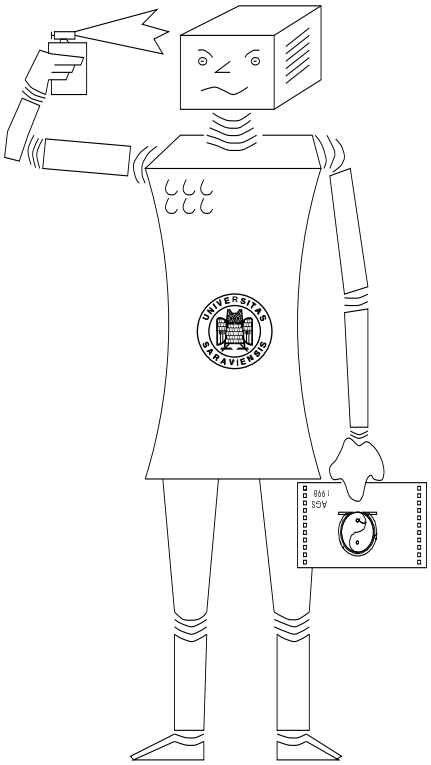
UNIVERSITÄT DES SAARLANDES
FACHBEREICH INFORMATIK
D-66041 SAARBRÜCKEN
GERMANY
WWW: http://www.ags.uni-sb.de/

SEKI Report



Semantical Investigation of Simultaneous Skolemization for First-Order Sequent Calculus

Serge Autexier and Heiko Mantel

SEKI Report SR-98-05

Semantical Investigation of Simultaneous Skolemization for First-Order Sequent Calculus

Serge Autexier
Universität des Saarlandes
FB 14 Informatik
Postfach 15 11 50
66041 Saarbrücken, Germany
E-mail: autexier@ags.uni-sb.de

Heiko Mantel
German Research Center
for Artificial Intelligence (DFKI GmbH)
Stuhlsatzenhausweg 3
66123 Saarbrücken, Germany
E-mail: mantel@dfki.de

Abstract

Simultaneous quantifier elimination in sequent calculus is an improvement over the well-known skolemization. It allows a lazy handling of instantiations as well as of the order of certain reductions. We prove the soundness of a sequent calculus which incorporates a rule for simultaneous quantifier elimination. The proof is performed by semantical arguments and provides some insights into the dependencies between various formulas in a sequent.

1 Introduction

Sequent calculi are a very common search space representation. Originally developed by Gentzen [6] they have been applied in automated deduction, in logic programming, in formal program development, and other areas. During analytic proof search formulas in a sequent are decomposed into sub-formulas in a stepwise manner. The structure of sub-formulas and of formulas which are not decomposed is preserved. The preservation of structure is especially beneficial when user interaction is required. A user can recognize structures which e.g. in the context of formal methods [7] originate from a specification.

The relation between standard presentations of Hilbert type, natural deduction, and sequent calculi has been investigated by Avron [2] for the propositional case. The additional structure in sequent calculi usually provides advantages in proof search. In the presence of quantifiers additional differences between these type of calculi arise. Gentzen's rules for the elimination of quantifiers employ an eager handling of instantiations. This causes a high-degree of non-determinism in proof search which can be avoided by a lazy handling of instantiations with meta-variables together with a computation of instantiations by unification. *Skolemization* [14] is a well-known technique which guarantees that proofs constructed with a lazy handling of instantiations can be validated in general. In the context of sequent calculi, skolemization has been investigated for classical [4] as well as for non-classical logics [13, 10].

The technique for simultaneous quantifier elimination [1] is specific to sequent calculi. It provides an optimization over the usual approach for lazy handling of instantiations. Our algebraic justification of its soundness clarifies the dependencies between the formulas of a sequent in the presence of quantifiers.

After presenting some fundamentals in section 2 we present a sequent calculus $\mathcal{K}$ with a rule for simultaneous quantifier elimination in section 3. We point out its advantages in

comparison to usual handling of quantifiers in sequent calculus proof search. The soundness of $\mathcal{K}$ is demonstrated in section 4 using semantical arguments. We conclude with some remarks on related work.

2 Fundamentals

Basing on [11], we define syntax and semantics of first-order logic. A *signature* Σ is a pair $(\mathcal{F}, \mathcal{P})$ consisting of a set $\mathcal{F}$ of operation symbols and a set $\mathcal{P}$ of predicate symbols. Each $f \in \mathcal{F}$ has an arity $n_f \in \mathbb{N}$ and each $p \in \mathcal{P}$ has an arity $n_p \in \mathbb{N}$. A Σ-*algebra* A has a *carrier set* S_A and assigns to each n_f-ary operation $f \in \mathcal{F}$ a total function $A(f) : (S_A)^{n_f} \to S_A$ and to each predicate $p \in \mathcal{P}$ a n_p-ary relation $A(p) \subseteq (S_A)^{n_p}$. *Constants* are 0-ary operations.

Syntax of First-Order Logic. The set $T_\Sigma(\mathcal{V})$ of *first-order terms* for a signature Σ and a set $\mathcal{V}$ of variables is defined recursively. For each $x \in \mathcal{V}$ holds $x \in T_\Sigma(\mathcal{V})$. If $t_1, \ldots, t_{n_f} \in T_\Sigma(\mathcal{V})$ then for any n_f-ary operation $f \in \mathcal{F}$ holds $f(t_1, \ldots, t_{n_f}) \in T_\Sigma(\mathcal{V})$. The set $\mathit{wff}(\Sigma, \mathcal{V})$ of *first-order formulas* for $\Sigma = (\mathcal{F}, \mathcal{P})$ and $\mathcal{V}$ is defined recursively. For $t_1, \ldots, t_{n_p} \in T_\Sigma(\mathcal{V})$ and $p \in \mathcal{P}$ with arity n_p the expression $p(t_1, \ldots, t_{n_p})$ is an *atomic formula* in $\mathit{wff}(\Sigma, \mathcal{V})$. If $\varphi, \psi \in \mathit{wff}(\Sigma, \mathcal{V})$ and $x \in \mathcal{V}$ then $\neg\varphi, \varphi \wedge \psi, \varphi \vee \psi, \forall x.\varphi, \exists x.\varphi \in \mathit{wff}(\Sigma, \mathcal{V})$ are formulas.

For a term t the function *Var* returns the variables and *Op* the operations which occur in t. The function *free* which returns the free variables of a formula is defined recursively over the structure of formulas, i.e. $\mathit{free}(p(t_1, \ldots, t_{n_p})) = \bigcup_{i=1}^{n_p} \mathit{Var}(t_i)$, $\mathit{free}(\neg\varphi) = \mathit{free}(\varphi)$, $\mathit{free}(\varphi \overset{\wedge}{\vee} \psi) = \mathit{free}(\varphi) \cup \mathit{free}(\psi)$, and $\mathit{free}(\overset{\forall}{\exists} x.\varphi) = \mathit{free}(\varphi) \setminus \{x\}$. *Op* returns for φ the operations which occur in φ.

Semantics of First-Order Logic. The *value* $A(\alpha)(t)$ of a term $t \in T_\Sigma(\mathcal{V})$ and the *value* $A(\alpha)(\varphi)$ of a formula $\varphi \in \mathit{wff}(\Sigma, \mathcal{V})$ for a Σ-algebra A and an assignment $\alpha : \mathcal{V} \to S_A$ where $\mathit{free}(\varphi) \subseteq \mathcal{V}$ is respectively an element of the carrier set S_A or a truth value (*true* or *false*).

- $A(\alpha)(x) = \alpha(x)$ for $x \in \mathcal{V}$,
- $A(\alpha)(f(t_1, \ldots, t_{n_f}) = A(f)(A(\alpha)(t_1), \ldots, A(\alpha)(t_{n_f}))$,
- $A(\alpha)(p(t_1, \ldots, t_{n_p})) = \mathit{true}$ iff $(A(\alpha)(t_1), \ldots, A(\alpha)(t_{n_p})) \in A(p)$,
- $A(\alpha)(\neg\varphi) = \mathit{true}$ iff $A(\alpha)(\varphi) = \mathit{false}$,
- $A(\alpha)(\varphi_1 \overset{\wedge}{\vee} \varphi_2) = \mathit{true}$ iff $A(\alpha)(\varphi_1) = \mathit{true} \overset{\text{and}}{\text{or}} A(\alpha)(\varphi_2) = \mathit{true}$,
- $A(\alpha)(\overset{\forall}{\exists} x.\varphi) = \mathit{true}$ iff $(A(\alpha[a/x])(\varphi) = \mathit{true} \overset{\text{for all}}{\text{for some}} a \in A(s))$.

where $\alpha[a/x]$ is the assignment: $\alpha[a/x](x) = a$ and $\alpha[a/x](y) = \alpha(y)$, if $y \neq x$. A formula φ is *valid in a* Σ-*algebra* A ($A \models_\Sigma \varphi$) iff for any assignment α holds $A(\alpha)(\varphi) = \mathit{true}$. A formula φ is *valid* ($\models_\Sigma \varphi$) iff it is valid in every Σ-algebra.

Substitutions. Let Σ be a signature and $\mathcal{V}$ be a set of variables for Σ. A function $\sigma : \mathcal{V} \to T_{\Sigma(\mathcal{V})}$ is called a *substitution*. The application of a substitution to a formula $\varphi \in \mathit{wff}(\Sigma, \mathcal{V})$ yields a formula $\sigma(\varphi)$, where all free occurrences of variables $x \in \mathcal{V}$ are replaced by $\sigma(x)$. If σ is the identity except for a finite number of variables $x_1, \ldots, x_n$, we denote σ by

$[\sigma(x_1)/x_1, \ldots, \sigma(x_n)/x_n]$. $Dom(\sigma) = \{x_1, \ldots, x_n\}$ is called the *domain* of σ. A substitution σ is *admissible* for φ if for every sub-formula $Qx.\varphi'$ of φ holds $x \notin \sigma(y)$ for all $y \in free(Qx.\varphi')$. We require substitutions to be idempotent and admissible.

The following theorem states a fundamental relationship between substitutions and assignments. For a proof we refer the interested reader to [11].

Theorem 1 [Substitution Theorem]
Let $\mathcal{V}$ be a set of variables for a signature Σ, $\sigma : \mathcal{V} \to T_{\Sigma(\mathcal{V})}$ a substitution, A a Σ-algebra, and $\beta : \mathcal{V} \to S_A$ an assignment. Then for every $t \in T_{\Sigma(\mathcal{V})}$ holds $A(\beta)(\sigma(t)) = A(\alpha)(t)$, where $\alpha : \mathcal{V} \to S_A$ is an assignment defined by $\alpha(x) := A(\beta)(\sigma(x))$ for every $x \in \mathcal{V}$.

We restrict ourselves throughout this report to formulas in *negation-normal form*, i.e. formulas where negation $\neg$ occurs only directly in front of atomic formulas. Using the de-Morgan laws any first-order formula can be transformed into an equivalent formula which is in negation normal form.

Sequents. A *(one-sided) sequent* s is a set Γ of formulas in negation-normal form denoted by $\longrightarrow \Gamma$. We define $free(\longrightarrow \Gamma) = \bigcup_{\varphi \in \Gamma} free(\varphi)$. Given an algebra A and an assignment $\alpha : \mathcal{V} \to S_A$ with $free(s) \subseteq \mathcal{V}$. The *value* $A(\alpha)(s)$ is *true* iff $A(\alpha)(\varphi) = true$ for some $\varphi \in \Gamma$. s is *valid in* an algebra A ($A \models_\Sigma s$) if for all assignments α $A(\alpha)(s) = true$. s is *valid* ($\models_\Sigma s$) if it is valid in all algebras.

A *sequent calculus* is a pair $\langle Ax, Inf \rangle$. Ax is a finite set of axiom schemes each of which is a decidable set of sequents. *Inf* is a finite set of inference rules. Each inference rule consists of a decidable set of pairs $(s_1, \ldots, s_n), s$ where $s_1, \ldots, s_n$ and s are sequents. s is called the *conclusion* and $s_1, \ldots, s_n$ the *premises* of the inference rule. A *principal formula* is a formula that occurs in the conclusion but not in any premise. Formulas which occur in a premise but not in the conclusion are called *side formulas*. All other formulas compose the *context*. Sequent rules can be represented graphically where the conclusion is written underneath the premises and separated from them by a horizontal line. A *derivation* of a sequent s from a set of sequents S is a finite sequence of sequents $s_1, \ldots, s_k$ with $k \geq 1$ and $s_k = s$ such that for each $i \leq k$ holds $s_i \in S$, s_i is an axiom in Ax, or there exist indices $i_1, \ldots, i_n$ such that there is an inference rule in *Inf* with conclusion s_i and premises $s_{i_1}, \ldots, s_{i_n}$. A sequent s is said to be *derivable* from a set of sequents S ($S \vdash s$) if there exists a derivation from S for it.

The one-sided sequent calculus $\mathcal{K}_c$ for formulas in negation normal form[1] is:

$$\frac{}{\longrightarrow \Gamma, \varphi, \neg\varphi}\ ax \qquad \frac{\longrightarrow \Gamma, \varphi_1 \quad \longrightarrow \Gamma, \varphi_2}{\longrightarrow \Gamma, \varphi_1 \wedge \varphi_2}\ \wedge \qquad \frac{\longrightarrow \Gamma, \varphi_1, \varphi_2}{\longrightarrow \Gamma, \varphi_1 \vee \varphi_2}\ \vee \qquad \frac{\longrightarrow \Gamma, \varphi[c/x]}{\longrightarrow \Gamma, \forall x.\varphi}\ \forall^* \qquad \frac{\longrightarrow \Gamma, \varphi[t/x]}{\longrightarrow \Gamma, \exists x.\varphi}\ \exists^{**}$$

$*$ c must not occur in $\longrightarrow \Gamma, \forall x.\varphi$ (*Eigenvariable condition*). $**$ t may be any term.

In *analytic proof search* with $\mathcal{K}_c$ one starts with the sequent to be proven and reduces it by application of rules until the *ax*-rule is applicable.

3 Simultaneous Quantifier Elimination

In this section we first introduce a calculus with a rule for conventional skolemization and point out its advantage for proof search with respect to the classical sequent calculus (sec-

[1] The restriction to formulas in negation-normal form and to one-sided sequents has only presentational purposes. The theory presented in this report could also be developed for arbitrary formulas and two-sided sequents.

tion 3.1). Then we introduce in section 3.3 a calculus with a simultaneous skolemization rule which overcomes some restrictions still present in the conventional skolemization rule. This is illustrated by a comparison in section 3.4 of both conventional and simultaneous skolemization.

3.1 Conventional Skolemization

The quantifier rules of $\mathcal{K}_c$ cause problems in analytic proof search. Whenever the $\exists$-rule is applied a term t must be guessed immediately. To postpone the choice of t until more information about good choices of t are at hand is a superior approach. In order to do so the rule $\exists'$ depicted below inserts a free variable X (sometimes also called *meta-variable*) which is implicitly existentially quantified. Thus, it may be instantiated later during proof search. However, precautions must be taken to guarantee the correctness of the resulting proofs because not all possible instantiations are admissible. *Skolemization* is used for this purpose. The rule *Skolem* inserts a *skolem-term* consisting of a new function symbol with all free variables of the sequent as arguments. Free variables may be instantiated during proof search. The instantiation of a variable affects all parts of a derivation where the variable occurs, i.e. *Inst* is a rewrite rule on derivations rather than an ordinary sequent rule. The *occur-check* ensures that a variable X can only be substituted by terms t which do not contain X.

$$\frac{\longrightarrow\Gamma,\varphi[X/x]}{\longrightarrow\Gamma,\exists x.\varphi}\ \exists' \qquad \frac{\longrightarrow\Gamma,\varphi[f(\vec{Z})/x]}{\longrightarrow\Gamma,\forall x.\varphi}\ \mathit{Skolem}^{*} \qquad \boxed{X} \rightsquigarrow \boxed{t}\ \mathit{Inst}(X,t)^{**}$$

* f must not occur in $s = \longrightarrow \Gamma, \forall x.\varphi$ and Z must contain all free variables of s.
** X must not occur in t and all variables and operations in t must also occur in the left-hand side proof-tree.

The calculus $\mathcal{K}_{sk}$ results from $\mathcal{K}_c$ by adding the rules $\exists'$, *Skolem*, and *Inst* while the rules $\exists$ and $\forall$ are removed.

The use of free variables and skolemization allows to postpone the instantiation until it can be computed, e.g. by unification. Nevertheless, if multiple quantified formulas occur in a sequent a principal formula must be determined. Although in some cases a principal formula can be chosen in a safe way, in general, the right order of reductions cannot be calculated from a sequent. This is demonstrated by the following example.

Example 2. Below a $\mathcal{K}_{sk}$-derivation with six rule applications is depicted.

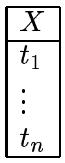
$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\longrightarrow\varphi(X_1,f_1(X_1),Z_1),\neg\varphi(Z_2,X_2,f_2(X_1,X_2))}{\longrightarrow\varphi(X_1,f_1(X_1),Z_1),\exists z_2.\neg\varphi(z_2,X_2,f_2(X_1,X_2))}\ \exists'}{\longrightarrow\exists z_1.\varphi(X_1,f_1(X_1),z_1),\exists z_2.\neg\varphi(z_2,X_2,f_2(X_1,X_2))}\ \exists'}{\longrightarrow\exists z_1.\varphi(X_1,f_1(X_1),z_1),\forall y_2.\exists z_2.\neg\varphi(z_2,X_2,y_2)}\ \mathit{Skolem}}{\longrightarrow\exists z_1.\varphi(X_1,f_1(X_1),z_1),\exists x_2.\forall y_2.\exists z_2.\neg\varphi(z_2,x_2,y_2)}\ \exists'}{\longrightarrow\forall y_1.\exists z_1.\varphi(X_1,y_1,z_1),\exists x_2.\forall y_2.\exists z_2.\neg\varphi(z_2,x_2,y_2)}\ \mathit{Skolem}}{\longrightarrow\exists x_1.\forall y_1.\exists z_1.\varphi(x_1,y_1,z_1),\exists x_2.\forall y_2.\exists z_2.\neg\varphi(z_2,x_2,y_2)}\ \exists'$$

The proof attempt would have failed if we first had reduced the second formula.

3.2 A Visualization of Conventional Skolemization

Illustrative Elements. We aim at illustrating what skolemization together with the occur-check achieves during analytic proof search. To this end we introduce a graphical notation for

variables and use it to illustrate the behavior of the calculus rules. The illustrative element we introduce are called *restriction boxes* in which for a given meta-variable X those terms are collected, which are not allowed to be instantiated for X. E.g., the following box is used as a graphical element to represent, that the terms $t_1, \ldots, t_n$ are not allowed to be instantiated for X.



For each (partial) $\mathcal{K}_{sk}$-proof there is an *environment* composed of restriction boxes. During analytical proof-search such an environment is successively constructed and updated. There are three basic operations: one to create a new restriction box for a variable and two update operations. Each of the operation corresponds to a rule of $\mathcal{K}_{sk}$.

1. Creation of a restriction box: This operation creates an empty restriction box for a given meta-variable X. This operation corresponds to the $\exists'$-rule. E.g., during the proof-step

$$\frac{\longrightarrow \forall y_1.\exists z_1.\varphi(X_1,y_1,z_1),\exists x_2.\forall y_2.\exists z_2.\neg\varphi(z_2,x_2,y_2)}{\longrightarrow \exists x_1.\forall y_1.\exists z_1.\varphi(x_1,y_1,z_1),\exists x_2.\forall y_2.\exists z_2.\neg\varphi(z_2,x_2,y_2)}\ \exists'$$

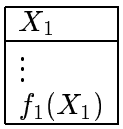
   the restriction box

   is introduced into the environment. This restriction box expresses, that X can be instantiated without any restrictions.

2. Updating of existing restriction boxes: This operation introduces a term in a restriction box and corresponds to the *Skolem* rule: E.g., during the proof step

$$\frac{\longrightarrow \exists z_1.\varphi(X_1,f_1(X_1),z_1),\exists x_2.\forall y_2.\exists z_2.\neg\varphi(z_2,x_2,y_2)}{\longrightarrow \forall y_1.\exists z_1.\varphi(X_1,y_1,z_1),\exists x_2.\forall y_2.\exists z_2.\neg\varphi(z_2,x_2,y_2)}\ \mathit{Skolem}$$

   the restriction box for X_1 is updated by inserting $f_1(X_1)$ yielding

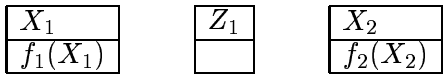


   .

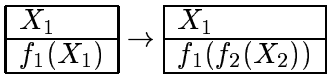
   This expresses the restriction that $f_1(X_1)$ must not be instantiated for X_1.

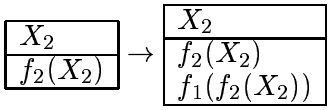
3. The third operation is an updating operation which corresponds to the application of the *Inst*-rule. If a meta-variable X is instantiated by a term t, then this is only allowed, if the variable and skolem-functions occurring in t do not occur in the restriction box of X. If the instantiation is valid, then (1) the instantiation is applied to all restriction boxes in the actual environment, in which X occurs and (2) all the terms occurring in

the restriction box of X are introduced in the restriction box of any variable occurring in t. E.g., assume we have the following environment:

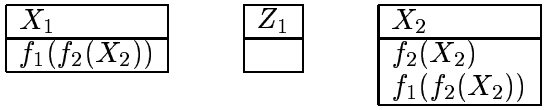


Applying the instantiation $\mathit{Inst}(X_1, f_2(X_2))$ entails the following updates:

(1) 

(2) 

The resulting environment is



3.3 A Rule for Simultaneous Quantifier Elimination

We now define a rule for simultaneous quantifier elimination. To define this rule in a general way, we want to be able to eliminate arbitrarily many leading quantifiers of formulas in a sequent. To this end we define the notion of *quantifier lists ql* and use them to describe the leading quantifiers of some formula. Quantifier lists ql are defined recursively starting from the empty list ϵ and for a variable x by $\forall x.$ql' and $\exists x.$ql'. E.g., the quantified formula $\forall x_1.\exists y_1 \forall z_1.\varphi$ can be decomposed in the following quantifier lists and formulas:

Quantifier List	Formula
ϵ	$\forall x_1.\exists y_1 \forall z_1.\varphi$,
$\forall x_1.\epsilon$	$\exists y_1 \forall z_1.\varphi$,
$\forall x_1.\exists y_1.\epsilon$	$\forall z_1.\varphi$ and
$\forall x_1.\exists y_1.\forall z_1.\epsilon$	φ

Furthermore, in order to simplify the following definitions, we assume generators *vgen* and *fgen* which respectively generate new symbols for variables and operations on every call.

We define the *quantifier elimination function* QE which takes a quantifier list ql, a formula φ, and a set Z of variables as arguments and returns a formula. ql determines which quantifiers shall be eliminated from φ. Z is used in order to determine the arguments of skolem functions.

- $\mathrm{QE}(\epsilon, \varphi, Z) := \varphi$,
- $\mathrm{QE}(\forall x.\mathrm{ql}, \varphi, Z) := \mathrm{QE}(\mathrm{ql}, \varphi[f(\vec{Z})/x], Z)$, where $f := \mathit{fgen}$ is new.
- $\mathrm{QE}(\exists x.\mathrm{ql}, \varphi, Z) := \mathrm{QE}(\mathrm{ql}, \varphi[X/x], Z \cup \{X\}$ where $X := \mathit{vgen}$ is new.

The rule *SQEl* for *simultaneous quantifier elimination* is depicted below.

$$\frac{\longrightarrow \Gamma, \psi_1, \ldots, \psi_n}{\longrightarrow \Gamma, \mathrm{ql}_1.\varphi_1, \ldots, \mathrm{ql}_n.\varphi_n}\ SQEl^*$$

$*$ For each i $(1 \leq i \leq n)$ must hold $\psi_i = \mathrm{QE}(\mathrm{ql}_i, \varphi_i, \mathit{free}(\longrightarrow \Gamma, \mathrm{ql}_1.\varphi_1, \ldots, \mathrm{ql}_n.\varphi_n))$.

The calculus $\mathcal{K}$ results from $\mathcal{K}_{sk}$ by replacing the rules $\exists'$ and *Skolem* by *SQEl* . $\mathcal{K}$ is complete with respect to $\mathcal{K}_{sk}$, i.e. for every sequent s which is $\mathcal{K}_{sk}$-derivable there is a $\mathcal{K}$-derivation, since $\exists'$ and *Skolem* can be simulated by *SQEl*. However, *SQEl* has advantages compared to these rules because one does not need to bother about the order of certain reductions.

Example 3. We reduce our example sequent by *SQEl*.

$$\frac{\longrightarrow \varphi(X_1, f_1(X_1), Z_1), \neg\varphi(Z_2, X_2, f_2(X_2))}{\longrightarrow \exists x_1.\forall y_1.\exists z_1.\varphi(x_1, y_1, z_1), \exists x_2.\forall y_2.\exists z_2.\neg\varphi(z_2, x_2, y_2)}\ SQEl$$

A comparison to example 2 shows the advantages of simultaneous quantifier elimination. First, one does not need to worry about the order of quantifier eliminations. Second, the skolem term in the second formula depends only on X_2 and not on both X_1 and X_2 as in example 2. This shows that the quantifier elimination of different formulas in a sequent do not depend on each other. For a more detailed comparison see section 3.4

The notion of derivations in a calculus with free variables and an instantiation rule differs from the one in a calculus without these concepts. So far we have used $\mathcal{K}$-derivations intuitively. In the sequel, we present a formal definition of $\mathcal{K}$-derivations which meet our intuitive understanding of $\mathcal{K}$.

A *$\mathcal{K}$-derivation* $\mathcal{D}$ from a set of sequents S, the assumptions, is a sequence of *levels*. Each level is a pair which consists of a sequence of sequents and a substitution. A derivation must fulfill the conditions 1–3 presented below.

$$\begin{aligned}\mathcal{D} = \langle \quad & (\langle s^1_{k_1}, \ldots, s^1_n\rangle, \quad \sigma_1),\\ & (\langle s^2_{k_2}, \ldots, s^2_{k_1}, \ldots, s^2_n\rangle, \quad \sigma_2),\\ & \ldots\\ & (\langle s^m_1, \ldots, s^m_{k_2}, \ldots, s^m_{k_1}, \ldots, s^m_n\rangle, \quad \sigma_m)\rangle\end{aligned}$$

1. For each $1 \leq j \leq m$ and $k_j \leq i \leq n$, i.e. for each sequent in $\mathcal{D}$, holds:
   if $j > 1$ and $i \geq k_{j-1}$ then $s^j_i = \sigma_j(s^{j-1}_i)$.

2. For each $1 \leq j \leq m$, $n_j \leq i \leq n$ holds: s^j_i is j-derivable (defined below) in $\mathcal{D}$.

3. σ_1 is the identity and for each $j > 1$ there is a substitution σ such that $\sigma_j = \sigma \circ \sigma_{j-1}$.

The above definition of $\mathcal{K}$-derivations captures the notion of analytic proof search (with goal s^1_n). The application of the instantiation rule with parameter σ on a derivation $\mathcal{D}$ with m levels adds a new level to $\mathcal{D}$ which is constructed by the application of σ on the former last level and by the substitution $\sigma \circ \sigma_m$. An analytic application of the rule $\wedge$, $\vee$, or *SQEl* adds a new sequent to level m. Intuitively, condition 1 ensures that the instantiation rule is applied correctly as a rewrite rule. In level j the sequents $s^j_{k_{j-1}}, \ldots s^j_n$ result from $s^{j-1}_{k_{j-1}}, \ldots s^{j-1}_n$ by an application of a substitution. Condition 2 requires that each sequent in the highest level (i.e. level m) of $\mathcal{D}$ can be justified. Condition 3 guarantees that a substitution is constructed by adapting the substitution of the previous level. Although proof search starts with s^1_n as

goal, $\mathcal{D}$ is a derivation of s_n^m. This reflects that free variables can be regarded as implicitly existentially quantified.

We now define the "*j-derivability of a sequent*" in a $\mathcal{K}$-proof: s_i^j is *j-derivable* in $\mathcal{D}$ where S are the assumptions if one of the following four conditions is fulfilled.

1. $j > 1$, $i \geq k_{j-1}$, and s_i^{j-1} is $(j-1)$-derivable.
2. $j > 1$ and $s_i^j = \sigma_j(s)$ for some assumption $s \in S$.
3. s_i^j is an axiom.
4. There exist indices $i_1, \ldots, i_l$ (all of which must be smaller than i) such that one of the inference rules $\wedge$, $\vee$, or *SQEl* can be instantiated with $s_{i_1}^j, \ldots s_{i_l}^j$ as premises and with s_i^j as conclusion.

Intuitively, a sequent is j-derivable if it can be justified already by looking at the first j levels of $\mathcal{D}$.

Lemma 4. *If a sequent is $(j-1)$-derivable according to one of the cases 2, 3, or 4 then it is also j-derivable respectively according to 2, 3, or 4 unless it has been derived by an application of* SQEl *where σ_j affects one of the variables introduced by* SQEl.

3.4 Visualization of Simultaneous Quantifier Elimination

Analogously to the restriction boxes for the $\mathcal{K}_{sk}$-calculus, restriction boxes can be defined for the $\mathcal{K}$-calculus; the boxes remain the same, but the construction of the environment is slightly altered. Indeed, the instantiation rule corresponds to the same update operation on environments as described in section 3.2. The two other operations, namely the construction of a restriction box for a new variable and the insertion of a new term into restriction boxes corresponds to the QE function as it is applied in the side condition of the *SQEl* rule.

- In the case $\mathrm{QE}(\epsilon, \varphi, Z)$ the construction terminates.
- In the case $\mathrm{QE}(\forall x.\mathrm{ql}, \varphi, Z) := \mathrm{QE}(\mathrm{ql}, \varphi[f(\vec{Z})/x], Z)$, where $f := \mathit{fgen}$ is new, the skolem term $f(\vec{Z})$ is inserted into the restriction boxes of all variables in $\vec{Z}$.
- In the case $\mathrm{QE}(\exists x.\mathrm{ql}, \varphi, Z) := \mathrm{QE}(\mathrm{ql}, \varphi[X/x], Z \cup \{X\})$ where $X := \mathit{vgen}$ is new, a new restriction box for X is inserted into the environment.

We now illustrate the difference in proof search between $\mathcal{K}_{sk}$ and $\mathcal{K}$ by the proof of the sequent

$$\longrightarrow \exists x_1.\forall y_1.\exists z_1.\varphi(x_1, y_1, z_1), \exists x_2.\forall y_2.\exists z_2.\neg\varphi(z_2, x_2, y_2) \quad .$$

$\mathcal{K}_{sk}$	$\mathcal{K}$
Performing the quantifier-elimination in the "right order" results in the sequent $\longrightarrow \varphi(X_1, f_1(X_1), Z_1), \neg\varphi(Z_2, X_2, f_2(X_1, X_2))$ and the following environment: $\begin{array}{\|l\|}\hline X_1 \\ \hline f_1(X_1) \\ f_2(X_1, X_2) \\ \hline \end{array}$ $\begin{array}{\|l\|}\hline Z_1 \\ \hline \\ \hline \end{array}$ $\begin{array}{\|l\|}\hline X_2 \\ \hline f_2(X_1, X_2) \\ \hline \end{array}$ $\begin{array}{\|l\|}\hline Z_2 \\ \hline \\ \hline \end{array}$	Performing the simultaneous quantifier elimination results in the sequent $\longrightarrow \varphi(X_1, f_1(X_1), Z_1), \neg\varphi(Z_2, X_2, f_2(X_2))$ and the following environment: $\begin{array}{\|l\|}\hline X_1 \\ \hline f_1(X_1) \\ \hline \end{array}$ $\begin{array}{\|l\|}\hline Z_1 \\ \hline \\ \hline \end{array}$ $\begin{array}{\|l\|}\hline X_2 \\ \hline f_2(X_2) \\ \hline \end{array}$ $\begin{array}{\|l\|}\hline Z_2 \\ \hline \\ \hline \end{array}$

At this stage the restriction on the further proof search applied by the conventional quantifier elimination rule becomes clearly visible: X_1 can not be instantiated by a term in which $f_2(X_1, X_2)$ occurs, although this is not necessary as simultaneous quantifier elimination shows. Furthermore, if the quantifiers were eliminated in the wrong order, the proof could not be completed and backtracking would have been necessary. In general, for any environment constructed by the conventional skolemization rule one can construct an equivalent or even less restrictive environment obtained by simultaneous skolemization but not the other way round. E.g, applying the instantiation $\mathit{Inst}(X_2, f_1(X_1))$ results in the following environments ("conventional" and "simultaneous" respectively).

$\begin{array}{\|l\|}\hline X_1 \\ \hline f_1(X_1) \\ f_2(X_1, f_1(X_1)) \\ \hline \end{array}$ $\begin{array}{\|l\|}\hline Z_1 \\ \hline \\ \hline \end{array}$ $\begin{array}{\|l\|}\hline X_2 \\ \hline f_2(X_1, f_1(X_1)) \\ \hline \end{array}$ $\begin{array}{\|l\|}\hline Z_2 \\ \hline \\ \hline \end{array}$ and $\begin{array}{\|l\|}\hline X_1 \\ \hline f_1(X_1) \\ f_2(f_1(X_1)) \\ \hline \end{array}$ $\begin{array}{\|l\|}\hline Z_1 \\ \hline \\ \hline \end{array}$ $\begin{array}{\|l\|}\hline X_2 \\ \hline f_2(f_1(X_1)) \\ \hline \end{array}$ $\begin{array}{\|l\|}\hline Z_2 \\ \hline \\ \hline \end{array}$

But the substitution $[X_1/f_2(X_2)]$ is only applicable in the "simultaneous" environment, since it does not restrict the instantiation of X_1, whereby the "conventional" environment does.

From these informal considerations about the differences between conventional skolemization and simultaneous skolemization the advantage of the latter becomes visible. Indeed, simultaneous skolemization avoids having to decide for a "right" order of quantifier eliminations during proof search, although the necessary restrictions for variable instantiations are captured. Hence, simultaneous skolemization reduces the search space and thus is much more adequate for automation than the conventional skolemization rule is.

Remark 5. In the Isabelle system [12] for example a dual technique to skolemization is employed. According to this technique a universally quantified formula $\forall x.\varphi(x)$ is reduced to $\bigwedge x.\varphi(x)$ and an existentially quantified formula $\exists x.\varphi(x)$ to $\varphi(?x)$ where $\bigwedge$ is a meta-logic quantifier and $?x$ is a (higher-order) meta-variable. Due to lifting over quantifiers meta-variables receive arguments which essentially determine which constants may be used in instantiations. The restrictions are represented positively by the constants of the signature which may be used by an instantiation. This causes close interdependencies between formulas in a sequent which are not present when skolemization is applied. Before a constant

may be instantiated, i.e. appear as an argument of a meta-variable, the corresponding quantifier must have been reduced already. In skolemization restrictions are represented negatively by variables. Compared to constants variables have the capability to propagate constraints. Therefore, it appears to be quite difficult to develop an optimized handling of quantifiers equivalent to *SQEl* for the Isabelle technique. For details of the technique we refer the interested reader to [12].

3.5 Analytical Environment Construction

In section 3.2 we have illustrated the effects of the $\mathcal{K}_{sk}$-rules $\exists'$, *Skolem* and *Inst* on a given environment as well as their restrictions in the case of the rule *Inst* by examples. In section 3.4 these effects and restrictions have been illustrated for the $\mathcal{K}$-rules *SQEl* and *Inst*. We now define formally how an environment is constructed during analytical proof-search with the calculi $\mathcal{K}_{sk}$ and $\mathcal{K}$.

Definition 6 [Environment Construction for QE on formulas]
The function QE has as parameters a quantifier list ql, a formula φ and a set of variables Z. The stepwise modification of the environment by $QE(ql, \varphi, Z)$ is defined inductively.

- *The initial environment contains for each variable $X \in Z$ an empty restriction box* $\boxed{\begin{array}{|l}\hline X \\ \hline \\ \hline\end{array}}$.
- *Case $QE(\epsilon, \varphi, Z)$: The construction terminates with the current environment.*
- *Case $QE(\exists x.ql, \varphi, Z)$: If E is the current environment then the new environment is*

$$E \cup \{\boxed{\begin{array}{|l}\hline X \\ \hline \\ \hline\end{array}}\}$$

  where X is the new meta-variable introduced for x and the construction is continued with $QE(ql, [X/x]\varphi, Z)$.

- *Case $QE(\forall x.ql, \varphi, Z)$: If E is the current environment then the new environment is*

$$\{\boxed{\begin{array}{|l}\hline X \\ \hline \vdots \\ f(\vec{Z}) \\ \hline\end{array}} \mid \boxed{\begin{array}{|l}\hline X \\ \hline \vdots \\ \hline\end{array}} \in E\}$$

  where $f(\vec{Z})$ is the skolem term introduced for x and the construction is continued with $QE(ql, [f(\vec{Z}/x]\varphi, Z)$.

Definition 7 [Environment Construction for QE on sequents]
Let $s = \longrightarrow \Gamma, ql_1.\varphi_1, \ldots, ql_n.\varphi_n$ be a sequent with free variables in Z. We define the environment constructed for the application of QE on s as the union of the environments for $QE(ql_i, \varphi_i, Z)$ such that the restriction boxes for variables in Z are combined. If $\boxed{\begin{array}{|l}\hline X \\ \hline sk_1, \ldots, sk_n \\ \hline\end{array}}$ *occurs in the environment for $QE(ql_i, \varphi_i, Z)$ and* $\boxed{\begin{array}{|l}\hline X \\ \hline sk_{n+1}, \ldots, sk_{n+m} \\ \hline\end{array}}$ *in the environment for $QE(ql_j, \varphi_j, Z)$ $(i \neq j)$, then the union contains*

X
$sk_1, \ldots, sk_n, sk_{n+1}, \ldots, sk_{n+m}$

.

We now define inductively the function *Env* which constructs an environment for a derivation in $\mathcal{K}_{sk}$ and $\mathcal{K}$.

Definition 8 [Environment Construction for a Derivation]
Both calculi $\mathcal{K}_{\mathrm{sk}}$ and $\mathcal{K}$ have derivations as defined in section 3.3, i.e., a derivation is a sequence of levels, where each level is a pair composed of a sequence of sequents and a substitution. Analytical proof-search for a sequent s starts with the initial $\mathcal{K}_{\mathrm{sk}}$- or $\mathcal{K}$-derivation

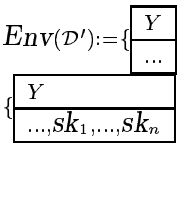
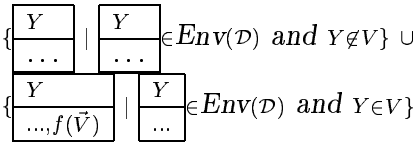
$$\mathcal{D}_0 = \langle(\langle s\rangle), \sigma_1\rangle$$

where σ_1 is the identity substitution. The environment $Env(\mathcal{D}_0)$ for this initial derivation contains only restriction boxes with an empty restriction list for the meta-variables occurring in s. I.e., if X is a meta-variable in s, then the environment $Env(\mathcal{D}_0)$ contains

X

.

Let the E be the environment for a derivation $\mathcal{D}$. Let $\mathcal{D}'$ be a derivation of s which results from $\mathcal{D}$ by the application of a calculus rule. We define the environment for $\mathcal{D}'$. In the left hand side column of the following table $\mathcal{K}_{\mathrm{sk}}$- and in the right hand side column $\mathcal{K}$-derivations are considered.

<table>
<tr><th>$\mathcal{K}_{\mathrm{sk}}$</th><th>$\mathcal{K}$</th></tr>
<tr><td colspan="2">Application of the rules $\wedge$-, $\vee$- and ax- results in a derivation $\mathcal{D}'$ with the same environment. Hence $Env(\mathcal{D}') := Env(\mathcal{D})$.</td></tr>
<tr><td>Rule $\exists'$ where a new X is introduced
$Env(\mathcal{D}') := Env(\mathcal{D}) \cup \{$  $\}$</td><td rowspan="2">Rule SQEl: Let Z be the meta-variables of the sequent s and E' be the environment obtained for the application of QE on s as it is used in the side condition of the rule.
$Env(\mathcal{D}') := \{$  </td></tr>
<tr><td>Rule Skolem where $f(\vec{V})$ is introduced with f new and V the meta-variables occurring in the sequent
$Env(\mathcal{D}') := \{$  </td></tr>
</table>

The application of the Inst*-rule results for both calculi in the same restrictions and the same effects.*

- *Restriction: $Inst(X,t)$ is only applicable if X does not occur in t. Due to our construction of environments this is equivalent to that variables and skolem-terms occurring in t do not occur in the restriction box of X, i.e.*

$$\forall \begin{array}{|c|}\hline X \\ \hline s_1 \ldots s_n \\ \hline \end{array} . \forall s_i \; . \; s_i \text{ does not occur in } t$$

- *Effect: Let the restriction box of X in $Env(\mathcal{D})$ be* $\begin{array}{|c|}\hline X \\ \hline t_1 \ldots t_n \\ \hline \end{array}$. *Then:*

$$Env(\mathcal{D}') := \{[X/t]\begin{array}{|c|}\hline Y \\ \hline \vdots \\ \hline \end{array} \in Env(\mathcal{D}) \mid Y \notin Var(t)\} \quad \cup \quad \{[X/t](\begin{array}{|c|}\hline Y \\ \hline \vdots \\ \hline t_1 \\ \vdots \\ t_n \\ \hline \end{array}) \mid \begin{array}{|c|}\hline Y \\ \hline \vdots \\ \hline \end{array} \in Env(\mathcal{D}) \text{ and } Y \in Var(t)\}$$

4 Semantical Justification

In this section we present a correctness proof for $\mathcal{K}$ using semantical arguments. For this purpose an auxiliary calculus $\mathcal{K}_{aux}$ is defined which allows to reason about sequents with substitutions. The explicitly stated substitutions are used for meta-level arguments only. We prove the soundness of $\mathcal{K}_{aux}$ and then conclude the soundness of $\mathcal{K}$ from that. In the process we introduce orderings on constants and variables, an approach which is motivated by orderings on positions in the context of matrix characterizations. [3, 16]

4.1 An auxiliary Calculus $\mathcal{K}_{aux}$

Sequents with Substitutions. A *sequent with substitution* s is a pair $\longrightarrow \Gamma; \sigma$ consisting of a sequent $\longrightarrow \Gamma$ and a substitution σ. We define $free(s) = \bigcup_{\varphi \in \Gamma} free(\sigma(\varphi))$. The *value* $A(\alpha)(s)$ is *true* in an algebra A under an assignment $\alpha : \mathcal{V} \to S_A$ where $free(s) \subseteq \mathcal{V}$ iff $A(\alpha)(\longrightarrow \sigma(\Gamma)) = true$.

Below, we define the *auxiliary quantifier elimination function* QE_{aux} which takes a quantifier list ql, a formula φ, a set O of constants and variables, and a binary relation $\ll$ over O as arguments. ql determines which quantified variables shall be eliminated from φ. In O the set of all constants and variables introduced during the elimination are collected while a relation over these symbols is collected in $\ll$. QE_{aux} returns a triple consisting of a formula φ', a set O' of constants and variables, and an ordering $\ll'$ over O'.

- $\mathrm{QE}_{aux}(\epsilon, \varphi, O, \ll) := \langle \varphi, O, \ll \rangle$,
- $\mathrm{QE}_{aux}(\forall x.\mathrm{ql}, \varphi, O, \ll) := \mathrm{QE}_{aux}(\mathrm{ql}, \varphi[c/x], O \cup \{c\}, \ll \cup \{(o,c) \mid \forall o \in O\})$,
  where $c := fgen$ is new.

- $\mathrm{QE}_{aux}(\exists x.\mathrm{ql}, \varphi, O, \ll) := \mathrm{QE}_{aux}(\mathrm{ql}, \varphi[X/x], O \cup \{X\}, \ll \cup\{(o, X) \mid \forall o \in O\})$
  where $X := vgen$ is new.

Note 9. The restriction boxes used for the visualization are similar to the ones for $\mathcal{K}_{aux}$ and for $\mathcal{K}$. However, there is a slight difference in the construction and update operations, since the substitution of a sequent with substitution has to be taken into account. The formal definition of the construction and update operations can be found in definition 15.

Example 10. The value of $(\forall x_1 \exists y_1 \forall z_1, \varphi(x_1, y_1, z_1), \emptyset, \emptyset)$ under QE_{aux} (with the appropriate symbols generated by *vgen* and *fgen*) is

$$\langle \varphi(X_1, c_1, Z_1), \{X_1, c_1, Z_1\}, \{(X_1, c_1), (X_1, Z_1), (c_1, Z_1)\}\rangle.$$

Orderings on Constants and Variables. QE_{aux} eliminates quantifiers in the order in which they occur in ql. This order is represented by the relation $\ll$ on the variables and constants introduced during elimination which is returned by QE_{aux}. Clearly, $\ll$ is an ordering, the *quantifier list ordering*.

For a set of variables $\mathcal{V}$, a set of constants $\mathcal{C}$, and a substitution σ we define two relations $\sim \subseteq \mathcal{V} \times \mathcal{V}$ and $\sqsubset \subseteq (\mathcal{V} \cup \mathcal{C}) \times \mathcal{V}$ as the minimal relations such that:

- for any $u, v \in \mathcal{V}$ if $\sigma(u) = v$ then $u \sim v$,
- for any $u \in \mathcal{V}$ and $v \in \mathcal{C} \cup \mathcal{V}$ if v occurs in $\sigma(u)$ and $\sigma(u) \neq v$ then $v \sqsubset u$,
- and for any $u, v \in \mathcal{V}$ if $v \sqsubset u$ and $u \sim u'$ then $v \sqsubset u'$.

We combine given orderings $\sqsubset$ and $\ll$ to a relation $\lhd \subseteq (\mathcal{C} \cup \mathcal{V})^2$, i.e. $\lhd = (\sqsubset \cup \ll)^+$ where $^+$ denotes the transitive closure. Indicating that $\lhd$ is intended to be usually irreflexive we call it a *reduction ordering*. If $\lhd$ is a reduction ordering over some set O of variables and constants and $O' \subseteq O$, then the *restriction* of $\lhd$ to O' is defined by $\lhd_{O'} := \{(o, o') \mid (o, o') \in \lhd \text{ and } o, o' \in O'\}$.

Calculus for Sequents with Substitutions. The auxiliary calculus $\mathcal{K}_{aux}$ is depicted below. Substitutions are explicitly stated and do not change in a $\mathcal{K}_{aux}$-proof. This is not problematic since we only reason about the soundness of the calculus but do not use it for proof search. Note, that in contrast to $\mathcal{K}$ in $\mathcal{K}_{aux}$ no skolem-terms are introduced during quantifier elimination.

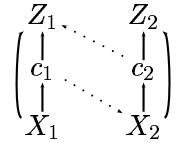
$$\frac{}{\longrightarrow \Gamma, \varphi, \neg\varphi; \sigma}\, ax \qquad \frac{\longrightarrow \Gamma, \varphi_1; \sigma \quad \longrightarrow \Gamma, \varphi_2; \sigma}{\longrightarrow \Gamma, \varphi_1 \wedge \varphi_2; \sigma}\, \wedge \qquad \frac{\longrightarrow \Gamma, \psi_1, \ldots, \psi_n; \sigma}{\longrightarrow \Gamma, \mathrm{ql}_1.\varphi_1, \ldots, \mathrm{ql}_n.\varphi_n; \sigma}\, SQEl^*_{aux} \qquad \frac{\longrightarrow \Gamma\sigma'; \sigma}{\longrightarrow \Gamma; \sigma}\, Subst^{**}$$

* For each $i \in \{1, \ldots, n\}$ holds $\langle \psi_i, O_i, \ll_i \rangle = \mathrm{QE}_{aux}(ql_i, \varphi_i, free(\longrightarrow \Gamma, \psi_1, \ldots, \psi_n), \emptyset)$, $\sqsubset$ the ordering from σ, $\ll \; = \; (\bigcup_{i=1}^n \ll_i)$, $\lhd = (\sqsubset \cup \ll)^+$, $O = \bigcup_{i=1}^n O_i$ the set of (free) variables and constants occurring in the premise and $\lhd_O$ is an irreflexive ordering.

** Where $\sigma \circ \sigma' = \sigma$ holds.

Example 11.

The orderings $\ll$, $\sqsubset$, and $\lhd_O$ for the following rule application are depicted in the diagram to the right. $\ll$ is symbolized by solid arrows and $\sqsubset$ by dashed arrows.



$$\frac{\longrightarrow \varphi(X_1, c_1, Z_1), \neg\varphi(X_2, c_2, Z_2); \{Z_2/X_1, c_1/X_2, c_2/Z_1\}}{\longrightarrow \exists x_1.\forall y_1.\exists z_1.\varphi(x_1, y_1, z_1), \exists x_2.\forall y_2.\exists z_2.\neg\varphi(x_2, y_2, z_2); \{Z_2/X_1, c_1/X_2, c_2/Z_1\}}\, SQEl$$

4.2 Visualization of Simultaneous Quantifier Elimination

We illustrate the diagram from example 11 using restriction boxes. Ignoring the substitution, we have the following environment for the sequent $\longrightarrow \varphi(X_1, c_1, Z_1), \neg\varphi(X_2, c_2, Z_2)$.

X_1	Z_1	X_2	Z_2
c_1		c_2	

The restrictions result from the formula tree of the formula in the conclusion of the rule. Indeed, the universally quantified variables y_1 and y_2 depend respectively on the existentially quantified variables x_1 and x_2. Hence, the *Eigenvariables* c_1 and c_2 respectively introduced for y_1 and y_2 occur in the restriction boxes of the meta-variables introduced for x_1 and x_2. This ensures the invariance of that the *Eigenvariables* are always new constants under applications of the *Inst* rule (*Eigenvariable* condition is always fulfilled).

The environment above represents roughly the solid lines of the diagram. Applying the substitution (which corresponds to the introduction of the dashed lines in the diagram) on this environment results in the following transformations of the environment:

1. Substituting c_1 for X_2 integrates c_2 into the restriction box of X_1 since c_1 occurs in the restriction box of X_1 and hence X_1 inherits the restrictions from X_2. This results in:

X_1
c_2
c_1

   .

2. Substituting c_2 for Z_1 does not change anything, since the restriction box of Z_1 is empty.

3. And finally the substitution of X_1 by Z_2 identifies both variables by integrating the whole restrictions of X_1 into the restrictions of the remaining variable Z_2

Z_2
c_1, c_2

   .

   Note that the substitution of a variable by another variable corresponds to the equivalence link $\sim$ established between the two variables during the definition of the reduction ordering $\lhd$. However, this is not represented in the diagram, since the diagram represents only the mixture of quantifier list ordering and the substitution ordering.

4. These substitutions result in the final environment

X_1	Z_1	X_2	Z_2
c_1		c_2	c_1
c_2			c_2

4.3 Analytical Environment Construction

We define, when an is compatible with some substitution σ. This is done in order to illustrate when the simultaneous quantifier elimination rule is applicable on a sequent with substitution $s;\sigma$.

Definition 12. *Let σ be a substitution and E an environment. Then E is compatible with σ iff*

$$\forall \ \boxed{\begin{array}{|c|}\hline X \\ \hline c_1,\ldots,c_n \\ \hline \end{array}} \in E \,.\, \forall c_i \,.\ \ \textit{holds } c_i \notin \sigma(X)$$

We now define how an environment is constructed during analytic proof-search with the calculus $\mathcal{K}_{aux}$.

Definition 13 [Environment Construction by $\mathbf{QE}_{aux}$ on Formulas]
The function QE_{aux} has as parameters a quantifier list ql, a formula φ and a set of variables Z. The stepwise modification of the environment by $QE_{aux}(ql, \varphi, Z)$ is defined inductively.

- *The initial environment contains for each variable $X \in Z$ an empty restriction box* $\boxed{\begin{array}{|c|}\hline X \\ \hline \\ \hline \end{array}}$.
- *Case $QE_{aux}(\epsilon, \varphi, Z)$: The construction terminates with the current environment.*
- *Case $QE_{aux}(\exists x.ql, \varphi, Z)$: If E is the current environment then the new environment is*

$$E \cup \{\boxed{\begin{array}{|c|}\hline X \\ \hline \\ \hline \end{array}}\}$$

  where X is the new meta-variable introduced for x.

- *Case $QE_{aux}(\forall x.ql, \varphi, Z)$: If E is the current environment then the new environment is*

$$\{\boxed{\begin{array}{|c|}\hline X \\ \hline \ldots,c \\ \hline \end{array}} \mid \boxed{\begin{array}{|c|}\hline X \\ \hline \ldots \\ \hline \end{array}} \in E\}$$

  where c is the Eigenvariable introduced for x.

Definition 14 [Environment Construction by $\mathbf{QE}_{aux}$ on Sequents]
Let $s = \longrightarrow \Gamma, ql_1.\varphi_1, \ldots, ql_n.\varphi_n, \sigma$ be a sequent with substitution with free variables in Z. We define the environment for the application of QE_{aux} on s by the union of the environments for $QE_{aux}(ql_i, \varphi_i, Z)$ such that restriction boxes for variables in Z are combined. If $\boxed{\begin{array}{|c|}\hline X \\ \hline c_1,\ldots,c_n \\ \hline \end{array}}$ *occurs in the environment for $QE_{aux}(ql_i, \varphi_i, Z)$ and* $\boxed{\begin{array}{|c|}\hline X \\ \hline c_{n+1},\ldots,c_{n+m} \\ \hline \end{array}}$ *in the environment for $QE_{aux}(ql_j, \varphi_j, Z)$ $(i \neq j)$, then the union contains*

$$\boxed{\begin{array}{|c|}\hline X \\ \hline c_1,\ldots,c_n,c_{n+1},\ldots,c_{n+m} \\ \hline \end{array}}.$$

We now define inductively the function *Env* which constructs an environment for a derivation in $\mathcal{K}_{aux}$.

Definition 15 [Analytical Environment Construction]
The calculus $\mathcal{K}_{\text{aux}}$ has derivations as defined in section 2, i.e., a derivation is a sequence of sequents with substitutions. Analytic proof-search for a sequent s with $\mathcal{K}_{\text{aux}}$ starts with the initial $\mathcal{K}_{\text{aux}}$-derivation

$$\mathcal{D}_0 = \langle s; \sigma \rangle$$

where σ is a guessed substitution. The environment $Env(\mathcal{D}_0)$ for this initial derivation contains only restriction boxes with an empty restriction list for the meta-variables occurring in $\sigma(s)$. I.e., if X is a meta-variable in $\sigma(s)$, then the environment $Env(\mathcal{D}_0)$ contains

$$\boxed{\begin{array}{|c|}\hline X \\ \hline \\ \hline \end{array}}\ .$$

Let E be the environment for a derivation $\mathcal{D}$. Let $\mathcal{D}'$ be a derivation of s which results from $\mathcal{D}$ by the application of a calculus rule. We define the environment for $\mathcal{D}'$.

- *Application of the rules $\wedge$-, $\vee$- and* ax- *results in a derivation $\mathcal{D}'$ with the same environment. Hence $Env(\mathcal{D}') := Env(\mathcal{D})$.*
- *Application of the* Subst*-rule results in a derivation $\mathcal{D}'$ with the same environment. Hence $Env(\mathcal{D}') := Env(\mathcal{D})$.*
- *Rule* SQEl_{aux} *on $s; \sigma$: Let E be the environment generated from the sequent s by QE_{aux}: If E is compatible with σ then the new environment $Env(\mathcal{D}')$ is defined by*

$$\begin{aligned} \textit{Let } E' := \quad & \{\boxed{\begin{array}{|c|}\hline Y \\ \hline \ldots \\ \hline \end{array}} \mid \boxed{\begin{array}{|c|}\hline Y \\ \hline \ldots \\ \hline \end{array}} \in Env(\mathcal{D}),\ \ Y \notin \sigma(X) \text{ for all } \boxed{\begin{array}{|c|}\hline X \\ \hline c_1,\ldots,c_n \\ \hline \end{array}} \in E\} \\ & \cup \{\boxed{\begin{array}{|c|}\hline X \\ \hline \ldots \\ \hline \end{array}} \in E \mid \text{for all } \boxed{\begin{array}{|c|}\hline Y \\ \hline \ldots \\ \hline \end{array}} \in Env(\mathcal{D}), X \neq Y\} \\ & \cup \{\boxed{\begin{array}{|c|}\hline Y \\ \hline \ldots, c_1^{X_1}, \ldots, c_n^{X_1}, \ldots, c_1^{X_m}, \ldots, c_n^{X_m} \\ \hline \end{array}} \mid \boxed{\begin{array}{|c|}\hline Y \\ \hline \ldots \\ \hline \end{array}} \in Env(\mathcal{D}),\ \ Y \in \sigma(X_i) \\ & \qquad\qquad \text{for all } \boxed{\begin{array}{|c|}\hline X_i \\ \hline c_1^{X_i},\ldots,c_n^{X_i} \\ \hline \end{array}} \in E\} \end{aligned}$$

Then $Env(\mathcal{D}')$ results from E' by the following manipulation of E':

For every $\boxed{\begin{array}{|c|}\hline X \\ \hline c_1^X, \ldots, c_k^X \\ \hline \end{array}} \in E'$ *and every c_i^X from the restriction box let $\mathbf{Y}_{\mathbf{c_i^X}}$ be the variables, such that $c_i^X \in \sigma(Y^j_{c_i^X})$, i.e. all variables for which c_i^X is substituted. For every $Y^j_{c_i^X}$ in E' let $Res(Y^j_{c_i^X})$ denote the terms in the restriction box of $Y^j_{c_i^X}$ and let $Res(\mathbf{Y}_{\mathbf{c_i^X}}) := \cup_j Res(Y^j_{c_i^X})$. $Env(\mathcal{D}')$ is build from E' by replacing every* $\boxed{\begin{array}{|c|}\hline X \\ \hline c_1^X, \ldots, c_k^X \\ \hline \end{array}}$ *by*

$$\boxed{\begin{array}{|c|}\hline X \\ \hline c_1^X, \ldots, c_k^X, Res(\mathbf{Y}_{\mathbf{c_1^X}}), \ldots, Res(\mathbf{Y}_{\mathbf{c_k^X}}) \\ \hline \end{array}}$$

Note that E' and $Env(\mathcal{D}')$ are identical, if the variables in the sequent s are not affected by σ.

The following definition is used in order to state in a lemma, that for any $\mathcal{K}_{aux}$-derivation an ordering can be constructed.

Definition 16. *Let E be an environment generated for a $\mathcal{K}_{\text{aux}}$-derivation. The ordering $<_E$ is defined by:*

$$X <_E c_i \quad \textit{iff} \quad \frac{X}{c_1, \ldots, c_n} \in E$$

The following lemma shows that an ordering $<_E$ which can be constructed for a $\mathcal{K}_{aux}$-derivation is irreflexive.

Lemma 17. *Let E be an environment constructed for a $\mathcal{K}_{\text{aux}}$-derivation. Then $<_E$ is irreflexive.*

Lemma 18. *Let E be an environment generated by QE_{aux} for some sequent with substitution and let E' be an environment of a $\mathcal{K}_{\text{aux}}$ proof. It holds:*

1. *The ordering $<_E$ generated from E is equivalent to the ordering $\ll$ from section 4.1 up to the deletion of relationships between two meta-variables.*
2. *The ordering $<_{E'}$ generated from E' a specialization of any ordering $\lhd$ (see section 4.1) generated during application of the rule* SQEl_{aux}.

Proof. The relationship between the ordering $<_E$ and the ordering $\ll$ from section 4.1 is trivial: $<_E$ is a subset of $\ll$, which can be obtained from $\ll$ by removing any relationship between two meta-variables.

The integration of an environment from QE_{aux} with an environment of a proof before SQEl_{aux}*-rule application, especially the part corresponding to the integration of the substitution corresponds to the construction of $\lhd$ from $\ll$ and $\sqsubset$.*

We use the ordering $<_E$ constructed from a (proof) environment in order to define a new side-condition of the $SQEl_{aux}$ rule in $\mathcal{K}_{aux}$. If a $\mathcal{K}_{aux}$ proof is constructed without checking the side-condition of $SQEl_{aux}$ then an environment can only be constructed if the side condition is fulfilled at any application of $SQEl_{aux}$.

Lemma 19. *Let $\mathcal{P}_{aux}$ be a proof constructed from the $\mathcal{K}_{\text{aux}}$-rules without checking the side condition of the* SQEl_{aux} *rule. If an environment can be constructed for $\mathcal{P}_{aux}$ then the side condition of any application of the* SQEl_{aux} *rule is fulfilled and, thus $\mathcal{P}_{aux}$ is a $\mathcal{K}_{\text{aux}}$-proof.*

Proof. Let $\mathcal{P}_{\text{aux}}$ be a proof constructed by the $\mathcal{K}_{\text{aux}}$-rules without checking the side-condition of the SQEl_{aux} *rule. Let us assume that an environment can be constructed for $\mathcal{P}_{\text{aux}}$ according to definition 15. From lemma 17 we conclude that the ordering $<_E$ specified by the environment is irreflexive. From lemma 18 we infer that the ordering is irreflexive. The irreflexivity of $<_E$ implies that the ordering $\lhd$ is also irreflexive as required in the side conditions of the rule* SQEl_{aux}*. Thus, the side-condition of the* SQEl_{aux} *rule is fulfilled.* □

According to lemma 19 we have two possible side-conditions of the $SQEl_{aux}$ rule.

4.4 Soundness of $\mathcal{K}_{aux}$.

Theorem 20 [Soundness]
If there exists a $\mathcal{K}_{\text{aux}}$-proof $\mathcal{P}$ of a sequent with substitution s then s is valid.

Proof. *The proof is by induction on the structure of $\mathcal{P}$. The base case where $\mathcal{P}$ consists only of an application of* ax *is trivial. In the induction step a case distinction depending on the last rule application in $\mathcal{P}$ is made. We concentrate on the interesting cases where $\wedge$,* Subst, *or* SQEl_{aux} *is applied. In each case we assume that all premises of the rule are valid and infer the validity of the conclusion.*

- *Let $\wedge$ be the last rule applied in $\mathcal{P}$. We assume that for every algebra A and every assignment α holds $A(\alpha)(\longrightarrow \Gamma, \varphi_1; \sigma) = true = A(\alpha)(\longrightarrow \Gamma, \varphi_2; \sigma)$. Let A be an arbitrary algebra and α be an arbitrary assignment. If there is a $F \in \Gamma$ such that $A(\alpha)(\sigma(F)) = true$, then $A(\alpha)(\longrightarrow \Gamma, \varphi_1 \wedge \varphi_2; \sigma) = true$ holds trivially. Otherwise, $A(\alpha)(\sigma(\varphi_1)) = true = A(\alpha)(\sigma(\varphi_2))$ must hold which implies $A(\alpha)(\longrightarrow \Gamma, \varphi_1 \wedge \varphi_2; \sigma) = true$.*

- *Let* Subst *be the last rule applied in $\mathcal{P}$. We assume that for every algebra A and every assignment α holds $A(\alpha)(\longrightarrow \sigma'(\Gamma); \sigma) = true$. According to the side condition of the rule holds $\sigma \circ \sigma' = \sigma$. Thus, $\longrightarrow \sigma(\Gamma) = \longrightarrow \sigma(\sigma'(\Gamma))$ and the validity of the conclusion follows.*

- *Let SQEl_{aux} be the last rule applied in $\mathcal{P}$. The proof is done by induction over the number m of quantifiers eliminated, i.e. the sum of the lengths of the ql_i. The base case where $m = 0$ is trivial, since the premise and the conclusion are the same sequent. We first prove the case $m = 1$ with $s = \longrightarrow \Gamma, Qx.\varphi; \sigma$.*

  - *If $Q = \exists$, we assume that for every algebra A and every assignment α holds $A(\alpha)(\longrightarrow \Gamma, \varphi[X/x]; \sigma) = true$. Let A be an arbitrary algebra and α be an arbitrary assignment. The interesting case is where $A(\alpha)(\sigma(F)) = false$ for all $F \in \Gamma$ and $A(\alpha)(\sigma(\varphi[X/x])) = true$. Let σ' be the restriction of σ to free$(\Gamma \cup \{\exists x.\varphi\})$. Then*

$$\begin{aligned} true &= A(\alpha)(\sigma(\varphi[X/x])) = A(\alpha)(\sigma'(\varphi[X/x])) = A(\alpha)((\sigma'(\varphi))[\sigma'(X)/x]) \\ &= A(\alpha[A(\alpha)(\sigma'(X))/x])(\sigma'(\varphi)) \text{ by substitution theorem} \\ &= A(\alpha)(\exists x.\sigma'(\varphi)) \text{ by definition of the semantics of } \exists \\ &= A(\alpha)(\sigma'(\exists x.\varphi)) = A(\alpha)(\sigma(\exists x.\varphi)) = A(\alpha)(\longrightarrow \Gamma, \exists x.\varphi; \sigma) \end{aligned}$$

  - *If $Q = \forall$, we assume that for every algebra A and every assignment α holds $A(\alpha)(\longrightarrow \Gamma, \varphi[c/x]; \sigma) = true$. Let A be an arbitrary algebra and α be an arbitrary assignment. The interesting case is where $A(\alpha)(\sigma(F)) = false$ for all $F \in \Gamma$ and $A(\alpha)(\sigma(\varphi[c/x])) = true$. We consider all* variants *A_a of A, i.e. all algebras which differ from A only in the interpretation of c such that $A_a(c) = a$. The side-condition of* SQEl_{aux} *and the definition of QE_{aux} ensure that $X \ll c$ holds for every $X \in$ free$(\longrightarrow \Gamma, \forall x.\varphi; \sigma)$. Because $\lhd$ is required to be irreflexive for any $F \in \Gamma$, c does not occur in $\sigma(F)$ and thus, $A_a(\alpha)(\sigma(F)) = A(\alpha)(\sigma(F)) = false$. Let σ' be the restriction of σ to free$(\Gamma \cup \{\forall x.\varphi)\}$. Then for all A_a holds*

$$\begin{aligned} true &= A_a(\alpha)(\sigma'(\varphi[c/x])) = A_a(\alpha)((\sigma'(\varphi))[c/x]) \text{ since } \sigma' \text{ is admissible} \\ &= A_a(\alpha[A_a(c)/x])(\sigma'(\varphi)) \text{ by substitution theorem} \\ &= A_a(\alpha[a/x])(\sigma'(\varphi)) \ . \end{aligned}$$

For every $a \in S_A$ there is an A_a, thus $A(\alpha)(\longrightarrow \Gamma, \forall x.\varphi; \sigma) = true$.

In the induction step we assume the soundness of $\mathrm{SQEl}_{\mathrm{aux}}$ *for the elimination of less than m quantifiers ($m > 1$) and show the soundness for m quantifiers. The irreflexivity of $\lhd$ ensures that there is a maximal element $o \in O$ with regard to $\lhd$. According to the definition of $\sqsubset$ o is not instantiated for any variable in O. Let o be introduced by the elimination of $Q_j x.\psi'_j$. We split the application of* $\mathrm{SQEl}_{\mathrm{aux}}$ *as follows into two applications of the rule where each of the applications reduces less than m quantifiers. Due to the choice of $Q_j x.\psi_j$ the side conditions for both rule applications are fulfilled.*

$$\dfrac{\dfrac{\longrightarrow \Gamma, \psi_1, \ldots, \psi_j, \ldots, \psi_n; \sigma}{\longrightarrow \Gamma, \psi_1, \ldots, Qx.\psi'_j, \ldots, \psi_n; \sigma}\ \mathrm{SQEl}_{\mathrm{aux}} \quad \text{1 quantifier elimination}}{\longrightarrow \Gamma,\ ql_1.\varphi_1, \ldots,\ ql_j.Qx.\varphi_j, \ldots,\ ql_n.\varphi_n; \sigma}\ \mathrm{SQEl}_{\mathrm{aux}} \quad (m-1) \text{ quantifier eliminations}$$

Remark 21. In the induction step of the above proof we show that it is always possible to focus on a single formula in a sequent. In the case where the rule $SQEl_{aux}$ is the last rule applied this is non-trivial because multiple formulas are reduced in a single rule application. Free variables in a sequent cause dependencies between formulas. Only the side condition of the rule $SQEl_{aux}$ allow us to single out a specific formula according to the reduction ordering and ensure in the case $\forall$ that this formula is valid in all variants of a specific algebra.

4.5 Justification of $\mathcal{K}$

There are three differences between $\mathcal{K}_{aux}$ and $\mathcal{K}$. In $\mathcal{K}_{aux}$ a substitution is explicitly stated in sequents, *Eigenvariables* have no arguments (i.e. are no skolem-terms), and the *Subst*-rule is a sequent rule while the *Inst*-rule of $\mathcal{K}$ is a rewrite rule on proof trees. Proof search in $\mathcal{K}_{aux}$ would require that an appropriate substitution is guessed before any rule may be applied. None of the rules in $\mathcal{K}_{aux}$ is capable to modify this substitution. This appears to be impractical, however, $\mathcal{K}_{aux}$ is only an auxiliary calculus. We show by the following theorem that the skolemization based technique applied in $\mathcal{K}$ is a realization of the constraints imposed by $\mathcal{K}_{aux}$.

Theorem 22 [Soundness]
If there exists a $\mathcal{K}$-proof for a sequent s then s is valid.

Proof. Let $\mathcal{P}$ be a $\mathcal{K}$-proof of some sequent s such that all meta-variables which occur in s are not instantiated in $\mathcal{P}$. Let σ be the substitution of the last level of $\mathcal{P}$. From $\mathcal{P}$ and σ we construct a $\mathcal{K}_{\mathrm{aux}}$*-proof $\widetilde{\mathcal{P}}$ and a substitution $\widetilde{\sigma}$ such that $\widetilde{\mathcal{P}}$ is a proof of $\widetilde{s}; \widetilde{\sigma}$. Theorem 20 ensures that $\widetilde{s}; \widetilde{\sigma}$ is valid. If no skolem-terms occur in s then $\widetilde{s} = s$ holds. According to the definition of validity for sequents with substitutions we can conclude that s is valid. Since s and $\mathcal{P}$ were chosen arbitrarily we can conclude the soundness of $\mathcal{K}$.*

We now fill the gaps in this proof sketch.

In order to simplify the proof, we (1) require that in $\mathcal{P}$ meta-variables are only instantiated immediately after their introduction and (2) that variables free in the initial sequent cannot be instantiated after a calculus rule different from Inst *has been applied. This restriction is complete, i.e. any $\mathcal{K}$-proof can be transformed into a proof which has this property. The only rule which introduces new meta-variables is* SQEl*. Applications of the* Inst *rule which do not affect variables introduced by* SQEl *can be permuted such that* Inst *is applied before* SQEl *where the order of rule applications is seen from the point of view of analytic proof search.* Inst *is also permutable with the rules $\wedge$ and $\vee$.*

Please keep in mind that $\mathcal{P}$ is a $\mathcal{K}$-proof which consists of a sequence of levels where each level contains a sequence of sequents and a substitution while $\widetilde{\mathcal{P}}$ is a sequence of sequents with substitutions.

For each skolem-function f which occurs in $\mathcal{P}$ we define a new constant symbol $\widetilde{f}$.

We define $\widetilde{\ }$ as a function on terms.

$$\begin{array}{lcll} \widetilde{X} & = & X & \text{, where } X \text{ is a meta-variable} \\ \widetilde{x} & = & x & \text{, where } x \text{ is a variable} \\ \widetilde{f(t_1,\ldots,t_n)} & = & \widetilde{f} & \text{, where } f \text{ is a skolem-function} \\ \widetilde{g(t_1,\ldots,t_n)} & = & g(\widetilde{t_1},\ldots,\widetilde{t_n}) & \text{, where } g \text{ is not a skolem-function} \end{array}$$

$\widetilde{\ }$ is extended homomorphically to formulas. For a substitution σ we define $\widetilde{\sigma} = \widetilde{\ } \circ \sigma$. The application of $\widetilde{\ }$ on an environment E results in the application of $\widetilde{\ }$ to all elements of the restriction boxes in E.

We construct a $\mathcal{K}_{\text{aux}}$-proof $\widetilde{\mathcal{P}}$ for $\widetilde{s_n^m}, \widetilde{\sigma_m}$ by induction from a $\mathcal{K}$-proof

$$\begin{array}{rl} \mathcal{P} = \langle & (\langle s^1_{k_1},\ldots,s^1_n\rangle,\ \sigma_1), \\ & (\langle s^2_{k_2},\ldots,s^2_{k_1},\ldots,s^2_n\rangle,\ \sigma_2), \\ & \ldots \\ & (\langle s^m_1,\ldots,s^m_{k_2},\ldots,s^m_{k_1},\ldots,s^m_n\rangle,\ \sigma_m)\rangle \end{array}$$

of s_n^m. The construction ensures that for each i the sequent $\widetilde{s_i^m};\widetilde{\sigma}$ occurs in $\widetilde{\mathcal{P}}$. Additionally, if E is the environment corresponding to $\mathcal{P}$ then $\widetilde{E}$ is the environment corresponding to $\widetilde{\mathcal{P}}$. The induction assumption is that for any $\mathcal{K}$-proof $\mathcal{P}'$ with length less than n, $\widetilde{\mathcal{P}'}$ is the corresponding $\mathcal{K}_{\text{aux}}$-proof and that if E' is the environment constructed from $\widetilde{\mathcal{P}'}$ then $E' = \widetilde{E}$ holds.

We make a case distinction depending on how s_n^m has been derived. According to lemma 4 it suffices to consider the cases where s_n^m can be derived on level m by an application of ax*, $\wedge$, or $\vee$ or s_n^m can be derived on some level $j \leq m$ by an application of* SQEl*. Let $\mathcal{P}'$ be the $\mathcal{K}$-proof where s_n^j has been removed from each level j in $\mathcal{P}$.*

- *If s_n^m can be derived on level m in $\mathcal{P}$ by an application of* ax*. Then, $\widetilde{\mathcal{P}'}, (\widetilde{s_n^m};\widetilde{\sigma_m})$ is a $\mathcal{K}_{\text{aux}}$-proof of $\widetilde{s_n^m};\widetilde{\sigma_m}$.*

  E and E' are the initial environments. Because of $\widetilde{\sigma_m}\widetilde{s_n^m} = \widetilde{s_n^m}$ holds $\widetilde{E} = E'$.

- *If s_n^m can be derived on level m in $\mathcal{P}$ by an application of $\wedge$. Let $s^m_{n_1}$ and $s^m_{n_2}$ be the premises of that rule application. According to the induction assumption $\widetilde{s^m_{n_1}};\widetilde{\sigma_m}$ and $\widetilde{s^m_{n_2}};\widetilde{\sigma_m}$ occur in $\widetilde{\mathcal{P}'}$. Thus, $\widetilde{\mathcal{P}'}, (\widetilde{s_n^m};\widetilde{\sigma_m})$ is a $\mathcal{K}_{\text{aux}}$-proof of $\widetilde{s_n^m};\widetilde{\sigma_m}$.*

  The construction of the environment is not affected by this rule in both calculi, thus, the induction assumption can be applied directly.

- *If s_n^m can be derived on level m in $\mathcal{P}$ by an application of $\vee$. Let $s_{n_1}^m$ be the premise of that rule application. According to the induction assumption $\widetilde{s_{n_1}^m};\widetilde{\sigma_m}$ occurs in $\widetilde{\mathcal{P}'}$. Thus, $\widetilde{\mathcal{P}'}, (\widetilde{s_n^m};\widetilde{\sigma_m})$ is a $\mathcal{K}_{\text{aux}}$-proof of $\widetilde{s_n^m};\widetilde{\sigma_m}$.*

  The construction of the environment is not affected by this rule in both calculi, thus, the induction assumption can be applied directly.

- *If s_n^m can be derived on level j $(j \leq m)$ in $\mathcal{P}$ by an application of* SQEl. *Let $s_{n_1}^j$ be the premise of that rule application. According to the induction assumption $\widetilde{s_{n_1}^m};\widetilde{\sigma_m}$ occurs in $\widetilde{\mathcal{P}'}$. The reduction of this sequent by* Subst *with parameter $\widetilde{\sigma_m}$ results in $\widetilde{s_n^j};\widetilde{\sigma_m}$. The subsequent application of* SQEl_{aux} *results in $\widetilde{s_n^m};\widetilde{\sigma_m}$. Thus, $\widetilde{\mathcal{P}'}$, $(\widetilde{s_n^j};\widetilde{\sigma_m})$, $(\widetilde{s_n^m};\widetilde{\sigma_m})$ is a $\mathcal{K}_{\text{aux}}$-proof of $\widetilde{s_n^m};\widetilde{\sigma_m}$.*

  If the variables introduced by SQEl_{aux} *are not affected by $\widetilde{\sigma_m}$ then according to definition 8 and definition 15 $\widetilde{E} = E'$ holds.*

  The case where some of the variables introduced are affected by $\widetilde{\sigma_m}$ is shown by induction on the number of variables which are affected. Let us assume, that $\widetilde{E} = E'$ holds when at most k variables are affected. We show that it also holds when $k+1$ variables are affected. This follows from definition 8 and 15 because an application of the Inst*-rule in $\mathcal{K}$ with parameters X and t has the same effect on the environment construction as the substitution $\widetilde{\sigma}$ with $\widetilde{\sigma}(X) = \widetilde{t}$ together with the restriction box of X in the environment construction for an $\mathcal{K}_{\text{aux}}$-proof.*

  Thus, $\widetilde{E} = E'$ holds.

  According to lemma 19 the construction of an environment for $\widetilde{P}$ guarantees that the side condition of SQEl_{aux} *is fulfilled. Thus, our application of* SQEl *in the construction of $\widetilde{P'}$ was admissible.*

5 Conclusion

We defined the sequent calculus $\mathcal{K}$ which incorporates a rule for simultaneous quantifier elimination [1]. This rule provides advantages for proof search: As conventional skolemization already postpones the choice of a term for an existentially quantified variable, the simultaneous quantifier elimination rule additionally overcomes the unnecessary decision for the ordering in which quantifiers are eliminated. This makes this rule especially attractive for automation. The simultaneous quantifier elimination rule presented in this report has been implemented in the VSE II system [8] for formal software development which is currently under development at the DFKI as a successor of the VSE system [7].

The calculus $\mathcal{K}$ is proven to be sound and complete. The more difficult proof of the soundness theorem has been carried out by semantical arguments. The proof clarifies the interdependencies between the formulas of a sequent. In order to visualize the relation between skolemization and the Eigenvariable condition the concept of restriction boxes has been introduced.

Other systems handle quantifiers with different degrees of sophistication. Lazy handling of instantiations has been used in classical as well as in non-classical logics and in first-order as well as in higher-order formalisms. Except in sequent calculus and natural deduction calculi,

Skolemization has been studied in the context of resolution, connection method and tableau calculi as well.

For instance in PVS [5] Gentzen-like quantifier elimination rules are used where instantiations must be guessed. Ketonen and Weyhrauch [9] present an approach where sequents are annotated by a substitution, like in the semantical part of this article, and used a technique similar to our quantifier list ordering. However, they have only classical quantifier rules with the corresponding non-determinism in proof search. In the Isabelle system [12] a technique dual to classical skolemization is used. In remark 5 we have pointed out that this technique causes close interdependencies between formulas in a sequent which make it quite difficult to develop an optimized handling of quantifiers equivalent to our simultaneous quantifier elimination rule.

References

[1] S. Autexier, H. Mantel and W. Stephan. *Simultaneous Quantifier Elimination*, In Proceedings of the 22nd German Conference on Artificial Intelligence KI-98, Springer, LNAI series, 1998.

[2] A. Avron. Simple Consequence Relations, In *Information and Computation 92*, p. 105–139, 1991.

[3] W. Bibel. *Automated Theorem Proving.* Vieweg Verlag, 2nd edition, 1987.

[4] K. A. Bowen. Programming with full first-order logic, In Hayes, Michie, Pao, Eds., *Machine Intelligence 10*, 1982.

[5] J. Crow, S. Owre, J. Rushby, N. Shankar and M. Srivas. *A Tutorial Introduction to PVS*, Presented at WIFT'95, 1995.

[6] G. Gentzen. *Untersuchungen über das logische Schließen*, Mathematische Zeitschrift. 39:179-210 and 405-431, 1935.

[7] D. Hutter, B. Langenstein, C. Sengler, J. Siekmann, W. Stephan and A. Wolpers. Verification Support Environment (VSE), *High Integrity Systems*, p. 523-530, 1996.

[8] D. Hutter, H. Mantel, G. Rock, W. Stephan, A. Wolpers, M. Balser, W. Reif, G. Schellhorn, K. Stenzel. *VSE: Controlling the Complexity in Formal Software Developments.* to appear in the Proceedings of the *International Workshop on Applied Formal Methods*, 1998.

[9] J. Ketonen and R. Weyhrauch. A Decidable Fragment of Predicate Calculus, In *Theoretical Computer Science, Volume 32-3*, p. 297-307, 1984.

[10] P. Lincoln and N. Shankar. Proof Search in First-Order Linear Logic and other Cut-Free Sequent Calculi, In *Proceedings of 9th LICS*, IEEE Computer Society Press, Los Alamitos, Ca., USA, p. 282-291, 1994.

[11] J. Loeckx, H.-D. Ehrich and M. Wolf. *Specification of Abstract Data Types*, Wiley-Teubner, 1996.

[12] L.C. Paulson. *Isabelle, A Generic Theorem Prover*, Springer Verlag, 1994.

[13] N. Shankar. Proof Search in the Intuitionistic Sequent Calculus, In D . Kapur, Ed., *Proceedings of CADE-11*, Springer, LNAI series, LNAI 607, p. 522–536, 1992.

[14] T. Skolem. Logisch-kombinatorische Untersuchungen über die Erfüllbarkeit oder Beweisbarkeit mathematischer Sätze, In *Skrifter utgit av Videnskapselskapet i Kristiania*, p. 4-36, 1920.

[15] A. Tarski. Introduction to Logic and to the Methodology of Deductive Sciences, number 24 Oxford Logic Guides, 4th edition, Oxford University Press, 1994.

[16] L. Wallen. *Automated Deduction in Non-Classical Logic.* MIT Press, 1990.