LOGIC JOURNAL of the IGPL

Volume 12 Number 1 January 2004

Interest Group in Pure and Applied Logics

OXFORD UNIVERSITY PRESS

ISSN 1367-0751
www.oup.co.uk/igpl
Printed in Great Britain by
Butler & Tanner

Subscriptions

A subscription to *Logic Journal of the IGPL* comprises 6 issues. Prices include postage by surface mail; or for subscribers in the USA and Canada, by airfreight; or in India, Japan, Australia and New Zealand, by Air Speeded Post. Airmail rates are available on request.

Annual Subscription Rate (Volume 12, 6 issues, 2004)
Institutional: Print edition and site-wide online access: £412/US$666; Print edition only: £391/US$633; Site-wide online access only: £370/US$600. *Personal*: Print edition only: £206/US$333. Please note: £ Sterling rates apply in Europe, US$ elsewhere. There may be other subscription rates available. For a complete listing, please visit www.oup.co.uk/igpl/subinfo.

Full prepayment, in the correct currency, is required for all orders. Orders are regarded as firm and payments are not refundable. Subscriptions are accepted and entered on a complete volume basis. Claims cannot be considered more than FOUR months after publication or date of order, whichever is later. All subscriptions in Canada are subject to GST. Subscriptions in the EU may be subject to European VAT. If registered, please supply details to avoid unnecessary charges. For subscriptions that include online versions, a proportion of the subscription price may be subject to UK VAT. Personal rate subscriptions are only available if payment is made by personal cheque or credit card and delivery is to a private address.

The current year and two previous years' issues are available from Oxford University Press. Previous volumes can be obtained from the Periodicals Service Company, 11 Main Street, Germantown, NY 12526, USA. Email: psc@periodicals.com. Tel: +1 (518) 537 4700. Fax: +1 (518) 537 5899.

For further information, please contact: Journals Customer Service Department, Oxford University Press, Great Clarendon Street, Oxford OX2 6DP, UK. Email: jnls.cust.serv@oupjour nals.org. Tel (and answerphone outside normal working hours): +44 (0)1865 353907. Fax: + 44 (0)1865 353485.

Methods of payment. Payment may be made by cheque (payable to Oxford University Press) in £ Sterling drawn on a UK bank or in US$ drawn on a US bank, bank transfer (£ Sterling – Barclays Bank Plc, Oxford City Office, PO Box 333, Oxford OX1 3HS, UK. Bank sort code: 20-65-18, Account No. 70299332, IBAN number GB54BARC20651870299332, Swift code BARCGB22), or by credit card (Mastercard, Visa, Switch or American Express).

Logic Journal of the IGPL (ISSN 1367-0751) is published bimonthly by Oxford University Press, Oxford, UK. Annual subscription price is £412/US$666. Logic Journal of the IGPL is distributed by Mercury International, 365 Blair Road, Avenel, NJ 07001, USA. Periodicals postage paid at Rahway, NJ and at additional entry points.

US Postmaster: send address changes to *Logic Journal of the IGPL* c/o Mercury International, 365 Blair Road, Avenel, NJ 07001, USA.

Permissions

For information on how to request permissions to reproduce articles/information from this journal, please visit www.oupjournals.org/permissions.

Advertising

Inquiries about advertising should be sent to Helen Pearson, Oxford Journals Advertising, PO Box 347, Abingdon OX14 1GJ, UK. Email: helen@oxfordads.com. Tel/Fax: +44 (0)1235 201904.

Disclaimer

Statements of fact and opinion in the articles in the *Logic Journal of the IGPL* are those of the respective authors and contributors and not of the *Logic Journal of the IGPL* or Oxford University Press. Neither Oxford University Press nor the *Logic Journal of the IGPL* make any representation, express or implied, in respect of the accuracy of the material in this journal and cannot accept any legal responsibility or liability for any errors or omissions that may be made. The reader should make his/her own evaluation as to the appropriateness or otherwise of any experimental technique described.

Logic Journal of the IGPL

Volume 12, Number 1, January 2004

Contents

Original Articles

Please visit the journal's World Wide Web site at
http://www.oup.co.uk/igpl

Logic Journal of the Interest Group in Pure and Applied Logics

Scope of the Journal

The *Journal* is the official publication of the International Interest Group in Pure and Applied Logics (IGPL), which is sponsored by The European Foundation for Logic, Language and Information (FoLLI), and currently has a membership of over a thousand researchers in various aspects of logic (symbolic, computational, mathematical, philosophical, etc.) from all over the world.

The *Journal* is published in hardcopy and in electronic form six times per year. Publication is fully electronic: submission, refereeing, revising, typesetting, checking proofs, and publishing, all is done via electronic mailing and electronic publishing.

Papers are invited in all areas of pure and applied logic, including: pure logical systems, proof theory, model theory, recursion theory, type theory, nonclassical logics, nonmonotonic logic, numerical and uncertainty reasoning, logic and AI, foundations of logic programming, logic and computation, logic and language, and logic engineering.

The *Journal* is an attempt to solve a problem in the logic (in particular, IGPL) community:

- ○ Long delays and large backlogs in publication of papers in current journals.
- ○ Very tight time and page number limits on submission.

Papers in the final form should be in LaTeX. The review process is quick, and is made mainly by other IGPL members.

Submissions

Submissions are made by sending a submission letter to the e-mail address: jigpl@dcs.kcl.ac.uk, giving the title and the abstract of the paper, and informing: of how to obtain the file electronically or, by sending 5 (five) hardcopies of the paper to the Editor-in-Chief.

As from Volume 10 (2002) the Journal is indexed in:
- Science Citation Index Expanded (SCIE)
- Research Alert
- CompuMath Citation Index (CMCI)

URL: www.oup.co.uk/igpl

Descente Infinie + Deduction

CLAUS-PETER WIRTH, *Dept. of Computer Science, Universität des Saarlandes,*
D–66123 Saarbrücken, Germany.
E-mail: cp@ags.uni-sb.de

Abstract

Inductive theorem proving in the form of *descente infinie* was known to the ancient Greeks and is the standard induction method of a working mathematician since it was reinvented in the middle of the 17th century. We present an integration of *descente infinie* into state-of-the-art free-variable sequent and tableau calculi. It is well-suited for an efficient interplay of human interaction and automation and combines raising, explicit representation of variable dependency, the liberalized δ-rule, preservation of solutions, and unrestricted applicability of lemmas and induction hypotheses. The semantical requirements are satisfied for a variety of two-valued logics, such as clausal logic, classical first-order logic, and higher-order modal logic.

Keywords: Mathematical Induction, Sequent and Tableau Calculi, Logical Foundations, Formalized Mathematics, Human-Oriented Interactive Automated Theorem Proving

Contents

1 *Descente Infinie*: An Introduction

1.1 What it Is

1.1.1 An Example

Inductive arguments are omnipresent in mathematics, theoretical computer science, or physics, and every freshman in these subjects is familiar with arguments of the following kind. Suppose we have the axioms:

$$
\begin{array}{lll}
(\mathsf{nat1}) & \forall x. & \big(\ x{=}\mathsf{0} \vee \exists y.\ x{=}\mathsf{s}(y)\ \big) \\
(\mathsf{plus1}) & \forall x. & x+\mathsf{0}=x \\
(\mathsf{plus2}) & \forall x,y. & x+\mathsf{s}(y)=\mathsf{s}(x+y)
\end{array}
$$

where $(\mathsf{nat1})$ says that any natural number is zero or the successor of a natural number, while $(\mathsf{plus1})$ and $(\mathsf{plus2})$ define the function '$+$', and the signature is as follows: We only have the single type nat of natural numbers. We use zero $\mathsf{0} : \mathsf{nat}$ and successor $\mathsf{s} : \mathsf{nat} \rightarrow \mathsf{nat}$ as constructors for the type nat. Moreover, $+ : \mathsf{nat} \rightarrow \mathsf{nat} \rightarrow \mathsf{nat}$ is a defined function on natural numbers, and x, y are variables of the type nat. Given this setting, how do we prove $\forall x.\ \mathsf{0}+x=x$, i.e. that the right-neutral element $\mathsf{0}$ of $(\mathsf{plus1})$ is also neutral to the left? An informal proof may run like this:

> We have to show
> $$\mathsf{0}+x=x. \tag{1}$$
> Using $(\mathsf{nat1})$, we have the following case analysis:
>
> $\underline{x=\mathsf{0}:}$ We have to show
> $$\mathsf{0}+\mathsf{0}=\mathsf{0}, \tag{1.1}$$
> which follows from $(\mathsf{plus1})$.
>
> $\underline{x=\mathsf{s}(y):}$ We have to show
> $$\mathsf{0}+\mathsf{s}(y)=\mathsf{s}(y). \tag{1.2}$$
> Using $(\mathsf{plus2})$ we can rewrite it into
> $$\mathsf{s}(\mathsf{0}+y)=\mathsf{s}(y). \tag{1.2.1}$$
> Setting $\{x{\mapsto}y\}$ in the induction hypothesis (1), we can rewrite this into the equality axiom
> $$\mathsf{s}(y)=\mathsf{s}(y). \tag{1.2.1.1}$$
> We still have to find an induction ordering $<$ and some weight $w(x)$ for (1) such that the instance of the applied induction hypothesis is smaller than the induction conclusion we are just proving, i.e. such that $w(y) < w(x)$. By our case assumption this is nothing but
> $$w(y) < w(\mathsf{s}(y)). \tag{1.2.1.2}$$

But this is trivial: We simply set the weight function w to the identity, $w(x) := x$, and we let the induction ordering $<$ be the ordering of the natural numbers, denoted by the symbol $\prec$. Now (1.2.1.2) turns into

$$y \prec \mathsf{s}(y). \qquad (1.2.1.2.1)$$

This is valid and follows from the properties of $\prec$, which include $\forall y.\ y \prec \mathsf{s}(y)$ and the well-foundedness of $\prec$.

Now, how can this kind of argument be formalized?

First, we have to settle on some specific logical calculus for deductive reasoning and, second, the actual form of the inductive argument has to be fixed within this calculus. We defer the answer to the first problem to Section 1.2. The second problem divided the research community into the two schools of *explicit* and *implicit induction*, of which the former represents the established mainstream community, which excels in the most powerful computer-based systems today. For comprehensive surveys on explicit induction cf. Walther (1994) and Bundy (2001). Implicit induction, however, evolved from the *Knuth–Bendix Completion Procedure* and comprises the alternative approaches of proof by consistency (inductionless induction), *descente infinie*, and syntactical induction orderings. While we are not going to discuss implicit induction here (cf., however, Wirth (2004) for a survey), it seems to be necessary to disambiguate *descente infinie* from mainstream work.

1.1.2 Axiomatization

As will be discussed in more detail in the following sections, proof search in the style of *descente infinie* was already known to the ancient Greeks and is the standard method of a working mathematician since it was reinvented in the fifties of the 17^{th} century by Pierre Fermat.

At Fermat's time, natural language was still the predominant tool for expressing terms and equations in mathematical writing, and it was too early for a formal axiomatization. Although an axiomatization captures only validity but in general does not induce a method of proof search, we should nevertheless discuss it here. Let us look at natural numbers and arithmetic to state our case:

In the 20^{th} century, Dedekind's axioms for arithmetic became popular under the name of *Peano's axioms*:

$$\begin{array}{lll}
(\mathsf{nat2}) & \forall x. & \mathsf{s}(x)\neq 0 \\
(\mathsf{nat3}) & \forall x,y. & (\mathsf{s}(x)=\mathsf{s}(y) \Rightarrow x=y) \\
(\mathsf{S}) & & P(0) \wedge \forall y.\ \big(\ P(y) \Rightarrow P(\mathsf{s}(y))\ \big) \ \Rightarrow\ \forall x.\ P(x)
\end{array}$$

The *axiom* (S) is called the axiom of *structural induction* because it follows the structure of the natural numbers built-up inductively by the constructors 0 and s. There are similar versions of structural induction for all inductive data types such as lists or trees. The axiom (S) can be seen either as a first-order scheme in P, or, if prefixed with "$\forall P.$ ", as a second-order axiom.

Theoretically, every theorem of arithmetic now follows from $(\mathsf{nat2})$, $(\mathsf{nat3})$, (S) and function definitions like $(\mathsf{plus1})$, $(\mathsf{plus2})$. *Practically*, however, it is next to impossible to find all proofs in arithmetic by structural induction, because some of the required instances for P in (S) are too complicated. On the contrary, induction *over arbitrary wellfounded relations* $<$, often called *Nötherian* induction after A. Emmy Nöther (1882–1935), is essential, both for search and communication of proofs:

$$(\mathsf{N}) \qquad \mathsf{Wellf}(<) \;\wedge\; \forall v.\ \big(\ \forall u.\ (u{<}v \Rightarrow P(u)) \;\;\Rightarrow\;\; P(v)\ \big) \quad\Rightarrow\quad \forall x.\ P(x)$$

The *theorem* (N) follows directly from the definition of wellfoundedness Wellf alone.

The wellfoundedness of the the successor relation $\lambda x, y.\ (\mathsf{s}(x){=}y)$ (which implies the wellfoundedness of the ordering $\prec$ of the natural numbers by Lemma 2.1) means that any nonempty subset B of the natural numbers contains an s-minimal element:

$$(\mathsf{Wellf}(\mathsf{s})) \qquad \forall B.\ \big(\ B{\neq}\emptyset \;\Rightarrow\; \exists y{\in}B.\ \forall w{\in}B.\ \mathsf{s}(w){\neq}y\ \big)$$

Using all of Peano's axioms, this can be shown by setting P in (S) to be

$$\lambda x.\ \big(\ \exists z{\in}B.\ \mathsf{s}(z){=}x \;\Rightarrow\; \exists y{\in}B.\ \forall w{\in}B.\ \mathsf{s}(w){\neq}y\ \big).$$

1.1.3 Explicit Induction

Vice versa, Peano's axioms follow from $\mathsf{Wellf}(\mathsf{s})$ and $(\mathsf{nat1})$. Indeed, $(\mathsf{nat2})$ and $(\mathsf{nat3})$ can be shown with the lemma $\forall x.\ \exists n{\in}\mathbf{N}.\ x{=}\mathsf{s}^n(\mathsf{0})$. Moreover, (S) follows from wellfoundedness of the successor relation $\mathsf{Wellf}(\mathsf{s})$ when we instantiate $u{<}v$ in (N) with the successor relation and apply the case analysis of $(\mathsf{nat1})$ to v. Indeed,

$$\forall u.\ (\mathsf{s}(u){=}\mathsf{0} \Rightarrow P(u)) \;\Rightarrow\; P(\mathsf{0})$$

simplifies to $P(\mathsf{0})$ by $(\mathsf{nat2})$.

Such first-order instances of (N) are called *induction axioms* in the school of explicit induction. Notice that in these induction axioms, the subformula

$$\forall u.\ (u{<}v \Rightarrow P(u))$$

of (N) is replaced with a conjunction of instances of $P(u)$ with predecessors of v like in (S). The induction axioms of explicit induction must not contain the induction ordering $<$.

The school of explicit induction was formed by computer scientists who were working on the automation of theorem proving and—inspired by J. Alan Robinson's resolution method (Robinson (1965))—tried to solve problems of logical inference via reduction to machine-oriented inference systems. Instead of implementing more advanced mathematical induction techniques, they decided to reduce the (second-order) theorem (N) to first-order *induction axioms* in the following fashion:

> Guess an induction axiom for which the wellfoundedness of $<$ can be automatically derived. Apply the induction axiom backwards. The rest is purely deductive first-order reasoning. If this fails, repeat the whole process.

For instance, our introductory example is solved via the induction axiom

$$\mathsf{0}+\mathsf{0}=\mathsf{0}\ \wedge\ \forall y.\ \big(\ \mathsf{0}+y=y\ \Rightarrow\ \mathsf{0}+\mathsf{s}(y)=\mathsf{s}(y)\ \big)\ \ \Rightarrow\ \ \forall x.\ \mathsf{0}+x=x$$

Note that the reduct $\mathsf{0}+\mathsf{0}=\mathsf{0}\ \wedge\ \forall y.\ \big(\ \mathsf{0}+y=y\ \Rightarrow\ \mathsf{0}+\mathsf{s}(y)=\mathsf{s}(y)\ \big)$ is valid in all models of $(\mathsf{plus1})$ and $(\mathsf{plus2})$, and can be shown just by deduction. Contrary to this, the conjectured proposition $\forall x.\ \mathsf{0}+x=x$ is only inductively valid.

The so-called "waterfall"-method of the pioneers of this approach (Boyer & Moore (1979)) refines this process into a fascinating heuristic, and the powerful inductive theorem proving system NQTHM (Boyer & Moore (1988)) shows the success of this reduction approach. Modern explicit induction systems such as INKA (Autexier &al. (1999)) or ACL2 (Kaufmann &al. (2000)) easily outperform even a good mathematician on the typical inductive proof problems that arise in his daily work or as subtasks in software verification. However, although there is still witness for considerable improvements over the years (cf. Hutter & Bundy (1999)), these methods and systems do not seem to scale up to hard mathematical problems, and we believe that there are *principle reasons* for this shortcoming.

1.1.4 *Descente Infinie* in the Working-Mathematician Style

In everyday mathematical practice of an advanced theoretical journal the frequent inductive arguments are hardly ever carried out explicitly, but instead the proof just reads something like "by structural induction on n, q.e.d." or "by induction on (x, y) over $<$, q.e.d.", expecting that the mathematically educated reader could easily expand the proof if in doubt.

In contrast, very difficult inductive arguments, sometimes covering several pages, such as the proofs of Hilbert's *first ε-theorem* (Hilbert & Bernays (1968/70), Vol. II) or Gentzen's *Hauptsatz* (Gentzen (1935)), or confluence theorems like the one in Gramlich & Wirth (1996) still require considerable ingenuity and *will* be carried out—but in a style that is very different from the explicit induction process as sketched above! The experienced mathematician engineers his proof more according to the following pattern:

> He starts with the conjecture and simplifies it by case analysis. When he realizes that the current goal becomes similar to an instance of the conjecture, he applies the instantiated conjecture just like a lemma, but keeps in mind that he has actually applied an induction hypothesis. Finally, he searches for some wellfounded ordering in which all the instances of the conjecture he has applied as an induction hypothesis are smaller than the original conjecture itself.

The hard problems in these proofs are

(i) to find the numerous induction hypotheses (as, e.g., to eliminate the Cut in the proof of Gentzen's Hauptsatz) and

(ii) to construct a wellfounded ordering that satisfies the ordering constraints of all these induction hypotheses in parallel (which was, e.g., the hard part for Wilhelm Ackermann in the elimination of the ε-formulas in the proof of the first ε-theorem).

1.1.5 *Descente Infinie* versus Explicit Induction

Explicit induction unfortunately must solve the hard problems (i) and (ii) of the previous section already *before* the proof has actually started. A proper induction axiom must be generated without any information on the structural difficulties that may arise in the proof later on. For this reason, we do not believe that an explicit-induction procedure will ever be able to guess the right induction axioms for very hard proofs in advance. Although the techniques for guessing the right induction axiom by an analysis of the syntax of the conjecture and of the recursive definitions are perhaps the most developed and fascinating applications of heuristic knowledge in artificial intelligence and computer science, even the disciples of explicit induction admit the limitations of this *recursion analysis*. In Protzen (1994) we find not only small verification examples already showing these limits, but also the conclusion:

> "We claim that computing the hypotheses"
>
> [i.e. the instantiation of $\forall u.\ (u{<}v \Rightarrow P(u))$ in (N) and the proof of $\mathsf{Wellf}(<)$]
>
> "*before* the proof is not a solution to the problem and so the central idea for the lazy method is to postpone the generation of hypotheses until it is evident which hypotheses are required for the proof."
>
> [Protzen (1994), p. 43]

This "lazy method" and the label "lazy induction" that was coined in this context are nothing but the reinvention of Fermat's *descente infinie* by the explicit induction community.

Descente infinie and explicit induction do not differ in the task (establishing inductive validity) but in the way the proof search is organized. For simple proofs there is always a straightforward translation between the two. The difference becomes obvious only for proofs of difficult theorems. While explicit induction remains the method of choice for routine tasks, it is an obstacle to progress in the automation of difficult proofs, where the proper induction axioms cannot be guessed in advance.

1.1.6 History and Soundness of *Descente Infinie*

As will be shown in this section, the soundness of the method for engineering hard induction proofs mentioned in Section 1.1.4 is easily seen when the argument is structured as a proof by contradiction, assuming a counterexample. For Fermat's historic reinvention of the method, it is thus just natural that he developed the method itself in terms of assumed counterexamples. He called it "*descente infinie ou indéfinie*". Here it is in modern language:

DEFINITION 1.1 (Method of Descente Infinie)
A proposition Γ can be proved by *descente infinie* as follows:

> *Find a wellfounded ordering $<$ and show that for each assumed counterexample of Γ there is another counterexample of Γ that is smaller in $<$.*

Now, why is this method sound?

> The argument is as follows: Let us assume that Γ is not valid. Then there is a counterexample for Γ. Thus, if we are successful in executing the *Method of Descente Infinie* for the wellfounded ordering $<$, there must be another counterexample for Γ that is smaller in $<$. Now we can iterate the last step *ad infinitum* to get an infinite sequence of counterexamples descending in $<$ (*descente infinie*),[1] but this contradicts the wellfoundedness of $<$, q.e.d.

Note that, although we argue in terms of counterexamples here, the positive argumentation of Section 1.1.4 in terms of application of induction hypotheses does not result in a different proof search, and the resulting proofs are identical if we abstract their structure from the verbalization. While the exact technical relationship between the positive and the negative argumentation can be found in Definition 2.36, the following negative verbalization of our positively stated example proof from Section 1.1.1 should make it intuitively clear:

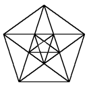
> Well, suppose that there is a counterexample for (1), i.e. some natural number x such that $0+x = x$ is not the case. Since we were successful in showing all cases of our proof, the counterexample must have escaped somehow. This is impossible within the deductive reasoning steps because they are sound. Thus, (1.2.1) must still have a counterexample. By our case assumption, this counterexample is the y with $x = \mathsf{s}(y)$. As (1.2.1) follows from the valid assertion (1.2.1.1) by a sound rewrite step with the equality $0+y=y$, the same y must be a counterexample for this equality, too. As it is an instance of our original proposition (1), to complete the execution of the *Method of Descente Infinie*, we only have to find a wellfounded ordering in which y is smaller than x, and—starting with (1.2.1.2)—we solved this task.

From a positive viewpoint, however, this *inductive* proof can also be seen as a program for computing—given a natural number x as input—a purely *deductive* proof:

> This program has to write down the proof with the exception of the part starting with (1.2.1.2) and then to call itself recursively with y as input. The omitted part of the proof, however, guarantees termination. Therefore, we know that after a finite number of recursive calls—although this number of descents may not be known, i.e. indefinite (*descente indéfinie*)—the program will end up in writing down the base case (1.1).

All in all, it does not really matter whether you prefer to think about *descente infinie* positively or negatively. What is important, however, is to know how to execute the method of proof search. And already the ancient Greeks knew how to do this:

Although we do not have any original Greek mathematical documents from the 5th century B.C. and only fragments from the 4th, the first known occurrence of *descente infinie* in history seems to be the proof of the irrationality of the golden number $\frac{1}{2}(1+\sqrt{5})$ by the Pythagorean mathematician Hippasos of Metapont (Italy) in the middle of the 5th century B.C., cf. Fritz (1945). This proof is carried out geometrically in a pentagram, where the golden number gives the proportion of the length of a line to the length of the side of the enclosing pentagon:



Under the assumption that this proportion is given by $m : n$ with natural numbers m and n, it can be shown that the proportion of the length of a line of a new pentagram drawn inside the inscribed pentagon to the length of the side of this pentagon is $m{-}n : 2n{-}m$, with $0 \prec m{-}n \prec m$, and so forth since the new inscribed pentagram is similar to the original one.

A myth says that the gods drowned Hippasos in the sea, as a punishment for destroying the Pythagoreans' belief that everything is given by positive rational numbers. The resulting confusion seems to have been one of the reasons for the ancient Greek culture to shift interest in mathematics from theorems to proofs.

Moreover, we find many occurrences of *descente infinie* in the famous collection "Elements" of Euclid of Alexandria, cf. Euclid (ca. 300 B.C.). There, the verbalization of an inductive proof has the form of a "*generalizable example*" in the sense that a special concrete counterexample is considered—instead of an arbitrary one—but the existence of a smaller counterexample is actually shown independently of this special choice.

I do not know of *descente infinie* in the following eighteen centuries (except that Euclid's Elements where copied again and again), but only of structural induction, which was known to the Muslim mathematicians around the year 1000 and occurs in a Hebrew book of Levi ben Gerson (Orange and Avignon) in 1321, cf. Katz (1998). Blaise Pascal (1623–1662) (Paris) knew structural induction from a book of Franciscus Maurolycus (Venice) published in 1575 (cf. Bussey (1917)) and used it for the proofs of his Arithmetical Triangle in 1654. While these inductive proofs are still presented as "generalizable examples", in the demonstration of "Conséquence XII" we find—for the first time in known history—a correct verbalization of the related instance of the axiom of structural induction, cf. Pascal (1954), p. 103. Moreover, in the 1650s Pascal exchanged letters on probability theory and *descente infinie* with Pierre Fermat (1607?–1665) (Toulouse), who was the first to describe the *Method of Descente Infinie* explicitly. Besides, Fermat seems to be also the first to provide a correct verbalization of proofs by *descente infinie* and to overcome the presentation of inductive proofs as "generalizable examples", which we would not accept as proper proofs today.

From the description of the *Method of Descente Infinie* in a letter to Pierre de Carcavi in 1659, cf. Fermat (1891), Vol. II, pp. 431–436, it becomes obvious that Fermat extended the analysis of concrete mathematical problems to the analysis of the process of their solution: Instead of just proving a theorem, he analyzed the method of proof search. Similarly, instead of a set of rules that sometimes did find a single solution to the "double equations" of Diophantos of Alexandria (3^{rd} century?) and sometimes did not, he was the first to describe a method to enumerate an infinite set of solutions, cf. Mahoney (1994).

As the competent lawyer and devoted judge Pierre de Fermat (cf. Barner (2001)) was reluctant to release his theorems and is still famous for omitting his proofs, we should be glad that he was generous enough to leave us some of his methods.

1.2 How we Do it

1.2.1 Design Goals for Inductive Inference Systems

Three decades of experience with automated inductive theorem provers, such as NQTHM, INKA, RRL, UNICOM, SPIKE, EXPANDER, &c., cf. Boyer & Moore (1979), Biundo &al. (1986), Kapur & Zhang (1989), Gramlich & Lindner (1991), Bouhoula & Rusinowitch (1995), Padawitz (1998), respectively, leave us with one important message: Successful application of an inductive theorem prover in "real-life" domains requires a knowledgeable human user who can interact with the system at various levels of abstraction. Hence, the development of a new theorem prover—including its inference system—should have an emphasis on its potential for user interaction. Therefore, the following two requirements are main design goals for our inductive inference systems:

I. We want the inference system to comply with natural human proof techniques and to support the user in stating his proof ideas.

II. The user should have no difficulties in understanding and searching for proofs represented within this inference system.

Refining the first design goal we obtain the following requirements:

I.1. All proof problems and sub-problems, the definitions, lemmas, *as well as the induction hypotheses* should be *homogeneously represented*, i.e. expressed in the same language.

I.2. The inference system should include *inference rules for all natural proof steps* (including the repeated application of induction hypotheses on the fly) such that the user can easily formulate his ideas and force the system to follow his proof ideas as closely as possible.

Refining the second design goal we obtain the following requirements:

II.1. The inference system should support a *natural flow of information* in the sense that a decision can be delayed or a commitment deferred, until the state of the proof attempt provides sufficient information for a successful choice. Examples for an unnatural flow of information are:

(a) Instantiating the induction hypotheses in *explicit induction* long before the hypotheses become applicable, cf. Protzen (1994).

(b) The γ-*rule* of a sequent or tableau calculus (without free variables) where an instance has to be guessed long before it becomes apparent whether it will be a successful one or not.

(c) The rules of $\vee$-introduction ($\vee$I) and indirect proof ($\bot_{\mathbf{c}}$ or $A \vee \neg A$) in *natural deduction* calculi for classical logic: The first requires a decision for one of two disjunctive alternatives and the second a decision of when to start an indirect proof.

II.2. Another important requirement for theorem proving is *goal-directedness*, which means that every problem in the graph of a proof attempt is connected to the theorem to be proved. For *inductive* theorem proving this is even more important than for *deductive* theorem proving as new lemmas often have to be invented to close the gap between the induction conclusion and the induction hypotheses. This step is usually guided by the user's knowledge of the domain, the applicable lemmas, the (expanded) induction conclusion, and the induction hypotheses. Without goal-directedness, i.e. without the connection to the induction conclusion *and* the induction hypotheses, the missing lemmas can hardly be guessed.
Note that such "creative steps" are not necessary for *deductive* theorem proving. By Gentzen's Hauptsatz on Cut elimination there is no need to invent new formulas in a proof of a deductive theorem. They can be restricted to "sub"-formulas of the theorem under consideration. In contrast to the lemma application (i.e. Cut) in a deductive proof tree, the application of induction hypotheses and lemmas inside an inductive reasoning cycle cannot generally be eliminated, cf. Kreisel (1965). Thus, for inductive theorem proving, "creativity" cannot be restricted to finding just the proper instances, but may require the invention of new lemmas.

In the spirit of the above design goals, we have an inference system in mind that explicitly provides the concepts of *induction hypothesis* and *induction ordering* and associated means of generating induction ordering conditions with sufficient expressiveness and flexibility, i.e. explicit *weights*. We also want an inference system that does not "hide" repeated applications of induction hypotheses in a single inference step, but instead it should include inference rules that explicitly provide or apply induction hypotheses, given that certain ordering conditions can be met. The inference system must be capable of representing an induction hypothesis as a whole and in a natural and recognizable form. No input normalization may decompose the inductive theorems into "sub"-formulas before the induction hypotheses have been extracted.

1.2.2 Sequent and Tableau Calculi

Obtaining an inference system for *explicit induction* is quite simple: Since the inductive argument is captured in the application of a single inference rule preceeding the call of the first-order deductive machinery, this "induction rule" can just be added to any deductive inference system.

When integrating *descente infinie*, however, the whole inference system is affected and *soundness* becomes a *global* problem. Thus, to go beyond a philosophical discussion, the *soundness* of this integration has to be proved with mathematical rigor.

Notice that we do not provide a proof of the *completeness* of our inference system because there is no appropriate and comprehensive notion of completeness yet: As the theory of arithmetic is not enumerable (Gödel (1931)), completeness w.r.t. the standard notion of validity cannot be achieved. And the common notions of validity for which completeness can be achieved (such as validity in Henkin models) are not sufficient for our goals in this paper, because we are actually not interested just in validity and the mere existence of proofs, but instead the interest is in proof search. Therefore, the proofs have to exist in a *special intentional* form.

The considerations of the previous section provide some guidance for an answer to the following question:

Which deductive inference system is best suited for the integration of descente infinie?

Neither *Hilbert* calculi nor *natural deduction* calculi are really adequate for proof search, because of their unnatural flow of information. Natural deduction is particularly problematic for *descente infinie* because the proofs are augmented with assumptions that conflict with our concept of induction hypothesis. Neither *Sergey Yu. Maslov's inversion technique* nor *non-refutational resolution* seem to be appropriate for proof search, because they lack goal-directedness.[2] Thus, a reasonable integration of *descente infinie* into resolution must be refutational. The only example of such an integration, however, seems to be the inductive theorem prover EXPANDER.[3]

Our choice of a deductive inference system is that of a *sequent* (Gentzen (1935), Lifschitz (1971)), *tableau* (Smullyan (1968), Fitting (1996)), or *matrix calculus* (Andrews (1981), Bibel (1987), Wallen (1990)). While matrix calculi have implementational advantages (cf. Section A.3), for simplicity of presentation we consider only sequent and tableau calculi in this paper.

Now the search for a proof proceeds as follows: Starting with a conjectured sequent, the problem of proving this *goal* is reduced to the problem of proving a set of other sequents as *sub-goals*. The recursive application of such reduction steps results in a tree-like sub-proof structure t_i for each proposition Γ_i. The whole proof consists of a forest of such trees, which are connected by applications of the propositions. Let us defer the discussion of the standard deductive steps within a single tree and have a look now at the new kind of proof steps establishing the connection between the trees.

Suppose we have a huge proof tree of a non-trivial theorem Γ_0. A mathematician organizes such a proof with the help of lemmas. Having identified a lemma Γ_1 in the proof tree, we can cut off (possibly several occurrences of) the subtree rooted by Γ_1, yielding two trees: one for Γ_1 as a new proposition and one for the original theorem Γ_0. Since the latter tree t_0 is incomplete now, we connect it to the new proposition by an inter-tree edge $(1, 0)$, which we call a *lemma application*. Even better than cutting a huge tree into pieces is to follow human practice and to apply lemmas whenever it seems appropriate, and prove them later. Thus, we should not let our tree grow too large. This can be prevented by our rule for lemma application when it introduces a yet unproved proposition as an *open lemma* with a trivial uncompleted proof tree.

While the graph of lemma application has to be acyclic for soundness, this is not the case for a more important but similar proof rule called *induction-hypothesis application.* The application of a proposition $\mathit{\Gamma}_j$ to a proof tree t_i *as an induction hypothesis* looks just like its application *as a lemma*, but starts a new extra sub-tree in t_i. The task of this sub-tree is to prove that the instance of the applied proposition $\mathit{\Gamma}_j$ (*induction hypothesis*) is smaller in some wellfounded ordering than the proposition $\mathit{\Gamma}_i$ (*induction conclusion*) of the proof tree t_i. Moreover—and this is the advantage of the application as an induction hypothesis in comparison to a lemma—the graph of induction-hypothesis application may be cyclic, as long as we still have a wellfounded ordering on it. In the simplest case of $i{=}j$, an induction hypothesis is applied to its own proof tree as in the introductory example of Section 1.1.1. If several trees are involved in a cycle of the application graph, we have *mutual* induction as in the example of Section 3.2.

The following concrete inference rules for deductive reasoning within a tree are presented in a sequent calculus style and may clear away some fog. They will be considered in more detail in Section 2.5. Note that in the good old days when trees grew upwards, Gerhard Gentzen would have inverted the inference rules such that passing the line means consequence. In our case, passing the line means reduction, and trees grow downwards. The inference rules are classified as α-, β-, γ-, and δ-rules (Smullyan (1968)):

α**-rules** describe the simple and

β**-rules** the case-splitting (or branching) propositional proof steps.

γ**-rules** show existential properties, either by exhibiting a term witnessing the existence or else by introducing a special kind of variable, called "dummy" in Prawitz (1960) and Kanger (1963), "free" in Fitting (1996) and in footnote 11 of Prawitz (1960), and "meta" in the field of planning and constraint solving. It may be used to delay the choice of a witnessing term until the state of the proof search provides more information. In this paper, however, as these names would be misleading, we call such a variable a *free γ-variable.*

δ**-rules** show universal properties using a new symbol, called a "parameter" or an "eigenvariable", about which nothing is known. We use nullary parameters called *free δ-variables.* These variables are not free in the sense that the terms to replace them may be chosen freely, but in the sense that their occurrences must not be bound by a quantifier or binder. The free δ-variables subdivide into the ordinary *free δ^--variables* introduced by standard δ-steps and the *free δ^+-variables* introduced together with a constraint (attached to the upper right of the rules) by liberalized δ-steps (δ^+-steps, cf. Section 2.1.5). Liberalized δ-rules differ from standard ones in the *variable-conditions* they introduce (attached to the lower right of the rules). Variable-conditions represent the dependency between free variables, cf. Prawitz (1960), Kanger (1963), Bibel (1987), Kohlhase (1995).

Other rules may be added for an appropriate treatment of frequent reasoning patterns such as rewriting with equalities or logical equivalences, unification, or the Cut.

Let A and B be formulas, $\mathit{\Gamma}$, $\mathit{\Pi}$, and $\mathit{\Lambda}$ be sequents, i.e. disjunctive lists of formulas. Let $x \in \mathrm{V}_{\mathrm{bound}}$ be a bound variable, and let $\mathcal{F}$ be the current proof forest, such that $\mathcal{V}(\mathcal{F})$ contains all variables already used, especially those from $\mathit{\Gamma}$, $\mathit{\Pi}$, and A:

α-rules:

$$\frac{\mathit{\Gamma}\ \neg\neg A\ \mathit{\Pi}}{A\ \mathit{\Gamma}\ \mathit{\Pi}} \qquad \frac{\mathit{\Gamma}\ (A{\vee}B)\ \mathit{\Pi}}{A\ B\ \mathit{\Gamma}\ \mathit{\Pi}} \qquad \frac{\mathit{\Gamma}\ \neg(A{\wedge}B)\ \mathit{\Pi}}{\overline{A}\ \overline{B}\ \mathit{\Gamma}\ \mathit{\Pi}} \qquad \frac{\mathit{\Gamma}\ (A{\Rightarrow}B)\ \mathit{\Pi}}{\overline{A}\ B\ \mathit{\Gamma}\ \mathit{\Pi}} \qquad \frac{\mathit{\Gamma}\ (A{\Leftarrow}B)\ \mathit{\Pi}}{A\ \overline{B}\ \mathit{\Gamma}\ \mathit{\Pi}}$$

β-rules: In the following rules we may choose to *fold down* none or one, but not both of the *side-formulas* in the optional brackets $[\cdots]$. For example, if we choose the first lower sequent of the first rule to be $A\ \overline{B}\ \mathit{\Gamma}\ \mathit{\Pi}$ then its second lower sequent must be $B\ \mathit{\Gamma}\ \mathit{\Pi}$.

$$\frac{\mathit{\Gamma}\ (A{\wedge}B)\ \mathit{\Pi}}{A\ [\,\overline{B}\,]\ \mathit{\Gamma}\ \mathit{\Pi} \qquad B\ [\,\overline{A}\,]\ \mathit{\Gamma}\ \mathit{\Pi}} \qquad\qquad \frac{\mathit{\Gamma}\ \neg(A{\vee}B)\ \mathit{\Pi}}{\overline{A}\ [\,B\,]\ \mathit{\Gamma}\ \mathit{\Pi} \qquad \overline{B}\ [\,A\,]\ \mathit{\Gamma}\ \mathit{\Pi}}$$

$$\frac{\mathit{\Gamma}\ \neg(A{\Rightarrow}B)\ \mathit{\Pi}}{A\ [\,B\,]\ \mathit{\Gamma}\ \mathit{\Pi} \qquad \overline{B}\ [\,\overline{A}\,]\ \mathit{\Gamma}\ \mathit{\Pi}} \qquad\qquad \frac{\mathit{\Gamma}\ \neg(A{\Leftarrow}B)\ \mathit{\Pi}}{\overline{A}\ [\,\overline{B}\,]\ \mathit{\Gamma}\ \mathit{\Pi} \qquad B\ [\,A\,]\ \mathit{\Gamma}\ \mathit{\Pi}}$$

γ-rules: Let t be any term:

$$\frac{\mathit{\Gamma}\ \ \exists x.A\ \ \mathit{\Pi}}{A\{x{\mapsto}t\}\ \ \mathit{\Gamma}\ \ \exists x.A\ \ \mathit{\Pi}} \qquad\qquad \frac{\mathit{\Gamma}\ \ \neg\forall x.A\ \ \mathit{\Pi}}{\overline{A\{x{\mapsto}t\}}\ \ \mathit{\Gamma}\ \ \neg\forall x.A\ \ \mathit{\Pi}}$$

δ-rules: Let $x^{\delta^-} \in \mathrm{V}_{\delta^-} \setminus \mathcal{V}(\mathcal{F})$ be a new[4] free δ^--variable:

$$\frac{\mathit{\Gamma}\ \ \forall x.A\ \ \mathit{\Pi}}{A\{x{\mapsto}x^{\delta^-}\}\ \ \mathit{\Gamma}\ \ \mathit{\Pi}}\ \ \mathcal{V}_{\gamma\delta^+}(A, \mathit{\Gamma}\mathit{\Pi}, \sqsupset) \times \{x^{\delta^-}\}$$

$$\frac{\mathit{\Gamma}\ \ \neg\exists x.A\ \ \mathit{\Pi}}{\overline{A\{x{\mapsto}x^{\delta^-}\}}\ \ \mathit{\Gamma}\ \ \mathit{\Pi}}\ \ \mathcal{V}_{\gamma\delta^+}(A, \mathit{\Gamma}\mathit{\Pi}, \sqsupset) \times \{x^{\delta^-}\}$$

Liberalized δ-rules: Let $x^{\delta^+} \in \mathrm{V}_{\delta^+} \setminus \mathcal{V}(\mathcal{F})$ be a new[5] free δ^+-variable:

$$\frac{\mathit{\Gamma}\ \ \forall x.A\ \ \mathit{\Pi}}{A\{x{\mapsto}x^{\delta^+}\}\ \ \mathit{\Gamma}\ \ \mathit{\Pi}}\ \ \begin{matrix}\{(x^{\delta^+}, \overline{A\{x{\mapsto}x^{\delta^+}\}}\,)\} \\ \mathcal{V}_{\mathrm{free}}(A) \times \{x^{\delta^+}\}\end{matrix}$$

$$\frac{\mathit{\Gamma}\ \ \neg\exists x.A\ \ \mathit{\Pi}}{\overline{A\{x{\mapsto}x^{\delta^+}\}}\ \ \mathit{\Gamma}\ \ \mathit{\Pi}}\ \ \begin{matrix}\{(x^{\delta^+}, A\{x{\mapsto}x^{\delta^+}\})\} \\ \mathcal{V}_{\mathrm{free}}(A) \times \{x^{\delta^+}\}\end{matrix}$$

Rewrite-Rules: Let s and t be terms (of the same type). Let B be one of the formulas $(s{\neq}t)$ or $(t{\neq}s)$. Let $A[t]$ denote the formula $A[s]$ with some occurrences of s replaced with t:

$$\frac{\mathit{\Gamma}\ \ A[s]\ \ \mathit{\Pi}\ \ B\ \ \mathit{\Lambda}}{A[t]\ \ \mathit{\Gamma}\ \ \mathit{\Pi}\ \ B\ \ \mathit{\Lambda}} \qquad\qquad \frac{\mathit{\Gamma}\ \ B\ \ \mathit{\Pi}\ \ A[s]\ \ \mathit{\Lambda}}{A[t]\ \ \mathit{\Gamma}\ \ B\ \ \mathit{\Pi}\ \ \mathit{\Lambda}}$$

Cut:

$$\frac{\mathit{\Gamma}}{A\ \ \mathit{\Gamma} \qquad \overline{A}\ \ \mathit{\Gamma}}$$

1.2.3 Skolemization

Contrary to most first-order deductive frameworks, *Skolemization* is not appropriate for *descente infinie*, whereas a dual of Skolemization called *raising* is unproblematic just as in Miller (1992), but for additional reasons. The problematic aspects of Skolemization in the context of *descente infinie* are the following two:

Firstly, Skolemization enriches the signature. Unless special care is taken, this may introduce objects into empty universes, change the notions of Herbrand and Henkin models and of *inductive validity* (cf. Wirth & Gramlich (1994b)), and it may imply the Axiom of Choice even if it is not part of the original theory. Apart from that, Skolem functions that cannot be translated back into the original signature may occur in answers to queries or in solutions of constraints.

Secondly, Skolemization destroys the locality of counterexamples we need for *descente infinie*. To see this, consider the following example: When we apply (outer) validity-invariant Skolemization to

$$\exists w.\ \forall x.\ \exists y.\ \forall z.\ \mathit{\Gamma}(w, y, x, z)$$

we get

$$\exists w.\ \exists y.\ \mathit{\Gamma}(w, y, x'(w), z'(w, y)),$$

where x' and z' are the new Skolem functions for x and z, respectively. Note that the dual unsatisfiability-invariant form of Skolemization applied in refutational resolution and tableau calculi would introduce Skolem functions for w and y instead. The validity of the latter formula is equivalent to the validity of the formula

$$\forall x'.\ \forall z'.\ \exists w.\ \exists y.\ \mathit{\Gamma}(w, y, x'(w), z'(w, y)).$$

Seen abstractly and independently from proving validity or unsatisfiability, Skolemization is the operation that moves the quantifiers of all δ-variables to the very left and gives them some γ-variables as arguments. Thus, Skolemization results in the following simplified quantificational structure:

> For all Skolem functions $\boldsymbol{u}$ there are solutions to the γ-variables $\boldsymbol{e}$ such that the quantifier-free theorem $\mathit{\Gamma}(\boldsymbol{e}, \boldsymbol{u})$ is valid (i.e. $\forall \boldsymbol{u}.\ \exists \boldsymbol{e}.\ \mathit{\Gamma}(\boldsymbol{e}, \boldsymbol{u})$).

When the state of the proof search is represented as the conjunction of the branches of a tree (as in sequent or tableau calculi), the γ-variables become “rigid” or “global”, i.e. a solution for a γ-variable must solve *all* occurrences of this variable in the whole proof tree. This is unfortunately so, because, if $B_0, \ldots, B_n$ denote the branches of a proof tree for $\mathit{\Gamma}(\boldsymbol{e}, \boldsymbol{u})$, then

$$\forall \boldsymbol{u}.\ \exists \boldsymbol{e}.\ (\ B_0 \wedge \ldots \wedge B_n\ )$$

is strictly stronger than

$$\forall \boldsymbol{u}.\ (\ \exists \boldsymbol{e}.\ B_0\ \wedge\ \ldots\ \wedge\ \exists \boldsymbol{e}.\ B_n\ )$$

Considering this tree structure, it can be easily seen that the quantificational structure resulting from Skolemization makes *descente infinie* impossible, however, because different applications of induction hypotheses may destroy the counterexample:

> Suppose we have some counterexample $\boldsymbol{u}$ for $\mathit{\Gamma}(\boldsymbol{e}, \boldsymbol{u})$ (i.e. there is no $\boldsymbol{e}$ such that $\mathit{\Gamma}(\boldsymbol{e}, \boldsymbol{u})$ is valid) then, for different $\boldsymbol{e}$, different branches B_i in the proof tree may cause the invalidity of the conjunction. If we have applied induction hypotheses in more than one branch, for different $\boldsymbol{e}$ we get different smaller counterexamples for different branches. What we would need, however, is *one single* smaller counterexample for all $\boldsymbol{e}$.

These problematic aspects are no longer present when Skolemization is replaced with *raising* (cf. Miller (1992)), which simplifies the quantificational structure to:

> There are raising functions $\boldsymbol{e}$ such that for all possible values of the free δ-variables $\boldsymbol{u}$ the quantifier-free theorem $\mathit{\Gamma}(\boldsymbol{e}, \boldsymbol{u})$ is valid (i.e. $\exists \boldsymbol{e}.\ \forall \boldsymbol{u}.\ \mathit{\Gamma}(\boldsymbol{e}, \boldsymbol{u})$).

The inverted order of universal and existential quantification of raising (compared to Skolemization) is advantageous in our case because now applications of induction hypotheses work well:

> When, for some—fixed—$\boldsymbol{e_0}$, we have some counterexample $\boldsymbol{u}$ for $\mathit{\Gamma}(\boldsymbol{e_0}, \boldsymbol{u})$ then *one single* branch B_i in the proof tree must cause the invalidity of the conjunction. If this branch is closed, then it contains the application of an induction hypothesis that is invalid for the $\boldsymbol{u}'$ resulting from the instantiation of the hypothesis. Thus, $\boldsymbol{u}'$ together with the induction hypothesis provides the strictly smaller counterexample we are looking for.

1.2.4 Preservation of Solutions

Question answering systems, such as PROLOG, compute answers to queries that contain free γ-variables to be instantiated. When the proof search is successfully completed, the existentially quantified query is known to be valid. Moreover, the substitution computed for the free γ-variables *solves* the query in the sense that its instance is a valid answer. Since the knowledge of mere existence is less useful than the knowledge of concrete witnesses, theorem proving should—if possible without overhead—always provide these solutions.

Regarding *descente infinie*, however, the following closely related property is not only desirable, but necessary for soundness.

> *All substitutions of free γ-variables that close a proof attempt for a proposition are also solutions of the original proposition. (Preservation of Solutions)*

Why do we need this property?

> Well, suppose that our original input theorem $\mathit{\Gamma}(\boldsymbol{e}, \boldsymbol{u})$ (cf. the discussion in the previous section) has been reduced to $G(\boldsymbol{e}, \boldsymbol{u})$ representing the state of the proof search. Furthermore, suppose that we have found some instance $\boldsymbol{e_0}$ such that, for each counterexample $\boldsymbol{u}$ of $G(\boldsymbol{e_0}, \boldsymbol{u})$, there is a counterexample $\boldsymbol{u}'$ for the original theorem (i.e. $\mathit{\Gamma}(\boldsymbol{e_0}, \boldsymbol{u}')$ is invalid) and that this $\boldsymbol{u}'$ is strictly smaller than $\boldsymbol{u}$ in some wellfounded ordering. In this case we have proved $\mathit{\Gamma}(\boldsymbol{e_0}, \boldsymbol{u})$ (and thus $\mathit{\Gamma}(\boldsymbol{e}, \boldsymbol{u})$) only if
>
> each counterexample $\boldsymbol{u}$ for $\mathit{\Gamma}(\boldsymbol{e_0}, \boldsymbol{u})$ is also a counterexample for $G(\boldsymbol{e_0}, \boldsymbol{u})$.
>
> The latter is the contrapositive—and therefore an equivalent—of the following property given by “preservation of solutions”:
>
> $G(\boldsymbol{e_0}, \boldsymbol{u})$ implies $\mathit{\Gamma}(\boldsymbol{e_0}, \boldsymbol{u})$ for each $\boldsymbol{u}$.

2 Formal Development

2.1 Technical Prerequisites

2.1.1 Basic Notions and Notation

'$\mathbf{N}$' denotes the set of natural numbers and '$\prec$' the ordering on $\mathbf{N}$. Let $\mathbf{N}_+ := \{\, n \in \mathbf{N} \mid 0 \neq n \,\}$. '$\mathbf{Z}$' denotes the set of integers. We use '$\uplus$' for the union of disjoint classes and 'id' for the identity function. For classes R, A, and B we define:

$$\begin{array}{lll}
\mathrm{dom}(R) := \{a \mid \exists b.\ (a,b) \in R\} & \textit{domain} & \\
{}_A{\upharpoonleft}R \quad := \{(a,b) \in R \mid a \in A\} & \textit{restriction to } A & \\
\langle A \rangle R \quad := \{b \mid \exists a \in A.\ (a,b) \in R\} & \textit{image of } A, & \text{i.e. } \langle A \rangle R = \mathrm{ran}({}_A{\upharpoonleft}R)
\end{array}$$

And the dual ones:

$$\begin{array}{lll}
\mathrm{ran}(R) \ := \{b \mid \exists a.\ (a,b) \in R\} & \textit{range} & \\
R{\upharpoonright}_B \quad := \{(a,b) \in R \mid b \in B\} & \textit{range-restriction to } B & \\
R\langle B \rangle \quad := \{a \mid \exists b \in B.\ (a,b) \in R\} & \textit{reverse-image of } B, & \text{i.e. } R\langle B \rangle = \mathrm{dom}(R{\upharpoonright}_B)
\end{array}$$

Furthermore, we use '$\emptyset$' to denote the empty set as well as the empty function. Functions are (right-) unique relations and the meaning of '$f \circ g$' is extensionally given by $(f \circ g)(x) = g(f(x))$. Note that we take the operator '$\circ$' to have higher priority than the operators '$\cup$' and '$\uplus$'. The *class of total functions from A to B* is denoted as $A \to B$. The *class of (possibly) partial functions from A to B* is denoted as $A \rightsquigarrow B$. Both $\to$ and $\rightsquigarrow$ associate to the right, i.e. $A \rightsquigarrow B \to C$ reads $A \rightsquigarrow (B \to C)$.

Let R be a binary relation. R is said to be a relation *on A* if

$$\mathrm{dom}(R) \cup \mathrm{ran}(R) \ \subseteq \ A.$$

R is *irreflexive* if $\mathrm{id} \cap R = \emptyset$. It is *$A$-reflexive* if ${}_A{\upharpoonleft}\mathrm{id} \subseteq R$. Speaking of a *reflexive* relation we refer to the largest A that is appropriate in the local context, and referring to this A we write R^0 to ambiguously denote ${}_A{\upharpoonleft}\mathrm{id}$. With $R^1 := R$, and $R^{n+1} := R^n \circ R$ for $n \in \mathbf{N}_+$, R^m denotes the m-step relation for R. The *transitive closure* of R is $R^+ := \bigcup_{n \in \mathbf{N}_+} R^n$. The *reflexive & transitive closure* of R is $R^* := \bigcup_{n \in \mathbf{N}} R^n$.

The *reverse* of R is $R^{-1} := \{\, (b,a) \mid (a,b) \in R \,\}$. A sequence $(s_i)_{i \in \mathbf{N}}$ is *non-terminating in R* if $s_i \; R \; s_{i+1}$ for all $i \in \mathbf{N}$. R is *terminating* if there are no non-terminating sequences in R. A relation R (on A) is *wellfounded* if any non-empty class B ($\subseteq A$) has an R-minimal element, i.e. $\exists a \in B.\ \neg \exists a' \in B.\ a' R\, a$.

A *quasi-ordering '$\lesssim$' on* a class A is an A-reflexive and transitive (binary) relation on A, and we define $a \gtrsim b$ if $b \lesssim a$. By an (irreflexive) *ordering* '$<$' we mean an irreflexive and transitive relation, called "strict partial ordering" by some authors. A *reflexive ordering '$\leq$' on A* is an A-reflexive, anti-symmetric, and transitive relation on A. The *ordering $<$ of a quasi-ordering or a reflexive ordering $\lesssim$* is $\lesssim \setminus \gtrsim$, and $\lesssim$ is called *wellfounded* if $<$ is wellfounded.

LEMMA 2.1
For a binary relation R we have the following equivalence:
R is wellfounded iff R^+ is a wellfounded ordering.

2.1.2 Dependent Choice, Wellfoundedness, and *Descente Infinie*

It is well-known that the Axiom of Foundation and the Axiom of Choice do not destroy the consistency of set theory (cf. Gödel (1986 ff.), Vol. II), but it is not always appropriate to assume their validity. As the Axiom of Choice implies all known forms of induction, its inclusion is inappropriate for a comparison of the logical strength of different forms of induction. A weak form (or proper logical consequence) of the Axiom of Choice is the following (cf. Rubin & Rubin (1985), p. 19; Howard & Rubin (1998), Form 43, p. 30):

DEFINITION 2.2 (Principle of Dependent Choice)
If R is a binary relation with $\mathrm{ran}(R) \subseteq \mathrm{dom}(R) \neq \emptyset$, then R is not terminating.

In this paper, we define wellfoundedness via the existence of minimal elements in classes, but a well-known alternative is to define it as termination of the reverse relation. While the converse of the following principle is tautological, the principle itself is not, and it makes wellfoundedness independent of the actual choice of its definition (cf. Howard & Rubin (1998), Form 43 R, p. 32):

DEFINITION 2.3 (Principle of Wellfoundedness)
If $<$ is an ordering and $>$ is terminating, then $<$ is wellfounded.

DEFINITION 2.4 (Principle of Descente Infinie)
If $<$ is an ordering and the class A has no $<$-minimal elements and either
(i) $> \cap (A{\times}A)$ is terminating, or
(ii) each $C \subseteq A$ that is totally ordered by $<$ has a $<$-minimal element
then A is empty.

The soundness of the Method of Descente Infinie of Definition 1.1 is achieved independently of the alternatives of the definition of wellfoundedness by setting A to be the class of counterexamples of Γ in (i) of Definition 2.4.[6] This version appears to be slightly stronger than (ii), which is listed in Howard & Rubin (1998), p. 31, as Form 43 K (formerly Form 43 W (Wirth?) in Note 146, p. 317f.). However, in fact, all these principles are equivalent:

LEMMA 2.5
The Principles of Dependent Choice, Wellfoundedness, and Descente Infinie (both (i) and (ii)) are logically equivalent in set theory, even without the axioms of Choice, Foundation, or Power-Set.

Finally, it deserves mentioning that it is theoretically possible to use, instead of the Principle of Dependent Choice, the strictly stronger Axiom of Choice (or Zorn's Lemma) to obtain a soundness principle for a stronger induction method than the Method of Descente Infinie. For this we would replace "$> \cap (A{\times}A)$ is terminating" in (i) of Definition 2.4 with "each non-terminating sequence in $> \cap (A{\times}A)$ has a $< \cap (A{\times}A)$-lower bound", cf. Geser (1995).

2.1.3 Syntax

To avoid the problem of binders capturing free variables (cf. below) and in the tradition of Gentzen (1935), Hilbert & Bernays (1968/70), and Snyder & Gallier (1989), we assume the following four sets of symbols to be disjoint:

V_{γ}	*free γ-variables*, i.e. the free variables of Fitting (1996)
V_{δ}	*free δ-variables*, i.e. nullary parameters, instead of Skolem functions
$\mathrm{V}_{\mathrm{bound}}$	*bound variables*, i.e. variables occurring only bound, cf. below
Σ	*constants*, i.e. the function (and predicate) symbols from the signature

We partition the free δ-variables V_{δ} into *free δ^{-}-variables* $\mathrm{V}_{\delta^{-}}$ that are introduced by the (non-liberalized) δ-rules; and *free δ^{+}-variables* $\mathrm{V}_{\delta^{+}}$ that are introduced by the liberalized δ-rules (δ^{+}-rules), cf. the end of Section 1.2.2 or Section 2.1.5:

$$\mathrm{V}_{\delta} = \mathrm{V}_{\delta^{-}} \uplus \mathrm{V}_{\delta^{+}}$$

We define the *free variables* by

$$\mathrm{V}_{\mathrm{free}} := \mathrm{V}_{\gamma} \uplus \mathrm{V}_{\delta}$$

and the *variables* by

$$\mathrm{V} := \mathrm{V}_{\mathrm{bound}} \uplus \mathrm{V}_{\mathrm{free}}$$

Finally:

$$\mathrm{V}_{\gamma\delta^{+}} := \mathrm{V}_{\gamma} \uplus \mathrm{V}_{\delta^{+}}$$

We use '$\mathcal{V}_{k}(\mathit{\Gamma})$' to denote the set of variables from V_{k} occurring in $\mathit{\Gamma}$.

We define a *sequent* to be a list of formulas. The *conjugate* of a formula A (written: $\overline{A}$) is the formula B if A is of the form $\neg B$, and the formula $\neg A$ otherwise. In the tradition of Hilbert & Bernays (1968/70), we do not permit binding of variables that already occur bound in a term or formula; that is: $\forall x.\ A$ is only a formula if no binder on x already occurs in A. The simple effect is that our formulas are easier to read and our γ- and δ-rules (and $\lambda\beta$-reduction) can replace *all* occurrences of x. Moreover, we assume that all binders have minimal scope, e.g. $\forall x, y.\ A \wedge B$ reads $(\forall x.\ \forall y.\ A) \wedge B$.

Let σ be a substitution. We say that σ is a *substitution on X* if $\mathrm{dom}(\sigma) \subseteq X$. We denote with '$\mathit{\Gamma}\sigma$' the result of replacing each occurrence of a variable $x \in \mathrm{dom}(\sigma)$ in $\mathit{\Gamma}$ with $\sigma(x)$. Unless otherwise stated, we tacitly assume that all occurrences of variables from $\mathrm{V}_{\mathrm{bound}}$ in a term or formula or in the range of a substitution are *bound occurrences* (i.e. that a variable $x \in \mathrm{V}_{\mathrm{bound}}$ occurs only in the scope of a binder on x) and that each substitution σ satisfies $\mathrm{dom}(\sigma) \subseteq \mathrm{V}_{\mathrm{free}}$, so that no bound occurrences of variables can be replaced and no additional variable occurrences can become bound (i.e. captured) when applying σ.

2.1.4 Semantical Requirements

Instead of defining validity from scratch, we just require some abstract properties as stated below, which normally hold in all two-valued semantics, such as in classical first-order, intensional, modal, or higher-order logic.

Validity is given relative to some Σ-structure $\mathcal{A}$, assigning a non-empty universe (or "carrier") to each type. For $\mathrm{X} \subseteq \mathrm{V}$ we denote the set of total $\mathcal{A}$-valuations of X (i.e. functions mapping variables to objects of the universe of $\mathcal{A}$ (respecting types)) with

$$\mathrm{X} \rightarrow \mathcal{A}$$

and the set of (possibly) partial $\mathcal{A}$-valuations of X with

$$\mathrm{X} \rightsquigarrow \mathcal{A}$$

For $\tau : \mathrm{X} \rightarrow \mathcal{A}$ we denote with '$\mathcal{A} \uplus \tau$' the extension of $\mathcal{A}$ to the variables of X. More precisely, we assume the existence of some evaluation function 'eval' such that $\mathrm{eval}(\mathcal{A} \uplus \tau)$ maps any term whose constants and free occurring variables are from $\Sigma \uplus \mathrm{X}$ into the universe of $\mathcal{A}$ (respecting types) such that for all $x \in \mathrm{X}$:

$$\mathrm{eval}(\mathcal{A} \uplus \tau)(x) \quad = \quad \tau(x)$$

Moreover, $\mathrm{eval}(\mathcal{A} \uplus \tau)$ maps any formula B whose constants and free occurring variables are from $\Sigma \uplus \mathrm{X}$ to TRUE or FALSE, such that

$$B \text{ is valid in } \mathcal{A} \uplus \tau \quad \text{iff} \quad \mathrm{eval}(\mathcal{A} \uplus \tau)(B) = \mathsf{TRUE}$$

Notice that we leave open what our formulas and what our Σ-structures exactly are. The latter can range from a first-order Σ-structure to a higher-order[7] modal[8] Σ-model, provided that the following two properties are satisfied:

EXPLICITNESS-LEMMA

(Andrews (1972), Lemma 2; Andrews (2002), Proposition 5400; Fitting (2002), Proposition 2.30)

Let B be a term or formula (possibly with some unbound occurrences of variables from $\mathrm{V}_{\mathrm{bound}}$).
Let $\mathcal{A}$ be a Σ-structure with valuation $\tau : \mathrm{V} \rightsquigarrow \mathcal{A}$.

The value of the evaluation function on B depends only on the valuation of those variables that actually occur free in B; formally:

For X being the set of variables that occur free in B, if $\mathrm{X} \subseteq \mathrm{dom}(\tau)$, then:

$$\mathrm{eval}(\mathcal{A} \uplus \tau)(B) \quad = \quad \mathrm{eval}(\mathcal{A} \;\uplus\; {}_{\mathrm{X}}\!\upharpoonleft\!\tau)(B).$$

SUBSTITUTION-LEMMA

(also called "Substitution-Value-Lemma"; Andrews (1972), Lemma 3; Andrews (2002), Lemma 5401(a); Enderton (1973), p. 127; Fitting (1996), p. 120; Fitting (2002), Proposition 2.31)

Let B be a term or formula (possibly with some unbound occurrences of variables from $\mathrm{V}_{\mathrm{bound}}$).
Let σ be a substitution. Let $\mathcal{A}$ be a Σ-structure with valuation $\tau : \mathrm{V} \rightsquigarrow \mathcal{A}$.

If the variables that occur free in $B\sigma$ belong to $\mathrm{dom}(\tau)$, then:

$$\mathrm{eval}(\mathcal{A} \uplus \tau)(B\sigma) \quad = \quad \mathrm{eval}\big(\; \mathcal{A} \;\;\uplus\;\; (\,\sigma \;\uplus\; {}_{\mathrm{V}\setminus\mathrm{dom}(\sigma)}\!\upharpoonleft\!\mathrm{id}\,) \circ \mathrm{eval}(\mathcal{A} \uplus \tau) \;\big)\big(\; B \;\big).$$

2.1.5 The Liberalized δ-rule

While the benefit of free γ-variables in γ-rules is to delay the choice of a witness term, it is sometimes unsound to instantiate a free γ-variable x^γ with a term containing a free δ-variable y^δ that was introduced later than x^γ:

EXAMPLE 2.6
The formula $\exists x.\ \forall y.\ (x = y)$
is not generally valid. We can start a proof attempt as follows:
γ-step: $\forall y.\ (x^\gamma = y),\quad \exists x.\ \forall y.\ (x = y)$
δ-step: $(x^\gamma = y^\delta),\quad \exists x.\ \forall y.\ (x = y)$
Now, if the free γ-variable x^γ could be substituted by the free δ-variable y^δ, we would get the tautology $(y^\delta = y^\delta)$, i.e. we would have proved an invalid formula. To prevent this, the δ-step has to record (x^γ, y^δ) in a variable-condition, where (x^γ, y^δ) means that x^γ is older than y^δ, so that we must not instantiate the free γ-variable x^γ with a term containing the free δ-variable y^δ.

DEFINITION 2.7 (Variable-Condition)
A *variable-condition* is a subset of $\mathrm{V}_{\mathrm{free}} \times \mathrm{V}_{\mathrm{free}}$.

To restrict the possible instantiations as little as possible, we should keep our variable-conditions as small as possible. Kanger (1963), Bibel (1987), and Wallen (1990) are quite generous in that they let their variable-conditions grow too much:

EXAMPLE 2.8
The valid formula $\exists x.\ \big(\ \forall y.\ \neg\mathsf{P}(y)\ \vee\ \mathsf{P}(x)\ \big)$
can be proved the following way:
γ-step: $\forall y.\ \neg\mathsf{P}(y)\ \vee\ \mathsf{P}(x^\gamma),\quad \exists x.\ \big(\ \forall y.\ \neg\mathsf{P}(y)\ \vee\ \mathsf{P}(x)\ \big)$
α-step: $\forall y.\ \neg\mathsf{P}(y),\quad \mathsf{P}(x^\gamma),\quad \exists x.\ \big(\ \forall y.\ \neg\mathsf{P}(y)\ \vee\ \mathsf{P}(x)\ \big)$
Liberalized δ-step: $\neg\mathsf{P}(y^{\delta^+}),\quad \mathsf{P}(x^\gamma),\quad \exists x.\ \big(\ \forall y.\ \neg\mathsf{P}(y)\ \vee\ \mathsf{P}(x)\ \big)$
Instantiation step: $\neg\mathsf{P}(y^{\delta^+}),\quad \mathsf{P}(y^{\delta^+}),\quad \exists x.\ \big(\ \forall y.\ \neg\mathsf{P}(y)\ \vee\ \mathsf{P}(x)\ \big)$
The final step is not allowed in the works cited above, so yet another γ-step must be applied to the original formula. Our instantiation step, however, is perfectly sound in classical logic: Since x^γ does not occur in $\forall y.\ \neg\mathsf{P}(y)$, the free variables x^γ and y^{δ^+} are independent and there is no reason to insist on x^γ being older than y^{δ^+}. Indeed, we can execute the δ-step introducing y^{δ^+} *before* the γ-step introducing x^γ, when we begin with moving-in the existential quantifier, transforming the original formula into the logically equivalent formula $\forall y.\ \neg\mathsf{P}(y)\ \vee\ \exists x.\ \mathsf{P}(x)$.

Keeping the variable-conditions small may lead to an exponential and even non-elementary reduction of the size of the smallest proof. The "liberalization of the δ-rule" and its reduction in the size of the smallest proof has the following history: Smullyan (1968), Hähnle & Schmitt (1994) (δ^+), Beckert &al. (1993) (δ^{+^+}), Baaz & Fermüller (1995) (δ^*), Giese & Ahrendt (1999) (δ^ε), Cantone & Nicolosi-Asmundo (2000) (δ^{*^*}). The step from δ^+ to δ^{+^+} (like the one from δ^{+^+} to δ^ε) does not reduce the variable-condition (as all others do) but reduces the number of Skolem symbols (just like the step from δ^* to δ^{*^*}). While already the earliest liberalized δ-rule of Smullyan (1968) proves the formula of Example 2.8 with a single γ-step, it is much more restrictive than the δ^+-rule which can be applied in the presence of free γ-variables.

Important for our goals in proof search, however, is that the liberalization of the δ-rule provides additional proofs that are not only shorter but also more natural and easier to find in the sense of the discussion in Section 1.2.1. The problematic step in our case is the one from the non-liberalized δ-rule to the liberalized δ^+-rule, because it destroys the preservation of solutions (cf. Section 1.2.4) as discussed in Section 2.2.4. Some further improvements on δ^+ are discussed in Section A.

Note that the liberalization of the δ-rule is not as simple as it may seem, because it may lead to an unsound calculus, cf. Kohlhase (1995) w.r.t. our Example 2.9 and Kohlhase (1998) w.r.t. our Example 2.50. The difficulty is with instantiation steps that relate previously unrelated variables:

EXAMPLE 2.9

The formula $\exists x.\ \forall y.\ \mathsf{Q}(x,y)\ \vee\ \exists u.\ \forall v.\ \neg\mathsf{Q}(v,u)$

is not generally valid (to wit, let Q be the identity relation on a non-trivial universe).

Consider the following proof attempt: One α-, two γ-, and two liberalized δ-steps result in

(2.9.1) $$\mathsf{Q}(x^{\gamma},y^{\delta^+}),\quad \neg\mathsf{Q}(v^{\delta^+},u^{\gamma}),\quad \exists x.\ \forall y.\ \mathsf{Q}(x,y),\quad \exists u.\ \forall v.\ \neg\mathsf{Q}(v,u)$$

with variable-condition

(2.9.2) $$R\ :=\ \{(x^{\gamma},y^{\delta^+}),\quad (u^{\gamma},v^{\delta^+})\}$$

Notice that the non-liberalized δ-rule would have produced in addition $(x^{\gamma},v^{\delta^+})$ or $(u^{\gamma},y^{\delta^+})$ or both, depending on the order of the proof steps. When we now instantiate x^{γ} with v^{δ^+}, we relate the previously unrelated variables u^{γ} and y^{δ^+}. Thus, our new goal

$$\mathsf{Q}(v^{\delta^+},y^{\delta^+}),\quad \neg\mathsf{Q}(v^{\delta^+},u^{\gamma}),\quad \exists x.\ \forall y.\ \mathsf{Q}(x,y),\quad \exists u.\ \forall v.\ \neg\mathsf{Q}(v,u)$$

must be equipped with the new variable-condition $(u^{\gamma},y^{\delta^+})$. Otherwise we could instantiate u^{γ} with y^{δ^+}, resulting in the tautology

$$\mathsf{Q}(v^{\delta^+},y^{\delta^+}),\quad \neg\mathsf{Q}(v^{\delta^+},y^{\delta^+}),\quad \ldots$$

Notice that in the standard framework of Skolemization and unification, this new variable-condition is automatically generated by the occur-check of unification:

When we instantiate x^{γ} with $v^{\delta^+}(u^{\gamma})$ in

$$\mathsf{Q}(x^{\gamma},y^{\delta^+}(x^{\gamma})),\quad \neg\mathsf{Q}(v^{\delta^+}(u^{\gamma}),u^{\gamma}),\quad \ldots$$

we get

$$\mathsf{Q}(v^{\delta^+}(u^{\gamma}),y^{\delta^+}(v^{\delta^+}(u^{\gamma}))),\quad \neg\mathsf{Q}(v^{\delta^+}(u^{\gamma}),u^{\gamma}),\quad \ldots$$

which cannot be reduced to a tautology because $y^{\delta^+}(v^{\delta^+}(u^{\gamma}))$ and u^{γ} cannot be unified.

When we instantiate the variables x^{γ} and u^{γ} in the sequence (2.9.1) in parallel via

(2.9.3) $$\sigma := \{x^{\gamma}{\mapsto}v^{\delta^+},\ u^{\gamma}{\mapsto}y^{\delta^+}\},$$

we have to check whether the newly imposed variable-conditions are consistent with the substitution itself. In particular, a cycle as

$$\begin{tikzcd}
x^{\gamma} \arrow[drr, "R"'] & & v^{\delta^+} \arrow[ll, "\sigma^{-1}"'] \\
u^{\gamma} \arrow[urr, "R"] & & y^{\delta^+} \arrow[ll, "\sigma^{-1}"]
\end{tikzcd}$$

(for the R of (2.9.2)) has to be disallowed by definition.

2.2.1 R-Substitutions

Several binary relations on free variables will be introduced in this and the following sections. The overall idea is that when (x, y) occurs in such a relation this means something like "x is necessarily older than y" or "the value of y depends on or is described in terms of x".

DEFINITION 2.10 (Γ_σ, Δ_σ)
For a substitution σ we define the Γ-*relation* to be

$$\Gamma_\sigma \;:=\; \{\, (z^\gamma, x) \mid x \in \mathrm{dom}(\sigma) \wedge z^\gamma \in \mathcal{V}_\gamma(\sigma(x)) \,\},$$

and the Δ-*relation* to be

$$\Delta_\sigma \;:=\; \{\, (y^\delta, x) \mid x \in \mathrm{dom}(\sigma) \wedge y^\delta \in \mathcal{V}_\delta(\sigma(x)) \,\}.$$

DEFINITION 2.11 (R-Substitution)
Let R be a variable-condition according to Definition 2.7.
σ is an R-*substitution* if σ is a substitution and $R \cup \Gamma_\sigma \cup \Delta_\sigma$ is wellfounded.

Note that, regarding syntax, $(x, z^\gamma) \in R$ is intended to mean that an R-substitution σ must not replace x with a term in which z^γ occurs, because x must have some meaning before z^γ comes into existence. To block this replacement, $(z^\gamma, x) \in \Gamma_\sigma$ must be disallowed, i.e. $R \cup \Gamma_\sigma$ must be wellfounded.

As another example, take from Example 2.9 the variable-condition R of (2.9.2) and the σ of (2.9.3). As explained there, σ must not be an R-substitution because the cycle

$$\begin{tikzcd}
x^\gamma \arrow[rd, "R"'] & v^{\delta+} \arrow[l, "\Delta_\sigma"'] \\
u^\gamma \arrow[ru, "R"] & y^{\delta+} \arrow[l, "\Delta_\sigma"']
\end{tikzcd}$$

contradicts the wellfoundedness of $R \cup \Delta_\sigma$.

Note that in practice w.l.o.g. R, Γ_σ, and Δ_σ can always be chosen to be finite. In this case,

$$R \cup \Gamma_\sigma \cup \Delta_\sigma \text{ is wellfounded iff it is acyclic.}$$

After application of an R-substitution σ, in case of $(x, y^\delta) \in R$, we have to update our variable-condition R to ensure that x is not replaced with a term containing y^δ via a future application of another R-substitution that replaces a free variable say u^γ occurring in $\sigma(x)$ with y^δ. In this case, the transitive closure of the updated variable-condition has to contain (u^γ, y^δ). But we have $u^\gamma\ \Gamma_\sigma\ x\ R\ y^\delta$. This means that $R \cup \Gamma_\sigma$ must be a subset of the updated variable-condition. Besides this, we have to add steps with Δ_σ again.

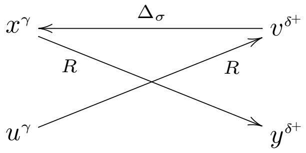
DEFINITION 2.12 (σ-Update)
Let R be a variable-condition and σ be a substitution.
The σ-*update of* R is $R \cup \Gamma_\sigma \cup \Delta_\sigma$.

EXAMPLE 2.13

In the proof attempt of Example 2.9, in a state with variable-condition

$$R = \{(x^{\gamma}, y^{\delta^+}),\ (u^{\gamma}, v^{\delta^+})\},$$

we applied the R-substitution $\sigma' := \{x^{\gamma}\mapsto v^{\delta^+}\}$. Note that $\Delta_{\sigma'} = \{(v^{\delta^+}, x^{\gamma})\}$ and $\Gamma_{\sigma'} = \emptyset$. Thus, the σ'-update R' of R is given by the following finite acyclic graph, which means that R' is wellfounded.



Our treatment of variable-conditions has the following characteristics.

- As explained in Section 2.1.5, the alternative approaches to variable-conditions in the literature restrict the construction of proofs either too much to admit short straightforward proofs, or not enough to guarantee soundness. Our solution, however, is less complicated and provides us with the proper level of restrictiveness.
- The possibility to represent Henkin quantifiers (or K. Jaakko J. Hintikka's IF logic, cf. Hintikka (1996)) was sacrificed for the liberalization of the δ-rule, cf. Section 2.1.5. While it is possible to make the alternative choice,[9] to my knowledge there is no sound approach to variable-conditions that combines the Henkin quantifier with the liberalized δ-rule.
- For efficiency, we never compute transitive closures, but simply keep adding new edges to a graph. The relevant wellfoundedness-checks can then be performed as acyclicity-checks in time which is linear in the number of edges. As any edge must be inspected, this is an optimal time complexity.
- To simplify the definitions, the proofs, and the implementation, we do not permit re-use and permutation of free γ-variables like $\{x^{\gamma}\mapsto u^{\gamma},\ u^{\gamma}\mapsto x^{\gamma}\}$. Indeed, these substitutions have a cyclic Γ-relation and thus are no R-substitutions according to the above definition. Re-use and permutation of free γ-variables are problematic in practice, because we would need an additional time reference to retrieve their solutions (in the sense of Section 1.2.4). Nevertheless, in a sequence of notes[10] we have developed an alternative technical solution that admits re-use and permutation of variables and could be more efficient in practice—even if no variables are re-used.

2.2.2 $(\mathcal{A}, R)$-Valuations

Let $\mathcal{A}$ be some Σ-structure. We now define semantical counterparts of our R-substitutions on V_{γ}, which we will call "$(\mathcal{A}, R)$-valuations".

As an $(\mathcal{A}, R)$-valuation plays the role of a raising function as defined in Section 1.2.3, it does not simply map each free γ-variable directly to an object of $\mathcal{A}$ (of the same type), but may additionally read the values of some free δ-variables under an $\mathcal{A}$-valuation $\delta : \mathrm{V}_{\delta} \to \mathcal{A}$. More precisely, an $(\mathcal{A}, R)$-valuation e takes some restriction of δ as a second argument, say $\delta' : \mathrm{V}_{\delta} \rightsquigarrow \mathcal{A}$ with $\delta' \subseteq \delta$. In short:

$$e : \mathrm{V}_{\gamma} \to (\mathrm{V}_{\delta} \rightsquigarrow \mathcal{A}) \rightsquigarrow \mathcal{A}.$$

Moreover, for each free γ-variable x^γ, we require that the set $\mathrm{dom}(\delta')$ of free δ-variables read by $e(x^\gamma)$ is identical for all δ. This identical set will be denoted with $S_e\langle\!\{x^\gamma\}\!\rangle$ below. Technically, we require that there is some "semantical relation" $S_e \subseteq \mathrm{V}_\delta{\times}\mathrm{V}_\gamma$ such that for all $x^\gamma \in \mathrm{V}_\gamma$:

$$e(x^\gamma) \;:\; (S_e\langle\!\{x^\gamma\}\!\rangle \rightarrow \mathcal{A}) \rightarrow \mathcal{A}.$$

Note that, for each $e : \mathrm{V}_\gamma \rightarrow (\mathrm{V}_\delta \rightsquigarrow \mathcal{A}) \rightsquigarrow \mathcal{A}$, at most one semantical relation exists, namely

$$S_e \;:=\; \{\, (y^\delta, x^\gamma) \mid x^\gamma \in \mathrm{V}_\gamma \wedge y^\delta \in \mathrm{dom}(\bigcup(\mathrm{dom}(e(x^\gamma)))) \,\}.$$

In the following definitions we are slightly more general because we want to apply the terminology not only to free γ-variables but also to free δ^+-variables.

DEFINITION 2.14 (Semantical Relation (S_e))
The *semantical relation for* e is

$$S_e \;:=\; \{\, (y, x) \mid x \in \mathrm{dom}(e) \wedge y \in \mathrm{dom}(\bigcup(\mathrm{dom}(e(x)))) \,\}.$$

e is *semantical* if e is a partial function on V such that for all $x \in \mathrm{dom}(e)$:

$$e(x) : (S_e\langle\!\{x\}\!\rangle \rightarrow \mathcal{A}) \rightarrow \mathcal{A}.$$

DEFINITION 2.15 (($\mathcal{A}, R$)-Valuation)
Let R be a variable-condition and let $\mathcal{A}$ be a Σ-structure.
e is an $(\mathcal{A}, R)$-*valuation* if $e : \mathrm{V}_\gamma \rightarrow (\mathrm{V}_\delta \rightsquigarrow \mathcal{A}) \rightsquigarrow \mathcal{A}$, e is semantical, and $R \cup S_e$ is wellfounded.

Finally, we need the technical means to turn an $(\mathcal{A}, R)$-valuation e together with a valuation δ of the free δ-variables into a valuation $\epsilon(e)(\delta)$ of the free γ-variables:

DEFINITION 2.16 (ϵ)
We define the function

$$\epsilon : \;\; (\mathrm{V} \rightsquigarrow (\mathrm{V} \rightsquigarrow \mathcal{A}) \rightsquigarrow \mathcal{A}) \;\;\rightarrow\;\; (\mathrm{V} \rightsquigarrow \mathcal{A}) \;\;\rightarrow\;\; \mathrm{V} \;\;\rightsquigarrow\;\; \mathcal{A}$$

for $\quad e : \mathrm{V} \rightsquigarrow (\mathrm{V} \rightsquigarrow \mathcal{A}) \rightsquigarrow \mathcal{A}, \qquad \delta : \mathrm{V} \rightsquigarrow \mathcal{A}, \qquad x \in \mathrm{V}$

by $\quad \epsilon(e)(\delta)(x) := e(x)({}_{S_e\langle\!\{x\}\!\rangle}\!\upharpoonleft\!\delta).$

2.2.3 R-Validity

Assuming that validity of formulas is already given as described in Section 2.1.4, we are now going to define a new notion of validity (of sets of sequents) that provides the free γ-variables with an existential semantics. As this new kind of validity depends on a variable-condition R, it is called "R-validity".

DEFINITION 2.17 (R-Validity, K)
Let R be a variable-condition, $\mathcal{A}$ a Σ-structure, and let G be a set of sequents.
G is *R-valid in $\mathcal{A}$* if there is an $(\mathcal{A}, R)$-valuation e such that G is $(e, \mathcal{A})$-valid.
G is $(e, \mathcal{A})$-*valid* if G is $(\delta, e, \mathcal{A})$-valid for all $\delta : \mathrm{V}_\delta \rightarrow \mathcal{A}$.
G is $(\delta, e, \mathcal{A})$-*valid* if G is valid in $\mathcal{A} \uplus \epsilon(e)(\delta) \uplus \delta$.
G is *valid in $\mathcal{A}$* if $\mathit{\Gamma}$ is valid in $\mathcal{A}$ for all $\mathit{\Gamma} \in G$.
A sequent $\mathit{\Gamma}$ is *valid in $\mathcal{A}$* if there is some formula listed in $\mathit{\Gamma}$ that is valid in $\mathcal{A}$.
Validity in a class of Σ-structures is understood as validity in each of the Σ-structures of that class.

If we omit the reference to a special Σ-structure we mean validity in some fixed class K of Σ-structures, e.g. the class of all Σ-structures or the class of Herbrand Σ-structures, cf. Wirth & Gramlich (1994b) for more interesting classes for establishing inductive validity.

EXAMPLE 2.18 (R-Validity)
For $x^\gamma \in \mathrm{V}_\gamma$, $y^\delta \in \mathrm{V}_\delta$, the sequent $x^\gamma{=}y^\delta$ is $\emptyset$-valid in any $\mathcal{A}$ because we can choose $S_e := \mathrm{V}_\delta{\times}\mathrm{V}_\gamma$ and $e(x^\gamma)(\delta) := \delta(y^\delta)$ for $\delta : \mathrm{V}_\delta \to \mathcal{A}$, resulting in

$$\epsilon(e)(\delta)(x^\gamma) = e(x^\gamma)({}_{S_e\langle\{x^\gamma\}\rangle}{\upharpoonleft}\delta) = e(x^\gamma)({}_{\mathrm{V}_\delta}{\upharpoonleft}\delta) = \delta(y^\delta).$$

This means that $\emptyset$-validity of $x^\gamma{=}y^\delta$ is the same as the validity of $\forall y.\ \exists x.\ x{=}y.$ Moreover, note that $\epsilon(e)(\delta)$ has access to the δ-value of y^δ just as a raising function f for x in the raised (i.e. dually Skolemized) version $f(y^\delta){=}y^\delta$ of $\forall y.\ \exists x.\ x{=}y.$

Contrary to this, for $R := \mathrm{V}_\gamma{\times}\mathrm{V}_\delta$, the same formula $x^\gamma{=}y^\delta$ is not R-valid in general because then the required wellfoundedness of $R \cup S_e$ implies $S_e{=}\emptyset$, and the value of x^γ cannot depend on $\delta(y^\delta)$ anymore, due to $e(x^\gamma)({}_{S_e\langle\{x^\gamma\}\rangle}{\upharpoonleft}\delta) = e(x^\gamma)({}_{\emptyset}{\upharpoonleft}\delta) = e(x^\gamma)(\emptyset)$. This means that $(\mathrm{V}_\gamma{\times}\mathrm{V}_\delta)$-validity of $x^\gamma{=}y^\delta$ is the same as the validity of $\exists x.\ \forall y.\ x{=}y.$ Moreover, note that $\epsilon(e)(\delta)$ has no access to the δ-value of y^δ just as a raising function c for x in the raised version $c{=}y^\delta$ of $\exists x.\,\forall y.\,x{=}y.$

For a more general example let $G = \{\ A_{i,0} \ldots A_{i,n_i-1} \mid i{\in}I\ \}$, where for $i \in I$ and $j \prec n_i$ the $A_{i,j}$ are formulas with free γ-variables from $\boldsymbol{e}$ and free δ-variables from $\boldsymbol{u}$. Then $(\mathrm{V}_\gamma{\times}\mathrm{V}_\delta)$-validity of G means

$$\exists \boldsymbol{e}.\ \forall \boldsymbol{u}.\ \forall i{\in}I.\ \exists j \prec n_i.\ A_{i,j}$$

whereas $\emptyset$-validity of G means

$$\forall \boldsymbol{u}.\ \exists \boldsymbol{e}.\ \forall i{\in}I.\ \exists j \prec n_i.\ A_{i,j}$$

2.2.4 Choice-Conditions

Roughly speaking, a set G_0 of sequents *reduces to* a set G_1 of sequents if validity of G_1 implies validity of G_0. This is too weak for our purpose, however, because we are not only interested in validity but also in preserving the solutions for the free γ-variables. As explained in Section 1.2.4, it is important that the solutions of G_1 are also solutions for G_0. Thus, a more appropriate definition would be: G_0 *R-reduces to* G_1 if $(e, \mathcal{A})$-validity of G_1 implies $(e, \mathcal{A})$-validity of G_0 for each $(\mathcal{A}, R)$-valuation e. This definition works well with all inference rules at the end of Section 1.2.2, with the exception of the liberalized δ-rules.

The additional solutions (i.e. R-substitutions on V_γ) resulting from the liberalization of the δ-rule admit additional proofs, which are shorter, more natural, and easier to find. These additional solutions do not impose any difficulty when interest is in validity only, cf. Hähnle & Schmitt (1994). But when the preservation of solutions is required, they pose problems because they may move some free δ^+-variable, say y^{δ^+}, out of its context, namely out of the scope of the quantifier eliminated by y^{δ^+}:

EXAMPLE 2.19 (Reduction & Liberalized δ-Steps)
In Example 2.8 a liberalized δ-step reduces $\quad \forall y.\ \neg\mathsf{P}(y),\quad \mathsf{P}(x^\gamma),\ \ldots$
to $\quad \neg\mathsf{P}(y^{\delta^+}),\quad \mathsf{P}(x^\gamma),\ \ldots$
with the empty variable-condition $R := \emptyset$. The lower sequent is $(e, \mathcal{A})$-valid for the $(\mathcal{A}, R)$-valuation e given by $e(x^\gamma)(\delta) := \delta(y^{\delta^+})$. The upper sequent, however, is not $(e, \mathcal{A})$-valid when $\mathsf{P}^{\mathcal{A}}(a)$ is TRUE and $\mathsf{P}^{\mathcal{A}}(b)$ is FALSE for some a, b from the universe of $\mathcal{A}$. To see this, take some valuation δ with $\delta(y^{\delta^+}) := b$.

How can we solve this problem, i.e. how can we change the notion of reduction such that the liberalized δ-step becomes a reduction step?

The[11] appropriate solution to the problem of the above Example 2.19 is the following: We disallow the value b for $\delta(y^{\delta^+})$ via a *choice-condition* $C(y^{\delta^+})$ that forces us to choose a value for y^{δ^+} such that $\mathsf{P}(y^{\delta^+})$ becomes true—if possible. Technically, this is achieved by setting $C(y^{\delta^+}) := \mathsf{P}(y^{\delta^+})$ and requiring the valuations to fulfill a compatibility condition. In the general case, the choice of a value for y^{δ^+} will depend on the free variables of the formula $C(y^{\delta^+})$. Therefore, we require the inclusion of this dependency into the reflexive & transitive closure of the variable-condition R in the following definition:

DEFINITION 2.20 (Choice-Condition)
C is an *R-choice-condition* if R is a wellfounded variable-condition, C is a partial function from V_{δ^+} into the set of formulas, and $z\ R^*\ y^{\delta^+}$ for all $y^{\delta^+} \in \mathrm{dom}(C)$ and $z \in \mathcal{V}_{\mathrm{free}}(C(y^{\delta^+}))$.

After global application of an R-substitution σ we now have to update both R and C:

DEFINITION 2.21 (Extended σ-Update)
Let C be an R-choice-condition and let σ be a substitution.
The *extended σ-update* (C', R') *of* (C, R) is given by:

$$C' := \{\, (x, B\sigma) \mid (x, B) \in C \wedge x \notin \mathrm{dom}(\sigma) \,\},$$

R' is the σ-update of R, cf. Definition 2.12.

LEMMA 2.22
If C is an R-choice-condition, σ an R-substitution, and if (C', R') is the extended σ-update of (C, R), then C' is an R'-choice-condition.

We now split our valuation $\delta : \mathrm{V}_{\delta} \to \mathcal{A}$; while $\tau : \mathrm{V}_{\delta^-} \to \mathcal{A}$ valuates the free δ^--variables, π valuates the remaining free δ^+-variables. As the choices of π may depend on τ, the technical realization is similar to that of the dependency of the $(\mathcal{A}, R)$-valuations on the free δ-variables, as described in Section 2.2.2.

DEFINITION 2.23 (Compatibility)
Let C be an R-choice-condition, $\mathcal{A}$ a Σ-structure, and e an $(\mathcal{A}, R)$-valuation.
π is $(e, \mathcal{A})$*-compatible with* (C, R) if

1. $\pi : \mathrm{V}_{\delta^+} \to (\mathrm{V}_{\delta^-} \leadsto \mathcal{A}) \leadsto \mathcal{A}$ is semantical (cf. Definition 2.14) and $R \cup S_e \cup S_\pi$ is wellfounded.
2. For all $y^{\delta^+} \in \mathrm{dom}(C)$, for all $\tau : \mathrm{V}_{\delta^-} \to \mathcal{A}$, and for all $\eta : \{y^{\delta^+}\} \to \mathcal{A}$, setting $B := C(y^{\delta^+})$, $\delta := \epsilon(\pi)(\tau) \uplus \tau$, and $\delta' := \eta \uplus {}_{\mathrm{V}\setminus\{y^{\delta^+}\}}{\upharpoonleft}\delta$ (i.e. δ' is the η-variant of δ):

   If B is $(\delta', e, \mathcal{A})$-valid, then B is also $(\delta, e, \mathcal{A})$-valid.

Roughly speaking, Item 1 of this definition says that the flow of information between variables (as expressed in R, e, and π) is acyclic. This is needed for lemma application.

To understand Item 2, let us consider an R-choice-condition $C := \{(y^{\delta^+}, B)\}$, which restricts the value of the single variable y^{δ^+} with the formula B. Then C simply requires that a different choice for the $\epsilon(\pi)(\tau)$-value of y^{δ^+} cannot give rise to the validity of the formula B in $\mathcal{A} \uplus \epsilon(e)(\delta) \uplus \delta$. Or—in other words—that $\epsilon(\pi)(\tau)(y^{\delta^+})$ is chosen such that B becomes valid, whenever such a choice is possible.

This is closely related to Hilbert's ε-operator in the sense that y^{δ^+} is given the value of

$$\varepsilon y.\ (B\{y^{\delta^+}\mapsto y\})$$

for a fresh bound variable y. For a motivational introduction to choice-conditions as an indefinite semantics for Hilbert's ε-terms, cf. Wirth (2002). For the technical treatment cf. Section B.2.

Note that the empty function $\emptyset$ is an R-choice-condition for any wellfounded variable-condition R. Furthermore, any π with $\pi : \mathrm{V}_{\delta^+} \to \{\emptyset\} \to \mathcal{A}$ is $(e, \mathcal{A})$-compatible with $(\emptyset, R)$ due to $S_\pi = \emptyset$. Indeed, a compatible π always exists:

LEMMA 2.24
If C is an R-choice-condition, $\mathcal{A}$ a Σ-structure, and e an $(\mathcal{A}, R)$-valuation, then there is some π that is $(e, \mathcal{A})$-compatible with (C, R).

Just like the variable-condition R, the R-choice-condition C grows during proofs. This kind of extension together with a simple soundness condition plays an important role:

DEFINITION 2.25 (Extension)
(C', R') is an *extension of* (C, R) if C is an R-choice-condition, C' is an R'-choice-condition, $C \subseteq C'$, and $R \subseteq R'$.

LEMMA 2.26
Let (C', R') be an extension of (C, R).
If e is an $(\mathcal{A}, R')$-valuation and π is $(e, \mathcal{A})$-compatible with (C', R'),
then e is also an $(\mathcal{A}, R)$-valuation and π is also $(e, \mathcal{A})$-compatible with (C, R).

2.2.5 (C, R)-Validity

While the notion of R-validity (cf. Definition 2.17) already provides the free γ-variables with an existential semantics, it fails to give the free δ^+-variables the proper semantics according to an R-choice-condition C. This deficiency is overcome in the following notion of "(C, R)-validity", which—roughly speaking—requires the following: For arbitrary values of the free δ^--variables, we must be able to choose values for the free δ^+-variables satisfying C and then arbitrary values for the free γ-variables such that the formula becomes valid. Note that the dependencies of these choices are restricted by R. In a formal top down representation, this reads:

DEFINITION 2.27 ((C, R)-Validity)
Let C be an R-choice-condition, let $\mathcal{A}$ be a Σ-structure, and let G be a set of sequents.
G is *(C, R)-valid in $\mathcal{A}$* if G is $(\pi, e, \mathcal{A})$-valid for some $(\mathcal{A}, R)$-valuation e and some[12] π that is $(e, \mathcal{A})$-compatible with (C, R).
G is *$(\pi, e, \mathcal{A})$-valid* if G is $\big(\ \epsilon(\pi)(\tau) \uplus \tau,\ \ e,\ \ \mathcal{A}\ \big)$-valid for each $\tau : \mathrm{V}_{\delta^-} \to \mathcal{A}$.

Note that the notion of $(\pi, e, \mathcal{A})$-validity with $\mathrm{dom}(\pi) = \mathrm{V}_{\delta^+}$ differs from $(\delta, e, \mathcal{A})$-validity with $\mathrm{dom}(\delta) = \mathrm{V}_\delta$ as given in Definition 2.17. Notice that (C, R)-validity treats the free δ^+-variables properly, whereas R-validity of Definition 2.17 does not. The logical strength of the two cannot be compared easily, but we do not need to know more than the following two lemmas.

LEMMA 2.28 (From R- to (C, R)-Validity)
Let C be an R-choice-condition, $\mathcal{A}$ a Σ-structure, and let G be a set of sequents.
If G is $(\mathrm{V}_\gamma \times \mathrm{V}_\delta)$-valid in $\mathcal{A}$, then G is R-valid and (C, R)-valid in $\mathcal{A}$.

On the other hand, from (C, R)-validity of a set of sequents G we can infer $(\emptyset, R')$-validity and R'-validity for some R' when we rename the free δ^+-variables in G to some new free γ-variables:

LEMMA 2.29 (From (C, R)- to R-Validity)
Let C be an R-choice-condition, $\mathcal{A}$ a Σ-structure, and let G be a set of sequents.
Let $\varsigma : \mathcal{V}_{\delta^+}(G) \to (\mathrm{V}_\gamma \setminus \mathcal{V}(G))$ be injective.
If G is (C, R)-valid in $\mathcal{A}$, then $G\varsigma$ is $(\emptyset, R')$-valid and R'-valid in $\mathcal{A}$ for any R' with

$$R' \quad \subseteq \quad ({}_{\mathrm{V}_{\delta^-} \cup \mathrm{V}_\gamma \setminus \mathrm{ran}(\varsigma)}\!\upharpoonright\!\mathrm{id} \;\uplus\; \varsigma^{-1}) \circ R^+\!\upharpoonright_{\mathrm{V}_{\delta^-} \cup \mathrm{V}_\gamma \setminus \mathrm{ran}(\varsigma)} \;\uplus\; \mathrm{V}_\gamma \times \mathrm{V}_{\delta^+}.$$

2.2.6 Reduction

DEFINITION 2.30 (Reduction)
Let C be an R-choice-condition, $\mathcal{A}$ a Σ-structure, and let G_0 and G_1 be sets of sequents. G_0 (C, R)-*reduces to* G_1 *in* $\mathcal{A}$ if for each $(\mathcal{A}, R)$-valuation e and each π that is $(e, \mathcal{A})$-compatible with (C, R):

if G_1 is $(\pi, e, \mathcal{A})$-valid, then G_0 is $(\pi, e, \mathcal{A})$-valid.

LEMMA 2.31 (Reduction)

Let C be an R-choice-condition; $\mathcal{A}$ a Σ-structure; G_0, G_1, G_2, and G_3 sets of sequents.

1. (Validity)
   If G_0 (C, R)-reduces to G_1 in $\mathcal{A}$ and G_1 is (C, R)-valid in $\mathcal{A}$,
   then G_0 is (C, R)-valid in $\mathcal{A}$, too.
2. (Reflexivity)
   In case of $G_0 \subseteq G_1$: G_0 (C, R)-reduces to G_1 in $\mathcal{A}$.
3. (Transitivity)
   If G_0 (C, R)-reduces to G_1 in $\mathcal{A}$ and G_1 (C, R)-reduces to G_2 in $\mathcal{A}$,
   then G_0 (C, R)-reduces to G_2 in $\mathcal{A}$.
4. (Additivity)
   If G_0 (C, R)-reduces to G_2 in $\mathcal{A}$ and G_1 (C, R)-reduces to G_3 in $\mathcal{A}$,
   then $G_0 \cup G_1$ (C, R)-reduces to $G_2 \cup G_3$ in $\mathcal{A}$.
5. (Monotonicity)
   For (C', R') being an extension of (C, R):
   (a) If G_0 is (C', R')-valid in $\mathcal{A}$, then G_0 is (C, R)-valid in $\mathcal{A}$.
   (b) If G_0 (C, R)-reduces to G_1 in $\mathcal{A}$, then G_0 (C', R')-reduces to G_1 in $\mathcal{A}$.
6. (Instantiation)
   For an R-substitution σ on V_γ and the extended σ-update (C', R') of (C, R):
   (a) If $G_0\sigma$ is (C', R')-valid in $\mathcal{A}$, then G_0 is (C, R)-valid in $\mathcal{A}$.
   (b) If G_0 (C, R)-reduces to G_1 in $\mathcal{A}$, then $G_0\sigma$ (C', R')-reduces to $G_1\sigma$ in $\mathcal{A}$.

2.3 The Inductive Machinery

2.3.1 Weights

Weights control the inductive reasoning cycles. While their syntax is given in the following definition, their semantics will be explained below.

DEFINITION 2.32 (Weight)
A *weight* is a triple $(w, <, \lesssim)$ consisting of the following three terms

Term	Name	In case our language is typed:
w	*weight term*	Let α be the type of w, i.e. $w : \alpha$
$<$	*induction ordering*	$< : \alpha \to \alpha \to \mathsf{bool}$ or $< : \alpha{\times}\alpha$
$\lesssim$	*induction quasi-ordering*	$\lesssim : \alpha \to \alpha \to \mathsf{bool}$ or $\lesssim : \alpha{\times}\alpha$

While we use upper case Greek letters for sequences, we denote our weights with the Hebrew letters ℵ aleph, ℶ beth, and ℸ daleth. While formulas and sequents are sufficient for deductive theorem proving, *weighted sequences* are the basic data structure for the formalization of *descente infinie*:

DEFINITION 2.33 (Weighted Sequent, Seq())
A *weighted sequent* is a pair $(\mathit{\Gamma}, \aleph)$ consisting of a sequent $\mathit{\Gamma}$ and a weight $\aleph$. The function 'Seq' extracts the sequents from a set G of weighted sequents: $\mathrm{Seq}(G) := \mathrm{dom}(G)$. Concrete instances of weighted sequents are written as $\mathit{\Gamma};\ w,\ <,\ \lesssim$ instead of $(\mathit{\Gamma}, (w, <, \lesssim))$.

Initially, the induction ordering $<$ and quasi-ordering $\lesssim$ of the weight $\aleph$ of a weighted sequent $(\mathit{\Gamma}, \aleph)$ should be new free γ-predicate variables $<^{\gamma}$ and $\lesssim^{\gamma}$, respectively. Moreover, the initial weight term of $\aleph$ should be the application $w^{\gamma}(x_0^{\delta^-}, \ldots, x_{n-1}^{\delta^-})$ of a new free γ-variable w^{γ} to the list $x_0^{\delta^-}, \ldots, x_{n-1}^{\delta^-}$ of the free δ^--variables of its sequent $\mathit{\Gamma}$. In our introductory example of Section 1.1.1, the initial sequent was (1) and the weight term was $w(x)$. In our notation here this is written as the weighted sequent

$$\mathsf{0} + x^{\delta^-} = x^{\delta^-};\ \ w^{\gamma}(x^{\delta^-}),\ <^{\gamma},\ \lesssim^{\gamma} \tag{1}$$

and within the proof we apply the R-substitution $\{\ w^{\gamma} \mapsto \lambda x.\ x,\quad <^{\gamma} \mapsto\ \prec\ \}$.

Notice that, although the terms of the induction ordering and quasi-ordering of a weight of a weighted sequent may be (free γ-) predicate variables or λ-terms, the sequents themselves can be restricted to first order because the weights have to interact with the sequents only after they have been instantiated and applied ($\lambda\beta$-reduced), just as in our introductory example.

Furthermore, note that the definition of a weight could be simplified by requiring $\lesssim$ to be a wellfounded quasi-ordering and $<$ to be its ordering. However, for proof-technical convenience and for reasoning on the induction ordering itself, we prefer weaker requirements.

> For example, if we want to prove formally that wellfoundedness of a—possibly non-transitive—relation R implies termination of the transitive closure of its reverse, it should be possible to set $<^{\gamma}$ and $\lesssim^{\gamma}$ to terms denoting R and the empty relation, respectively.

So we decided to have no requirements on the two terms $<$ and $\lesssim$ of a weight $(w, <, \lesssim)$ at all (besides on their types in case of a typed language), but instead we introduce the minimal set of necessary requirements (such as wellfoundedness) on $<$ and $\lesssim$ when counterexamples are compared, cf. Definition 2.35.

Moreover, notice that, although the term $\lesssim$ of the induction quasi-ordering is not visible in the example proof, it may be non-trivial and necessary for simplification in other proofs.[13] Furthermore, even the term $<$ of the induction ordering is not always needed: With very few exceptions,[14] inductive theorem proving systems admit only a single built-in wellfounded induction ordering. In this case, the only part of a weight that has to be implemented is the weight term, and we indeed omitted the induction ordering and quasi-ordering in the implementation of the QUODLIBET system, cf. Section 3.2.1. Nevertheless, to cover all cases, the *general* concept of a weight has to include both $<$ and $\lesssim$.

2.3.2 Counterexamples

The weight of an induction hypothesis $(\mathit{\Delta}, \beth)$ must be smaller than the weight of the goal $(\mathit{\Gamma}, \aleph)$, and for *powerful* inductive theorem proving, we have to be able to restrict this test to the special case semantically described by the sequence $\mathit{\Gamma}$. This can be achieved by considering only such instances of $\aleph$ and $\beth$ that invalidate $\mathit{\Gamma}$. A weighted sequent (cf. Definition 2.33) augmented with such a valuation providing extra information on the invalidity of its sequent in some Σ-structure $\mathcal{A}$ is our formal means to capture the notion of "counterexample".

DEFINITION 2.34 (Counterexample)
Let $\mathcal{A}$ be a Σ-structure from K, let C be an R-choice-condition and e be an $(\mathcal{A}, R)$-valuation, and finally let π be $(e, \mathcal{A})$-compatible with (C, R).
(S, τ) is an $(\pi, e, \mathcal{A})$*-counterexample (for* S*)* if S is a weighted sequent, $\tau : \mathrm{V}_{\delta^-} \to \mathcal{A}$, and $\mathrm{Seq}(\{S\})$ is not $\big(\ \epsilon(\pi)(\tau) \uplus \tau,\ \ e,\ \ \mathcal{A}\ \big)$-valid, cf. Definition 2.17.

Thus, for a weighted sequent $(\mathit{\Gamma}, \aleph)$, the sequent $\mathit{\Gamma}$ is $(\pi, e, \mathcal{A})$-valid (cf. Definition 2.27) iff $(\mathit{\Gamma}, \aleph)$ has no $(\pi, e, \mathcal{A})$-counterexamples.

DEFINITION 2.35 (Ordering on Counterexamples)
Let $\mathcal{A}$ be a Σ-structure from K, let C be an R-choice-condition and e be an $(\mathcal{A}, R)$-valuation, and finally let π be $(e, \mathcal{A})$-compatible with (C, R).
Let (S_0, τ_0) and (S_1, τ_1) be $(\pi, e, \mathcal{A})$-counterexamples. Then, for $i \in \{0, 1\}$, their weighted sequents are of the form $S_i = (\mathit{\Gamma}_i, (w_i, <_i, \lesssim_i))$ and we set $\delta_i := \epsilon(\pi)(\tau_i) \uplus \tau_i$, $\mathcal{B}_i := \mathcal{A} \uplus \epsilon(e)(\delta_i) \uplus \delta_i$, $\bar{w}_i := \mathrm{eval}(\mathcal{B}_i)(w_i)$, $\lhd_i := \mathrm{eval}(\mathcal{B}_i)(<_i)$, and $\underset{\sim}{\lhd}_i := \mathrm{eval}(\mathcal{B}_i)(\lesssim_i)$.
As the following two notions hold only for the case that $\lhd_0 = \lhd_1$ and $\underset{\sim}{\lhd}_0 = \underset{\sim}{\lhd}_1$, we write $\lhd$ for $\underset{\sim}{\lhd}_0$ and $\underset{\sim}{\lhd}_1$ as well as $\underset{\sim}{\lhd}$ for $\underset{\sim}{\lhd}_0$ and $\underset{\sim}{\lhd}_1$:
(S_1, τ_1) *is* $(\pi, e, \mathcal{A})$*-smaller than* (S_0, τ_0) if $\bar{w}_1\ (\underset{\sim}{\lhd} \cup \lhd)^*\ \bar{w}_0$.
(S_1, τ_1) *is strictly* $(\pi, e, \mathcal{A})$*-smaller than* (S_0, τ_0) if $\bar{w}_1 \lhd^+ \bar{w}_0$, $\lhd \circ \underset{\sim}{\lhd} \subseteq \lhd^+$, and $\lhd$ is wellfounded.

Note that in case of "$<_i$" and "$\lesssim_i$" being no proper terms of our (possibly first-order) logic language, "$\mathrm{eval}(\mathcal{B}_i)(<_i)$" is to be taken a shorthand for

$$\big\{\ (a, b)\ \big|\ \mathrm{eval}\big(\ \mathcal{B}_i \uplus \{x{\mapsto}a,\ y{\mapsto}b\}\ \big)\big(\ x <_i y\ \big) = \mathsf{TRUE}\ \big\},$$

for two new distinct variables $x, y \in \mathrm{V}_{\mathrm{bound}} \setminus \mathcal{V}(<_i)$.

Moreover, note that our induction ordering is semantical in the sense that it does not depend on the syntactical term structure of a weight w, but only on the value of w under the evaluation function, cf. Definition 13.7 of Wirth (1997). In Wirth (1997) we have investigated the price one has to pay for the possibility to have induction orderings also depending on the syntax of weights. For powerful concrete inference systems this price turned out to be surprisingly high. Besides this, after improving the ordering information in *descente infinie* by our introduction of explicit weights (cf. Wirth & Becker (1995)), contrary to Bachmair (1988) we no longer feel the need for sophisticated induction orderings that exploit the term structure.

2.3.3 Groundedness

The notion of *groundedness*[15] is for induction as crucial as the notion of reduction is for deduction.

Groundedness is defined in terms of counterexamples, according to the somehow negative argumentation of the *Method of Descente Infinie* as presented in Definition 1.1. Nevertheless, it captures the positive view on *descente infinie* via application of induction hypotheses.

The notion of groundedness as given by Definition 2.36 is sufficiently general to cover the practical and technical requirements of a variety of application domains and inference systems. It also bridges the gap to the simple and clear abstract view on induction given in Lemma 2.37, which abstracts the algebraic structure we need in the following sections from the concrete representation in this section.

For the benefit of the reader's intuition of groundedness, consider the metaphor of building a supporting frame in a swamp.

Note that in the following 'H' stands for the induction hypotheses, 'G_1' for the sub-goals of the goals 'G_0', and 'L' for the lemmas of the proof.

We can fix a construction element G_0 to a construction element (G_1, L) on the same or lower level of the supporting frame resulting in the construction

$$G_0 \rightarrow (G_1, L)$$

In the world of induction this means that if an element of G_0 has a counterexample, then there is a counterexample for an element of G_1 or L. Moreover, if this counterexample is from G_1, then it has to be smaller or equal in the induction quasi-ordering $\lesssim$ that must be identical for both counterexamples.

We can fix a construction element G_0 partly to a construction element (G_1, L) on the same or lower level and partly to a construction element H on a strictly lower level of the supporting frame resulting in the construction

$$\begin{tikzcd}
G_0 \arrow[r] \arrow[d] & (G_1, L) \\
H &
\end{tikzcd}$$

for which we write $G_0 \mathrel{\downarrow\!\!\rightarrow} (H, G_1, L)$. In the world of induction this means that if an element of G_0 has a counterexample, then there is a counterexample for an element of H, G_1, or L. Moreover, if this counterexample is from H then it has to be strictly smaller and if it is from G_1 it has to be equal or smaller than the original counterexample from G_0 in the induction ordering they share.

Now, if we have a supporting frame of the form $H \mathrel{\downarrow\!\!\rightarrow} (H, G_1, L)$, i.e.

$$\begin{tikzcd}
H \arrow[r] \arrow[d] & (G_1, L) \\
H \arrow[r] \arrow[d] & (G_1, L) \\
\vdots &
\end{tikzcd}$$

and we know that the swamp is wellfounded (i.e. we find solid ground eventually if we only go deep enough) then we know that H is sufficiently supported—and hence will not sink—by the element (G_1, L) alone, i.e. $H \rightarrow (G_1, L)$. In the world of induction this means that all sequents of the elements of the set H are inductively valid provided that the base cases in G_1 and the lemmas in L are, cf. Lemma 2.37(7).

Note that $\{S\} \mathrel{\downarrow\!\!\rightarrow} (H, \emptyset, \emptyset) \ \vee \ \{S\} \rightarrow (G_1, L)$ implies $\{S\} \mathrel{\downarrow\!\!\rightarrow} (H, G_1, L)$ for a weighted sequent S, but the converse does not hold in general, because different counterexamples for S may have smaller counterexamples in different sets.

DEFINITION 2.36 (Groundedness)
Let C be an R-choice-condition. Let G_0, G_1, H, L be sets of weighted sequents.
G_0 is (C, R)-*grounded* on (H, G_1, L) (denoted by $G_0 \mapsto_{C,R} (H, G_1, L)$) if for any Σ-structure $\mathcal{A}$ from K, for any $(\mathcal{A}, R)$-valuation e, for any π that is $(e, \mathcal{A})$-compatible with (C, R), and for any $(\pi, e, \mathcal{A})$-counterexample (S_0, τ_0) with $S_0 \in G_0$, there is an $(\pi, e, \mathcal{A})$-counterexample (S_1, τ_1) satisfying one of the following cases:

Induction Hypothesis: $S_1 \in H$ and (S_1,τ_1) is strictly $(\pi,e,\mathcal{A})$-smaller than (S_0,τ_0).

Sub-Goal: $S_1 \in G_1$ and (S_1,τ_1) is $(\pi, e, \mathcal{A})$-smaller than (S_0,τ_0).

Lemma: $S_1 \in L$.

Finally, we write $G_0 \rightarrow_{C,R} (G_1, L)$ as a shorthand for $G_0 \mapsto_{C,R} (\emptyset, G_1, L)$.

Note that $H \rightarrow_{C,R} (\emptyset, L)$ iff $\mathrm{Seq}(H)$ (C, R)-reduces to $\mathrm{Seq}(L)$ in all $\mathcal{A} \in$ K.

Finally, note that Section 2.4 depends only on the general properties of groundedness given in the following lemma. It is is similar to Lemma 2.31, but it extends reduction to groundedness.

LEMMA 2.37 (Groundedness)
Let C be an R-choice-condition, and let G_i, G'_i, H_i, L_i be sets of weighted sequents.

1. (Validity)
   Assume $G_0 \rightarrow_{C,R} (G_1, L_1)$. Let $\mathcal{A} \in$ K.
   (a) If $\mathrm{Seq}(G_1 \cup L_1)$ is (C, R)-valid in $\mathcal{A}$, then $\mathrm{Seq}(G_0)$ is (C, R)-valid in $\mathcal{A}$, too.
   (b) Let e be an $(\mathcal{A}, R)$-valuation and let π be $(e, \mathcal{A})$-compatible with (C, R).
   If $\mathrm{Seq}(G_1 \cup L_1)$ is $(\pi, e, \mathcal{A})$-valid, then $\mathrm{Seq}(G_0)$ is $(\pi, e, \mathcal{A})$-valid, too.
2. (Reflexivity)
   In case of $G_0 \subseteq G_1 \cup L_1$: $G_0 \mapsto_{C,R} (H_1, G_1, L_1)$.
3. (Transitivity)
   (a) If $G_0 \rightarrow_{C,R} (G_1, L_1)$ and $G_1 \mapsto_{C,R} (H_2, G_2, L_2)$,
   then $G_0 \mapsto_{C,R} (H_2, G_2, L_1 \cup L_2)$.
   (b) If $G_0 \rightarrow_{C,R} (G_1, L_1)$ and $L_1 \rightarrow_{C,R} (G_2, L_2)$, then $G_0 \rightarrow_{C,R} (G_1, G_2 \cup L_2)$.
4. (Additivity)
   If $G_i \mapsto_{C,R} (H_i, G'_i, L_i)$ for all $i \in I$,
   then $\bigcup_{i \in I} G_i \mapsto_{C,R} \left(\bigcup_{i \in I} H_i \,,\, \bigcup_{i \in I} G'_i \,,\, \bigcup_{i \in I} L_i \right)$.
5. (Monotonicity)
   For (C', R') being an extension of (C, R):
   If $G_0 \rightarrow_{C,R} (G_1, L_1)$, then $G_0 \rightarrow_{C',R'} (G_1, L_1)$.
6. (Instantiation)
   For an R-substitution σ on V_γ and the extended σ-update (C', R') of (C, R):
   If $G_0 \rightarrow_{C,R} (G_1, L_1)$, then $G_0\sigma \rightarrow_{C',R'} (G_1\sigma, L_1\sigma)$.
7. (Descente Infinie)
   If $H_1 \mapsto_{C,R} (H_1, G_1, L_1)$, then $H_1 \rightarrow_{C,R} (G_1, L_1)$.

2.4 Abstract Sequent and Tableau Calculus

Now we are going to describe an abstract sequent and tableau calculus for *descente infinie*. The standard state-of-the-art deductive calculi are instances of this calculus, and its design is not as *ad hoc* as it may seem, cf. Wirth & Becker (1995), Wirth (1997) for a discussion of alternatives. The benefit of an *abstract* calculus is that each instance is automatically sound. For the design of purely deductive calculi, such an abstract calculus is not really helpful because their soundness is a local property of each inference rule. For *descente infinie*, however, soundness becomes a *global* problem of the whole inference system. Moreover, the inference rules usually have to be improved over a long period of practical testing until they meet the design goals of Section 1.2.1. And in this setting, such an abstract calculus turned out to be very useful indeed, cf. Wirth (1997).

DEFINITION 2.38 ($\mathcal{AX}$)
The set $\mathcal{AX}$ of *axioms* may be any set of sequents that is $(\mathrm{V}_\gamma{\times}\mathrm{V}_\delta)$-valid in all $\mathcal{A} \in \mathrm{K}$.

By Lemma 2.28, this means that $\mathcal{AX}$ is R-valid and (C,R)-valid for any R-choice-condition C. For K cf. the last sentence in Definition 2.17. Typically, $\mathcal{AX}$ contains all sequents of the forms $\mathit{\Gamma}\ A\ \mathit{\Pi}\ \overline{A}\ \mathit{\Lambda}$ and $\mathit{\Gamma}\ (s{=}s)\ \mathit{\Pi}$ for sequents $\mathit{\Gamma}$, $\mathit{\Pi}$, $\mathit{\Lambda}$, formulas A, and terms s.

In *inductive* proof trees, each sequent has a weight which controls the inductive loops, i.e. the sequents of deductive calculi are replaced with weighted sequents. Inductive *tableau* trees differ from deductive tableau trees in that each root is labeled with a weight instead of a formula.

Both sequent and tableau trees will be used in the following. In the definitions we describe the sequent version and add the alternative text for the tableau version enclosed in double parenthesis, as in the following definition.

DEFINITION 2.39 (Proof Forest)
An inductive *proof forest* in a sequent and tableau calculus is a quintuple

$$(F, C, R, L, H)$$

where C is an R-choice-condition, $L, H \subseteq \mathbf{N}_+{\times}\mathbf{N}_+$, and F is a partial function from $\mathbf{N}_+$ into the set of pairs (S,t), where S is a weighted sequent and t is a tree whose nodes are labeled with weighted sequents ((*t is a tree whose root is labeled with a weight and whose other nodes are labeled with formulas*)).

Here L records the lemma applications and H the induction-hypothesis applications, and the tree t represents a proof attempt for the proposition S. In case of a *tableau* tree, the nodes of t are labeled with formulas; the root, however, with a weight. In case of a *sequent* tree, all nodes are labeled with weighted sequents.

While the weighted sequents at the leaves of a *sequent* tree represent its goals, in a *tableau* tree we have to collect all ancestors to make up a weighted sequent, and—moreover—the labeling formulas are in negated form:

DEFINITION 2.40 (Goals(), Closedness)
Let T be a set of trees. '$\mathrm{Goals}(T)$' denotes the set of weighted sequents labeling the leaves of the trees in T ((*the set of weighted sequents $(\mathit{\Delta}, \sqsupset)$ where $\mathit{\Delta}$ results from listing the conjugates of the formulas labeling a branch from a leaf to the root (exclusively) in a tree t in T and $\sqsupset$ is the label of the root of the tree t*)).
A tree t is *closed* if $\mathrm{Seq}(\mathrm{Goals}(\{t\})) \subseteq \mathcal{AX}$.

What is the conceptual reason for a forest instead of a single proof tree? We want to separate lemma and induction-hypothesis application from the standard reductive proof steps. This has already been explained in detail in Section 1.2.2. In our formalization, lemma and induction-hypothesis application now look as follows:

If we have two proof trees $F(i) = ((\mathit{\Gamma}, \aleph), t)$ and $F(i') = ((\mathit{\Gamma}', \aleph'), t')$, we can apply $\mathit{\Gamma}'$ as a lemma in the tree t of $(\mathit{\Gamma}, \aleph)$ and record this lemma application by inserting (i', i) into L. We can also apply $(\mathit{\Gamma}', \aleph')$ instantiated with a substitution ϱ on V_{δ^-} as an induction hypothesis in the tree t of $(\mathit{\Gamma}, \aleph)$ and record this induction hypothesis application by inserting (i', i) into H. In the latter case we additionally have to implant a new branch into t whose goals express that the weight term of $\aleph'\varrho$ is strictly smaller than the weight term of $\aleph$ and that the induction (quasi-) orderings of $\aleph'\varrho$ and $\aleph$ are identical.

Provided that the lemma-application relation $L{\circ}H^*$ is wellfounded and all trees t'' with $F(i'') = (S'', t'')$ and $i''\ (L{\cup}H)^*\ i$ are closed, this proves that $\mathit{\Gamma}$ is (C, R)-valid.

The following definition introduces the abstract and mnemonic '$\mathrm{Propos}()$' and '$\mathrm{Trees}()$' for the '$\mathrm{dom}()$' and '$\mathrm{ran}()$' of our concrete representation.

DEFINITION 2.41 ($\mathrm{Propos}()$, $\mathrm{Trees}()$)
For A being a set of pairs (S, t) consisting of a weighted sequent S and a tree t, we define the propositions of A by $\mathrm{Propos}(A) := \mathrm{dom}(A)$ and the trees of A by $\mathrm{Trees}(A) := \mathrm{ran}(A)$.

The following definition is based on three abstract proof steps: an *Instantiation* step globally instantiates some free variables in the proof forest; a *Hypothesizing* step starts a new proof tree for a newly conjectured proposition; and an *Expansion* step expands a proof tree.

DEFINITION 2.42 (Abstract Sequent and Tableau Calculus)
We start with the empty proof forest $(F, C, R, L, H) := (\emptyset, \emptyset, \emptyset, \emptyset, \emptyset)$ and then iterate the following modifications of (F, C, R, L, H), resulting in (F', C', R', L', H'):

Instantiation: Let σ be an R-substitution on V_{γ}. Let (C', R') be the extended σ-update of (C, R).
Set $L' := L$, $H' := H$, and

$$F' \ := \ \{ \ (\ i,\ ((\mathit{\Gamma}\sigma, \aleph\sigma), t\sigma)\ ) \ \big|\ (\ i,\ ((\mathit{\Gamma}, \aleph), t)\ ) \in F \ \}.$$

Hypothesizing: Let $i \in \mathbf{N}_+ \setminus \mathrm{dom}(F)$. Let $(\mathit{\Gamma}, \aleph)$ be a weighted sequent.
Let t be a new tree with a single node, and label this node with $(\mathit{\Gamma}, \aleph)$.
⦅*Let t be a new tree with a single branch, such that $\mathit{\Gamma}$ is the list of the conjugates of the formulas labeling the branch from the leaf to the root (exclusively) and $\aleph$ is the label of the root.*⦆
Let (C', R') be an extension of (C, R).
Set $L' := L$, $H' := H$, and $F' := F \cup \{ \ (\ i,\ ((\mathit{\Gamma}, \aleph), t)\ )\ \}$.

Expansion: Let $(i, (S, t)) \in F$, let l be a leaf in t, let $(\mathit{\Delta}, \beth)$ be the label of l
⦅*let $(\mathit{\Delta}, \beth)$ result from listing the conjugates of the formulas labeling the branch from l to the root (exclusively) and let $\beth$ be the label of the root of t*⦆.
Let G be a set of weighted sequents
⦅*let M be a set of sequents and set $G := \{\ (\mathit{\Pi}\mathit{\Delta}, \beth)\ |\ \mathit{\Pi} \in M\ \}$*⦆.
Let (C', R') be an extension of (C, R), and let $N_{\mathrm{L}}, N_{\mathrm{H}} \subseteq \mathrm{dom}(F)$, such that

$$\{(\mathit{\Delta}, \beth)\} \mapsto_{C', R'} \big(\ \mathrm{Propos}(\langle N_{\mathrm{H}} \rangle F),\ G,\ \mathrm{Propos}(\langle N_{\mathrm{L}} \rangle F)\ \big). \qquad (\$)$$

Set $L' := L \cup N_{\mathrm{L}}{\times}\{i\}$, $H' := H \cup N_{\mathrm{H}}{\times}\{i\}$, and

$$F' := \left(\ F\backslash\{(i,(S,t))\}\ \right) \cup \{(i,(S,t'))\},$$

where t' results from t by adding, for each weighted sequent S' in G, a new child node labeled with S' to the former leaf l *((by adding, for each sequent Π in M, a new child branch to the former leaf l, such that Π is the list of the conjugates of the formulas labeling the branch from the leaf to the new child node of l)).*

Expansion steps are parameterized with a goal $(\varDelta,\sqsupset)$, with two sets N_{H}, N_{L} of numbers of proof trees, and with a set of sequents G such that ($) holds. $\mathrm{Propos}(\langle N_{\mathrm{H}}\rangle F)$ and $\mathrm{Propos}(\langle N_{\mathrm{L}}\rangle F)$ contain the propositions of the proof trees that are applied as induction hypotheses and lemmas, respectively. For the $\langle\ldots\rangle F$ notation cf. Section 2.1.1. The weighted sequents in G become the new child nodes of the former leaf node labeled with $(\varDelta,\sqsupset)$. For *tableau* trees, however, this set G of weighted sequents must actually have the form of $\{\ (\Pi\varDelta,\sqsupset) \mid \Pi \in M\ \}$, because an Expansion step cannot remove formulas from ancestor nodes (as they are also part of the goals associated with other leaves in the proof tree). To be precise, in addition to the standard notion of a *tree* (cf. Knuth (1997 f.), Vol. I), we assume an explicit representation of leaves, so that, when we add the elements of G as children to the leaf node l, this l is no longer a leaf, even if G is empty. Finally note that an Instantiation step can actually apply a substitution even on $\mathrm{V}_{\gamma} \cup \mathrm{V}_{\delta^+}$, cf. Section B.3.

2.4.1 Soundness

The following invariant captures the soundness of our proof trees. Roughly speaking, the validity of the goals of a tree imply the validity of the sequent of this tree; i.e.: "The leaves imply the root."

DEFINITION 2.43 (Invariant for Soundness)
The *invariant for soundness of* (F,C,R,L,H) is that (F,C,R,L,H) is a proof forest and that, for all $(i,(S,t)) \in F$,

$$\{S\} \rightarrow_{C,R} \left(\ \mathrm{Goals}(\mathrm{Trees}(\langle I\rangle F)),\ \ \mathrm{Propos}(\langle L\langle I\rangle\rangle F)\ \right) \text{ for } I := H^*\langle\!\langle\{i\}\rangle\!\rangle.$$

Note that I is the set of the number i plus the numbers of the proof trees whose propositions have been applied in the tree t as induction hypotheses. $\mathrm{Goals}(\mathrm{Trees}(\langle I\rangle F))$ is the set of goals of these proof trees. Moreover,

$$\mathrm{Propos}(\langle L\langle I\rangle\rangle F)\ =\ \{\ S' \mid i \in I \wedge i' L i \wedge F(i') = (S',t')\ \}$$

is the set of the lemmas S depends on.

THEOREM 2.44 (Soundness)
The invariant for soundness is always satisfied for the abstract sequent and tableau calculus.

THEOREM 2.45 (Successful Proof)
Suppose the invariant for soundness of (F,C,R,L,H) holds. Let $(i,((\varGamma,\aleph),t)) \in F$.
If all trees in $\mathrm{Trees}(\langle(L{\cup}H)^*\langle\!\langle\{i\}\rangle\!\rangle\rangle F)$ (i.e. in $\{\ t' \mid i'\ (L{\cup}H)^*\ i\ \wedge\ F(i') = (S',t')\ \}$)
are closed and if $L{\circ}H^*$ is wellfounded, then $\varGamma$ is (C,R)-valid.

Notice that (C,R)-validity of $\varGamma$ implies $(\emptyset,R')$- and R'-validity of $\varGamma\varsigma$, for R' and ς satisfying the requirements of Lemma 2.29.

2.4.2 Safeness

While the invariant for soundness ("the leaves imply the root") is essential, its converse, namely "the root implies the leaves", which we call *safeness*, is useful in practice for failure detection.

Failure detection is especially important for inductive theorem proving as the standard technique to generalize (i.e. to strengthen) induction hypotheses easily leads to *over-generalization*. As a valid input theorem easily produces an invalid sub-goal by over-generalization, the early and localized detection of this invalidity is of major practical importance, cf. also Section 3.2.3.

DEFINITION 2.46 (Invariant for Safeness)
The *invariant for safeness of* (F, C, R, L, H) is that, for all $(i, ((\mathit{\Gamma}, \aleph), t)) \in F$,

$$\mathrm{Seq}(\mathrm{Goals}(\{t\})) \quad (C, R)\text{-reduces to} \quad \{\mathit{\Gamma}\}.$$

We extend Definition 2.42 of the abstract sequent and tableau calculus as follows:

DEFINITION 2.47 (Safeness of Steps and Sub-rules)
Instantiation^{16} and Hypothesizing steps are always *safe*. Also Expansion steps in a *tableau* tree are always *safe*. An Expansion step in a *sequent* tree is *safe* if $\mathrm{Seq}(G)$ (C', R')-reduces to $\{\mathit{\Delta}\}$. A sub-rule of the Expansion rule is *safe* if it describes only safe Expansion steps.

THEOREM 2.48 (Safeness)
The invariant for safeness is always satisfied for the abstract sequent and tableau calculus, provided the individual steps are safe.

Suppose we have disproved a goal of a tree t with $(i, ((\mathit{\Gamma}, \aleph), t)) \in F$, i.e. we have found out that the goal is invalid. In this case we should backtrack to a possibly unsafe step that may have caused this invalidity. If, however, all steps in t are safe, then the proposition $\mathit{\Gamma}$ is invalid. This may have two reasons: Either a Hypothesizing step introduced an invalid proposition, or the proposition was modified later by an invalidating Instantiation step:

- If there have been no Instantiation steps affecting the sequent $\mathit{\Gamma}$, then we should remove $(i, ((\mathit{\Gamma}, \aleph), t))$ from the proof forest F and undo all its applications as a lemma or as an induction hypothesis, i.e. the Expansion steps where i occurs in the sets N_{L}, N_{H}.

- Otherwise, we should undo an Instantiation step affecting the sequent $\mathit{\Gamma}$, and then see whether we can still detect a failure by disproving the disinstantiated goal.

2.5 Concrete Sequent and Tableau Calculus

The concrete sequent and tableau calculus described in this section results from the abstract sequent and tableau calculus of the previous section by presenting concrete sub-rules of the Expansion rule.

2.5.1 Expansion Steps Within a Single Tree

The α-, β-, γ-, δ-rules as well as the liberalized δ-, Rewrite-, and Cut-rules at the end of Section 1.2.2 can be modeled as safe Expansion steps as follows:
Let $\mathcal{F} = (F, C, R, L, H)$. Let

$$\frac{\varDelta}{\varPi_0 \quad \ldots \quad \varPi_{n-1}} \begin{array}{l} C'' \\ R'' \end{array}$$

denote a sub-rule of the Expansion rule in sequent trees of the abstract sequent and tableau calculus of Definition 2.42 where $N_{\mathrm{L}} := N_{\mathrm{H}} := \emptyset$ (i.e. no application of lemmas or induction hypotheses), $G := \{(\varPi_0, \sqsupset), \ldots, (\varPi_{n-1}, \sqsupset)\}$, $C' := C \cup C''$, and $R' := R \cup R''$. If C'' and R'' are not explicitly denoted, this stands for the special case of $C'' = R'' = \emptyset$.

The respective rules for *tableau* trees differ only in that M consists of the sub-sequents containing the new (i.e. the first one or two) formulas of the sequents below the bar.

For such a rule being a safe sub-rule of the Expansion rule of the abstract sequent and tableau calculus of Definition 2.42 we have to show that C' is an R'-choice-condition, that $\{(\varDelta, \sqsupset)\} \rightarrow_{C',R'} (G, \emptyset)$, and that $\mathrm{Seq}(G)$ (C', R')-reduces to $\{\varDelta\}$.

THEOREM 2.49
The α-, β-, γ-, δ-rules as well as the liberalized δ-, Rewrite-, and Cut-rules at the end of Section 1.2.2 are safe sub-rules of the Expansion rule of the abstract sequent and tableau calculus.

The following example shows that R'' of the liberalized δ-rule at the end of Section 1.2.2 must indeed contain $\mathcal{V}_{\delta}(A) \times \{x^{\delta^+}\}$ besides $\mathcal{V}_{\gamma}(A) \times \{x^{\delta^+}\}$, and that the transitive closure over R' must be considered for an R'-substitution on V_{γ}.

EXAMPLE 2.50
The formula $\exists y.\ \forall x.\ \big(\ \forall z.\ \mathsf{Q}(x,z)\ \vee\ \neg\mathsf{Q}(x,y)\ \big)$
is not generally valid (to wit, let Q be the identity relation on a non-trivial universe).
γ-step: $\forall x.\ \big(\ \forall z.\ \mathsf{Q}(x,z)\ \vee\ \neg\mathsf{Q}(x,y^{\gamma})\ \big),\quad \exists y.\ \forall x.\ \big(\ \forall z.\ \mathsf{Q}(x,z)\ \vee\ \neg\mathsf{Q}(x,y)\ \big)$
Liberalized or non-liberalized δ-step:
$\big(\ \forall z.\ \mathsf{Q}(x^{\delta},z)\ \vee\ \neg\mathsf{Q}(x^{\delta},y^{\gamma})\ \big),\quad \exists y.\ \forall x.\ \big(\ \forall z.\ \mathsf{Q}(x,z)\ \vee\ \neg\mathsf{Q}(x,y)\ \big)$
with variable-condition $R := \{(y^{\gamma}, x^{\delta})\}$.
α-step: $\forall z.\ \mathsf{Q}(x^{\delta},z),\quad \neg\mathsf{Q}(x^{\delta},y^{\gamma}),\quad \exists y.\ \forall x.\ \big(\ \forall z.\ \mathsf{Q}(x,z)\ \vee\ \neg\mathsf{Q}(x,y)\ \big)$
Liberalized δ-step: $\mathsf{Q}(x^{\delta},z^{\delta^+}),\quad \neg\mathsf{Q}(x^{\delta},y^{\gamma}),\quad \exists y.\ \forall x.\ \big(\ \forall z.\ \mathsf{Q}(x,z)\ \vee\ \neg\mathsf{Q}(x,y)\ \big)$
with additional choice-condition $C'' := \{(z^{\delta^+}, \neg\mathsf{Q}(x^{\delta}, z^{\delta^+}))\}$ and additional variable-condition $R'' := \{(x^{\delta}, z^{\delta^+})\}$, i.e. the current variable-condition R' is given by

$$y^{\gamma} \xrightarrow[R]{} x^{\delta} \xrightarrow[R'']{} z^{\delta^+}$$

Note that now we have $y^{\gamma}\ R'^{+}\ z^{\delta^+}$ although y^{γ} does not appear in $\mathsf{Q}(x^{\delta}, z)$.

Thus, both the inclusion of the free δ-variables of the principle formula into the domain of the variable-condition and its transitive closure are necessary for $\sigma := \{y^{\gamma}{\mapsto}z^{\delta^+}\}$ not being an R'-substitution in our state of proof. The latter fact is, however, essential for soundness, because without it we could complete the proof attempt by application of σ in an Instantiation step, leading to the tautology

$$\mathsf{Q}(x^{\delta}, z^{\delta^+}),\quad \neg\mathsf{Q}(x^{\delta}, z^{\delta^+}),\quad \exists y.\ \forall x.\ \big(\ \forall z.\ \mathsf{Q}(x, z)\ \vee\ \neg\mathsf{Q}(x, y)\ \big)$$

2.5.2 Applying Lemmas and Induction Hypotheses

Now we present two rules for applying $(\varPhi, \daleth)$ as a lemma or as an induction hypothesis to expand a goal $(\varDelta, \beth)$ of a proof tree t. We formulate them as Expansion steps in tableau trees (sequent trees analogously) of the abstract sequent and tableau calculus of Definition 2.42 as follows.

Let (F, C, R, L, H), i, and $(\varDelta, \beth)$ be given as in the Expansion rule.

As there is no reason for updating the variable-condition R or the R-choice-condition,
set $(C', R') := (C, R)$.

Let $(j, ((\varPhi, \daleth), t'')) \in F$ be the proof tree whose proposition we want to apply.

Set $Y := \{\ y^{\delta^-} {\in} \mathcal{V}_{\delta^-}(\varPhi, \daleth) \bigm| \mathcal{V}_{\gamma\delta^+}(\varPhi, \daleth) \times \{y^{\delta^-}\} \subseteq R'\ \}$. Note that Y contains exactly those free δ^--variables of $(\varPhi, \daleth)$ that have neither free γ-variables nor free δ^+-variables of $(\varPhi, \daleth)$ in their "R'-scopes". In other words, the variables in Y are those free δ^--variables upon which neither a solution for the free γ-variables nor a choice-condition for the free δ^+-variables in $(\varPhi, \daleth)$ depends. Therefore, the variables in Y are those which we can instantiate when applying $(\varPhi, \daleth)$.[17]

Thus, let ϱ be a substitution on Y.

To complete the description of a sub-rule of the Expansion rule in a tableau tree we have to present the sets N_{L} (applied lemmas), N_{H} (applied induction hypotheses), and M (sequents generating the sub-goals). These sets differ for lemma and induction-hypothesis application. A lemma is simply added to the context of the goal $(\varDelta, \beth)$. In case of an induction hypothesis, we also have to add sub-goals which express that (2) the instantiated induction hypothesis is smaller than the goal, (3) the induction ordering is wellfounded, (4) the induction orderings and (5) the induction quasi-orderings of the instantiated hypothesis and the goal are identical and (6) compatible.

Lemma Application: Set $N_{\mathrm{L}} := \{j\}$ and $N_{\mathrm{H}} := \emptyset$. As it would be fatal to destroy the wellfoundedness of $L{\circ}H^*$ required in Theorem 2.45, it is reasonable to forbid $i\ (L{\cup}H)^*\ j$. Let M be the set containing the single-formula sequents $\overline{B\varrho}$ for each formula B listed in the sequent $\varPhi$.

Induction-Hypothesis Application: Set $N_{\mathrm{L}} := \emptyset$ and $N_{\mathrm{H}} := \{j\}$. As it would be fatal to destroy the wellfoundedness of $L{\circ}H^*$ required in Theorem 2.45, it is reasonable to forbid $i\ H^*{\circ}(L{\circ}H^*)^+\ j$. Set $(w, <, \lesssim) := \beth$ and $(w', <', \lesssim') := \daleth$. Let α be the common type of w and w'. Let M be the set containing the following single-formula sequents:

(1) $\overline{B\varrho}$ for each formula B listed in the sequent Φ

(2) $w'\varrho < w$

(3) $\forall p : \alpha \to \mathsf{bool}.\ \big(\ \exists a{:}\alpha.\ p(a)\ \Rightarrow\ \exists a{:}\alpha.\ \big(\ p(a) \wedge \neg\exists a'{:}\alpha.\ (p(a') \wedge a'{<}a)\,\big)\,\big)$ [18]

(4) $\forall x, y : \alpha.\ \big(\ x < y\ \Leftrightarrow\ x\ ({<'}\varrho)\ y\ \big)$

(5) $\forall x, y : \alpha.\ \big(\ x \lesssim y\ \Leftrightarrow\ x\ ({\lesssim'}\varrho)\ y\ \big)$

(6) $\forall x, y, z : \alpha.\ \big(\ (x < y \wedge y \lesssim z) \Rightarrow\ x < z\ \big)$

Each of the above sequents (3)–(6) can be omitted if the following holds, respectively, for any $\mathcal{A} \in \mathrm{K}$, $(\mathcal{A}, R')$-valuation e, and π and τ such that π is $(e, \mathcal{A})$-compatible with (C', R') and $((\Delta, \sqsupset), \tau)$ is an $(\pi, e, \mathcal{A})$-counterexample, and for $\delta := \epsilon(\pi)(\tau) \uplus \tau$, $\lhd := \mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta) \uplus \delta)(<)$, and $\underset{\sim}{\lhd} := \mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta) \uplus \delta)(\lesssim)$:

(3) $\lhd$ is wellfounded

(4) $\lhd = \mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta) \uplus \delta)({<'}\varrho)$

(5) $\underset{\sim}{\lhd} = \mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta) \uplus \delta)({\lesssim'}\varrho)$

(6) $\lhd \circ \underset{\sim}{\lhd} \subseteq \lhd^{+}$

Thus, the sequents (3)–(6) can be omitted if we have a fixed wellfounded induction (quasi-) ordering, as described at the end of Section 2.3.1. The sequents (5) and (6) can also be omitted in the important special case (cf. Section 3.4) that the third component $\underset{\sim}{\lhd}$ of the weights is restricted to be the empty relation $\emptyset$.

THEOREM 2.51
The rules for lemma and induction-hypothesis application described above are safe sub-rules of the Expansion rule of the abstract sequent and tableau calculus.

Detailed examples showing how Theorem 2.51 should be used are given in Section 3.1 (lemma application) and Section 3.2 ff. (induction-hypothesis application).

Note that there is no analogon of Theorem 2.51 instantiating a set of free δ^{+}-variables instead of the set Y of free δ^{-}-variables. Thus, free δ^{-}-variables are necessary even if we are not interested in non-liberalized δ-steps. As explained in Section 3.1, we should always use free δ^{-}-variables in Hypothesizing steps. Moreover, to have more useful lemmas and induction hypotheses, we sometimes have to split a tree at an inner position with a Hypothesizing step introducing a new proposition with free δ^{-}-variables replacing the free δ^{+}-variables and apply this new proposition as a lemma to the new leaf of the old tree, closing this branch, cf. the discussion at the end of Section 3.2.3.

2.5.3 Other Concrete Inference Steps

More specialized sub-rules of the Expansion rule are appropriate for practical inference systems such as the one presented in Wirth (1997), Kühler (2000), but for our purposes here, the basic rules of Theorem 2.49 and Theorem 2.51 are sufficient.

3 Examples

3.1 An Example for Lemma Application

In this example, the proofs are presented as tableau trees, which we do not depict because they all have branching degree 1. As there are no inductive proofs, we omit the weights completely. As no liberalized δ-rules are applied, the choice-conditions are always empty. Assume that in the signature Σ we have the operator $*$, the constant 1, and the inverse function inv.

We begin with the empty proof forest $(F, C, R, L, H) := (\emptyset, \emptyset, \emptyset, \emptyset, \emptyset)$.

Then we start a new proof tree with number 1 for the associativity of $*$ as

(1) $$x_1^{\delta^-} * (y_1^{\delta^-} * z_1^{\delta^-}) = (x_1^{\delta^-} * y_1^{\delta^-}) * z_1^{\delta^-}$$

by a Hypothesizing step in the tableau calculus of Definition 2.42, just as two new proof trees for

(2) $$\mathsf{1} * x_2^{\delta^-} = x_2^{\delta^-}$$

(3) $$\mathsf{inv}(x_3^{\delta^-}) * x_3^{\delta^-} = \mathsf{1}$$

With these three trees we have the axioms of group theory at hand via lemma application.

Now we really want to prove something. We start the new proof tree number 4 for

(4) $$\forall x.\ \ x * \mathsf{inv}(x) = \mathsf{1}$$

by a Hypothesizing step. The the root of proof tree 4 is labeled with

$$\neg\forall x.\ \ x * \mathsf{inv}(x) = \mathsf{1}$$

A δ-step (cf. Theorem 2.49) adds the child

$$x_4^{\delta^-} * \mathsf{inv}(x_4^{\delta^-}) \neq \mathsf{1}$$

Our variable-condition is still empty because no free variables occur in $\neg\forall x.\ x * \mathsf{inv}(x) = \mathsf{1}$.
Applying the sequent of proof tree 3 in the way of Theorem 2.51 with $\varrho := \{x_3^{\delta^-} \mapsto y_1^{\gamma}\}$ adds the new child

$$\mathsf{inv}(y_1^{\gamma}) * y_1^{\gamma} = \mathsf{1}$$

to proof tree 4 and inserts the pair $(3, 4)$ into L. A Rewrite step (cf. Theorem 2.49) with this equality from right to left produces the new child

$$x_4^{\delta^-} * \mathsf{inv}(x_4^{\delta^-}) \neq \mathsf{inv}(y_1^{\gamma}) * y_1^{\gamma}$$

Applying the sequent of proof tree 2 in the way of Theorem 2.51 with $\varrho := \{x_2^{\delta^-} \mapsto y_1^{\gamma}\}$ adds the new child

$$\mathsf{1} * y_1^{\gamma} = y_1^{\gamma}$$

to proof tree 4 and inserts the pair $(2, 4)$ into L.
This new child can be used for a Rewrite step from right to left adding the child

$$x_4^{\delta^-} * \mathsf{inv}(x_4^{\delta^-}) \neq \mathsf{inv}(y_1^{\gamma}) * (\mathsf{1} * y_1^{\gamma})$$

Applying the sequent of proof tree 3 in the way of Theorem 2.51 adds the new child

$$\mathsf{inv}(y_2^{\gamma}) * y_2^{\gamma} = \mathsf{1}$$

A Rewrite step (cf. Theorem 2.49) with this equality from right to left produces the new child

$$x_4^{\delta^-} * \mathsf{inv}(x_4^{\delta^-}) \neq \mathsf{inv}(y_1^{\gamma}) * ((\mathsf{inv}(y_2^{\gamma}) * y_2^{\gamma}) * y_1^{\gamma})$$

With two applications of the sequent of proof tree 1, this can be rewritten into

$$x_4^{\delta^-} * \mathsf{inv}(x_4^{\delta^-}) \neq (\mathsf{inv}(y_1^{\gamma}) * \mathsf{inv}(y_2^{\gamma})) * (y_2^{\gamma} * y_1^{\gamma})$$

Note that now $L = \{1, 2, 3\} \times \{4\}$.
Applying the sequent of proof tree 3 in the way of Theorem 2.51 adds the new child

$$\mathsf{inv}(y_3^{\gamma}) * y_3^{\gamma} = \mathsf{1}$$

While the proof up to now required some ingenuity, the following can be easily automated. To use the latter new child for a Rewrite step from left to right at the position 1 of the right-hand side of the previous one, we apply the unifier $\sigma := \{y_1^\gamma{\mapsto}\mathsf{inv}(y_2^\gamma),\ y_3^\gamma{\mapsto}\mathsf{inv}(y_2^\gamma)\}$ to the whole proof forest and—after the Rewrite step—get the new child

$$x_4^{\delta^-} * \mathsf{inv}(x_4^{\delta^-}) \neq 1 * (y_2^\gamma * \mathsf{inv}(y_2^\gamma))$$

Note that σ is an R-substitution on V_γ in our proof state with $R = \emptyset$, and that the σ-update R' of R is given by

$$\begin{tikzcd} y_1^\gamma & y_2^\gamma \arrow[l, "\Gamma_\sigma"'] \arrow[dl, "\Gamma_\sigma"] \\ y_3^\gamma & \end{tikzcd}$$

. After global application of σ, the free γ-variables y_1^γ and y_3^γ do not occur anywhere in our current proof forest. Thus, even the updated variable-condition does not put any restrictions on R-substitutions on V_γ, unless we would re-use y_1^γ or y_3^γ. [19]

With an application of the sequent of proof tree 2, the formula of the last new node can be rewritten into

$$x_4^{\delta^-} * \mathsf{inv}(x_4^{\delta^-}) \neq y_2^\gamma * \mathsf{inv}(y_2^\gamma)$$

An Instantiation step applying $\{y_2^\gamma \mapsto x_4^{\delta^-}\}$ turns this into

$$x_4^{\delta^-} * \mathsf{inv}(x_4^{\delta^-}) \neq x_4^{\delta^-} * \mathsf{inv}(x_4^{\delta^-})$$

Now the tree is closed because all sequents of the form $(t = t)\ \Lambda$ are assumed to be in our axioms $\mathcal{AX}$. By Theorem 2.45 we now know that $\forall x.\ \ x * \mathsf{inv}(x) = 1$ is $\emptyset$-valid, provided that the proof trees 1, 2, and 3 are closed, which is the case when we assume their sequents to be in $\mathcal{AX}$.

Now we start proof tree 5 for

(5) $$x_5^{\delta^-} * \mathsf{inv}(x_5^{\delta^-}) = 1$$

by a Hypothesizing step. Note that the sequent is not really different from that of proof tree 4. We prefer the form of proof tree 5 because it will be more useful for *descente infinie*. For purely deductive theorem proving, the two only differ in that the form of proof tree 5 is handier for lemma application. To see this, we will prove each with the help of the other. A lemma application according to Theorem 2.51 of the sequent of proof tree 4 to proof tree 5 whose root is labeled with $x_5^{\delta^-} * \mathsf{inv}(x_5^{\delta^-}) \neq 1$ adds the child

$$\forall x.\ \ x * \mathsf{inv}(x) = 1.$$

A γ-step adds the child

$$x_5^{\delta^-} * \mathsf{inv}(x_5^{\delta^-}) = 1.$$

Now proof tree 5 is closed because all sequents of the form $A\ \Lambda\ \overline{A}$ are assumed to be in our axioms $\mathcal{AX}$.

Finally, we start another proof tree number 6 for the sequent of proof tree 4. The root is again labeled with

$$\neg\forall x.\ \ x * \mathsf{inv}(x) = 1$$

A δ-step adds the child

$$x_6^{\delta^-} * \mathsf{inv}(x_6^{\delta^-}) \neq 1$$

Applying the sequent of proof tree 5 in the way of Theorem 2.51 adds the new child

$$x_6^{\delta^-} * \mathsf{inv}(x_6^{\delta^-}) = 1$$

Now proof tree 6 is also closed because all sequents of the form $A\ \overline{A}\ \Lambda$ are assumed to be in our axioms $\mathcal{AX}$.

Note that finally we have $L = \{1,2,3\} \times \{4\} \cup \{(4,5),\ (5,6)\}$ and $H = \emptyset$, so that $L{\circ}H^*$ is wellfounded and Theorem 2.45 can be applied indeed.

3.2 An Example for Mutual Induction

3.2.1 Induction Ordering in QuodLibet

While for general *descente infinie*—as described in Section 2.3.1—not only the weights but also the induction ordering can be chosen for each proof differently, in QuodLibet, a tactic-based inductive theorem proving system for clausal logic, cf. Wirth (1997), Kühler (2000), Avenhaus &al. (2003), it has turned out to be adequate to use the following fixed wellfounded quasi-ordering depending on the signature Σ:

The *semantical length* of a ground term is the syntactical length of a constructor ground term equal to it. The admissibility conditions guarantee that there is at most one such term. The lexicographic extension up to a fixed finite length[20] of the lifting of the semantical length results in a wellfounded quasi-ordering on the objects of each of the models that establish the inductive validity of QuodLibet (i.e. type-C in Wirth & Gramlich (1994b)).

Although the induction ordering is fixed, the lazy substitution of the second-order weight variables during the proofs provides sufficient flexibility for the intended application domain of partially defined recursive functions, cf. Kühler & Wirth (1996), Wirth & Gramlich (1994a).

3.2.2 The P & Q Example

The toy example of this section illustrates how mutual induction works in our framework. As the proof requires mutual induction with non-trivial weights, it cannot be performed in many inductive theorem proving systems or the lean induction calculus of Baaz &al. (1997). The signature is the one presented in Section 1.1.1, enriched with the predicates $\mathsf{P} : \mathsf{nat} \to \mathsf{bool}$ and $\mathsf{Q} : \mathsf{nat} \to \mathsf{nat} \to \mathsf{bool}$. Besides the axiom $(\mathsf{nat1})$ of Section 1.1.1, we have the following axioms, defining the special predicates of our example.

(P1) $\qquad \mathsf{P}(\mathsf{0})$

(P2) $\quad \forall x.\ \ (\ \mathsf{P}(\mathsf{s}(x)) \Leftarrow (\ \mathsf{P}(x) \wedge \mathsf{Q}(x, \mathsf{s}(x))\ )\ )$

(Q1) $\qquad \forall x.\ \ \mathsf{Q}(x, \mathsf{0})$

(Q2) $\quad \forall x, y.\ \ (\ \mathsf{Q}(x, \mathsf{s}(y)) \Leftarrow (\ \mathsf{Q}(x, y) \wedge \mathsf{P}(x)\ )\ )$

We want to show that both predicates are tautological:

(1) $\qquad \mathsf{P}(x_0^{\delta^-});\ \ w_1^{\gamma}(x_0^{\delta^-})$

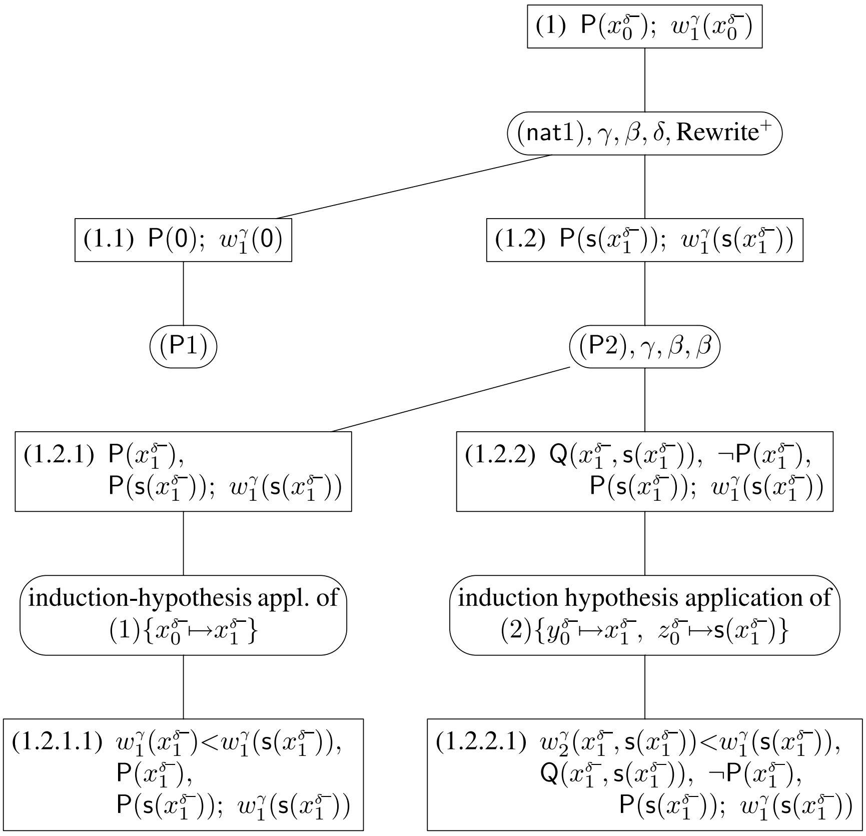
(2) $\qquad \mathsf{Q}(y_0^{\delta^-}, z_0^{\delta^-});\ \ w_2^{\gamma}(y_0^{\delta^-}, z_0^{\delta^-})$

Note that weights consist only of weight terms (like $w_1^{\gamma}(x_0^{\delta^-})$ in (1)) because we fix the induction (quasi-) ordering to be the single one of the QuodLibet system, as discussed in Section 3.2.1. Therefore—as discussed in Section 2.5—the items (3)–(6) of Theorem 2.51 can be omitted in the following.

In the Hypothesizing steps for (1) and (2) we introduce the variable-condition

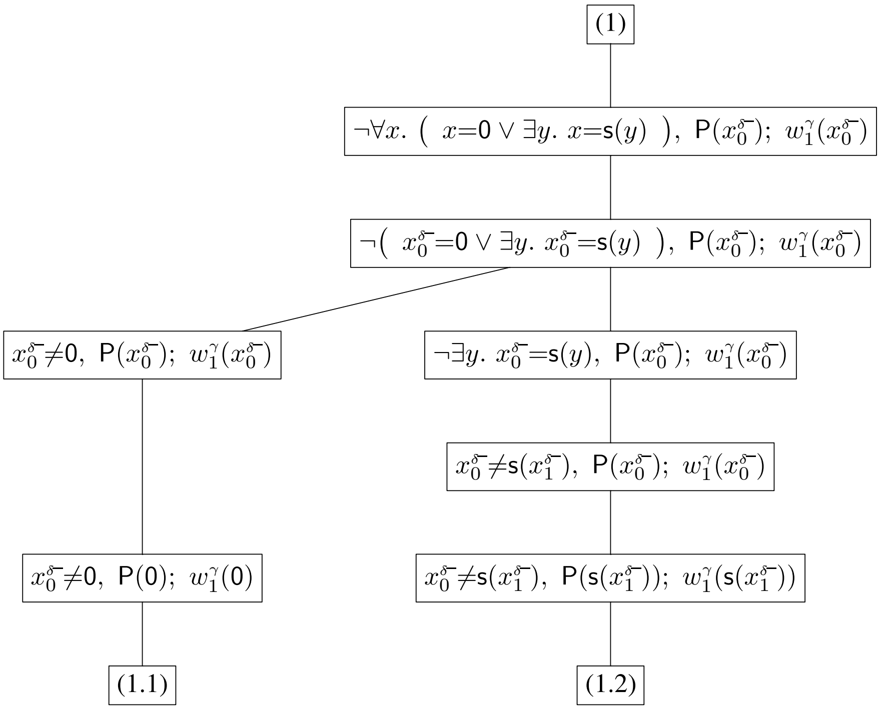
$$R := \left(\begin{array}{cc} & \mathcal{V}_{\gamma\delta^+}((1)) \times \mathcal{V}_{\delta^-}((1)) \\ \cup & \mathcal{V}_{\gamma\delta^+}((2)) \times \mathcal{V}_{\delta^-}((2)) \end{array} \right) = \left(\begin{array}{cc} & \{w_1^\gamma\} \times \{x_0^{\delta^-}\} \\ \cup & \{w_3^\gamma\} \times \{y_0^{\delta^-}, z_0^{\delta^-}\} \end{array} \right)$$

to have all free δ^--variables of (1) or (2) in the set Y of Theorem 2.51. After several inference steps, QUODLIBET presents a sequent tree for (1) similar to following:

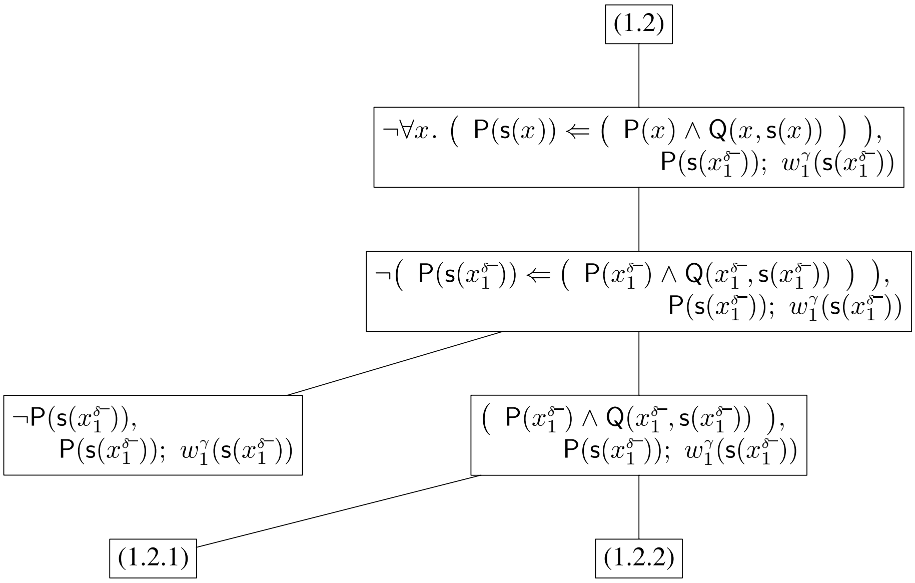

(1) $\mathsf{P}(x_0^{\delta^-});\ w_1^\gamma(x_0^{\delta^-})$
$(\mathsf{nat1}), \gamma, \beta, \delta, \text{Rewrite}^+$
(1.1) $\mathsf{P}(0);\ w_1^\gamma(0)$
(1.2) $\mathsf{P}(\mathsf{s}(x_1^{\delta^-}));\ w_1^\gamma(\mathsf{s}(x_1^{\delta^-}))$
$(\mathsf{P1})$
$(\mathsf{P2}), \gamma, \beta, \beta$
(1.2.1) $\mathsf{P}(x_1^{\delta^-}),\ \mathsf{P}(\mathsf{s}(x_1^{\delta^-}));\ w_1^\gamma(\mathsf{s}(x_1^{\delta^-}))$
(1.2.2) $\mathsf{Q}(x_1^{\delta^-}, \mathsf{s}(x_1^{\delta^-})),\ \neg\mathsf{P}(x_1^{\delta^-}),\ \mathsf{P}(\mathsf{s}(x_1^{\delta^-}));\ w_1^\gamma(\mathsf{s}(x_1^{\delta^-}))$
induction-hypothesis appl. of $(1)\{x_0^{\delta^-} \mapsto x_1^{\delta^-}\}$
induction hypothesis application of $(2)\{y_0^{\delta^-} \mapsto x_1^{\delta^-},\ z_0^{\delta^-} \mapsto \mathsf{s}(x_1^{\delta^-})\}$
(1.2.1.1) $w_1^\gamma(x_1^{\delta^-}) < w_1^\gamma(\mathsf{s}(x_1^{\delta^-})),\ \mathsf{P}(x_1^{\delta^-}),\ \mathsf{P}(\mathsf{s}(x_1^{\delta^-}));\ w_1^\gamma(\mathsf{s}(x_1^{\delta^-}))$
(1.2.2.1) $w_2^\gamma(x_1^{\delta^-}, \mathsf{s}(x_1^{\delta^-})) < w_1^\gamma(\mathsf{s}(x_1^{\delta^-})),\ \mathsf{Q}(x_1^{\delta^-}, \mathsf{s}(x_1^{\delta^-})),\ \neg\mathsf{P}(x_1^{\delta^-}),\ \mathsf{P}(\mathsf{s}(x_1^{\delta^-}));\ w_1^\gamma(\mathsf{s}(x_1^{\delta^-}))$

The square boxes are the nodes of the proof tree, whereas the round-edged boxes show applications of inference rules of Theorem 2.49 and Theorem 2.51, which are more elementary than the inference rules in QUODLIBET. We can check whether the tree is closed simply by realizing that all leaves are round-edged nodes. This is not only useful for implementation purposes (where we have to record somewhere why a branch is closed) but also immediately realizes the explicit representation of leaves required by Definition 2.42.

For example, "$(\mathsf{nat1}), \gamma, \beta, \delta, \text{Rewrite}^+$" in the first round-edged box means that we use the axiom $(\mathsf{nat1})$ as a lemma in Theorem 2.51, and then apply a γ-, a β-, and a δ-step and several Rewrite-steps of Theorem 2.49 to get the following proof tree below, where in the last inference steps (resulting in (1.1) and (1.2)) the left-most literals of the parents of the leaf nodes are safely (cf. Section 2.4.2) removed because $x_0^{\delta^-}$ is in solved[21] form.


(1)
$\neg\forall x.\ (\ x{=}0 \lor \exists y.\ x{=}\mathsf{s}(y)\ ),\ \mathsf{P}(x_0^{\delta^-});\ w_1^\gamma(x_0^{\delta^-})$
$\neg(\ x_0^{\delta^-}{=}0 \lor \exists y.\ x_0^{\delta^-}{=}\mathsf{s}(y)\ ),\ \mathsf{P}(x_0^{\delta^-});\ w_1^\gamma(x_0^{\delta^-})$
$x_0^{\delta^-}{\neq}0,\ \mathsf{P}(x_0^{\delta^-});\ w_1^\gamma(x_0^{\delta^-})$
$\neg\exists y.\ x_0^{\delta^-}{=}\mathsf{s}(y),\ \mathsf{P}(x_0^{\delta^-});\ w_1^\gamma(x_0^{\delta^-})$
$x_0^{\delta^-}{\neq}\mathsf{s}(x_1^{\delta^-}),\ \mathsf{P}(x_0^{\delta^-});\ w_1^\gamma(x_0^{\delta^-})$
$x_0^{\delta^-}{\neq}0,\ \mathsf{P}(0);\ w_1^\gamma(0)$
$x_0^{\delta^-}{\neq}\mathsf{s}(x_1^{\delta^-}),\ \mathsf{P}(\mathsf{s}(x_1^{\delta^-}));\ w_1^\gamma(\mathsf{s}(x_1^{\delta^-}))$
(1.1)
(1.2)

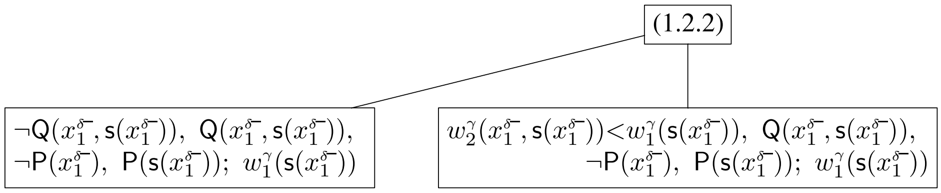
Let us have a closer look at the inference below (1.2). The defining formula (P2) is applied as a lemma in Theorem 2.51, i.e. its single formula is added in negated form. Thus, the round-edged node labeled with “$(\mathsf{P2}), \gamma, \beta, \beta$” can be replaced with the following subtree. Note that the leftmost leaf of the tree below is closed and can be omitted in the global tree.


(1.2)
$\neg\forall x.\ (\ \mathsf{P}(\mathsf{s}(x)) \Leftarrow (\ \mathsf{P}(x) \land \mathsf{Q}(x,\mathsf{s}(x))\ )\ ),$
$\mathsf{P}(\mathsf{s}(x_1^{\delta^-}));\ w_1^\gamma(\mathsf{s}(x_1^{\delta^-}))$
$\neg(\ \mathsf{P}(\mathsf{s}(x_1^{\delta^-})) \Leftarrow (\ \mathsf{P}(x_1^{\delta^-}) \land \mathsf{Q}(x_1^{\delta^-},\mathsf{s}(x_1^{\delta^-}))\ )\ ),$
$\mathsf{P}(\mathsf{s}(x_1^{\delta^-}));\ w_1^\gamma(\mathsf{s}(x_1^{\delta^-}))$
$\neg\mathsf{P}(\mathsf{s}(x_1^{\delta^-})),$
$\mathsf{P}(\mathsf{s}(x_1^{\delta^-}));\ w_1^\gamma(\mathsf{s}(x_1^{\delta^-}))$
$(\ \mathsf{P}(x_1^{\delta^-}) \land \mathsf{Q}(x_1^{\delta^-},\mathsf{s}(x_1^{\delta^-}))\ ),$
$\mathsf{P}(\mathsf{s}(x_1^{\delta^-}));\ w_1^\gamma(\mathsf{s}(x_1^{\delta^-}))$
(1.2.1)
(1.2.2)

Even more interesting is what happens below (1.2.2). We instantiate the meta-variables of Theorem 2.51 as follows:

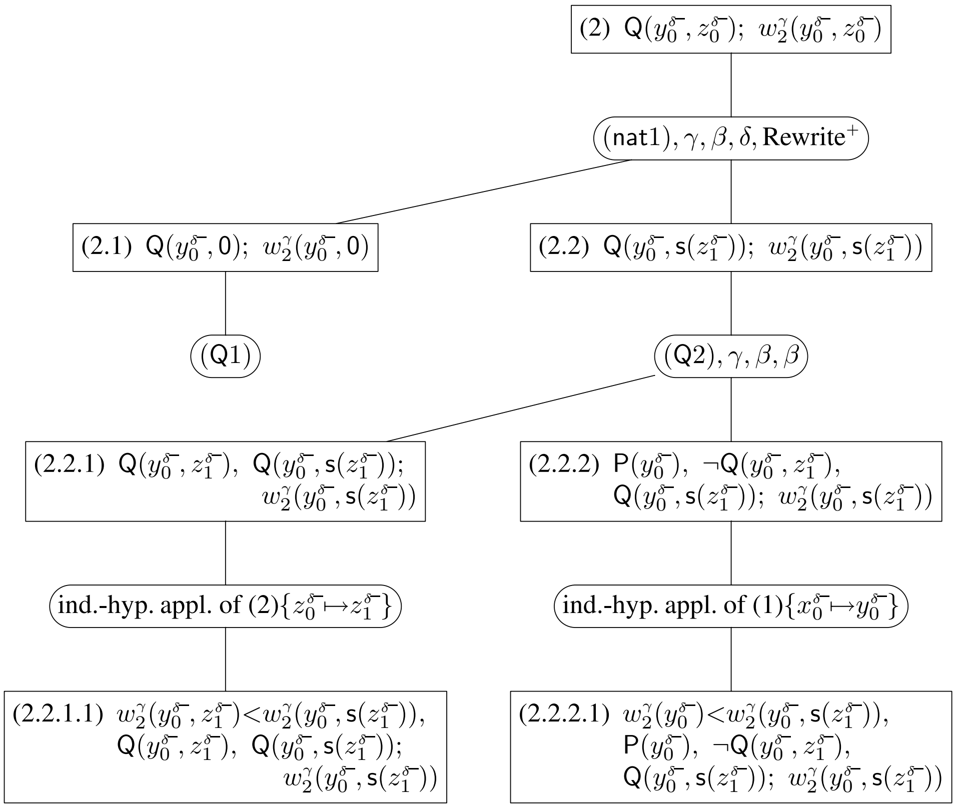
$$\begin{array}{rcl}
\Phi & := & \mathsf{Q}(y_0^{\delta^-}, z_0^{\delta^-}) \\
\daleth & := & w_2^{\gamma}(y_0^{\delta^-}, z_0^{\delta^-}) \\
Y & := & \{y_0^{\delta^-}, z_0^{\delta^-}\} \\
\varrho & := & \{y_0^{\delta^-}{\mapsto}x_1^{\delta^-},\ z_0^{\delta^-}{\mapsto}\mathsf{s}(x_1^{\delta^-})\} \\
M & := & \{\neg\mathsf{Q}(x_1^{\delta^-}, \mathsf{s}(x_1^{\delta^-})),\ w_2^{\gamma}(x_1^{\delta^-}, \mathsf{s}(x_1^{\delta^-})){<}w_1^{\gamma}(\mathsf{s}(x_1^{\delta^-}))\}
\end{array}$$

This results in the tree below. Its left leaf is closed and its right leaf is (1.2.2.1).



(1.2.2)
$\neg\mathsf{Q}(x_1^{\delta^-}, \mathsf{s}(x_1^{\delta^-})),\ \mathsf{Q}(x_1^{\delta^-}, \mathsf{s}(x_1^{\delta^-})),\ \neg\mathsf{P}(x_1^{\delta^-}),\ \mathsf{P}(\mathsf{s}(x_1^{\delta^-}));\ w_1^{\gamma}(\mathsf{s}(x_1^{\delta^-}))$
$w_2^{\gamma}(x_1^{\delta^-}, \mathsf{s}(x_1^{\delta^-})){<}w_1^{\gamma}(\mathsf{s}(x_1^{\delta^-})),\ \mathsf{Q}(x_1^{\delta^-}, \mathsf{s}(x_1^{\delta^-})),\ \neg\mathsf{P}(x_1^{\delta^-}),\ \mathsf{P}(\mathsf{s}(x_1^{\delta^-}));\ w_1^{\gamma}(\mathsf{s}(x_1^{\delta^-}))$

For (2) we get a sequent tree very similar to that of (1):



(2) $\mathsf{Q}(y_0^{\delta^-}, z_0^{\delta^-});\ w_2^{\gamma}(y_0^{\delta^-}, z_0^{\delta^-})$
$(\mathsf{nat1}), \gamma, \beta, \delta, \text{Rewrite}^+$
(2.1) $\mathsf{Q}(y_0^{\delta^-}, 0);\ w_2^{\gamma}(y_0^{\delta^-}, 0)$
(2.2) $\mathsf{Q}(y_0^{\delta^-}, \mathsf{s}(z_1^{\delta^-}));\ w_2^{\gamma}(y_0^{\delta^-}, \mathsf{s}(z_1^{\delta^-}))$
(Q1)
(Q2), γ, β, β
(2.2.1) $\mathsf{Q}(y_0^{\delta^-}, z_1^{\delta^-}),\ \mathsf{Q}(y_0^{\delta^-}, \mathsf{s}(z_1^{\delta^-}));\ w_2^{\gamma}(y_0^{\delta^-}, \mathsf{s}(z_1^{\delta^-}))$
(2.2.2) $\mathsf{P}(y_0^{\delta^-}),\ \neg\mathsf{Q}(y_0^{\delta^-}, z_1^{\delta^-}),\ \mathsf{Q}(y_0^{\delta^-}, \mathsf{s}(z_1^{\delta^-}));\ w_2^{\gamma}(y_0^{\delta^-}, \mathsf{s}(z_1^{\delta^-}))$
ind.-hyp. appl. of (2)$\{z_0^{\delta^-}{\mapsto}z_1^{\delta^-}\}$
ind.-hyp. appl. of (1)$\{x_0^{\delta^-}{\mapsto}y_0^{\delta^-}\}$
(2.2.1.1) $w_2^{\gamma}(y_0^{\delta^-}, z_1^{\delta^-}){<}w_2^{\gamma}(y_0^{\delta^-}, \mathsf{s}(z_1^{\delta^-})),\ \mathsf{Q}(y_0^{\delta^-}, z_1^{\delta^-}),\ \mathsf{Q}(y_0^{\delta^-}, \mathsf{s}(z_1^{\delta^-}));\ w_2^{\gamma}(y_0^{\delta^-}, \mathsf{s}(z_1^{\delta^-}))$
(2.2.2.1) $w_2^{\gamma}(y_0^{\delta^-}){<}w_2^{\gamma}(y_0^{\delta^-}, \mathsf{s}(z_1^{\delta^-})),\ \mathsf{P}(y_0^{\delta^-}),\ \neg\mathsf{Q}(y_0^{\delta^-}, z_1^{\delta^-}),\ \mathsf{Q}(y_0^{\delta^-}, \mathsf{s}(z_1^{\delta^-}));\ w_2^{\gamma}(y_0^{\delta^-}, \mathsf{s}(z_1^{\delta^-}))$

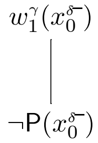
We have applied each of the two weighted sequents (1) and (2) in each of their two proof trees 1 and 2. Luckily we used induction hypothesis application instead of lemma application. The latter would have resulted in a lemma application relation of $\{1,2\} \times \{1,2\}$ which is not wellfounded and our proof trees would have been useless because we would never be able to apply Theorem 2.45. As we have used induction hypothesis application instead of lemma application, we have produced the four additional leaves (1.2.1.1), (1.2.2.1), (2.2.1.1), and (2.2.2.1), which are still open. We choose our 2nd order weight functions according to $w_1^{\gamma}(x) := (x)$ and $w_2^{\gamma}(x,y) := (x,y)$, using the lexicographic combination of Section 3.2.1. Now the proof attempt can be successfully completed: E.g., the first literal of (1.2.1.1) turns into $(x_1^{\delta^-}, \mathsf{0}) < (\mathsf{s}(x_1^{\delta^-}), \mathsf{0})$, which simplifies to QUODLIBET's ordering axiom $x_1^{\delta^-} < \mathsf{s}(x_1^{\delta^-})$.

Which steps in this proof were typical for *inductive* theorem proving in the sense that their soundness relies on notions of inductive validity instead of the stronger notion of validity in all models?

> Besides the four induction hypothesis applications, the final branch closure rules for $<$-literals are typical for induction because they require that, in all models in **K**, the successor of each natural number is different from that natural number and each natural number is built-up from zero by a finite number of successor steps (i.e. there are neither cycles nor **Z**-chains in the models, cf. Enderton (1973)).

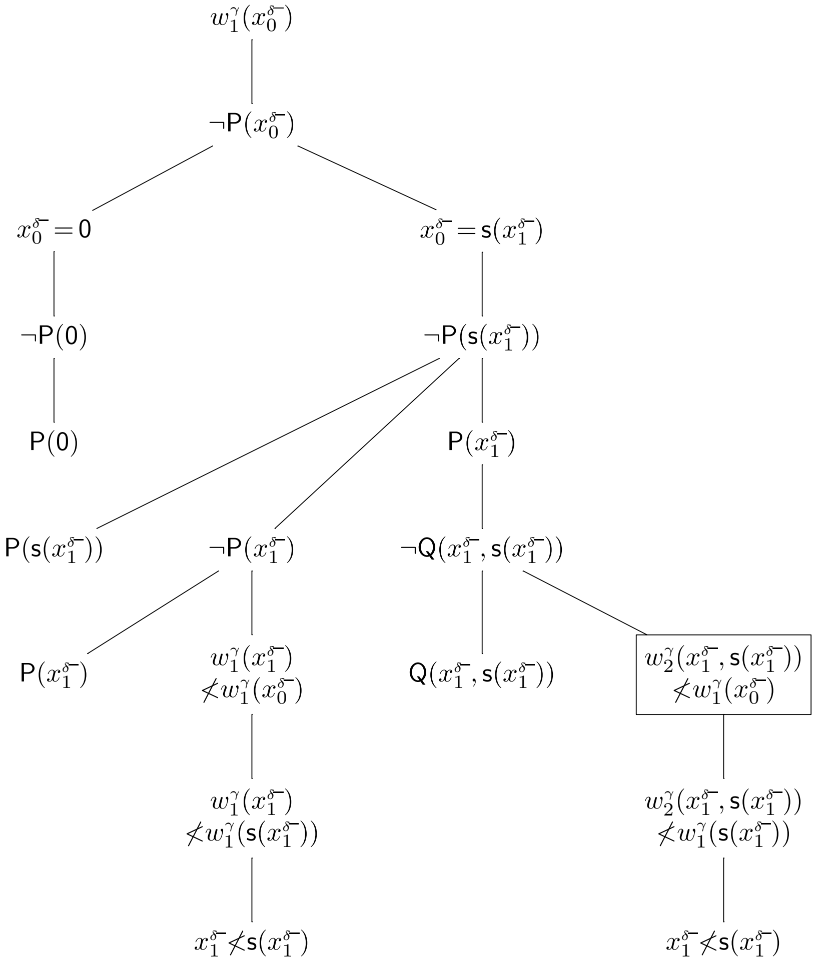
3.2.3 Sequents versus Tableaus in *Descente Infinie*

In this section we are going to compare the appropriateness of sequent versus tableau trees under the special aspect of *descente infinie*. To this end we first see what the sequent tree of (1) of Section 3.2.2 would look like as a tableau tree. After the first Hypothesizing step, the initial tableau for (1) looks the following way.


$w_1^{\gamma}(x_0^{\delta^-})$
$\neg\mathsf{P}(x_0^{\delta^-})$

Note that this differs from (1) in duality. While this is not a hindrance for completely automatic ITP systems, it poses considerable practical problems in systems where user-guidance is possible: The primitive process of switching duality is a typical source of errors for human beings (or me at least).

For the closed complete proof tree for (1) on the following page, we have chosen a representation according to clausal tableau calculi because there is not enough space for non-atomic formulas here. Let us have a closer look at the boxed formula in this tableau. It results from induction hypothesis application of (2). Note that the only difference to an Extension step in Model Elimination tableaus (cf. Baumgartner &al. (1997)) lies with the additional child (the boxed node), which asks us to show that the instance of the hypothesis is smaller than the weight of our proof tree. Indeed: As the induction ordering is fixed here, hypothesis application differs from the standard lemma (or axiom) application only in producing an additional $<$-goal. This makes hypothesis application a little more expensive than lemma application.


$w_1^\gamma(x_0^{\delta^-})$
$\neg\mathsf{P}(x_0^{\delta^-})$
$x_0^{\delta^-}=0$
$x_0^{\delta^-}=\mathsf{s}(x_1^{\delta^-})$
$\neg\mathsf{P}(0)$
$\neg\mathsf{P}(\mathsf{s}(x_1^{\delta^-}))$
$\mathsf{P}(0)$
$\mathsf{P}(x_1^{\delta^-})$
$\mathsf{P}(\mathsf{s}(x_1^{\delta^-}))$
$\neg\mathsf{P}(x_1^{\delta^-})$
$\neg\mathsf{Q}(x_1^{\delta^-},\mathsf{s}(x_1^{\delta^-}))$
$\mathsf{P}(x_1^{\delta^-})$
$w_1^\gamma(x_1^{\delta^-}) \not< w_1^\gamma(x_0^{\delta^-})$
$\mathsf{Q}(x_1^{\delta^-},\mathsf{s}(x_1^{\delta^-}))$
$w_2^\gamma(x_1^{\delta^-},\mathsf{s}(x_1^{\delta^-})) \not< w_1^\gamma(x_0^{\delta^-})$
$w_1^\gamma(x_1^{\delta^-}) \not< w_1^\gamma(\mathsf{s}(x_1^{\delta^-}))$
$w_2^\gamma(x_1^{\delta^-},\mathsf{s}(x_1^{\delta^-})) \not< w_1^\gamma(\mathsf{s}(x_1^{\delta^-}))$
$x_1^{\delta^-} \not< \mathsf{s}(x_1^{\delta^-})$
$x_1^{\delta^-} \not< \mathsf{s}(x_1^{\delta^-})$

The left-hand term $w_2^\gamma(x_1^{\delta^-},\mathsf{s}(x_1^{\delta^-}))$ is the weight term of (2) instantiated via $\{y_0^{\delta^-}\mapsto\mathsf{s}(x_1^{\delta^-}),\ z_0^{\delta^-}\mapsto x_1^{\delta^-}\}$ because this substitution enables the left sibling of the boxed node to close its branch with the instantiated formula of (2). The right-hand term $w_1^\gamma(x_0^{\delta^-})$ comes down from the root of the tree. Contrary to the sequent tree where the weight of the root is carried along and updated on its way down, we have to rewrite the variable $x_0^{\delta^-}$ in it with an ancestor equality literal to know what the root weight means in the local context.

Note that the sequent tree is not equal to the result of the standard transformation of the tableau tree. The standard transformation of a tableau tree into a sequent tree works for inductive trees just as for deductive trees:

1. Bottom-up replace the label of each node with the weighted sequent listing the conjugates of the formulas and the weight labeling the (partial) branch from this node to the root.
2. Remove the root part of the tree where the nodes are ancestors of a node of the initial Hypothesizing step (in our example: remove the root node).

This standard transformation multiplies the number of formulas labeling each proof tree with at most nearly the depth of that tree, but does not use the advantages of sequent trees, namely the ability to simplify formulas that label ancestor nodes in a tableau tree. For example, in the above tableau tree it is not possible to rewrite the literal $\neg\mathsf{P}(x_0^{\delta^-})$ with the equality literals below it in place. In tableau trees, an equality literal can be used to rewrite formulas of its offspring in place, whereas it must copy ancestor formulas beforehand down to its offspring because the ancestor is also part of other branches that do not include the equality literal. Moreover, the weight term can be rewritten in the sequent tree, which again is not possible in the tableau version where the weight is at the root node. Since $x_0^{\delta^-}$ is in solved form after the Rewrite steps, we know that validity cannot rely on the equality literals containing it. This means that we can safely remove both equality literals in the sequent tree so that they do not appear in (1.1) and (1.2). Removing redundant formulas is the most important simplification step besides contextual rewriting. This is impossible in tableau trees unless the redundancy of the formula is due to the ancestor nodes only, which is the case only for useless formulas that should not have been added at all.

Note that formulas like (nat1) from Section 1.1.1 make equality omnipresent in inductive theorem proving and that these simplification steps are even more important in inductive than in deductive theorem proving: Not only do they play a role in the generation of appropriate induction hypotheses; in addition to the detection of invalid input theorems they are an essential part of the failure detection process that has to compensate for *over-generalization* of induction hypotheses: Indeed, many induction proofs can only be successful when we try to show propositions that are more general than the ones we initially intended to show. This is because—in an induction proof—a proposition is not only a task (as a goal) but also a tool (as an induction hypothesis). This generalization is *unsafe* in the sense that it may over-generalize a valid hypothesis into an invalid one. Therefore, generalization should not be modeled in Expansion steps within a tree. Instead, the generalized sequent should start a new tree (Hypothesizing step) and be later applied to the original tree as a lemma or an induction hypothesis. Since even a valid input theorem may result in an invalid goal due to over-generalization, the ability of an inductive theorem proving system to detect invalid goals is of major importance in practice, cf. Section 2.4.2.

In Wirth (1997) and in QUODLIBET the Expansion from (1) into (1.1) and (1.2) is done in a single inference step called "substitution add" applying a "covering set of substitutions". Note that the state of the sequent proof resulting from this step is much simpler than the corresponding state of the tableau proof. The former consists of the nodes (1.1) and (1.2) and has two formulas and one variable. The latter consists of a six node tree with five formulas and two variables. This is of practical importance because tactics for proof search are more easily confused with less concise proof state representations. The rest of the whole sequent proof is analogous to the tableau proof with the exception that all rewrite steps of the tableau tree are omitted since there are no equality literals to rewrite with and the terms are already in normal form.

Another possibility restricted to sequent trees is that each weighted sequent labeling a node in the trees could be applied as an induction hypothesis. We do not see a real advantage in this because splitting the tree in two above such an induction hypothesis results in a better structure of the proof forest and in more successful proofs because we can adjust the weighted sequent appropriately:

Suppose we had not started a new proof tree for the hypothesis for Q but instead kept the hypothesis for Q down in the tree (1) at position (1.2.2). Several unsafe generalization steps would have been necessary before

$$\mathsf{Q}(x_1^{\delta^-}, \mathsf{s}(x_1^{\delta^-})),\ \neg\mathsf{P}(x_1^{\delta^-}),\ \mathsf{P}(\mathsf{s}(x_1^{\delta^-}));\ w_0^{\gamma}(\mathsf{s}(x_1^{\delta^-}))$$

would have become useful as an induction hypothesis, namely removing the second and third formula, generalizing $\mathsf{s}(x_1^{\delta^-})$ to a new variable, and switching to a weight that measures also this new variable.

Moreover, in practice one should not apply the hypothesis for Q in the tree for P before it is obvious that the tree for Q mutually needs the hypothesis for P: Most of the time a proof for Q can be completed in a proof forest not containing the tree for P. In this case, not only the number of trees in the proof forest for Q gets smaller, but also the tree for P because (2) can then be applied as a lemma and not as an induction hypothesis, which would cut off the rightmost $<$-branch of the proof tree of P.

3.3 An Example for Eager Hypotheses Generation

Let us try to find a lower bound for the Ackermann function $\mathsf{ack} : \mathsf{nat} \longrightarrow \mathsf{nat} \longrightarrow \mathsf{nat}$ w.r.t. the ordering on natural numbers $\mathsf{less} : \mathsf{nat} \longrightarrow \mathsf{nat} \longrightarrow \mathsf{bool}$, assuming the following axioms.

$(\mathsf{ack1})$ $\forall y.\ \ \mathsf{ack}(\mathsf{0}, y){=}\mathsf{s}(y)$
$(\mathsf{ack2})$ $\forall x, y.\ \ \mathsf{ack}(\mathsf{s}(x), \mathsf{0}){=}\mathsf{ack}(x, \mathsf{s}(\mathsf{0}))$
$(\mathsf{ack3})$ $\forall x, y.\ \ \mathsf{ack}(\mathsf{s}(x), \mathsf{s}(y)){=}\mathsf{ack}(x, \mathsf{ack}(\mathsf{s}(x), y))$

$(\mathsf{less1})$ $\forall y.\ \ \mathsf{less}(\mathsf{0}, \mathsf{s}(y)){=}\mathsf{true}$
$(\mathsf{less2})$ $\forall x.\ \ \mathsf{less}(x, \mathsf{0}){=}\mathsf{false}$
$(\mathsf{less3})$ $\forall x, y.\ \ \mathsf{less}(\mathsf{s}(x), \mathsf{s}(y)){=}\mathsf{less}(x, y)$

Standard lemmas for less proved automatically by QUODLIBET are:

$(\mathsf{less4})$ $\forall x.\ \ \mathsf{less}(x, \mathsf{s}(x))$

$(\mathsf{less5})$ $\forall x, y.\ \ \big(\ \mathsf{less}(x, y) \Rightarrow \mathsf{less}(x, \mathsf{s}(y))\ \big)$

$(\mathsf{less6})$ $\forall x, y.\ \ \big(\ \mathsf{less}(\mathsf{s}(x), y) \Rightarrow \mathsf{less}(x, y)\ \big)$

$(\mathsf{less7})$ $\forall x, y, z.\ \ \left(\left(\begin{array}{rl} & \mathsf{less}(x, y) \\ \wedge & \mathsf{less}(y, z) \end{array} \right) \Rightarrow \mathsf{less}(\mathsf{s}(x), z) \right)$

Note that for Boolean terms t we abbreviate the equation $t{=}\mathsf{true}$ with t. Moreover, note that $(\mathsf{less7})$ is a strengthened version of transitivity. The simple transitivity is a simple consequence of it, using $(\mathsf{less6})$.

Let us start with a Hypothesizing step in the sequent calculus of Definition 2.42, posing the query for a lower bound $z_0^\gamma : \mathsf{nat} \longrightarrow \mathsf{nat} \longrightarrow \mathsf{nat}$

(1) $\mathsf{less}(z_0^\gamma(x_0^{\delta^-}, y_0^{\delta^-}), \mathsf{ack}(x_0^{\delta^-}, y_0^{\delta^-}));\ \ w_1^\gamma(x_0^{\delta^-}, y_0^{\delta^-})$

with variable-condition $R := \{z_0^\gamma, w_1^\gamma\} \times \{x_0^{\delta^-}, y_0^{\delta^-}\}$.

Note that z_0^γ must be higher order: If z_0^γ were a first-order variable, it could not depend on $x_0^{\delta^-}$ and $y_0^{\delta^-}$ due to R, resulting in a constant lower bound, which would not be too interesting. If we did not include z_0^γ into $\mathrm{dom}(R)$, however, we could not do induction on the variables $x_0^{\delta^-}$ and $y_0^{\delta^-}$ because they would not be elements of the set Y of Theorem 2.51.

Applying $(\mathsf{nat1})$ (cf. Section 1.1.1) as a lemma yields the two goals

(1.1) $\mathsf{less}(z_0^\gamma(\mathsf{0}, y_0^{\delta^-}), \mathsf{ack}(\mathsf{0}, y_0^{\delta^-}));\ \ w_1^\gamma(\mathsf{0}, y_0^{\delta^-})$

(1.2) $\mathsf{less}(z_0^\gamma(\mathsf{s}(x_1^{\delta^-}), y_0^{\delta^-}), \mathsf{ack}(\mathsf{s}(x_1^{\delta^-}), y_0^{\delta^-}));\ \ w_1^\gamma(\mathsf{s}(x_1^{\delta^-}), y_0^{\delta^-})$

just as it was explained in Section 3.2.2, adding $\{z_0^{\gamma}, w_1^{\gamma}\} \times \{x_1^{\delta^-}\}$ to the variable-condition. The same procedure again yields

(1.2.1) $\mathsf{less}(z_0^{\gamma}(\mathsf{s}(x_1^{\delta^-}), \mathsf{0}), \mathsf{ack}(\mathsf{s}(x_1^{\delta^-}), \mathsf{0}));\ w_1^{\gamma}(\mathsf{s}(x_1^{\delta^-}), \mathsf{0})$

(1.2.2) $\mathsf{less}(z_0^{\gamma}(\mathsf{s}(x_1^{\delta^-}), \mathsf{s}(y_1^{\delta^-})), \mathsf{ack}(\mathsf{s}(x_1^{\delta^-}), \mathsf{s}(y_1^{\delta^-})));\ w_1^{\gamma}(\mathsf{s}(x_1^{\delta^-}), \mathsf{s}(y_1^{\delta^-}))$

adding $\{z_0^{\gamma}, w_1^{\gamma}\} \times \{y_1^{\delta^-}\}$ to the variable-condition.

Rewriting (1.1), (1.2.1), and (1.2.2) with (ack1), (ack2), and (ack3), resp., yields

(1.1.1) $\mathsf{less}(z_0^{\gamma}(\mathsf{0}, y_0^{\delta^-}), \mathsf{s}(y_0^{\delta^-}));\ w_1^{\gamma}(\mathsf{0}, y_0^{\delta^-})$

(1.2.1.1) $\mathsf{less}(z_0^{\gamma}(\mathsf{s}(x_1^{\delta^-}), \mathsf{0}), \mathsf{ack}(x_1^{\delta^-}, \mathsf{s}(\mathsf{0})));\ w_1^{\gamma}(\mathsf{s}(x_1^{\delta^-}), \mathsf{0})$

(1.2.2.1) $\mathsf{less}(z_0^{\gamma}(\mathsf{s}(x_1^{\delta^-}), \mathsf{s}(y_1^{\delta^-})), \mathsf{ack}(x_1^{\delta^-}, \mathsf{ack}(\mathsf{s}(x_1^{\delta^-}), y_1^{\delta^-})));\ w_1^{\gamma}(\mathsf{s}(x_1^{\delta^-}), \mathsf{s}(y_1^{\delta^-}))$

In our previous examples the generation of induction hypotheses was always lazy in the sense of Protzen (1994). In this case, however, to be able to use goal-directedness also w.r.t. the induction hypotheses, we should generate them eagerly in the way suggested by the recursion analysis of explicit induction, cf. Section 1.1.3. Recursion analysis and eager hypotheses generation are very useful for finding simple proofs completely automatically. Note that eager hypotheses generation is not possible with the induction rules of Baaz &al. (1997). Although the inductive theorem proving system NQTHM (cf. Boyer & Moore (1988)) cannot accept (1) because it does not have any free γ-variables (not even existential quantification), if we instantiate (1) with the proper lower bound, NQTHM proves (1) completely automatically, even when the lemma (less7) is not provided and the function 'less' is redefined so that the built-in features for treating arithmetic cannot help. Moreover, during this proof the fascinating NQTHM guesses (less7) completely automatically using the goal-directedness w.r.t. the eagerly generated induction hypotheses. Indeed, if the eagerly generated induction hypotheses happen to be the right ones, they can help us to find missing lemmas or to find proper instantiations for free γ-variables.

Since it is folklore heuristic knowledge in inductive theorem proving that a strong lower bound is often found by first finding a weaker one and then improving it, we should not look for an optimal lower bound with a difficult proof but for a reasonable lower bound with a simple proof.

In our example, the induction hypotheses suggested for (1.2.1.1) and (1.2.2.1) result from matching the ack-subterm of (1) to the ack-subterms of (1.2.1.1) and (1.2.2.1). For (1.2.1.1) we get the substitution $\{x_0^{\delta^-}{\mapsto}x_1^{\delta^-},\ y_0^{\delta^-}{\mapsto}\mathsf{s}(\mathsf{0})\}$ and for (1.2.2.1) the substitutions $\{x_0^{\delta^-}{\mapsto}x_1^{\delta^-},\ y_0^{\delta^-}{\mapsto}\mathsf{ack}(\mathsf{s}(x_1^{\delta^-}), y_1^{\delta^-})\}$ and $\{x_0^{\delta^-}{\mapsto}\mathsf{s}(x_1^{\delta^-}),\ y_0^{\delta^-}{\mapsto}y_1^{\delta^-}\}$ resulting in:

(1.2.1.1.1) $\neg\mathsf{less}(z_0^{\gamma}(x_1^{\delta^-}, \mathsf{s}(\mathsf{0})), \mathsf{ack}(x_1^{\delta^-}, \mathsf{s}(\mathsf{0}))),$
$\mathsf{less}(z_0^{\gamma}(\mathsf{s}(x_1^{\delta^-}), \mathsf{0}), \mathsf{ack}(x_1^{\delta^-}, \mathsf{s}(\mathsf{0})));\ w_1^{\gamma}(\mathsf{s}(x_1^{\delta^-}), \mathsf{0})$

(1.2.1.1.2) $w_1^{\gamma}(x_1^{\delta^-}, \mathsf{s}(\mathsf{0})) < w_1^{\gamma}(\mathsf{s}(x_1^{\delta^-}), \mathsf{0}),\ \mathsf{less}(z_0^{\gamma}(\mathsf{s}(x_1^{\delta^-}), \mathsf{0}), \mathsf{ack}(x_1^{\delta^-}, \mathsf{s}(\mathsf{0})));\ w_1^{\gamma}(\mathsf{s}(x_1^{\delta^-}), \mathsf{0})$

(1.2.2.1.1) $\neg\mathsf{less}(z_0^{\gamma}(x_1^{\delta^-}, \mathsf{ack}(\mathsf{s}(x_1^{\delta^-}), y_1^{\delta^-})), \mathsf{ack}(x_1^{\delta^-}, \mathsf{ack}(\mathsf{s}(x_1^{\delta^-}), y_1^{\delta^-}))),$
$\mathsf{less}(z_0^{\gamma}(\mathsf{s}(x_1^{\delta^-}), \mathsf{s}(y_1^{\delta^-})), \mathsf{ack}(x_1^{\delta^-}, \mathsf{ack}(\mathsf{s}(x_1^{\delta^-}), y_1^{\delta^-})));\ w_1^{\gamma}(\mathsf{s}(x_1^{\delta^-}), \mathsf{s}(y_1^{\delta^-}))$

(1.2.2.1.2) $w_1^{\gamma}(x_1^{\delta^-}, \mathsf{ack}(\mathsf{s}(x_1^{\delta^-}), y_1^{\delta^-})) < w_1^{\gamma}(\mathsf{s}(x_1^{\delta^-}), \mathsf{s}(y_1^{\delta^-})),\ \ldots$

(1.2.2.1.1.1) $\neg\mathsf{less}(z_0^{\gamma}(\mathsf{s}(x_1^{\delta^-}), y_1^{\delta^-}), \mathsf{ack}(\mathsf{s}(x_1^{\delta^-}), y_1^{\delta^-})),$
$\neg\mathsf{less}(z_0^{\gamma}(x_1^{\delta^-}, \mathsf{ack}(\mathsf{s}(x_1^{\delta^-}), y_1^{\delta^-})), \mathsf{ack}(x_1^{\delta^-}, \mathsf{ack}(\mathsf{s}(x_1^{\delta^-}), y_1^{\delta^-}))),$
$\mathsf{less}(z_0^{\gamma}(\mathsf{s}(x_1^{\delta^-}), \mathsf{s}(y_1^{\delta^-})), \mathsf{ack}(x_1^{\delta^-}, \mathsf{ack}(\mathsf{s}(x_1^{\delta^-}), y_1^{\delta^-})));\ w_1^{\gamma}(\mathsf{s}(x_1^{\delta^-}), \mathsf{s}(y_1^{\delta^-}))$

(1.2.2.1.1.2) $w_1^{\gamma}(\mathsf{s}(x_1^{\delta^-}), y_1^{\delta^-}) < w_1^{\gamma}(\mathsf{s}(x_1^{\delta^-}), \mathsf{s}(y_1^{\delta^-})),\ \ldots$

After setting $w_1^\gamma(x,y) := (x,y)$, the goals (1.2.1.1.2), (1.2.2.1.2), and (1.2.2.1.1.2) can be closed due to their first formulas. The whole proof up to now is the “eager induction hypotheses generation” suggested by recursion analysis of (1).

Now, (1.2.2.1.1.1) cries for a lemma application of $(\mathsf{less}7)$. Indeed, the lemma can close it, provided that we can identify the pairs $(\mathsf{s}(z_0^\gamma(\mathsf{s}(x_1^{\delta^-}), y_1^{\delta^-})), z_0^\gamma(\mathsf{s}(x_1^{\delta^-}), \mathsf{s}(y_1^{\delta^-})))$ and $(\mathsf{ack}(\mathsf{s}(x_1^{\delta^-}), y_1^{\delta^-}), z_0^\gamma(x_1^{\delta^-}, \mathsf{ack}(\mathsf{s}(x_1^{\delta^-}), y_1^{\delta^-})))$, which is achieved by their most general $\lambda\beta$-unifier, the projection $z_0^\gamma(x,y) := y$.

Now (1.1.1) reads

(1.1.1’) $\mathsf{less}(y_0^{\delta^-}, \mathsf{s}(y_0^{\delta^-}));\ (\mathsf{0}, y_0^{\delta^-})$

which can be closed by an application of lemma $(\mathsf{less}4)$.

The only branch that is still open is

(1.2.1.1.1’) $\neg\mathsf{less}(\mathsf{s}(\mathsf{0}), \mathsf{ack}(x_1^{\delta^-}, \mathsf{s}(\mathsf{0}))),\ \mathsf{less}(\mathsf{0}, \mathsf{ack}(x_1^{\delta^-}, \mathsf{s}(\mathsf{0})));\ (\mathsf{s}(x_1^{\delta^-}), \mathsf{0})$

which can be closed by an application of lemma $(\mathsf{less}6)$.

This completes the proof of (1) with the answer that z_0^γ can be the projection to its second argument, i.e. the lower bound is $y_0^{\delta^-}$.

Note that it is possible to find this proof with the first-order system QUODLIBET because one can use a symbol for an undefined function instead of the 2nd order variable z_0^γ. There is no 2nd order unification but the user can set this function to be the projection during the proof.

Since QUODLIBET guarantees consistency of the specification (i.e. the existence of models where semantical equality of constructor ground terms implies syntactical equality) (provided arithmetic is consistent, cf. Gentzen (1938)) and admits partially defined and non-terminating functions, the actual proof in QUODLIBET differs from the presented one by some additional subgoals that can be closed by a lemma stating that ack is a total function, which has a simple inductive proof. For the details cf. Kühler & Wirth (1996).

3.4 An Example for a Variable Induction Ordering

In this section we are going to prove a generalized version of a lemma of M. H. A. Newman, namely that local commutation of two relations implies their commutation, provided that the reverse of their union is wellfounded.

$\mathsf{0}$	:	nat
s	:	$\mathsf{nat} \rightarrow \mathsf{nat}$
$\mathsf{true},\ \mathsf{false}$	:	bool
$*,\ \mathsf{Rev}$	:	$(\mathsf{A} \rightarrow \mathsf{A} \rightarrow \mathsf{bool}) \rightarrow \mathsf{A} \rightarrow \mathsf{A} \rightarrow \mathsf{bool}$
Union	:	$(\mathsf{A} \rightarrow \mathsf{A} \rightarrow \mathsf{bool}) \rightarrow (\mathsf{A} \rightarrow \mathsf{A} \rightarrow \mathsf{bool}) \rightarrow \mathsf{A} \rightarrow \mathsf{A} \rightarrow \mathsf{bool}$
$\mathsf{Comm},\ \mathsf{LComm}$	:	$(\mathsf{A} \rightarrow \mathsf{A} \rightarrow \mathsf{bool}) \rightarrow (\mathsf{A} \rightarrow \mathsf{A} \rightarrow \mathsf{bool}) \rightarrow \mathsf{bool}$
Wellf	:	$(\mathsf{A} \rightarrow \mathsf{A} \rightarrow \mathsf{bool}) \rightarrow \mathsf{bool}$

Our simply-typed higher-order signature is used to denote the following: $*(\longrightarrow)$ contains the reflexive & transitive closure of the binary relation $\longrightarrow$ on A, $\mathsf{Rev}(\longrightarrow)$ is its reverse relation, and $\mathsf{Union}(\longrightarrow, \longrightarrow')$ is its union with $\longrightarrow'$. For all our Boolean terms t we abbreviate the equation $t{=}\mathsf{true}$ with t. For $\longrightarrow : \mathsf{A} \rightarrow \mathsf{A} \rightarrow \mathsf{bool}$, instead of $\longrightarrow(x, y)$ we write $x{\longrightarrow}y$, instead of $*(\longrightarrow, x, y)$ we write $x\overset{*}{\longrightarrow}y$, and instead of $\mathsf{Union}(\longrightarrow, \longrightarrow')$ we write $\longrightarrow \cup \longrightarrow'$.

$$(*1) \qquad \forall{\longrightarrow}, x, z.\ \left(\ x\overset{*}{\longrightarrow}z \Leftrightarrow \left(\begin{array}{ll} & x{=}z \\ \vee & \exists y.\ \left(\begin{array}{ll} & x{\longrightarrow}y \\ \wedge & y\overset{*}{\longrightarrow}z \end{array}\right) \end{array}\right)\ \right)$$

$$(\mathsf{Union1}) \qquad \forall{\longrightarrow}, {\longrightarrow'}, x, y.\ \left(\ x(\longrightarrow \cup \longrightarrow')y \Leftrightarrow \left(\begin{array}{ll} & x{\longrightarrow}y \\ \vee & x{\longrightarrow'}y \end{array}\right)\ \right)$$

$$(\mathsf{Rev1}) \qquad \forall{\longrightarrow}, x, y.\ \left(\ \mathsf{Rev}(\longrightarrow, x, y) \Leftrightarrow y{\longrightarrow}x\ \right)$$

$(\mathsf{Comm1})$, $(\mathsf{LComm1})$, and $(\mathsf{Wellf1})$ are the properties of commutation, local commutation, and wellfoundedness, resp.:

$$(\mathsf{Comm1}) \qquad \forall{\longrightarrow_0}, {\longrightarrow_1}.\ \left(\begin{array}{ll} & \mathsf{Comm}(\longrightarrow_0, \longrightarrow_1) \\ \Leftrightarrow & \forall x, y_0, y_1.\ \left(\begin{array}{ll} & \left(\begin{array}{ll} & x\overset{*}{\longrightarrow}_0 y_0 \\ \wedge & x\overset{*}{\longrightarrow}_1 y_1 \end{array}\right) \\ \Rightarrow & \exists z.\ \left(\begin{array}{ll} & y_0\overset{*}{\longrightarrow}_1 z \\ \wedge & y_1\overset{*}{\longrightarrow}_0 z \end{array}\right) \end{array}\right) \end{array}\right)$$

$$(\mathsf{LComm1}) \qquad \forall{\longrightarrow_0}, {\longrightarrow_1}.\ \left(\begin{array}{ll} & \mathsf{LComm}(\longrightarrow_0, \longrightarrow_1) \\ \Leftrightarrow & \forall x, y_0, y_1.\ \left(\begin{array}{ll} & \left(\begin{array}{ll} & x\longrightarrow_0 y_0 \\ \wedge & x\longrightarrow_1 y_1 \end{array}\right) \\ \Rightarrow & \exists z.\ \left(\begin{array}{ll} & y_0\overset{*}{\longrightarrow}_1 z \\ \wedge & y_1\overset{*}{\longrightarrow}_0 z \end{array}\right) \end{array}\right) \end{array}\right)$$

$$(\mathsf{Wellf1}) \qquad \forall r : \mathsf{A} \rightarrow \mathsf{A} \rightarrow \mathsf{bool}.$$

$$\left(\begin{array}{ll} & \mathsf{Wellf}(r) \\ \Leftrightarrow & \forall p : \mathsf{A} \rightarrow \mathsf{bool}.\ \left(\begin{array}{ll} & \exists x.\ p(x) \\ \Rightarrow & \exists x.\ \left(\begin{array}{ll} & p(x) \\ \wedge & \neg\exists y.\ \left(\begin{array}{ll} & p(y) \\ \wedge & r(y, x) \end{array}\right) \end{array}\right) \end{array}\right) \end{array}\right)$$

Note that wellfoundedness and termination are no first-order properties.[22]

The transitivity lemma

$$(1)\ \ u_0^{\delta^-}\overset{*}{\longrightarrow}{}^{\delta^-}u_2^{\delta^-},\ \neg u_0^{\delta^-}\overset{*}{\longrightarrow}{}^{\delta^-}u_1^{\delta^-},\ \neg u_1^{\delta^-}\overset{*}{\longrightarrow}{}^{\delta^-}u_2^{\delta^-},\ \neg\mathsf{Wellf}(\mathsf{Rev}(\longrightarrow^{\delta^-}));\ u_0^{\delta^-}$$

can be shown by induction on $u_0^{\delta^-}$ in $\longrightarrow^{\delta^-}$. Note that we need the wellfoundedness because otherwise $\overset{*}{\longrightarrow}{}^{\delta^-}$ may be a proper super-relation of the reflexive & transitive closure of $\longrightarrow^{\delta^-}$. I.e. the reflexive & transitive closure is the smallest solution of $(*1)$ and in case of wellfoundedness there is only one single solution.

The following lemmas have very simple non-inductive proofs that expand the definition $(\mathsf{Wellf1})$ twice. Note that the expansion of a logical equivalence is nothing but (γ-steps followed by) a kind of Rewrite-step because formulas can be seen as higher-order terms of type bool and the logical equivalence as the equality of type bool.

$$(2a)\ \ \neg\mathsf{Wellf}(\mathsf{Rev}(\longrightarrow_0^{\delta^-}\cup\longrightarrow_1^{\delta^-})),\ \mathsf{Wellf}(\mathsf{Rev}(\longrightarrow_0^{\delta^-}))$$

$$(2b)\ \ \neg\mathsf{Wellf}(\mathsf{Rev}(\longrightarrow_0^{\delta^-}\cup\longrightarrow_1^{\delta^-})),\ \mathsf{Wellf}(\mathsf{Rev}(\longrightarrow_1^{\delta^-}))$$

Note that commutativity of Union implies that (2a) and (2b) are equivalent, but to prove $\longrightarrow_0^{\delta^-}\cup\longrightarrow_1^{\delta^-} = \longrightarrow_1^{\delta^-}\cup\longrightarrow_0^{\delta^-}$ we need extensionality, which we do not want to discuss here. Cf. Benzmüller &al. (2004) for a comprehensive discussion of extensionality.
Now we are going to show the generalized Newman lemma, namely that wellfoundedness of the reverse of the union plus local commutation implies commutation.

$$\neg\mathsf{Wellf}(\mathsf{Rev}(\longrightarrow_0^{\delta^-}\cup\longrightarrow_1^{\delta^-})),\ \neg\mathsf{LComm}(\longrightarrow_0^{\delta^-},\longrightarrow_1^{\delta^-}),\ \mathsf{Comm}(\longrightarrow_0^{\delta^-},\longrightarrow_1^{\delta^-})$$

Expanding the definition $(\mathsf{Comm1})$, three liberalized δ-steps, and two α-steps yield

$$\neg x^{\delta^+}\overset{*}{\longrightarrow}{}_0^{\delta^-}z_0^{\delta^+},\ \neg x^{\delta^+}\overset{*}{\longrightarrow}{}_1^{\delta^-}z_1^{\delta^+},\ \exists z.\left(\begin{array}{cc} & z_0^{\delta^+}\overset{*}{\longrightarrow}{}_1^{\delta^-}z \\ \wedge & z_1^{\delta^+}\overset{*}{\longrightarrow}{}_0^{\delta^-}z\end{array}\right),$$
$$\neg\mathsf{Wellf}(\mathsf{Rev}(\longrightarrow_0^{\delta^-}\cup\longrightarrow_1^{\delta^-})),\ \neg\mathsf{LComm}(\longrightarrow_0^{\delta^-},\longrightarrow_1^{\delta^-})$$

Now, since we want to do induction on x^{δ^+}, we start a new proof tree for

$$(3)\ \ \neg x^{\delta^-}\overset{*}{\longrightarrow}{}_0^{\delta^-}z_0^{\delta^-},\ \neg x^{\delta^-}\overset{*}{\longrightarrow}{}_1^{\delta^-}z_1^{\delta^-},\ \exists z.\left(\begin{array}{cc} & z_0^{\delta^-}\overset{*}{\longrightarrow}{}_1^{\delta^-}z \\ \wedge & z_1^{\delta^-}\overset{*}{\longrightarrow}{}_0^{\delta^-}z\end{array}\right),$$
$$\neg\mathsf{Wellf}(\mathsf{Rev}(\longrightarrow_0^{\delta^-}\cup\longrightarrow_1^{\delta^-})),\ \neg\mathsf{LComm}(\longrightarrow_0^{\delta^-},\longrightarrow_1^{\delta^-});$$
$$x^{\delta^-},\ <^{\gamma}(x^{\delta^-},z_0^{\delta^-},z_1^{\delta^-},\longrightarrow_0^{\delta^-},\longrightarrow_1^{\delta^-})$$

Note that this differs from the previous sequent (which can immediately be closed by lemma application of (3)) not only in that all free δ^+-variables are replaced with free δ^--variables now (which we also could have achieved by using non-liberalized δ-steps before instead of the liberalized ones) but also in that x^{δ^-} is included into the weight, which is necessary for our intended induction. Actually we have set the weight directly to x^{δ^-} for simplicity. Note that if the heuristic knowledge to recognize the above sequent as the likely induction hypothesis is not present, our calculi violate our design goal of a natural flow of information (cf. Section 1.2.1) because we sometime later realize that we should have started a new proof tree. With implemented calculi, however, this violation is no problem because one just has to implement a destructive inference rule that automatically splits a proof tree at a given position, reorganizes the former subtree into a new individual proof tree, and closes the cut branch by lemma or induction hypothesis application of the sequent of the new tree.

Moreover, we have added a free γ-variable for the induction ordering

$$<^{\gamma} : \mathsf{A} \rightarrow \mathsf{A} \rightarrow \mathsf{A} \rightarrow (\mathsf{A} \rightarrow \mathsf{A} \rightarrow \mathsf{bool}) \rightarrow (\mathsf{A} \rightarrow \mathsf{A} \rightarrow \mathsf{bool}) \rightarrow \mathsf{A} \rightarrow \mathsf{A} \rightarrow \mathsf{bool}$$

where the last two arguments will be supplied in infix notation below. Note that we have not supplied any induction quasi-ordering, but instead assume it to be the empty relation as in the discussion after Theorem 2.51, so that the sequents (5) and (6) can be omitted from the set M in Theorem 2.51. We set our variable-condition to $R := \{<^{\gamma}\} \times \{x^{\delta^-}, z_0^{\delta^-}, z_1^{\delta^-}, \longrightarrow_0^{\delta^-}, \longrightarrow_1^{\delta^-}\}$ to have all free δ^--variables of (3) in the set Y of Theorem 2.51. Expansion of the equivalence $(*1)$ in the first formula of (3), a β-, a liberalized δ- and an α-step yield:

$$\text{(3.1)}\ \ x^{\delta^-} \neq z_0^{\delta^-},\ \neg x^{\delta^-} \overset{*}{\longrightarrow}{}_1^{\delta^-} z_1^{\delta^-},\ \exists z. \left(\begin{array}{rl} & z_0^{\delta^-} \overset{*}{\longrightarrow}{}_1^{\delta^-} z \\ \wedge & z_1^{\delta^-} \overset{*}{\longrightarrow}{}_0^{\delta^-} z \end{array} \right),\ \ \ldots;\ \ldots$$

$$\begin{array}{l} \text{(3.2)}\ \ \neg x^{\delta^-} \longrightarrow_0^{\delta^-} y_0^{\delta^+},\ \neg y_0^{\delta^+} \overset{*}{\longrightarrow}{}_0^{\delta^-} z_0^{\delta^-},\ \neg x^{\delta^-} \overset{*}{\longrightarrow}{}_1^{\delta^-} z_1^{\delta^-},\ \exists z. \left(\begin{array}{rl} & z_0^{\delta^-} \overset{*}{\longrightarrow}{}_1^{\delta^-} z \\ \wedge & z_1^{\delta^-} \overset{*}{\longrightarrow}{}_0^{\delta^-} z \end{array} \right), \\ \qquad\qquad\qquad \neg\mathsf{Wellf}(\mathsf{Rev}(\longrightarrow_0^{\delta^-} \cup \longrightarrow_1^{\delta^-})),\ \neg\mathsf{LComm}(\longrightarrow_0^{\delta^-}, \longrightarrow_1^{\delta^-}); \\ \hfill x^{\delta^-},\ <^{\gamma}(x^{\delta^-}, z_0^{\delta^-}, z_1^{\delta^-}, \longrightarrow_0^{\delta^-}, \longrightarrow_1^{\delta^-}) \end{array}$$

Rewriting with the first formula of (3.1) yields:

$$\text{(3.1.1)}\ \ \neg x^{\delta^-} \overset{*}{\longrightarrow}{}_1^{\delta^-} z_1^{\delta^-},\ \exists z. \left(\begin{array}{rl} & x^{\delta^-} \overset{*}{\longrightarrow}{}_1^{\delta^-} z \\ \wedge & z_1^{\delta^-} \overset{*}{\longrightarrow}{}_0^{\delta^-} z \end{array} \right),\ \ \ldots;\ \ldots$$

which is easily proved by setting z to $z_1^{\delta^-}$ in a γ-step. Expansion of the equivalence $(*1)$ in the third formula of (3.2), a β-, a liberalized δ- and an α-step yield:

$$\text{(3.2.1)}\ \ x^{\delta^-} \neq z_1^{\delta^-},\ \neg x^{\delta^-} \longrightarrow_0^{\delta^-} y_0^{\delta^+},\ \neg y_0^{\delta^+} \overset{*}{\longrightarrow}{}_0^{\delta^-} z_0^{\delta^-},\ \exists z. \left(\begin{array}{rl} & z_0^{\delta^-} \overset{*}{\longrightarrow}{}_1^{\delta^-} z \\ \wedge & z_1^{\delta^-} \overset{*}{\longrightarrow}{}_0^{\delta^-} z \end{array} \right),\ \ \ldots;\ \ldots$$

$$\begin{array}{l} \text{(3.2.2)}\ \ \neg x^{\delta^-} \longrightarrow_1^{\delta^-} y_1^{\delta^+},\ \neg y_1^{\delta^+} \overset{*}{\longrightarrow}{}_1^{\delta^-} z_1^{\delta^-},\ \neg x^{\delta^-} \longrightarrow_0^{\delta^-} y_0^{\delta^+},\ \neg y_0^{\delta^+} \overset{*}{\longrightarrow}{}_0^{\delta^-} z_0^{\delta^-}, \\ \qquad \exists z. \left(\begin{array}{rl} & z_0^{\delta^-} \overset{*}{\longrightarrow}{}_1^{\delta^-} z \\ \wedge & z_1^{\delta^-} \overset{*}{\longrightarrow}{}_0^{\delta^-} z \end{array} \right),\ \neg\mathsf{Wellf}(\mathsf{Rev}(\longrightarrow_0^{\delta^-} \cup \longrightarrow_1^{\delta^-})),\ \neg\mathsf{LComm}(\longrightarrow_0^{\delta^-}, \longrightarrow_1^{\delta^-}); \\ \hfill x^{\delta^-},\ <^{\gamma}(x^{\delta^-}, z_0^{\delta^-}, z_1^{\delta^-}, \longrightarrow_0^{\delta^-}, \longrightarrow_1^{\delta^-}) \end{array}$$

Rewriting with the first formula of (3.2.1) yields:

$$\text{(3.2.1.1)}\ \ \neg x^{\delta^-} \longrightarrow_0^{\delta^-} y_0^{\delta^+},\ \neg y_0^{\delta^+} \overset{*}{\longrightarrow}{}_0^{\delta^-} z_0^{\delta^-},\ \exists z. \left(\begin{array}{rl} & z_0^{\delta^-} \overset{*}{\longrightarrow}{}_1^{\delta^-} z \\ \wedge & x^{\delta^-} \overset{*}{\longrightarrow}{}_0^{\delta^-} z \end{array} \right),\ \ \ldots;\ \ldots$$

Now we have to regenerate the literal $\neg x^{\delta^-} \overset{*}{\longrightarrow}{}_0^{\delta^-} z_0^{\delta^-}$ (which a tableau proof would still have available from (3)) by application of $(*1)$ and then close this subtree by setting z to $z_0^{\delta^-}$.
Expansion of $(\mathsf{LComm1})$ in (3.2.2), γ-, β- and liberalized δ-steps yield two tautologies plus

$$\begin{aligned}
&(3.2.2.1)\ \ \neg y_0^{\delta+}\overset{*}{\longrightarrow}{}^{\delta-}_{1}\,y_2^{\delta+},\ \ \neg y_1^{\delta+}\overset{*}{\longrightarrow}{}^{\delta-}_{0}\,y_2^{\delta+},\\
&\qquad\neg x^{\delta-}\longrightarrow{}^{\delta-}_{1}\,y_1^{\delta+},\ \ \neg y_1^{\delta+}\overset{*}{\longrightarrow}{}^{\delta-}_{1}\,z_1^{\delta-},\ \ \neg x^{\delta-}\longrightarrow{}^{\delta-}_{0}\,y_0^{\delta+},\ \ \neg y_0^{\delta+}\overset{*}{\longrightarrow}{}^{\delta-}_{0}\,z_0^{\delta-},\\
&\qquad\exists z.\left(\begin{array}{ll} & z_0^{\delta-}\overset{*}{\longrightarrow}{}^{\delta-}_{1}\,z\\ \wedge & z_1^{\delta-}\overset{*}{\longrightarrow}{}^{\delta-}_{0}\,z\end{array}\right),\ \ \neg\mathsf{Wellf}(\mathsf{Rev}(\longrightarrow{}^{\delta-}_{0}\cup\longrightarrow{}^{\delta-}_{1})),\ \ \neg\mathsf{LComm}(\longrightarrow{}^{\delta-}_{0},\longrightarrow{}^{\delta-}_{1});\\
&\qquad\qquad x^{\delta-},\ \ <^{\gamma}(x^{\delta-},z_0^{\delta-},z_1^{\delta-},\longrightarrow{}^{\delta-}_{0},\longrightarrow{}^{\delta-}_{1})
\end{aligned}$$

Applying (3) as an induction hypothesis according to Theorem 2.51 with substitution $\{x^{\delta-}\mapsto y_0^{\delta+},\ z_1^{\delta-}\mapsto y_2^{\delta+}\}$ yields four tautologies and

$$\begin{aligned}
&(3.2.2.1.1)\ \ \neg z_0^{\delta-}\overset{*}{\longrightarrow}{}^{\delta-}_{1}\,y_3^{\delta+},\ \ \neg y_2^{\delta+}\overset{*}{\longrightarrow}{}^{\delta-}_{0}\,y_3^{\delta+},\ \ \neg y_0^{\delta+}\overset{*}{\longrightarrow}{}^{\delta-}_{1}\,y_2^{\delta+},\ \ \neg y_1^{\delta+}\overset{*}{\longrightarrow}{}^{\delta-}_{0}\,y_2^{\delta+},\\
&\qquad\neg x^{\delta-}\longrightarrow{}^{\delta-}_{1}\,y_1^{\delta+},\ \ \neg y_1^{\delta+}\overset{*}{\longrightarrow}{}^{\delta-}_{1}\,z_1^{\delta-},\ \ \neg x^{\delta-}\longrightarrow{}^{\delta-}_{0}\,y_0^{\delta+},\ \ \neg y_0^{\delta+}\overset{*}{\longrightarrow}{}^{\delta-}_{0}\,z_0^{\delta-},\\
&\qquad\exists z.\left(\begin{array}{ll} & z_0^{\delta-}\overset{*}{\longrightarrow}{}^{\delta-}_{1}\,z\\ \wedge & z_1^{\delta-}\overset{*}{\longrightarrow}{}^{\delta-}_{0}\,z\end{array}\right),\ \ \neg\mathsf{Wellf}(\mathsf{Rev}(\longrightarrow{}^{\delta-}_{0}\cup\longrightarrow{}^{\delta-}_{1})),\ \ \neg\mathsf{LComm}(\longrightarrow{}^{\delta-}_{0},\longrightarrow{}^{\delta-}_{1});\\
&\qquad\qquad x^{\delta-},\ \ <^{\gamma}(x^{\delta-},z_0^{\delta-},z_1^{\delta-},\longrightarrow{}^{\delta-}_{0},\longrightarrow{}^{\delta-}_{1})
\end{aligned}$$

$$(3.2.2.1.2)\ \ y_0^{\delta+}<^{\gamma}(x^{\delta-},z_0^{\delta-},z_1^{\delta-},\longrightarrow{}^{\delta-}_{0},\longrightarrow{}^{\delta-}_{1})\ x^{\delta-},\ \ \ldots,\ \ \neg x^{\delta-}\longrightarrow{}^{\delta-}_{0}\,y_0^{\delta+},\ \ \ldots$$

$$\begin{aligned}
&(3.2.2.1.3)\ \ \forall p:\mathsf{A}\to\mathsf{bool}.\\
&\qquad\left(\exists x.\ p(x)\Rightarrow\exists x.\left(\begin{array}{ll} & p(x)\\ \wedge & \neg\exists y.\left(\begin{array}{ll} & p(y)\\ \wedge & y<^{\gamma}(x^{\delta-},z_0^{\delta-},z_1^{\delta-},\longrightarrow{}^{\delta-}_{0},\longrightarrow{}^{\delta-}_{1})\ x\end{array}\right)\end{array}\right)\right),\\
&\qquad\qquad\ldots,\ \neg\mathsf{Wellf}(\mathsf{Rev}(\longrightarrow{}^{\delta-}_{0}\cup\longrightarrow{}^{\delta-}_{1})),\ \ \ldots
\end{aligned}$$

$$(3.2.2.1.4)\ \ \forall x,y:\mathsf{A}.\left(\begin{array}{ll} & x<^{\gamma}(x^{\delta-},z_0^{\delta-},z_1^{\delta-},\longrightarrow{}^{\delta-}_{0},\longrightarrow{}^{\delta-}_{1})\ y\\ \Leftrightarrow & x<^{\gamma}(y_0^{\delta+},z_0^{\delta-},y_2^{\delta+},\longrightarrow{}^{\delta-}_{0},\longrightarrow{}^{\delta-}_{1})\ y\end{array}\right),\ \ \ldots$$

where (3.2.2.1.1) is presented after application of a liberalized δ- and an α-step. The situation of the first two lines of (3.2.2.1.1) (seen as an antecedent) can be depicted as follows:

$$\begin{tikzcd}
x^{\delta-} \arrow[r, "0"'] \arrow[d, "1"'] & y_0^{\delta+} \arrow[r, "*", "0"'] \arrow[d, "1"', "*"] & z_0^{\delta-} \arrow[d, "1"', "*"] \\
y_1^{\delta+} \arrow[r, "*", "0"'] \arrow[d, "1"', "*"] & y_2^{\delta+} \arrow[r, "*", "0"'] & y_3^{\delta+} \\
z_1^{\delta-} & &
\end{tikzcd}$$

Application of (1) as a lemma yields (besides a sequent that can be closed by lemma application of (2a))

$$\begin{aligned}
&(3.2.2.1.1.1)\ \ \neg y_1^{\delta+}\overset{*}{\longrightarrow}{}^{\delta-}_{0}\,y_3^{\delta+},\ \ \neg z_0^{\delta-}\overset{*}{\longrightarrow}{}^{\delta-}_{1}\,y_3^{\delta+},\ \ \neg y_2^{\delta+}\overset{*}{\longrightarrow}{}^{\delta-}_{0}\,y_3^{\delta+},\ \ \neg y_0^{\delta+}\overset{*}{\longrightarrow}{}^{\delta-}_{1}\,y_2^{\delta+},\ \ \neg y_1^{\delta+}\overset{*}{\longrightarrow}{}^{\delta-}_{0}\,y_2^{\delta+},\\
&\qquad\neg x^{\delta-}\longrightarrow{}^{\delta-}_{1}\,y_1^{\delta+},\ \ \neg y_1^{\delta+}\overset{*}{\longrightarrow}{}^{\delta-}_{1}\,z_1^{\delta-},\ \ \neg x^{\delta-}\longrightarrow{}^{\delta-}_{0}\,y_0^{\delta+},\ \ \neg y_0^{\delta+}\overset{*}{\longrightarrow}{}^{\delta-}_{0}\,z_0^{\delta-},\\
&\qquad\exists z.\left(\begin{array}{ll} & z_0^{\delta-}\overset{*}{\longrightarrow}{}^{\delta-}_{1}\,z\\ \wedge & z_1^{\delta-}\overset{*}{\longrightarrow}{}^{\delta-}_{0}\,z\end{array}\right),\ \ \neg\mathsf{Wellf}(\mathsf{Rev}(\longrightarrow{}^{\delta-}_{0}\cup\longrightarrow{}^{\delta-}_{1})),\ \ \neg\mathsf{LComm}(\longrightarrow{}^{\delta-}_{0},\longrightarrow{}^{\delta-}_{1});\\
&\qquad\qquad x^{\delta-},\ \ <^{\gamma}(x^{\delta-},z_0^{\delta-},z_1^{\delta-},\longrightarrow{}^{\delta-}_{0},\longrightarrow{}^{\delta-}_{1})
\end{aligned}$$

Applying (3) as an induction hypothesis with substitution $\{x^{\delta-}\mapsto y_1^{\delta+},\ z_0^{\delta-}\mapsto y_3^{\delta+}\}$ yields four tautologies and

(3.2.2.1.1.1.1) $\neg z_0^{\delta^-} \overset{*}{\longrightarrow}{}_1^{\delta^-} y_4^{\delta^+},\ \neg z_1^{\delta^-} \overset{*}{\longrightarrow}{}_0^{\delta^-} y_4^{\delta^+},\ \neg y_3^{\delta^+} \overset{*}{\longrightarrow}{}_1^{\delta^-} y_4^{\delta^+},$
$\neg y_1^{\delta^+} \overset{*}{\longrightarrow}{}_0^{\delta^-} y_3^{\delta^+},\ \neg z_0^{\delta^-} \overset{*}{\longrightarrow}{}_1^{\delta^-} y_3^{\delta^+},\ \ldots,\ \exists z. \left(\begin{array}{ll} & z_0^{\delta^-} \overset{*}{\longrightarrow}{}_1^{\delta^-} z \\ \wedge & z_1^{\delta^-} \overset{*}{\longrightarrow}{}_0^{\delta^-} z \end{array}\right),\ \ldots$

(3.2.2.1.1.1.2) $y_1^{\delta^+} <^{\gamma}(x^{\delta^-}, z_0^{\delta^-}, z_1^{\delta^-}, \longrightarrow_0^{\delta^-}, \longrightarrow_1^{\delta^-})\ x^{\delta^-},\ \ldots,\ \neg x^{\delta^-} \longrightarrow_1^{\delta^-} y_1^{\delta^+},\ \ldots$

(3.2.2.1.1.1.3) $\forall p : \mathsf{A} \rightarrow \mathsf{bool}.$

$$\left(\begin{array}{l} \exists x.\ p(x) \Rightarrow \exists x. \left(\begin{array}{ll} & p(x) \\ \wedge & \neg\exists y. \left(\begin{array}{ll} & p(y) \\ \wedge & y <^{\gamma}(x^{\delta^-}, z_0^{\delta^-}, z_1^{\delta^-}, \longrightarrow_0^{\delta^-}, \longrightarrow_1^{\delta^-})\ x \end{array}\right) \end{array}\right) \end{array}\right),$$

$\ldots,\ \neg\mathsf{Wellf}(\mathsf{Rev}(\longrightarrow_0^{\delta^-} \cup \longrightarrow_1^{\delta^-})),\ \ldots$

(3.2.2.1.1.1.4) $\forall x, y : \mathsf{A}. \left(\begin{array}{ll} & x <^{\gamma}(x^{\delta^-}, z_0^{\delta^-}, z_1^{\delta^-}, \longrightarrow_0^{\delta^-}, \longrightarrow_1^{\delta^-})\ y \\ \Leftrightarrow & x <^{\gamma}(y_1^{\delta^+}, y_3^{\delta^+}, z_1^{\delta^-}, \longrightarrow_0^{\delta^-}, \longrightarrow_1^{\delta^-})\ y \end{array}\right),\ \ldots$

where (3.2.2.1.1.1.1) is presented after application of a liberalized δ- and an α-step, whose resulting sequent's antecedent can be depicted as

$$\begin{tikzcd}
x^{\delta^-} \arrow[r, "0"'] \arrow[d, "1"'] & y_0^{\delta^+} \arrow[r, "*", "0"'] & z_0^{\delta^-} \arrow[d, "1"', "*"] \\
y_1^{\delta^+} \arrow[rr, "*", "0"'] \arrow[d, "1"', "*"] & & y_3^{\delta^+} \arrow[d, "1"', "*"] \\
z_1^{\delta^-} \arrow[rr, "*", "0"'] & & y_4^{\delta^+}
\end{tikzcd}$$

and also after a lemma application of the transitivity lemma (1) (producing another goal closed by lemma application of (2b)).

Now (3.2.2.1.1.1.1) can be closed after setting z to $y_4^{\delta^+}$ in a γ-step. When we finally apply the R-substitution $\{<^{\gamma} \mapsto \lambda v_0, \ldots, v_4.\ (\mathsf{Rev}(v_3 \cup v_4))\}$ and $\lambda\beta$-reduce, we get the following open goals:

(3.2.2.1.2') $\mathsf{Rev}(\longrightarrow_0^{\delta^-} \cup \longrightarrow_1^{\delta^-}, y_0^{\delta^+}, x^{\delta^-}),\ \ldots,\ \neg x^{\delta^-} \longrightarrow_0^{\delta^-} y_0^{\delta^+},\ \ldots$

(3.2.2.1[.1.1].3') $\forall p : \mathsf{A} \rightarrow \mathsf{bool}.$

$$\left(\begin{array}{l} \exists x.\ p(x) \Rightarrow \exists x. \left(\begin{array}{ll} & p(x) \\ \wedge & \neg\exists y. \left(\begin{array}{ll} & p(y) \\ \wedge & \mathsf{Rev}(\longrightarrow_0^{\delta^-} \cup \longrightarrow_1^{\delta^-}, y, x) \end{array}\right) \end{array}\right) \end{array}\right),$$

$\ldots,\ \neg\mathsf{Wellf}(\mathsf{Rev}(\longrightarrow_0^{\delta^-} \cup \longrightarrow_1^{\delta^-})),\ \ldots$

(3.2.2.1[.1.1].4') $\forall x, y : \mathsf{A}. \left(\begin{array}{ll} & \mathsf{Rev}(\longrightarrow_0^{\delta^-} \cup \longrightarrow_1^{\delta^-}, x, y) \\ \Leftrightarrow & \mathsf{Rev}(\longrightarrow_0^{\delta^-} \cup \longrightarrow_1^{\delta^-}, x, y) \end{array}\right),\ \ldots$

(3.2.2.1.1.1.2') $\mathsf{Rev}(\longrightarrow_0^{\delta^-} \cup \longrightarrow_1^{\delta^-}, y_1^{\delta^+}, x^{\delta^-}),\ \ldots,\ \neg x^{\delta^-} \longrightarrow_1^{\delta^-} y_1^{\delta^+},\ \ldots$

which can be easily closed.

4 Conclusion

We have shown how to integrate *descente infinie* in the style of a working mathematician into state-of-the-art free-variable sequent and tableau calculi which are well-suited for an efficient interplay of human interaction and automation. The semantical requirements are satisfied for a variety of two-valued logics, such as clausal logic, classical first-order logic, and higher-order modal logic.

For the special case of first-order universally quantified clausal logic we have realized this style of inductive theorem proving in the tactic-based inductive theorem prover QUODLIBET, cf. Wirth (1997), Kühler (2000), Avenhaus &al. (2003). The extension of QUODLIBET's approach to full first-order logic, however, turned out to be more difficult than expected, because the standard state-of-the-art free-variable first-order sequent and tableau calculi destroy the wellfoundedness of *descente infinie*. The foundational problems that ensued from the combination of *descente infinie* with these calculi are solved now for the first time by our technique of combining the liberalized δ-rules with

- raising (instead of Skolemization),
- an explicit representation of variable dependencies, and
- a preservation of solutions (i.e. closing substitutions).

Lemma and induction-hypothesis application are included for the first time in all calculi treated in this paper: Wirth (1999) included only hypothesis application for the weak version (i.e. without free δ^+-variables and liberalized δ-rules) of the calculi of Wirth (1998). To apply lemmas and induction hypotheses freely and easily also in the strong version, we surprisingly had to change the notion of (C, R)-validity, cf. Note 12. With this exception and besides minor improvements, the calculi of this paper are the ones of the strong version of Wirth (1998) with the free δ^--variables and the non-liberalized δ-rules of the weak version added.

Our comprehensive (or "fat") integration of *descente infinie* differs from the "lean" calculus of Baaz &al. (1997) in the following aspects: We can have mutual induction and variable induction orderings. Our induction hypotheses can be arbitrary sequents instead of a single preset literal. Finally, we can also generate induction hypotheses eagerly in the style of explicit induction, which enables goal-directedness w.r.t. induction hypotheses. Indeed, in our framework all the heuristic knowledge and automatization of the field of explicit induction is still applicable and indispensable.[23] We have just opened a door to a new formal basis that provides the flexibility and the support a mathematician needs when he searches for hard induction proofs.

Acknowledgments: After three years of solid—and solitary—work on this paper, when I was not able to see the wood for the trees, Jörg H. Siekmann showed me a path out of the wilderness. I owe more to Jörg than I can express here. I would also like to thank Ulrich Kühler for the development of the QUODLIBET system, and Paul Howard for a short but very helpful e-mail communication on the Principle of Dependent Choice.

A Optimizations

A.1 The Low Price and High Value of Choice-Conditions

Note that (as far as Theorem 2.49 and Theorem 2.51 are concerned) the choice-conditions do not have any influence on our proofs as long as we never instantiate free δ^+-variables and always choose a completely new free δ^+-variable x^{δ^+} in the liberalized δ-steps. Thus, in implementations of our calculus, the choice-conditions may be omitted. We could, however, use them for the following purposes:

1. We can use the choice-conditions to weaken our requirements for our set of axioms $\mathcal{AX}$: Instead of $\mathrm{V}_\gamma{\times}\mathrm{V}_\delta$-validity of $\mathcal{AX}$, (C, R)-validity of $\mathcal{AX}$ (which is logically weaker, cf. Lemma 2.28) is sufficient for Theorem 2.45.

2. We can simulate the behavior of an improved version of the δ^{+^+}-rule of Beckert &al. (1993) by equating different free δ^+-variables whose C-values are initially equal or have become logically equivalent during the proof. Note that this does not anymore require a functional and extensional behavior of choice-conditions as in Wirth (1998). There we had to require that, for $(x^{\delta^+}, A) \in C$, the value for x^{δ^+} is not just an arbitrary one from the set of values that make A valid, but a unique element of this set given by some choice-function. In the present version (due to the changed notion of (C, R)-validity) it is possible to globally replace not only a free γ-variable, but also a free δ^+-variable x^{δ^+} with any term that (if possible) makes $C(x^{\delta^+})$ true, cf. Section B.3.
   Expressed with Hilbert's ε-terms (as indicated in Section 2.2.4), our treatment is similar to a structure sharing version of the merely intensional treatment of ε-terms in Giese & Ahrendt (1999). Note that our choice-conditions even do not imply a functional dependence of $\epsilon(\pi)(\tau)(y^{\delta^+})$ from $C(y^{\delta^+})$; instead the choice of a special value is a step in a proof similar to the instantiation of a free γ-variable, and we do not have to commit to this choice for other occurrences of the same ε-term. This means that our choice-conditions work like the word "some" in the in the English language. E.g., "Some human loves some human" is like $\mathsf{Loves}(x^{\delta^+}, y^{\delta^+})$ with $C(x^{\delta^+}) = \mathsf{Human}(x^{\delta^+})$ and $C(y^{\delta^+}) = \mathsf{Human}(y^{\delta^+})$, or like

   $$\mathsf{Loves}(\varepsilon x.\ \mathsf{Human}(x),\ \varepsilon x.\ \mathsf{Human}(x))$$

   and follows from $\mathsf{Loves}(\mathsf{Jack}, \mathsf{Jill})$, $\mathsf{Human}(\mathsf{Jack})$, and $\mathsf{Human}(\mathsf{Jill})$. There is more on this subject in Wirth (2002).

3. Moreover, the choice-conditions may be used to get more interesting solutions to query variables, as explained in the following example.

EXAMPLE A.1
Starting with the empty proof forest and hypothesizing

$$\forall x.\ \mathsf{Q}(x,x),\quad \exists y.\ \big(\ \neg\mathsf{Q}(y,y)\wedge\neg\mathsf{P}(y)\ \big),\quad \mathsf{P}(z^{\gamma})$$

with the rules at the end of Section 1.2.2 we can produce a proof tree with the leaves

$$\neg\mathsf{Q}(x^{\delta^+},x^{\delta^+}),\quad \mathsf{Q}(x^{\delta^+},x^{\delta^+}),\quad \exists y.\ \big(\ \neg\mathsf{Q}(y,y)\wedge\neg\mathsf{P}(y)\ \big),\quad \mathsf{P}(z^{\gamma})$$

and

$$\neg\mathsf{P}(x^{\delta^+}),\quad \mathsf{Q}(x^{\delta^+},x^{\delta^+}),\quad \exists y.\ \big(\ \neg\mathsf{Q}(y,y)\wedge\neg\mathsf{P}(y)\ \big),\quad \mathsf{P}(z^{\gamma})$$

and the $\emptyset$-choice-condition $\{(x^{\delta^+},\neg\mathsf{Q}(x^{\delta^+},x^{\delta^+}))\}$.

The $\emptyset$-substitution $\{z^{\gamma}\mapsto x^{\delta^+}\}$ closes the proof tree via an Instantiation step. The solution x^{δ^+} for our query variable z^{γ} is not very interesting unless the choice-condition tells us to choose x^{δ^+} in such a way that $\mathsf{Q}(x^{\delta^+},x^{\delta^+})$ becomes false.

Note that if we had applied the δ-rule instead of the liberalized δ-rule in the above proof, i.e. if we had introduced x^{δ^-} instead of x^{δ^+}, then we would not only be unable to provide any information on our query variable (because the choice-condition is empty), but we would even be unable to finish our proof because—due to the new variable-condition $R = \{(z^{\gamma},x^{\delta^-})\}$—we cannot apply $\{z^{\gamma}\mapsto x^{\delta^-}\}$ anymore, because it is not an R-substitution. With the δ^--rule, all we can show instead is

$$\forall x.\ \mathsf{Q}(x,x),\quad \exists y.\ \big(\ \neg\mathsf{Q}(y,y)\wedge\neg\mathsf{P}(y)\ \big),\quad \exists z.\ \mathsf{P}(z)$$

Thus, it is obvious that the liberalized δ-rule typically is not only superior[24] to the (non-liberalized) δ-rule w.r.t. reductive theorem proving but also w.r.t. computation of answers and solutions.

Nevertheless—unless we conjecture propositions that already contain free δ^+-variables from the very beginning—the choice-conditions do not produce any overhead in an implementation because they can simply be omitted; thereby leaving the free δ^+-variables unspecified just like the Skolem functions in Skolemizing deduction.

The only overhead compared to the standard framework of Skolemization seems to be that we have to compute transitive closures when checking whether a substitution σ is really an R-substitution on V_{γ} and when computing the σ-update of R. But we actually do not have to compute the transitive closure at all, because we only have to check for acyclicity, which can be done on a graph generating the transitive closures. This check is in the worst case linear in

$$|R| + \sum_{\sigma}\big(\ \ |\Delta_{\sigma}|\ +\ |\Gamma_{\sigma}|\ \ \big)$$

and is expected to perform at least as well as an optimally integrated version (i.e. one without conversion of term-representation) of the linear unification algorithm of Paterson & Wegman (1978) in the standard framework of Skolemization and unification. (Of course, the check for being an R-substitution can also be implemented with any other first-order unification algorithm.)

A.2 Smaller Variable-Condition versus Less Free δ^+-Variables

Not computing the transitive closure of variable-conditions enables another refinement that allows us to go even beyond the fascinating *strong Skolemization* of Nonnengart (1996), whose basic idea can be translated into our framework in the following simplified way.

Instead of proving $\forall x.\ (A{\vee}B)$ it may be advantageous to prove the stronger $\forall x.A \vee \forall x.B$, because after applications of α- and liberalized δ-rules to $\forall x.A \vee \forall x.B$, resulting in $A\{x{\mapsto}x^{\delta^+}_A\},\ B\{x{\mapsto}x^{\delta^+}_B\}$, the variable-conditions introduced for $x^{\delta^+}_A$ and $x^{\delta^+}_B$ may be smaller than the variable-condition introduced for y^{δ^+} after applying these rules to $\forall x.\ (A{\vee}B)$, resulting in $A\{x{\mapsto}y^{\delta^+}\},\ B\{x{\mapsto}y^{\delta^+}\}$, i.e. $\mathcal{V}_{\mathrm{free}}(A)$ and $\mathcal{V}_{\mathrm{free}}(B)$ may be *proper* subsets of $\mathcal{V}_{\mathrm{free}}(A,B)$. Therefore, the proof of $\forall x.A \vee \forall x.B$ may be simpler than the proof of $\forall x.\ (A{\vee}B)$. The nice aspect of strong Skolemization roughly translated into our framework is that the intermediately sized $\mathcal{V}_{\mathrm{free}}(A) \times \{x^{\delta^+}_A\} \ \cup\ \mathcal{V}_{\mathrm{free}}(A,B) \times \{x^{\delta^+}_B\}$ is added to the variable-condition, but only a single Skolem function f is introduced with $x^{\delta^+}_A$ represented as $f(A',X)$ and $x^{\delta^+}_B$ represented as $f(A', B'{\setminus}A')$ where X are some new free γ-variables, $A' := \mathrm{V}_{\gamma} \cap R^*\langle\mathcal{V}_{\mathrm{free}}(A)\rangle$, and $B' := \mathrm{V}_{\gamma} \cap R^*\langle\mathcal{V}_{\mathrm{free}}(B)\rangle$. Thus, $x^{\delta^+}_B$ still becomes dependent on the free variables of the whole disjunction, so that—due to this asymmetry[25]—it may make a difference to prove $\forall x.\ (A{\vee}B)$ or $\forall x.\ (B{\vee}A)$.

Now, if we do not really compute the transitive closures as indicated in Section A.1, we can try to prove $A\{x{\mapsto}x^{\delta^+}_A\},\ B\{x{\mapsto}x^{\delta^+}_B\}$ first, and—if this fails—may later switch directly to prove the weaker $A\{x{\mapsto}y^{\delta^+}\},\ B\{x{\mapsto}y^{\delta^+}\}$ instead, simply by merging the nodes for $x^{\delta^+}_A$ and $x^{\delta^+}_B$ and substituting $x^{\delta^+}_A$ and $x^{\delta^+}_B$ by y^{δ^+}. Of course, we have to check that R stays wellfounded.

Finally note that the same conflict and solution apply to

$$\forall x.\ (A{\wedge}B) \quad \text{versus} \quad \forall x.A \wedge \forall x.B,$$

although these formulas are logically equivalent: The latter in general produces smaller variable-conditions (unless $\mathcal{V}_{\mathrm{free}}(A) = \mathcal{V}_{\mathrm{free}}(B)$) but the former less free δ^+-variables (Skolem functions) and each of the two effects may enable additional proofs and reduce the size of minimal proofs.

A.3 Improving Multiple γ-Rule Applications and Matrix Calculi

Another optimization, inspired by the ideas of Section 7 of Giese (1998) and Appendix B of Wirth (1997), improves the behavior of multiple γ-rule applications to the same formula. It requires a new kind of free γ-variables which are not used for direct instantiation but as generators for the usual kind of free γ-variables. In the tableau community these variables are sometimes called "universal" (cf. e.g. Beckert & Hähnle (1998)), but they have nothing to do with our free δ-variables here. Thus, we call them generator variables and denote them with x^{γ^+} and V_{γ^+}. Instead of the γ-rule say

$$\frac{\varGamma \quad \exists x.A \quad \varPi}{A\{x{\mapsto}t\} \quad \varGamma \quad \exists x.A \quad \varPi}$$

we take a rule like

$$\frac{\varGamma \quad \exists x.A \quad \varPi}{A\{x{\mapsto}x^{\gamma^+}\} \quad \varGamma \quad \varPi}$$

where $\exists x.A$ is removed and x^{γ^+} is a new generator variable. Other γ-rules are changed analogously. The α-rules are not changed and the β-rules at the end of Section 1.2.2 are restricted in their applicability by the restriction of $\mathcal{V}_{\gamma^+}(A) \cap \mathcal{V}_{\gamma^+}(B) = \emptyset$, which (together with the condition that generator variables do not occur in root sequents of Hypothesizing steps and substitutions of Instantiation steps) guarantees that for each generator variable there is always a single branch in a tree that contains all its occurrences. To enable blocked β-steps and for Instantiation, we need a *generation* rule like

$$\frac{\varGamma \quad A \quad \varPi}{A\{x^{\gamma^+} \mapsto t\} \quad \varGamma \quad A \quad \varPi}$$

The δ-rules at the end of Section 1.2.2 either get restricted by $\mathcal{V}_{\gamma^+}(A) = \emptyset$ or otherwise we can proceed in the following less simple but more powerful way: We treat generator variables like free γ-variables and the generation rule replaces each free δ^+-variable y^{δ^+} (free δ^--variables analogously) from $\mathcal{V}_{\delta^+}(A) \cap \langle\!\langle\{x^{\gamma^+}\}\rangle\!\rangle R^+$ in A with a new one, say $y_{\prime}^{\delta^+}$, and add to the variable-condition R a copy of the graph of $\langle\!\langle\{x^{\gamma^+}\}\rangle\!\rangle R^*$ with $y_{\prime}^{\delta^+}$ instead of y^{δ^+}, and add to the choice-condition C something like $(y_{\prime}^{\delta^+}, (C(y^{\delta^+}))\{y^{\delta^+} \mapsto y_{\prime}^{\delta^+}, \ \ldots\})$. The nice treatment in Section 7 of Giese (1998) makes the reason for this seemingly complicated procedure obvious by means of Hilbert's ε-terms.

To include this into our framework, the crucial step is to change the notion of $(\delta, e, \mathcal{A})$-validity such that a generator variable x^{γ^+} is treated like a free γ-variable with the exception that its value may also be chosen from the values of its instances in the generation rule; i.e. a value of x^{γ^+} establishing the validity must exist among $\epsilon(e)(\delta)(x^{\gamma^+})$ and the values of $\mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta) \uplus \delta)(t)$ for the terms t introduced for x^{γ^+} in the generation rule. Without this flexibility, the generation rule would not preserve solutions.

Now, if the formula A in the γ-rule above is a literal or a blocked β-formula, then the new γ-rule plus n generation steps have the effect of n applications of the old γ-rule and no improvement takes place. Otherwise, however, several inference rules may be applied after the new γ-rule, and when we suddenly discover that we need say $\mathsf{P}(x^{\gamma^+})$ twice, then we can apply two generation steps instead of repeating the whole subtree up to the γ-rule application.

All in all, we have to admit that the possibilities to improve multiple γ-rule applications are poor in sequent and tableau calculi. In a matrix representation like in Wallen (1990), however, it is possible to dynamically increase the multiplicity and to let all γ-variables be generating, no matter whether all occurrences of a variable are on the same branch or not. Thus, an implementation should use matrix calculi instead of the presentationally simpler sequent and tableau calculi used in this paper, because there the β- and δ-formulas do not suffer from the severe restrictions explained above.

Moreover, as explained in the $\lim +$-example in Wirth &al. (2003), matrix representation helps to find the right ordering of β-steps (especially of the ones that are critical due to consecutive δ-steps) and to answer the question of downfolding to the left or right in β-steps, simply by delaying the decisions a sequent or tableau representation forces us to do prematurely.

Thus, to follow our design goal of a *natural flow of information* of Section 1.2.1, instead of a sequent or tableau representation, we should use a matrix representation for an implementation of our calculi, similar to the one the CoRe system of Autexier (2003).

B Tools for the Proofs

B.1 Technical Lemmas

The following technical lemma says—roughly speaking—that the Substitution-Lemma can be lifted to $(\mathcal{A}, R)$-valuations as expected.

LEMMA B.1

Let $\mathcal{A}$ be a Σ-structure, and let R be a variable-condition and σ an R-substitution.

1. If R' is a variable-condition with $R \subseteq R'$,
   then each $(\mathcal{A}, R')$-valuation is also an $(\mathcal{A}, R)$-valuation.

2. Let R' be the σ-update of R.
   For each $(\mathcal{A}, R')$-valuation e' there is some $(\mathcal{A}, R)$-valuation e such that

   $$S_e \;=\; S_{e'} \circ ({}_{\mathrm{V}_\gamma\setminus\mathrm{dom}(\sigma)}{\upharpoonleft}\mathrm{id} \cup \Gamma_\sigma{\upharpoonright}_{\mathrm{V}_\gamma}) \;\cup\; \Delta_\sigma{\upharpoonright}_{\mathrm{V}_\gamma}$$

   and for all $\delta : \mathrm{V}_\delta \to \mathcal{A}$:

   $$\epsilon(e)(\delta) \;=\; ({}_{\mathrm{V}_\gamma\setminus\mathrm{dom}(\sigma)}{\upharpoonleft}\mathrm{id} \cup {}_{\mathrm{V}_\gamma}{\upharpoonleft}\sigma) \;\circ\; \mathrm{eval}(\mathcal{A} \uplus \epsilon(e')(\delta) \uplus \delta).$$

3. Let (C', R') be the extended σ-update of (C, R). For each $(\mathcal{A}, R')$-valuation e' and each π' that is $(e', \mathcal{A})$-compatible with (C', R'), there is some $(\mathcal{A}, R)$-valuation e such that

   $$R \;\cup\; S_e \;\cup\; {}_{\mathrm{V}_\delta}{\upharpoonleft}(R' \cup S_{e'} \cup S_{\pi'})^{+}{\upharpoonright}_{\mathrm{V}_\delta} \text{ is wellfounded,}$$

   $$S_e \;=\; (S_{\pi'} \cup {}_{\mathrm{V}_{\delta-}}{\upharpoonleft}\mathrm{id}) \;\circ\; (S_{e'} \circ ({}_{\mathrm{V}_\gamma\setminus\mathrm{dom}(\sigma)}{\upharpoonleft}\mathrm{id} \cup \Gamma_\sigma{\upharpoonright}_{\mathrm{V}_\gamma}) \cup \Delta_\sigma{\upharpoonright}_{\mathrm{V}_\gamma}),$$

   and for all $\delta : \mathrm{V}_\delta \to \mathcal{A}$ and $\tau := {}_{\mathrm{V}_{\delta-}}{\upharpoonleft}\delta$:

   $$\epsilon(e)(\delta) \;=\; ({}_{\mathrm{V}_\gamma\setminus\mathrm{dom}(\sigma)}{\upharpoonleft}\mathrm{id} \cup {}_{\mathrm{V}_\gamma}{\upharpoonleft}\sigma) \;\circ\; \mathrm{eval}(\mathcal{A} \uplus \epsilon(e')(\epsilon(\pi')(\tau) \uplus \tau) \uplus \epsilon(\pi')(\tau) \uplus \tau).$$

B.2 Generalized Notions

As Hilbert's ε-terms can be used to constrain variables in a very general sense with a vast number of applications, the possibility to include a representation of ε-terms into our inference system provides considerable evidence for the quality of our combination of *descente infinie* and deduction. This inclusion, however, now only requires a minor generalization of our choice-conditions. For a motivational introduction to choice-conditions as an indefinite semantics for Hilbert's ε-terms, cf. Wirth (2002). Since we do not want to publish the long proofs twice, all proofs are omitted in Wirth (2002). As a consequence, the proofs in this paper have to include the following generalization of the notion of choice-condition and show with little additional effort that our combination of *descente infinie* and deduction admits the inclusion of Hilbert's ε-terms.

The generalizations in the following definitions—as compared to the ones of Section 2.2.4—additionally model the so-called "subordinate" ε-terms by extending the possible value of a choice-condition from a simple formula B to a formula-valued λ-term $\lambda v_0.\ \ldots\lambda v_{l-1}.\ B$ with a formula B in which the variables $v_0, \ldots, v_{n-1}$ may occur free. Notice that, for $l=0$, all generalized definitions specialize to the original definitions.

DEFINITION B.2 (Choice-Condition, generalized) *(Cf. Definition 2.20)*

More generally than stated in Definition 2.20, the values of an *R-choice-condition C* can be formula-valued λ-terms (instead of formulas)
where, for $y^{\delta+} \in \mathrm{dom}(C)$ and $C(y^{\delta+}) = \lambda v_0.\ \ldots\lambda v_{l-1}.\ B$,

B is a formula whose free occurring variables from $\mathrm{V}_{\mathrm{bound}}$ are among $\{v_0, \ldots, v_{l-1}\} \subseteq \mathrm{V}_{\mathrm{bound}}$

and where, for $v_0 : \alpha_0,\ \ldots,\ v_{l-1} : \alpha_{l-1}$, we have

$$y^{\delta+} : \alpha_0 \rightarrow \ldots \rightarrow \alpha_{l-1} \rightarrow \alpha_l \text{ for some type } \alpha_l,$$

and any occurrence of $y^{\delta+}$ in B must be of the form $y^{\delta+}(v_0)\cdots(v_{l-1})$.

DEFINITION B.3 (Compatibility, generalized) *(Cf. Definition 2.23)*

Item 2 of Definition 2.23 is generalized to the following:

2. For all $y^{\delta+} \in \mathrm{dom}(C)$ with $C(y^{\delta+}) = \lambda v_0.\ \ldots\lambda v_{l-1}.\ B$ for a formula B, for all $\tau : \mathrm{V}_{\delta-} \rightarrow \mathcal{A}$, for all $\eta : \{y^{\delta+}\} \rightarrow \mathcal{A}$, and for all $\chi : \{v_0, \ldots, v_{l-1}\} \rightarrow \mathcal{A}$, setting $\delta := \epsilon(\pi)(\tau) \uplus \tau \uplus \chi$, $\delta' := \eta \uplus_{\mathrm{V}\setminus\{y^{\delta+}\}}\!\upharpoonleft\!\delta$ (i.e. δ' is the η-variant of δ):

   If B is $(\delta', e, \mathcal{A})$-valid, then B is also $(\delta, e, \mathcal{A})$-valid.

B.3 Instantiation of Free δ^+-Variables

Due to the existential treatment of free δ^+-variables in Definition 2.27 ("some π"), an Instantiation step may replace not only the free γ-variables but also the free δ^+-variables globally. For doing so, we need means of expressing the requirement on an R-substitution on $\mathrm{V}_{\gamma} \cup \mathrm{V}_{\delta^+}$ to replace the free δ^+-variables in accordance with the compatibility requirement of Definition B.3(2):

DEFINITION B.4 (Q_C)
For an R-choice-condition C, we let Q_C be a total function from $\mathrm{dom}(C)$ into the set of single-formula sequents such that for each $y^{\delta^+} \in \mathrm{dom}(C)$ with
$C(y^{\delta^+}) = \lambda v_0.\ \ldots \lambda v_{l-1}.\ B$ for a formula B, we have $Q_C(y^{\delta^+}) =$

$$\forall v_0.\ \ldots \forall v_{l-1}.\ \big(\ \exists y.\ (B\{y^{\delta^+}(v_0)\cdots(v_{l-1}) \mapsto y\})\ \Rightarrow\ B\ \big)$$

for an arbitrary $y \in \mathrm{V}_{\mathrm{bound}} \backslash \mathcal{V}(C(y^{\delta^+}))$.

Note that $Q_C(y^{\delta^+})$ is (C, R)-valid and can serve as an axiom in $\mathcal{AX}$ as indicated in item 1 of Section A.1. Indeed, directly by Definition B.3, $Q_C(y^{\delta^+})$ is even $(\pi, e, \mathcal{A})$-valid for each $(\mathcal{A}, R)$-valuation e and each π that is $(e, \mathcal{A})$-compatible with (C, R).

For dealing with R-substitutions on $\mathrm{V}_{\gamma} \cup \mathrm{V}_{\delta^+}$ semantically, we need the following technical lemma used in the proofs of Lemma B.6 and Lemma B.7, which again are essential for Lemma 2.31(5) and Lemma 2.37(5), respectively.

Note that, considering those variables that are constrained by the choice-condition C and replaced by the substitution σ, on the one hand, the set O contains the variables whose replacements are supported by the lemmas $(\langle O \rangle Q_C)\sigma$. On the other hand, the set N contains the variables that are not supported by such lemmas, plus all the variables that are constrained by C and suffer from this missing support in the sense that they depend on these variables via the variable-condition R.

LEMMA B.5
Let C be an R-choice-condition, let $\mathcal{A}$ be a Σ-structure, and let σ be an R-substitution on $\mathrm{V}_{\gamma} \cup \mathrm{V}_{\delta^+}$. Let (C', R') be the extended σ-update of (C, R).
Assume that we have O and N with $O \subseteq \mathrm{dom}(C) \cap \mathrm{dom}(\sigma) \subseteq O \uplus N$,
$N \subseteq \mathrm{dom}(C) \setminus O$, and $\mathrm{dom}(C) \cap \langle N \rangle R^+ \subseteq N$.
Now, for any $(\mathcal{A}, R')$-valuation e' and any π' that is $(e', \mathcal{A})$-compatible with (C', R') such that $(\langle O \rangle Q_C)\sigma$ is $(\pi', e', \mathcal{A})$-valid, there are an $(\mathcal{A}, R)$-valuation e and a π that is $(e, \mathcal{A})$-compatible with (C, R) for which the following holds:

1. For any term or formula B (possibly with some unbound occurrences of variables from a set $W \subseteq \mathrm{V}_{\mathrm{bound}}$) with $N \cap \mathcal{V}(B) = \emptyset$, and for any $\tau : \mathrm{V}_{\delta^-} \to \mathcal{A}$ and $\chi : W \to \mathcal{A}$, when we set $\delta' := \epsilon(\pi')(\tau) \uplus \tau$ and $\delta := \epsilon(\pi)(\tau) \uplus \tau$:
$$\mathrm{eval}(\mathcal{A} \uplus \epsilon(e')(\delta') \uplus \delta' \uplus \chi)(B\sigma) = \mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta) \uplus \delta \uplus \chi)(B).$$

2. For any set of sequents G with $N \cap \mathcal{V}(G) = \emptyset$:
$$G\sigma \text{ is } (\pi', e', \mathcal{A})\text{-valid iff } G \text{ is } (\pi, e, \mathcal{A})\text{-valid.}$$

The following two lemmas generalize Lemma 2.31(6) and Lemma 2.37(6), respectively:

LEMMA B.6 (Reduction & Instantiation)
For an R-substitution σ on $\mathrm{V}_\gamma \cup \mathrm{V}_{\delta^+}$, and the extended σ-update (C', R') of (C, R),
and for O, N with $O \subseteq \mathrm{dom}(C) \cap \mathrm{dom}(\sigma) \subseteq O \uplus N$,

$$N \subseteq \mathrm{dom}(C) \setminus O,\ \ \mathrm{dom}(C) \cap \langle N \rangle R^+ \subseteq N, \text{ and } N \cap \mathcal{V}(G_0, G_1) = \emptyset:$$

1. If $G_0\sigma \cup (\langle O \rangle Q_C)\sigma$ is (C', R')-valid in $\mathcal{A}$, then G_0 is (C, R)-valid in $\mathcal{A}$.
2. If G_0 (C, R)-reduces to G_1 in $\mathcal{A}$,
   then $G_0\sigma$ (C', R')-reduces to $G_1\sigma \cup (\langle O \rangle Q_C)\sigma$ in $\mathcal{A}$.

LEMMA B.7 (Groundedness and Instantiation)
For an R-substitution σ on $\mathrm{V}_\gamma \cup \mathrm{V}_{\delta^+}$, and the extended σ-update (C', R') of (C, R),
and for O, N with $O \subseteq \mathrm{dom}(C) \cap \mathrm{dom}(\sigma) \subseteq O \uplus N$,

$$N \subseteq \mathrm{dom}(C) \setminus O,\ \ \mathrm{dom}(C) \cap \langle N \rangle R^+ \subseteq N, \text{ and } N \cap \mathcal{V}(G_0, G_1, L_1) = \emptyset,$$

and L_2 a set of weighted sequents with $\mathrm{Seq}(L_2) = (\langle O \rangle Q_C)\sigma$:
If $G_0 \rightharpoonup_{C,R} (G_1, L_1)$, then $G_0\sigma \rightharpoonup_{C',R'} (G_1\sigma, L_1\sigma \cup L_2)$.

The following definition extends the Instantiation rule of Definition 2.42 to the application of R-substitutions even on $\mathrm{V}_\gamma \cup \mathrm{V}_{\delta^+}$ instead of V_γ only. Note that it specializes to the original definition for R-substitutions on V_γ.

Every replacement of a free δ^+-variable y^{δ^+} must be justified by a lemma $Q_C(y^{\delta^+})\sigma$ given by the proposition of a proof tree number $j_{y^{\delta^+}}$, which must be added in a preceding Hypothesizing step unless it is already present. Note that this lemma is special in the sense that it is not applied locally in some proof tree but globally. Especially problematic is the possibility that y^{δ^+} occurs in the proof of the lemma itself. If we are not very careful, the lemma becomes a lemma of itself, resulting in a cyclic lemma application relation. Therefore, since we do not want to reintroduce the lemma as an open lemma and prove it again, we take a very close look on which of our (possibly open) propositions really depend on the justifying lemma $Q_C(y^{\delta^+})\sigma$ after global application of σ. Our solution is that the lemma is relevant for any proof tree number i whose proof state (i.e. the goals, the proposition itself, and the lemmas) contains free δ^+-variables that may depend on y^{δ^+}; i.e. for any i with $y^{\delta^+} \in D_i$, for the D_i given below.

(Cf. Definition 2.42, Definition 2.47)
DEFINITION B.8 (Generalized Instantiation Rule and Soundness of Steps)

Let σ be an R-substitution on $\mathrm{V}_\gamma \cup \mathrm{V}_{\delta^+}$.
Let (C', R') be the extended σ-update of (C, R).
Set and $H' := H$ and $F' := \{\ (\ i,\ ((\varGamma\sigma, \aleph\sigma), t\sigma)\ )\ \big|\ (\ i,\ ((\varGamma, \aleph), t)\ ) \in F\ \}$.
Assume that for each $y^{\delta^+} \in \mathrm{dom}(C) \cap \mathrm{dom}(\sigma)$ there is some $j_{y^{\delta^+}} \in \mathrm{dom}(F)$ with

$$\mathrm{Seq}(\mathrm{Propos}(\langle\!\{j_{y^{\delta^+}}\}\!\rangle F')) = \{Q_C(y^{\delta^+})\sigma\}.$$

For each $i \in \mathrm{dom}(F)$ set $I := H^*\langle\!\{i\}\!\rangle$ and

$$D_i := \mathrm{dom}(C) \cap \mathrm{dom}(\sigma) \cap R^*\Big\langle \mathcal{V}_{\delta^+}\Big(\mathrm{Goals}\big(\,\mathrm{Trees}(\langle I \rangle F)\,\big),\ \mathrm{Propos}\big(\,\langle \{i\} \cup L\langle I \rangle \rangle F\,\big)\Big)\Big\rangle.$$

Set $L' := L \cup \{\ (j_{y^{\delta^+}}, i)\ \big|\ y^{\delta^+} \in D_i \wedge i \in \mathrm{dom}(F)\ \}$.

Such a generalized Instantiation step is *safe* if, for all $y^{\delta^+} \in \mathrm{dom}(C) \cap \mathrm{dom}(\sigma)$,
$Q_C(y^{\delta^+})\sigma$ is (C', R')-valid by satisfying the requirements of Theorem 2.45,
i.e. all trees in $\mathrm{Trees}\big(\ \langle (L' \cup H')^* \langle\!\{j_{y^{\delta^+}}\}\!\rangle \rangle F'\ \big)$ are closed and $L' \circ H'^*$ is wellfounded.

C The Proofs

Proof of Lemma 2.1

The backward implication is trivial because R^+-minimality in a class A implies R-minimality in A due to $R \subseteq R^+$. For the forward implication, since R^+ is clearly transitive, it suffices to show that it is wellfounded, because then it is irreflexive. Thus, suppose that there is some class A with $\forall a \in A.\ \exists a' \in A.\ a' R^+ a$. We have to show that A must be empty. Set $B := \{\ b \mid \exists a \in A.\ aR^*b\ \}$.

Claim 1: For any $b \in B$, there is some $b' \in B$ with $b'Rb$.

Proof of Claim 1: By definition of B and the property of A, there is some $a \in A$ with aR^+b. Thus, there is some b' with $aR^*b'Rb$. Q.e.d. (Claim 1)

By Claim 1 and the assumption that R is wellfounded, we get $B = \emptyset$. Then, we also have $A = \emptyset$ due to $A \subseteq B$. **Q.e.d. (Lemma 2.1)**

Proof of Lemma 2.5

2.2 ⇒ 2.4: Let $<$ be an ordering with $\forall a \in A.\ \exists a' \in A.\ a > a'$. Set $R := {>} \cap (A \times A)$. Now $\mathrm{dom}(R) = A \supseteq \mathrm{ran}(R)$. Assume $A \neq \emptyset$. By the Principle of Dependent Choice, R is not terminating. This contradicts (i) and (ii) of the Principle of Descente Infinie.

2.4 ⇒ 2.2: Let R be a binary relation with $\mathrm{ran}(R) \subseteq \mathrm{dom}(R) \neq \emptyset$. We are going to show that R is not terminating.

Set $A := \left\{\ a\ \middle|\ \exists n \in \mathbf{N}.\ \begin{pmatrix} & a : \{0, \ldots, n\} \to \mathrm{dom}(R) \\ \wedge & \forall i \prec n.\ a_i R a_{i+1} \end{pmatrix} \right\}$.

Define $\lesssim$ on A by $a \gtrsim a'$ if $\begin{pmatrix} & \mathrm{dom}(a) \subseteq \mathrm{dom}(a') \\ \wedge & \forall i \in \mathrm{dom}(a).\ a_i = a'_i \end{pmatrix}$. Let $<$ be the ordering of $\lesssim$.

Claim 1: $\forall a \in A.\ \exists a' \in A.\ a > a'$.

Proof of Claim 1: For $a : \{0, \ldots, n\} \to \mathrm{dom}(R)$ we have to show the existence of some $a' : \{0, \ldots, n, n{+}1\} \to \mathrm{dom}(R)$ with $a > a'$. When we set $a'_i := a_i$ for $i \preceq n$ then (due to $a_n \in \mathrm{dom}(R)$) there exists an a'_{n+1} with $a_n\ R\ a'_{n+1}$, and then $a'_{n+1} \in \mathrm{ran}(R) \subseteq \mathrm{dom}(R)$. Q.e.d. (Claim 1)

Since $\mathrm{dom}(R) \neq \emptyset$ we have $A \neq \emptyset$. Thus, by Claim 1 and the Principle of Descente Infinie (i) there is some non-terminating sequence $(a_i)_{i \in \mathbf{N}}$ in $>$ and we set $C := \mathrm{ran}(a)$ or (ii) there is some $C \subseteq A$ totally ordered by $<$ that has no $<$-minimal element. But then $\bigcup C$ is a non-terminating sequence in R.

2.4(i) ⇒ 2.3: If $<$ is not wellfounded, then there is some non-empty class A with $\forall a \in A.\ \exists a' \in A.\ a {>} a'$. Thus, by the Principle of Descente Infinie 2.4(i), ${>} \cap (A \times A)$ is not terminating, which implies that $>$ is not terminating.

2.3 ⇒ 2.4(i): Let $<$ be an ordering. Then ${<} \cap (A \times A)$ is an ordering, too. Thus, if ${>} \cap (A \times A)$ is terminating, by the Principle of Wellfoundedness, ${<} \cap (A \times A)$ is a wellfounded ordering. In case of $\forall a \in A.\ \exists a' \in A.\ a {>} a'$, this means that A must be empty. **Q.e.d. (Lemma 2.5)**

Proof of Lemma 2.22
Here we denote concatenation (product) of relations ‘$\circ$’ simply by juxtaposition and assume it to have higher priority than any other binary operator. R'^{+} is a wellfounded ordering simply because R' is the σ-update of R and σ is an R-substitution. Now it suffices to show the following two claims for an arbitrary $y^{\delta+} \in \mathrm{dom}(C')$:
Claim 1: For all $z^{\delta} \in \mathcal{V}_{\delta}(C'(y^{\delta+}))\setminus\{y^{\delta+}\}$: $z^{\delta} R'^{+} y^{\delta+}$.
Claim 2: For all $u^{\gamma} \in \mathcal{V}_{\gamma}(C'(y^{\delta+}))$: $u^{\gamma} R'^{+} y^{\delta+}$.
Proof of Claim 1: Let $z^{\delta} \in \mathcal{V}_{\delta}(C'(y^{\delta+}))\setminus\{y^{\delta+}\}$. By the definition of C' this means $z^{\delta} \in \mathcal{V}_{\delta}(C(y^{\delta+}))\setminus\{y^{\delta+}\}$ or there is some $u \in \mathcal{V}(C(y^{\delta+}))$ with $z^{\delta}\ \Delta_{\sigma}\ u$. Since C is an R-choice-condition, we have $z^{\delta}\ R^{+}\ y^{\delta+}$ or $z^{\delta}\ \Delta_{\sigma}\ u\ R^{*}\ y^{\delta+}$. As R' is the σ-update of R, we have $R \cup \Delta_{\sigma} \subseteq R'$.[26] Thus $z^{\delta} R'^{+} y^{\delta+}$.
Q.e.d. (Claim 1)
Proof of Claim 2: Let $u^{\gamma} \in \mathcal{V}_{\gamma}(C'(y^{\delta+}))$. By the definition of C' there is some $v \in \mathcal{V}(C(y^{\delta+}))$ with $u^{\gamma}\ ({}_{\mathrm{V}_{\gamma}\setminus\mathrm{dom}(\sigma)}\upharpoonright\mathrm{id} \cup \Gamma_{\sigma})\ v$. Since C is an R-choice-condition, we have $v\ R^{*}\ y^{\delta+}$, i.e. $u^{\gamma}\ ({}_{\mathrm{V}_{\gamma}\setminus\mathrm{dom}(\sigma)}\upharpoonright\mathrm{id} \cup \Gamma_{\sigma})R^{*}\ y^{\delta+}$. As R' is the σ-update of R, we have $({}_{\mathrm{V}_{\gamma}\setminus\mathrm{dom}(\sigma)}\upharpoonright\mathrm{id} \cup \Gamma_{\sigma})R^{*} \subseteq (R \cup \Gamma_{\sigma})^{*} \subseteq R'^{*}$.[27] Thus $u^{\gamma} R'^{+} y^{\delta+}$. Q.e.d. (Claim 2) **Q.e.d. (Lemma 2.22)**

Proof of Lemma 2.24
Set $\lhd := (R \cup S_{e})^{+}$ and $S_{\pi} := \lhd \cap (\mathrm{V}_{\delta-}\times\mathrm{V}_{\delta+})$. As e is an $(\mathcal{A}, R)$-valuation, $\lhd$ is a wellfounded ordering. With the help of a choice function and by recursion on $y^{\delta+} \in \mathrm{V}_{\delta+}$ in $\lhd$ we can define $\pi(y^{\delta+}) : (S_{\pi}\langle\{y^{\delta+}\}\rangle \to \mathcal{A}) \to \mathcal{A}$ in the following way:
Let $\tau : S_{\pi}\langle\{y^{\delta+}\}\rangle \to \mathcal{A}$.
In case of $y^{\delta+} \in \mathrm{V}_{\delta+}\setminus\mathrm{dom}(C)$ we choose an arbitrary value for $\pi(y^{\delta+})(\tau)$ from the universe of $\mathcal{A}$ (of the appropriate type). Note that universes are assumed to be non-empty, cf. Section 2.1.4.
In case of $y^{\delta+} \in \mathrm{dom}(C)$, we have the following situation: $C(y^{\delta+}) = \lambda v_0.\ \ldots\lambda v_{l-1}.\ B$, B is a formula whose unbound variables from $\mathrm{V}_{\mathrm{bound}}$ are among $\{v_0, \ldots, v_{l-1}\} \subseteq \mathrm{V}_{\mathrm{bound}}$ and where, for $v_0 : \alpha_0, \ldots, v_{l-1} : \alpha_{l-1}$, we have $y^{\delta+} : \alpha_0 \to \ldots \to \alpha_{l-1} \to \alpha_l$ for some type α_l, and any occurrence of $y^{\delta+}$ in B is of the form $y^{\delta+}(v_0)\cdots(v_{l-1})$. In this case, we let $\pi(y^{\delta+})(\tau)$ be the function f that for $\chi : \{v_0, \ldots, v_{l-1}\} \to \mathcal{A}$ chooses a value from the universe of $\mathcal{A}$ for $f(\chi(v_0))\cdots(\chi(v_{l-1}))$ such that, if possible, B is $(\epsilon(\pi)(\tau \uplus \tau')\uplus\tau \uplus \tau' \uplus \chi, e, \mathcal{A})$-valid for an arbitrary $\tau' : (\mathrm{V}_{\delta-}\setminus\mathrm{dom}(\tau)) \to \mathcal{A}$. Note that this definition of $f(\chi(v_0))\cdots(\chi(v_{l-1}))$ does not depend on the values of $f(\chi'(v_0))\cdots(\chi'(v_{l-1}))$ for a different $\chi' : \{v_0, \ldots, v_{l-1}\} \to \mathcal{A}$ because any occurrence of $y^{\delta+}$ in B is of the form $y^{\delta+}(v_0)\cdots(v_{l-1})$.
Claim 1: For $z^{\delta} \in \mathrm{V}_{\delta}$ with $z^{\delta} \lhd y^{\delta+}$, $(\epsilon(\pi)(\tau \uplus \tau') \uplus \tau \uplus \tau')(z^{\delta})$ depends only τ, $\pi(z^{\delta})$, and z^{δ}.
Claim 2: For $x^{\gamma} \in \mathrm{V}_{\gamma}$ with $z^{\gamma} \lhd y^{\delta+}$, $\epsilon(e)(\epsilon(\pi)(\tau \uplus \tau') \uplus \tau \uplus \tau')(x^{\gamma})$ depends only τ, ${}_{\lhd\langle\{y^{\delta+}\}\rangle}\upharpoonright\pi$ and $e(x^{\gamma})$.
Claim 3: The definition of $\pi(y^{\delta+})(\tau)$ depends only on such $\pi(v^{\delta+})$ with $v^{\delta+} \lhd y^{\delta+}$.
Claim 4: The definition of $\pi(y^{\delta+})(\tau)$ does not depend on τ'.
Proof of Claim 1: For $z^{\delta} \in \mathrm{V}_{\delta-}$ we have $(\epsilon(\pi)(\tau \uplus \tau') \uplus \tau \uplus \tau')(z^{\delta}) = \tau(z^{\delta})$ due to $z^{\delta} \in S_{\pi}\langle\{y^{\delta+}\}\rangle$. Moreover, for $z^{\delta} \in \mathrm{V}_{\delta+}$, we have $S_{\pi}\langle\{z^{\delta}\}\rangle \subseteq S_{\pi}\langle\{y^{\delta+}\}\rangle$, and therefore $(\epsilon(\pi)(\tau \uplus \tau') \uplus \tau \uplus \tau')(z^{\delta}) = \pi(z^{\delta})({}_{S_{\pi}\langle\{z^{\delta}\}\rangle}\upharpoonright(\tau \uplus \tau')) = \pi(z^{\delta})({}_{S_{\pi}\langle\{z^{\delta}\}\rangle}\upharpoonright\tau)$. Q.e.d. (Claim 1)
Proof of Claim 2: As $S_{e}\langle\{x^{\gamma}\}\rangle \subseteq \lhd\langle\{y^{\delta+}\}\rangle$, and $\epsilon(e)(\epsilon(\pi)(\tau \uplus \tau') \uplus \tau \uplus \tau')(x^{\gamma}) = e(x^{\gamma})({}_{S_{e}\langle\{x^{\gamma}\}\rangle}\upharpoonright(\epsilon(\pi)(\tau \uplus \tau') \uplus \tau \uplus \tau'))$ this follows from Claim 1. Q.e.d. (Claim 2)
Proof of Claim 3 and 4: Since C is an R-choice-condition, we have $z \lhd y^{\delta+}$ for all $z \in \mathcal{V}_{\mathrm{free}}(C(y^{\delta+}))\setminus\{y^{\delta+}\}$. Thus, this follows from Claim 1 and Claim 2. Q.e.d. (Claim 3, 4)

Now π is well-defined by Claim 3 and Claim 4 and obviously semantical. Thus, Item 1 of Definition 2.23 is satisfied because $(R \cup S_e \cup S_\pi)^+ = \lhd$ is a wellfounded ordering. For showing Item 2 of Definition B.3, let $\tau : \mathrm{V}_{\delta-} \to \mathcal{A}$, $y^{\delta^+} \in \mathrm{dom}(C)$, and $C(y^{\delta^+}) = \lambda v_0. \ \ldots \lambda v_{l-1}. \ B$, and assume to the contrary that, for some $\eta : \{y^{\delta^+}\} \to \mathcal{A}$ and $\chi : \{v_0, \ldots, v_{l-1}\} \to \mathcal{A}$, B is $(\delta', e, \mathcal{A})$-valid, but not $(\delta, e, \mathcal{A})$-valid for $\delta := \epsilon(\pi)(\tau) \uplus \tau \uplus \chi$ and $\delta' := \eta \uplus {}_{\mathrm{V} \setminus \{y^{\delta^+}\}}\!\upharpoonleft\!\delta$. This contradicts the definition of $\pi(y^{\delta^+})({}_{S_\pi \langle\!\{y^{\delta+}\}\!\rangle}\!\upharpoonleft\!\tau)$ from above due to Claim 4. **Q.e.d. (Lemma 2.24)**

Proof of Lemma 2.26
Due to $R \subseteq R'$, by Lemma B.1(1), e is an $(\mathcal{A}, R)$-valuation, too. As π is $(e, \mathcal{A})$-compatible with (C', R'), $C \subseteq C'$, and the choice-condition occurs only in Item 2 of Definition 2.23 and Definition B.3, π is $(e, \mathcal{A})$-compatible with (C, R'). As $R \subseteq R'$, and the variable-condition occurs only in Item 1 of Definition 2.23, π is $(e, \mathcal{A})$-compatible with (C, R). **Q.e.d. (Lemma 2.26)**

Proof of Lemma 2.28
As G is $(\mathrm{V}_\gamma \times \mathrm{V}_\delta)$-valid in $\mathcal{A}$, there is some $(\mathcal{A}, \mathrm{V}_\gamma \times \mathrm{V}_\delta)$-valuation e s.t. G is $(e, \mathcal{A})$-valid.
Claim 1: e is an $(\mathcal{A}, R)$-valuation.
Proof of Claim 1: As C is an R-choice-condition, R is wellfounded. As e is an $(\mathcal{A}, \mathrm{V}_\gamma \times \mathrm{V}_\delta)$-valuation, $S_e \circ (\mathrm{V}_\gamma \times \mathrm{V}_\delta)$ is irreflexive. This means $S_e = \emptyset$, i.e. $R \cup S_e = R$. This means that $R \cup S_e$ is wellfounded, as was to be shown. Q.e.d. (Claim 1)
By Claim 1, G is immediately R-valid. Moreover, by Claim 1 and Lemma 2.24, there is some π that is $(e, \mathcal{A})$-compatible with (C, R). As G is $(e, \mathcal{A})$-valid, G is also $(\epsilon(\pi)(\tau) \uplus \tau, e, \mathcal{A})$-valid for all $\tau : \mathrm{V}_{\delta-} \to \mathcal{A}$. Then, as π is $(e, \mathcal{A})$-compatible with (C, R), and by Claim 1, G is (C, R)-valid in $\mathcal{A}$.
Q.e.d. (Lemma 2.28)

Proof of Lemma 2.29
As G is (C, R)-valid in $\mathcal{A}$, there are some $(\mathcal{A}, R)$-valuation e and some π s.t. π is $(e, \mathcal{A})$-compatible with (C, R) and G is $(\pi, e, \mathcal{A})$-valid.
Set $S_{e'} := ({}_{\mathrm{V}_{\delta-}}\!\upharpoonleft\!\mathrm{id} \cup S_\pi) \circ S_e\!\upharpoonright_{\mathrm{V}_\gamma \setminus \mathrm{ran}(\varsigma)} \ \cup \ S_\pi \circ \varsigma$. We define e' via:
For $x^\gamma \in \mathrm{V}_\gamma \setminus \mathrm{ran}(\varsigma)$: For $\tau : S_{e'}\langle\!\{x^\gamma\}\!\rangle \to \mathcal{A}$:

$$e'(x^\gamma)(\tau) := e(x^\gamma)({}_{S_e \langle\!\{x^\gamma\}\!\rangle}\!\upharpoonleft\!(\epsilon(\pi)(\tau \uplus \tau') \uplus \tau \uplus \tau'))$$

where $\tau' : (\mathrm{V}_{\delta-} \setminus \mathrm{dom}(\tau)) \to \mathcal{A}$. Note that this right-hand side is okay because $\mathrm{dom}(\tau) \subseteq \mathrm{V}_{\delta-}$; indeed, due to $x^\gamma \notin \mathrm{ran}(\varsigma)$, we have $S_{e'}\langle\!\{x^\gamma\}\!\rangle = (\mathrm{V}_{\delta-} \cap S_e\langle\!\{x^\gamma\}\!\rangle) \cup (S_\pi \circ S_e)\langle\!\{x^\gamma\}\!\rangle \subseteq \mathrm{V}_{\delta-}$. Furthermore, note that this right-hand side does not depend on τ' because $\mathrm{V}_{\delta-} \cap S_e\langle\!\{x^\gamma\}\!\rangle \subseteq S_{e'}\langle\!\{x^\gamma\}\!\rangle = \mathrm{dom}(\tau)$, and for $y^{\delta^+} \in S_e\langle\!\{x^\gamma\}\!\rangle$, we have $S_\pi\langle\!\{y^{\delta^+}\}\!\rangle \subseteq (S_\pi \circ S_e)\langle\!\{x^\gamma\}\!\rangle \subseteq S_{e'}\langle\!\{x^\gamma\}\!\rangle$ and therefore $\epsilon(\pi)(\tau \uplus \tau')(y^{\delta^+}) = \pi(y^{\delta^+})({}_{S_\pi \langle\!\{y^{\delta+}\}\!\rangle}\!\upharpoonleft\!(\tau \uplus \tau')) = \pi(y^{\delta^+})({}_{S_\pi \langle\!\{y^{\delta+}\}\!\rangle}\!\upharpoonleft\!\tau)$.
For $x^\gamma \in \mathrm{ran}(\varsigma)$: For $\tau : S_{e'}\langle\!\{x^\gamma\}\!\rangle \to \mathcal{A}$: $e'(x^\gamma)(\tau) := \pi(\varsigma^{-1}(x^\gamma))(\tau)$.
Note that this right-hand side is okay because, due to $x^\gamma \in \mathrm{ran}(\varsigma)$, we have $S_{e'}\langle\!\{x^\gamma\}\!\rangle = S_\pi\langle\!\{\varsigma^{-1}(x^\gamma)\}\!\rangle \subseteq \mathrm{V}_{\delta-}$.
Claim 1: e' is an $(\mathcal{A}, R')$-valuation.

Proof of Claim 1: Here we denote concatenation (product) of relations '$\circ$' again by juxtaposition and assume it to have higher priority than any other binary operator. It suffices to show that $R' \cup S_{e'}$ is wellfounded. As π is $(e, \mathcal{A})$-compatible with (C, R), we know that $R \cup S_e \cup S_\pi$ is wellfounded. Thus, the subset
$({}_{\mathrm{V}_{\delta-} \cup \mathrm{V}_\gamma \setminus \mathrm{ran}(\varsigma)}{\upharpoonleft}\mathrm{id} \cup S_\pi \varsigma \varsigma^{-1}) R^+ {\upharpoonright}_{\mathrm{V}_{\delta-} \cup \mathrm{V}_\gamma \setminus \mathrm{ran}(\varsigma)} \ \cup \ ({}_{\mathrm{V}_{\delta-}}{\upharpoonleft}\mathrm{id} \cup S_\pi) S_e {\upharpoonright}_{\mathrm{V}_\gamma \setminus \mathrm{ran}(\varsigma)}$ of its transitive closure is wellfounded, too.
Since the domain of this relation and $\mathrm{dom}(S_\pi)$ are disjoint from $\mathrm{ran}(\varsigma)$, we know that
$({}_{\mathrm{V}_{\delta-} \cup \mathrm{V}_\gamma \setminus \mathrm{ran}(\varsigma)}{\upharpoonleft}\mathrm{id} \cup S_\pi \varsigma \varsigma^{-1}) R^+ {\upharpoonright}_{\mathrm{V}_{\delta-} \cup \mathrm{V}_\gamma \setminus \mathrm{ran}(\varsigma)} \ \cup \ ({}_{\mathrm{V}_{\delta-}}{\upharpoonleft}\mathrm{id} \cup S_\pi) S_e {\upharpoonright}_{\mathrm{V}_\gamma \setminus \mathrm{ran}(\varsigma)} \ \cup \ S_\pi \varsigma$
is wellfounded, too. Since the domain of this relation and V_γ are disjoint from $\mathrm{V}_{\delta+}$,
$({}_{\mathrm{V}_{\delta-} \cup \mathrm{V}_\gamma \setminus \mathrm{ran}(\varsigma)}{\upharpoonleft}\mathrm{id} \cup S_\pi \varsigma \varsigma^{-1}) R^+ {\upharpoonright}_{\mathrm{V}_{\delta-} \cup \mathrm{V}_\gamma \setminus \mathrm{ran}(\varsigma)} \ \cup \ \mathrm{V}_\gamma \times \mathrm{V}_{\delta+} \ \cup \ ({}_{\mathrm{V}_{\delta-}}{\upharpoonleft}\mathrm{id} \cup S_\pi) S_e {\upharpoonright}_{\mathrm{V}_\gamma \setminus \mathrm{ran}(\varsigma)} \ \cup \ S_\pi \varsigma$
is wellfounded, too. Since a step with this relation that can precede a step with ς^{-1} can only be a step with $S_\pi \varsigma$ (due to $\mathrm{dom}(\varsigma^{-1}) = \mathrm{ran}(\varsigma) \subseteq \mathrm{V}_\gamma$),
$({}_{\mathrm{V}_{\delta-} \cup \mathrm{V}_\gamma \setminus \mathrm{ran}(\varsigma)}{\upharpoonleft}\mathrm{id} \cup \varsigma^{-1}) R^+ {\upharpoonright}_{\mathrm{V}_{\delta-} \cup \mathrm{V}_\gamma \setminus \mathrm{ran}(\varsigma)} \ \cup \ \mathrm{V}_\gamma \times \mathrm{V}_{\delta+} \ \cup \ ({}_{\mathrm{V}_{\delta-}}{\upharpoonleft}\mathrm{id} \cup S_\pi) S_e {\upharpoonright}_{\mathrm{V}_\gamma \setminus \mathrm{ran}(\varsigma)} \ \cup \ S_\pi \varsigma$
is wellfounded, too; just like its subset $R' \cup S_{e'}$. Q.e.d. (Claim 1)

As the universes are assumed to be non-empty (cf. Section 2.1.4), there is some $\delta : \mathrm{V}_{\delta+} \longrightarrow \mathcal{A}$ by the Axiom of Choice. Define π' by $\pi'(y^{\delta+})(\emptyset) := \delta(y^{\delta+})$.

Claim 2: π' is $(e', \mathcal{A})$-compatible with $(\emptyset, R')$.

Proof of Claim 2: We have $S_{\pi'} = \emptyset$. Thus, $R' \cup S_{e'} \cup S_{\pi'}$ is equal to $R' \cup S_{e'}$, which is wellfounded by Claim 1. Q.e.d. (Claim 2)

Claim 3: For $\tau : \mathrm{V}_{\delta-} \longrightarrow \mathcal{A}$ and $x^\gamma \in \mathcal{V}_\gamma(G)$:

$$\epsilon(e')(\epsilon(\pi')(\tau) \uplus \tau)(x^\gamma) = \epsilon(e)(\epsilon(\pi)(\tau) \uplus \tau)(x^\gamma).$$

Proof of Claim 3: We have $x^\gamma \in \mathcal{V}_\gamma(G) \subseteq \mathrm{V}_\gamma \setminus \mathrm{ran}(\varsigma)$. Thus, by the discussion of the first case of the definition of e', we have $\epsilon(e')(\epsilon(\pi')(\tau) \uplus \tau)(x^\gamma) = e'(x^\gamma)({}_{S_{e'}\langle\{x^\gamma\}\rangle}{\upharpoonleft}\tau) = e(x^\gamma)({}_{S_e\langle\{x^\gamma\}\rangle}{\upharpoonleft}(\epsilon(\pi)(\tau) \uplus \tau)) = \epsilon(e)(\epsilon(\pi)(\tau) \uplus \tau)(x^\gamma)$. Q.e.d. (Claim 3)

Claim 4: For $\tau : \mathrm{V}_{\delta-} \longrightarrow \mathcal{A}$ and $y^{\delta+} \in \mathcal{V}_{\delta+}(G)$: $\quad \epsilon(e')(\epsilon(\pi')(\tau) \uplus \tau)(\varsigma(y^{\delta+})) = \epsilon(\pi)(\tau)(y^{\delta+})$.

Proof of Claim 4: Since $\varsigma(y^{\delta+}) \in \mathrm{ran}(\varsigma)$, by the discussion of the second case of the definition of e', we have $\epsilon(e')(\epsilon(\pi')(\tau) \uplus \tau)(\varsigma(y^{\delta+})) = e'(\varsigma(y^{\delta+}))({}_{S_{e'}\langle\{\varsigma(y^{\delta+})\}\rangle}{\upharpoonleft}\tau) = \pi(\varsigma^{-1}(\varsigma(y^{\delta+})))({}_{S_\pi\langle\{\varsigma^{-1}(\varsigma(y^{\delta+}))\}\rangle}{\upharpoonleft}\tau) = \pi(y^{\delta+})({}_{S_\pi\langle\{y^{\delta+}\}\rangle}{\upharpoonleft}\tau) = \epsilon(\pi)(\tau)(y^{\delta+})$. Q.e.d. (Claim 4)

Claim 5: $G\varsigma$ is $(\pi', e', \mathcal{A})$-valid.

Proof of Claim 5: Let $\tau : \mathrm{V}_{\delta-} \longrightarrow \mathcal{A}$ be arbitrary. First by the Substitution-Lemma, second by Claim 3, $\mathcal{V}_{\delta+}(G) \subseteq \mathrm{dom}(\varsigma)$, Claim 4, and third as G is $(\pi, e, \mathcal{A})$-valid, we get:

$$\mathrm{eval}(\mathcal{A} \ \uplus \ \epsilon(e')(\epsilon(\pi')(\tau) \uplus \tau) \ \uplus \ \epsilon(\pi')(\tau) \ \uplus \ \tau)(G\varsigma) \ = $$
$$\mathrm{eval}\left(\begin{array}{ll} & \mathcal{A} \\ \uplus & \epsilon(e')(\epsilon(\pi')(\tau) \uplus \tau) \\ \uplus & {}_{\mathrm{V}_{\delta+} \setminus \mathrm{dom}(\varsigma)}{\upharpoonleft}(\epsilon(\pi')(\tau)) \ \uplus \ \varsigma \circ (\epsilon(e')(\epsilon(\pi')(\tau) \uplus \tau)) \\ \uplus & \tau \end{array}\right)(G) \ =$$
$$\mathrm{eval}\big(\ \mathcal{A} \ \uplus \ \epsilon(e)(\epsilon(\pi)(\tau) \uplus \tau) \ \uplus \ \epsilon(\pi)(\tau) \ \uplus \ \tau \ \big)(G) = \mathsf{TRUE}$$

Q.e.d. (Claim 5)

Claim 6: $G\varsigma$ is R'-valid in $\mathcal{A}$.

Proof of Claim 6: First note that by Claim 1, e' is an $(\mathcal{A}, R')$-valuation.
Let $\tau' : \mathrm{V}_\delta \rightarrow \mathcal{A}$ be arbitrary. When we set $\delta := {}_{\mathrm{V}_{\delta+}}{\upharpoonleft}\tau'$ and $\tau := {}_{\mathrm{V}_{\delta-}}{\upharpoonleft}\tau'$, we get
$\mathrm{eval}(\mathcal{A} \uplus \epsilon(e')(\tau') \uplus \tau')(G\varsigma) = \mathrm{eval}(\mathcal{A} \uplus \epsilon(e')(\epsilon(\pi')(\tau) \uplus \tau) \uplus \epsilon(\pi')(\tau) \uplus \tau)(G\varsigma) = \mathsf{TRUE}$, where the last step is due to Claim 5. Q.e.d. (Claim 6)

Now we conclude that $G\varsigma$ is $(\emptyset, R')$-valid in $\mathcal{A}$ (by Claim 1, Claim 2, and Claim 5) and R'-valid in $\mathcal{A}$ (by Claim 6). **Q.e.d. (Lemma 2.29)**

Proof of Lemma 2.31

(1), (2), (3), and (4) are trivial.

(5a): As G_0 is (C', R')-valid in $\mathcal{A}$, there is an $(\mathcal{A}, R')$-valuation e and some π s.t. π is $(e, \mathcal{A})$-compatible with (C', R') and G_0 is $(\pi, e, \mathcal{A})$-valid. By Lemma 2.26, e is also an $(\mathcal{A}, R)$-valuation and π is also $(e, \mathcal{A})$-compatible with (C, R). Thus, G_0 is (C, R)-valid in $\mathcal{A}$.

(5b): Suppose that e is an $(\mathcal{A}, R')$-valuation and π is $(e, \mathcal{A})$-compatible with (C', R'), and that G_1 is $(\pi, e, \mathcal{A})$-valid. By Lemma 2.26, e is also an $(\mathcal{A}, R)$-valuation and π is also $(e, \mathcal{A})$-compatible with (C, R). Thus, since G_0 (C, R)-reduces to G_1, also G_0 is $(\pi, e, \mathcal{A})$-valid as was to be shown.

(6): By Lemma B.6, setting $O := \emptyset$ and $N := \emptyset$. **Q.e.d. (Lemma 2.31)**

Proof of Lemma 2.37

(1), (2), (3), and (4) are trivial.

(5): Let $\mathcal{A} \in \mathrm{K}$, $S \in G_0$, let e be an $(\mathcal{A}, R')$-valuation, and π be $(e, \mathcal{A})$-compatible with (C', R'). Suppose that (S_0, τ_0) is an $(\pi, e, \mathcal{A})$-counterexample. By Lemma 2.26, e is also an $(\mathcal{A}, R)$-valuation and π is also $(e, \mathcal{A})$-compatible with (C, R). By assumption, $G_0 \rightharpoonup_{C,R} (G_1, L_1)$. Thus, there is some $(\pi, e, \mathcal{A})$-counterexample (S_1, τ_1) with $S_1 \in L_1$ or $S_1 \in G_1$ and in the latter case (S_1, τ_1) is $(\pi, e, \mathcal{A})$-smaller than (S_0, τ_0).

(6): By Lemma B.7, setting $O := \emptyset$ and $N := \emptyset$.

(7): To show $H_1 \rightharpoonup_{C,R} (G_1, L_1)$, let $\mathcal{A} \in \mathrm{K}$, e be some $(\mathcal{A}, R)$-valuation, and π be $(e, \mathcal{A})$-compatible with (C, R). W.l.o.g. we may assume that L_1 does not have an $(\pi, e, \mathcal{A})$-counterexample. Let D be the class of $(\pi, e, \mathcal{A})$-counterexamples (S_0, τ_0) with $S_0 \in H_1$ for which there is no $(\pi, e, \mathcal{A})$-counterexample (S_1, τ_1) s.t. $S_1 \in G_1$ and (S_1, τ_1) is $(\pi, e, \mathcal{A})$-smaller than (S_0, τ_0). It suffices to show that D is empty. We show this by the Method of Descente Infinie on the meta-level. Suppose there is some meta-counterexample $((\mathit{\Gamma}_0, (w_0, <_0, \lesssim_0)), \tau_0) \in D$. Due to the assumed $H_1 \mapsto_{C,R} (H_1, G_1, L_1)$, there must be an $(\pi, e, \mathcal{A})$-counterexample $((\mathit{\Gamma}_1, (w_1, <_1, \lesssim_1)), \tau_1)$ s.t. $(\mathit{\Gamma}_1, (w_1, <_1, \lesssim_1)) \in H_1$ and $((\mathit{\Gamma}_1, (w_1, <_1, \lesssim_1)), \tau_1)$ is strictly $(\pi, e, \mathcal{A})$-smaller than $((\mathit{\Gamma}_0, (w_0, <_0, \lesssim_0)), \tau_0)$. The latter means that there are $\lhd$ and $\underset{\sim}{\lhd}$ s.t., for $i \in \{0, 1\}$, $\delta_i := \epsilon(\pi)(\tau_i) \uplus \tau_i$, $\mathcal{B}_i := \mathcal{A} \uplus \epsilon(e)(\delta_i) \uplus \delta_i$, $\bar{w}_i := \mathrm{eval}(\mathcal{B}_i)(w_i)$, we have $\lhd = \mathrm{eval}(\mathcal{B}_i)(<_i)$, $\underset{\sim}{\lhd} = \mathrm{eval}(\mathcal{B}_i)(\lesssim_i)$, $\bar{w}_1 \lhd^+ \bar{w}_0$, and $\lhd$ is wellfounded. By Lemma 2.1, $\lhd^+$ is a wellfounded ordering.

$((\mathit{\Gamma}_1, (w_1, <_1, \lesssim_1)), \tau_1) \in D$: In this case we have found the meta-counterexample we are looking for. It is important that we indeed have a *single* meta-induction ordering here, which is defined as follows: $((\mathit{\Gamma}'_0, (w'_0, <'_0, \lesssim'_0)), \tau'_0)$ is strictly smaller than $((\mathit{\Gamma}'_1, (w'_1, <'_1, \lesssim'_1)), \tau'_1)$ if for $i \in \{0, 1\}$, $\delta'_i := \epsilon(\pi)(\tau'_i) \uplus \tau'_i$, $\mathcal{B}'_i := \mathcal{A} \uplus \epsilon(e)(\delta'_i) \uplus \delta'_i$, $\bar{w}'_i := \mathrm{eval}(\mathcal{B}'_i)(w'_i)$, we have some wellfounded $\lhd'$ with $\lhd' = \mathrm{eval}(\mathcal{B}'_i)(<'_i)$ and $\bar{w}_1 \lhd'^+ \bar{w}_0$.

$((\mathit{\Gamma}_1, (w_1, <_1, \lesssim_1)), \tau_1) \notin D$: In this case, there must be some $(\pi, e, \mathcal{A})$-counterexample $((\mathit{\Gamma}_2, (w_2, <_2, \lesssim_2)), \tau_2)$ s.t. $(\mathit{\Gamma}_2, (w_2, <_2, \lesssim_2)) \in G_1$ and $((\mathit{\Gamma}_2, (w_2, <_2, \lesssim_2)), \tau_2)$ is $(\pi, e, \mathcal{A})$-smaller than $((\mathit{\Gamma}_1, (w_1, <_1, \lesssim_1)), \tau_1)$. The latter means, for $\delta_2 := \epsilon(\pi)(\tau_2) \uplus \tau_2$, $\mathcal{B}_2 := \mathcal{A} \uplus \epsilon(e)(\delta_2) \uplus \delta_2$, $\bar{w}_2 := \mathrm{eval}(\mathcal{B}_2)(w_2)$, we have $\lhd = \mathrm{eval}(\mathcal{B}_2)(<_2)$, $\underset{\sim}{\lhd} = \mathrm{eval}(\mathcal{B}_2)(\lesssim_2)$, and $\bar{w}_2\ (\underset{\sim}{\lhd} \cup \lhd)^*\ \bar{w}_1$. But then we also have $\bar{w}_2\ (\underset{\sim}{\lhd} \cup \lhd)^*\ \bar{w}_0$. This, however, contradicts $((\mathit{\Gamma}_0, (w_0, <_0, \lesssim_0)), \tau_0) \in D$. **Q.e.d. (Lemma 2.37)**

Proof of Theorem 2.44

As $\emptyset$ is an $\emptyset$-choice-condition, $(\emptyset,\emptyset,\emptyset,\emptyset,\emptyset)$ vacuously satisfies the invariant for soundness.

For the iteration steps, let $(i'',((\Gamma'',\aleph''),t''))\in F'$ be arbitrary. Assuming the invariant for soundness of (F,C,R,L,H) and using the abbreviations

$$\begin{array}{rcl|rcl} I & := & H^{*}\langle\!\langle\{i''\}\rangle\!\rangle & I' & := & H'^{*}\langle\!\langle\{i''\}\rangle\!\rangle \\ A & := & \mathrm{Goals}(\mathrm{Trees}(\langle I'\rangle F)) & A' & := & \mathrm{Goals}(\mathrm{Trees}(\langle I'\rangle F')) \\ & & & B' & := & \mathrm{Propos}(\langle L'\langle I'\rangle\rangle F'). \end{array}$$

we have to show that C' is an R'-choice-condition and that $\{(\Gamma'',\aleph'')\}\rightarrow_{C',R'}(A',B')$.

Hypothesizing: Note that F' is a partial function on $\mathbf{N}_{+}$ just like F because of $i\in\mathbf{N}_{+}\setminus\mathrm{dom}(F)$. Note that R is a variable-condition and that R^{+} is a wellfounded ordering because C is an R-choice-condition (because (F,C,R,L,H) is assumed to be a proof forest).

$i''\in\mathrm{dom}(F)$: By assumption,

$$\{(\Gamma'',\aleph'')\}\rightarrow_{C,R}(\mathrm{Goals}(\mathrm{Trees}(\langle I\rangle F)),\mathrm{Propos}(\langle L\langle I\rangle\rangle F)).$$

As (C',R') is an extension of (C,R) and by Lemma 2.37(5), this means

$$\{(\Gamma'',\aleph'')\}\rightarrow_{C',R'}(\mathrm{Goals}(\mathrm{Trees}(\langle I\rangle F)),\mathrm{Propos}(\langle L\langle I\rangle\rangle F)).$$

Due to $H=H'$ we have $I=I'$, and then due to $L=L'$ and $F\subseteq F'$, we have

$$\mathrm{Goals}(\mathrm{Trees}(\langle I\rangle F))\subseteq A' \text{ and } \mathrm{Propos}(\langle L\langle I\rangle\rangle F)\subseteq B'.$$

Thus, by Lemma 2.37(2), we have

$$\mathrm{Goals}(\mathrm{Trees}(\langle I\rangle F))\rightarrow_{C',R'}(A',\emptyset) \text{ and } \mathrm{Propos}(\langle L\langle I\rangle\rangle F)\rightarrow_{C',R'}(\emptyset,B').$$

Thus, by Lemma 2.37(3a,b), we have $\{(\Gamma'',\aleph'')\}\rightarrow_{C',R'}(A',B')$.

$i''=i$: Then $\{(\Gamma'',\aleph'')\}=\{(\Gamma,\aleph)\}=\mathrm{Goals}(\{t\})=\mathrm{Goals}(\{t''\})\subseteq A'\subseteq A'\cup B'$. Thus, by Lemma 2.37(2), $\{(\Gamma'',\aleph'')\}\rightarrow_{C',R'}(A',B')$.

Expansion: Note that $\mathrm{Propos}(\langle J\rangle F)=\mathrm{Propos}(\langle J\rangle F')$ for all $J\subseteq\mathbf{N}_{+}$.

Claim 1: $\mathrm{Propos}(\langle I'\rangle F)\rightarrow_{C',R'}(A,B')$.

Claim 2: $A\hookrightarrow_{C',R'}(\mathrm{Propos}(\langle I'\rangle F),A',B')$.

By Claim 1, Claim 2, and Lemma 2.37(3a), we get

$$\mathrm{Propos}(\langle I'\rangle F)\hookrightarrow_{C',R'}(\mathrm{Propos}(\langle I'\rangle F),A',B').$$

By Lemma 2.37(7), we get $\mathrm{Propos}(\langle I'\rangle F)\rightarrow_{C',R'}(A',B')$. Since $\{(\Gamma'',\aleph'')\}\subseteq\mathrm{Propos}(\langle I'\rangle F)$, we have $\{(\Gamma'',\aleph'')\}\rightarrow_{C',R'}(\mathrm{Propos}(\langle I'\rangle F),\emptyset)$ by Lemma 2.37(2). Thus, by Lemma 2.37(3a), we get $\{(\Gamma'',\aleph'')\}\rightarrow_{C',R'}(A',B')$.

Proof of Claim 1: By Lemma 2.37(4) it suffices to show $\mathrm{Propos}(\langle\!\langle\{i'''\}\rangle\!\rangle F)\rightarrow_{C',R'}(A,B')$ for any $i'''\in I'$. We have

$$\mathrm{Propos}(\langle\!\langle\{i'''\}\rangle\!\rangle F)\rightarrow_{C,R}(\mathrm{Goals}(\mathrm{Trees}(\langle I'''\rangle F)),\mathrm{Propos}(\langle L\langle I'''\rangle\rangle F))$$

for $I''':=H^{*}\langle\!\langle\{i'''\}\rangle\!\rangle$ by assumption. As (C',R') is an extension of (C,R) and by Lemma 2.37(5),

$$\mathrm{Propos}(\langle\!\langle\{i'''\}\rangle\!\rangle F)\rightarrow_{C',R'}(\mathrm{Goals}(\mathrm{Trees}(\langle I'''\rangle F)),\mathrm{Propos}(\langle L\langle I'''\rangle\rangle F)).$$

Due to $H\subseteq H'$, we have $I'''\subseteq H'^{*}\langle\!\langle\{i'''\}\rangle\!\rangle\subseteq H'^{*}\langle I'\rangle=I'$. Thus, $\mathrm{Goals}(\mathrm{Trees}(\langle I'''\rangle F))\subseteq A$ and (due to $L\subseteq L'$) $\mathrm{Propos}(\langle L\langle I'''\rangle\rangle F)\subseteq\mathrm{Propos}(\langle L'\langle I'\rangle\rangle F)=B'$. Thus, by Lemma 2.37(2), we get $\mathrm{Goals}(\mathrm{Trees}(\langle I'''\rangle F))\rightarrow_{C',R'}(A,\emptyset)$ and $\mathrm{Propos}(\langle L\langle I'''\rangle\rangle F)\rightarrow_{C',R'}(\emptyset,B')$. By Lemma 2.37(3a,b): $\mathrm{Propos}(\langle\!\langle\{i'''\}\rangle\!\rangle F)\rightarrow_{C',R'}(A,B')$. Q.e.d. (Claim 1)

Proof of Claim 2: If $i\notin I'$, then we have $A=A'$ and Claim 2 follows from Lemma 2.37(2). Thus, we may assume $i\in I'$. By construction of t' we have $A\backslash\{(\Delta,\beth)\}\subseteq A'$. Thus, by Lemma 2.37(2),

$$A\backslash\{(\Delta,\beth)\}\hookrightarrow_{C',R'}(\mathrm{Propos}(\langle I'\rangle F),A',B').$$

By assumption we have

$$\{(\Delta,\beth)\}\hookrightarrow_{C',R'}(\mathrm{Propos}(\langle N_{\mathrm{H}}\rangle F),G,\mathrm{Propos}(\langle N_{\mathrm{L}}\rangle F)).$$

By Lemma 2.37(4), we get Claim 2 due to $\mathrm{Propos}(\langle N_{\mathrm{H}}\rangle F)\subseteq\mathrm{Propos}(\langle I'\rangle F)$, $G\subseteq\mathrm{Goals}(\{t'\})=\mathrm{Goals}(\mathrm{Trees}(\langle\!\langle\{i\}\rangle\!\rangle F'))\subseteq A'$, and $\mathrm{Propos}(\langle N_{\mathrm{L}}\rangle F)\subseteq\mathrm{Propos}(\langle L'\langle I'\rangle\rangle F)=B'$, which hold due

to $N_{\mathrm{H}} \subseteq H'\langle\!\langle\{i\}\rangle\!\rangle \subseteq H'\langle I'\rangle = I'$, the construction of t', and $N_{\mathrm{L}} \subseteq L'\langle\!\langle\{i\}\rangle\!\rangle \subseteq L'\langle I'\rangle$, respectively.
Q.e.d. (Claim 2)

Instantiation: Not that here we take into account the generalization of the Instantiation rule of Definition 2.42 given by Definition B.8.

By Lemma 2.22, C' is an R'-choice-condition.
Set $O := D_{i''}$ and $N := \mathrm{dom}(C) \cap \langle(\mathrm{dom}(C) \cap \mathrm{dom}(\sigma)) \setminus O\rangle R^*$.
Claim 3: $O \subseteq \mathrm{dom}(C) \cap \mathrm{dom}(\sigma) \subseteq O \uplus N$, $\mathrm{dom}(C) \cap \langle N\rangle R^+ \subseteq N$, $N \subseteq \mathrm{dom}(C) \setminus O$, and $N \cap \mathcal{V}(\mathrm{Goals}(\mathrm{Trees}(\langle I\rangle F)), \mathrm{Propos}(\langle\{i''\}\cup L\langle I\rangle\rangle F)) = \emptyset$.
Proof of Claim 3: By definition of D_i and N, the first, second, and third statement are trivial with the exception of $N \cap O = \emptyset$, which we will show together with the last statement: Set $M := R^*\langle\mathcal{V}_{\delta+}(\mathrm{Goals}(\mathrm{Trees}(\langle I\rangle F)), \mathrm{Propos}(\langle\{i''\}\cup L\langle I\rangle\rangle F))\rangle$. It now suffices to show $N \cap M = \emptyset$. If $z_1^{\delta+} \in N$, there is some $z_0^{\delta+} \in (\mathrm{dom}(C) \cap \mathrm{dom}(\sigma)) \setminus O$ with $z_0^{\delta+}\ R^*\ z_1^{\delta+}$, but then, if $z_1^{\delta+} \in M$, we get $z_0^{\delta+} \in M$ and the contradictory $z_0^{\delta+} \in O$ by definition of O.
Q.e.d. (Claim 3)
By assumption $\mathrm{Propos}(\langle\!\langle\{i''\}\rangle\!\rangle F) \rightharpoondown_{C,R} (\mathrm{Goals}(\mathrm{Trees}(\langle I\rangle F)), \mathrm{Propos}(\langle L\langle I\rangle\rangle F))$. Set $B'' := \bigcup_{y^{\delta+} \in O} \mathrm{Propos}(\langle\!\langle\{j_{y^{\delta+}}\}\rangle\!\rangle F')$. Then we have $\mathrm{Seq}(B'') = (\langle O\rangle Q_C)\sigma$ according to the requirements of the Instantiation rule. By Claim 3 we can apply Lemma B.7 to get:

$$\begin{aligned}\{(\mathit{\Gamma}'', \aleph'')\} = \mathrm{Propos}(\langle\!\langle\{i''\}\rangle\!\rangle F)\sigma \rightharpoondown_{C',R'} &(\mathrm{Goals}(\mathrm{Trees}(\langle I\rangle F))\sigma, \mathrm{Propos}(\langle L\langle I\rangle\rangle F)\sigma \cup B'')\\ = \quad &(\mathrm{Goals}(\mathrm{Trees}(\langle I\rangle F')), \mathrm{Propos}(\langle L\langle I\rangle\rangle F') \cup B'')\\ = \quad &(A', \mathrm{Propos}(\langle L\langle I'\rangle\rangle F') \cup B''),\end{aligned}$$

the latter step being due to $I = I'$.
By definition of L' we have $\{\, j_{y^{\delta+}} \mid y^{\delta+} \in O \,\} \subseteq L'\langle\!\langle\{i''\}\rangle\!\rangle \subseteq L'\langle I'\rangle$. Thus, we have $B'' \subseteq B'$. Moreover, due to $L \subseteq L'$, we have $\mathrm{Propos}(\langle L\langle I'\rangle\rangle F') \subseteq B'$. Together this implies $\mathrm{Propos}(\langle L\langle I'\rangle\rangle F') \cup B'' \rightharpoondown_{C',R'} (\emptyset, B')$, by Lemma 2.37(2). By Lemma 2.37(3b) we get $\{(\mathit{\Gamma}'', \aleph'')\} \rightharpoondown_{C',R'} (A', B')$.
Q.e.d. (Theorem 2.44)

Proof of Theorem 2.45

Let $\mathcal{A} \in \mathrm{K}$ be arbitrary. Since $\mathcal{AX}$ is $\mathrm{V}_\gamma \times \mathrm{V}_\delta$-valid in $\mathcal{A}$ (cf. Definition 2.38) and C is an R-choice-condition, $\mathcal{AX}$ is (C, R)-valid in $\mathcal{A}$ by Lemma 2.28. By definition, this means that there is some $(\mathcal{A}, R)$-valuation e and some π that is $(e, \mathcal{A})$-compatible with (C, R) s.t. $\mathcal{AX}$ is $(\pi, e, \mathcal{A})$-valid.
Claim 1: For i' with $i'\,(L \cup H)^*\,i$ and for $(i', ((\mathit{\Gamma}', \aleph'), t')) \in F$: $\mathit{\Gamma}'$ is $(\pi, e, \mathcal{A})$-valid.
Proof of Claim 1: By induction on i' in $L \circ H^*$: Set $I := H^*\langle\!\langle\{i'\}\rangle\!\rangle$. Due to $I \subseteq (L \cup H)^*\langle\!\langle\{i\}\rangle\!\rangle$ and by the closedness assumption of the theorem we have
$\mathrm{Seq}(\mathrm{Goals}(\mathrm{Trees}(\langle I\rangle F))) \subseteq \mathrm{Seq}(\mathrm{Goals}(\mathrm{Trees}(\langle (L \cup H)^*\langle\!\langle\{i\}\rangle\!\rangle\rangle F))) \subseteq \mathcal{AX}$. Thus,
$\mathrm{Seq}(\mathrm{Goals}(\mathrm{Trees}(\langle I\rangle F)))$ is $(\pi, e, \mathcal{A})$-valid. By induction hypothesis,
$\mathrm{Seq}(\mathrm{Propos}(\langle L\langle I\rangle\rangle F))$ is $(\pi, e, \mathcal{A})$-valid. Together this means that
$\mathrm{Seq}(\mathrm{Goals}(\mathrm{Trees}(\langle I\rangle F)) \cup \mathrm{Propos}(\langle L\langle I\rangle\rangle F))$ is $(\pi, e, \mathcal{A})$-valid, too. (Note that the last step would not be possible for (C, R)-validity instead of $(\pi, e, \mathcal{A})$-validity.)

Since $(i', ((\mathit{\Gamma}', \aleph'), t')) \in F$ and (F, C, R, L, H) satisfies the invariant for soundness, $\{(\mathit{\Gamma}', \aleph')\} \rightharpoondown_{C,R} (\mathrm{Goals}(\mathrm{Trees}(\langle I\rangle F)), \mathrm{Propos}(\langle L\langle I\rangle\rangle F))$. All in all, by Lemma 2.37(1b), $\mathit{\Gamma}'$ is $(\pi, e, \mathcal{A})$-valid.
Q.e.d. (Claim 1)
For $i' = i$, Claim 1 says that $\mathit{\Gamma}$ is (C, R)-valid in $\mathcal{A}$. **Q.e.d. (Theorem 2.45)**

Proof of Theorem 2.48

The empty proof forest trivially satisfies the invariant for safeness.

Hypothesizing: When we assume the old trees from F to satisfy the invariant for safeness for (C, R), then they also satisfy it for (C', R') by Lemma 2.31(5b) because (C', R') is an extension of (C, R). The new tree $(i, ((\mathit{\Gamma}, \aleph), t))$ satisfies the invariant for safeness because $\mathrm{Seq}(\mathrm{Goals}(\{t\})) = \{\mathit{\Gamma}\}$ and $\{\mathit{\Gamma}\}$ (C', R')-reduces to $\{\mathit{\Gamma}\}$ by Lemma 2.31(2).

Expansion: When we assume the non-expanded trees to satisfy the invariant for safeness for (C, R), then they also satisfy it for the extension (C', R') of (C, R) by Lemma 2.31(5b). For the new tree $(i, (\mathit{\Gamma}, t'))$ we have to show that $\mathrm{Seq}(\mathrm{Goals}(\{t'\}))$ (C', R')-reduces to $\{\mathit{\Gamma}\}$.

Claim 1: $\mathrm{Seq}(G)$ (C', R')-reduces to $\{\mathit{\Delta}\}$.

Proof of Claim 1: In case of a sequent calculus this is given by the additional requirement of safeness of the Expansion step. In case of a tableau calculus we have $\mathrm{Seq}(G) = \{\, \mathit{\Pi\Delta} \mid \mathit{\Pi} \in M \,\}$, and the claim follows because because $(\pi, e, \mathcal{A})$-validity of $\mathit{\Delta}$ implies $(\pi, e, \mathcal{A})$-validity of $\mathit{\Pi\Delta}$. Q.e.d. (Claim 1)

Claim 2: $\mathrm{Seq}(\mathrm{Goals}(\{t'\}))$ (C', R')-reduces to $\mathrm{Seq}(\mathrm{Goals}(\{t\}))$.

Proof of Claim 2: As $\mathrm{Goals}(\{t'\}) \setminus G \subseteq \mathrm{Goals}(\{t\})$, we have $\mathrm{Seq}(\mathrm{Goals}(\{t'\})) \setminus \mathrm{Seq}(G) \subseteq \mathrm{Seq}(\mathrm{Goals}(\{t'\}) \setminus G) \subseteq \mathrm{Seq}(\mathrm{Goals}(\{t\}))$, so that $\mathrm{Seq}(\mathrm{Goals}(\{t'\})) \setminus \mathrm{Seq}(G)$ (C', R')-reduces to $\mathrm{Seq}(\mathrm{Goals}(\{t\}))$ by Lemma 2.31(2). Thus, by Claim 1, the claim follows by Lemma 2.31(4) due to $\mathit{\Delta} \in \mathrm{Seq}(\mathrm{Goals}(\{t\}))$. Q.e.d. (Claim 2)

When we assume the old tree $(i, ((\mathit{\Gamma}, \aleph), t))$ to satisfy the invariant for safeness for (C, R), then $\mathrm{Seq}(\mathrm{Goals}(\{t\}))$ (C', R')-reduces to $\{\mathit{\Gamma}\}$ by Lemma 2.31(5b). By Lemma 2.31(3), together with Claim 2 this implies that $\mathrm{Seq}(\mathrm{Goals}(\{t'\}))$ (C', R')-reduces to $\{\mathit{\Gamma}\}$, as was to be shown.

Instantiation: Not that here we take into account the generalization of the Instantiation rule of Definition 2.42 given by Definition B.8.

Assume any old tree $(i, ((\mathit{\Gamma}, \aleph), t)) \in F$ to satisfy the invariant for safeness for (C, R), i.e. $\mathrm{Seq}(\mathrm{Goals}(\{t\}))$ (C, R)-reduces to $\{\mathit{\Gamma}\}$. Set $O := D_i$ and $N := \mathrm{dom}(C) \cap \langle(\mathrm{dom}(C) \cap \mathrm{dom}(\sigma)) \setminus O\rangle R^*$.

Claim 3: $O \subseteq \mathrm{dom}(C) \cap \mathrm{dom}(\sigma) \subseteq O \uplus N$, $\mathrm{dom}(C) \cap \langle N\rangle R^+ \subseteq N$, $N \subseteq \mathrm{dom}(C) \setminus O$, and $N \cap \mathcal{V}(\mathrm{Goals}(\mathrm{Trees}(\langle I\rangle F)), \mathrm{Propos}(\langle\{i\} \cup L\langle I\rangle\rangle F)) = \emptyset$.

Proof of Claim 3: Just like the proof of Claim 3 in the proof of Theorem 2.44. Q.e.d. (Claim 3)

By Lemma B.6 and Claim 3, $\mathrm{Seq}(\mathrm{Goals}(\{t\sigma\}))$ (C', R')-reduces to $\{\mathit{\Gamma}\sigma\} \cup (\langle O\rangle Q_C)\sigma$. As the Instantiation step is safe by assumption, by Theorem 2.44 and Theorem 2.45, $(\langle O\rangle Q_C)\sigma$ is (C', R')-valid. Thus, $\mathrm{Seq}(\mathrm{Goals}(\{t\sigma\}))$ (C', R')-reduces to $\{\mathit{\Gamma}\sigma\}$, as was to be shown. **Q.e.d. (Theorem 2.48)**

Proof of Theorem 2.49

To illustrate our techniques, we just prove the first rule of each kind to be a safe sub-rule of the Expansion rule, all other case are similar.

Due to $\mathrm{ran}(G) = \{\sqsupset\}$, for the α-, β-, γ-, Rewrite-, and Cut-rules, it suffices to show that, for each Σ-structure $\mathcal{A}$, each $(\mathcal{A}, R)$-valuation e, each π that is $(e, \mathcal{A})$-compatible with (C, R), each $\tau : \mathrm{V}_{\delta^-} \to \mathcal{A}$, and for $\delta := \epsilon(\pi)(\tau) \uplus \tau$, the $(\delta, e, \mathcal{A})$-validity of $\{\mathit{\Delta}\}$ is logically equivalent to $(\delta, e, \mathcal{A})$-validity of $\mathrm{Seq}(G)$.

α-rule: $(\delta, e, \mathcal{A})$-validity of $\{\mathit{\Gamma}\ (A{\vee}B)\ \mathit{\Pi}\}$ is indeed logically equivalent to $(\delta, e, \mathcal{A})$-validity of $\{A\ B\ \mathit{\Gamma}\ \mathit{\Pi}\}$.

β-rule: $(\delta, e, \mathcal{A})$-validity of $\{\mathit{\Gamma}\ (A{\wedge}B)\ \mathit{\Pi}\}$ is indeed logically equivalent to $(\delta, e, \mathcal{A})$-validity of $\{A\ \mathit{\Gamma}\ \mathit{\Pi},\quad B\ [\,\overline{A}\,]\ \mathit{\Gamma}\ \mathit{\Pi}\}$.

γ-rule: $(\delta, e, \mathcal{A})$-validity of $\{\mathit{\Gamma}\ \ \exists x.\ A\ \ \mathit{\Pi}\}$ is indeed logically equivalent to $(\delta, e, \mathcal{A})$-validity of $\{A\{x{\mapsto}t\}\ \ \mathit{\Gamma}\ \exists x.\ A\ \ \mathit{\Pi}\}$.

The implication from left to right is simple because the former sequent is a sub-sequent of the latter. For the other direction, assume that $A\{x{\mapsto}t\}$ is $(\delta, e, \mathcal{A})$-valid. Let $y^\delta \in \mathrm{V}_\delta \backslash \mathcal{V}(A)$. Then, since $A\{x{\mapsto}y^\delta\}\{y^\delta{\mapsto}t\}$ is equal to $A\{x{\mapsto}t\}$, we know that $A\{x{\mapsto}y^\delta\}\{y^\delta{\mapsto}t\}$ is valid in $\mathcal{A} \uplus \epsilon(e)(\delta) \uplus \delta$. Then, by the Substitution-Lemma, $A\{x{\mapsto}y^\delta\}$ is valid in $\mathcal{A} \uplus \epsilon(e)(\delta) \uplus \delta'$ for $\delta' : \mathrm{V}_\delta \to \mathcal{A}$ given by ${}_{\mathrm{V}_\delta \backslash \{y^\delta\}}{\upharpoonright}\delta' := {}_{\mathrm{V}_\delta \backslash \{y^\delta\}}{\upharpoonright}\delta$ and $\delta'(y^\delta) := \mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta) \uplus \delta)(t)$. By the standard semantical definition of $\exists$ (cf. e.g. Enderton (1973), p. 82) and since binding of x cannot occur in A (as $\exists x.\ A$ is a formula in our restricted sense, cf. Section 2.1.3), this means that $\exists x.\ (A\{x{\mapsto}y^\delta\}\{y^\delta{\mapsto}x\})$ is valid in $\mathcal{A} \uplus \epsilon(e)(\delta) \uplus \delta$. Since y^δ does not occur in A, this formula is equal to $\exists x.\ A$, which means that the former sequent is $(\delta, e, \mathcal{A})$-valid.

Rewrite-rule: We have to show that $(\delta, e, \mathcal{A})$-validity of $\{\mathit{\Gamma}\ \ A[s]\ \ \mathit{\Pi}\ \ B\ \ \mathit{\Lambda}\}$ is logically equivalent to $(\delta, e, \mathcal{A})$-validity of $\{A[t]\ \ \mathit{\Gamma}\ \ \mathit{\Pi}\ \ B\ \ \mathit{\Lambda}\}$.

If $\mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta) \uplus \delta)(s) \neq \mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta) \uplus \delta)(t)$, then both are $(\delta, e, \mathcal{A})$-valid because B is. Note that B is of the form $(s{\neq}t)$ or $(t{\neq}s)$.

Otherwise, we set $a := \mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta) \uplus \delta)(s)$, choose some $z^\delta \in \mathrm{V}_\delta \backslash \mathcal{V}(A[s])$, and define $\delta' : \mathrm{V}_\delta \to \mathcal{A}$ by ${}_{\mathrm{V}_\delta \backslash \{z^\delta\}}{\upharpoonright}\delta' := {}_{\mathrm{V}_\delta \backslash \{z^\delta\}}{\upharpoonright}\delta$ and $\delta'(z^\delta) := a$. Then $a = \mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta) \uplus \delta)(t)$. Moreover, by the Substitution-Lemma:

$$\begin{array}{ll}
\mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta) \uplus \delta)(A[s]) & = \\
\mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta) \uplus \delta)(A[z^\delta]\{z^\delta{\mapsto}s\}) & = \\
\mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta) \uplus \delta')(A[z^\delta]) & = \\
\mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta) \uplus \delta)(A[z^\delta]\{z^\delta{\mapsto}t\}) & = \\
\mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta) \uplus \delta)(A[t]). &
\end{array}$$

Note that the usual problems with variables getting captured by binders cannot occur in our context, because the unbound occurrence of variables from $\mathrm{V}_{\mathrm{bound}}$ in formulas (like $(s{\neq}t)$) is not permitted, cf. Section 2.1.3.

Cut: Trivial.

δ-rule: Note that in this proof, we only use the weaker conditions on the occurrence of x^{δ^-} given in Note 4.

Claim 1: (C', R') is an extension of (C, R).

Proof of Claim 1: Since (F, C, R, L, H) is a proof forest, C is an R-choice-condition. Moreover, $C{\subseteq}C'$ and $R{\subseteq}R'$ are trivial, because the rule says that $C'' := \emptyset$, $R'' := \mathcal{V}_{\gamma\delta^+}(A, \mathit{\Gamma}\mathit{\Pi}, \sqsupset) \times \{x^{\delta^-}\}$, $C' := C \cup C'$, $R' := R \cup R''$. Thus, we only have to show that C' is an R'-choice-condition. As $C' = C$, we only have to show that R' is wellfounded. As $\mathrm{ran}(R'') = \{x^{\delta^-}\}$ and as $\{x^{\delta^-}\} \cap \mathrm{dom}(R) = \emptyset$ is required in Note 4, we have $R'' {\circ} R = \emptyset$. As $\mathrm{ran}(R'') \cap \mathrm{dom}(R'') \subseteq \mathrm{V}_{\delta^-} \cap (\mathrm{V}_\gamma \cup \mathrm{V}_{\delta^+}) = \emptyset$, we have $R'' {\circ} R'' = \emptyset$. Therefore, as R is wellfounded, R' is wellfounded, too. Q.e.d. (Claim 1)

Now, we have to show that

$$\{(\mathit{\Gamma}\ \ \forall x.\ A\ \ \mathit{\Pi},\ \sqsupset)\}\ \to_{C',R'}\ (\{(A\{x{\mapsto}x^{\delta^-}\}\ \mathit{\Gamma}\ \mathit{\Pi},\ \sqsupset)\}, \emptyset)$$

Let e and π be arbitrary s.t. e is an $(\mathcal{A}, R')$-valuation and π is $(e, \mathcal{A})$-compatible with (C', R'). Assume that $((\mathit{\Gamma}\ \ \forall x.\ A\ \ \mathit{\Pi},\ \sqsupset), \tau)$ is an $(\pi, e, \mathcal{A})$-counterexample. Then, $\mathit{\Gamma}\mathit{\Pi}$ is invalid in $\mathcal{A} \uplus \epsilon(e)(\epsilon(\pi)(\tau) \uplus \tau) \uplus \epsilon(\pi)(\tau) \uplus \tau$.

Claim 2: $\mathit{\Gamma}\mathit{\Pi}$ is invalid in $\mathcal{A}\uplus\epsilon(e)(\epsilon(\pi)(\tau')\uplus\tau')\uplus\epsilon(\pi)(\tau')\uplus\tau'$ and

$$\begin{aligned}&\mathrm{eval}(\mathcal{A}\uplus\epsilon(e)(\epsilon(\pi)(\tau')\uplus\tau')\uplus\epsilon(\pi)(\tau')\uplus\tau')(\sqsupset)\\ =\ &\mathrm{eval}(\mathcal{A}\uplus\epsilon(e)(\epsilon(\pi)(\tau)\uplus\tau)\uplus\epsilon(\pi)(\tau)\uplus\tau)(\sqsupset)\end{aligned}$$

for all $\tau':\mathrm{V}_{\delta^-}\to\mathcal{A}$ with ${}_{\mathrm{V}_{\delta^-}\setminus\{x^{\delta^-}\}}{\upharpoonleft}\tau'={}_{\mathrm{V}_{\delta^-}\setminus\{x^{\delta^-}\}}{\upharpoonleft}\tau$.

Proof of Claim 2: Otherwise, there must be some $u\in\mathcal{V}_{\gamma\delta^+}(\mathit{\Gamma}\mathit{\Pi},\sqsupset)$ with $x^{\delta^-}\ S_\pi\circ S_e\ u$ (the first occurrence of τ' makes a difference) or $x^{\delta^-}\ S_e\ u$ (the second occurrence of τ' makes a difference) or $x^{\delta^-}\ S_\pi\ u$ when the third occurrence of τ' makes a difference. Note that the fourth occurrence of τ' cannot make a difference simply because x^{δ^-} does not occur in $\mathcal{V}(\mathit{\Gamma}\mathit{\Pi},\sqsupset)$ according to Note 4. Since $u\ R''\ x^{\delta^-}$, we know that $R'\cup S_e\cup S_\pi$ is not wellfounded, which contradicts π being $(e,\mathcal{A})$-compatible with (C',R'). Q.e.d. (Claim 2)

Now, if there is any $\tau':\mathrm{V}_{\delta^-}\to\mathcal{A}$ with ${}_{\mathrm{V}_{\delta^-}\setminus\{x^{\delta^-}\}}{\upharpoonleft}\tau'={}_{\mathrm{V}_{\delta^-}\setminus\{x^{\delta^-}\}}{\upharpoonleft}\tau$ s.t. $A\{x\mapsto x^{\delta^-}\}$ is invalid in $\mathcal{A}\uplus\epsilon(e)(\epsilon(\pi)(\tau')\uplus\tau')\uplus\epsilon(\pi)(\tau')\uplus\tau'$, then due to Claim 2 $((A\{x\mapsto x^{\delta^-}\}\ \mathit{\Gamma}\ \mathit{\Pi},\ \sqsupset),\tau')$ is the $(\pi,e,\mathcal{A})$-counterexample we are searching for. Thus, we only have to derive a contradiction from the assumption that $A\{x\mapsto x^{\delta^-}\}$ is valid in $\mathcal{A}\uplus\epsilon(e)(\epsilon(\pi)(\tau')\uplus\tau')\uplus\epsilon(\pi)(\tau')\uplus\tau'$ for all $\tau':\mathrm{V}_{\delta^-}\to\mathcal{A}$ with ${}_{\mathrm{V}_{\delta^-}\setminus\{x^{\delta^-}\}}{\upharpoonleft}\tau'={}_{\mathrm{V}_{\delta^-}\setminus\{x^{\delta^-}\}}{\upharpoonleft}\tau$.

Claim 4: $A\{x\mapsto x^{\delta^-}\}$ is valid in $\mathcal{A}\uplus\epsilon(e)(\epsilon(\pi)(\tau)\uplus\tau)\uplus\epsilon(\pi)(\tau)\uplus\tau'$ for all $\tau':\mathrm{V}_{\delta^-}\to\mathcal{A}$ with ${}_{\mathrm{V}_{\delta^-}\setminus\{x^{\delta^-}\}}{\upharpoonleft}\tau'={}_{\mathrm{V}_{\delta^-}\setminus\{x^{\delta^-}\}}{\upharpoonleft}\tau$.

Proof of Claim 4: Otherwise there must be some $u\in\mathcal{V}_{\gamma\delta^+}(A\{x\mapsto x^{\delta^-}\})$ with $x^{\delta^-}\ S_\pi\circ S_e\ u$ (the first occurrence of τ makes a difference) or $x^{\delta^-}\ S_e\ u$ (the second occurrence of τ makes a difference) or $x^{\delta^-}\ S_\pi\ u$ when the third occurrence of τ makes a difference. Since $u\ R''\ x^{\delta^-}$, we know that $R'\cup S_e\cup S_\pi$ is not wellfounded, which contradicts π being $(e,\mathcal{A})$-compatible with (C',R'). Q.e.d. (Claim 4)

By the standard semantical definition of $\forall$ (cf. e.g. Enderton (1973), p. 82) and since binding of x cannot occur in A (as $\forall x.\ A$ is a formula in our restricted sense, cf. Section 2.1.3), Claim 4 means that $\forall x.\ (A\{x\mapsto x^{\delta^-}\}\{x^{\delta^-}\mapsto x\})$ is valid in $\mathcal{A}\uplus\epsilon(e)(\epsilon(\pi)(\tau)\uplus\tau)\uplus\epsilon(\pi)(\tau)\uplus\tau$, i.e. $(\epsilon(\pi)(\tau)\uplus\tau,e,\mathcal{A})$-valid. Since x^{δ^-} does not occur in A according to Note 4, this formula is equal to $\forall x.\ A$, which contradicts $((\mathit{\Gamma}\ \ \forall x.\ A\ \ \mathit{\Pi},\ \sqsupset),\tau)$ being an $(\pi,e,\mathcal{A})$-counterexample.

Finally, for the safeness proof, assume that $\mathit{\Gamma}\ \ \forall x.\ A\ \ \mathit{\Pi}$ is $(\pi,e,\mathcal{A})$-valid. For arbitrary $\tau:\mathrm{V}_{\delta^-}\to\mathcal{A}$ we have to show that $A\{x\mapsto x^{\delta^-}\}\ \mathit{\Gamma}\ \mathit{\Pi}$ is $(\delta,e,\mathcal{A})$-valid for $\delta:=\epsilon(\pi)(\tau)\uplus\tau$. If some formula in $\mathit{\Gamma}\mathit{\Pi}$ is $(\delta,e,\mathcal{A})$-valid, then the latter sequent is $(\delta,e,\mathcal{A})$-valid, too. Otherwise, $\forall x.\ A$ is $(\delta,e,\mathcal{A})$-valid. Then, by the standard semantical definition of $\forall$, $A\{x\mapsto x^{\delta^-}\}$ is $(\delta,e,\mathcal{A})$-valid, too, as was to be shown.

Liberalized δ-rule: Note that in this proof, we only use the weaker conditions on the occurrence of x^{δ^+} given in Note 5.

Claim 5: (C',R') is an extension of (C,R).

Proof of Claim 5: Since (F,C,R,L,H) is a proof forest, C is an R-choice-condition. Moreover, $C{\subseteq}C'$ and $R{\subseteq}R'$ are trivial, because the rule says that $C'':=\{(x^{\delta^+},\ \overline{A\{x\mapsto x^{\delta^+}\}}\,)\}$, $R'':=\mathcal{V}_{\mathrm{free}}(A)\times\{x^{\delta^+}\}$, $C':=C\cup C'$, $R':=R\cup R''$. Thus, we only have to show that C' is an R'-choice-condition. As $x^{\delta^+}\in\mathrm{V}_{\delta^+}\setminus\mathrm{dom}(C)$ by Note 5, C' is a partial function on V_{δ^+}, too. As $\mathrm{ran}(R'')=\{x^{\delta^+}\}$ and as $\{x^{\delta^+}\}\cap\mathrm{dom}(R)=\emptyset$ by Note 5, we have $R''{\circ}R=\emptyset$. As $\mathrm{ran}(R'')\cap\mathrm{dom}(R'')=\{x^{\delta^+}\}\cap\mathcal{V}_{\mathrm{free}}(A)=\{x^{\delta^+}\}\cap\mathcal{V}(A)=\emptyset$ by Note 5, we have $R''{\circ}R''=\emptyset$. Therefore, as R is wellfounded, R' is a wellfounded, too. Moreover, for $y^{\delta^+}\in\mathrm{dom}(C')$, we either have $y^{\delta^+}\in\mathrm{dom}(C)$ and then $\mathcal{V}_{\mathrm{free}}(C'(y^{\delta^+}))\times\{y^{\delta^+}\}=\mathcal{V}_{\mathrm{free}}(C(y^{\delta^+}))\times\{y^{\delta^+}\}\subseteq R^*\subseteq R'^*$, or $y^{\delta^+}=x^{\delta^+}$ and then $\mathcal{V}_{\mathrm{free}}(C'(y^{\delta^+}))\times\{y^{\delta^+}\}=\mathcal{V}_{\mathrm{free}}(A\{x\mapsto x^{\delta^+}\})\times\{x^{\delta^+}\}\subseteq(\mathcal{V}_{\mathrm{free}}(A)\cup\{x^{\delta^+}\})\times\{x^{\delta^+}\}\subseteq R''^*\subseteq R'^*$. Q.e.d. (Claim 5)

Now, due to $\mathrm{ran}(G) = \{\beth\}$, it suffices to show that, for each Σ-structure $\mathcal{A}$, each $(\mathcal{A}, R')$-valuation e, each π that is $(e, \mathcal{A})$-compatible with (C', R'), each $\tau : \mathrm{V}_{\delta^-} \to \mathcal{A}$, and for $\delta := \epsilon(\pi)(\tau) \uplus \tau$, the $(\delta, e, \mathcal{A})$-validity of

$\mathit{\Gamma}\ \forall x.\ A\ \ \mathit{\Pi}$ is logically equivalent to $(\delta, e, \mathcal{A})$-validity of $A\{x{\mapsto}x^{\delta^+}\}\ \mathit{\Gamma}\ \mathit{\Pi}$.

For the soundness direction, we have to show that the former sequent is $(\delta, e, \mathcal{A})$-valid under the assumption that the latter is. If some formula in $\mathit{\Gamma\Pi}$ is $(\delta, e, \mathcal{A})$-valid, then the former sequent is $(\delta, e, \mathcal{A})$-valid, too. Otherwise, this means that $A\{x{\mapsto}x^{\delta^+}\}$ is $(\delta, e, \mathcal{A})$-valid. Since π is $(e, \mathcal{A})$-compatible with (C', R'), by Item 2 Definition 2.23, we know that $A\{x{\mapsto}x^{\delta^+}\}$ is $(\delta', e, \mathcal{A})$-valid for all $\delta' : \mathrm{V}_\delta \to \mathcal{A}$ with ${}_{\mathrm{V}_\delta \setminus \{x^{\delta^+}\}}{\upharpoonright}\delta' = {}_{\mathrm{V}_\delta \setminus \{x^{\delta^+}\}}{\upharpoonright}\delta$. This means that $A\{x{\mapsto}x^{\delta^+}\}$ is valid in $\mathcal{A} \uplus \epsilon(e)(\delta') \uplus \delta'$ for all $\delta' : \mathrm{V}_\delta \to \mathcal{A}$ with ${}_{\mathrm{V}_\delta \setminus \{x^{\delta^+}\}}{\upharpoonright}\delta' = {}_{\mathrm{V}_\delta \setminus \{x^{\delta^+}\}}{\upharpoonright}\delta$.

<u>Claim 6:</u> $A\{x{\mapsto}x^{\delta^+}\}$ is valid in $\mathcal{A} \uplus \epsilon(e)(\delta) \uplus \delta'$ for all $\delta' : \mathrm{V}_\delta \to \mathcal{A}$ with ${}_{\mathrm{V}_\delta \setminus \{x^{\delta^+}\}}{\upharpoonright}\delta' = {}_{\mathrm{V}_\delta \setminus \{x^{\delta^+}\}}{\upharpoonright}\delta$.

<u>Proof of Claim 6:</u> Otherwise we have $x^{\delta^+} S_e u^\gamma$ for some $u^\gamma \in \mathcal{V}_\gamma(A\{x{\mapsto}x^{\delta^+}\})$. But then $u^\gamma \in \mathcal{V}_{\mathrm{free}}(A)$ and then $u^\gamma R'' x^{\delta^+}$. This means that $R' \cup S_e$ is not wellfounded, which contradicts e being an $(\mathcal{A}, R')$-valuation. <u>Q.e.d. (Claim 6)</u>

By the standard semantical definition of $\forall$ (cf. e.g. Enderton (1973), p. 82) and since binding of x cannot occur in A (as $\forall x.\ A$ is a formula in our restricted sense, cf. Section 2.1.3), Claim 6 means that $\forall x.\ (A\{x{\mapsto}x^{\delta^+}\}\{x^{\delta^+}{\mapsto}x\})$ is valid in $\mathcal{A} \uplus \epsilon(e)(\delta) \uplus \delta$. Since x^{δ^+} does not occur in A by Note 5, this formula is equal to $\forall x.\ A$, which means that the former sequent is $(\delta, e, \mathcal{A})$-valid as was to be shown.

For the safeness direction, we have to show that the latter sequent is $(\delta, e, \mathcal{A})$-valid under the assumption that the former is. If some formula in $\mathit{\Gamma\Pi}$ is $(\delta, e, \mathcal{A})$-valid, then the latter sequent is $(\delta, e, \mathcal{A})$-valid, too. Otherwise, $\forall x.\ A$ is $(\delta, e, \mathcal{A})$-valid. Then, by the standard semantical definition of $\forall$, $A\{x{\mapsto}x^{\delta^+}\}$ is $(\delta, e, \mathcal{A})$-valid, too, as was to be shown. **Q.e.d. (Theorem 2.49)**

Proof of Theorem 2.51

Let $G := \{\ (\mathit{\Pi\Delta}, \beth) \mid \mathit{\Pi} \in M\ \}$ as in the Expansion rule in tableau trees. According to Definition 2.42 we have to show

$$\{(\mathit{\Delta}, \beth)\} \looparrowright_{C', R'} (\{(\mathit{\Phi}, \daleth)\}, G, \emptyset)$$

in case of “induction hypothesis application” and

$$\{(\mathit{\Delta}, \beth)\} \looparrowright_{C', R'} (\emptyset, G, \{(\mathit{\Phi}, \daleth)\})$$

in case of “lemma application”. According to Definition 2.36 and Definition 2.35 and due to $\mathrm{ran}(G) = \{\beth\}$, it is sufficient to show that, for $\mathcal{A} \in \mathrm{K}$, e an $(\mathcal{A}, R')$-valuation, and π $(e, \mathcal{A})$-compatible with (C', R'), for any $(\pi, e, \mathcal{A})$-counterexample $((\mathit{\Delta}, \beth), \tau)$, under the assumption that $\mathrm{Seq}(G)$ is $(\delta, e, \mathcal{A})$-valid for $\delta := \epsilon(\pi)(\tau) \uplus \tau$, there is an $(\pi, e, \mathcal{A})$-counterexample $((\mathit{\Phi}, \daleth), \tau')$ such that, for $\lhd := \mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta) \uplus \delta)(<)$, $\underset{\approx}{\lhd} := \mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta) \uplus \delta)(\lesssim)$, $\bar{w} := \mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta) \uplus \delta)(w)$, $\delta' := \epsilon(\pi)(\tau') \uplus \tau'$, we have (in case of hypothesis application only):

$$\lhd = \mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta') \uplus \delta')(<'),$$
$$\underset{\approx}{\lhd} = \mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta') \uplus \delta')(\lesssim'),$$
$$\mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta') \uplus \delta')(w') \lhd \bar{w},$$

and $\lhd \circ \underset{\approx}{\lhd} \subseteq \lhd^+$ and $\lhd$ is wellfounded.

Since, for all $\mathit{\Pi} \in M$, $\mathit{\Pi}\mathit{\Delta} \in \mathrm{Seq}(G)$ is assumed to be $(\delta, e, \mathcal{A})$-valid whereas $\mathit{\Delta}$ is assumed to be not, we know that M is $(\delta, e, \mathcal{A})$-valid. By the definition of M, this means that $\mathit{\Phi}\varrho$ is not $(\delta, e, \mathcal{A})$-valid (due to (1)) and (in case of hypothesis application only):

$$\begin{aligned} \lhd &= \mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta) \uplus \delta)(<'\varrho) && \text{(due to (4)),}\\ \precsim\!\!\lhd &= \mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta) \uplus \delta)(\precsim'\varrho) && \text{(due to (5)),}\\ &\mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta) \uplus \delta)(w'\varrho) \lhd \bar{w} && \text{(due to (2)),} \end{aligned}$$

and $\lhd \circ \precsim\!\!\lhd \subseteq \lhd^+$ (due to (6)), and $\lhd$ is wellfounded (due to (3)). To complete the proof, we have to get rid of the ϱ here by stepping from δ to δ' given by some appropriate τ' as indicated above.

Define $\tau'(y^{\delta^-}) := \left\{ \begin{array}{ll} \mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta) \uplus \delta)(\varrho(y^{\delta^-})) & \text{for } y^{\delta^-} \in Y \\ \tau(y^{\delta^-}) & \text{for } y^{\delta^-} \in \mathrm{V}_{\delta^-} \backslash Y \end{array} \right\}$.

Claim 1: For $v^{\delta^+} \in \mathcal{V}_{\delta^+}(\mathit{\Phi}, \urcorner)$ we have $\epsilon(\pi)(\tau)(v^{\delta^+}) = \epsilon(\pi)(\tau')(v^{\delta^+})$.

Proof of Claim 1: Otherwise there must be some $y^{\delta^-} \in Y$ with $y^{\delta^-}\ S_\pi\ v^{\delta^+}$. Since $v^{\delta^+} \in \mathcal{V}_{\delta^+}(\mathit{\Phi}, \urcorner)$ we have $v^{\delta^+}\ R'\ y^{\delta^-}$ by definition of Y. But then $R' \cup S_e \cup S_\pi$ is not wellfounded, which contradicts π being $(e, \mathcal{A})$-compatible with (C', R'). Q.e.d. (Claim 1)

Claim 3: For $x^\gamma \in \mathcal{V}_\gamma(\mathit{\Phi}, \urcorner)$ we have $\epsilon(e)(\delta)(x^\gamma) = \epsilon(e)(\delta')(x^\gamma)$.

Proof of Claim 3: Otherwise we have $\epsilon(e)(\epsilon(\pi)(\tau) \uplus \tau)(x^\gamma) \neq \epsilon(e)(\epsilon(\pi)(\tau') \uplus \tau')(x^\gamma)$. Then there must be some $y^{\delta^-} \in Y$ with $y^{\delta^-}\ S_\pi \circ S_e\ x^\gamma$ (i.e. the first occurrence of τ' makes a difference) or $y^{\delta^-}\ S_e\ x^\gamma$ when the second occurrence of τ' makes a difference. Since $x^\gamma \in \mathcal{V}_\gamma(\mathit{\Phi}, \urcorner)$ we have $x^\gamma\ R'\ y^{\delta^-}$ by definition of Y. But then $R' \cup S_e \cup S_\pi$ is not wellfounded, which contradicts π being $(e, \mathcal{A})$-compatible with (C', R'). Q.e.d. (Claim 3)

The respective values of $\mathit{\Phi}$, w', $<'$, and $\precsim'$ under $\mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta') \uplus \delta')$ are the same as the values of $\mathit{\Phi}$, w', $<'$, and $\precsim'$ under $\mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta) \uplus \epsilon(\pi)(\tau) \uplus \tau')$ by definition of δ', Claim 1, Claim 3, and the Explicitness-Lemma, which again are the same as the values of $\mathit{\Phi}\varrho$, $w'\varrho$, $<'\varrho$, and $\precsim'\varrho$ under $\mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta) \uplus \delta)$ by the Substitution-Lemma and the definition of δ. Thus, due to $\mathit{\Phi}\varrho$ not being $(\delta, e, \mathcal{A})$-valid, $((\mathit{\Phi}, \urcorner), \tau')$ is an $(\pi, e, \mathcal{A})$-counterexample with (in case of hypothesis application only):

$$\begin{aligned} \lhd &= \mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta) \uplus \delta)(<'\varrho) = \mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta') \uplus \delta')(<'),\\ \precsim\!\!\lhd &= \mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta) \uplus \delta)(\precsim'\varrho) = \mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta') \uplus \delta')(\precsim'),\\ &\mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta') \uplus \delta')(w') = \mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta) \uplus \delta)(w'\varrho) \lhd \bar{w}. \end{aligned}$$

Q.e.d. (Theorem 2.51)

Proof of Lemma B.1

Here we denote concatenation (product) of relations '$\circ$' simply by juxtaposition and assume it to have higher priority than any other binary operator.

(1): When e is an $(\mathcal{A}, R')$-valuation, $R' \cup S_e$ is wellfounded. In case of $R \subseteq R'$, we have $R \cup S_e \subseteq R' \cup S_e$ and $R \cup S_e$ is wellfounded, too.

(2): Set $\sigma' := {}_{\mathrm{V}_\gamma \backslash \mathrm{dom}(\sigma)}\!\upharpoonleft\!\mathrm{id} \cup {}_{\mathrm{V}_\gamma}\!\upharpoonleft\!\sigma$. Let e' be an $(\mathcal{A}, R')$-valuation. Define $S_e := S_{e'}({}_{\mathrm{V}_\gamma \backslash \mathrm{dom}(\sigma)}\!\upharpoonleft\!\mathrm{id} \cup \Gamma_\sigma\!\upharpoonright_{\mathrm{V}_\gamma}) \cup \Delta_\sigma\!\upharpoonright_{\mathrm{V}_\gamma}$ and the $(\mathcal{A}, R)$-valuation e by $(x \in \mathrm{V}_\gamma,\ \tau' : S_e\langle\!\{x\}\!\rangle \to \mathcal{A})$: $e(x)(\tau') := \mathrm{eval}(\mathcal{A} \uplus \epsilon(e')(\tau) \uplus \tau)(\sigma'(x))$ where $\tau : \mathrm{V}_\delta \to \mathcal{A}$ is an arbitrary extension of τ'. For this definition to be okay, we have to prove the following claims:

Claim 1: For $x \in \mathrm{V}_\gamma$, $y \in \mathcal{V}_\delta(\sigma'(x))$, the choice of $\tau \supseteq \tau'$ does not influence the value of $\tau(y)$.

Claim 2: For $x \in \mathrm{V}_\gamma$, $x' \in \mathcal{V}_\gamma(\sigma'(x))$, the choice of $\tau \supseteq \tau'$ does not influence the value of $\epsilon(e')(\tau)(x')$.

Claim 3: $R \cup S_e$ is wellfounded.

Proof of Claim 1: $y \in \mathcal{V}_\delta(\sigma'(x))$ means $(y, x) \in \Delta_\sigma {\upharpoonright}_{\mathrm{V}_\gamma}$. By definition of S_e we have $(y, x) \in S_e$, i.e. $y \in S_e\langle\!\{x\}\!\rangle = \mathrm{dom}(\tau')$. Q.e.d. (Claim 1)

Proof of Claim 2: $x' \in \mathcal{V}_\gamma(\sigma'(x))$ means $(x', x) \in {}_{\mathrm{V}_\gamma\setminus\mathrm{dom}(\sigma)}{\upharpoonleft}\mathrm{id} \cup \Gamma_\sigma{\upharpoonright}_{\mathrm{V}_\gamma}$. Thus by definition of S_e we have $S_{e'}\{(x', x)\} \subseteq S_e$, i.e. $S_{e'}\langle\!\{x'\}\!\rangle \subseteq S_e\langle\!\{x\}\!\rangle = \mathrm{dom}(\tau')$. Therefore $\epsilon(e')(\tau)(x') = e'(x')({}_{S_{e'}\langle\!\{x'\}\!\rangle}{\upharpoonleft}\tau) = e'(x')({}_{S_{e'}\langle\!\{x'\}\!\rangle}{\upharpoonleft}\tau')$. Q.e.d. (Claim 2)

Proof of Claim 3: $R' \cup S_{e'}$ is wellfounded because e' is an $(\mathcal{A}, R')$-valuation. Moreover, as R' is the σ-update of R, we have[28] $R' = R \cup \Gamma_\sigma \cup \Delta_\sigma$. Thus, $(R \cup \Gamma_\sigma \cup \Delta_\sigma \cup S_{e'})^+$ is a wellfounded ordering, just like its subset $(R \ \cup \ S_{e'}({}_{\mathrm{V}_\gamma\setminus\mathrm{dom}(\sigma)}{\upharpoonleft}\mathrm{id} \cup \Gamma_\sigma{\upharpoonright}_{\mathrm{V}_\gamma}) \ \cup \ \Delta_\sigma{\upharpoonright}_{\mathrm{V}_\gamma})^+$, which is equal to $(R \cup S_e)^+$. Q.e.d. (Claim 3)

Now, for $\tau : \mathrm{V}_\delta \to \mathcal{A}$ and $x \in \mathrm{V}_\gamma$ we have

$$\epsilon(e)(\tau)(x) = e(x)({}_{S_e\langle\!\{x\}\!\rangle}{\upharpoonleft}\tau) = \mathrm{eval}(\mathcal{A} \uplus \epsilon(e')(\tau) \uplus \tau)(\sigma'(x)),$$

i.e.

$$\epsilon(e)(\tau) = \sigma' \circ \mathrm{eval}(\mathcal{A} \uplus \epsilon(e')(\tau) \uplus \tau).$$

(3): Set $\sigma' := {}_{\mathrm{V}_\gamma\setminus\mathrm{dom}(\sigma)}{\upharpoonleft}\mathrm{id} \cup {}_{\mathrm{V}_\gamma}{\upharpoonleft}\sigma$.

Define $S_e := (S_{\pi'} \cup {}_{\mathrm{V}_{\delta-}}{\upharpoonleft}\mathrm{id})(S_{e'}({}_{\mathrm{V}_\gamma\setminus\mathrm{dom}(\sigma)}{\upharpoonleft}\mathrm{id} \cup \Gamma_\sigma{\upharpoonright}_{\mathrm{V}_\gamma}) \cup \Delta_\sigma{\upharpoonright}_{\mathrm{V}_\gamma})$ and the $(\mathcal{A}, R)$-valuation e by $(x \in \mathrm{V}_\gamma, \tau' : S_e\langle\!\{x\}\!\rangle \to \mathcal{A})$:

$$e(x)(\tau') := \mathrm{eval}(\mathcal{A} \uplus \epsilon(e')(\epsilon(\pi')(\tau) \uplus \tau) \uplus \epsilon(\pi')(\tau) \uplus \tau)(\sigma'(x))$$

where $\tau : \mathrm{V}_{\delta-} \to \mathcal{A}$ is an arbitrary extension of τ'.

For this definition to be okay, we have to prove the following claims:

Claim 4: For $x \in \mathrm{V}_\gamma$ and $y \in \mathcal{V}(\sigma'(x))$, the choice of $\tau \supseteq \tau'$ does not influence:

(a) In case of $y \in \mathrm{V}_{\delta-}$, the value of $\tau(y)$.

(b) In case of $y \in \mathrm{V}_{\delta+}$, the value of $\epsilon(\pi')(\tau)(y)$.

(c) In case of $y \in \mathrm{V}_\gamma$, the value of $\epsilon(e')(\epsilon(\pi')(\tau) \uplus \tau)(y)$.

Claim 5: $R \cup S_e \cup (R' \cup S_{e'} \cup S_{\pi'})^+{\upharpoonright}_{\mathrm{V}_\delta}$ is wellfounded.

Proof of Claim 4a: $y \in \mathcal{V}_{\delta-}(\sigma'(x))$ means $(y, x) \in {}_{\mathrm{V}_{\delta-}}{\upharpoonleft}\Delta_\sigma{\upharpoonright}_{\mathrm{V}_\gamma}$. By definition of S_e we have $(y, x) \in S_e$, i.e. $y \in S_e\langle\!\{x\}\!\rangle = \mathrm{dom}(\tau')$. Q.e.d. (Claim 4a)

Proof of Claim 4b: $y \in \mathcal{V}_{\delta+}(\sigma'(x))$ means $(y, x) \in \Delta_\sigma{\upharpoonright}_{\mathrm{V}_\gamma}$. Thus by definition of S_e we have $S_{\pi'}\{(y, x)\} \subseteq S_e$, i.e. $S_{\pi'}\langle\!\{y\}\!\rangle \subseteq S_e\langle\!\{x\}\!\rangle = \mathrm{dom}(\tau')$. Therefore $\epsilon(\pi')(\tau)(y) = \pi'(y)({}_{S_{\pi'}\langle\!\{y\}\!\rangle}{\upharpoonleft}\tau) = \pi'(y)({}_{S_{\pi'}\langle\!\{y\}\!\rangle}{\upharpoonleft}\tau')$. Q.e.d. (Claim 4b)

Proof of Claim 4c: $y \in \mathcal{V}_\gamma(\sigma'(x))$ means $(y, x) \in {}_{\mathrm{V}_\gamma\setminus\mathrm{dom}(\sigma)}{\upharpoonleft}\mathrm{id} \cup \Gamma_\sigma{\upharpoonright}_{\mathrm{V}_\gamma}$. If the value of $\epsilon(e')(\epsilon(\pi')(\tau) \uplus \tau)(y) = e'(y)({}_{S_{e'}\langle\!\{y\}\!\rangle}{\upharpoonleft}(\epsilon(\pi')(\tau) \uplus \tau))$ depended on the choice of $\tau \supseteq \tau'$, then there would be some $z \in S_{e'}\langle\!\{y\}\!\rangle$ with $(S_{\pi'} \cup {}_{\mathrm{V}_{\delta-}}{\upharpoonleft}\mathrm{id})\langle\!\{z\}\!\rangle \not\subseteq \mathrm{dom}(\tau')$, which is contradictory to $(S_{\pi'} \cup {}_{\mathrm{V}_{\delta-}}{\upharpoonleft}\mathrm{id})\langle\!\{z\}\!\rangle \subseteq ((S_{\pi'} \cup {}_{\mathrm{V}_{\delta-}}{\upharpoonleft}\mathrm{id})S_{e'})\langle\!\{y\}\!\rangle \subseteq ((S_{\pi'} \cup {}_{\mathrm{V}_{\delta-}}{\upharpoonleft}\mathrm{id})S_{e'}({}_{\mathrm{V}_\gamma\setminus\mathrm{dom}(\sigma)}{\upharpoonleft}\mathrm{id} \cup \Gamma_\sigma{\upharpoonright}_{\mathrm{V}_\gamma}))\langle\!\{x\}\!\rangle \subseteq S_e\langle\!\{x\}\!\rangle = \mathrm{dom}(\tau')$. Q.e.d. (Claim 4c)

Proof of Claim 5: $R' \cup S_{e'} \cup S_{\pi'}$ is wellfounded because π' is $(e', \mathcal{A})$-compatible with (C', R'). Moreover, as R' is the σ-update of R, we have[29] $R' = R \cup \Gamma_\sigma \cup \Delta_\sigma$. Thus, $R \cup \Gamma_\sigma \cup \Delta_\sigma \cup R' \cup S_{e'} \cup S_{\pi'}$ is wellfounded, just like the subset

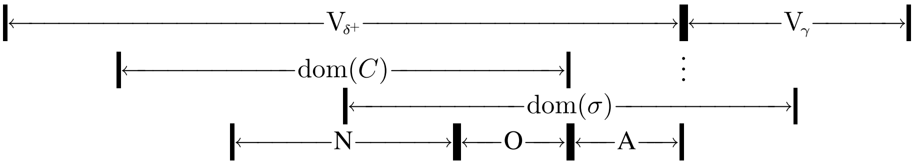
$$R \ \cup \ (S_{\pi'} \cup {}_{\mathrm{V}_{\delta-}}{\upharpoonleft}\mathrm{id})(S_{e'}({}_{\mathrm{V}_\gamma\setminus\mathrm{dom}(\sigma)}{\upharpoonleft}\mathrm{id} \cup \Gamma_\sigma{\upharpoonright}_{\mathrm{V}_\gamma}) \cup \Delta_\sigma{\upharpoonright}_{\mathrm{V}_\gamma}) \ \cup \ {}_{\mathrm{V}_\delta}{\upharpoonleft}(R' \cup S_{e'} \cup S_{\pi'})^+{\upharpoonright}_{\mathrm{V}_\delta}$$

of its transitive closure, which is again equal to $R \cup S_e \cup {}_{\mathrm{V}_\delta}{\upharpoonleft}(R' \cup S_{e'} \cup S_{\pi'})^+{\upharpoonright}_{\mathrm{V}_\delta}$. Q.e.d. (Claim 5)

Q.e.d. (Lemma B.1)

Proof of Lemma B.5

Assuming σ, C, R, C', R', O, N, $\mathcal{A}$, e', π' as described in the lemma, we set $A := (\mathrm{V}_{\delta+} \cap \mathrm{dom}(\sigma)) \setminus (N \uplus O)$. As σ is a substitution on $\mathrm{V}_\gamma \cup \mathrm{V}_{\delta+}$, we have $\mathrm{V}_\delta \cap \mathrm{dom}(\sigma) \subseteq N \uplus O \uplus A \subseteq \mathrm{V}_{\delta+}$. This leaves us in the following situation:



Note that C' is an R'-choice-condition due to Lemma 2.22.
As π' is $(e', \mathcal{A})$-compatible with (C', R'),

$$\lhd := (R' \cup S_{e'} \cup S_{\pi'})^+$$

is a wellfounded ordering.
Let e be the $(\mathcal{A}, R)$-valuation given by Lemma B.1(3) for e'. Then

$$S_e \;=\; (S_{\pi'} \cup {}_{\mathrm{V}_{\delta-}}\rceil\mathrm{id}) \;\circ\; (S_{e'} \circ ({}_{\mathrm{V}_\gamma \setminus \mathrm{dom}(\sigma)}\rceil\mathrm{id} \cup \Gamma_\sigma \upharpoonright_{\mathrm{V}_\gamma}) \cup \Delta_\sigma \upharpoonright_{\mathrm{V}_\gamma}) \tag{B.5.1}$$

and for all $\delta : \mathrm{V}_\delta \to \mathcal{A}$ and $\tau := {}_{\mathrm{V}_{\delta-}}\rceil\delta$:

$$\epsilon(e)(\delta) = ({}_{\mathrm{V}_\gamma \setminus \mathrm{dom}(\sigma)}\rceil\mathrm{id} \cup {}_{\mathrm{V}_\gamma}\rceil\sigma) \circ \mathrm{eval}(\mathcal{A} \uplus \epsilon(e')(\epsilon(\pi')(\tau) \uplus \tau) \uplus \epsilon(\pi')(\tau) \uplus \tau) \tag{B.5.2}$$

and

$$R \;\cup\; S_e \;\cup\; {}_{\mathrm{V}_\delta}\rceil\lhd\upharpoonright_{\mathrm{V}_\delta} \text{ is wellfounded.} \tag{B.5.3}$$

Claim 1: For any term or formula B (possibly with some unbound occurrences of variables from a set $W \subseteq \mathrm{V}_{\mathrm{bound}}$) and any $\tau : \mathrm{V}_{\delta-} \to \mathcal{A}$, $\chi : W \to \mathcal{A}$, and $\delta, \delta', \bar{\delta}' : \mathrm{V}_\delta \to \mathcal{A}$ with ${}_{\mathrm{V}_{\delta-}}\rceil\delta = \tau$, ${}_{\mathcal{V}_\delta(\langle\mathcal{V}_\gamma(B)\rangle\sigma)}\rceil\bar{\delta}' = {}_{\mathcal{V}_\delta(\langle\mathcal{V}_\gamma(B)\rangle\sigma)}\rceil\delta'$, $\delta' = \epsilon(\pi')(\tau) \uplus \tau$:

$$\begin{aligned} &\mathrm{eval}(\mathcal{A} \uplus \epsilon(e')(\delta') \uplus \bar{\delta}' \uplus \chi)(B\sigma) \\ = \;&\mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta) \uplus {}_{\mathrm{V}_\delta}\rceil\sigma \circ \mathrm{eval}(\mathcal{A} \uplus \epsilon(e')(\delta') \uplus \bar{\delta}') \uplus {}_{\mathrm{V}\setminus\mathrm{dom}(\sigma)}\rceil\bar{\delta}' \uplus \chi)(B). \end{aligned}$$

Proof of Claim 1: $\mathrm{eval}(\mathcal{A} \uplus \epsilon(e')(\delta') \uplus \bar{\delta}' \uplus \chi)(B\sigma) =$ (by the Substitution-Lemma)
$\mathrm{eval}(\mathcal{A} \uplus ({}_{\mathrm{V}\setminus\mathrm{dom}(\sigma)}\rceil\mathrm{id} \uplus \sigma) \circ \mathrm{eval}(\mathcal{A} \uplus \epsilon(e')(\delta') \uplus \bar{\delta}' \uplus \chi))(B) =$
(by the Explicitness-Lemma: as the variables of W do not occur free in $\mathrm{ran}(\sigma)$ and by ${}_{\mathcal{V}_\delta(\langle\mathcal{V}_\gamma(B)\rangle\sigma)}\rceil\bar{\delta}' = {}_{\mathcal{V}_\delta(\langle\mathcal{V}_\gamma(B)\rangle\sigma)}\rceil\delta'$)

$$\mathrm{eval}\left(\begin{array}{ll} & \mathcal{A} \\ \uplus & ({}_{\mathrm{V}_\gamma\setminus\mathrm{dom}(\sigma)}\rceil\mathrm{id} \uplus {}_{\mathrm{V}_\gamma}\rceil\sigma) \circ \mathrm{eval}(\mathcal{A} \uplus \epsilon(e')(\delta') \uplus \delta') \\ \uplus & {}_{\mathrm{V}_\delta}\rceil\sigma \circ \mathrm{eval}(\mathcal{A} \uplus \epsilon(e')(\delta') \uplus \bar{\delta}') \\ \uplus & {}_{\mathrm{V}\setminus\mathrm{dom}(\sigma)}\rceil\bar{\delta}' \uplus \chi \end{array}\right)(\;B\;) =$$

(by (B.5.2), ${}_{\mathrm{V}_{\delta-}}\rceil\delta = \tau$, and $\delta' = \epsilon(\pi')(\tau) \uplus \tau$)
$\mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta) \uplus {}_{\mathrm{V}_\delta}\rceil\sigma \circ \mathrm{eval}(\mathcal{A} \uplus \epsilon(e')(\delta') \uplus \bar{\delta}') \uplus {}_{\mathrm{V}\setminus\mathrm{dom}(\sigma)}\rceil\bar{\delta}' \uplus \chi)(B)$. Q.e.d. (Claim 1)

Claim 2: ${}_{\mathrm{V}_\delta}\rceil(R^+) \subseteq \lhd$, $\Delta_\sigma \circ R^+ \subseteq \lhd$, and $S_e \circ R^+ \subseteq \lhd$.

Proof of Claim 2: As R' is the σ-update of R, we have[30] $R' = R \cup \Gamma_\sigma \cup \Delta_\sigma$. Thus, the first two statements of Claim 2 are trivial by definition of $\lhd$ and the third follows from (B.5.1). Q.e.d. (Claim 2)

Set $S_\pi := \lhd \cap (\mathrm{V}_{\delta-} \times \mathrm{V}_{\delta+})$.

Claim 3: $R \cup S_e \cup S_\pi$ is wellfounded.

Proof of Claim 3: This follows from (B.5.3). Q.e.d. (Claim 3)

The idea for the definition of the π we have to find is—roughly speaking—as follows: For $y^{\delta+} \notin N \uplus O \uplus A$ we take $\pi(y^{\delta+})$ to be $\pi'(y^{\delta+})$. For $y^{\delta+} \in O$ we evaluate $\sigma(y^{\delta+})$ in $(\pi', e', \mathcal{A})$ because we know that $(\langle O\rangle Q_C)\sigma$ is valid there by assumption of the lemma. For $y^{\delta+} \in A$ we take the same because this case is unproblematic. For $y^{\delta+} \in N$, however, we have to take care of $(e, \mathcal{A})$-compatibility with (C, R) explicitly in an $\lhd$-recursive definition.
Let π be defined by $(y^{\delta+} \in \mathrm{V}_{\delta+},\ \tau : \mathrm{V}_{\delta-} \rightarrow \mathcal{A})$
$\pi(y^{\delta+})({}_{S_\pi\langle\{y^{\delta+}\}\rangle}\upharpoonleft\tau) :=$

$$\begin{cases} f & \text{if } y^{\delta+} \in N \\ \mathrm{eval}(\mathcal{A} \uplus \epsilon(e')(\epsilon(\pi')(\tau) \uplus \tau) \uplus \epsilon(\pi')(\tau) \uplus \tau)(\sigma(y^{\delta+})) & \text{if } y^{\delta+} \in O \uplus A \\ \pi'(y^{\delta+})({}_{S_{\pi'}\langle\{y^{\delta+}\}\rangle}\upharpoonleft\tau) & \text{otherwise} \end{cases}$$

where (for details cf. the proof of Lemma 2.24) f is chosen s.t., for
$C(y^{\delta+}) = \lambda v_0.\ \ldots \lambda v_{l-1}.\ B$ for a formula B, and for any $\chi : \{v_0, \ldots, v_{l-1}\} \rightarrow \mathcal{A}$
B becomes—if possible— $({}_{\mathrm{V}_{\delta+}\setminus\{y^{\delta+}\}}\upharpoonleft(\epsilon(\pi)(\tau)) \uplus \{y^{\delta+} \mapsto f\} \uplus \tau \uplus \chi, e, \mathcal{A})$-valid.
Note that this definition is okay because the only part of τ that is relevant on the right-hand side is ${}_{S_\pi\langle\{y^{\delta+}\}\rangle}\upharpoonleft\tau$ (we have $(\Gamma_\sigma \cup \Delta_\sigma)\upharpoonright_{\mathrm{V}_\delta} \subseteq R'$ due to R' being the σ-update of R) and because it is recursive in $\lhd$; indeed, for $x^\gamma \in \mathcal{V}_\gamma(C(y^{\delta+}))$ we have $x^\gamma\ R^+\ y^{\delta+}$ (as C is an R-choice-condition) and then for $v^\delta\ S_e\ x^\gamma$ we have $v^\delta \lhd y^{\delta+}$ by Claim 2, and for $z^\delta \in \mathcal{V}_\delta(C(y^{\delta+}))\setminus\{y^{\delta+}\}$ we have $z^\delta\ R^+\ y^{\delta+}$ and then $z^\delta \lhd y^{\delta+}$ by Claim 2.
Claim 4: For all $y^{\delta+} \in O \uplus A$ and $\tau : \mathrm{V}_{\delta-} \rightarrow \mathcal{A}$, when we set $\delta' := \epsilon(\pi')(\tau) \uplus \tau$:

$$\epsilon(\pi)(\tau)(y^{\delta+}) = \mathrm{eval}(\mathcal{A} \uplus \epsilon(e')(\delta') \uplus \delta')(\sigma(y^{\delta+})).$$

Proof of Claim 4: Immediately by the definition of π. Q.e.d. (Claim 4)
Claim 5: For all $y^{\delta+} \in \mathrm{V}_{\delta+}\setminus(N \uplus O \uplus A)$ and $\tau : \mathrm{V}_{\delta-} \rightarrow \mathcal{A}$: $\epsilon(\pi)(\tau)(y^{\delta+}) = \epsilon(\pi')(\tau)(y^{\delta+})$.
Proof of Claim 5: Immediately by the definition of π. Q.e.d. (Claim 5)
Claim 6: For any term or formula B (possibly with some unbound occurrences of variables from a set $W \subseteq \mathrm{V}_{\mathrm{bound}}$) with $N \cap \mathcal{V}(B) = \emptyset$, and for any $\tau : \mathrm{V}_{\delta-} \rightarrow \mathcal{A}$ and $\chi : W \rightarrow \mathcal{A}$, when we set $\delta := \epsilon(\pi)(\tau) \uplus \tau$ and $\delta' := \epsilon(\pi')(\tau) \uplus \tau$, we have

$$\mathrm{eval}(\mathcal{A} \uplus \epsilon(e')(\delta') \uplus \delta' \uplus \chi)(B\sigma) = \mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta) \uplus \delta \uplus \chi)(B).$$

Proof of Claim 6: $\mathrm{eval}(\mathcal{A} \uplus \epsilon(e')(\delta') \uplus \delta' \uplus \chi)(B\sigma) =$ (by Claim 1)
$\mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta) \uplus {}_{\mathrm{V}_\delta}\upharpoonleft\sigma \circ \mathrm{eval}(\mathcal{A} \uplus \epsilon(e')(\delta') \uplus \delta') \quad \uplus {}_{\mathrm{V}\setminus\mathrm{dom}(\sigma)}\upharpoonleft\delta' \uplus \chi)(B) =$
(by $O \uplus A \subseteq \mathrm{V}_\delta \cap \mathrm{dom}(\sigma) \subseteq N \uplus O \uplus A$ and $N \cap \mathcal{V}(B) = \emptyset$)
$\mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta) \uplus {}_{O \uplus A}\upharpoonleft\sigma \circ \mathrm{eval}(\mathcal{A} \uplus \epsilon(e')(\delta') \uplus \delta') \uplus {}_{\mathrm{V}\setminus(N \uplus O \uplus A)}\upharpoonleft\delta' \ \uplus \chi)(B) =$
(by Claim 4 and Claim 5)
$\mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta) \uplus \delta \uplus \chi)(B)$. Q.e.d. (Claim 6)
Claim 7: For any set of sequents G' (possibly with some unbound occurrences of variables from a set $W \subseteq \mathrm{V}_{\mathrm{bound}}$) with $N \cap \mathcal{V}(G') = \emptyset$, and for any $\tau : (\mathrm{V}_{\delta-} \cup W) \rightarrow \mathcal{A}$:
$(\epsilon(\pi)(\tau) \uplus \tau, e, \mathcal{A})$-validity of G' is logically equivalent to $(\epsilon(\pi')(\tau) \uplus \tau, e', \mathcal{A})$-validity of $G'\sigma$.
Proof of Claim 7: This is a trivial consequence of Claim 6. Q.e.d. (Claim 7)
Claim 8: For $y^{\delta+} \in \mathrm{dom}(C) \setminus N$ we have $N \cap \mathcal{V}(C(y^{\delta+})) = \emptyset$.
Proof of Claim 8: Otherwise there is some $z^{\delta+} \in N \cap \mathcal{V}(C(y^{\delta+}))$, but then $z^{\delta+}\ R^*\ y^{\delta+}$ as C is an R-choice-condition, and then, as $\mathrm{dom}(C) \cap \langle N\rangle R^+ \subseteq N$, we have the contradicting $y^{\delta+} \in N$.
Q.e.d. (Claim 8)

Claim 9: Let $y^{\delta+} \in \mathrm{dom}(C)$ and $C(y^{\delta+}) = \lambda v_0.\ \ldots \lambda v_{l-1}.\ B$. Let $\tau : \mathrm{V}_{\delta-} \rightarrow \mathcal{A}$ and $\chi : \{v_0, \ldots, v_{l-1}\} \rightarrow \mathcal{A}$ and suppose that, for some $\eta : \{y^{\delta+}\} \rightarrow \mathcal{A}$, B is $(\bar{\delta}, e, \mathcal{A})$-valid for $\bar{\delta} := {}_{\mathrm{V}_{\delta+}\backslash\{y^{\delta+}\}}{\upharpoonright}(\epsilon(\pi)(\tau)) \uplus \eta \uplus \tau \uplus \chi$. Now: B is $(\delta, e, \mathcal{A})$-valid for $\delta := \epsilon(\pi)(\tau) \uplus \tau \uplus \chi$.

Proof of Claim 9: Set $\bar{\delta}' := {}_{\mathrm{V}_{\delta+}\backslash\{y^{\delta+}\}}{\upharpoonright}(\epsilon(\pi')(\tau)) \uplus \eta \uplus \tau \uplus \chi$ and $\delta' := \epsilon(\pi')(\tau) \uplus \tau \uplus \chi$.
$y^{\delta+} \notin O{\uplus}N$: In this case, we have $y^{\delta+} \notin \mathrm{dom}(\sigma)$ because of $\mathrm{dom}(C) \cap \mathrm{dom}(\sigma) \subseteq O{\uplus}N$. Thus, as (C', R') is the extended σ-update of (C, R), we have $C'(y^{\delta+}) = (C(y^{\delta+}))\sigma$. By Claim 8 we have $N{\cap}\mathcal{V}(B) = \emptyset$. For later application of Claim 1, note that ${}_{\mathcal{V}_\delta(\langle\mathcal{V}_\gamma(B)\rangle\sigma)}{\upharpoonright}\bar{\delta}' = {}_{\mathcal{V}_\delta(\langle\mathcal{V}_\gamma(B)\rangle\sigma)}{\upharpoonright}\delta'$; otherwise there would be some $x^\gamma \in \mathcal{V}_\gamma(B) = \mathcal{V}_\gamma(C(y^{\delta+}))$ with $y^{\delta+}\ \Delta_\sigma\ x^\gamma$, and then, as C is an R-choice-condition, $y^{\delta+}\ \Delta_\sigma\ x^\gamma\ R^+\ y^{\delta+}$, and then, by Claim 2, $y^{\delta+} \lhd y^{\delta+}$, which contradicts the wellfoundedness of $\lhd$.
Note that ${}_{\mathcal{V}_{\delta+}(B)}{\upharpoonright}\sigma \circ \mathrm{eval}(\mathcal{A} \uplus \epsilon(e')(\delta') \uplus \delta') = {}_{\mathcal{V}_{\delta+}(B)}{\upharpoonright}\sigma \circ \mathrm{eval}(\mathcal{A} \uplus \epsilon(e')(\delta') \uplus \bar{\delta}')$; otherwise there would be some $z^{\delta+} \in \mathcal{V}_{\delta+}(C(y^{\delta+}))$ with $y^{\delta+} \in \mathcal{V}(\sigma(z^{\delta+}))$, which implies $y^{\delta+}\ R'\ z^{\delta+}\ R^*\ y^{\delta+}$ (as R' is the σ-update of R and C is an R-choice-condition), and then, by Claim 2, $y^{\delta+} \lhd z^{\delta+} \unlhd y^{\delta+}$, which contradicts the wellfoundedness of $\lhd$. Moreover:
${}_{\mathcal{V}(B)}{\upharpoonright}\bar{\delta} =$ (due to $y^{\delta+} \notin \mathrm{dom}(\sigma)$, $N \cup (\mathrm{dom}(\sigma) \cap \mathrm{V}_\delta) = N{\uplus}O{\uplus}A$, $N{\cap}\mathcal{V}(B) = \emptyset$, Claim 5)
${}_{\mathcal{V}(B)\backslash\mathrm{dom}(\sigma)}{\upharpoonright}\bar{\delta}' \uplus {}_{(O\uplus A)\cap\mathcal{V}_{\delta+}(B)}{\upharpoonright}(\epsilon(\pi)(\tau)) =$ (by Claim 4)
${}_{\mathcal{V}(B)\backslash\mathrm{dom}(\sigma)}{\upharpoonright}\bar{\delta}' \uplus {}_{(O\uplus A)\cap\mathcal{V}_{\delta+}(B)}{\upharpoonright}\sigma \circ \mathrm{eval}(\mathcal{A} \uplus \epsilon(e')(\delta') \uplus \delta') =$ (cf. above)
${}_{\mathcal{V}(B)\backslash\mathrm{dom}(\sigma)}{\upharpoonright}\bar{\delta}' \uplus {}_{(O\uplus A)\cap\mathcal{V}_{\delta+}(B)}{\upharpoonright}\sigma \circ \mathrm{eval}(\mathcal{A} \uplus \epsilon(e')(\delta') \uplus \bar{\delta}')$.
Now: $\mathsf{TRUE} =$ (by assumption of Claim 9)
$\mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\bar{\delta}) \uplus \bar{\delta})(B) =$ (by the above and $\mathrm{dom}(\sigma) \cap \mathcal{V}_\delta(B) = (O{\uplus}A) \cap \mathcal{V}_{\delta+}(B)$)
$\mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\bar{\delta}) \uplus {}_{\mathrm{V}_\delta}{\upharpoonright}\sigma \circ \mathrm{eval}(\mathcal{A} \uplus \epsilon(e')(\delta') \uplus \bar{\delta}') \uplus {}_{\mathrm{V}\backslash\mathrm{dom}(\sigma)}{\upharpoonright}\bar{\delta}')(B) =$
(by Claim 1 instantiated with the substitution $\{\delta{\mapsto}\bar{\delta}\}$)
$\mathrm{eval}(\mathcal{A} \uplus \epsilon(e')(\delta') \uplus \bar{\delta}')(B\sigma) =$ (as otherwise for some $x^\gamma \in \mathcal{V}_\gamma(B\sigma) = \mathcal{V}_\gamma(C'(y^{\delta+}))$
we have $y^{\delta+}\ S_{e'}\ x^\gamma\ R'^{+}\ y^{\delta+}$, i.e. $y^{\delta+} {\lhd} y^{\delta+}$)
$\mathrm{eval}(\mathcal{A} \uplus \epsilon(e')(\bar{\delta}') \uplus \bar{\delta}')(B\sigma)$. As π' is $(e', \mathcal{A})$-compatible with (C', R'), we know that $B\sigma$ is $(\delta', e', \mathcal{A})$-valid. Thus, by Claim 7, B is $(\delta, e, \mathcal{A})$-valid.
$y^{\delta+} \in O$: $N{\cap}\mathcal{V}(B) = \emptyset$ by Claim 8. Let $y \in \mathrm{V}_{\mathrm{bound}}\backslash\mathcal{V}(C(y^{\delta+}))$ and D be the formula

$$\exists y.\ (B\{y^{\delta+}(v_0)\cdots(v_{l-1}) \mapsto y\})$$

s.t. $Q_C(y^{\delta+})$ is equal to $\forall v_0.\ \ldots\forall v_{l-1}.\ (D \Rightarrow B)$. We have $N{\cap}\mathcal{V}(D) = \emptyset$. As B is valid in $\mathcal{A}{\uplus}\epsilon(e)(\bar{\delta}){\uplus}\bar{\delta}$, for $w^\delta \in \mathrm{V}_\delta\backslash\mathcal{V}(B)$ and $\bar{w} := \mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\bar{\delta}) \uplus \bar{\delta})(y^{\delta+}(v_0)\cdots(v_{l-1}))$ we have

$$\begin{aligned}\mathsf{TRUE} &= \mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\bar{\delta}) \uplus \bar{\delta})(B\{y^{\delta+}(v_0)\cdots(v_{l-1}) \mapsto w^\delta\}\{w^\delta \mapsto y^{\delta+}(v_0)\cdots(v_{l-1})\})\\ &= \mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\bar{\delta}) \uplus {}_{\mathrm{V}\backslash\{w^\delta\}}{\upharpoonright}\bar{\delta} \uplus \{w^\delta{\mapsto}\bar{w}\})(B\{y^{\delta+}(v_0)\cdots(v_{l-1}) \mapsto w^\delta\})\end{aligned}$$

by the Substitution-Lemma. Thus, by the standard semantical definition of $\exists$ (cf. e.g. Enderton (1973), p. 82), D is valid in $\mathcal{A} \uplus \epsilon(e)(\bar{\delta}) \uplus \bar{\delta}$, too; and then (as $y^{\delta+}$ does not occur in D anymore (as all occurrences of $y^{\delta+}$ in B are of the form $y^{\delta+}(v_0)\cdots(v_{l-1})$ according to Definition B.2)) also valid in $\mathcal{A} \uplus \epsilon(e)(\bar{\delta}) \uplus \delta$. Moreover, D is even valid in $\mathcal{A} \uplus \epsilon(e)(\delta) \uplus \delta$; otherwise there would be some $v^\gamma \in \mathcal{V}_\gamma(D)$ with $y^{\delta+}\ S_e\ v^\gamma$, but then $v^\gamma \in \mathcal{V}(C(y^{\delta+}))\backslash\{y^{\delta+}\}$ and (as C is an R-choice-condition) $v^\gamma\ R^+\ y^{\delta+}$, which contradicts the wellfoundedness of $R{\cup}S_e$, which contradicts e being an $(\mathcal{A}, R)$-valuation. By Claim 7, $D\sigma$ is $(\delta', e', \mathcal{A})$-valid. But by assumption of the lemma on $(\langle O\rangle Q_C)\sigma$ and by the standard definition of $\forall$, we know that $(D \Rightarrow B)\sigma$ is $(\delta', e', \mathcal{A})$-valid. Thus, $B\sigma$ is $(\delta', e', \mathcal{A})$-valid. By Claim 7, B is $(\delta, e, \mathcal{A})$-valid.
$y^{\delta+} \in N$: By definition of π. Q.e.d. (Claim 9)

By Claim 3 and Claim 9, π is $(e, \mathcal{A})$-compatible with (C, R), and then items 1 and 2 of the lemma are trivial consequences of Claim 6, Claim 7, resp.. **Q.e.d. (Lemma B.5)**

Proof of Lemma B.6

<u>(1):</u> As $G_0\sigma \cup (\langle O\rangle Q_C)\sigma$ is (C', R')-valid in $\mathcal{A}$, there is an $(\mathcal{A}, R')$-valuation e' and some π' s.t. π' is $(e', \mathcal{A})$-compatible with (C', R') and $G_0\sigma \cup (\langle O\rangle Q_C)\sigma$ is $(\pi', e', \mathcal{A})$-valid. Let e and π be given as in Lemma B.5. Then G_0 is $(\pi, e, \mathcal{A})$-valid. Moreover, as π is $(e, \mathcal{A})$-compatible with (R, C) and as e is an $(\mathcal{A}, R)$-valuation, G_0 is (C, R)-valid in $\mathcal{A}$.

<u>(2):</u> Let e' be an $(\mathcal{A}, R')$-valuation, π' be $(e', \mathcal{A})$-compatible with (C', R'), and suppose that $G_1\sigma \cup (\langle O\rangle Q_C)\sigma$ is $(\pi', e', \mathcal{A})$-valid. Let π and the $(\mathcal{A}, R)$-valuation e be given as in Lemma B.5. Then π is $(e, \mathcal{A})$-compatible with (C, R), and G_1 is $(\pi, e, \mathcal{A})$-valid. By assumption, G_0 (C, R)-reduces to G_1. Thus, G_0 is $(\pi, e, \mathcal{A})$-valid, too. By Lemma B.5(2), this means that $G_0\sigma$ is $(\pi', e', \mathcal{A})$-valid as was to be shown. **Q.e.d. (Lemma B.6)**

Proof of Lemma B.7

Let $\mathcal{A} \in \mathrm{K}$, let e' be an $(\mathcal{A}, R')$-valuation, and π' be $(e', \mathcal{A})$-compatible with (C', R'). Let $(\mathit{\Gamma}, (w, <, \lesssim)) \in G_0$ and assume that $((\mathit{\Gamma}\sigma, (w\sigma, <\sigma, \lesssim\sigma)), \tau)$ is an $(\pi', e', \mathcal{A})$-counterexample. Assuming that there is no $(\pi', e', \mathcal{A})$-counterexample of $L_1\sigma \cup L_2$, we have to find some $(\pi', e', \mathcal{A})$-counterexample $((\mathit{\Gamma}'\sigma, (w'\sigma, <'\sigma, \lesssim'\sigma)), \tau')$ with $(\mathit{\Gamma}', (w', <', \lesssim')) \in G_1$, s.t. $((\mathit{\Gamma}'\sigma, (w'\sigma, <'\sigma, \lesssim'\sigma)), \tau')$ is $(\pi', e', \mathcal{A})$-smaller than $((\mathit{\Gamma}\sigma, (w\sigma, <\sigma, \lesssim\sigma)), \tau)$. By our assumption on no $(\pi', e', \mathcal{A})$-counterexamples of L_2, we can apply Lemma B.5 to get a an $(\mathcal{A}, R)$-valuation e and a π that is $(e, \mathcal{A})$-compatible with (C, R). Moreover, by this lemma, $((\mathit{\Gamma}, (w, <, \lesssim)), \tau)$ is an $(\pi, e, \mathcal{A})$-counterexample. By assumption, $G_0 \rightharpoondown_{C,R} (G_1, L_1)$. Thus, there is some $(\pi, e, \mathcal{A})$-counterexample $((\mathit{\Gamma}', (w', <', \lesssim')), \tau')$ with $(\mathit{\Gamma}', (w', <', \lesssim')) \in L_1$ or both $(\mathit{\Gamma}', (w', <', \lesssim')) \in G_1$ and $((\mathit{\Gamma}', (w', <', \lesssim')), \tau')$ is $(\pi, e, \mathcal{A})$-smaller than $((\mathit{\Gamma}, (w, <, \lesssim)), \tau)$. By Lemma B.5(2), $((\mathit{\Gamma}'\sigma, (w'\sigma, <'\sigma, \lesssim'\sigma)), \tau')$ is an $(\pi', e', \mathcal{A})$-counterexample, and by our assumption on $L_1\sigma$, we then have $(\mathit{\Gamma}', (w', <', \lesssim')) \in G_1$ and $((\mathit{\Gamma}', (w', <', \lesssim')), \tau')$ is $(\pi, e, \mathcal{A})$-smaller than $((\mathit{\Gamma}, (w, <, \lesssim)), \tau)$. We only have left to show that $((\mathit{\Gamma}'\sigma, (w'\sigma, <'\sigma, \lesssim'\sigma)), \tau')$ is $(\pi', e', \mathcal{A})$-smaller than $((\mathit{\Gamma}\sigma, (w\sigma, <\sigma, \lesssim\sigma)), \tau)$. This is nearly implied by Lemma B.5(1); the only problem is that $<$, $\lesssim$, $<'$, $\lesssim'$ are possibly no terms (so that the B of Lemma B.5(1) cannot be instantiated with them). Thus, for arbitrary $\tau : \mathrm{V}_{\delta^-} \rightarrow \mathcal{A}$ and δ and δ' given as in Lemma B.5(1), we still have to prove say $\mathrm{eval}(\mathcal{A} \uplus \epsilon(e')(\delta') \uplus \delta')(<\sigma) = \mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta) \uplus \delta)(<)$. After expanding the shorthand on both sides for some distinct $x, y \in \mathrm{V}_{\mathrm{bound}} \backslash \mathcal{V}(<, \mathrm{dom}(\sigma), \mathrm{ran}(\sigma))$, this follows from
$\mathrm{eval}(\mathcal{A} \uplus \epsilon(e')(\delta') \uplus \delta' \uplus \{x{\mapsto}a,\ y{\mapsto}b\})(x\ (<\sigma)\ y) =$ (as $x, y \notin \mathrm{dom}(\sigma)$)
$\mathrm{eval}(\mathcal{A} \uplus \epsilon(e')(\delta') \uplus \delta' \uplus \{x{\mapsto}a,\ y{\mapsto}b\})((x < y)\sigma) =$ (due to Lemma B.5.1)
$\mathrm{eval}(\mathcal{A} \uplus \epsilon(e)(\delta) \uplus \delta \uplus \{x{\mapsto}a,\ y{\mapsto}b\})(x < y)$. **Q.e.d. (Lemma B.7)**

D Notes

Note 1: This step is actually superfluous because we can simply take the class of all counterexamples for a contradiction. But at this early stage, we want to be independent of the two alternative notions of wellfoundedness as discussed in Section 2.1.2.

Note 2: For *inductive* theorem proving, however, Sergey Yu. Maslov's inversion technique (cf. Maslov (1971)) (note that this is more general than Maslov's *inverse method*, cf. Lifschitz (1989)) and non-refutational resolution (cf. Lee (1967); Leitsch (1997), Theorem of Lee, p. 203) could be organized in a goal-directed manner by starting with the axioms *plus the induction hypotheses*, and a formula that subsumes the induction conclusion is to be inferred. However, this form of goal-directedness is still insufficient: As a myriad of lemmas are applicable, it is practically impossible to find the appropriate ones unless the conclusion has been considerably expanded. Furthermore, since inductive proofs typically follow the form of the recursive definitions, non-refutational resolution requires to paramodulate with the defining rules from right to left, resulting in a high branching rate. All in all, we conclude that non-refutational resolution as well as Maslov's inversion technique are not adequate for our purpose.

Note 3: In EXPANDER, cf. Padawitz (1996), Padawitz (1998), the induction hypotheses are super-clauses (i.e. disjunctions of super-literals, which are conjunctions of literals) with additional existentially quantified variables. They generate inference rules operating on clauses, similar to the super-clauses in *Sergey Yu. Maslov's inverse method*, cf. Lifschitz (1989). Moreover, goal-directedness w.r.t. the induction conclusion is achieved in EXPANDER by starting from the negated induction conclusion in the form of a set of "goals", i.e. clauses in dual notation for readability. Contrary to this, the inverse method starts from the set of tautologies, which has the advantage of deductive completeness but lacks goal-directedness w.r.t. the induction conclusion. Nevertheless, from my experiences with EXPANDER, it does not seem to satisfy our main design goals (I) and (II) of Section 1.2.1 particularly well.

Note 4: Note that for soundness and safeness of the δ-rule it is sufficient that

$$x^{\delta^-} \notin \mathcal{V}(A, \Gamma\Pi, \sqsupset) \cup \mathrm{dom}(R),$$

cf. the proof of Theorem 2.49. Nevertheless, we require the stronger condition

$$x^{\delta^-} \notin \mathcal{V}(\mathcal{F}) \text{ for } \mathcal{F} = (F, C, R, L, H),$$

because we do not want to lose possible proofs.

Note 5: Note that for soundness and safeness of the liberalized δ-rule it is sufficient that

$$x^{\delta^+} \notin \mathcal{V}(A) \cup \mathrm{dom}(C \cup R),$$

cf. the proof of Theorem 2.49. Nevertheless, we require the stronger condition

$$x^{\delta^+} \notin \mathcal{V}(\mathcal{F}) \text{ for } \mathcal{F} = (F, C, R, L, H),$$

because we do not want to lose possible proofs.

Note 6: An anonymous referee of a previous version of this text wrote:

> "A minor item: After stating the relevant induction principle the author writes: 'Now by the Principle of Dependent Choice (cf. Rubin & Rubin (1985)) ....' I find this reference quite inappropriate: Of course, one needs some form of the Axiom of Choice to prove the existence of minimal elements *in general*, however in the context of inductive reasoning the used ordering is always *concretely given* and consequently the fact that 'a class without minimal elements contains a chain without a least element' is always obvious in any particular scenario of theorem proving".

The problem, however, is that there may be several counterexamples and the induction ordering only partial. So we have to pick again and again smaller counterexamples from unstructured non-empty classes. Nevertheless, because of this remark we finally changed the definition of wellfoundedness from non-termination of the reverse relation to the existence of minimal elements, which resulted in an immediate soundness of the Method of Descente Infinie without the Principle of Descente Infinie.

Note 7: The typical problems of higher-order logic—incompleteness, undecidability of unifiability, and Skolemization—do not burden this paper: We neither Skolemize nor show completeness. Moreover, unification is not treated in this paper, we just assume the right instance.

Note 8: It may be objected that in the modal logics of, say, Fitting (1999), Cerrito & Cialdea (2001), Fitting (2002), the Substitution-Lemma is not valid because it only holds for the substitution of rigid and rigidified (grounded, annotated, non-relativized) terms. This is, however, a wrong view: Those substitutions for which the Substitution-Lemma does not hold are no proper substitutions. They cannot occur in proof steps because such proof steps would be unsound. And therefore we do not need them at all, and simply do not call them substitutions, which renders the Substitution-Lemma valid again. Indeed, the substitutions for which the Substitution-Lemma does not hold when applied to a certain term or formula B, are not "free" for B in some sense. The problem is that an implicit variable is captured by some quantifier. We explain this for the higher-order modal logic of Fitting (2002) because there the relativization operator $\downarrow$ makes this obvious. For a term t of intensional type $\uparrow\alpha$, the term $\downarrow t$ has the extensional type α. Instead of $\downarrow t$ one could also write tw where w is a variable valuated to the current world, so that tw is the extension of t at world w. The quantifiers $\Box$, $\Diamond$ and the binder λ implicitly bind this implicit variable w. Let us now have a look on the standard example for the violation of the Substitution-Lemma. Let x, y be variables of the extensional type 0. Let h, p be constants of the intensional type $\uparrow 0$ standing for the intentional notions of Hesperus (morning star) and Phosphorus (evening star), and assume that $\Box$ means "all former highly developed civilizations knew" or simply "the ancients knew". Then

$$x=y \quad\Rightarrow\quad \Box(x=y)$$

is valid because the ancients knew that two identical things are identical. On the other hand its instance

$$\downarrow h=\downarrow p \quad\Rightarrow\quad \Box(\downarrow h=\downarrow p)$$

via the "substitution" $\{x\mapsto\downarrow h,\quad y\mapsto\downarrow p\}$ is not valid in our world because here the extensions of Hesperus and Phosphorus are identical but the ancients did not know that. But with the variable w made explicit, the first formula reads

$$x=y \quad\Rightarrow\quad \Box w.\ (x=y)$$

for which the "substitution" $\{x\mapsto hw,\quad y\mapsto pw\}$ is obviously not "free" because the w is captured by the quantifier $\Box w$.

Note 9: Consider the valid Henkin quantified IF logic formula

$$\forall x_0.\ \forall x_1.\ \exists y_0/x_1.\ \exists y_1/x_0.\ (\ x_0{=}y_0 \wedge x_1{=}y_1\ )$$

or its logically equivalent raised form

$$\exists y_0.\ \exists y_1.\ \forall x_0.\ \forall x_1.\ (\ x_0{=}y_0(x_0) \wedge x_1{=}y_1(x_1)\ )$$

Its representation in our framework as the formula $x_0^\delta{=}y_0^\gamma \wedge x_1^\delta{=}y_1^\gamma$ with variable-condition $R=\{(y_0^\gamma,x_1^\delta),(y_1^\gamma,x_0^\delta)\}$ fails to be R-valid. Indeed, while $\{y_0^\gamma{\mapsto}x_0^\delta\}$ and $\{y_1^\gamma{\mapsto}x_1^\delta\}$ are R-substitutions on V_γ, their combination $\sigma=\{y_0^\gamma{\mapsto}x_0^\delta,\ y_1^\gamma{\mapsto}x_1^\delta\}$ is no R-substitution:

$$\begin{tikzcd}
y_0^\gamma \arrow[rd, "R"'] & x_0^\delta \arrow[l, "\Delta_\sigma"'] \\
y_1^\gamma \arrow[ru, "R"'] & x_1^\delta \arrow[l, "\Delta_\sigma"']
\end{tikzcd}$$

Now, if you want to turn this wrong representation into a proper one, you have to use the notions from the weak version of Wirth (1998) instead. Reformulated according to the slightly different notion of a substitution used in this paper, they read:

DEFINITION NOTE 9.1 (Weak Variable-Condition) *(Cf. Definition 2.7)*
A *variable-condition* is a subset of $\mathrm{V}_\gamma\times\mathrm{V}_\delta$.

DEFINITION NOTE 9.2 (Weak R-Substitution) *(Cf. Definition 2.11)*
Let R be a variable-condition.
σ is an *R-substitution* if σ is a substitution and $\Delta_\sigma\circ R$ is irreflexive.

DEFINITION NOTE 9.3 (Weak σ-Update) *(Cf. Definition 2.12)*
Let R be a variable-condition and σ be a substitution.
The *σ-update of R* is $({}_{\mathrm{V}_\gamma\setminus\mathrm{dom}(\sigma)}{\upharpoonright}\mathrm{id}\cup\Gamma_\sigma)\circ R$.

Note that for this weak version we have to pay the price that we cannot use a liberalized version of the δ-rule, which makes our proofs dependent on the order in which we eliminated quantifiers, thereby violating our design goal of a natural flow of information, cf. Section 1.2.1.

Note 10: If you nevertheless want to have re-use and permutations of free γ-variables you have to use the following alternative notions instead.

DEFINITION NOTE 10.1 (Alternative Variable-Condition) *(Cf. Definition 2.7)*
A *variable-condition* is a subset of $\mathrm{V}_{\mathrm{free}} \times \mathrm{V}_\delta$.

DEFINITION NOTE 10.2 (Alternative R-Substitution) *(Cf. Definition 2.11)*
Let R be a variable-condition. σ is an *R-substitution* if
σ is a substitution and $\left({}_{\mathrm{V}_\delta\cup(\mathrm{V}_\gamma\setminus\mathrm{dom}(\sigma))}\upharpoonleft\mathrm{id} \cup \Gamma_\sigma \cup \Delta_\sigma\right) \circ R \ \cup\ \left(\Gamma_\sigma \cup \Delta_\sigma\right)\upharpoonright_{\mathrm{V}_\delta}$ is wellfounded.

DEFINITION NOTE 10.3 (Alternative σ-Update) *(Cf. Definition 2.12)*
Let R be a variable-condition and σ be a substitution.
The *σ-update of R* is $\left({}_{\mathrm{V}_\delta\cup(\mathrm{V}_\gamma\setminus\mathrm{dom}(\sigma))}\upharpoonleft\mathrm{id} \cup \Gamma_\sigma \cup \Delta_\sigma\right) \circ R \ \cup\ \left(\Gamma_\sigma \cup \Delta_\sigma\right)\upharpoonright_{\mathrm{V}_\delta}$.

In an implementation, substituted free γ-variables should get new nodes while their old nodes lose their labels. E.g., (where we have boxed the old occurrences of the re-used free γ-variables x^γ and u^γ) for

$$R := \{(\boxed{x^\gamma}, y^\delta),\ (\boxed{x^\gamma}, z_0^\delta),\ (\boxed{x^\gamma}, z_1^\delta),\ (\boxed{x^\gamma}, z_2^\delta),\ (\boxed{u^\gamma}, v^\delta),\ (w^\gamma, v^\delta)\}.$$

and the R-substitution on V_γ (in the alternative sense!)

$$\sigma := \{\boxed{x^\gamma} \mapsto (u^\gamma + v^\delta),\ \ \boxed{u^\gamma} \mapsto x^\gamma,\ \ \boxed{y^\gamma} \mapsto v^\delta\}$$

we should update

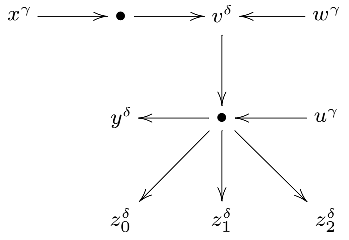
$$\begin{tikzcd}
\boxed{u^\gamma} \arrow[r, "R"'] & v^\delta & w^\gamma \arrow[l, "R"']
\end{tikzcd}$$

$$\begin{tikzcd}
y^\delta & \boxed{x^\gamma} \arrow[l, "R"'] \arrow[dl, "R"'] \arrow[d, "R"] \arrow[dr, "R"] & \\
z_0^\delta & z_1^\delta & z_2^\delta
\end{tikzcd}$$

first to

$$\begin{tikzcd}
 & & \boxed{y^\gamma} & \\
x^\gamma \arrow[r, "\Gamma_\sigma"'] & \boxed{u^\gamma} \arrow[r, "R"'] & v^\delta \arrow[u, "\Delta_\sigma"'] \arrow[d, "\Delta_\sigma"'] & w^\gamma \arrow[l, "R"'] \\
 & y^\delta & \boxed{x^\gamma} \arrow[l, "R"'] \arrow[dl, "R"'] \arrow[d, "R"] \arrow[dr, "R"] & u^\gamma \arrow[l, "\Gamma_\sigma"'] \\
 & z_0^\delta & z_1^\delta & z_2^\delta
\end{tikzcd}$$

and then to



representing the σ-update of R in the alternative sense. Note that the edge from v^δ to $\boxed{y^\gamma}$ has been completely removed in the last step because y^γ has no out-going R-edge. This may be an efficiency advantage over the non-alternative version, cf. also Note 19 in Section 3.1.

Note 11: A first alternative approach one may try is to admit a slight modification of e to e' such that $e'(x^\gamma)(\delta) = a$. However, such a modification does not conform to our requirement on preservation of solutions. Moreover, this approach fails because it is not possible to preserve reduction under instantiation steps:

E.g., an instantiation step with the R-substitution $\{x^\gamma \mapsto y^{\delta^+}\}$ transforms the reduction of Example 2.19 into the reduction of
$\forall y.\ \neg\mathsf{P}(y),\quad \mathsf{P}(y^{\delta^+})$
to $\neg\mathsf{P}(y^{\delta^+}),\quad \mathsf{P}(y^{\delta^+})$

Taking δ, e, and $\mathcal{A}$ as in Example 2.19, the new lower sequent is still $(e, \mathcal{A})$-valid. There is, however, no modification e' of e such that the new upper sequent is $(\delta, e', \mathcal{A})$-valid.

Another alternative approach is to admit a slight modification of δ instead. E.g., for the reduction step of Example 2.19, one would require the existence of some $\pi : \{y^{\delta^+}\} \to \mathcal{A}$ such that the upper sequent is $(\pi \uplus {}_{\mathrm{V}_\delta\setminus\{y^{\delta^+}\}}\!\upharpoonleft\delta,\ e,\ \mathcal{A})$-valid instead of $(\delta, e, \mathcal{A})$-valid. Choosing $\pi := \{y^{\delta^+}\mapsto a\}$ would solve the problem of Example 2.19 then: Indeed, the upper sequent is $(\pi \uplus {}_{\mathrm{V}_\delta\setminus\{y^{\delta^+}\}}\!\upharpoonleft\delta,\ e,\ \mathcal{A})$-valid because for the e of Example 2.19 we have $e(x^\gamma)(\pi \uplus {}_{\mathrm{V}_\delta\setminus\{y^{\delta^+}\}}\!\upharpoonleft\delta) = (\pi \uplus {}_{\mathrm{V}_\delta\setminus\{y^{\delta^+}\}}\!\upharpoonleft\delta)(y^{\delta^+}) = a$. Moreover, with this approach, reduction is preserved under instantiation steps. However, the difficulty with this approach is that neither the choice of a single π for all δ or nor the admission of a different π for each δ solves the problem:

EXAMPLE NOTE 11.1
Consider the following liberalized δ-step where the additional free δ-variable z^δ occurs in the principal formula, namely the reduction of $\forall y.\ z^\delta{\neq}y,\quad z^\delta{=}x^\gamma$
to $z^\delta{\neq}y^{\delta^+},\quad z^\delta{=}x^\gamma$
For the e of Example 2.19 (which gives x^γ the value of y^{δ^+}) the lower sequent is $(e, \mathcal{A})$-valid.

Different π: The admission of a different π for each δ seems to be necessary due to the following argumentation: In case of $R = \emptyset$, the upper sequent must be $(\pi \uplus {}_{\mathrm{V}_\delta\setminus\{y^{\delta^+}\}}\!\upharpoonleft\delta, e, \mathcal{A})$-valid for all δ. This holds only when $\pi : \{y^{\delta^+}\} \to \mathcal{A}$ changes when the δ-value of z^δ changes:

E.g., for $\delta := \{y^{\delta^+}\mapsto a,\ z^\delta\mapsto b\}$ we need $\pi(y^{\delta^+}) := b$,
while for $\delta := \{y^{\delta^+}\mapsto b,\ z^\delta\mapsto a\}$ we need $\pi(y^{\delta^+}) := a$.

Indeed, in the reduction above, y^{δ^+} is functionally dependent on z^δ. This dependency is the main reason for our requirement of the liberalized δ-rule to insert (z^δ, y^{δ^+}) into the variable-condition, cf. the end of Section 1.2.2. (The other reason is that we do not have to insert $R\langle\!\langle\{z^\delta\}\rangle\!\rangle \times \{y^{\delta^+}\}$ into the variable-condition R anymore (as was the case in Wirth (1998)) because the transitive closure now takes care of this.)

Single π: The restriction to a single π for all δ seems to be necessary due to the following argumentation: In case of $R = \{(x^\gamma, z^\delta)\}$, the upper sequent of Example Note 11.1 is not R-valid in general. Thus, to preserve the connection between reduction and validity (cf. Lemma 2.31(1)), the step of Example Note 11.1 must not be a reduction, i.e. the upper sequent must not be $(\pi \uplus {}_{\mathrm{V}_\delta\setminus\{y^{\delta^+}\}}\!\upharpoonleft\delta, e, \mathcal{A})$-valid for all δ. Therefore, π must not depend on the δ-value of z^δ, contrary to the item above. Note that such a dependency would effectively allow x^γ to read the value of z^δ, which is explicitly forbidden by the variable-condition R.

Thus, the only solution can be that π (just like e) depends on some values of δ but not on others. Since we are interested in extracting information on the solution of free γ-variables of the original theorem from a completed proof, we want to have the additional possibility to look up what role the free δ^+-variables introduced by liberalized δ-steps really play. And this is what the *choice-conditions* are all about.

Note 12: It should be pointed out that the "some π" in this definition is something we can play around with. Indeed, in Wirth (1998), Definition 5.7 (resp. Definition 4.4 in short version), we can read "each π" instead, which is just the other extreme. The reason why we prefer "some π" to "each π" here and in Wirth (2002) is that the latter results in more valid formulas (e.g. (E2) in Wirth (2002)) and makes theorem proving easier. Contrary to the former and to all semantics for Hilbert's ε in the literature, the latter frees us from considering all possible choices: We just have to pick a single one and fix it in a proof step. As the major notion here and in Wirth (2002) is not validity but reduction (cf. Definition 2.30), where the quantification of π must be universal no matter how we quantify in the notion of (C, R)-validity, changing the quantification of π in Definition 2.27 would only have very local consequences. Roughly speaking, only Lemma 2.31(5a) and Lemma B.6(1) become false for a different choice on the quantification of π in Definition 2.27.

Note 13: For example, a drawback of the implicit induction calculus of Bachmair (1988) (implemented as the UNICOM system, cf. Gramlich & Lindner (1991)) is that every simplification has to reduce the induction conclusion in the induction ordering $<$. Thus, the more reduction steps, the smaller the goals, and the less likely a successful completion of the proof, because this means to find an induction hypothesis being smaller than the goals in $\lesssim$. This can be avoided in our framework by requiring the simplified induction conclusion to be smaller only in $\lesssim$ instead of in $<$.

Note 14: Exceptions are: EXPANDER (Padawitz (1998)) admits any relation, but soundness holds only if it is wellfounded. UNICOM (Gramlich & Lindner (1991)) and SPIKE (Bouhoula & Rusinowitch (1995)) admit the adjustment of some parameters for the induction ordering in advance. Note that NQTHM and other explicit induction systems can be seen to have a fixed induction ordering when we augment the weight terms with the information on how the induction ordering is constructed from a fixed set of combinations, such as it is done in QUODLIBET (Avenhaus &al. (2003)). E.g., instead of comparing a tuple like (x, y, z) in "length–lex($\prec$)" we can take it to be the ordinal number $\omega^2(x{+}1) + \omega(y{+}1) + (z{+}1)$.

Note 15: Groundedness was first defined in Wirth & Becker (1995) under the name "foundedness", which, however, is too easily confused with "wellfoundedness".

Note 16: Note that an Instantiation step can be unsafe if free δ^+-variables are instantiated, cf. Definition B.8.

Note 17: Although it might be possible to instantiate more variables than the ones from Y, this does not seem to be necessary due to the following arguments:

1. To include any $y^{\delta^-} \in \mathcal{V}_{\delta^-}(\varPhi, \daleth)$ into Y we can extend R' with
$$\mathcal{V}_{\gamma\delta^+}(\varPhi, \daleth) \times \{y^{\delta^-}\}$$
provided that R' is still wellfounded after the extension. If this extension of R' makes a query variable useless (i.e. blocks a solution for a free γ-variable), we have to take a higher-order query variable instead, cf. Section 3.3.

2. I do not known a more general approach in the literature. For example, in Baaz &al. (1997), an application of a δ-rule triggers an induction on the variable y of the quantifier removed by the δ-rule. In our approach, the δ-rule application replaces y with a new free δ^--variable y^{δ^-} and extends the variable-condition with $\mathcal{V}_{\gamma\delta^+}(\varPhi, \daleth) \times \{y^{\delta^-}\}$ so that $y^{\delta^-} \in Y$ holds indeed.

Note 18: Note that "$p(a)$" may abbreviate "$p(a)=\mathsf{true}$". To express the wellfoundedness of $<$ properly, "$\alpha\rightarrow\mathsf{bool}$" must have the *standard* interpretation of a predicate over "α".

Note 19: Indeed, for the alternative notions in Note 10, we get $R' := \emptyset$ here because (y^γ_2, y^γ_1) and (y^γ_2, y^γ_3) from Γ_σ are removed, just as the edge from v^{δ^+} to y^γ in the example of Note 10, because there are no out-going R'-edges from y^γ_1 and y^γ_3.

Note 20: Note that we cannot take arbitrary length because the lexicographic combination of arbitrary length of wellfounded orderings is not wellfounded: $(1) > (0,1) > (0,0,1) > \cdots$. This length is not limiting the QUODLIBET system, however, because it is not implemented: If a proof attempt is successful it has used only a finite number of finite terms and we can assume that the limit is the maximum length of lexicographic combination occurring in them.

Note 21: $x^{\delta^-}\in\mathrm{V}_{\delta^-}$ is in *solved form* in the weighted sequent $\varGamma\ (x^{\delta^-}{\neq}t)\ \varPi;\ \sqsupset$ if
$x^{\delta^-}\notin\mathcal{V}(t,\varGamma\varPi,\sqsupset)$ and $\mathcal{V}_{\gamma\delta^+}(t,\varGamma\varPi,\sqsupset)\subseteq R^+\langle\!\langle\{x^{\delta^-}\}\rangle\!\rangle$.

Note 22: If wellfoundedness or termination were a first-order property, the first-order theory of the Peano algebra of natural numbers would be first-order axiomatizable and enumerable, but it is not even arithmetically definable, cf. e.g. Enderton (1973), p. 228.

Note 23: Actually, the possibility to be lazy simplifies things a little bit when different induction schemes are in conflict with each other. To get an idea on this, compare Walther (1992) with Kühler (2000), Section 8.3.

Note 24: In reductive theorem proving, there is one disadvantage, however, of the liberalized δ-rule compared to the non-liberalized δ-rule. Sometimes the liberalized δ-rule results in a larger variable-condition because it introduces dependencies from the free δ^--variables of the principle formula. This is necessary for the soundness of lemma and induction-hypothesis application. One consequence of this is that simplification becomes more difficult: For example, in the second tree in Section 3.2.2, we *safely* removed the literal $x^{\delta^-}_0{\neq}\mathsf{s}(x^{\delta^-}_1)$ from

$$x^{\delta^-}_0{\neq}\mathsf{s}(x^{\delta^-}_1),\ \mathsf{P}(\mathsf{s}(x^{\delta^-}_1));\ w^\gamma_1(\mathsf{s}(x^{\delta^-}_1))$$

because $x^{\delta^-}_0$ was in solved form in this sequent, cf. Note 21. If we had applied the *liberalized* δ-rule instead, we would have got

$$x^{\delta^-}_0{\neq}\mathsf{s}(x^{\delta^+}_1),\ \mathsf{P}(\mathsf{s}(x^{\delta^+}_1));\ w^\gamma_1(\mathsf{s}(x^{\delta^+}_1))$$

where $x^{\delta^-}_0$ is not in solved form because this sequent contains the free δ^+-variable $x^{\delta^+}_1$ which is not in $R^+\langle\!\langle\{x^{\delta^-}_0\}\rangle\!\rangle$. Moreover, we cannot extend the variable-condition R such that $R^+\langle\!\langle\{x^{\delta^-}_0\}\rangle\!\rangle$ contains $x^{\delta^+}_1$ because the liberalized δ-rule has introduced the dependency $(x^{\delta^-}_0, x^{\delta^+}_1)$ into R, so that R would become cyclic. Note that $x^{\delta^+}_1$ stands for $\varepsilon y.\ (x^{\delta^-}_0{=}\mathsf{s}(y))$, which means that $x^{\delta^-}_0$ still occurs hidden the latter sequent. Indeed, under the variable-condition $R := \{(x^{\delta^-}_0, x^{\delta^+}_1)\}$, the choice-condition $C := \{(x^{\delta^+}_1, (x^{\delta^-}_0{=}\mathsf{s}(x^{\delta^+}_1)))\}$, and $(\mathsf{nat1})$ from Section 1.1.1, the removal of $x^{\delta^-}_0{\neq}\mathsf{s}(x^{\delta^+}_1)$ from $x^{\delta^-}_0{\neq}\mathsf{s}(x^{\delta^+}_1),\ x^{\delta^+}_1{\neq}\mathsf{0};\ \ldots$ is not safe in the sense of Definition 2.47; to wit, let $\mathcal{A}$ have the universe $\{+,-\}\times\mathbf{N}$ with $\mathsf{s}^{\mathcal{A}}(+,n) := (+,n{+}1)$, $\mathsf{s}^{\mathcal{A}}(-,n{+}1) := (-,n)$, $\mathsf{s}^{\mathcal{A}}(-,0) := (+,1)$, and $\mathsf{0}^{\mathcal{A}} := (+,0)$, and set

$$\pi(x^{\delta^+}_1)(\tau) := \left\{\begin{array}{ll}(+,0) & \text{if } \tau(x^{\delta^-}_0){=}(+,0)\\ (-,0) & \text{if } \tau(x^{\delta^-}_0){=}(+,1)\\ (+,n{+}1) & \text{if } \tau(x^{\delta^-}_0){=}(+,n{+}2)\\ (-,n{+}1) & \text{if } \tau(x^{\delta^-}_0){=}(-,n)\end{array}\right\},$$

which is compatible with (C,R). Moreover, considering $\tau(x^{\delta^-}_0)=\mathsf{0}^{\mathcal{A}}$, it can be easily seen that

$$x^{\delta^-}_0{\neq}\mathsf{s}(x^{\delta^+}_1),\ x^{\delta^+}_1{\neq}\mathsf{0}$$

is $(\pi, e, \mathcal{A})$-valid, but $x^{\delta^+}_1{\neq}\mathsf{0}$ is not, thereby violating safeness.

There is, however, a general way to overcome this shortcoming for constructive domains. For our special case of natural numbers it looks as follows: When we add the axiom $(\mathtt{nat3})$ from Section 1.1.2, then the removal of $x_0^{\delta^-}{\neq}\mathsf{s}(x_1^{\delta^+})$ from $x_0^{\delta^-}{\neq}\mathsf{s}(x_1^{\delta^+}),\ \mathit{\Gamma};\ \ldots$ is always safe because the image of the predecessor *function* on the universe without $\mathsf{0}^{\mathcal{A}}$ is the whole universe and if $\mathit{\Gamma}$ is false for $\tau(x_0^{\delta^-}){=}\mathsf{0}^{\mathcal{A}}$ then $x_0^{\delta^-}{\neq}\mathsf{s}(x_1^{\delta^+}),\ \mathit{\Gamma}$ is false for the τ' which differs from τ in $\tau'(x_0^{\delta^-}){=}\mathsf{s}^{\mathcal{A}}(\pi(x_1^{\delta^+})(\tau))$ because then $\pi(x_1^{\delta^+})(\tau'){=}\pi(x_1^{\delta^+})(\tau)$.

Note 25: This asymmetry results from the following line of argumentation: For some new variable $z \in \mathrm{V}_{\mathrm{bound}}$ and t denoting the term $\varepsilon z.\ ((\neg A \Rightarrow x{=}z) \wedge \neg A\{x{\mapsto}z\})$, using the logical equivalence of $\forall x.\ (A{\vee}B)$ with $\forall x.\ A \vee \forall x.\ (B\{x{\mapsto}t\})$ and then the logical equivalence of $\forall x.\ A$ with $\exists x.\ (A\{x{\mapsto}t\})$, we see that $\forall x.\ (A{\vee}B)$ is logically equivalent with $\exists x.\ (A\{x{\mapsto}t\})\ \vee\ \forall x.\ (B\{x{\mapsto}t\})$.

Note 26: For the alternative notions in Note 10, we have to replace this sentence with the following: As R' is the σ-update of R, we have

$$\Delta_\sigma R^* {\upharpoonright}_{\mathrm{V}_\delta} \subseteq \Delta_\sigma R R^* \cup \Delta_\sigma {\upharpoonright}_{\mathrm{V}_\delta} = \Delta_\sigma R({}_{\mathrm{V}_\delta}{\upharpoonleft} R)^* \cup \Delta_\sigma {\upharpoonright}_{\mathrm{V}_\delta} \subseteq R'^{+},$$

the second step being due to $\mathrm{ran}(R) \subseteq \mathrm{V}_\delta$ for any *alternative* variable-condition R. Similarly, ${}_{\mathrm{V}_\delta}{\upharpoonleft}(R^+) = ({}_{\mathrm{V}_\delta}{\upharpoonleft} R)^+ \subseteq R'^{+}$.
Q.e.d. (Claim 1)

Note 27: For the alternative notions in Note 10, we have to replace this sentence with the following: As R' is the σ-update of R, we have

$$\begin{aligned}({}_{\mathrm{V}_\gamma\setminus\mathrm{dom}(\sigma)}{\upharpoonleft}\mathrm{id} \cup \Gamma_\sigma)R^* {\upharpoonright}_{\mathrm{V}_\delta} \ \subseteq\ ({}_{\mathrm{V}_\gamma\setminus\mathrm{dom}(\sigma)}{\upharpoonleft}\mathrm{id} \cup \Gamma_\sigma)RR^* \ \cup\ {}_{\mathrm{V}_{\mathrm{free}}}{\upharpoonleft}\mathrm{id} \ \cup\ \Gamma_\sigma {\upharpoonright}_{\mathrm{V}_\delta} \ =\\ ({}_{\mathrm{V}_\gamma\setminus\mathrm{dom}(\sigma)}{\upharpoonleft}\mathrm{id} \cup \Gamma_\sigma)R({}_{\mathrm{V}_\delta}{\upharpoonleft}R)^* \ \cup\ {}_{\mathrm{V}_{\mathrm{free}}}{\upharpoonleft}\mathrm{id} \ \cup\ \Gamma_\sigma {\upharpoonright}_{\mathrm{V}_\delta} \ \subseteq\ R'^{*},\end{aligned}$$

the second step being due to $\mathrm{ran}(R) \subseteq \mathrm{V}_\delta$ for any *alternative* variable-condition R. Q.e.d. (Claim 2)

Note 28: For the alternative notions in Note 10, we have to deviate here in the following way: Moreover, as R' is the σ-update of R, we have

$$R' = ({}_{\mathrm{V}_\delta\cup(\mathrm{V}_\gamma\setminus\mathrm{dom}(\sigma))}{\upharpoonleft}\mathrm{id} \cup \Gamma_\sigma \cup \Delta_\sigma)R \cup (\Gamma_\sigma \cup \Delta_\sigma){\upharpoonright}_{\mathrm{V}_\delta}.$$

As $(R' \cup S_{e'})^+$ is a wellfounded ordering, so is its subset

$$({}_{\mathrm{V}_\delta\cup(\mathrm{V}_\gamma\setminus\mathrm{dom}(\sigma))}{\upharpoonleft}R \cup S_{e'}{\upharpoonright}_{\mathrm{V}_\gamma\setminus\mathrm{dom}(\sigma)} \cup S_{e'}\Gamma_\sigma{\upharpoonright}_{\mathrm{V}_\gamma}R \cup \Delta_\sigma{\upharpoonright}_{\mathrm{V}_\gamma}R)^+.$$

The alternative version of a variable-condition guarantees $\mathrm{ran}(R) \subseteq \mathrm{V}_\delta$. Thus, additional steps with ${}_{\mathrm{V}_\gamma\cap\mathrm{dom}(\sigma)}{\upharpoonleft}R$ must cause immediate termination; i.e.

$$(R \cup S_{e'}{\upharpoonright}_{\mathrm{V}_\gamma\setminus\mathrm{dom}(\sigma)} \cup S_{e'}\Gamma_\sigma{\upharpoonright}_{\mathrm{V}_\gamma}R \cup \Delta_\sigma{\upharpoonright}_{\mathrm{V}_\gamma}R)^+$$

is a wellfounded ordering, too. As $\mathrm{ran}(\Gamma_\sigma{\upharpoonright}_{\mathrm{V}_\gamma} \cup \Delta_\sigma{\upharpoonright}_{\mathrm{V}_\gamma}) \subseteq \mathrm{V}_\gamma$ and $\mathrm{dom}(S_{e'} \cup \Delta_\sigma) \subseteq \mathrm{V}_\delta$

$$(R \cup S_{e'}{\upharpoonright}_{\mathrm{V}_\gamma\setminus\mathrm{dom}(\sigma)} \cup S_{e'}\Gamma_\sigma{\upharpoonright}_{\mathrm{V}_\gamma} \cup \Delta_\sigma{\upharpoonright}_{\mathrm{V}_\gamma})^+$$

is a wellfounded ordering, which is equal to $(R \cup S_e)^+$ by definition of S_e. Q.e.d. (Claim 3)

Note 29: For the alternative notions in Note 10, we have to deviate here in the following way: Moreover, as R' is the σ-update of R, we have

$$R' = ({}_{\mathrm{V}_\delta\cup(\mathrm{V}_\gamma\setminus\mathrm{dom}(\sigma))}{\upharpoonleft}\mathrm{id} \cup \Gamma_\sigma \cup \Delta_\sigma)R \cup (\Gamma_\sigma \cup \Delta_\sigma){\upharpoonright}_{\mathrm{V}_\delta}.$$

As $R' \cup S_{e'} \cup S_{\pi'}$ is wellfounded, the subset

$${}_{\mathrm{V}_\delta\cup(\mathrm{V}_\gamma\setminus\mathrm{dom}(\sigma))}{\upharpoonleft}R \cup (S_{\pi'} \cup {}_{\mathrm{V}_{\delta\text{–}}}{\upharpoonleft}\mathrm{id})(S_{e'}({}_{\mathrm{V}_\gamma\setminus\mathrm{dom}(\sigma)}{\upharpoonleft}\mathrm{id} \cup \Gamma_\sigma{\upharpoonright}_{\mathrm{V}_\gamma}) \cup \Delta_\sigma{\upharpoonright}_{\mathrm{V}_\gamma})R \cup {}_{\mathrm{V}_\delta}{\upharpoonleft}(R' \cup S_{e'} \cup S_{\pi'})^+{\upharpoonright}_{\mathrm{V}_\delta}$$

of its transitive closure is wellfounded, too.
The alternative version of a variable-condition guarantees $\mathrm{ran}(R) \subseteq \mathrm{V}_\delta$. Thus, additional steps with ${}_{\mathrm{V}_\gamma\cap\mathrm{dom}(\sigma)}{\upharpoonleft}R$ must cause immediate termination; i.e.

$$R \cup (S_{\pi'} \cup {}_{\mathrm{V}_{\delta}-}\!\upharpoonleft\mathrm{id})(S_{e'}({}_{\mathrm{V}_{\gamma}\setminus\mathrm{dom}(\sigma)}\!\upharpoonleft\mathrm{id} \cup \Gamma_{\sigma}\!\upharpoonright_{\mathrm{V}_{\gamma}}) \cup \Delta_{\sigma}\!\upharpoonright_{\mathrm{V}_{\gamma}})R \cup {}_{\mathrm{V}_{\delta}}\!\upharpoonleft(R' \cup S_{e'} \cup S_{\pi'})^{+}\!\upharpoonright_{\mathrm{V}_{\delta}}$$

is wellfounded, too.
As $\mathrm{ran}(S_{e'}({}_{\mathrm{V}_{\gamma}\setminus\mathrm{dom}(\sigma)}\!\upharpoonleft\mathrm{id} \cup \Gamma_{\sigma}\!\upharpoonright_{\mathrm{V}_{\gamma}}) \cup \Delta_{\sigma}\!\upharpoonright_{\mathrm{V}_{\gamma}}) \subseteq \mathrm{V}_{\gamma}$ and $\mathrm{dom}(S_{\pi'} \cup {}_{\mathrm{V}_{\delta}-}\!\upharpoonleft\mathrm{id}) \subseteq \mathrm{V}_{\delta}$

$$R \cup (S_{\pi'} \cup {}_{\mathrm{V}_{\delta}-}\!\upharpoonleft\mathrm{id})(S_{e'}({}_{\mathrm{V}_{\gamma}\setminus\mathrm{dom}(\sigma)}\!\upharpoonleft\mathrm{id} \cup \Gamma_{\sigma}\!\upharpoonright_{\mathrm{V}_{\gamma}}) \cup \Delta_{\sigma}\!\upharpoonright_{\mathrm{V}_{\gamma}}) \cup {}_{\mathrm{V}_{\delta}}\!\upharpoonleft(R' \cup S_{e'} \cup S_{\pi'})^{+}\!\upharpoonright_{\mathrm{V}_{\delta}}$$

is wellfounded, which is equal to $R \cup S_{e} \cup {}_{\mathrm{V}_{\delta}}\!\upharpoonleft(R' \cup S_{e'} \cup S_{\pi'})^{+}\!\upharpoonright_{\mathrm{V}_{\delta}}$ by definition of S_{e}. Q.e.d. (Claim 5)

Note 30: For the alternative notions in Note 10, we have to deviate here in the following way: As R' is the σ-update of R, we have

$$R' = ({}_{\mathrm{V}_{\delta}\cup(\mathrm{V}_{\gamma}\setminus\mathrm{dom}(\sigma))}\!\upharpoonleft\mathrm{id} \cup \Gamma_{\sigma} \cup \Delta_{\sigma})\circ R \cup (\Gamma_{\sigma} \cup \Delta_{\sigma})\!\upharpoonright_{\mathrm{V}_{\delta}}.$$

As the alternative version of a variable-condition guarantees $\mathrm{ran}(R) \subseteq \mathrm{V}_{\delta}$ and $\lhd$ is transitive, we have ${}_{\mathrm{V}_{\delta}}\!\upharpoonleft(R^{+}) \subseteq \lhd$ and $({}_{\mathrm{V}_{\gamma}\setminus\mathrm{dom}(\sigma)}\!\upharpoonleft\mathrm{id} \cup \Gamma_{\sigma} \cup \Delta_{\sigma})\circ R^{+} \subseteq \lhd$. The latter implies $S_{e}\circ R^{+} \subseteq \lhd$ by (B.5.1). Q.e.d. (Claim 2)

E References

Peter B. Andrews (1972). *General Models, Descriptions, and Choice in Type Theory.* J. Symbolic Logic **37**, pp. 385–394.

Peter B. Andrews (1981). *Theorem Proving via General Matings.* J. ACM **28**, pp. 193–214, ACM Press.

Peter B. Andrews (2002). *An Introduction to Mathematical Logic and Type Theory: To Truth Through Proof.* 2nd ed., Academic Press.

Serge Autexier (2003). *Hierarchical Contextual Reasoning.* Ph.D. thesis, FR Informatik, Saarland Univ..

Serge Autexier, Dieter Hutter, Heiko Mantel, Axel Schairer (1999). INKA *5.0 — A Logical Voyager.* 16th CADE 1999, LNAI 1632, pp. 207–211, Springer.

Jürgen Avenhaus, Ulrich Kühler, Tobias Schmidt-Samoa, Claus-Peter Wirth (2003). *How to Prove Inductive Theorems?* QUODLIBET*!*. 19th CADE 2003, LNAI 2741, pp. 328–333, Springer.

Matthias Baaz, Christian G. Fermüller (1995). *Non-elementary Speedups between Different Versions of Tableaus.* 4th TABLEAU 1995, LNAI 918, pp. 217–230, Springer.

Matthias Baaz, Uwe Egly, Christian G. Fermüller (1997). *Lean Induction Principles for Tableaus.* 6th TABLEAU 1997, LNAI 1227, pp. 62–75, Springer.

Leo Bachmair (1988). *Proof By Consistency in Equational Theories.* 3rd IEEE symposium on Logic In Computer Sci., pp. 228–233, IEEE Press.

Klaus Barner (2001). *Das Leben Fermats.* DMV-Mitteilungen 3/2001, pp. 12–26.

Peter Baumgartner, Ulrich Furbach, Frieder Stolzenburg (1997). *Computing Answers with Model Elimination.* Artificial Intelligence **90**, pp. 135–176.

Bernhard Beckert, Reiner Hähnle (1998). *Analytic Tableaus.* In: Bibel & Schmitt (1998), Vol. 1, pp. 11–41.

Bernhard Beckert, Reiner Hähnle, Peter H. Schmitt (1993). *The Even More Liberalized δ-Rule in Free-Variable Semantic Tableaus.* Kurt Gödel Colloquium, LNCS 713, pp. 108–119, Springer.

Christoph Benzmüller, Chad Brown, Michaël Kohlhase (2004). *Higher Order Semantics and Extensionality.* To appear, J. Symbolic Logic.

Wolfgang Bibel (1987). *Automated Theorem Proving.* 2nd rev. ed., Vieweg, Braunschweig.

Wolfgang Bibel, Peter H. Schmitt (eds.) (1998). *Automated Deduction — A Basis for Applications.* Kluwer.

Susanne Biundo, Birgit Hummel, Dieter Hutter, Christoph Walther (1986). *The Karlsruhe Induction Theorem Proving System.* 8th CADE 1986, LNCS 230, pp. 672–674, Springer.

Adel Bouhoula, Michaël Rusinowitch (1995). *Implicit Induction in Conditional Theories.* J. Automated Reasoning **14**, pp. 189–235, Kluwer.

Robert S. Boyer, J Strother Moore (1979). *A Computational Logic.* Academic Press.

Robert S. Boyer, J Strother Moore (1988). *A Computational Logic Handbook.* Academic Press.

Alan Bundy (2001). *The Automation of Proof by Mathematical Induction.* In: Robinson & Voronkov (2001), Vol. 1, pp. 845–911.

W. H. Bussey (1917). *The Origin of Mathematical Induction.* American Mathematical Monthly **XXIV**, pp. 199–207.

Ricardo Caferra, Gernot Salzer (eds.) (2000). *Automated Deduction in Classical and Non-Classical Logics.* LNAI 1761, Springer.

Domenico Cantone, Marianna Nicolosi-Asmundo (2000). *A Further and Effective Liberalization of the δ-Rule in Free-Variable Semantic Tableaus.* In: Caferra & Salzer (2000), pp. 109–125.

Serenella Cerrito, Marta Cialdea (2001). *Free-Variable Tableaus for Constant-Domain Quantified Modal Logics with Rigid and Non-Rigid Designation.* 1st IJCAR 2001, LNAI 2083, pp. 137–151, Springer.

Herbert B. Enderton (1973). *A Mathematical Introduction to Logic.* 2nd printing, Academic Press.

Euclid of Alexandria (ca. 300 B.C.). *Elements.* D. E. Joyce, Dept. Math. & Comp. Sci., Clark Univ., Worcester, MA. `http://aleph0.clarku.edu/~djoyce/java/elements/Euclid.html (March 24, 2003).`

Pierre Fermat (1891). *Œuvres de Fermat.* Paul Tannery, Charles Henry (eds.), Gauthier-Villars, Paris.

Melvin C. Fitting (1996). *First-Order Logic and Automated Theorem Proving.* 2nd extd. ed., Springer.

Melvin C. Fitting (1999). *On Quantified Modal Logic.* Fundamenta Informaticae **39**, pp. 105–121.

Melvin C. Fitting (2002). *Types, Tableaus, and Gödel's God.* Kluwer.

Kurt von Fritz (1945). *Die Entdeckung der Inkommensurabilität durch Hippasos von Metapont.* Annals of Mathematics **46**, pp. 242–264. Also in: Oscar Becker. *Zur Geschichte der griechischen Mathematik*, pp. 271–308.

Dov M. Gabbay, C. J. Hogger, J. Alan Robinson (eds.) (1993 ff.). *Handbook of Logic in Artificial Intelligence and Logic Programming.* Clarendon Press.

Gerhard Gentzen (1935). *Untersuchungen über das logische Schließen.* Mathematische Zeitschrift **39**, pp. 176-210, 405–431.

Gerhard Gentzen (1938). *Die gegenwärtige Lage in der mathematischen Grundlagenforschung – Neue Fassung des Widerspruchsfreiheitsbeweises für die reine Zahlentheorie.* Forschungen zur Logik und zur Grundlegung der exakten Wissenschaften, Folge 4, Leipzig.

Alfons Geser (1995). *A Principle of Non-Wellfounded Induction.* In: Tiziana Margaria (ed.). Kolloquium Programmiersprachen und Grundlagen der Programmierung, MIP–9519, pp. 117–124, Univ. Passau.

Martin Giese (1998). *Integriertes automatisches und interaktives Beweisen: die Kalkülebene.* Master's thesis, Univ. Karlsruhe. `http://i11www.ira.uka.de/~giese/da.ps.gz (May 09, 2000).`

Martin Giese, Wolfgang Ahrendt (1999). *Hilbert's ε-Terms in Automated Theorem Proving.* 8th TABLEAU 1999, LNAI 1617, pp. 171–185, Springer.

Leonard Gillman (1987). *Writing Mathematics Well.* The Mathematical Association of America.

Kurt Gödel (1931). *Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I.* Monatshefte für Mathematik und Physik **38**, pp. 173–198.

Kurt Gödel (1986 ff.). *Collected Works.* Solomon Feferman (ed.), Oxford Univ. Press.

Bernhard Gramlich, Wolfgang Lindner (1991). *A Guide to* UNICOM*, an Inductive Theorem Prover Based on Rewriting and Completion Techniques.* SEKI-Report SR–91–17 (SFB), FB Informatik, Univ. Kaiserslautern. `http://agent.informatik.uni-kl.de/seki/1991/Lindner.SR-91-17.ps.gz (May 09, 2000).`

Bernhard Gramlich, Claus-Peter Wirth (1996). *Confluence of Terminating Conditional Term Rewriting Systems Revisited.* 7th RTA 1996, LNCS 1103, pp. 245–259, Springer.

Reiner Hähnle, Peter H. Schmitt (1994). *The Liberalized δ-Rule in Free-Variable Semantic Tableaus.* J. Automated Reasoning **13**, pp. 211–221, Kluwer.

David Hilbert, Paul Bernays (1968/70). *Grundlagen der Mathematik.* 2nd ed., Springer.

K. Jaakko J. Hintikka (1996). *The Principles of Mathematics Revisited.* Cambridge Univ. Press.

Paul Howard, Jean E. Rubin (1998). *Consequences of the Axiom of Choice.* Math. Surv. and Monographs, Vol. 58, American Math. Society. `http://www.math.purdue.edu/~jer/Papers/conseq.html (Oct. 15, 2002).`

Dieter Hutter, Alan Bundy (1999). *The Design of the CADE-16 Inductive Theorem Prover Contest.* 16th CADE 1999, LNAI 1632, pp. 374–377, Springer.

Stig Kanger (1963). *A Simplified Proof Method for Elementary Logic.* In: Siekmann & Wrightson (1983), Vol. 1, pp. 364–371.

Deepak Kapur, Hantao Zhang (1989). *An Overview of Rewrite Rule Laboratory (*RRL*).* 3rd RTA 1989, LNCS 355, pp. 559–563, Springer.

Victor J. Katz (1998). *A History of Mathematics: An Introduction.* 2nd ed., Addison-Wesley.

Matt Kaufmann, Panagiotis Manolios, J Strother Moore (2000). *Computer-Aided Reasoning: An Approach.* Kluwer.

Donald E. Knuth (1997 f.). *The Art of Computer Programming.* 3rd ed., Addison-Wesley.

Michaël Kohlhase (1995). *Higher-Order Tableaus.* 4th TABLEAU 1995, LNAI 918, pp. 294–309, Springer. Revised version is: Kohlhase (1998).

Michaël Kohlhase (1998). *Higher-Order Automated Theorem Proving.* In: Bibel & Schmitt (1998), Vol. 1, pp. 431–462.

Georg Kreisel (1965). *Mathematical Logic.* In: T. L. Saaty (ed.). Lectures on Modern Mathematics, Vol. III, pp. 95–195, John Wiley & Sons, New York.

Ulrich Kühler (2000). *A Tactic-Based Inductive Theorem Prover for Data Types with Partial Operations.* Ph.D. thesis, Infix, Sankt Augustin.

Ulrich Kühler, Claus-Peter Wirth (1996). *Conditional Equational Specifications of Data Types with Partial Operations for Inductive Theorem Proving.* SEKI-Report SR–96–11, FB Informatik, Univ. Kaiserslautern. Short version in: 8th RTA 1997, LNCS 1232, pp. 38–52, Springer. `http://www.ags.uni-sb.de/~cp/p/rta97/welcome.html (Aug. 05, 2001).`

Richard C.-T. Lee (1967). *A Completeness Theorem and a Computer Program for Finding Theorems Derivable from Given Axioms.* Ph.D. thesis, Univ. of California, Berkeley.

Alexander Leitsch (1997). *The Resolution Calculus.* Springer.

Vladimir A. Lifschitz (1971). *Specialization of the Form of Deduction in the Predicate Calculus with Equality and Function Symbols.* In: Orevkov (1971), pp. 1–24.

Vladimir A. Lifschitz (1989). *What Is the Inverse Method?.* J. Automated Reasoning **5**, pp. 1–23, Kluwer.

Michael Sean Mahoney (1994). *The Mathematical Career of Pierre de Fermat.* 2nd ed., Princeton Univ. Press.

Sergey Yu. Maslov (1971). *The Inverse Method for Establishing Deducibility for Logic Calculi.* In: Orevkov (1971), pp. 25–96.

Dale Miller (1992). *Unification under a Mixed Prefix.* J. Symbolic Computation **14**, pp. 321–358, Academic Press.

Andreas Nonnengart (1996). *Strong Skolemization.* MPI–I–96–2–010, Max Planck-Inst. für Informatik, Saarbrücken.

Vladimir P. Orevkov (1971). *The Calculi of Symbolic Logic.I.* American Mathematical Society, Providence, Rhode Island.

Peter Padawitz (1996). *Inductive Theorem Proving for Design Specifications.* J. Symbolic Computation **21**, pp. 41–99, Academic Press.

Peter Padawitz (1998). EXPANDER. *A System for Testing and Verifying Functional Logic Programs.* `http://LS5.cs.uni-dortmund.de/~peter/ExpaTex.ps.gz (Sept. 14, 1999).`

Blaise Pascal (1954). *Œuvres Complètes.* Jacques Chevalier (ed.), Gallimard, Paris.

Michaël S. Paterson, Mark N. Wegman (1978). *Linear Unification.* J. Computer and System Sci. **16**, pp. 158–167, Academic Press.

Dag Prawitz (1960). *An Improved Proof Procedure.* In: Siekmann & Wrightson (1983), Vol. 1, pp. 159–199.

Martin Protzen (1994). *Lazy Generation of Induction Hypotheses.* 12th CADE 1994, LNAI 814, pp. 42–56, Springer. Long version in: Protzen (1995).

Martin Protzen (1995). *Lazy Generation of Induction Hypotheses and Patching Faulty Conjectures.* Ph.D. thesis, Infix, Sankt Augustin.

J. Alan Robinson (1965). *A Machine-Oriented Logic based on the Resolution Principle.* In: Siekmann & Wrightson (1983), Vol. 1, pp. 397–415.

J. Alan Robinson, Andrei Voronkov (eds.) (2001). *Handbook of Automated Reasoning.* Elsevier.

Herman Rubin, Jean E. Rubin (1985). *Equivalents of the Axiom of Choice.* Elsevier.

Jörg H. Siekmann, Graham Wrightson (eds.) (1983). *Automation of Reasoning.* Springer.

Raymond M. Smullyan (1968). *First-Order Logic.* Springer.

Wayne Snyder, Jean Gallier (1989). *Higher-Order Unification Revisited: Complete Sets of Transformations.* J. Symbolic Computation **8**, pp. 101–140, Academic Press.

Lincoln A. Wallen (1990). *Automated Proof Search in Non-Classical Logics.* MIT Press. Cf., however, `http://www.ags.uni-sb.de/~cp/p/wallen/all.txt` for some obsolete aspects of this fascinating book.

Christoph Walther (1992). *Computing Induction Axioms.* 3rd LPAR 1992, LNAI 624, pp. 381–392, Springer.

Christoph Walther (1994). *Mathematical Induction.* In: Gabbay &al. (1993 ff.), Vol. 2, pp. 127–228.

Claus-Peter Wirth (1997). *Positive/Negative-Conditional Equations: A Constructor-Based Framework for Specification and Inductive Theorem Proving.* Ph.D. thesis, Verlag Dr. Kovač, Hamburg.

Claus-Peter Wirth (1998). *Full First-Order Sequent and Tableau Calculi With Preservation of Solutions and the Liberalized δ-Rule but Without Skolemization.* Report 698/1998, FB Informatik, Univ. Dortmund. Short version in: Gernot Salzer, Ricardo Caferra (eds.). Proc. 2nd Int. Workshop on First-Order Theorem Proving (FTP'98), pp. 244–255, Tech. Univ. Vienna, 1998. Short version also in: Caferra & Salzer (2000), pp. 283–298. `http://www.ags.uni-sb.de/~cp/p/ftp98/welcome.html (Aug. 05, 2001).`

Claus-Peter Wirth (1999). *Full First-Order Free-Variable Sequents and Tableaus in Implicit Induction.* 8th TABLEAU 1999, LNAI 1617, pp. 293–307, Springer. `http://www.ags.uni-sb.de/~cp/p/tab99/welcome.html (Aug. 05, 2001).`

Claus-Peter Wirth (2002). *A New Indefinite Semantics for Hilbert's epsilon.* 11th TABLEAU 2002, LNAI 2381, pp. 298–314, Springer. `http://www.ags.uni-sb.de/~cp/p/epsi/welcome.html (Feb. 04, 2002).`

Claus-Peter Wirth (2004). *History and Future of Implicit and Inductionless Induction: Beware the old jade and the zombie!.* Festschrift in Honor of Jörg H. Siekmann, to appear, LNAI, Springer. `http://www.ags.uni-sb.de/~cp/p/zombie/welcome.html (Dec. 02, 2002).`

Claus-Peter Wirth, Klaus Becker (1995). *Abstract Notions and Inference Systems for Proofs by Mathematical Induction.* 4th CTRS 1994, LNCS 968, pp. 353–373, Springer. `http://www.ags.uni-sb.de/~cp/p/ctrs94/welcome.html (Aug. 05, 2001).`

Claus-Peter Wirth, Bernhard Gramlich (1994a). *A Constructor-Based Approach for Positive/Negative-Conditional Equational Specifications.* J. Symbolic Computation **17**, pp. 51–90, Academic Press. `http://www.ags.uni-sb.de/~cp/p/jsc94/welcome.html (Aug. 05, 2001).`

Claus-Peter Wirth, Bernhard Gramlich (1994b). *On Notions of Inductive Validity for First-Order Equational Clauses.* 12th CADE 1994, LNAI 814, pp. 162–176, Springer. `http://www.ags.uni-sb.de/~cp/p/cade94/welcome.html (Aug. 05, 2001).`

Claus-Peter Wirth, Christoph Benzmüller, Armin Fiedler, Andreas Meier, Serge Autexier, Martin Pollet, Carsten Schürmann (2003). *Human-Oriented Theorem Proving — Foundations and Applications.* Lecture course at Universität des Saarlandes, winter semester 2003/4. `http://www.ags.uni-sb.de/~cp/teaching/hotp (Sept. 12, 2003).`

Received 12 September 2003.

Interest Group in Pure and Applied Logics (IGPL)

The Interest Group in Pure and Applied Logics (IGPL) is sponsored by The European Association for Logic, Language and Information (FoLLI), and currently has a membership of over a thousand researchers in various aspects of logic (symbolic, mathematical, computational, philosophical, etc.) from all over the world (currently, more than 50 countries). Our main activity is that of a research and information clearing house.

Our activities include:

- Exchanging information about research problems, references and common interest among group members, and among different communities in pure and applied logic.
- Helping to obtain photocopies of papers to colleagues (under the appropriate copyright restrictions), especially where there may be difficulties of access.
- Supplying review copies of books through the journals on which some of us are editors.
- Helping to organise exchange visits and workshops among members.
- Advising on papers for publication.
- Editing and distributing a Newsletter and a Journal (the first scientific journal on logic which is FULLY electronic: submission, refereeing, revising, typesetting, publishing, distribution; first issue: July 1993): the Logic Journal of the Interest Group on Pure and Applied Logics. (For more information on the Logic Journal of the IGPL, see the Web homepage: http://www.oup.co.uk/igpl)
- Keeping a public archive of papers, abstracts, etc., accessible via ftp.
- Wherever possible, obtaining reductions on group (6 or more) purchases of logic books from publishers.

If you are interested, please send your details (name, postal address, phone, fax, e-mail address, research interests) to:

IGPL Headquarters
c/o Prof. Dov Gabbay
King's College, Dept of Computer Science
Strand
London WC2R 2LS
United Kingdom
e-mail: dg@dcs.kcl.ac.uk

For the organisation: Dov Gabbay, Ruy de Queiroz and Hans Jürgen Ohlbach.