How to Universally Close the Existential Rule

Kai Brünnler

Institut für angewandte Mathematik und Informatik
Universität Bern, Neubrückstr. 10, CH – 3012 Bern, Switzerland
http://www.iam.unibe.ch/~kai/

Abstract This paper introduces a nested sequent system for predicate logic. The system features a structural universal quantifier and a universally closed existential rule. One nice consequence of this is that proofs of sentences cannot contain free variables. Another nice consequence is that the assumption of a non-empty domain is isolated in a single inference rule. This rule can be removed or added at will, leading to a system for free logic or classical predicate logic, respectively. The system for free logic is interesting because it has no need for an existence predicate. We see syntactic cut-elimination and completeness results for these two systems as well as two standard applications: Herbrand's Theorem and interpolation.

Table of Contents

1 Introduction

The mismatch. Traditional analytic proof systems like Gentzen's sequent calculus often cannot capture non-classical logics. Various formalisms have been designed to overcome this problem. The most prominent ones seem to be hypersequents [1], the display calculus [3], labelled systems [18], the calculus of structures [12], and nested sequents [5, 6]. All these formalisms work by enriching the structural level of proofs. We can thus see the problem of the traditional sequent calculus as a *mismatch* between the logic and the structural level of proofs, and we can see these formalisms as ways of repairing it. See also the note [11] by Guglielmi for an exposition of this mismatch.

The mismatch in predicate logic. Our proposition here is that the mismatch even affects sequent systems for classical predicate logic. Technically, there is obviously no match between structural and logical connectives, in particular there is no structural counterpart for quantification. The question is whether that is a problem. It turns out that it is if we either want to extend the logic, such as to modal predicate logic, or to slightly weaken the logic, such as by admitting models with an empty domain.

It is a well-known problem in modal predicate logic that combining a traditional sequent system for modal logic with one for predicate logic results in a system which forces the provability of the converse Barcan formula. The existential (right) rule is responsible for that. As it happens, the same existential rule forces the provability of the formula $\forall x A \supset \exists x A$ and thus restricts us to non-empty domains.

Repairing the mismatch. We now take a quick look at Hilbert-style axiom systems, because there these two problems also occur, but have an elegant solution. The analogue of the existential rule, and equally problematic, is the axiom of instantiation. It typically has the form:

$$\forall x A \supset A[x := y] \quad ,$$

where it suffices to have a variable y instead of an arbitrary term because we have no function symbols. This axiom can be universally closed as follows:

$$\forall y(\forall x A \supset A[x := y]) \quad .$$

And indeed, this closed form of instantiation leads to an axiomatisation for modal predicate logic which does not force the converse Barcan formula and is thus modular in the sense that we can add or remove the converse Barcan formula at will. It also leads to an axiomatisation for predicate logic which does not force non-empty domains, so we can add or remove the non-emptyness requirement at will. This trick of universally closing the instantiation axiom is attributed to Kripke in the context of modal predicate logic in [8] and to Lambert in the context of free logic in [4]. The purpose of the present work is essentially to bring this trick to cut-free systems.

Nested sequents. To that end, we use nested sequents. Compared to usual sequents, nested sequents simply add more structural counterparts for logical connectives. In modal logic, for example, in addition to having the comma (on the right) as a structural counterpart for disjunction, one also has a structural connective for the modality. Since in the present work we are concerned with predicate logic, we have a structural counterpart for universal quantification.

The first use of nested sequents under that name seems to be by Kashima in [13] for tense logics. However, the concept is very natural, and it has been used independently many times in different places. The earliest references I am aware of are from the seventies, by Dunn [7] and by Mints [14], on proof systems for relevance logics. More recent uses for modal logics can be found in the work by the author [5], by Goré, Tiu and Postniece [10], by Dyckhoff and Sadrzadeh [16], or by Poggiolesi [15].

Why repair the mismatch? For now, repairing the mismatch gives us a proof system in which the assumption of a non-empty domain is isolated in a single inference rule. This rule can be removed or added at will, leading to a system for free logic or classical predicate logic, respectively. This is of course not possible in the standard sequent calculus. The system for free logic is interesting because it does not rely on a so-called existence predicate, contrary to traditional sequent systems for free logic. This allows for a syntactic proof of an interpolation theorem, which is slightly stronger than what can be (easily) obtained from the traditional systems: the existence predicate does not occur in the interpolant. However, this is just for now. I think that repairing the mismatch becomes more fruitful the further we move away from classical logic. I am currently working on proof systems for modal predicate logic, in which both the Barcan and converse Barcan formula can be added in a modular fashion. I also think that nested sequents can provide proof systems for logics which currently do not seem to have cut-free systems such as (first-order) Gödel-logic without constant domains, see for example [2].

Outline. The outline of this paper is as follows. In Section 2 we introduce our two nested sequent systems: System Q for predicate logic and a subsystem of it, called System FQ, for free logic. In Section 3 we show some basic properties such as invertibility and some admissible rules. Section 4 relates System Q to a traditional system for predicate logic, and Section 5 relates System FQ to traditional proof systems for free logic. In Section 6 we prove cut-elimination. Finally, in Section 7 we show two simple consequences of cut-admissibility: Herbrand's Theorem and interpolation.

2 The Sequent Systems

Formulas. We assume given an infinite set of predicate symbols for each arity $n \geq 0$ and an infinite set of variables. Predicate symbols are denoted by P, variables by x, y, z. A *proposition*, denoted by p, is an expression $P(x_1, \ldots, x_n)$ where P is a predicate symbol of arity n and $x_1, \ldots, x_n$ are variables. *Formulas*, denoted by A, B, C, D are given by the grammar

$$A ::= p \mid \bar{p} \mid (A \vee A) \mid (A \wedge A) \mid \exists x A \mid \forall x A \quad .$$

Propositions p and their *negations* $\bar{p}$ are called *atoms*. Given a formula A, its *negation*

$\bar{A}$ is defined as usual using the double negation law and the De Morgan laws, $A \supset B$ is defined as $\bar{A} \vee B$ and both $\top$ and $\bot$ are respectively defined as $p \vee \bar{p}$ and $p \wedge \bar{p}$ for some proposition p. The binary connectives are left-associative, and we drop parentheses whenever possible, so for example $A \vee B \vee C$ denotes $((A \vee B) \vee C)$.

Nested sequents. A *nested sequent* is defined inductively as one of the following: 1) a finite multiset of formulas, 2) the singleton multiset containing the expression $\forall x[\Gamma]$ where Γ is a nested sequent, or 3) the multiset union of two nested sequents. The expression $\forall x[\ ]$ in $\forall x[\Gamma]$ is called the *structural universal quantifier*: the intention is that it relates to the universal quantifier just like comma relates to disjunction. It binds variables in the same way as usual quantifiers, these variables are then called *structurally bound*. We will often save brackets and write $\forall x\Gamma$ instead of $\forall x[\Gamma]$ and $\forall xy\Gamma$ instead of $\forall x\forall y\Gamma$ if it does not lead to confusion. In the following, a *sequent* is a nested sequent. Sequents are denoted by Γ and Δ. We adopt the usual notational conventions for sequents, in particular the comma in the expression Γ, Δ is multiset union. A sequent Γ is always of the form

$$A_1, \dots, A_m, \forall x_1[\Delta_1], \dots, \forall x_n[\Delta_n] \quad .$$

The *corresponding formula* $\underline{\Gamma}_{\mathsf{F}}$ of the above sequent is

$$A_1 \vee \dots \vee A_m \vee \forall x_1 \underline{\Delta_1}_{\mathsf{F}} \vee \dots \vee \forall x_n \underline{\Delta_n}_{\mathsf{F}} \quad ,$$

where an empty disjunction is $\bot$ and both the shown formulas and the shown variables are ordered according to some fixed total order.

Structural α-equivalence. Two sequents are *structurally α-equivalent* if they are equal up to the naming of structurally bound variables. This is a weaker equivalence than the usual notion of α-equivalence, which identifies formulas and sequents up to the naming of all bound variables. In particular, given different variables x and y, the sequents $\forall x[A]$ and $\forall y[A[x := y]]$ are structurally α-equivalent while the sequents $\forall xA$ and $\forall yA[x := y]$ are not. Our inference rules will apply modulo structural α-equivalence.

Sequent contexts. Informally, a context is a sequent with holes. We will mostly encounter sequents with just one hole. A *unary context* is a sequent with exactly one occurrence of the symbol $\{\ \}$, the *hole*, which does not occur inside formulas. Such contexts are denoted by $\Gamma\{\ \}$, $\Delta\{\ \}$, and so on. The hole is also called the *empty context*. The sequent $\Gamma\{\Delta\}$ is obtained by replacing $\{\ \}$ inside $\Gamma\{\ \}$ by Δ. For example, if $\Gamma\{\ \} = A, \forall x[B, \{\ \}]$ and $\Delta = C, D$ then $\Gamma\{\Delta\} = A, \forall x[B, C, D]$. More generally, a *context* is a sequent with $n \geq 0$ occurrences of $\{\ \}$, which do not occur inside formulas, and which are linearly ordered. A context with n holes is denoted by $\Gamma\underbrace{\{\ \}\dots\{\ \}}_{n-\text{times}}$.

Holes can be filled with sequents, or contexts, in general. For example, if $\Gamma\{\ \}\{\ \} = A, \forall x[B, \{\ \}], \{\ \}$ and $\Delta\{\ \} = C, \{\ \}$ then

$$\Gamma\{\Delta\{\ \}\}\{\ \} = A, \forall x[B, C, \{\ \}], \{\ \} \quad ,$$

$$\mathsf{id}\,\frac{}{\Gamma\{p,\bar{p}\}} \qquad \wedge\,\frac{\Gamma\{A\} \quad \Gamma\{B\}}{\Gamma\{A \wedge B\}} \qquad \vee\,\frac{\Gamma\{A,B\}}{\Gamma\{A \vee B\}}$$

$$\forall\,\frac{\Gamma\{\forall x[A]\}}{\Gamma\{\forall x A\}} \qquad \mathsf{scp}\,\frac{\forall x[\Gamma\{\Delta\}]}{\Gamma\{\forall x[\Delta]\}}\ \text{where } x \text{ does not occur in } \Gamma\{\ \}$$

$$\exists\mathsf{c}_1\,\frac{\Gamma\{\exists x A, A[x := y]\}}{\Gamma\{\exists x A\}}\ \text{where } \Gamma\{\ \} \text{ binds } y \qquad \exists\mathsf{c}_2\,\frac{\Gamma\{\exists x A, \forall x[A]\}}{\Gamma\{\exists x A\}}$$

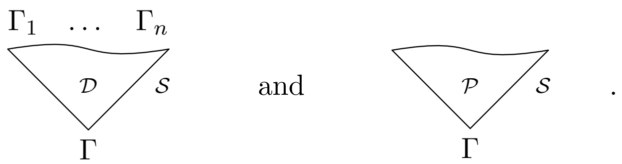
Figure 1: System Q

where in all contexts the holes are ordered from left to right as shown.

Inference rules, derivations, and proofs. *Inference rules* are of the form

$$\rho\,\frac{\Gamma_1 \ \ldots \ \Gamma_n}{\Gamma} \quad ,$$

where the $\Gamma_{(i)}$ are schematic sequents and ρ is the *name* of the rule. We sometimes write ρ^n to denote n instances of ρ and ρ^* to denote an unspecified number of instances of ρ. A *system*, denoted by $\mathcal{S}$, is a set of inference rules. A *derivation* in a system $\mathcal{S}$ is a finite tree which is built in the usual way from instances of inference rules from $\mathcal{S}$, which are applied modulo structural α-equivalence. The sequent at the root is the *conclusion* and the sequents at the leaves are the *premises* of the derivation. An *axiom* is a rule without premises. A *proof* of a sequent Γ in a system is a derivation in this system with conclusion Γ where each premise is an instances of an axiom. Derivations are denoted by $\mathcal{D}$ and proofs are denoted by $\mathcal{P}$. A derivation $\mathcal{D}$ in system $\mathcal{S}$ with premises $\Gamma_1 \ldots \Gamma_n$ and conclusion Γ and a proof $\mathcal{P}$ in system $\mathcal{S}$ of Γ are respectively denoted as



Systems Q and FQ. We say that a unary context *binds y* if it is of the form $\Gamma_1\{\forall y[\Gamma_2\{\ \}]\}$. Figure 1 shows *System Q*, the set of rules $\{\mathsf{id}, \wedge, \vee, \forall, \mathsf{scp}, \exists\mathsf{c}_1, \exists\mathsf{c}_2\}$. The id-rule is also called the *identity axiom* and the scp-rule is also called the *scope rule*. Notice how the existential rule is closed, and how the system thus has the *free variable property*: all free variables that occur in a proof occur in its conclusion. The $\exists\mathsf{c}_2$-rule postulates that the domain is non-empty. Removing it from System Q gives us *System FQ*, our system for free logic.

$$
\begin{array}{rc}
\mathsf{id} & \overline{\phantom{\forall xz[\forall y[\overline{P(z)}, P(y), \overline{P(x)}, P(z)]]}} \\
 & \forall xz[\forall y[\overline{P(z)}, P(y), \overline{P(x)}, P(z)]] \\
\mathsf{scp} & \overline{\phantom{\forall xz[\forall y[\overline{P(z)}, P(y)], \overline{P(x)}, P(z)]}} \\
 & \forall xz[\forall y[\overline{P(z)}, P(y)], \overline{P(x)}, P(z)] \\
\vee,\forall & \overline{\phantom{\forall xz[\forall y(P(z) \supset P(y)), \overline{P(x)}, P(z)]}} \\
 & \forall xz[\forall y(P(z) \supset P(y)), \overline{P(x)}, P(z)] \\
\exists_1 & \overline{\phantom{\forall xz[\exists x\forall y(P(x) \supset P(y)), \overline{P(x)}, P(z)]}} \\
 & \forall xz[\exists x\forall y(P(x) \supset P(y)), \overline{P(x)}, P(z)] \\
= & \overline{\phantom{\forall xy[\exists x\forall y(P(x) \supset P(y)), \overline{P(x)}, P(y)]}} \\
 & \forall xy[\exists x\forall y(P(x) \supset P(y)), \overline{P(x)}, P(y)] \\
\mathsf{scp}^2 & \overline{\phantom{\exists x\forall y(P(x) \supset P(y)), \forall x[\forall y[\overline{P(x)}, P(y)]]}} \\
 & \exists x\forall y(P(x) \supset P(y)), \forall x[\forall y[\overline{P(x)}, P(y)]] \\
\vee,\forall & \overline{} \\
 & \exists x\forall y(P(x) \supset P(y)), \forall x[\forall y(P(x) \supset P(y))] \\
\exists\mathsf{c}_2 & \overline{} \\
 & \exists x\forall y(P(x) \supset P(y))
\end{array}
$$

Figure 2: A proof of the drinker's formula

Provisos modulo renaming. Note that, since rules apply modulo structural α-equivalence, so do their provisos. In particular, the following is a valid instance of the scp-rule:

$$
\mathsf{scp}\ \frac{\forall x[P(x), \forall x Q(x)]}{\forall x[P(x)], \forall x Q(x)} \quad .
$$

A proof of the drinker's formula. Figure 2 shows a proof of the drinker's formula: there is a man such that when he drinks, everyone drinks. The proof makes use of the $\exists_1$-rule, which is just the $\exists\mathsf{c}_1$-rule without built-in contraction and will be shown to be admissible in the next section. The notation scp^2 denotes two instances of the scope rule and the notation $\vee, \forall$ denotes an instance of the $\vee$-rule followed by an instance of the $\forall$-rule. Note also that implication is a defined connective.

The drinker's formula is not provable in System FQ. Notice also that this is very easy to see: the $\exists\mathsf{c}_1$-rule does not apply because of its proviso and thus no rule at all is applicable to the formula.

System Q is complete for sentences only. System Q does not prove all valid sequents. In particular it is easy to see that it does not prove the valid sequent $\exists x P(x), \overline{P(y)}$. It only proves its universal closure $\forall y[\exists x P(x), \overline{P(y)}]$. This is by design: the non-closed sequent is not valid in varying domain semantics for modal predicate logic. Note that, thanks to the free variable property, restricting ourselves to sentences is less problematic than in the usual sequent calculus.

Why structural α-equivalence? Without α-renaming our system would not be complete even for sentences. The closed valid sequent $\forall y[\exists x \exists y P(x), \overline{P(y)}]$ would not be provable. Of course this situation is well-known. Many systems handle it by treating formulas modulo α-equivalence. In System Q it is sufficient to allow for structural α-renaming. The intention here is that a connective should be free while its corresponding structural connective is subject to some equations. We have disjunction and its corresponding comma

$$
\text{gid}\ \frac{}{\Gamma\{A,\bar{A}\}} \qquad \text{wk}\ \frac{\Gamma\{\varnothing\}}{\Gamma\{\Delta\}} \qquad \text{ctr}\ \frac{\Gamma\{\Delta,\Delta\}}{\Gamma\{\Delta\}} \qquad \text{ins}\ \frac{\Gamma\{\forall x[\Delta]\}}{\Gamma\{\Delta[x:=y]\}}\ \text{where } \Gamma\{\ \} \text{ binds } y
$$

$$
\text{gen}\ \frac{\Gamma}{\forall x[\Gamma]} \qquad \text{per}\ \frac{\Gamma\{\forall x[\forall y[\Delta]]\}}{\Gamma\{\forall y[\forall x[\Delta]]\}} \qquad \text{vac}\ \frac{\Gamma\{\forall x[\Delta]\}}{\Gamma\{\Delta\}}\ \text{where } x \text{ does not occur in } \Delta
$$

$$
\text{cut}\ \frac{\Gamma\{A\} \quad \Gamma\{\bar{A}\}}{\Gamma\{\varnothing\}} \qquad \exists_1\ \frac{\Gamma\{A[x:=y]\}}{\Gamma\{\exists x A\}}\ \text{where } \Gamma\{\ \} \text{ binds } y \qquad \exists_2\ \frac{\Gamma\{\forall x[A]\}}{\Gamma\{\exists x A\}}
$$

Figure 3: Admissible rules

(with associativity and commutativity) and similarly we have the universal quantifier and its corresponding structural universal quantifier (with α-renaming).

3 Admissible Rules and Invertibility

Figure 3 show some rules which are admissible for System Q. Their names are: *general identity*, *weakening*, *contraction*, *instantiation*, *generalisation*, *permutation*, *vacuous quantification*, *cut*, *first existential* and *second existential*. To show their admissibility we first need some definitions of standard notions.

Cut rank. The *depth* of a formula A, denoted $depth(A)$, is defined as usual, the depth of possibly negated propositions being zero. Given an instance of the *cut-rule* as shown in Figure 3, its *cut formula* is A and its *cut rank* is one plus the depth of its cut formula. For $r \geq 0$ we define the cut_r*-rule* which is cut with at most rank r. The *cut rank* of a derivation is the supremum of the cut ranks of its cuts.

Admissibility and derivability. An inference rule ρ is *(depth-preserving) admissible* for a system $\mathcal{S}$ if for each proof in $\mathcal{S} \cup \{\rho\}$ there is a proof in $\mathcal{S}$ with the same conclusion (and with at most the same depth). An inference rule ρ is *derivable* for a system $\mathcal{S}$ if for each instance of ρ there is a derivation $\mathcal{D}$ in $\mathcal{S}$ with the same conclusion and the same set of premises. A rule is *cut-rank (and depth-) preserving admissible* for a system $\mathcal{S}$ if for all $r \geq 0$ the rule is (depth-preserving) admissible for $\mathcal{S} + \mathsf{cut}_r$.

Invertibility. For each rule ρ there is its *inverse*, denoted by ρ^{-1}, which is obtained by exchanging premise and conclusion. The inverse of a two-premise-rule allows each premise as a conclusion. An inference rule ρ is *(depth-preserving) invertible* for a system $\mathcal{S}$ if ρ^{-1} is (depth-preserving) admissible for $\mathcal{S}$. An inference rule ρ is *cut-rank (and depth-) preserving invertible* for a system $\mathcal{S}$ if ρ^{-1} is cut-rank (and depth-) preserving admissible for $\mathcal{S}$.

We are now ready to state our lemma on admissibility and invertibility.

Lemma 3.1 (Admissibility and Invertibility)
(i) For each system $\mathcal{S} \in \{\mathsf{FQ}, \mathsf{Q}\}$, each rule of $\mathcal{S}$ is cut-rank and depth-preserving invertible for $\mathcal{S}$.
(ii) For each system $\mathcal{S} \in \{\mathsf{FQ}, \mathsf{Q}\}$, each of the first six rules from Figure 3 is cut-rank and depth-preserving admissible for $\mathcal{S}$.
(iii) The vacuous quantification rule is depth-preserving admissible for System Q.

Proof. To show (i) we need admissibility of the weakening and permutation rules for both systems, so we show this admissibility first. The proof of cut-rank and depth-preserving admissibility is as usual, by induction of the number of instances of the respective rule in a given proof. We choose a top-most instance and remove it by induction on the depth of the proof above it and a case analysis on its lowermost rule. For weakening admissibility all cases are trivial.

Admissibility of per. For permutation admissibility the only two interesting cases are as follows:

$$\mathsf{per}\,\dfrac{\mathsf{scp}\,\dfrac{\forall y \Gamma\{\forall x \Delta\}}{\Gamma\{\forall y \forall x \Delta\}}}{\Gamma\{\forall x \forall y \Delta\}} \quad \rightsquigarrow \quad \mathsf{scp}\,\dfrac{\forall y \Gamma\{\forall x \Delta\}}{\Gamma\{\forall x \forall y \Delta\}}$$

and

$$\mathsf{per}\,\dfrac{\mathsf{scp}\,\dfrac{\forall x \Gamma\{\forall y \Delta\}}{\Gamma\{\forall y \forall x \Delta\}}}{\Gamma\{\forall x \forall y \Delta\}} \quad \rightsquigarrow \quad \mathsf{scp}\,\dfrac{\forall x \Gamma\{\forall y \Delta\}}{\Gamma\{\forall x \forall y \Delta\}} \quad .$$

We go on to show (i). Invertibility is trivial for all rules except for the scope rule. So we just show the invertibility of scope, meaning the admissibility of its inverse scp^{-1}.

Invertibility of scp. The only interesting case is when an instance of scp is above it. We make a case analysis on the position of the active quantifier in the scp^{-1}-rule with respect to the active quantifier in the scp-rule. We have the following four cases. First, the two scopes coincide:

$$\mathsf{scp}^{-1}\,\dfrac{\mathsf{scp}\,\dfrac{\forall x \Gamma\{\Delta\}}{\Gamma\{\forall x \Delta\}}}{\forall x \Gamma\{\Delta\}} \quad \rightsquigarrow \quad \forall x \Gamma\{\Delta\} \quad ,$$

second, the scope of the first includes the second:

$$\mathsf{scp}^{-1}\,\dfrac{\mathsf{scp}\,\dfrac{\forall y \Gamma_1\{\forall x \Gamma_2\{\Delta\}\}}{\Gamma_1\{\forall x \Gamma_2\{\forall y \Delta\}\}}}{\forall x \Gamma_1\{\Gamma_2\{\forall y \Delta\}\}} \quad \rightsquigarrow \quad \mathsf{scp}\,\dfrac{\mathsf{per}\,\dfrac{\mathsf{scp}^{-1}\,\dfrac{\forall y \Gamma_1\{\forall x \Gamma_2\{\Delta\}\}}{\forall x \forall y \Gamma_1\{\Gamma_2\{\Delta\}\}}}{\forall y \forall x \Gamma_1\{\Gamma_2\{\Delta\}\}}}{\forall x \Gamma_1\{\Gamma_2\{\forall y \Delta\}\}} \quad ,$$

third, the scope of the second includes the first:

$$\mathsf{scp}^{-1}\,\dfrac{\mathsf{scp}\,\dfrac{\forall y\Gamma_1\{\Gamma_2\{\forall x\Delta\}\}}{\Gamma_1\{\forall y\Gamma_2\{\forall x\Delta\}\}}}{\forall x\Gamma_1\{\forall y\Gamma_2\{\Delta\}\}} \quad\rightsquigarrow\quad \mathsf{scp}\,\dfrac{\mathsf{per}\,\dfrac{\mathsf{scp}^{-1}\,\dfrac{\forall y\Gamma_1\{\Gamma_2\{\forall x\Delta\}\}}{\forall x\forall y\Gamma_1\{\Gamma_2\{\Delta\}\}}}{\forall y\forall x\Gamma_1\{\Gamma_2\{\Delta\}\}}}{\forall x\Gamma_1\{\forall y\Gamma_2\{\Delta\}\}}\ ,$$

and fourth, the two scopes are disjoint:

$$\mathsf{scp}^{-1}\,\dfrac{\mathsf{scp}\,\dfrac{\forall y\Gamma\{\Delta_1\}\{\forall x\Delta_2\}}{\Gamma\{\forall y\Delta_1\}\{\forall x\Delta_2\}}}{\forall x\Gamma\{\forall y\Delta_1\}\{\Delta_2\}} \quad\rightsquigarrow\quad \mathsf{scp}\,\dfrac{\mathsf{per}\,\dfrac{\mathsf{scp}^{-1}\,\dfrac{\forall y\Gamma\{\Delta_1\}\{\forall x\Delta_2\}}{\forall x\forall y\Gamma\{\Delta_1\}\{\Delta_2\}}}{\forall y\forall x\Gamma\{\Delta_1\}\{\Delta_2\}}}{\forall x\Gamma\{\forall y\Delta_1\}\{\Delta_2\}}\ .$$

This concludes the proof of (i) for both Systems FQ and Q. To prove (ii) we now look at the non-trivial cases in the admissibility arguments for the remaining rules, namely general identity, contraction, instantiation, and generalisation.

Admissibility of gid. The admissibility argument for the general identity is the standard induction on the structure of the active formula. The case where the main connective of the active formula is a quantifier is as follows:

$$\mathsf{gid}\,\dfrac{}{\Gamma\{\forall xA, \exists x\bar{A}\}} \quad\rightsquigarrow\quad \mathsf{scp},\forall\,\dfrac{\exists\mathsf{c}_1\,\dfrac{\mathsf{gid}\,\dfrac{}{\forall x\Gamma\{A, \bar{A}, \exists x\bar{A}\}}}{\forall x\Gamma\{A, \exists x\bar{A}\}}}{\Gamma\{\forall xA, \exists x\bar{A}\}}\ .$$

Admissibility of ctr. For contraction, all rules above it except scope are handled by using invertibility:

$$\mathsf{ctr}\,\dfrac{\rho\,\dfrac{\Gamma\{\Delta, \Delta'\}}{\Gamma\{\Delta, \Delta\}}}{\Gamma\{\Delta\}} \quad\rightsquigarrow\quad \rho\,\dfrac{\mathsf{ctr}\,\dfrac{\rho^{-1}\,\dfrac{\Gamma\{\Delta, \Delta'\}}{\Gamma\{\Delta', \Delta'\}}}{\Gamma\{\Delta'\}}}{\Gamma\{\Delta\}}\ ,$$

and scope is handled as follows:

$$\mathsf{ctr}\,\dfrac{\mathsf{scp}\,\dfrac{\forall x\Gamma_1\{\Gamma_2\{\Delta\}, \Gamma_2\{\forall x\Delta\}\}}{\Gamma_1\{\Gamma_2\{\forall x\Delta\}, \Gamma_2\{\forall x\Delta\}\}}}{\Gamma_1\{\Gamma_2\{\forall x\Delta\}\}} \quad\rightsquigarrow\quad \mathsf{scp}\,\dfrac{\mathsf{ctr}\,\dfrac{\mathsf{ins}\,\dfrac{\forall x\Gamma_1\{\Gamma_2\{\Delta\}, \Gamma_2\{\forall x\Delta\}\}}{\forall x\Gamma_1\{\Gamma_2\{\Delta\}, \Gamma_2\{\Delta\}\}}}{\forall x\Gamma_1\{\Gamma_2\{\Delta\}\}}}{\Gamma_1\{\Gamma_2\{\forall x\Delta\}\}}\ .$$

Admissibility of ins. For instantiation there are two interesting cases. The first case is when it is below an $\exists\mathsf{c}_1$-instance, which relies on the structural quantifier that the instantiation rule removes (seen downwards). Notice that the provisos on the right are fulfilled since $\Gamma_1\{\ \}$ binds z:

$$\mathsf{ins}\,\dfrac{\exists\mathsf{c}_1\,\dfrac{\Gamma_1\{\forall x\Gamma_2\{\exists yA, A[y:=x]\}\}}{\Gamma_1\{\forall x\Gamma_2\{\exists yA\}\}}}{\Gamma_1\{\Gamma_2\{\exists yA\}[x:=z]\}} \quad\leadsto\quad \exists\mathsf{c}_1\,\dfrac{=\,\dfrac{\mathsf{ins}\,\dfrac{\Gamma_1\{\forall x\Gamma_2\{\exists yA, A[y:=x]\}\}}{\Gamma_1\{\Gamma_2\{\exists yA, A[y:=x]\}[x:=z]\}}}{\Gamma_1\{\Gamma_2\{\exists yA, A[y:=z]\}[x:=z]\}}}{\Gamma_1\{\Gamma_2\{\exists yA\}[x:=z]\}} \quad .$$

The second interesting case is instantiation below scope, if the same structural universal quantifier is active in both rules:

$$\mathsf{ins}\,\dfrac{\mathsf{scp}\,\dfrac{\forall x\Gamma\{\Delta\}}{\Gamma\{\forall x\Delta\}}}{\Gamma\{\Delta[x:=y]\}} \quad\leadsto\quad \mathsf{scp}\,\dfrac{=\,\dfrac{\mathsf{ins}\,\dfrac{\mathsf{scp}^{-1}\,\dfrac{\forall x\Gamma\{\Delta\}}{\forall y\forall x\Gamma'\{\Delta\}}}{\forall y(\Gamma'\{\Delta\}[x:=y])}}{\forall y\Gamma'\{\Delta[x:=y]\}}}{\Gamma\{\Delta[x:=y]\}}$$

Admissibility of gen. For the generalisation rule there is just one interesting case, namely when it is below a scope:

$$\mathsf{gen}\,\dfrac{\mathsf{scp}\,\dfrac{\forall y\Gamma\{\Delta\}}{\Gamma\{\forall y\Delta\}}}{\forall x\Gamma\{\forall y\Delta\}} \quad\leadsto\quad \mathsf{scp}\,\dfrac{\mathsf{per}\,\dfrac{\mathsf{gen}\,\dfrac{\forall y\Gamma\{\Delta\}}{\forall x\forall y\Gamma\{\Delta\}}}{\forall y\forall x\Gamma\{\Delta\}}}{\forall x\Gamma\{\forall y\Delta\}} \quad .$$

This concludes the proof of (ii) for both Systems FQ and Q. We go on to show (iii), the admissibility of the vacuous quantification rule for Q.

Admissibility of vac. There is just one interesting case, namely when the vac-rule is below an $\exists\mathsf{c}_1$-rule which relies on the vacuous quantifier. It is handled as follows:

$$\mathsf{vac}\,\dfrac{\exists\mathsf{c}_1\,\dfrac{\Gamma\{\forall y\Delta\{\exists xA, A[x:=y]\}\}}{\Gamma\{\forall y\Delta\{\exists xA\}\}}}{\Gamma\{\Delta\{\exists xA\}\}} \quad\leadsto\quad \exists\mathsf{c}_2\,\dfrac{=\,\dfrac{\mathsf{scp}\,\dfrac{\mathsf{scp}^{-1}\,\dfrac{\Gamma\{\forall y\Delta\{\exists xA, A[x:=y]\}\}}{\forall y\Gamma\{\Delta\{\exists xA, A[x:=y]\}\}}}{\Gamma\{\Delta\{\exists xA, \forall y[A[x:=y]]\}\}}}{\Gamma\{\Delta\{\exists xA, \forall x[A]\}\}}}{\Gamma\{\Delta\{\exists xA\}\}} \quad .$$

□

Note that the vacuous quantification rule is not admissible for System FQ: the sequent $\forall x[\exists y(p \vee \bar{p})]$ is provable, but not the sequent $\exists y(p \vee \bar{p})$.

By the $\forall$- and $\vee$-rules and their invertibility we get the following corollary.

Corollary 3.2 Each system from $\{\mathsf{FQ}, \mathsf{Q}\}$ proves a sequent iff it proves its corresponding formula.

By weakening admissibility we also get the following corollary.

Corollary 3.3 The $\exists_1$-rule is admissible for each system from $\{\mathsf{FQ}, \mathsf{Q}\}$. The $\exists_2$-rule is admissible for System Q.

4 Relation between System Q and the usual Sequent Calculus

We now see how to embed System Q into a standard sequent system and vice versa. The specific standard system we use is a variant of System $\mathsf{GS1}$ from [17]. We call this variant *System* $\mathsf{LK1}$ and the only difference between $\mathsf{LK1}$ and $\mathsf{GS1}$ is that $\mathsf{LK1}$ does not treat formulas modulo α-equivalence. Instead it requires that free and bound variables come from disjoint sets, just like in Gentzen's original System LK [9]. So an $\mathsf{LK1}$-*sequent* is a multiset of formulas for which the set of free variables and the set of bound variables are disjoint.

We denote sets of variables by $\vec{x}$ and write $\vec{x}, y$ for $\vec{x} \cup \{y\}$. Given a set of variables $\vec{x}$, the corresponding sequence of structural universal quantifiers is written as $\forall\vec{x}[\ ]$, where variables are ordered according to the fixed total order. The *universal closure* of a sequent Γ is the sequent $\forall\vec{x}[\Gamma]$ where $\vec{x}$ is the set of free variables of Γ.

Theorem 4.1 (Embedding System $\mathsf{LK1}$ into System Q) If System $\mathsf{LK1}$ proves an $\mathsf{LK1}$-sequent then System Q proves its universal closure.

Proof. We proceed by induction on the depth of the given proof in $\mathsf{LK1}$. The identity axiom and the rules for disjunction and conjunction are trivial, so we just show the cases for the quantifier rules. For the universal rule there are two cases. Here is the first, where x occurs free in A (and thus y occurs free in the premise of the $\forall$-rule). Note that the proviso of the scope rule and of the renaming below it are fulfilled because of the proviso of the $\forall$-rule:

$$\forall\,\frac{\Gamma, A[x := y]}{\Gamma, \forall x A}\quad \begin{array}{c}\text{where } y \text{ does}\\ \text{not occur in}\\ \text{conclusion}\end{array} \qquad \leadsto \qquad \forall\,\cfrac{=\,\cfrac{\mathsf{scp}\,\cfrac{\mathsf{per}^*\,\cfrac{\forall\vec{x}, y[\Gamma, A[x := y]]}{\forall y \forall\vec{x}[\Gamma, A[x := y]]}}{\forall\vec{x}[\Gamma, \forall y[A[x := y]]]}}{\forall\vec{x}[\Gamma, \forall x[A]]}}{\forall\vec{x}[\Gamma, \forall x A]} \quad .$$

In the second case, where x does not occur free in A, the translation is almost the same, just the instances of the permutation rule are replaced by an instance of generalisation.

For the existential rule, there are three cases. Here is the first, where y is free in the conclusion:

$$\exists\,\frac{\Gamma, A[x := y]}{\Gamma, \exists x A} \qquad \leadsto \qquad \exists_1\,\frac{\forall\vec{x}, y[\Gamma, A[x := y]]}{\forall\vec{x}, y[\Gamma, \exists x A]} \quad .$$

The second case, where y is free in the premise of the $\exists$-rule but not in the conclusion is as follows. Note that here the proviso in the scope rule is fulfilled because free and bound variables are disjoint in $\mathsf{LK1}$-proofs:

$$\exists\,\frac{\Gamma, A[x := y]}{\Gamma, \exists x A} \quad \rightsquigarrow \quad \exists_2\,\cfrac{=\,\cfrac{\mathsf{scp}\,\cfrac{\mathsf{per}^*\,\cfrac{\forall \vec{x}, y[\Gamma, A[x := y]]}{\forall y \forall \vec{x}[\Gamma, A[x := y]]}}{\forall \vec{x}[\Gamma, \forall y[A[x := y]]]}}{\forall \vec{x}[\Gamma, \forall x[A]]}}{\forall \vec{x}[\Gamma, \exists x A]} \quad .$$

The third case, where y is neither free in premise nor conclusion is similar to the second case, with the instances of permutation replaced by one instance of generalisation. □

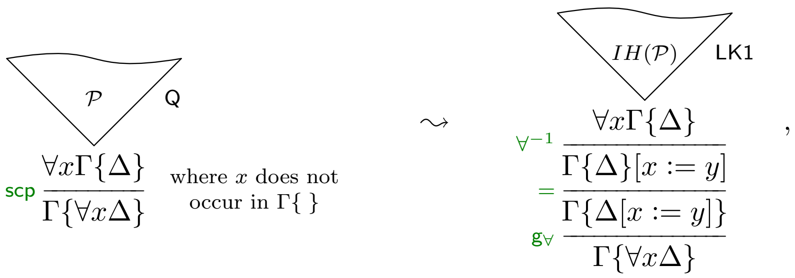
We now see the reverse direction, that is, we embed System Q into System $\mathsf{LK1}$. We first need some definitions. Define *formula contexts* in analogy to sequent contexts. Let a *restricted formula context* be one where the hole occurs in the scope of at most disjunction and universal quantification. The *glue rules* are shown in Figure 4, where $F\{\ \}$ is a restricted formula context. They are just a technical device useful for embedding System Q into System $\mathsf{LK1}$.

Lemma 4.2 (Glue for LK1) The glue rules are admissible for System $\mathsf{LK1}$.

Proof. By an induction on the depth of $F\{\ \}$ and using invertibility of the $\forall$-rule and of a $\mathsf{G3}$-style $\vee$-rule, we can reduce the admissibility of each glue rule to the admissibility of its restriction to an empty formula context $F\{\ \}$. The admissibility of the restricted glue rules is easy to see. □

Theorem 4.3 If System Q proves a sequent, then System $\mathsf{LK1}$ proves its corresponding formula.

Proof. Since by the previous lemma the $\{\mathsf{g_c}, \mathsf{g_a}, \mathsf{g}_\alpha\}$-rules are admissible for $\mathsf{LK1}$ we assume in the following equiprovability of sequents that only differ modulo commutativity and associativity of disjunction and renaming of universally bound variables. We proceed by induction on the depth of the given proof in System Q. The propositional rules are just translated to the corresponding glue rules which are admissible for $\mathsf{LK1}$ by the previous lemma. The case for the scope rule is as follows:



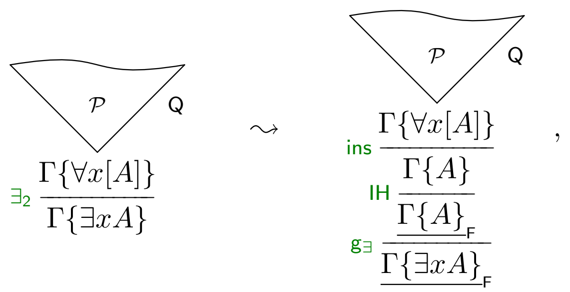
$$\mathsf{g_c}\ \frac{\Gamma, F\{A \vee B\}}{\Gamma, F\{B \vee A\}} \qquad \mathsf{g_a}\ \frac{\Gamma, F\{(A \vee B) \vee C\}}{\Gamma, F\{A \vee (B \vee C)\}} \qquad \mathsf{g}_\alpha\ \frac{\Gamma, F\{\forall x A\}}{\Gamma, F\{\forall y A[x := y]\}}$$

$$\mathsf{g_{id}}\ \frac{}{\Gamma, F\{p \vee \bar{p}\}} \qquad \mathsf{g}_\wedge\ \frac{\Gamma, F\{A\} \quad \Gamma, F\{B\}}{\Gamma, F\{A \wedge B\}} \qquad \mathsf{g_{ctr}}\ \frac{\Gamma, F\{A \vee A\}}{\Gamma, F\{A\}}$$

$$\mathsf{g}_\forall\ \frac{\Gamma, F\{A[x := y]\}}{\Gamma, F\{\forall x A\}}\ \text{where } y \text{ does not occur in the conclusion} \qquad \mathsf{g}_\exists\ \frac{\Gamma, F\{A[x := y]\}}{\Gamma, F\{\exists x A\}}$$

Figure 4: Some glue for embedding System Q into System LK1

where on the right y is a fresh variable, the equality is justified by the proviso of the scope rule, and we have just written sequents to denote their corresponding formulas. The case for the $\exists\mathsf{c}_1$ is handled easily by splitting it into an instance of $\exists_1$ and contraction and using the corresponding glue rules. The case for the $\exists\mathsf{c}_2$-rule is also handled splitting it into $\exists_2$ and contraction, and translating $\exists_2$ as follows:



where on the right ins is depth-preserving admissible for Q and IH denotes applying the induction hypothesis to the proof above. □

By soundness and completeness of LK1 our embeddings give us soundness and completeness for System Q.

Theorem 4.4 (System Q is sound and complete) System Q proves a sentence iff it is valid (for the standard notion of validity in classical predicate logic).

5 Relation between System FQ and Free Logic

We will now see that System FQ is sound and complete with respect to free logic. For completeness we embed a Hilbert-style system for free logic into System FQ + cut

(Tautology)	a propositional tautology
(Distributivity)	$\forall x(A \supset B) \supset (\forall x A \supset \forall x B)$
(Vacuous)	$A \supset \forall x A$ where x does not occur free in A
(Instantiation)	$\forall y(\forall x A \supset A[x := y])$
(Permutation)	$\forall x \forall y A \supset \forall y \forall x A$

Figure 5: Axioms of System FQC

and then use cut elimination for System FQ, which is proved in the next section. The specific Hilbert-system we use is *System FQC* from Bencivenga [4], which consists of modus ponens and all generalisations of instances of the axioms shown in Figure 5. As usual, a *generalisation* of a formula is obtained by prefixing it with any sequence of universal quantifiers.

It is easy to see that the axioms of FQC are provable in System FQ and that modus pones is admissible for System $\mathsf{FQ} + \mathsf{cut}$, so we have the following theorem.

Theorem 5.1 (Embedding System FQC into System FQ + cut) If a formula is provable in System FQC then it is provable in System $\mathsf{FQ} + \mathsf{cut}$.

To show soundness of System FQ we embed it into a sequent system for free logic, a variant of a system from Bencivenga [4]. *System FLK1* is System LK1 with the quantifier rules replaced by the ones shown in Figure 6, where E, the *existence predicate*, is a fixed unary predicate symbol, which is not used in formulas.

The proof of the following lemma is standard.

Lemma 5.2 (Invertibility for FLK1) All rules of System FLK1 are depth-preserving invertible for System FLK1.

The *free glue rules* are shown in Figure 7, where $F\{\ \}$ is a restricted formula context. The proof of the following lemma follows the lines of the corresponding proof for LK1 and makes use of the invertibility of rules in FLK1.

Lemma 5.3 (Glue for FLK1) The rules $\{\mathsf{g_c}, \mathsf{g_a}, \mathsf{g}_\alpha, \mathsf{g_{id}}, \mathsf{g}_\wedge, \mathsf{g_{ctr}}\}$ and the free glue rules are admissible for System FLK1.

We can now embed FQ into FLK1.

Theorem 5.4 (Embedding System FQ into System FLK1) If System FQ proves a sequent, then System FLK1 proves its corresponding formula.

Proof. The proof is similar to the embedding of System Q into System LK1. The difference is in the translation of the scope rule and the $\exists \mathsf{c_1}$-rule, and, of course, the fact that

$$
\mathsf{\forall_F}\ \frac{\Gamma, A[x := y], \overline{E(y)}}{\Gamma, \forall x A}\ \text{where } y \text{ does not occur in the conclusion}
\qquad
\mathsf{\exists_F}\ \frac{\Gamma, A[x := y] \quad \Gamma, E(y)}{\Gamma, \exists x A}
$$

Figure 6: Quantifier rules of System FLK1

$$
\mathsf{g\forall_F}\ \frac{\Gamma, F\{A[x := y]\}, \overline{E(y)}}{\Gamma, F\{\forall x A\}}\ \text{where } y \text{ does not occur in the conclusion}
\qquad
\mathsf{g\exists_F}\ \frac{\Gamma, F\{A[x := y]\} \quad \Gamma, F\{E(y)\}}{\Gamma, F\{\exists x A\}}
$$

Figure 7: Glue for FLK1

we do not need to translate the $\exists c_2$-rule. Here is the case for the scope rule:

$$
\mathsf{scp}\ \frac{\forall x \Gamma\{\Delta\}}{\Gamma\{\forall x \Delta\}}\ \text{where } x \text{ does not occur in } \Gamma\{\ \}
\quad \rightsquigarrow \quad
\mathsf{g\forall_F}\ \dfrac{=\ \dfrac{\mathsf{\forall_F^{-1}}\ \dfrac{\forall x \Gamma\{\Delta\}}{\Gamma\{\Delta\}[x := y], \overline{E(y)}}}{\Gamma\{\Delta[x := y]\}, \overline{E(y)}}}{\Gamma\{\forall x \Delta\}}\ ,
$$

where on the right y is a fresh variable, the equality is justified by the proviso of the scope rule, and we have just written sequents to denote their corresponding formulas.

For simplicity we just show $\exists_1$ since $\exists c_1$ is derivable for $\exists_1$ and contraction:

$$
\mathsf{\exists_1}\ \frac{\Gamma\{A[x := y]\}}{\Gamma\{\exists x A\}}\ \text{where } \Gamma\{\ \} \text{ binds } y
\quad \rightsquigarrow \quad
\mathsf{g\exists_F}\ \dfrac{\Gamma\{A[x := y]\} \qquad \mathsf{\forall_F^*,\vee^*,wk^*}\ \dfrac{\mathsf{\forall_F}\ \dfrac{\mathsf{id}\ \dfrac{}{E(z), \overline{E(z)}}}{\forall y E(y)}}{\Gamma\{E(y)\}}}{\Gamma\{\exists x A\}}\ ,
$$

□

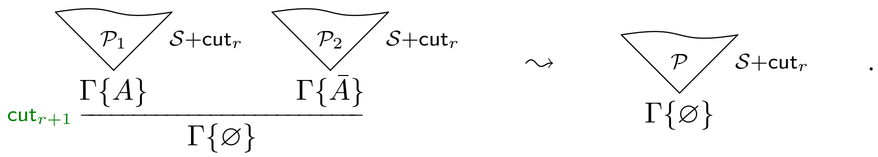
By soundness of FLK1 and completeness of FQC with respect to free logic, our embeddings, and cut-elimination for FQ we obtain soundness and completeness for System FQ with respect to free logic.

Theorem 5.5 (System FQ is sound and complete) System FQ proves a formula iff it is a theorem of free logic.

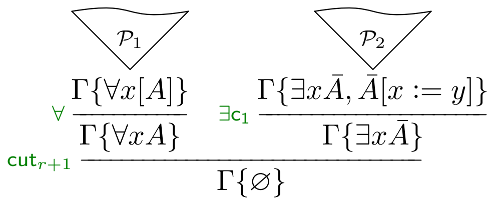
6 Syntactic Cut-Elimination

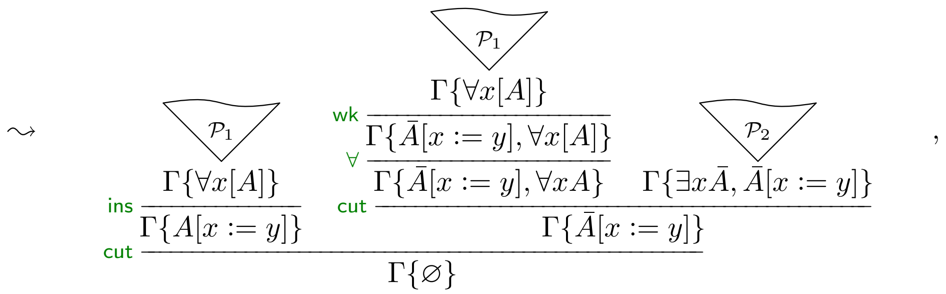
We now turn to cut-elimination. As usual, the reduction lemma is the centerpiece of the cut elimination argument.

Lemma 6.1 (Reduction Lemma) Let $\mathcal{S} \in \{\mathsf{FQ}, \mathsf{Q} + \mathsf{vac}\}$. If there is a proof as shown on the left, then there is a proof as shown on the right:

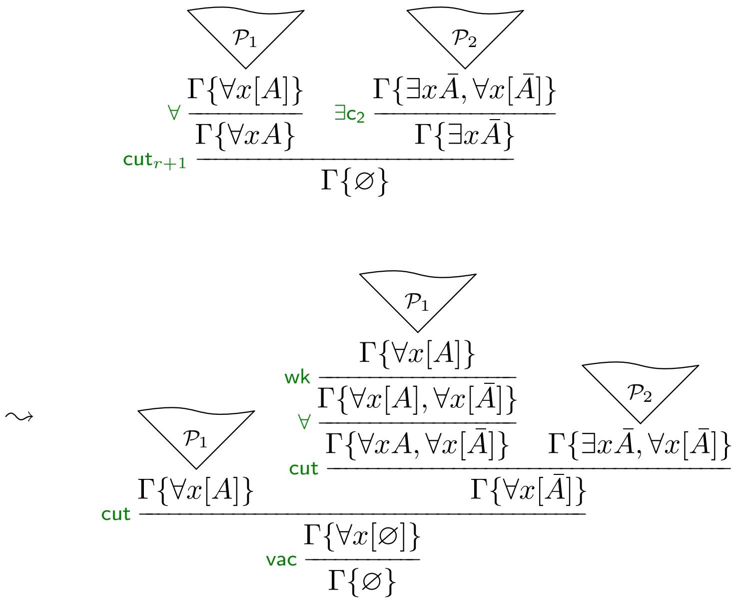


Proof. We first prove the lemma for System FQ. We proceed as usual, by an induction on the sum of the depths of $\mathcal{P}_1$ and $\mathcal{P}_2$ and by a case analysis on their lowermost rules. All passive cases (that is, those where the lowermost rule does not apply to the cut-formula) are handled using invertibility, and using contraction admissibility in the passive conjunction case. Note that scope can only be passive. The active cases for the axiom and the propositional connectives are as usual. We now look at the only interesting case, namely $\forall$ vs. $\exists\mathsf{c}_1$, which is handled as follows:





where the proviso of the ins-rule is fulfilled because of the proviso of the $\exists\mathsf{c}_1$-rule. This concludes the proof for System FQ. The proof for System $\mathsf{Q} + \mathsf{vac}$ is the same, but with the additional case $\forall$ vs. $\exists c_2$:



□

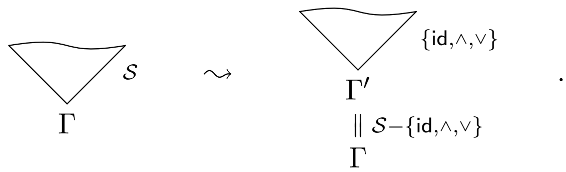
Cut-elimination for System FQ now follows from a routine induction on the cut-rank of the given proof with a subinduction on the depth of the proof, using the reduction lemma in the case of a maximal-rank cut. To get cut-elimination for System Q we first prove it for System $\mathsf{Q} + \mathsf{vac}$, in the same way as we did for FQ, and then use the admissibility of the vac-rule. So we have the following theorem.

Theorem 6.2 (Cut-Elimination) Let $\mathcal{S} \in \{\mathsf{FQ}, \mathsf{Q}\}$. If a sequent is provable in System $\mathcal{S} + \mathsf{cut}$ then it is provable in System $\mathcal{S}$.

7 Herbrand's Theorem and Interpolation

We now see two simple applications of our cut-free system: Herbrand's Theorem and interpolation, both for classical predicate logic and free logic. The point here is of course not that these results are new, the point is that our cut-free systems are useful enough to easily provide them. That said, in the case of free logic the syntactic proof of interpolation seems to be new: it is not easy to see how to get it from System FLK1 without allowing the existence predicate to occur in the interpolant.

Theorem 7.1 (Mid-Sequent Theorem) Let $\mathcal{S} \in \{\mathsf{FQ}, \mathsf{Q}\}$ and let Γ contain only prenex formulas. If there is a proof as shown on the left, then there is a proof as shown on the right:



Proof. We first establish the claim that each rule in $\{\forall, \mathsf{scp}, \exists\mathsf{c}_1, \exists\mathsf{c}_2\}$ is depth-preserving invertible for System $\{\mathsf{id}, \wedge, \vee\}$.

To prove the decomposition theorem we now proceed by induction on the depth of the given proof and a case analysis on the lowermost rule. Consider the case when that is the $\wedge$-rule, which is the only interesting case. We first decompose the left subproof by induction hypothesis. Then we permute all the quantifier rules we obtained down below the $\wedge$-rule, using invertibility to compensate on its right premise. Note that the depth of the right subproof is preserved. Now we decompose the right subproof by induction hypothesis. Then we permute all the quantifier rules that we obtained down below the $\wedge$-rule, using the claim to compensate on its left premise. We have now obtained a proof of the desired form. □

The mid-sequent Γ' may still contain formulas with quantifiers as well as structural universal quantifiers. However, since its proof contains only propositional rules, both can be easily removed. So we have the following corollary.

Corollary 7.2 (Herbrand's Theorem) Let $\mathcal{S} \in \{\mathsf{FQ}, \mathsf{Q}\}$ and let A be a prenex formula which is provable in System $\mathcal{S}$. Then there is a sequent which 1) consists only of substitution instances of the matrix of A and 2) is propositionally provable.

We also easily obtain the following interpolation theorem, by a standard induction on the depth of the given proof.

Theorem 7.3 (Interpolation) Let $\mathcal{S} \in \{\mathsf{FQ}, \mathsf{Q}\}$. If $\mathcal{S}$ proves the sequent $\forall\vec{z}[\Gamma, \Delta]$ then there is a formula C, called *interpolant*, such that 1) each predicate symbol and each free variable in C occurs both in Γ and in Δ, and 2) system $\mathcal{S}$ proves both the sequent $\forall\vec{z}[\Gamma, C]$ and the sequent $\forall\vec{z}[\bar{C}, \Delta]$.

Proof. We just show the case for System Q since the case for FQ is properly contained in it. We first separate the existential rules in the given proof from contraction. We then proceed by induction on the depth of the obtained proof and a case analysis on the lowermost rule. The cases for the propositional rules are as usual, and the cases for the $\forall$-rule and the $\exists_2$-rule are trivial. This leaves the cases of the scp- and $\exists_1$-rule. For scp we have a proof which ends in:

$$\mathsf{scp}\,\frac{\forall x\vec{z}[\Gamma_1, \Gamma_2\{\Delta\}]}{\forall\vec{z}[\Gamma_1, \Gamma_2\{\forall x[\Delta]\}]} \quad ,$$

for which we construct the desired proofs as follows:

$$\forall\,\dfrac{\mathsf{scp}\,\dfrac{\forall x\vec{z}[\Gamma_1, C]}{\forall\vec{z}[\Gamma_1, \forall x[C]]}}{\forall\vec{z}[\Gamma_1, \forall x C]} \qquad \text{and} \qquad \mathsf{scp}\,\dfrac{\exists_1\,\dfrac{\forall x\vec{z}[\bar{C}, \Gamma_2\{\Delta\}]}{\forall x\vec{z}[\exists x\bar{C}, \Gamma_2\{\Delta\}]}}{\forall\vec{z}[\exists x\bar{C}, \Gamma_2\{\forall x[\Delta]\}]} \quad ,$$

where C is the interpolant obtained by induction hypothesis. For the case of the $\exists_1$-rule we only show the subcase where we need to prevent y from occuring free in the interpolant, which is the only non-trivial subcase. It is the one where $\Gamma_2\{\ \}$ does not bind y and y is not free in $\Gamma_2\{\ \}$:

$$\exists_1\,\dfrac{\forall\vec{z}y[\Gamma_1, \Gamma_2\{A[x := y]\}]}{\forall\vec{z}y[\Gamma_1, \Gamma_2\{\exists x A\}]} \quad .$$

We construct the desired proofs as follows:

$$\exists_1\,\dfrac{\forall\vec{z}y[\Gamma_1, C]}{\forall\vec{z}y[\Gamma_1, \exists y C]} \qquad \text{and} \qquad =\,\dfrac{\forall,\mathsf{scp}\,\dfrac{\exists_1\,\dfrac{\mathsf{scp}\,\dfrac{\mathsf{gen,per}^*\,\dfrac{\forall\vec{z}y[\bar{C}, \Gamma_2\{A[x := y]\}]}{\forall y'y\vec{z}[\bar{C}, \Gamma_2\{A[x := y]\}]}}{\forall y\vec{z}y'[\bar{C}, \Gamma_2\{A[x := y]\}]}}{\forall y\vec{z}y'[\bar{C}, \Gamma_2\{\exists x A\}]}}{\forall\vec{z}y'[\forall y\bar{C}, \Gamma_2\{\exists x A\}]}}{\forall\vec{z}y[\forall y\bar{C}, \Gamma_2\{\exists x A\}]} \quad ,$$

where again C is the interpolant obtained by induction hypothesis. □

References

[1] Arnon Avron. The method of hypersequents in the proof theory of propositional non-classical logics. In Wilfrid Hodges, Martin Hyland, Charles Steinhorn, and John Truss, editors, *Logic: from foundations to applications. Proc. Logic Colloquium, Keele, UK, 1993*, pages 1–32. Oxford University Press, New York, 1996.

[2] Matthias Baaz, Agata Ciabattoni, and Christian G. Fermüller. Hypersequent calculi for Gödel logics: a survey. *Journal of Logic and Computation*, 13:1–27, 2003.

[3] Nuel D. Belnap, Jr. Display logic. *Journal of Philosophical Logic*, 11:375–417, 1982.

[4] E. Bencivenga. Free logics. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic: Volume III: Alternatives to Classical Logic*, pages 373–426. Reidel, Dordrecht, 1986.

[5] Kai Brünnler. Deep sequent systems for modal logic. *Archive for Mathematical Logic*, 48(6):551–577, 2009. `http://www.iam.unibe.ch/~kai/Papers/2009dssml.pdf`.

[6] Kai Brünnler and Lutz Straßburger. Modular sequent systems for modal logic. In Martin Giese and Arild Waaler, editors, *Proceedings of TABLEAUX*, volume 5607 of *Lecture Notes in Computer Science*, pages 152–166. Springer, 2009.

[7] J. Michael Dunn. A 'Gentzen' system for positive relevant implication. *The Journal of Symbolic Logic*, 38:356–357, 1974. (Abstract).

[8] M. Fitting and R.L. Mendelsohn. *First-order Modal Logic*, volume 277 of *Synthese library*. Kluwer, Dordrecht, Holland, 1998.

[9] Gerhard Gentzen. Investigations into logical deduction. In M. E. Szabo, editor, *The Collected Papers of Gerhard Gentzen*, pages 68–131. North-Holland Publishing Co., Amsterdam, 1969.

[10] Rajeev Goré, Linda Postniece, and Alwen Tiu. Taming displayed tense logics using nested sequents with deep inference. In Martin Giese and Arild Waaler, editors, *Proceedings of TABLEAUX*, volume 5607 of *Lecture Notes in Computer Science*, pages 189–204. Springer, 2009.

[11] Alessio Guglielmi. Mismatch. `http://cs.bath.ac.uk/ag/p/AG9.pdf`, 2003.

[12] Alessio Guglielmi. A system of interaction and structure. *ACM Transactions on Computational Logic*, 8(1):1–64, 2007.

[13] Ryo Kashima. Cut-free sequent calculi for some tense logics. *Studia Logica*, 53:119–135, 1994.

[14] G. Mints. Cut-elimination theorem in relevant logics. *The Journal of Soviet Mathematics*, 6:422–428, 1976.

[15] Francesca Poggiolesi. The tree-hypersequent method for modal propositional logic. In Jacek Malinowski David Makinson and Heinrich Wansing, editors, *Towards Mathematical Philosophy*, Trends in Logic, pages 9–30. Springer, 2009.

[16] Mehrnoosh Sadrzadeh and Roy Dyckhoff. Positive logic with adjoint modalities: Proof theory, semantics and reasoning about information. *Review of Symbolic Logic (to appear)*, 2010.

[17] A.S. Troelstra and H. Schwichtenberg. *Basic Proof Theory.* Cambridge University Press, 1996.

[18] Luca Viganò. *Labelled Non-Classical Logics.* Kluwer Academic Publishers, Dordrecht, 2000.