Proof Transformations in Higher-Order Logic

Frank Pfenning

January 1987

Submitted in partial fulfillment

of the requirements

for the Degree of

Doctor of Philosophy

Department of Mathematics

Carnegie Mellon University

Pittsburgh, Pennsylvania

Abstract

We investigate the problem of translating between different styles of proof systems in higher-order logic: *analytic* proofs which are well suited for automated theorem proving, and *non-analytic* deductions which are well suited for the mathematician. Analytic proofs are represented as expansion proofs, $\mathcal{H}$, a form of the sequent calculus we define, non-analytic proofs are represented by natural deductions. A non-deterministic translation algorithm between expansion proofs and $\mathcal{H}$-deductions is presented and its correctness is proven. We also present an algorithm for translation in the other direction and prove its correctness. A cut-elimination algorithm for expansion proofs is given and its partial correctness is proven. Strong termination of this algorithm remains a conjecture for the full higher-order system, but is proven for the first-order fragment. We extend the translations to a non-analytic proof system which contains a primitive notion of equality, while leaving the notion of expansion proof unaltered. This is possible, since a non-extensional equality is definable in our system of type theory. Next we extend analytic and non-analytic proof systems and the translations between them to include extensionality. Finally, we show how the methods and notions used so far apply to the problem of translating expansion proofs into natural deductions. Much care is taken to specify this translation in a modular way (through tactics) which leaves a large number of choices and therefore a lot of room for heuristics to produce elegant natural deductions. A practically very useful extension, called symmetric simplification, produces natural deductions which make use of lemmas and are often much more intuitive than the normal deductions which would be created by earlier algorithms.

Acknowledgements

First and foremost I would like to thank my advisor Peter Andrews for teaching me the subjects of logic and automated theorem proving, for his continued guidance of my research, and for providing financial support during my many years at Carnegie-Mellon University. Also, without him I would probably not have come to Pittsburgh and not have met my wife, Nancy, who I am also very grateful to. Her understanding when I had to work late and her moral support helped me through the mathematically rough times. Thanks also to my son Andreas and daughter Marina for making me laugh even if I didn't feel like it, and to Nancy's family who spent many an hour babysitting when I should have been at home. I am also indebted to Wilfried Sieg for his encouragement and enthusiasm in my work and to Richard Statman and Dana Scott for their time and their comments on earlier drafts. I would also like to thank Dale Miller for many discussions concerning the ideas in this dissertation. Finally, I thank Bill Scherlis, Gene Rollins and the rest of the ERGO group at Carnegie-Mellon University for providing the working and computing environment in which the later versions of this thesis were drafted.

Contents

Chapter 1

Introduction

1.1 Motivation and Applications

All the work in this thesis is motivated by the desire to integrate machine-oriented and human-oriented forms of deduction. Higher-order logic was chosen, since most concepts and theorems in mathematics can be expressed very naturally in type theory. It is my belief that this ease of expression will eventually lead to powerful and user-friendly tools assisting the student of mathematics and logic and, in the farther future, the mathematician in his research.

When one teaches logic it is all too obvious that to the student the form often obscures the content. Computer-assisted logic instruction can help alleviate this problem, since "form" is left to the computer and the student can concentrate on "content". This is idealized, of course, but experience suggests that students learn more with the aid of a computer to check their deductions. The immediate feedback the student receives when he tries to apply an inference illegally is invaluable.

Wouldn't it be nice if we could give sophisticated help to a student who does not know how to proceed, rather than just telling him if he did something illegal? Here the gap between machine-oriented and human-oriented forms of deductions becomes painfully apparent. Theorem proving is expensive — it costs time in which the student has to wait for advice. Moreover, the fact that a given line in the proof is in fact a theorem is not of much help to the student. Unfortunately, the most widely used theorem proving procedures use a representation far from a natural deduction, which I assume the student is to produce. Also, first-order logic is undecidable, and the theorem prover will thus not always be able to provide help. This problem is even greater in higher-order logic, where theorem proving procedures are rare and weaker in practice.

We offer at least a partial solution, which was first proposed by Andrews [2] and developed by Miller [23]. If we can find a suitable representation of the machine proof, we could use it as a plan which would guide us through the deduction. The representation we use in this thesis is a generalization of Miller's expansion tree proofs. Miller solves the problem of translating expansion proofs into a sequent calculus and into natural deductions, but he makes use of certain derived rules of inference which cannot easily be eliminated. We extend his work in

several directions. Since our notion of expansion proof is more general, we need to generalize his translation algorithm. This is done in Chapter 3. We also translate into a different logical system which is more convenient for our purposes. In Chapter 6 we present a new translation algorithm into a pure natural deduction system which does not make use of any derived rules of inference. For the purposes of proof guidance to students, this is most useful, since many of the derived rules may not be available to the student. Also, there is a certain beauty and simplicity to unembellished natural deduction.

This, like all previously mentioned translation algorithms, will produce only normal deductions. However, many desirable and intuitive proofs are not normal. At the end of Chaper 6 we present a significant improvement of the translation which introduces lemmas into the deduction in certain situations instead of resorting to the Rule of Indirect Proof. In practical examples this improvement turned out to be extremely valuable, producing much more intuitive deductions, and thus giving much better help to the student.

It should also be noted that these translation procedures are highly non-deterministic, that is, they leave a lot of choice between different possible inferences in a given partial deduction. This is important, since it means that help can be given in many situations, since many natural deductions will correspond to the same expansion proof.

So far, however, we have solved only half of the problem. How are we to get this machine-oriented proof? In first-order logic we can run a theorem prover and then convert the result into an expansion proof. This can be done for mating proofs (see Andrews et al. [3]) which are very closely related to expansion proofs, and, with a little more work, for resolution proofs (see Pfenning [26]). What do we do if the theorem prover is not powerful enough to prove the theorem, as will often be the case in a higher-order logic? The teacher could give one or more sample deductions which are then stored and compared against the situation in which the student has asked for help. The hope would be that the student's proof attempt will be sufficiently close to the sample deduction so that we can still give advice. This does not seem a very promising approach, since deductions vary not only in essentials, but also in many inessential details and it would be hard to determine whether we can use a certain step profitably. We explore an alternative solution, which is to convert the sample deduction into our abstract format, that is, into an expansion proof. As noted above, we will then be able to give advice from this expansion proof in many different situations. In Chapter 3 we give an algorithm which does this translation from a cut-free deduction into an expansion proof. A similar algorithm was presented by Miller [23], but produced highly redundant expansion proofs in many cases. We improved his algorithm so that the translation between deductions and expansion proofs is the inverse of the translation in the opposite direction. The extension of this algorithm to natural deductions is straightforward and not presented in this thesis. The translation from deductions using cut (or deductions which are not normal) requires a cut-elimination algorithm of some form, since expansion proofs are inherently cut-free, that is, have the strong subformula property. We give a cut-elimination algorithm for expansion proofs and show that it is partially correct. The proof of total correctness, that is, termination on all inputs, is only proven for the first-order fragment and remains a conjecture for the full higher-order system.

The ability to interpret a student's deduction as an expansion proof has additional benefits.

We can take a deduction, translate it into an expansion proof, then generate another deduction in the hope that it will be a cleaned-up version of his original deduction with essentially the same contents. In another application the translation may help the researcher in developing heuristics for the automatic proof of theorems through expansion proofs, since the expansion proof counterpart of a natural deduction can be easily viewed and analysed.

We also present two additional practically useful extensions. The deductive systems considered above are unsatisfactory in at least two ways. There is no primitive notion of equality, as it is often used in first-order logic, and the system is non-extensional. We define a new system with a primitive equality and show that all the translations mentioned above can be modified to go to and from this new logical system. Thus expansion proofs can provide advice for students writing proofs in a logical system with equality.

The second extension centers around the fact that in mathematics one usually assumes the axiom of extensionality. However, expansion proofs are non-extensional. We define a notion of extensional expansion proof and show that they are sound and complete with respect to a system of type theory with extensionality. These proofs are again given explicitly through translations which means all the applications listed above now apply to a logical system with extensionality.

1.2 Overview

In this thesis we will study two styles of inference systems for higher-order logic and the connections between them.

One of the systems, we call it $\mathcal{H}$, is very much like Gentzen's sequent calculus [11], but we go beyond that and consider natural deduction (Prawitz [28]). We call these *non-analytic*, since the rule of cut or modus ponens (in the sequent calculus) and maximal formulas (in natural deduction) is important in these systems, even though it may be eliminated from most of them.

Expansion proofs form the other style of logical calculus we investigate. They were introduced by Miller [23] and generalize Herbrand expansions to higher-order logic. Expansion proofs we call *analytic*, since they are inherently cut-free.

In Chapter 2 we define a language $\mathcal{L}$ and an inference system $\mathcal{H}$. $\mathcal{H}$ is based on the sequent calculus as refined by Tait [36], where negations of non-atomic formulas are considered to be defined rather than primitive. We generalize this system to type theory. Moreover, later applications make it necessary to base deductions on multisets of formulas instead of sets of formulas. The cut-elimination properties of $\mathcal{H}$ are different from Tait's and Gentzen's [11] systems and are investigated in this chapter. We give a very general notion of proper reduction sequence. The first-order fragment of $\mathcal{H}$ is important for educational applications discussed later, and the total correctness proof of a non-deterministic cut-elimination algorithm for first-order $\mathcal{H}$-deductions is given. However, for the full higher-order system the algorithm is proved only partially correct — the termination of the algorithm on all deductions remains a conjecture. It should be noted that the termination of cut-elimination algorithms for classical higher-order sequent and natural deduction systems is still an open problem of proof theory.

In Chapter 3 we introduce expansion proofs, which were first defined by Miller [23] and form a purely analytic inference system and are closely related to Herbrand expansions [16]. Our formulation is more general than Miller's in several respects; most importantly we allow arbitrary subnodes of the expansion tree to be mated, thereby mostly eliminating the need for *focusing* as defined by Miller [22]. Moreover, we banish negation from all but atomic formulas. Then we show the soundness and completeness of expansion proofs with respect to the system $\mathcal{H}$ by giving explicit translation in both directions. These translations are at once more general than Miller's by allowing more choices and more refined because they have the important property that they are inverses of each other. That is to say that an expansion proof, when translated into one of many possible $\mathcal{H}$-deductions and then mapped back into an expansion proof will yield the original expansion proof or a simpler one. This practically important property is achieved by an improvement of the author's merging algorithm for expansion proofs presented in [26], which itself was an improvement over Miller's original algorithm in [23]. We then give a cut-elimination algorithm based directly on expansion proofs, that is, we show how to construct an expansion proof for $A \supset C$, given one for $A \supset B$ and for $B \supset C$. It is shown partially correct, that is, if it terminates it yields an expansion proof, but unfortunately the termination proof again remains a conjecture. It is shown that in the first-order fragment every strong reduction sequence terminates.

In Chapter 4 we extend the language $\mathcal{L}$ and deduction system $\mathcal{H}$ to include a primitive notion of equality accompanied by a substitution rule and an axiom schema asserting reflexivity of equality. Since a (non-extensional) equality is definable in our formulation of type theory, every theorem in $\mathcal{H}^=$ has an expansion proof after we instantiate the definition of equality. This poses the question whether we can recover an $\mathcal{H}^=$-deduction of the original theorem (with primitive equality) from the expansion proof for the instantiated theorem. We answer this question affirmatively. Most of this chapter is devoted to developing and proving the correctness of the algorithm which constructs an $\mathcal{H}^=$-deduction from an expansion proof for the instaniated theorem. It is shown that cut-elimination does not hold in $\mathcal{H}^=$. We add a dual substitution rule (which substitutes the left-hand side for the right-hand side instead of vice versa) to obtain a system $\mathcal{H}^*$. We show that cut-elimination holds in the first-order fragment of $\mathcal{H}^*$ (it still does not hold for the full system) and improve our earlier translation procedure to produce more natural and elegant deductions in $\mathcal{H}^*$. As an aside we also show how to restrict the search for an expansion proof of a theorem containing equality. This is important since instantiating the definition of equality introduces a higher-order quantifier for which there is a potentially very large set of possible substitution terms. It is proven that if one allows only literals to instantiate such higher-order quantifiers, the expansion proof system remains complete. This is especially important if one is only interested in obtaining first-order proofs for first-order theorems containing equality. The translation from $\mathcal{H}^=$ deductions into expansion proofs is a straightforward extension of the algorithm presented in Chapter 3 and is not given in this thesis.

In Chapter 5 we consider extensionality. If one adds an extensionality rule to $\mathcal{H}$, the cut-elimination theorem still holds in the resulting system $\mathcal{H}^e$, as shown by Takahashi [37]. Since an extensional equality is not definable in our type theory, we now need to define a new notion of expansion proof. We define extensional expansion proofs which have an additional type of node, called extensionality nodes. We give translations between cut-free $\mathcal{H}^e$-deductions and

extensional expansion proofs and prove them correct. This establishes soundness and completeness of extensional expansion proofs. In the final section of this chapter we indicate how to further improve the translation of extensional expansion proofs. As in the case of equality, one can restrict the number of situations in which to apply extensionality when searching for an extensional expansion proof. This observation is based on a theorem given at the end of this chapter.

In Chapter 6 we consider a very pure system $\mathcal{N}$ of natural deduction and show that expansion proofs contain the necessary information to guide the building of a deduction in $\mathcal{N}$. The main organizational tool for specifying and implementing the translation from expansion proofs into natural deductions is a generalization of tactics as introduced by Gordon et al. [14]. Our tactics are more general, since they can make use of the information in the expansion proof and thus do not have to perform any real search. The correctness and completeness of the given set of tactics is proven. This proof makes heavy use of well-founded relations on expansion proofs defined in Chapter 3. We then address the question how to find more elegant deductions than the translations presented so far give us. We present an algorithm called *symmetric simplification* and three tactics which make use of this algorithm to introduce applications of the Proof by Cases into the deduction. Since the formulas which define the cases are not necessarily subformulas of the theorem, the resulting deduction will not in general be normal. These three tactics seem to be very useful in practice since they help to avoid unintuitive uses of the Rule of Indirect Proof.

1.3 Historical Perspective

1.3.1 Gentzen's Hauptsatz

Ever since Gentzen's fundamental paper [11] there existed a dichotomy in proof theory between natural deduction and sequent-like proof systems. This dichotomy in proof theory reflects the division between classical and intuitionistic logic. All the proof-theoretical work done subsequently shows that natural deduction is the "natural" system to express intuitionistic reasoning, while less adequate for classical proofs. On the other hand sequent-like proof systems seem to be the most appropriate vehicle for proof-theoretic investigations of classical logic.

We will look at some results and methods used in the investigation of the proof-theoretical properties of classical and intuitionistic logic. It will become apparent that certain research programs have never been completed. Most glaringly, no meaningful cut-elimination algorithm for classical type theory has been proven correct. We will look at this problem from two differents points of view: in a sequent-like calculus in the tradition of Gentzen and in a calculus of expansion proofs in the spirit of Herbrand [16].

Gentzen [11] started by introducing his natural deduction systems NJ and NK for intuitionistic and classical logic, respectively. He then asserts that for the proof of his Hauptsatz these systems were inadequate for technical reasons, and proceeds to define the sequent calculus (systems LJ and LK). In hindsight it becomes clear that NJ was actually very well suited for a proof of the Hauptsatz for intutionistic logic — it was his goal to give a uniform treatment to classical

and intuitionistic logic which forced him to invent the sequent calculus.

Gentzen shows the equivalence of his calculi, in the sense that one can deduce the same theorems. He does not give a strong correspondence (in the sense of Zucker [41]) between the systems which would relate the meaning of his Hauptsatz for natural deductions and sequent calculus proofs.

In subsequent work, we have to distinguish further proof-theoretic investigations from model-theoretic extensions of the Hauptsatz. The former have "constructive" or "algorithmic" character, while the latter are proved by semantic means which do not have meaningful computational consequences.

1.3.2 Constructive Extensions of the Hauptsatz

Let us first follow the tracks of proof-theory. It strikes one immediately that most of this work was done in an intuitionistic setting until very recently. These investigations focused on natural deduction proofs. The concepts of normalization and strong normalization were introduced and normalization theorems were proved for first-order natural deduction by Prawitz [28]. The concept of *strong normalization* would have been entirely new to Gentzen, but the Normalization Theorem may be viewed as the analogue of the Hauptsatz for the sequent calculus in natural deduction.

The extension of the Normalization Theorem to second-order logic is inherently hard, because normalization does directly imply consistency of second-order number theory (see Statman [35]). An idea of Girard [13] finally brought the breakthrough and the Normalization Theorem was proved for second-order (intuitionistic) logic by Martin-Löf [21]. At the same time, Prawitz [27] proved the corresponding Strong Normalization Theorem. It was generally accepted that this would easily generalize to full type intuitionistic type theory, and Martin-Löf [20] provided a proof of Strong Normalization for full intuitionistic type theory. Since then these methods have been extended to prove strong normalization for even more powerful intuitionistic type theories, such as Martin-Löf's [19] or Coquand and Huet's Calculus of Constructions [9].

Another very natural and important question was answered in very much detail by Zucker [41]: What is the relation between cut-elimination in sequent-style systems and normalization in natural deduction. Zucker actually provides more: he defines a notion of strong cut-elimination and shows that it corresponds to strong normalization. He stays entirely in the bounds of intuitionistic logic. An attempt by the author to carry over his analysis to classical logic failed. None of the known interpretations of classical in intuitionistic logic have the property that cut-conversions correspond to reductions of natural deductions. Zucker pointed out some of the difficulties in trying to extend the correspondence even in the intuitionistic case with more liberal conversion rules. Part of the problem seems to be that we cannot allow a very natural reduction rule which interchanges cuts and therefore cannot truly claim to have a "strong" normalization result. Intuitively it seems reasonable that two deductions present the same proof, if they only differ in the order in which two lemmas are applied. But then not every sequence of mix reductions will terminate and we have to place some technical restriction

on the application of conversion rules to save the normalization theorem. Therefore we follow Gentzen's example and do not consider a conversion which allows interchange of cuts.

Little work has been done in extending normalization or cut-elimination to higher-order *classical* logics with proof-theoretic means. Statman [35], [34] and Prawitz [29] arrive at different versions of what it means for a classical natural deduction proof to be normal. They prove (strong) normalization for their respective notions of normality in a system of second order classical natural deduction.

1.3.3 Non-Constructive Extensions of the Hauptsatz

A wholly different approach to the questions of normalizability and cut-free proofs may be characterized as "non-constructive" or "model-theoretic". Here, cut-elimination theorems are proved via semantic methods. The disadvantage of these proofs is that they do not provide a meaningful algorithm which transforms a proof with cuts or maximal formulae into a cut-free or normal proof. Tait in [36] extended Gentzen's work to second-order logic in this fashion. Later, also by a semantic argument, Takahashi [38] proved what was known as Takeuti's Conjecture [39]: The cut-elimination theorem for simple type theory.

From here it was a small step to include extensionality. A cut-elimination theorem for type theory with extensionality was announced by Takahashi in [37]. One proof can be found in [40]. A completeness result for a system also including a description operator can be found in Andrews [1].

1.4 Open Problems

The non-constructive results for classical logic are very valuable in their own right, but unsatisfactory for an automated reasoning program, since no practical algorithm for eliminating a cut from a given derivation follows.

In this thesis we propose an algorithm for eliminating cuts from a sequent-like proof system. Unfortunately we were only able to show that it terminates when applied to first-order proofs.

One natural attempt would be to interpret the classical system in the intuitionistic one. This was carried out for a minimal second-order natural deduction system by Statman [34]. However, for none of the known interpretations do cut-elimination steps commute with the interpretation. In particular, there are cut-reductions in the sequent calculus such that the deduction corresponding to the reduced form is incomparable (with respect to normalization) to the deduction before the reduction. However, this is not to say that there could be no such commuting interpretation.

A direct method of proof seems more promising, though of course in absence of a proof this is only idle speculation. Clearly a simple syntactic argument as for the first-order case must fail because of Gödel's second incompleteness theorem. However, this paradox was avoided by the methods used to obtain the normalization results for intuitionistic second order logic

(Martin-Löf [21], Prawitz [27]) and type theory (Girard [13], Martin-Löf [20]). One may try to reinterpret Martin-Löf's *computability predicates* or Prawitz's *regular sets* in the framework of Henkin models [15] for type theory. Girard [12] and later Martin-Löf [18] realized that the computability predicate constructions could serve as models for intuitionistic type theory. With this point of view in mind we would like to build a structure $\mathbf{D} = \{\mathbf{D}^\alpha\}_{\alpha\in\mathcal{T}}$. $\mathbf{D}$ itself would not be a frame in the sense of Henkin [15] or Andrews [1], but is what could be called a "universal deduction model". Given any consistent set of sentences, we can take a kind of "quotient" structure which is a submodel of a Boolean-valued model for type theory (in the sense of Scott [31]; see [6] for an exposition of Boolean-valued models in set theory). From these we could build frames with two truth values in the domain $\mathbf{D}^o$ as shown in [6]. Such a deduction model would differ from term models ordinarily constructed for completeness proofs or semantic cut-elimination proofs in that deduction models are made up from sets of deductions, rather than from terms of some extended language.

Another desirable constructive extension would be to include the axiom schemata of extensionality and choice. A constructive proof of cut-elimination for type theory with extensionality is still an open research question, though it has been proven by model-theoretic methods by Takahashi [37].

Another major open question as far as mathematical reasoning in type theory is concerned is the axiom schema of choice. Kreisel argued in [17] that no satisfactory notion of cut-freeness could result in a cut-free system which includes the axiom of choice. In first-order logic it is immediately clear what we mean by a cut-free proof and it is often described as a proof with the *subformula property*. This notion must be modified for higher-order logics, since every formula is a subformula of any quantified formula, since we have to allow instantiations. Takeuti noted that the subformula property becomes meaningless, for example, when one is willing to use $\forall A[A\rightarrow A]$ as an assumption. This uncertainty of what cut-elimination really means can be resolved. We can give a kind of "operational" definition in the framework of an automated reasoning system. We say a proof system is *analytic* if we only need to find formulas to instantiate the quantifiers, i.e. we never need to look for lemmas when trying to prove a theorem (that we would *want* to look for or make use of lemmas when actually proving a theorem is a different matter). Corroborating our contention that this is the "right" notion of cut-freeness for type theory is the fact that, when it is restricted to first order logic, we obtain the usual definition via the subformula property.

This definition does not satisfy all of Kreisel's criterion for a satisfactory notion of cut-freeness and his theorem does not apply. Indeed, given the completeness of type theory with extensionality and choice (see Henkin [15]) and also for extensionality and description proven by Andrews [1], it seems quite possible that there is a cut-free formulation for type theory with the axiom schemata of extensionality and choice or description.

Arithmetic is another vital element of mathematical reasoning and we certainly want to be able to add Peano's axioms or some axiom of infinity. Here the expressive power of type theory helps: these principles may be formulated as single axioms and not axiom schemata and do not affect the cut-elimination properties of the deductive system.

Chapter 2

The System $\mathcal{H}$

2.1 The Language $\mathcal{L}$

The complexity of some of the syntactic proofs forced us to go a somewhat unusual route in the definition of language and derivation. We first define the language proper, with primitives $\sim$, $\wedge$, $\vee$, $\forall$, $\exists$, λ. We then define notions of λ-conversion, negation conversions, and $\lambda^{\sim}$-normal form.

The derivations in $\mathcal{H}$ do not contain formulas in their lines, but rather equivalence classes of formulas modulo $\lambda^{\sim}$-conversion. This is very similar to considering formulas identical when they only differ in the names of their bound variables, as is often done. When giving definitions or proofs by induction on the structure of equivalence classes of formulas, we use the unique (up to $\alpha\beta$-conversion) representative of the equivalence class which is in $\lambda^{\sim}$-normal form.

This technical device is a tremendous help in formulating the logical system in a concise way. Below we will usually talk about "formulas" when we really mean "equivalence classes of formulas". Whenever it is important to point out the distinction, we will do so.

A deduction thus conceived and ending in a class of formulas could be considered to derive any formula in that class. Alternatively, we could make up new rules which allow $\lambda^{\sim}$-conversions as inference rules, but they would clutter up the presentation and add nothing essential.

There are more economical definitions of the similar languages of type theory put forth by Church [8] which are appropriate for a more axiomatic treatment. When considering sequent-like deductions, and also for expansion proofs, it is convenient to have all the usual quantifiers and connectives available from the start. Somehow the connectives given above seem to form a very natural basis for classical logic — for intuitionistic type theory the fragment based on $\forall$ and $\supset$ seems to be the most useful complete set of quantifiers and connectives. Compare, for example, Martin-Löf [20], or Prawitz [28].

Definition 1 We inductively define $\mathcal{T}$, the *set of type symbols* by

1. $\iota \in \mathcal{T}$. ι is the type of individuals. We could allow an arbitrary set of ground types.

2. $o \in \mathcal{T}$. o is the type of truth values.

3. $\alpha\beta \in \mathcal{T}$ if $\alpha \in \mathcal{T}$ and $\beta \in \mathcal{T}$. $\alpha\beta$ stands for the type of functions from elements of type β to elements of α.

We will attach type symbols as subscripts to variables, parameters, and constants in the language and to meta-variables for arbitrary formulas. In the language $\mathcal{L}$, variables will always be bound variables. We use parameters for free variables, thereby distinguishing free and bound variables syntactically.

Definition 2 A formula in the language $\mathcal{L}$ is defined inductively by

1. A variable x_α is a formula of type α.

2. A parameter a_α is a formula of type α.

3. A constant k_α is a formula of type α.

4. $[B_{\alpha\beta}C_\beta]$ is a formula of type α for any formulas $B_{\alpha\beta}$ and C_β.

5. $[\lambda x_\beta A_\alpha]$ is a formula of type $\alpha\beta$ for a variable x_β and formula A_α.

6. $\sim A_o$ is a formula of type o for A_o a formula of type o.

7. $A_o \wedge B_o$ is a formula of type o for formulas A_o and B_o.

8. $A_o \vee B_o$ is a formula of type o for formulas A_o and B_o.

9. $\forall x_\alpha A_o$ is a formula of type o for a variable x_α and formula A_o.

10. $\exists x_\alpha A_o$ is a formula of type o for a variable x_α and formula A_o.

We assume that any formula we mention is well-typed.

Definition 3 We introduce some common abbreviations. $\underset{\text{def}}{=}$ is our symbol for definition in the logical system.

1. $A \supset B \underset{\text{def}}{=} \sim A \vee B$.

2. $A \equiv B \underset{\text{def}}{=} [A \supset B] \wedge [B \supset A]$.

In Chapter 4 we will also consider a different language where equality is primitive. Here let us merely note that a (non-extensional) equality is definable in $\mathcal{H}$.

Definition 4 (*Substitution*) Substitution for a parameter or variable is defined as usual, taking care to rename bound variables to avoid name clashes. We think of a substitution as a mapping from formulas to formulas which is completely specified by the images of the parameters. Our notation for the substitution θ which maps a_1 to t_1, a_2 to t_2, etc. will be

$$\theta = [a_1 \mapsto t_1, a_2 \mapsto t_2, \ldots]$$

We also apply the convention that a parameter not explicitly mentioned in the list of pairs defining θ will be mapped to itself. We write $\theta + [a \mapsto t]$ for the substitution which maps a to t and behaves like θ on all other parameters. We write θA for the result of applying the substitution θ to A.

We assume that all substitution are type-consistent, that is, the type of the parameter or variable and the type of its substitution term are identical.

We also use an implicit notation when we substitute in a context where a variable appears bound. For any binder Q (we will use it for $\forall$, $\exists$, and λ), and formula A (of correct type), we write $QxA(x)$ to indicate that x may (but need not) occur free in A. The result of substituting t for *all* free occurrences of x in A is then written as $A(t)$. If a formula depends on multiple bound variables, they are separated by commas, as in $\forall x \exists y A(x, y)$.

Definition 5 (*Lambda Conversion*) We define *β-reduction* through substitution at the *β-redex* $[\lambda x.A]B$

$$[\lambda x.A]B \underset{\beta}{\longrightarrow} [x \mapsto B]A$$

In our shortened notation, this rule may also be formulated as

$$[\lambda x.A(x)]B \underset{\beta}{\longrightarrow} A(B)$$

We say two formulas A and B are equal up to λ-conversion, if there is a sequence of β-reductions, inverses of β-reductions, and renaming of bound variables in A which yields B. We write $A \underset{\lambda}{=} B$. We say that A is in λ-normal form if there is no B such that $A \underset{\beta}{\longrightarrow} B$.

We state that our typed calculus has the strong Church-Rosser property. For a proof, see Barendregt [4], for example.

Lemma 6 Every formula A in $\mathcal{L}$ has a λ-normal form B such that $A \underset{\lambda}{=} B$. Moreover, this normal form is unique up to renaming of bound variables.

Definition 7 (*Negation Conversion*) We define $\sim$-conversion for formulas in λ-normal form. These are the usual rules for converting a formula into negation normal form.

1. $\sim\sim A \underset{\sim}{\longrightarrow} A$
2. $\sim[A \vee B] \underset{\sim}{\longrightarrow} [\sim A \wedge \sim B]$
3. $\sim[A \wedge B] \underset{\sim}{\longrightarrow} [\sim A \vee \sim B]$
4. $\sim[\forall x A] \underset{\sim}{\longrightarrow} [\exists x \sim A]$
5. $\sim[\exists x A] \underset{\sim}{\longrightarrow} [\forall x \sim B]$

We say two formulas A and B in λ-normal form are equal up to $\sim$-conversion if there is a sequence of $\sim$-reductions, inverses of $\sim$-reductions of subformulas of A which yields a A' such that $A' \underset{\lambda}{=} B$. We say that A is in $\lambda^{\sim}$-normal form (or *negation normal form*) if A is in λ-normal and $\sim$-normal form (that is no further $\sim$-reduction can be applied).

We write $A \underset{\lambda^{\sim}}{=} B$ for formulas which are equal up to λ and negation conversions.

The following extension of the strong Church-Rosser property to include negation conversions is well known.

Lemma 8 Every formula A in $\mathcal{L}$ has a $\lambda^{\sim}$-normal form B such that $A \underset{\lambda^{\sim}}{=} B$. Moreover, this normal form is unique up to renaming of bound variables.

Definition 9 A is an *atom*, if its $\lambda^{\sim}$-normal form is **not** of the form $\sim B$, $B \wedge C$, $\forall x B$, or $\exists x B$. A wff A of type o is a *literal*, if $A \underset{\lambda^{\sim}}{=} B$ or $A \underset{\lambda^{\sim}}{=} \sim B$, for an atom B.

Remark 10 (*Typographic Conventions*) We will use

1. $A, B, C, \ldots$, for arbitrary (equivalence classes of) formulas.
2. x, y, z for variables.
3. a, b, c for parameters.
4. U, V, W for multisets of (equivalence classes of) formulas.
5. $\mathcal{D}, \mathcal{E}, \mathcal{F}, \ldots$ for arbitrary deductions.

2.2 Inference Rules in $\mathcal{H}$

The system $\mathcal{H}$ is similar to the sequent calculus of Gentzen [11] as modified by Tait [36]. Another similar formulation can be found in Schütte [30]. However, we cannot consider lines in the deduction as sets of formulas. If one does this, an important and non-trivial inference rule, namely contraction, is made implicit. However, the order of the formulas on the line is unimportant, so assertions of lines in the deduction will be multi-sets of formulas.

Moreover, we will really have equivalence classes of formulas, rather than the formulas themselves, as members of the lines for reasons outlined at the beginning of this chapter.

Strictly speaking, we have to index inferences with the occurrences of a formula in the premiss of the inference. Sometimes we will make this explicit, but most of the time it can be inferred from the context. If several distinct occurrences of a given formula must be considered in the same rule, we will use numerical superscripts to distinguish them.

The inference rules can be divided into *structural rules*, *propositional rules*, *quantificational rules*, and later *extensionality* and *choice rules*. We will consider separately the rule of *Cut*. Together with the rules, we also define the notion of *active*, *passive*, and *contracted* formula occurrence for a given inference.

Definition 11 (*System* $\mathcal{H}$) There is only one structural rule in $\mathcal{H}$ namely *contraction* (C). There is one propositional rule for each propositional connective: *initial deductions* for negation, $\wedge$-*introduction* ($\wedge I$) and $\vee$-*introduction* ($\vee I$). There is also exactly one rule for the quantifiers: $\exists$-*introduction* ($\exists I$) and $\forall$-*introduction* ($\forall I$).

In all rules, any formula $C \in U$ is passive.

1. Initial deductions

$$\frac{}{U, A, \sim A}\, I : A, \sim A$$

for any formula A of type o and multiset U. A and $\sim A$ are active, any formula in U is passive. In order to distinguish an active A from a passive A, and also resolve ambiguities in initial deductions like $B, \sim B, A, \sim A$, the occurrences of A and $\sim A$ which are considered active are provided explicitly. The arguments of an inference are always occurrences of formulas, not just formulas. Usually, it will be clear from the context, which occurrences are considered active.

2. Contraction

$$\frac{U, A^1, A^2}{U, A^{12}}\, C : A^1, A^2$$

Here any formula occurrence in U is passive and A^{12} is contracted. There is no active formula occurrence in the conclusion of a contraction. Whenever it will be necessary

to distinguish different occurrences of a given formula, we will attach superscripts to the occurrences. We will adopt the convention of giving the name A^{ij} to the result of contracting of A^i and A^j.

3. Propositional rules

$$\frac{U, A, B}{U, A \vee B} \vee I : A, B$$

$A \vee B$ is active.

$$\frac{U, A \qquad V, B}{U, V, A \wedge B} \wedge I : A, B$$

$A \wedge B$ is active.

4. Quantificational rules

$$\frac{U, A(B)}{U, \exists x A(x)} \exists I : \exists x A$$

Here $\exists x A(x)$ is active.

$$\frac{U, A(a)}{U, \forall x A(x)} \forall I : \forall x A$$

where a is a parameter not free in U or $\forall x A(x)$. $\forall x A(x)$ is active.

5. Cut

$$\frac{U, A \qquad \sim A, V}{U, V} Cut : A, \sim A$$

Here any $C \in U$ and $C \in V$ are passive, but there is no active or contracted occurrence in the conclusion. A and $\sim A$ are called the *cut formulas*. Note that this is not Gentzen's rule of mix, since we distinguish occurrences and only one occurrence of A and one occurrence of $\sim A$ are eliminated by the cut.

Definition 12 The system $\mathcal{H}^-$ is like $\mathcal{H}$, but without the rule of cut.

2.3 Basic Properties of Deductions in $\mathcal{H}$

Definition 13 We introduce some notation.

1. We define $\texttt{Final}(\mathcal{D})$ as the multiset in the final line of $\mathcal{D}$.
2. We write $\begin{matrix}\mathcal{D}\\ U\end{matrix}$ for a deduction $\mathcal{D}$ with final line U.
3. We write $\dfrac{U'}{\overline{U}}\, r$ for several successive applications of the rule r, usually contraction.
4. The deductions ending in the premisses of the final inference of a deduction $\mathcal{D}$ are the *immediate subderivations* of $\mathcal{D}$. We write $\mathcal{D}' \lhd \mathcal{D}$ when $\mathcal{D}'$ is an immediate subderivation of $\mathcal{D}$. Note that a deduction ending in cut or $\wedge I$, has two such immediate subderivations. We define inductively that a *subderivation* of $\mathcal{D}$ is either an immediate subderivation of $\mathcal{D}$, or a subderivation of one of the immediate subderivations of $\mathcal{D}$.
5. Given $\mathcal{D} = \begin{matrix}\mathcal{D}\\ U, A\end{matrix}$ and $\mathcal{E} = \begin{matrix}\mathcal{E}\\ \sim A, V\end{matrix}$, we write $\texttt{Cut}(\mathcal{D}, \mathcal{E}, A, \sim A)$ for the deduction

$$\dfrac{\begin{matrix}\mathcal{D}\\ U, A\end{matrix} \qquad \begin{matrix}\mathcal{E}\\ \sim A, V\end{matrix}}{U, V}\, Cut$$

Lemma 14 (*Weakening*) Given a deduction $\mathcal{D}$ of U, then there is a deduction $\mathcal{D}$ of equal length for U, A for any formula A.

Proof: We define a set of deductions $\mathcal{D} \oplus A$ by adjoining A to every step in $\mathcal{D}$. Each deduction in $\mathcal{D} \oplus A$ ends in U, A. The definition is inductive on the length of $\mathcal{D}$.

1. $\mathcal{D}$ is initial. Then $\mathcal{D} \oplus A$ is the singleton set containing the deduction U, A which is again initial with A passive.
2. $\mathcal{D}$ ends a one-premiss inference r ($\vee I$, $\forall I$, $\exists I$). Then for $\mathcal{D}' \lhd \mathcal{D}$ and for any deduction $\mathcal{E}' \in \mathcal{D}' \oplus A$, $E = \dfrac{\begin{matrix}\mathcal{E}'\\ U', A\end{matrix}}{U, A}\, r$ is a deduction in $\mathcal{D} \oplus A$ (one may need to rename parameters).
3. $\mathcal{D}$ ends in a two-premiss inference rule r ($\wedge I$, Cut) with left and right premiss $\mathcal{D}'$ and $\mathcal{D}''$, respectively. Then for any $\mathcal{E}' \in \mathcal{D}' \oplus A$, $E = \dfrac{\begin{matrix}\mathcal{E}'\\ U', A\end{matrix} \qquad \begin{matrix}\mathcal{D}''\\ U''\end{matrix}}{U, A}\, r$ is in $\mathcal{D} \oplus A$. Also for any $\mathcal{E}'' \in \mathcal{D}'' \oplus A$, $\mathcal{E} = \dfrac{\begin{matrix}\mathcal{D}'\\ U'\end{matrix} \qquad \begin{matrix}\mathcal{E}''\\ U'', A\end{matrix}}{U, A}\, r$ is in $\mathcal{D} \oplus A$.

Note that the new formula A is passive in all inferences of any deduction in $\mathcal{D} \oplus A$. We shall extend the notation in the usual way to allow multisets of formulas to be adjoined to a deduction, as in $\mathcal{D} \oplus U$. We shall also write $\mathcal{D} \oplus A$ in place of a deduction when we mean some deduction in the set $\mathcal{D} \oplus A$.

□

Definition 15 (*Substitution into a deduction*) If θ is a substitution, we can extend it from formulas to deductions, by applying it to each line in the deduction. We may have to rename some bound variables or names for parameters to avoid name clashes in the usual way. We write $\theta\mathcal{D}$ for the result of applying θ to $\mathcal{D}$. Sometimes we also write $\mathcal{D}(a)$ and then $\mathcal{D}(B)$ for the result of substituting B for *every* free occurrence of a in $\mathcal{D}$.

2.4 Cut Conversions in $\mathcal{H}$

Our cut reductions are based on similar algorithms of Gentzen [11] and Smullyan [33], but formulated in the system $\mathcal{H}$. Note that compared to Gentzen's original formulation we have abandoned the structural rule of interchange. This is a consequence of considering the lines of the proof as multisets, rather than sequences of formulas.

We also do not need the otherwise customary rules of λ-conversion, since the "formulas" in the deductions are really equivalence classes of formulas modulo $\lambda^{\sim}$-conversion. This is possible because of the strong Church-Rosser property of the typed λ-calculus as well as the existence of negation normal forms. In first-order logic, negation is often taken as defined for everything but atomic formulas, but this approach is impossible here since a formula $\sim x_o$ may become non-atomic after substituting for x_o.

But we also depart from the usual presentation in a different respect. When one considers cut rather than mix, one immediately is faced with the problem noted by Zucker [41], namely that the natural set of conversion rules has non-terminating reduction sequences. This happens already in the first-order case. Therefore one can only expect to prove a weak normalization theorem stating that there is a sequence of reductions ending in a normal proof. We will follow Zucker's terminolgy and define proper sequences of reductions. We can then show in the first-order fragment that every proper sequence of reductions will terminate. Still, this is much weaker than the corresponding result for the $\rightarrow\forall$ fragment of intuitionistic natural deduction which states that all reductions sequences will terminate in the *same* normal deduction.

In the first-order fragment of our type theory, the conversion rules below have exactly the same description, and we can prove cut-elimination by a simple double induction.

In type theory this argument fails, since in case 4 in Definition 21 below, the complexity of the cut formula after the reduction may actually increase.

As a matter of fact, the function which would give a bound on the size of a proof without cuts, given one with cuts, cannot be provably recursive in type theory. A discussion of this can be found in Statman [35]. In second-order logic, the termination of a different algorithm in a different inference system (natural deduction) has been shown by Statman [34] and, again with a different notion of "normal", by Prawitz [29].

One natural attempt would be to interpret our Tait-style system in one of the systems used by Statman or Prawitz. However, the known interpretations do not commute with reductions, so that this method seems difficult to apply. Zucker noted this, by pointing out the limitations of his interpretation of natural deductions in sequents for the intuitionstic logic. Also, for the natural deduction systems we have strong normalization, here we can only expect normalization, as some counterexamples show. Part of the problem also seems to be that natural deduction is very much less "natural" for a classical logic, than for an intuitionistic system. As the discrepancy between the notion of "normal" for Statman and Prawitz shows, it is not exactly clear what this notion should be. We cannot answer this for the natural deduction system, since our attempts at finding an interpretation which commutes with reductions has failed.

First we define a relation which we need to consider when defining which reductions apply

when a cut formula is contracted.

Definition 16 We call a formula occurrence A^i in the premiss of an inference rule a *predecessor* of an occurrence A^j in the conclusion if A^i is passive and $i = j$ or A^i and some other A^k are contracted to A^j. The *ancestor* relation is the reflexive and transitive closure of the predecessor relation.

Definition 17 We define a relation $\ll^{\mathcal{D}}$ on occurrences of identical formulas on the final line in a deduction $\mathcal{D}$. $\ll$ is defined only when no occurrence for the formula is passive in a cut in $\mathcal{D}$.

1. All occurrences of A are passive in the last inference of $\mathcal{D}$. Then $A^i \ll^{\mathcal{D}} A^j$ iff for some premiss $\mathcal{D}' \lhd \mathcal{D}$, $A^i \ll^{\mathcal{D}'} A^j$. This means one of the two cases below must apply.
   1.1. The last inference is $\wedge I$. Then $A^i \ll^{\mathcal{D}} A^j$ iff A^i and A^j come from the same premiss $\mathcal{D}'$ and $A^i \ll^{\mathcal{D}'} A^j$.
   1.2. The last inference is not $\wedge I$. Note that it cannot be a cut, since we assumed that no occurrence of A may be passive in a cut inference. Then $A^i \ll^{\mathcal{D}} A^j$ iff $A^i \ll^{\mathcal{D}'} A^j$ for $\mathcal{D}' \lhd \mathcal{D}$.
2. One occurrence of A is active. Then $A^i \ll^{\mathcal{D}} A^j$ iff
   2.1. A^i is active in the last inference, or
   2.2. $A^i \ll^{\mathcal{D}'} A^j$ for some $\mathcal{D}' \lhd \mathcal{D}$.
3. One occurrence of A is contracted in the last inference. This is the critical case. Then $A^i \ll^{\mathcal{D}} A^j$ iff there are predecessors $A^{i'}$ of A^i and $A^{j'}$ of A^j such that $A^{i'} \ll^{\mathcal{D}'} A^{j'}$.

Definition 18 We say a deduction $\mathcal{D}$ ending in U is *contraction normal* if $\ll^{\mathcal{D}}$ is acyclic on every set of identical formulas in U.

Informally, a deduction is contraction normal if occurrences of a formula which were introduced earlier (higher up in the tree), are also contracted earlier. Care is taken to make the definition as general as possible. For example, occurrences of formulas introduced in disjoint branches of the proof tree are incomparable (with respect to $\ll$) and therefore we assume nothing about the order in which they are contracted when we try to decide if a deduction is contraction normal.

The property that a deduction is contraction normal is crucial, since we would like our reduction rules to be local in nature. Jumping ahead a little bit: when one needs to propagate a cut where one of the cut-formulas was contracted, we will get two new cuts. It is necessary to ensure termination that the occurrence which was introduced later is also cut later.

Example 19 The following deduction $\mathcal{F}$ is *not* contraction normal, since $[\exists x A(x)]^{123} \ll^{\mathcal{F}} [\exists x A(x)]^{123}$. That can be seen since in the immediate subderivation $\mathcal{F}'$, $[\exists x A(x)]^{1} \ll^{\mathcal{F}'} [\exists x A(x)]^{23} \ll^{\mathcal{F}'} [\exists x A(x)]^{1}$, which in turn is true because in the next subderivation $\mathcal{F}''$, $[\exists x A(x)]^{2} \ll^{\mathcal{F}''} [\exists x A(x)]^{1}$ and $[\exists x A(x)]^{1} \ll^{\mathcal{F}''} [\exists x A(x)]^{3}$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overset{\mathcal{D}}{A(t), A(s), A(r)}}{A(t), A(s), [\exists x A(x)]^{3}}\exists \mathrm{I}}{[\exists x A(x)]^{1}, A(s), [\exists x A(x)]^{3}}\exists \mathrm{I}}{[\exists x A(x)]^{1}, [\exists x A(x)]^{2}, [\exists x A(x)]^{3}}\exists \mathrm{I}}{[\exists x A(x)]^{1}, [\exists x A(x)]^{23}}C^{2,3}}{[\exists x A(x)]^{123}}C^{1,23}$$

This can easily be converted into a contraction normal deduction, which looks almost the same, except for the superscripts of the contractions. The proof of Lemma 30 will give the general algorithm for converting deductions into contraction normal form by reindexing contractions.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overset{\mathcal{D}}{A(t), A(s), A(r)}}{A(t), A(s), [\exists x A(x)]^{3}}\exists \mathrm{I}}{[\exists x A(x)]^{1}, A(s), [\exists x A(x)]^{3}}\exists \mathrm{I}}{[\exists x A(x)]^{1}, [\exists x A(x)]^{2}, [\exists x A(x)]^{3}}\exists \mathrm{I}}{[\exists x A(x)]^{2}, [\exists x A(x)]^{13}}C^{1,3}}{[\exists x A(x)]^{213}}C^{2,13}$$

Lemma 20 (*Contraction Normal Form*) Given a deduction $\mathcal{D}$, we can obtain a contraction-normal deduction $\mathcal{E}$ by reindexing contractions.

Proof: by structural induction on $\mathcal{D}$.

We give the proof assuming that there is only one set of formula occurrences for which $\ll^{\mathcal{D}}$ is cyclic. It can easily be generalized.

Assume we are given $\overset{\mathcal{D}}{U, A^1, \ldots, A^n, A^{n+1}, \ldots, U}$ such that $A^1 \ll^{\mathcal{D}} A^2 \ll^{\mathcal{D}} \cdots \ll^{\mathcal{D}} A^n \ll^{\mathcal{D}} A^1$.

We distinguish cases depending on the last inference in $\mathcal{D}$.

1. All A^i are passive in the last inference in $\mathcal{D}$.
2. One A^i is active in $\mathcal{D}$. Then $A^i \ll^{\mathcal{D}} A^j$ for all $j \neq i$. Hence $i \geq n+1$. By the induction hypothesis we can reindex contractions in $\mathcal{D}'$ to get $\mathcal{E}'$ such that $\mathcal{E}'$ is contraction normal. Let $\mathcal{E}$ be $\mathcal{E}'$ followed by the last inference in $\mathcal{D}$. Then $\mathcal{E}$ is contraction normal and is obtained from $\mathcal{D}$ by reindexing contractions.

3. One A^i is contracted. This is the critical case. By induction hypothesis we can reindex contractions in $\mathcal{D}'$ to obtain $\mathcal{E}'$ such that $\ll^{\mathcal{E}'}$ is acyclic (on A). We now have to find two occurrences A^k and A^l such that contracting them will result in an $\mathcal{E}$ such that $\ll^{\mathcal{E}}$ is acyclic (on A).

   Claim: For any choice of A^k and A^l such that there is no m such that $A^k \ll^{\mathcal{E}'} A^m \ll^{\mathcal{E}'} A^l$, $\ll^{\mathcal{E}}$ is acyclic on A for $E = \dfrac{\overset{\mathcal{E}'}{A^0, A^1, \ldots, A^k, \ldots, A^l \ldots, U}}{A^0, A^1, \ldots, A^{kl}, \ldots, A^{l-1}, A^{l+1}, \ldots, U} C^{l,k}$.

   Proof of claim: Note that away from A^k and A^l, $\ll^{\mathcal{E}}$ agrees with $\ll^{\mathcal{E}'}$. Thus there could only be a cycle in $\ll^{\mathcal{E}}$ if for some A^m, $A^{kl} \ll^{\mathcal{E}} A^m \ll^{\mathcal{E}} A^{kl}$. Because $\ll^{\mathcal{E}'}$ does not have a cycle, this can only happen if $A^k \ll^{\mathcal{E}'} A^m \ll^{\mathcal{E}'} A^l$ or $A^l \ll^{\mathcal{E}'} A^m \ll^{\mathcal{E}'} A^k$. But the existence of such an m was explicitly excluded in the assumption of the claim

   It remains to show that we can always find such A^k and A^l. But this is easy: take, for instance, A^k minimal with respect to $\ll^{\mathcal{E}'}$ and A^l immediately above A^k.

□

Now we have completed all the necessary definitions to describe the cut reduction in $\mathcal{H}$. We divide them into three different classes, again following Zucker [41].

1. The cut formula is active in both premises. These reductions are called *essential* conversions.
2. Cut formula is passive in a premise. We call these *permutative* reductions.
3. Cut formula is contracted in a premise. There is one conversion rules for contraction, called *contraction* conversion.

Definition 21 (*Essential Reductions*). We write $\mathcal{D} \Rightarrow \mathcal{E}$ if we can obtain $\mathcal{E}$ from $\mathcal{D}$ by one of the rewrites given below. Note that some given deduction may rewrite to several distinct deductions, even in the case of essential reductions.

1. One of the premises of the cut is initial. Then we eliminate the cut immediately:

$$\dfrac{U, A, {\sim}A \qquad \overset{\mathcal{D}}{A, V}}{U, A, V}\, Cut \qquad \Rightarrow \qquad \begin{matrix}\mathcal{D} \oplus U \\ U, A, V\end{matrix}$$

   For the remaining essential reductions we assume that both cut formulas are active in the last inference.

2. The cut formula is a literal A. Then the previous case must apply, since a literal can only be active in an initial deduction.

3. The cut formula is $A \vee B$. There are two possible reductions.

$$\dfrac{\dfrac{\overset{\mathcal{D}}{U, A, B}}{U, A \vee B} \vee I \qquad \dfrac{\overset{\mathcal{E}_1}{V_1, \sim A} \qquad \overset{\mathcal{E}_2}{V_2, \sim B}}{V_1, V_2, \sim A \wedge \sim B} \wedge I}{U, V_1, V_2} Cut \quad \Rightarrow \quad \dfrac{\dfrac{\overset{\mathcal{D}}{U, A, B} \qquad \overset{\mathcal{E}_1}{V_1, \sim A}}{U, V_1, B} Cut \qquad \overset{\mathcal{E}_2}{V_2, \sim B}}{U, V_1, V_2} Cut$$

$$\dfrac{\dfrac{\overset{\mathcal{D}}{U, A, B}}{U, A \vee B} \vee I \qquad \dfrac{\overset{\mathcal{E}_1}{V_1, \sim A} \qquad \overset{\mathcal{E}_2}{V_2, \sim B}}{V_1, V_2, \sim A \wedge \sim B} \wedge I}{U, V_1, V_2} Cut \quad \Rightarrow \quad \dfrac{\overset{\mathcal{E}_1}{V_1, \sim A} \qquad \dfrac{\overset{\mathcal{D}}{U, A, B} \qquad \overset{\mathcal{E}_2}{V_2, \sim B}}{U, V_2, A} Cut}{U, V_1, V_2} Cut$$

4. The cut formula is $\forall y A(y)$.

$$\dfrac{\dfrac{\overset{\mathcal{D}(a)}{U, A(a)}}{U, \forall y A(y)} \forall I \qquad \dfrac{\overset{\mathcal{E}}{\sim A(C), V}}{\exists y \sim A(y), V} \exists I}{U, V} Cut \quad \Rightarrow \quad \dfrac{\overset{\mathcal{D}(C)}{U, A(C)} \qquad \overset{\mathcal{E}}{\sim A(C), V}}{U, V} Cut$$

Note that substituting C for the parameter a is a legal operation, transforming one deduction into another.

Definition 22 Now we consider *commutative* conversions. A commutative reduction is possible if there is a premiss of the cut in which the cut formula is passive. We just write out the cases where the cut formula is passive on the left. The other cases are completely symmetrical.

1. Last inference is initial. In this case the cut reduces to an initial deduction.

$$\dfrac{U, A, \sim A, X \qquad \overset{\mathcal{E}}{\sim X, V}}{U, A, \sim A, V} Cut \qquad \underset{p}{\rightarrow} \qquad U, A, \sim A, V$$

2. Last inference is $\vee I$.

$$\dfrac{\dfrac{\overset{\mathcal{D}}{U, A, B, X}}{U, A \vee B, X} \vee I \qquad \overset{\mathcal{E}}{\sim X, V}}{U, A \vee B, V} Cut \qquad \underset{p}{\rightarrow} \qquad \dfrac{\dfrac{\overset{\mathcal{D}}{U, A, B, X} \qquad \overset{\mathcal{E}}{\sim X, V}}{U, A, B, V} Cut}{U, A \vee B, V} \vee I$$

3. Last inference is $\wedge I$. Assume the cut formula (occurrence) appears in the left premiss of the $\wedge I$.

$$\dfrac{\dfrac{\overset{\mathcal{D}_1}{U_1, A, X} \qquad \overset{\mathcal{D}_2}{U_2, B}}{U_1, U_2, A \wedge B, X} \wedge I \qquad \overset{\mathcal{E}}{\sim X, V}}{U_1, U_2, A \wedge B, V} Cut \qquad \underset{p}{\rightarrow}$$

$$\dfrac{\dfrac{\overset{\mathcal{D}_1}{U_1, A, X} \quad \overset{\mathcal{E}}{\sim X, V}}{U_1, A, V}\, Cut \quad \overset{\mathcal{D}_2}{U_2, B}}{U_1, U_2, A \wedge B, V}\, \wedge I$$

If X appears in the right premiss of the $\wedge I$, the symmetric reduction is allowed.

4. Last inference is $\exists I$.

$$\dfrac{\dfrac{\overset{\mathcal{D}}{U, A(C), X}}{U, \exists y A(y), X}\, \exists I \qquad \overset{\mathcal{E}}{\sim X, V}}{U, \exists y A(y), V}\, Cut \quad \underset{p}{\rightarrow} \quad \dfrac{\overset{\mathcal{D}}{U, A(C), X} \qquad \overset{\mathcal{E}}{\sim X, V}}{\dfrac{U, A(C), V}{U, \exists y A(y), V}\, \exists I}\, Cut$$

5. Last inference is $\forall I$.

$$\dfrac{\dfrac{\overset{\mathcal{D}}{U, A(a), X}}{U, \forall y A(y), X}\, \forall I \qquad \overset{\mathcal{E}}{\sim X, V}}{U, V, \forall y A(y)}\, Cut \quad \underset{p}{\rightarrow} \quad \dfrac{\overset{\mathcal{D}}{U, A(a), X} \qquad \overset{\mathcal{E}}{\sim X, V}}{\dfrac{U, A(a), V}{U, \forall y A(y), V}\, \forall I}\, Cut$$

If a is free in V, rename a to a new parameter b everywhere in $\mathcal{E}$.

6. Last inference is contraction.

$$\dfrac{\dfrac{\overset{\mathcal{D}}{U, A, A, X}}{U, A, X}\, C \qquad \overset{\mathcal{E}}{\sim X, V}}{U, A, V}\, Cut \quad \underset{p}{\rightarrow} \quad \dfrac{\overset{\mathcal{D}}{U, A, A, X} \qquad \overset{\mathcal{E}}{\sim X, V}}{\dfrac{U, A, A, V}{U, A, V}\, C}\, Cut$$

Definition 23 The last remaining conversion is *contraction* conversion. Here the cut formula must be contracted in an antecedent. We assume that the deduction with the contracted formula is contraction normal and $A^1 \not\gg^{\mathcal{D}} A^2$.

1. The cut formula is contracted in a premiss.

$$\dfrac{\dfrac{\overset{\mathcal{D}}{U, A^1, A^2}}{U, A^{12}}\, C \qquad \overset{\mathcal{E}}{\sim A, V}}{U, V}\, Cut \quad \underset{c}{\rightarrow} \quad \dfrac{\dfrac{\overset{\mathcal{D}}{U, A^1, A^2} \qquad \overset{\mathcal{E}}{\sim A, V}}{U, V, A^1}\, Cut^2 \qquad \overset{\mathcal{E}}{\sim A, V}}{\dfrac{U, V, V}{U, V}\, C}\, Cut^1$$

Remark 24 The contraction conversion is what destroys the strong normalization property of the system with $\underset{p}{\rightarrow} \cup \Rightarrow \cup \underset{c}{\rightarrow}$, even with the given restriction on contraction reduction.

Example 25 Counterexample to show that $\rightarrow \underset{\text{def}}{=} \underset{p}{\rightarrow} \cup \Rightarrow \cup \underset{c}{\rightarrow}$ has infinite reduction sequences.

$$\dfrac{\dfrac{\overset{\mathcal{D}}{U, A^1, A^2}}{U, A^{12}}\, C \quad \dfrac{\overset{\mathcal{E}}{\sim A^1, \sim A^2, V}}{\sim A^{12}, V}\, C}{U, V}\, Cut$$

In this example, $\mathcal{D}$ and $\mathcal{E}$ are completely arbitrary. After one contraction conversion we obtain

$$\dfrac{\dfrac{\dfrac{\overset{\mathcal{D}}{U, A^1, A^2} \qquad \dfrac{\overset{\mathcal{E}}{\sim A^1, \sim A^2, V}}{\sim A^{12}, V}\, C}{U, A^1, V}\, Cut^{2,12} \qquad \dfrac{\overset{\mathcal{E}}{\sim A^1, \sim A^2, V}}{\sim A^{12}, V}\, C}{U, V, V}\, Cut^{1,12}}{U, V}\, C$$

Another contraction conversion yields

$$\dfrac{\dfrac{\dfrac{\overset{\mathcal{D}}{U, A^1, A^2} \qquad \dfrac{\overset{\mathcal{D}}{U, A^1, A^2} \qquad \overset{\mathcal{E}}{V, \sim A^1, \sim A^2}}{U, A^1, \sim A^2, V}\, Cut^{2,1}}{U, U, A^1, A^1, V}\, Cut^{2,2}}{U, A^{11}, V}\, C \qquad \dfrac{\overset{\mathcal{E}}{\sim A^1, \sim A^2, V}}{\sim A^{12}, V}\, C}{U, V, V}\, cut^{11,12}}{U, V}\, C$$

Let us replace a big piece of this deduction by a deduction variable, since it will not matter what it is. It will then be easier to see how this can lead to an infinite sequence of reductions. Let us also rename some side formulas.

$$\dfrac{\dfrac{\overset{\mathcal{F}}{U', A^1, A^1}}{U', A^{11}}\, C \quad \dfrac{\overset{\mathcal{E}}{\sim A^1, \sim A^2, V}}{\sim A^{12}, V}\, C}{U', V}\, Cut$$

One can see that this cut has exactly the form of our original cut. Hence there is an infinite sequence of reductions, since no matter how the upper cuts in $\mathcal{F}$ are eliminated, one can apply the analogous two reduction steps to the result.

A proper sequence of reductions in the sense of Definition 31 will prohibit such infinite sequence of reductions by ensuring that the uppermost of the two cuts after a contraction conversion is permuted upwards until no duplication of the remaining occurrence of the cut formula is possible. But this requires the use of *contraction normal* deductions — only for them such a requirement can be enforced by restricting possible sequences of reductions alone.

We would also like the requirement to be as general as possible, so as to keep the number of forbidden reduction sequences low. The definitions have been formulated carefully to achieve this goal. For example, we do not require any relation between occurrences of the same formula on disjoint branches of the proof tree.

The conversion rule which allows to interchange one cut with another was not considered by Gentzen [11]. Since his algorithm always eliminates the topmost cuts first, this omission does not present a problem. We follow Gentzen here in not considering cut-cut conversion a proper reduction, even though it is a very natural operation. It would amount to changing the order in which lemmas are applied. Unfortunately, as noted by Zucker [41], cut elimination does not hold in any strong sense. One could perhaps still obtain a weak cut elimination result, in which we declare successive interchanges of cuts improper. The absence of this natural conversion rule shows that our normalization result is in some sense weaker than the corresponding result for the $\rightarrow\forall$ fragment of intuitionistic natural deduction.

Definition 26 (*Cut-cut conversion*)

$$\dfrac{\dfrac{\begin{matrix}\mathcal{D}\\ U, A, B\end{matrix} \qquad \begin{matrix}\mathcal{E}_1\\ \sim A, V_1\end{matrix}}{U, V_1, B}\,Cut \qquad \begin{matrix}\mathcal{E}_2\\ \sim B, V_2\end{matrix}}{U, V_1, V_2}\,Cut \quad \rightarrow \quad \dfrac{\dfrac{\begin{matrix}\mathcal{D}\\ U, A, B\end{matrix} \qquad \begin{matrix}\mathcal{E}_2\\ \sim B, V_2\end{matrix}}{U, A, V_2}\,Cut \qquad \begin{matrix}\mathcal{E}_1\\ \sim A, V_1\end{matrix}}{U, V_1, V_2}\,Cut$$

Remark 27 Another problem of the cut-cut conversion can be realized by considering the contraction conversions. If cut-cut conversion was allowed, the restriction on the contraction conversion would be meaningless, since one could simply interchange the two cuts on the right-hand side. However the restriction is necessary for termination.

2.5 Cut Elimination in $\mathcal{H}$

Definition 28 Let $\longrightarrow\!\!\!\!\!\rightarrow$ be the reflexive and transitive closure of $\underset{p}{\rightarrow} \cup \underset{c}{\rightarrow}$. This includes permutative and contraction conversions, but excludes essential conversions.

Definition 29 Let $\mathcal{D}$ be a deduction with a formula occurrence $A \in \texttt{Final}(\mathcal{D})$. We define $\mathrm{cr}^{\mathcal{D}}(A)$ the *contraction rank* of A in $\mathcal{D}$ by induction on $\mathcal{D}$.

1. A is passive in $\mathcal{D}$. If $\mathcal{D}$ is initial then $\mathrm{cr}^{\mathcal{D}}(A) = 1$. Otherwise let $\mathcal{D}'$ be the immediate subderivation of $\mathcal{D}$ such that A occurs in $\mathcal{D}'$. Then $\mathrm{cr}^{\mathcal{D}}(A) = \mathrm{cr}^{\mathcal{D}'}(A)$

2. A is active in $\mathcal{D}$. Then $\mathrm{cr}^{\mathcal{D}}(A) = 1$.

3. A is contracted from A^1 and A^2. Then $\mathrm{cr}^{\mathcal{D}}(A) = \mathrm{cr}^{\mathcal{D}'}(A^1) + \mathrm{cr}^{\mathcal{D}'}(A^2)$, where $\mathcal{D}' \lhd \mathcal{D}$.

4. $\mathcal{D} \in \mathcal{D}' \oplus A$ for some $\mathcal{D}'$. Then $\mathrm{cr}^{\mathcal{D}}(A) = 1$. This clause is necessary in the absence of a rule of weakening (which could be a derived rule of inference).

5. $\mathcal{D}$ ends in cut and the previous case does not apply. Then $\mathrm{cr}^{\mathcal{D}}(A)$ is undefined.

The *contraction lemma* is the key to understanding the restriction on reductions which gives us a "weak" cut-elimination result. Since the proof of the existential statement is constructive, an algorithm can be extracted from it.

Lemma 30 (*Contraction Lemma*) Given is

$$\mathcal{F} = \quad \dfrac{\begin{array}{cc}\overset{\mathcal{D}}{} & \overset{\mathcal{E}}{}\\ U, A^1, \ldots, A^{n-1}, A^n & \sim A, V\end{array}}{U, A^1, \ldots, A^{n-1}, V}\, Cut$$

such that

1. $\mathcal{D}$ is contraction normal.

2. $A^i \not\gg A^n$ for $1 \leq i \leq n-1$

3. $\mathrm{cr}^{\mathcal{D}}(A^i)$ is defined for $1 \leq i \leq n$.

Then there exists a $\mathcal{G}$ such that $\mathcal{F} \longrightarrow\!\!\!\!\!\rightarrow \mathcal{G}$ and $\mathrm{cr}^{\mathcal{G}}(A^i) = \mathrm{cr}^{\mathcal{D}}(A^i)$ for $1 \leq i \leq n-1$.

Note that $\mathrm{cr}^{\mathcal{F}}(A^i)$ itself is not defined, since $\mathcal{F}$ ends in cut.

Proof: By double induction on $\mathrm{cr}^{\mathcal{D}}(A^n)$ and $\mathcal{D}$.

1. $\mathcal{D}$ ends in a cut. Then $\mathcal{F}$ itself can serve as $\mathcal{G}$. Since $\mathrm{cr}^{\mathcal{D}}(A^i)$ is defined, but $\mathcal{D}$ ends in a cut, $\mathcal{D}$ must be of the form $\mathcal{D}' \oplus A^1 \oplus \cdots \oplus A^{n-1} \oplus A^n$ for some $\mathcal{D}'$ and hence $\mathcal{F}$ is of the form $\mathcal{F}' \oplus A^1 \oplus \cdots \oplus A^{n-1}$, and $\mathrm{cr}^{\mathcal{F}}(A^i) = 1 = \mathrm{cr}^{\mathcal{D}}(A^i)$ for $1 \leq i \leq n-1$.

2. All A^i are passive. If $\mathcal{D}$ is initial the claim follows immediately, since $\mathcal{F} \underset{p}{\rightarrow} \mathcal{G}$ and $\mathrm{cr}^{\mathcal{D}}(A^i) = 1 = \mathrm{cr}^{\mathcal{G}}(A^i)$ for $1 \leq i \leq n-1$, since $\mathcal{G}$ is also initial. Otherwise let $\mathcal{D}' \lhd \mathcal{D}$ contain A^n. Then $\mathcal{F} \underset{p}{\rightarrow} \mathcal{F}'$, where $\mathcal{F}'$ is $\mathtt{Cut}(\mathcal{D}', \mathcal{E}, A^n)$ followed by the last inference in $\mathcal{D}$. For those A^i such that $A^i \in \mathtt{Final}(\mathcal{D}')$ the lemma follows by induction hypothesis on the length of $\mathcal{D}$ applied to $\mathcal{D}'$. For the A^i which do not occur in $\mathcal{D}'$ ($\mathcal{D}$ must have ended in $\wedge I$), it follows directly that $\mathrm{cr}^{\mathcal{G}}(A^i) = \mathrm{cr}^{\mathcal{D}}(A^i)$ for *any* $\mathcal{G}$ such that $\mathcal{F}' \underset{p}{\longrightarrow\!\!\!\!\!\rightarrow} \mathcal{G}$.

3. Some A^i for $1 \leq i \leq n-1$ is active. If $\mathcal{D}$ is initial the lemma follows from the equation $\mathrm{cr}^{\mathcal{E} \oplus W}(B) = 1$ for $B \in W$ applied to $W = U, \{A^j : j \neq i\}$ and $B \in \{A^j : j \neq i\}$. If we had not made the definition that the contraction rank of an adjoined formula occurrence is 1, we would have had to use a rule of weakening in order to make sure that this base case for initial $\mathcal{D}$ is always satisfied, since $\mathcal{E}$ may contain applications of cut.

4. A^n is active. Then $n = 1$ because $A^i \not\gg A^n$ and the lemma is trivially satisfied.

5. Some A^i for $1 \leq i \leq n-1$ is contracted. Again the lemma follows directly from the induction hypothesis on the construction of $\mathcal{D}$.

6. A^n is contracted. This is the most interesting case. Here we need the induction hypothesis on the contraction rank of A^n.

$$\mathcal{F} = \dfrac{\dfrac{\overset{\mathcal{D}'}{U, A^1, \ldots, A^{n-1}, A^{n1}, A^{n2}}}{U, A^1, \ldots, A^{n-1}, A^n} C^{n1,n2} \quad \dfrac{\overset{\mathcal{E}}{\sim A, V}}{U, A^1, \ldots, A^{n-1}, V} Cut^n$$

$$\underset{c}{\rightarrow}$$

$$\mathcal{F}' = \dfrac{\dfrac{\overset{\mathcal{D}'}{U, A^1, \ldots, A^{n-1}, A^{n1}, A^{n2}} \quad \overset{\mathcal{E}}{\sim A, V}}{U, A^1, \ldots, A^{n-1}, A^{n1}, V} Cut^{n2} \quad \overset{\mathcal{E}}{\sim A, V}}{\dfrac{U, A^1, \ldots, A^{n-1}, V, V}{U, V} C} Cut^{n1}$$

where $A^{n1} \not\gg A^{n2}$.

Therefore we can apply the induction hypothesis to Cut^{n2} to find a deduction $\mathcal{G}'$ such that $Cut^{n2} \longrightarrow\!\!\!\!\!\rightarrow \mathcal{G}'$ and $\mathrm{cr}^{\mathcal{G}'}(A^i) = \mathrm{cr}^{\mathcal{D}'}(A^i)$ and $\mathrm{cr}^{\mathcal{G}'}(A^{n1}) = \mathrm{cr}^{\mathcal{D}'}(A^{n2})$.

Also,

$$\mathcal{F}' \longrightarrow\!\!\!\!\!\rightarrow \dfrac{\dfrac{\overset{\mathcal{G}'}{U, A^1, \ldots, A^{n-1}, A^{n1}, V} \quad \overset{\mathcal{E}}{\sim A, V}}{\dfrac{U, A^1, \ldots, A^{n-1}, V, V}{U, V} C} Cut^{n1}$$

and we can apply the inductive hypothesis on the contraction rank, since $\mathrm{cr}^{\mathcal{D}}(A^n) > \mathrm{cr}^{\mathcal{G}'}(A^{n1})$.

□

Definition 31 Given is a contraction normal deduction $\mathcal{D}$ ending in a contraction of A^1 and A^2 to A^{12} such that $\mathrm{cr}^{\mathcal{D}}(A^{12})$ is defined. Let $\mathcal{F}$ be a deduction ending in a cut of A^{12} from $\mathcal{D}$. Then a reduction sequence $\mathcal{F} \longrightarrow\!\!\!\!\!\rightarrow \mathcal{G}$ is called *proper* iff for every cut of an ancestor A^k of A^{12} in $\mathcal{G}$, the contraction rank of A^k is strictly less that the contraction rank of A^{12}

Lemma 30 shows that such proper reductions sequences always exist. Since the proof is constructive, it specifies an algorithm for performing proper reduction sequences if the cut formula is contracted in the premiss.

Definition 32 A reduction sequence $\mathcal{F} \longrightarrow\!\!\!\!\!\rightarrow \mathcal{G}$ is *proper* iff every segment of it is proper.

Theorem 33 (*Cut-elimination in the first-order fragment*) In the first-order fragment of $\mathcal{H}$, every proper reduction sequence terminates.

Proof: The proof is by a double induction on the contraction rank and complexity of the cut formula. If we let the complexity of a first-order formula be the number of connectives and quantifiers, then after every essential reduction the complexity of all new cut formulas is smaller than the complexity of the original cut formula. For every proper reduction sequence the complexity of the cut formula is not changed, but the contraction rank of each of the new cut formula occurrences is less than the contraction rank of the original cut. Hence, by a double induction, every proper reduction sequence will terminate.

□

Conjecture 34 Every proper reduction sequence in $\mathcal{H}$ terminates.

See a discussion of this conjecture in the introduction.

Chapter 3

Expansion Proofs

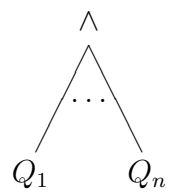
3.1 Expansion Proofs

Analytic proofs in this thesis are presented as expansion trees. Expansion trees very concisely and naturally represent the information contained in an analytic proof, as we hope to show. They were first introduced by Miller [23] and are somewhat similar to Herbrand expansions [16]. Some redundancies can easily be eliminated for an actual implementation as done by Miller in the context of higher order logic. The shallow formula of an expansion tree will correspond to the theorem; the deep formula is akin to a Herbrand-expansion proving the theorem. Our formulation of expansion trees differs from Miller's in [22] in several respects. Firstly, we found it convenient to allow n-ary conjunction and disjunction instead of treating them as binary operations. Secondly, we allow arbitrary formulas at the leaves of expansion trees instead of requiring them to be literals. And thirdly we allow the mating to be a relation on arbitrary nodes in the expansion tree, rather than merely on the leaves. On the other hand we do not allow negations to appear at arbitrary levels in the expansion tree.

Definition 35 We define *expansion trees* inductively. Simultaneously, we also define Q^D, the *deep formula* of an expansion tree, and Q^S, the *shallow formula* of an expansion tree. We furthermore place the restriction that no parameter in an expansion tree may be selected more than once.

1. A formula A_o is an expansion tree. $Q^D(A) = Q^S(A) = A$. Formulas form the leaves of expansion trees.

2. If $Q_1, \ldots, Q_n$, $n \geq 0$, are expansion trees, so is

$Q =$ 

Then $Q^D = Q_1^D \wedge \cdots \wedge Q_n^D$, and $Q^S = Q_1^S \wedge \cdots \wedge Q_n^S$.

3. If $Q_1, \ldots, Q_n$, $n \geq 0$, are expansion trees, so is

$$Q = \quad \begin{array}{ccccc} & & \vee & & \\ & \swarrow & & \searrow & \\ Q_1 & & \cdots & & Q_n \end{array}$$

Then $Q^D = Q_1^D \vee \cdots \vee Q_n^D$, and $Q^S = Q_1^S \vee \cdots \vee Q_n^S$.

4. If $Q_1, \ldots, Q_n$ are expansion trees such that $Q_i^S = A(t_i)$, for $1 \leq i \leq n, n \geq 1$, then

$$Q = \quad \begin{array}{ccccc} & & \exists x A(x) & & \\ & \swarrow & & \searrow & \\ t_1 & & \cdots & & t_n \\ Q_1 & & & & Q_n \end{array} \quad \text{is an expansion tree.}$$

Then $Q^D = Q_1^D \vee \cdots \vee Q_n^D$, and $Q^S = \exists x A(x)$.

Q is called an *expansion node*; x is the *expanded variable*; $t_1, \ldots, t_n$ are the *expansion terms*. Note that expansion terms are treated as if indexed, that is, two expansion terms s and t may perhaps be the same formula, but nevertheless may be distinct expansion terms, even in the same expansion node.

5. If Q_0 is an expansion tree such that $Q_0^S = A(a)$ for a parameter a, so is

$$Q = \quad \begin{array}{c} \forall x A(x) \\ \Big| a \\ Q_0 \end{array}$$

Then $Q^D = Q_0^D$, and $Q^S = \forall x A(x)$.

Q is called a *selection node*; a is the *parameter selected* for this occurrence of x.

When given a formula occurrence A and expansion tree Q we will write Q_A for the node with shallow formula A corresponding to the given occurrence of A. The nature of the correspondence will depend on the context. Similarly, $Q|_U$, the restriction of Q to U, is the part of the expansion tree Q corresponding to the multiset of formulas U. We similarly restrict matings and write $\mathcal{M}|_U$ for the restriction of $\mathcal{M}$ to nodes in $Q|_U$.

We explicitly allow disjunction and conjunction nodes with no successors. They will turn out to be useful in Chapter 6, since they behave like truth and falshood, respectively. In Chapters 2, 3, 4, and 5 we restrict ourselves to expansion proofs where all disjunction and conjunction nodes except for the disjunction node at the root must have exactly two successors. However, lemmas concerning expansion proofs are formulated that they either directly apply in the case of n-ary disjunctions and conjunctions, are can easily be extended to apply in this case, also.

Since traditional proof systems do not contain Skolem-functions, we need a different mechanism to insure the soundness of our proofs. Following an idea of Bibel [7], which was picked up by Miller [23], we introduce a relation $<_Q$ on occurrences of expansion terms. The condition that $<_Q$ be acyclic replaces Skolemization in our analytic proof system. The correctness if this definition will become clear in the proof of soundess of expansion proofs (see Algorithm 77) . $<_Q$ is dual to $\prec_Q$ and it is shown in [23] that they are equivalent.

Definition 36 Let Q be an expansion tree. $<_Q^0$ is a relation on occurrences of expansion terms such that $t <_Q^0 s$ iff there is a parameter selected for a node below t in Q which is free in s. $<_Q$, the *dependency relation*, is the transitive closure of $<_Q^0$.

Definition 37 Let Q be an expansion tree. $\prec_Q^0$ is a relation on selected parameters such that $a \prec_Q^0 b$ if there is an expansion term occurrence t such that a is free in t and b is selected below t in Q. $\prec_Q$, the *imbedding relation*, is the transitive closure of $\prec_Q^0$.

Usually clauses are defined only for quantifier-free formulas. In the context of higher-order logic it is convenient to allow arbitrary formulas in a clause. The structure of the formulas in clauses is only analyzed to the extent that we check whether $A \underset{\lambda^\sim}{=} \sim B$. Remember that in general equality is up to λ and $\sim$-conversion.

We made a similar decision, when we allowed arbitrary formulas A in the initial lines in an $\mathcal{H}$ deduction. The system where only literals are allowed is also complete.

We define what we mean by a full clause in an expansion tree. Full clauses differ from the ordinary concept of clause in two ways. First of all, the elements of a clause may represent arbitrary formulas instead of merely literals. Secondly, we enter not only the leaves of the expansion tree, but also intermediate nodes into the full clause. This generalizes the notion of clause for higher-order logic given by Miller [23]. The reason for our definition is that it allows arbitrary nodes in the an expansion tree to be mated (see Definition 41).

Definition 38 We write $\langle e_1, \ldots, e_n \rangle$ for a list with elements e_1 through e_n. $l_1 @ \cdots @ l_n$ is the result of appending the lists $l_1, \ldots, l_n$.

Definition 39 Let Q be an expansion tree. A *full clause* in Q is a list of nodes defined inductively by

1. Q is a leaf node. Then $C = \langle A \rangle$ is the only full clause in Q.

2.

$$Q = \quad \overset{\vee}{\underset{Q_1 \quad \cdots \quad Q_n}{}}$$

Then for all clauses c_1 in Q_1, ..., c_n in Q_n, the concatenation $\langle Q \rangle \texttt{@} c_1 \texttt{@} \ldots \texttt{@} c_n$ is a full clause in Q.

3.

$$Q = \quad \overset{\wedge}{\underset{Q_1 \quad \cdots \quad Q_n}{}}$$

Then for any full clause c in some Q_i, the list $\langle Q \rangle \texttt{@} c$ is a full clause in Q.

4.

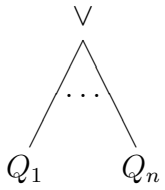
$$Q = \quad \overset{\forall x A(x)}{\underset{Q_0}{\Big| a}}$$

Then for every full clause c in Q_0, the list $\langle Q \rangle \texttt{@} c$ is a full clause in Q.

5.

$$Q = \quad \overset{\exists x A(x)}{\underset{Q_1 \quad \cdots \quad Q_n}{t_1 \quad \cdots \quad t_n}}$$

Then for every full clause c in



the list $\langle Q\rangle @c$ is a full clause in Q.

Definition 40 We define $\mathrm{fc}(Q)$ to be the set of full clauses of the expansion tree Q. $\mathrm{fc}(Q)\backslash Q_0$ is the set of full clauses of Q where Q_0 has been deleted from every clause in which it occurs.

Definition 41 A symmetric relation on nodes of an expansion tree Q is a *mating* $\mathcal{M}$ if $\sim l^S \underset{\lambda^\sim}{=} k^S$ for any unordered pair $(l,k) \in \mathcal{M}$. Furthermore we require that for any pair $(l,k) \in \mathcal{M}$, there is at least one full clause in Q containing both l and k. If $(l,k) \in \mathcal{M}$, l and k are said to be $\mathcal{M}$-mated.

Definition 42 Let $\mathcal{M}$ be a mating on Q. A node l in Q is said to *occur* in $\mathcal{M}$ if there is a k such that $(l,k) \in \mathcal{M}$.

Definition 43 A mating $\mathcal{M}$ is said to *span a full clause* c if there are nodes $l, k \in c$ such that $(l,k) \in \mathcal{M}$. A mating $\mathcal{M}$ is said to be *clause-spanning* on an expansion tree Q if every full clause in Q is spanned by $\mathcal{M}$.

Example 44 This example illustrates that in the setting of higher-order logic a node may be mated to a node below it. Names for the nodes are written into the tree.

$$
\begin{array}{ll}
[\exists AA] & : Q_0 \\
\quad\Big| \exists BB & \\
{[\exists BB]} & : Q_1 \\
\quad\Big| \forall C{\sim}C & \\
{[\forall C{\sim}C]} & : Q_2
\end{array}
$$

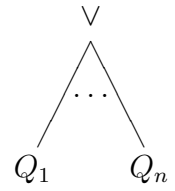
There is only one full clause, namely $\langle Q_0, Q_1, Q_2\rangle$, and it would be spanned by the pair (Q_0, Q_2). One can easily see that the shallow formulas are complementary.

Definition 45 A pair $(Q, \mathcal{M})$ is called an *expansion proof* for a formula A if

1. $Q^S = A$.
2. No selected parameter is free in Q^S.
3. $<_Q$ is acyclic.
4. $\mathcal{M}$, a *mating* on Q, is clause-spanning on Q.

We will establish soundness and completeness of expansion proofs. We will give translations from expansion proof to deductions in $\mathcal{H}$ and vice versa. We rely on the soundness and completeness of $\mathcal{H}$, which is well-known in first-order logic (for example Smullyan [33], or Tait [36]). In higher-order logic, this was known as "Takeuti's Conjecture" (in a slightly different formulation of the logical system) and was established by Takahashi [37]. A different proof was given by Miller [23] which used an abstract consistency property which was introduced by Smullyan [33] and generalized to higher-order logic by Andrews [1]. Another presentation may be found in Takeuti [40].

Definition 46 We say an expansion proof Q is an expansion proof for a multiset U of formulas if there is a ordering $\langle A_1, \ldots, A_n \rangle$ of the formulas occurrences in U such that

$Q =$ 

and $Q_1^S = A_1, \ldots, Q_n^S = A_n$.

3.2 Basic Operations on Expansion Proofs

The motivations for some of the operations presented in this section will become clear later, as we discuss translations between expansion proofs and deductions in $\mathcal{H}$. Thus the reader may wish to skip this sections and refer back to its definitions as they are used in the next three sections.

We begin by giving the definition of the operations on trees, and later apply that definition to expansion trees and expansion proofs. We do not have much occasion to explicitly manipulate the set of nodes and arcs which represent an expansion tree. If we need to, an arc will be an ordered pair $\langle l, k \rangle$ of nodes l and k, and every node k, except for the root, will have a unique predecessor l such that $\langle l, k \rangle \in Q$,

Definition 47 The result of *deletion* below Q_i in Q, $\texttt{del}(Q, Q_i)$, is the tree obtained by deleting the whole subtree below Q_i from Q.

Definition 48 Let $(Q, \mathcal{M})$ be an expansion proof with node Q_0 such that Q_0 is not the root and does not occur in $\mathcal{M}$. Let $\texttt{shallow}(Q_0, (Q, \mathcal{M}))$ be the result of replacing unique arc $\langle l, Q_0\rangle$ in Q by the set of arcs $\{\langle l, k\rangle : \langle Q_0, k\rangle \in Q\}$. Note that $\texttt{shallow}(Q_0, (Q, \mathcal{M}))$ is not always an expansion proof (see Lemma 53 for some sufficient conditions).

Definition 49 For trees Q and R, we define $Q \sqsubset_D R$ if Q is the result of a sequence of shallowings and deletions in R.

Lemma 50 $\sqsubset_D$ is well-founded.

Proof: This is obvious, because the number of arcs decreases with each shallowing and deletion and is finite.

□

Definition 51 Let $(Q, \mathcal{M})$ be an expansion proof with node Q_0. We say Q_0 is *accessible* if Q_0 is not below any expansion or selection node, and no node above Q_0 in Q is $\mathcal{M}$-mated. A formula A will be called accessible in U if A is not in the scope of any universal or existential quantifier and not inside an atomic formula in U.

The notion of *accessible* will be very important later. When a node is accessible in an expansion proof we could replace it by a tree with another shallow formula and still get an expansion proof. If the node lies below an expansion node that is in general not possible, since shallow and deep formulas of the expansion node will no longer match. Also, if the node itself or one above is $\mathcal{M}$-mated, the node and its mate will no longer have the same shallow formula, thus invalidating the mating. The condition that the node itself is not mated is the definition of *single*.

Definition 52 Let $(Q, \mathcal{M})$ be an expansion proof with node Q_0. We say Q_0 is *single* if neither Q_0 nor any node above Q_0 in Q occur in $\mathcal{M}$.

Lemma 53 (*Shallowing Lemma*) Let $(Q, \mathcal{M})$ be an expansion proof with single and accessible node Q_0 such that either

1. Q_0 is an expansion node with only one expansion term t and t is admissible in Q.
2. Q_0 is a selection, conjunction or disjunction node.

Then $(R, \mathcal{N}) = \texttt{shallow}(Q_0, (Q, \mathcal{M}))$ is again an expansion proof and $(R, \mathcal{N}) \sqsubset_D (Q, \mathcal{M})$.

Proof: by cases.

First note that in all cases conditions 4 and 5 in the definition of expansion tree (Definition 35) will still be satisfied in R, since Q_0 was assumed not to be below any selection or expansion node in Q.

1. Q_0 is an expansion node with only one expansion term t and t is admissible in Q. Because t is admissible, no parameter free in t is selected in Q. Therefore no parameter selected in R will be free in R. $\mathcal{N} = \mathcal{M}$ will be clause-spanning on R, because $\mathrm{fc}(R) = \mathrm{fc}(Q)\backslash Q_0$ and Q_0 does not occur in $\mathcal{M}$ by assumption.

2. Q_0 is a selection node. Because no parameter may be selected in more than one node, no parameter in R will be selected in R. It follows just as in case 1 that $\mathcal{N} = \mathcal{M}$ is clause-spanning on R.

3. Q_0 is a disjunction node. As in case 1, $\mathcal{N} = \mathcal{M}$ is clause-spanning on R. All other conditions are trivially satisfied.

4. Q_0 is a conjunction node. Here the set of full clauses actually changes non-trivially. However, any clause in R is an extension of a clause in Q since it will contain nodes from each of the subtress of Q_0 and not just one of them. By this observation and the assumption that Q_0 itself did not appear in $\mathcal{M}$, $\mathcal{N} = \mathcal{M}$ is clause-spanning on R.

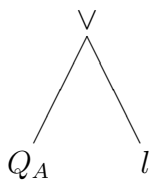
□

Remark 54 In the case of a conjunction, Lemma 53 by itself is not very useful, because the result of shallowing $Q_{A\wedge B}$ in expansion proof $(Q, \mathcal{M})$ for $U, A \wedge B$ will be an expansion proof for U, A, B. However, we will also show that, say A, is inessential and so we can obtain an expansion proof for U, B from one for $U, A \wedge B$ by a shallowing followed by an erasure.

Definition 55 Let $(Q, \mathcal{M})$ be an expansion proof with node Q_0. Then $\mathtt{erase}(Q_0, (Q, \mathcal{M}))$ is the result deleting Q_0 from Q. Note that this may create a new leaf if the last successor of a disjunction, conjunction, expansion or selection node is erased.

Lemma 56 Let $(Q, \mathcal{M})$ be an expansion proof with node Q_0 such that Q_0 is unnecessary (see Definition 93) or inessential (see Definition 67) . Then $\mathtt{erase}(Q_0, (Q, \mathcal{M}))$ is again an expansion proof with $\mathtt{erase}(Q_0, (Q, \mathcal{M})) \sqsubset_D (Q, \mathcal{M})$.

Definition 57 Let $(Q, \mathcal{M})$ be an expansion proof with node Q_A, such that Q_A appears in $\mathcal{M}$, Q_A is not a leaf of Q, and no node above Q_A is $\mathcal{M}$-mated. Then let $\mathtt{crleaf}(Q_A, (Q, \mathcal{M}))$ be the result of replacing Q_A by



where l is a new leaf with shallow formula $l^S = A$. Modify $\mathcal{M}$ by replacing occurrences of Q_A by l.

Lemma 58 If Q_A is accessible, not a leaf, and occurs in $\mathcal{M}$ then $\texttt{crleaf}(Q_A, (Q, \mathcal{M}))$ is again an expansion proof.

Proof: The only part not completely obvious is to check that $\mathcal{N}$ is clause-spanning on R if $(R, \mathcal{N}) = \texttt{crleaf}(Q_A, (Q, \mathcal{M}))$. The crux in this proof is that every full clause through R which contains Q_A also contains l by definition of a full clause through an expansion tree. Given a full clause c' through R. Define the preimage c of c' through Q to by erasing of l. By assumption, $\mathcal{M}$ spans c. If the spanning pair is of the form (k, Q_A) then (k, l) will span c'. Otherwise the same pair will still be in $\mathcal{N}$ therefore span c'.

□

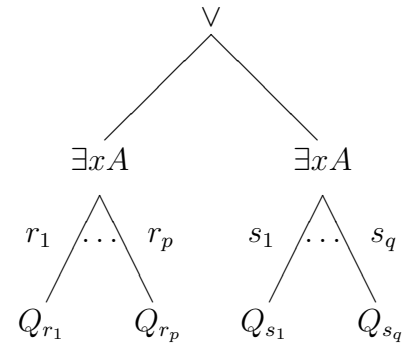
Definition 59 We say $(Q, \mathcal{M}) \sqsubset_C (R, \mathcal{N})$ if $(Q, \mathcal{M})$ is the result of creating leaves at nodes which are not leaves, but appear in $\mathcal{N}$.

Lemma 60 $\sqsubset_C$ is a well-ordering.

Proof: Trivial, since the number of non-leaves occurring in the mating decreases and is finite.

□

Definition 61 Let $(Q, \mathcal{M})$ be an expansion proof with expansion node Q_0. Moreover, let $T = \{t_1, \ldots, t_n\}$ be the set of expansion term occurrences of Q_0 and $\{r_1, \ldots, r_p\}, \{s_1, \ldots s_n\}$ a disjoint partition of T. Define $\texttt{split}(Q_0, \{r_1, \ldots, r_p\}, \{s_1, \ldots s_n\}, (Q, \mathcal{M}))$ as the result of replacing Q_0 by Q_0', where

$Q_0' =$ 

Lemma 62 Let $(Q, \mathcal{M})$ be an expansion proof with single and accessible expansion node Q_0 with expansion term occurrences T. Then for any disjoint partition T', T'' of T, $\texttt{split}(Q_0, T', T'', (Q, \mathcal{M}))$ is again an expansion proof.

Definition 63 We say $(Q, \mathcal{M}) \sqsubset_S (R, \mathcal{N})$ if $(Q, \mathcal{M})$ is the result of splitting expansion nodes in $(R, \mathcal{N})$.

Lemma 64 $\sqsubset_S$ is a well-ordering.

Proof: This is again easy to see, because the multiset-ordering on the multi-set of numbers of expansion terms in R decreases while splitting.

□

Definition 65 Given expansion proofs $(Q, \mathcal{M})$ and $(R, \mathcal{N})$. We define $\sqsubset$ as the lexicographic order obtained from $\sqsubset_S$, $\sqsubset_C$, and $\sqsubset_D$ in this order $(\sqsubset_S, \sqsubset_C, \sqsubset_D)$.

Lemma 66 $\sqsubset$ is a well-ordering.

Proof: One merely has to note that a shallowing or erasing in $(Q, \mathcal{M})$ does not increase the number of non-leaf nodes in $\mathcal{M}$ and also does not increase the number of expansion terms in Q.

□

3.3 Translating Expansion Proofs into $\mathcal{H}$

In this section we will present a non-deterministic algorithm for translating expansion proofs into $\mathcal{H}$. This is not a complete recipe for how to do this kind of translation in practice, since a lot of choices still remain, but it establishes soundness of expansion proofs. We will remark on possible heuristics and give a more practically useful translation algorithm into natural deductions in Chapter 6. We will also begin to introduce the basic definitions necessary to formulate a cut-elimination algorithm directly for expansion proofs.

Similar translations were given by Miller in [23], [22], and jointly with Felty in [24]. Here the presentation of expansion proofs is different, so the previously given algorithms must be modified. In particular, arbitrary nodes in the tree may be $\mathcal{M}$-mated, which eliminates the need for Miller's *focusing*. To focus the deduction in this context would require checking at each step in the translation process whether the current theorem is propositionally valid. If it is propositionally valid, one would invoke a derived rule of inference which allows one to assert all tautologies as initial deductions. One strong motivation for focusing is that a simple theorem like $\forall x Px \supset \forall x Px$ would otherwise require two quantificational inferences. Here the nodes corresponding to the two occurrences of $\forall x Px$ can be mated and the deduction will be the natural one consisting of one inference. Also, conjunction and contraction are treated in a novel, more general way, which increases the non-determinism of the algorithm, while still preserving its termination properties. This is desirable, since one would like to recognize many different deductions as incarnations of expansion proofs. The commitments to the order of inference rule applications should be minimal, thereby allowing good guidance for the completion of a partial proof in many situations. Closest to this presentation is the approach taken in the author's own paper on *Analytic and Non-Analytic Proofs* [26], but generalized to higher-order logic.

All the basic operations on expansion proofs are formulated in a very general way, and we continue to make use of them in later chapters. The concentration here is on what is really

essential to the connection between expansion proofs and deductions. The shortcomings of the deductive system $\mathcal{H}$ will be remedied in Chapter 6 where we consider a natural deduction system $\mathcal{N}$ very close to Gentzen's [11] system LK when generalized to type theory. While much of the formula structure is ignored in $\mathcal{H}$ since it operates on equivalence classes of formulas, $\mathcal{N}$ will preserve all of the formula structure.

Definition 67 Let $(Q, \mathcal{M})$ be an expansion proof for U, A. We say an occurrence A is *inessential* if $\mathcal{M}|_U$ is clause-spanning on $Q|_U$. We will also call Q_A inessential.

Definition 68 Let $(Q, \mathcal{M})$ be an expansion proof for U and Q_A be accessible in Q. We write $U[\![A]\!]$ for U to indicate the occurrence of A corresponding to Q_A. Then $U[\![B]\!]$ is the result of replacing the indicated occurrence of A in U by B.

Definition 69 Whenever we consider an expansion tree Q with shallow formula $U[\![A]\!]$ we write Q_A to be the subtree of Q corresponding to the occurrence of A in U. Note that this notation implies that A is accessible.

Definition 70 We say a node Q_0 is *top-level* in an expansion tree Q, if Q_0 is immediately below the root node of Q.

Definition 71 Let $(Q, \mathcal{M})$ be an expansion proof for $U, A \wedge B$. $U' \subseteq U$ is called *sufficient for* A if $(Q|_{U',A}, \mathcal{M}|_{U',A})$ is an expansion proof for U', A. Similarly, $U'' \subseteq U$ is called *sufficient for* B if $(Q|_{U'',B}, \mathcal{M}|_{U'',B})$ is an expansion proof for U'', B.

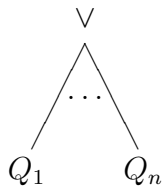
Lemma 72 Let $(Q, \mathcal{M})$ be an expansion proof for $U, A \wedge B$. Then U is sufficient for A and U is sufficient for B.

Definition 73 Let $(Q, \mathcal{M})$ be an expansion proof for $U, A \wedge B$. We say U^A, U^B *separates* $A \wedge B$ if $U^A \subseteq U$ is sufficient for A and $U^B \subseteq U$ is sufficient for B.

Definition 74 Let $(Q, \mathcal{M})$ be an expansion proof for $U[\![\exists x A]\!]$. An expansion term t for x is *admissible* if there is no (free) parameter in t which is selected in Q.

Lemma 75 Let $(Q, \mathcal{M})$ be an expansion proof for $\exists x_1 A_1, \ldots, \exists x_n A_n$. Then there is an x_i and an expansion term t for $\exists x_i A_i$ such that t is admissible.

Proof: First note that not all nodes Q_i can be leaves. Let $T \neq \{\}$ be the set of expansion term occurrences for $\exists x_1 A_1, \ldots, \exists x_n A_n$ in Q. Since $<_Q$ has no cycles there is some t minimal in T. This means that for no $s \in T$, $s <_Q t$. t is admissible. Because t is minimal, no free parameter in t could be selected below any expansion term $s \in T$. But all selections in Q lie below some expansion term $s \in T$, because Q is of the form



where each Q_i has an expansion node at its root. Hence no free parameter of t is selected in Q, and t is admissible.

□

Next we will combine the basic operations of shallowing, creating leaves, and splitting to define (non-deterministic) functions which correspond to steps in a deduction in $\mathcal{H}$. Since we are first concerned with the translation from expansion proofs into $\mathcal{H}$, the first set of operations will be such that $f((Q,\mathcal{M})) \sqsubset (Q,\mathcal{M})$ for every operation f. This will ensure the termination of our translation procedure.

As we will see, most of these operations may be thought of as elimination rules.

Definition 76 Below we assume that $(Q,\mathcal{M})$ is an expansion proof. The fact that the result of these operations are again expansion proofs follows from the lemmas in Section 3.2, in particular from Lemmas 53, 56, and 58.

1. `double`. Let Q_A be top-level, $\mathcal{M}$-mated, and not a leaf of Q. Then we define $(R,\mathcal{N}) = \texttt{double}(Q_A,(Q,\mathcal{M})) = \texttt{shallow}(Q'_A,(Q',\mathcal{M}'))$, where $(Q',\mathcal{M}') = \texttt{crleaf}(Q_A,(Q,\mathcal{M}))$.

2. $\vee$E. Let $Q_{A\vee B}$ be top-level and single. Then $(R,\mathcal{N}) = \vee\mathrm{E}(Q_{A\vee B},(Q,\mathcal{M})) = \texttt{shallow}(Q_{A\vee B},(Q,\mathcal{M}))$.

3. $\forall$E. Let $Q_{\forall xA}$ be accessible and single. Then $(R,\mathcal{N}) = \forall\mathrm{E}(Q_{\forall xA},(Q,\mathcal{M})) = \texttt{shallow}(Q_{\forall xA},(Q,\mathcal{M}))$.

4. $\exists$E. Let $Q_{\exists xA}$ be accessible, single, and assume that it has a unique expansion term t and t is admissible. Then $(R,\mathcal{N}) = \exists\mathrm{E}(Q_{\exists xA},(Q,\mathcal{M})) = \texttt{shallow}(Q_{\exists xA},(Q,\mathcal{M}))$.

5. $\wedge$E. Let $Q_{A\wedge B}$ be top-level and single. Then for any U^A sufficient for A, we define $(R,\mathcal{N}) = \wedge\mathrm{E}_1(Q_{A\wedge B},U^A,(Q,\mathcal{M}))$ in stages.

   First let $(Q',\mathcal{M}') = \texttt{shallow}(Q_{A\wedge B},Q)$. Then we erase Q'_B and every tree in $Q|_{U''}$ to get $(R',\mathcal{N}')$, the expansion proof for $U',C_1,\ldots,C_n$. In order to show that it is an expansion proof, we have to show that Q'_B and every Q'_D for $D \in U''$ are inessential. That Q'_B is inessential follows by considering the full clauses in $(Q'',\mathcal{M}'') = \texttt{erase}(Q'_B,(Q',\mathcal{M}'))$: each of them already occurs in $\mathrm{fc}(Q)$ and must therefore be closed by $\mathcal{M}$ (remember the assumption that $Q_{A\wedge B}$ does not occur in $\mathcal{M}$). That $Q|_{U''}$ is inessential follows directly from the assumption that U^A and U^B separate $A\wedge B$.

   The algorithm amounts to obtaining $(R,\mathcal{N})$ for U^A, A as $(Q|_{U^A,A},\mathcal{M}|_{U^A,A})$.

   $\wedge\mathrm{E}_2$ is obtained completely analogously, using a U^B which is sufficient for B.

We now give a non-deterministic algorithm for translating an expansion proof for U into a deduction in $\mathcal{H}$ of U. We will show that this non-deterministic algorithm terminates strongly, that is, every possible computation terminates.

Algorithm 77 Given is a line L and an expansion proof $(Q, \mathcal{M})$ for L. The goal is to construct and $\mathcal{H}$-deduction for L. One step in the algorithm fills in the final inference (or inferences in Case 7) and creates a list of subgoals, each of which again consists of a line with associated expansion proof, and to which the algorithm is applied recursively. When the list of subgoals is empty, the line L can be an initial deduction. Thus more and more inferences are filled in until there are no remaining subgoals.

The algorithm is non-deterministic in the sense that it could produce many different $\mathcal{H}$-deductions, depending on the choices mentioned below.

We make the global restriction that a formula A such that Q_A is a leaf of Q is never active (see Definition 11) in an inference other than an initial deduction. Note that A may still be contracted.

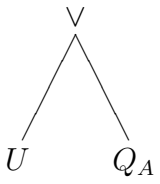
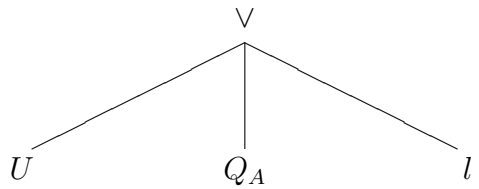
Also note that all our transformations on the expansion proof have the property that the critical node is directly below the root node, and therefore will be accessible.

The definition is by cases, depending on the input L and $(Q, \mathcal{M})$. The cases are not necessarily exclusive, which is one source of non-determinism in the algorithm.

1. $L = U, A$ such that Q_A appears in $\mathcal{M}$ and Q_A is not a leaf of Q. Then infer L by

$$\frac{U, A, A}{U, A}\, C$$

and let $(Q', \mathcal{M}') = \texttt{crleaf}(Q, \mathcal{M})$. We can then obtain an expansion proof $(R, \mathcal{N})$ for U, A, A by letting $(R, \mathcal{N}) = \texttt{shallow}(Q'_{A \vee A}, (Q', \mathcal{M}'))$. Graphically,

Modify $Q =$  to get $R =$ 

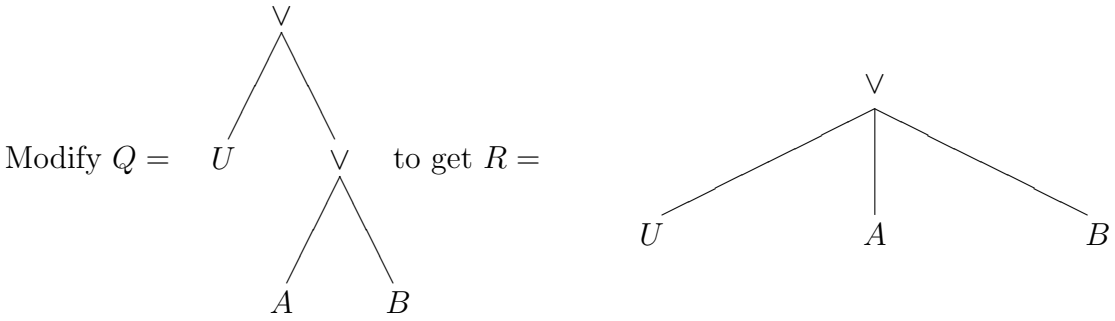
where l is a new leaf node with $l^S = A$. Modify $\mathcal{M}$ by replacing occurrences of Q_A by l.

2. $L = U, A, \sim A$ such that $(Q_A, Q_{\sim A}) \in \mathcal{M}$. Then let L be an initial deduction with active formula (occurrences) A and $\sim A$.

3. $L = U, A \vee B$. Infer L by

$$\frac{U, A, B}{U, A \vee B}\, \vee I$$

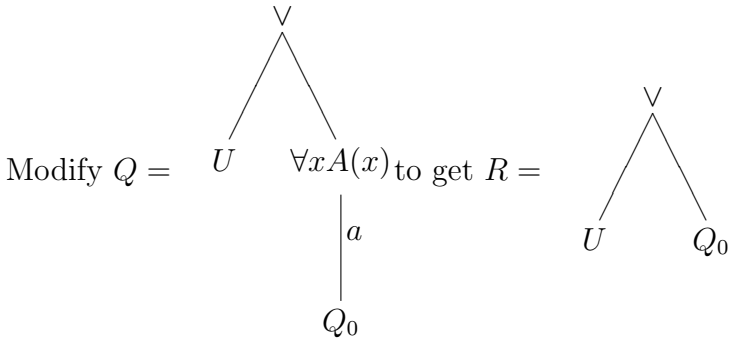
and obtain $(R, \mathcal{M}) = \texttt{shallow}(Q_{A \vee B}, (Q, \mathcal{M}))$. Graphically,



4. $L = U, \forall x A(x)$. Infer L by

$$\frac{U, A(a)}{U, \forall x A(x)} \forall I$$

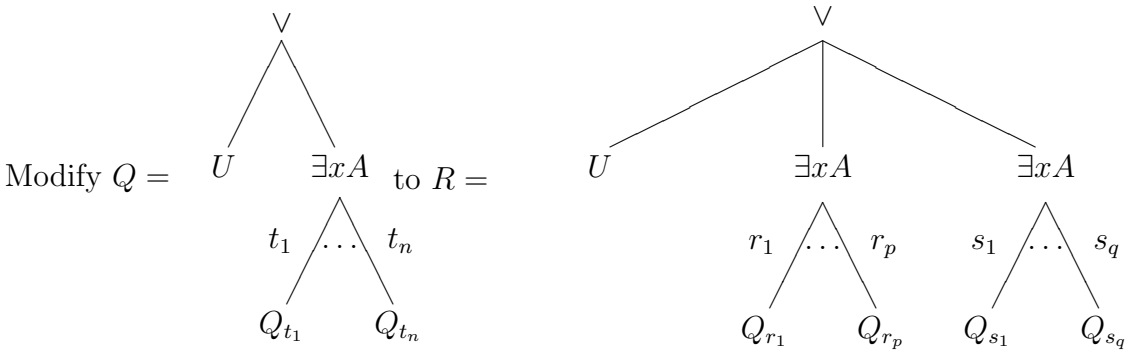
where a is the parameter selected for this occurrence of $A(a)$. Let $(R, \mathcal{N}) = \texttt{shallow}(Q_{\forall x A(x)}, (Q, \mathcal{M}))$. Graphically,



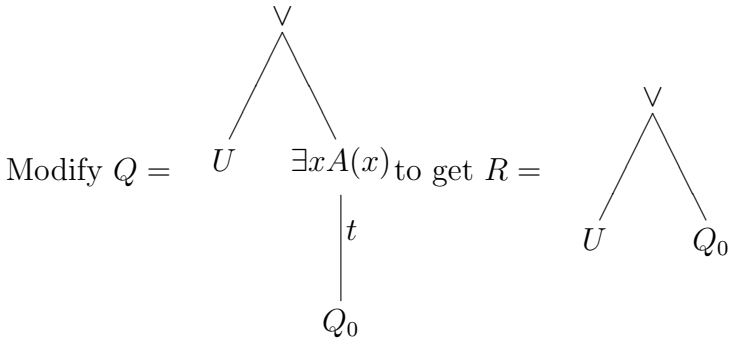
5. $L = U, \exists x A(x)$ and $\exists x A(x)$ has n, $n \geq 2$ successors in Q. Infer L by

$$\frac{U, \exists x A(x), \exists x A(x)}{U, \exists x A(x)} C$$

There are many transformations which still yield an expansion proof for $U, \exists x A, \exists x A$. In order to show that our algorithm terminates, we need to make some "progress". For any disjoint partition $\{r_1, \ldots, r_p\} \cup \{s_1, \ldots, s_q\} = \{t_1, \ldots, t_n\}$ we first define $(Q', \mathcal{M}') = \texttt{split}(Q_{\exists x A}, \{r_1, \ldots, r_p\}, \{s_1, \ldots s_q\}, (Q, \mathcal{M}))$. In order to get $(R, \mathcal{N})$ we merely eliminate the new node $Q'_{\exists x A \vee \exists x A}$ by shallowing. Let $(R, \mathcal{N}) = \texttt{shallow}(Q'_{\exists x A \vee \exists x A}, (Q', \mathcal{M}'))$. Graphically,



6. $L = U, \exists x A(x)$, and $\exists x A(x)$ has exactly one expansion term t, and t is admissible. Infer L by $\dfrac{U, A(t)}{U, \exists x A(x)}\,\exists I$ and let $(R, \mathcal{N}) = \texttt{shallow}(Q_{\exists x A(x)}, (Q, \mathcal{M}))$. Graphically,



7. $L = U, A \wedge B$. Let U^A, U^B separate $A \wedge B$. Furthermore, let $U^A \cap U^B = \{C_1, \ldots, C_n\}$, $U' = U^A - \{C_1, \ldots, C_n\}$, and $U'' = U^B - \{C_1, \ldots, C_n\}$. Then $L = U, A \wedge B = U', U'', C_1, \ldots, C_n, A \wedge B$ and we infer L by

$$\dfrac{\dfrac{U', C_1, \ldots, C_n, A \qquad U'', C_1, \ldots, C_n, B}{U', U'', C_1, \ldots, C_n, C_1, \ldots, C_n, A \wedge B}\,\wedge I}{U', U'', C_1, \ldots, C_n, A \wedge B}\,n \times C$$

   We obtain the the new expansion proofs $(R', \mathcal{N}')$ and $(R'', \mathcal{N}'')$ by $(R', \mathcal{N}') = \wedge\mathrm{E}_1(Q_{A \wedge B}, U^A, (Q, \mathcal{M}))$ and $(R'', \mathcal{N}'') = \wedge\mathrm{E}_2(Q_{A \wedge B}, U^B, (Q, \mathcal{M}))$.

Remark 78 In the previous algorithm, when we try to infer a conjunction and $U^A \cap U^B = \{\}$ we can infer it directly without any contractions.

Remark 79 The expansion proofs corresponding to the intermediate deductions in case 7 are not given. They are more "complicated" (with respect to $\sqsubset$) than $(Q, \mathcal{M})$ and are therefore not of interest. Once may construct them, however, by using the methods for constructing expansion proofs from $\mathcal{H}$-deductions (see Algorithm 87).

Theorem 80 (*Soundness*) If $(Q, \mathcal{M})$ is an expansion proof for U, then there is an $\mathcal{H}$ deduction for U.

Proof: In order to show that this algorithm always terminates, we use the relation $\sqsubset$ on pairs $(Q, \mathcal{M})$ which we have shown to be well-founded.

1. Whenever the algorithm terminates, we have a complete deduction for U. What we need to show here is that one of the cases always applies, until all remaining deductions are initial. But this follows from lemmas 72 and 75. Lemma 72 shows that there always are U^A and U^B separating $A \wedge B$ in case 7. Thus we only have to show that if all formula occurrences in a line start with existential quantifiers, one of them will be admissible. But this is just the contents of lemma 75.

2. The algorithm always terminates. Note that for each case and for each new expansion proof $(R, \mathcal{N})$, $(R, \mathcal{N}) \sqsubset (Q, \mathcal{M})$. Since $\sqsubset$ is well-founded, the algorithm must terminate.

□

3.4 Translating from $\mathcal{H}$ into Expansion Proofs

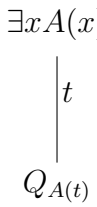
Generally there will be many $\mathcal{H}$ deductions corresponding to one expansion proof. One important property of our translations should be that they are inverses in the sense that if we build an $\mathcal{H}$ deduction from an expansion proof and then rebuild an expansion proof, they should be the same. This will not be exactly true; the resulting expansion trees will be the same, but the mating will only be a submating of the original mating. If we made sure that the original mating was minimal, then this retranslation property applies. Miller's MERGE and also the author's Merge [26] are significantly improved here in order for the translations to be inverses of each other.

This is one of the many places were we make strong use of the fact that we formulated $\mathcal{H}$ to operate on multisets rather than on sets of formulas. If one sees the relative complexity of the MERGE algorithm, it becomes clear that contraction is an interesting rule and it is therefore advantageous to make it explicit. Sets of formulas as used in many presentation of cut-elimination algorithms for first-order logic will not work, since they hide too much of the structure of the deduction in case two identical formulas are "accidentally" combined into one element of the set. Clearly, this is also impractical for implementations, where lines should probably be represented by lists.

We now define the "inverses" to the elimination operations of the previous section. In the next section we will make this correspondence precise.

Definition 81 Given below is an expansion proof $(Q, \mathcal{M})$ for U. We define some operations which generally make the expansion tree more complex and give them suggestive names.

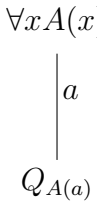
1. $\exists$I. Let $Q_{A(t)}$ be accessible such that no node above it is $\mathcal{M}$-mated. Then let $\exists\text{I}(Q_{A(t)}, \exists xA, (Q, \mathcal{M})) = (R, \mathcal{N})$, where $(R, \mathcal{N})$ is obtained from $(Q, \mathcal{M})$ by replacing $Q_{A(t)}$ by



   and $\mathcal{N} = \mathcal{M}$.

   If x is not free in A, we pick a new variable h to be t, h not selected in Q and not free in U.

2. $\forall$I. Let $Q_{A(a)}$ be accessible such that no node above it is $\mathcal{M}$-mated and let a be a parameter which is not free in Q or $\forall xA(x)$. Then $\forall\text{I}(Q_{A(a)}, \forall xA, (Q, \mathcal{M})) = (R, \mathcal{N})$, where $\mathcal{R}$ is obtained from $\mathcal{Q}$ by replacing $Q_{A(a)}$ by



   and $\mathcal{N} = \mathcal{M}$.

   If x does not appear free in A, we pick a new parameter a not free in U or selected in Q.

Lemma 82 If $(Q, \mathcal{M})$ is an expansion proof for $U[\![A(t)]\!]$, then $(R, \mathcal{N}) = \exists\text{I}(Q_{A(t)}, \exists xA(x), (Q, \mathcal{M}))$ is an expansion proof for $U[\![\exists xA(x)]\!]$.

Proof: The full clauses in R and Q are the same except for the addition of $R_{\exists xA(x)}$. But then every full clause is still spanned by $\mathcal{M}$.

$<_R$ is acyclic. Let a be a parameter selected below $\exists xA(x)$ in R. There may be expansion terms s_i, $t <_R^0 s_i$, but there is no term s such that $s <_R^0 t$. If $s <_R^0 t$ would hold, there had to be a variable b selected in R, and b free in t. But then also b free in $A(t)$ (otherwise t was selected to be a new variable), and hence b free in Q^S which contradicts the assumption that $(Q, \mathcal{M})$ is an expansion proof for $U[\![A(t)]\!]$.

□

Lemma 83 If $(Q, \mathcal{M})$ is an expansion proof for $U[\![A(a)]\!]$,
then $(R, \mathcal{N}) = \forall\text{I}(Q_{A(a)}, \forall xA(x), (Q, \mathcal{M}))$ is an expansion proof for $U[\![\forall xA(x)]\!]$.

Proof: Since $\mathrm{fc}(R)$ and $\mathrm{fc}(Q)$ agree except for the addition of $Q_{\forall x A(x)}$ to some full clauses, every full clause in R will still be spanned by $\mathcal{M}$.

Since a is not free in $U, \forall x A(x)$, a is a valid selection. Moreover, a could not have been selected in Q, since a occurs free in $A(a)$ or had been chosen not to be selected in Q. Thus a is selected in R only once.

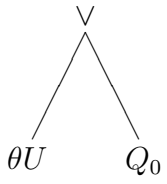
All other conditions on expansion proofs follow immediatly from the assumptions on $(Q, \mathcal{M})$. □

The significance of the following algorithm for merging expansion terms is in Lemma 85 and Theorem 99.

The problem with a straightforward, recursive merging of two expansions trees is that, if done in the wrong order, two initially distinct selected parameters could be identified later, thereby making two initially distinct expansion terms identical. If we would like Lemma 85 to hold, we have to make sure that this cannot happen. Earlier algorithms presented by Miller [23] and the author [26] did not take this problem into account and therefore resulted in much bigger, redundant expansion proofs.

We therefore introduce the auxiliary function `m`. `m` is a shallow merge: it stops on encountering expansion nodes. We maintain a global state in two variables: `mergelist` is a list of pairs of expansion term occurrence lists which still need to be examined and θ is a global substitution renaming the parameters.

Algorithm 84 Given is an expansion proof $(Q, \mathcal{M})$ and two subtrees Q_1 and Q_2 such that $Q_1^S = Q_2^S$ and both Q_1 and Q_2 are immediately below an accessible disjunction node or an expansion node. To compute $\texttt{merge}(Q_1, Q_2, (Q, \mathcal{M}))$ we first merge the expansion trees Q_1 and Q_2 with the `treemerge` algorithm given below. Let $(Q_0, \theta) = \texttt{treemerge}(Q_1, Q_2)$ and let

$R =$ 

Each node R_0 in R either existed in Q or was the result of applying `m` to nodes in Q or intermediate expansion trees created during the merge algorithm. In any case, each node in Q can be found somewhere in R, possibly after identification with other nodes. Replace each node in Q in $\mathcal{M}$ by this corresponding node in R to obtain $\mathcal{N}$. Return $(R, \mathcal{N})$ as the result of $\texttt{merge}(Q_1, Q_2, (Q, \mathcal{M}))$. The proof that $(R, \mathcal{N})$ will always be an expansion proof is in the proof of Lemma 85.

To compute $\texttt{treemerge}(Q_1, Q_2)$, where $Q_1^S = Q_2^S$:

1. Let `mergelist` $= \langle\rangle$ and θ the empty substitution.
2. Let $R = \mathtt{m}(Q_1^S, Q_2^S)$. This has a side-effect on `mergelist` and θ (`m` is defined below).

3. Try to select a pair $(S, T) \in \texttt{mergelist}$ and $s \in S$, $t \in T$ such that $\theta s = \theta t$.

4. If there is no such pair, return (R, θ).

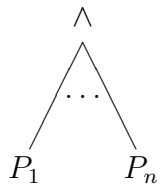
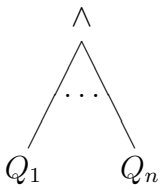
5. Otherwise, let E be the node where s and t are expansion terms and let E_s and E_t be the expansion trees immediately below s and t. Replace E in R by $\mathtt{m}(E_s, E_t)$. Go to 3.

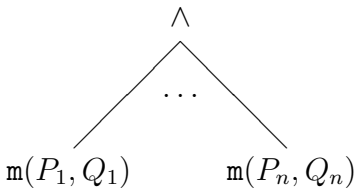
For the shallow merge $\mathtt{m}(P, Q)$ where $P^S = Q^S$ we consider several cases.

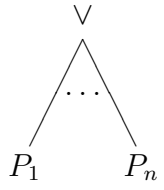
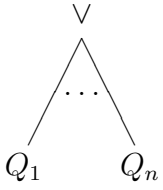
1. $P = l$ and $Q = m$ for leaves l and m. Then $\mathtt{m}(P, Q) = R$, where R is a new leaf with formula $\theta l = \theta m$.

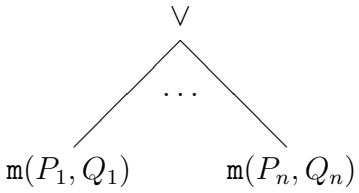
2. $P = l$ and Q is not a leaf. Then $\mathtt{m}(P, Q) = Q$.

3. $Q = l$ and P is not a leaf. Then $\mathtt{m}(P, Q) = P$.

4. $P =$  and $Q =$ 

Then $\mathtt{m}(P, Q) =$ 

5. $P =$  and $Q =$ 

Then $\mathtt{m}(P, Q) =$ 

6.

$$P = \begin{array}{c} \forall x A \\ | a \\ P_0 \end{array} \text{ and } Q = \begin{array}{c} \forall x A \\ | b \\ Q_0 \end{array}$$

Then we first pick a new parameter name c and update $\theta := \theta + [a \mapsto c] + [b \mapsto c]$. Next we return

$$\mathtt{m}(P, Q) = \begin{array}{c} \forall x A \\ | c \\ \mathtt{m}(P_0, Q_0) \end{array}$$

7.

$$P = \begin{array}{ccc} & \exists x A & \\ r_1 & \cdots & r_p \\ P_{r_1} & & P_{r_p} \end{array} \quad \text{and } Q = \begin{array}{ccc} & \exists x A & \\ s_1 & \cdots & s_q \\ Q_{s_1} & & Q_{s_q} \end{array}$$

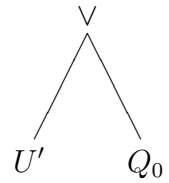
Add the pair $(\{r_1, \ldots, r_p\}, \{s_1, \ldots, s_q\})$ to `mergelist`. Return

$$\mathtt{m}(P, Q) = \begin{array}{ccc} & \exists x A & \\ r_1 & \cdots & s_q \\ P_{r_1} & & Q_{s_q} \end{array}$$

Lemma 85 Given an expansion proof $(Q, \mathcal{M})$ such that

$$Q = \begin{array}{ccccc} & & \vee & & \\ & \swarrow & | & \searrow & \\ U & & Q_1 & & Q_2 \end{array}$$

and $Q_1^S = Q_2^S$. Then there is an expansion proof $(R, \mathcal{N})$ such that

$R =$ 

$Q_0^S = Q_1^S = Q_2^S$ and $R|_{U'}$ differs from $Q|_U$ only through the renaming of some parameters.

Proof: We let $(R, \mathcal{N}) = \texttt{merge}(Q_1, Q_2, (Q, \mathcal{M}))$. It remains to show:

1. `merge` always terminates. This is easy to see, because the number of expansion terms in an expansion tree is finite.

2. $(R, \mathcal{N})$ is an expansion proof.

   2.1. No selected parameter is free in R. This is clear, since there are no free parameters to be selected in Q, and no parameters become free.

   2.2. $<_R$ is acyclic. We show this by induction on the number of applications of `m`. Hence the dependency relation will be acyclic for all the intermediate expansion trees constructed during `merge`.

   We define a sequence of relations $<_Q = <^0, <^1, \ldots, <^n = <_R$ such that each $<^i$, $1 \leq i \leq n$ is acyclic.

   Note first that $<^0$ is acyclic, since $<_Q$ is. `m` was not applied, we are done since $<_R = <_Q$. Otherwise the only step affecting the dependency relation is Step 5. How does it change the dependency relation when $\texttt{m}(E_s, E_t)$ is called? Every term selected below s previously, is now also selected below $r = \theta s = \theta t$ and vice versa. For the induction step, let us assume that $<^0, \ldots, <^i$ have already been defined and are acyclic. Now we define, according to the note above that $u <^{i+1} v$ iff $u <^i v$ or $u = r = \theta s = \theta t$ and $s <^i v$ or $t <^i v$. If this were to introduce a cycle, then $r <^{i+1} r$ would have to hold. Because of the way we extended $<^i$ this can only happen if $s <^i t$ or $t <^i s$. We also know that $\theta t = \theta s$. If also $t = s$ we are done, since then $s <^i t$ implies $s <^i s$, since s and t have the same free parameters. Otherwise let us use a for the (not necessarily unique) parameter which is selected below s and free in t, and b for the corresponding parameter in t. Since $\theta s = \theta t$, we know that $\theta a = \theta b$, which means that a and b were identified in a previous call to `m`. But this is a contradiction, since `m` is a shallow merge and had not been applied any node below s. Hence $<^{i+1}$ must be acyclic, and therefore $<_R$ by induction.

   2.3. $\mathcal{N}$ is clause-spanning on R. This is easy to see since any preimage of a full clause from $\mathrm{fc}(R)$ in $\mathrm{fc}(Q)$ must be spanned. That spanning pair, possibly disguised by some applications of `m` can still be found in $\mathcal{N}$.

□

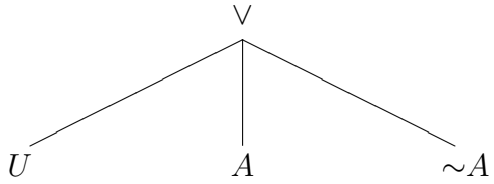
Remark 86 As mentioned before, there are some important properties of `merge` which are not covered by this lemma and are not necessary to show completeness, but we will need them later.

Algorithm 87 We assume we are given a deduction $\mathcal{D}$. We will show how to construct an expansion proof for $\mathcal{D}$, given expansion proofs for the premises of the last inference in $\mathcal{D}$.

The expansion proof for the premise will be called $(Q, \mathcal{M})$ for one-premise inference rules and $(Q_1, \mathcal{M}_1)$, $(Q_2, \mathcal{M}_2)$ in the case of $\wedge I$. The expansion proof for the conclusion will be $(R, \mathcal{N})$.

The definition is by cases, one case for each possible inference in $\mathcal{H}$.

1. $\mathcal{D} = U, A, \sim A$. Then $\mathcal{N} = \{(\sim A, A)\}$ and

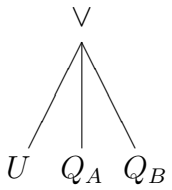
   $R =$ 

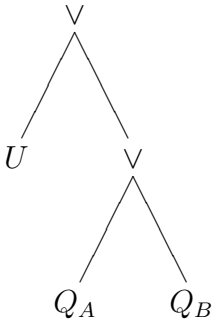
   Every formula of $Q|_U$ is a leaf directly below the root of R.

   Note that all inference rules including initial deductions must be indexed, so we know which $(A, \sim A)$ to enter into $\mathcal{N}$, even if $\mathcal{D} = A, \sim A, B, \sim B$.

2. $\mathcal{D} = \dfrac{U, A, B}{U, A \vee B} \vee I$.

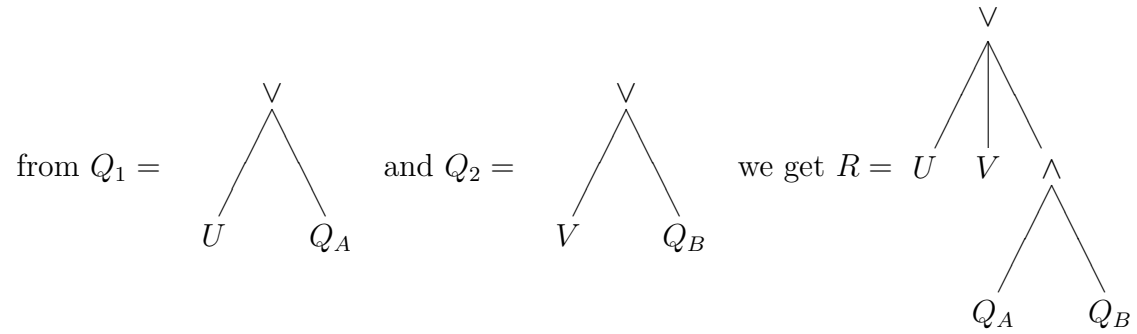
   Here $\mathcal{N} = \mathcal{M}$ and from

   $Q =$ 

   we pass to $R =$ 

3. $\mathcal{D} = \dfrac{U, A \qquad V, B}{U, V, A \wedge B} \wedge I$.

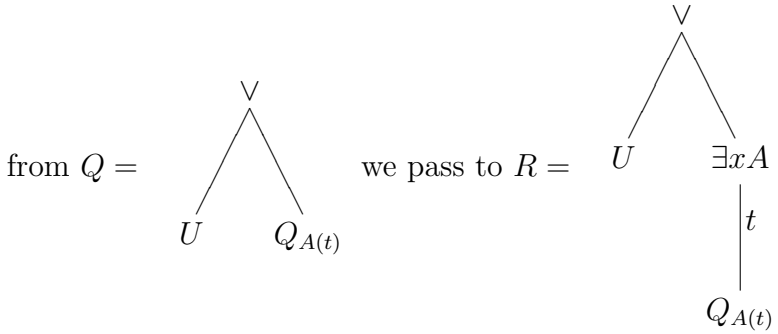
   Here $\mathcal{N} = \mathcal{M}_1 \cup \mathcal{M}_2$ and



In the new tree R we may have to rename the parameters selected for some universally quantified variables, to make sure that no free or selected parameter from one branch of the $\mathcal{H}$ deduction is selected in the other branch.

4. $\mathcal{D} = \dfrac{U, A(t)}{U, \exists x A(x)}\,\exists I$.

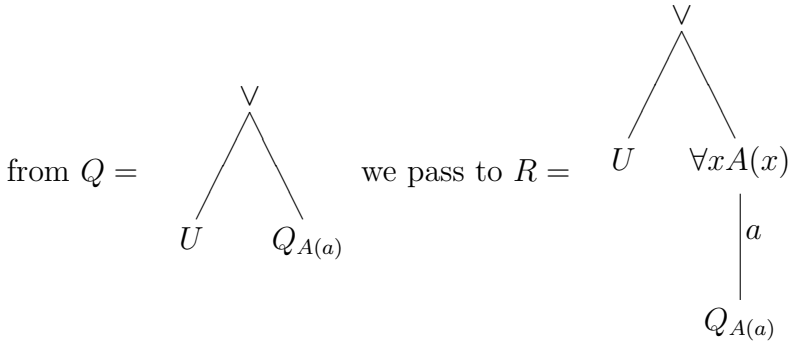
   Here we let $(R, \mathcal{N}') = \exists\mathrm{I}(Q_{A(t)}, \exists x A(x), (Q, \mathcal{M}))$. Graphically,



   and $\mathcal{N}' = \mathcal{M}$.

5. $\mathcal{D} = \dfrac{U, A(a)}{U, \forall x A(x)}\,\forall I$, a a parameter not free in U or $\forall x A(x)$.

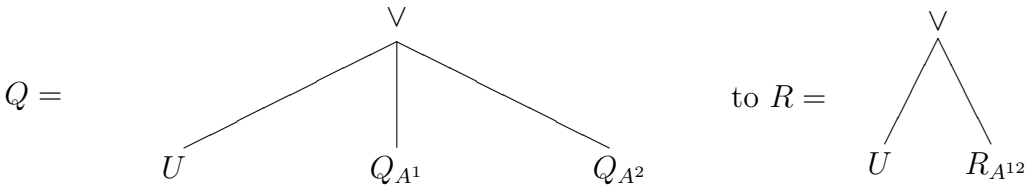
   Here we let $(R, \mathcal{N}') = \forall\mathrm{I}(Q_{A(a)}, \forall x A(x), (Q, \mathcal{M}))$. Graphically,



and $\mathcal{N} = \mathcal{M}$.

6. $\mathcal{D} = \dfrac{U, A^1, A^2}{U, A^{12}}\, C$.

   We apply the recursive `merge` algorithm to obtain an expansion tree $R_{A^{12}}$ for the occurrence of A^{12} in the conclusion. Let $(R, \mathcal{N}) = \texttt{merge}(Q_{A^1}, Q_{A^2}, (Q, \mathcal{M}))$. Then we pass from



Note that here $Q|_U$ is not always identical to $R|_U$ changes because of renaming of selected parameters and merging of identical expansion terms.

Theorem 88 (*Completeness*) For every cut-free deduction in $\mathcal{H}$ ending in U there exists an expansion proof $(Q, \mathcal{M})$ for U.

Proof: The proof follows from the correctness of Algorithm 87 which is easy to see, given the crucial Lemmas 82, 83, and 85. The proof is then a straightforward induction on the structure of the deduction in $\mathcal{H}$.

□

3.5 Properties and Refinements of the Translations

The properties established below show that our notion of expansion proof is "right". It contains the essential information necessary for the proof of a theorem. That many different deductions lead to the same expansion proof should not be surprising, since one can interchange inferences in a deduction without altering the "idea" of the proof.

We cannot expect to recover any deduction $\mathcal{D}$ which induces $(Q, \mathcal{M})$. In particular, those with unnecessary passive formulas which are later contracted cannot be obtained by our algorithm, if we would like to maintain strong termination.

Remark 89 If one actually carries out these transformations, redundancies will become obvious, even if unnecessary formulas are disregarded. This is not due to properties of the mating, but rather of expansions. A simple example should illustrate this.

Example 90

$$\forall x \exists y P x[f[fy]] \supset \forall a \exists u P a u \wedge \forall b \exists v P b[fv]$$

Here we need two instances for x: the parameters selected for a and b. However, only one instance is needed for each branch after the deduction has been split into two subdeductions (one of $\forall a \exists u P a u$ and one of $\forall b \exists v P b[fv]$).

Remark 91 One could have avoided the problem here if we had simplified the expansion proofs for the immediate subderivations of the application of $\wedge I$ by erasing some unnecessary expansion terms. This is not always possible, as can be seem by considering a slight elaboration of the example above: $\sim\forall x\exists y Px[f[fy]] \wedge \sim\forall s\forall t Pst \vee \forall a\exists u Pau \wedge \forall b\exists v Pb[fv]$.

Motivated by this example, we introduce a refinement to our Algorithm 77 which allows *expansion simplification* after each step. It will be clear that if we keep our expansions minimal, we only need to apply expansion simplification after Step 7.

Definition 92 Let Q be an expansion tree and let H be the set of all parameters which are free in some expansion term in Q, but not free or selected in Q. We rename all those parameters to one. Let $\theta = \{[a \mapsto h] : a \in H\}$, where h is some new parameter not selected or free in Q. Then θQ is again an expansion tree, and for any clause-spanning mating $\mathcal{M}$ on Q, $\mathcal{M}$ is again clause-spanning on θQ. We call θQ the result of *identifying free parameters*.

h plays the same role as 1 in Herbrand [16]. It is the seed of the Herbrand universe (in the first-order case).

After identifying free parameters, previously distinct expansion terms could now be identical.

Definition 93 Let $(Q, \mathcal{M})$ be an expansion proof, Q_0 an expansion node in Q with expansion term t. Furthermore, let Q' be the result of erasing Q_t from Q (note that Q_0 becomes a leaf, if t was the only expansion term of Q_0). We say the occurrence t is *unnecessary* iff $(Q', \mathcal{M}|_{Q'})$ is an expansion proof.

Definition 94 We say $(Q', \mathcal{M}')$ is an *expansion deletion* of $(Q, \mathcal{M})$ iff it is the result of erasing some unnecessary expansion term occurrences.

Definition 95 We say $(Q', \mathcal{M}')$ is the result of *expansion simplification* of $(Q, \mathcal{M})$ if it can be obtained from $(Q, \mathcal{M})$ by any number of free parameter identifications, deletion of unnecessary expansion term occurrences, or merging of identical expansion terms. We say $(Q', \mathcal{M}')$ is *expansion minimal* if no expansion simplification is possible.

Remark 96 It may be too costly to keep expansion proofs expansion minimal. Note, however, that the property of being expansion minimal is preserved by all steps in Algorithm 77 except step 7.

Remark 97 One may be content with the following faster check: Given an expansion term t. If no node below t occurs in $\mathcal{M}$, then t is unnecessary.

If we knew that the mating was minimal, the check of the previous remark would actually be sufficient.

Lemma 98 Let $(Q, \mathcal{M})$ be an expansion proof such that $\mathcal{M}$ is minimal, that is, no proper subset of $\mathcal{M}$ spans every full clause in Q. Then a subtree Q_0 immediately below an expansion or disjunction node of Q is necessary iff some node in Q_0 occurs in $\mathcal{M}$.

Proof: If no node in Q_0 occurs in $\mathcal{M}$, Q_0 is clearly unnecessary since every full clause in Q will still be spanned by $\mathcal{M}$. If some node Q_1 in Q_0 occurs in $\mathcal{M}$, Q_0 cannot be unnecessary. If it were, we could erase every pair with a node from Q_0 from $\mathcal{M}$ and still have a clause-spanning mating. This contradicts the minimality assumption on $\mathcal{M}$.

□

Theorem 99 Let $\mathcal{D}$ be the result of translating an expansion proof $(Q, \mathcal{M})$ into $\mathcal{H}$ with Algorithm 77 and let $(R, \mathcal{N})$ be the expansion proof obtained from $\mathcal{D}$ with Algorithm 87. Then $R = Q$ and $\mathcal{N} \subseteq \mathcal{M}$. Moreover, if expansion simplifications are allowed between steps in Algorithm 77, then $(R, \mathcal{N})$ is an expansion simplification of $(Q, \mathcal{M})$.

Proof: The proof is by induction on the measure used to show termination of the algorithm translating expansion proofs into $\mathcal{H}$. Thus we have to check for each of the cases in Algorithm 77 that the result of applying one (or several) steps in Algorithm 87 to obtain L leads to the same expansion proof and possibly a simpler mating than the one we started out with. Then the second part of the theorem follows by the straightforward fact that expansion simplification which would have to be applied between steps in the translation from $\mathcal{H}$-deductions into expansion proofs can be combined and applied summarily to the final deduction.

It can be immediately seen that the steps in Algorithm 77 and Algorithm 87 are inverses of each other except in the case of an $\wedge I$ followed by some contractions (as they are produced in Step 7 of Algorithm 77) and when the deduction is initial.

If the deduction is initial, the mating may be simplified, since the mating generated by Algorithm 87 has exactly one element, while original mating may have been redundant. Thus in this case $R = Q$ and $\mathcal{N} \subseteq \mathcal{M}$.

The remaing and critical case is contraction, and that is where earlier versions of the merge algorithm proposed by Miller [23] and improved by the author [26] were inadequate.

When applying $\wedge$E to the expansion proof, some selected parameters will appear in both new expansion proofs for the two subdeductions. Before the `merge`, they have to be renamed, since a parameter may only be selected once in an expansion proof. Let us assume we have expansion proofs $(Q, \mathcal{M})$ and $(Q', \mathcal{M}')$ which were duplicated during an $\wedge I$. Can parameters which were the same before the $\wedge I$ be identified by `merge`? Yes. The proof is by induction on the imbedding relation $\prec_Q$.

Let P be the set of selected parameters in Q which would need to be identified to obtain the original expansion proof, P' be the corresponding parameters in Q', and Q and Q' are being merged.

First note that a pair of renamed selected parameters a in Q and a' in Q' which are minimal in P (and P') with respect to $\prec_Q$ (and $\prec_{Q'}$, respectively) will be identified because either they

are not below any expansion terms, or they are below identical expansion terms. If the expansion terms would differ, then they would differ by a parameter name, say c and c'. But then $c \prec_Q a$, which is a contradiction. Let us use $\min P$ to denote the minimal elements in a set P with respect to $\prec_Q$.

But this merging of the minimal selected parameters potentially adds new pairs to `mergelist`, because expansion terms previously differing by one of those parameter names now become identical. Hence now all parameters in $\min\{P - \min P\}$ and $\min\{P' - \min P'\}$ will be identified when the expansion terms above them are selected from the `mergelist`.

Since $\prec_Q$ and $\prec_{Q'}$ are acyclic, this procedure will eventually identify all parameters in P with the corresponding parameters in P'.

□

3.6 A Cut Elimination Algorithm for Expansion Proofs

In this section we investigate the following question: given expansion proofs for $A \supset B$ and $B \supset C$, how do we *construct* an expansion proof for $A \supset C$?

One obvious algorithm would use the translations provided in the previous sections. We could translate the expansion proofs for $A \supset B$ and $B \supset C$ into $\mathcal{H}$ deductions, then eliminate the cut of those two deductions with the cut-elimination procedure given in Chapter 2, and translate the resulting deduction back into an expansion proof.

This naive procedure is unsatisfactory for several reasons. First of all it is impractical in that many steps in the translation are unnecessary. Secondly we will see that we can substantially improve this algorithm (at least in practical terms) by doing the elimination directly on expansion proofs. There are fewer rules to consider at each step in the algorithm, and moreover, we do not have to split the deduction apart as one necessarily must to when using the reduction which permutes a cut of a passive formula occurrence upwards past an $\wedge I$. We will point out another important advantage later, but roughly speaking we do not need to commit ourselves to a particular, heuristically chosen translation, but can choose translation steps depending on what is appropriate in order to eliminate the cut.

This algorithm will suppport our contention that expansion proofs embody essential proof information for analytic proofs. Many different $\mathcal{H}$ deductions may correspond to it and we will fully exploit this in the algorithm. In intuitionistic logic, natural deductions seem to fill that role of a universal analytic proof system.

The improvements over the cut-elimination algorithm in $\mathcal{H}$ are such that the termination of this new procedure still remains a conjecture. It does not follow from the Conjecture 34. Again, we can only give a termination proof when restricting ourselves to first-order expansion trees.

The algorithm is open for further improvements. For example, one can make use of properties of matings for the propositional calculus and eliminate the cut formula directly when it is purely propositional, that is, does not contain any selection or expansion nodes.

Now let us assume we have two expansion proofs,

$$(Q, \mathcal{M}) \text{ for } U, A$$

and

$$(R, \mathcal{N}) \text{ for } \sim A, V$$

We give an algorithm to construct an expansion proof

$$(S, \mathcal{O}) \text{ for } U, V.$$

Immediately we are faced with a problem. When investigating $\mathcal{H}$-deductions it was simple to introduce a new non-analytic inference rule Cut. Or rather, we had the reverse situation of having to show (constructively) that the rule of Cut may be eliminated from every $\mathcal{H}$ deduction.

Expansion proofs are inherently analytic. There is no easy way to represent a non-analytic inference in them. Therefore we will have to find a new language in which to express non-analytic proofs involving expansion trees. The critical step will of course be to represent the problem outlined above: given two expansion proofs $(Q, \mathcal{M})$ for U, A and $(R, \mathcal{N})$ for $\sim A, V$, we would like to represent somehow a non-analytic expansion proof for U, V.

3.6.1 Expansion Developments

In analogy to $\mathcal{H}$ deductions, we will represent these non-analytic expansion proofs as trees whose leaves are (usual) expansion proofs. Let us call these trees *expansion developments*. Initially we will have the inference rules of $\forall$I, $\exists$I, and `merge`.

One should keep in mind that in the descriptions below expansion developments are often written out completely and contain no variables for arbitrary subdevelopments as was the case for $\mathcal{H}$-deductions where $\mathcal{D}$ often stood for an arbitrary deduction with the correct final line. This is possible since the initial expansion deductions (axioms or leaves) are expansion proofs.

We will use $\mathcal{Q}$ and $\mathcal{R}$ as meta-variables representing expansion developments. Every line in an expansion development will have an assertion which intuitively corresponds to the formula proven. In case we have an expansion proof, the assertion is merely its shallow formula.

Definition 100 (*Expansion Developments*) These are very similar in flavor to $\mathcal{H}$ deductions, but we do not need the propositional rules. This results in a simplification of the elimination procedure, since the splitting which needed to be done in the case of $\wedge I$ for expansion deductions is no longer needed. The definition proceeds inductively.

1. Every expansion proof $(Q, \mathcal{M})$ is an expansion development for Q^S, the shallow formula of Q.

2. If $\mathcal{Q}$ is an expansion development with accessible formula (occurrence) $A(t)$ then

$$\dfrac{\begin{matrix}\mathcal{Q}\\ U[\![A(t)]\!]\end{matrix}}{U[\![\exists x A(x)]\!]}\ \exists \mathrm{I} : \exists x A(x)$$

   is an expansion development.

3. If $\mathcal{Q}$ is an expansion development with an accessible formula (occurrence) $A(a)$, then

$$\dfrac{\begin{matrix}\mathcal{Q}\\ U[\![A(a)]\!]\end{matrix}}{U[\![\forall x A(x)]\!]}\ \forall \mathrm{I} : \forall x A(x)$$

   is an expansion development.

4. If $\mathcal{Q}$ is an expansion development with accessible formula (occurrence) $A^1 \vee A^2$ then

$$\frac{\overset{\mathcal{Q}}{U[\![A^1 \vee A^2]\!]}}{U[\![A]\!]}\,\mathtt{merge} A^1, A^2$$

is an expansion development.

5. If $\mathcal{Q}$ and $\mathcal{R}$ are expansion developments then

$$\frac{\overset{\mathcal{Q}}{U, A} \qquad \overset{\mathcal{R}}{\sim A, V}}{U, V}\,\mathtt{elim}$$

is an expansion development.

3.6.2 Conversions between Expansion Developments

The first set of reduction rules will look rather strange, because they have no analogue in $\mathcal{H}$ deductions. These reduction rules are possible, because expansion proofs can "encode" every cut-free proof directly.

After these definitions we make use of the results in the previous sections and view them in a new light.

Definition 101 *Initial expansion reductions.* In the definition below, $(Q, \mathcal{M})$ is an expansion proof with $Q^S = U$. Note that in each conversion, the resulting expansion deduction is initial, or, in other words, the right hand side of each reduction is an expansion proof. We write $\mathcal{Q} \underset{i}{\Longrightarrow} \mathcal{R}$ if $\mathcal{R}$ is an initial expansion reduction of $\mathcal{Q}$. The following are the reduction rules:

1. initial-$\exists$I.

$$\frac{(Q, \mathcal{M})}{U[\![\exists x A(x)]\!]}\,\exists\mathrm{I} \qquad \underset{i}{\Longrightarrow} \qquad \exists\mathrm{I}(Q_{A(t)}, \exists x A(x), (Q, \mathcal{M}))$$

2. initial-$\forall$I.

$$\frac{(Q, \mathcal{M})}{U[\![\forall x A(x)]\!]}\,\forall\mathrm{I} \qquad \underset{i}{\Longrightarrow} \qquad \forall\mathrm{I}(Q_{A(a)}, \forall x A(x), (Q, \mathcal{M}))$$

3. initial-`merge`.

$$\frac{(Q, \mathcal{M})}{U[\![A]\!]}\,\mathtt{merge} A^1, A^2 \qquad \underset{i}{\Longrightarrow} \qquad \mathtt{merge}(Q_{A^1}, Q_{A^2}, (Q, \mathcal{M}))$$

Definition 102 (*Permutative expansion reductions*) These closely correspond to the permutative reductions of $\mathcal{H}$-deductions. We write $\mathcal{Q} \underset{p}{\Longrightarrow} \mathcal{R}$ if $\mathcal{R}$ is the result of a permutative expansion reduction of $\mathcal{Q}$. The reduction rules are:

1. Last inference is initial. In this case the cut reduces to an initial deduction.

2. Last inference is $\exists I$.

$$\dfrac{\dfrac{\overset{\mathcal{Q}}{U[\![A(C)]\!], X}}{U[\![\exists y A(y)]\!], X}\exists I \qquad \overset{\mathcal{R}}{\sim X, V}}{U[\![\exists y A(y)]\!], V}\,\texttt{elim} \quad \underset{p}{\Longrightarrow} \quad \dfrac{\dfrac{\overset{\mathcal{Q}}{U[\![A(C)]\!], X} \qquad \overset{\mathcal{R}}{\sim X, V}}{U[\![A(C)]\!], V}\,\texttt{elim}}{U[\![\exists y A(y)]\!], V}\exists I$$

3. Last inference is $\forall I$.

$$\dfrac{\dfrac{\overset{\mathcal{Q}}{U[\![A(a)]\!], X}}{U[\![\forall y A(y)]\!], X}\forall I \qquad \overset{\mathcal{R}}{\sim X, V}}{U[\![\forall y A(y)]\!], V}\,\texttt{elim} \quad \underset{p}{\Longrightarrow} \quad \dfrac{\dfrac{\overset{\mathcal{Q}}{U[\![A(a)]\!], X} \qquad \overset{\mathcal{R}}{\sim X, V}}{U[\![A(a)]\!], V}\,\texttt{elim}}{U[\![\forall y A(y)]\!], V}\forall I$$

   If a happens to be free in V, replace a by a new parameter b everywhere in $\mathcal{R}$.

4. Last inference is a `merge`.

$$\dfrac{\dfrac{\overset{\mathcal{Q}}{U[\![A^1 \vee A^2]\!], X}}{U[\![A]\!], X}\,\texttt{merge} \qquad \overset{\mathcal{R}}{\sim X, V}}{U[\![A]\!], V}\,\texttt{elim} \quad \underset{p}{\Longrightarrow} \quad \dfrac{\dfrac{\overset{\mathcal{Q}}{U[\![A^1 \vee A^2]\!], X} \qquad \overset{\mathcal{R}}{\sim X, V}}{U[\![A^1 \vee A^2]\!], V}\,\texttt{elim}}{U[\![A]\!], V}\,\texttt{merge}$$

The essential reductions will have some restrictions pertaining to them. Still, as we will see in a later lemma, whenever we have an application of `elim` to two expansion proofs, either a permutative reduction, a merge reduction, or an essential reduction will apply.

Definition 103 (*Essential expansion reductions*) We write $\mathcal{Q} \underset{e}{\Longrightarrow} \mathcal{R}$ if one can obtain $\mathcal{R}$ from $\mathcal{Q}$ by an *essential expansion reduction* as defined by the following cases.

1. One of the eliminated formulas, say A, corresponds to a leaf Q_A in its respective expansion proof. Then we replace the `elim` immediately by applying `chain` (see Algorithm 112) yielding an expansion proof (an initial expansion development). Thus

$$\dfrac{\begin{array}{cc}(Q, \mathcal{M}) & (R, \mathcal{N}) \\ U, A & \sim A, V\end{array}}{U, V}\,\texttt{elim} \quad \underset{e}{\Longrightarrow} \quad \texttt{chain}(Q_A, R_{\sim A}, (Q, \mathcal{M}), (R, \mathcal{N}))$$

2. One of the cut formulas is a single conjunction. Then there are several possible reductions. For every choice of U^A and U^B sufficient for A and B respectively, we have two possible reductions.

$$\dfrac{\begin{array}{cc}(Q,\mathcal{M}) & (R,\mathcal{N})\\ U, A\wedge B & \sim A\vee\sim B, V\end{array}}{U,V}\,\texttt{elim} \quad \underset{e}{\Longrightarrow} \quad \dfrac{\begin{array}{cc}\mathcal{Q}_2 & \dfrac{\begin{array}{cc}\mathcal{Q}_1 & (R,\mathcal{N})\\ U^A, A & \sim A,\sim B, V\end{array}}{U^A,\sim B, V}\,\texttt{elim}\\ U^B, B & \end{array}}{\dfrac{U^B, U^A, V}{U,V}\,\texttt{merge}}\,\texttt{elim}$$

where $\mathcal{Q}_1 = \wedge\mathrm{E}_1(Q_{A\wedge B}, U^A, (Q,\mathcal{M}))$ and $\mathcal{Q}_2 = \wedge\mathrm{E}_2(Q_{A\wedge B}, U^B, (Q,\mathcal{M}))$. Also, the reduction to the symmetric expansion development where B is eliminated first is allowed.

3. One of the cut formulas is a single existentially quantified formula with exactly one expansion term t. Let a be the parameter selected in the other eliminated formula.

$$\dfrac{\begin{array}{cc}(Q,\mathcal{M}) & (R,\mathcal{N})\\ U, \exists x A(x) & V, \forall x\sim A(x)\end{array}}{U,V}\,\texttt{elim} \quad \underset{e}{\Longrightarrow} \quad \dfrac{\begin{array}{cc}(Q',\mathcal{M}') & (R',\mathcal{N}')\\ U, A(t) & \sim A(t), V\end{array}}{U,V}\,\texttt{elim}$$

where $(Q',\mathcal{M}') = \exists\mathrm{E}(Q_{\exists x A(x)}, (Q,\mathcal{M}))$ and $(R',\mathcal{N}') = [a\mapsto t](\forall\mathrm{E}(R_{\forall x A(x)}, (R,\mathcal{N})))$.

Definition 104 (*Merge reductions*) We write $\mathcal{Q}\underset{m}{\Longrightarrow}\mathcal{R}$ if $\mathcal{R}$ is the result of a merge reduction in $\mathcal{Q}$. There are two possible situations in which merge reductions may be applied. These restrictions are important, since they must be general enough to ensure that some reduction will always apply and restrictive enough to ensure termination of reduction sequences.

The reduction

$$\dfrac{\begin{array}{cc}\dfrac{\begin{array}{c}(Q,\mathcal{M})\\ U, A^1, A^2\end{array}}{U, A^{12}}\,\texttt{merge} & (R,\mathcal{N})\\ & \sim A, V\end{array}}{U,V}\,\texttt{elim} \quad \underset{m}{\Longrightarrow} \quad \dfrac{\begin{array}{cc}\dfrac{\begin{array}{cc}(Q,\mathcal{M}) & (R,\mathcal{N})\\ U, A^1, A^2 & \sim A, V\end{array}}{U, V, A^1}\,\texttt{elim}^2 & (R,\mathcal{N})\\ & \sim A, V\end{array}}{\dfrac{U,V,V}{U,V}\,\texttt{merge}}\,\texttt{elim}^1$$

may be applied when either

1. $Q_{A^{12}}$ is an expansion node and Q_{A^1} has exactly one expansion term t, and t is admissible.

   or

2. $Q_{A^{12}}$ is not a leaf, but $\mathcal{M}$-mated and Q_{A^1} is not a leaf and not $\mathcal{M}$-mated, and Q_{A^2} is a leaf and $\mathcal{M}$-mated.

3.6.3 The Elimination Algorithm

Definition 105 Let $\mathcal{Q}$ and $\mathcal{R}$ be expansion developments. We define

1. $\overset{*}{\underset{i}{\Longleftrightarrow}}$ for the symmetric, reflexive, and transitive closure of $\underset{i}{\Longrightarrow}$.
2. $\Longrightarrow$ for the union of $\underset{p}{\Longrightarrow}$, $\underset{m}{\Longrightarrow}$, and $\underset{e}{\Longrightarrow}$.
3. $\mathcal{Q}\overset{*}{\Longleftrightarrow}\mathcal{R}$ if there exists a sequence $\mathcal{Q} = \mathcal{Q}_0 \overset{*}{\underset{i}{\Longleftrightarrow}} \mathcal{Q}_1 \Longrightarrow \mathcal{Q}_2 \overset{*}{\underset{i}{\Longleftrightarrow}} \cdots \mathcal{Q}_{n-2} \Longrightarrow \mathcal{Q}_{n-1} \overset{*}{\underset{i}{\Longleftrightarrow}} \mathcal{Q}_n = \mathcal{R}$. We call such a sequence a *strong reduction sequence*.

Theorem 106 For every expansion development $\mathcal{Q}$ for U there exists an expansion proof $(R, \mathcal{N})$ for U.

Proof: Follows from the completeness theorem for expansion proofs and Lemma 108. Note, however, that the completeness proof is not constructive, as long as the completeness proof for $\mathcal{H}$-deductions remains non-constructive.

□

Theorem 107 For any expansion development $\mathcal{Q}$ with an application of the `elim` rule, there exist expansion developments $\mathcal{Q}'$ and $\mathcal{R}$ such that $\mathcal{Q} \overset{*}{\underset{i}{\Longleftrightarrow}} \mathcal{Q}' \Longrightarrow \mathcal{R}$.

Proof: The essential lemmas are given below. From Lemma 108 we know that any `elim` with no other `elim` inferences above it is $\overset{*}{\underset{i}{\Longrightarrow}}$-equivalent to one where both premisses are expansion proofs. Once we have an application of `elim` to two initial expansion developments, we have to show that permutative, essential, or merge reductions can be made applicable through initial expansion conversions of the premisses. This is the content of Lemma 109.

□

Lemma 108 Let $\mathcal{Q}$ be an expansion development without an application of the `elim` rule. Then there exists an expansion proof (initial expansion development) $(R, \mathcal{N})$ such that $Q\overset{*}{\underset{i}{\Longrightarrow}}(R, \mathcal{N})$.

Proof: Follows directly from the definition of expansion developments and initial expansion reductions, given the proofs of Theorems 80 and 88.

□

Lemma 109 Given an expansion development $\mathcal{S} = \texttt{elim}(A, \sim A, (Q, \mathcal{M}), (R, \mathcal{N}))$. Then either

1. there is an $\mathcal{S}'$ such that $\mathcal{S} \Longrightarrow \mathcal{S}'$,

   or

2. there is a $\mathcal{Q}$ such that $\mathcal{Q} \underset{i}{\Longrightarrow} (Q, \mathcal{M})$ and an $\mathcal{S}'$ such that $\texttt{elim}(A, \sim A, \mathcal{Q}, (R, \mathcal{N})) \Longrightarrow \mathcal{S}'$, or

3. there is a $\mathcal{R}$ such that $\mathcal{R} \underset{i}{\Longrightarrow} (R, \mathcal{N})$ and an $\mathcal{S}'$ such that $\texttt{elim}(A, \sim A, (Q, \mathcal{M}), \mathcal{R}) \Longrightarrow \mathcal{S}'$.

Proof: One has to distinguish cases for A. If it is a disjunction or conjunction the corresponding conversion is directly applicable. The only critical step is to show that one can always convert the expansion proof for an existentially quantified formula such that there will be an admissible expansion term. This is achieved through the algorithm `seek` (see Algorithm 117) whose correctness is proven in Lemma 118.

□

Throughout the remainder of this section we assume that we are considering the expansion development $\mathcal{S} = \texttt{elim}(A, \sim A, (Q, \mathcal{M}), (R, \mathcal{N}))$, where $(Q, \mathcal{M})$ is an expansion proof for U, A and $(R, \mathcal{N})$ is an expansion proof for $V, \sim A$.

Lemma 110 If A is inessential, $(Q, \mathcal{M})$ is an expansion proof for U, V.

Lemma 111 If A is a leaf of an expansion proof $(Q, \mathcal{M})$ then either

1. A is inessential, or

2. A is $\mathcal{M}$-mated to at least one other leaf in Q.

In the case analysis below we could therefore assume that A is essential. This is not necessary, but checking whether A or $\sim A$ are essential may significantly improve the performance of the algorithm.

Algorithm 112 $\texttt{chain}(Q_A, R_{\sim A}, (Q, \mathcal{M}), (R, \mathcal{N}))$

We assume that we have an expansion proof $(Q, \mathcal{M})$ for U, A such that Q_A is a leaf of Q and Q_A appears in $\mathcal{M}$. Given an expansion proof $(R, \mathcal{N})$ for $V, \sim A$, we would like to find an expansion proof $(S, \mathcal{O})$ for U, V.

We make the further assumption that Q_A is $\mathcal{M}$-mated only to leaves of Q.

1. Let $\texttt{mated}(Q_A) = \{k : (Q_A, k) \in \mathcal{M}\}$.

2. Let S be the expansion tree for U, V obtained by joining $Q|_U$ with $R|_V$ and then replacing every leaf $k \in \texttt{mated}(Q_A)$ by a copy of $R_{\sim A}$. We call R'_k any new occurrence of a subtree R' in $R_{\sim A}$ where it replaces k.

3. Let $\mathcal{N}' = \{(R_0, R_1) : R_0 \in R_{\sim A} \text{ and } (R_0, R_1) \in \mathcal{N}\}$.

4. For every $k \in \texttt{mated}(Q_A)$, let $\mathcal{N}'_k$ be the result of replacing every $R' \in R_{\sim A}$ in $\mathcal{N}'$ by R'_k.

5. Let $\mathcal{O} = \mathcal{M}|_U \cup \mathcal{N}|_V \cup \bigcup_{k \in \mathtt{mated}(Q_A)} \mathcal{N}'_k$.

6. Return $(S, \mathcal{O})$.

Lemma 113 If the conditions for calling `chain` are satisfied, then

$$(S, \mathcal{O}) = \mathtt{chain}(Q_A, R_{\sim A}, (Q, \mathcal{M}), (R, \mathcal{N}))$$

is an expansion proof for U, V.

Proof: All conditions except for the following two are immediate.

1. $\mathcal{O}$ is clause-spanning on S. Let c be a full clause in S, and let $c|_U$ and $c|_V$ be the restrictions to U and V, respectively. If the preimage c' on Q is spanned by a pair entirely $c'|_U$, this pair will still span c. Similarly for the preimage c'' on R. If not, then c' must be spanned by a pair (Q_A, k). Let us define the path d on R to behave on V like c and on $R_{\sim A}$ like c on k. By assumption, d is spanned by a pair (l', l''). By Step 4 in the `chain` algorithm, a (renamed) copy of this pair will be in $\mathcal{O}$ and close c.

2. $<_S$ is acyclic. Since no parameter selected in R should appear in an expansion term in Q (rename it otherwise) and vice versa, no cycle can be introduced. One obtains $<_S$ by joining $<_R$ and $<_Q$ and only adding pairs of the form $s <_S t$ for s in Q and t in R.

□

In order to defined `seek` precisely, we first define an extension $\lessdot\!\!\lhd_Q$ of the dependency relation $<_Q$.

Definition 114 For two nodes Q_0 and Q_1 in an expansion tree Q we say $Q_0 \lhd Q_1$ if Q_0 is (strictly) above Q_1. We extend this notion to occurrences of expansion terms t and s by defining $t \lhd s$ if $Q_t \lhd Q_s$, where Q_t and Q_s are the nodes below the arcs labeled with t and s respectively. Similarly for selected parameters. We write $Q_0 \trianglelefteq_Q Q_1$ if $Q_0 \lhd_Q Q_1$ or $Q_0 = Q_1$.

Definition 115 Let $\lessdot\!\!\lhd_Q$ be the transitive closure of $<_Q \cup \lhd_Q$. ($\lhd_Q$ restricted to expansion terms.)

Lemma 116 If $<_Q$ is acyclic, so is $\lessdot\!\!\lhd_Q$.

Proof: Note that $\lhd_Q$ is acyclic. Thus, if we had a cycle, we must have one of the form

$$t_1 <_Q t_2 \lhd_Q t_3 <_Q t_4 \lhd_Q t_5 <_Q \cdots \lhd_Q t_1$$

But when $t_2 \lhd_Q t_3$, then every parameter selected below t_3 is also selected below t_2. Thus if $t_3 <_Q s$, then also $t_2 <_Q s$. Hence we can collapse the cycle to one of the form

$$t_1 <_Q t_2 \lhd t_1$$

But then $t_1 <_Q t_2 \lhd_Q t_1 <_Q t_2$, so we must also have $t_2 <_Q t_2$ which was excluded by assumption. Hence $\ll\!\lhd_Q$ does not contain a cycle.

□

If the elimination node $Q_{\exists xA(x)}$ of an `elim` inference is existential, but none of the expansion terms is admissible, we have to "break down" the rest of Q until one or more of the expansion terms becomes admissible. It is a consequence of the Soundness Theorem 80 that this is always possible.

Here we use initial conversions to expand an initial expansion deduction into one which ends with an inference on the side-formula of the `elim`. The guiding light is the expansion proof $(Q, \mathcal{M})$, and the task is to remove some of the non-determinism from algorithm 77.

The basic idea of the algorithm is to pick some expansion term s which we can make accessible, without first making other expansion terms of $Q_{\exists xA(x)}$ accessible. Then we work our way backwards, making the terms it depends on accessible, and expanding them.

The result of `seek` will be an expansion deduction $\mathcal{Q}'$, such that s is accessible in every leaf $(Q', \mathcal{M}')$ of $\mathcal{Q}'$. It is important to note that $\mathcal{Q}' \overset{*}{\underset{i}{\Longleftrightarrow}} \mathcal{Q}$.

Again, we will allow some non-determinism in `seek`, which may be used to the advantage of an implementation which would use some heuristics to decide how to proceed.

Algorithm 117 `seek`$(s, (Q, \mathcal{M}))$

We assume we are given an expansion proof $(Q, \mathcal{M})$ and an expansion term occurrence s in Q. We would like to apply a sequence of initial conversions to obtain $\mathcal{Q}'$, such that s is admissible in the initial expansion development of $\mathcal{Q}'$, call it $(Q', \mathcal{M}')$.

We define the relation $\underset{\texttt{seek}(s)}{\longrightarrow}$, from may be though of as describing a non-deterministic algorithm for computing $\texttt{seek}(s, (Q, \mathcal{M}))$. For simplicity let us assume that no non-leaves occur in $(Q, \mathcal{M})$. If they do occur one must apply merge expansion reductions (see Definition 104) to make the selection nodes Q_{b_0} in step 6 and the expansion nodes Q_{t_0} in step 33.1 accessible before applying ∀E and ∃E, respectively.

1. Let $S = \{t : t \ll\!\lhd_Q s\}$. Pick an arbitrary t_0 such that t_0 is minimal in S.
2. Let $P = \{a : a \text{ free in } t_i \text{ and } a \text{ selected in } Q\}$.
3. If $P = \{\}$ then
   - 3.1. if t_0 is the sole expansion term of Q_{t_0}, let $(Q, \mathcal{M}) \underset{\texttt{seek}(s)}{\longrightarrow} \exists\mathrm{E}(Q_{t_0}, (Q, \mathcal{M}))$.

3.2. if t_0 is not the sole expansion term of Q_{t_0} then let $(Q, \mathcal{M}) \underset{\texttt{seek}(s)}{\longrightarrow} \texttt{split}(Q_{\exists x A(x)}, \{t_0\}, T - \{t_0\}, (Q, \mathcal{M}))$.

4. Let $P' = \{b :$ there is an $a \in P$ such that $b \trianglelefteq_Q a\}$
5. Pick a b_0 minimal (with respect to $\trianglelefteq_Q$).
6. Then $\mathcal{Q} \underset{\texttt{seek}(s)}{\longrightarrow} \forall \mathrm{E}(Q_{b_0}, (Q, \mathcal{M}))$.

Lemma 118 For any expansion proof $(Q, \mathcal{M})$ and any expansion term occurrence s in $(Q, \mathcal{M})$, $\texttt{seek}(s, (Q, \mathcal{M}))$ is an expansion proof in which s is admissible.

Proof: One can easily see that the algorithm must terminate (use $\sqsubset$). Also, in case 6, Q_{b_0} is accessible. Moreover, in case 33.1 it is always true that Q_{t_0} is admissible and accessible. Hence by lemma 53, if $\mathcal{Q} \underset{\texttt{seek}(s)}{\longrightarrow} \mathcal{R}$, then $\mathcal{R}$ is an expansion developement.

□

Theorem 119 For every first-order expansion development $\mathcal{Q}$ there exists a first-order expansion proof $(R, \mathcal{N})$ such that $\mathcal{Q} \overset{*}{\Longleftrightarrow} (R, \mathcal{N})$. Moreover, every strong reduction sequence terminates in an `elim`-free expansion development.

Proof: From the general Theorem 107 it follows that permutative, merge, or essential reductions will always be applicable.

What remains to be shown is termination. The argument is somewhat similar to the termination argument for $\mathcal{H}$-deductions, except that we do not need to consider the contraction rank. The termination order is by a triple induction. If $\mathcal{Q} \underset{p}{\Longrightarrow} \mathcal{R}$, then $(Q, \mathcal{M}) \sqsubset (R, \mathcal{N})$ for the expansion proofs $(Q, \mathcal{M})$ and $(R, \mathcal{N})$ of the premisses of the `elim` rule before and after the reduction. Thus, either the `merge` will be eliminated, or one reaches the point where a merge or essential conversion is applicable. If $\mathcal{Q} \underset{m}{\Longrightarrow} \mathcal{R}$ then $Q_{A^1} \sqsubset Q_{A^{12}}$ and $Q_{A^2} \sqsubset Q_{A^{12}}$ where $(Q, \mathcal{M})$ and A are as in Definition 104. The conditions on the merge reduction imply that Q_{A^1} does not increase in complexity (with respect to $\sqsubset$) when eliminating $\texttt{elim}^2$. Finally, if $\mathcal{Q} \underset{e}{\Longrightarrow} \mathcal{R}$ then the complexity (number of connectives and quantifiers) of the cut formula is decreased. This is where the argument breaks down for general, higher-order expansion developments.

□

Conjecture 120 Every strong reduction sequence starting from an arbitrary expansion development terminates in an `elim`-free expansion development.

Chapter 4

Adding Equality

In this chapter we will extend the language $\mathcal{L}$ to $\mathcal{L}^=$ which contains a primitive symbol for equality. The inference system $\mathcal{H}$ will also be extended to a system $\mathcal{H}^=$ which contains a rule of substitution and the axiom schema asserting reflexivity of equality. We then extend Algorithm 77 to translate expansion proofs where equality is treated as defined into $\mathcal{H}^=$ deductions where equality is primitive. This translation requires a precomputation stage during which we transform the given expansion proof into a more suitable one satisfying certain requirements for the translation proper.

Since a non-extensional equality is definable in our type theory, what is the purpose of this exercise? The reasons are two-fold.

Firstly, we wish to support our claim that expansion proofs are in a sense a universal structure for representing proofs in classical logic. Since there are only two connectives and two quantifiers in expansion proofs, we should be able to show how other definable notions may be represented in expansion proofs. The answer is simple: through their definition. But the problem of the disparity between the logical systems poses a new problem: how do we make the connection between primitive notions of a deductive system and the defined notions of expansion proofs? For simple definable connectives like implication, the answer is also simple. We will discuss negation and implication in some detail in Chapter 6. For a complex and inherently polymorphic notion like equality, the answer will be less straightforward.

Secondly, treating equality as defined results in an escalation of the order of the formulas. In first-order systems, equality cannot be defined. Much effort has been devoted to treating equality in a first-order setting in automated theorem proving. For a survey see Fages and Huet [10] or Siekmann [32]. We will not attempt here to extend such work to theorem proving in higher-order logic. Rather, we provide a translation algorithm from expansion proofs where equality is treated as a defined notion to deductions where equality is considered primitive. When this algorithm is applied to first-order theorems involving equality, we will obtain first-order deductions.

The translation from cut-free $\mathcal{H}^=$-deductions into expansion proofs is a straightforward extension of Algorithm 87 and we will not present it here.

The full system $\mathcal{H}^=$ does not have the cut-elimination property (see Theorem 136). How-

ever, if we add one more rule of inference (the dual to the substitution rule which replaces the left-hand side for the right-hand side instead of vice versa) then the first-order fragment will have the cut-elimination property. We give an improved translation procedure into this modified system $\mathcal{H}^*$ which will always produce normal deductions. This also requires proving that the quantifiers coming from instantiating an equality in the original theorem need only be instantiated with literals when one searches for an expansion proof. Since there are potientially very many instances of these higher-order quantifiers, this is a practically very significant result.

4.1 A system $\mathcal{H}^=$ with equality

In order to obtain a system similar in spirit to $\mathcal{H}$, but with a the notion of equality primitive in it, we add, for each type α a new symbol $\doteq$ and the following formulas of type o

$$A_\alpha \doteq B_\alpha.$$

Note that we do not regard "$\doteq$" itself as a proper formula, which simplifies the exposition below. Since we do not assume extensionality, this means that the equality predicate for a given type α, itself in type $o\alpha\alpha$, is not the interpretation of $\doteq_{o\alpha\alpha}$ (since it does not have an interpretation), but of $[\lambda x \lambda y.x \doteq y]$. This avoids the complication arising from the fact that, if "$\doteq_{o\alpha\alpha}$" were an interpreted symbol, we could not prove it to be equal to $[\lambda x \lambda y.x \doteq y]$. This is analogous to the decision we made when making $\wedge, \vee$, etc., not constants of the given type.

Definition 121 The language $\mathcal{L}^=$ is obtained from $\mathcal{L}$ by adding one clause to Definition 2

11. $A_\alpha \doteq B_\alpha$ for arbitrary formulas A_α and B_α.

We also add the inference rule of substitution and some new initial deductions.

Definition 122 $\mathcal{H}^=$ is obtained by adding the following rules of inference to $\mathcal{H}$ (see Definition 11)

6. Equality Axioms

$$\frac{}{U, A \doteq A}\, R\doteq : A \doteq A$$

7. Substitution Inference

$$\frac{U, \mathbf{P}A}{U, \sim[A \doteq B], \mathbf{P}B}\, S\doteq : A \doteq B, \mathbf{P}A, \mathbf{P}B$$

Here $\mathbf{P}$ is an arbitrary term (of correct type). A way of expressing the rule of substitution would be to say that we obtain $\mathbf{P}B$ from $\mathbf{P}A$ by substituting zero or more occurrences of A in $\mathbf{P}A$ by B. Avoiding clashes of variables names is taken care of through the λ-conversion rules. Let us call this system $\mathcal{H}^=$.

We do not copy $\sim A \doteq B$ to the premiss of the inference for the same reason that we did not copy the $\exists x A$ in the rule of $\exists$I. We need to isolate several "uses" of the same formula (the equality in one case, the existentially quantified formula in the other) into single uses and contraction. This is necessary because of the complexity of contraction when applied to expansion proofs (see the `merge` algorithm 84). This is not just a property of expansion proofs, when one considers the complexities introduced into cut-elimination algorithms by the possibility of contraction.

4.2 Some Properties of Defined Equality

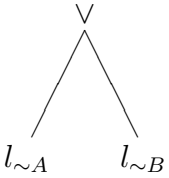
We will treat definitions in expansion trees by enlarging our notion of equivalence class of formulas. Two formulas are now equivalent if they are equal up to $\lambda^{\sim}$-conversion or instantiation of definitions.

Thus when we write $A \doteq B$ in the context of expansion trees, this is the same as writing out the instantiated form $\forall \mathbf{q}[\mathbf{q}A \supset \mathbf{q}B]$.

Lemma 123 Given an expansion proof $(Q, \mathcal{M})$ of U. Then we can find an expansion proof $(R, \mathcal{N})$ of U such that

1. All non-leaf nodes in Q are single, i. e., are not $\mathcal{M}$-mated.
2. All leaf nodes in Q are literals.

Proof: First we show that we can find a $(Q', \mathcal{M}')$ such that all non-leaf nodes in $(Q', \mathcal{M}')$ are single. We show how to eliminate nodes which are above no other $\mathcal{M}$-mated non-leaf node in Q. The proof is by induction on the complexity of the expansion tree below an $\mathcal{M}$-mated node Q_0

1. Q_0 is a leaf. This is the basis of the induction in which the assertion holds trivially.
2. Q_0 is a conjunction node with shallow formula $A \wedge B$. Then each of its mates is a disjunction node Q^i with shallow formula $\sim A \vee \sim B$ or a leaf l. We replace any such leaf l by $Q^i =$ 

, where $l_{\sim A}$ and $l_{\sim B}$ are new leaves with formulas $\sim A$ and $\sim B$, respectively. The result is still an expansion proof with the same deep and shallow formulas. Now we replace every pair $(Q_0, Q^i) \in \mathcal{M}$ by two new pairs $(Q_A, Q^i_{\sim A})$ and $(Q_B, Q^i_{\sim B})$ to obtain $\mathcal{M}'$.

   It is easy to check that $\mathcal{M}'$ is spans every full clause on the new expansion tree Q'. Also the complexity of the expansion tree below the two new $\mathcal{M}$-mated nodes Q_A and Q_B is strictly smaller that the complexity of the expansion tree below Q_0 (they are proper subtrees).

The same holds for any disjunction node Q^i which was $\mathcal{M}$-mated to Q_0. As for leaves, which were $\mathcal{M}$-mated: they are replaced by new leaves in $\mathcal{M}'$ thus neither increasing nor decreasing the complexity.

3. Q_0 is a disjunction node with shallow formula $A \vee B$. This case is symmetric to the previous one.

4. Q_0 is an expansion node with shallow formula $\exists x A(x)$. First we replace each of its mates which are leaves by $Q^i = \forall x{\sim}A(x)$, where all the a^i are new and distinct from each other.

$$\begin{array}{c} \Big| \, a^i \\ \Big| \\ l_{\sim A(a^i)} \end{array}$$

Again we obtain an expansion proof with the same shallow formula. We may now assume that all nodes mated to Q_0 are selection nodes. For any pair $(Q_0, Q^i) \in \mathcal{M}$, $1 \leq i \leq n$ we now add a new expansion term a^i, where a^i is the parameter selected for Q^i.

To get $\mathcal{M}'$ we replace every pair $(Q_0, Q^i) \in \mathcal{M}$ by $(Q_{a^i}, Q_{\sim A(a^i)})$, where Q_{a^i} is the node below Q_0 corresponding to one of the new expansion term a^i.

It is easy to check the $(Q', \mathcal{M}')$ satisfies the conditions of an expansion proof and has the same shallow formula as $(Q, \mathcal{M})$. As in the previous case, every new $\mathcal{M}$-mated node either came from and remains a leaf or has a strict subtree of previously $\mathcal{M}$-mated nodes below it.

5. Q_0 is a selection node. This is symmetric to the case of an expansion node.

In order to see that we can always achieve that the shallow formulas of leaves are literals, we can continue the algorithm outlined above even if all $\mathcal{M}$-mated nodes are leaves. The proof of termination then relies on the complexity of the leaf formulas, rather than on the complexity of the expansion proof.

□

Definition 124 Given an expansion proof $(Q, \mathcal{M})$ with leaf Q_0 such that its shallow formula Q_0^S is not a literal. Then let $\texttt{deepen}(Q_0, (Q, \mathcal{M}))$ be the operation defined in the proof of Lemma 123 which transforms Q_0 into an interior node of the tree, its kind depending on the main connective of Q_0^S.

For further development in this section we will need to extend the imbedding relation $\prec_Q$, which was defined on the selected parameters of an expansion tree Q (see Definition 37.) This extension is analogous to the extension of the dependency relation $<_Q$ defined on expansion term occurrences (see Definition 36) as was necessary for the definition of `seek` (see Definitions 114 and 115).

Definition 125 For two nodes Q_0 and Q_1 in an expansion tree Q we say $Q_0 \prec\!\!\!\cdot\, Q_1$ if Q_0 is (strictly) below Q_1. We extend this notion to selected parameters a and b by defining $a \prec\!\!\!\cdot\, b$ if $Q_a \prec\!\!\!\cdot\, Q_b$, where Q_a and Q_b are the nodes below the arcs labeled with a and b respectively.

Definition 126 Let $\prec\!\!\!\triangleleft_Q$ be the transitive closure of $\prec_Q \cup \triangleleft_Q$.

Lemma 127 If $\prec_Q$ is acyclic, so is $\prec\!\!\!\triangleleft_Q$.

Proof: We omit the proof. It is dual to the the proof given for Lemma 116.

□

A common operation on expansion trees we will need in the rest of this chapter is an inverse to `merge`. For simplicity of the description we will show how to duplicate an expansion term in a given expansion node. In order to create more than two instances of the given term, one can iterate the construction.

Of course there are many trivial inverses of `merge`. As we will see below, we have additional constraints which require a non-trivial algorithm to achieve the duplication.

The next lemma will be required in the proof of Theorem 132. Intuitively it says that if an expansion term t contains an occurrence of a parameter a which is selected in another conjunct, then we can modify the expansion proof in such a way that t no longer contains a. Moreover, this can be done without increasing the depth of the expansion proof.

If there is a conjunction immediately below the disjunctive root such that t occurs in one conjunct, and a is selected in another, then the lemma is trivial. We simply find two sets U^t and U^a sufficient for the conjunct with t and a, respectively. Then applying $\wedge$E followed by $\wedge I$ (see case 3 of Algorithm 87) yields an expansion proof with the desired property.

Definition 128 Given an expansion proof $(Q, \mathcal{M})$ with an expansion node $Q_{\exists yB}$ and two expansion terms s and t of $Q_{\exists yB}$. We say $\mathcal{M}$ is *strongly clause-spanning* with respect to s and t if the following two conditions hold. Given a full clause p,

1. the clause obtained by erasing all nodes l which are below s in Q is spanned $\mathcal{M}$, and
2. the clause obtained by erasing all nodes l which are below t in Q is spanned by $\mathcal{M}$.

This condition is much stronger than clause-spanning.

Lemma 129 Given an expansion proof $(Q, \mathcal{M})$ with an expansion node $Q_{\exists yB}$ and an expansion term s of $Q_{\exists yB}$. Then there is an expansion proof $(Q', \mathcal{M}')$ with equal depth and two copies, s^1 and s^2, of s such $\mathcal{M}'$ is strongly clause-spanning with respect to s^1 and s^2.

Proof: To obtain Q' we duplicate some expansion terms in Q. The construction we will use in the proof of Lemma 130 is similar, but somewhat more complicate. Here we let

$$\begin{aligned}
\texttt{dupa} &= \{a\} \cup \{b : b \prec\!\!\!\triangleleft_Q a\} \\
\texttt{dupt} &= \{t\} \cup \{s : \text{ there is a parameter } b \in \texttt{dupa} \text{ such that } b \text{ is free in } s\} \\
\texttt{maxdupt} &= \{s \in \texttt{dupt} : \text{ there is no } r \in \texttt{dupt} \text{ such that } r \lhd_Q s\} \\
\theta' &= \{[b \mapsto b'] : b \in \texttt{dupa}\}
\end{aligned}$$

We then duplicate every term $t \in \texttt{maxdupt}$, applying θ' to one copy. Note that t must be in $\texttt{maxdupt}$ (otherwise $\prec_Q$ would be cyclic). This achieves an expansion tree Q'.

Next we have to construct the mating $\mathcal{M}'$. Informally we try to mate the new copies of nodes in Q' "parallel" to the mates of the old copies. Let $\texttt{dupl}$ be the set of nodes in Q which were duplicated, that is,

$$\texttt{dupl} = \{l : l \in Q \text{ and there is a } t \in \texttt{maxdupt} \text{ such that } l \prec\!\!\!\prec_Q t\}$$

Then we define $\mathcal{M}'$ by cases. Let $(l, k) \in \mathcal{M}$ be given. Then

1. If both $l \in \texttt{dupl}$ and $k \in \texttt{dupl}$, add the two pairs (l, k) and (l', k') to $\mathcal{M}'$.
2. If $l \in \texttt{dupl}$, but $k \notin \texttt{dupl}$ we add (l, k) and (l', k) to $\mathcal{M}'$. Note that this is possible, since k cannot contain any parameter $b \in \texttt{dupa}$. If it did, k would have to be in $\texttt{dupl}$.
3. $l \notin \texttt{dupl}$ and $k \notin \texttt{dupl}$. Then add (l, k) to $\mathcal{M}'$.

Clearly, $\mathcal{M}$ is mating, that is, mated nodes have complementary shallow formulas, and mated nodes share at least one clause.

It remains to show that $(Q', \mathcal{M}')$ is an expansion proof. This means we have to verify that $\mathcal{M}'$ is clause-spanning on Q' and that $<_{Q'}$ is acyclic.

We name the nodes in Q' identically to the nodes in Q, except for the nodes $l \in \texttt{dupl}$. The node which was identically copied we continue to call l, the copy to which we applied θ' we call $\theta' l$. $\theta'\texttt{dupl}$ is the set of all such nodes.

1. $\mathcal{M}'$ is clause-spanning on Q'. Actually, we would like to show more, namely that $\mathcal{M}'$ is strongly clause-spanning with respect to t^1 and t^2. The two cases are symmetric, so it is enough to show that $\mathcal{M}'$ closes any path after erasing all nodes below t^2.

   Let p' be a full clause on Q'. We erase all nodes below t^2 and define the preimage p of p' to contain l if p' contains $\theta' l$. By assumption, $\mathcal{M}$ was clause-spanning on Q, and therefore p is spanned by a pair (l, k).

   1.1. $l \in \texttt{dupl}$ and $k \in \texttt{dupl}$. By definition, (l, k) and (l', k') are both in $\mathcal{M}'$. Hence (l', k') spans p'.

   1.2. $l \in \texttt{dupl}$, but $k \notin \texttt{dupl}$. Then (l, k) and (l', k) are in $\mathcal{M}'$ and p' is spanned by (l', k).

   1.3. $l \notin \texttt{dupl}$ and $k \notin \texttt{dupl}$. Then $(l, k) \in \mathcal{M}'$ spans p'.

   This exhausts all possible cases. Hence $\mathcal{M}'$ is strongly clause-spanning on Q' with respect ot t^1 and t^2.

2. $<_{Q'}$ is acyclic.

   This part of the proof follows a common pattern. We assume that there is a cycle in $<_{Q'}$ and show that there must have been a cycle in $<_Q$, which is a contradiction.

   We name the expansion terms in Q' identically to the expansion terms in Q, except for $t \in \texttt{dupl}$. The node which was identically copied we continue to call t, the copy to which we applied θ' we call $\theta' t$. $\theta'\texttt{dupt}$ is the set of all such nodes. We let u_i be the preimage of u'_i on Q, that is u'_i if $u'_i \notin \theta'\texttt{dupt}$ and u_i if u'_i is of the form $\theta' u_i$. Note that some expansion terms may have the same preimage.

   Assume there is a cycle $u'_1 <^0_{Q'} u'_2 <^0_{Q'} \cdots <^0_{Q'} u'_n = u'_1$.

   We claim that $u_1 <^0_Q u_2 <^0_Q \cdots <^0_Q u_n = u_1$.

   The proof is by cases on the left-hand and right-hand side of a related pair $u' <^0_{Q'} v'$. We show that for the preimages u and v, $u <^0_Q v$.

   2.1. $u' \in \theta'\texttt{dupt}$, $v' \in \theta'\texttt{dupt}$. By definition, there is a parameter b selected below u', free in v'. But then b must be some b' (since θ' is applied below u') and hence b is free in v. Of course b is also selected below u, so $u <^0_Q v$.

   2.2. $u' \in \theta'\texttt{dupt}$, $v' \notin \theta'\texttt{dupt}$. This is not possible, since a parameter selected below u' must be some b', that is, of the form $\theta' b$ for $b \in \texttt{dupa}$.

   2.3. $u' \notin \theta'\texttt{dupt}$. A parameter c selected below $u' \notin \theta'\texttt{dupt}$ is also selected below u, unless c is selected below some $t \in \texttt{exempt}$. In either case, $u <^0_Q v$, because the preimage of c is selected below u and free in v.

□

Lemma 130 Given an expansion proof $(Q, \mathcal{M})$ with a selection node $Q_{\forall xA}$ with selected parameter a. Let $Q_{\exists yB}$ be an expansion node such that there is no full clause on Q containing both $Q_{\forall xA}$ and $Q_{\exists yB}$. Let $\texttt{dupa}$ be $\{a\} \cup \{b : b \prec_Q a$ but b not selected below $Q_{\exists yB}\}$. Then there is an expansion proof with equal depth and just one copy of $Q_{\exists yB}$ such that no expansion term below $Q_{\exists yB}$ contains a parameter $b \in \texttt{dupa}$.

Proof: First we describe how to obtain the expansion tree, call it Q'. Intuitively, we would like to make two copies of expansion terms containing the parameter a, rename a to a' in one copy rename all occurrences of a in $Q_{\exists yB}$ from a to a'. As described this is an unsound procedure, because duplicating an expansion term also duplicates all parameters which are selected below it. Since no parameter may be selected more than once, which would require renaming of those parameters. This of course again may require duplication of expansion term etc.

Here is where the imbedding relation comes to the rescue. If we let

$$\texttt{dupa} = \{a\} \cup \{b : b \prec\!\!\!\prec_Q a\}$$

then $\texttt{dupa}$ is precisely the set of parameters of which we will have two copies in Q'.

Now we have to determine which expansion terms need to be duplicated. First note that we do not want to duplicate any expansion term for $Q_{\exists yB}$ or below $Q_{\exists yB}$. Moreover we do not want to duplicate $Q_{\exists yB}$ itself. If we were to duplicate an expansion term above $Q_{\exists yB}$, we would also duplicate $Q_{\exists yB}$. Therefore we let

$$\texttt{above} = \{t : t \text{ is an expansion term, there exists a } b \in \texttt{dupa} \text{ such that } b \text{ is free in } t, \text{ and } t \text{ is above } Q_{\exists yB}\}$$

If `above` is empty, we exclude all expansion terms below $Q_{\exists yB}$ from duplication. Otherwise, there must be a unique topmost term in `above`. In this case we exclude that term and all expansion terms below it from duplication. If `above` is non-empty, let t_{max} be the topmost $t \in \texttt{above}$. Note that in this case, t_{max} is not above $Q_{\forall xA}$, because $b \in \texttt{dupa}$ is free in t_{max}. If it were also above a, then $a \prec_Q b$, but also $b \prec_Q a$ (since $b \in \texttt{dupa}$), and hence $\prec_Q$ would have a cycle.

$$\texttt{exempt} = \begin{cases} \{t : t \trianglelefteq_Q Q_{\exists yB}\} & \text{if } \texttt{above} = \{\} \\ \{t : t \trianglelefteq_Q t_{max}\} & \text{if } \texttt{above} \neq \{\} \end{cases}$$

Then the expansion terms we will have to duplicate are the ones which contain free occurrences of parameters in `dupa`, but are not exempt.

$$\texttt{dupt} = \{t : \text{ there is a parameter } b \in \texttt{dupa} \text{ such that } b \text{ is free in } t\} - \texttt{exempt}$$

Of course duplicating the topmost expansion terms in `dupt` will automatically create copies of expansion terms below it. Hence

$$\texttt{maxdupt} = \{t \in \texttt{dupt} : \text{ there is no } s \in \texttt{dupt} \text{ such that } s \trianglelefteq_Q t\}$$

Thus `maxdupt` contains only the expansion terms in `dupt` which have no other terms in `dupt` above them. We will need to rename some selected parameters. Let

$$\theta' = \{[b \mapsto b'] : b \in \texttt{dupa}\}$$

Now we are finally prepared to describe Q'. For every expansion term $t \in \texttt{maxdupt}$ we add an identical expansion term t to its expansion node. Then we apply θ' to one copy of t and Q_t, the expansion tree below it. Moreover we apply θ' to every expansion term $t \in \texttt{exempt}$ and the tree below every $t \in \texttt{exempt}$.

We show that Q' is an expansion tree.

1. $Q_{\forall xA}$ was not copied. Assume it were. Then there must have been an expansion term t above $Q_{\forall xA}$ such that some $b \in \texttt{dupa}$ free in t. But then $a \lessdot_Q b$, but also by definition of `dupa`, $b \lessdot_Q a$, so $\lessdot_Q$ would have a cycle. By Lemma 127 and the assumption that Q is an expansion proof, this is a contradiction.

2. $Q_{\exists yB}$ was not copied. This follows from the construction of `exempt`.

3. No parameter is selected twice. This held for Q and now still holds since θ' was applied to all new copies of selection nodes. It remains to show that any duplicated selection node is in the domain of θ'. But this follows from the construction of `dupa` and `dupt`. If there is a parameter b selected below one of the expansion terms in `maxdupt`, then $b \prec\!\!\!\prec_Q a$, and therefore $b \in$ `dupa`. But `dupa` is the domain of θ'.

Next we have to construct the mating $\mathcal{M}'$. Informally we try to mate the new copies of nodes in Q' "parallel" to the mates of the old copies. Let `dupl` be the set of nodes in Q which were duplicated, that is,

$$\texttt{dupl} = \{l : l \in Q \text{ and there is a } t \in \texttt{maxdupt} \text{ such that } l \prec_Q t\}$$

$$\texttt{exemptl} = \{l : l \in Q \text{ and there is a } t \in \texttt{exempt} \text{ such that } l \prec_Q t\}$$

For each $l \in$ `dupl` we have two nodes in Q'. Let the occurrence which resulted from applying θ' be l', the other one l.

The definition of $\mathcal{M}'$ is by cases. Let $(l, k) \in \mathcal{M}$ be given. Then

1. If both $l \in$ `dupl` and $k \in$ `dupl`, add the two pairs (l, k) and (l', k') to $\mathcal{M}'$.

2. If $l \in$ `dupl`, but $k \notin$ `dupl` we distinguish two subcases

   2.1. $k \in$ `exemptl`. Add (l', k) to $\mathcal{M}'$. Note that is is possible, since we applied θ' to such a k.

   2.2. Otherwise, add (l, k) and (l', k) to $\mathcal{M}'$. Note that this is possible, since k cannot contain any parameter $b \in$ `dupa`. If it did, k would be either in `dupl` or in `exemptl`.

3. $\lambda \notin$ `dupl` and $k \notin$ `dupl`. Then add (l, k) to $\mathcal{M}'$. Since no $l \in$ `exemptl` has a path in common with $Q_{\forall xA}$ and therefore cannot be mated to some k below $Q_{\forall xA}$ (see Definition 41), l and k will still be complementary.

Clearly, $\mathcal{M}$ is mating, that is, mated nodes have complementary shallow formulas, and mated nodes share at least one clause.

It remains to show that $(Q', \mathcal{M}')$ is an expansion proof. This means we have to verify that $\mathcal{M}'$ is clause-spanning on Q' and that $<_{Q'}$ is acyclic.

We name the nodes in Q' identically to the nodes in Q, except for the nodes $l \in$ `dupl`. The node which was identically copied we continue to call l, the copy to which we applied θ' we call $\theta' l$. θ'`dupl` is the set of all such nodes.

1. $\mathcal{M}'$ is clause-spanning on Q'.

   Let p' be a full clause on Q'. We define the preimage p of p' on Q by cases:

   1.1. p' contains nodes in `exemptl`. Informally, we force p to behave on Q as p' on θ'`dupl`. Given $l \in p'$. If $l \in$ `exemptl` we let $l \in p$. If $l \in \theta'$`dupl`, we let $l \in p$. If $l \notin$ `dupl` we let $l \in p$. Note that the nodes $l \in$ `dupl` have no preimage on p, but it will turn out that even the fewer nodes in p will guarantee that p' is closed.

   1.2. p' contains nodes below $Q_{\forall xA}$. Informally, we force p to behave on Q as p' on `dupl`. Given $l \in p'$. If $l \prec_{Q'} Q_{\forall xA}$ we let $l \in p$. If $l \in$ `dupl`, we let $l \in p$. If $l \notin \theta'$`dupl` we let $l \in p$. Note that the nodes $l \in \theta'$`dupl` have no preimage on p.

   1.3. p' contains nodes neither in `exemptl` nor below $Q_{\forall xA}$. In this case we arbitrarily pick p to behave on Q as p' on `dupl`. Given $l \in p'$, let $l \in p$.

   By assumption, $\mathcal{M}$ was clause-spanning on Q, and therefore p is spanned by a pair (l, k). We now distinguish cases.

   1.1. $l \in$ `dupl` and $k \in$ `dupl`. By definition, (l, k) and (l', k') are both in $\mathcal{M}'$. Depending on where l and k came from in p', one of these pairs spans p'.

   1.2. $l \in$ `dupl`, but $k \notin$ `dupl`. If $k \in$ `exemptl`, then $(l', k) \in \mathcal{M}'$. If $k \notin$ `exemptl` then (l, k) and (l', k) are in $\mathcal{M}'$ and p' is spanned, since l must be the preimage of either l or l'.

   1.3. $l \notin$ `dupl` and $k \notin$ `dupl`. Then $(l, k) \in \mathcal{M}'$ and also $l \in p'$ and $k \in p'$. Again, $\mathcal{M}'$ spans p'.

   This exhausts all possible cases. Therefore $\mathcal{M}'$ spans an arbitrary clause p'. Hence $\mathcal{M}'$ is clause-spanning on Q'.

2. $<_{Q'}$ is acyclic.

   This part of the proof is identical to the the part of the proof of Lemma 129 showing that $<_{Q'}$ is acyclic. We do not repeat it here.

□

Definition 131 We say that a term $\mathbf{P}_{o\alpha}$ is *literalic* if it is of the form $[\lambda z.A]$ for a literal A.

The following theorem is of interest not just in the translation of proofs, but also in the search for a proof. It tells us that we only need to consider literalic expansion terms for nodes which have a shallow formula equivalent to $\sim[A \doteq B]$.

Theorem 132 Given an expansion proof $(Q, \mathcal{M})$ for U. Then there is an expansion proof $(R, \mathcal{N})$ for U in which all expansion terms for nodes with a shallow formula of the form $\exists\mathbf{Q}[\mathbf{Q}f \wedge \sim\mathbf{Q}g]$ are literalic.

Proof: We will show how to eliminate one non-literalic expansion term. It then follows by induction on the number of such terms that all of them can be eliminated.

The proof is by an induction on the complexity of the tree below the expansion node $Q_{\exists\mathbf{Q}[\mathbf{Q}f\wedge\sim\mathbf{Q}g]}$. Call this node Q^0, and let $\mathbf{P}$ be an expansion term of Q^0 such that $\mathbf{P}$ is not literalic.

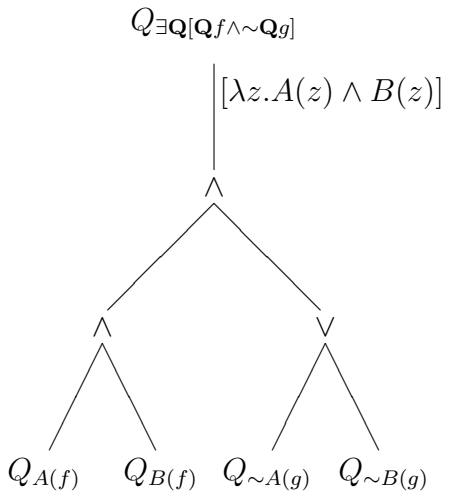
First note that if the $\lambda^{\sim}$-normal form of $\mathbf{P}$ does not begin with a λ-binder, we are done, since we can replace $\mathbf{P}$ by $[\lambda z.\mathbf{P}z]$ and obtain a node with the same shallow and deep formulas, but a literalic expansion term. Note that this is not an application of extensionality.

We will omit the details of dealing with leaf nodes which are not literals. In general, one can apply `deepen` (see Definition 124) to such nodes. In order to make sure that the complexity of the relevant part of the tree before and after each step does not increase one merely has to check that the new subtrees can be made into leaves again. This transforms leaves into new leaves, while the complexity of the shallow formula of the leaf is decreased. This is analogous to the proof of Lemma 123. However, we cannot do this process once initially, since the substitution into expansion trees in case 3 may transform a leaf nodes with a literal shallow formula into one with a more complex shallow formula. The same can be said for non-leaf nodes which are $\mathcal{M}$-mated. The algorithm given in the proof of Lemma 123 shows how these non-leaf nodes may be eliminated. It will only be necessary to eliminate a non-leaf node from the mating, if it disappears from the expansion tree.

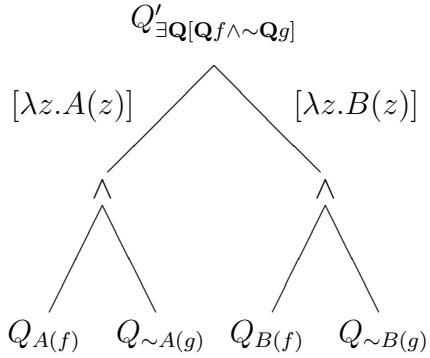
Of all the non-literalic expansion terms $\mathbf{P}$ in question, we pick one maximal with respect to $\ll_Q$ to be reduced. This will be important in case 3 in order to ensure that no other non-literalic expansion terms in question are duplicated.

For our main induction, we have to distinguish cases, depending on the form of $\mathbf{P}$.

1. $\mathbf{P} = [\lambda z.A(z) \wedge B(z)]$. Then we replace the expansion term $\mathbf{P}$ of Q^0 by two, namely $\mathbf{P}^1 = [\lambda z.A(z)]$ and $\mathbf{P}^2 = [\lambda z.B(z)]$. Omitting additional expansion terms of Q^0, the transformation on the expansion proof then is from



to



Let Q' be the result of replacing $Q_{\exists\mathbf{Q}[\mathbf{Q}f\wedge\sim\mathbf{Q}g]}$ by $Q'_{\exists\mathbf{Q}[\mathbf{Q}f\wedge\sim\mathbf{Q}g]}$ in Q. Furthermore let $\mathcal{M}' = \mathcal{M}$. If one of the intermediate non-leaf nodes below Q^0 was in $\mathcal{M}$ we apply `single` as indicated in a remark above.

We now have to show that $(Q', \mathcal{M}')$ is again an expansion proof. Most conditions are immediate, except for the following two.

1.1. $\mathcal{M}'$ is clause-spanning on Q'.

Let p' be a full clause on Q'. We define the preimage p of p' on Q. Every node in p' outside $Q'_{\exists\mathbf{Q}[\mathbf{Q}f\wedge\sim\mathbf{Q}g]}$ is also in p. If p' does not contain nodes from inside $Q'_{\exists\mathbf{Q}[\mathbf{Q}f\wedge\sim\mathbf{Q}g]}$ we are done, since then $p' = p$ and both are spanned. If p' contains nodes from $Q_{A(f)}$, let p contain the same nodes from $Q_{A(f)}$ in Q. If p' contains nodes from $Q_{B(f)}$ (but not $Q_{A(f)}$), let p contain nodes from $Q_{B(f)}$ in Q. If neither of these two cases applies then p' must contain nodes from $Q_{\sim A(g)}$ and $Q_{\sim B(g)}$. Let p contain these nodes. It can easily be checked that p, if extended by two intermediate nodes, is a full clause on Q. We have assumed that these intermediate nodes do not occur in $\mathcal{M}$. Therefore, p is spanned by a pair in $\mathcal{M}$ which also appears in $\mathcal{M}'$. Since p' was arbitrary, $\mathcal{M}'$ is clause-spanning on Q'.

1.2. $<_{Q'}$ is acyclic.

The proof is by contradiction. We assume that there is a cycle $t'_1 <^0_{Q'} t'_2 <^0_{Q'} \cdots <^0_{Q'} t'_n = t'_1$. We will construct a cycle in $<_Q$, which contradicts the assumption that $(Q, \mathcal{M})$ is an expansion proof.

Let t_i stand for the expansion term (occurrence) in Q corresponding to t'_i in Q'.

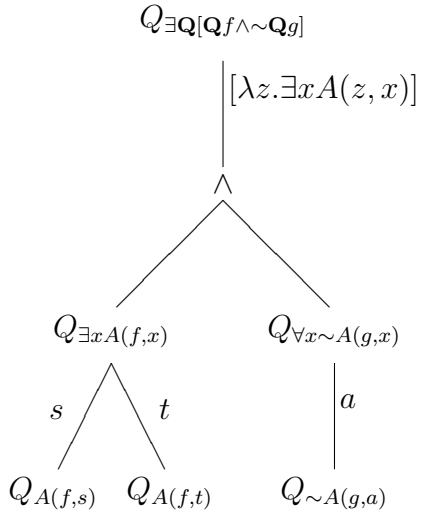
If none of the t'_i are $\mathbf{P}^1$ or $\mathbf{P}^2$, we are done. Without loss of generality assume that $t'_2 = \mathbf{P}^1$. But then also $\mathbf{P} <^0_Q t_3$, since a parameter selected below $\mathbf{P}^1$ in Q' must be selected below $\mathbf{P}$ in Q. Also, $t_1 <^0_Q \mathbf{P}$, since a parameter free in $\mathbf{P}^1$ is also free in $\mathbf{P}$. Therefore, by induction, we can construct a cycle in $<_Q$ by replacing t'_i by t_i and $\mathbf{P}^1$ and $\mathbf{P}^2$ by $\mathbf{P}$. But this is a contradiction.

2. $\mathbf{P} = [\lambda z.A \vee B]$. This case is symmetric to the case of a conjunction.

3. $\mathbf{P} = [\lambda z.\exists x A]$.

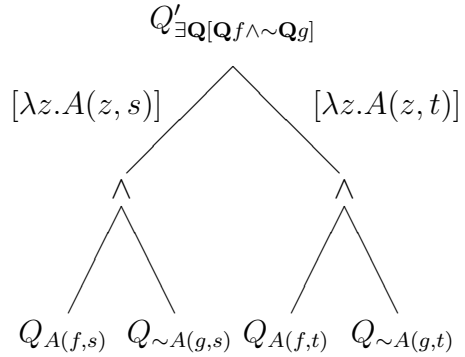
   This is the critical and most difficult case. For the sake of simplicity we assume that the expansion node $Q_{\exists x A(f,x)}$ below $Q_{\exists\mathbf{Q}[\mathbf{Q}f\wedge\sim\mathbf{Q}g]}$ has only two expansion terms, call them s and t.

   Under these assumptions, the expansion tree at Q^0 has the following form:



   Recall that $\mathbf{P}$ was chosen maximal (with respect to $\lll_Q$) among non-literalic substitution terms for nodes with a shallow formula of the form $\sim[A \doteq B]$. Since a is selected below $Q_{\exists\mathbf{Q}[\mathbf{Q}f\wedge\sim\mathbf{Q}g]}$ this means that a is not free in any other non-literalic expansion term for a negative equality node. Moreover, no other non-literalic expansion term lies below an expansion term in which a is free.

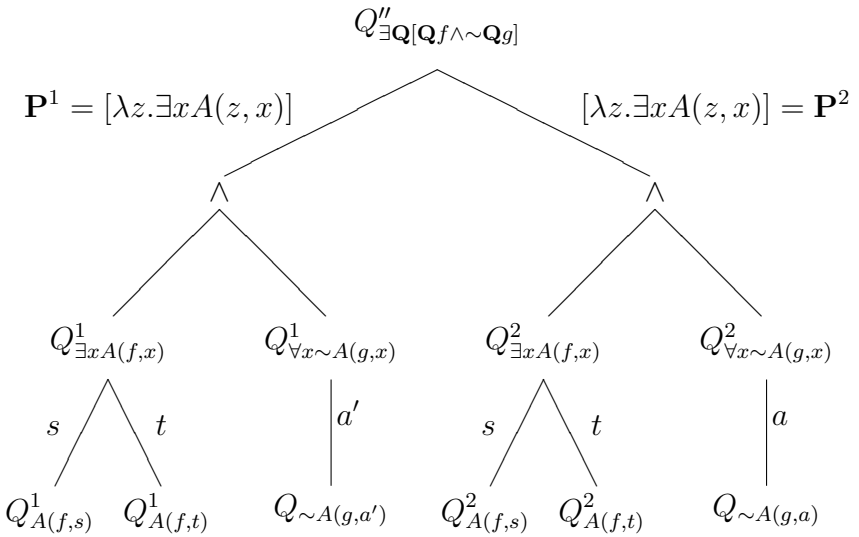
   Our goal is to replace $Q_{\exists\mathbf{Q}[\mathbf{Q}f\wedge\sim\mathbf{Q}g]}$ by $Q'_{\exists\mathbf{Q}[\mathbf{Q}f\wedge\sim\mathbf{Q}g]}$, where we construct two new nodes from $Q_{\sim A(g,a)}$ in Q. Let $Q_{\sim A(g,s)} = [a \leftarrow s]Q_{\sim A(g,a)}$ and $Q_{\sim A(g,t)} = [a \leftarrow t]Q_{\sim A(g,a)}$. Then we would like to construct $Q'_{\exists\mathbf{Q}[\mathbf{Q}f\wedge\sim\mathbf{Q}g]}$ as



   In order to facilitate the description of the transformation and correctness proof, we will construct $(Q, \mathcal{M})'$ in several steps.

3.1. Achieve that neither a nor any selected parameter $b \prec\!\!\prec_Q a$ is free in an expansion term below $Q_{\exists x A(f,x)}$. By Lemma 130 such an expanion proof, call it $(Q', \mathcal{M}')$ exists and is no deeper than $(Q, \mathcal{M})$.

3.2. Duplicate expansion term $\mathbf{P} = [\lambda z.\exists x A(z,x)]$ in $(Q', \mathcal{M}')$ such that in the new expansion proof $(Q'', \mathcal{M}'')$, the mating $\mathcal{M}''$ is strongly clause-spanning with respect to the two copies $\mathbf{P}^1$ and $\mathbf{P}^2$ of $\mathbf{P}$. This is possible by Lemma 129 without increasing the depth of the expansion tree below $\mathbf{P}$. The replacement for $Q_{\exists \mathbf{Q}[\mathbf{Q}f \wedge \sim \mathbf{Q}g]}$ now looks like (we have relabeled identical nodes with superscripts)



3.3. Erase $Q^1_{A(f,t)}$ and $Q^2_{A(f,s)}$.

This cannot be done immediately, since those subtrees are not unnecessary (in the sense of Definition 93). But it turns out that we can change the mating $\mathcal{M}''$ to render them unnecessary.

It is merely necessary to note that, because of step 33.1, the trees $Q^1_{A(f,t)}$ and $Q^2_{A(f,t)}$ do not contain parameters b such that $b \prec\!\!\prec_Q a$, nor do they contain a. Define a substitution

$$\theta = \{[c^1 \mapsto c^2] : c^1 \text{ selected below } Q^1_{A(f,t)}\} \cup \{[c^2 \mapsto c^1] : c^2 \text{ selected below } Q^2_{A(f,s)}\}$$

We now erase $Q^1_{A(f,t)}$ and $Q^2_{A(f,s)}$ and globally apply θ to Q''. This results in an expansion tree, call it Q'''

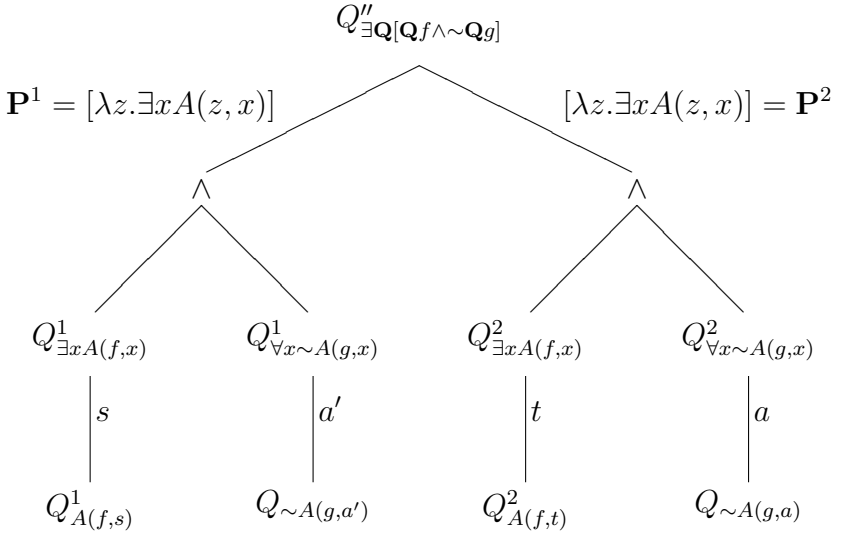
We define $\mathcal{M}'''$ by cases. Let $(l,k) \in \mathcal{M}''$ be given. Let us be content with merely defining the critcal case. If $l^1 \in Q^1_{A(f,t)}$ and $k^1 \in Q^1_{A(f,s)}$. By definition of $\mathcal{Q}'''$ it follows that $\theta k^1 = \theta l^2$ and that we can therefore mate k^1 with l^2 in $\mathcal{M}'''$. This and the symmetric case are the only new pairs in $\mathcal{M}'''$, after all pairs involving nodes in $Q^1_{A(f,t)}$ and $Q^2_{A(f,s)}$ have been erased.

It remains to show that $(Q''', \mathcal{M}''')$ is an expansion proof.

3.3.1. $\mathcal{M}'''$ is clause-spanning on $\mathcal{Q}'''$. This follows easily by the usual method of defining a preimage of a clause p by cases, and then check that $\mathcal{M}'''$ was constructed large enough.

3.3.2. $<_{Q'''}$ is acyclic. This follows because the selected parameters which we identified by θ were incomparable with respect to $\prec_{Q''}$.

The expansion tree Q''' now has the form



3.4. Eliminate the intermediate expansion and selection nodes.

This step finally achieves the expansion proof we were aiming for. Let $\mathbf{P}^s = [\lambda z.A(z,s)]$ and $\mathbf{P}^t = [\lambda z.A(z,t)]$. We obtain $Q_{\sim A(g,s)}$ as $Q_{\sim A(g,s)} = [a' \leftarrow s]Q_{\sim A(g,a)}$ and $Q_{\sim A(g,t)} = [a \leftarrow t]Q_{\sim A(g,a)}$. We apply shallowing at $Q^1_{\exists x A(f,x)}$, $Q^1_{\forall x\sim A(g,x)}$, $Q^2_{\exists x A(f,x)}$, and $Q^2_{\forall x\sim A(g,x)}$. Furthermore we apply $\theta = \{[a' \leftarrow s], [a \leftarrow t]\}$ globally. Of course, shallowing is not generally a sound operation at an inaccessible node, but here replacing $\mathbf{P}^1$ and $\mathbf{P}^2$ by $\mathbf{P}^s$ and $\mathbf{P}^t$ leaves the shallow and deep formulas of the tree internally consistent.

The structure of the full clauses not change, except that a few nodes have been erased through shallowing. Therefore $\mathcal{M}'''$ remains clause-spanning. If one of erased nodes appeared in $\mathcal{M}'''$ one has to apply deepening to the mated nodes. Note that these mated nodes are outside $Q_{\exists\mathbf{Q}[\mathbf{Q}f\wedge\sim\mathbf{Q}g]}$ and therefore do not increase the depth of the tree below an equality expansion term.

The critical property we must check here is that $<_{Q''''}$ is acyclic. But this follows analogously to the previous arguments of that kind. We assume that there is a cycle in $<_{Q''''}$ and show that then there must have been a cycle already present in $<_{Q'''}$. One has to be careful in defining the preimage of a cycle: $\mathbf{P}^s$, for example, may map to s or $\mathbf{P}^1$, depending on whether the parameter free in $\mathbf{P}^s$ is free in s or $\mathbf{P}^1$. However, once the problem is observed the details are straightforward. The cycle in $<_{Q'''}$ would not have to be any longer than the assumed cycle in $<_{Q''''}$.

Even though we made two (modified) copies of $Q_{\sim A(g,a)}$, no new non-literalic negative equality nodes have been created, since $\mathbf{P}$ was maximal with respect to $\ll\!\lhd_Q$.

4. $\mathbf{P} = [\lambda z.\forall x A]$. This case is symmetric to the case of an existential quantifier.

□

Theorem 133 We can always transform a proof in $\mathcal{H}$ with inferences of the form

$$\frac{U, \mathbf{P}f \wedge \sim\mathbf{P}g}{U, \exists\mathbf{Q}[\mathbf{Q}f \wedge \sim\mathbf{Q}g]}\,\exists I$$

into one where all substitution terms $\mathbf{P}$ for such inferences are literalic.

Proof: Given an $\mathcal{H}$ deduction, we can translate it into an expansion proof. By Theorem 132 we can eliminate non-literalic expansion terms for expansion nodes with appropriate assertions. We can then translate this new expansion proof back into an $\mathcal{H}$ deduction. Since expansion terms become instantiation terms, all instantiation terms for formulas of the required form will be literalic.

□

Remark 134 One has to be careful in interpreting this theorem. Even though $\mathbf{P}$ may be literalic, $\mathbf{P}f$ may not be a literal! This is possible in higher-order logic since $\mathbf{P}$ may be of the form $[\lambda z.zt_1 \dots t_n]$, which is literalic. The following example shows that this lemma cannot be extended.

Example 135 $[p \doteq [p \wedge q]] \supset [p \supset q]$. We interpret this formula in $\mathcal{L}$ (not $\mathcal{L}^=$) and give an $\mathcal{H}$-deduction for it.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{p, \sim p \qquad \dfrac{\sim p, \sim q, q}{\sim p \vee \sim q, q}\vee I}{p \wedge \sim[p \wedge \sim q], \sim p, q}\wedge I}{\exists\mathbf{Q}[\mathbf{Q}p \wedge \sim\mathbf{Q}[p \wedge q]], \sim p, q}\exists \mathrm{I} : [\lambda zz]}{\exists\mathbf{Q}[\mathbf{Q}p \wedge \sim\mathbf{Q}[p \wedge q]], p \supset q}\vee I}{[p \doteq [p \wedge q]] \supset [p \supset q]}\vee I$$

It can be seen that, if the instantiation term for $\mathbf{Q}$ is constrained to be literalic, it must be either $[\lambda zz]$ or $[\lambda z.\sim z]$ in order to be able to prove the theorem. In both cases $\mathbf{Q}[p \wedge \sim q]$ would not be a literal.

Theorem 136 Cut-elimination does not hold in $\mathcal{H}^=$.

Proof: Consider $[p \wedge q] \doteq [p \vee q] \supset [p \supset q]$. Duplication will get us nowhere, since we have to make use of $p \supset q$. After two steps we have

$$\dfrac{\dfrac{\overset{\mathcal{D}}{\sim[[p \wedge q] \doteq [p \vee q]], \sim p, q}}{\sim[[p \wedge q] \doteq [p \vee q]], [p \supset q]} \vee I}{[p \wedge q] \doteq [p \vee q] \supset [p \supset q]} \vee I$$

There is no way in which the $S\doteq$ rule can be applied such that $\mathbf{P}B$ and $\mathbf{P}A$ are in different equivalence classes (notation from Definition 122), since $\mathbf{P}$ could not depend on its argument ($p \vee q$ does not appear in the line except in the equality). Hence we must use cut to force $p \vee q$ to appear. This completes the proof.

□

It turns out that $p \vee q$ by itself is a good candidate for the cut formula in the above counterexample, and would also have been suggested by the algorithm in the proof of Theorem 141. The final deduction we would obtain, and which is quite natural:

$$\dfrac{\dfrac{\dfrac{\sim p, p, q}{\sim p, p \vee q} \vee I \qquad \dfrac{\dfrac{\sim p, \sim q, q}{\sim[p \wedge q], q} \vee I}{\sim[p \vee q], \sim[[p \wedge q] \doteq [p \vee q]], q} S\doteq}{\sim[[p \wedge q] \doteq [p \vee q]], \sim p, q} Cut}{\dfrac{\sim[[p \wedge q] \doteq [p \vee q]], [p \supset q]}{[p \wedge q] \doteq [p \vee q] \supset [p \supset q]} \vee I} \vee I$$

4.3 Translating Expansion Proofs into $\mathcal{H}^=$

Here an expansion proof is an expansion proof of the original assertion, where all the definitions of equality have been expanded according to

$$A \doteq B \underset{\text{def}}{=} \forall \mathbf{q}[\mathbf{q}A \supset \mathbf{q}B]$$

Since equality is primitive in $\mathcal{H}^=$ the translation algorithm from expansion proofs into $\mathcal{H}$-deductions must be modified. An equality in the statement of the theorem to be proven cannot simply be instantiated in $\mathcal{H}^=$, but must be used according to the substitution rule and the reflexivity axiom available. It turns out that this will require an elaborate modification of the expansion tree before the translation process begins, which in itself is rather straightforward.

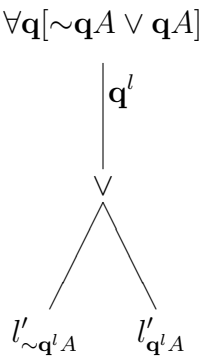
If one examines the expansion proof in which equality was instantiated, two questions arise: how do we eliminate references to the selected parameter which comes from instantiating a positive occurrence of equality, and how do we interpret the expansion terms of an existential

quantifier which comes from instantiating a negatively occurring equality? These expansion terms and selected parameters cannot have a direct analogue in the $\mathcal{H}$-deduction, since the expansion node does not correspond to an existentially quantified formula, and the selection node does not correspond to a univerally quantified formula. Instead, they correspond to a negatively and positively occurring equality, respectively.

These questions are answered in Lemma 139 and in the proof of Theorem 141. Lemma 139 intuitively states that the expansion proof can be modified such that all selection nodes which correspond to a positively occurring equality will be instances of the reflexivity axioms when translated into an $\mathcal{H}^=$-deduction. The proof of Theorem 141 shows that an expansion term $\mathbf{P}$ of an expansion node corresponding to a negatively occurring equality, say $\sim[A \doteq B]$ means that we will substitute A for B in $\mathbf{P}B$ when translating into an $\mathcal{H}^=$-deduction.

We will have frequent occasion to use the expansion proof for $A \doteq A$, not only by itself but as subtrees of larger expansion proofs. We call such an expansion tree an *initial equality tree*. An additional condition on the mating ensures that the parameter selected for the hidden universal quantifier is not used elsewhere.

Definition 137 An *initial equality tree* in $(Q, \mathcal{M})$ is a subtree of an expansion proof $(Q, \mathcal{M})$ of the form



such that $(l'_{\sim \mathbf{q}^l A}, l'_{\mathbf{q}^l A}) \in \mathcal{M}$, but $\mathbf{q}^l$ does not occur in any expansion term in Q. Since $\mathbf{q}^l$ also does not occur free in Q, this means than neither $l'_{\sim \mathbf{q}^l A}$, nor $l'_{\mathbf{q}^l A}$ occur elsewhere in $\mathcal{M}$.

Theorem 138 Let $\texttt{eqsel}((Q, \mathcal{M}))$ be the set of parameters selected at selection nodes with shallow formula of the form $\forall \mathbf{q}[\mathbf{q}A \supset \mathbf{q}B]$, where $\mathbf{q}$ not free in A or B. Then for any expansion proof $(Q, \mathcal{M})$ for U there exists an expansion proof $(R, \mathcal{N})$ for U such that no parameter in $\texttt{eqsel}((R, \mathcal{N}))$ is free in any expansion term occurring in R.

Proof: The proof will be entirely constructive, thus giving an algorithm for construction $(R, \mathcal{N})$ given $(Q, \mathcal{M})$.

The proof is easier to understand if we often write the uninstantiated form of equality. Note, however, that this does not affect the expansion tree itself due to the convention that formulas which can be obtained from each other by instantiating definitions are equivalent.

We show how to eliminate occurrences of the parameter selected for one expansion node with shallow formula $\forall\mathbf{q}[\mathbf{q}A \supset \mathbf{q}B]$ (note: $\mathbf{q}$ not free in A or B). By induction on the number of such selection nodes it follows that all occurrences of such parameters may be eliminated.

Pick such node Q^0 with shallow formula $\forall\mathbf{q}[\mathbf{q}A \supset \mathbf{q}B]$. Q^0 could be a leaf, or a selection node. If it is a leaf, we are done, because there is no selected parameter for this positive occurrence of equality. If Q^0 is a selection node, its two immediate subtrees must be leaves, since their shallow formulas are literals. For simplicity, let us assume the selected parameter for $\mathbf{q}$ is also called $\mathbf{q}$.

There are two cases to consider.

If the deep formula of such a node is $\sim\mathbf{q}A \vee \mathbf{q}B$ and $Q_{\sim\mathbf{q}A}$ was $\mathcal{M}$-mated to $Q_{\mathbf{q}B}$, we leave this pair in the mating, but try to eliminate any other mated pairs in $\mathcal{M}$ containing $Q_{\sim\mathbf{q}A}$ or $Q_{\mathbf{q}B}$. Note that this case implies that A is equal to B, i. e., they are in the same formula equivalence class.

If the deep formula of such a node is $\sim\mathbf{q}A \vee \mathbf{q}B$, where $Q_{\sim\mathbf{q}A}$ and $Q_{\mathbf{q}B}$ are not $\mathcal{M}$-mated, we try to eliminate the selection node Q^0 and replace it by a leaf with the same shallow formula. This means that the selected parameter $\mathbf{q}$ disappears from the expansion proof.

To eliminate $\mathbf{q}$, let us first note that $\mathbf{q}$ cannot be free in Q because of the restrictions on expansion proofs (see Definition 45).

Because of the asymmetry in the definition of equality, there are two ways of eliminating $\mathbf{q}$ from expansion terms. See Remark 140 for some heuristics on how to decide which one to use in real proof transformation situations.

We are now trying to construct an expansion proof $(Q', \mathcal{M}')$ in which $\mathbf{q}$ does not appear in any expansion term. To construct Q' we can either apply the substitution

$$\mathbf{q} \leftarrow [\lambda x.A \doteq x]$$

or the substitution

$$\mathbf{q} \leftarrow [\lambda x.{\sim}x \doteq B]$$

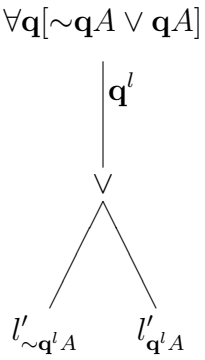
We will only prove that the first alternative is correct — the proof for the second alternative is symmetric. A heuristic for making the choice between the substitutions is discussed in Remark 140.

As a notational convention, we write N' for the node in Q' corresponding to N in Q.

$\mathcal{M}'$ is obtained from $\mathcal{M}$ by replacing some of the mated pairs. Note that $\mathcal{M}'$ will have exactly as many elements as $\mathcal{M}$ did.

In the deep formula of Q every literal with $l^S = \mathbf{q}A$ becomes l' with $l'^S = [\lambda x.A \doteq x]A = [A \doteq A]$.

Let L be the set of leaves $\mathcal{M}$-mated to QnA. Certainly, for $l \in L$, $l^S = \mathbf{q}A$. Therefore $l'^S = [A \doteq A] = [\forall\mathbf{q}.{\sim}\mathbf{q}A \vee \mathbf{q}A]$. We select a new parameter $\mathbf{q}^l$ for this new node and replace it by an initial equality tree.



In $\mathcal{M}'$ we replace every pair $(l, Q_{\sim \mathbf{q}A}) \in \mathcal{M}$ by $(l'_{\sim \mathbf{q}^l A}, l'_{\mathbf{q}^l A})$. Call the set of all new literals L^{new} and call l the ancestor of $l'_{\sim \mathbf{q}^l A}$ and $l'_{\mathbf{q}^l A}$.

Every literal with $l^S = \sim \mathbf{q}B$ becomes l' with $l'^S = \sim[\lambda x.A \doteq x]B = \sim[A \doteq B]$.

Let K be the set of leaves $\mathcal{M}$-mated to QB. For every $k \in K$ we replace the pair $(k, Q_{\mathbf{q}B}) \in \mathcal{M}$ by $(k', Q^{0'})$ to obtain $\mathcal{M}'$. Recall that Q^0 was the original selection node with shallow formula $A \doteq B$ and can therefore be mated to k' with shallow formula $\sim[A \doteq B]$.

Now that we have described how to obtain $(Q', \mathcal{M}')$ we have to prove that it is an expansion proof for U.

First note that $Q^S = Q'^S$, because the $\mathbf{q}$ was not free in Q^S.

Thus it remains to show that $\mathcal{M}'$ spans every full clause on Q' and that $<_{Q'}$ is acyclic.

1. $\mathcal{M}'$ spans every full clause on Q'.

   Let p' be a full clause through Q'. Let p be the preimage of p' on Q defined as follows.

   If p' contains $Q^{0'}$, p contains $Q_{\sim \mathbf{q}A}$, $Q_{\mathbf{q}B}$, and $\mathcal{Q}^0$. If p' contains some new $l^{new} \in L^{new}$, p contains the corresponding ancestor l.

   By assumption we know that p was spanned by a pair (l, k). There are four cases to consider.

   1.1. If both l and k are distinct from $Q_{\sim \mathbf{q}A}$ and $Q_{\mathbf{q}B}$, then $(l', k') \in \mathcal{M}'$ and p' is spanned by $\mathcal{M}'$.

   1.2. If p is spanned by the pair $(Q_{\sim \mathbf{q}A}, Q_{\mathbf{q}B})$, p' will still be spanned by the corresponding pair $(Q'_{\sim \mathbf{q}A}, Q'_{\mathbf{q}B})$.

   1.3. If p is spanned by a pair $(l, Q_{\sim \mathbf{q}A})$, l distinct from $Q_{\mathbf{q}B}$, then $(l, Q_{\sim \mathbf{q}A})$ was replaced by a pair $(l'_{\sim \mathbf{q}^l A}, l'_{\mathbf{q}^l A})$. Both of these literals have to be on p' and therefore p' is spanned by $\mathcal{M}$.

   1.4. If p is spanned by a pair $(k, Q_{\mathbf{q}B})$, k distinct from $Q_{\sim \mathbf{q}A}$, then $(k, Q_{\mathbf{q}B})$ was replaced by $(k', Q^{0'})$. By the construction of p, both k' and $Q^{0'}$ have to be on p', and hence p' is spanned by $\mathcal{M}$.

2. $<_{Q'}$ is acyclic.

   Recall that $t <^0_{Q'} s$ if there is a parameter selected for a node below t which is free in s, and that $<_{Q'}$ is the transitive closure of $<^0_{Q'}$.

   The proof is by contradiction. We assume that there is a cycle $t'_1 <^0_{Q'} t'_2 <^0_{Q'} \cdots <^0_{Q'} t'_n = t'_1$. We will construct a cycle in $<_Q$, which contradicts the assumption that $(Q, \mathcal{M})$ is an expansion proof.

   Let t_i stand for the expansion term (occurrence) in Q corresponding to t'_i in Q'.

   If $\mathbf{q}$ is not free in any t_i we are done, because then $t_1 <^0_Q t_2 <^0_Q \cdots <^0_Q t_n = t_1$ which is a contradiction.

   Otherwise, we pick a term with an occurrence of $\mathbf{q}$. Without loss of generality, let it be $t_2(\mathbf{q})$. If we can show that $t_1 <_Q t_2$ it follows by induction on the number of t_i with $\mathbf{q}$ free, that $t_1 <_Q t_2 <_Q \cdots <_Q t_n = t_1$ which would be a contradiction.

   t'_2 is of the form $t_2([\lambda x.x \doteq A])$. Since $t'_1 <^0_{Q'} t'_2$, there must be parameter free in t'_2 which is selected below t'_1.

   If this parameter occurs free in t'_2 outside A, we are done, since then it also appears free in t_2.

   Then t'_2 is of the form $t_2([\lambda x.A(c) \doteq x])$ for some parameter c which is selected below t_1. $A(c)$ is also free below the selection node Q^0 with selected parameter $\mathbf{q}$ (its deep formula being $\sim\mathbf{q}A \vee \mathbf{q}B$). Now we have to consider two cases:

   2.1. c is selected above Q^0. This means that $\mathbf{q}$ is selected below t_1, since c was assumed to be selected below t_1. Hence by definition $t_1 <^0_Q t_2(\mathbf{q})$.

   2.2. c is free in some expansion term $s(c)$ above Q^0. Then, by definition, $s(c) <^0_Q t_2(\mathbf{q})$, since $\mathbf{q}$ is free in t_2 and $\mathbf{q}$ is selected below $s(c)$. But c is free in $s(c)$ and selected below t_1 and therefore also $t_1 <^0_Q s(c)$. Since $<_Q$ is the transitive closure of $<^0_Q$, $t_1 <_Q t_2(\mathbf{q})$. But this is the contradiction we were looking for.

In order to complete the overall induction we have to show that while eliminating occurrences of $\mathbf{q}$ in expansion terms, we do not introduce occurrences of other equalities whose selected parameter occurs in expansion terms. But that is easy to see, since there the only new expansion term were obtained by substituting for $\mathbf{q}$ in existing ones, and the new substitution term does not contain any new selected parameters.

□

Lemma 139 Let $(Q, \mathcal{M})$ be an expansion proof for U such that no parameter in $\texttt{eqsel}((Q, \mathcal{M}))$ occurs in an expansion term in Q. Then there exists an expansion proof $(R, \mathcal{N})$ for U such that all subtrees with shallow formula $\forall\mathbf{q}[\mathbf{q}A \supset \mathbf{q}B]$ either have the form of an initial equality tree, or are leaves of R.

Proof: Since $\mathbf{q}$ can not occur free in Q (since it is selected), $\mathbf{q}A$ and $\mathbf{q}A$ can not be mated to any other node except to each other. If they are mated to each other, the subtree will be an initial equality tree. If they are not mated at all, the subtree can be replace with a leaf with the correct shallow formula without invalidating the expansion proof.

□

Remark 140 In the algorithm which proved Theorem 138 we had a choice between a positive and a negative substitution for a selected parameter $\mathbf{q}$. This choice will affect the $\mathcal{H}^=$ deduction constructed from the expansion proof, and therefore must be made with care. A good heuristic may be to choose the "most positive" replacement of of $\mathbf{q}$. If $\mathbf{q}$ appears positively, replace it by the positive term, if it appears negatively, replace it by the negative term. Of course, sometimes $\mathbf{q}$ will occur positively and negatively, in which case counting positive and negative occurrences may extend the heuristic. Another heuristic would try to minimize the number of occurrences of formulas of the form $A \doteq A$ in the final expansion proof. Uses of axioms of the form $A \doteq A$ in an $\mathcal{H}^=$ deduction often look clumsy and it may therefore be a good strategy to avoid them as much as possible.

Theorem 141 Let U be a multiset of equivalence classes of formulas in the language $\mathcal{L}^=$, and let $(Q, \mathcal{M})$ be an expansion proof for U. Then there exists a deduction in $\mathcal{H}^=$ of U.

Proof: The proof is entirely constructive and thus gives a translation procedure from expansion proofs into $\mathcal{H}^=$ deductions.

The invariant of other translation procedures presented earlier (see, for example, Algorithm 77) was that the multiset of shallow formulas immediately below the root of the expansion proof was equal to the multiset of formulas in the assertion of the $\mathcal{H}$-deduction.

This no longer holds, since the language of $\mathcal{H}$-deductions is an extension of the language for expansion proofs. Rather, we say that the multiset of shallow formulas immediately below the root of the expansion proof and the assertion of the $\mathcal{H}^=$ deduction are equal up to instantiation of equality symbols.

Given a $\mathcal{H}^=$-deduction for $U^=$ and an expansion proof for U, we denote the $\mathcal{L}^=$ formula in $U^=$ corresponding to the $\mathcal{L}$ formula A in U by $A^=$.

Now we describe the translation algorithm.

By virtue of Theorem 138 we may assume that no expansion term in $(Q, \mathcal{M})$ contains a parameter selected for a node with shallow formula $\forall\mathbf{q}[\mathbf{q}A \supset \mathbf{q}B]$.

Be Lemma 139 we may also assume that all subtrees with shallow formula $A \doteq B$ either have the form of initial equality trees, or are leaves of Q.

All of the steps given in Algorithm 77 remain the same, if applicable. It remains to discuss three additional situations:

1. $L = U, A \doteq A$ such that $Q_{A \doteq A}$ has the form of an initial equality tree (see 137). In this case we may apply rule $R\doteq$ and L is initial.

2. $L = U, A \doteq B$ such that the previous case does not apply. Note that in this case, $Q_{A \doteq B}$ must be a leaf of Q. Here we do not add any new transformations.

3. $L = U, \sim[A \doteq B]$ such that $Q_{\sim[A \doteq B]} = Q_{\exists \mathbf{P}[\mathbf{P}A \wedge \sim \mathbf{P}B]}$ is an expansion node with 2 or more expansion terms. Then we apply `split` just as in case 5 of Algorithm 77. L is then inferred by

$$\frac{U, \sim[A \doteq B], \sim[A \doteq B]}{U, \sim[A \doteq B]}\, C$$

where each of the two new subnodes with shallow formula $\sim[A \doteq B]$ has fewer expansion terms.

4. $L = U, \sim[A \doteq B]$ such that $Q_{\exists \mathbf{P}[\mathbf{P}A \wedge \sim \mathbf{P}B]}$ has exactly one expansion term $\mathbf{P}$, and $\mathbf{P}$ is admissible. From looking at Example 135 and Theorem 136 it is clear that we will have to introduce a cut sometimes.

   Let us follow Algorithm 77 to create an $\mathcal{H}$-deduction. If we commit ourselves to using case 6 ($\exists$I), followed by case 7 ($\wedge I$), we obtain the following $\mathcal{H}$-deduction (for some $U^{\mathbf{P}A}$ sufficient for $\mathbf{P}A$ and some $U^{\sim \mathbf{P}B}$ sufficient for $\sim \mathbf{P}B$):

$$\dfrac{\dfrac{\dfrac{U^{\mathbf{P}A}, \mathbf{P}A \qquad U^{\sim \mathbf{P}B}, \sim \mathbf{P}B}{U^{\mathbf{P}A}, U^{\sim \mathbf{P}B}, \mathbf{P}A \wedge \sim \mathbf{P}B}\, \wedge I}{U, \mathbf{P}A \wedge \sim \mathbf{P}B}\, C}{U, \exists \mathbf{Q}[\mathbf{Q}A \wedge \sim \mathbf{Q}B]}\, \exists \mathrm{I} : \mathbf{P}$$

   Thus Algorithm 77 will give us two new expansion proofs for $U^{\mathbf{P}A}, \mathbf{P}A$ and $U^{\sim \mathbf{P}B}, \sim \mathbf{P}B$. Call them $(Q^{\mathbf{P}A}, \mathcal{M}^{\mathbf{P}A})$ and $(Q^{\sim \mathbf{P}B}, \mathcal{M}^{\sim \mathbf{P}B})$, respectively.

   In the system $\mathcal{H}^=$, where $\doteq$ is primitive, we use Cut and $S\doteq$ instead of of $\exists$I and $\wedge I$, but we achieve the same subdeductions. Infer L by

$$\dfrac{\dfrac{\dfrac{U^{\mathbf{P}A}, \mathbf{P}A}{U^{\mathbf{P}A}, \sim[A \doteq B], \mathbf{P}B}\, S\doteq \qquad \sim \mathbf{P}B, U^{\sim \mathbf{P}B}}{U^{\mathbf{P}A}, \sim[A \doteq B], U^{\sim \mathbf{P}B}}\, Cut}{U, \sim[A \doteq B]}\, C$$

□

The algorithm above can be significantly improved in order to avoid the use of cut in many cases. The inherent asymmetry in the substitution prevents the system from being cut-free, even in the first-order restriction. However, if we introduce a new rule which substitutes the right-hand side of the equality for the left-hand side instead of vice versa, the resulting system $\mathcal{H}^*$ is cut-free in the first-order case. That this extension is really necessary will become clear in Example 148.

Definition 142 Let $\mathcal{H}^*$ be like $\mathcal{H}^=$ augmented with the following dual to the substitution rule

$$\frac{U, \mathbf{P}B}{U, \sim[A \doteq B], \mathbf{P}A} S^*\doteq : A \doteq B, \mathbf{P}A, \mathbf{P}B$$

The following theorem shows an important subcase in which we do not have to introduce a cut into the $\mathcal{H}^*$ deduction.

Lemma 143 Given an expansion proof $(Q, \mathcal{M})$ for $U, \sim[A \doteq B]$ such that $Q_{\exists\mathbf{P}[\mathbf{P}A \wedge \sim\mathbf{P}B]}$ has exactly one expansion term $\mathbf{P}$, $Q_{\sim\mathbf{P}B}$ is a leaf of Q, and $Q_{\mathbf{P}A}$ is either a leaf or an initial equality tree in $(Q, \mathcal{M})$. Then there is an $\mathcal{H}^=$-deduction where this occurrence of $\sim[A \doteq B]$ is inferred without the use of cut.

Proof: The idea of the proof is that either the expansion term $\mathbf{P}$ is unnecessary or $Q_{\sim\mathbf{P}B}$ is $\mathcal{M}$-mated to at least one other node in Q. In the second case we delay the attempt to apply the substitution rule until one of the nodes mated to it appears as an element of the assertion in the deduction. Let us assume that $Q_{[\mathbf{P}A \wedge \sim\mathbf{P}B]}$ is not mated, otherwise apply deepening to achieve that it is not mated. Not all the following cases are exclusive, that is, sometimes more than one transformation may be applied in order to obtain a cut-free application of substitution.

1. $Q_{\sim\mathbf{P}B}$ is not $\mathcal{M}$-mated. Every full clause through $Q_{\sim\mathbf{P}B}$ is spanned by $\mathcal{M}$. But then every full clause through $Q_{\mathbf{P}A}$ will also be spanned by $\mathcal{M}$ without involving a node in $Q_{\mathbf{P}A}$, simply choose the pair which spanned the parallel clause through $Q_{\sim\mathbf{P}B}$. Hence no subtree of $Q_{\exists\mathbf{P}[\mathbf{P}A \wedge \sim\mathbf{P}B]}$ is mated, and therefore $\mathbf{P}$ is unnecessary. This means that it will never be used as an equality for substitution, as can be seen by checking the proof of Theorem 141.

2. $Q_{\sim\mathbf{P}B}$ has at least on $\mathcal{M}$-mate. Note that $\mathbf{P}$ is maximal with respect to $<_Q$, that is, there is no parameter selected below $\mathbf{P}$ which is free in any other substitution term. If $Q_{\mathbf{P}A}$ is a leaf, this follows immediately, if $Q_{\mathbf{P}A}$ is an initial equality tree it follows because the only parameter selected below $\mathbf{P}$ is not free in any other expansion term in Q by definition of initial equality tree. Since $\mathbf{P}$ is maximal, there will never be a point where all top-level nodes in $(Q, \mathcal{M})$ are existential with $\mathbf{P}$ the only minimal expansion term. Hence we can proceed with Algorithm 141 until we have reached a point where we have an expansion proof $(Q', \mathcal{M}')$ for a line either of the form $U', \sim[A \doteq B]$ such that $\mathbf{P}$ is unnecessary, or $U, \sim[A \doteq B], \mathbf{P}B$. In the first case we are done, in the second case we can apply $S\doteq$ immediately. This follows from Algorithm 77 by the observation that $\mathbf{P}B$ is sufficient (in the sense of Definition 71) for $\mathbf{P}$. Thus one may infer (in $\mathcal{H}$!)

$$\dfrac{\dfrac{U', \mathbf{P}A \qquad \sim\mathbf{P}B, \mathbf{P}B}{U', \mathbf{P}A \wedge \sim\mathbf{P}B, \mathbf{P}B} \wedge I}{U, \exists\mathbf{Q}[\mathbf{Q}A \wedge \sim\mathbf{Q}B], \mathbf{P}B} \exists \mathrm{I} : \mathbf{P}$$

Thus Algorithm 77 yields a new expansion proof $(Q'', \mathcal{M}'')$ for $U', \mathbf{P}A$. Hence we can use substitution directly in $\mathcal{H}^=$ and infer

$$\frac{U', \mathbf{P}A}{U', \sim[A \doteq B], \mathbf{P}B} S\doteq$$

and let $(Q'', \mathcal{M}'')$ be the new expansion proof for $U', \mathbf{P}A$.

□

Lemma 144 Given an expansion proof $(Q, \mathcal{M})$ for $U, \sim[A \doteq B]$ such that $Q_{\exists\mathbf{P}[\mathbf{P}A \wedge \sim\mathbf{P}B]}$ has exactly one expansion term $\mathbf{P}$, $Q_{\mathbf{P}A}$ is a leaf of Q, and $Q_{\sim\mathbf{P}B}$ is either a leaf or an initial equality tree in $(Q, \mathcal{M})$. Then there is an $\mathcal{H}^=$-deduction where this occurrence of $\sim[A \doteq B]$ is inferred without the use of cut.

Proof: The proof is dual to the proof of Lemma 143. We will just indicate the crucial step where we need the rule $S^*\doteq$. If $Q_{\mathbf{P}A}$ has at least one mate, one may infer in $\mathcal{H}$ (eventually)

$$\frac{\dfrac{U', \sim\mathbf{P}B \qquad \mathbf{P}A, \sim\mathbf{P}A}{U', \mathbf{P}A \wedge \sim\mathbf{P}B, \sim\mathbf{P}A} \wedge I}{U, \exists\mathbf{Q}[\mathbf{Q}A \wedge \sim\mathbf{Q}B], \sim\mathbf{P}A} \exists \mathrm{I} : \mathbf{P}$$

Therefore we get in $\mathcal{H}^*$:

$$\frac{U', \sim\mathbf{P}B}{U', \sim[A \doteq B], \sim\mathbf{P}A} S^*\doteq$$

□

Remark 145 It follows that in the first-order fragment, the system $\mathcal{H}^*$ is cut-free, using Theorem 132 and Lemmas 143 and 144.

Theorem 146 Given an expansion proof $(Q, \mathcal{M})$ with a node $Q_{\sim[A\doteq B]}$. If $Q_{\sim[A\doteq B]}$ has exactly one expansion term $\mathbf{P}$, and $\mathbf{P} = [\lambda x.A \doteq x]$ or $\mathbf{P} = [\lambda x.\sim[x \doteq B]]$ then one may eliminate the expansion term $\mathbf{P}$ from $(Q, \mathcal{M})$.

Proof: This is somewhat different from the usual methods for eliminating expansion terms. Usually one shows that the term is unnecessary and then erases it and the tree below it from Q. Here we note that either $\mathbf{P}A$ or $\mathbf{P}B$ has exactly the same shallow formula as $Q_{\sim[A\doteq B]}$. Again we assume that $Q_{[\mathbf{P}A\wedge\sim\mathbf{P}B]}$ has no $\mathcal{M}$-mates. Let us only treat the case the $\mathbf{P} = [\lambda x.A \doteq x]$, the other case is symmetric. Then $Q_{\mathbf{P}A}$ can be transformed into an initial equality tree in Q. (see the proof of Theorem 138). In particular we can achieve that $Q_{\mathbf{P}A}$ does not occur in $\mathcal{M}$. Now

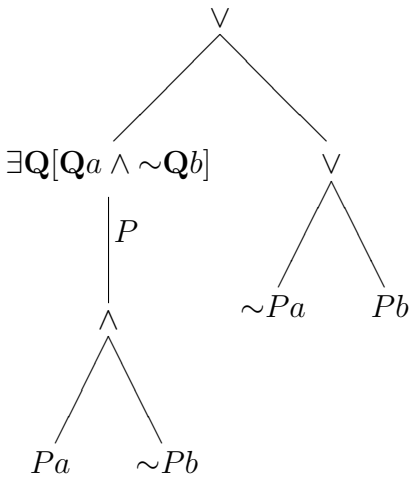
we obtain Q' by replacing $Q_{\sim[A \doteq B]}$ by $Q_{\sim \mathbf{P}B}$. Note that they have the same shallow formula, namely $\sim[A = B]$ which is equivalent to $\exists\mathbf{Q}[\mathbf{Q}A \wedge \sim\mathbf{Q}B]$. $\mathcal{M}'$ is obtained by deleting the pair from the initial equality tree $Q_{\mathbf{P}A}$ and substituting $Q_{\sim \mathbf{P}B}$ for $Q_{\sim[A \doteq B]}$ elsewhere. Every full clause through $Q_{\sim \mathbf{P}B}$ remains spanned. It is also easy to see that $<_{Q'}$ is acyclic, since no new pairs are added to $<^0_Q$.

□

In the following examples we will replace a selection, expansion, or leaf node with its shallow formula to increase the legibility of the expansion tree. Thus, for example, we write $\exists\mathbf{Q}[\mathbf{Q}a \wedge \sim\mathbf{Q}b]$ instead of $Q_{\exists\mathbf{Q}[\mathbf{Q}a \wedge \sim\mathbf{Q}b]}$.

Example 147 $a \doteq b \supset [Pa \supset Pb]$.

This is almost the simplest possible example. The obvious expansion proof for this theorem is



If we apply the improved algorithm for translating this into $\mathcal{H}^=$, we do not have any nondeterministic choices. The resulting deduction is a natural one. Note that we make use of our convention that formulas with a definition (here $\supset$) are equivalent to their "instantiated" form. Therefore we do not need any explicit rules to convert implications into disjunctions or vice versa.

$$\dfrac{\dfrac{\dfrac{\sim Pa, Pa}{\sim a \doteq b, \sim Pa, Pb}\, S\doteq}{\sim a \doteq b, Pa \supset Pb}\, \vee I}{a \doteq b \supset [Pa \supset Pb]}\, \vee I$$

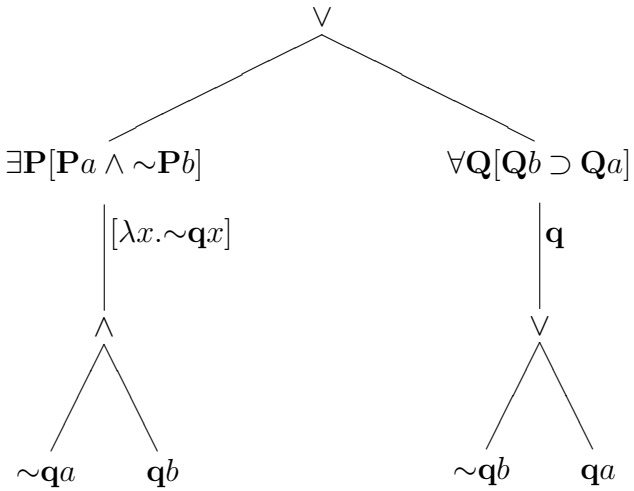
Note that we did not have any choice, since according to the proof of Lemma 143 we have to wait before applying $S\doteq$, even though the substitution term P is admissible in the line with assertion $\sim a \doteq b, Pa \supset Pb$.

Example 148 $a \doteq b \supset b \doteq a$.

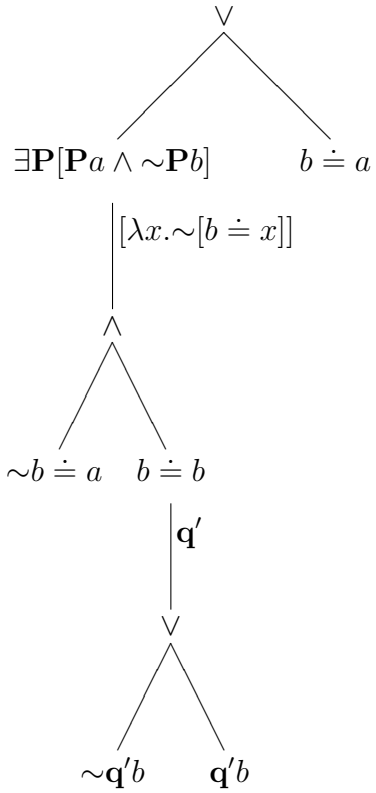
This example is interesting for two reasons. First of all we will have to apply Theorem 138 to eliminate the parameter selected for $b \doteq a$ from the expansion term for $a \doteq b$. Secondly, we will recover an intuitionistically valid proof when performing this step.

The expansion proof we assume has the simplest expansion term for $a \doteq b$, since it contains only a negation. After eliminating parameters selected for equality the complexity of the expansion term is much greater: it contains an existential quantifier and an implication.

The original expansion proof is



Now, following the proof of Theorem 138, we have a choice. Let us assume that we substitute $\mathbf{q} \leftarrow [\lambda x.b \doteq x]$. Then the expansion proof becomes



The $\mathcal{H}^=$ deduction we obtain with the improved algorithm again turns out to be a natural one.

$$\dfrac{\dfrac{b \doteq b}{\sim[a \doteq b], b \doteq a}\, S^*\doteq}{a \doteq b \supset b \doteq a}\, \vee I$$

Notice that here the use of $S^*\doteq$ is essential. If it were not available, we would have had to use cut and we would have been left with the following deduction

$$\dfrac{\dfrac{\dfrac{b \doteq a, \sim[b \doteq a]}{b \doteq a, \sim[a \doteq b], \sim[b \doteq b]}\, S\doteq \qquad b \doteq b}{\sim[a \doteq b], b \doteq a}\, Cut}{a \doteq b \supset b \doteq a}\, \vee I$$

However, in this particular example we could have chosen the other substitution term for $\mathbf{q}$, namely $\mathbf{q} \leftarrow [\lambda x.\sim[x = a]]$. This would have given us a different expansion proof and consequently a different $\mathcal{H}^*$ deduction. For this different expansion proof it would not have been necessary to use $S^*\doteq$, but it cannot always be avoided. The deduction would then have been:

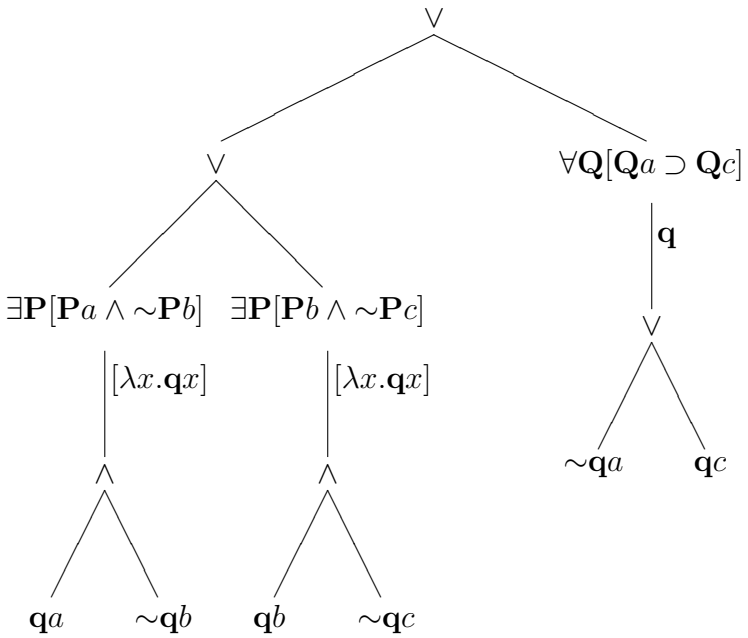
$$\dfrac{\dfrac{\dfrac{}{a \doteq a}\, S\doteq}{\sim[a \doteq b], b \doteq a}\, \vee I}{a \doteq b \supset b \doteq a}$$

Remark 149 This translation also can recover constructive contents in classical proofs. For example, the initial expansion proof of the symmetry of equality (see Example 148) is not intuitionistically valid (when translated into an $\mathcal{H}$-deduction). However, the transformed expansion proof given in the example yields an intuitionistically valid proof of the symmetry of equality.

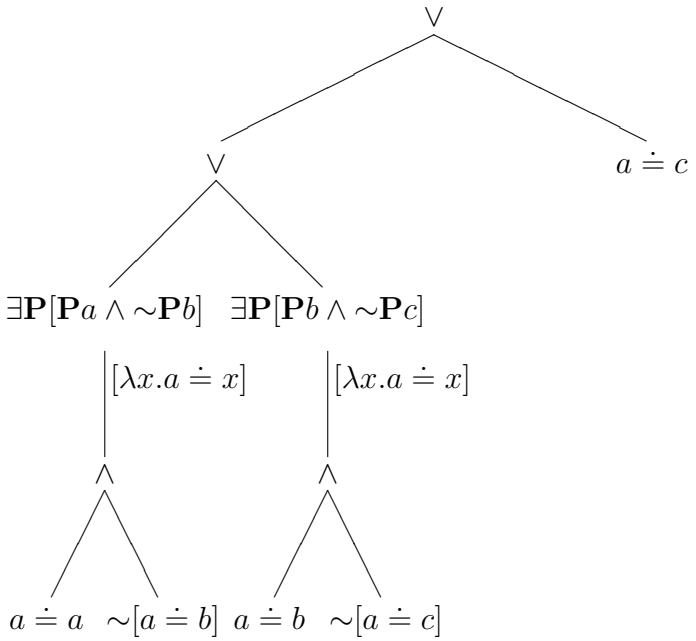
Example 150 $a \doteq b \wedge b \doteq c \supset a \doteq c$.

In this example we will make use of Theorem 146 to simplify the expansion proof. This will allow us to obtain a nice $\mathcal{H}^=$ deduction.

The original expansion tree is

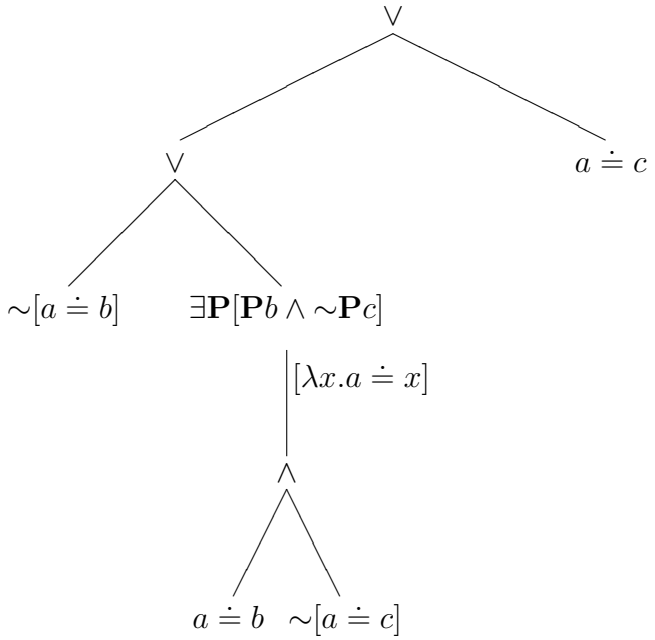


Again, we have two choices for the substituent for $\mathbf{q}$. Let us use $\mathbf{q} \leftarrow [\lambda x.a \doteq x]$. Then the expansion tree becomes



where the tree at node $a \doteq a$ is an initial equality tree in Q.

Now we note that the conditions of Theorem 146 are satisfied for the node $Q_{\exists \mathbf{P}[\mathbf{P}a \wedge \sim \mathbf{P}b]}$. Applying that theorem means to replace that node by the leaf with the same formula which is below it. This results in the following simplified expansion tree



The resulting $\mathcal{H}^=$-deduction is

$$\dfrac{\dfrac{\dfrac{\sim a \doteq b, a \doteq b}{\sim[a \doteq b], \sim[b \doteq c], a \doteq c}\, S\doteq}{\sim[a \doteq b] \vee \sim[b \doteq c], a \doteq c}\, \vee I}{a \doteq b \wedge b \doteq c \supset a \doteq c}\, \vee I$$

Chapter 5

Adding Extensionality

The extension of expansion proofs to be complete for an extensional higher-order logic is fundamentally different from the extension in the previous chapter to deal with equality. Equality is definable and therefore we did not need to extend our notion of expansion proof. Rather we showed how to translate an expansion tree where equality was treated as defined into a deductive system where equality was primitive.

The situation with extensionality is fundamentally different, since the set of theorems changes. Moreover, since extensionality is a theorem schema (parameterized over types) it is not possible to consider the problem as one where we try to show $A \supset B$, where A is a new axiom and B is to be proven assuming A. We therefore need to extend our notion of expansion proof to directly include extensionality. Ideally, we would have a new kind of node, call it *extensionality node*, which somehow serves to represent an application of extensionality.

It turns out that this desired generalization of expansion trees can be done very cleanly. We will first present a cut-free deductive system similar to the one introduced by Takahashi [37] which was also used by Takeuti [40]. Both present proofs that the system described below is cut-free. However, the proofs are non-constructive and neither offers a concrete cut-elimination algorithm.

5.1 The System $\mathcal{H}^e$

One has to be careful when defining $\mathcal{H}^e$, if one wants the resulting system to be cut-free. We also would like to consider extensionality separately from equality, since equality is definable and not a primitive in $\mathcal{H}$ or in expansion trees. Takahashi has a similar system; it differs in that he uses sequents instead of multisets of formulas in a Tait-style inference system.

Formulated as axioms, the extensionality axiom schemas fall into two categories, one for functional types, and one for propositional types. Function types are of the form $\iota\alpha_1 \dots \alpha_n$ for a base-type ι, propositional types are of the form $o\alpha_1 \dots \alpha_n$, where o is the type of propositions. Propositional types includes propositions, sets, and n-ary relations; functional types include

everything else. Remember that $\doteq$ is not primitive, but merely an abbreviation. We thus have the following axiom schemata (for $n \geq 0$):

$$E^{o\alpha_1 \ldots \alpha_n}: \quad \forall x^1_{\alpha_1} \ldots \forall x^n_{\alpha_n} [[px^1 \ldots x^n] \equiv [qx^1 \ldots x^n]] \supset p \doteq q$$

$$E^{\iota\alpha_1 \ldots \alpha_n}: \quad \forall x^1_{\alpha_1} \ldots \forall x^n_{\alpha_n} [[px^1 \ldots x^n] \doteq [qx^1 \ldots x^n]] \supset p \doteq q$$

The cut-elimination theorem no longer holds when the axioms are restricted to the cases $n = 1$ and $n = 0$, though the logical system with cut remains complete.

We would like to have a unified notation for functional and propositional types. Here, and in the remainder of this chapter, let f and g stand for arbitrary formulas in $\mathcal{L}$ of the correct type, which is determined by the context.

Definition 151 Let f_γ and g_γ be given. Then we define $E(f, g)$ by cases.

1. γ is a functional type of the form $\iota\alpha_1 \ldots \alpha_n$. Then

$$E(f, g) \underset{\text{def}}{=} \forall x^1_{\alpha_1} \ldots \forall x^n_{\alpha_n} [[fx^1 \ldots x^n] \doteq [gx^1 \ldots x^n]]$$

   where none of the $x^i_{\alpha_i}$ is free in f or g.

2. γ is a propositional type of the form $o\alpha_1 \ldots \alpha_n$. Then

$$E(f, g) \underset{\text{def}}{=} \forall x^1_{\alpha_1} \ldots \forall x^n_{\alpha_n} [[fx^1 \ldots x^n] \equiv [gx^1 \ldots x^n]]$$

   where none of the $x^i_{\alpha_i}$ is free in f or g.

Then the extensionality axiom reads

$$E^\gamma: \quad E(f, g) \supset f_\gamma \doteq g_\gamma$$

Instead of using an axiom schema, we will follow Takahashi and introduce an extensionality rule. In order for the system to remain cut-free, one has to allow "simultaneous" application of extensionality in the arguments of a property. Note, however, that $\mathbf{P}$ may not be literalic (see Definition 131).

Extensionality Rule

$$\frac{U, E(f^1, g^1) \wedge \ldots \wedge E(f^n, g^n)}{U, \sim\mathbf{P}f^1 \ldots f^n, \mathbf{P}g^1 \ldots g^n} \, Ext$$

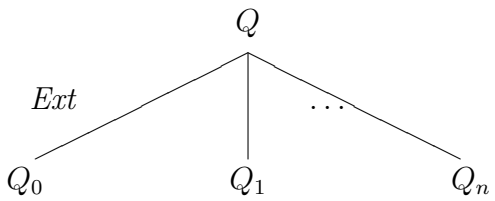
We will call the system which contains all rules of $\mathcal{H}$ plus the extensionality rule $\mathcal{H}^e$.

5.2 Extensional Expansion Proofs

We first define extensional expansion trees by adding another case to the definition of expansion tree (see 35). Then we amend the definition of a full clause to take extensionality nodes into account. Finally we define notion of extensional expansion proof by combining these two.

Definition 152 An *extensional expansion tree* is an expansion tree with an additional type of node, an *extensionality node*. We thus change cases 1 through 5 of Definition 35 by replacing "expansion tree" by "extensional expansion tree" and add the following case:

6. If Q_0 is an extensional expansion tree with shallow formula $\mathbf{P}f_1 \ldots f_n$ and $Q_1, \ldots Q_n$ is are extensional expansion trees with shallow formulas $E(f_1, g_1), \ldots, E(f_n, g_n)$, respectively, then



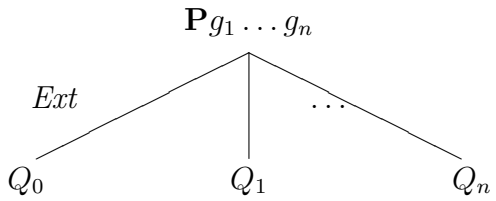
is an extensional expansion proof with $Q^S = \mathbf{P}g_1 \ldots g_n$ and $Q^D = \mathbf{P}f_1 \ldots f_n \wedge E(f_1, g_1) \wedge \cdots \wedge E(f_n, g_n)$.

In order to increase the legibility we will as usual write the shallow formula of an extensionality node instead of the name of the node itself.

Definition 153 A *full clause* in an extensional expansion tree is defined as in Definition 39 with the following additional case.

6.

$Q =$



Then for any full clause c in some Q_i, the list $\langle Q \rangle @ c$ is a full clause in Q.

5.3 Translating Extensional Expansion Proofs into $\mathcal{H}^e$

In this section we will establish soundness of extensional expansion proofs relative to $\mathcal{H}^e$. Algorithm 77 remains basically intact, except that we have to add a case for extensionality nodes in Q.

Algorithm 154 Under the assumption that $Q_{\mathbf{P}g_1\ldots g_n}$ is single (that is not $\mathcal{M}$-mated), we have the following new case

8. $L = U, \mathbf{P}g_1 \ldots g_n$ such that $Q_{\mathbf{P}g_1\ldots g_n}$ is an extensionality node in Q. Then we infer L by

$$\dfrac{U^1, \mathbf{P}f_1 \ldots f_n \qquad \dfrac{E(f_1,g_1) \wedge \cdots \wedge E(f_n,g_n), U^2}{\sim\mathbf{P}f_1 \ldots f_n, \mathbf{P}g_1 \ldots g_n, U^2}\,Ext}{\dfrac{U^1, U^2, \mathbf{P}g_1 \ldots g_n}{U, \mathbf{P}g_1 \ldots g_n}\,C}\,Cut$$

Here U^1 and U^2 are multisets of formulas which are sufficient for $\mathbf{P}f_1 \ldots f_n$ and $\mathbf{P}g_1 \ldots g_n$, respectively. Since $Q_{\mathbf{P}g_1\ldots g_n}$ is single, these sufficient sets exist, perhaps not uniquely. This can be seen by applying Lemma 72 after observing that $(Q^*, \mathcal{M})$ is an expansion proof for $U, \mathbf{P}f_1 \ldots f_n \wedge [E(f_1,g_1) \wedge \cdots \wedge E(f_n,g_n)]$. Here Q^* is the result of replacing the extensionality node $Q_{\mathbf{P}g_1\ldots g_n}$ by a conjunction node with the same successors. $Q_{\mathbf{P}g_1\ldots g_n}$ is accessible and single, so this is a legal operation. Moreover, by definition, it has the same full clauses, if $Q_{\mathbf{P}g_1\ldots g_n}$ is removed. But since $Q_{\mathbf{P}g_1\ldots g_n}$ was not $\mathcal{M}$-mated, every path is still closed.

This argument also shows how to obtain new expansion proofs $(Q', \mathcal{M}')$ for $U^1, \mathbf{P}f_1 \ldots f_n$ and $(Q'', \mathcal{M}'')$ for $U^2, E(f_1,g_1) \wedge \cdots \wedge E(f_n,g_n)$.
Namely, $(Q', \mathcal{M}') = \wedge\mathrm{E}_1(Q^*_{\mathbf{P}f_1\ldots f_n \wedge [E(f_1,g_1)\wedge\cdots\wedge E(f_n,g_n)]}, U^1, (Q^*, \mathcal{M}))$ and
$(Q'', \mathcal{M}'') = \wedge\mathrm{E}_2(Q^*_{\mathbf{P}f_1\ldots f_n \wedge [E(f_1,g_1)\wedge\cdots\wedge E(f_n,g_n)]}, U^2, (Q^*, \mathcal{M}))$. It is easy to check that these have the correct shallow formulas.

Theorem 155 If $(Q, \mathcal{M})$ is an extensional expansion proof for U, then there is an $\mathcal{H}^e$ deduction for U.

Proof: By Algorithm 154. Since $(Q', \mathcal{M}') \sqsubset (Q, \mathcal{M})$ and $(Q'', \mathcal{M}'') \sqsubset (Q, \mathcal{M})$ in case 8 of the algorithm, termination will be assured by the well-founded ordering $\sqsubset$.

□

5.4 Translating from $\mathcal{H}^e$ into Extensional Expansion Proofs

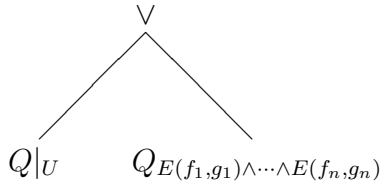
Let us give the algorithm for translating cut-free deductions in $\mathcal{H}$ into extensional expansion proofs. Cuts can be eliminated from $\mathcal{H}^e$ deductions. A proof of the cut-elimination theorem can be found in Takeuti [40] and was anounced earlier by Takahashi [37]. However, the proofs are non-constructive. A constructive proof of cut-elimination in the intuitionistic type theory was given by a student of Gandy's in his thesis [42]. It may be interesting to determine if that result carries over to classical logic.

Algorithm 156 This algorithm follows the pattern of Algorithm 87 except that we have to add another case, where $\mathcal{D}$ ends in an extensionality inference.

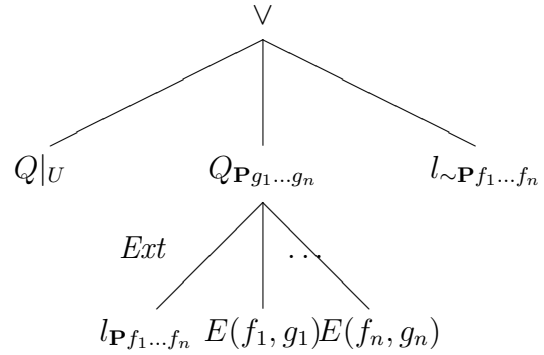
7. $\mathcal{D} = \dfrac{U, E(f_1, g_1) \wedge \cdots \wedge E(f_n, g_n)}{U, \sim\mathbf{P}f_1 \ldots f_n, \mathbf{P}g_1 \ldots g_n}\, Ext.$

   Here we construct a new expansion tree R from Q. Given

$Q =$



then $R =$



   Here $l_{\mathbf{P}f_1\ldots f_n}$ and $l_{\sim\mathbf{P}f_1\ldots f_n}$ are leaves of R. $\mathcal{N} = \mathcal{M} \cup \{(l_{\mathbf{P}f_1\ldots f_n}, l_{\sim\mathbf{P}f_1\ldots f_n})\}$.

Theorem 157 If $\mathcal{D}$ is an $\mathcal{H}^e$ deduction for U, then there is a extensional expansion proof $(Q, \mathcal{M})$ for U.

Proof: By Algorithm 156.

□

5.5 Translation Improvements

Even though Algorithm 154 establishes the soundness of extensional expansion proofs, it is unsatisfactory for practical purposes. The main drawback of the given translation is that it introduces a cut, where it is often inelegant. Unlike the system $\mathcal{H}^=$, $\mathcal{H}^e$ is cut-free and we can therefore always avoid the use of cut.

A simple refinement is similar to the refinement of Algorithm 141 which translates expansion proofs to $\mathcal{H}^=$-deductions. The basic underlying idea is the same: if $\mathbf{P}f_1 \dots f_n$ is a leaf of Q, then either it is unnecessary, or we can defer applying the extensionality inference until one of its mates is part of U. Then we can apply extensionality without the detour of a cut.

Algorithm 158 Assume that every extensionality successor of an extensionality node in an extensional expansion proof $(Q, \mathcal{M})$ for U is a leaf of Q. Then there is a cut-free $\mathcal{H}^e$-deduction of U.

We replace Case 8 in Algorithm 154 with the following case

8. $L = U, \sim\mathbf{P}f_1 \dots f_n, \mathbf{P}g_1 \dots g_n$ such that $Q_{\mathbf{P}g_1 \dots g_n}$ is an extensionality node in Q with extensionality successor $Q_{\sim\mathbf{P}f_1 \dots f_n}$. Then we infer L by

$$\mathcal{D} = \frac{U, E(f_1, g_1) \wedge \dots \wedge E(f_n, g_n)}{U, \sim\mathbf{P}f_1 \dots f_n, \mathbf{P}g_1 \dots g_n}\, Ext$$

We do not give a complete proof here, but simply indicate how a correctness proof could proceed.

The critical step in proving correctness of this refined algorithm is to show that at least one of the cases will always be applicable. First we transform Q to an expansion tree such that no l depends on any of its mates.

Then we distinguish cases.

1. $Q_{\mathbf{P}f_1 \dots f_n}$ is not $\mathcal{M}$-mated. Since every full clause through Q is spanned by $\mathcal{M}$, every full clause through $Q_{\mathbf{P}f_1 \dots f_n}$ is spanned by some pair (l, k). This same pair will then also span every full clause through $Q_{E(f_i, g_i)}$ for $1 \leq i \leq n$. Therefore we can erase the whole tree below $Q_{\mathbf{P}g_1 \dots g_n}$ and convert it into a leaf with the same shallow formula. This modified expansion tree remains an expansion proof with the same shallow formula. Hence we will never need to apply extensionality to the given occurrence $\mathbf{P}g_1 \dots g_n$.

2. $Q_{\mathbf{P}f_1 \dots f_n}$ is $\mathcal{M}$-mated. Let k be a minimal (with respect to $<_Q$ extended to literals) mate of $Q_{\mathbf{P}f_1 \dots f_n}$. By assumption either $Q_{\mathbf{P}g_1 \dots g_n}$ will be unnecessary, other inferences are possible, or the line has the form $U', \sim\mathbf{P}f_1 \dots f_n, \mathbf{P}g_1 \dots g_n$.

Clearly, this method cannot be extended to the case where the extensionality successor is not a leaf. If the extensionality successor is not a leaf, one cannot defer applying the extensionality rule since its complement may not even appear in the expansion proof.

The only avenue open here is to somehow transform the deduction into one where all extensionality successors are leaves. It is not clear whether this is of practical value for providing a translation algorithm from extensional expansion proofs to $\mathcal{H}^e$ deductions, since the use of cut in such cases actually seems quite natural. However, this transformation would provide a completeness proof for a search procedure where one only creates an extensionality node from a leaf, where the extensionality successor is a leaf already present and shares a path with the extensionality node.

As a first step we establish that it is sufficient if we restrict ourselves to literalic $\mathbf{P}$.

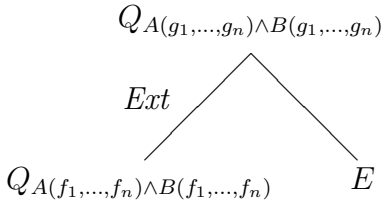
Definition 159 We call an extensionality node *literalic* if its shallow formula is of the form $\mathbf{P}g_1 \dots g_n$, the extensionality successor is of the form $\mathbf{P}f_1 \dots f_n$, and $\mathbf{P}$ is literalic.

Theorem 160 Given an extensional expansion proof $(Q, \mathcal{M})$ for U. Then there is an extensional expansion proof $(Q', \mathcal{M}')$ for U such that all extensionality nodes are literalic.

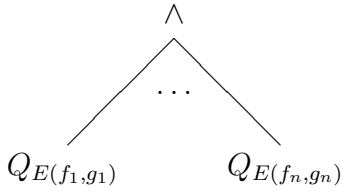
Proof: We will show how to eliminate any given non-literalic extensionality node in favor of several literalic ones. This will be done without creating new ones in the rest of Q, so the theorem follows by induction on the number of non-literalic extensionality nodes. However, the order in which we eliminate the non-literalic extensionality nodes is important. At each stage we eliminate some extensionality node which is minimal with respect to $\lessdot$.

The proof that any given non-literalic extensionality node can be eliminated is by induction on the depth of the expansion proof below the extensionality node. It follows a pattern similar to the proof of Theorem 132, but is somewhat simpler. First note that we may assume that $\mathbf{P}$ is of the form $[\lambda x_1 \dots \lambda x_n.A_o(x_1, \dots, x_n)]$. This is true since there is always a wff A such that $\mathbf{P}g_1 \dots g_n = [\lambda x_1 \dots \lambda x_n.A_o(x_1, \dots, x_n)]g_1 \dots g_n$.

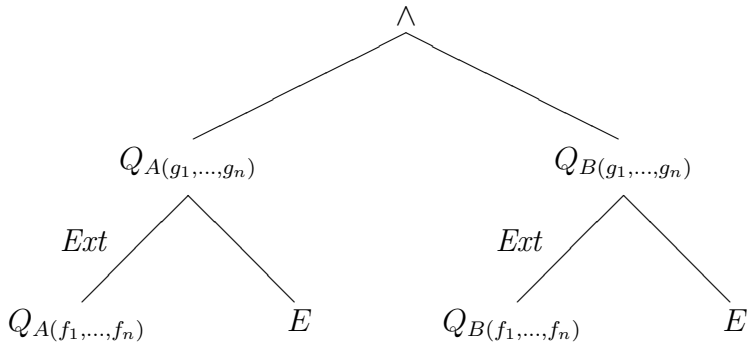
1. $\mathbf{P} = [\lambda x_1 \dots \lambda x_n.A(x_1, \dots, x_n) \wedge B(x_1, \dots, x_n)]$. Then $Q_{\mathbf{P}g_1 \dots g_n}$ has the form



   where E stands for the tree



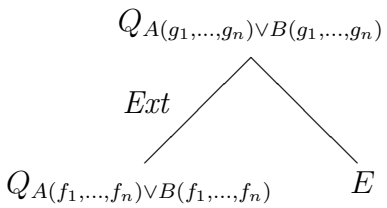
Here we change the extensionality node into a conjunction node, introducing two new extensionality nodes, but with a simpler form of $\mathbf{P}$.



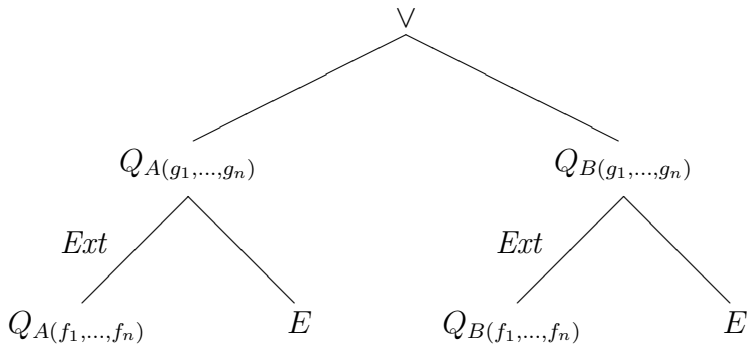
The new tree Q' has the same shallow formula, namely $A(g_1,\ldots,g_n)\wedge B(g_1,\ldots,g_n)$. Since we are copying E we will have to rename selected parameters and double some expansion terms. Since $Q_{\mathbf{P}g_1\ldots g_n}$ was chosen $\prec\!\!\prec_Q$-minimal, no other non-literalic expansion terms will be doubled (as in the proof of Theorem 132).

$\mathcal{M}'$ is like $\mathcal{M}$, except for the possible duplication of some pairs. It is easy to see that $\mathcal{M}'$ is clause-spanning.

2. $\mathbf{P} = [\lambda x_1 \ldots \lambda x_n.A(x_1,\ldots,x_n) \vee B(x_1,\ldots,x_n)]$. Then $Q_{\mathbf{P}g_1\ldots g_n}$ has the form



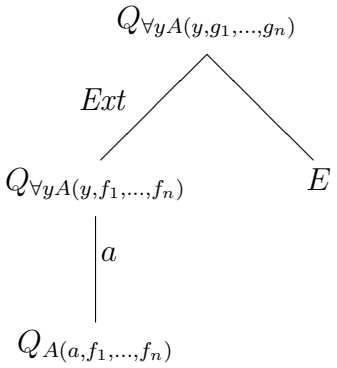
Here we change the extensionality node into a disjunction node, introducing two new extensionality nodes, but simpler than $\mathbf{P}$.



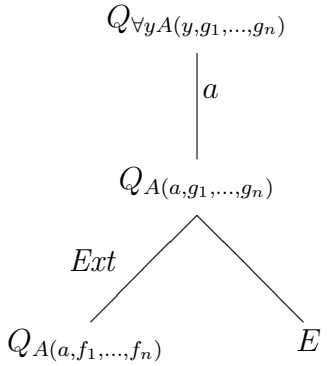
As in the previous case, the shallow formula stays the same. Also, we will again have to duplicate because of renaming of the parameters selected in the two copies of E.

The new mating $\mathcal{M}'$ is obtained as in the proof of Theorem 132. It is clause-spanning.

3. $\mathbf{P} = [\lambda x_1 \ldots \lambda x_n . \forall y A(y, x_1, \ldots, x_n)]$. Then $Q_{\mathbf{P}g_1 \ldots g_n}$ has the form

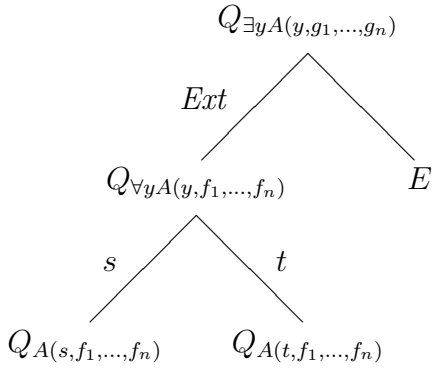


Again, we push down the application of extensionality.

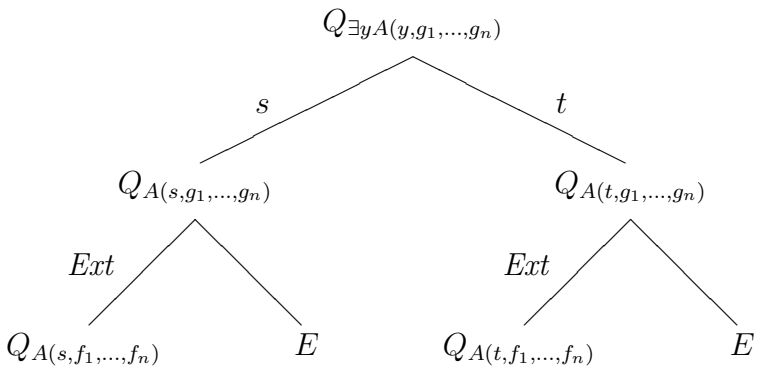


Here we have to make sure that a is not free in any expansion term in E. This can be achieved by duplication (see Lemma 129).

4. $\mathbf{P} = [\lambda x_1 \ldots \lambda x_n . \exists y A(y, x_1, \ldots, x_n)]$. Then $Q_{\mathbf{P}g_1 \ldots g_n}$ has the form (we assume without loss of generality that $Q_{\exists yA(y,f_1,\ldots,f_n)}$ has at most two expansion nodes.



Again, we push down the application of extensionality.



$\mathcal{M}'$ is obtained like in the case of a disjunction node and is therefore clause-spanning. Again, duplication of E requires renaming of some parameters, but without duplication of other non-literalic extensionality nodes, since $Q_{\mathbf{P}g_1\ldots g_n}$ was chosen $\prec\!\!\prec_Q$-minimal.

□

Chapter 6

Applications

The logical system $\mathcal{H}$ was well suited to demonstrate the role of expansion proofs with respect to deductions. However, some useful applications of the algorithms and transformations are not directly covered by the material presented in Chapters 3 and 4. These extensions are centered around the observation that sequent-style systems like $\mathcal{H}$ are still not very well suited for the human reader — natural deductions seem to fill this role best. The asymmetry between (possibly many) assumptions and the single desired conclusion seem to be crucial and cannot be achieved in the classical sequent calculus, where one may sometimes have to use several "conclusions".

In this chapter we show how to apply the translation paradigms presented in previous chapters to obtain natural deductions and also present significant extensions which are motivated by the goal of achieving intuitive natural deductions. Part of this will be to use formulas directly in natural deductions instead of equivalence classes of formulas we used in $\mathcal{H}$.

First we define a system of natural deduction and show how expansion proofs may be transformed into natural deductions by means of proof transformation tactics. Proof transformation tactics are in the spirit of rule-based computation systems and provide a modular and succinct way of describing the rules and heuristics governing the translations. The natural deduction systems presented by Miller in [23] and [22] used many derived rules of inference which makes them less useful in a pedagogical setting where one is interested in teaching natural deductions. It should not be a precondition for the translation that derived rules like RuleP (which allows one to assert any tautologous formula as an axiom) are available. However, we do not preclude the use of derived rules of inference. On the contrary, the rule-based presentation is ideal for adding derived rules of inference and tactics which suggest them in situations where they are useful.

Next we show how to further refine the initial translation by providing tactics which do not produce normal deductions, but make judicious use of cut to obtain more intuitive deductions in many cases. The heart of these lemma introducing tactics is a procedure called *symmetric simplification* which determines formulas which may serve as useful lemmas and also produces expansion proofs for the subdeductions which must be carried out when the suggested lemma is used. Symmetric simplification is by no means a complete method for finding lemmas — only certain types of lemmas can be found. However, these lemmas turned out to be very useful and

intuitive in many examples.

These two extensions show clearly that expansion proofs are extremely useful as an abstract representation of the contents of a proof and can serve as high-level plans in constructing natural deductions. As an application, logic students can be given very sophisticated help when they are trying to prove a theorem within a system of natural deduction on the computer. In this setting a student works on what Miller [23] calls proof outlines, that is, deductions in which gaps remain to be filled. The student then enters the rule of inference he wishes to apply and where, and the rule will be applied if legal. When the student asks for advice, the tactics are tested for applicability and the applicable ones are suggested. An implementation of expansion proofs and proof transformation tactics is currently under way in the TPS project at Carnegie-Mellon University (see, for example, Andrews [3]). The translation from $\mathcal{H}$-deductions into expansion proofs presented in Section 3.4 can be extended to natural deductions in a straightforward way. This extended translation algorithm can be used to interpret student's deduction as expansion proofs, and perhaps show them how they could have produced the same deduction more elegantly. Another use allows the teacher to give a sample deduction for an exercise and have it converted automatically to an expansion proof. This expansion proof could then be used to give help for a large variety of partial deductions which differ from the teacher's in many ways, such as inessential instantiation terms, or a different order of steps. It could even detect cases when the student attempts to use parameters which are not yet free in the deduction and give an appropriate warning. This particular error seems to be one of the biggest initial stumbling blocks for students who are learning to prove theorems in the predicate calcnulus.

6.1 A System $\mathcal{N}$ of Natural Deduction

Our system of natural deduction which we describe below is basically Gentzen's system NK [11] or the system which may be found in Prawitz [28]. A detailed, goal-oriented presentation of the intuitionistic fragment may be found in Zucker [41]. In our presentation we use the rule of indirect proof rather than the double-negation rule, thus combining what would be two deduction steps in NK into one.

Because of the complications which are introduced in higher-order logic due to the fact that expansion proofs are based on equivalence classes of formulas, we will divide the presentation into two parts. The first part deals with the translation from first-order expansion proofs into first-order natural deductions. The second part deals with the higher-order constructs.

The translation into the first-order fragment has many applications in its own right, since most logic courses do not cover higher-order logic. Thus the reader who is only interested in the first-order fragment of the logical system need not be burdened with the complications which arise when higher-order quantifiers are introduced.

In our context, we think of natural deduction as a process rather than the completed deduction. In this goal-oriented presentation, we are trying to solve a collection of goals at each stage in the deductive process. A *line* is a formula in the tree representing the deduction. A goal is defined by a *planned line* (the theorem) and a multi-set of *support lines* which may be used to

prove the goal. Support lines usually are assumptions or consequences thereof.

Remark 161 For practical purposes it is extremely important to keep the set of support lines small in order to avoid the combinatorial explosion which results when one keeps accumulating support lines monotonically. Often the decision whether to retain or drop support lines can be made even without the aid of an expansion proof. For example, once one has inferred both A and B from a conjunctive support line $A \wedge B$, one no longer needs $A \wedge B$ as a support line. Sometimes expansion proof information is vital, as is the case when one instantiates a universal quantifier in a support line. One may need the universal statement more than once and thus cannot delete it from the support, unless one can check (as can be done through the expansion proof) that no more instantiations are needed.

The formula language underlying the natural deduction system now contains implication, negation, and falsehood as primitive. In order to distinguish them from the symbols in $\mathcal{L}$, we will write them as $\rightarrow$, $\neg$, and $\perp$ respectively. We use a different symbol for negation to remind the reader that dealing with formulas rather than equivalence classes of formulas. The language of types has not changed. We will only present the definition of the first-order fragment using type symbols — those could be easily elimiated from the definition if desired. The straightforward extension to a full higher-order language is given in Definition 245. No confusion should arise when we use the identical symbols $\wedge$, $\vee$, $\forall$, and $\exists$ in $\mathcal{LN}$ and $\mathcal{L}$, since intuitively they have the same meaning and it will be clear from the context which language the formula belongs to.

Remember that o is the type of propositions, ι is the type of individuals, and $\alpha\beta$ is the type of functions from elements of type β to elements of type α.

Definition 162 A formula in the language $\mathcal{LN}^1$ is defined inductively by

1. A variable x_α is a formula of type α.
2. A parameter a_α is a formula of type α.
3. A constant k_α is a formula of type α.
4. $[B_{\alpha\beta}C_\beta]$ is a formula of type α for any formulas $B_{\alpha\beta}$ and C_β.
5. $\perp$ is a formula of type o.
6. $\neg A_o$ for A_o a formula of type o.
7. $A_o \wedge B_o$ for formulas A_o and B_o.
8. $A_o \vee B_o$ for formulas A_o and B_o.
9. $A_o \rightarrow B_o$ for formulas A_o and B_o.
10. $\forall x_\iota A_o$ for a variable x_ι and formula A_o.

11. $\exists x_\iota A_o$ for a variable x_ι and formula A_o.

Note that the quantifiers are restricted to range over individuals. The removal of this restriction and the addition of λ-abstraction are the only changes needed in order to extend natural deductions to the higher-order language to $\mathcal{LN}$.

We now present the inference rules of the system $\mathcal{N}$ of natural deduction. Cancelled assumptions are written as $[\![A]\!]$. An assumption is cancelled in the inference with the same superscript. An inference may cancel 0 or more assumptions with the same formula, so the inference remains correct of the assumption does not occur in the deduction. Later in this chapter we will mostly omit these superscripts, since it should be obvious where the assumptions are cancelled.

Definition 163 (*System* $\mathcal{N}$)

1. Minimal Propositional Rules.

$$\frac{A \qquad B}{A \wedge B}\,\wedge I \qquad \frac{A \wedge B}{A}\,\wedge \mathrm{E}_L \qquad \frac{A \wedge B}{B}\,\wedge \mathrm{E}_R$$

$$\frac{A}{A \vee B}\,\vee \mathrm{I}_L \qquad \frac{B}{A \vee B}\,\vee \mathrm{I}_R \qquad \frac{\begin{array}{ccc} & [\![A]\!]^1 & [\![B]\!]^1 \\ & \vdots & \vdots \\ A \vee B & C & C \end{array}}{C}\,\vee \mathrm{E}^1$$

$$\frac{\begin{array}{c}[\![A]\!]^1 \\ \vdots \\ B\end{array}}{A \to B}\,\to \mathrm{I}^1 \qquad \frac{A \qquad A \to B}{B}\,\to \mathrm{E}$$

$$\frac{\begin{array}{c}[\![A]\!]^1 \\ \vdots \\ \bot\end{array}}{\neg A}\,\neg \mathrm{I}^1 \qquad \frac{A \qquad \neg A}{\bot}\,\neg \mathrm{E}$$

2. Intuitionistic Absurdity Rule

$$\frac{\bot}{A}\,\bot_I$$

3. Classical Proof by Contradiction

$$\begin{array}{c} [\![\neg A]\!]^1 \\ \vdots \\ \dfrac{\bot}{A}\,\bot_C{}^1 \end{array}$$

4. Quantification Rules.

$$\frac{A(a)}{\forall x A(x)}\,\forall\mathrm{I}$$

where a is a parameter not free in $\forall x A(x)$ or any uncancelled assumption.

$$\frac{A(t)}{\exists x A(x)}\,\exists\mathrm{I}$$

$$\frac{\forall x A(x)}{A(t)}\,\forall\mathrm{E}$$

$$\begin{array}{cc} & [\![A(a)]\!]^1 \\ & \vdots \\ \exists x A(x) & C \end{array}\Bigg/\, C\;\exists\mathrm{E}^1$$

where a is a parameter not free in $\exists x A(x)$, C, or any uncancelled assumption.

6.2 Tactics for Constructing Natural Deductions

6.2.1 Proof Transformation Tactics

Tactics as they were introduced in Gordon et al. [14] embody control knowledge in the search for a proof. They can be thought of as functions which map a goal to a list of subgoals and a validation which shows how to construct a solution for the goal, given the solutions to the subgoals. A tactic may also fail. The basic tactics are most commonly described in rule format: a set of conditions which must be met for the tactic to be applicable and a transformation which maps the goal into the subgoals. More complicated tactics are then built up from basic tactics by combining them through the use of tacticals. Examples of tacticals are `OR`, `COMPOSE`, `ITERATE`, or `TRY`. T `OR` S first tries to apply T. If that fails it tries to apply S. T `COMPOSE` S tries to apply T and then S to every subgoal which is created by T. `ITERATE` S applies S repeatedly

to the created subgoals, until it fails. `TRY` T tries to apply T. If there are sugoals remaining `TRY` T will fail, otherwise succeed (with the empty list of subgoals). The practicality of tactics as a language for expressing search-control knowledge in theorem proving has been demonstrated for example by Prl [5] and also by Mulmuley [25].

In the setting of natural deductions, a goal consists of a planned line and a collection of support lines. The validations are short pieces of deductions connecting the subgoals to the original goal. We can view the validation as a function from a list of deductions to a deduction. Its arguments are the completed subdeductions, its result is the deduction for the original problem.

A transformation tactic is similar, except that we make explicit use of a plan to achieve a goal. Thus the data we manipulate are pairs consisting of a goal and a plan.

In our setting a proof transformation tactic operates on pairs, each pair consisting of a goal and an expansion proof. The goal is a planned line B and a collection of support lines $A_1, \ldots, A_n$, the expansion proof has shallow formula $\sim\overline{A_1} \vee \ldots \vee \sim\overline{A_n} \vee \overline{B}$. Here $\overline{}$ is the natural translation of formulas in $\mathcal{LN}$ into $\mathcal{L}$.

We will not give a very formal description of how the basic tactics are combined to one tactic which constructs the complete natural deduction given an expansion proof for the theorem. We will describe the interactions between the different basic tactics informally.

The language $\mathcal{LN}$ contains a new primitive constant, $\bot$. It cannot be interpreted directly in $\mathcal{L}$. However, our general formulation of expansion trees will allow us to model $\bot$ as the empty disjunction, and $\neg\bot$ as the empty conjunction.

Thus we extend $\mathcal{L}$ to $\mathcal{L}'$ by adding `T` and `F` as new constants.

Definition 164 (*Language* $\mathcal{L}'$) We extend the language $\mathcal{L}$ to language $\mathcal{L}'$ by adding the following clauses to Definition 2

11. `F` is a formula of type o.
12. `T` is a formula of type o.

We also extend the definition of negation conversion to include rules concerning `T` and `F`.

Definition 165 (*Negation Conversion in* $\mathcal{L}'$) We extend the $\sim$-conversion rules by the equations

6. $\sim\mathtt{F} \underset{\sim}{\longrightarrow} \mathtt{T}$.
7. $\sim\mathtt{T} \underset{\sim}{\longrightarrow} \mathtt{F}$.

Note that in the extended $\lambda^{\sim}$-normal form, `T` and `F` will not appear in the scope of any negations.

The general definition of expansion tree did not preclude empty conjunctions or disjunctions. We now allow these also in expansion proofs with deep and shallow formulas in $\mathcal{L}'$ and interpret the empty disjunction as $\mathtt{F}$ and the empty conjunction as $\mathtt{T}$.

As before, we restrict the expansion tree, but this time so that all disjunction or conjunction nodes (except for the top-level disjunction node) are either binary or nullary.

Definition 166 (*Empty Conjunction and Disjunction*)

1. If Q is a disjunction node with no successors, then $Q^S = Q^D = \mathtt{F}$. Let us call this expansion tree $Q^{\mathtt{F}}$.
2. If Q is a conjunction node with no successors, then $Q^S = Q^D = \mathtt{T}$. Let us call this expansion tree $Q^{\mathtt{T}}$.

Remark 167 Full clauses were defined in such a way that empty disjunctions and conjunctions are handled correctly (see Definition 39). In particular, one can check that the empty mating $\{\}$ spans every full clause in a conjunction node with no successors, which means that $(Q^{\mathtt{T}}, \{\})$ is an expansion proof for $\mathtt{T}$. Also, if Q is the disjunction node with no successors, then $\langle Q \rangle$ is the only full clause in Q.

Definition 168 $\overline{}$ is the natural translation from formulas in $\mathcal{LN}$ into $\mathcal{L}'$. Inductively,

1. $\overline{\bot} = \mathtt{F}$.
2. $\overline{\neg A} = \sim\overline{A}$.
3. $\overline{A \wedge B} = \overline{A} \wedge \overline{B}$.
4. $\overline{A \vee B} = \overline{A} \vee \overline{B}$.
5. $\overline{A \to B} = \overline{A} \supset \overline{B} = \sim\overline{A} \vee \overline{B}$.
6. $\overline{\exists x A(x)} = \exists x \overline{A(x)}$.
7. $\overline{\forall x A(x)} = \forall x \overline{A(x)}$.

The writing and reading of overlined formulas tends to get tiresome, so we have mostly omitted overlines. The subscripts for expansion trees which represent their shallow formulas should of course be overlined. This extends to multisets of support lines, when they need to appear explicitly to denote parts of expansion trees. Wherever confusion is possible, we will make explicit use of the natural translation as defined above.

We have five classes of tactics:

1. bookkeeping tactics which do not apply inference rules,

2. minimal planned line tactics which suggest introduction rules,
3. minimal support line tactics which suggest elimination rules,
4. non-minimal tactics which suggest indirect proof or absurdity rule,
5. lemma tactics which introduce formulas which are not necessarily subformulas of the planned line or the support line and do not lead to normal deductions.

The main connective of the planned line or support line suggests a possible inference. These suggested inferences are not always possible. Sometimes they have to be deferred until they become applicable, other times only an application of indirect proof or the absurdity rule will allow the deduction to progress.

The benefit of using expansion proofs as plans becomes striking here. We will have to decide whether a certain rule may be applied profitably at a given step in the deduction. The expansion proof allows this decision. In general the decisions involve solving an NP-complete problem, but in practice they are feasible and actually quite fast. Without the guidance of expansion proofs, one would have to invoke a general-purpose theorem prover to decide whether certain steps are applicable. This is possible and probably feasible in the propositional calculus, but in first-order logic it is of course undecidable. Still one may argue that even in the first-order case invoking a theorem prover may not be so bad. However, most theorem provers are meant to prove theorems, rather than to show that something is not a theorem. It is exactly those quick failures which make expansion proofs so valuable. For example, consider a planned line $A \vee B$. We could infer it from A by $\vee \mathrm{I}_L$ if we knew that A by itself would already follow from the support lines, and from B if B by itself would follow from the support lines. However, such decisions have to be made often and we cannot wait for the time-out of a theorem prover if neither of the disjuncts follows directly. The expansion proof information is vital, because it fails quickly if neither a proof of A nor a proof of B is possible (relative to the expansion proof).

Moreover, the algorithm for converting deductions into expansion proofs given in Section 3.4 shows that it is sufficient to give a *deduction* of a theorem — no automated theorem prover is necessary to obtain the expansion proofs. This makes the view of expansion proofs as a concise representation of the contents of a proof very attractive. They abstract the inessential details, like the order of many rule applications, but they preserve the essential, namely the instantiation terms and the mating. Especially in higher-order logic where little is known about good procedures for finding instantiation terms, this information is indispensable in proof guidance.

Let us give the name *minimal tactics* to any tactic bookkeeping, planned line or support line tactic. Minimal tactics will produce deductions in minimal logic with the strict subformula property. The *strict subformula property* requires that each formula in the deduction is an instance of a subformula of the theorem we are trying to prove, or $\bot$. The addition of tactics suggesting indirect proof produces deductions satisfying only the weak subformula property. The *weak subformula property* states that each formula in the deduction is either an instance of a subformula, or a negation of such an instance. When we generalize $\mathcal{L}^1$ to the higher-order language $\mathcal{L}$ these concepts become less meaningful, because higher-order quantifiers allow instantiations with formulas. We will still be able to see the distinction between deductions

which are normal and those which are not. Normal natural deductions roughly correspond to cut-free $\mathcal{H}$-deductions (see Zucker [41] for an explanation of the connections between "normal" and "cut-free" for intuitionistic natural deductions and sequent calculi, respectively). If we translate into normal deductions, we do not have to "invent" formulas which do not already appear in the expansion proof.

We will give very suggestive names to the tactics by relating them to the rules which they apply. The reader should be keep in mind that tactics unlike inference rules have conditions associated with them which may refer to the expansion proof and ensure that the new subgoals can be solved.

The tactics are organized so that most of them apply only to planned lines or to support lines with one particular main connective. Additional applicability conditions ensure that the resulting subgoals can be solved. There are two types of impasses that one can encounter. The *temporary impasse* occurs when a particular planned or support line can not be broken down, because the applicability condition of the rule is not satisfied, but other tactics can be applied to other lines so that eventually the tactic becomes applicable. A *critical impasse* occurs if none of the bookkeeping, planned line or support line tactics apply to any planned or support line. It is often hard to decide if a given impasse is temporary or critical, and we just continue to apply the tactics until the impasse is resolved or no further minimal tactics can be applied. When a critical impasse is encountered only the absurdity rule or indirect proof can be applied. We also distinguish between *planned line impasses*, when application of a tactic suggesting an introduction rule is impossible, and *support line impasses*, when application of a tactic suggesting an elimination rule is impossible. Let us give two examples to illustrate those types of impasses.

Example 169 $A \vee B \to B \vee A$. After one step in the deduction process we reach

$$\dfrac{\begin{array}{c}[\![A \vee B]\!] \\ \vdots \\ B \vee A\end{array}}{A \vee B \to B \vee A} \to\mathrm{I}$$

Here we have a temporary planned line impasse, since $\vee I$ has to be deferred until we have distinguished cases, that is, used $\vee$E on the support line. After the use of $\vee$E, the impasse is resolved in both subgoals.

$$\dfrac{\dfrac{\dfrac{[\![A]\!]}{B \vee A} \vee\mathrm{I}_R \qquad \dfrac{[\![B]\!]}{B \vee A} \vee\mathrm{I}_L \qquad\qquad [\![A \vee B]\!]}{B \vee A} \vee\mathrm{E}}{A \vee B \to B \vee A} \to\mathrm{I}$$

Example 170 $A \vee \neg A$. Here neither disjunct can be inferred directly, so again we have a planned line impasse. Moreover, the impasse is critcal, since no other minimal tactic applies (there are no assumptions). We have to use the rule of indirect proof.

Looking ahead a bit, let us give the deduction which is obtained without the use of tactics which violate the weak subformula property.

$$\dfrac{\dfrac{\dfrac{[\![A]\!]}{A \vee \neg A}\vee\mathrm{I}_L \qquad [\![\neg[A \vee \neg A]]\!]}{\dfrac{\bot}{\neg A}\neg\mathrm{I}}\neg\mathrm{E}}{\dfrac{[\![\neg[A \vee \neg A]]\!] \qquad \dfrac{\dfrac{\bot}{A \vee \neg A}\vee\mathrm{I}_R}{A \vee \neg A}\neg\mathrm{E}}{\dfrac{\bot}{A \vee \neg A}\bot_C}}$$

The alternative, more symmetric deduction given below is harder to find. However, it can be found using the tactic presented in Section 6.5. Note that this deduction is *not normal*, since an application of ¬I is followed by a ¬E. Normalizing this deduction leads to the one given above.

$$\dfrac{\dfrac{\dfrac{[\![A]\!]}{A \vee \neg A}\vee\mathrm{I}_L \qquad [\![\neg[A \vee \neg A]]\!]}{\dfrac{\bot}{\neg A}\neg\mathrm{I}}\neg\mathrm{E} \qquad \dfrac{\dfrac{\dfrac{[\![\neg A]\!]}{A \vee \neg A}\vee\mathrm{I}_R \qquad [\![\neg[A \vee \neg A]]\!]}{\dfrac{\bot}{\neg\neg A}\neg\mathrm{I}}\neg\mathrm{E}}{\dfrac{\bot}{A \vee \neg A}\bot_C}$$

Remark 171 Because expansion proofs are based on formula equivalence classes a problem which has an almost trivial expansion proof sometimes can be quite complex. There we need to use transformations which only change the expansion proof trivially, but make progress in the natural deduction. The next examples illustrates this.

Example 172 $\neg\neg A \to A$.

$$\dfrac{\dfrac{[\![\neg\neg A]\!] \qquad [\![\neg A]\!]}{\dfrac{\bot}{A}\bot_C}\neg\mathrm{E}}{\neg\neg A \to A}\to\mathrm{I}$$

In the following subsections we will present a complete set of tactics which produces normal deductions. The presentation is devised to be highly modular. This was motivated by the desire to have one, well-defined "action", that is, transformation for a tactic. This is generally in the spirit of rule-based systems, and should aid implementation.

6.2.2 Bookkeeping Tactics

The most important here is called *Closing a Gap*. It notices if a planned line is identical to a support line and declares a subgoal as achieved. In practice, this may be most useful if composed with other tactics, so that the trivial subgoals are not even created. This is the only basic tactic which achieves a goal, that is, returns an empty list of subgoals.

Tactic 173 (*Closing a Gap*) If a planned line and a support line are identical we can close the gap – the subgoal is achieved.

Example 174 $A \to A$.

$$\begin{array}{c}\vdots\\ A \to A\end{array} \quad \Longrightarrow \quad \begin{array}{c}[\![A]\!]\\ \vdots\\ A\end{array}\over{A \to A}\,{\to}\mathrm{I} \quad \Longrightarrow \quad \frac{[\![A]\!]}{A \to A}\,{\to}\mathrm{I}$$

Remark 175 In order to facilitate a fast check for identical planned and support lines, one may add the restriction to the tactic, that $Q_{\sim A}$ (corresponding to the support line) and Q_A (corresponding to the planned line) be $\mathcal{M}$-mated. This restriction will not affect the completeness of the algorithm. One could say, that if the expansion proof "did not notice" of "did not use the fact" that those two formulas are complementary, perhaps the natural deduction does not need to either.

Tactic 176 (*Inessential Support Line*) If the expansion tree $Q_{\sim A}$ corresponding to a support line A is *inessential*, it may be dropped from the support (and the expansion proof).

Remark 177 A sufficient condition: Q_A is inessential in $(Q, \mathcal{M})$ if no node in Q_A is $\mathcal{M}$-mated.

Remark 178 A refinement: Q_A is inessential in $(Q, \mathcal{M})$ if there is a submating $\mathcal{M}'$ of $\mathcal{M}$ clause-spanning on Q such that no node in Q_A is $\mathcal{M}'$-mated.

Remark 179 A further refinement: Q_A is inessential in $(Q, \mathcal{M})$ if there is an extension $\overline{\mathcal{M}}$ of $\mathcal{M}$ such that there is a submating $\mathcal{M}'$ of $\overline{\mathcal{M}}$ clause-spanning on Q such that no node in Q_A is $\mathcal{M}'$-mated.

Remark 180 All of these can be computed efficiently in practice, though theoretically it is an NP-complete problem to determine if a subtree is essential.

Tactic 181 (*Erasing Unnecessary Expansion Terms*) If the planned line is of the form $\exists x A(x)$ and $Q_{\exists x A(x)}$ has more than one expansion term, erase any unnecessary expansion terms (see Definition 93).

Remark 182 This tactic may be relatively expensive to check, so it is best to apply it only directly after an application of the $\vee$ *Elimination* tactic (see Tactic 202), because an application of $\vee$E to the support lines is what may render an expansion term unnecessary. Other elimination rules applied to the support lines will not affect whether an expansion term is unnecessary or not.

Tactic 183 (*Mated Support Line*) If a support line A corresponds to $Q_{\sim A}$, where $Q_{\sim A}$ is not a leaf, but is $\mathcal{M}$-mated, create a new support line with the same formula A. The new expansion tree is obtained as $(R, \mathcal{N}) = \texttt{double}(Q_A, (Q, \mathcal{M}))$ (see Definition 76.1.) Even though we are creating more planned lines, we have $(R, \mathcal{N}) \sqsubset_C (Q, \mathcal{M})$ and therefore $(R, \mathcal{N}) \sqsubset' (Q, \mathcal{M})$ (see Definitions 59 and 240).

Tactic 184 (*Mating Simplification*) Given a planned line A and node Q_A corresponding to A. If Q_A is not a leaf, but $\mathcal{M}$-mated, check if $\mathcal{M}$ restricted to pairs not containing Q_A is still clause-spanning on Q. If yes, restrict $\mathcal{M}$.

The next tactic is necessary to take care of the problems which arise due to the fact that expansion proofs are allowed to mate formulas which are equal only up to $\lambda^{\sim}$-conversion.

Tactic 185 (*Deepening the Expansion Proof*) Assume there is a support line A and a planned line B such that $\overline{A} \underset{\lambda^{\sim}}{=} \overline{B}$, and $Q_{\overline{A}}$ and $Q_{\overline{B}}$ are $\mathcal{M}$-mated. In such a case one must deepen the expansion proof, that is write out one more level of the formula as a node, making new leaves below. Formally, let $(R, \mathcal{N}) = \texttt{deepen}(Q_{\overline{A}}, (Q, \mathcal{M}))$ (see Definition 124).

6.2.3 Minimal Planned Lines Tactics

Throughout this section we will assume that the node Q_C which corresponds to the planned line C is neither a leaf of Q, nor $\mathcal{M}$-mated.

Tactic 186 ($\wedge$ *Introduction*)

$$\begin{array}{c} \mathcal{S} \\ \vdots \\ A \wedge B \end{array} \quad \rightarrow \quad \dfrac{\begin{array}{cc} \mathcal{S}_A & \mathcal{S}_B \\ \vdots & \vdots \\ A & B \end{array}}{A \wedge B} \wedge I$$

where $\mathcal{S}_A$ sufficient for A and $\mathcal{S}_B$ sufficient for B. Both are sub(multi)sets of $\mathcal{S}$. The expansion proofs for the two new sugoals are $\wedge\mathrm{E}_1(Q_{A\wedge B}, \mathcal{S}_A, (Q, \mathcal{M}))$ and $\wedge\mathrm{E}_2(Q_{A\wedge B}, \mathcal{S}_B, (Q, \mathcal{M}))$ (see Definition 76.5).

Remark 187 Candidates for $\mathcal{S}_A$ and $\mathcal{S}_B$ can be determined in a way similar to determining whether a subtree of Q is essential.

Tactic 188 ($\vee$ *Introduction Left*)

$$\begin{array}{c}\mathcal{S}\\ \vdots\\ A \vee B\end{array} \quad \rightarrow \quad \dfrac{\begin{array}{c}\mathcal{S}\\ \vdots\\ A\end{array}}{A \vee B}\,\vee\mathrm{I}_L$$

provided B is inessential. It is probably also worthwile to check that all of $A \vee B$ is not also inessential, in which case the absurdity rule may be more useful.

The new expansion proof $(R, \mathcal{N})$ is obtained by a shallowing at $Q_{A\vee B}$ followed by erasing Q_B. By Definition 67, the result is an expansion proof for the new subgoal.

Tactic 189 ($\vee$ *Introduction Right*)

$$\begin{array}{c}\mathcal{S}\\ \vdots\\ A \vee B\end{array} \quad \rightarrow \quad \dfrac{\begin{array}{c}\mathcal{S}\\ \vdots\\ B\end{array}}{A \vee B}\,\vee\mathrm{I}_R$$

provided A is inessential. Again, one probably want to make sure that all of $A \vee B$ is not also inessential. The new expansion proof is obtained analogously to the previous tactic.

Remark 190 This leaves a rather large number of cases, where neither A nor B is inessential. However, the impasse may be temporary, which suggests deferring even the check whether A or B is inessential until no further support line tactics can be applied (see Example 169). Other times, the impasse will be critical and we will have to apply indirect proof eventually (see Example 170).

Tactic 191 ($\rightarrow$ *Introduction*)

$$\begin{array}{c}\mathcal{S}\\ \vdots\\ A \rightarrow B\end{array} \quad \rightarrow \quad \dfrac{\begin{array}{c}\mathcal{S}, [\![A]\!]\\ \vdots\\ B\end{array}}{A \rightarrow B}\,\rightarrow\mathrm{I}$$

The expansion proof $(R, \mathcal{N})$ for the new subgoal is simply $\vee\mathrm{E}(Q_{\sim A\vee B}, (Q, \mathcal{M})) = \texttt{shallow}(Q_{\sim A\vee B}, (Q, \mathcal{M}))$. See Definition 76.2.

Tactic 192 ($\neg$ *Introduction*)

$$\begin{array}{c}\mathcal{S}\\ \vdots\\ \neg A\end{array} \quad \rightarrow \quad \begin{array}{c}\mathcal{S}, [\![A]\!]\\ \vdots\\ \dfrac{\bot}{\neg A}\,\neg\mathrm{I}\end{array}$$

The new expansion proof $(R, \mathcal{N})$ is obtained by adjoining $Q^{\mathtt{F}}$ as a new leaf to the top-level disjunction node in $\mathcal{Q}$ and leaving $\mathcal{M}$ unchanged. From the definition of a full clause (Definition 39) it is clear that every full clause in R will be spanned by $\mathcal{M}$. See also Remark 167.

Remark 193 $\neg$ *Introduction* does not by itself make any progress: the expansion proofs for both goals are identical, except for the inessential $Q^{\mathtt{F}}$. As can be seen in the proof of Theorem 243, termination of the tactics is nevertheless assured, since after a $\neg$I some other minimal support line tactic will apply.

Tactic 194 ($\forall$ *Introduction*)

$$\begin{array}{c}\mathcal{S}\\ \vdots\\ \forall x A(x)\end{array} \quad \rightarrow \quad \begin{array}{c}\mathcal{S}\\ \vdots\\ \dfrac{A(a)}{\forall x A(x)}\,\forall\mathrm{I}\end{array}$$

The choice for a is the selected parameter in the expansion proof and therefore legal (as in the case of $\forall$I in $\mathcal{H}$-deductions, see Definition 76.3). $(R, \mathcal{N})$ is obtained by $\forall\mathrm{E}(Q_{\forall x A(x)}, \mathcal{M})$.

Tactic 195 ($\exists$ *Introduction*)

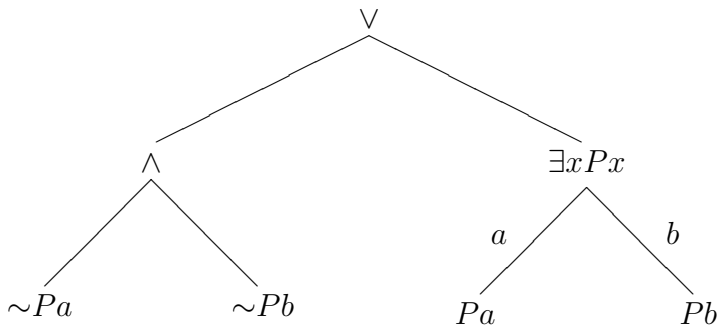
$$\begin{array}{c}\mathcal{S}\\ \vdots\\ \exists x A(x)\end{array} \quad \rightarrow \quad \begin{array}{c}\mathcal{S}\\ \vdots\\ \dfrac{A(t)}{\exists x A(x)}\,\exists\mathrm{I}\end{array}$$

provided t is the unique expansion term of $Q_{\exists x A(x)}$ and t is admissible. The new expansion proof is obtained by $(R, \mathcal{N}) = \exists\mathrm{E}(Q_{\exists x A(x),(Q,\mathcal{M})}) = \mathtt{shallow}(Q_{\exists x A(x)}, (Q, \mathcal{M}))$ (see Definition 76.6).

Remark 196 As in the case of $\vee I$, this leaves a rather large number of cases with different solutions. There may be a unique expansion term, but it is not admissible. Or there could be multiple expansion terms. In either case, the impasse may be temporary or critical. The two examples below should help to understand these impasses and how a temporary impasse could be resolved. See also Example 250 for a case where a multiple expansion term impasse is critical.

Example 197 $Pa \vee Pb \rightarrow \exists x Px$. Here a temporary, multiple expansion term impasse will arise.

Assume the expansion proof



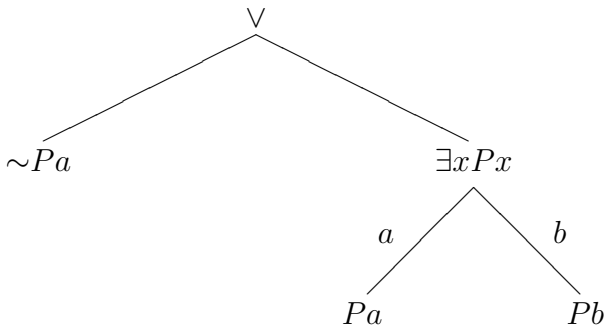
The impasse occurs in the situation

$$\dfrac{\begin{array}{c}[\![Pa \vee Pb]\!] \\ \vdots \\ \exists x Px\end{array}}{Pa \vee Pb \rightarrow \exists x Px} \rightarrow \mathrm{I}$$

because there are two expansion term for $Q_{\exists x Px}$. However, the $\vee$ *Elimination* tactic is applicable, after which we have two sugoals:

$$\dfrac{\dfrac{\begin{array}{ccc}[\![Pa]\!] & [\![Pb]\!] & \\ \vdots & \vdots & \\ \exists x Px & \exists x Px & [\![Pa \vee Pb]\!]\end{array}}{\exists x Px} \vee \mathrm{E}}{Pa \vee Pb \rightarrow \exists x Px} \rightarrow \mathrm{I}$$

The expansion proof for the leftmost subgoal is:



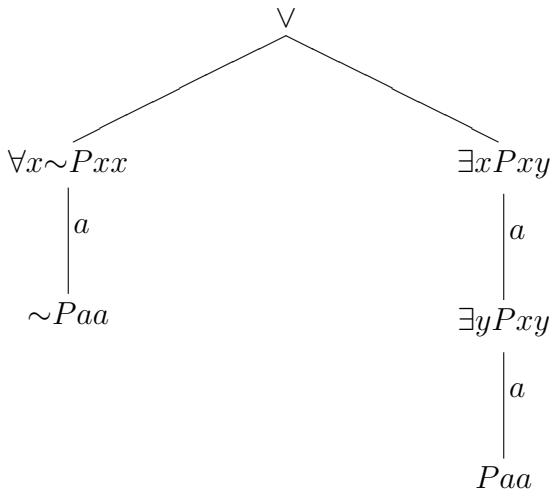
Here the expansion term b is unnecessary (see Definition 93) and can be deleted. After the deletion, the conditions for a the $\exists$ *Introduction* tactic are satisfied.

The corresponding natural deduction:

$$\dfrac{\dfrac{\dfrac{[\![Pa]\!]}{\exists xPx}\exists\mathrm{I} \qquad \dfrac{[\![Pb]\!]}{\exists xPx}\exists\mathrm{I} \qquad Pa \vee Pb}{\exists xPx}\vee\mathrm{E}}{Pa \vee Pb \rightarrow \exists xPx}\rightarrow\mathrm{I}$$

Example 198 $\exists xPxx \rightarrow \exists x\exists yPxy$. Here the temporary impasse arises because the expansion term for x contains a free parameter which must first be selected. The selection is by an application of the $\exists$E rule, which would have been suggested by the *Simple* $\exists$ *Elimination* tactic (see Tactic 220).

The expansion proof:



The corresponding natural deduction:

$$\dfrac{\dfrac{\dfrac{\dfrac{[\![Paa]\!]}{\exists yPay}\exists\mathrm{I}}{\exists x\exists yPxy}\exists\mathrm{I} \qquad [\![\exists xPxx]\!]}{\exists x\exists yPxy}\exists\mathrm{E}}{\exists xPxx \rightarrow \exists x\exists yPxy}\rightarrow\mathrm{I}$$

The analysis of introduction rules which may be applied to a planned line leaves open the following cases:

1. The planned line is $A \vee B$ and both A and B are essential.
2. The planned line is $\exists xA(x)$ and there is more than one expansion term.
3. The planned line is $\exists xA(x)$ and there is a unique expansion term, but it is not admissible.
4. The planned line C is $\mathcal{M}$-mated, but not a leaf.

6.2.4 Minimal Tactics for Support Lines

Again we assume that the node $Q_{\sim A}$ corresponding to the support line A in question is neither a leaf of Q, nor $\mathcal{M}$-mated. Because of the *Mated Support Line* tactic (Tactic 183) this is not as restrictive an assumption as it is for the planned line, which we cannot simply double.

Tactic 199 (*Simple $\wedge$ Elimination Left*)

$$\begin{array}{c}\mathcal{S}, A \wedge B \\ \vdots \\ C\end{array} \quad \rightarrow \quad \begin{array}{c}\mathcal{S}, \quad \dfrac{A \wedge B}{A}\,\wedge\mathrm{E}_L \\ \vdots \\ C\end{array}$$

This tactic applies if Q_B is inessential (and $Q_{\sim[A\wedge B]}$ is essential). The expansion proof transformation is merely an erasing of Q_B after a shallowing. Formally, or $(R, \mathcal{N}) = \texttt{erase}(Q_B, \vee\mathrm{E}(Q_{\sim[A\wedge B]}))$.

Tactic 200 (*Simple $\wedge$ Elimination Right*) Analogous to *Simple $\wedge$ Elimination Left*.

Tactic 201 (*Duplicating Conjunctive Support Line*)

This tactic prepares simple $\wedge$ elimination, in case neither A nor B in a support line $A \wedge B$ are inessential.

$$\begin{array}{c}\mathcal{S}, A \wedge B \\ \vdots \\ C\end{array} \quad \rightarrow \quad \begin{array}{c}\mathcal{S}, A \wedge B, A \wedge B \\ \vdots \\ C\end{array}$$

We let $(R, \mathcal{N}) = \texttt{csplit}(Q_{\sim[A\wedge B]}, (Q, \mathcal{M}))$ (see Definition 236). Now *Simple $\wedge$ Elimination Left* and *Simple $\wedge$ Elimination Right* apply to the left and right copy of $A \wedge B$, respectively.

Tactic 202 ($\vee$ *Elimination*)

$$\begin{array}{c}\mathcal{S}, A \vee B \\ \vdots \\ C\end{array} \quad \rightarrow \quad \dfrac{\begin{array}{ccc}\mathcal{S}_A, [\![A]\!] & \mathcal{S}_B, [\![B]\!] & \\ \vdots & \vdots & \\ C & C & A \vee B\end{array}}{C}\,\vee\mathrm{E}$$

where $\mathcal{S}_A$ and and $\mathcal{S}_B$ are sufficient for the corresponding subgoals and both are sub(multi)sets of $\mathcal{S}$.

The two new expansion proofs are $(R', \mathcal{N}') = \wedge\mathrm{E}_1(Q_{\sim[A\vee B]}, \mathcal{S}_A, (Q, \mathcal{M}))$ and $(R'', \mathcal{N}'') = \wedge\mathrm{E}_2(Q_{\sim[A\vee B]}, \mathcal{S}_B, (Q, \mathcal{M}))$.

Remark 203 The $\vee$E rule often has the effect of resolving temporary impasses. In particular, if the planned line C is a disjunction which was deferred because each disjunct was essential, or if C was existentially quantified and deferred because there was more than one expansion term. In each of these cases it would be worthwile to reevaluate if the impasse has been resolved, that is if some disjunct has become inessential or some expansion term unnecessary.

Tactic 204 (*Simple* $\forall$ *Elimination*)

$$\begin{array}{c} \mathcal{S}, \forall x A(x) \\ \vdots \\ C \end{array} \quad \rightarrow \quad \begin{array}{l} \mathcal{S}, \quad \dfrac{\forall x A(x)}{A(t)}\,\forall\mathrm{E} \\ \qquad \vdots \\ \qquad C \end{array}$$

where t is the only expansion term for $Q_{\sim[\forall x A(x)]}$ and t is admissible. $(R, \mathcal{N}) = \exists\mathrm{E}(Q_{\sim[\forall x A(x)]}, (Q, \mathcal{M}))$.

Tactic 205 (*Duplication Universal Support Line*)

If a support line $\forall x A(x)$ corresponds to $Q_{\sim[\forall x A(x)]}$ with more than one expansion term, double the support line.

$$\begin{array}{c} \mathcal{S}, \forall x A(x) \\ \vdots \\ C \end{array} \quad \rightarrow \quad \begin{array}{c} \mathcal{S}, \forall x A(x), \forall x A(x) \\ \vdots \\ C \end{array}$$

Let T be the set of expansion term of $Q_{\sim[\forall x A(x)]}$. For any partition T_0, T_1 of T we can obtain a new expansion proof $(R, \mathcal{N}) = \texttt{split}(Q_{\sim[\forall x A(x)]}, T_0, T_1, (Q, \mathcal{M}))$.

Remark 206 The most common applications of the previous tactic let T_0 be t for some admissible expansion term in T, and $T_1 = T - T_0$. As one can see from the argument in the proof of Theorem 243, the tactics remain complete with this additional restriction.

Tactic 207 ($\exists$ *Elimination*)

$$\begin{array}{c} \mathcal{S}, \exists x A(x) \\ \vdots \\ C \end{array} \quad \rightarrow \quad \begin{array}{l} \mathcal{S}, [\![A(a)]\!] \\ \qquad \vdots \\ \dfrac{C \qquad \exists x A(x)}{C}\,\exists\mathrm{E} \end{array}$$

where a is the parameter selected for $Q_{\sim[\exists x A(x)]}$. $(R, \mathcal{N}) = \forall\mathrm{E}Q_{\sim[\exists x A(x)]}, (Q, \mathcal{M})$.

Remark 208 If the planned line C is existentially quantified and was defered because its unique expansion term was not admissible, reevaluate whether it is admissible now.

Tactic 209 ($\rightarrow$ *Elimination*)

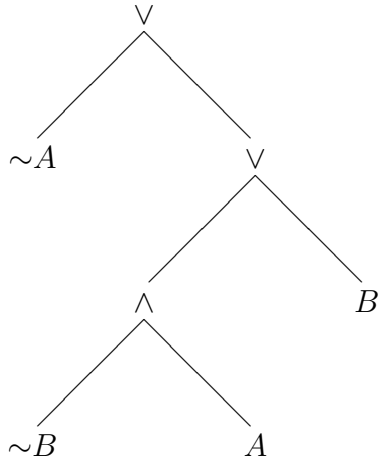
$$\begin{array}{c} \mathcal{S},\, A \rightarrow B \\ \vdots \\ C \end{array} \qquad \rightarrow \qquad \begin{array}{c} \mathcal{S}_A \\ \vdots \\ A \qquad\qquad A \rightarrow B \\ \hline \mathcal{S}_B, \qquad B \\ \vdots \\ C \end{array} \rightarrow\mathrm{E}$$

provided there are subsets $\mathcal{S}_A$ and $\mathcal{S}_B$ of $\mathcal{S}$ such that $\mathcal{S}_B, B$ are sufficient for C and $\mathcal{S}_A$ is sufficient for A. The two new expansion proofs are obtained by $(R', \mathcal{N}') = \wedge\mathrm{E}_1(Q_{\sim[A \supset B]}, \mathcal{S}_A, (Q, \mathcal{M}))$ and $(R'', \mathcal{N}'') = \wedge\mathrm{E}_2(Q_{\sim[A \supset B]}, \mathcal{S}_{\sim B}, (Q, \mathcal{M}))$.

Remark 210 This is the first minimal support line tactic which can lead to an impasse, namely when there is no $\mathcal{S}_A$ which is sufficient for A. This does not contradict Lemma 72, because here $\mathcal{S}_A$ cannot contain C, the planned line.

Example 211 $A \rightarrow [[\neg B \rightarrow \neg A] \rightarrow B]$.

An expansion proof is



The construction of the natural deduction is straightforward until we reach

$$\begin{array}{c} A,\, \neg B \rightarrow \neg A \\ \vdots \\ B \end{array}$$

At this point the tactic is not applicable, since of the two possible subgoals

$$\begin{array}{c} A,\neg A \\ \vdots \\ B \end{array} \qquad\qquad \begin{array}{c} A \\ \vdots \\ \neg B \end{array}$$

only the first could be achieved.

However, we can wait until after the rule of indirect proof has been applied:

$$\begin{array}{c} A,\ \neg B \to \neg A,\ [\![\neg B]\!] \\ \vdots \\ \dfrac{\bot}{B}\,\bot_C \end{array}$$

in which case we are left with two trivial subgoals:

$$\begin{array}{c} [\![\neg B]\!] \\ \vdots \\ \dfrac{\neg B \qquad \neg B \to \neg A}{A,\ \neg A}\,\to\!\mathrm{E} \\ \vdots \\ \bot \end{array}$$

Negation Elimination, like $\bot_C$, plays a special role in the search for a deduction. ¬E eliminates a negation, and makes progress in that sense. However, the negation may have been created by a use of $\bot_C$, which could introduce a non-terminating sequence of deductions. Thus care must be taken when attempting to use ¬E, and we will only apply it if the current planned line is $\bot$ and if one of the minimal planned line tactics applies immediately to the new planned line. Thus conceptually there are six ¬-elimination tactics, one for each possible main connective or quantifier and one atoms.

Tactic 212 (¬ *Atom Elimination*) If a support line is of the form $\neg A$ for an atom A, and the planned line is $\bot$, and A is another support line, infer $\bot$ from A and $\neg A$ by ¬E. The remaining subgoal is to infer the new planned line A from the support line A, which can be solved directly with tactic *Closing a Gap* (Tactic 173).

Remark 213 This tactic may be restricted to cases where $Q_{\sim A}$ and Q_A are $\mathcal{M}$-mated. This saves time, since no formulas need to be compared. The completeness of the tactics is not affected.

Tactic 214 ($\neg\neg$ *Elimination*) If a support line is of the form $\neg\neg A$, and the planned line is $\bot$, apply $\neg$E.

$$\begin{array}{c} \mathcal{S}, \neg\neg A \\ \vdots \\ \bot \end{array} \quad \rightarrow \quad \begin{array}{c} \mathcal{S} \\ \vdots \\ \dfrac{\neg A \qquad \neg\neg A}{\bot}\,\neg\text{E} \end{array}$$

The expansion proof only changes trivially, since $Q^{\mathtt{F}}$ is erased. $\neg$E will be followed by an application of the $\neg$ *Introduction* tactic.

Tactic 215 ($\neg\wedge$ *Elimination*) If a support line is of the form $\neg[A \wedge B]$, and the planned line is $\bot$, apply $\neg$E.

$$\begin{array}{c} \mathcal{S}, \neg[A \wedge B] \\ \vdots \\ \bot \end{array} \quad \rightarrow \quad \begin{array}{c} \mathcal{S} \\ \vdots \\ \dfrac{A \wedge B \qquad \neg[A \wedge B]}{\bot}\,\neg\text{E} \end{array}$$

The expansion proof only changes trivially. $\neg$E will be followed by an application of the $\wedge$ *Introduction* tactic.

Tactic 216 ($\neg\rightarrow$ *Elimination*) If a support line is of the form $\neg[A \rightarrow B]$, and the planned line is $\bot$, apply $\neg$E.

$$\begin{array}{c} \mathcal{S}, \neg[A \rightarrow B] \\ \vdots \\ \bot \end{array} \quad \rightarrow \quad \begin{array}{c} \mathcal{S} \\ \vdots \\ \dfrac{A \rightarrow B \qquad \neg[A \rightarrow B]}{\bot}\,\neg\text{E} \end{array}$$

The expansion proof only changes trivially. $\neg$E enables an application of the $\rightarrow$ *Introduction* tactic.

Tactic 217 ($\neg\forall$ *Elimination*) If a support line is of the form $\neg\forall x A(x)$, and the planned line is $\bot$, apply $\neg$E.

$$\begin{array}{c} \mathcal{S}, \neg\forall x A(x) \\ \vdots \\ \bot \end{array} \quad \rightarrow \quad \begin{array}{c} \mathcal{S} \\ \vdots \\ \dfrac{\forall x A(x) \qquad \neg\forall x A(x)}{\bot}\,\neg\text{E} \end{array}$$

The expansion proof only changes trivially. $\neg$E enables an application of the $\forall$ *Introduction* tactic.

Tactic 218 (*Simple* $\neg\vee$ *Elimination*) If a support line is of the form $\neg[A \vee B]$, and the planned line is $\bot$, and either A or B is inessential (but not both simultaneously, in which case the *Inessential Support Line* tactic applies), apply $\neg$E.

$$\begin{array}{c} \mathcal{S}, \neg[A \vee B] \\ \vdots \\ \bot \end{array} \quad \rightarrow \quad \begin{array}{c} \mathcal{S} \\ \vdots \\ \dfrac{A \vee B \qquad \neg[A \vee B]}{\bot}\,\neg\mathrm{E} \end{array}$$

The expansion proof only changes trivially. $\neg$E will be followed by an application of one the *Simple* $\vee$ *Introduction* tactics.

Tactic 219 (*Duplicating* $\neg\vee$ *Elimination*) If a support line is of the form $\neg[A \vee B]$, and the planned line is $\bot$, and neither A nor B are inessential, apply $\neg$E.

$$\begin{array}{c} \mathcal{S}, \neg[A \vee B] \\ \vdots \\ \bot \end{array} \quad \rightarrow \quad \begin{array}{c} \mathcal{S}, \neg[A \vee B] \\ \vdots \\ \dfrac{A \vee B \qquad \neg[A \vee B]}{\bot}\,\neg\mathrm{E} \end{array}$$

For the first time a $\neg$ *Elimination* tactic must change the expansion proof in a non-trivial way. Since $\neg[A \vee B]$ still appears in the support, the expansion proof must simplify. To obtain R from Q we replace Q_A by a new expansion tree $R^0_{A\vee B}$ with successors Q_A and l_B, and we replace Q_B by a new expansion tree $R^1_{A\vee B}$ with successors l_A and Q_B. This operation is analogous to the `split` operation, except that it occurs on disjunction rather than expansion nodes. The mating $\mathcal{M}$ is still clause-spanning on R, since every full clause through R is an extension of a full clause in $\mathrm{fc}(Q)\backslash Q_{A\vee B}$ by l_A, l_B, and a few more interior nodes. But we assumed that $Q_{A\vee B}$ was not $\mathcal{M}$-mated, so $\mathcal{M}$ spans every full clause in R. In short $(R, \mathcal{N}) = \texttt{csplit}(Q_{A\vee B}, (Q, \mathcal{M}))$ (see Definition 236).

See Lemma 239 to see that this is "progress".

Tactic 220 (*Simple* $\neg\exists$ *Elimination*) If a support line is of the form $\neg\exists x A(x)$, the planned line is $\bot$, and there is exactly one expansion term t for $Q_{\exists x A(x)}$, and t is admissible, then apply $\neg$E.

$$\begin{array}{c} \mathcal{S}, \neg\exists x A(x) \\ \vdots \\ \bot \end{array} \quad \rightarrow \quad \begin{array}{c} \mathcal{S} \\ \vdots \\ \dfrac{\exists x A(x) \qquad \neg\exists x A(x)}{\bot}\,\neg\mathrm{E} \end{array}$$

The expansion proof only changes trivially. $\neg$E enables the $\exists$ *Introduction* tactic.

Tactic 221 (*Duplicating* $\neg\exists$ *Elimination*) If a support line is of the form $\neg\exists x A(x)$, the planned line is $\bot$, and there is more than one expansion term for $Q_{\exists x A(x)}$, one of which, say t, is admissible, then apply $\neg$E.

$$\begin{array}{c} \mathcal{S},\, \neg\exists x A(x) \\ \vdots \\ \bot \end{array} \quad \rightarrow \quad \begin{array}{c} \mathcal{S},\, \neg\exists x A(x) \\ \vdots \\ \dfrac{\exists x A(x) \qquad \neg\exists x A(x)}{\bot}\,\neg\mathrm{E} \end{array}$$

This case is similar to the *Duplicating* $\neg\vee$ *Elimination* tactic. We let $(R, \mathcal{N}) = \texttt{split}(Q_{\exists x A(x)}, \{t\}, T - \{t\}, (Q, \mathcal{M}))$, where T is the set of expansion terms for $Q_{\exists x A(x)}$. $R^{t}_{\exists x A(x)}$, the node with the expansion term t, corresponds to the new planned line, $R^{T-\{t\}}_{\exists x A(x)}$ corresponds to the support line $\neg\exists x A(x)$.

The condition that t was admissible means that the $\exists$ *Introduction* tactic applies to the new planned line.

Example 222 $A \vee \neg A$.

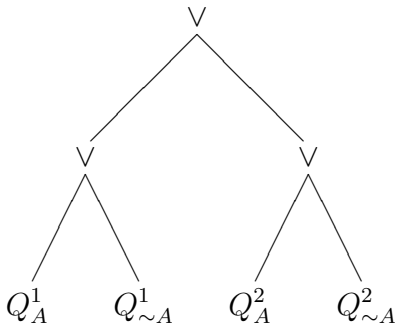
As noted before, we are forced to start with an indirect proof:

$$\begin{array}{c} [\![\neg[A \vee \neg A]]\!] \\ \vdots \\ \dfrac{\bot}{A \vee \neg A}\,\bot_C \end{array}$$

Applying the $\neg\vee$ *Elimination* tactic:

$$\begin{array}{c} [\![\neg[A \vee \neg A]]\!] \\ \vdots \\ \dfrac{\dfrac{A \vee \neg A \qquad [\![\neg[A \vee \neg A]]\!]}{\bot}\,\neg\mathrm{E}}{A \vee \neg A}\,\bot_C \end{array}$$

We have made progress compared to the original goal, since we have one additional support line, but more importantly, A, the left disjunct, has become inessential. This can be seen by examining the expansion proof (the superscript is only to disambiguate):

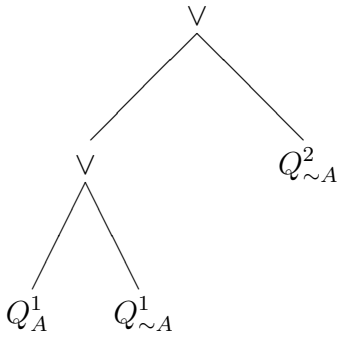


with the mating $\{(Q^1_A, Q^2_{\sim A})\}$.

After the $\vee \mathrm{I}_R$ and $\neg$I we arrive at

$$\dfrac{\dfrac{\dfrac{[\![\neg[A \vee \neg A]]\!], [\![A]]\!]}{\vdots}}{\dfrac{\dfrac{\bot}{\neg A}\,\neg\mathrm{I}}{A \vee \neg A}\,\vee \mathrm{I}_R} \quad [\![\neg[A \vee \neg A]]\!]}{\dfrac{\bot}{A \vee \neg A}\,\bot_C}\,\neg\mathrm{E}$$

with associated expansion proof



and mating $\{(Q^1_A, Q^2_{\sim A})\}$.

We can again apply the $\neg$E, but this time we do not need to copy the support line $\neg[A \vee \neg A]$. The tactic that applies is *Simple* $\neg\vee$ *Elimination*, since $Q^1_{\sim A}$ is inessential (which can be seen easily since it is not $\mathcal{M}$-mated).

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[\![A]\!]}{\vdots}}{A \vee \neg A} \qquad \dfrac{}{[\![\neg[A \vee \neg A]]\!]}\neg\mathrm{E}}{\dfrac{\bot}{\neg A}\neg\mathrm{I}}}{A \vee \neg A}\vee\mathrm{I}_R \qquad \dfrac{}{[\![\neg[A \vee \neg A]]\!]}\neg\mathrm{E}}{\bot}}{A \vee \neg A}\bot_C$$

Remark 223 The resulting deductions are often inelegant, because using ¬E on the negated assumption after an indirect proof amounts to saying that the contradiction we are trying to derive will be between the original planned line and its negation.

6.2.5 Non-Minimal Tactics

These tactics are not minimal, that is, the resulting deduction will be in the intuitionistic or classical logic, but not in its minimal fragment.

Tactic 224 (*Inessential Planned Line*) If Q_A corresponding to a planned line A is inessential, and A is different from $\bot$, infer it by $\bot_I$.

$$\begin{array}{c}\mathcal{S}\\ \vdots\\ A\end{array} \qquad \rightarrow \qquad \dfrac{\begin{array}{c}\mathcal{S}\\ \vdots\\ \bot\end{array}}{A}\bot_I$$

Example 225 This is not always desirable. For example $A \rightarrow [\neg A \rightarrow [B \rightarrow B]]$.

$$\dfrac{\dfrac{\dfrac{[\![A]\!], [\![\neg A]\!]}{\vdots}}{B \rightarrow B}}{\neg A \rightarrow [B \rightarrow B]}\rightarrow\mathrm{I}}{A \rightarrow [\neg A \rightarrow [B \rightarrow B]]}\rightarrow\mathrm{I}$$

could be completed to either of

$$\dfrac{\dfrac{\dfrac{\dfrac{[\![A]\!] \qquad [\![\neg A]\!]}{\bot}\,\neg\mathrm{E}}{B \to B}\,\bot_I}{\neg A \to [B \to B]}\,{\to}\mathrm{I}}{A \to [\neg A \to [B \to B]]}\,{\to}\mathrm{I} \qquad\qquad \dfrac{\dfrac{\dfrac{[\![B]\!]}{B \to B}\,{\to}\mathrm{I}}{\neg A \to [B \to B]}\,{\to}\mathrm{I}}{A \to [\neg A \to [B \to B]]}\,{\to}\mathrm{I}$$

The last remaining rule which is not yet suggested by any other tactic is also the most troublesome. The problem is that a use of $\bot_C$ by itself does not make any progress and infinite sequences of subgoals can arise if one is not careful. An example are alternations of $\bot_C$ and $\neg$E.

All of the tactics below preserve the *weak* subformula property: the deductions produced are normal. We have already pointed out that this is not always desirable, and given some examples (see Examples 170, 172, and 211). Therefore some of the tactics below have to be taken with a grain of salt. They only provide a complete translation procedure, but the tactics one really would want to use in practice are introduced in Section 6.5.

Luckily, the problems associated with the use of $\bot_C$ can be handled by using $\neg$E with care. This was the main reason for presenting so many $\neg$E tactics: we had to avoid infinite non-progressing deductions by placing restrictions on the use of $\neg$E. Perhaps the same goal can be achieved by placing restrictions on applications of $\bot_C$ alone, but it would seem unlikely.

Thus progress in the deduction is ensured by the fact that some $\neg$E tactic will apply after $\bot_C$ has been applied, in effect (in perhaps a few more steps) simplifying the expansion proof (with respect to the $\sqsubset'$ ordering).

Remark 226 Because of the way we formulated the other tactics, indirect proof will not be applied if the planned line is $\bot$.

We distinguish different tactics for different types of impasses, even though the deduction transformation for all inferences below is of the form

$$\begin{array}{c}\mathcal{S}\\ \vdots\\ C\end{array} \qquad \to \qquad \begin{array}{c}\mathcal{S}, [\![\neg C]\!]\\ \vdots\\ \dfrac{\bot}{C}\,\bot_C\end{array}$$

Tactic 227 (*Mated Planned Line*) If the planned line C corresponds to Q_C, which is a mated non-leaf, and no support line tactic applies, apply $\bot_C$.

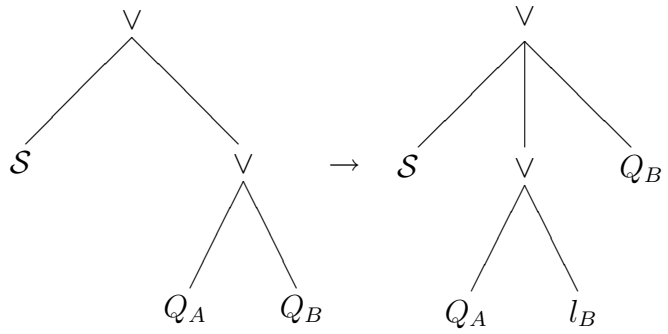
Remark 228 In this case we can follow the indirect proof by the *Mated Support Line* tactic (see Tactic 183). $\bot_C$ only changes the expansion proof by adjoining $Q^{\mathtt{F}}$, so $(R, \mathcal{N}) \sqsubset' (Q, \mathcal{M})$ (since $Q^{\mathtt{F}}$ is inessential).

Tactic 229 (*Classical Disjunction*) If the planned line is $A \vee B$ and neither A nor B is inessential, and no support line tactic applies, infer $A \vee B$ by $\perp_C$.

Example 230 We illustrate how the deduction progresses if the $\perp_C$ is immediately followed by an application of the *Duplicating* $\neg\vee$ tactic. Because of the symmetry of the problem, there are actually two choices: the new planned line could be A or B. The deduction is transformed as follows:

$$\begin{array}{c} \mathcal{S} \\ \vdots \\ A \vee B \end{array} \quad \rightarrow \quad \dfrac{\dfrac{\begin{array}{c}\mathcal{S},\, [\![\neg[A \vee B]]\!] \\ \vdots \\ B\end{array}}{A \vee B}\vee \mathrm{I}_R \qquad \neg[A \vee B]}{\dfrac{\perp}{A \vee B}\perp_C}\neg \mathrm{E}$$

The copy of Q_B^1 which corresponds to the copy of B still in the planned line is no longer mated, nor is any of its subnodes. The node Q_B^2 corresponding to the new planned line B actually looks exactly like Q_B before the transformation steps, and Q_B^1 is an unmated leaf l_B. Graphically,



The case where A is the new planned line is dual.

Remark 231 The decision whether A or B should be the new planned line depends on the structure of A and B. A short look-ahead can decide if a planned line tactic would directly apply to A or B in the new situation. If yes, that disjunct should be preferred.

Tactic 232 (*Multiple Expansion Terms*) If the planned line is $\exists x A(x)$ with two or more expansion terms, one of which is admissible, and no support line tactic applies, infer $\exists x A(x)$ by $\perp_C$.

Example 233 If we were to apply the *Duplicating* $\neg\exists$ *Elimination* tactic immediately after the $\perp_C$, we would get the following deduction transformation.

$$\begin{array}{c}\mathcal{S}\\ \vdots\\ \exists x A(x)\end{array} \quad \rightarrow \quad \dfrac{\dfrac{\begin{array}{c}\mathcal{S},[\![\neg[\exists x A(x)]]\!]\\ \vdots\\ A(t)\end{array}}{\exists x A(x)}\exists\mathrm{I} \qquad [\![\neg[\exists x A(x)]]\!]}{\dfrac{\bot}{\exists x A(x)}\bot_C}\neg\mathrm{E}$$

The expansion proof transformation is simply a split: $(R, \mathcal{N}) = \texttt{split}(Q_{\exists x A(x)}, \{t\}, T - \{t\}, (Q, \mathcal{M}))$, where T is the set of expansion terms of $Q_{\exists x A(x)}$ and t is an admissible term in T.

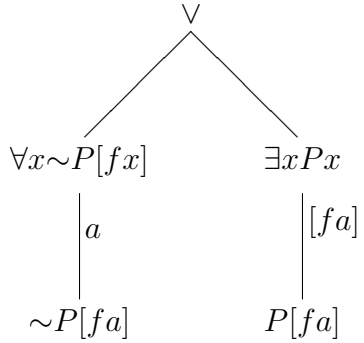
Tactic 234 (*Inadmissible Expansion Terms*) If the planned line is $\exists x A(x)$ with at least one expansion term, such that all of them are inadmissible, infer $\exists x A(x)$ by $\bot_C$.

Example 235 Illustrating the last type of impasse: $\neg\forall x \neg P[fx] \rightarrow \exists x P x$.

We get the subgoal

$$\begin{array}{c}\neg\forall x \neg P[fx]\\ \vdots\\ \exists x P x\end{array}$$

with expansion proof



The sole expansion term for $Q_{\exists x P x}$ is not admissible, since a has not yet been selected. We cannot use a $\neg$ *Elimination* tactic since we are not trying to prove $\bot$, and are forced to an indirect proof.

$$\dfrac{\begin{array}{c}[\![\neg\exists x P x]\!], \neg\forall x \neg P[fx]\\ \vdots\\ \bot\end{array}}{\exists x P x}\bot_C$$

Now the tactic can be applied and we obtain:

$$\dfrac{\dfrac{\neg\forall x\neg P[fx] \qquad \forall x\neg P[fx]}{\bot}\,\neg\mathrm{E}}{\exists xPx}\,\bot_C \qquad \begin{array}{c}[\![\neg\exists xPx]\!]\\ \vdots\end{array}$$

Again we have a situation where $\neg$ elimination cannot be applied. But this is not an impasse, since we can work on the planned line.

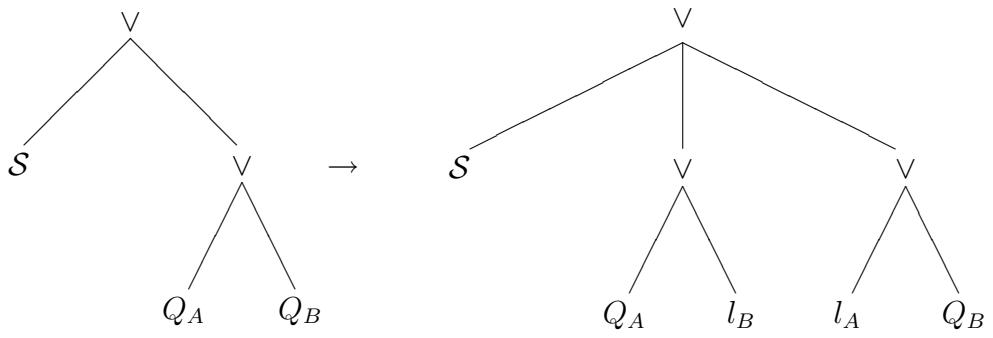
The rest is routine, yielding

$$\dfrac{\dfrac{\neg\forall x\neg P[fx] \qquad \dfrac{\dfrac{[\![\neg\exists xPx]\!] \qquad \dfrac{[\![P[fa]]\!]}{\exists xPx}\,\exists\mathrm{I}}{\bot}\,\neg\mathrm{E}}{\neg P[fa]}\,\neg\mathrm{I}}{\forall x\neg P[fx]}\,\forall\mathrm{I}}{\bot}\,\neg\mathrm{E}}{\exists xPx}\,\bot_C$$

6.3 Correctness and Completeness of the Tactics

In order to prove correctness and completeness of the tactics, we need a slight refinement of the $\sqsubset$ relation which was defined earlier.

Definition 236 Let $(Q, \mathcal{M})$ be an expansion proof with disjunction node $Q_{A\vee B}$ with immediate subtree Q_A and Q_B. Moreover, assume that neither Q_A or Q_B is inessential. We obtain R by replacing $Q_{A\vee B}$ by two new expansion trees with the same shallow formula. In the first one we replace Q_B by a new leaf l_B with the same shallow formula. In the second copy we replace Q_A by a new leaf l_A with the same shallow formula. Then define $\texttt{csplit}(Q_{A\vee B}, (Q, \mathcal{M})) = (R, \mathcal{M})$. We say that $(R, \mathcal{M})$ is the result of splitting the disjunction node $Q_{A\vee B}$ in $(Q, \mathcal{M})$. Graphically,



Lemma 237 Given an expansion proof $(Q, \mathcal{M})$ with single and accessible node $Q_{A \vee B}$. Then $\texttt{csplit}(Q_{A \vee B}, (Q, \mathcal{M}))$ is an expansion proof.

Proof: Every full clause c in R has a preimage in Q which one can obtain by erasing all occurrences of the new nodes l_A, l_B, and a few intermediate nodes. This preimage is spanned by $\mathcal{M}$ and hence c is spanned.

□

Definition 238 Let $(Q, \mathcal{M}) \sqsubset_O (R, \mathcal{N})$ if $(Q, \mathcal{M})$ is the result of splitting expansion nodes in $(R, \mathcal{N})$, and each split node is such that neither disjunct is inessential.

Lemma 239 $\sqsubset_O$ is a well-ordering.

Proof: The number of disjunction nodes such that both disjuncts are essential decreases. There can only be a finite number of such nodes, and hence $\sqsubset_O$ is a well-ordering.

□

Definition 240 We define $\sqsubset_E$ such that $(Q, \mathcal{M}) \sqsubset_E (R, \mathcal{N})$ whenever $(Q, \mathcal{M})$ is the result of some shallowings and deletions in $(R, \mathcal{N})$ such that they do not create new disjunctions where both disjuncts are essential. The $\sqsubset'$ is defined as the lexicographic order obtained from $\sqsubset_S$, $\sqsubset_O$, $\sqsubset_C$, and $\sqsubset_E$ in this order $(\sqsubset_S, \sqsubset_O, \sqsubset_C, \sqsubset_E)$.

Lemma 241 $\sqsubset'$ is a well-ordering.

Proof: See the proof of Lemma 66. We only have to add a check that applying `csplit` does not increase the expansion proof with respect ot $\sqsubset_S$, which is clear since no new expansion nodes are created, nor are any existing ones changed. The restriction which distinguishes $\sqsubset_E$ from $\sqsubset_D$ ensures that $\sqsubset_O$ is not increased through shallowings or deletions.

□

Theorem 242 Given a goal with planned line B, support lines $A_1, \ldots, A_n$ and expansion proof $(Q, \mathcal{M})$ with shallow formula $\sim\overline{A_1} \vee \ldots \vee \overline{A_n} \vee \overline{B}$. Then for any tactic given above and every resulting subgoal with planned line B', support lines $A'_1, \ldots, A'_m$, and plan $(R, \mathcal{N})$, $(R, \mathcal{N})$ is an expansion proof with shallow formula $\sim\overline{A_1} \vee \ldots \vee \overline{A_n} \vee \overline{B}$.

Proof: The correctness of the tactics follows easily from the work done in Sections 3.2 and 3.3, in particular Lemmas 53, 56, 58, and 62.

□

Theorem 243 Given a goal with planned line B, support lines $\mathcal{S}$ and expansion proof $(Q, \mathcal{M})$. Then at least one of the tactics will apply.

Proof: This follows the same pattern as the proof for Theorem 80. One can just enumerate all possibilities for planned and support lines, and it becomes obvious that the only critical part of the proof is to show that there is no support line impasse when all support lines are of the form $\forall x A(x)$ or $\neg\exists x A(x)$ and the planned line is $\bot$ (if the planned line is not $\bot$, one could always apply one of the indirect proof tactics). This follows immediately from Lemma 75, since it says that at least one of the expansion terms for the top-level nodes of Q has to be admissible, and for the corresponding support line either the $\forall$ *Elimination* tactic (possibly preceded by a *Duplicating Universal Support Line*), *Simple* $\neg\exists$ *Elimination*, or *Duplicating* $\neg\exists$ *Elimination* will apply.

□

Theorem 244 The given tactics create no infinite sequences of subgoals.

Proof: This is by far the most difficult to prove. We have to make use of of $\sqsubset'$ which is the relevant well-ordering, and also take into account sequences of applications which eliminate successive negations.

Let us use the name *negation tactics* for $\neg$ *Introduction* (Tactic 192) and $\neg\neg$ *Elimation* (Tactic 214). Also, the tactics suggesting indirect proof will be called *indirect proof tactics*.

First note that the expansion proof for the subgoal created by an application of a negation or indirect proof tactic is identical to the expansion proof before the tactic was applied.

The following tactics, which we shall call *critical tactics*, also have this property: $\neg\wedge$ *Elimination* (Tactic 215), $\neg\rightarrow$ *Elimination* (Tactic 216), $\neg\forall$ *Elimination* (Tactic 217), *Simple* $\neg\vee$ *Elimination* (Tactic 218), *Simple* $\neg\exists$ *Elimination* (Tactic 220). All other tactics create subgoals with simpler (with respect to $\sqsubset'$) associated expansion proofs. Let us call those tactics *expansion simplifying tactics*.

We say a tactic is *enabled* when its applicability condition is satisfied, *disabled* otherwise. Every critical tactic directly enables an *expansion simplifying tactic*. For example, applying the $\neg\rightarrow$ *Elimination* tactic always enables the $\rightarrow$ *Introduction* tactic.

Moreover, applying a critical tactic disables any further critical tactics or indirect proof tactics. This is because the tactics which eliminate a negation in a support line apply only when the planned line is $\bot$. This is also the reason why we distinguished between *Simple* and *Duplicating* tactics in some cases: we have to make sure that no indirect proof tactic is enabled by applying a critical tactic. Note that the duplicating tactics are not critical, since they are expansion simplifying tactics.

Applying an indirect proof tactic will either directly enable an expansion simplifying tactic or a critical tactic and of course disable any indirect proof tactics.

The remaining cases are the negation tactics, which can indeed be applied alternatingly. However, the number of consecutive applications of the negations tactics is bounded by the number of outermost double negations occurring in the support lines and the planned line and infinite sequences of negation tactics applications cannot occur.

□

6.4 Extension to Higher-Order Logic

When we extend the first-order treatment above, we have to add the rule of extensionality. The reason is that in using equivalence classes of formulas in expansion proofs, we use a limited amount of extensionality when equivalent, but not equal, formulas appear in the arguments to predicates or higher-order functions.

Definition 245 (*Language $\mathcal{LN}$*) We replace the clauses 10 and 11 in Definition 162 by allowing the quantifier to range over variables of arbitrary type, and add λ-abstraction as a new construct.

10. $\forall x_\alpha A_o$ for a variable x_α and formula A_o.
11. $\exists x_\alpha A_o$ for a variable x_α and formula A_o.
12. $[\lambda x_\beta A_\alpha]$ is a formula of type $\alpha\beta$ for a variable x_β and formula A_α.

In order to avoid explicit uses of λ-conversion rules, we require substitution to return the result in λ-normal form, and also that the original theorem be in λ-normal form. A less attractive alternative would be to explicitly introduce a rule of λ-conversion into the natural deduction calculus.

Definition 246 (*Natural Deduction with Extensionality $\mathcal{N}^e$*) $\mathcal{N}^e$ is obtained from $\mathcal{N}$ by adding the following rule of extensionality (see Definition 151 for the definition of E)

$$\frac{\mathbf{P}f^1 \dots f^n \qquad E(f^1, g^1) \quad \dots \quad E(f^n, g^n)}{\mathbf{P}g^1 \dots g^n}\, Ext$$

The methods given in Chapter 5 carry over directly to the case of natural deductions and we will not repeat them here. If one decides to continue to use expansion trees rather than extensional expansion trees, extensionality will only be necessary in a case where the *Closing a Gap* tactic (Tactic 173) fails, because the two atomic formulas A and B are such that $\overline{A} \underset{\lambda^\sim}{=} \overline{B}$, but $A \neq B$. In this case we only need the following extensionality tactic.

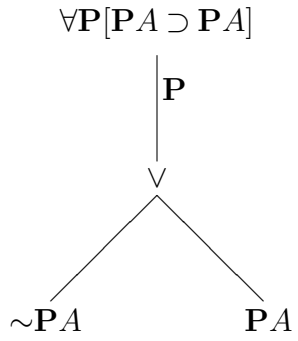
Tactic 247 (*Atomic Extensionality*) If an atomic planned line B and atomic support line A are such that $\overline{A} = \overline{B}$, but A and B are different as $\mathcal{LN}$ formulas, infer B by extensionality.

$$\frac{\mathbf{P}f^1 \dots f^n \qquad E(f^1, g^1) \quad \dots \quad E(f^n, g^n)}{\mathbf{P}g^1 \dots g^n}\, Ext$$

where $\mathbf{P}f^1 \dots f^n \underset{\lambda}{\longrightarrow} A$ and $\mathbf{P}g^1 \dots g^n \underset{\lambda}{\longrightarrow} B$ and f^i distinct from g^i for $1 \leq i \leq n$.

Remark 248 One should also check to make sure that the subdeductions are not redundant, that is the $E(f^i, g^i)$ are pairwise syntactically distinct.

Example 249 $A = \neg\neg A$. Expanding the definition, we get the following expansion proof with shallow formula equivalent to $\forall \mathbf{P}[\mathbf{P}A \supset \mathbf{P}A]$



In the natural deduction we have to make use of the extensionality rule, since it is simply not a theorem in $\mathcal{N}$ without extensionality. Here we could have made the following deduction:

$$\dfrac{\dfrac{\dfrac{\dfrac{[\![\mathbf{P}A]\!] \qquad \vdots \quad [A \to \neg\neg A] \wedge [\neg\neg A \to A]}{\mathbf{P}[\neg\neg A]}\, E}{\mathbf{P}A \to \mathbf{P}[\neg\neg A]}\, \to\!\mathrm{I}}{\forall \mathbf{P}[\mathbf{P}A \supset \mathbf{P}[\neg\neg A]]}\, \forall\mathrm{I}$$

Theorems 242, 243, and 244 remain true for $\mathcal{N}^e$ if the *Atomic Extensionality* tactic is added. This is easy to see since none of our proofs made use of the fact that the underlying language was first-order. We took care to express the termination ordering purely in terms of expansion proofs rather than complexity of formulas. In only remains to add that an application of the atomic extensionality tactic reduces the number of positions inside atoms at which two mated atoms are equal merely up to $\sim$-conversion, but not directly equal. Therefore the number of applications of the atomic extensionality tactic must also be finite.

6.5 Symmetric Simplification

After the rule of indirect proof has been applied, we have $\perp$ as a planned line. This does not give us any formula structure to work on and it is desirable to apply some elimination rule to establish one or more new subgoals with some more complicated formula as planned line.

Often the resulting deductions do not seem very natural, and do not resemble an informal argument one would be inclined to give (see Examples 170 and 250).

One source of this problem is that fact that the contradictory formulas which one uses in informal arguments are not necessarily weak subformulas, and therefore cannot be found by the tactics in Section 6.2.

Three closely related ways of making such proofs more intuitive are explored.

1. Try to find simpler formulas B and $\neg B$ as planned lines, effecting the following proof transformation:

$$\begin{array}{c}\mathcal{S}\\ \vdots\\ A\end{array} \quad \rightarrow \quad \dfrac{\begin{array}{cc}\mathcal{S}_B,\,[\![\neg A]\!] & \mathcal{S}_{\neg B},\,[\![\neg A]\!]\\ \vdots & \vdots\\ B & \neg B\end{array}}{\dfrac{\bot}{A}\,\bot_C}\,\neg\mathrm{E}$$

2. Assuming an additional derived rule of inference (the rule of excluded middle) and using it to avoid indirect proof. Here the proof transformation is

$$\begin{array}{c}\mathcal{S}\\ \vdots\\ A\end{array} \quad \rightarrow \quad \dfrac{\begin{array}{ccc}\mathcal{S}_B,[\![B]\!] & \mathcal{S}_{\neg B},[\![\neg B]\!] & \\ \vdots & \vdots & \\ A & A & B\vee\neg B\end{array}}{A}\,\vee\mathrm{E}$$

3. Item 2 is improved by allowing any two formulas B and C such that $\overline{B} \underset{\lambda^{\sim}}{=} \overline{C}$.

$$\begin{array}{c}\mathcal{S}\\ \vdots\\ A\end{array} \quad \rightarrow \quad \dfrac{\begin{array}{ccc}\mathcal{S}_B,[\![B]\!] & \mathcal{S}_C,[\![C]\!] & \\ \vdots & \vdots & \vdots\\ A & A & B\vee C\end{array}}{A}\,\vee\mathrm{E}$$

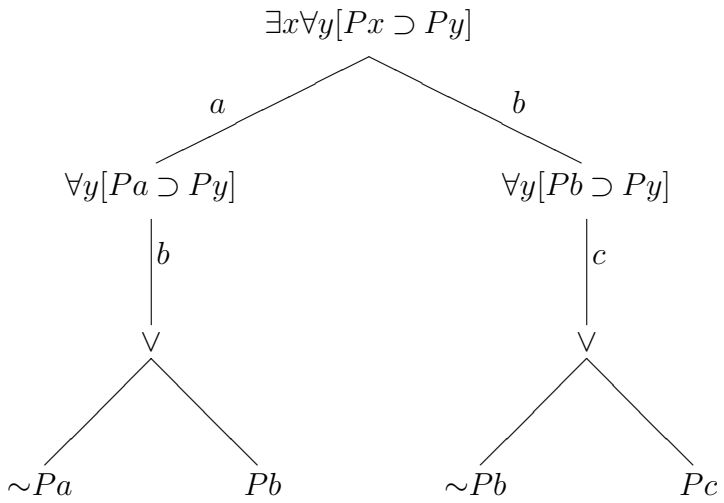
From the point of view of expansion proofs those three solutions are indistinguishable, since expansion proofs for the new subgoals can be identical in all cases.

The problem to be solved here is entirely different from all the problems we have dealt with so far, since we need to construct formulas. These formulas will in general not be subformulas of the planned line or any of the support lines, but nevertheless, they will be constructed from the formulas available in the deduction.

We will first simply present some more examples of desirable deductions. All of these can be obtained by using the *Symmetric Simplification* procedure described in Algorithm 261.

Example 250 (*Math Logic Baffler*) $\exists x \forall y[Px \to Py]$

The expansion proof again



Without any special improvements we get

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[\![Pb]\!] \qquad [\![\neg Pb]\!]}{\bot}\neg\mathrm{E}}{Pc}\bot_I}{Pb \to Pc}\to\mathrm{I}}{\forall y[Pb \to Py]}\forall\mathrm{I}}{\exists x \forall y[Px \to Py]}\exists\mathrm{I} \qquad [\![\neg\exists x \forall y[Px \to Py]]\!]}{\bot}\neg\mathrm{E}}{Pb}\bot_C}{Pa \to Pb}\to\mathrm{I}}{\forall y[Pa \to Py]}\forall\mathrm{I}}{\exists x \forall y[Px \to Py]}\exists\mathrm{I} \qquad [\![\neg\exists x \forall y[Px \to Py]]\!]}{\bot}\neg\mathrm{E}}{\exists x \forall y[Px \to Py]}\bot_C$$

What we would like in the proof by cases: either $\exists x{\sim}Px$ or ${\sim}\exists x{\sim}Px$. This would yield

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[\![Pb]\!] \qquad [\![\neg Pb]\!]}{\bot}\neg\mathrm{E}}{Pc}\bot_I}{Pb \to Pc}\to\mathrm{I}}{\forall y[Pb \to Py]}\forall\mathrm{I}}{\exists x\forall y[Px \to Py]}\exists\mathrm{I} \qquad [\![\exists x\neg Px]\!]}{\exists x\forall y[Px \to Py]}\exists\mathrm{E} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[\![\neg Pb]\!]}{\exists x\neg Px}\exists\mathrm{I} \qquad [\![\neg\exists x\neg Px]\!]}{\bot}\neg\mathrm{E}}{Pb}\bot_C}{Pa \to Pb}\to\mathrm{I}}{\forall y[Pa \to Py]}\forall\mathrm{I}}{\exists x\forall y[Px \to Py]}\exists\mathrm{I} \qquad \exists x\neg Px \vee \neg\exists x\neg Px}{\exists x\forall y[Px \to Py]}\vee\mathrm{E}$$

Once we have found an appropriate lemma in the expansion proof, all we have really done is find an equivalence class of formulas in $\mathcal{LN}$. We have to select a representative of this equivalence class. If the representatives are closely related, one (in general) obtains a simpler lemma (as in the example above), but perhaps more complicated subdeductions. If the representatives diverge further, the lemma will be harder to prove, but the subdeductions become simpler.

It seems to us that a more difficult proof for the lemma is a small price to pay for an improvement in the main deduction. The lemmas are often of a general nature, and may even be among a fixed set of lemmas one is allowed to use.

To illustrate this point, we go back to preceding example and present yet another deduction, perhaps even better.

Example 251

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[\![Pb]\!] \qquad [\![\neg Pb]\!]}{\bot}\neg\mathrm{E}}{Pc}\bot_I}{Pb \to Pc}\to\mathrm{I}}{\forall y[Pb \to Py]}\forall\mathrm{I}}{\exists x\forall y[Px \to Py]}\exists\mathrm{I} \qquad [\![\exists x\neg Px]\!]}{\exists x\forall y[Px \to Py]}\exists\mathrm{E} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[\![\forall xPx]\!]}{Pb}\forall\mathrm{I}}{Pa \to Pb}\to\mathrm{I}}{\forall y[Pa \to Py]}\forall\mathrm{I}}{\exists x\forall y[Px \to Py]}\exists\mathrm{I} \qquad \begin{array}{c}\vdots\\ \exists x\neg Px \vee \forall xPx\end{array}}{\exists x\forall y[Px \to Py]}\vee\mathrm{E}$$

In the remainder of this section we deal with the problem of finding equivalence classes of formulas appropriate as disjunctive lemmas. We have already indicated how this also allows us to find simple contradictory formulas for indirect proofs.

This algorithm is most useful when all support lines are negations of disjunctions (such that neither disjunct is inessential) or negations of existentially quantified formulas. In the second case it is usually best to apply it to a support line whose corresponding expansion node has more than one expansion term, at least one of which is admissible.

These cases can be recognized as the ones where the *Simple* $\neg\vee$ *Elimination* and *Simple* $\neg\exists$ *Elimination* tactics (Tactics 218 and 220) do not apply, and one has to double the support line in question.

The basic observation is that the planned line (before the indirect proof, when we are looking for a disjunctive lemma), or the negated assumption (after the indirect proof, when we are looking for two contradictory formulas) by themselves are sufficient as a disjuncts for the lemma.

We then build an expansion proof under the assumption that the whole formula will be a disjunct in the lemma. We then try to simplify this formula, or rather the new expansion proof. If simplification succeeds, the simplified formula is suggested as a lemma. If no simplification is possible, we interpret it as failure and make no use of the lemma. If symmetric simplification is called only in the two cases outlined above, simplification is always possible, though perhaps only to a minor degree.

First let us describe more concretely which problem is solved by the symmetric simplification algorithm.

> Given an expansion proof $(Q, \mathcal{M})$ for $U, C \wedge \sim C, A, B$ such that U, A and U, B separate $C \wedge \sim C$. This means that $M|_{U,C,A}$ is clause-spanning on U, C, A, and $M|_{U,\sim C,B}$ is clause-spanning on $U, \sim C, B$ (see Definition 73).
>
> Find a formula D and expansion proof $(R, \mathcal{N})$ for $U, D \wedge \sim D, A, B$ such that U, A and U, B separate $D \wedge \sim D$.

Of course, using C for D and $(Q, \mathcal{M})$ for $(R, \mathcal{N})$ will satisfy this specification, so the algorithm we provide will be a heuristic improvement of this trivial one. What is being improved? We can only state informally that the natural deductions corresponding to the "symmetrically simplified" expansion proof are more intuitive.

In the first-order case we could require that D's complexity (number of quantifiers and connectives) is less than C's, but in the general higher-order case this is not desirable, since the instantiation phase (see below) may actually increase the formula's complexity.

We would also like $(R, \mathcal{N})$ to be minimal, but it is undecidable if $(R, \mathcal{N})$ is a minimal solution, since we do not require a relation between R_A and Q_A or R_B and Q_B. We do not require such a connection so that some substitution can still be carried out by our algorithm.

From now on in this section we relate a node in Q_C to the corresponding node in $Q_{\sim C}$ by putting a prime on one or the other node name.

The algorithm iterates three phases. The *substitution phase* shallows the expansion proof at some accessible node $Q_{\exists x E(x)}$ in $Q_{C \wedge \sim C}$ and shallows and substitutes for the parameter selected at $Q'_{\forall x \sim E(x)}$. The *deletion phase* deletes corresponding subtrees from Q_C and $Q_{\sim C}$. The *mating phase* changes the mating so as to allow possibly more substitutions and deletions during the next iteration of the three phases. If the mating phase cannot make useful changes, the algorithm terminates.

Algorithm 252 (*Single Instantiation*) Let $Q_{\exists xE(x)}$ be a single and accessible node in Q_C or Q'_C with exactly one expansion term t and t is admissible. Assume moreover that $Q'_{\forall xE(x)}$ is also single, that is, not $\mathcal{M}$-mated.

Then apply $\texttt{shallow}(Q_{\exists xE(x)}, (Q, \mathcal{M}))$ to obtain an intermediate expansion tree. $Q'_{\forall x \sim E(x)}$, now will have an incorrect shallow formula, since it is not the negation of $E(t)$. In order to achieve that, we first apply $\texttt{shallow}(Q_{\forall x \sim E(x)})$ which results in the shallow $\sim E(a)$ for the selected parameter a. Now we apply $\theta = [a \mapsto t]$ to the entire expansion proof and obtain $\sim E(t)$ as shallow formula. Let us call the result R, and let $\texttt{snginst}(Q_{\exists xE(x)}, (Q, \mathcal{M})) = (R, \mathcal{M})$.

Lemma 253 Given an expansion proof $(Q, \mathcal{M})$ for $U, C[\![\exists xE(x)]\!] \wedge \sim C[\![\exists xE(x)]\!], A, B$ such that U, A and U, B separate $C \wedge \sim C$. If the conditions for applying *Single Instantiation* are satisfied, then $(R, \mathcal{M}) = \texttt{snginst}(Q_{\exists xE(x)}, (Q, \mathcal{M}))$ is an expansion proof for $U, C[\![E(t)]\!] \wedge \sim C[\![E(t)]\!], A, B$ and U, A and U, B separate $C[\![E(t)]\!] \wedge \sim C[\![E(t)]\!]$.

Proof: Since the mating has not changed, it will still be separating, and it is clause-spanning since the deleted nodes did not occur in it. The shallowing is legal, since t was admissible (see Lemma 53).

□

Definition 254 We call a node Q_A a *disjunct in* Q if it is directly below a disjunction node in Q. Similarly, we call Q_A a *conjunct in* Q if it is directly below a conjunction node in Q.

Algorithm 255 (*Single Deletion*) Let E be a subformula of the shallow formula of C or $\sim C$, such that Q_{E^i} for each, possibly instantiated, copy E^i of E in Q is a disjunct, and none of the E^i or any of its subtrees is $\mathcal{M}$-mated.

Then erase all Q_{E^i} and $Q'_{\sim E^j}$, where $Q'_{\sim E^j}$ are all the possibly instantiated nodes coming from $\sim E$ in $Q_{\sim C}$. Let us call this expansion tree R. $\mathcal{N}$ is obtained from $\mathcal{M}$ simply by restricting it to elements in R. Then $(R, \mathcal{N}) = \texttt{sngdel}(E, (Q, \mathcal{M}))$.

Note that the $\sim E^j$ could be mated, or even be expansion or selection nodes with complicated expansion trees below them. This is why the algorithm is so useful.

Lemma 256 When the conditions for calling *Single Deletion* are satisfied, then $(R, \mathcal{N}) = \texttt{sngdel}(E, (Q, \mathcal{M}))$ is an expansion proof for $U, D \wedge \sim D, A, B$ such that U, A and U, B are separating $D \wedge \sim D$ and D is obtained from C by erasing E.

Proof: The shallow formula of R is of the correct form since all copies of E and $\sim E$ are erased from Q. If Q_E is not accessible one has to check that if it lies below an expansion node, the shallow formula of all of its successor is the instantiation of the shallow formula of the expansion node. But this is true since we erased even instantiated copies of E.

The main work here is to check that the mating is still clause-spanning, since we erased some possibly mated nodes of the form $Q'_{\sim E^j}$. Note that since all Q_{E^i}'s were disjuncts in Q, all $Q_{\sim E^j}$ are conjuncts in Q.

Assume without loss of generality that Q_E was a disjunct in Q_C. Now let c be a full clause in R. It either contains nodes in R_C or $R_{\sim C}$, but not both. There is a preimage c' in Q, which may just be a partial clause.

If c it contains nodes in R_C, then any extension of c' in Q by nodes in some Q_{E^i} will be spanned by a pair $(l,k) \in \mathcal{M}$. Since $\mathcal{M}$ does not mate any of the Q_{E^i} or nodes below, this same pair (l,k) belongs to $\mathcal{N}$, and c is spanned by $\mathcal{N}$.

If c contains nodes in $Q_{\sim C}$ then the preimage c' already occurs in Q, since only conjuncts have been deleted. The pair (l,k) which spanned c' will still span c in R and belong to $\mathcal{N}$.

□

After some subtrees and subformulas have been erased, it often helps to see if the mating can be manipulated so that more of C and $\sim C$ can be erased during the next iteration of the overall algorithm. This is done by systematically enumerating the full clauses and marking those pairs in $\mathcal{M}$ which are the only possible pair closing a path. One has to be careful to exclude those pairs which would destroy the separation property we require.

Definition 257 Given a symmetric simplification problem $U, C \wedge \sim C, A, B$ with expansion proof $(Q, \mathcal{M})$ such that U, A and U, B separate $C \wedge \sim C$. We call a pair of nodes (l,k) *separation preserving* if either both l and k are in $Q|_{U,C,A}$, or both are in $Q|_{U,\sim C,B}$.

Algorithm 258 (*Single Mating Change*)

1. Let `nec` $:= \{\}$. `nec` contains the pairs in $\mathcal{M}$ which have been determined to be strictly necessary.
2. For each full clause c in $\mathrm{fc}(Q)$ do
   2.1. If c is closed by more than one pair in $\mathcal{M}$, nothing.
   2.2. If c is closed by exactly one pair p and this pair contains a node in Q_C or $Q_{\sim C}$, check if there is another separation preserving pair (l,k) on c such that $l^S \underset{\lambda\sim}{=} \sim k^S$ and neither l nor k are nodes in Q_C and $Q_{\sim C}$. If yes, do nothing. Else let `nec` $:=$ `nec` $\cup \{p\}$. p can not be deleted from the mating.
   2.3. Otherwise do nothing.
3. Reduce $\mathcal{M}$ to $\mathcal{M}'$ by deleting an arbitrary pair which has an element in Q_C or $Q_{\sim C}$, but has not been marked as necessary, that is, does not belong to `nec`. If no such pair exists, return $\mathcal{M} = \mathtt{sngmat}(Q, \mathcal{M})$.
4. Enumerate the full clauses once again, adding pairs to $\mathcal{N}$ whenever a clause is encountered which is not spanned by $\mathcal{M}'$ or the $\mathcal{N}$ built up so far. These spanning pairs are guaranteed to exist because of the condition in Step 2.
5. Return $\mathcal{N} = \mathtt{sngmat}(\mathcal{M})$

Lemma 259 Given a symmetric simplification problem as stated above. Then $\mathcal{N} = \texttt{sngmat}(\mathcal{M})$ is still a separating and clause-spanning mating on Q.

Proof: Let c be a full clause in Q. If it was closed by exactly one pair, that pair either did not contain a node in Q_C or $Q_{\sim C}$, or it was marked as necessary, or there is another separation-preserving pair of nodes in c. In the first two cases, the pair will remain in $\mathcal{N}$. In the second pair one of those separation preserving, complementary pairs will be added in Step 4.

If c was closed by more than one pair, at most one of the pairs could have been deleted in Step 3 and it will still be spanned by $\mathcal{N}$.

□

Remark 260 Since potentially many pairs are added in Step 4 this may not seem like much of an improvement. However, the number of pairs in $\mathcal{M}$ with elements in Q_C or $Q_{\sim C}$ will decrease, or remain the same, if no pair is expendable.

It remains to put the pieces together into one algorithm:

Algorithm 261 (*Symmetric Simplification*) Given a symmetric simplification problem $U, C \wedge \sim C, A, B$ with expansion proof $(Q, \mathcal{M})$ such that U, A and U, B separate $C \wedge \sim C$. Do:

1. Traverse the expansion trees Q_C and $Q_{\sim C}$ simultaneously stopping at mated interior nodes, and expansion nodes. Apply *Single Instantiation* if possible. If *Single Instantiation* was possible, traverse the new expansion proof at that point, also.
2. Traverse the result of Step 1 from the bottom up, applying *Single Deletion* wherever possible.
3. Apply *Single Mating Change* once. If there is no change in the mating, no further simplifications will be possible. Return. Otherwise repeatedly apply *Single Mating Change* until no longer any improvement is made. Then go to Step 1.

Theorem 262 The *Symmetric Simplification Algorithm* is correct and will always terminate.

Proof: The correctness follows easily from Lemmas 253, 256, and 259, since these steps are merely iterated. By correctness we mean that any application of the symmetric simplification algorithm will result in a valid expansion proof.

It is also easy to see that the iterations at each step terminate: *Single Instantiation* will make the expansion proof shallower, *Single Deletion* is merely an erasing of subtrees, and *Single Mating Change* decreases the number of pairs in the mating with elements in Q_C or $Q_{\sim C}$ until no more of these pairs can be deleted. But then *Symmetric Simplification* also stops this inner iteration.

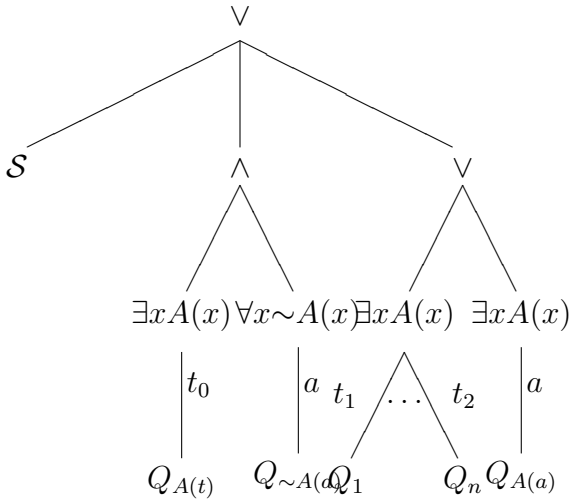
Since the three operations do not interact in the sense that one simplification will complicate another measure, the whole algorithm must eventually terminate.

□

Remark 263 Mated interior nodes, that is, nodes which are not leaves, seem to inhibit this algorithm. Therefore it may be a good idea to do some deepening (see Definition 124) before or during the algorithm.

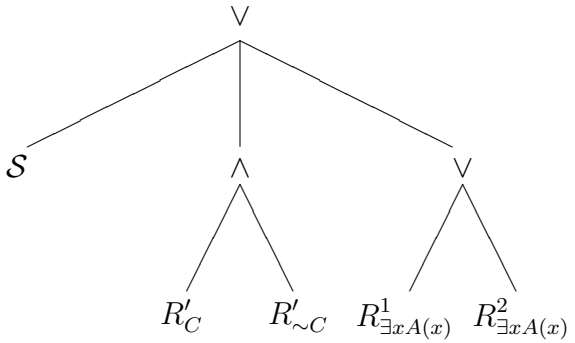
When and how is it useful to apply symmetric simplification? We present two tactics which seem to cover most of our examples.

Tactic 264 (∃ *Lemma Tactic*) If the planned line is of the form $\exists x A(x)$ and the set of expansion terms at $Q_{\exists x A(x)}$ is $\{t_0, t_1, \ldots, t_n\}$ for some $n \geq 1$ and t_0 is admissible, then construct the following symmetric simplification problem $(R, \mathcal{N})$. Let $R =$

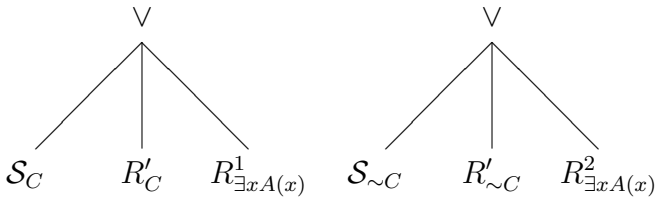


The mating $\mathcal{N}$ behaves on $\mathcal{S}$, $Q_{A(t)}$ and $Q_1, \ldots, Q_n$ like $\mathcal{M}$ before. $\mathcal{N}$ also mates every leaf in $Q_{\sim A(a)}$ with the correspoding complementary leaves in $Q_{A(a)}$. Clearly $\mathcal{N}$ is clause-spanning on R. It also separates $\exists x A(x)$ and $\forall x \sim A(x)$.

Now apply symmetric simplification to $(R, \mathcal{N})$ to obtain $(R', \mathcal{N}')$. Note that t_0 will be instantiated since it was chosen to be admissible. This leads to a new expansion proof of the following structure



and $\mathcal{N}'$ still separates $C \wedge \sim C$. Now we use $\wedge \mathrm{E}_1$ and $\wedge \mathrm{E}_2$ on expansion proofs to obtain the following two subproofs

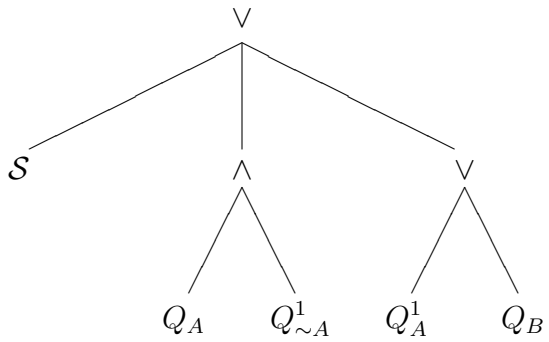


which are expansion proofs for subgoals remaining in

$$\dfrac{\begin{array}{ccc}\mathcal{S}_D, [\![D]\!] & \mathcal{S}_E, [\![E]\!] & \\ \vdots & \vdots & \vdots \\ \exists x A(x) & \exists x A(x) & D \vee E\end{array}}{\exists x A(x)} \vee \mathrm{E}$$

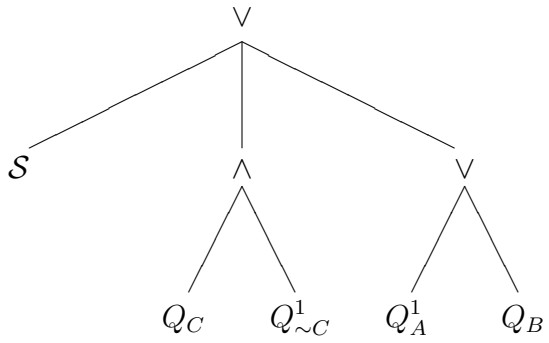
where D and E are chosen such that $\overline{D} = C$ and $\overline{E} = \sim C$. Also, there is another but trivial expansion proof for the subgoal of proving $D \vee E$, if it is not a derived rule of inference.

Tactic 265 ($\vee$ *Lemma Tactic Left*) If the planned line is of the form $A \vee B$ such that neither A not B is unnecessary, then construct the following symmetric simplification problem $(R, \mathcal{N})$. Let $R =$



where the mating $\mathcal{N}$ mates corresponding nodes in $Q^1_{\sim A}$ and Q^1_A and is otherwise like $\mathcal{M}$.

Now apply symmetric simplification to obtain $(R', \mathcal{N}')$, where $R' =$



Since we still have separation of $C \wedge {\sim}C$, we can again get two subgoals by useing $\wedge\mathrm{E}_1$ and $\wedge\mathrm{E}_2$ for some sufficent subsets of $\mathcal{S}$ and get the following piece of deduction

$$\dfrac{\begin{array}{ccccc} \mathcal{S}_D, [\![D]\!] & & \mathcal{S}_E, [\![E]\!] & & \\ \vdots & & \vdots & & \vdots \\ A \vee B & & A \vee B & & D \vee E \end{array}}{A \vee B} \vee\mathrm{E}$$

where D and E are chosen such that $\overline{D} = C$ and $\overline{E} = {\sim}C$. Also, there is another but trivial expansion proof for the subgoal of proving $D \vee E$, if it is not a derived rule of inference.

Tactic 266 ($\vee$ *Lemma Tactic Right*) Symmetric to the previous tactic, except that the symmetric simplification starts with $B \vee {\sim}B$ as initial lemma, rather than $A \vee {\sim}A$.

Bibliography

[1] Peter B. Andrews. *An Introduction to Mathematical Logic and Type Theory: To Truth Through Proof*. Academic Press, 1986.

[2] Peter B. Andrews. Transforming matings into natural deduction proofs. In *Proceedings of the 5th Conference on Automated Deduction, Les Arcs, France*, pages 281–292, Springer-Verlag, 1980.

[3] Peter B. Andrews, Dale Miller, Eve Cohen, and Frank Pfenning. Automating higher-order logic. *Contemporary Mathematics*, 29:169–192, August 1984.

[4] Hendrik P. Barendregt. *The Lambda-Calculus: Its Syntax and Semantics*. North-Holland, 1980.

[5] Joseph Bates and Robert Constable. Proofs as programs. *ACM Transactions on Programming Languages and Systems*, 7(1):113–136, January 1985.

[6] John L. Bell. *Boolean-Valued Models and Independence Proofs in Set Theory. Oxford Logic Guides*, Clarendon Press, Oxford, 1977.

[7] Wolfgang Bibel. On matrices with connections. *Journal of the Association of Computing Machinery*, 28:633–645, 1981.

[8] Alonzo Church. A formulation of the simple theory of types. *Journal of Symbolic Logic*, 5:56–68, 1940.

[9] Thierry Coquand and Gerard Huet. Constructions: a higher order proof system for mechanizing mathematics. In *EUROCAL85*, Springer-Verlag LNCS 203, 1985.

[10] François Fages and Gérard Huet. Unification and matching in equational theories. In *CAAP 83*, pages 205–220, Springer Verlag (LNCS 159), 1983.

[11] Gerhard Gentzen. Untersuchungen über das logische Schließen. *Mathematische Zeitschrift*, 39:176–210, 405–431, 1935.

[12] Jean-Yves Girard. *Interprétation fonctionelle et élimination des coupures de l'arithmétique d'ordere supérieur*. PhD thesis, Université Paris VII, 1972.

[13] Jean-Yves Girard. Une extension de l'interpretation de Gödel a l'analyse, et son application a l'elimination des coupures dans l'analyse et la theorie des types. In J. E. Fenstad, editor, *Proceedings of the Second Scandinavian Logic Symposium*, pages 63–92, North-Holland Publishing Co., Amsterdam, London, 1971.

[14] Michael J. Gordon, Arthur J. Milner, and Christopher P. Wadsworth. *Edinburgh LCF*. Volume 78 of *Lecture Notes in Computer Science*, Springer Verlag, 1979.

[15] Leon Henkin. Completeness in the theory of types. *Journal of Symbolic Logic*, 15:81–91, 1950.

[16] Jacques Herbrand. Recherches sur la théorie de la démonstration. *Travaux de la Société des Sciences et de Letrres de Varsovic*, 33, 1930.

[17] Georg Kreisel. Stanford report on the consistency of analysis, appendix ii. 1962.

[18] Per Martin-Löf. About models for intuitionistic type theories and the notion of definitional equality. In S. Kanger, editor, *Proceedings of the Third Scandinavian Logic Symposium*, pages 81–109, North Holland Publishing Co., Amsterdam, 1975.

[19] Per Martin-Löf. Constructive mathematics and computer programming. In *Logic, Methodology and Philosophy of Science VI*, pages 153–175, North-Holland, 1980.

[20] Per Martin-Löf. *Hauptsatz for Intuitionistic Simple Type Theory*, pages 279–290. Volume IV, North-Holland Publishing Co., Amsterdam, 1973.

[21] Per Martin-Löf. Hauptsatz for the theory of species. In J. E. Fenstad, editor, *Proceedings of the Second Scandinavian Logic Symposium*, pages 217–233, North-Holland Publishing Co., Amsterdam, London, 1971.

[22] Dale Miller. Expansion tree proofs and their conversin to natural deduction proofs. In R. E. Shostak, editor, *Proceedings of the 7th Conference on Automated Deduction*, pages 375–393, Springer Verlag, Heidelberg, May 1984.

[23] Dale Miller. *Proofs in Higher-Order Logic*. PhD thesis, Carnegie-Mellon University, October 1983.

[24] Dale Miller and Amy Felty. An integration of resolution and natural deduction theorem proving. In *Proceedings of the National Conference on Artificial Intelligence*, pages 633–645, American Association for Artificial Intelligence, August 1986.

[25] Ketan Mulmuley. The mechanization of existence proofs of recursive predicates. In R. E. Shostak, editor, *Proceedings of the 7th International Conference on Automated Deduction*, pages 460–475, Springer-Verlag, May 1984.

[26] Frank Pfenning. Analytic and non-analytic proofs. In R. E. Shostak, editor, *Proceedings of the 7th Conference on Automated Deduction*, pages 394–413, Springer Verlag, Heidelberg, May 1984.

[27] Dag Prawitz. Ideas and results in proof theory. In J. E. Fenstad, editor, *Proceedings of the Second Scandinavian Logic Symposium*, pages 235–307, North-Holland Publishing Co., Amsterdam, London, 1971.

[28] Dag Prawitz. *Natural Deduction*. Almquist & Wiksell, Stockholm, 1965.

[29] Dag Prawitz. Validity and normalizability of proofs in 1st and 2nd order classical and intuitionistic logic. 1975. Lecture at Oxford.

[30] Kurt Schütte. Schlußweisen-kalküle der prädikatenlogik. *Mathematische Annalen*, 39(122):47–65, 1951.

[31] Dana Scott. Boolean-valued models for higher-order logic. 1966. Duplicated Notes, Stanford University.

[32] Jörg Siekmann. Universal unification. In *Proceedings of the 7th Conference on Automated Deduction*, pages 1–42, Springer Verlag (LNCS 170), 1984.

[33] Raymond Smullyan. *First-Order Logic*. Volume 5, Springer Verlag, 1968.

[34] Richard Statman. Bounds for proof-search and speed-up in the predicate calculus. *Annals of Mathematical Logic*, (15):225–287, 1978.

[35] Richard Statman. *Structural Complexity of Proofs*. PhD thesis, Stanford University, 1974.

[36] W. Tait. A nonconstructive proof of gentzen's hauptsatz for second order predicate logic. *Bulletin of the American Mathematical Society*, (72):980–983, 1966.

[37] Moto-o Takahashi. Cut-elimination in simple type theory with extensionality. *Journal of the Mathematical Society of Japan*, (20), 1968.

[38] Moto-o Takahashi. A proof of cut-elimination in simple type theory. *Journal of the Mathematical Society of Japan*, (19):399–410, 1967.

[39] Gaisi Takeuti. On a generalized logical calculus. *Japanese Journal of Mathematics*, 23:39–96, 1953.

[40] Gaisi Takeuti. *Proof Theory*. Volume 81 of *Studies in Logic and the Foundations of Mathematics*, North-Holland, Amsterdam, 1975.

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

[42] Unknown Author (a student advised by Professor Gandy). *Unknown Title (manuscript without title page in my possession)*. PhD thesis, Worcester College, Unknown Year.

Glossary

$\mathcal{T}$	Set of type symbols, 9
ι	Type of individuals, 9
o	Type of propositions, 9
$\mathcal{L}$	Language, 10
$x, y, \ldots$	Variables, 10
$a, b, c, \ldots$	Parameters, 10
λ	Lambda binder, 10
$\sim$	Negation, 10
$\wedge$	Conjunction, 10
$\vee$	Disjunction, 10
$\forall$	Universal Quantifier, 10
$\exists$	Existential Quantifier, 10
$\underset{\text{def}}{=}$	Defined as, 10
$\supset$	Implication (defined), 10
$\equiv$	Equivalence (defined), 10
θ	Substitution, 10
$[a_1 \mapsto t_1, a_2 \mapsto t_2, \ldots]$	Substitution, 10
$\theta \oplus [a \mapsto t]$	θ modified at a, 10
θA	Application of θ to A, 10
$\underset{\beta}{\longrightarrow}$	Beta reduction, 11

$\underset{\lambda}{=}$	Equal up to λ-conversion, 11
$\underset{\sim}{\longrightarrow}$	Negation reduction, 11
$\underset{\lambda^\sim}{=}$	Equal up to $\lambda^\sim$-conversion, 11
$A, B, C, \ldots$	Arbitrary formulas, 12
$U, V, W, \ldots$	Multisets of formulas, 12
$\mathcal{D}, \mathcal{E}, \mathcal{F}, \ldots$	Arbitrary Deductions, 12
$\mathcal{H}$	Deductive system, 12
C	Contraction, 13
$\wedge I$	And Introduction Rule, 13
$\vee I$	Or Introduction Rule, 13
$\forall$I	Forall Introduction Rule, 13
$\exists$I	Exists Introduction Rule, 13
Cut	Rule of Cut, 13
I	Initial deductions, 13
$\mathcal{H}^-$	System $\mathcal{H}$ without cut, 14
$\mathtt{Final}(\mathcal{D})$	Multiset in final line of $\mathcal{D}$, 14
$\begin{matrix}\mathcal{D}\\U\end{matrix}$	Deduction of U, 14
$\dfrac{\dfrac{U'}{=}}{U}\, r$	Several applications of r, 14
$\lhd$	Immediate subderivation, 15
$\mathtt{Cut}(\mathcal{D}, \mathcal{E}, A, \sim A)$	Cut of $\mathcal{D}$ and $\mathcal{E}$, eliminating A, 15
$\mathcal{D} \oplus A$	Adjoining A to $\mathcal{D}$, 15
$\mathcal{D} \oplus U$	Adoining U to $\mathcal{D}$, 15
$\ll^{\mathcal{D}}$	Of lower rank in $\mathcal{D}$, 17
$\mathcal{D} \Rightarrow \mathcal{E}$	$\mathcal{D}$ goes to $\mathcal{E}$ by an essential reduction, 19

$\mathcal{D} \underset{p}{\rightarrow} \mathcal{E}$	$\mathcal{D}$ goes to $\mathcal{E}$ by a commutative reduction, 20
$\mathcal{D} \underset{c}{\rightarrow} \mathcal{E}$	$\mathcal{D}$ goes to $\mathcal{E}$ by a contraction reduction, 21
$\longrightarrow\!\!\!\!\rightarrow$	Reduces by commutative or contraction reductions, 23
$\mathrm{cr}^{\mathcal{D}}(A)$	Contraction rank of A in $\mathcal{D}$, 23
Q^D	Deep formula of Q, 27
Q^S	Shallow formula of Q, 27
Q_A	Node in Q with shallow formula A, 28
$Q\|_U$	Restriction of Q to U, 28
$\mathcal{M}\|_U$	Restriction of $\mathcal{M}$ to $Q\|_U$, 28
$t <^0_Q s$	t depends directly on s, 29
$t <_Q s$	t depends on s, 29
$a \prec^0_Q b$	a is directly imbedded in b, 29
$a \prec_Q b$	a is imbedded in b, 29
$\langle e_1, \ldots, e_n \rangle$	List of e_1 through e_n, 29
$\mathrm{fc}(Q)$	Full clauses in Q, 31
$\mathrm{fc}(Q) \backslash Q_0$	Full clauses without Q_0, 31
$\mathtt{shallow}(Q_0, (Q, \mathcal{M}))$	Shallow $(Q, \mathcal{M})$ at Q_0, 32
$Q \sqsubset_D R$	Q from R by shallowings and deletions, 32
$\mathtt{erase}(Q_0, (Q, \mathcal{M}))$	Erasing Q_0 form $(Q, \mathcal{M})$, 34
$\mathtt{crleaf}(Q_A, (Q, \mathcal{M}))$	Creating new leaf at Q_A in $(Q, \mathcal{M})$, 34
$(Q, \mathcal{M}) \sqsubset_C (R, \mathcal{N})$	$(Q, \mathcal{M})$ from $(R, \mathcal{N})$ by creation of leaves, 35
$\mathtt{split}$	Splitting an expansion proof, 35
$(Q, \mathcal{M}) \sqsubset_S (R, \mathcal{N})$	$(Q, \mathcal{M})$ from $(R, \mathcal{N})$ by splitting expansion nodes, 35
$(Q, \mathcal{M}) \sqsubset (R, \mathcal{N})$	$(Q, \mathcal{M})$ is simpler than $(R, \mathcal{N})$, 35
$U[\![A]\!]$	A accessible in U, 36
Q_A	Node in Q with shallow formula A, 36

$\texttt{double}(Q_A, (Q, \mathcal{M}))$	Double Q_A in $(Q, \mathcal{M})$, 38
$\vee\mathrm{E}(Q_{A\vee B}, (Q, \mathcal{M}))$	Or elimination in expansion proofs, 38
$\forall\mathrm{E}(Q_{\forall xA}, (Q, \mathcal{M}))$	Forall elimination in expansion proofs, 38
$\exists\mathrm{E}(Q_{\exists xA}, (Q, \mathcal{M}))$	Exists elimination in expansion proofs, 38
$\wedge\mathrm{E}_1(Q_{A\wedge B}, U^A, (Q, \mathcal{M}))$	And elimination in expansion proofs, 38
$\wedge\mathrm{E}_2(Q_{A\wedge B}, U^B, (Q, \mathcal{M}))$	And elimination in expansion proofs, 38
$\exists\mathrm{I}(Q_{A(t)}, \exists xA(x), (Q, \mathcal{M}))$	Exists introduction in expansion proofs, 42
$\forall\mathrm{I}(Q_{A(a)}, \forall xA(x), (Q, \mathcal{M}))$	Forall introduction in expansion proofs, 42
$\texttt{merge}(Q_1, Q_2, (Q, \mathcal{M}))$	Merging Q_1 and Q_2 in $(Q, \mathcal{M})$, 43
$\mathtt{m}(P, Q)$	Shallow merge of P and Q, 43
$\mathcal{Q}, \mathcal{R}, \ldots$	Expansion developments, 54
$\mathcal{Q} \underset{i}{\Longrightarrow} \mathcal{R}$	Initial expansion reduction, 55
$\mathcal{Q} \underset{p}{\Longrightarrow} \mathcal{R}$	Permutative expansion reductions, 56
$\mathcal{Q} \underset{e}{\Longrightarrow} \mathcal{R}$	Essential Expansion Reduction, 56
$\mathcal{Q} \underset{m}{\Longrightarrow} \mathcal{R}$	Merge expansion reduction, 57
$\mathcal{Q} \overset{*}{\Longleftrightarrow} \mathcal{R}$	$\mathcal{Q}$ reduces to $\mathcal{R}$, 57
$\mathcal{Q} \overset{*}{\underset{i}{\Longleftrightarrow}} \mathcal{R}$	convertible by initial conversions, 57
$\mathcal{Q} \Longrightarrow \mathcal{R}$	convertible by $\underset{p}{\Longrightarrow}$, $\underset{m}{\Longrightarrow}$, or $\underset{e}{\Longrightarrow}$, 57
$Q_0 \lhd Q_1$	Q_0 above Q_1, 60
$t \lll s$	t above or dependent on s, 60
$\mathcal{L}^=$	Language $\mathcal{L}$ with primitve equality, 64
$\doteq$	Primitive Equality, 64
$\mathcal{H}^=$	System $\mathcal{H}$ with primitive equality, 64
$R\doteq$	Reflexivity Axiom, 64

$S\doteq$	Rule of Substitution, 64
$\texttt{deepen}(Q_0, (Q, \mathcal{M}))$	Deepen $(Q, \mathcal{M})$ at Q, 66
$\prec$	Below, 66
$\prec\!\!\prec$	Imbedded or below, 66
$\mathcal{H}^*$	$\mathcal{H}^=$ with reverse substitution rule, 85
$\mathcal{H}^e$	System $\mathcal{H}$ with extensionality, 94
$\mathcal{LN}^1$	Language, 105
$\bot$	Absurdity, 105
$\neg$	Negation, 105
$\wedge$	Conjunction, 105
$\vee$	Disjunction, 105
$\rightarrow$	Implication, 105
$\forall$	Universal Quantifier, 105
$\exists$	Existential Quantifier, 105
$\wedge I$	And Introduction Rule in $\mathcal{N}$, 106
$\wedge \mathrm{E}_L$	And Elimination Left in $\mathcal{N}$, 106
$\wedge \mathrm{E}_R$	And Eliminiation Right in $\mathcal{N}$, 106
$\vee \mathrm{I}_L$	Or Introduction Left in $\mathcal{N}$, 106
$\vee \mathrm{I}_R$	Or Introduction Right in $\mathcal{N}$, 106
$\vee \mathrm{E}$	Or Elimination in $\mathcal{N}$, 106
$\rightarrow \mathrm{I}$	Arrow Introduction in $\mathcal{N}$, 106
$\rightarrow \mathrm{E}$	Arrow Elimination in $\mathcal{N}$, 106
$\neg \mathrm{I}$	Negation Introduction in $\mathcal{N}$, 106
$\neg \mathrm{E}$	Negation Elimination in $\mathcal{N}$, 106
$\bot_I$	Intuitionistic Absurdity Rule, 106
$\bot_C$	Classical Proof by Contradiction, 106

$\forall$I	Forall Introduction in $\mathcal{N}$, 106
$\forall$E	Forall Elimination in $\mathcal{N}$, 106
$\exists$I	Exists Introduction in $\mathcal{N}$, 106
$\exists$E	Exists Elimination in $\mathcal{N}$, 106
$\mathcal{L}'$	Language, 108
$\mathtt{F}$	Falsehood in $\mathcal{L}'$, 108
$\mathtt{T}$	Truth in $\mathcal{L}'$, 108
$Q \sqsubset_O R$	Q from R by splitting a disjunction, 131
$(Q, \mathcal{M}) \sqsubset' (R, \mathcal{N})$	$(Q, \mathcal{M})$ is simpler than $(R, \mathcal{N})$, 131
λ	Lambda binder, 133
Ext	Extensionality Rule in $\mathcal{N}^e$, 133

Index