A)$ is arbitrary. Then we let $f = d$ and $\mathcal{F} = \mathcal{D}$. The representation of this case is immediate.

```
ca_axiom_r : ca A D ([h:hyp A] axiom h) D.
```

Next we consider a case where the cut formula A is the principal formula of the last inference in both $\mathcal{D}$ and $\mathcal{E}$.

Case:

$$\mathcal{D} = \frac{\begin{array}{c} \mathcal{D}_2 \\ \Gamma, h_1{:}A_1 \longrightarrow d_2 : A_2 \end{array}}{\Gamma \longrightarrow \text{impr}\, (\lambda h_1{:}A_1.\, d_2) : A_1 \supset A_2} \supset R$$

and

$$\mathcal{E} = \frac{\begin{array}{c} \mathcal{E}_1 \\ \Gamma, h{:}A_1 \supset A_2 \longrightarrow e_1 : A_1 \end{array} \qquad \begin{array}{c} \mathcal{E}_2 \\ \Gamma, h{:}A_1 \supset A_2, h_2{:}A_2 \longrightarrow e_2 : C \end{array}}{\Gamma, h{:}A_1 \supset A_2 \longrightarrow \text{impl}\, e_1\, (\lambda h_2{:}A_2.\, e_2)\, h : C} \supset L.$$

Here $d = \mathrm{impr}\,(\lambda h_1{:}A_1.\ d_2)$ and $e = \mathrm{impl}\, e_1\, (\lambda h_2{:}A_2.\ e_2)\, h$. In this case we first need to eliminate the remaining copies of $A_1 \supset A_2$ from the hypotheses of $\mathcal{E}_1$ and $\mathcal{E}_2$. To this end we apply the induction hypothesis with all of $\mathcal{D}$ and the subderivation $\mathcal{E}_1$ to obtain an e_1' and $\mathcal{E}_1'$ such that

$\mathcal{E}_1' :: (\Gamma \longrightarrow e_1' : A_1)$ By i.h. on $A_1 \supset A_2$, d, and e_1

Similarly, we would like to eliminate the hypothesis h from $\mathcal{E}_2$, but $\mathcal{E}_2$ has an additional hypothesis h_2. Thus we must first weaken $\mathcal{D}$ to $(\mathcal{D}, h_2{:}A_2) :: (\Gamma, h_2{:}A_2 \longrightarrow d : A_1 \supset A_2)$. By Lemma 3(1), this does not change the proof term d. We can thus apply the induction hypothesis and obtain

$\mathcal{E}_2' :: (\Gamma, h_2{:}A_2 \longrightarrow e_2' : C)$ By i.h. on $A_1 \supset A_2$, d, and e_2

Now that we have eliminated the additional copies of $A_1 \supset A_2$ we can apply the ordinary step of reducing the cut to two new cuts, but both on smaller formulas (A_1 and A_2). Note that the proof terms e_1', e_2', and d_2' involved in these cuts may be much bigger, since they are the result of earlier appeals to the induction hypothesis.

$\mathcal{D}_2' :: (\Gamma \longrightarrow d_2' : A_2)$ By i.h. on A_1, e_1', and d_2

$\mathcal{F} :: (\Gamma \longrightarrow f : C)$ By i.h. on A_2, d_2', and e_2'

In the Elf representation, each of the four appeals to the induction hypothesis are implemented as recursive calls to `ca`. We use `A <- B` for `B -> A` to emphasize the operational reading of this declaration as part of a logic program in Elf to perform cut elimination. The backwards arrow associates to the left.

```
ca_imp  : ca (A1 imp A2) (impr D2)
           ([h:hyp (A1 imp A2)] impl (E1 h) (E2 h) h) F
           <- ca (A1 imp A2) (impr D2) E1 E1'
           <- ({h2:hyp A2}
                 ca (A1 imp A2) (impr D2)
                 ([h:hyp (A1 imp A2)] E2 h h2) (E2' h2))
           <- ca A1 E1' D2 D2'
           <- ca A2 D2' E2' F.
```

The weakening we mentioned above is implemented by the weakening which holds for LF: in the second subgoal, `D2` slips inside the scope of `h2`, but it may not actually depend on it.

Next we show a case where the cut formula A is a side formula of the last inference in $\mathcal{E}$. The idea in all cases where the cut formula is a side formula of the last inference R is the same: we appeal to the induction hypothesis on the premise(s) and then apply R to the resulting derivation(s).

Case:

$$\mathcal{E} = \dfrac{\begin{array}{c}\mathcal{E}_1\\ \Gamma', H{:}B_1 \wedge B_2, h_1{:}B_1, h{:}A \longrightarrow e_1 : C\end{array}}{\Gamma', H{:}B_1 \wedge B_2, h{:}A \longrightarrow \mathrm{andl}_1\,(\lambda h_1{:}B_1.\ e_1)\, H : C}\wedge L_1$$

and $\mathcal{D} :: (\Gamma', H{:}B_1 \wedge B_2 \longrightarrow d : A)$ is arbitrary. In this case, $\Gamma = \Gamma', H{:}B_1 \wedge B_2$ and $e = \mathrm{andl}_1\,(\lambda h_1{:}B_1.\ e_1)\, H$. After weakening $\mathcal{D}$ (without changing the proof term d!) we "cut" $(\mathcal{D}, h_1{:}B_1)$ and E_1 to obtain

$\mathcal{E}_1' :: (\Gamma', H{:}B_1 \wedge B_2, h_1{:}B_1 \longrightarrow e_1' : C)$ By i.h. on A, d, e_1

We now obtain $\mathcal{F}$ by applying the $\wedge L_1$ to $\mathcal{E}_1'$.

$$\mathcal{F} = \dfrac{\begin{array}{c}\mathcal{E}_1' \\ \Gamma', H{:}B_1 \wedge B_2, h_1{:}B_1 \longrightarrow e_1' : C\end{array}}{\Gamma', H{:}B_1 \wedge B_2 \longrightarrow \mathrm{andl}_1\,(\lambda h_1{:}B_1.\, e_1')\, H : C}\wedge L_1$$

Note how explicit abstractions and applications are employed to represent scoping of hypotheses in the Elf implementation of this case.

```
car_andl1: ca A D ([h:hyp A] andl1 (E1 h) H) (andl1 E1' H)
             <- ({h1:hyp B1} ca A D ([h:hyp A] E1 h h1) (E1' h1)).
```

All the remaining cases (there are 35 altogether) follow similar patterns. They are given in a more compact form in Appendix A.1. They require the usual substitution of terms for individual parameters in the cases for quantifiers, and formulas for propositional parameters in the case of negation. In the quantifier case we need that $[t/x]A$ is a subformula of $\forall x.\ A$, which can be justified by an appropriate structural induction principle for first-order formulas. □

What does the proof representation we show above achieve? First of all, it is operationally adequate, that is, it provides an implementation of an algorithm that eliminates cuts from sequent derivations. Execution of the signature above as an Elf program is illustrated through an example below. Furthermore, the implementation describes not just any admissibility proof of cut, but captures the computational content of the particular, informal constructive proof we presented. Clearly, this must remain an informal statement, since our "constructive proof" is not a formal, mathematical object. Our implementation is partially verified by the type checker which ensures correctness of the result of applying cut to the two given derivations. Here the dependent types play a critical role in guaranteeing the validity of all sequent derivations in the signature statically, which is the subject of the adequacy theorem (Theorem 4). On the other hand, due to the absence of induction principles in LF, parts of the informal argument are not formally verified through the type checker. They require an additional argument external to LF which is possible to carry out by hand, but exceedingly tedious. Automation of this external check is the subject of ongoing research.

Even if an external checker would verify that the signature represented a proof of admissibility of cut, there would still remain the issue if such a check can always be trusted (there may be bugs in the implementation, for example). Thus we believe that it is important that we should be able to recover informal, mathematical proofs from their formalization in a framework, that is, a proof checker should be able to "explain itself". For this particular case study we have implemented a program that translates the Elf signature into the critical parts of the informal argument, namely the sequences of appeals to the induction hypotheses in each case of the admissibility proof. We then inspected each of the 35 cases in the same way we would judge a proof in a paper submitted to a journal and verified the correctness of the implementation of the proof in Elf. The complete implementation and the informal presentation of each case are given in full detail in Appendix A.1. It remains to convince oneself that all cases are covered, which is not difficult since they are enumerated systematically.

In the remainder of this section, we illustrate how Elf can be used to execute the (constructive) proof of the admissibility theorem for cut. The first derivation (called $\mathcal{D}$ in the statement of the theorem) is

$$\dfrac{\dfrac{\dfrac{}{(\exists x.\,(Ax \vee Bx)), (Aa \vee Ba), Aa \longrightarrow Aa}\,I}{(\exists x.\,(Ax \vee Bx)), (Aa \vee Ba), Aa \longrightarrow (\exists x.\,Ax)}\,\exists R}{(\exists x.\,(Ax \vee Bx)), (Aa \vee Ba), Aa \longrightarrow ((\exists x.\,Ax) \vee (\exists x.\,Bx))}\,\vee R_1 \quad \dfrac{\dfrac{\dfrac{}{(\exists x.\,(Ax \vee Bx)), (Aa \vee Ba), Ba \longrightarrow Ba}\,I}{(\exists x.\,(Ax \vee Bx)), (Aa \vee Ba), Ba \longrightarrow (\exists x.\,Bx)}\,\exists R}{(\exists x.\,(Ax \vee Bx)), (Aa \vee Ba), Ba \longrightarrow ((\exists x.\,Ax) \vee (\exists x.\,Bx))}\,\vee R_2$$

$$\dfrac{\dfrac{\cdots \qquad \cdots}{(\exists x.\,(Ax \vee Bx)), (Aa \vee Ba) \longrightarrow ((\exists x.\,Ax) \vee (\exists x.\,Bx))}\,\vee L}{(\exists x.\,(Ax \vee Bx)) \longrightarrow ((\exists x.\,Ax) \vee (\exists x.\,Bx))}\,\exists L^a$$

which is represented by

```
[A:i -> o] [B:i -> o]
[h1:hyp (exists [x:i] (A x or B x))]

(existsl ([a:i] [h2:hyp (A a or B a)]
            (orl ([h3:hyp (A a)] orr1 (existsr a (axiom h3)))
               ([h4:hyp (B a)] orr2 (existsr a (axiom h4)))
               h2))
   h1)
```

The second derivation (slightly more general than what we need) is

$$\dfrac{\dfrac{\dfrac{}{(A' \vee B'), A' \longrightarrow A'}\,I}{(A' \vee B'), A' \longrightarrow (B' \vee A')}\,\vee R_2 \quad \dfrac{\dfrac{}{(A' \vee B'), B' \longrightarrow B'}\,I}{(A' \vee B'), B' \longrightarrow (B' \vee A')}\,\vee R_1}{(A' \vee B') \longrightarrow (B' \vee A')}\,\vee L$$

which is represented by

```
[A':o] [B':o]
[h:hyp (A' or B')]

(orl ([h2:hyp A'] orr2 (axiom h2))
   ([h3:hyp B'] orr1 (axiom h3))
   h)
```

In this second derivation we instantiate the meta-variables A' and B' to $\exists x.\,Ax$ and $\exists x.\,Bx$, respectively. To obtain a cut-free derivation of $\exists x.\,(Ax \vee Bx) \longrightarrow (\exists x.\,Bx) \vee (\exists x.\,Ax)$ we then pose the following query.

```
?- {A:i -> o} {B:i -> o}
   {h1:hyp (exists [x:i] (A x or B x))}
   ca ((exists [x:i] A x) or (exists [x:i] B x))
      (existsl ([a:i] [h2:hyp (A a or B a)]
```

```
              (orl ([h3:hyp (A a)] orr1 (existsr a (axiom h3)))
                 ([h4:hyp (B a)] orr2 (existsr a (axiom h4)))
                 h2))
      h1)
   ([h:hyp ((exists [x:i] A x) or (exists [x:i] B x))]
      (orl ([h2:hyp (exists [x:i] A x)] orr2 (axiom h2))
         ([h3:hyp (exists [x:i] B x)] orr1 (axiom h3))
         h))
   (F A B h1).
```

The only free variable in this query is `F` which may depend on `A`, `B`, and the hypothesis `h1`. The first (and in this case only) answer we obtain is the substitution

```
F =
   [A:i -> o] [B:i -> o] [h1:hyp (exists [x:i] A x or B x)]
      existsl
         ([a:i] [h:hyp (A a or B a)]
             orl ([h11:hyp (A a)] orr2 (existsr a (axiom h11)))
                 ([h2:hyp (B a)] orr1 (existsr a (axiom h2))) h) h1.
```

which represents the expected derivation

$$
\dfrac{\dfrac{
\dfrac{\dfrac{\dfrac{}{(\exists x.\,(Ax \lor Bx)), (Aa \lor Ba), Aa \longrightarrow Aa}\,I}{(\exists x.\,(Ax \lor Bx)), (Aa \lor Ba), Aa \longrightarrow (\exists x.\,Ax)}\,\exists R}{(\exists x.\,(Ax \lor Bx)), (Aa \lor Ba), Aa \longrightarrow ((\exists x.\,Bx) \lor (\exists x.\,Ax))}\,{\lor}R_2
\qquad
\dfrac{\dfrac{\dfrac{}{(\exists x.\,(Ax \lor Bx)), (Aa \lor Ba), Ba \longrightarrow Ba}\,I}{(\exists x.\,(Ax \lor Bx)), (Aa \lor Ba), Ba \longrightarrow (\exists x.\,Bx)}\,\exists R}{(\exists x.\,(Ax \lor Bx)), (Aa \lor Ba), Ba \longrightarrow ((\exists x.\,Bx) \lor (\exists x.\,Ax))}\,{\lor}R_1
}{(\exists x.\,(Ax \lor Bx)), (Aa \lor Ba) \longrightarrow ((\exists x.\,Bx) \lor (\exists x.\,Ax))}\,{\lor}L}{(\exists x.\,(Ax \lor Bx)) \longrightarrow ((\exists x.\,Bx) \lor (\exists x.\,Ax))}\,\exists L^a
$$

6 Extension to Classical Logic

In natural deduction we obtain classical logic by adding another inference rule that breaks the symmetry of introduction and elimination rules. This rule might be excluded middle, indirect proof, or double negation elimination. In sequent calculus, classical logic is usually handled by allowing multiple conclusions, that is, a sequent has the form $\Gamma \longrightarrow \Delta$, where both Γ and Δ are lists (or multi-sets) of formulas. This exhibits deep symmetries in classical logic which are not so obvious in natural deduction form. The duality between left and right rules is now perfect, as is the duality of conjunction and disjunction, truth and falsehood, universal and existential quantification, and the self-duality of negation. Unfortunately, the gap between natural deduction and sequent calculus has become wider, so our rules are motivated by an extension of the intuitionistic case to multiple conclusions, rather than directly from natural deduction. For the proof of cut elimination and our representation it is important that Gentzen's structural rules remain implicit: The principal formula of an inference must always be copied to all premises along with all side formulas.

$$
\dfrac{}{\Gamma, A \longrightarrow A, \Delta}\,I
$$

$$\frac{\Gamma \longrightarrow A, A \wedge B, \Delta \qquad \Gamma \longrightarrow B, A \wedge B, \Delta}{\Gamma \longrightarrow A \wedge B, \Delta} \wedge R \qquad \frac{\Gamma, A \wedge B, A \longrightarrow \Delta}{\Gamma, A \wedge B \longrightarrow \Delta} \wedge L_1 \qquad \frac{\Gamma, A \wedge B, B \longrightarrow \Delta}{\Gamma, A \wedge B \longrightarrow \Delta} \wedge L_2$$

$$\frac{\Gamma, A \longrightarrow B, A \supset B, \Delta}{\Gamma \longrightarrow A \supset B, \Delta} \supset R \qquad \frac{\Gamma, A \supset B \longrightarrow A, \Delta \qquad \Gamma, A \supset B, B \longrightarrow \Delta}{\Gamma, A \supset B \longrightarrow \Delta} \supset L$$

$$\frac{\Gamma \longrightarrow A, A \vee B, \Delta}{\Gamma \longrightarrow A \vee B, \Delta} \vee R_1 \qquad \frac{\Gamma \longrightarrow B, A \vee B, \Delta}{\Gamma \longrightarrow A \vee B, \Delta} \vee R_2 \qquad \frac{\Gamma, A \vee B, A \longrightarrow \Delta \qquad \Gamma, A \vee B, B \longrightarrow \Delta}{\Gamma, A \vee B \longrightarrow \Delta} \vee L$$

$$\frac{\Gamma, A \longrightarrow \neg A, \Delta}{\Gamma \longrightarrow \neg A, \Delta} \neg R \qquad \frac{\Gamma, \neg A \longrightarrow A, \Delta}{\Gamma, \neg A \longrightarrow \Delta} \neg L$$

$$\frac{}{\Gamma \longrightarrow \top, \Delta} \top R$$

$$\frac{}{\Gamma, \bot \longrightarrow \Delta} \bot L$$

$$\frac{\Gamma \longrightarrow [a/x]A, \forall x.\ A, \Delta}{\Gamma \longrightarrow \forall x.\ A, \Delta} \forall R^a \qquad \frac{\Gamma, \forall x.\ A, [t/x]A \longrightarrow \Delta}{\Gamma, \forall x.\ A \longrightarrow \Delta} \forall L$$

$$\frac{\Gamma \longrightarrow [t/x]A, \exists x.\ A, \Delta}{\Gamma \longrightarrow \exists x.\ A, \Delta} \exists R \qquad \frac{\Gamma, \exists x.\ A, [a/x]A \longrightarrow \Delta}{\Gamma, \exists x.\ A \longrightarrow \Delta} \exists L^a$$

As in the intuitionistic case, we exclude cut from the system and show that it is admissible. It has the form

$$\frac{\Gamma \longrightarrow A, \Delta \qquad \Gamma, A \longrightarrow \Delta}{\Gamma \longrightarrow \Delta} \text{Cut}$$

The classical calculus satisfies *weakening* and *contraction* on both sides, and also the usual substitution properties. Weakening and contraction do not change the structure of the proof, which is made explicit below in Lemma 6. The equivalence to Gentzen's calculus LK is also easy to establish by inserting or removing appropriate structural rules; we skip the routine details here.

The assignment of proof terms reflects the symmetry between the left- and right-hand sides of a sequent in that we label both negative (left-hand side) and positive (right-hand side) formulas with *variables*. A proof term d then annotates the whole sequent; we write it above the sequent arrow:

$$n_1{:}A_1, \ldots, n_j{:}A_j \xrightarrow{d} p_1{:}C_1, \ldots, p_k{:}C_k.$$

We use n (*negative*) for labels of formulas occurring on the left of the sequent arrow and p (*positive*) for labels of formulas occurring on the right of the sequent arrow. As in the intuitionistic calculus, our proof terms faithfully record the structure of the sequent derivation and have no immediate connection to computational interpretations. We again use λ and the idea of higher-order abstract syntax to delimit scope.

$$\frac{}{\Gamma, n{:}A \xrightarrow{\mathrm{axiom}\, n\, p} p{:}A, \Delta}\, I$$

$$\frac{\Gamma \xrightarrow{d_1} p_1{:}A, p{:}A \wedge B, \Delta \qquad \Gamma \xrightarrow{d_2} p_2{:}B, p{:}A \wedge B, \Delta}{\Gamma \xrightarrow{\mathrm{andr}\, (\lambda p_1{:}A.\, d_1)\, (\lambda p_2{:}B.\, d_2)\, p} p{:}A \wedge B, \Delta}\, \wedge R$$

$$\frac{\Gamma, n{:}A \wedge B, n_1{:}A \xrightarrow{d_1} \Delta}{\Gamma, n{:}A \wedge B \xrightarrow{\mathrm{andl}_1\, (\lambda n_1{:}A.\, d_1)\, n} \Delta}\, \wedge L_1$$

$$\frac{\Gamma, n{:}A \wedge B, n_2{:}B \xrightarrow{d_2} \Delta}{\Gamma, n{:}A \wedge B \xrightarrow{\mathrm{andl}_2\, (\lambda n_2{:}B.\, d_2)\, n} \Delta}\, \wedge L_2$$

$$\frac{\Gamma, n_1{:}A \xrightarrow{d} p_2{:}B, p{:}A \supset B, \Delta}{\Gamma \xrightarrow{\mathrm{impr}\, (\lambda n_1{:}A.\, \lambda p_2{:}B.\, d)\, p} p{:}A \supset B, \Delta}\, \supset R \qquad \frac{\Gamma, n{:}A \supset B \xrightarrow{d_1} p_1{:}A, \Delta \qquad \Gamma, n{:}A \supset B, n_2{:}B \xrightarrow{d_2} \Delta}{\Gamma, n{:}A \supset B \xrightarrow{\mathrm{impl}\, (\lambda p_1{:}A.\, d_1)\, (\lambda n_2{:}B.\, d_2)\, n} \Delta}\, \supset L$$

$$\frac{\Gamma \xrightarrow{d_1} p_1{:}A, p{:}A \vee B, \Delta}{\Gamma \xrightarrow{\mathrm{orr}_1\, (\lambda p_1{:}A.\, d_1)\, p} p{:}A \vee B, \Delta}\, \vee R_1$$

$$\frac{\Gamma \xrightarrow{d_2} p_2{:}B, p{:}A \vee B, \Delta}{\Gamma \xrightarrow{\mathrm{orr}_2\, (\lambda p_2{:}B.\, d_2)\, p} p{:}A \vee B, \Delta}\, \vee R_2$$

$$\frac{\Gamma, n{:}A \vee B, n_1{:}A \xrightarrow{d_1} \Delta \qquad \Gamma, n{:}A \vee B, n_2{:}B \xrightarrow{d_2} \Delta}{\Gamma, n{:}A \vee B \xrightarrow{\mathrm{orl}\, (\lambda n_1{:}A.\, d_1)\, (\lambda n_2{:}B.\, d_2)\, n} \Delta}\, \vee L$$

$$\frac{\Gamma, n{:}A \xrightarrow{d} p{:}\neg A, \Delta}{\Gamma \xrightarrow{\mathrm{notr}\, (\lambda n{:}A.\, d)\, p} p{:}\neg A, \Delta}\, \neg R \qquad \frac{\Gamma, n{:}\neg A \xrightarrow{d} p{:}A, \Delta}{\Gamma, n{:}\neg A \xrightarrow{\mathrm{notl}\, (\lambda p{:}A.\, d)\, n} \Delta}\, \neg L$$

$$\frac{}{\Gamma \xrightarrow{\mathrm{truer}\, p} p{:}\top, \Delta}\top R$$

$$\frac{}{\Gamma, n{:}\bot \xrightarrow{\mathrm{falsel}\, n} \Delta}\bot L$$

$$\frac{\Gamma \xrightarrow{d} p_1{:}[a/x]A, p{:}\forall x.\ A, \Delta}{\Gamma \xrightarrow{\mathrm{forallr}\,(\lambda a{:}i.\ \lambda p_1{:}[a/x]A.\ d)\, p} p{:}\forall x.\ A, \Delta}\forall R^a \qquad \frac{\Gamma, n{:}\forall x.\ A, n_1{:}[t/x]A \xrightarrow{d} \Delta}{\Gamma, n{:}\forall x.\ A \xrightarrow{\mathrm{foralll}\, t\,(\lambda n_1{:}[t/x]A.\ d)\, n} \Delta}\forall L$$

$$\frac{\Gamma \xrightarrow{d} p_1{:}[t/x]A, p{:}\exists x.\ A, \Delta}{\Gamma \xrightarrow{\mathrm{existsr}\, t\,(\lambda p_1{:}[t/x]A.\ d)\, p} p{:}\exists x.\ A, \Delta}\exists R \qquad \frac{\Gamma, n{:}\exists x.\ A, n_1{:}[a/x]A \xrightarrow{d} \Delta}{\Gamma, n{:}\exists x.\ A \xrightarrow{\mathrm{existsl}\,(\lambda a{:}i.\ \lambda n_1{:}[a/x]A.\ d)\, n} \Delta}\exists L^a$$

We generalize the various notions of substitution and weakening from the intuitionistic case in the obvious way. Substitution for formula parameters is not necessary here, since negation is handled in a different way. We then have:

Lemma 6 (Basic Properties of Classical Sequent Calculus with Proof Terms) *The classical sequent calculus with proof terms satisfies the following properties.*

1. *(Weakening) If $\mathcal{D} :: (\Gamma \xrightarrow{d} \Delta)$ then $(\mathcal{D}, n{:}A) :: (\Gamma, n{:}A \xrightarrow{d} \Delta)$ and $(\mathcal{D}, p{:}A) :: (\Gamma \xrightarrow{d} p{:}A, \Delta)$, where n and p are new labels.*
2. *(Contraction) If $\mathcal{D} :: (\Gamma, n_1{:}A, n_2{:}A \xrightarrow{d} \Delta)$ then $[n_1/n_2]\mathcal{D} :: (\Gamma, n_1{:}A \xrightarrow{[n_1/n_2]d} \Delta)$. Furthermore, if $\mathcal{D} :: (\Gamma \xrightarrow{d} p_1{:}A, p_2{:}A, \Delta)$ then $[p_1/p_2]\mathcal{D} :: (\Gamma \xrightarrow{[p_1/p_2]d} p_1{:}A, \Delta)$.*
3. *(Term Substitution) If $\mathcal{D} :: (\Gamma \xrightarrow{d} \Delta)$ is a derivation with free individual parameter a then $[t/a]\mathcal{D} :: ([t/a]\Gamma \xrightarrow{[t/a]d} [t/a]\Delta)$.*
4. *(Uniqueness) If $\mathcal{D} :: (\Gamma \xrightarrow{d} \Delta)$ and $\mathcal{D}' :: (\Gamma \xrightarrow{d} \Delta)$ then $\mathcal{D} = \mathcal{D}'$.*

Proof: By simple structural inductions. □

The LF representation closely models proof terms and is thus also symmetric with respect to formulas on the left and right: Both appear in the context of the LF typing judgment. That is, a cut-free derivation

$$\begin{array}{c}\mathcal{D}\\ n_1{:}A_1, \ldots, n_j{:}A_j \xrightarrow{d} p_1{:}C_1, \ldots, p_k{:}C_k\end{array}$$

with free individual parameters among $a_1, \ldots, a_m$ is represented by a term $M = \ulcorner\mathcal{D}\urcorner$ such that

$$a_1{:}i, \ldots, a_m{:}i, n_1{:}\mathrm{neg}\ulcorner A_1\urcorner, \ldots, n_j{:}\mathrm{neg}\ulcorner A_j\urcorner, p_1{:}\mathrm{pos}\ulcorner C_1\urcorner, \ldots, p_k{:}\mathrm{pos}\ulcorner C_k\urcorner \vdash^{LF} M : \#,$$

where neg and pos are type families indexed by formulas, and # is a new type, the type of every valid proof term. If we interpreted a sequent calculus as a refutational calculus, # would represent a contradiction. Below we show the representation of cut-free sequent derivations as an LF signature in the concrete syntax of Elf.

```
# : type.
neg : o -> type.
pos : o -> type.

axiom' : (neg A -> pos A -> #).

andr'  : (pos A -> #)
          -> (pos B -> #)
          -> (pos (A and B) -> #).

andl1' : (neg A -> #)
          -> (neg (A and B) -> #).

andl2' : (neg B -> #)
          -> (neg (A and B) -> #).

impr'  : (neg A -> pos B -> #)
          -> (pos (A imp B) -> #).

impl'  : (pos A -> #)
          -> (neg B -> #)
          -> (neg (A imp B) -> #).

orr1'  : (pos A -> #)
         -> (pos (A or B) -> #).

orr2'  : (pos B -> #)
          -> (pos (A or B) -> #).

orl'   : (neg A -> #)
         -> (neg B -> #)
         -> (neg (A or B) -> #).

notr'  : (neg A -> #)
          -> (pos (not A) -> #).

notl'  : (pos A -> #)
          -> (neg (not A) -> #).

truer' : (pos (true) -> #).

falsel' : (neg (false) -> #).

forallr' : ({a:i} pos (A a) -> #)
            -> (pos (forall A) -> #).

foralll' : {T:i} (neg (A T) -> #)
            -> (neg (forall A) -> #).

existsr' : {T:i} (pos (A T) -> #)
            -> (pos (exists A) -> #).

existsl' : ({a:i} neg (A a) -> #)
            -> (neg (exists A) -> #).
```

The cut rule can be added in a similar style (see Appendix B.2). The representations is adequate and compositional, we skip the routine formulation of such a theorem.

We consider two examples of classical sequent derivations and their representation in Elf. The first is the law of excluded middle.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A \longrightarrow \neg(A), A, (A \vee \neg(A))}\,I}{\longrightarrow \neg(A), A, (A \vee \neg(A))}\,\neg R}{\longrightarrow A, (A \vee \neg(A))}\,\vee R_2}{\longrightarrow (A \vee \neg(A))}\,\vee R_1$$

Note how multiple conclusions are necessary so that both right rules for disjunction may be applied in succession. This derivation is represented by the term

```
([A:o] [p:pos (A or not A)]
   orr1' ([p1:pos A]
```

```
            orr2' ([p2:pos (not A)]
                     notr' ([n1:neg A] axiom' n1 p1)
                     p2)
            p)
   p)
:
{A:o} pos (A or not A) -> #.
```

The following example provides another illustration of the differences between intuitionistic and classical reasoning in the sequent calculus.

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\neg((\forall x.\, Ax)), Aa \longrightarrow \neg(Aa), Aa, (\forall x.\, Ax), (\exists x.\, \neg(Ax))}\,I}{\neg((\forall x.\, Ax)) \longrightarrow \neg(Aa), Aa, (\forall x.\, Ax), (\exists x.\, \neg(Ax))}\,\neg R}{\neg((\forall x.\, Ax)) \longrightarrow Aa, (\forall x.\, Ax), (\exists x.\, \neg(Ax))}\,\exists R}{\neg((\forall x.\, Ax)) \longrightarrow (\forall x.\, Ax), (\exists x.\, \neg(Ax))}\,\forall R^a}{\neg((\forall x.\, Ax)) \longrightarrow (\exists x.\, \neg(Ax))}\,\neg L
$$

It is represented by the term

```
([A:i -> o]
   [n:neg (not (forall [x] A x))]
   [p:pos (exists [x] not (A x))]
   notl' ([p1:pos (forall [x] A x)]
            forallr' ([a:i] [p2:pos (A a)]
                        existsr' a ([p3:pos (not (A a))]
                                      notr' ([n1:neg (A a)]
                                               axiom' n1 p2)
                                      p3)
                        p)
            p1)
   n)
:
{A:i -> o}
   neg (not (forall [x] A x))
   -> pos (exists [x] not (A x))
   -> #.
```

The admissibility of the cut rule for the cut-free calculus is once again the central lemma for cut elimination. There are now more cases, since there may be side formulas on the right-hand sides of sequents. However, due to the symmetry of the rules, the proof is even more systematic than in the intuitionistic case. It also follows by three nested structural inductions.

Theorem 7 (Classical Admissibility of Cut) *Let $\mathcal{D} :: (\Gamma \xrightarrow{d} p{:}A, \Delta)$ and $\mathcal{E} :: (\Gamma, n{:}A \xrightarrow{e} \Delta)$ be cut-free derivations in the classical sequent calculus G_3. Then there is a proof term f and a cut-free sequent derivation of $\mathcal{F} :: (\Gamma \xrightarrow{f} \Delta)$.*

Proof: By three nested structural inductions on A, d, and e. □

The notion of a *cross-cut* [Gal93], though without multiplicities, surfaces naturally in this proof: Since formulas are never discarded they must be eliminated explicitly from both premises of a cut in a "cross-cut" fashion before the essential cut reduction can take place. The implementation of the proof is by a type family

```
ca' : {A:o} (pos A -> #) -> (neg A -> #) -> # -> type.
```

that implements the relation between A, the derivation $\mathcal{D}$, the derivation $\mathcal{E}$ and the resulting derivation $\mathcal{F}$.

We only show the representation of one case in the proof of admissibility here; The complete proof representation including an informal version of each case may be found in Appendix A.2. The first three appeals to the induction hypothesis below are cross-cuts.

```
ca_imp' :
  ca' (A imp B) ([p] impr' (D1 p) p) ([n] impl' (E1 n) (E2 n) n) F
     <- ({p1:pos A} ca' (A imp B) ([p] impr' (D1 p) p) ([n] E1 n p1) (E1' p1))
     <- ({n2:neg B} ca' (A imp B) ([p] impr' (D1 p) p) ([n] E2 n n2) (E2' n2))
     <- ({n1:neg A} {p2:pos B}
            ca' (A imp B) ([p] D1 p n1 p2) ([n] impl' (E1 n) (E2 n) n)
            (D1' n1 p2))
     <- ({p2:pos B} ca' A ([p1] E1' p1) ([n1] D1' n1 p2) (F2 p2))
     <- ca' B ([p2] F2 p2) ([n2] E2' n2) F.
```

7 Conclusion

We have presented new proofs of cut elimination for intuitionistic and classical sequent calculi. The proof in the intuitionistic case is motivated by maintaining a close correspondence between proof search for natural deduction and sequent derivations. It is this proximity that permits a natural representation of the sequent calculus in LF. Furthermore, we show how the proof of cut elimination can be implemented in Elf, although the fact that this implementation models the informal argument is still partly an informal property, just like the adequacy of the LF encoding of derivations. The proof representation is extremely concise and much shorter than an informal proof of the same argument (if all the cases were given, of course). In the two appendices below we give the details of the proofs which were obtained via a program that translates the internal form of Elf declarations to LaTeX source. This "informalized" version of the proof representation can be inspected for correctness like ordinary informal mathematical proofs.

In order to give the reader a feel for the efficiency of LF representation techniques and the Elf implementation I give a brief summary of the development history of the work described here. For a long time I had thought that a representation of a cut elimination proof in LF would be prohibitively complex—if I were to undertake it, I would use a system like Coq in which tactics can be used to automate long chains of trivial reasoning steps; Elf does not provide this sort of automation technique. The basic difficulty can be traced to the representation of sequents and sequent derivations themselves. It is often informally stated that a sequent calculus may be viewed as a calculus of proof search for natural deduction. Once I took this remark literally, it took me a day to write out the representation of sequent derivations and the proof of admissibility for the intuitionistic calculus in Elf (fortunately, the first approach I tried worked). It took me another half day to write out and debug admissibility of cut for the classical sequent calculus. From these

I reconstructed and checked the informal arguments by hand, which took me another couple of days. The combined sources of representations and proofs for intuitionistic and classical case are 739 lines of Elf code (with only a few comments, but without white-space compression). Type-checking the proof takes about 2 secs on a Dec Alpha. Writing the program which generates the informal version of the proof from its formalization took about 2 weeks, with several false starts (this involved programming in Elf, Emacs Lisp, and LaTeX).

Another analysis of cut elimination for a small propositional fragment of classical logic is given by Matthews [Mat94] in FS_0. His proof is traditional—sequents are represented as lists, and termination is proved by induction on a standard complexity measure. It has not yet been implemented, but it is clear from the sketched development that it would require much time and effort just to prove basic properties of sequent derivations, their lengths, *etc.* For the predicate calculus this overhead would be even higher, since a theory binding would have to be developed first.

Once the structural proof of admissibility has been found and implemented, it is natural to ask if it can also be encoded in stronger frameworks such as Coq [DFH+93] so that structural inductions are made explicit and the proof is fully formally verified. There are several aspects of our proof which make this difficult. The first is the use of higher-order abstract syntax, which is not available in a similarly straightforward fashion in other candidate environments. Thus one either has to try ideas from [DH94] (which we have not attempted) or use an encoding such as de Bruijn indices and explicitly represent contexts. In either case one has to prove a number of auxiliary lemmas regarding substitutions which are not needed in our representation. The second difficulty arises from the non-deterministic nature of the cut elimination algorithm contained in the proof. Making it deterministic in the form of a primitive recursion (which would be required for a functional framework) would lead to an explosion in the number of cases that would have to be considered. It appears the only way to avoid at least some of this combinatorial explosion is to introduce termination measures after all, which requires a new sequence of lemmas regarding sizes of formulas and derivations. We conclude that a similarly elegant representation of cut elimination in other systems is a non-trivial challenge which, we hope, others will take up.

In future work we plan to verify mechanically that the given signatures indeed implement proofs. The prototype implementation of the schema-checker sketched in [Roh94] currently accepts them, but the (meta-meta-)theoretical analysis of schema-checker itself is not yet complete. In other future work we plan to reexamine the connection between normalization and cut elimination (see, for example, [Zuc74]) in the same framework. Another direction is to study cut elimination in a formulation as a higher-order rewrite system along the lines of Nipkow [Nip91], but using dependent types. We first note that our system of rules is terminating (note that we cannot permute adjacent cuts!). Assuming the completeness of a critical pair criterion for the dependently typed calculus, the system is confluent modulo Kleene's permutations of adjacent inference rules in the cut-free system. This means that our cut conversions do not identify intuitively unrelated sequent derivations, which has been a problem in other systems as noted by Lafont (see [Gal93]).

Finally, we have a formulation of a sequent calculus for classical linear logic based on the ideas in this paper. There is a combinatorial explosion of cases (and we have not checked all of them), but we conjecture that a structural proof of cut-elimination is still possible. To represent such a proof concisely would require a *linear* framework, which is the subject of other current research.

A Detailed Admissibility Proofs for Cut

In this appendix we give the details of the admissibility of cut for intuitionistic and classical sequent calculi. For each case in the two proofs we show the formalization as an Elf declaration, followed by an automatically generated informal rendering of the case. In order to make the informal proof cases more readable we omit explicit proof terms. This means that appeals to weakening and contraction lemmas are not visible (see the implicit contraction in the first case, for example). We apologize for the unintuitive naming of variables. Variable names are chosen by Elf during type reconstruction and printing, and our naming heuristics are currently too simplistic.

Substitution for individual and propositional parameters arises in the admissibility proof for cut in a few cases. When a formula or derivation may depend on a variable x or parameter a we indicate this, for example, by writing Ax or $\mathcal{D}a$. Instead of $[t/x]A$ or $[t/a]\mathcal{D}$ we then write At or $\mathcal{D}t$ for the result of a substitution t for x or a. This is more perspicuous and also closer to the Elf implementation and therefore much easier to generate.

A.1 Intuitionistic Calculus

A case in the proof of admissibility of cut in the intuitionistic sequent calculus is represented as a transformation

$$\begin{array}{c}\mathcal{D}\\ \Gamma \longrightarrow A\end{array} \quad \otimes \quad \begin{array}{c}\mathcal{E}\\ \Gamma, A \longrightarrow C\end{array} \quad \Longrightarrow \quad \begin{array}{c}\mathcal{F}\\ \Gamma \longrightarrow C\end{array}$$

where $\mathcal{F}$ may refer to derivations constructed by appeals to the induction hypothesis. These are given below the first line (which identifies the case under consideration) in an appropriate order. In all cases the decreasing structural component should be apparent; it would have been awkward to include this information, since proof terms have been omitted. In the remarks we loosely refer to the *principal formula* or *side formula* when properly speaking we mean the *principal formula occurrence* or *side formula occurrence*. We could be pedantic using labelled hypotheses, but only at a heavy cost in legibility.

The relation between $\mathcal{D}$, $\mathcal{E}$, and $\mathcal{F}$ is implemented as a type family

```
ca : {A:o} conc A -> (hyp A -> conc C) -> conc C -> type.
```

The cases below are divided into the four classes mentioned in the proof of Theorem 5. In analogy to other published proofs we call them *initial conversions* (one of $\mathcal{D}$ or $\mathcal{E}$ is initial with the cut formula as a principal formula), *essential conversions* (cut formula is principal in $\mathcal{D}$ and $\mathcal{E}$), *left commutative conversions* (cut formula is side formula in $\mathcal{D}$), and *right commutative conversions* (cut formula is side formula in $\mathcal{E}$).

Initial Conversions. These are the cases in the proof where either $\mathcal{D}$ or $\mathcal{E}$ is an initial sequent with principal formula being the cut formula A.

```
ca_axiom_l : ca A (axiom H) E (E H).
```

$$\dfrac{}{\Gamma, A_1 \longrightarrow A_1}I \quad \otimes \quad \begin{array}{c}N\\ \Gamma, A_1, A_1 \longrightarrow A\end{array} \quad \Longrightarrow \quad \begin{array}{c}N\\ \Gamma, A_1 \longrightarrow A\end{array}$$

```
ca_axiom_r : ca A D ([h:hyp A] axiom h) D.
```

$$\begin{array}{c} N \\ \Gamma \longrightarrow A \end{array} \quad \otimes \quad \dfrac{}{\Gamma, A \longrightarrow A}\, I \quad \Longrightarrow \quad \begin{array}{c} N \\ \Gamma \longrightarrow A \end{array}$$

Essential Conversions. These are the steps in the proof where the cut formula is the principal formula of the last inference in both $\mathcal{D}$ and $\mathcal{E}$.

```
ca_and1 : ca (A1 and A2) (andr D1 D2)
          ([h:hyp (A1 and A2)] andl1 (E1 h) h) F
          <- ({h1:hyp A1}
                ca (A1 and A2) (andr D1 D2)
                ([h:hyp (A1 and A2)] E1 h h1) (E1' h1))
          <- ca A1 D1 E1' F.
```

$$\dfrac{\begin{array}{c} N \\ \Gamma \longrightarrow A_1 \end{array} \qquad \begin{array}{c} N_3 \\ \Gamma \longrightarrow A_2 \end{array}}{\Gamma \longrightarrow (A_1 \wedge A_2)}\, \wedge R \quad \otimes \quad \dfrac{\begin{array}{c} N_4 \\ \Gamma, (A_1 \wedge A_2), A_1 \longrightarrow A \end{array}}{\Gamma, (A_1 \wedge A_2) \longrightarrow A}\, \wedge L_1 \quad \Longrightarrow \quad \begin{array}{c} N_2 \\ \Gamma \longrightarrow A \end{array}$$

$$\dfrac{\begin{array}{c} N \\ \Gamma, A_1 \longrightarrow A_1 \end{array} \qquad \begin{array}{c} N_3 \\ \Gamma, A_1 \longrightarrow A_2 \end{array}}{\Gamma, A_1 \longrightarrow (A_1 \wedge A_2)}\, \wedge R \quad \otimes \quad \begin{array}{c} N_4 \\ \Gamma, A_1, (A_1 \wedge A_2) \longrightarrow A \end{array} \quad \Longrightarrow \quad \begin{array}{c} N_1 \\ \Gamma, A_1 \longrightarrow A \end{array}$$

$$\begin{array}{c} N \\ \Gamma \longrightarrow A_1 \end{array} \quad \otimes \quad \begin{array}{c} N_1 \\ \Gamma, A_1 \longrightarrow A \end{array} \quad \Longrightarrow \quad \begin{array}{c} N_2 \\ \Gamma \longrightarrow A \end{array}$$

```
ca_and2 : ca (A1 and A2) (andr D1 D2)
          ([h:hyp (A1 and A2)] andl2 (E2 h) h) F
          <- ({h2:hyp A2}
                ca (A1 and A2) (andr D1 D2)
                ([h:hyp (A1 and A2)] E2 h h2) (E2' h2))
          <- ca A2 D2 E2' F.
```

$$\dfrac{\begin{array}{c} N_3 \\ \Gamma \longrightarrow A_2 \end{array} \qquad \begin{array}{c} N \\ \Gamma \longrightarrow A_1 \end{array}}{\Gamma \longrightarrow (A_2 \wedge A_1)}\, \wedge R \quad \otimes \quad \dfrac{\begin{array}{c} N_4 \\ \Gamma, (A_2 \wedge A_1), A_1 \longrightarrow A \end{array}}{\Gamma, (A_2 \wedge A_1) \longrightarrow A}\, \wedge L_2 \quad \Longrightarrow \quad \begin{array}{c} N_2 \\ \Gamma \longrightarrow A \end{array}$$

$$\dfrac{\begin{array}{c} N_3 \\ \Gamma, A_1 \longrightarrow A_2 \end{array} \qquad \begin{array}{c} N \\ \Gamma, A_1 \longrightarrow A_1 \end{array}}{\Gamma, A_1 \longrightarrow (A_2 \wedge A_1)}\, \wedge R \quad \otimes \quad \begin{array}{c} N_4 \\ \Gamma, A_1, (A_2 \wedge A_1) \longrightarrow A \end{array} \quad \Longrightarrow \quad \begin{array}{c} N_1 \\ \Gamma, A_1 \longrightarrow A \end{array}$$

$$\begin{array}{c} N \\ \Gamma \longrightarrow A_1 \end{array} \quad \otimes \quad \begin{array}{c} N_1 \\ \Gamma, A_1 \longrightarrow A \end{array} \quad \Longrightarrow \quad \begin{array}{c} N_2 \\ \Gamma \longrightarrow A \end{array}$$

```
ca_imp  : ca (A1 imp A2) (impr D2)
           ([h:hyp (A1 imp A2)] impl (E1 h) (E2 h) h) F
           <- ca (A1 imp A2) (impr D2) E1 E1'
           <- ({h2:hyp A2}
                 ca (A1 imp A2) (impr D2)
                 ([h:hyp (A1 imp A2)] E2 h h2) (E2' h2))
           <- ca A1 E1' D2 D2'
           <- ca A2 D2' E2' F.
```

$$\dfrac{\overset{N_4}{\Gamma, A_2 \longrightarrow A_1}}{\Gamma \longrightarrow (A_2 \supset A_1)}{\supset}R \quad \bigotimes \quad \dfrac{\overset{N_6}{\Gamma, (A_2 \supset A_1) \longrightarrow A_2} \qquad \overset{N_5}{\Gamma, (A_2 \supset A_1), A_1 \longrightarrow A}}{\Gamma, (A_2 \supset A_1) \longrightarrow A}{\supset}L \quad \Longrightarrow \quad \overset{N_2}{\Gamma \longrightarrow A}$$

$$\dfrac{\overset{N_4}{\Gamma, A_2 \longrightarrow A_1}}{\Gamma \longrightarrow (A_2 \supset A_1)}{\supset}R \quad \bigotimes \quad \overset{N_6}{\Gamma, (A_2 \supset A_1) \longrightarrow A_2} \quad \Longrightarrow \quad \overset{N_3}{\Gamma \longrightarrow A_2}$$

$$\dfrac{\overset{N_4}{\Gamma, A_1, A_2 \longrightarrow A_1}}{\Gamma, A_1 \longrightarrow (A_2 \supset A_1)}{\supset}R \quad \bigotimes \quad \overset{N_5}{\Gamma, A_1, (A_2 \supset A_1) \longrightarrow A} \quad \Longrightarrow \quad \overset{N_1}{\Gamma, A_1 \longrightarrow A}$$

$$\overset{N_3}{\Gamma \longrightarrow A_2} \quad \bigotimes \quad \overset{N_4}{\Gamma, A_2 \longrightarrow A_1} \quad \Longrightarrow \quad \overset{N}{\Gamma \longrightarrow A_1}$$

$$\overset{N}{\Gamma \longrightarrow A_1} \quad \bigotimes \quad \overset{N_1}{\Gamma, A_1 \longrightarrow A} \quad \Longrightarrow \quad \overset{N_2}{\Gamma \longrightarrow A}$$

```
ca_or1  : ca (A1 or A2) (orr1 D1)
           ([h:hyp (A1 or A2)] orl (E1 h) (E2 h) h) F
           <- ({h1:hyp A1}
                 ca (A1 or A2) (orr1 D1)
                 ([h:hyp (A1 or A2)] E1 h h1) (E1' h1))
           <- ca A1 D1 E1' F.
```

$$\dfrac{\overset{N}{\Gamma \longrightarrow A_1}}{\Gamma \longrightarrow (A_1 \vee A_2)}\vee R_1 \quad \bigotimes \quad \dfrac{\overset{N_3}{\Gamma, (A_1 \vee A_2), A_1 \longrightarrow A} \qquad \overset{N_4}{\Gamma, (A_1 \vee A_2), A_2 \longrightarrow A}}{\Gamma, (A_1 \vee A_2) \longrightarrow A}\vee L \quad \Longrightarrow \quad \overset{N_2}{\Gamma \longrightarrow A}$$

$$\dfrac{\overset{N}{\Gamma, A_1 \longrightarrow A_1}}{\Gamma, A_1 \longrightarrow (A_1 \vee A_2)}\vee R_1 \quad \bigotimes \quad \overset{N_3}{\Gamma, A_1, (A_1 \vee A_2) \longrightarrow A} \quad \Longrightarrow \quad \overset{N_1}{\Gamma, A_1 \longrightarrow A}$$

$$\overset{N}{\Gamma \longrightarrow A_1} \quad \bigotimes \quad \overset{N_1}{\Gamma, A_1 \longrightarrow A} \quad \Longrightarrow \quad \overset{N_2}{\Gamma \longrightarrow A}$$

```
ca_or2  : ca (A1 or A2) (orr2 D2)
           ([h:hyp (A1 or A2)] orl (E1 h) (E2 h) h) F
           <- ({h2:hyp A2}
                 ca (A1 or A2) (orr2 D2)
                 ([h:hyp (A1 or A2)] E2 h h2) (E2' h2))
           <- ca A2 D2 E2' F.
```

$$\dfrac{\overset{N}{\Gamma \longrightarrow A_1}}{\Gamma \longrightarrow (A_2 \vee A_1)}\vee R_2 \quad \bigotimes \quad \dfrac{\overset{N_4}{\Gamma, (A_2 \vee A_1), A_2 \longrightarrow A} \qquad \overset{N_3}{\Gamma, (A_2 \vee A_1), A_1 \longrightarrow A}}{\Gamma, (A_2 \vee A_1) \longrightarrow A}\vee L \quad \Longrightarrow \quad \overset{N_2}{\Gamma \longrightarrow A}$$

$$\dfrac{\overset{N}{\Gamma, A_1 \longrightarrow A_1}}{\Gamma, A_1 \longrightarrow (A_2 \vee A_1)}\vee R_2 \quad \bigotimes \quad \overset{N_3}{\Gamma, A_1, (A_2 \vee A_1) \longrightarrow A} \quad \Longrightarrow \quad \overset{N_1}{\Gamma, A_1 \longrightarrow A}$$

$$\overset{N}{\Gamma \longrightarrow A_1} \quad \bigotimes \quad \overset{N_1}{\Gamma, A_1 \longrightarrow A} \quad \Longrightarrow \quad \overset{N_2}{\Gamma \longrightarrow A}$$

```
ca_not  : ca (not A1) (notr D1)
           ([h:hyp (not A1)] notl (E1 h) h) (F2 C)
           <- ca (not A1) (notr D1) E1 F1
           <- ({p:o} ca A1 F1 ([h1:hyp A1] D1 p h1) (F2 p)).
```

$$\cfrac{\begin{array}{c}N_1 p_1\\ \Gamma, A \longrightarrow p_1\end{array}}{\Gamma \longrightarrow \neg(A)}\neg R^{p_1} \quad \bigotimes \quad \cfrac{\begin{array}{c}N_3\\ \Gamma, \neg(A) \longrightarrow A\end{array}}{\Gamma, \neg(A) \longrightarrow A_1}\neg L \quad \Longrightarrow \quad \begin{array}{c}N_2 A_1\\ \Gamma \longrightarrow A_1\end{array}$$

$$\cfrac{\begin{array}{c}N_1 p_1\\ \Gamma, A \longrightarrow p_1\end{array}}{\Gamma \longrightarrow \neg(A)}\neg R^{p_1} \quad \bigotimes \quad \begin{array}{c}N_3\\ \Gamma, \neg(A) \longrightarrow A\end{array} \quad \Longrightarrow \quad \begin{array}{c}N\\ \Gamma \longrightarrow A\end{array}$$

$$\begin{array}{c}N\\ \Gamma \longrightarrow A\end{array} \quad \bigotimes \quad \begin{array}{c}N_1 p\\ \Gamma, A \longrightarrow p\end{array} \quad \Longrightarrow \quad \begin{array}{c}N_2 p\\ \Gamma \longrightarrow p\end{array}$$

```
ca_forall : ca (forall A1) (forallr D1)
            ([h:hyp (forall A1)] foralll T (E1 h) h) F
            <- ({h2:hyp (A1 T)}
                  ca (forall A1) (forallr D1)
                  ([h:hyp (forall A1)] E1 h h2) (E1' h2))
            <- ca (A1 T) (D1 T) E1' F.
```

$$\cfrac{\begin{array}{c}N a\\ \Gamma \longrightarrow A_1 a\end{array}}{\Gamma \longrightarrow (\forall x.\, A_1 x)}\forall R^a \quad \bigotimes \quad \cfrac{\begin{array}{c}N_3\\ \Gamma, (\forall x.\, A_1 x), A_1 t \longrightarrow A\end{array}}{\Gamma, (\forall x.\, A_1 x) \longrightarrow A}\forall L \quad \Longrightarrow \quad \begin{array}{c}N_2\\ \Gamma \longrightarrow A\end{array}$$

$$\cfrac{\begin{array}{c}N a\\ \Gamma, A_1 t \longrightarrow A_1 a\end{array}}{\Gamma, A_1 t \longrightarrow (\forall x.\, A_1 x)}\forall R^a \quad \bigotimes \quad \begin{array}{c}N_3\\ \Gamma, A_1 t, (\forall x.\, A_1 x) \longrightarrow A\end{array} \quad \Longrightarrow \quad \begin{array}{c}N_1\\ \Gamma, A_1 t \longrightarrow A\end{array}$$

$$\begin{array}{c}N t\\ \Gamma \longrightarrow A_1 t\end{array} \quad \bigotimes \quad \begin{array}{c}N_1\\ \Gamma, A_1 t \longrightarrow A\end{array} \quad \Longrightarrow \quad \begin{array}{c}N_2\\ \Gamma \longrightarrow A\end{array}$$

```
ca_exists : ca (exists A1) (existsr T D1)
            ([h:hyp (exists A1)] existsl (E1 h) h) F
            <- ({a:i} {h1:hyp (A1 a)}
                  ca (exists A1) (existsr T D1)
                  ([h:hyp (exists A1)] E1 h a h1) (E1' a h1))
            <- ca (A1 T) D1 (E1' T) F.
```

$$\dfrac{\begin{array}{c}N\\ \Gamma \longrightarrow A_1 t\end{array}}{\Gamma \longrightarrow (\exists x.\, A_1 x)}\exists R \quad \otimes \quad \dfrac{\begin{array}{c}N_3 a_2\\ \Gamma, (\exists x.\, A_1 x), A_1 a_2 \longrightarrow A\end{array}}{\Gamma, (\exists x.\, A_1 x) \longrightarrow A}\exists L^{a_2} \quad \Longrightarrow \quad \begin{array}{c}N_2\\ \Gamma \longrightarrow A\end{array}$$

$$\dfrac{\begin{array}{c}N\\ \Gamma, A_1 a \longrightarrow A_1 t\end{array}}{\Gamma, A_1 a \longrightarrow (\exists x.\, A_1 x)}\exists R \quad \otimes \quad \begin{array}{c}N_3 a\\ \Gamma, A_1 a, (\exists x.\, A_1 x) \longrightarrow A\end{array} \quad \Longrightarrow \quad \begin{array}{c}N_1 a\\ \Gamma, A_1 a \longrightarrow A\end{array}$$

$$\begin{array}{c}N\\ \Gamma \longrightarrow A_1 t\end{array} \quad \otimes \quad \begin{array}{c}N_1 t\\ \Gamma, A_1 t \longrightarrow A\end{array} \quad \Longrightarrow \quad \begin{array}{c}N_2\\ \Gamma \longrightarrow A\end{array}$$

Left Commutative Conversions. In these cases the cut formula is a side formula in the deduction $\mathcal{D} :: (\Gamma \longrightarrow A)$. Note that the deduction $\mathcal{D}$ must end in a left rule, since otherwise A would be its principal formula.

```
cal_andl1  : ca A (andl1 D1 H) E (andl1 D1' H)
              <- {h1:hyp B1} ca A (D1 h1) E (D1' h1).
```

$$\dfrac{\begin{array}{c}N\\ \Gamma, (A \wedge A_3), A \longrightarrow A_2\end{array}}{\Gamma, (A \wedge A_3) \longrightarrow A_2}\wedge L_1 \quad \otimes \quad \begin{array}{c}N_1\\ \Gamma, (A \wedge A_3), A_2 \longrightarrow A_1\end{array} \quad \Longrightarrow \quad \dfrac{\begin{array}{c}N_2\\ \Gamma, (A \wedge A_3), A \longrightarrow A_1\end{array}}{\Gamma, (A \wedge A_3) \longrightarrow A_1}\wedge L_1$$

$$\begin{array}{c}N\\ \Gamma, (A \wedge A_3), A \longrightarrow A_2\end{array} \quad \otimes \quad \begin{array}{c}N_1\\ \Gamma, (A \wedge A_3), A, A_2 \longrightarrow A_1\end{array} \quad \Longrightarrow \quad \begin{array}{c}N_2\\ \Gamma, (A \wedge A_3), A \longrightarrow A_1\end{array}$$

```
cal_andl2  : ca A (andl2 D2 H) E (andl2 D2' H)
              <- {h2:hyp B2} ca A (D2 h2) E (D2' h2).
```

$$\dfrac{\begin{array}{c}N\\ \Gamma, (A_3 \wedge A), A \longrightarrow A_2\end{array}}{\Gamma, (A_3 \wedge A) \longrightarrow A_2}\wedge L_2 \quad \otimes \quad \begin{array}{c}N_1\\ \Gamma, (A_3 \wedge A), A_2 \longrightarrow A_1\end{array} \quad \Longrightarrow \quad \dfrac{\begin{array}{c}N_2\\ \Gamma, (A_3 \wedge A), A \longrightarrow A_1\end{array}}{\Gamma, (A_3 \wedge A) \longrightarrow A_1}\wedge L_2$$

$$\begin{array}{c}N\\ \Gamma, (A_3 \wedge A), A \longrightarrow A_2\end{array} \quad \otimes \quad \begin{array}{c}N_1\\ \Gamma, (A_3 \wedge A), A, A_2 \longrightarrow A_1\end{array} \quad \Longrightarrow \quad \begin{array}{c}N_2\\ \Gamma, (A_3 \wedge A), A \longrightarrow A_1\end{array}$$

```
cal_impl   : ca A (impl D1 D2 H) E (impl D1 D2' H)
              <- ({h2:hyp B2} ca A (D2 h2) E (D2' h2)).
```

$$\cfrac{\overset{N_3}{\Gamma,(A_3\supset A)\longrightarrow A_3} \qquad \overset{N}{\Gamma,(A_3\supset A),A\longrightarrow A_2}}{\Gamma,(A_3\supset A)\longrightarrow A_2}\,{\supset}L \quad \otimes \quad \overset{N_1}{\Gamma,(A_3\supset A),A_2\longrightarrow A_1}$$

$$\Longrightarrow \quad \cfrac{\overset{N_3}{\Gamma,(A_3\supset A)\longrightarrow A_3} \qquad \overset{N_2}{\Gamma,(A_3\supset A),A\longrightarrow A_1}}{\Gamma,(A_3\supset A)\longrightarrow A_1}\,{\supset}L$$

$$\overset{N}{\Gamma,(A_3\supset A),A\longrightarrow A_2} \quad \otimes \quad \overset{N_1}{\Gamma,(A_3\supset A),A,A_2\longrightarrow A_1} \quad \Longrightarrow \quad \overset{N_2}{\Gamma,(A_3\supset A),A\longrightarrow A_1}$$

```
cal_orl    : ca A (orl D1 D2 H) E (orl D1' D2' H)
              <- ({h1:hyp B1} ca A (D1 h1) E (D1' h1))
              <- ({h2:hyp B2} ca A (D2 h2) E (D2' h2)).
```

$$\cfrac{\overset{N_3}{\Gamma,(A_3\vee A),A_3\longrightarrow A_2} \qquad \overset{N}{\Gamma,(A_3\vee A),A\longrightarrow A_2}}{\Gamma,(A_3\vee A)\longrightarrow A_2}\,{\vee}L \quad \otimes \quad \overset{N_1}{\Gamma,(A_3\vee A),A_2\longrightarrow A_1}$$

$$\Longrightarrow \quad \cfrac{\overset{N_4}{\Gamma,(A_3\vee A),A_3\longrightarrow A_1} \qquad \overset{N_2}{\Gamma,(A_3\vee A),A\longrightarrow A_1}}{\Gamma,(A_3\vee A)\longrightarrow A_1}\,{\vee}L$$

$$\overset{N_3}{\Gamma,(A_3\vee A),A_3\longrightarrow A_2} \quad \otimes \quad \overset{N_1}{\Gamma,(A_3\vee A),A_3,A_2\longrightarrow A_1} \quad \Longrightarrow \quad \overset{N_4}{\Gamma,(A_3\vee A),A_3\longrightarrow A_1}$$

$$\overset{N}{\Gamma,(A_3\vee A),A\longrightarrow A_2} \quad \otimes \quad \overset{N_1}{\Gamma,(A_3\vee A),A,A_2\longrightarrow A_1} \quad \Longrightarrow \quad \overset{N_2}{\Gamma,(A_3\vee A),A\longrightarrow A_1}$$

```
cal_notl : ca A (notl D1 H) E (notl D1 H).
```

$$\cfrac{\overset{N}{\Gamma,\neg(A_2)\longrightarrow A_2}}{\Gamma,\neg(A_2)\longrightarrow A_1}\,\neg L \quad \otimes \quad \overset{N_1}{\Gamma,\neg(A_2),A_1\longrightarrow A} \quad \Longrightarrow \quad \cfrac{\overset{N}{\Gamma,\neg(A_2)\longrightarrow A_2}}{\Gamma,\neg(A_2)\longrightarrow A}\,\neg L$$

```
cal_falsel  : ca A (falsel H) E (falsel H).
```

$$\dfrac{}{\Gamma, \bot \longrightarrow A_1}\bot L \quad \otimes \quad \begin{array}{c} N \\ \Gamma, \bot, A_1 \longrightarrow A \end{array} \quad \Longrightarrow \quad \dfrac{}{\Gamma, \bot \longrightarrow A}\bot L$$

```
cal_foralll : ca A (foralll T D1 H) E (foralll T D1' H)
               <- ({h} ca A (D1 h) E (D1' h)).
```

$$\dfrac{\begin{array}{c} N \\ \Gamma, (\forall x.\, Ax), At \longrightarrow A_2 \end{array}}{\Gamma, (\forall x.\, Ax) \longrightarrow A_2}\forall L \quad \otimes \quad \begin{array}{c} N_1 \\ \Gamma, (\forall x.\, Ax), A_2 \longrightarrow A_1 \end{array} \quad \Longrightarrow \quad \dfrac{\begin{array}{c} N_2 \\ \Gamma, (\forall x.\, Ax), At \longrightarrow A_1 \end{array}}{\Gamma, (\forall x.\, Ax) \longrightarrow A_1}\forall L$$

$$\begin{array}{c} N \\ \Gamma, (\forall x.\, Ax), At \longrightarrow A_2 \end{array} \quad \otimes \quad \begin{array}{c} N_1 \\ \Gamma, (\forall x.\, Ax), At, A_2 \longrightarrow A_1 \end{array} \quad \Longrightarrow \quad \begin{array}{c} N_2 \\ \Gamma, (\forall x.\, Ax), At \longrightarrow A_1 \end{array}$$

```
cal_existsl : ca A (existsl D1 H) E (existsl D1' H)
               <- ({a:i} {h:hyp (B1 a)}
                     ca A (D1 a h) E (D1' a h)).
```

$$\dfrac{\begin{array}{c} N a_1 \\ \Gamma, (\exists x.\, Ax), Aa_1 \longrightarrow A_2 \end{array}}{\Gamma, (\exists x.\, Ax) \longrightarrow A_2}\exists L^{a_1} \quad \otimes \quad \begin{array}{c} N_1 \\ \Gamma, (\exists x.\, Ax), A_2 \longrightarrow A_1 \end{array} \quad \Longrightarrow \quad \dfrac{\begin{array}{c} N_2 a_1 \\ \Gamma, (\exists x.\, Ax), Aa_1 \longrightarrow A_1 \end{array}}{\Gamma, (\exists x.\, Ax) \longrightarrow A_1}\exists L^{a_1}$$

$$\begin{array}{c} N a \\ \Gamma, (\exists x.\, Ax), Aa \longrightarrow A_2 \end{array} \quad \otimes \quad \begin{array}{c} N_1 \\ \Gamma, (\exists x.\, Ax), Aa, A_2 \longrightarrow A_1 \end{array} \quad \Longrightarrow \quad \begin{array}{c} N_2 a \\ \Gamma, (\exists x.\, Ax), Aa \longrightarrow A_1 \end{array}$$

Right Commutative Conversions. In these cases, the formula A in $\mathcal{E} :: (\Gamma, A \longrightarrow C)$ is a side formula of the last inference in $\mathcal{E}$. These cases are not necessarily exclusive with the left commutative conversions above. There are three classes of subcases: The last inference in $\mathcal{E}$ may be an axiom, a left rule, or a right rule.

```
car_axiom : ca A D ([h:hyp A] axiom H1) (axiom H1).
```

$$\begin{array}{c} N \\ \Gamma, A \longrightarrow A_1 \end{array} \quad \otimes \quad \dfrac{}{\Gamma, A, A_1 \longrightarrow A}I \quad \Longrightarrow \quad \dfrac{}{\Gamma, A \longrightarrow A}I$$

```
car_andr : ca A D ([h:hyp A] andr (E1 h) (E2 h)) (andr E1' E2')
            <- ca A D E1 E1'
            <- ca A D E2 E2'.
```

$$\overset{N}{\Gamma \longrightarrow A_1} \quad \otimes \quad \dfrac{\overset{N_3}{\Gamma, A_1 \longrightarrow A_2} \qquad \overset{N_1}{\Gamma, A_1 \longrightarrow A}}{\Gamma, A_1 \longrightarrow (A_2 \wedge A)}\wedge R \quad \Longrightarrow \quad \dfrac{\overset{N_4}{\Gamma \longrightarrow A_2} \qquad \overset{N_2}{\Gamma \longrightarrow A}}{\Gamma \longrightarrow (A_2 \wedge A)}\wedge R$$

$$\overset{N}{\Gamma \longrightarrow A_1} \quad \otimes \quad \overset{N_3}{\Gamma, A_1 \longrightarrow A_2} \quad \Longrightarrow \quad \overset{N_4}{\Gamma \longrightarrow A_2}$$

$$\overset{N}{\Gamma \longrightarrow A_1} \quad \otimes \quad \overset{N_1}{\Gamma, A_1 \longrightarrow A} \quad \Longrightarrow \quad \overset{N_2}{\Gamma \longrightarrow A}$$

```
car_andl1: ca A D ([h:hyp A] andl1 (E1 h) H) (andl1 E1' H)
           <- ({h1:hyp B1} ca A D ([h:hyp A] E1 h h1) (E1' h1)).
```

$$\overset{N}{\Gamma, (A \wedge A_3) \longrightarrow A_2} \quad \otimes \quad \dfrac{\overset{N_1}{\Gamma, (A \wedge A_3), A_2, A \longrightarrow A_1}}{\Gamma, (A \wedge A_3), A_2 \longrightarrow A_1}\wedge L_1 \quad \Longrightarrow \quad \dfrac{\overset{N_2}{\Gamma, (A \wedge A_3), A \longrightarrow A_1}}{\Gamma, (A \wedge A_3) \longrightarrow A_1}\wedge L_1$$

$$\overset{N}{\Gamma, (A \wedge A_3), A \longrightarrow A_2} \quad \otimes \quad \overset{N_1}{\Gamma, (A \wedge A_3), A, A_2 \longrightarrow A_1} \quad \Longrightarrow \quad \overset{N_2}{\Gamma, (A \wedge A_3), A \longrightarrow A_1}$$

```
car_andl2: ca A D ([h:hyp A] andl2 (E2 h) H) (andl2 E2' H)
           <- ({h2:hyp B2} ca A D ([h:hyp A] E2 h h2) (E2' h2)).
```

$$\overset{N}{\Gamma, (A_3 \wedge A) \longrightarrow A_2} \quad \otimes \quad \dfrac{\overset{N_1}{\Gamma, (A_3 \wedge A), A_2, A \longrightarrow A_1}}{\Gamma, (A_3 \wedge A), A_2 \longrightarrow A_1}\wedge L_2 \quad \Longrightarrow \quad \dfrac{\overset{N_2}{\Gamma, (A_3 \wedge A), A \longrightarrow A_1}}{\Gamma, (A_3 \wedge A) \longrightarrow A_1}\wedge L_2$$

$$\overset{N}{\Gamma, (A_3 \wedge A), A \longrightarrow A_2} \quad \otimes \quad \overset{N_1}{\Gamma, (A_3 \wedge A), A, A_2 \longrightarrow A_1} \quad \Longrightarrow \quad \overset{N_2}{\Gamma, (A_3 \wedge A), A \longrightarrow A_1}$$

```
car_impr : ca A D ([h:hyp A] impr (E2 h)) (impr E2')
           <- ({h1:hyp B1} ca A D ([h:hyp A] E2 h h1) (E2' h1)).
```

$$\overset{N}{\Gamma \longrightarrow A_2} \quad \otimes \quad \dfrac{\overset{N_1}{\Gamma, A_2, A \longrightarrow A_1}}{\Gamma, A_2 \longrightarrow (A \supset A_1)}\supset R \quad \Longrightarrow \quad \dfrac{\overset{N_2}{\Gamma, A \longrightarrow A_1}}{\Gamma \longrightarrow (A \supset A_1)}\supset R$$

$$\overset{N}{\Gamma, A \longrightarrow A_2} \quad \otimes \quad \overset{N_1}{\Gamma, A, A_2 \longrightarrow A_1} \quad \Longrightarrow \quad \overset{N_2}{\Gamma, A \longrightarrow A_1}$$

```
car_impl : ca A D ([h:hyp A] impl (E1 h) (E2 h) H) (impl E1' E2' H)
            <- ca A D E1 E1'
            <- ({h2:hyp B2} ca A D ([h:hyp A] E2 h h2) (E2' h2)).
```

$$\overset{N}{\Gamma, (A_3 \supset A) \longrightarrow A_2} \quad \otimes \quad \dfrac{\overset{N_3}{\Gamma, (A_3 \supset A), A_2 \longrightarrow A_3} \qquad \overset{N_1}{\Gamma, (A_3 \supset A), A_2, A \longrightarrow A_1}}{\Gamma, (A_3 \supset A), A_2 \longrightarrow A_1} \supset L$$

$$\Longrightarrow \quad \dfrac{\overset{N_4}{\Gamma, (A_3 \supset A) \longrightarrow A_3} \qquad \overset{N_2}{\Gamma, (A_3 \supset A), A \longrightarrow A_1}}{\Gamma, (A_3 \supset A) \longrightarrow A_1} \supset L$$

$$\overset{N}{\Gamma, (A_3 \supset A) \longrightarrow A_2} \quad \otimes \quad \overset{N_3}{\Gamma, (A_3 \supset A), A_2 \longrightarrow A_3} \quad \Longrightarrow \quad \overset{N_4}{\Gamma, (A_3 \supset A) \longrightarrow A_3}$$

$$\overset{N}{\Gamma, (A_3 \supset A), A \longrightarrow A_2} \quad \otimes \quad \overset{N_1}{\Gamma, (A_3 \supset A), A, A_2 \longrightarrow A_1} \quad \Longrightarrow \quad \overset{N_2}{\Gamma, (A_3 \supset A), A \longrightarrow A_1}$$

```
car_orr1 : ca A D ([h:hyp A] orr1 (E1 h)) (orr1 E1')
            <- ca A D E1 E1'.
```

$$\overset{N}{\Gamma \longrightarrow A_1} \quad \otimes \quad \dfrac{\overset{N_1}{\Gamma, A_1 \longrightarrow A}}{\Gamma, A_1 \longrightarrow (A \vee A_2)} \vee R_1 \quad \Longrightarrow \quad \dfrac{\overset{N_2}{\Gamma \longrightarrow A}}{\Gamma \longrightarrow (A \vee A_2)} \vee R_1$$

$$\overset{N}{\Gamma \longrightarrow A_1} \quad \otimes \quad \overset{N_1}{\Gamma, A_1 \longrightarrow A} \quad \Longrightarrow \quad \overset{N_2}{\Gamma \longrightarrow A}$$

```
car_orr2 : ca A D ([h:hyp A] orr2 (E2 h)) (orr2 E2')
            <- ca A D E2 E2'.
```

$$\overset{N}{\Gamma \longrightarrow A_1} \quad \otimes \quad \dfrac{\overset{N_1}{\Gamma, A_1 \longrightarrow A}}{\Gamma, A_1 \longrightarrow (A_2 \vee A)} \vee R_2 \quad \Longrightarrow \quad \dfrac{\overset{N_2}{\Gamma \longrightarrow A}}{\Gamma \longrightarrow (A_2 \vee A)} \vee R_2$$

$$\overset{N}{\Gamma \longrightarrow A_1} \quad \otimes \quad \overset{N_1}{\Gamma, A_1 \longrightarrow A} \quad \Longrightarrow \quad \overset{N_2}{\Gamma \longrightarrow A}$$

```
car_orl  : ca A D ([h:hyp A] orl (E1 h) (E2 h) H) (orl E1' E2' H)
            <- ({h1:hyp B1} ca A D ([h:hyp A] E1 h h1) (E1' h1))
            <- ({h2:hyp B2} ca A D ([h:hyp A] E2 h h2) (E2' h2)).
```

$$\overset{N}{\Gamma,(A_3\vee A)\longrightarrow A_2} \quad\bigotimes\quad \dfrac{\overset{N_3}{\Gamma,(A_3\vee A),A_2,A_3\longrightarrow A_1}\qquad \overset{N_1}{\Gamma,(A_3\vee A),A_2,A\longrightarrow A_1}}{\Gamma,(A_3\vee A),A_2\longrightarrow A_1}\vee L$$

$$\Longrightarrow\quad \dfrac{\overset{N_4}{\Gamma,(A_3\vee A),A_3\longrightarrow A_1}\qquad \overset{N_2}{\Gamma,(A_3\vee A),A\longrightarrow A_1}}{\Gamma,(A_3\vee A)\longrightarrow A_1}\vee L$$

$$\overset{N}{\Gamma,(A_3\vee A),A_3\longrightarrow A_2} \quad\bigotimes\quad \overset{N_3}{\Gamma,(A_3\vee A),A_3,A_2\longrightarrow A_1} \quad\Longrightarrow\quad \overset{N_4}{\Gamma,(A_3\vee A),A_3\longrightarrow A_1}$$

$$\overset{N}{\Gamma,(A_3\vee A),A\longrightarrow A_2} \quad\bigotimes\quad \overset{N_1}{\Gamma,(A_3\vee A),A,A_2\longrightarrow A_1} \quad\Longrightarrow\quad \overset{N_2}{\Gamma,(A_3\vee A),A\longrightarrow A_1}$$

```
car_notr : ca A D ([h:hyp A] notr (E1 h)) (notr E1')
            <- ({p:o} {h1:hyp B1} ca A D ([h:hyp A] E1 h p h1) (E1' p h1)).
```

$$\overset{N}{\Gamma\longrightarrow A_1} \quad\bigotimes\quad \dfrac{\overset{N_1 p_1}{\Gamma,A_1,A\longrightarrow p_1}}{\Gamma,A_1\longrightarrow\neg(A)}\neg R^{p_1} \quad\Longrightarrow\quad \dfrac{\overset{N_2 p_1}{\Gamma,A\longrightarrow p_1}}{\Gamma\longrightarrow\neg(A)}\neg R^{p_1}$$

$$\overset{N}{\Gamma,A\longrightarrow A_1} \quad\bigotimes\quad \overset{N_1 p}{\Gamma,A,A_1\longrightarrow p} \quad\Longrightarrow\quad \overset{N_2 p}{\Gamma,A\longrightarrow p}$$

```
car_notl : ca A D ([h:hyp A] notl (E1 h) H) (notl E1' H)
            <- ca A D E1 E1'.
```

$$\overset{N}{\Gamma,\neg(A)\longrightarrow A_1} \quad\bigotimes\quad \dfrac{\overset{N_1}{\Gamma,\neg(A),A_1\longrightarrow A}}{\Gamma,\neg(A),A_1\longrightarrow A_2}\neg L \quad\Longrightarrow\quad \dfrac{\overset{N_2}{\Gamma,\neg(A)\longrightarrow A}}{\Gamma,\neg(A)\longrightarrow A_2}\neg L$$

$$\overset{N}{\Gamma,\neg(A)\longrightarrow A_1} \quad\bigotimes\quad \overset{N_1}{\Gamma,\neg(A),A_1\longrightarrow A} \quad\Longrightarrow\quad \overset{N_2}{\Gamma,\neg(A)\longrightarrow A}$$

```
car_truer: ca A D ([h:hyp A] truer) (truer).
```

$$\overset{N}{\Gamma \longrightarrow A} \quad \bigotimes \quad \dfrac{}{\Gamma, A \longrightarrow \top}\top R \quad \Longrightarrow \quad \dfrac{}{\Gamma \longrightarrow \top}\top R$$

```
car_falsel : ca A D ([h:hyp A] falsel H) (falsel H).
```

$$\overset{N}{\Gamma, \bot \longrightarrow A_1} \quad \bigotimes \quad \dfrac{}{\Gamma, \bot, A_1 \longrightarrow A}\bot L \quad \Longrightarrow \quad \dfrac{}{\Gamma, \bot \longrightarrow A}\bot L$$

```
car_forallr : ca A D ([h:hyp A] forallr (E1 h)) (forallr E1')
               <- ({a:i} ca A D ([h:hyp A] E1 h a) (E1' a)).
```

$$\overset{N}{\Gamma \longrightarrow A_1} \quad \bigotimes \quad \dfrac{\overset{N_1 a_1}{\Gamma, A_1 \longrightarrow A a_1}}{\Gamma, A_1 \longrightarrow (\forall x.\, Ax)}\forall R^{a_1} \quad \Longrightarrow \quad \dfrac{\overset{N_2 a_1}{\Gamma \longrightarrow A a_1}}{\Gamma \longrightarrow (\forall x.\, Ax)}\forall R^{a_1}$$

$$\overset{N}{\Gamma \longrightarrow A_1} \quad \bigotimes \quad \overset{N_1 a}{\Gamma, A_1 \longrightarrow Aa} \quad \Longrightarrow \quad \overset{N_2 a}{\Gamma \longrightarrow Aa}$$

```
car_foralll: ca A D ([h:hyp A] foralll T (E1 h) H) (foralll T E1' H)
              <- ({h1} ca A D ([h:hyp A] E1 h h1) (E1' h1)).
```

$$\overset{N}{\Gamma, (\forall x.\, Ax) \longrightarrow A_2} \quad \bigotimes \quad \dfrac{\overset{N_1}{\Gamma, (\forall x.\, Ax), A_2, At \longrightarrow A_1}}{\Gamma, (\forall x.\, Ax), A_2 \longrightarrow A_1}\forall L \quad \Longrightarrow \quad \dfrac{\overset{N_2}{\Gamma, (\forall x.\, Ax), At \longrightarrow A_1}}{\Gamma, (\forall x.\, Ax) \longrightarrow A_1}\forall L$$

$$\overset{N}{\Gamma, (\forall x.\, Ax), At \longrightarrow A_2} \quad \bigotimes \quad \overset{N_1}{\Gamma, (\forall x.\, Ax), At, A_2 \longrightarrow A_1} \quad \Longrightarrow \quad \overset{N_2}{\Gamma, (\forall x.\, Ax), At \longrightarrow A_1}$$

```
car_existsr : ca A D ([h:hyp A] existsr T (E1 h)) (existsr T E1')
               <- ca A D E1 E1'.
```

$$\begin{array}{c} N \\ \Gamma \longrightarrow A_1 \end{array} \quad \otimes \quad \dfrac{\begin{array}{c} N_1 \\ \Gamma, A_1 \longrightarrow At \end{array}}{\Gamma, A_1 \longrightarrow (\exists x.\, Ax)} \exists R \quad \Longrightarrow \quad \dfrac{\begin{array}{c} N_2 \\ \Gamma \longrightarrow At \end{array}}{\Gamma \longrightarrow (\exists x.\, Ax)} \exists R$$

$$\begin{array}{c} N \\ \Gamma \longrightarrow A_1 \end{array} \quad \otimes \quad \begin{array}{c} N_1 \\ \Gamma, A_1 \longrightarrow At \end{array} \quad \Longrightarrow \quad \begin{array}{c} N_2 \\ \Gamma \longrightarrow At \end{array}$$

```
car_existsl : ca A D ([h:hyp A] existsl (E1 h) H) (existsl E1' H)
               <- ({a:i} {h1:hyp (B1 a)}
                     ca A D ([h:hyp A] E1 h a h1) (E1' a h1)).
```

$$\begin{array}{c} N \\ \Gamma, (\exists x.\, Ax) \longrightarrow A_2 \end{array} \quad \otimes \quad \dfrac{\begin{array}{c} N_1 a_1 \\ \Gamma, (\exists x.\, Ax), A_2, Aa_1 \longrightarrow A_1 \end{array}}{\Gamma, (\exists x.\, Ax), A_2 \longrightarrow A_1} \exists L^{a_1} \quad \Longrightarrow \quad \dfrac{\begin{array}{c} N_2 a_1 \\ \Gamma, (\exists x.\, Ax), Aa_1 \longrightarrow A_1 \end{array}}{\Gamma, (\exists x.\, Ax) \longrightarrow A_1} \exists L^{a_1}$$

$$\begin{array}{c} N \\ \Gamma, (\exists x.\, Ax), Aa \longrightarrow A_2 \end{array} \quad \otimes \quad \begin{array}{c} N_1 a \\ \Gamma, (\exists x.\, Ax), Aa, A_2 \longrightarrow A_1 \end{array} \quad \Longrightarrow \quad \begin{array}{c} N_2 a \\ \Gamma, (\exists x.\, Ax), Aa \longrightarrow A_1 \end{array}$$

A.2 Classical Calculus

We list the cases using the same conventions as for the intuitionistic calculus above. A transformation now has the form

$$\begin{array}{c} \mathcal{D} \\ \Gamma \longrightarrow A, \Delta \end{array} \quad \otimes \quad \begin{array}{c} \mathcal{E} \\ \Gamma, A \longrightarrow \Delta \end{array} \quad \Longrightarrow \quad \begin{array}{c} \mathcal{F} \\ \Gamma \longrightarrow \Delta \end{array}$$

where $\mathcal{F}$ may refer to derivations constructed by appeals to the induction hypothesis. This relation is implemented by a type family

```
ca' : {A:o} (pos A -> #) -> (neg A -> #) -> # -> type.
```

Initial Conversions. Here either $\mathcal{D}$ or $\mathcal{E}$ is initial with the cut formula A as the principal formula. Note that appeals to contraction are implicit since we omit proof terms in this presentation. Recall that contraction does not change the structure of the derivation (only the proof term by substituting one formula label for another).

```
ca_axiom'l : ca' A ([p] axiom' N p) E (E N).
```

$$\dfrac{}{\Gamma, A \longrightarrow A, \Delta} I \quad \otimes \quad \begin{array}{c} N \\ \Gamma, A, A \longrightarrow \Delta \end{array} \quad \Longrightarrow \quad \begin{array}{c} N \\ \Gamma, A \longrightarrow \Delta \end{array}$$

```
ca_axiom'r : ca' A D ([n] axiom' n P) (D P).
```

$$\begin{array}{c} N \\ \Gamma \longrightarrow A, A, \Delta \end{array} \quad \otimes \quad \dfrac{}{\Gamma, A \longrightarrow A, \Delta} I \quad \Longrightarrow \quad \begin{array}{c} N \\ \Gamma \longrightarrow A, \Delta \end{array}$$

Essential Conversions. Here the cut formula A is the principal formula of the last inference in both $\mathcal{D}$ and $\mathcal{E}$.

```
ca_and1' :
  ca' (A and B) ([p] andr' (D1 p) (D2 p) p) ([n] andl1' (E1 n) n) F
     <- ({p1:pos A} ca' (A and B) ([p] D1 p p1) ([n] andl1' (E1 n) n) (D1' p1))
     <- ({n1:neg A}
           ca' (A and B) ([p] andr' (D1 p) (D2 p) p)
           ([n] E1 n n1) (E1' n1))
     <- ca' A ([p1] D1' p1) ([n1] E1' n1) F.
```

$$\dfrac{\overset{N_3}{\Gamma \longrightarrow A, (A \wedge A_1), \Delta} \qquad \overset{N_4}{\Gamma \longrightarrow A_1, (A \wedge A_1), \Delta}}{\Gamma \longrightarrow (A \wedge A_1), \Delta}\wedge R \quad \otimes \quad \dfrac{\overset{N_5}{\Gamma, (A \wedge A_1), A \longrightarrow \Delta}}{\Gamma, (A \wedge A_1) \longrightarrow \Delta}\wedge L_1$$

$$\Longrightarrow \quad \overset{N_2}{\Gamma \longrightarrow \Delta}$$

$$\overset{N_3}{\Gamma \longrightarrow (A \wedge A_1), A, \Delta} \quad \otimes \quad \dfrac{\overset{N_5}{\Gamma, (A \wedge A_1), A \longrightarrow A, \Delta}}{\Gamma, (A \wedge A_1) \longrightarrow A, \Delta}\wedge L_1 \quad \Longrightarrow \quad \overset{N}{\Gamma \longrightarrow A, \Delta}$$

$$\dfrac{\overset{N_3}{\Gamma, A \longrightarrow A, (A \wedge A_1), \Delta} \qquad \overset{N_4}{\Gamma, A \longrightarrow A_1, (A \wedge A_1), \Delta}}{\Gamma, A \longrightarrow (A \wedge A_1), \Delta}\wedge R \quad \otimes \quad \overset{N_5}{\Gamma, A, (A \wedge A_1) \longrightarrow \Delta}$$

$$\Longrightarrow \quad \overset{N_1}{\Gamma, A \longrightarrow \Delta}$$

$$\overset{N}{\Gamma \longrightarrow A, \Delta} \quad \otimes \quad \overset{N_1}{\Gamma, A \longrightarrow \Delta} \quad \Longrightarrow \quad \overset{N_2}{\Gamma \longrightarrow \Delta}$$

```
ca_and2' :
  ca' (A and B) ([p] andr' (D1 p) (D2 p) p) ([n] andl2' (E2 n) n) F
     <- ({p2:pos B} ca' (A and B) ([p] D2 p p2) ([n] andl2' (E2 n) n) (D2' p2))
     <- ({n2:neg B}
           ca' (A and B) ([p] andr' (D1 p) (D2 p) p)
           ([n] E2 n n2) (E2' n2))
     <- ca' B ([p2] D2' p2) ([n2] E2' n2) F.
```

$$\dfrac{\overset{N_3}{\Gamma \longrightarrow A_1, (A_1 \wedge A), \Delta} \qquad \overset{N_4}{\Gamma \longrightarrow A, (A_1 \wedge A), \Delta}}{\Gamma \longrightarrow (A_1 \wedge A), \Delta}\wedge R \quad \otimes \quad \dfrac{\overset{N_5}{\Gamma, (A_1 \wedge A), A \longrightarrow \Delta}}{\Gamma, (A_1 \wedge A) \longrightarrow \Delta}\wedge L_2$$

$$\Longrightarrow \quad \overset{N_2}{\Gamma \longrightarrow \Delta}$$

$$\overset{N_4}{\Gamma \longrightarrow (A_1 \wedge A), A, \Delta} \quad \otimes \quad \dfrac{\overset{N_5}{\Gamma, (A_1 \wedge A), A \longrightarrow A, \Delta}}{\Gamma, (A_1 \wedge A) \longrightarrow A, \Delta}\wedge L_2 \quad \Longrightarrow \quad \overset{N}{\Gamma \longrightarrow A, \Delta}$$

$$\dfrac{\overset{N_3}{\Gamma, A \longrightarrow A_1, (A_1 \wedge A), \Delta} \qquad \overset{N_4}{\Gamma, A \longrightarrow A, (A_1 \wedge A), \Delta}}{\Gamma, A \longrightarrow (A_1 \wedge A), \Delta}\wedge R \quad \otimes \quad \overset{N_5}{\Gamma, A, (A_1 \wedge A) \longrightarrow \Delta}$$

$$\Longrightarrow \quad \overset{N_1}{\Gamma, A \longrightarrow \Delta}$$

$$\overset{N}{\Gamma \longrightarrow A, \Delta} \quad \otimes \quad \overset{N_1}{\Gamma, A \longrightarrow \Delta} \quad \Longrightarrow \quad \overset{N_2}{\Gamma \longrightarrow \Delta}$$

```
ca_imp' :
  ca' (A imp B) ([p] impr' (D1 p) p) ([n] impl' (E1 n) (E2 n) n) F
     <- ({p1:pos A} ca' (A imp B) ([p] impr' (D1 p) p) ([n] E1 n p1) (E1' p1))
     <- ({n2:neg B} ca' (A imp B) ([p] impr' (D1 p) p) ([n] E2 n n2) (E2' n2))
     <- ({n1:neg A} {p2:pos B}
           ca' (A imp B) ([p] D1 p n1 p2) ([n] impl' (E1 n) (E2 n) n)
           (D1' n1 p2))
     <- ({p2:pos B} ca' A ([p1] E1' p1) ([n1] D1' n1 p2) (F2 p2))
     <- ca' B ([p2] F2 p2) ([n2] E2' n2) F.
```

$$\dfrac{\overset{N_5}{\Gamma, A_1 \longrightarrow A, (A_1 \supset A), \Delta}}{\Gamma \longrightarrow (A_1 \supset A), \Delta}\supset R \quad \bigotimes \quad \dfrac{\overset{N_6}{\Gamma, (A_1 \supset A) \longrightarrow A_1, \Delta} \qquad \overset{N_7}{\Gamma, (A_1 \supset A), A \longrightarrow \Delta}}{\Gamma, (A_1 \supset A) \longrightarrow \Delta}\supset L$$

$$\Longrightarrow \quad \overset{N_2}{\Gamma \longrightarrow \Delta}$$

$$\dfrac{\overset{N_5}{\Gamma, A_1 \longrightarrow A, (A_1 \supset A), A_1, \Delta}}{\Gamma \longrightarrow (A_1 \supset A), A_1, \Delta}\supset R \quad \bigotimes \quad \overset{N_6}{\Gamma, (A_1 \supset A) \longrightarrow A_1, \Delta} \quad \Longrightarrow \quad \overset{N_3}{\Gamma \longrightarrow A_1, \Delta}$$

$$\dfrac{\overset{N_5}{\Gamma, A, A_1 \longrightarrow A, (A_1 \supset A), \Delta}}{\Gamma, A \longrightarrow (A_1 \supset A), \Delta}\supset R \quad \bigotimes \quad \overset{N_7}{\Gamma, A, (A_1 \supset A) \longrightarrow \Delta} \quad \Longrightarrow \quad \overset{N_1}{\Gamma, A \longrightarrow \Delta}$$

$$\overset{N_5}{\Gamma, A_1 \longrightarrow (A_1 \supset A), A, \Delta} \quad \bigotimes \quad \dfrac{\overset{N_6}{\Gamma, A_1, (A_1 \supset A) \longrightarrow A_1, A, \Delta} \qquad \overset{N_7}{\Gamma, A_1, (A_1 \supset A), A \longrightarrow A, \Delta}}{\Gamma, A_1, (A_1 \supset A) \longrightarrow A, \Delta}\supset L$$

$$\Longrightarrow \quad \overset{N_4}{\Gamma, A_1 \longrightarrow A, \Delta}$$

$$\overset{N_3}{\Gamma \longrightarrow A_1, A, \Delta} \quad \bigotimes \quad \overset{N_4}{\Gamma, A_1 \longrightarrow A, \Delta} \quad \Longrightarrow \quad \overset{N}{\Gamma \longrightarrow A, \Delta}$$

$$\overset{N}{\Gamma \longrightarrow A, \Delta} \quad \bigotimes \quad \overset{N_1}{\Gamma, A \longrightarrow \Delta} \quad \Longrightarrow \quad \overset{N_2}{\Gamma \longrightarrow \Delta}$$

```
ca_or1' :
  ca' (A or B) ([p] orr1' (D1 p) p) ([n] orl' (E1 n) (E2 n) n) F
     <- ({n1:neg A} ca' (A or B) ([p] orr1' (D1 p) p) ([n] E1 n n1) (E1' n1))
     <- ({p1:pos A} ca' (A or B) ([p] D1 p p1) ([n] orl' (E1 n) (E2 n) n)
           (D1' p1))
     <- ca' A D1' E1' F.
```

$$\dfrac{\overset{N_3}{\Gamma \longrightarrow A, (A \vee A_1), \Delta}}{\Gamma \longrightarrow (A \vee A_1), \Delta}\vee R_1 \quad \otimes \quad \dfrac{\overset{N_4}{\Gamma, (A \vee A_1), A \longrightarrow \Delta} \qquad \overset{N_5}{\Gamma, (A \vee A_1), A_1 \longrightarrow \Delta}}{\Gamma, (A \vee A_1) \longrightarrow \Delta}\vee L$$

$$\Longrightarrow \quad \overset{N_2}{\Gamma \longrightarrow \Delta}$$

$$\dfrac{\overset{N_3}{\Gamma, A \longrightarrow A, (A \vee A_1), \Delta}}{\Gamma, A \longrightarrow (A \vee A_1), \Delta}\vee R_1 \quad \otimes \quad \overset{N_4}{\Gamma, A, (A \vee A_1) \longrightarrow \Delta} \quad \Longrightarrow \quad \overset{N_1}{\Gamma, A \longrightarrow \Delta}$$

$$\overset{N_3}{\Gamma \longrightarrow (A \vee A_1), A, \Delta} \quad \otimes \quad \dfrac{\overset{N_4}{\Gamma, (A \vee A_1), A \longrightarrow A, \Delta} \qquad \overset{N_5}{\Gamma, (A \vee A_1), A_1 \longrightarrow A, \Delta}}{\Gamma, (A \vee A_1) \longrightarrow A, \Delta}\vee L$$

$$\Longrightarrow \quad \overset{N}{\Gamma \longrightarrow A, \Delta}$$

$$\overset{N}{\Gamma \longrightarrow A, \Delta} \quad \otimes \quad \overset{N_1}{\Gamma, A \longrightarrow \Delta} \quad \Longrightarrow \quad \overset{N_2}{\Gamma \longrightarrow \Delta}$$

```
ca_or2' :
  ca' (A or B) ([p] orr2' (D2 p) p) ([n] orl' (E1 n) (E2 n) n) F
     <- ({n2:neg B} ca' (A or B) ([p] orr2' (D2 p) p) ([n] E2 n n2) (E2' n2))
     <- ({p2:pos B} ca' (A or B) ([p] D2 p p2) ([n] orl' (E1 n) (E2 n) n)
           (D2' p2))
     <- ca' B D2' E2' F.
```

$$\dfrac{\overset{N_3}{\Gamma \longrightarrow A, (A_1 \vee A), \Delta}}{\Gamma \longrightarrow (A_1 \vee A), \Delta}\vee R_2 \quad \bigotimes \quad \dfrac{\overset{N_4}{\Gamma, (A_1 \vee A), A_1 \longrightarrow \Delta} \qquad \overset{N_5}{\Gamma, (A_1 \vee A), A \longrightarrow \Delta}}{\Gamma, (A_1 \vee A) \longrightarrow \Delta}\vee L$$

$$\Longrightarrow \quad \overset{N_2}{\Gamma \longrightarrow \Delta}$$

$$\dfrac{\overset{N_3}{\Gamma, A \longrightarrow A, (A_1 \vee A), \Delta}}{\Gamma, A \longrightarrow (A_1 \vee A), \Delta}\vee R_2 \quad \bigotimes \quad \overset{N_5}{\Gamma, A, (A_1 \vee A) \longrightarrow \Delta} \quad \Longrightarrow \quad \overset{N_1}{\Gamma, A \longrightarrow \Delta}$$

$$\overset{N_3}{\Gamma \longrightarrow (A_1 \vee A), A, \Delta} \quad \bigotimes \quad \dfrac{\overset{N_4}{\Gamma, (A_1 \vee A), A_1 \longrightarrow A, \Delta} \qquad \overset{N_5}{\Gamma, (A_1 \vee A), A \longrightarrow A, \Delta}}{\Gamma, (A_1 \vee A) \longrightarrow A, \Delta}\vee L$$

$$\Longrightarrow \quad \overset{N}{\Gamma \longrightarrow A, \Delta}$$

$$\overset{N}{\Gamma \longrightarrow A, \Delta} \quad \bigotimes \quad \overset{N_1}{\Gamma, A \longrightarrow \Delta} \quad \Longrightarrow \quad \overset{N_2}{\Gamma \longrightarrow \Delta}$$

```
ca_not' :
  ca' (not A) ([p] notr' (D1 p) p) ([n] notl' (E1 n) n) F
     <- ({p1:pos A} ca' (not A) ([p] notr' (D1 p) p) ([n] E1 n p1) (E1' p1))
     <- ({n1:neg A} ca' (not A) ([p] D1 p n1) ([n] notl' (E1 n) n) (D1' n1))
     <- ca' A E1' D1' F.
```

$$\dfrac{\overset{N_3}{\Gamma, A \longrightarrow \neg(A), \Delta}}{\Gamma \longrightarrow \neg(A), \Delta}\neg R \quad \otimes \quad \dfrac{\overset{N_4}{\Gamma, \neg(A) \longrightarrow A, \Delta}}{\Gamma, \neg(A) \longrightarrow \Delta}\neg L \quad \Longrightarrow \quad \overset{N_2}{\Gamma \longrightarrow \Delta}$$

$$\dfrac{\overset{N_3}{\Gamma, A \longrightarrow \neg(A), A, \Delta}}{\Gamma \longrightarrow \neg(A), A, \Delta}\neg R \quad \otimes \quad \overset{N_4}{\Gamma, \neg(A) \longrightarrow A, \Delta} \quad \Longrightarrow \quad \overset{N}{\Gamma \longrightarrow A, \Delta}$$

$$\overset{N_3}{\Gamma, A \longrightarrow \neg(A), \Delta} \quad \otimes \quad \dfrac{\overset{N_4}{\Gamma, A, \neg(A) \longrightarrow A, \Delta}}{\Gamma, A, \neg(A) \longrightarrow \Delta}\neg L \quad \Longrightarrow \quad \overset{N_1}{\Gamma, A \longrightarrow \Delta}$$

$$\overset{N}{\Gamma \longrightarrow A, \Delta} \quad \otimes \quad \overset{N_1}{\Gamma, A \longrightarrow \Delta} \quad \Longrightarrow \quad \overset{N_2}{\Gamma \longrightarrow \Delta}$$

```
ca_forall' :
  ca' (forall A) ([p] forallr' (D1 p) p) ([n] foralll' T (E1 n) n) F
     <- ({n1} ca' (forall A) ([p] forallr' (D1 p) p) ([n] E1 n n1) (E1' n1))
     <- ({p1} ca' (forall A) ([p] D1 p T p1) ([n] foralll' T (E1 n) n) (D1' p1))
     <- ca' (A T) D1' E1' F.
```

$$\dfrac{\overset{N_3 a}{\Gamma \longrightarrow Aa, (\forall x.\ Ax), \Delta}}{\Gamma \longrightarrow (\forall x.\ Ax), \Delta}\forall R^a \quad \otimes \quad \dfrac{\overset{N_4}{\Gamma, (\forall x.\ Ax), At \longrightarrow \Delta}}{\Gamma, (\forall x.\ Ax) \longrightarrow \Delta}\forall L \quad \Longrightarrow \quad \overset{N_2}{\Gamma \longrightarrow \Delta}$$

$$\dfrac{\overset{N_3 a}{\Gamma, At \longrightarrow Aa, (\forall x.\ Ax), \Delta}}{\Gamma, At \longrightarrow (\forall x.\ Ax), \Delta}\forall R^a \quad \otimes \quad \overset{N_4}{\Gamma, At, (\forall x.\ Ax) \longrightarrow \Delta} \quad \Longrightarrow \quad \overset{N_1}{\Gamma, At \longrightarrow \Delta}$$

$$\overset{N_3 t}{\Gamma \longrightarrow (\forall x.\ Ax), At, \Delta} \quad \otimes \quad \dfrac{\overset{N_4}{\Gamma, (\forall x.\ Ax), At \longrightarrow At, \Delta}}{\Gamma, (\forall x.\ Ax) \longrightarrow At, \Delta}\forall L \quad \Longrightarrow \quad \overset{N}{\Gamma \longrightarrow At, \Delta}$$

$$\overset{N}{\Gamma \longrightarrow At, \Delta} \quad \otimes \quad \overset{N_1}{\Gamma, At \longrightarrow \Delta} \quad \Longrightarrow \quad \overset{N_2}{\Gamma \longrightarrow \Delta}$$

```
ca_exists' :
  ca' (exists A) ([p] existsr' T (D1 p) p) ([n] existsl' (E1 n) n) F
     <- ({n1} ca' (exists A) ([p] existsr' T (D1 p) p) ([n] E1 n T n1) (E1' n1))
     <- ({p1} ca' (exists A) ([p] D1 p p1) ([n] existsl' (E1 n) n) (D1' p1))
     <- ca' (A T) D1' E1' F.
```

$$\dfrac{\begin{array}{c}N_3\\ \Gamma \longrightarrow At, (\exists x.\ Ax), \Delta\end{array}}{\Gamma \longrightarrow (\exists x.\ Ax), \Delta}\exists R \quad \otimes \quad \dfrac{\begin{array}{c}N_4 a\\ \Gamma, (\exists x.\ Ax), Aa \longrightarrow \Delta\end{array}}{\Gamma, (\exists x.\ Ax) \longrightarrow \Delta}\exists L^a \quad \Longrightarrow \quad \begin{array}{c}N_2\\ \Gamma \longrightarrow \Delta\end{array}$$

$$\dfrac{\begin{array}{c}N_3\\ \Gamma, At \longrightarrow At, (\exists x.\ Ax), \Delta\end{array}}{\Gamma, At \longrightarrow (\exists x.\ Ax), \Delta}\exists R \quad \otimes \quad \begin{array}{c}N_4 t\\ \Gamma, At, (\exists x.\ Ax) \longrightarrow \Delta\end{array} \quad \Longrightarrow \quad \begin{array}{c}N_1\\ \Gamma, At \longrightarrow \Delta\end{array}$$

$$\begin{array}{c}N_3\\ \Gamma \longrightarrow (\exists x.\ Ax), At, \Delta\end{array} \quad \otimes \quad \dfrac{\begin{array}{c}N_4 a\\ \Gamma, (\exists x.\ Ax), Aa \longrightarrow At, \Delta\end{array}}{\Gamma, (\exists x.\ Ax) \longrightarrow At, \Delta}\exists L^a \quad \Longrightarrow \quad \begin{array}{c}N\\ \Gamma \longrightarrow At, \Delta\end{array}$$

$$\begin{array}{c}N\\ \Gamma \longrightarrow At, \Delta\end{array} \quad \otimes \quad \begin{array}{c}N_1\\ \Gamma, At \longrightarrow \Delta\end{array} \quad \Longrightarrow \quad \begin{array}{c}N_2\\ \Gamma \longrightarrow \Delta\end{array}$$

Right Commutative Conversions. Here the cut formula is a side formula of the last inference in $\mathcal{E}$.

```
car_axiom' : ca' A D ([n] axiom' N P) (axiom' N P).
```

$$\begin{array}{c}N\\ \Gamma, A_1 \longrightarrow A, A_1, \Delta\end{array} \quad \otimes \quad \dfrac{}{\Gamma, A_1, A \longrightarrow A_1, \Delta}I \quad \Longrightarrow \quad \dfrac{}{\Gamma, A_1 \longrightarrow A_1, \Delta}I$$

```
car_andr' :
  ca' A D ([n] andr' (E1 n) (E2 n) P) (andr' F1 F2 P)
     <- ({p1:pos B1} ca' A D ([n] E1 n p1) (F1 p1))
     <- ({p2:pos B2} ca' A D ([n] E2 n p2) (F2 p2)).
```

$$\overset{N}{\Gamma \longrightarrow A_1, (A_2 \wedge A), \Delta} \quad \bigotimes \quad \dfrac{\overset{N_3}{\Gamma, A_1 \longrightarrow A_2, (A_2 \wedge A), \Delta} \qquad \overset{N_1}{\Gamma, A_1 \longrightarrow A, (A_2 \wedge A), \Delta}}{\Gamma, A_1 \longrightarrow (A_2 \wedge A), \Delta}\wedge R$$

$$\Longrightarrow \quad \dfrac{\overset{N_4}{\Gamma \longrightarrow A_2, (A_2 \wedge A), \Delta} \qquad \overset{N_2}{\Gamma \longrightarrow A, (A_2 \wedge A), \Delta}}{\Gamma \longrightarrow (A_2 \wedge A), \Delta}\wedge R$$

$$\overset{N}{\Gamma \longrightarrow A_1, A_2, (A_2 \wedge A), \Delta} \quad \bigotimes \quad \overset{N_3}{\Gamma, A_1 \longrightarrow A_2, (A_2 \wedge A), \Delta} \quad \Longrightarrow \quad \overset{N_4}{\Gamma \longrightarrow A_2, (A_2 \wedge A), \Delta}$$

$$\overset{N}{\Gamma \longrightarrow A_1, A, (A_2 \wedge A), \Delta} \quad \bigotimes \quad \overset{N_1}{\Gamma, A_1 \longrightarrow A, (A_2 \wedge A), \Delta} \quad \Longrightarrow \quad \overset{N_2}{\Gamma \longrightarrow A, (A_2 \wedge A), \Delta}$$

```
car_andl1' :
  ca' A D ([n] andl1' (E1 n) N) (andl1' F1 N)
     <- ({n1:neg B1} ca' A D ([n] E1 n n1) (F1 n1)).
```

$$\overset{N}{\Gamma, (A \wedge A_2) \longrightarrow A_1, \Delta} \quad \bigotimes \quad \dfrac{\overset{N_1}{\Gamma, (A \wedge A_2), A_1, A \longrightarrow \Delta}}{\Gamma, (A \wedge A_2), A_1 \longrightarrow \Delta}\wedge L_1 \quad \Longrightarrow \quad \dfrac{\overset{N_2}{\Gamma, (A \wedge A_2), A \longrightarrow \Delta}}{\Gamma, (A \wedge A_2) \longrightarrow \Delta}\wedge L_1$$

$$\overset{N}{\Gamma, (A \wedge A_2), A \longrightarrow A_1, \Delta} \quad \bigotimes \quad \overset{N_1}{\Gamma, (A \wedge A_2), A, A_1 \longrightarrow \Delta} \quad \Longrightarrow \quad \overset{N_2}{\Gamma, (A \wedge A_2), A \longrightarrow \Delta}$$

```
car_andl2' :
  ca' A D ([n] andl2' (E2 n) N) (andl2' F2 N)
     <- ({n2:neg B2} ca' A D ([n] E2 n n2) (F2 n2)).
```

$$\overset{N}{\Gamma, (A_2 \wedge A) \longrightarrow A_1, \Delta} \quad \bigotimes \quad \dfrac{\overset{N_1}{\Gamma, (A_2 \wedge A), A_1, A \longrightarrow \Delta}}{\Gamma, (A_2 \wedge A), A_1 \longrightarrow \Delta}\wedge L_2 \quad \Longrightarrow \quad \dfrac{\overset{N_2}{\Gamma, (A_2 \wedge A), A \longrightarrow \Delta}}{\Gamma, (A_2 \wedge A) \longrightarrow \Delta}\wedge L_2$$

$$\overset{N}{\Gamma, (A_2 \wedge A), A \longrightarrow A_1, \Delta} \quad \bigotimes \quad \overset{N_1}{\Gamma, (A_2 \wedge A), A, A_1 \longrightarrow \Delta} \quad \Longrightarrow \quad \overset{N_2}{\Gamma, (A_2 \wedge A), A \longrightarrow \Delta}$$

```
car_impr' :
  ca' A D ([n] impr' (E1 n) P) (impr' F1 P)
     <- ({n1:neg B1} {p2:pos B2}
           ca' A D ([n] E1 n n1 p2) (F1 n1 p2)).
```

$$\begin{array}{c} N \\ \Gamma \longrightarrow A_2, (A \supset A_1), \Delta \end{array} \otimes \dfrac{\begin{array}{c} N_1 \\ \Gamma, A_2, A \longrightarrow A_1, (A \supset A_1), \Delta \end{array}}{\Gamma, A_2 \longrightarrow (A \supset A_1), \Delta} {\supset}R \quad \Longrightarrow \quad \dfrac{\begin{array}{c} N_2 \\ \Gamma, A \longrightarrow A_1, (A \supset A_1), \Delta \end{array}}{\Gamma \longrightarrow (A \supset A_1), \Delta} {\supset}R$$

$$\begin{array}{c} N \\ \Gamma, A \longrightarrow A_2, A_1, (A \supset A_1), \Delta \end{array} \otimes \begin{array}{c} N_1 \\ \Gamma, A, A_2 \longrightarrow A_1, (A \supset A_1), \Delta \end{array} \Longrightarrow \begin{array}{c} N_2 \\ \Gamma, A \longrightarrow A_1, (A \supset A_1), \Delta \end{array}$$

```
car_impl' :
  ca' A D ([n] impl' (E1 n) (E2 n) N) (impl' F1 F2 N)
     <- ({p1:pos B1} ca' A D ([n] E1 n p1) (F1 p1))
     <- ({n2:neg B2} ca' A D ([n] E2 n n2) (F2 n2)).
```

$$\begin{array}{c} N \\ \Gamma, (A_2 \supset A) \longrightarrow A_1, \Delta \end{array} \otimes \dfrac{\begin{array}{c} N_3 \\ \Gamma, (A_2 \supset A), A_1 \longrightarrow A_2, \Delta \end{array} \qquad \begin{array}{c} N_1 \\ \Gamma, (A_2 \supset A), A_1, A \longrightarrow \Delta \end{array}}{\Gamma, (A_2 \supset A), A_1 \longrightarrow \Delta} {\supset}L$$

$$\Longrightarrow \quad \dfrac{\begin{array}{c} N_4 \\ \Gamma, (A_2 \supset A) \longrightarrow A_2, \Delta \end{array} \qquad \begin{array}{c} N_2 \\ \Gamma, (A_2 \supset A), A \longrightarrow \Delta \end{array}}{\Gamma, (A_2 \supset A) \longrightarrow \Delta} {\supset}L$$

$$\begin{array}{c} N \\ \Gamma, (A_2 \supset A) \longrightarrow A_1, A_2, \Delta \end{array} \otimes \begin{array}{c} N_3 \\ \Gamma, (A_2 \supset A), A_1 \longrightarrow A_2, \Delta \end{array} \Longrightarrow \begin{array}{c} N_4 \\ \Gamma, (A_2 \supset A) \longrightarrow A_2, \Delta \end{array}$$

$$\begin{array}{c} N \\ \Gamma, (A_2 \supset A), A \longrightarrow A_1, \Delta \end{array} \otimes \begin{array}{c} N_1 \\ \Gamma, (A_2 \supset A), A, A_1 \longrightarrow \Delta \end{array} \Longrightarrow \begin{array}{c} N_2 \\ \Gamma, (A_2 \supset A), A \longrightarrow \Delta \end{array}$$

```
car_orr1' :
  ca' A D ([n] orr1' (E1 n) P) (orr1' F1 P)
     <- ({p1:pos B1} ca' A D ([n] E1 n p1) (F1 p1)).
```

$$\begin{array}{c} N \\ \Gamma \longrightarrow A_1, (A \vee A_2), \Delta \end{array} \otimes \dfrac{\begin{array}{c} N_1 \\ \Gamma, A_1 \longrightarrow A, (A \vee A_2), \Delta \end{array}}{\Gamma, A_1 \longrightarrow (A \vee A_2), \Delta} {\vee}R_1 \quad \Longrightarrow \quad \dfrac{\begin{array}{c} N_2 \\ \Gamma \longrightarrow A, (A \vee A_2), \Delta \end{array}}{\Gamma \longrightarrow (A \vee A_2), \Delta} {\vee}R_1$$

$$\begin{array}{c} N \\ \Gamma \longrightarrow A_1, A, (A \vee A_2), \Delta \end{array} \otimes \begin{array}{c} N_1 \\ \Gamma, A_1 \longrightarrow A, (A \vee A_2), \Delta \end{array} \Longrightarrow \begin{array}{c} N_2 \\ \Gamma \longrightarrow A, (A \vee A_2), \Delta \end{array}$$

```
car_orr2' :
  ca' A D ([n] orr2' (E2 n) P) (orr2' F2 P)
     <- ({p2:pos B2} ca' A D ([n] E2 n p2) (F2 p2)).
```

$$\begin{array}{c}N\\ \Gamma \longrightarrow A_1, (A_2 \vee A), \Delta\end{array} \quad \bigotimes \quad \dfrac{\begin{array}{c}N_1\\ \Gamma, A_1 \longrightarrow A, (A_2 \vee A), \Delta\end{array}}{\Gamma, A_1 \longrightarrow (A_2 \vee A), \Delta}\vee R_2 \quad \Longrightarrow \quad \dfrac{\begin{array}{c}N_2\\ \Gamma \longrightarrow A, (A_2 \vee A), \Delta\end{array}}{\Gamma \longrightarrow (A_2 \vee A), \Delta}\vee R_2$$

$$\begin{array}{c}N\\ \Gamma \longrightarrow A_1, A, (A_2 \vee A), \Delta\end{array} \quad \bigotimes \quad \begin{array}{c}N_1\\ \Gamma, A_1 \longrightarrow A, (A_2 \vee A), \Delta\end{array} \quad \Longrightarrow \quad \begin{array}{c}N_2\\ \Gamma \longrightarrow A, (A_2 \vee A), \Delta\end{array}$$

```
car_orl' :
  ca' A D ([n] orl' (E1 n) (E2 n) N) (orl' F1 F2 N)
     <- ({n1:neg B1} ca' A D ([n] E1 n n1) (F1 n1))
     <- ({n2:neg B2} ca' A D ([n] E2 n n2) (F2 n2)).
```

$$\begin{array}{c}N\\ \Gamma, (A_2 \vee A) \longrightarrow A_1, \Delta\end{array} \quad \bigotimes \quad \dfrac{\begin{array}{c}N_3\\ \Gamma, (A_2 \vee A), A_1, A_2 \longrightarrow \Delta\end{array} \qquad \begin{array}{c}N_1\\ \Gamma, (A_2 \vee A), A_1, A \longrightarrow \Delta\end{array}}{\Gamma, (A_2 \vee A), A_1 \longrightarrow \Delta}\vee L$$

$$\Longrightarrow \quad \dfrac{\begin{array}{c}N_4\\ \Gamma, (A_2 \vee A), A_2 \longrightarrow \Delta\end{array} \qquad \begin{array}{c}N_2\\ \Gamma, (A_2 \vee A), A \longrightarrow \Delta\end{array}}{\Gamma, (A_2 \vee A) \longrightarrow \Delta}\vee L$$

$$\begin{array}{c}N\\ \Gamma, (A_2 \vee A), A_2 \longrightarrow A_1, \Delta\end{array} \quad \bigotimes \quad \begin{array}{c}N_3\\ \Gamma, (A_2 \vee A), A_2, A_1 \longrightarrow \Delta\end{array} \quad \Longrightarrow \quad \begin{array}{c}N_4\\ \Gamma, (A_2 \vee A), A_2 \longrightarrow \Delta\end{array}$$

$$\begin{array}{c}N\\ \Gamma, (A_2 \vee A), A \longrightarrow A_1, \Delta\end{array} \quad \bigotimes \quad \begin{array}{c}N_1\\ \Gamma, (A_2 \vee A), A, A_1 \longrightarrow \Delta\end{array} \quad \Longrightarrow \quad \begin{array}{c}N_2\\ \Gamma, (A_2 \vee A), A \longrightarrow \Delta\end{array}$$

```
car_notr' :
  ca' A D ([n] notr' (E1 n) P) (notr' F1 P)
     <- ({n1:neg B1} ca' A D ([n] E1 n n1) (F1 n1)).
```

$$\begin{array}{c}N\\ \Gamma \longrightarrow A_1, \neg(A), \Delta\end{array} \quad \bigotimes \quad \dfrac{\begin{array}{c}N_1\\ \Gamma, A_1, A \longrightarrow \neg(A), \Delta\end{array}}{\Gamma, A_1 \longrightarrow \neg(A), \Delta}\neg R \quad \Longrightarrow \quad \dfrac{\begin{array}{c}N_2\\ \Gamma, A \longrightarrow \neg(A), \Delta\end{array}}{\Gamma \longrightarrow \neg(A), \Delta}\neg R$$

$$\begin{array}{c}N\\ \Gamma, A \longrightarrow A_1, \neg(A), \Delta\end{array} \quad \bigotimes \quad \begin{array}{c}N_1\\ \Gamma, A, A_1 \longrightarrow \neg(A), \Delta\end{array} \quad \Longrightarrow \quad \begin{array}{c}N_2\\ \Gamma, A \longrightarrow \neg(A), \Delta\end{array}$$

```
car_notl' :
  ca' A D ([n] notl' (E1 n) N) (notl' F1 N)
     <- ({p1:pos B1} ca' A D ([n] E1 n p1) (F1 p1)).
```

$$\dfrac{N}{\Gamma, \neg(A) \longrightarrow A_1, \Delta} \quad \otimes \quad \dfrac{\dfrac{N_1}{\Gamma, \neg(A), A_1 \longrightarrow A, \Delta}}{\Gamma, \neg(A), A_1 \longrightarrow \Delta}\neg L \quad \Longrightarrow \quad \dfrac{\dfrac{N_2}{\Gamma, \neg(A) \longrightarrow A, \Delta}}{\Gamma, \neg(A) \longrightarrow \Delta}\neg L$$

$$\dfrac{N}{\Gamma, \neg(A) \longrightarrow A_1, A, \Delta} \quad \otimes \quad \dfrac{N_1}{\Gamma, \neg(A), A_1 \longrightarrow A, \Delta} \quad \Longrightarrow \quad \dfrac{N_2}{\Gamma, \neg(A) \longrightarrow A, \Delta}$$

```
car_truer' :
  ca' A D ([n] truer' P) (truer' P).
```

$$\dfrac{N}{\Gamma \longrightarrow A, \top, \Delta} \quad \otimes \quad \dfrac{}{\Gamma, A \longrightarrow \top, \Delta}\top R \quad \Longrightarrow \quad \dfrac{}{\Gamma \longrightarrow \top, \Delta}\top R$$

```
car_falsel' :
  ca' A D ([n] falsel' N) (falsel' N).
```

$$\dfrac{N}{\Gamma, \bot \longrightarrow A, \Delta} \quad \otimes \quad \dfrac{}{\Gamma, \bot, A \longrightarrow \Delta}\bot L \quad \Longrightarrow \quad \dfrac{}{\Gamma, \bot \longrightarrow \Delta}\bot L$$

```
car_forallr' :
  ca' A D ([n] forallr' (E1 n) P) (forallr' F1 P)
     <- ({a:i} {p1:pos (B1 a)}
           ca' A D ([n] E1 n a p1) (F1 a p1)).
```

$$\dfrac{N}{\Gamma \longrightarrow A_1, (\forall x.\, Ax), \Delta} \quad \otimes \quad \dfrac{\dfrac{N_1 a_1}{\Gamma, A_1 \longrightarrow Aa_1, (\forall x.\, Ax), \Delta}}{\Gamma, A_1 \longrightarrow (\forall x.\, Ax), \Delta}\forall R^{a_1} \quad \Longrightarrow \quad \dfrac{\dfrac{N_2 a_1}{\Gamma \longrightarrow Aa_1, (\forall x.\, Ax), \Delta}}{\Gamma \longrightarrow (\forall x.\, Ax), \Delta}\forall R^{a_1}$$

$$\dfrac{N}{\Gamma \longrightarrow A_1, Aa, (\forall x.\, Ax), \Delta} \quad \otimes \quad \dfrac{N_1 a}{\Gamma, A_1 \longrightarrow Aa, (\forall x.\, Ax), \Delta} \quad \Longrightarrow \quad \dfrac{N_2 a}{\Gamma \longrightarrow Aa, (\forall x.\, Ax), \Delta}$$

```
car_foralll' :
  ca' A D ([n] foralll' T (E1 n) N) (foralll' T F1 N)
     <- ({n1} ca' A D ([n] E1 n n1) (F1 n1)).
```

$$\begin{array}{c} N \\ \Gamma, (\forall x.\ Ax) \longrightarrow A_1, \Delta \end{array} \quad \otimes \quad \dfrac{\begin{array}{c} N_1 \\ \Gamma, (\forall x.\ Ax), A_1, At \longrightarrow \Delta \end{array}}{\Gamma, (\forall x.\ Ax), A_1 \longrightarrow \Delta} \forall L \quad \Longrightarrow \quad \dfrac{\begin{array}{c} N_2 \\ \Gamma, (\forall x.\ Ax), At \longrightarrow \Delta \end{array}}{\Gamma, (\forall x.\ Ax) \longrightarrow \Delta} \forall L$$

$$\begin{array}{c} N \\ \Gamma, (\forall x.\ Ax), At \longrightarrow A_1, \Delta \end{array} \quad \otimes \quad \begin{array}{c} N_1 \\ \Gamma, (\forall x.\ Ax), At, A_1 \longrightarrow \Delta \end{array} \quad \Longrightarrow \quad \begin{array}{c} N_2 \\ \Gamma, (\forall x.\ Ax), At \longrightarrow \Delta \end{array}$$

```
car_existsr' :
  ca' A D ([n] existsr' T (E1 n) P) (existsr' T F1 P)
     <- ({p1} ca' A D ([n] E1 n p1) (F1 p1)).
```

$$\begin{array}{c} N \\ \Gamma \longrightarrow A_1, (\exists x.\ Ax), \Delta \end{array} \quad \otimes \quad \dfrac{\begin{array}{c} N_1 \\ \Gamma, A_1 \longrightarrow At, (\exists x.\ Ax), \Delta \end{array}}{\Gamma, A_1 \longrightarrow (\exists x.\ Ax), \Delta} \exists R \quad \Longrightarrow \quad \dfrac{\begin{array}{c} N_2 \\ \Gamma \longrightarrow At, (\exists x.\ Ax), \Delta \end{array}}{\Gamma \longrightarrow (\exists x.\ Ax), \Delta} \exists R$$

$$\begin{array}{c} N \\ \Gamma \longrightarrow A_1, At, (\exists x.\ Ax), \Delta \end{array} \quad \otimes \quad \begin{array}{c} N_1 \\ \Gamma, A_1 \longrightarrow At, (\exists x.\ Ax), \Delta \end{array} \quad \Longrightarrow \quad \begin{array}{c} N_2 \\ \Gamma \longrightarrow At, (\exists x.\ Ax), \Delta \end{array}$$

```
car_existsl' :
  ca' A D ([n] existsl' (E1 n) N) (existsl' F1 N)
     <- ({a:i} {n1:neg (B1 a)}
           ca' A D ([n] E1 n a n1) (F1 a n1)).
```

$$\begin{array}{c} N \\ \Gamma, (\exists x.\ Ax) \longrightarrow A_1, \Delta \end{array} \quad \otimes \quad \dfrac{\begin{array}{c} N_1 a_1 \\ \Gamma, (\exists x.\ Ax), A_1, Aa_1 \longrightarrow \Delta \end{array}}{\Gamma, (\exists x.\ Ax), A_1 \longrightarrow \Delta} \exists L^{a_1} \quad \Longrightarrow \quad \dfrac{\begin{array}{c} N_2 a_1 \\ \Gamma, (\exists x.\ Ax), Aa_1 \longrightarrow \Delta \end{array}}{\Gamma, (\exists x.\ Ax) \longrightarrow \Delta} \exists L^{a_1}$$

$$\begin{array}{c} N \\ \Gamma, (\exists x.\ Ax), Aa \longrightarrow A_1, \Delta \end{array} \quad \otimes \quad \begin{array}{c} N_1 a \\ \Gamma, (\exists x.\ Ax), Aa, A_1 \longrightarrow \Delta \end{array} \quad \Longrightarrow \quad \begin{array}{c} N_2 a \\ \Gamma, (\exists x.\ Ax), Aa \longrightarrow \Delta \end{array}$$

Left Commutative Conversions. Here the cut formula A is a side formula of the last inference in $\mathcal{D}$.

```
cal_axiom' : ca' A ([p] axiom' N P) E (axiom' N P).
```

$$\begin{array}{c} N \\ \Gamma, A_1 \longrightarrow A, A_1, \Delta \end{array} \quad \otimes \quad \dfrac{}{\Gamma, A_1, A \longrightarrow A_1, \Delta} I \quad \Longrightarrow \quad \dfrac{}{\Gamma, A_1 \longrightarrow A_1, \Delta} I$$

```
cal_andr' :
  ca' A ([p] andr' (D1 p) (D2 p) P) E (andr' F1 F2 P)
     <- ({p1:pos B1} ca' A ([p] D1 p p1) E (F1 p1))
     <- ({p2:pos B2} ca' A ([p] D2 p p2) E (F2 p2)).
```

$$\dfrac{\overset{N_3}{\Gamma \longrightarrow A_2, A_1, (A_2 \wedge A), \Delta} \qquad \overset{N}{\Gamma \longrightarrow A, A_1, (A_2 \wedge A), \Delta}}{\Gamma \longrightarrow A_1, (A_2 \wedge A), \Delta}\wedge R \quad \bigotimes \quad \overset{N_1}{\Gamma, A_1 \longrightarrow (A_2 \wedge A), \Delta}$$

$$\Longrightarrow \quad \dfrac{\overset{N_4}{\Gamma \longrightarrow A_2, (A_2 \wedge A), \Delta} \qquad \overset{N_2}{\Gamma \longrightarrow A, (A_2 \wedge A), \Delta}}{\Gamma \longrightarrow (A_2 \wedge A), \Delta}\wedge R$$

$$\overset{N_3}{\Gamma \longrightarrow A_1, A_2, (A_2 \wedge A), \Delta} \quad \bigotimes \quad \overset{N_1}{\Gamma, A_1 \longrightarrow A_2, (A_2 \wedge A), \Delta} \quad \Longrightarrow \quad \overset{N_4}{\Gamma \longrightarrow A_2, (A_2 \wedge A), \Delta}$$

$$\overset{N}{\Gamma \longrightarrow A_1, A, (A_2 \wedge A), \Delta} \quad \bigotimes \quad \overset{N_1}{\Gamma, A_1 \longrightarrow A, (A_2 \wedge A), \Delta} \quad \Longrightarrow \quad \overset{N_2}{\Gamma \longrightarrow A, (A_2 \wedge A), \Delta}$$

```
cal_andl1' :
  ca' A ([p] andl1' (D1 p) N) E (andl1' F1 N)
     <- ({n1:neg B1} ca' A ([p] D1 p n1) E (F1 n1)).
```

$$\dfrac{\overset{N}{\Gamma, (A \wedge A_2), A \longrightarrow A_1, \Delta}}{\Gamma, (A \wedge A_2) \longrightarrow A_1, \Delta}\wedge L_1 \quad \bigotimes \quad \overset{N_1}{\Gamma, (A \wedge A_2), A_1 \longrightarrow \Delta} \quad \Longrightarrow \quad \dfrac{\overset{N_2}{\Gamma, (A \wedge A_2), A \longrightarrow \Delta}}{\Gamma, (A \wedge A_2) \longrightarrow \Delta}\wedge L_1$$

$$\overset{N}{\Gamma, (A \wedge A_2), A \longrightarrow A_1, \Delta} \quad \bigotimes \quad \overset{N_1}{\Gamma, (A \wedge A_2), A, A_1 \longrightarrow \Delta} \quad \Longrightarrow \quad \overset{N_2}{\Gamma, (A \wedge A_2), A \longrightarrow \Delta}$$

```
cal_andl2' :
  ca' A ([p] andl2' (D2 p) N) E (andl2' F2 N)
     <- ({n2:neg B2} ca' A ([p] D2 p n2) E (F2 n2)).
```

$$\dfrac{\overset{N}{\Gamma, (A_2 \wedge A), A \longrightarrow A_1, \Delta}}{\Gamma, (A_2 \wedge A) \longrightarrow A_1, \Delta}\wedge L_2 \quad \bigotimes \quad \overset{N_1}{\Gamma, (A_2 \wedge A), A_1 \longrightarrow \Delta} \quad \Longrightarrow \quad \dfrac{\overset{N_2}{\Gamma, (A_2 \wedge A), A \longrightarrow \Delta}}{\Gamma, (A_2 \wedge A) \longrightarrow \Delta}\wedge L_2$$

$$\overset{N}{\Gamma, (A_2 \wedge A), A \longrightarrow A_1, \Delta} \quad \bigotimes \quad \overset{N_1}{\Gamma, (A_2 \wedge A), A, A_1 \longrightarrow \Delta} \quad \Longrightarrow \quad \overset{N_2}{\Gamma, (A_2 \wedge A), A \longrightarrow \Delta}$$

```
cal_impr' :
  ca' A ([p] impr' (D1 p) P) E (impr' F1 P)
     <- ({n1:neg B1} {p2:pos B2}
           ca' A ([p] D1 p n1 p2) E (F1 n1 p2)).
```

$$\dfrac{\overset{N}{\Gamma, A \longrightarrow A_1, A_2, (A \supset A_1), \Delta}}{\Gamma \longrightarrow A_2, (A \supset A_1), \Delta}\supset R \quad \otimes \quad \overset{N_1}{\Gamma, A_2 \longrightarrow (A \supset A_1), \Delta} \quad \Longrightarrow \quad \dfrac{\overset{N_2}{\Gamma, A \longrightarrow A_1, (A \supset A_1), \Delta}}{\Gamma \longrightarrow (A \supset A_1), \Delta}\supset R$$

$$\overset{N}{\Gamma, A \longrightarrow A_2, A_1, (A \supset A_1), \Delta} \quad \otimes \quad \overset{N_1}{\Gamma, A, A_2 \longrightarrow A_1, (A \supset A_1), \Delta} \quad \Longrightarrow \quad \overset{N_2}{\Gamma, A \longrightarrow A_1, (A \supset A_1), \Delta}$$

```
cal_impl' :
  ca' A ([p] impl' (D1 p) (D2 p) N) E (impl' F1 F2 N)
     <- ({p1:pos B1} ca' A ([p] D1 p p1) E (F1 p1))
     <- ({n2:neg B2} ca' A ([p] D2 p n2) E (F2 n2)).
```

$$\dfrac{\overset{N_3}{\Gamma, (A_2 \supset A) \longrightarrow A_2, A_1, \Delta} \qquad \overset{N}{\Gamma, (A_2 \supset A), A \longrightarrow A_1, \Delta}}{\Gamma, (A_2 \supset A) \longrightarrow A_1, \Delta}\supset L \quad \otimes \quad \overset{N_1}{\Gamma, (A_2 \supset A), A_1 \longrightarrow \Delta}$$

$$\Longrightarrow \quad \dfrac{\overset{N_4}{\Gamma, (A_2 \supset A) \longrightarrow A_2, \Delta} \qquad \overset{N_2}{\Gamma, (A_2 \supset A), A \longrightarrow \Delta}}{\Gamma, (A_2 \supset A) \longrightarrow \Delta}\supset L$$

$$\overset{N_3}{\Gamma, (A_2 \supset A) \longrightarrow A_1, A_2, \Delta} \quad \otimes \quad \overset{N_1}{\Gamma, (A_2 \supset A), A_1 \longrightarrow A_2, \Delta} \quad \Longrightarrow \quad \overset{N_4}{\Gamma, (A_2 \supset A) \longrightarrow A_2, \Delta}$$

$$\overset{N}{\Gamma, (A_2 \supset A), A \longrightarrow A_1, \Delta} \quad \otimes \quad \overset{N_1}{\Gamma, (A_2 \supset A), A, A_1 \longrightarrow \Delta} \quad \Longrightarrow \quad \overset{N_2}{\Gamma, (A_2 \supset A), A \longrightarrow \Delta}$$

```
cal_orr1' :
  ca' A ([p] orr1' (D1 p) P) E (orr1' F1 P)
     <- ({p1:pos B1} ca' A ([p] D1 p p1) E (F1 p1)).
```

$$\dfrac{\overset{N}{\Gamma \longrightarrow A, A_1, (A \vee A_2), \Delta}}{\Gamma \longrightarrow A_1, (A \vee A_2), \Delta}\vee R_1 \quad \otimes \quad \overset{N_1}{\Gamma, A_1 \longrightarrow (A \vee A_2), \Delta} \quad \Longrightarrow \quad \dfrac{\overset{N_2}{\Gamma \longrightarrow A, (A \vee A_2), \Delta}}{\Gamma \longrightarrow (A \vee A_2), \Delta}\vee R_1$$

$$\overset{N}{\Gamma \longrightarrow A_1, A, (A \vee A_2), \Delta} \quad \otimes \quad \overset{N_1}{\Gamma, A_1 \longrightarrow A, (A \vee A_2), \Delta} \quad \Longrightarrow \quad \overset{N_2}{\Gamma \longrightarrow A, (A \vee A_2), \Delta}$$

```
cal_orr2' :
  ca' A ([p] orr2' (D2 p) P) E (orr2' F2 P)
     <- ({p2:pos B2} ca' A ([p] D2 p p2) E (F2 p2)).
```

$$\dfrac{\overset{N}{\Gamma \longrightarrow A, A_1, (A_2 \vee A), \Delta}}{\Gamma \longrightarrow A_1, (A_2 \vee A), \Delta}\vee R_2 \quad \bigotimes \quad \overset{N_1}{\Gamma, A_1 \longrightarrow (A_2 \vee A), \Delta} \quad \Longrightarrow \quad \dfrac{\overset{N_2}{\Gamma \longrightarrow A, (A_2 \vee A), \Delta}}{\Gamma \longrightarrow (A_2 \vee A), \Delta}\vee R_2$$

$$\overset{N}{\Gamma \longrightarrow A_1, A, (A_2 \vee A), \Delta} \quad \bigotimes \quad \overset{N_1}{\Gamma, A_1 \longrightarrow A, (A_2 \vee A), \Delta} \quad \Longrightarrow \quad \overset{N_2}{\Gamma \longrightarrow A, (A_2 \vee A), \Delta}$$

```
cal_orl' :
  ca' A ([p] orl' (D1 p) (D2 p) N) E (orl' F1 F2 N)
     <- ({n1:neg B1} ca' A ([p] D1 p n1) E (F1 n1))
     <- ({n2:neg B2} ca' A ([p] D2 p n2) E (F2 n2)).
```

$$\dfrac{\overset{N_3}{\Gamma, (A_2 \vee A), A_2 \longrightarrow A_1, \Delta} \qquad \overset{N}{\Gamma, (A_2 \vee A), A \longrightarrow A_1, \Delta}}{\Gamma, (A_2 \vee A) \longrightarrow A_1, \Delta}\vee L \quad \bigotimes \quad \overset{N_1}{\Gamma, (A_2 \vee A), A_1 \longrightarrow \Delta}$$

$$\Longrightarrow \quad \dfrac{\overset{N_4}{\Gamma, (A_2 \vee A), A_2 \longrightarrow \Delta} \qquad \overset{N_2}{\Gamma, (A_2 \vee A), A \longrightarrow \Delta}}{\Gamma, (A_2 \vee A) \longrightarrow \Delta}\vee L$$

$$\overset{N_3}{\Gamma, (A_2 \vee A), A_2 \longrightarrow A_1, \Delta} \quad \bigotimes \quad \overset{N_1}{\Gamma, (A_2 \vee A), A_2, A_1 \longrightarrow \Delta} \quad \Longrightarrow \quad \overset{N_4}{\Gamma, (A_2 \vee A), A_2 \longrightarrow \Delta}$$

$$\overset{N}{\Gamma, (A_2 \vee A), A \longrightarrow A_1, \Delta} \quad \bigotimes \quad \overset{N_1}{\Gamma, (A_2 \vee A), A, A_1 \longrightarrow \Delta} \quad \Longrightarrow \quad \overset{N_2}{\Gamma, (A_2 \vee A), A \longrightarrow \Delta}$$

```
cal_notr' :
  ca' A ([p] notr' (D1 p) P) E (notr' F1 P)
     <- ({n1:neg B1} ca' A ([p] D1 p n1) E (F1 n1)).
```

$$\dfrac{\overset{N}{\Gamma, A \longrightarrow A_1, \neg(A), \Delta}}{\Gamma \longrightarrow A_1, \neg(A), \Delta}\neg R \quad \bigotimes \quad \overset{N_1}{\Gamma, A_1 \longrightarrow \neg(A), \Delta} \quad \Longrightarrow \quad \dfrac{\overset{N_2}{\Gamma, A \longrightarrow \neg(A), \Delta}}{\Gamma \longrightarrow \neg(A), \Delta}\neg R$$

$$\overset{N}{\Gamma, A \longrightarrow A_1, \neg(A), \Delta} \quad \bigotimes \quad \overset{N_1}{\Gamma, A, A_1 \longrightarrow \neg(A), \Delta} \quad \Longrightarrow \quad \overset{N_2}{\Gamma, A \longrightarrow \neg(A), \Delta}$$

```
cal_notl' :
  ca' A ([p] notl' (D1 p) N) E (notl' F1 N)
     <- ({p1:pos B1} ca' A ([p] D1 p p1) E (F1 p1)).
```

$$\dfrac{\overset{N}{\Gamma, \neg(A) \longrightarrow A, A_1, \Delta}}{\Gamma, \neg(A) \longrightarrow A_1, \Delta}\neg L \quad \bigotimes \quad \overset{N_1}{\Gamma, \neg(A), A_1 \longrightarrow \Delta} \quad \Longrightarrow \quad \dfrac{\overset{N_2}{\Gamma, \neg(A) \longrightarrow A, \Delta}}{\Gamma, \neg(A) \longrightarrow \Delta}\neg L$$

$$\overset{N}{\Gamma, \neg(A) \longrightarrow A_1, A, \Delta} \quad \bigotimes \quad \overset{N_1}{\Gamma, \neg(A), A_1 \longrightarrow A, \Delta} \quad \Longrightarrow \quad \overset{N_2}{\Gamma, \neg(A) \longrightarrow A, \Delta}$$

```
cal_truer' :
  ca' A ([p] truer' P) E (truer' P).
```

$$\dfrac{}{\Gamma \longrightarrow A, \top, \Delta}\top R \quad \bigotimes \quad \overset{N}{\Gamma, A \longrightarrow \top, \Delta} \quad \Longrightarrow \quad \dfrac{}{\Gamma \longrightarrow \top, \Delta}\top R$$

```
cal_falsel' :
  ca' A ([p] falsel' N) E (falsel' N).
```

$$\dfrac{}{\Gamma, \bot \longrightarrow A, \Delta}\bot L \quad \bigotimes \quad \overset{N}{\Gamma, \bot, A \longrightarrow \Delta} \quad \Longrightarrow \quad \dfrac{}{\Gamma, \bot \longrightarrow \Delta}\bot L$$

```
cal_forallr' :
  ca' A ([p] forallr' (D1 p) P) E (forallr' F1 P)
     <- ({a:i} {p1:pos (B1 a)}
           ca' A ([p] D1 p a p1) E (F1 a p1)).
```

$$\dfrac{\overset{N a_1}{\Gamma \longrightarrow A a_1, A_1, (\forall x.\, A x), \Delta}}{\Gamma \longrightarrow A_1, (\forall x.\, A x), \Delta}\forall R^{a_1} \quad \bigotimes \quad \overset{N_1}{\Gamma, A_1 \longrightarrow (\forall x.\, A x), \Delta} \quad \Longrightarrow \quad \dfrac{\overset{N_2 a_1}{\Gamma \longrightarrow A a_1, (\forall x.\, A x), \Delta}}{\Gamma \longrightarrow (\forall x.\, A x), \Delta}\forall R^{a_1}$$

$$\overset{N a}{\Gamma \longrightarrow A_1, A a, (\forall x.\, A x), \Delta} \quad \bigotimes \quad \overset{N_1}{\Gamma, A_1 \longrightarrow A a, (\forall x.\, A x), \Delta} \quad \Longrightarrow \quad \overset{N_2 a}{\Gamma \longrightarrow A a, (\forall x.\, A x), \Delta}$$

```
cal_foralll' :
  ca' A ([p] foralll' T (D1 p) N) E (foralll' T F1 N)
     <- ({n1} ca' A ([p] D1 p n1) E (F1 n1)).
```

$$\dfrac{\begin{array}{c}N\\ \Gamma, (\forall x.\, Ax), At \longrightarrow A_1, \Delta\end{array}}{\Gamma, (\forall x.\, Ax) \longrightarrow A_1, \Delta}\forall L \quad \otimes \quad \begin{array}{c}N_1\\ \Gamma, (\forall x.\, Ax), A_1 \longrightarrow \Delta\end{array} \quad \Longrightarrow \quad \dfrac{\begin{array}{c}N_2\\ \Gamma, (\forall x.\, Ax), At \longrightarrow \Delta\end{array}}{\Gamma, (\forall x.\, Ax) \longrightarrow \Delta}\forall L$$

$$\begin{array}{c}N\\ \Gamma, (\forall x.\, Ax), At \longrightarrow A_1, \Delta\end{array} \quad \otimes \quad \begin{array}{c}N_1\\ \Gamma, (\forall x.\, Ax), At, A_1 \longrightarrow \Delta\end{array} \quad \Longrightarrow \quad \begin{array}{c}N_2\\ \Gamma, (\forall x.\, Ax), At \longrightarrow \Delta\end{array}$$

```
cal_existsr' :
  ca' A ([p] existsr' T (D1 p) P) E (existsr' T F1 P)
     <- ({p1} ca' A ([p] D1 p p1) E (F1 p1)).
```

$$\dfrac{\begin{array}{c}N\\ \Gamma \longrightarrow At, A_1, (\exists x.\, Ax), \Delta\end{array}}{\Gamma \longrightarrow A_1, (\exists x.\, Ax), \Delta}\exists R \quad \otimes \quad \begin{array}{c}N_1\\ \Gamma, A_1 \longrightarrow (\exists x.\, Ax), \Delta\end{array} \quad \Longrightarrow \quad \dfrac{\begin{array}{c}N_2\\ \Gamma \longrightarrow At, (\exists x.\, Ax), \Delta\end{array}}{\Gamma \longrightarrow (\exists x.\, Ax), \Delta}\exists R$$

$$\begin{array}{c}N\\ \Gamma \longrightarrow A_1, At, (\exists x.\, Ax), \Delta\end{array} \quad \otimes \quad \begin{array}{c}N_1\\ \Gamma, A_1 \longrightarrow At, (\exists x.\, Ax), \Delta\end{array} \quad \Longrightarrow \quad \begin{array}{c}N_2\\ \Gamma \longrightarrow At, (\exists x.\, Ax), \Delta\end{array}$$

```
cal_existsl' :
  ca' A ([p] existsl' (D1 p) N) E (existsl' F1 N)
     <- ({a:i} {n1:neg (B1 a)}
           ca' A ([p] D1 p a n1) E (F1 a n1)).
```

$$\dfrac{\begin{array}{c}N a_1\\ \Gamma, (\exists x.\, Ax), Aa_1 \longrightarrow A_1, \Delta\end{array}}{\Gamma, (\exists x.\, Ax) \longrightarrow A_1, \Delta}\exists L^{a_1} \quad \otimes \quad \begin{array}{c}N_1\\ \Gamma, (\exists x.\, Ax), A_1 \longrightarrow \Delta\end{array} \quad \Longrightarrow \quad \dfrac{\begin{array}{c}N_2 a_1\\ \Gamma, (\exists x.\, Ax), Aa_1 \longrightarrow \Delta\end{array}}{\Gamma, (\exists x.\, Ax) \longrightarrow \Delta}\exists L^{a_1}$$

$$\begin{array}{c}N a\\ \Gamma, (\exists x.\, Ax), Aa \longrightarrow A_1, \Delta\end{array} \quad \otimes \quad \begin{array}{c}N_1\\ \Gamma, (\exists x.\, Ax), Aa, A_1 \longrightarrow \Delta\end{array} \quad \Longrightarrow \quad \begin{array}{c}N_2 a\\ \Gamma, (\exists x.\, Ax), Aa \longrightarrow \Delta\end{array}$$

B Cut Elimination

In this appendix we define intuitionistic and classical sequent calculi with a primitive rule of cut and show that they can be translated to cut-free derivations. In both cases the proof is a straightforward induction on the structure of derivations, taking advantage of admissibility of cut in the cut-free system.

B.1 Intuitionistic Calculus

We use $\Gamma \longrightarrow_+ C$ for sequents in the system G_3^+ with cut which is obtained by adding

$$\frac{\Gamma \longrightarrow_+ A \qquad \Gamma, A \longrightarrow_+ C}{\Gamma \longrightarrow_+ C}\,\mathrm{Cut}$$

to the rules of the cut-free system G_3. With proof terms this rule reads

$$\frac{\Gamma \longrightarrow_+ d : A \qquad \Gamma, h{:}A \longrightarrow_+ e : C}{\Gamma \longrightarrow_+ \mathrm{cut}\, d\, (\lambda h{:}A.\ e) : C}\,\mathrm{Cut}.$$

In order to represent derivations in G_3^+ we introduce another judgment, `conc* A`, rename all the rules for the cut-free calculus and add

```
cut* : {A:o} conc* A
       -> (hyp A -> conc* C)
       -> conc* C.
```

The complete implementation is given below. Note that we do not need to rename the hypothesis judgment `hyp`, since hypothesis play the same role in both systems. The main lemmas concerning G_3^+ such as weakening, contraction, substitution, and the adequacy of the encoding follow as before.

```
conc* : o -> type.

cut* : {A:o} conc* A
        -> (hyp A -> conc* C)
        -> conc* C.

axiom* : (hyp A -> conc* A).

andr*  : conc* A
          -> conc* B
          -> conc* (A and B).

andl1* : (hyp A -> conc* C)
          -> (hyp (A and B) -> conc* C).

andl2* : (hyp B -> conc* C)
          -> (hyp (A and B) -> conc* C).

impr*  : (hyp A -> conc* B)
          -> conc* (A imp B).

impl*  : conc* A
          -> (hyp B -> conc* C)
          -> (hyp (A imp B) -> conc* C).

orr1*  : conc* A
          -> conc* (A or B).
```

```
orr2*  : conc* B
          -> conc* (A or B).

orl*   : (hyp A -> conc* C)
          -> (hyp B -> conc* C)
          -> (hyp (A or B) -> conc* C).

notr*  : ({p:o} hyp A -> conc* p)
         -> conc* (not A).

notl*  : conc* A
         -> (hyp (not A) -> conc* C).

truer* : conc* (true).

falsel* : (hyp (false) -> conc* C).

forallr* : ({a:i} conc* (A a))
            -> conc* (forall A).

foralll* : {T:i} (hyp (A T) -> conc* C)
            -> (hyp (forall A) -> conc* C).

existsr* : {T:i} conc* (A T)
            -> conc* (exists A).

existsl* : ({a:i} hyp (A a) -> conc* C)
            -> (hyp (exists A) -> conc* C).
```

The theorem of cut elimination explicitly relates derivations in G_3^+ to G_3.

Theorem 8 (Cut Elimination) *If $\mathcal{D}^* :: (\Gamma \longrightarrow_+ d^* : C)$ is a derivation in G_3^+ then there exists a cut-free derivation $\mathcal{D} :: (\Gamma \longrightarrow d : C)$ in G_3.*

Proof: By structural induction on d. When the last inference R is not a cut we appeal to the induction hypothesis on the premise(s) of the last inference and combine the resulting cut-free derivation(s) with R. If the last inference is a cut we generate cut-free derivations of the premises by induction hypothesis and then use admissibility of cut to obtain a cut-free derivation for the conclusion. □

This proof is implemented by a relation between sequent derivations in the system with cut and sequent derivations in the system without cut. We use the convention that variables whose name ends in a star (*) represent derivations that may contain cut.

```
ce : conc* C -> conc C -> type.

ce_cut   : ce (cut* A D1* D2*) D
            <- ce D1* D1
            <- ({h1:hyp A} ce (D2* h1) (D2 h1))
            <- ca A D1 D2 D.
```

```
ce_axiom : ce (axiom* H) (axiom H).

ce_andr  : ce (andr* D1* D2*) (andr D1 D2)
            <- ce D1* D1
            <- ce D2* D2.

ce_andl1 : ce (andl1* D1* H) (andl1 D1 H)
            <- ({h1:hyp A} ce (D1* h1) (D1 h1)).

ce_andl2 : ce (andl2* D2* H) (andl2 D2 H)
            <- ({h2:hyp B} ce (D2* h2) (D2 h2)).

ce_impr  : ce (impr* D1*) (impr D1)
            <- ({h1:hyp A} ce (D1* h1) (D1 h1)).

ce_impl  : ce (impl* D1* D2* H) (impl D1 D2 H)
            <- ce D1* D1
            <- ({h2:hyp B} ce (D2* h2) (D2 h2)).

ce_orr1  : ce (orr1* D1*) (orr1 D1)
            <- ce D1* D1.

ce_orr2  : ce (orr2* D2*) (orr2 D2)
            <- ce D2* D2.

ce_orl   : ce (orl* D1* D2* H) (orl D1 D2 H)
            <- ({h1:hyp A} ce (D1* h1) (D1 h1))
            <- ({h2:hyp B} ce (D2* h2) (D2 h2)).

ce_notr  : ce (notr* D1*) (notr D1)
            <- ({p:o} {h1:hyp A} ce (D1* p h1) (D1 p h1)).

ce_notl  : ce (notl* D1* H) (notl D1 H)
            <- ce D1* D1.

ce_truer : ce (truer*) (truer).

ce_falsel : ce (falsel* H) (falsel H).

ce_forallr : ce (forallr* D1*) (forallr D1)
              <- {a:i} ce (D1* a) (D1 a).

ce_foralll : ce (foralll* T D1* H) (foralll T D1 H)
              <- ({h1} ce (D1* h1) (D1 h1)).

ce_existsr : ce (existsr* T D1*) (existsr T D1)
              <- ce D1* D1.

ce_existsl : ce (existsl* D1* H) (existsl D1 H)
              <- ({a:i} {h1:hyp (A1 a)} ce (D1* a h1) (D1 a h1)).
```

B.2 Classical Calculus

We write $\Gamma \longrightarrow_+ \Delta$ for a sequent in the classical system with cut as a primitive rule of inference. It is obtained by adding

$$\frac{\Gamma \longrightarrow_+ A, \Delta \qquad \Gamma, A \longrightarrow_+ \Delta}{\Gamma \longrightarrow_+ \Delta}\,\mathrm{Cut}$$

to the other rules of inference. With proof terms we have

$$\frac{\Gamma \xrightarrow{d}_+ p{:}A, \Delta \qquad \Gamma, n{:}A \xrightarrow{e}_+ \Delta}{\Gamma \xrightarrow{\mathrm{cut}\,(\lambda p{:}A.\ d)\,(\lambda n{:}A.\ e)}_+ \Delta}\,\mathrm{Cut}$$

This system continues to satisfy weakening, contraction, and substitution lemmas. The signature below specifies an adequate encoding of this extended calculus in LF. We need a new judgment `@` that replaces `#` as the type of proof terms in the representation. We systematically copy all declarations from the cut-free system (appending `^` to their name) and add the cut rule as `cut^`.

```
@ : type.

cut^ : (pos A -> @)
        -> (neg A -> @)
        -> @.

axiom^ : (neg A -> pos A -> @).

andr^  : (pos A -> @)
          -> (pos B -> @)
          -> (pos (A and B) -> @).

andl1^ : (neg A -> @)
          -> (neg (A and B) -> @).

andl2^ : (neg B -> @)
          -> (neg (A and B) -> @).

impr^  : (neg A -> pos B -> @)
          -> (pos (A imp B) -> @).

impl^  : (pos A -> @)
          -> (neg B -> @)
          -> (neg (A imp B) -> @).

orr1^  : (pos A -> @)
         -> (pos (A or B) -> @).

orr2^  : (pos B -> @)
          -> (pos (A or B) -> @).

orl^   : (neg A -> @)
```

```
         -> (neg B -> @)
         -> (neg (A or B) -> @).

notr^  : (neg A -> @)
          -> (pos (not A) -> @).

notl^  : (pos A -> @)
          -> (neg (not A) -> @).

truer^ : (pos (true) -> @).

falsel^ : (neg (false) -> @).

forallr^ : ({a:i} pos (A a) -> @)
            -> (pos (forall A) -> @).

foralll^ : {T:i} (neg (A T) -> @)
            -> (neg (forall A) -> @).

existsr^ : {T:i} (pos (A T) -> @)
            -> (pos (exists A) -> @).

existsl^ : ({a:i} neg (A a) -> @)
            -> (neg (exists A) -> @).
```

Cut elimination follows by a simple structural induction from the admissibility of cut in the cut-free system. We present here only the Elf code implementing this proof.

Theorem 9 (Classical Cut Elimination) *Let $\mathcal{D} :: (\Gamma \xrightarrow{d}_{+} \Delta)$ be a classical sequent derivation possibly containing cut. Then there exists a cut-free derivation $\mathcal{D}' :: (\Gamma \xrightarrow{d'} \Delta)$*

Proof: By structural induction on d. For each inference rule except cut we apply the induction hypothesis to the premises and then reconstruct a cut-free derivation with the same inference rule. In the case of cut, we appeal to the induction hypothesis and then to admissibility of cut on the resulting two cut-free derivations. □

This proof is implementated as a type family relating derivations with cut to cut-free derivations. Note how the appeal to admissibility in the case of a cut is implemented as a call to `ca'`.

```
ce' : @ -> # -> type.

ce_cut'  : ce' (cut^ D E) F
            <- ({p:pos A} ce' (D p) (D' p))
            <- ({n:neg A} ce' (E n) (E' n))
            <- ca' A D' E' F.

ce_axiom': ce' (axiom^ N P) (axiom' N P).

ce_andr' : ce' (andr^ D1 D2 P) (andr' D1' D2' P)
            <- ({p1} ce' (D1 p1) (D1' p1))
```

```
            <- ({p2} ce' (D2 p2) (D2' p2)).

ce_andl1': ce' (andl1^ N1 N) (andl1' N1' N)
            <- ({n1} ce' (N1 n1) (N1' n1)).

ce_andl2': ce' (andl2^ N2 N) (andl2' N2' N)
            <- ({n2} ce' (N2 n2) (N2' n2)).

ce_impr' : ce' (impr^ D1 P) (impr' D1' P)
            <- ({n1}{p2} ce' (D1 n1 p2) (D1' n1 p2)).

ce_impl' : ce' (impl^ D1 D2 N) (impl' D1' D2' N)
            <- ({p1} ce' (D1 p1) (D1' p1))
            <- ({n2} ce' (D2 n2) (D2' n2)).

ce_orr1' : ce' (orr1^ D1 P) (orr1' D1' P)
            <- ({p1} ce' (D1 p1) (D1' p1)).

ce_orr2' : ce' (orr2^ D2 P) (orr2' D2' P)
            <- ({p2} ce' (D2 p2) (D2' p2)).

ce_orl'  : ce' (orl^ D1 D2 N) (orl' D1' D2' N)
            <- ({n1} ce' (D1 n1) (D1' n1))
            <- ({n2} ce' (D2 n2) (D2' n2)).

ce_notr' : ce' (notr^ D1 P) (notr' D1' P)
            <- ({n1} ce' (D1 n1) (D1' n1)).

ce_norl' : ce' (notl^ D1 N) (notl' D1' N)
            <- ({p1} ce' (D1 p1) (D1' p1)).

ce_truer': ce' (truer^ P) (truer' P).

ce_falsel': ce' (falsel^ N) (falsel' N).

ce_forallr': ce' (forallr^ D1 P) (forallr' D1' P)
              <- ({a:i} {p1:pos (A1 a)} ce' (D1 a p1) (D1' a p1)).

ce_foralll': ce' (foralll^ T D1 N) (foralll' T D1' N)
              <- ({n1} ce' (D1 n1) (D1' n1)).

ce_existsr': ce' (existsr^ T D1 P) (existsr' T D1' P)
              <- ({p1} ce' (D1 p1) (D1' p1)).

ce_existsl': ce' (existsl^ D1 N) (existsl' D1' N)
              <- ({a:i} {n1:neg (A1 a)} ce' (D1 a n1) (D1' a n1)).
```

Acknowledgments

I would like to thank Iliano Cervesato, Jean Gallier, Dale Miller, Richard Statman and Roberto Virga for discussions regarding the topic of this paper and David Basin and Sean Matthews for comments on a draft.

References

[Abr93] Samson Abramsky. Computational interpretations of linear logic. *Theoretical Computer Science*, 111:3–57, 1993.

[BTKP93] Val Breazu-Tannen, Delia Kesner, and Laurence Puel. A typed pattern calculus. In Moshe Y. Vardi, editor, *Proceedings of the Eight Annual IEEE Symposium on Logic in Computer Science*, pages 262–274, Montreal, Canada, June 1993.

[DFH+93] Gilles Dowek, Amy Felty, Hugo Herbelin, Gérard Huet, Chet Murthy, Catherine Parent, Christine Paulin-Mohring, and Benjamin Werner. The Coq proof assistant user's guide. Rapport Techniques 154, INRIA, Rocquencourt, France, 1993. Version 5.8.

[DH94] Joëlle Despeyroux and André Hirschowitz. Higher-order abstract syntax with induction in Coq. In Frank Pfenning, editor, *Proceedings of the 5th International Conference on Logic Programming and Automated Reasoning*, pages 159–173, Kiev, Ukraine, July 1994. Springer-Verlag LNAI 822.

[Fel89] Amy Felty. *Specifying and Implementing Theorem Provers in a Higher-Order Logic Programming Language*. PhD thesis, University of Pennsylvania, August 1989. Available as Technical Report MS-CIS-89-53.

[Gal93] Jean Gallier. Constructive logics Part I: A tutorial on proof systems and typed λ-calculi. *Theoretical Computer Science*, 110:249–339, 1993.

[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.

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

[Gir93] Jean-Yves Girard. On the unity of logic. *Annals of Pure and Applied Logic*, 59:201–217, 1993.

[HHP93] Robert Harper, Furio Honsell, and Gordon Plotkin. A framework for defining logics. *Journal of the Association for Computing Machinery*, 40(1):143–184, January 1993.

[Kle52] Stephen Cole Kleene. *Introduction to Metamathematics*. North-Holland, 1952.

[Mat94] Seán Matthews. A theory and its metatheory in FS_0. In Dov Gabbay and Franz Guenthner, editors, *What is a Logical System?* Oxford University Press, 1994. To appear.

[Mil94] Dale Miller. A multiple-conclusion meta-logic. In S. Abramsky, editor, *Ninth Annual IEEE Symposium on Logic in Computer Science*, pages 272–281, Paris, France, July 1994.

[MNPS91] Dale Miller, Gopalan Nadathur, Frank Pfenning, and Andre Scedrov. Uniform proofs as a foundation for logic programming. *Annals of Pure and Applied Logic*, 51:125–157, 1991.

[MP91] Spiro Michaylov and Frank Pfenning. Natural semantics and some of its meta-theory in Elf. In L.-H. Eriksson, L. Hallnäs, and P. Schroeder-Heister, editors, *Proceedings of the Second International Workshop on Extensions of Logic Programming*, pages 299–344, Stockholm, Sweden, January 1991. Springer-Verlag LNAI 596.

[Nip91] Tobias Nipkow. Higher-order critical pairs. In G. Kahn, editor, *Sixth Annual IEEE Symposium on Logic in Computer Science*, pages 342–349, Amsterdam, The Netherlands, July 1991.

[Pfe91] Frank Pfenning. Logic programming in the LF logical framework. In Gérard Huet and Gordon Plotkin, editors, *Logical Frameworks*, pages 149–181. Cambridge University Press, 1991.

[Roh94] Ekkehard Rohwedder. Verifying the meta-theory of deductive systems. Thesis Proposal, February 1994.

[Wal90] Lincoln A. Wallen. *Automated Proof Search in Non-Classical Logics: Efficient Matrix Proof Methods for Modal and Intuitionistic Logics.* MIT Press, 1990.

[Zuc74] Jeffrey Zucker. The correspondence between cut-elimination and normalization I. *Annals of Mathematical Logic*, 7:1–112, 1974.