eq

PROOFS IN HIGHER-ORDER LOGIC

Dale A. Miller

October 1983
Tech Report: MS-CIS-83-37

This dissertation was submitted to the Mathematics Department of Carnegie-Mellon University on 27 August 1983 in partial fulfillment of the requirements for the degree of Doctor of Philosophy. In October 1983 it was released as Tech Report MS-CIS-83-37 from the Computer Science and Information Department at the University of Pennsylvania. The current version is the result of recompiling the original TeX source files (not LaTeX files) in August 2004. In the process, three figures have not been reproduced (they are not particularly important for the text).

This work was supported by NSF grant MCS81-02870.

Abstract

Expansion trees are defined as generalizations of Herbrand instances for formulas in a nonextensional form of higher-order logic based on Church's simple theory of types. Such expansion trees can be defined with or without the use of skolem functions. These trees store substitution terms and either critical variables or skolem terms used to instantiate quantifiers in the original formula and those resulting from instantiations. An expansion tree is called an expansion tree proof (ET-proof) if it encodes a tautology, and, in the form not using skolem functions, an "imbedding" relation among the critical variables be acyclic. The relative completeness result for expansion tree proofs not using skolem functions, *i.e.* if A is provable in higher-order logic then A has such an expansion tree proof, is based on Andrews' formulation of Takahashi's proof of the cut-elimination theorem for higher-order logic. If the occurrences of skolem functions in instantiation terms are restricted appropriately, the use of skolem functions in place of critical variables is equivalent to the requirement that the imbedding relation is acyclic. This fact not only resolves the open question of what is a sound definition of skolemization in higher-order logic but also provides a direct, syntactic proof of its correctness.

Since subtrees of expansion trees are also expansion trees (or their dual) and expansion trees store substitution terms and critical variables explicitly, ET-proofs can be directly converted into sequential and natural deduction proofs. A naive translation will often produce proofs which contain a lot of redunancies and will often use implicational lines in an awkward fashion. An improved translation process is presented. This process will produce only focused proofs in which much of the redunancy has been eliminated and backchaining on implicational lines was automatically selected if it was applicable. The information necessary to construct focused proofs is provided by a certain connection scheme, called a mating, of the boolean atoms within the tautology encoded by an ET-proof.

Acknowledgments

I am greatly indebted to my advisor and my mentor Peter Andrews. During the five years I spent at Carnegie-Mellon University, he not only taught me my first logic lesson but also all my important logic lessons. He was always very patient, encouraging, and exacting whenever I would confront him with half-baked solutions to problems I often did not fully understand. His financial support of me as a programmer and assistant on his theorem proving project is greatly appreciated. The other members of my committee, Professors George Fix, Daniel Leivant, and Dana Scott, also made several comments on and correction to the actual text of this dissertation. I alone must be responsible for any errors which remain.

I would also like to thank Frank Pfenning for many discussions concerning the ideas in this dissertation. His comments and challenges often helped to shape much of the presentation of this material. I am also grateful to Richard Statman for offering several valuable comments and helpful insights.

Table of Contents

CHAPTER 1

Introduction and Summary

This dissertation is a presentation of various metatheoretical results about higher-order logic (HOL). Although many of these results should be of interest from a formal proof theory point-of-view, they were motivated by problems encountered in the construction of automatic theorem provers for this logic. (We will not explore such applications here.) The need to develop the metatheory for HOL is quite clear when one notes that techniques used by theorem provers in first order logic (FOL) cannot be naively extended to the higher-order setting. Such extensions have turned out to be very incomplete in some cases and unsound in other cases.

Analytic proofs. In order to extend automatic theorem proving techniques to the higher-order logic, the nature of cut-free, or using Smullyan's term, analytic (see [Smullyan68]) proofs in HOL must be understood. The theorem prover imbedded in the computer system TPS (see [Miller82]) is essentially a first-order theorem prover in which Huet's higher-order unification algorithm (see [Huet75]) is used to find unifying substitutions. This mix of techniques enables TPS to find a proof of Cantor's Theorem for sets (which is a genuinely higher-order theorem), but does not enable it to have even a chance of finding proofs for other theorems of HOL. This is because the structure of analytic proofs in HOL is complicated by the fact that higher-order substitution terms can include boolean connectives and quantifiers. A naive use of Huet's algorithm does not encompass this richer nature of HOL substitutions.

The completeness of most first-order theorem provers can generally be proved by showing that the theorem prover enumerates compound Herbrand instances of a proposed theorem in such a fashion that if a tautologous, compound instance exists, the prover will find one. Such a tautologous, compound instance can be thought of as a proof of the proposed theorem (see [Andrews81]). It is this representation of analytic proofs which seems most appropriate to extend to HOL. Our analogue to compound Herbrand instances will be called *expansion trees*. These trees will actually contain more information than compound instances since various nonterminal nodes of such trees will be labeled with the substitutions used in making the compound instances. Such an explicit structure is very useful when we later show how to convert expansion trees to other styles of proof, in particular, to proofs in natural deduction style. Expansion trees also have the pleasing property that they do not require the use of any normal form other than λ-normal form, and that they essentially are formulas with additional structure. Hence, they can often be manipulated much as formulas are. Expansion trees will use *selected variables* (also known in other settings as critical variables or eigenvariables) instead of skolem functions. An expansion tree will be considered a proof, called an ET-proof, if it encodes a tautologous formula (*i.e.* if nonterminal nodes which are labeled with substitutions are ignored) and if a certain relationship among the selected variables is acyclic. Chapter 2 concludes with a proof that ET-proofs are both sound and relatively complete for our system of HOL. These results can be considered to be a higher-order version of Herbrand's Theorem. The necessary proof-theoretical tools for establishing the completeness property of ET-proofs is the Abstract Consistency Property which Andrews defines in [Andrews71]. This is a generalization of Smullyan's Analytic Consistency Property [Smullyan68] to HOL and is a direct consequence of the cut-elimination theorem for HOL established independently by Takahashi in [Takahashi67] and Prawitz in [Prawitz68].

Skolemization. The naive extension of the skolemizations process to HOL is not sound. Andrews in [Andrews71] defined a system of resolution for HOL in which existential quantifiers were removed by using *choice parameters*. In this setting, these choice parameters could be used to assume the role usually played by skolem functions. This resolution system, however, turned out to be too strong since the negation of instances of the Axiom of Choice, which is known not to be derivable in our formulation of HOL, could be refuted. Hence, that form of resolution was not sound. Although Andrews was aware of how the skolemization process might be fixed, how to prove its soundness remained an open question. In Chapter 3 we will introduce a variant of expansion trees which uses skolem functions instead of selected variables. Those skolem expansion trees which encode tautologous formulas are called ST-proofs. An acyclic condition is not needed in ST-proofs since the nesting of skolem terms in other skolem terms provides an equivalent restriction on the proofs. It is necessary, however, to restrict the occurrences of skolem functions within the substitutions used in the skolem expansion trees in order to make their use sound. When applied in the first order setting, this restriction has no effect. We shall show that ST-proofs and ET-proofs are interconvertible, and this provides a direct, syntactic proof of the correctness of skolemization.

List representations. In Chapter 4 we define a list representation for expansion trees which will be more convenient to use, for both theoretical and practical concerns, and is more succinct than the tree structures. With these list representations, we are able to give straightforward, syntactic proofs of the independence of the axioms of extensionality, choice, and descriptions in our version of HOL. The proofs of these independence results are placed in Appendix 1. The reader is advised to look at this appendix for examples of ET-proofs.

Sequential and natural deductions proofs. Finally in Chapter 5, we deal with algorithms for converting ET-proofs into two more conventional and more readable proof formats, namely sequential calculus proof trees and natural deduction proofs. The first such algorithm considered will convert an ET-proof to both a cut-free sequential proof (in a calculus which is a slight extensions of Gentzen's LK-calculus) and a natural deduction proof. This is, therefore, a direct demonstration that Herbrand's theorem (our completeness result for ET-proofs) implies Gentzen's *Hauptsatz* for HOL. Apart from this theoretical fact, there are numerous other more practical concerns for investigating this transformation process. For example, once an automatic theorem prover has been successful in finding an ET-proof (or finding an ST-proof which can easily be converted to an ET-proof as outlined in Chapter 3), it should be possible for the theorem prover to convert that proof into a more readable explanation of the proof's structure. Also, this conversion should also be possible without any further search. The algorithm just mentioned can construct natural deduction proofs which generally qualify as being readable. This particular algorithm, however, will often produce rather inelegant proofs. Much research could be done in the area of finding those criteria which can be used to produce elegant proofs. One particular fact which makes this algorithm's proofs inelegant is that it does not know when it can backchain on an implicational fact that it has already established. Deciding this requires a certain amount of "look-ahead" on the algorithm's part. This look-ahead can be built into the algorithm by having it examine information which is available within the tautology encoded in the given ET-proof. This information is contained in a *clause-spanning mating* and is used to define the notion of a *focused* construction of a natural deduction proof. We give an improved algorithm which will build focused proofs. These proofs are generally quite readable and natural in many respects. This chapter is an extension of [Andrews80].

CHAPTER 2

Expansions Trees as Proofs

Section 2.1: The Logical System $\mathcal{T}$

Let $\mathcal{T}$ be the theory of HOL formulated by Church in [Church40] which uses only his axioms 1 through 6 (listed below). Formulas are built up from logical constants, variables, and parameters (non-logical constants) by lambda abstraction and function application. The first formulas of $\mathcal{T}$ we will consider contain only the logical constants $\sim_{oo}$, $\vee_{ooo}$, and $\Pi_{o(o\alpha)}$, where $[\forall x_\alpha A_o]$ is an abbreviation for $[\Pi_{o(o\alpha)} \lambda x_\alpha A_o]$, $A_o \wedge B_o$ is an abbreviation for $\sim.\sim A_o \vee \sim B_o$, and $A_o \supset B_o$ is an abbreviation for $\sim A_o \vee B_o$. This system is nonextensional. We shall generally not adorn formulas with type symbols, but rather, when the type of a formula, say A, cannot be determined from context, we will add the phrase "where A is a formula$_\alpha$" to imply that A has type α.

We shall freely use many of the definitions and results of §§ 2 and 3 of [Andrews71]. We now introduce some new definitions and state some simple theorems which the reader will already know.

2.1.1. Definition. If x is a variable$_\alpha$ and t is a formula$_\alpha$, we shall denote by $\underset{\cdot}{\mathbf{S}}^x_t A$ the formula which is the result of replacing all free occurrences of x in A with t. We shall assume that bound variable names are systematically changed to avoid variable capture. ▮

2.1.2. Definition. We say that formula$_\alpha$ B comes from formula$_\alpha$ A by $\lambda Rule1$ if B is the result of replacing a subformula of A of the form $[\lambda x C]$ with the subformula $[\lambda z \underset{\cdot}{\mathbf{S}}^x_z C]$, provided that z is a variable$_\beta$ which does not occur in C and x is a variable$_\beta$ which is not bound in C. We say that B comes from A by $\lambda Rule2$ if B is the result of replacing a subformula of A of the form $[\lambda x C]E$ with $\underset{\cdot}{\mathbf{S}}^x_E C$, provided that the bound variables of C are distinct from both the variable$_\beta$ x and from the free variables of the formula$_\beta$ E. A comes from B by $\lambda Rule3$ if B comes from A by λRule2.

We shall write A conv B (resp. A conv-*I*-*II* B) (resp. A conv-*I* B) if there is a sequence of applications of λRules 1, 2 and 3 (resp. 1 and 2) (resp. 1) which transforms A into B. ▮

Below we list the axioms and rules of inference for the logical calculus $\mathcal{T}$. First the axioms:

(1) $p \vee p \supset p$

(2) $p \supset p \vee q$

(3) $p \vee q \supset .q \vee p$

(4) $p \supset q \supset .r \vee p \supset .r \vee q$

(5) $\Pi_{o(o\alpha)} f_{o\alpha} \supset f_{o\alpha} x_\alpha$

(6) $\forall x_\alpha\ [p \vee f_{o\alpha} x_\alpha] \supset p \vee \Pi_{o(o\alpha)} f_{o\alpha}$

The rules of inference are the following:

(1) λRule1, 2, 3

(2) *Substitution:* From $F_{o\alpha} x_\alpha$ to infer $F_{o\alpha} A_\alpha$ provided that x_α is not a free variable of $F_{o\alpha}$.

(3) *Modus Ponens:* From $[A \supset B]$ and A to infer B.

(4) *Generalization:* From $F_{o\alpha} x_\alpha$ to infer $\Pi_{o(o\alpha)} F_{o\alpha}$, provided that x_α is not a free variable of $F_{o\alpha}$.

Those axioms and rules of inference which contain the type variable α are considered to be schema. We say that a formula$_o$, A, is a theorem of $\mathcal{T}$, written $\vdash_{\mathcal{T}} A$, if there is a list of formulas$_o$, $A_1, \ldots, A_n = A$ $(n \geq 1)$ such that for each i, $1 \leq i \leq n$, A_i is either an axiom or is derived from one or two previous formulas by a rule of inference.

2.1.3. Definition. A formula$_\alpha$ is in *ρ-normal* form if A is in λ-normal form and for each subformula $[\lambda x C]$ of A, x is the first variable in alphabetical order which is distinct from the other free variables of C. It is clear that for any formula$_\alpha$, B , there is a unique formula$_\alpha$ A in ρ-normal form such that A conv B. We shall write ρB to represent this formula. (ρ-normal form is identical to the η-normal form defined in [Andrews71]. We have changed its name here to avoid confusion later with η-convertibility, with which it has no relation.) As is noted in [Andrews71], ρ-normal formulas have the following properties: (a) If A is in ρ-normal form then every subformula of A is also in ρ-normal form, and (b) $\rho[A_{\alpha\beta} B_\beta] = [(\rho A_{\alpha\beta})(\rho B_\beta)]$. Neither of these properties will be used within this presentation, so, if the reader wishes, ρ-normal form can be taken to mean principal normal form, as in the sense used in [Church41]. ∎

2.1.4. Definition. Let A be a formula$_o$. An occurrence of a subformula B in A is a *boolean subformula occurrence* if it is in the scope of only $\sim$ and $\vee$, or if A is B. A boolean subformula occurrence is either positive or negative, depending on whether it is in the scope of an even or odd number of occurrences of $\sim$. A formula$_o$ A is an *atom* if its leftmost non-bracket symbol is a variable or a parameter. A formula B is a *boolean atom* (*b-atom*, for short) if its leftmost non-bracket symbol is a variable, parameter, or Π. A *signed atom* (*b-atom*) is a formula which is either an atom (b-atom) or the negation of an atom (b-atom). Two signed atoms, A_1 and A_2, are said to be *complementary* if either $\sim A_1$ conv-I A_2 or $\sim A_2$ conv-I A_1. ∎

2.1.5. **Substitutivity of Implication.** *Let A, B, and C be formulas$_o$, and assume that $\vdash_{\mathcal{T}} A \supset B$. Let D be the result of either replacing some positive boolean subformula occurrence of A in C with B, or some negative boolean subformula occurrence of B in C with A. If $\vdash_{\mathcal{T}} C$ then $\vdash_{\mathcal{T}} D$.*

2.1.6. Definition. A formula$_o$ is *tautologous* if it is an alphabetic variant of a substitution instance of a tautology. We shall use the statement "$A \equiv B$" to be the metalanguage assertion that $[A \supset B] \wedge [B \supset A]$ is tautologous, *i.e.* that A and B are truth functionally equivalent. ∎

2.1.7. **Theorem.** *If A is tautologous, then $\vdash_{\mathcal{T}} A$. Also, if x is variable$_\alpha$ which is not free in the formula$_{o\alpha}$ B, then $\vdash_{\mathcal{T}} [\Pi B \supset \forall x\, Bx] \wedge [\forall x\, Bx \supset \Pi B]$.*

Proof. See [Church40].

2.1.8. Definition. Let B be a boolean atom occurrence in the formula$_o$ A. If the leftmost non-bracket symbol of B is not a Π, then we say that B is *neutral*. Otherwise, we say that B is *existential* if it is in the scope of an odd number of negations and *universal* if it is in the scope of an even number of negations. We say that boolean atom occurrences come in these three *kinds*. ∎

Section 2.2: Abstract Derivability Property

The principal proof-theoretic tool we will use to establish the completeness result in Section 2.5 is called the abstract derivability property. This is essentially the dual notion to what Andrews in [Andrews71] calls the abstract consistency property, which is itself a generalization of Smullyan's analytic consistency property described in [Smullyan68]. Below we define both the abstract consistency and derivability properties.

2.2.9. Definition. A property Γ of finite sets of formulas$_o$ is an *abstract consistency property* if for all finite sets $\mathcal{S}$ of formulas, the following holds:

ACP1 If $\Gamma(\mathcal{S})$, then there is no atomic formula$_o$, A, such that $A \in \mathcal{S}$ and $\sim A \in \mathcal{S}$.

ACP2 If $\Gamma(\mathcal{S} \cup \{A\})$, then $\Gamma(\mathcal{S} \cup \{\rho A\})$.

ACP3 If $\Gamma(\mathcal{S} \cup \{\sim\sim A\})$, then $\Gamma(\mathcal{S} \cup \{A\})$.

ACP4 If $\Gamma(\mathcal{S} \cup \{A \vee B\})$, then $\Gamma(\mathcal{S} \cup \{A\})$ or $\Gamma(\mathcal{S} \cup \{B\})$

ACP5 If $\Gamma(\mathcal{S} \cup \{\sim.A \vee B\})$, then $\Gamma(\mathcal{S} \cup \{\sim A, \sim B\})$.

ACP6 If $\Gamma(\mathcal{S} \cup \{\Pi A\})$, then for each B, $\Gamma(\mathcal{S} \cup \{\Pi A, AB\})$.

ACP7 If $\Gamma(\mathcal{S} \cup \{\sim\Pi A\})$, then for any variable or parameter c which does not occur free in A or any formula in $\mathcal{S}$, $\Gamma(\mathcal{S} \cup \{\sim Ac\})$. ∎

If $\mathcal{S}$ is a finite set of formulas$_o$, then $\vee\mathcal{S}$ denotes the formula which is the disjunction of the members of $\mathcal{S}$ in some, undetermined order. Also, let $\sim\mathcal{S}$ be the set of the negations of formulas in $\mathcal{S}$. The important result concerning abstract consistency properties is the following theorem (see Theorem 3.5 in [Andrews71]).

2.2.10. **Theorem.** *If Γ is an abstract consistency property and $\mathcal{S}$ is a finite set of formulas$_o$ such that $\Gamma(\mathcal{S})$, then $\mathcal{S}$ is consistent, i.e. it is not the case that $\vdash_{\mathcal{T}} \vee \sim\mathcal{S}$.*

2.2.11. Definition. A property Λ of finite sets of formulas$_o$ is an *abstract derivability property* if for all finite sets $\mathcal{S}$ of formulas, the following holds:

ADP1 If there is an atomic formula$_o$, A, such that $A \in \mathcal{S}$ and $\sim A \in \mathcal{S}$, then $\Lambda(\mathcal{S})$.

ADP2 If $\Lambda(\mathcal{S} \cup \{\rho A\})$ then, $\Lambda(\mathcal{S} \cup \{A\})$.

ADP3 $\Lambda(\mathcal{S} \cup \{A\})$ if and only if $\Lambda(\mathcal{S} \cup \{\sim\sim A\})$.

ADP4 If $\Lambda(\mathcal{S} \cup \{\sim A\})$ and $\Lambda(\mathcal{S} \cup \{\sim B\})$, then $\Lambda(\mathcal{S} \cup \{\sim.A \vee B\})$.

ADP5 If $\Lambda(\mathcal{S} \cup \{A, B\})$, then $\Lambda(\mathcal{S} \cup \{A \vee B\})$.

ADP6 If $\Lambda(\mathcal{S} \cup \{\sim\Pi A, \sim AB\})$ for some B, then $\Lambda(\mathcal{S} \cup \{\sim\Pi A\})$.

ADP7 If for some variable or parameter c which does not occur free in A or any formula in $\mathcal{S}$, $\Lambda(\mathcal{S} \cup \{Ac\})$, then $\Lambda(\mathcal{S} \cup \{\Pi A\})$. ∎

It is easy to verify from the description of provability in $\mathcal{T}$ that the property $\Lambda(\mathcal{S})$ of finite sets $\mathcal{S}$ which asserts that $\vdash_{\mathcal{T}} \vee \mathcal{S}$, is an abstract derivability property. The reason for defining this second property, which is essentially the dual of the first, is that it is a positive statement about the nature of proof systems. Generally, abstract derivability properties, $\Lambda(\mathcal{S})$, are of the form "$\vee\mathcal{S}$ has a proof in system $\mathcal{X}$," for some proof system $\mathcal{X}$. In this way, each of the ADP conditions can be thought of as specifying some minimal properties of a proof system in order for it to be relatively complete. Notice that the proofs of Theorems 4.10 and 5.3 in [Andrews71], which are concerned with the completeness of a cut-free proof system and a resolution system resp., use essentially the contrapositive form of the abstract consistency property. The abstract derivability property permits a more direct approach to proving such completeness results.

2.2.12. **Lemma.** *Let Λ be an abstract derivability property. Define Γ to be the property of finite sets of formulas$_o$, $\mathcal{S}$, such that $\Gamma(\mathcal{S}) := \sim\Lambda(\sim\mathcal{S})$. Γ is an abstract consistency property.*

Proof. Let $\mathcal{S}$ be a finite set of formulas$_o$, and let A and B be any formulas$_o$. Below, we prove the contrapositive form of each of the abstract consistency property conditions.

(1) Assume that there is an atom A such that $A \in \mathcal{S}$ and $\sim A \in \mathcal{S}$. By ADP1, $\Lambda(\sim\mathcal{S} \cup \{A, \sim A\})$, and by ADP3, $\Lambda(\sim\mathcal{S} \cup \{\sim A, \sim\sim A\})$. But $\sim A \in \sim\mathcal{S}$ and $\sim\sim A \in \sim\mathcal{S}$. Hence, $\Lambda(\sim\mathcal{S})$. But this is the same as $\sim\Gamma(\mathcal{S})$.

(2) Assume $\sim\Gamma(\mathcal{S} \cup \{\rho A\})$. Then $\Lambda(\sim\mathcal{S} \cup \{\sim\rho A\})$ and $\Lambda(\sim\mathcal{S} \cup \{\rho\sim A\})$. By ADP2, we then have $\Lambda(\sim\mathcal{S} \cup \{\sim A\})$ and $\sim\Gamma(\mathcal{S} \cup \{A\})$.

(3) Assume $\sim\Gamma(\mathcal{S} \cup \{A\})$. Then $\Lambda(\sim\mathcal{S} \cup \{\sim A\})$ and by ADP3, $\Lambda(\sim\mathcal{S} \cup \{\sim\sim\sim A\})$. Hence, $\sim\Gamma(\mathcal{S} \cup \{\sim\sim A\})$.

(4) Assume $\sim\Gamma(\mathcal{S} \cup \{A\})$ and $\sim\Gamma(\mathcal{S} \cup \{B\})$. Hence, $\Lambda(\sim\mathcal{S} \cup \{\sim A\})$ and $\Lambda(\sim\mathcal{S} \cup \{\sim B\})$, and by ADP4, $\Lambda(\sim\mathcal{S} \cup \{\sim.A \vee \sim B\})$. Using the definition of Λ, we have $\sim\Gamma(\mathcal{S} \cup \{A \vee B\})$.

(5) Assume $\sim\Gamma(\mathcal{S} \cup \{\sim A, \sim B\})$. Then $\Lambda(\sim\mathcal{S} \cup \{\sim\sim A, \sim\sim B\})$ and $\Lambda(\sim\mathcal{S} \cup \{A, B\})$, by ADP3. By ADP5, we have $\Lambda(\sim\mathcal{S} \cup \{A \vee B\})$ and by ADP3, $\Lambda(\sim\mathcal{S} \cup \{\sim\sim.A \vee B\})$ which is $\sim\Gamma(\mathcal{S} \cup \{\sim.A \vee B\})$.

(6) Assume that there is a formula$_\alpha$ B such that $\sim\Gamma(\mathcal{S} \cup \{\Pi A, AB\})$. Then $\Lambda(\sim\mathcal{S} \cup \{\sim\Pi A, \sim AB\})$ and by ADP6, $\Lambda(\sim\mathcal{S} \cup \{\sim\Pi A\})$, which is $\sim\Gamma(\mathcal{S} \cup \{\Pi A\})$.

(7) Assume that for some parameter$_\alpha$ or variable$_\alpha$, c, which does not occur free in A or in any formula in $\mathcal{S}$, $\sim\Gamma(\mathcal{S} \cup \{\sim Ac\})$, where A is a formula$_{o\alpha}$. Then $\Lambda(\sim\mathcal{S} \cup \{\sim\sim Ac\})$. By ADP3, $\Lambda(\sim\mathcal{S} \cup \{Ac\})$. By ADP7, $\Lambda(\sim\mathcal{S} \cup \{\Pi A\})$ or $\Lambda(\sim\mathcal{S} \cup \{\sim\sim\Pi A\})$ and finally, $\sim\Gamma(\mathcal{S} \cup \{\sim\Pi A\})$. **Q.E.D.**

2.2.13. **Relative Completeness Theorem for Abstract Derivability Properties.** *Let Λ be an abstract derivability property. Whenever $\mathcal{S}$ is a finite set of formulas$_o$ such that $\vdash_{\mathcal{T}} \vee\mathcal{S}$, then $\Lambda(\mathcal{S})$.*

The reason (and need) for using the term *relative completeness* instead of completeness is explained at the end of Section 2.3.

Proof. Define $\Gamma(\mathcal{S})$ to be $\sim\Lambda(\sim\mathcal{S})$. By Lemma 2.12, we know that Γ is an abstract consistency property. Now assume that $\sim\Lambda(\mathcal{S})$ for a finite set $\mathcal{S}$ of formulas$_o$. Then $\sim\Lambda(\sim\sim\mathcal{S})$ by ADP3. Then $\Gamma(\sim\mathcal{S})$, and by Theorem 2.10, $\sim\mathcal{S}$ is consistent, *i.e.* it is not true that $\vdash_{\mathcal{T}} \vee\sim\sim\mathcal{S}$ or $\vdash_{\mathcal{T}} \vee\mathcal{S}$. **Q.E.D.**

The definition of abstract derivability is not actually dual to abstract consistency, mainly since abstract derivability permits stronger manipulation of double negations. With a dual and, hence weaker form of abstract derivability, this theorem would only offer the final conclusion that $\vdash_{\mathcal{T}} A$ implies $\Lambda(\{\sim\sim A\})$. Since most useful abstract derivability properties treat double negations in the stronger sense, we have constructed our definition accordingly.

Section 2.3: Expansion Tree Proofs

We shall now define our generalization of compound Herbrand expansions, by defining expansion trees and ET-proofs. All references to trees will actually refer to finite, ordered trees in which the nodes and arcs may or may not be labeled, and where labels, if present, are formulas. In particular, nodes may be labeled with the formulas which are just the logical connectives $\sim$ and $\vee$. We shall picture our trees with their roots at the top and their leaves (terminal nodes) at the bottom. In this setting, we say that one-node *dominates* another node if it they are on a common branch and the first node is *higher* in the tree than the second. This dominance relation shall be considered reflexive. All nodes except the root node will have *in-arcs* while all nodes except the leaves will have *out-arcs*. A node labeled with $\sim$ will always have one out-arc, while a node labeled with $\vee$ will always have two out-arcs. We shall also say that an arc dominates a node if the node which terminates the arc dominates the given node. In particular, an arc dominates the node in which it terminates.

2.3.14. Definition. Formulas$_o$ of $\mathcal{T}$ can be considered as trees in which the nonterminal nodes are labeled with $\sim$ or $\vee$, and the terminal nodes are labeled with b-atoms. Given a formula$_o$, A, we shall refer to this tree as the *tree representation of* A. ▮

2.3.15. Example. Below is the tree representation of $\sim[\Pi B \vee Ax] \vee \sim\sim\Pi[\lambda x.Ax \vee Bx]$.

Figure 2.1:

▮

We shall adopt the following linear fashion of representing trees. If the root of the tree Q is labeled with $\sim$ we write $Q = \sim Q'$, where Q' is the subtree dominated by Q's root. Likewise, if the root of Q has label $\vee$, with left subtree Q' and right subtree Q'', then we write $Q = Q' \vee Q''$. The expression $Q' \wedge Q''$ is an abbreviation for the tree $\sim[\sim Q' \vee \sim Q'']$.

2.3.16. Definition. Let Q be a tree, and let N be a node in Q. We say that N occurs positively (negatively) if the path from the root of Q to N contains an even (odd) number of nodes labeled with $\sim$. In particular, the root of Q occurs positively in Q. If a node N in Q is labeled with a formula of the form ΠB, then we say that N is *universal* (*existential*) if it occurs positively (negatively) in Q. A universal (existential) node which is not dominated by any other universal or existential node is called a *top-level universal (existential) node*. A labeled arc is a *top-level labeled arc* if it is not dominated by any other labeled arc. ▮

2.3.17. Definition. Let Q, Q' be two trees. Let N be a node in Q and let l be a label. We shall denote by $Q +^l_N Q'$ the tree which results from adding to N an arc, labeled l, which joins N to the root of the tree Q'. This new arc on N comes after the other arcs from N (if there are any). In the case that the tree Q is a one-node tree, N must be the root of Q. In this case, we write $A +^l Q'$ instead of $Q +^l_N Q'$, where A is the formula which labels N. ▮

Figure 2.2: Figure showing the three trees Q, Q', and $Q +^c_N Q'$.

2.3.18. Example. Below we have three trees, Q, Q' and $Q +^c_N Q'$, where N is a node of Q and c is some label. The nodes and labels of Q may or may not have their own labels.

∎

2.3.19. Definition. Let Q be a tree with a terminal node N labeled with the formula ΠB, for some formula$_{o\alpha}$ B. If N is existential, then an *expansion* of Q at N with respect to the list of formulas$_\alpha$, $\langle t_1, \dots, t_n\rangle$, is the tree $Q +^{t_1}_N Q_1 +^{t_2}_N \cdots +^{t_n}_N Q_n$ (associating to the left), where for $1liln$, Q_i is the tree representation for some λ-normal form of Bt_i. If N is universal, then a *selection* of Q at N with respect to the variable$_\alpha$ y, is the tree $Q +^y_N Q'$, where Q' is the tree representation of some λ-normal form of By, and y does not label an out-arc of any universal node in Q.

The set of all *expansion trees* is the smallest set of trees which contains the tree representations of all λ-normal formulas$_o$ and which is closed under expansions and selections. ∎

2.3.20. Definition. Let Q be an expansion tree for A. A *derivation list* for Q is a list $\langle Q_1, \dots, Q_n\rangle$, $n \geq 1$, such that Q_1 is the tree representation of some λ-normal form of A, $Q_n = Q$, and for $i = 1, \dots, n-1$, Q_{i+1} is either an expansion or selection of Q_i. Notice that all the trees in this list are expansion trees, and that a tree is an expansion tree if and only if it has a derivation list. ∎

2.3.21. Definition. Assume that Q is an expansion tree. Let $\mathbf{S}_Q$ be the set of all *selected variables* of Q, *i.e.* $\mathbf{S}_Q$ is the set of all variables which label the out-arcs from (nonterminal) universal nodes in Q. A node N of Q is said to be *selected by* $y \in \mathbf{S}_Q$ if N is a universal node of Q and y labels the (unique) out-arc of N. Let Θ_Q be the set of *occurrences of expansion terms* in Q, *i.e.* Θ_Q is the set of all formulas which label out-arcs of (nonterminal) existential nodes of Q. A node N in Q is said to be result of an expansion by $t \in \Theta_Q$ if the in-arc of N is labeled with this occurrence of t. Alternatively, we could think of Θ_Q as a set of arcs instead of occurrences of expansion term occurrences. All labeled arcs of the expansion tree Q are represented by a member in either $\mathbf{S}_Q$ or Θ_Q. Notice that the same node can be selected on and also be the result of an expansion. ∎

The expansion trees we are considering in this section will not use skolem functions as is generally the case with Herbrand instances. It turns out that the way skolem terms imbed themselves in other skolem terms actually places a restriction on the selected variables which are used to stand in the place of skolem terms. In order to do without skolem functions, we need to place a restriction on the selected variables of an expansion tree. There are two equivalent such restrictions, each requiring that a certain relation, one defined on $\mathbf{S}_Q$ and the other on Θ_Q, be acyclic.

2.3.22. Definition. Let Q be an expansion tree. Let $\prec^0_Q$ be the binary relation on $\mathbf{S}_Q$ such that $z \prec^0_Q y$ if there exists a $t \in \Theta_Q$ such that z is free in t and a node dominated by (the arc labeled with) t is selected by y. $\prec_Q$, the transitive closure of $\prec^0_Q$, is called the *imbedding relation* and plays an important part in the analysis of skolemization in the next chapter. ∎

2.3.23. Definition. Let Q be an expansion tree. Let $<^0_Q$ be the binary relation on Θ_Q such that $t <^0_Q s$ if there exists a variable which is selected for a node dominated by t and which is free in s. $<_Q$, the transitive closure of $<^0_Q$, is called the *dependency relation* for Q and plays an important part in the soundness proof in the next section. ∎

2.3.24. **Proposition.** *$<_Q$ is acyclic if and only if $\prec_Q$ is acyclic.*

Proof. Let $<_Q$ be cyclic. That is, assume that there are expansion term occurrences $t_1, \ldots, t_m \in \Theta_Q$ such that $t_1 <^0_Q \cdots <^0_Q t_m <^0_Q t_{m+1} = t_1$ for $m \geq 1$ (see figure below). Let y_i, for $i = 1, \ldots, m$, be chosen from $\mathbf{S}_Q$ so that y_i is selected for a node dominated by t_i and y_i is free in t_{i+1}. If we identify y_{m+1} with y_1, then we have $y_i \prec^0_Q y_{i+1}$, for $i = 1, \ldots, m$, since y_{i+1} is selected for a node dominated by t_{i+1} and y_i is free in the formula t_{i+1}. Hence, $y_1 \prec^0_Q \cdots \prec^0_Q y_m \prec^0_Q y_1$, and $\prec_Q$ is cyclic.

Figure 2.3: Figure showing the relationship among various nodes and arcs within Q.

The proof in the other direction is very similar and is omitted. **Q.E.D.**

An expansion tree represents two formulas in its structure. The formula $Fm(Q)$, defined below, is the "deep" representation of the expansion tree Q since it is composed of the b-atoms which are the leaves of Q. The formula $Sh(Q)$ is the "shallow" representation of Q since it is composed of b-atoms which label nodes in Q that are not dominated by any other existential or universal node.

2.3.25. Definition. Let Q be a tree such that either Q or $\sim Q$ is an expansion tree. We define $Fm(Q)$ by induction on the structure of Q.

(1) If Q is a one-node tree, then $Fm(Q) := A$, where A is the formula which labels that one-node.

(2) If $Q = \sim Q'$ then $Fm(Q) := \sim Fm(Q')$.

(3) If $Q = Q' \vee Q''$ then $Fm(Q) := Fm(Q') \vee Fm(Q'')$.

(4) If $Q = \Pi B +^{l_1} Q_1 + \ldots +^{l_n} Q_n$ then $Fm(Q) := Fm(Q_1) \wedge \ldots \wedge Fm(Q_n)$. ∎

Notice, that if A is a formula$_o$, and Q is the tree representation of A, then $Fm(Q) = A$.

2.3.26. Definition. Let Q be a tree such that either Q or $\sim Q$ is an expansion tree. We define $Sh(Q)$ by induction on the top-level boolean structure of Q.

(1) If Q is a one-node tree, whose sole node is labeled with the formula$_o$ A, then $Sh(Q) := A$.

(2) If $Q = \sim Q'$ then $Sh(Q) := \sim Sh(Q')$.

(3) If $Q = Q' \vee Q''$ then $Sh(Q) := Sh(Q') \vee Sh(Q'')$.

(4) If $Q = \Pi B +^{l_1} Q_1 + \ldots +^{l_n} Q_n$ then $Sh(Q) := \Pi B$. ∎

2.3.27. Definition. Let Q be a tree, x a variable$_\alpha$, and t a formula$_\alpha$. We define $\mathbf{S}^x_t Q$ to be the tree which results from replacing all free occurrences of x with t in all formulas which label nodes and arcs, and then placing all formula labels in λ-normal form. If a leaf is left which is labeled with a formula$_o$ which is a top-level $\sim$ or $\vee$, then the node is replaced with the tree representation of this label. We assume that changes in bound variables are made in some systematic fashion to avoid variable capture.

A variable is *new to* Q if it is not free in any formula which is a label in Q. ∎

2.3.28. Definition. An expansion tree Q is an expansion tree *for* A if $Sh(Q)$ is a λ-normal form of A and no variable in $\mathbf{S}_Q$ is free in A. An *ET-proof* for a formula$_o$, A, is an expansion tree Q for A such that $Fm(Q)$ is tautologous and $<_Q$ is acyclic. ∎

2.3.29. Example. Let A be the theorem $\exists y\, \forall x\, .Px \supset Py$. An ET-proof for A would then be the tree Q given as:

$$\begin{aligned}\sim[[\Pi\lambda y.\sim\Pi\lambda x.\sim Px \vee Py] +^u \sim[[\Pi\lambda x.\sim Px \vee Pu] +^v [\sim Pv \vee Pu]]\\ +^v \sim[[\Pi\lambda x.\sim Px \vee Pv] +^w [\sim Pw \vee Pv]]].\end{aligned}$$

Here,

$$\begin{aligned}&Fm(Q) = \sim[\sim[\sim Pv \vee Pu] \wedge \sim[\sim Pw \vee Pv]],\\ &\Theta_Q = \{u, v\}, \text{ and}\\ &\mathbf{S}_Q = \{v, w\}\,.\end{aligned}$$

The dependency relation is given by the pair $u <_Q v$, while the imbedding relation is given by the pair $v \prec_Q w$. Notice, that if we had used u instead of w, $<_Q$ and $\prec_Q$ would have been cyclic. ∎

2.3.30. Definition. An expansion tree is *grounded* if none of its terminal nodes are labeled with formulas of the form ΠB. An ET-proof is a grounded ET-proof if it is also a grounded expansion tree. ∎

2.3.31. **Proposition.** *Let Q be an expansion tree, $x \in \mathbf{S}_Q$, and let y be new to Q.*

(1) *$Q' := \mathbf{S}^x_{.y} Q$ is an expansion tree.*

(2) *If $<_Q$ is acyclic, so is $<_{Q'}$.*

(3) *$Fm(Q') = [\mathbf{S}^x_{.y} Fm(Q)]$.*

(4) *If Q is an expansion tree for A, and y is not free in A, then Q' is an expansion tree for A.*

(5) *If Q is an ET-proof for A, and y is not free in A, then Q' is an ET-proof for A.*

Proof. We first verify (1). Let $\langle Q_1, \ldots, Q_n \rangle$ be a derivation list for Q. We will show that $\langle \mathbf{S}^x_{.y} Q_1, \ldots, \mathbf{S}^x_{.y} Q_n \rangle$ is a derivation list for $\mathbf{S}^x_{.y} Q$. Clearly, since Q_1 is a tree representation of A, $\mathbf{S}^x_{.y} Q_1$ is a tree representation of $\mathbf{S}^x_{.y} A$. We now proceed by induction on i, for $i = 1, \ldots, n-1$. Assume that $\mathbf{S}^x_{.y} Q_i$ is an expansion tree. We consider two cases:

(a) Q_{i+1} is a selection on Q_i, *i.e.* $Q_{i+1} = Q_i +^z_N Q'$ for some terminal, universal node N of Q_i labeled with a formula ΠB for some formula$_{o\alpha}$, B, and for some variable$_\alpha$, z, and where Q' is the tree representation of some λ-normal form of Bz. Let $z' := \mathbf{S}^x_{.y} z$. z' is either z or y. Clearly, $\mathbf{S}^x_{.y} Q_{i+1} = [\mathbf{S}^x_{.y} Q_i] +^{z'}_{N'} [\mathbf{S}^x_{.y} Q']$, where N' in $\mathbf{S}^x_{.y} Q_i$ corresponds to N in Q_i. Now, N' in $\mathbf{S}^x_{.y} Q_i$ is labeled with $\Pi[\mathbf{S}^x_{.y} B]$ while $\mathbf{S}^x_{.y} Q'$ is the tree representation of some λ-normal form of $[\mathbf{S}^x_{.y} B]z' = \mathbf{S}^x_{.y}[Bz]$. Hence, since $\mathbf{S}^x_{.y} Q_{i+1}$ is a selection on $\mathbf{S}^x_{.y} Q_i$, and since the latter tree is an expansion tree, $\mathbf{S}^x_{.y} Q_{i+1}$ is too.

(b) Q_{i+1} is an expansion of Q_i, *i.e.* $Q_{i+1} = Q_i +^{t_1}_N P_1 + \ldots +^{t_m}_N P_m$ where N is a terminal, existential node of of Q_i labeled with ΠB for some formula$_{o\alpha}$, B, and for $j = 1, \ldots, m$, t_j is a formula$_\alpha$ and P_j is the tree representation of some λ-normal form of Bt_j. Now, $\underset{\cdot}{\mathbf{S}}{}^x_y Q_{i+1} = \underset{\cdot}{\mathbf{S}}{}^x_y Q_i +^{t'_1}_N \underset{\cdot}{\mathbf{S}}{}^x_y P_1 \ldots +^{t'_m}_N \underset{\cdot}{\mathbf{S}}{}^x_y P_m$, where $t'_j := \underset{\cdot}{\mathbf{S}}{}^x_y t_j$ for $j = 1, \ldots, m$. Once again, $\underset{\cdot}{\mathbf{S}}{}^x_y P_j$ is a tree representation for some λ-normal form of $[\underset{\cdot}{\mathbf{S}}{}^x_y B]t'_j = \underset{\cdot}{\mathbf{S}}{}^x_y [Bt_j]$. Hence, $\underset{\cdot}{\mathbf{S}}{}^x_y Q_{i+1}$ is an expansion of $\underset{\cdot}{\mathbf{S}}{}^x_y Q_i$ and is, therefore, an expansion tree itself.

To verify (2), assume that $t' <^0_{Q'} s'$, for two expansion term occurrences $t', s' \in \Theta_{Q'}$. Let $t, s \in \Theta_Q$ be the corresponding expansion term occurrences in Q. Hence, there is a variable $z \in \mathbf{S}_{Q'}$ such that the node, say H' of Q' which is selected by z is dominated by t', and z is free in s'. If $z \neq y$, then $t <^0_Q s$. If $z = y$ then since y is new to Q, H (corresponding to H') is selected by x and x is free in t. Again, $t <^0_Q s$. Thus, if $<_{Q'}$ contained a cyclic, then so would $<_Q$.

(3) and (4) follow trivially. (5) follows immediately from all the preceding cases. **Q.E.D.**

2.3.32. **Corollary.** *Let $\mathcal{B}$ be a finite set of variables. If A has an ET-proof, then it has an ET-proof in which no selected variable is a member of $\mathcal{B}$.*

Proof. Let Q be an ET-proof for A. We proceed by induction of the cardinality of the set $\mathcal{B} \cap \mathbf{S}_Q$. If this set is empty, we are finished. Otherwise, pick $x \in \mathcal{B} \cap \mathbf{S}_Q$ and let $y \notin \mathcal{B}$ be a variable which is not free in A and which is new to Q. By Proposition 2.31 (5), $Q' := \underset{\cdot}{\mathbf{S}}{}^x_y Q$ is an ET-proof for A. Since, $x \notin \mathbf{S}_{Q'}$ and $\mathbf{S}_{Q'} \subset \mathbf{S}_Q$, the inductive hypothesis finishes our proof. **Q.E.D.**

2.3.33. **Proposition.** *If Q is an expansion tree for A and if A' is a positively occurring boolean subformula in A, then the corresponding subtree Q' in Q is an expansion tree for A'.*

Proof. Let $\langle Q_1, \ldots, Q_m \rangle$, be a derivation sequence for Q, and let Q'_i, for $1lilm$, be the subtree of Q_i whose relative position in Q_i corresponds to the position of A' in A. The sequence $Q'_1, \ldots, Q'_m$ is such that Q'_1 is the tree representation of some λ-normal form of A', and either Q'_{i+1} is equal to Q'_i or it comes from Q'_i by an expansion or a selection. Hence, $Q' = Q'_m$ is an expansion tree for A'. **Q.E.D.**

We now must justify calling certain expansion trees proofs, *i.e.* we must show that a formula of $\mathcal{T}$ is a theorem if and only if that formula has an ET-proof. Since $\mathcal{T}$ is nonextensional (see Section 5.2), Henkin-style frame semantics (see [Henkin50]) will not be strong enough to formulate an adequate definition of validity since Henkin-sytle models are always extensional (see [Andrews72b]). Thus we are not able to prove the strong forms of completeness, *i.e.* that if a sentence is valid it has an ET-proof. Hence, we shall prove the weaker form of this metatheorem relative to provability in $\mathcal{T}$. In the next section we shall prove soundness of ET-proofs, and in the last section, their relative completeness.

Section 2.4: Soundness for ET-Proofs

In this section we shall show that if a formula$_o$ A has an ET-proof then $\vdash_{\mathcal{T}} A$. In order to motivate the few definitions and lemmas which we will need to prove this result, we briefly outline the soundness proof which concludes this section.

Given an ET-proof Q for a formula$_o$ we shall construct a sequence of expansion trees $Q = Q_1, \ldots, Q_m$ by eliminating "top-level" expansion terms or selected variables, so that the last tree Q_m contains no labeled arcs. The process of eliminating a labeled arc from Q_i to get Q_{i+1} is essentially a substitution of a formula into $Sh(Q_i)$ to get $Sh(Q_{i+1})$. Looked at in reverse, $Sh(Q_i)$ will be either a universal or existential generalization of $Sh(Q_{i+1})$. In order to actually insure that this is the case we must be careful when eliminating arcs from existential nodes. The terms introduced in this fashion cannot introduce into the shallow formula any variables which are still selected in the expansion tree (trees satisfying this property are said to be *sound*).

These terms are called *admissible.* It is the acyclic nature of $<_Q$ which guarantees that we can require that expansion terms be eliminated only when they are admissible and that we can still manage to eliminate all labeled arcs. Finally, since $Fm(Q_m) = Sh(Q_m)$ and this formula is tautologous, $\vdash_{\mathcal{T}} Sh(Q_m)$. By application of universal and existential generalization, $\vdash_{\mathcal{T}} Sh(Q_1)$. But A conv-*I*-*II* $Sh(Q_1)$, so $\vdash_{\mathcal{T}} A$ by λ-conversion.

2.4.34. Definition. A node N in an expansion tree Q is *instantiated* if it is a nonterminal, universal or existential node. A term t is *admissible* in Q if no variable free in t is contained in $\mathbf{S}_Q$. Q is *sound* if no variable in $\mathbf{S}_Q$ is free in $Sh(Q)$. We can *eliminate a top-level labeled arc*, *i.e.* a selected variable or an expansion term, in one of the following ways:

(1) If N is a top-level, instantiated, universal node, then it is the root of a subtree of Q of the form $\Pi B +^{y} Q'$, where B is a formula$_{o\alpha}$ and y is a selected variable$_\alpha$. The tree which results by replacing this subtree by Q' is called the *result of eliminating* y *from* Q. In the resulting tree, y is no longer a selected variable.

(2) If N is a top-level, instantiated, existential node, then it is the root of a subtree $Q_0 := \Pi B +^{t_1} Q_1 + \ldots +^{t_n} Q_n$ where $n \geq 1$, B is a formula$_{o\alpha}$ and $t_1, \ldots, t_n$ are expansion terms$_\alpha$ of Q. If $n = 1$ and t_1 is admissible in Q, then let Q' be the result of replacing Q_0 with Q_1. If $n > 1$ and for some i, $1 l i l n$, t_i is admissible in Q, then let Q' be the result of replacing Q_0 with the tree

$$[\Pi B +^{t_1} Q_1 + \ldots +^{t_{i-1}} Q_{i-1} +^{t_{i+1}} Q_{i+1} + \ldots +^{t_n} Q_n] \wedge Q_i.$$

If in the first case, we set $i := 1$, then in either case, Q' is called the *result of eliminating* t_i *from* Q. Notice, that t_i does not correspond to an occurrence of an expansion term in the resulting tree. ∎

Sound expansion trees are those trees which are expansion trees *for* some formula. In particular, if Q is a sound expansion tree, then Q is an expansion tree for $Sh(Q)$.

2.4.35. **Lemma.** *If Q' is the result of eliminating a labeled arc from the expansion tree Q then*

(1) Q' has fewer labeled arcs than Q,

(2) if $<_Q$ is acyclic, then so is $<_{Q'}$,

(3) $Fm(Q')$ is truth-functionally equivalent to $Fm(Q)$,

(4) if Q is sound then so is Q', and

(5) if Q is sound and $\vdash_{\mathcal{T}} Sh(Q')$ then $\vdash_{\mathcal{T}} Sh(Q)$.

Proof. Part (1) is immediate. Notice that an expansion tree has an instantiated node if and only if it has a labeled arc.

Since only top-level expansion terms can be eliminated, it is easy to verify that there is a natural imbedding of $<_{Q'}$ into $<_Q$. Hence, $<_{Q'}$ is acyclic and we have part (2).

Since $Fm(\Pi B +^{y} Q') = Fm(Q')$ and

$$\begin{aligned} Fm(\Pi B +^{t_1} Q_1 + \ldots +^{t_n} Q_n) &= Fm(Q_1) \wedge \ldots \wedge Fm(Q_n) \\ &\equiv Fm([\Pi B +^{t_1} Q_1 + \ldots +^{t_{i-1}} Q_{i-1} \\ &\qquad +^{t_{i+1}} Q_{i+1} + \ldots +^{t_n} Q_n] \wedge Q_i) \end{aligned}$$

we obtain (3) by substitutivity of equivalence.

If Q is sound and Q' arises by eliminating a selected variable $y \in \mathbf{S}$, then Q' must also be sound, since the selected variable y, which may now be free in $Sh(Q')$, is not selected in Q', since otherwise y would be selected twice in Q. Let Q be sound and let Q' arise by eliminating an expansion term $t \in \Theta_Q$ from Q. Here, t is admissible in Q. $Sh(Q')$ can be formed by replacing the existential b-atom ΠB with a λ-normal form of either $\Pi B \wedge Bt$ or of Bt. Assume that Q' is not sound. Then there must be some $z \in \mathbf{S}_{Q'} = \mathbf{S}_Q$ which is free in $Sh(Q')$. Hence, z is either free in $Sh(Q)$ or in Bt. Since Q is sound and $z \in \mathbf{S}_Q$, z is not free in $Sh(Q)$ or in B. Hence, z must be free in t. But this contradicts the fact that t was admissible in Q. Hence, Q' is sound, and we have part (4).

Let Q be sound. By (4), Q' is sound. Also assume that $\vdash_{\mathcal{T}} Sh(Q')$. Now assume that Q' is the result of eliminating the top-level, selected variable $y \in \mathbf{S}_Q$. By universal generalization, $\vdash_{\mathcal{T}} \forall y\, Sh(Q')$. Since Q is sound and since $Sh(Q')$ can be formed by replacing the universal b-atom ΠB in $Sh(Q)$ with some λ-normal form of By, if y is free in $Sh(Q')$, then it is free only within this boolean subformula occurrence. Hence, by using the substitutivity of implication in the positive form with the implications

$$\vdash_{\mathcal{T}} \forall y\, [R \vee Sy] \supset .R \vee \forall y\, Sy \quad \text{and} \quad \vdash_{\mathcal{T}} [\forall y \sim Sy] \supset .{\sim}\exists y\, Sy$$

and substitutivity of implication in the negative form with the implications

$$\vdash_{\mathcal{T}} [R \vee \exists y\, Sy] \supset .\exists y\, [R \vee Sy] \quad \text{and} \quad \vdash_{\mathcal{T}} [{\sim}\forall y\, Sy] \supset .\exists y \sim Sy$$

we can push the quantifier on $\forall y\, Sh(Q')$ in until we obtain $Sh(Q)$, by a sequence of implications. Hence, $\vdash_{\mathcal{T}} \forall y\, Sh(Q') \supset Sh(Q)$ and by *modus ponens*, we finally obtain $\vdash_{\mathcal{T}} Sh(Q)$.

Now assume that Q' is the result of elimination a top-level expansion term $t \in \Theta_Q$. $Sh(Q')$ can be formed by replacing an existential b-atom of the form ΠB with some λ-normal form of Bt or $\Pi B \wedge Bt$. Since

$$\vdash_{\mathcal{T}} \Pi B \supset Bt \quad \text{and} \quad \vdash_{\mathcal{T}} \Pi B \supset .\Pi B \wedge Bt$$

we have $\vdash_{\mathcal{T}} Sh(Q') \supset Sh(Q)$ by using the negative form of the substitutivity of implication. By *modus ponens*, we then have $\vdash_{\mathcal{T}} Sh(Q)$. This concludes the proof of (5). **Q.E.D.**

2.4.36. **Lemma.** *If the expansion tree Q has a labeled arc and $<_Q$ is acyclic, then some top-level labeled arc can be eliminated.*

Proof. If Q has a top-level selected variable, then this arc can be eliminated. Assume that Q has no top-level instantiated universal nodes. Let $t_1, \dots, t_m$ be the list of all the top-level expansion terms of Q. Assume that none of these expansion terms can be eliminated since they are all inadmissible in Q. Let i be an arbitrary integer such that $1lilm$. Since t_i is inadmissible in Q, then there is a variable $y \in \mathbf{S}_Q$ such that y is free in t_i. Since Q has no top-level selected variables, y must label an arc which is dominated by t_j for some j such that $1ljlm$. Hence, $t_j <_Q^0 t_i$. Since t_i was chosen arbitrarily from the list $t_1, \dots, t_m$, each of these term occurrences has an $<_Q^0$-descendant in this list. But this is possible only if $<_Q$ has a cycle, which is a contradiction. Hence, we must be able to eliminate one of the expansion terms $t_1, \dots, t_m$. **Q.E.D.**

2.4.37. **Soundness Theorem for ET-Proofs.** *If the formula A has an ET-proof, then $\vdash_{\mathcal{T}} A$.*

Notice that since $\mathcal{T}$ is sound, if $\vdash_{\mathcal{T}} A$, then A is satisfied by all Henkin-style models of $\mathcal{T}$.

Proof. Let Q be an ET-proof for A. We can now construct a list of expansion trees, $Q_1, \dots, Q_m$, such that $Q_1 := Q$, and for $1li < m$, Q_{i+1} is the result of eliminating a top-level, labeled arc from Q_i. Since $<_{Q_1}$ is acyclic, by Lemma 2.35 (2) and Lemma 2.36, we know that such a construction is possible. Lemma 2.35 (1) guarantees that this construction can be made to terminate so that Q_m has no labeled arcs. Since $Q = Q_1$ is an expansion tree for A, Q_1 is sound. By Lemma 2.35 (4), all the trees Q_i are sound, for i such that $1lilm$. By Lemma 2.35 (3), we know that $Fm(Q_m)$, being truth-functionally equivalent to $Fm(Q_1)$, is tautologous. Since Q_m contains no labeled arcs, $Sh(Q_m) = Fm(Q_m)$ and $Sh(Q_m)$ is tautologous. Hence $\vdash_{\mathcal{T}} Sh(Q_m)$, and by Lemma 2.35 (5), $\vdash_{\mathcal{T}} Sh(Q_1)$. Now, Q_1 is an expansion tree for A, $Sh(Q)$ conv-*I-II* A, and $\vdash_{\mathcal{T}} A$ by λ-convertibility. **Q.E.D.**

Section 2.5: Relative Completeness for ET-Proofs

Before we jump into the completeness proof for ET-proofs, we prove the following useful lemma.

2.5.38. **Lemma.** *Let A be a formula$_o$ which has a boolean level, universal subformula occurrence, ΠB, for some formula$_{o\alpha}$ B. Let y be some variable$_\alpha$ which is not free in A and let A' be the result of replacing ΠB with By. A' has a grounded ET-proof if and only if A has a grounded ET-proof.*

Proof. Let Q' be a grounded ET-proof for A'. By Proposition 2.32, we may assume that $y \notin \mathbf{S}_{Q'}$. Then by Proposition 2.33, Q' has a subtree, Q'', which is an expansion tree for By. Let Q be the result of replacing Q'' in Q' with $\Pi C +^y Q''$, where C is a λ-normal form of B. Clearly, Q is a grounded expansion tree for A. Since $<_Q$ and $<_{Q'}$ are isomorphic, and $Fm(Q) = Fm(Q')$, Q is a grounded ET-proof of A.

Let Q be a grounded ET-proof for A. Again, we can assume that $y \notin \mathbf{S}_Q$. By Proposition 2.33 and the fact that Q is grounded, Q has a subtree of the form $\Pi C +^z Q''$, where C is a λ-normal form of B, which is an expansion tree for ΠB. Then, let $Q''' := \dot{\mathbf{S}}^z_y Q$, which has a subtree $\Pi C +^y [\dot{\mathbf{S}}^z_y Q'']$, which is also an expansion tree for ΠB. Let Q' be the result of replacing this last subtree in Q''' with $\dot{\mathbf{S}}^z_y Q''$. Q' is a grounded expansion tree for A'. Since $<_Q$ and $<_{Q'}$ are isomorphic,$<_{Q'}$ is acyclic. Now $Fm(Q') = [\dot{\mathbf{S}}^z_y Fm(Q)]$, and $Fm(Q')$ is tautologous. In other words, Q' is a grounded ET-proof of A'. **Q.E.D.**

The following lemma, along with Theorem 2.13, is required to prove the relative completeness result (Theorem 2.42). Its proof is the most involved one presented to this point.

2.5.39. **Theorem.** *Let $\Lambda(\mathcal{S})$ be the property about finite sets of formulas$_o$ which asserts that $\vee\mathcal{S}$ has a grounded ET-proof. Λ is an abstract derivability property.*

Proof. First, we must show that $\Lambda(\mathcal{S})$ is well-defined, *i.e.* is not dependent on the order in which the disjunction $\vee\mathcal{S}$ is formed. This is immediate since, if some disjunction of $\mathcal{S}$ has an ET-proof, Q, then any rearrangement of that disjunction has an ET-proof which is the corresponding rearrangement of Q.

In the lines below, let $S := \vee\mathcal{S}$, where the disjunction is taken in any order. If $\mathcal{S}$ is empty, we take S to be the empty disjunction, and we identify $S \vee A$ with A.

Proof of ADP1. Let $\mathcal{S}$ be a finite set of formulas such that there is an atomic A with $A \in \mathcal{S}$ and $\sim A \in \mathcal{S}$. Let $\mathcal{S}'$ be the result of removing A and $\sim A$ from $\mathcal{S}$. Let $S' := \vee\mathcal{S}'$. Since the simple tree representation of $S' \vee [A \vee \sim A]$ is an ET-proof of S, we only need to produce a grounded version of this tree. To do this, simply expand or select on any existential or universal terminal node of this tree with a variable which is new to the current tree. Since the number of Π's within the tree's Fm value is reduced by one in each such step, we will eventually get a grounded expansion tree Q. Here the dependency relation is empty and, therefore, acyclic, and $Fm(Q)$ contains $A \vee \sim A$ as a disjunct and is therefore tautologous. Hence, Q is a grounded ET-proof.

Proof of ADP2. Let Q be an ET-proof for $S \vee \rho A$. Since $S \vee \rho A$ may have more free variables than $S \vee A$, some of which may have been selected in Q, Q may not be an expansion tree *for* $S \vee A$. By Proposition 2.32, we know that $S \vee A$ has a grounded ET-proof Q' such that no member of $\mathbf{S}_{Q'}$ is free in $S \vee A$. Q' is then a expansion tree for $S \vee A$.

Proof of ADP3. We shall prove a stronger form of ADP3 for this particular Λ. Let Q be an ET-proof for $S \vee A$ and let C be the result of replacing a subformula B ($\sim\sim B$) of A which is in the scope of only occurrences of $\vee$ and $\sim$ with $\sim\sim B$ (B). Let Q' be the result of replacing the corresponding subtree Q_0 ($\sim\sim Q_0$) with the subtree $\sim\sim Q_0$ (Q_0). Q' is a grounded ET-proof for $S \vee C$. Obviously, ADP3 follows immediately.

The proof of ADP4 is postpone until later.

Proof of ADP5. This follows immediately from the discussion above concerning the well-definedness of $\Lambda(\mathcal{S})$.

Proof of ADP6. Let Q be a grounded ET-proof for $S \vee [\sim\Pi A \vee \sim AB]$. Q then decomposes into subtrees $Q_0, \ldots, Q_{m+1}$ with $m \geq 1$, such that

$$Q = Q_0 \vee [\sim[\Pi A' +^{t_1} Q_1 + \ldots +^{t_m} Q_m] \vee \sim Q_{m+1}]$$

where A' is a λ-normal form of A, Q_0 is an expansion tree for S and $\sim Q_{m+1}$ is an expansion tree for $\sim AB$. Now let Q' be the tree

$$Q_0 \vee \sim[\Pi A' +^{t_1} Q_1 + \ldots +^{t_m} Q_m +^{B} Q_{m+1}].$$

Q' is a grounded expansion tree for $S \vee \sim\Pi A$. We now verify that it is in fact a grounded ET-proof of $S \vee \sim\Pi A$.

Notice that

$$\begin{aligned} &Fm(Q) = Fm(Q_0) \vee \sim[Fm(Q_1) \wedge \ldots \wedge Fm(Q_m)] \vee \sim Fm(Q_{m+1}) \text{ and} \\ &Fm(Q') = Fm(Q_0) \vee \sim[Fm(Q_1) \wedge \ldots \wedge Fm(Q_{m+1})]. \end{aligned}$$

Since these are truth-functionally equivalent and since $Fm(Q)$ is tautologous, so too is $Fm(Q')$.

Let $<_Q$ and $<_{Q'}$ be the dependency relations for the trees Q and Q', resp. Assume that $<_{Q'}$ has a cycle. Then there are expansion term occurrences $\{t_1, \ldots, t_p\} \subset \Theta_{Q'}$ with $p > 1$ such that $t_1 <^0_{Q'} t_2 <^0_{Q'} \ldots <^0_{Q'} t_p$ and $t_1 = t_p$. Then $B = t_j$ for some j, $1 < jlp$, since otherwise, this $<_{Q'}$-cycle would correspond to a $<_Q$-cycle. Since $t_{j-1} <^0_{Q'} B$, we know that there is a variable y, selected in Q' which is free in B. However, that would mean that y is selected in Q, while y is free in $S \vee [\sim\Pi A \vee \sim AB]$. But this contradicts the fact that Q is an expansion tree for $S \vee [\Pi A \vee \sim AB]$. Hence, $<_{Q'}$ is acyclic, and Q' is a grounded ET-proof of $S \vee \sim\Pi A$.

Proof of ADP7. Follows immediately from Lemma 2.38.

Proof of ADP4. First notice that whenever an abstract derivability property satisfies the strong form of ADP3 which was proved above, the condition ADP4 is equivalent to the following condition: If $\Lambda(\mathcal{S} \cup \{A\})$ and $\Lambda(\mathcal{S} \cup \{B\})$, then $\Lambda(\mathcal{S} \cup \{A \wedge B\})$. For convenience, we shall prove this latter condition here.

Next, we provide an algorithm and a lemma.

2.5.40. Merge Algorithm. Let S be a λ-normal formula$_o$ which has no universal boolean subformula. We define $\text{Merge}(Q_1, Q_2)$ when Q_1 and Q_2 are grounded expansion trees for S or when $\sim Q_1$ and $\sim Q_2$ are grounded expansion trees for S. Here S would have the form $\sim S'$. In either case, the selected variables in the two expansion trees must be disjoint.

(1) If Q_1 is a one-node tree, then so is Q_2. Set $\text{Merge}(Q_1, Q_2) := Q_1$.

(2) If $Q_1 = \sim Q_1'$ then $Q_2 = \sim Q_2'$. Set $\text{Merge}(Q_1, Q_2) := \sim\text{Merge}(Q_1', Q_2')$.

(3) If $Q_1 = Q_1' \vee Q_1''$ then $Q_2 = Q_2' \vee Q_2''$. Set

$$\text{Merge}(Q_1, Q_2) := \text{Merge}(Q_1', Q_2') \vee \text{Merge}(Q_1'', Q_2'').$$

(4) If $Q_1 = \Pi B_1 +^{t_1} Q_1^1 + \ldots +^{t_n} Q_1^n$ then $Q_2 = \Pi B_2 +^{s_1} Q_2^1 + \ldots +^{s_m} Q_2^m$, where B_1, B_2 are formulas$_o$, B_1 conv-I B_2, $t_1, \ldots, t_n, s_1, \ldots, s_m$ are formulas$_\alpha$, and $n, m \geq 1$. Set

$$\text{Merge}(Q_1, Q_2) := \Pi B_1 +^{t_1} Q_1^1 + \ldots +^{t_n} Q_1^n +^{s_1} Q_2^1 + \ldots +^{s_m} Q_2^m.$$

Since we do not have top-level universal nodes, we need to only consider this one case. ∎

Notice, that if Q_1 and Q_2 are expansion trees for S, then so is $\text{Merge}(Q_1, Q_2)$. If $\sim Q_1$ and $\sim Q_2$ are expansion trees for S then so is $\sim\text{Merge}(Q_1, Q_2)$.

2.5.41. **Lemma.** *Let S be a λ-normal formula$_o$ which has no universal boolean subformulas. If Q_1 and Q_2 are grounded expansion trees for S which share no selected variables in common and $Q :=$ Merge(Q_1, Q_2) then*

$$[Fm(Q_1) \vee Fm(Q_2)] \supset Fm(Q) \text{ is tautologous.} \tag{*}$$

If $\sim Q_1$ and $\sim Q_2$ are grounded expansion trees for S which share no selected variables in common and $Q :=$ Merge(Q_1, Q_2) then

$$Fm(Q) \supset .Fm(Q_1) \wedge Fm(Q_2) \text{ is tautologous.} \tag{**}$$

Proof. The proof is by induction on the boolean structure of Q_1 and, therefore, also on the boolean structure of S.

(1) If Q_1 is the one-node tree, so too is Q_2 and $Fm(Q_1)$ conv-I $Fm(Q_2)$. In either case $(*)$ or $(**)$, the implication is tautologous.

(2) Assume that Q_1 and Q_2 are expansion trees for S. If $Q_1 = \sim Q_1'$ then $Q_2 = \sim Q_2'$. Setting $Q' :=$ Merge(Q_1', Q_2') it is seen that the following three formulas are tautologous.

$$\begin{array}{rl} Fm(Q') \supset .Fm(Q_1') \wedge Fm(Q_2') & \text{by the inductive hypothesis} \\ [\sim Fm(Q_1') \vee \sim Fm(Q_2')] \supset \sim Fm(Q') & \text{by the contrapositive rule} \\ [Fm(\sim Q_1') \vee Fm(\sim Q_2')] \supset Fm(\sim Q') & \text{by the definition of } Fm \end{array}$$

and, finally, we have $(*)$ where $Q = \sim Q'$. We prove $(**)$ in a very similar fashion.

(3) Assume that Q_1 and Q_2 are expansion trees for S. If $Q_1 = Q_1' \vee Q_1''$ then $Q_2 = Q_2' \vee Q_2''$. Set $Q' :=$ Merge(Q_1', Q_2') and $Q'' :=$ Merge(Q_1'', Q_2''). By the inductive hypothesis, both of the following formulas are tautologous.

$$\begin{array}{l} [Fm(Q_1') \vee Fm(Q_2')] \supset Fm(Q') \\ [Fm(Q_1'') \vee Fm(Q_2'')] \supset Fm(Q'') \end{array}$$

The conjunction of both these formulas truth-functionally implies $(*)$. The other case is proved similarly.

(4) Otherwise, $Q_1 = \Pi B_1 +^{t_1} Q_1^1 + \ldots +^{t_n} Q_1^n$ and $Q_2 = \Pi B_2 +^{s_1} Q_2^1 + \ldots +^{s_m} Q_2^m$, where B_1, B_2 are formulas$_{o\alpha}$, B_1 conv-I B_2, $t_1, \ldots, t_n, s_1, \ldots, s_m$ are formulas$_\alpha$, and $n, m \geq 1$. Since

$$Fm(Q) = Fm(Q_1^1) \wedge \ldots \wedge Fm(Q_1^n) \wedge Fm(Q_2^1) \wedge \ldots \wedge Fm(Q_2^m)$$

the formula $Fm(Q) \supset .Fm(Q_1) \wedge Fm(Q_2)$ is tautologous. **Q.E.D.**

Now we prove ADP4. Assume that $S \vee A$ and $S \vee B$ have grounded ET-proofs. Let S' be the result of repeatly instantiating universal boolean subformulas of S with new variables as described in Lemma 2.38. By Lemma 2.38, both $S' \vee A$ and $S' \vee B$ have ET-proofs, so let $Q_1 \vee Q_2$ and $Q_3 \vee Q_4$ be grounded ET-proofs for $S' \vee A$ and $S' \vee B$, respectively. By use of Proposition 2.32, we may assume that $Q_1 \vee Q_2$ and $Q_3 \vee Q_4$ share no selected variables and that these selected variables have no occurrences in the other's tree. We will show below that $S' \vee [A \wedge B]$ has a grounded ET-proof. By Lemma 2.38 again, we shall be able to conclude that $S \vee [A \wedge B]$ has a grounded ET-proof, thus finishing the proof of ADP4.

Set $Q_5 := \text{Merge}(Q_1, Q_3)$. We claim that $Q := Q_5 \vee [Q_2 \wedge Q_4]$ is a grounded ET-proof for $S' \vee [A \wedge B]$. Clearly, Q is a grounded expansion tree for $S' \vee [A \wedge B]$. Since $Q_1 \vee Q_2$ and $Q_3 \vee Q_4$ are ET-proofs, $Fm(Q_1) \vee Fm(Q_2)$ and $Fm(Q_3) \vee Fm(Q_4)$ are tautologous. By Lemma 2.41, $[Fm(Q_1) \vee Fm(Q_3)] \supset Fm(Q_5)$ is tautologous. Hence, so too is

$$Fm(Q) = Fm(Q_5) \vee [Fm(Q_2) \wedge Fm(Q_4)].$$

Let $<_Q$, $<_{12}$, and $<_{34}$ be the dependency relations for Q, $Q_1 \vee Q_2$, and $Q_3 \vee Q_4$, respectively. Also, let Θ_Q, Θ_{12}, and Θ_{34} be defined with respect to these same trees. There is a natural correspondence between the term occurrences in Θ_{12} and Θ_{34} with those in Θ_Q. Assume that $<_Q$ has a cycle, *i.e.*

$$t_1 <^0_Q t_2 <^0_Q \dots <^0_Q t_m$$

for $m > 1$, $t_i \in \Theta_Q$ and $t_m = t_1$. Each t_i corresponds to either a term occurrence in Θ_{12} or in Θ_{34}. Let i be an arbitrary integer, $1li < m$, and assume that t_i corresponds to a term occurrence in Θ_{12}. Since $t_i <^0_Q t_{i+1}$, there is a variable y which is selected for some node M of Q dominated by t_i such that y is free in t_{i+1}. Since y has an occurrence in $Q_1 \vee Q_2$, it cannot be selected in $Q_3 \vee Q_4$, so M must correspond to node in $Q_1 \vee Q_2$ while t_{m+1} corresponds to an expansion term occurrence in the same tree. Since i was arbitrary, if any one of the expansion term occurrences $t_1, \dots, t_{m-1}$ corresponds to a term occurrence in Θ_{12}, then they all do. The same is true with respect to the set Θ_{34}. Hence, either $<_{12}$ or $<_{34}$ must have a cycle, which is a contradiction. Thus, $<_Q$ is acyclic, and Q is indeed an ET-proof of $S' \vee [A \wedge B]$. This ends the proof of ADP4 and of Theorem 2.39. **Q.E.D.**

2.5.42. **Relative completeness theorem for ET-proofs.** *Grounded ET-proofs are relatively complete for $\mathcal{T}$, i.e. if A is a formula$_o$ such that $\vdash_{\mathcal{T}} A$, then A has a grounded ET-proof.*

Proof. Follows immediately from Theorems 2.39 and 2.13. **Q.E.D.**

As a result of our soundness and completeness results, we can now give a simple but nonconstructive proof of the following proposition.

2.5.43. **Proposition.** *If A has an ET-proof, then it has a grounded ET-proof.*

Proof. Assume that A has an ET-proof. By the Soundness Theorem 2.37, we have $\vdash_{\mathcal{T}} A$. By the Completeness Theorem above, we then know that A has a grounded ET-proof. **Q.E.D.**

CHAPTER 3

Skolemization

Section 3.1: Introduction

In the resolution system described by Andrews in [Andrews71], skolemization was done by permitting choice functions to be used to do existential instantiations. (Remember that in a resolution system, we work with the negation of a proposed theorem. Hence, existential instantiation corresponds to universal instantiation in our situation.) The critical part of the resolution definition is

$$\text{From } M \vee \sim\Pi_{o(o\alpha)} A_{o\alpha} \text{ to infer } M \vee \sim A[k_{\alpha(o\alpha)} A]$$

where $k_{\alpha(o\alpha)}$ is called an *existential parameter*, which behaves somewhat like a skolem function. There is no restriction on how these existential parameters are used within substitution terms used in doing resolution, so in fact, it is possible (as shown by Andrews in [Andrews73]) to prove the following instance of the Axiom of Choice (see Section 5.3).

$$(AC^{\iota}) \qquad \exists c_{\iota(o\iota)}\, \forall p_{o\iota}\, .[\exists x_{\iota}\, px] \supset p.cp$$

Here, c gets instantiated by the formula

$$\lambda p_{o\iota}.k_{\iota(o\iota)}.\lambda x_{\iota}.\sim px.$$

Hence, the negation of AC^{ι} is refuted because the above existential instantiation inference rule implicitly uses the axiom of choice. We need to restrict the occurrences of the existential (choice) parameter k within substitution terms in order to use it correctly. Skolem functions can be ill-used in exactly the same way. In this chapter, we shall define a variant of ET-proofs, called ST-proofs, which use skolem functions in place of selected variables. We shall restrict the occurrences of skolem functions within substitution formulas in such a fashion that the above formula could not appear in a proof structure, assuming that k is actually a skolem function. It is, however, the case that the existential parameters used above are strictly stronger than the skolem functions we shall use. For example, even if we do not restrict the occurrences of skolem functions in substitution instances, we cannot "prove" AC^{ι} while we can "prove"

$$\forall x_{\iota}\, \exists y_{\iota}\, P_{o\iota\iota}xy \supset \exists f_{\iota\iota}\, \forall z_{\iota}\, .Pz.fz$$

which is not a theorem of $\mathcal{T}$. (This can be established by methods identical to the ones used in Appendix 1.) Using the proper restriction of skolem function occurrences in ST-proofs will not enable us to "prove" this last formula.

Section 3.2: Skolem Expansion Trees

3.2.1. Definition. The list $\sigma := \langle\alpha, \beta_1, \ldots, \beta_p\rangle$, where $\alpha, \beta_1, \ldots, \beta_p$ are type symbols ($p \geq 0$), is called a *signature* (for a skolem function). For each signature, σ, let $\mathcal{K}_\sigma$ be a denumerably infinite set of functions symbols all of type $(\ldots(\alpha\beta_1)\ldots\beta_p)$ which are not in the formulation of $\mathcal{T}$ and such that if σ_1 and σ_2 are two different signatures then $\mathcal{K}_{\sigma_1}$ and $\mathcal{K}_{\sigma_2}$ are disjoint. If $f \in \mathcal{K}_\sigma$, f is called a *skolem functions of signature σ with arity p.* Let $\mathcal{T}^*$ be the formulation of $\mathcal{T}$ in which all the skolem functions are added. ∎

Notice that two skolem functions may have the same type while they have different arities. For example, if α is of the form $\alpha'\beta_0$, then a skolem term with signature $\langle\alpha, \beta_1, \ldots, \beta_p\rangle$ and one with the signature $\langle\alpha', \beta_0, \beta_1, \ldots, \beta_p\rangle$ have different arities but have the same type. Since types can generally be determined from context while arity often cannot be, we shall frequently write skolem functions with a superscripted non-negative integer to denote its arity, *i.e.* f^p.

3.2.2. Definition. We shall define a set $\mathcal{U}$, called the *Herbrand Universe*, of formulas of $\mathcal{T}^*$. ($\mathcal{U}_\alpha$ will denote the set of formulas in $\mathcal{U}$ of type α.) Let $\mathcal{U}$ be the smallest set of formulas of $\mathcal{T}^*$ such that

(1) All variables and parameters are in $\mathcal{U}$. (For convenience, we shall consider skolem functions to be different from variables or parameters.)

(2) If $p \geq 0$, $t_i \in \mathcal{U}_{\beta_i}$ for $i = 1, \ldots, p$, and f has signature $\langle\alpha, \beta_1, \ldots, \beta_p\rangle$, then $ft_1 \ldots t_p \in \mathcal{U}_\alpha$. Formulas such as $ft_1 \ldots t_p$ are called *skolem terms* and the terms $t_1, \ldots, t_p$ are called the *necessary arguments* of f. Since such skolem terms may be of any functional type, α in this case, skolem functions may occur in formulas with more than their arity-number of arguments.

(3) If $A \in \mathcal{U}_{\alpha\beta}$ and $B \in \mathcal{U}_\beta$ then $[AB] \in \mathcal{U}_\alpha$.

(4) If $A \in \mathcal{U}_\alpha$ and x is a variable$_\beta$ which is not free in any neccessary argument of any skolem function occurrence in A, then $[\lambda x A] \in \mathcal{U}_{\alpha\beta}$.

The important clause in this definition is (4). It will be formally justified in Section 3.4 where we define "deskolemizing." ∎

Notice that if $A \in \mathcal{U}$ and f^p has an occurrence in A, then that occurrence is applied to p arguments, and if some variable has a free occurrence in one of these arguments then that occurrence is free in A.

3.2.3. Example. If f, g are skolem terms with signature $\langle\iota, \iota\rangle$, x, w are variables$_\iota$, and A is a variable$_{o(o\iota)}$ then $[f.gx] \in \mathcal{U}_\iota$, $[\lambda x.x] \in \mathcal{U}_{\iota\iota}$, $[\lambda w.Aw.gx] \in \mathcal{U}_{o\iota}$, while $f \notin \mathcal{U}$, $[\lambda x.fx] \notin \mathcal{U}$, and $[\lambda w.A[gx].fw] \notin \mathcal{U}$. In particular, if we treat the existential parameter $k_{\iota(o\iota)}$ (in the preceding section) as a skolem function of arity 1, the substitution term $[\lambda p_{o\iota}.k_{\iota(o\iota)}.\lambda x_\iota.\sim px]$ is not a member of $\mathcal{U}$. ∎

Notice that $\mathcal{U}$ is not closed under λ-convertibility. For example, if f is a skolem function of signature $\langle\iota, \iota\rangle$ then a λ-expansion of $[\lambda y_\iota.y] \in \mathcal{U}_{\iota\iota}$ is $[\lambda x_{\iota\iota}\lambda y.y]f$ which is not in $\mathcal{U}$. This lack of closure means that we must be careful of our use of the Herbrand Universe. We can, however, prove the following two propositions concerning λ-convertibility and $\mathcal{U}$.

3.2.4. **Proposition.** *If $C \in \mathcal{U}_\alpha$, $D \in \mathcal{U}_\beta$, and x is a variable$_\beta$, then* $\underset{\cdot}{\mathbf{S}}^x_D C \in \mathcal{U}_\alpha$.

Proof. We prove this by induction on the structure of C.

(1) C is a variable or parameter. Then $\underset{\cdot}{\mathbf{S}}^x_D C$ is either C or D. In either case, the result is in $\mathcal{U}_\alpha$.

(2) C is $f^p t_1 \dots t_p$, for some skolem function f^p. Since $t_i \in \mathcal{U}$, the inductive hypothesis yields $\underset{\cdot}{\mathbf{S}}^x_D t_i \in \mathcal{U}$, for $i = 1, \dots, p$. Hence, $\underset{\cdot}{\mathbf{S}}^x_D [f^p t_1 \dots t_p] = f^p [\underset{\cdot}{\mathbf{S}}^x_D t_1] \dots [\underset{\cdot}{\mathbf{S}}^x_D t_p] \in \mathcal{U}_\alpha$.

(3) C is $[EF]$ for $E \in \mathcal{U}_{\alpha\gamma}$, and $F \in \mathcal{U}_\gamma$. Then by the inductive hypothesis, $\underset{\cdot}{\mathbf{S}}^x_D E \in \mathcal{U}_{\alpha\gamma}$, $\underset{\cdot}{\mathbf{S}}^x_D F \in \mathcal{U}_\gamma$, and $\underset{\cdot}{\mathbf{S}}^x_D [EF] = [\underset{\cdot}{\mathbf{S}}^x_D E][\underset{\cdot}{\mathbf{S}}^x_D F] \in \mathcal{U}_\alpha$.

(4) C is $[\lambda y E]$ where $\alpha = (\gamma\delta)$, y is a variable$_\delta$, $E \in \mathcal{U}_\gamma$ and y is not free in any necessary argument of any occurrence of a skolem function in E. If $x = y$ then $\underset{\cdot}{\mathbf{S}}^x_D C = C \in \mathcal{U}_\alpha$. Assume that $x \neq y$. We may also assume that y is not free in D, since we would change y to some new variable in a systematic fashion to avoid variable capture. Now, by the inductive hypothesis, $\underset{\cdot}{\mathbf{S}}^x_D E \in \mathcal{U}_\gamma$. But y is not free in any necessary argument of any occurrence of a skolem function in $\underset{\cdot}{\mathbf{S}}^x_D E$, since this is true of E and y is not free in D. Hence, $\underset{\cdot}{\mathbf{S}}^x_D [\lambda y E] = \lambda y [\underset{\cdot}{\mathbf{S}}^x_D E] \in \mathcal{U}_\alpha$. **Q.E.D.**

3.2.5. **Proposition.** *If $A \in \mathcal{U}_\alpha$ and A* conv-*I-II* B, *then* $B \in \mathcal{U}_\alpha$.

Proof. First, notice that if $[\lambda x C]D \in \mathcal{U}$ then $C, D \in \mathcal{U}$, and hence, $\underset{\cdot}{\mathbf{S}}^x_D C \in \mathcal{U}$. Second, if $C \in \mathcal{U}$ and C and D are $\alpha\beta$-variants, then $D \in \mathcal{U}$. Third, if $C, D, E \in \mathcal{U}$ and D and E are of the same type, and the free variables of E are free in D, and F is the result of replacing an occurrence of D in C with E, then $F \in \mathcal{U}$. Using these three facts we have that every application of λRule1 and λRule2 carries a formula in $\mathcal{U}$ to a formula in $\mathcal{U}$. **Q.E.D.**

3.2.6. Definition. Let Q be a tree with a terminal node N which is labeled with ΠB for some formula$_o a$ B. If N is universal, then let $N_1, \dots, N_p$, $p \geq 0$ be those nodes which dominate N, are immediate descendants of existential nodes of Q, and whose in-arcs are labeled with formulas from $\mathcal{U}$. Also assume that if $1li < jlp$ then N_i dominates N_j. Let $t_i \in \mathcal{U}_{\beta_i}$ be the label on the in-arc of N_i. A *skolem instantiation of Q at N with respect to the skolem function f^p of signature* $\langle \alpha, \beta_1, \dots, \beta_p \rangle$ is the tree $Q +^s_N Q'$ where $s := f^p t_1 \dots t_p$ and Q' is the tree representation of some λ-normal form of Bs. There must also be the proviso that no other skolem instantiation in Q is done with respect to f^p.

The set of all *skolem expansion trees* is the smallest set of trees which contains the tree representations of all λ-normal formulas$_o$ of $\mathcal{T}$ and which is closed under expansions, with expansion terms taken from $\mathcal{U}$, and skolem instantiations. ∎

Notice that the definitions for derivation lists (2.20), Θ_Q (2.21), $Fm(Q)$ (2.25), and $Sh(Q)$ (2.26) can easily be extended to the case where Q is a skolem expansion tree.

3.2.7. Definition. A skolem expansion tree Q is a skolem expansion tree *for A* if $Sh(Q)$ is a λ-normal form of A and A contains no skolem function. Q is an ST-proof for A if Q is a skolem expansion tree for A and $Fm(Q)$ is tautologous. ∎

3.2.8. Example. Let A be the theorem $\exists y\, \forall x\, .Px \supset Py$, and let f and g be skolem functions with signature $\langle \iota, \iota \rangle$. A skolem expansion tree for A would then be the tree Q_1 given as (compare with Example 2.29):

$$\begin{aligned}&\sim[[\Pi\lambda y.\sim\Pi\lambda x.\sim Px \vee Py] +^u \sim[[\Pi\lambda x.\sim Px \vee Pu] +^{fu} [\sim P[fu] \vee Pu]]\\ &\qquad\qquad +^v \sim[[\Pi\lambda x.\sim Px \vee Pv] +^{gv} [\sim P[gv] \vee Pv]]].\end{aligned}$$

An ST-proof for A would then be the tree Q_2 given as:

$$\begin{aligned}&\sim[[\Pi\lambda y.\sim\Pi\lambda x.\sim Px \vee Py] +^u \sim[[\Pi\lambda x.\sim Px \vee Pu] +^{fu} [\sim P[fu] \vee Pu]]\\ &\qquad\qquad +^{fu} \sim[[\Pi\lambda x.\sim Px \vee P.fu] +^{g.fu} [\sim P[g.fu] \vee P.fu]]].\end{aligned}$$

∎

The usefulness of ST-proofs follows from this next proposition.

3.2.9. **Proposition.** *Let A be a formula$_o$ and Q be a skolem expansion tree for A. If x is a variable$_\alpha$ not free in A and $B \in \mathcal{U}_\alpha$ then $\underset{\cdot}{\mathbf{S}}^x_B Q$ is a skolem expansion tree for A.*

Proof. This follows immediately from the definitions of $\underset{\cdot}{\mathbf{S}}^x_B$ and skolem expansion trees, and from Proposition 3.4. **Q.E.D.**

In Example 3.8, Q_2, which is an ST-proof, is the result of substituting fu for v in Q_1, which is not an ST-proof. Notice that this is a much stronger version of the corresponding Proposition 2.31 (1) for expansion trees, where only a variable could be substituted for a variable. One method for attempting a search for ST-proofs of a proposed theorem would be to pick a skolem expansion tree, say Q, in which the expansion terms are all variables and then search for a substitution φ such that $\varphi Fm(Q)$ is tautologous. The search procedure described in [Andrews80] is essentially a refinement of this general strategy for FOL. In the higher-order case, it is necessary to restrict the substitution φ so that the range of φ is contained in $\mathcal{U}$. In Section 3.5, we describe how to modify Huet's unification algorithm [Huet75] so that only these kinds of substitutions are produced.

3.2.10. Definition. A skolem expansion tree is *grounded* if none of its terminal nodes are labeled with formulas of the form ΠB. An ST-proof for Q is a grounded ST-proof if it is also a grounded skolem expansion tree. ∎

Section 3.3: The Relative Completeness Theorem for ST-Proofs

3.3.11. **Relative Completeness Theorem for ST-proofs.** *If* $\vdash_{\mathcal{T}} A$ *then* A *has a grounded ST-proof.*

Proof. Assume that $\vdash_{\mathcal{T}} A$. By the Relative Completeness Theorem for ET-proofs (Theorem 2.42), A has a grounded ET-proof Q. We may assume that formulas labeling arcs are all in λ-normal form. We now describe how to convert Q into a grounded ST-proof of A.

Since Q is an ET-proof, $\prec_Q$ is acyclic. Let $\langle y_1, \ldots, y_r \rangle$ be a list of the variables in $\mathbf{S}_Q$ such that whenever $y_i \prec_Q y_j$, $i < j$.

For any selected variable y in Q we defined an associated skolem term s. Let N be the node selected by y, and let $t_1, \ldots, t_p$ be the expansion terms in Q which dominate N. Assume that these terms are ordered so that if $1ll < klp$ then t_l dominates t_k. Let f be a skolem function with signature $\langle \alpha, \beta_1, \ldots, \beta_p \rangle$ (where t_j is a formula$_{\beta_j}$), and set $s := ft_1 \ldots t_p$. Since none of the formulas, $t_1, \ldots, t_p$ contain skolem functions, $s \in \mathcal{U}_\alpha$. Now for each i such that $1lilr$, associate with y_i such a skolem term s_i, where we assume that no two of these skolem terms share the same skolem function as their head.

Let i and j be such that $1li, jlr$ and y_j is free in s_i. Then y_j is free in some expansion term which dominates the node selected by y_i. Hence, $y_j \prec^0_Q y_i$ and so $j < i$. Thus, $1liljlr$ implies that y_j is *not* free in s_i.

Let $\varphi := \mathbf{S}^{y_1}_{s_1} \circ \cdots \circ \mathbf{S}^{y_r}_{s_r}$. A simple induction argument shows that if B is a formula of $\mathcal{T}$, then φB will be a formula of $\mathcal{T}^*$ in which none of the variables $y_1, \ldots, y_r$ are free. Also, φy_i is a skolem term with top-level skolem function f_i. We now verify that $Q' := \varphi Q$ is a grounded ST-proof for A.

Let $\langle Q_1, \ldots, Q_n \rangle$ be a derivation list for Q. We show that φQ is an ST-proof for A by showing that $\langle \varphi Q_1, \ldots, \varphi Q_n \rangle$ is a derivation list for φQ. Since $Sh(Q_1)$ contains no selected variables of Q, then $Sh(\varphi Q_1) = Sh(Q_1)$ and, hence, φQ_1 is a tree representation of some λ-normal form of A. We now show that for $i = 1, \ldots, n-1$, φQ_{i+1} is either an expansion or skolem instantiation of φQ_i. We must consider two cases.

(1) Q_{i+1} is an expansion of Q_i at the existential, terminal node N (labeled with ΠB) with the formulas $t_1, \ldots, t_m \in \mathcal{U}$, *i.e.* $Q_{i+1} = Q_i +^{t_1}_N P_1 + \cdots +^{t_m}_N P_m$, where P_j is the tree representation of some λ-normal form of Bt_j. By repeated use of Proposition 3.4, $\varphi t_j \in \mathcal{U}$ for all $j = 1, \ldots, m$. Since Bt_j conv-*I-II* $Sh(P_j)$, $\varphi(B)\varphi(t_j) = \varphi(Bt_j)$ conv-*I-II* $\varphi Sh(P_j) = Sh(\varphi P_j)$. Since each t_j is λ-normal and the skolem terms $s_1, \ldots, s_r$ are λ-normal and are not top-level abstractions, $Sh(\varphi P_i)$ is in λ-normal form. Hence, $\varphi Q_{i+1} = \varphi Q_i +^{\varphi t_1}_{N'} \varphi P_1 + \cdots +^{\varphi t_m}_{N'} \varphi P_m$ and since φQ_i was assumed to be a skolem expansion tree, φQ_{i+1} is too. Here, N' is the node in φQ_i which corresponds to N in Q_i.

(2) Q_{i+1} is a selection of Q_i at the universal, terminal node N (labeled with ΠB) with the variable y_l (for some l, $1lllr$), *i.e.* $Q_{i+1} = Q_i +^{y_l}_N P$. As we reasoned above, $\varphi(B)\varphi(y_l) = \varphi(By_l)$ conv-*I-II* $\varphi Sh(P) = Sh(\varphi P)$ which is λ-normal. Let $t_1, \ldots, t_p$ (listed in the order of relative dominance) be the expansion trees in Q which dominate N. Hence, by definition, $s_l = f_l t_1 \ldots t_p$. For all j such that $lljlr$, y_j is not free in s_l and, thus, not in the terms $t_1, \ldots, t_p$. Hence, if we set $\varphi' := \mathbf{S}^{y_1}_{s_1} \circ \ldots \circ \mathbf{S}^{y_{l-1}}_{s_{l-1}}$, then $\varphi t_i = \varphi' t_i$. Thus, $\varphi y_l = \varphi' s_l = f_l(\varphi' t_1) \ldots (\varphi' t_p) = f_l(\varphi t_1) \ldots (\varphi t_p)$ (where f_l is the head of the skolem term s_l). Hence, the skolem term used to instantiate φQ_i has the correct arguments. Since φQ_i is assumed to be a skolem expansion tree, φQ_{i+1} is also since it is a skolem instantiation of φQ_i with respect to f_l.

Hence, $\varphi Q_n = \varphi Q$ is a skolem expansion tree. Since $Sh(\varphi Q_n) = Sh(\varphi Q_{n-1}) = \ldots = Sh(\varphi Q_1) = Sh(Q_1)$ is a λ-normal form for A, then φQ is in fact a skolem expansion tree for A. Also, since $Fm(Q)$ is tautologous, then $\varphi Fm(Q) = Fm(\varphi Q)$ is tautologous. We finally can conclude that φQ is an ST-proof for A. **Q.E.D.**

Section 3.4: The Soundness Theorem for ST-Proofs

3.4.12. Definition. Let $A \in \mathcal{U}_\alpha$, s be a skolem term$_\beta$, and y be a variable$_\beta$ which does not appear in A or in s. Let $D^s_y A$ be the result of replacing in A every subformula, t, such that t conv s, by y. ∎

Notice, that $D^s_y A \in \mathcal{U}_\alpha$ and $\underset{\cdot}{\mathbf{S}}^y_s D^s_y A$ conv A, where A, s, and y are as in the above definition.

3.4.13. Example. Let f be a skolem function with signature $\langle \iota\iota, \iota \rangle$. We then have the following:

$$D^{fv}_y [\lambda z_\iota . f v_\iota z] = \lambda z . y_{\iota\iota} z$$
$$D^{fv}_y P_{o(o\iota)} z . f[[\lambda w_\iota . w] v] = P z y$$

∎

The next two lemmas are required to show that λ-contractions are preserved by application of this "deskolemizing" operator.

3.4.14. **Lemma.** *Let $A, B \in \mathcal{U}$, s a skolem term$_\beta$, and y a variable$_\beta$ which has no occurrences in A, B, and s. If B arises from A by one application of λRule1, then $D^s_y A$ conv-I $D^s_y B$.*

Proof. Let $[\lambda x C]$ be the subformula A which is replaced by $[\lambda z . \underset{\cdot}{\mathbf{S}}^x_z C]$ to get B. Here we have the provisos that x is not bound in C and z does not appear in C. We must distinguish between two cases.

(1) If $[\lambda x C]$ is a subformula of a subformula t of A which is convertible to s, then the corresponding subformula t' of B is such that t' conv t. Thus $D^s_y A = D^s_y B$.

(2) If $[\lambda x C]$ does not occur in such a subformula, then every occurrence of such a subformula t in C is such that $\underset{\cdot}{\mathbf{S}}^x_z t$ conv s since x cannot occur free in s. Hence, $D^s_y B$ arises from $D^S_y B$ by replacing $[\lambda x . D^s_y C]$ with $[\lambda z . \underset{\cdot}{\mathbf{S}}^x_z D^s_y C]$, that is $D^s_y A$ conv-I $D^s_y B$. **Q.E.D.**

3.4.15. **Lemma.** *Let $A, B \in \mathcal{U}$, s be a skolem term$_\beta$, and y be a variable$_\beta$ which has no occurrences in A, B, and s. If B arises from A by one application of λRule2, then $D^s_y A$ conv-I-II $D^s_y B$.*

Proof. This proof resembles the one for the preceding Lemma. Let $[\lambda x C]E$ be the subformula of A which is replaced with $\underset{\cdot}{\mathbf{S}}^x_E C$, with the proviso that the bound variables of C are distinct both from x and from the free variables of E. We have two cases to consider.

(1) If $[\lambda x C]E$ is a subformula of a subformula t of A which is convertible to s, then the corresponding subformula t' of B is such that t' conv t. Thus $D^s_y A = D^s_y B$.

(2) If $[\lambda x C]E$ is not a subformula of such a subformula t, then every occurrence of such a subformula t in C is such that $\underset{\cdot}{\mathbf{S}}^x_E t$ conv s since x cannot occur free in s. Hence, $D^s_y B$ arises from $D^s_y B$ by replacing $[\lambda x . D^s_y C][D^s_t E]$ with $[\underset{\cdot}{\mathbf{S}}^x_{D^s_y E} D^s_y C]$, that is $D^s_y A$ conv-I-II $D^s_y B$. **Q.E.D.**

3.4.16. **Proposition.** *Let $A, B \in \mathcal{U}_\alpha$, s a skolem term$_\beta$, and y a variable$_\beta$ which has no occurrences in A, B, and s. If A conv-I-II B, then $D^s_y A$ conv-I-II $D^s_y B$.*

Proof. Follows immediately from Lemmas 3.14 and 3.15. **Q.E.D.**

Note: The requirement that $A \in \mathcal{U}$ is very important here. Let $A := [\lambda p_{o\iota} . k_{\iota(o\iota)} . \lambda x_\iota . {\sim} p x]$ be the substitution formula mentioned in Section 3.1, where $k_{\iota(o\iota)}$ is considered to be a skolem function of arity 1. If we set $s := [k_{\iota(o\iota)} . \lambda x_\iota . {\sim} p x]$ and $B := [\lambda q_{o\iota} . k_{\iota(o\iota)} . \lambda x_\iota . {\sim} q x]$, then it is not the case that $D^s_y A$ conv-I $D^s_y B$. The term s does not behave as simply a name, but rather as a important part in the function defined in A. It is this reason that the term s cannot be replaced by the name y. The need to remove skolem terms with the D^s_y operator is the reason why occurrences of skolem functions within substitution terms must be restricted. Notice that this kind of example does not occur if we restrict our attention to first order formulas only.

3.4.17. **Proposition.** *Let $A, B \in \mathcal{U}_\alpha$, $s \in \mathcal{U}_\beta$ be a skolem term, and y be a variable$_\beta$ which has no occurrences in A, B, and s. If B is a λ-normal form of A, then $D^s_y B$ is a λ-normal form of $D^s_y B$.*

Proof. A conv-*I-II* B and B has no λ-contractible subformulas. Hence, $D^s_y A$ conv-*I-II* $D^s_y B$. Clearly, $D^s_y B$ contains no contractible parts. **Q.E.D.**

3.4.18. **Soundness Theorem for ST-Proofs.** *If A has an ST-proof then $\vdash_{\mathcal{T}} A$.*

Proof. Let Q be an ST-proof for A and let $\mathcal{V}$ be the set of ρ-normal forms of all skolem terms which are subformulas of formulas used to do expansions or skolem instantiations in Q. We shall assume that all the formulas labeling arcs in Q are in λ-normal form. Thus, if $s \in \mathcal{V}$ then $s = f^p t_1 \ldots t_p$ where f^p is some p-arity skolem function and $t_1, \ldots, t_p$ are its arguments. Let $\langle s_1, \ldots, s_r \rangle$ be an ordering of $\mathcal{V}$ such that whenever s_j is an alphabetic variant of a subformula of s_i then $i < j$. Let $y_1, \ldots, y_r$ be r distinct variables new to Q and A such that y_i has the same type as s_i. Let φ be the "deskolemizing" operator

$$\varphi := D^{s_n}_{y_n} \circ \cdots \circ D^{s_1}_{y_1}.$$

Now $\varphi s_i = y_i$, for all $i = 1, \ldots, r$, since $j < i$ implies that $D^{s_j}_{y_j} s_i = s_i$. Also by Proposition 3.16, t conv-*I-II* s_i implies that $\varphi t = y_i$. We claim that φQ is an ET-proof of A, and hence, by the soundness for ET-proofs $\vdash_{\mathcal{T}} A$.

Let $\langle Q_1, \ldots, Q_m \rangle$ be a derivation list for Q. We now show that φQ is an expansion tree for A by showing that $\langle \varphi Q_1, \ldots, \varphi Q_m \rangle$ is a derivation list for φQ. Since Q_1 contains no skolem functions, $\varphi Q_1 = Q_1$ and φQ_1 is the tree representation of some λ-normal form of A. We now assume that for some $i = 1, \ldots, m-1$, φQ_i is an expansion tree for A. We consider two cases:

(a) Q_{i+1} is a skolem instantiation of Q_i at the universal, terminal node N (labeled with ΠB), *i.e.* for some $s_j \in \mathcal{V}$, $Q_{i+1} = Q_i +^t_N Q'$, where t conv-*I-II* s_j and Q' is the tree representation of some λ-normal form of Bt, and hence, also of Bs_j. Since $\varphi t = y_j$, $\varphi Q_{i+1} = \varphi Q_i +^{y_j}_{N'} \varphi Q'$ where N' is the node in φQ_i which corresponds to N in Q_i. Now Bt conv-*I-II* $Sh(Q')$, so by repeated use of Proposition 3.16, $\varphi(B) y_j = \varphi(B s_j)$ conv-*I-II* $\varphi Sh(Q') = Sh(\varphi Q')$, which is λ-normal by Proposition 3.17. Hence, $\varphi Q'$ is the tree representation of some λ-normal form of $\varphi(B) y_j$, and φQ_{i+1} is a selection of φQ_i.

(b) Q_{i+1} is an expansion of Q_i at the existential, terminal node N (labeled with ΠB) of Q_i, *i.e.* for some list of formulas$_\alpha$, $\langle t_1, \ldots, t_n \rangle$, $Q_{i+1} = Q_i +^{t_1}_N P_1 + \ldots +^{t_n}_N P_n$, where P_j is the tree representation of some λ-normal form of Bt_j. But, $\varphi Q_{i+1} = \varphi Q_i +^{\varphi t_1}_{N'} \varphi P_1 + \ldots +^{\varphi t_n}_{N'} \varphi P_n$, where N' is the node in φQ_i which corresponds to N in Q_i. As before, since Bt_j conv-*I-II* $Sh(P_j)$, we have by Proposition 3.16 that $\varphi(B)\varphi(t_j) = \varphi(Bt_j)$ conv-*I-II* $\varphi Sh(P_j) = Sh(\varphi P_j)$, which is in λ-normal form. Hence, φQ_{i+1} is an expansion of φQ_i.

Thus φQ is an expansion tree for A. Also, since $Fm(\varphi Q) = \varphi Fm(Q)$, $Fm(\varphi Q)$ is tautologous. We now only need to prove that $\prec_{\varphi Q}$ is acyclic. Assume that $y_i \prec^0_{\varphi Q} y_j$ for some $y_i, y_j \in \mathbf{S}_{\varphi Q}$. Then y_j is selected for a node, say H in φQ, which is dominated by some expansion term t which has y_i free in it. Let H' be the node in Q which corresponds to H in φQ, and let t' be the formula labeling the arc in Q corresponding to the arc labeled with t in φQ. In the skolem expansion tree, t' would have a subformula w such that $s_i = \rho w$ while the skolem term labeling the out-arc of H', say w', is such that $s_j = \rho w'$. Since w' is a skolem term with argument w, s_i is an alphabetic variant of a subformula of s_j and, therefore, $j < i$. Finally we conclude that, if $y_i \prec_{\varphi Q} y_j$ then $j < i$. Thus $\prec_{\varphi Q}$ is acyclic. **Q.E.D.**

Section 3.5: Skolemization and Unification

We now present the changes to Huet's unification algorithm which will ensure that if a disagreement set consists of pairs from $\mathcal{U}$, then the unifying substitutions for such pairs will also be in $\mathcal{U}$. We shall assume that the reader is familiar with this algorithm (see [Huet75]).

Of the two major portions of this algorithm, SIMPL and MATCH, MATCH is the only one which produces unifying substitutions. These substitutions are of two kinds — those produced by the *imitation* rule and those produced by the *projection* rule. All substitutions produced by the latter are members of $\mathcal{U}$. Hence, we need only look at terms produced by the imitation rule. When unifying a flexible term e_1 and a rigid term e_2 (both in $\mathcal{U}$) of the form

$$\begin{array}{l} e_1 = \lambda u_1 \ldots \lambda u_{n_1}.f(e^1_1, \ldots, e^1_{p_1}), \quad n_1 \geq 0, \quad p_1 \geq 0 \\ e_2 = \lambda v_1 \ldots \lambda w_{n_2}.@(e^2_1, \ldots, e^2_{p_2}), \quad n_2 \geq 0, \quad p_2 \geq 0, \end{array}$$

if the rigid head @ is not a skolem function, then the resulting substitutions are once again all members of $\mathcal{U}$. Hence, we must only consider the case when @ is a skolem function. Let q be the arity of @. The prescription of the imitation rule produces terms of the form

$$f \rightarrow \lambda w_1 \ldots \lambda w_m.@(E_1, \ldots, E_r)$$

where $E_i := h_i(w_1, \ldots, w_m)$ for each $i = 1, \ldots, r$ and where $w_1, \ldots, w_m$ and $h_1, \ldots, h_r$ are "new" variables. Here m and r are determined in various ways from n_1, n_2, p_1, and p_2. The substitution for f can fail to be in $\mathcal{U}$ for two reasons. The first is when $r < q$ (a cases we need not consider if we are using the η-rule). Hence, our first restriction is that $r \geq q$. Secondly, the terms $E_1, \ldots, E_q$ cannot be of such a general form since they contain variables occurrences (namely $w_1, \ldots, w_m$) which are not free in the full term. Hence, we must restrict E_i to be of the form h_i for $i = 1, \ldots, q$. With these two restrictions applied to those substitutions for f otherwise produced by the imitation rule in this case, we produce only substitution terms in $\mathcal{U}$.

CHAPTER 4

List Representation of Expansion Trees

Section 4.1: Introduction

For the purposes of the rest of this text, we shall add to $\mathcal{T}$ the logical constants $\wedge$ and $\supset$, and the quantifiers $\forall$ and $\exists$. When it is important to replace these connectives and quantifiers with $\sim$, $\vee$, and Π, they shall stand for the following: $A \wedge B$ stands for $\sim[\sim A \vee \sim B]$, $A \supset B$ stands for $\sim A \vee B$, $\forall x\ P$ stands for $\Pi[\lambda x P]$, and $\exists x\ P$ stands for $\sim\Pi[\lambda x \sim P]$. We add these connectives and quantifiers here since we wish to deal with a more conventional formulation of HOL, and especially because in the next chapter we shall describe sequential and natural deduction proofs for $\mathcal{T}$. Discussions of such systems would be quite awkward without considering the full complement of connectives and quantifiers usually considered.

We shall find it very convenient to formulate an alternative representation for expansion trees which are defined as list structures instead of labeled trees. The way in which we choose to do this will result in a much more succinct presentation of the information present in expansion trees. In this chapter, we define these list representations and show that they faithfully represent the information of expansion trees.

Section 4.2: The Definition of List Representations

We shall now present a representation of expansion trees which is more succinct and more suitable for direct implementation on computer systems, especially those written in LISP. The set of all list structures over a given set, Ξ, is defined to be the smallest set which contains Ξ and is closed under building finite tuples.

Notice that expansion and selection nodes in an expansion tree must have the right parity, so when we attempt to build up larger expansion trees from smaller ones, we must be careful how we imbed expansion trees under negations. This fact explains why we need to consider so many cases in the following definition.

4.2.1. Definition. Let Ξ be the set which contains the labels SEL and EXP and all formulas of $\mathcal{T}$. Let $\mathcal{E}$ be the smallest set of pairs $\langle R, A\rangle$, where R is a list structure over Ξ and A is a formula$_o$, which satisfies the conditions below. We say that a variable y is *selected* in the list structure R if it occurs in a sublist of the form (SEL y R').

(1) If A is a boolean atom and R is a λ-normal form of A, then $\langle R, A\rangle \in \mathcal{E}$ and $\langle \sim R, \sim A\rangle \in \mathcal{E}$. Here, $\sim R$ is shorthand for the two element list $(\sim R)$.

(2) If $\langle R, A\rangle \in \mathcal{E}$ then $\langle R, B\rangle \in \mathcal{E}$ where A conv B.

(3) If $\langle R, A\rangle \in \mathcal{E}$ then $\langle \sim\sim R, \sim\sim A\rangle \in \mathcal{E}$.

In cases (4), (5), and (6), we assume that R_1 and R_2 share no selected variables in common and that A_1 (resp. A_2) has no free variable selected in R_2 (resp. R_1).

(4) If $\langle R_1, A_1\rangle \in \mathcal{E}$ and $\langle R_2, A_2\rangle \in \mathcal{E}$ then $\langle(\vee\ R_1\ R_2), A_1 \vee A_2\rangle \in \mathcal{E}$ and $\langle(\wedge\ R_1\ R_2), A_1 \wedge A_2\rangle \in \mathcal{E}$.

(5) If $\langle{\sim}R_1, {\sim}A_1\rangle \in \mathcal{E}$ and $\langle{\sim}R_2, {\sim}A_2\rangle \in \mathcal{E}$ then $\langle{\sim}(\vee\ R_1\ R_2), {\sim}.A_1 \vee A_2\rangle \in \mathcal{E}$ and $\langle{\sim}(\wedge\ R_1\ R_2), {\sim}.A_1 \wedge A_2\rangle \in \mathcal{E}$.

(6) If $\langle{\sim}R_1, {\sim}A_1\rangle \in \mathcal{E}$ and $\langle R_2, A_2\rangle \in \mathcal{E}$ then $\langle(\supset\ R_1\ R_2), A_1 \supset A_2\rangle \in \mathcal{E}$ and $\langle{\sim}(\supset\ R_2\ R_1), {\sim}.A_2 \supset A_1\rangle \in \mathcal{E}$.

In cases (7), (8), and (9), we assume that y is not selected in R and that y is not free in $[\lambda x P]$ or in B.

(7) If $\langle R, [\lambda x P]y\rangle \in \mathcal{E}$ then $\langle(\text{SEL}\ \ y\ \ R), \forall x\, P\rangle \in \mathcal{E}$.

(8) If $\langle{\sim}R, {\sim}[\lambda x P]y\rangle \in \mathcal{E}$ then $\langle({\sim}(\text{SEL}\ \ y\ \ R)), {\sim}\exists x\, P\rangle \in \mathcal{E}$.

(9) If $\langle R, By\rangle \in \mathcal{E}$ then $\langle(\text{SEL}\ \ y\ \ R), \Pi B\rangle \in \mathcal{E}$.

Let $n \geq 1$. In cases (10), (11), and (12), we assume that for distinct i, j such that $1li, jln$, R_i and R_j share no selected variables and that no variable free in $[\lambda x P]t_i$ is selected in R_j.

(10) If for $i = 1, \ldots, n$, $\langle R_i, [\lambda x P]t_i\rangle \in \mathcal{E}$ then $\langle(\text{EXP}\ \ (t_1\ R_1) \ldots (t_n\ R_n)), \exists x\, P\rangle \in \mathcal{E}$.

(11) If for $i = 1, \ldots, n$, $\langle{\sim}R_i, {\sim}[\lambda x P]t_i\rangle \in \mathcal{E}$ then $\langle{\sim}(\text{EXP}\ \ (t_1\ R_1) \ldots (t_n\ R_n)), {\sim}\forall x\, P\rangle \in \mathcal{E}$.

(12) If for $i = 1, \ldots, n$, $\langle{\sim}R_i, {\sim}Bt_i\rangle \in \mathcal{E}$ then $\langle{\sim}(\text{EXP}\ \ (t_1\ R_1) \ldots (t_n\ R_n)), \Pi B\rangle \in \mathcal{E}$. ∎

The pair $\langle R, A\rangle \in \mathcal{E}$ represents — in a succinct fashion — an expansion tree. Notice, that the only formulas stored in R are those used for expansions and selections and those which are the leaves of the expansion tree. Expansion trees, as defined in Chapter 2, contain additional formulas which are used as "shallow formulas" to label expansion and selection nodes. These formulas, however, can be determined up to λ-convertibility if we know what the expansion tree is an expansion for. Informally, R can be considered an expansion tree for A.

Notice that if $\langle R, A\rangle \in \mathcal{E}$, then either R is a b-atom in λ-normal form, or it is a list whose first element is either $\sim$, $\vee$, $\wedge$, $\supset$, SEL or EXP.

4.2.2. **Proposition.** *If $\langle R, A\rangle \in \mathcal{E}$ then no variable selected in R is free in A.*

Proof. This is guaranteed by the conditions reguarding free and selected variables in Definition 4.1.

Q.E.D.

4.2.3. Example. Let $A := {\sim}\exists x_\iota\, \forall z_\iota\, C_{o\iota\iota} xz$ and $R := {\sim}(\text{SEL}\ \ y_\iota\ \ (\text{EXP}\ \ (a_\iota\, Cya)))$. We demonstrate that $\langle R, A\rangle \in \mathcal{E}$.

$$\begin{array}{rl}
\langle{\sim}Cya, {\sim}Cya\rangle \in \mathcal{E} & \text{by (1)} \\
\langle{\sim}Cya, {\sim}[\lambda z.Cyz]a\rangle \in \mathcal{E} & \text{by (2)} \\
\langle{\sim}(\text{EXP}\ \ (a\ Cya)), {\sim}\forall z\ Cyz\rangle \in \mathcal{E} & \text{by (11)} \\
\langle{\sim}(\text{EXP}\ \ (a\ Cya)), {\sim}[\lambda x \forall z\ Cyz]y\rangle \in \mathcal{E} & \text{by (2)} \\
\langle{\sim}(\text{SEL}\ \ y\ \ (\text{EXP}\ \ (a\ Cya))), {\sim}\exists x\ \forall z\ Cxz\rangle \in \mathcal{E} & \text{by (8)}
\end{array}$$

The expansion tree given in Example 2.29 on page 10 can be written as

$$(\text{EXP}\ \ (u\ (\text{SEL}\ \ v\ \ (\vee\ {\sim}Pv\ Pu)))(v\ (\text{SEL}\ \ w\ \ (\vee\ {\sim}Pw\ Pv)))).$$

∎

Notice that the converse of (2) in Definition 4.1 is not true — that is, if $\langle R, A\rangle \in \mathcal{E}$ and $\langle R, B\rangle \in \mathcal{E}$ then A and B are not necessarily λ-convertible. For example, the list structure (EXP $(a\ Paa)$) can be paired with $\exists x\ Pax$, $\exists x\ Pxa$ and $\exists x\ Paa$. Hence, the pairing of a list structure with a formula is important in order to know which expansion tree is being considered.

The rules for moving negations over quantifiers is mirrored within this list representation. For example, $\langle\sim(\text{EXP}\ \ (t_1\ R_1)\ldots(t_n\ R_n)), \sim\forall x\ P\rangle \in \mathcal{E}$ if and only if for $i = 1, \ldots, n$, $\langle\sim R_i, \sim[\lambda x P]t_i\rangle \in \mathcal{E}$ if and only if for $i = 1, \ldots, n$, $\langle\sim R_i, [\lambda x{\sim}P]t_i\rangle \in \mathcal{E}$ if and only if $\langle(\text{EXP}\ \ (t_1\ {\sim}R_1)\ldots(t_n\ {\sim}R_n)), \exists x\ {\sim}P\rangle \in \mathcal{E}$. Similarly, it is easy to show that $\langle\sim(\text{SEL}\ \ y\ \ R), \sim\exists x\ P\rangle \in \mathcal{E}$ if and only if $\langle(\text{SEL}\ \ y\ \ {\sim}R), \forall x\ {\sim}P\rangle \in \mathcal{E}$.

These pairs of list structures and formulas can be considered to be abbreviations of expansion trees. In order to confirm this, we define the function rep$[\![R, A]\!]$ whose value (when $\langle R, A\rangle \in \mathcal{E}$) is the expansion tree represented by this pair.

4.2.4. Definition. If A is a formula, define A^0 to be the result of eliminating the abbreviations for $\wedge$, $\supset$, $\forall$, and $\exists$ in A.

4.2.5. Definition. Let R and A be such that either $\langle R, A\rangle \in \mathcal{E}$ or $\langle\sim R, \sim A\rangle \in \mathcal{E}$.

(1) If A is not in ρ-normal form, then rep$[\![R, A]\!]$:= rep$[\![R, \rho A]\!]$.

In all the remaining cases, we shall assume that A is in ρ-normal form. Hence, the top-level structure of R is mirrored in the top-level structure of A.

(2) If R is a formula$_o$ and $\langle R, A\rangle \in \mathcal{E}$ then rep$[\![R, A]\!]$ is the tree representation of ρR^0.

(3) If $R = {\sim}R_1$ for some expansion tree R, then $A = {\sim}A_1$ for some formula A_1. Set rep$[\![R, A]\!]$:= ${\sim}$rep$[\![R_1, A_1]\!]$.

(4) If $R = (\vee\ R_1\ R_2)$ then $A = A_1 \vee A_2$. Set

$$\text{rep}[\![R, A]\!] := \text{rep}[\![R_1, A_1]\!] \vee \text{rep}[\![R_2, A_2]\!].$$

(5) If $R = (\wedge\ R_1\ R_2)$ then $A = A_1 \wedge A_2$. Set

$$\text{rep}[\![R, A]\!] := {\sim}[{\sim}\text{rep}[\![R_1, A_1]\!] \vee {\sim}\text{rep}[\![R_2, A_2]\!]].$$

(6) If $R = (\supset\ R_1\ R_2)$ then $A = A_1 \supset A_2$. Set

$$\text{rep}[\![R, A]\!] := [{\sim}\text{rep}[\![R_1, A_1]\!] \vee \text{rep}[\![R_2, A_2]\!]].$$

(7) If $R = (\text{SEL}\ \ y\ \ R_1)$ then we consider two cases: If $\langle R, A\rangle \in \mathcal{E}$ then A is of the form $\forall x\ P$ for some formula$_o$ P and some variable x or ΠB for some formula B. If $A = \forall x\ P$ then set $B := [\lambda x P]$. In either case, set

$$\text{rep}[\![R, A]\!] := \Pi B^0 +^y \text{rep}[\![R_1, \rho(By)]\!].$$

Otherwise, $\langle\sim R, \sim A\rangle \in \mathcal{E}$ and $A = \exists x\ P$. In this case, set

$$\text{rep}[\![R, A]\!] := {\sim}.\Pi[\lambda x{\sim}P^0] +^y \text{rep}[\![{\sim}R_1, \rho([\lambda x{\sim}P]y)]\!].$$

(8) If $R = (\text{EXP}\ \ (t_1\ \ R_1)\ldots(t_n\ \ R_n))$, then we consider two cases: If $\langle R, A\rangle \in \mathcal{E}$ then $A = \exists x\ P$. Set

$$\begin{aligned}\mathrm{rep}[\![R,A]\!] := {\sim}.\Pi[\lambda x{\sim}P^0]&\\ +^{t_1^0}{\sim}\mathrm{rep}[\![R_1,\rho([\lambda xP]t_1)]\!] + \ldots +^{t_n^0} {\sim}\mathrm{rep}[\![R_n,\rho([\lambda xP]t_n)]\!].&\end{aligned}$$

Otherwise, $\langle{\sim}R,{\sim}A\rangle \in \mathcal{E}$ and A is of the form $\forall x\ P$ or ΠB. If $A = \forall x\ P$ then set $B := [\lambda xP]$. Now set

$$\mathrm{rep}[\![R,A]\!] := \Pi B^0 +^{t_1^0} \mathrm{rep}[\![R_1,\rho(Bt_1)]\!] + \ldots +^{t_n^0} \mathrm{rep}[\![R_n,\rho(Bt_n)]\!]).$$

∎

4.2.6. Example. Let A and R be as in the previous example. We now compute $\mathrm{rep}[\![R,A]\!]$.

$$\begin{aligned}\mathrm{rep}[\![R,A]\!] &= {\sim}\mathrm{rep}[\![(\mathrm{SEL}\ \ y\ \ (\mathrm{EXP}\ \ (a\ Cya))), \exists x\,\forall z\ Cxz]\!] && \text{by (2)}\\ &= {\sim}{\sim}(\Pi[\lambda x{\sim}\Pi\lambda zCxz] +^y \mathrm{rep}[\![{\sim}(\mathrm{EXP}\ \ (a\ Cya)), {\sim}\forall z\ Cyz]\!]) && \text{by (6)}\\ &= {\sim}{\sim}(\Pi[\lambda x{\sim}\Pi\lambda zCxz] +^y {\sim}\mathrm{rep}[\![(\mathrm{EXP}\ \ (a\ Cya)), \forall z\ Cyz)]\!] && \text{by (2)}\\ &= {\sim}{\sim}(\Pi[\lambda x{\sim}\Pi\lambda zCxz] +^y {\sim}(\Pi[\lambda zCyz] +^a \mathrm{rep}[\![Cya, Cya]\!])) && \text{by (7)}\\ &= {\sim}{\sim}(\Pi[\lambda x{\sim}\Pi\lambda zCxz] +^y {\sim}(\Pi[\lambda zCyz] +^a Cya)) && \text{by (1)}\end{aligned}$$

Notice that $\mathrm{rep}[\![R,A]\!]$ is an expansion tree for $A^0 = {\sim}{\sim}\Pi[\lambda x.{\sim}\Pi[\lambda z.Cxz]]$. This relationship between $\mathrm{rep}[\![R,A]\!]$ and A^0 will be proved in the next section. ∎

Section 4.3: The Correctness of List Representations

In this section, we prove that the list structures defined in the previous section correctly represent expansion trees. If you are convinced of this fact, you may skip the rest of this chapter. The proofs below offer no new insights into the structure of expansion trees or of their list representations.

4.3.7. **Proposition.** *If* $\langle R,A\rangle \in \mathcal{E}$ *then* $\mathrm{rep}[\![R,A]\!]$ *is an expansion tree.*

Proof. We prove this by first proving the following compound statement by induction on the structure of R: If A is a λ-normal formula$_o$ then, if $\langle R,A\rangle \in \mathcal{E}$ then $\mathrm{rep}[\![R,A]\!]$ is an expansion tree, and if $\langle{\sim}R, {\sim}A\rangle \in \mathcal{E}$ then ${\sim}\mathrm{rep}[\![R,A]\!]$ is an expansion tree. The general case for formulas$_o$ not necessarily in λ-normal form follows from the following argument: If $\langle R,A'\rangle \in \mathcal{E}$ and A is a λ-normal form of A', then $\langle R,A\rangle \in \mathcal{E}$ and $\mathrm{rep}[\![R,A]\!]$ is an expansion tree. But $\mathrm{rep}[\![R,A]\!] = \mathrm{rep}[\![R,\rho A]\!] = \mathrm{rep}[\![R,A']\!]$. The inductive argument is below.

Let R be a b-atom. If $\langle R,A\rangle \in \mathcal{E}$ then $\mathrm{rep}[\![R,A]\!]$ is the tree representation for ρR^0 and is therefore an expansion tree. If $\langle{\sim}R,{\sim}A\rangle \in \mathcal{E}$ then $\mathrm{rep}[\![{\sim}R,{\sim}A]\!] = {\sim}\mathrm{rep}[\![R,A]\!]$ and ${\sim}\mathrm{rep}[\![R,A]\!]$ is a tree representation of ${\sim}\rho R^0$ and is, therefore, an expansion tree.

If $R = {\sim}R_1$ then $A = {\sim}A_1$. If $\langle R,A\rangle \in \mathcal{E}$ then $\langle{\sim}R_1,{\sim}A_1\rangle \in \mathcal{E}$ and by the inductive hypothesis, ${\sim}\mathrm{rep}[\![R_1,A_1]\!]$ is an expansion tree. But ${\sim}\mathrm{rep}[\![R_1,A_1]\!] = \mathrm{rep}[\![{\sim}R_1,{\sim}A_1]\!] = \mathrm{rep}[\![R,A]\!]$. If $\langle{\sim}R,{\sim}A\rangle \in \mathcal{E}$ then $\langle{\sim}{\sim}R_1,{\sim}{\sim}A_1\rangle \in \mathcal{E}$. Then also, $\langle R_1,A_1\rangle \in \mathcal{E}$, so by the inductive hypothesis, $\mathrm{rep}[\![R_1,A_1]\!]$ is an expansion tree. Then so too is ${\sim}{\sim}\mathrm{rep}[\![R_1,A_1]\!] = {\sim}\mathrm{rep}[\![{\sim}R_1,{\sim}A_1]\!] = {\sim}\mathrm{rep}[\![R,A]\!]$.

If $R = (\wedge\ R_1\ R_2)$, then $A = A_1 \wedge A_2$. If $\langle R,A\rangle \in \mathcal{E}$ then $\langle R_1,A_1\rangle \in \mathcal{E}$ and $\langle R_2,A_2\rangle \in \mathcal{E}$ and R_1 and R_2 share no selected variables in common. By the inductive hypothesis, if $Q_1 := \mathrm{rep}[\![R_1,A_1]\!]$ and $Q_2 := \mathrm{rep}[\![R_2, A_2]\!]$ then Q_1 and Q_2 are expansion trees. But $\mathrm{rep}[\![R,A]\!] = {\sim}[{\sim}Q_1 \vee {\sim}Q_2]$ is then an expansion tree. If $\langle{\sim}R, {\sim}A\rangle \in \mathcal{E}$ then $\langle{\sim}R_1,{\sim}A_1\rangle \in \mathcal{E}$ and $\langle{\sim}R_2,{\sim}A_2\rangle \in \mathcal{E}$. By the inductive hypothesis, if $Q_1 := \mathrm{rep}[\![R_1,A_1]\!]$ and $Q_2 := \mathrm{rep}[\![R_2,A_2]\!]$ then ${\sim}Q_1$ and ${\sim}Q_2$ are expansion trees. But ${\sim}\mathrm{rep}[\![R,A]\!] = {\sim}{\sim}[{\sim}Q_1 \vee {\sim}Q_2]$ is then an expansion tree.

The cases for R equal to $(\vee\ R_1\ R_2)$ and $(\supset\ R_1\ R_2)$ are very similar.

Let $R = (\text{SEL} \;\; y \;\; R_1)$. If $\langle R, A\rangle \in \mathcal{E}$ then A is either $\forall x\, P$ or ΠB. In the first case, set $B := [\lambda x P]$. Now $\langle R_1, \rho(By)\rangle \in \mathcal{E}$ so by the inductive hypothesis, $\text{rep}[\![R_1, \rho(By)]\!]$ is an expansion tree. Then so too is $\text{rep}[\![R, A]\!] = \Pi B^0 +^y \text{rep}[\![R_1, \rho(By)]\!]$. If $\langle \sim R, \sim A\rangle \in \mathcal{E}$ then A is $\exists x\, P$ and $\langle \sim R_1, \sim\rho([\lambda x P]y)\rangle \in \mathcal{E}$. By the inductive hypothesis, $\sim\text{rep}[\![R_1, \rho(By)]\!]$ is an expansion tree and, therefore, so is $\sim\text{rep}[\![R, A]\!] = \sim\sim(\Pi[\lambda x \sim P^0] +^y \sim\text{rep}[\![R_1, \rho([\lambda x P]y)]\!])$.

Let $R = (\text{EXP} \;\; (t_1 \; R_1) \ldots (t_n \; R_n))$. If $\langle R, A\rangle \in \mathcal{E}$ then $A = \exists x\, P$ and for $i = 1, \ldots, n$, $\langle R_i, \rho([\lambda x P]t_i)\rangle \in \mathcal{E}$. By the inductive hypothesis, $Q_i := \text{rep}[\![R_i, \rho([\lambda x P]t_i)]\!]$ is an expansion tree for each i. Then so must $\text{rep}[\![R, A]\!] = \sim(\Pi[\lambda x \sim P^0] +^{t_1^0} Q_1 + \ldots +^{t_n^0} Q_n)$. If $\langle \sim R, \sim A\rangle \in \mathcal{E}$ then A is either $\forall x\, P$ or ΠB. In the first case, set $B := [\lambda x P]$. Since $\langle \sim R_i, \sim\rho(Bt_i)\rangle \in \mathcal{E}$ then $Q_i := \text{rep}[\![R_i, \rho(Bt_i)]\!]$ is such that $\sim Q_i$ is an expansion tree for all i. Hence, $\sim\text{rep}[\![R, A]\!] = \sim(\Pi B^0 +^{t_1^0} Q_1 + \ldots +^{t_n^0} Q_n)$ is also an expansion tree. **Q.E.D.**

4.3.8. **Proposition.** *Let $\langle R, A\rangle \in \mathcal{E}$ or $\langle \sim R, \sim A\rangle \in \mathcal{E}$ and set $Q := \text{rep}[\![R, A]\!]$. Then $A^0 \text{ conv } Sh(Q)$.*

Proof. First notice that either Q or $\sim Q$ is an expansion tree, so Sh is defined for Q. We shall prove this Proposition, first in the case that A is in λ-normal form, by induction on the structure of R. The general case follows by the following argument: If $\langle R, A'\rangle \in \mathcal{E}$ and A is a λ-normal form of A', then $\langle R, A\rangle \in \mathcal{E}$ and, hence, $Q := \text{rep}[\![R, A]\!] = \text{rep}[\![R, A']\!]$ is such that $Sh(Q) \text{ conv } A^0 \text{ conv } A'^0$. Now assume that A is in λ-normal form.

If R is a b-atom, then A conv-I R. But Q is then the tree representation of ρR^0. Hence, $Sh(Q) = \rho R^0$ and A^0 conv-I $Sh(Q)$.

If $R = \sim R_1$, then $A = \sim A_1$, with A_1 in λ-normal form. Set $Q_1 := \text{rep}[\![R_1, A_1]\!]$. Then $Q = \sim Q_1$. By the inductive hypothesis, A_1^0 conv-I $Sh(Q_1)$, so $\sim A_1^0$ conv-I $\sim Sh(Q_1)$ and A^0 conv-I $Sh(Q)$.

If $R = (\wedge \;\; R_1 \;\; R_2)$ then $A = A_1 \wedge A_2$ with A_1 and A_2 in λ-normal form. Setting $Q_1 := \text{rep}[\![R_1, A_1]\!]$ and $Q_2 := \text{rep}[\![R_2, A_2]\!]$, we have by the inductive hypothesis that A_1^0 conv-I $Sh(Q_1)$ and A_2^0 conv-I $Sh(Q_2)$. Hence, $A^0 = \sim[\sim A_1^0 \vee \sim A_2^0]$ conv-I $\sim[\sim Sh(Q_1) \vee \sim Sh(Q_2)] = Sh(\sim[\sim Q_1 \vee \sim Q_2]) = Sh(Q)$.

The cases for when R is $(\vee \;\; R_1 \;\; R_2)$ or $(\supset \;\; R_1 \;\; R_2)$ is very similar and omitted here.

Let R be $(\text{SEL} \;\; y \;\; R_1)$. In the case that $\langle R, A\rangle \in \mathcal{E}$, A is $\forall x\, P$ or ΠB. In the first case, set $B := [\lambda x P]$. Now $Q = \Pi B^0 +^y \text{rep}[\![R_1, \rho(By)]\!]$, so $Sh(Q) = \Pi B^0 = A^0$. If $\langle \sim R, \sim A\rangle \in \mathcal{E}$ then A is $\exists x\, P$ and $Q = \sim(\Pi[\lambda x \sim P^0] +^y \text{rep}[\![\sim R_1, \rho([\lambda x \sim P]y)]\!]$ and $Sh(Q) = \sim\Pi[\lambda x \sim P^0] = A^0$.

Let R be $(\text{EXP} \;\; (t_1 \; R_1) \ldots (t_n \; R_n))$. In the case that $\langle R, A\rangle \in \mathcal{E}$, A is $\exists x\, P$ and Q is $\sim(\Pi[\lambda x \sim P^0] + \ldots)$. Hence, $Sh(Q) = \sim\Pi[\lambda x \sim P^0] = A^0$. If $\langle \sim R, \sim A\rangle \in \mathcal{E}$, then A is either $\forall x\, P$ or ΠB. In the first case, set $B := [\lambda x P]$. Then Q is $\Pi B^0 + \ldots$ so $Sh(Q) = \Pi B^0 = A^0$. **Q.E.D.**

4.3.9. **Theorem.** *If $\langle R, A\rangle \in \mathcal{E}$ then $\text{rep}[\![R, A]\!]$ is an expansion tree for A^0.*

Proof. Let $\langle R, A\rangle \in \mathcal{E}$ and set $Q := \text{rep}[\![R, A]\!]$. By Proposition 4.7, Q is an expansion tree. By Proposition 4.8, $Sh(Q)$ conv A^0, and by Proposition 4.2, no free variables in A, and therefore in A^0, are selected in Q. Hence, Q is an expansion tree for A^0. **Q.E.D.**

Notice that there is an obvious mapping of expansion trees into list structures. Hence, $\mathcal{E}$ contains no new expansion trees — just convenient representations of already defined expansion trees. A similar list structure could be introduced to abbreviate skolem expansion trees. If $\langle R, A\rangle \in \mathcal{E}$ and $Q := \text{rep}[\![R, A]\!]$, then we say that Q is represented by the list structure R.

CHAPTER 5

Sequential and Natural Deduction Proofs

Section 5.1: Introduction

In this chapter, we shall show how to transform ET-proofs for theorems into both cut-free sequential proofs (in a calculus which is a slight extension of Gentzen's LK calculus) and natural deduction proofs. Hence, we will explicitly show how Herbrand's Theorem in $\mathcal{T}$, which is essentially our Relative Completeness result for ET-proofs, implies the *Hauptsatz* for $\mathcal{T}$.

Within the context of natural deduction proofs, we shall investigate various criteria relevant to the "naturalness" and "readability" of proofs and how to have such criteria followed in our transformation process. In particular, we shall define the notion of a "focused proof outline" and show how to construct such proof outlines.

We shall use the symbol $\perp$ to denote falsehood. We will not consider $\perp$ as being part of our formulation of $\mathcal{T}$ since in the occasions when we need to use it, it merely stands as an indicator that we are attempting to prove a contradiction from certain hypotheses. It will never be used within a formula. We adopt the convention that the one-node tree whose node is labeled with $\perp$ is an expansion tree for $\perp$. By convention, also, we let $\perp$ be the list representation for this expansion tree.

Section 5.2: Sequential Proofs

The logical system, LKH defined below, is a higher-order extension to Gentzen's LK classical, logistic system. We shall, however, make a few necessary and convenient modifications.

5.2.1. Definition. The following are the inference rules of the LKH proof system. We shall assume that the reader is already familiar with Gentzen's LK proof system (see [Gentzen35]). Here, A, C, P and A' are formulas$_o$, such that A conv A'. B is a formula$_{o\alpha}$, x and y are variables$_\alpha$, and t is a formula$_\alpha$. Γ, Δ, Θ and Λ represent possibly empty, finite lists of formulas$_o$.

$$\frac{\Gamma \longrightarrow \Theta}{A, \Gamma \longrightarrow \Theta}\ \text{Thinning} \qquad \frac{\Gamma \longrightarrow \Theta}{\Gamma \longrightarrow \Theta, A}\ \text{Thinning}$$

$$\frac{A, A, \Gamma \longrightarrow \Theta}{A, \Gamma \longrightarrow \Theta}\ \text{Contraction} \qquad \frac{\Gamma \longrightarrow \Theta, A, A}{\Gamma \longrightarrow \Theta, A}\ \text{Contraction}$$

$$\frac{\Delta, A, C, \Gamma \longrightarrow \Theta}{\Delta, C, A, \Gamma \longrightarrow \Theta}\ \text{Interchange} \qquad \frac{\Gamma \longrightarrow \Theta, A, C, \Lambda}{\Gamma \longrightarrow \Theta, C, A, \Lambda}\ \text{Interchange}$$

$$\frac{A, \Gamma \longrightarrow \Theta}{A', \Gamma \longrightarrow \Theta}\ \lambda \qquad \frac{\Gamma \longrightarrow \Theta, A}{\Gamma \longrightarrow \Theta, A'}\ \lambda$$

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

$$\frac{\Gamma \longrightarrow \Theta, A \qquad \Gamma \longrightarrow \Theta, C}{\Gamma \longrightarrow \Theta, A \wedge C}\ \wedge\text{−IS}$$

$$\frac{A, \Gamma \longrightarrow \Theta}{A \wedge C, \Gamma \longrightarrow \Theta}\ \wedge\text{−IA} \qquad \frac{C, \Gamma \longrightarrow \Theta}{A \wedge C, \Gamma \longrightarrow \Theta}\ \wedge\text{−IA}$$

$$\frac{A, \Gamma \longrightarrow \Theta \qquad C, \Gamma \longrightarrow \Theta}{A \vee C, \Gamma \longrightarrow \Theta}\ \vee\text{−IA}$$

$$\frac{\Gamma \longrightarrow \Theta, A}{\Gamma \longrightarrow \Theta, A \vee C}\ \vee\text{−IS} \qquad \frac{\Gamma \longrightarrow \Theta, C}{\Gamma \longrightarrow \Theta, A \vee C}\ \vee\text{−IS}$$

$$\frac{\Gamma \longrightarrow \Theta, A}{\sim A, \Gamma \longrightarrow \Theta}\ \sim\text{−IA} \qquad \frac{A, \Gamma \longrightarrow \Theta}{\Gamma \longrightarrow \Theta, \sim A}\ \sim\text{−IS}$$

$$\frac{A, \Gamma \longrightarrow \Theta, C}{\Gamma \longrightarrow \Theta, A \supset C}\ \supset\text{−IS}$$

$$\frac{\Gamma \longrightarrow \Theta, A \qquad C, \Delta \longrightarrow \Lambda}{A \supset C, \Gamma, \Delta \longrightarrow \Theta, \Lambda}\ \supset\text{−IA}$$

$$\frac{[\lambda x P]t, \Gamma \longrightarrow \Theta}{\forall x\ P, \Gamma \longrightarrow \Theta}\ \forall\text{−IA} \qquad \frac{\Gamma \longrightarrow \Theta, [\lambda x P]y}{\Gamma \longrightarrow \Theta, \forall x\ P}\ \forall\text{−IS}^*$$

$$\frac{Bt, \Gamma \longrightarrow \Theta}{\Pi B, \Gamma \longrightarrow \Theta}\ \Pi\text{−IA} \qquad \frac{\Gamma \longrightarrow \Theta, By}{\Gamma \longrightarrow \Theta, \Pi B}\ \Pi\text{−IS}^*$$

$$\frac{[\lambda x P]y, \Gamma \longrightarrow \Theta}{\exists x\ P, \Gamma \longrightarrow \Theta}\ \exists\text{−IA}^* \qquad \frac{\Gamma \longrightarrow \Theta, [\lambda x P]t}{\Gamma \longrightarrow \Theta, \exists x\ P}\ \exists\text{−IS}$$

(*) The following proviso is placed on $\forall$−IS, Π−IS, and $\exists$−IA: The variable y is not free in any formula in the lower sequent for each of these rules. We shall generally assume that the λ rule can be applied without mention whenever one of the six quantifier rules is used.

Axioms in LKH are sequents of the form $A \longrightarrow A$, where A is any formula$_o$. Derivation trees are the same as defined in [Gentzen35]. A derivation tree is an *LKH-proof* of A if the root of the tree is the sequent $\longrightarrow A$. ∎

There are numerous ways to simply Gentzen's LK-calculus into equivalent calculi which contain fewer connectives and fewer rules. We have decided to use Gentzen's original formulation since the derivation trees in this system are more difficult to build and the intuitive use of the connectives is important when we discuss the "readability" of proofs. Since we shall be able to automate the building of LKH-proofs, it should be clear that we could do the same with many of the variations of this proof system.

5.2.2. Example. The following is an LKH-proof of the theorem

$$[\exists c_{\iota(o\iota)}\, \forall p_{o\iota}\, .[\exists u_\iota\, .pu] \supset .p.cp] \supset .[\forall x_\iota\, \exists y_\iota\, .P_{o\iota\iota}xy] \supset \exists f_{\iota\iota}\, \forall z_\iota\, .Pz.fz.$$

$$\dfrac{\dfrac{\exists y\,.Pxy \longrightarrow \exists y\,.Pxy}{\exists y\,.Pxy \longrightarrow \exists u\,.Pxu}\ \lambda \qquad Pz.c.Pz \longrightarrow Pz.c.Pz}{[\exists u\,.Pzu] \supset Pz.c.Pz, \exists y\,.Pzy \longrightarrow Pz.c.Pz}\ \supset\text{–IA}$$

$$\begin{array}{cl}
[\exists u\,.Pzu] \supset Pz.c.Pz, \exists y\,.Pzy \longrightarrow Pz.c.Pz & \\ \hline
\forall p\,.[\exists u\,.pu] \supset p.cp, \exists y\,.Pzy \longrightarrow Pz.c.Pz & \forall\text{–IA} \\ \hline
\forall p\,.[\exists u\,.pu] \supset p.cp, \forall x\, \exists y\,.Pxy \longrightarrow Pz.c.Pz & \forall\text{–IA} \\ \hline
\forall p\,.[\exists u\,.pu] \supset p.cp, \forall x\, \exists y\,.Pxy \longrightarrow \forall z\,.Pz.c.Pz & \forall\text{–IS} \\ \hline
\forall p\,.[\exists u\,.pu] \supset p.cp, \forall x\, \exists y\,.Pxy \longrightarrow \exists f\, \forall z\,.Pz.fz & \exists\text{–IS} \\ \hline
\exists c\, \forall p\,.[\exists u\,.pu] \supset p.cp, \forall x\, \exists y\,.Pxy \longrightarrow \exists f\, \forall z\,.Pz.fz & \exists\text{–IA} \\ \hline
\exists c\, \forall p\,.[\exists u\,.pu] \supset .p.cp \longrightarrow [\forall x\, \exists y\,.Pxy] \supset \exists f\, \forall z\,.Pz.fz & \supset\text{–IS} \\ \hline
\longrightarrow [\exists c\, \forall p\,.[\exists u\,.pu] \supset .p.cp] \supset .[\forall x\, \exists y\,.Pxy] \supset \exists f\, \forall z\,.Pz.fz & \supset\text{–IS}
\end{array}$$

5.2.3. **Proposition.** *If the sequent* $\Gamma, \sim A, \Delta \longrightarrow \Theta$ *has a cut-free LKH-proof, then so must the sequent* $\Gamma, \Delta \longrightarrow \Theta, A$. *If the sequent* $\Gamma \longrightarrow \Delta, \sim A, \Theta$ *has a cut-free LKH-proof, then so must the sequent* $\Gamma, A \longrightarrow \Delta, \Theta$.

Proof. This is easily proved by standard methods of moving $\sim$-introduction rules higher (*i.e.* closer to the leaves) in cut-free LKH-derivations. **Q.E.D.**

5.2.4. Definition. By merit of the preceding proposition, we define the inference rules $\sim$–EA, for negation elimination from the antecedent and $\sim$–ES for negation elimination from the succedent. Any cut-free LKH-proof which contains these inference rules can be converted to a cut-free LKH-proof without these inference rules. ∎

Section 5.3: Natural Deduction Proofs

The list of natural deduction inference rules below is not a minimal set of inference rules. Instead they represent the actual set of rules which our proof building algorithms described in subsequent sections will use.

5.3.5. Definition. The *inference rules* we will be using in natural deduction proofs are listed below. Here, A, A_0, $A_1, \ldots, A_n$ are formulas$_o$, B is a formula$_{o\alpha}$, x is a variable$_\alpha$, t is a formula$_\alpha$, and $\mathcal{H}$ is a possibly empty set of formulas$_o$.

Hypothesis Rule. From any set of hypotheses, we can assert one of its members.

$$\mathcal{H}, A \quad \vdash \quad A \qquad Hyp$$

Rule of λ-conversion. Here, A conv A_0.

$$\begin{array}{lll} \mathcal{H} & \vdash & A \\ \hline \mathcal{H} & \vdash & A_0 \qquad \lambda \end{array}$$

Rule of Propositional Calculus. Here, $\mathcal{H}_1 \cup \ldots \cup \mathcal{H}_n \subset \mathcal{H}$ and the formula $[A_1 \wedge \ldots \wedge A_n] \supset A$ is tautologous.

$$\begin{array}{lll} \mathcal{H}_1 & \vdash & A_1 \\ \vdots & & \vdots \\ \mathcal{H}_n & \vdash & A_n \\ \hline \mathcal{H} & \vdash & A \qquad RuleP \end{array}$$

Rule of Indirect Proof. If a contradiction can be inferred from the negation of A then we can infer A.

$$\begin{array}{lll} \mathcal{H}, \sim A & \vdash & \bot \\ \hline \mathcal{H} & \vdash & A \qquad IP \end{array}$$

Universal Instantiation.

$$\begin{array}{lll} \mathcal{H} & \vdash & \forall x\,.P \\ \hline \mathcal{H} & \vdash & [\lambda x.P]t \qquad \forall I \end{array} \qquad\qquad \begin{array}{lll} \mathcal{H} & \vdash & \Pi B \\ \hline \mathcal{H} & \vdash & Bt \qquad \forall I \end{array}$$

Universal Generalization. x is not free in any formula in $\mathcal{H}$ or in B.

$$\begin{array}{lll} \mathcal{H} & \vdash & P \\ \hline \mathcal{H} & \vdash & \forall x\; P \qquad \forall G \end{array} \qquad\qquad \begin{array}{lll} \mathcal{H} & \vdash & Bx \\ \hline \mathcal{H} & \vdash & \Pi B \qquad \forall G \end{array}$$

Rule of Choice. y is not free in $\mathcal{H}$, B, or A.

$$\begin{array}{lll} \mathcal{H} & \vdash & \exists x\; P \\ \mathcal{H}, [\lambda x P]y & \vdash & A \\ \hline \mathcal{H} & \vdash & A \qquad RuleC \end{array} \qquad\qquad \begin{array}{lll} \mathcal{H} & \vdash & \sim\Pi B \\ \mathcal{H}, \sim By & \vdash & A \\ \hline \mathcal{H} & \vdash & A \qquad RuleC \end{array}$$

Existential Generalization.

$$\begin{array}{lll} \mathcal{H} & \vdash & [\lambda x.P]t \\ \hline \mathcal{H} & \vdash & \exists x\; P \qquad \exists G \end{array} \qquad\qquad \begin{array}{lll} \mathcal{H} & \vdash & \sim.Bt \\ \hline \mathcal{H} & \vdash & \sim.\Pi B \qquad \exists G \end{array}$$

Rule of Cases. Here, $\mathcal{H}' \subset \mathcal{H}$.

$$\begin{array}{lll} \mathcal{H}' & \vdash & A \vee B \\ \mathcal{H}', A & \vdash & C \\ \mathcal{H}', B & \vdash & C \\ \hline \mathcal{H} & \vdash & C \qquad Cases \end{array}$$

Deduction Rule.

$$\begin{array}{lll} \mathcal{H}, A_1 & \vdash & A_2 \\ \hline \mathcal{H} & \vdash & A_1 \supset A_2 \qquad Deduct \end{array}$$

Quantifier and Negation Rule. Negations can be moved in or out over quantifiers.

$$\begin{array}{lll} \mathcal{H} & \vdash & \sim\forall x\; P \\ \hline \mathcal{H} & \vdash & \exists x \sim P \qquad RuleQ \end{array} \qquad\qquad \begin{array}{lll} \mathcal{H} & \vdash & \sim\exists x\; P \\ \hline \mathcal{H} & \vdash & \forall x \sim P \qquad RuleQ \end{array}$$

$$\frac{\mathcal{H} \quad \vdash \quad \forall x \sim P}{\mathcal{H} \quad \vdash \quad \sim\exists x\ P \qquad\qquad \mathit{RuleQ}} \qquad\qquad \frac{\mathcal{H} \quad \vdash \quad \exists x \sim P}{\mathcal{H} \quad \vdash \quad \sim\forall x\ P \qquad\qquad \mathit{RuleQ}}$$

We shall often combine the λ rule with the rule of universal instantiation and existential generalization, so that we can directly infer $\underset{\cdot}{\mathbf{S}}^x_t P$ from $\forall x\ P$ and $\exists x\ P$ from $\underset{\cdot}{\mathbf{S}}^x_t P$. ∎

5.3.6. Definition. A *natural deduction proof*, ND-proof for short, is a list of *proof lines*, each of which must follow from zero or more previous proof lines by one of the above rules of inference. A proof line is written as

$$(l) \quad \mathcal{H} \quad \vdash \quad A \qquad\qquad\qquad J : l_1, \ldots, l_p$$

where A is the line's *assertion*, $\mathcal{H}$ is the possibly empty list of *hypotheses* on which the assertion relies, l is the line's *label*, J is its *justification* which is the name of an inference rule, and $l_1, \ldots, l_p$ are the lines used in the inference. We will often use a proof line label to denote its assertion. Hence, a list of hypotheses will generally be written as a set of labels. Since an ND-proof is a list of proof lines, the labels, which are assumed to correspond in a $1-1$ fashion to the proof lines, have an implied order. Hence, all the labels $l_1, \ldots, l_p$ must appear prior to label l. For the convenience of reading examples, we shall list the term t in the justification field when the inference rule used is either universal instantiation or existential generalization.

An *incomplete ND-proof* is an ND-proof in which some lines contain the non-justification *NJ*. Although these lines have no justification, they may be used to justify lines which follow them.

We say that such a list of proof lines is an ND-proof of A, if the last line in the list asserts A and has an empty list of hypotheses.

The variable occurrence x in the Rule of Choice or in Universal Generalization is said to be used *critically* in the ND-proof. We call such variable occurrences *critical* variable occurrences. ∎

5.3.7. Example. The following is an ND-proof of the theorem used in Example 5.2. This particular ND-proof is not very elegant largely since the implication in line (6) is changed into its equivalent disjunctive form. A more appropriate use of this line is to apply *modus ponens* to it and line (4). We have chosen to construct this proof in this fashion because the first algorithm which we will use for constructing ND-proofs from ET-proofs will produces ND-proofs of this kind. Later, in Section 5.6 we will describe another algorithm which will realize that *modus ponens* would have been a better choice.

(1)	1	$\vdash$	$\exists c\, \forall p\, .[\exists u\, .pu] \supset .p.cp$	Hyp
(2)	2	$\vdash$	$\forall x\, \exists y\, .Pxy$	Hyp
(3)	3	$\vdash$	$\forall p\, .[\exists u\, .pu] \supset .p.cp$	Hyp
(4)	2	$\vdash$	$\exists y\, .Pzy$	$\forall I\ : z, 2$
(5)	5	$\vdash$	Pzy	Hyp
(6)	3	$\vdash$	$[\exists u\, .Pzu] \supset .Pz.c.Pz$	$\forall I\ : Pz, 3$
(7)	3	$\vdash$	$[\sim.\exists u\, .Pzu] \vee .Pz.c.Pz$	$RuleP : 6$
(8)	8	$\vdash$	$\sim.\exists u\, .Pzu$	Hyp
(9)	8	$\vdash$	$\sim.Pzy$	$RuleQ, \forall I\ : y, 8$
(10)	$5, 8$	$\vdash$	$Pz.c.Pz$	$RuleP : 5, 9$
(11)	11	$\vdash$	$Pz.c.Pz$	Hyp
(12)	11	$\vdash$	$Pz.c.Pz$	$RuleP : 11$
(13)	$3, 5$	$\vdash$	$Pz.c.Pz$	$Cases : 7, 10, 12$
(14)	$2, 3$	$\vdash$	$Pz.c.Pz$	$RuleC : 4, 13$
(15)	$2, 3$	$\vdash$	$\forall z\, .Pz.c.Pz$	$\forall G\ : 14$
(16)	$2, 3$	$\vdash$	$\exists f\, \forall z\, .Pz.fz$	$\exists G\ : [\lambda v.c.Pv], 15$

(17)	$1, 2$	$\vdash$	$\exists f\, \forall z\, .Pz.fz$	$RuleC : 1, 16$
(18)	1	$\vdash$	$[\forall x\, \exists y\, .Pxy] \supset .\exists f\, \forall z\, .Pz.fz$	$Deduct : 17$
(19)		$\vdash$	$[\exists c\, \forall p\, .[\exists u\, .pu] \supset .p.cp] \supset$ $[\forall x\, \exists y\, .Pxy] \supset \exists f\, \forall z\, .Pz.fz$	$Deduct : 18$

Section 5.4: Outline Transformations

In this section and the next we will show how to convert ET-proofs to natural deduction proofs. This investigation is an immediate extension of the work described by Andrews in [Andrews80]. In that paper Andrews described the key ideas of how to process a natural deduction proof in a top-down and bottom-up fashion under the direction of information stored in a proof structure called a *plan.* Plans, when restricted to FOL, are essentially the same as ET-proofs, although their definition is a bit more awkward. For example, an important part of a plan is called a *replication scheme*, which specifies how often existential quantifiers are instantiated. Use of replication schemes places several inconvenient restrictions on the names of bound variables in a formula before and after the replication is applied. It is also difficult to perform a partial replication on a formula — this being a particularly important operation in the process of constructing natural deduction proofs. Both of these problems are characterized by the fact that replication schemes are defined with respect to global properties of a formula. Expansion trees avoid these problems by being defined with respect to local properties of formulas. What corresponds to partial application of a replication in an ET-proof is the process of eliminating an expansion term from an expansion tree. These inconveniences of plans complicated giving a complete analysis of this transformation process, and Andrews did not give one. We shall show in this section that ET-proofs not only extend this process to HOL but also provide a much more appropriate proof structure upon which to base this transformation.

In the rest of this chapter, references to ET-proofs shall actually be to grounded ET-proofs.

5.4.8. Definition. A *proof outline*, $\mathcal{O}$, is a triple, $\langle L, \Sigma, \{R_l\}\rangle$, where:

(1) L is a list of proof lines which is a complete or incomplete ND-proof. A line with the justification NJ represents a piece of a proof which must be completed. Let L_0 be the set of all lines labels in L which have this justification. These are called the *sponsoring* lines of $\mathcal{O}$.

(2) $\Sigma = \{\Gamma_l \rightarrow l \mid l \in L_0\}$ is a set of sequents, where $\Gamma_l \subset L \setminus L_0$ for each $l \in L_0$. Also, the line labels in Γ_l must precede l. The lines in Γ_l are said to *support* l and are called *supporting* lines. A line is *active* if it is either a supporting line or a sponsoring line which does not assert $\perp$.

(3) $\{R_l\}$ represents a set of list structures, one for each active line. If l is a supporting line, then $\langle \sim R_l, \sim l\rangle \in \mathcal{E}$. If l is a sponsoring line, then $\langle R_l, l\rangle \in \mathcal{E}$.

If Σ is not empty, we define the following formulas and tree structures. For each active line l, set $Q_l := \mathrm{rep}[\![R_l, l]\!]$. Let $\sigma \in \Sigma$. Then σ is the sequent $\Gamma_z \rightarrow z$, for some $z \in L_0$. If z does not assert $\perp$ and Γ_z is not empty then let A_σ be the formula $[\bigvee_{c\in\Gamma_z} \sim c] \vee z$ and let Q_σ be the expansion tree $[\bigvee_{c\in\Gamma_z} \sim Q_c] \vee Q_z$. If z asserts $\perp$ then let A_σ be the formula $\bigvee_{c\in\Gamma_z} \sim c$ and Q_σ be the expansion tree $\bigvee_{c\in\Gamma_z} \sim Q_c$. If Γ_z is empty, then let A_σ be the formula A_z and let Q_σ be the expansion tree Q_z. The following two conditions must also be satisfied by an outline.

(4) If line a supports line z then the hypotheses of a are a subset of the hypotheses of z.

(5) If Σ is not empty, then Q_σ is a (grounded) ET-proof for A^0_σ for each $\sigma \in \Sigma$.

It is easy to show that $\mathcal{O}$ has an active line if and only if Σ is not empty. We say that $\mathcal{O}$ is an outline *for* A if the last line in $\mathcal{O}$ (more precisely, in L) has no hypotheses and asserts A. ∎

5.4.9. Definition. Let A be a formula and R a list representation such that $\mathrm{rep}[\![R, A]\!]$ is an ET-proof for A^0. Let z be the label for the proof line

$$(z) \qquad \vdash \quad A \qquad\qquad\qquad\qquad NJ,$$

and set $L := \{z\}$, $\Sigma := \{\rightarrow z\}$ and $R_z := R$. Then $\mathcal{O}_0 := \langle L, \Sigma, \{R_z\}\rangle$ is clearly an outline. We call this outline the *trivial outline for* A *based on* R. ∎

5.4.10. Example. A proof outline for the theorem in Example 5.7 is given by setting $L = \langle 1, 2, 3, 16, 17, 18, 19\rangle$, $\Sigma = \{2, 3 \rightarrow 16\}$ and

$$R_2 = (\mathrm{EXP} \;\; (z\ (\mathrm{SEL} \;\; y \;\; Pzy)))$$
$$R_3 = (\mathrm{EXP} \;\; (Pz\ (\supset (\mathrm{EXP} \;\; (y\ Pzy))\ Pz.c.Pz)))$$
$$R_{16} = (\mathrm{EXP} \;\; ([\lambda v.c.Pv] \;\;\; (\mathrm{SEL} \;\; z \;\; Pz.c.Pz))),$$

where the lines in L are:

(1)	1	$\vdash$	$\exists c\, \forall p\, .[\exists u\, .pu] \supset .p.cp$	Hyp
(2)	2	$\vdash$	$\forall x\, \exists y\, .Pxy$	Hyp
(3)	3	$\vdash$	$\forall p\, .[\exists u\, .pu] \supset .p.cp$	Hyp
(16)	2, 3	$\vdash$	$\exists f\, \forall z\, .Pz.fz$	NJ
(17)	1, 2	$\vdash$	$\exists f\, \forall z\, .Pz.fz$	$RuleC: 1, 16$
(18)	1	$\vdash$	$[\forall x\, \exists y\, .Pxy] \supset .\exists f\, \forall z\, .Pz.fz$	$Deduct: 17$
(19)		$\vdash$	$[\exists c\, \forall p\, .[\exists u\, .pu] \supset .p.cp] \supset$ $[\forall x\, \exists y\, .Pxy] \supset \exists f\, \forall z\, .Pz.fz$	$Deduct: 18$

It is easy to verify that $\mathrm{rep}[\![{\sim}R_2 \vee {\sim}R_3 \vee R_{16}, {\sim}2 \vee {\sim}3 \vee 16]\!]$ is an ET-proof for ${\sim}2 \vee {\sim}3 \vee 16$ and that $\langle L, \Sigma, \{R_2, R_3, R_{16}\}\rangle$ is an outline. ∎

5.4.11. Definition. A formula t is *admissible in* $\mathcal{O}$ if no free variable in t is selected in R_l for any active line l. In other words, if t is admissible in $\mathcal{O}$ then t is admissible in Q_σ (see Definition 2.34) for all $\sigma \in \Sigma$. ∎

5.4.12. Definition. Below we list 18 transformations. These take an outline, $\mathcal{O} = \langle L, \Sigma, \{R_l\}\rangle$ in which Σ is not empty, and produce a new structure, $\mathcal{O}' = \langle L', \Sigma', \{R'_l\}\rangle$, which we shall soon verify is also an outline. We shall assume that any sequent of the form $\Gamma \rightarrow \bot$ is simply another way to write the sequent $\Gamma \rightarrow$ in which the succedent is empty.

The D- rules will be responsible for simplifying the complexity of support lines, while the P- rules simplify the complexity of sponsoring lines. The two transformations, RuleP1 and RuleP2, will be responsible for giving a justification to a sponsoring line without creating a new sponsoring line. In this case, Σ' result from removing a sequent from Σ.

The transformations below explicitly describe how to compute new members of Σ' and $\{R'_l\}$ from members of Σ and $\{R_l\}$. If a sequent, σ, or active line, l, in $\mathcal{O}$ is unaffected by the transformation, then we assume that $\sigma \in \Sigma'$ and $R'_l = R_l$. The similar description for computing L' from L is given by showing two boxes of proof lines separated by an arrow. The box on the left contains lines present in L, while the box on the right contains lines present in L'. If a line appears in the box on the right but not in the box on the left, we add this new line to L' in the position indicated by the alphabetical ordering of the line labels. If a line appears in both boxes, then its justification has been changed from NJ in L to a new justification in L'. It is always the case that all the lines in L are contained in L'.

If Σ' is not empty, then each sequent $\sigma' \in \Sigma'$ is of two kinds. If no line in σ' was altered or inserted by the transformation, then $\sigma' \in \Sigma$. Otherwise, σ' is *constructed* from a unique $\sigma \in \Sigma$. The D-Disj and P-Conj transformations are the only transformations which will construct two sequents in Σ' from a sequent in Σ. All the other D- and P- transformations will construct one sequent in Σ' from one in Σ.

D-Lambda

Let a be a supporting line with assertion A. If A is not in λ-normal form, let B be a λ-normal form of A. Set $R'_b := R_a$ and construct Σ' by replacing line a with the line b (shown below) in each sequent of Σ. *RuleX* represents any valid justification.

(a)	$\mathcal{H}$	$\vdash$	A	$RuleX$

$\Longrightarrow$

(b)	$\mathcal{H}$	$\vdash$	B	$\lambda : a$

D-Conj

Here a is a supporting line in $\mathcal{O}$. Σ' is the result of replacing a with the lines b, c everywhere in Σ. Since $\langle \sim R_a, \sim . A_1 \wedge A_2 \rangle \in \mathcal{E}$, $R_a = (\wedge\ R_1\ R_2)$, so set $R'_b := R_1$ and $R'_c := R_2$. It may be the case that both lines b and c are really needed to prove all the lines supported by line a. Often, one of these lines may be unnecessary, but to actually determine this requires doing a certain amount of "looking ahead." For now, we must be conservative in giving supports to sponsoring lines, but later, after we introduce matings in Section 5.6, we will be able to determine which supports are truly necessary. This same comment applies equally well to many of the following transformations.

(a)	$\mathcal{H}$	$\vdash$	$A_1 \wedge A_2$	$RuleX$

$\Longrightarrow$

(b)	$\mathcal{H}$	$\vdash$	A_1	$RuleP : a$
(c)	$\mathcal{H}$	$\vdash$	A_2	$RuleP : a$

D-Disj

Let a be a disjunctive support line and let line z be a sponsor for line a. Enter the proof lines shown below. Here $\mathcal{H}' \subset \mathcal{H}$. Also, build Σ' by replacing the sequent $\Gamma_z \rightarrow z$ with the two sequents $\Gamma, b \rightarrow m$ and $\Gamma, n \rightarrow y$, where $\Gamma := \Gamma_z \setminus \{a\}$. Set $R'_m := R_z$ and $R'_y := R_z$. Since $\langle \sim R_a, \sim . A_1 \vee A_2 \rangle \in \mathcal{E}$, $R_a = (\vee\ R_1\ R_2)$. Thus, set $R'_b := R_1$ and $R'_n := R_2$.

(a)	$\mathcal{H}'$	$\vdash$	$A_1 \vee A_2$	$RuleX$
(z)	$\mathcal{H}$	$\vdash$	C	NJ

$\Longrightarrow$

(b)	b	$\vdash$	A_1	Hyp
(m)	$\mathcal{H}, b$	$\vdash$	C	NJ
(n)	n	$\vdash$	A_2	Hyp
(y)	$\mathcal{H}, n$	$\vdash$	C	NJ
(z)	$\mathcal{H}$	$\vdash$	C	$Cases : a, m, y$

D-Imp

With the set of transformations described in this section, we treat implication as if it were an abbreviation of a disjunction. Σ' is the result of replacing a with b in each sequent of Σ. Also, $R'_b := (\vee \sim R_1\ R_2)$ where $R_a = (\supset\ R_1\ R_2)$. Apply D-Disj immediately on line b. In Section 5.6, we will introduce two transformations, D-ModusPonens and D-ModusTollens, which will operate on implicational, support lines without needing to convert them to their equivalent disjunctive form. One of those two rules may not always be applicable, so we will not always be able to avoid using D-Imp.

(a)	$\mathcal{H}$	$\vdash$	$A_1 \supset A_2$	$RuleX$

$\Longrightarrow$

(b)	$\mathcal{H}$	$\vdash$	$\sim A_1 \vee A_2$	$RuleP$

D-All

If a is a universally quantified support line, then R_a has the form (EXP $(t_1\ R_1) \ldots (t_n\ R_n)$). If none of the terms, $t_1, \ldots, t_n$ are admissible in $\mathcal{O}$, then we cannot apply this transformation to a. Otherwise, assume that for some i, such that $1liln$, t_i is admissible in $\mathcal{O}$. Enter line b with $R'_b := R_i$. If $n = 1$, then line a should no longer be active, so we replace every occurrence of a with b in the sequents of Σ to get Σ'. If $n > 1$, then we require that both lines a and b are made active by placing b in the antecedent of every sequent which contains a. In this case, we also change R_a to be the expansion tree in which the subtree R_i is removed.

(a)	$\mathcal{H}$	$\vdash$	$\forall x\ P$	$RuleX$

$\Longrightarrow$

(b)	$\mathcal{H}$	$\vdash$	$[\lambda x . P] t_i$	$\forall I\ : a$

(a)	$\mathcal{H}$	$\vdash$	ΠB	$RuleX$

$\Longrightarrow$

(b)	$\mathcal{H}$	$\vdash$	$B t_i$	$\forall I\ : a$

D-Exists

If a is an existentially quantified support line, then R_a is of the form (SEL y R). Let z be a sponsor of a. Construct Σ' by replacing the sequent $\Gamma_z \rightarrow z$ with $\Gamma_z \setminus \{a\}, b \rightarrow y$. Also, set $R'_y := R_z$ and $R'_b := R$.

$$\begin{array}{|llll r|}\hline (a) & \mathcal{H} & \vdash & \exists x\ P & RuleX \\ (z) & \mathcal{H} & \vdash & C & NJ \\ \hline \end{array} \Longrightarrow \begin{array}{|llll r|}\hline (b) & b & \vdash & [\lambda x P]y & Hyp \\ (y) & \mathcal{H}, b & \vdash & C & NJ \\ (z) & \mathcal{H} & \vdash & C & RuleC: a, y \\ \hline \end{array}$$

$$\begin{array}{|llll r|}\hline (a) & \mathcal{H} & \vdash & \sim.\Pi B & RuleX \\ (z) & \mathcal{H} & \vdash & C & NJ \\ \hline \end{array} \Longrightarrow \begin{array}{|llll r|}\hline (b) & b & \vdash & \sim.By & Hyp \\ (y) & \mathcal{H}, b & \vdash & C & NJ \\ (z) & \mathcal{H} & \vdash & C & RuleC: a, y \\ \hline \end{array}$$

D-NotExists

If a asserts the formula $\sim\exists x\ P$, then R_a is of the form $\sim$(EXP $(t_1\ R_1)\ldots(t_n\ R_n)$). If none of the terms, $t_1, \ldots, t_n$ are admissible in $\mathcal{O}$, then we cannot apply this transformation to a. Otherwise, assume that for some i, such that $1liln$, t_i is admissible in $\mathcal{O}$. Enter line b with $R'_b := \sim R_i$. If $n = 1$, then line a should no longer be active, so we replace every occurrence of a with b in the sequents of Σ to get Σ'. If $n > 1$, then we require that both lines a and b are made to be active by placing b is the antecedent of every sequent which contains a. In this case, we also change R_a to be the expansion tree in which the subtree R_i is removed.

$$\begin{array}{|llll r|}\hline (a) & \mathcal{H} & \vdash & \sim\exists x\ P & RuleX \\ \hline \end{array} \Longrightarrow \begin{array}{|llll r|}\hline (b) & \mathcal{H} & \vdash & [\lambda x \sim P]t_i & RuleQ, \forall I: a \\ \hline \end{array}$$

D-Neg

Apply one of the following five sub-transformations to line a, depending on which one matches the structure of a. Σ' is the result of replacing a with b in each sequent of Σ. R'_b is the result of applying the corresponding negation rule to R_a. In all but the first sub-transformation, we must immediately apply the other transformation indicated.

$$\begin{array}{|llll r|}\hline (a) & \mathcal{H} & \vdash & \sim\sim A & RuleX \\ \hline \end{array} \Longrightarrow \begin{array}{|llll r|}\hline (b) & \mathcal{H} & \vdash & A & RuleP: a \\ \hline \end{array}$$

$$\begin{array}{|llll r|}\hline (a) & \mathcal{H} & \vdash & \sim.A_1 \vee A_2 & RuleX \\ \hline \end{array} \Longrightarrow \begin{array}{|llll r|}\hline (b) & \mathcal{H} & \vdash & \sim A_1 \wedge \sim A_2 & RuleP: a \\ \hline \end{array}$$

Apply D-Conj to line b.

$$\begin{array}{|llll r|}\hline (a) & \mathcal{H} & \vdash & \sim.A_1 \wedge A_2 & RuleX \\ \hline \end{array} \Longrightarrow \begin{array}{|llll r|}\hline (b) & \mathcal{H} & \vdash & \sim A_1 \vee \sim A_2 & RuleP: a \\ \hline \end{array}$$

Apply D-Disj to line b.

$$\begin{array}{|llll r|}\hline (a) & \mathcal{H} & \vdash & \sim.A_1 \supset A_2 & RuleX \\ \hline \end{array} \Longrightarrow \begin{array}{|llll r|}\hline (b) & \mathcal{H} & \vdash & A_1 \wedge \sim A_2 & RuleP: a \\ \hline \end{array}$$

Apply D-Conj to line b.

$$\begin{array}{|llll r|}\hline (a) & \mathcal{H} & \vdash & \sim\forall x\ P & RuleX \\ \hline \end{array} \Longrightarrow \begin{array}{|llll r|}\hline (b) & \mathcal{H} & \vdash & \exists x \sim P & RuleQ: a \\ \hline \end{array}$$

Apply D-Exists to line b.

Each of the P- rules listed below will "process" a sponsoring line z. Let $\Sigma_0 := \Sigma \setminus \{\Gamma_z \rightarrow z\}$.

P-Lambda

If A is not in λ-normal form, let B be a λ-normal form of A. Set $\Sigma' := \Sigma_0 \cup \{\Gamma_z \rightarrow y\}$ and $R'_y := R_z$.

$$\begin{array}{|llll r|}\hline (z) & \mathcal{H} & \vdash & A & NJ \\ \hline \end{array} \Longrightarrow \begin{array}{|llll r|}\hline (y) & \mathcal{H} & \vdash & B & NJ \\ (z) & \mathcal{H} & \vdash & A & \lambda: y \\ \hline \end{array}$$

P-Conj

Set $\Sigma' := \Sigma_0 \cup \{\Gamma_z \rightarrow m, \Gamma_z \rightarrow y\}$. Since $\langle R_z, A_1 \wedge A_2\rangle \in \mathcal{E}$, then $R_z = (\wedge\ R_1\ R_2)$. Set $R'_m := R_1$ and $R'_y := R_2$.

$$\begin{array}{|llll r|}\hline (z) & \mathcal{H} & \vdash & A_1 \wedge A_2 & NJ \\ \hline \end{array} \Longrightarrow \begin{array}{|llll r|}\hline (m) & \mathcal{H} & \vdash & A_1 & NJ \\ (y) & \mathcal{H} & \vdash & A_2 & NJ \\ (z) & \mathcal{H} & \vdash & A_1 \wedge A_2 & RuleP: m, y \\ \hline \end{array}$$

P-Disj1

Set $\Sigma' := \Sigma_0 \cup \{\Gamma_z, a \to x\}$. Since $\langle R_z, A_1 \lor A_2\rangle \in \mathcal{E}$, $R_z = (\lor\ R_1\ A_2)$. Set $R'_a := {\sim}R_1$ and $R'_x := R_2$. We shall introduce a variant of this rule, P-Disj2, later.

$$\begin{array}{|lllll|}\hline (z) & \mathcal{H} & \vdash & A_1 \lor A_2 & NJ \\ \hline \end{array} \quad \Longrightarrow \quad \begin{array}{|lllll|}\hline (a) & a & \vdash & {\sim}A_1 & Hyp \\ (x) & \mathcal{H}, a & \vdash & A_2 & NJ \\ (y) & \mathcal{H} & \vdash & {\sim}A_1 \supset A_2 & Deduct : x \\ (z) & \mathcal{H} & \vdash & A_1 \lor A_2 & RuleP : y \\ \hline \end{array}$$

P-Imp

Set $\Sigma' := \Sigma \cup \{\Gamma_z, a \to y\}$ and $R_a := R_1$, $R_y := R_2$, where $R_z := (\supset\ R_1\ R_2)$.

$$\begin{array}{|lllll|}\hline (z) & \mathcal{H} & \vdash & A_1 \supset A_2 & NJ \\ \hline \end{array} \quad \Longrightarrow \quad \begin{array}{|lllll|}\hline (a) & a & \vdash & A_1 & Hyp \\ (y) & \mathcal{H}, a & \vdash & A_2 & NJ \\ (z) & \mathcal{H} & \vdash & A_1 \supset A_2 & Deduct : y \\ \hline \end{array}$$

P-All

If z is a universally quantified sponsoring line, then $R_z = (\text{SEL}\ \ v\ \ R)$ for some variable v. Set $R'_y := R$ and $\Sigma' := \Sigma_0 \cup \{\Gamma_z \to y\}$.

$$\begin{array}{|lllll|}\hline (z) & \mathcal{H} & \vdash & \forall x\ P & NJ \\ \hline \end{array} \quad \Longrightarrow \quad \begin{array}{|lllll|}\hline (y) & \mathcal{H} & \vdash & [\lambda x P]v & NJ \\ (z) & \mathcal{H} & \vdash & \forall x\ P & \forall G : y \\ \hline \end{array}$$

$$\begin{array}{|lllll|}\hline (z) & \mathcal{H} & \vdash & \Pi B & NJ \\ \hline \end{array} \quad \Longrightarrow \quad \begin{array}{|lllll|}\hline (y) & \mathcal{H} & \vdash & Bv & NJ \\ (z) & \mathcal{H} & \vdash & \Pi B & \forall G : y \\ \hline \end{array}$$

P-Exists

If z is an existentially quantified sponsoring line, then $R_z = (\text{EXP}\ \ (t_1\ R_1) \ldots (t_n\ R_n))$ for some $n \geq 1$ and terms $t_1, \ldots, t_n$. If none of these terms are admissible in $\mathcal{O}$, then we cannot apply P-Exists to line z. Otherwise, we must distinguish two cases. If $n = 1$, then we use $\exists G$ to process line z. In this case, we set $R'_y := R_1$ and $\Sigma' := \Sigma_0 \cup \{\Gamma_z \to y\}$. If $n > 1$, then we use the method of indirect proof to process this line. Let t_i be an admissible term for $\mathcal{O}$. Here, $R'_b := R_i$ and R'_a is the negation of the result of removing R_i from the tree R_z. Also, $R'_y := \perp$ and $\Sigma' := \Sigma_0 \cup \{\Gamma_z, a, b \to y\}$.

$$\begin{array}{|lllll|}\hline (z) & \mathcal{H} & \vdash & \exists x\ P & NJ \\ \hline \end{array} \quad \Longrightarrow \quad \begin{array}{|lllll|}\hline (y) & \mathcal{H} & \vdash & [\lambda x.P]t_1 & NJ \\ (z) & \mathcal{H} & \vdash & \exists x\ P & \exists G : y \\ \text{or} & & & & \\ (a) & a & \vdash & {\sim}\exists x\ P & Hyp \\ (b) & a & \vdash & [\lambda x {\sim} P]t_i & RuleQ, \forall I : a \\ (y) & \mathcal{H}, y & \vdash & \perp & NJ \\ (z) & \mathcal{H} & \vdash & \exists x\ P & IP : y \\ \hline \end{array}$$

(z)	$\mathcal{H}$	$\vdash$	$\sim.\Pi B$	NJ

$\Longrightarrow$

(y)	$\mathcal{H}$	$\vdash$	$\sim.Bt_1$	NJ
(z)	$\mathcal{H}$	$\vdash$	$\sim.\Pi B$	$\exists G : y$
or				
(a)	a	$\vdash$	ΠB	Hyp
(b)	a	$\vdash$	Bt_i	$\forall I\ : a$
(y)	$\mathcal{H}, y$	$\vdash$	$\bot$	NJ
(z)	$\mathcal{H}$	$\vdash$	$\sim.\Pi B$	$IP : y$

P-NotAll

If z asserts $\sim\forall x\ P$, then $R_z = \sim(\text{EXP}\ \ (t_1\ R_1)\ldots(t_n\ R_n))$ for some $n \geq 1$ and terms $t_1, \ldots, t_n$. If none of these terms are admissible in $\mathcal{O}$, then we cannot apply P-NotAll to line z. Otherwise, we must distinguish two cases. If $n = 1$, then we use RuleQ and $\exists G$ to process line z. In this case, we set $R'_y := \sim R_1$ and $\Sigma' := \Sigma_0 \cup \{\Gamma_z \rightarrow y\}$. If $n > 1$, then we use the method of indirect proof to process this line. Let t_i be an admissible term for $\mathcal{O}$. Here, $R'_b := \sim R_i$ and R'_a is $(\text{EXP}\ \ (t_1\ R_1)\ldots(t_{i-1}\ R_{i-1})(t_{i+1}\ R_{i+1})\ldots(t_n\ R_n))$. Also, $R'_y := \bot$ and $\Sigma' := \Sigma_0 \cup \{\Gamma_z, a, b \rightarrow y\}$.

(z)	$\mathcal{H}$	$\vdash$	$\sim\forall x\ P$	NJ

$\Longrightarrow$

(y)	$\mathcal{H}$	$\vdash$	$[\lambda x.\sim P]t_1$	NJ
(z)	$\mathcal{H}$	$\vdash$	$\sim\forall x\ P$	$\exists G, RuleQ : y$
or				
(a)	a	$\vdash$	$\forall x\ P$	Hyp
(b)	a	$\vdash$	$[\lambda x P]t_i$	$\forall I : a$
(y)	$\mathcal{H}, a$	$\vdash$	$\bot$	NJ
(z)	$\mathcal{H}$	$\vdash$	$\sim\forall x\ P$	$RuleP, IP : y$

P-Neg

Apply one of the following five sub-transformations. Set $\Sigma' := \Sigma_0 \cup \{\Gamma_z \rightarrow y\}$. Set R'_y to be the corresponding negation rule applied to R_z. In all but the first sub-transformation, we must immediately apply the other transformation indicated.

(z)	$\mathcal{H}$	$\vdash$	$\sim\sim A$	NJ

$\Longrightarrow$

(y)	$\mathcal{H}$	$\vdash$	A	NJ
(z)	$\mathcal{H}$	$\vdash$	$\sim\sim A$	$RuleP : y$

(z)	$\mathcal{H}$	$\vdash$	$\sim.A_1 \vee A_2$	NJ

$\Longrightarrow$

(y)	$\mathcal{H}$	$\vdash$	$\sim A_1 \wedge \sim A_2$	NJ
(z)	$\mathcal{H}$	$\vdash$	$\sim.A_1 \vee A_2$	$RuleP : y$

Apply P-Conj to line y.

(z)	$\mathcal{H}$	$\vdash$	$\sim.A_1 \wedge A_2$	NJ

$\Longrightarrow$

(y)	$\mathcal{H}$	$\vdash$	$\sim A_1 \vee \sim A_2$	NJ
(z)	$\mathcal{H}$	$\vdash$	$\sim.A_1 \wedge A_2$	$RuleP : y$

Apply P-Disj1 to line y.

(z)	$\mathcal{H}$	$\vdash$	$\sim.A_1 \supset A_2$	NJ

$\Longrightarrow$

(y)	$\mathcal{H}$	$\vdash$	$A_1 \wedge \sim A_2$	NJ
(z)	$\mathcal{H}$	$\vdash$	$\sim.A_1 \supset A_2$	$RuleP : y$

Apply P-Conj to line y.

(z)	$\mathcal{H}$	$\vdash$	$\sim\exists x\ P$	NJ

$\Longrightarrow$

(y)	$\mathcal{H}$	$\vdash$	$\forall x \sim P$	NJ
(z)	$\mathcal{H}$	$\vdash$	$\sim\exists x\ P$	$RuleQ : y$

Apply P-All to line y.

RuleP1
If A_1 and A_2 are complementary, λ-normal signed atoms, then we have proved line z indirectly. Here $\mathcal{H}_1 \cup \mathcal{H}_2 \subset \mathcal{H}$. Set $\Sigma' := \Sigma_0$.

$$\boxed{\begin{array}{lllll}(a_1) & \mathcal{H}_1 & \vdash & A_1 & \mathit{RuleX} \\ (a_2) & \mathcal{H}_2 & \vdash & A_2 & \mathit{RuleY} \\ (z) & \mathcal{H} & \vdash & A & NJ\end{array}} \implies \boxed{\begin{array}{lllll}(z) & \mathcal{H} & \vdash & A & \mathit{RuleP}: a_1, a_2\end{array}}$$

RuleP2
If A_1 and A_2 are λ-normal, signed atoms such that A_1 conv-I A_2, then we have proved line z directly. Here $\mathcal{H}_1 \subset \mathcal{H}_2$. Set $\Sigma' := \Sigma_0$.

$$\boxed{\begin{array}{lllll}(a) & \mathcal{H}_1 & \vdash & A_1 & \mathit{RuleX} \\ (z) & \mathcal{H}_2 & \vdash & A_2 & NJ\end{array}} \implies \boxed{\begin{array}{lllll}(z) & \mathcal{H}_2 & \vdash & A_2 & \mathit{RuleP}: a\end{array}}$$

Notice that after a D- transformation is applied, the line a may or may not still be active, while after a P- transformation is applied, the line z is no longer active. ∎

The definition of these transformations may look more complicated then they need to be, and in a sense, that is the case. For example, the D-Neg and P-Neg transformations are presented here as a composite transformation, *i.e.* in most cases, when one of these transformations is applied, it is immediately followed by the application of another transformation. This is done in this setting since we will show later (Proposition 5.15) that each transformation must eliminate one of the connectives $\wedge$, $\vee$, $\supset$, or quantifiers $\forall$ and $\exists$, so that a corresponding introduction can be made in a parallel LKH-proof figure. The same comments apply also to P-NotAll and D-NotExists. These two transformations could be simplified to P-Exists and D-All if we added the obvious sub-transformation to both D-Neg and P-Neg. We have avoided doing this since Proposition 5.15 would not be provable. In an implementation of these transformations in which we only desire to build ND-proofs, these transformations can be simplified.

For the rest of this chapter, we shall assume that $\mathcal{O}$ and $\mathcal{O}'$ always refer to the outlines $\langle L, \Sigma, \{R_l\}\rangle$ and $\langle L', \Sigma', \{R'_l\}\rangle$, respectively.

5.4.13. **Lemma.** *Let $\mathcal{O}'$ be the result of applying P-All, P-Exists, P-NotAll, D-All, D-Exists, or D-NotExists to $\mathcal{O}$, let $\sigma' \in \Sigma'$ which is not in Σ, and let $\sigma \in \Sigma$ be the sequent from which σ' is constructed. Then $Q_{\sigma'}$ is the result of eliminating a top-level instantiated node of Q_σ (see Definition 2.34), modulo adding or dropping double negations from the top-level boolean structure of $Q_{\sigma'}$.*

Proof. Of the six cases to consider, we shall show the case where the transformation applied is P-Exists since this is the hardest case. The others follow similarly. Let lines a, b, y, z be as in the definition of P-Exists, and let $R_z = (\text{EXP}\ \ (t_1\ R_1) \ldots (t_n\ R_n))$ for some $n \geq 1$. Then $Q_z = \text{rep}[\![R_z, z]\!] = {\sim}(\Pi[\lambda x{\sim}P^0] +^{t_1^0} {\sim}Q_1 + \ldots +^{t_n^0} {\sim}Q_n)$, where $Q_i := \text{rep}[\![R_i, \rho([\lambda x P]t_i)]\!]$ for $i = 1, \ldots, n$. σ must be the sequent $\Gamma_z \to z$. $Q_\sigma = [\bigvee_{c \in \Gamma_z} {\sim}Q_c] \vee Q_z$.

Now we consider two cases. If $n = 1$, then y replaces z in σ, so σ' is equal to the sequent $\Gamma_z \to y$. Now $Q_y = \text{rep}[\![R_1, y]\!] = \text{rep}[\![R_1, \rho([\lambda x P]t_1)]\!] = Q_1$. Hence, $Q_{\sigma'}$ is the result of replacing the subtree ${\sim}(\Pi[\lambda x{\sim}P^0] +^{t_1^0} {\sim}Q_1)$ with Q_1 while, if we were to eliminate the existential node, we would be replacing it with ${\sim}{\sim}Q_1$.

Now assume that $n > 1$. Then the sequent σ' is equal to $\Gamma_z, a, b \to$. But

$$\begin{aligned} Q_a &= \text{rep}[\![{\sim}(\text{EXP}\ \ (t_1\ R_1) \ldots (t_{i-1}\ R_{i-1})(t_{i+1}\ R_{i+1}) \ldots (t_n\ R_n)), {\sim}\exists x\ P]\!] \\ &= {\sim}{\sim}(\Pi[\lambda x{\sim}P^0] +^{t_1^0} {\sim}Q_1 + \ldots +^{t_{i-1}^0} {\sim}Q_{i-1} +^{t_{i+1}^0} {\sim}Q_{i+1} + \ldots +^{t_n^0} {\sim}Q_n) \end{aligned}$$

and $Q_b = \sim Q_i$. Hence, $Q_{\sigma'}$ is the result of replacing the subtree $\sim(\Pi[\lambda x \sim P^0] +^{t_1^0} \sim Q_1 + \ldots +^{t_n^0} \sim Q_n)$ with $\sim Q_a \vee \sim Q_b$, while if we eliminate the i^{th} descendant of this existential node, we would get

$$\sim[(\Pi[\lambda x \sim P^0] +^{t_1^0} \sim Q_1 + \ldots +^{t_{i-1}^0} \sim Q_{i-1} +^{t_{i+1}^0} \sim Q_{i+1} + \ldots +^{t_n^0} \sim Q_n) \wedge \sim Q_i].$$

Modulo double negations, this is the same as $\sim Q_a \vee \sim Q_b$. **Q.E.D.**

5.4.14. **Proposition.** *Let $\mathcal{O}$ be an outline, and let $\mathcal{O}'$ be the result of applying one of the transformations described in Definition 5.12. Then $\mathcal{O}'$ is an outline.*

Proof. It is straightforward to verify that conditions (1)–(4) of the definition for outlines are satisfied by $\mathcal{O}'$ no matter which transformation is applied. We need to clearly examine condition (5). If the transformation which was applied was either RuleP1 or RuleP2, then this condition is trivial to check since $\Sigma' \subset \Sigma$. So assume that the transformation applied was other than these two. Let $\sigma' \in \Sigma'$. If $\sigma' \in \Sigma$ then $Q_{\sigma'}$ is clearly an ET-proof for $A_{\sigma'}$. Thus, assume that $\sigma' \notin \Sigma$ and let $\sigma \in \Sigma$ be the sequent from which σ' was constructed. We must verify that $Q_{\sigma'}$ is an ET-proof for $A_{\sigma'}$.

We must first show that $Q_{\sigma'}$ is an expansion tree *for* $A_{\sigma'}$. Since Q_z is an expansion tree for z when z is a sponsoring line, and $\sim Q_a$ is an expansion tree for $\sim a$ when a is a supporting line, we must show that the combination of these trees in $Q_{\sigma'}$ is an expansion tree for the combination of these assertions in $A_{\sigma'}$. This means showing that $Q_{\sigma'}$ is a sound expansion tree. First remember that Q_σ is sound. In the case of all the transformations other than D-All, D-Exists, D-NotExists, P-All, P-Exists, and P-NotAll, the free variables in $Sh(Q_\sigma)$ and the selected variables in Q_σ do not change in $Sh(Q_{\sigma'})$ and $Q_{\sigma'}$, respectively. Thus, in this case, $Q_{\sigma'}$ must be sound. In the case of one of these six transformations, $Q_{\sigma'}$ is essentially the elimination of a top-level instantiated node (by the above lemma). Hence, by Proposition 2.35 (4), $Q_{\sigma'}$ is sound. Note that adding or dropping double negations does not change the soundness of an expansion tree. Thus, $Q_{\sigma'}$ is sound and, therefore, an expansion tree for $A_{\sigma'}$.

Next we must verify that $<_{Q_{\sigma'}}$ is acyclic, given that $<_{Q_\sigma}$ is acyclic. In the case that the transformation applied is other than D-All, D-Exists, D-NotExists, P-All, P-Exists, or P-NotAll, the relative dominance of existential and universal nodes in $Q_{\sigma'}$ is the same as it is in Q_σ. Hence, $<_{Q_{\sigma'}}$ is acyclic. If the transformation applied was one of the above six, then, by the above lemma and Proposition 2.35 (2), we know that $<_{Q_{\sigma'}}$ is acyclic.

Finally, we need to show that $Fm(Q_{\sigma'})$ is tautologous. First assume that the transformation applied is either D-All, D-Exists, D-NotExists, P-All, P-Exists, or P-NotAll. Then, again by the above lemma and Proposition 2.35 (3), since $Fm(Q_\sigma)$ is tautologous, then so is $Fm(Q_{\sigma'})$. Notice that adding or dropping double negations to an expansion tree corresponds to the same operation in the tree's Fm value. If the transformation was D-Lambda or P-Lambda, then $Q_\sigma = Q_{\sigma'}$ and the result follows trivially. If the transformation was either D-Neg or P-Neg, then we either drop a double negation, in which case the result is immediate, or we reduce this case to another transformation. If the transformation was D-Imp, then we essentially reduce this problem to one for D-Disj. Hence, we only have to consider the following five cases.

Case D-Conj: Here σ is $\Gamma, a \rightarrow z$ and σ' is $\Gamma, b, c \rightarrow z$ where $\Gamma := \Gamma_z \setminus \{a\}$. Now $Q_{\sigma'}$ is the expansion tree $[\bigvee_{c \in \Gamma} \sim Q_c] \vee \sim Q'_b \vee \sim Q'_c \vee Q_z$ while Q_σ is the expansion tree $[\bigvee_{c \in \Gamma} \sim Q_c] \vee \sim[Q'_b \wedge Q'_c] \vee Q_z$. Hence, $Fm(Q_\sigma) \equiv Fm(Q_{\sigma'})$ and $Fm(Q_{\mathcal{O}'})$ is tautologous.

Case P-Disj1: Here σ is $\Gamma_z \rightarrow z$ and σ' is $\Gamma_z, a \rightarrow x$. $Q_{\sigma'}$ is the expansion tree $[\bigvee_{c \in \Gamma_z} \sim Q_c] \vee \sim\sim Q'_a \vee Q'_x$ while Q_σ is the expansion tree $[\bigvee_{c \in \Gamma_z} \sim Q_c] \vee [Q'_a \vee Q'_x]$. Hence, $Fm(Q_{\sigma'}) \equiv Fm(Q_\sigma)$ and $Fm(Q_{\mathcal{O}'})$ is tautologous.

Case P-Imp: Here σ is $\Gamma_z \to z$ and σ' is $\Gamma_z, a \to y$. $Q_{\sigma'}$ is the expansion tree $[\bigvee_{c \in \Gamma_z} {\sim} Q_c] \vee {\sim} Q'_a \vee Q'_y$ while Q_σ is the expansion tree $[\bigvee_{c \in \Gamma_z} {\sim} Q_c] \vee [Q'_a \supset Q'_x]$. Hence, $Fm(Q_{\sigma'}) \equiv Fm(Q_\sigma)$ and $Fm(Q_{\mathcal{O}'})$ is tautologous.

Case P-Conj: Here σ is $\Gamma_z \to z$ and σ' is either $\Gamma_z \to m$ or $\Gamma_z \to y$. $Q_{\sigma'}$ is either the expansion tree $[\bigvee_{c \in \Gamma_z} {\sim} Q_c] \vee Q'_m$ or $[\bigvee_{c \in \Gamma_z} {\sim} Q_c] \vee Q'_y$ while Q_σ is the expansion tree $[\bigvee_{c \in \Gamma_z} {\sim} Q_c] \vee [Q'_m \wedge Q'_y]$. Hence, $Fm(Q_\sigma)$ truth-functionally implies $Fm(Q_{\sigma'})$ which must, therefore, be tautologous.

Case D-Disj: Here, σ is $\Gamma, a \to z$ and σ' is either $\Gamma, b \to z$ or $\Gamma, n \to z$, where $\Gamma := \Gamma_z \setminus \{a\}$. Now Q_σ is $[\bigvee_{c \in \Gamma} {\sim} Q_c] \vee {\sim}[Q'_b \vee Q'_n] \vee Q_z$ while $Q_{\sigma'}$ is either $[[\bigvee_{c \in \Gamma} {\sim} Q_c] \vee {\sim} Q'_b \vee Q_z]$ or $[[\bigvee_{c \in \Gamma} {\sim} Q_c] \vee {\sim} Q'_n \vee Q_z]$. Hence, $Fm(Q_\sigma)$ truth-functionally implies $Fm(Q_{\sigma'})$, which must, therefore, be tautologous. **Q.E.D.**

5.4.15. **Proposition.** *Let $\mathcal{O}'$ be the result of applying one of the transformations in Definition 5.12 to the outline $\mathcal{O}$. Let Σ and Σ' be the sequent sets associated with $\mathcal{O}$ and $\mathcal{O}'$. If each sequent in Σ' has a cut-free LKH-proof, then each sequent in Σ has a cut-free LKH-proof.*

Proof. If the transformation which was applied was either RuleP1 or RuleP2, then $\Sigma' \subset \Sigma$ and the result is immediate. In the cases where a D- or P- transformation was applied, either one or two sequents in Σ' are constructed from a sequent in Σ. (More than one or two sequents in Σ' may have been constructed, however, from the application of some transformations.) Below we show how to combine cut-free LKH-proofs for those one or two sequents to give a cut-free LKH-proof of the original sequent in Σ. Let C denote the formula asserted by a line supported by a. We shall not specify when the inference rule of interchange is used, since it will be easy for the reader to insert them in the inference figure where they are required.

Case D-Lambda:

$$\frac{\Gamma, B \longrightarrow C}{\Gamma, A \longrightarrow C}\ \lambda$$

Case D-Conj:

$$\frac{\dfrac{\dfrac{\Gamma, A_1, A_2 \longrightarrow C}{\Gamma, A_1 \wedge A_2, A_2 \longrightarrow C}\ \wedge\text{–IA}}{\Gamma, A_1 \wedge A_2, A_1 \wedge A_2 \longrightarrow C}\ \wedge\text{–IA}}{\Gamma, A_1 \wedge A_2 \longrightarrow C}\ \text{Contraction}$$

Case D-Disj:

$$\frac{\Gamma, A_1 \longrightarrow C \qquad\qquad \Gamma, A_2 \longrightarrow C}{\Gamma, A_1 \vee A_2 \longrightarrow C}\ \vee\text{–IA}$$

Case D-Imp:

$$\frac{\dfrac{\Gamma, {\sim} A_1 \longrightarrow C}{\Gamma \longrightarrow C, A_1}\ {\sim}\text{–EA} \qquad\qquad \Gamma, A_2 \longrightarrow C}{A_1 \supset A_2, \Gamma \longrightarrow C}\ \supset\text{–IA}$$

Case D-All: First assume that line a asserts $\forall x\, P$. If $n > 1$, then

$$\frac{\dfrac{\Gamma, \forall x\, P, [\lambda x.P]t_i \longrightarrow C}{\Gamma, \forall x\, P, \forall x\, P \longrightarrow C}\ \forall\text{–IA}}{\Gamma, \forall x\, P \longrightarrow C}\ \text{Contraction}$$

If $n = 1$, the the figure is simply

$$\frac{\Gamma, [\lambda x.P]t_1 \longrightarrow C}{\Gamma, \forall x\, P \longrightarrow C}\ \forall\text{−IA}$$

If line a asserts ΠB, then the above two figures are repeated, except that Π−IA is used instead of $\forall$−IA.

Case D-Exists: If line a asserts $\exists x\, P$, then the figure is

$$\frac{\Gamma, [\lambda x P]y \longrightarrow C}{\Gamma, \exists x\, P \longrightarrow C}\ \exists\text{−IA}$$

If line a asserts $\sim\Pi B$, then the figure is

$$\cfrac{\cfrac{\cfrac{\Gamma, \sim By \longrightarrow C}{\Gamma \longrightarrow C, By}\ \sim\text{−EA}}{\Gamma \longrightarrow C, \Pi B}\ \Pi\text{−IS}}{\Gamma, \sim\Pi B \longrightarrow C}\ \sim\text{−IA}$$

The proviso on the eigenvariable y is met since Q_σ is a sound expansion tree for A_σ, *i.e.* since $y \in \mathbf{S}_{Q_\sigma}$, y is not free in A_σ and, therefore, not in P, C, B, or any formula in Γ.

Case D-NotExists:

If $n = 1$ then

$$\cfrac{\cfrac{\cfrac{\cfrac{\Gamma, [\lambda x{\sim}P]t_1 \longrightarrow C}{\Gamma, \sim[\lambda x P]t_1 \longrightarrow C}\ \lambda}{\Gamma \longrightarrow C, [\lambda x P]t_1}\ \sim\text{−EA}}{\Gamma \longrightarrow C, \exists x\, P}\ \exists\text{−IS}}{\Gamma, \sim\exists x\, P \longrightarrow C}\ \sim\text{−IA}$$

If $n > 1$ then

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\Gamma, \sim\exists x\, P, [\lambda x{\sim}P]t_i \longrightarrow C}{\Gamma, \sim\exists x\, P, \sim[\lambda x P]t_i \longrightarrow C}\ \lambda}{\Gamma, \sim\exists x\, P \longrightarrow C, [\lambda x P]t_i}\ \sim\text{−EA}}{\Gamma, \sim\exists x\, P \longrightarrow C, \exists x\, P}\ \exists\text{−IS}}{\Gamma, \sim\exists x\, P, \sim\exists x\, P \longrightarrow C}\ \sim\text{−IA}}{\Gamma, \sim\exists x\, P \longrightarrow C}\ \text{Contraction}$$

Case D-Neg:

If a asserts $\sim\sim A$ then

$$\dfrac{\dfrac{\Gamma, A \longrightarrow C}{\Gamma \longrightarrow C, \sim A}\ \sim\!-\text{IS}}{\Gamma, \sim\sim A \longrightarrow C}\ \sim\!-\text{IA}$$

If a asserts $\sim .A_1 \vee A_2$ then

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma, \sim A_1, \sim A_2 \longrightarrow C}{\Gamma, \sim A_2 \longrightarrow C, A_1}\ \sim\!-\text{EA}}{\Gamma \longrightarrow C, A_1, A_2}\ \sim\!-\text{EA}}{\Gamma \longrightarrow C, A_1 \vee A_2, A_2}\ \vee\!-\text{IS}}{\Gamma \longrightarrow C, A_1 \vee A_2, A_1 \vee A_2}\ \vee\!-\text{IS}}{\Gamma \longrightarrow C, A_1 \vee A_2}\ \text{Contraction}}{\Gamma, \sim .A_1 \vee A_2 \longrightarrow C}\ \sim\!-\text{IA}$$

If a asserts $\sim .A_1 \wedge A_2$ then

$$\dfrac{\dfrac{\dfrac{\Gamma, \sim A_1 \longrightarrow C}{\Gamma \longrightarrow C, A_1}\ \sim\!-\text{EA} \qquad \dfrac{\Gamma, \sim A_2 \longrightarrow C}{\Gamma \longrightarrow C, A_2}\ \sim\!-\text{EA}}{\Gamma \longrightarrow C, A_1 \wedge A_2}\ \wedge\!-\text{IA}}{\Gamma, \sim .A_1 \wedge A_2 \longrightarrow C}\ \sim\!-\text{IA}$$

If a asserts $\sim .A_1 \supset A_2$ then

$$\dfrac{\dfrac{\dfrac{\Gamma, A_1, \sim A_2 \longrightarrow C}{\Gamma, A_1 \longrightarrow C, A_2}\ \sim\!-\text{EA}}{\Gamma \longrightarrow C, A_1 \supset A_2}\ \supset\!-\text{IS}}{\Gamma, \sim .A_1 \supset A_2 \longrightarrow C}\ \sim\!-\text{IA}$$

If a asserts $\sim\forall x\ P$ then

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma, [\lambda x \sim P]y \longrightarrow C}{\Gamma, \sim[\lambda x P]y \longrightarrow C}\ \lambda}{\Gamma \longrightarrow C, [\lambda x P]y}\ \sim\!-\text{EA}}{\Gamma \longrightarrow C, \forall x\ P}\ \forall\!-\text{IS}}{\Gamma, \sim\forall x\ P \longrightarrow C}\ \sim\!-\text{IA}$$

As in the D-Exists case, the proviso on the eigenvariable is meet.

Case P-Lambda:

$$\frac{\Gamma \longrightarrow B}{\Gamma \longrightarrow A}\ \lambda$$

Case P-Conj:

$$\frac{\Gamma \longrightarrow A_1 \qquad\qquad \Gamma \longrightarrow A_2}{\Gamma \longrightarrow A_1 \wedge A_2}\ \wedge\text{−IA}$$

Case P-Disj1:

$$\cfrac{\cfrac{\cfrac{\Gamma, \sim A_1 \longrightarrow A_2}{\Gamma \longrightarrow A_1, A_2}\ \sim\text{−EA}}{\Gamma \longrightarrow A_1 \vee A_2, A_1 \vee A_2}\ \vee\text{−IS, twice}}{\Gamma \longrightarrow A_1 \vee A_2}\ \text{Contraction}$$

Case P-Imp:

$$\frac{\Gamma, A_1 \longrightarrow A_2}{\Gamma \longrightarrow A_1 \supset A_2}\ \supset\text{−IS}$$

Case P-All:

If a asserts $\forall x\ P$, then

$$\frac{\Gamma \longrightarrow [\lambda x P]y}{\Gamma \longrightarrow \forall x\ P}\ \forall\text{−IS}$$

If a asserts ΠB, then repeat the above figure using Π−IS. As in the D-Exists case, the proviso on the eigenvariable y is meet.

Case P-Exists:

If $n = 1$ and a asserts $\exists x\ P$, then

$$\frac{\Gamma \longrightarrow [\lambda x P]t_1}{\Gamma \longrightarrow \exists x\ P}\ \exists\text{−IS}$$

If $n > 1$ and a asserts $\exists x\ P$, then

$$\cfrac{\cfrac{\cfrac{\Gamma, \sim\exists x\ P, \sim[\lambda x P]t_i \longrightarrow}{\Gamma \longrightarrow \exists x\ P, [\lambda x P]t_i}\ \sim\text{−EA twice}}{\Gamma \longrightarrow \exists x\ P, \exists x\ P}\ \exists\text{−IS}}{\Gamma \longrightarrow \exists x\ P}\ \text{Contraction}$$

If $n = 1$ and a asserts ΠB, then

$$\cfrac{\cfrac{\cfrac{\Gamma \longrightarrow \sim B t_1}{\Gamma, B t_1 \longrightarrow}\ \sim\text{−ES}}{\Gamma, \Pi B \longrightarrow}\ \Pi\text{−IA}}{\Gamma \longrightarrow \sim\Pi B}\ \sim\text{−IS}$$

If $n > 1$ and a asserts ΠB, then

$$\dfrac{\dfrac{\dfrac{\Gamma, \Pi B, Bt_i \;\longrightarrow}{\Gamma, \Pi B, \Pi B \;\longrightarrow}\;\Pi{-}\mathrm{IA}}{\Gamma, \Pi B \;\longrightarrow}\;\text{Contraction}}{\Gamma \;\longrightarrow\; {\sim}\Pi B}\;{\sim}{-}\mathrm{IS}$$

Case P-NotAll:

If $n = 1$ then

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma, [\lambda x{\sim}P]t_1 \;\longrightarrow\; C}{\Gamma, {\sim}[\lambda x P]t_1 \;\longrightarrow\; C}\;\lambda}{\Gamma \;\longrightarrow\; C, [\lambda x P]t_1}\;{\sim}{-}\mathrm{EA}}{\Gamma \;\longrightarrow\; C, \exists x\, P}\;\exists{-}\mathrm{IS}}{\Gamma, {\sim}\exists x\, P \;\longrightarrow\; C}\;{\sim}{-}\mathrm{IA}$$

If $n > 1$ then

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma, {\sim}\exists x\, P, [\lambda x{\sim}P]t_1 \;\longrightarrow\; C}{\Gamma, {\sim}\exists x\, P, {\sim}[\lambda x P]t_1 \;\longrightarrow\; C}\;\lambda}{\Gamma, {\sim}\exists x\, P \;\longrightarrow\; C, [\lambda x P]t_1}\;{\sim}{-}\mathrm{EA}}{\Gamma, {\sim}\exists x\, P \;\longrightarrow\; C, \exists x\, P}\;\exists{-}\mathrm{IS}}{\Gamma, {\sim}\exists x\, P, {\sim}\exists x\, P \;\longrightarrow\; C}\;{\sim}{-}\mathrm{IA}}{\Gamma, {\sim}\exists x\, P \;\longrightarrow\; C}\;\text{Contraction}$$

Case P-Neg:

If z asserts ${\sim}{\sim}A$, then

$$\dfrac{\dfrac{\Gamma \;\longrightarrow\; A}{\Gamma, {\sim}A \;\longrightarrow}\;{\sim}{-}\mathrm{IA}}{\Gamma \;\longrightarrow\; {\sim}{\sim}A}\;{\sim}{-}\mathrm{IS}$$

If z asserts ${\sim}.A_1 \vee A_2$ then

$$\dfrac{\dfrac{\dfrac{\Gamma \;\longrightarrow\; {\sim}A_1}{\Gamma, A_1 \;\longrightarrow}\;{\sim}{-}\mathrm{ES} \qquad \dfrac{\Gamma \;\longrightarrow\; {\sim}A_2}{\Gamma, A_2 \;\longrightarrow}\;{\sim}{-}\mathrm{ES}}{\Gamma, A_1 \vee A_2 \;\longrightarrow}\;\vee{-}\mathrm{IA}}{\Gamma \;\longrightarrow\; {\sim}.A_1 \vee A_2}\;{\sim}{-}\mathrm{IS}$$

If z asserts ${\sim}.A_1 \wedge A_2$ then

$$\begin{array}{cl}
\Gamma, {\sim}{\sim}A_1 \longrightarrow {\sim}A_2 & \\ \cline{1-1}
\Gamma \longrightarrow {\sim}A_2, {\sim}A_1 & {\sim}{-}\mathrm{EA} \\ \cline{1-1}
\Gamma, A_1 \longrightarrow {\sim}A_2 & {\sim}{-}\mathrm{ES} \\ \cline{1-1}
\Gamma, A_1, A_2 \longrightarrow & {\sim}{-}\mathrm{ES} \\ \cline{1-1}
\Gamma, A_1 \wedge A_2, A_1 \wedge A_2 \longrightarrow & \wedge{-}\mathrm{IA}\ \text{twice} \\ \cline{1-1}
\Gamma, A_1 \wedge A_2 \longrightarrow & \text{Contraction} \\ \cline{1-1}
\Gamma \longrightarrow {\sim}.A_1 \wedge A_2 & {\sim}{-}\mathrm{IS}
\end{array}$$

If z asserts ${\sim}.A_1 \supset A_2$ then

$$\dfrac{\Gamma \longrightarrow A_1 \qquad \dfrac{\Gamma \longrightarrow {\sim}A_2}{\Gamma, A_2 \longrightarrow}\ {\sim}{-}\mathrm{ES}}{\Gamma, \Gamma, A_1 \supset A_2 \longrightarrow}\ \supset{-}\mathrm{IA}$$

$$\begin{array}{cl}
\Gamma, \Gamma, A_1 \supset A_2 \longrightarrow & \\ \cline{1-1}
\Gamma, A_1 \supset A_2 \longrightarrow & \text{several Contractions} \\ \cline{1-1}
\Gamma \longrightarrow {\sim}.A_1 \supset A_2 & {\sim}{-}\mathrm{IA}
\end{array}$$

If z asserts ${\sim}\exists x\ P$ then

$$\begin{array}{cl}
\Gamma \longrightarrow [\lambda x {\sim} P]y & \\ \cline{1-1}
\Gamma \longrightarrow {\sim}[\lambda x P]y & \lambda \\ \cline{1-1}
\Gamma, [\lambda x P]y \longrightarrow & {\sim}{-}\mathrm{ES} \\ \cline{1-1}
\Gamma, \exists x\ P \longrightarrow & \exists{-}\mathrm{IA} \\ \cline{1-1}
\Gamma \longrightarrow {\sim}\exists x\ P & {\sim}{-}\mathrm{IS}
\end{array}$$

Q.E.D.

Section 5.5: Naive Construction of Proof Outlines

In this section, we will present an algorithm which will non-deterministically select outline transformations to be applied to a given outline. If the resulting outline contains active lines, this selection process is repeated. Hence, the final outline produced by this algorithm will contain no active lines and will, therefore, represent an ND-proof. Before we present the algorithm, we first prove two propositions which will guarantee that the selection of transformations is possible at various steps in the algorithm.

5.5.16. **Proposition.** *If $\mathcal{O}$ is an outline which contains an active line which does not assert a λ-normal, signed atom, then some D- or P- transformation can be applied to $\mathcal{O}$.*

Proof. Let $\mathcal{O}$ be an outline and let l be an active line of $\mathcal{O}$ which does not assert a λ-normal, signed atom. If line l asserts a formula which is not in λ-normal form, apply either D-Lambda or P-Lambda (depending on whether or not l is supporting or sponsoring). If l asserts a formula which is a conjunction, disjunction, implication or the negation of such a line, then apply either D-Conj, P-Conj, D-Disj, P-Disj1, D-Imp, P-Imp, or D-Neg, P-Neg to line l. If the assertion of l is a double negation, then apply either D-Neg or P-Neg. If l is a supporting line which asserts a formula of the form $\exists x\ P$ or $\sim\Pi B$, apply D-Exists, or if it is in the form $\sim\forall x\ P$ then apply D-Neg. If l is a sponsoring line which asserts a formula of the form $\forall x\ P$ or ΠB apply P-All, or if it asserts a formula of the form $\sim\exists x\ P$ then apply P-Neg. None of the transformations mentioned above have any provisos attached to them, so they may be applied whenever an appropriate active line is present.

Now assume that the only active lines assert either signed atoms or are supporting lines asserting formulas of the form $\forall x\ P$, ΠB, or $\sim\exists x\ P$ or are sponsoring lines asserting formulas of the form $\exists x\ P$, $\sim\Pi x$, or $\sim\forall x\ P$. Let $\sigma \in \Sigma$ be such that it contains an active line with such a quantified assertion (such as l). Thus Q_σ contains top-level, existential nodes. Since Q_σ is grounded, these existential nodes are instantiated. Also, since Q_σ is an ET-proof for A_σ, $<_{Q_\sigma}$ is acyclic, and by Proposition 2.36, one of these nodes can be eliminated. But this means that the proviso for D-All, P-Exists, D-NotExists, or P-NotAll concerning the admissibility of an expansion term can be meet. Thus one of these four transformations must be applicable. **Q.E.D.**

5.5.17. **Proposition.** *If $\mathcal{O}$ is an outline in which all the active lines assert λ-normal, signed atoms, then either RuleP1 or RuleP2 can be applied to σ, for each $\sigma \in \Sigma$.*

Proof. Assume that there is some sequent $\sigma \in \Sigma$ for which RuleP1 or RuleP2 does not apply. Let σ be given by the sequent $l_1, \ldots, l_m \longrightarrow l$ for active lines $l_1, \ldots, l_m, l$. (If l asserts $\perp$, the following argument is simplified.) $Fm(Q_\sigma)$ is then $\sim l_1 \vee \ldots \vee \sim l_m \vee l$. Now none of these disjuncts are complementary, since, if $\sim l_i$ and $\sim l_j$ were complementary (for some i, j such that $1li, jlm$) then l_i and l_j would be complementary and RuleP1 is applicable, and if $\sim l_i$ and l were complementary, then l_i conv-I l and RuleP2 is applicable. But this implies that $Fm(Q_\sigma)$ cannot be tautologous, and this contradicts the fact that Q_σ is an ET-proof for A_σ. Hence, either RuleP1 or RuleP2 must be applicable to σ. **Q.E.D.**

5.5.18. Algorithm. Apply transformations to the initial outline $\mathcal{O}_0$, stopping when there are no active lines left to process. The resulting outline $\mathcal{O}$ is an ND-proof of the same formula for which $\mathcal{O}_0$ was an outline.

(1) *Initialize:* Set $\mathcal{O} := \mathcal{O}_0$.

(2) *λ-normalize:* If any active line in $\mathcal{O}$ is not in λ-normal form, apply either P- or D-Lambda to it, and repeat step 2. Otherwise, do step 3.

(3) *Remove top-level double negations:* If any active line is a top-level double negation, then apply either P- or D-Neg to that line and repeat step 3. Otherwise, do step 4.

(4) If all active lines of $\mathcal{O}$ are λ-normal, signed atoms, then do step (6), otherwise do step (5).

(5) By Proposition 5.16, some D- or P- transformation can be applied to $\mathcal{O}$. Set $\mathcal{O}$ to the result of applying any such transformation, and then do step 2.

(6) By Proposition 5.17, either RuleP1 or RuleP2 can be applied to each of the sequents in $\mathcal{O}$. Set $\mathcal{O}$ to be the result of applying one of these transformations to each of the sequents. $\mathcal{O}$ will have no active lines, and the algorithm is finished. ▮

In order to prove that this algorithm terminates we must define measures for tautologous formulas, expansion trees (their list representations), and outlines. The measure for a tautologous formula is bases on the number of clauses it contains. We give our own recursive definition of clauses below. Here, if $\mathcal{A}_1$ and $\mathcal{A}_2$ are sets, then $\mathcal{A}_1 \uplus \mathcal{A}_2 := \{\xi_1 \cup \xi_2 \mid \xi_1 \in \mathcal{A}_1, \xi_2 \in \mathcal{A}_2\}$.

5.5.19. Definition. Let D be a λ-normal formula$_o$. We shall define two sets, $\mathcal{C}_D$ and $\mathcal{V}_D$, which are both sets of sets of b-atom subformula occurrences in D, by joint induction on the boolean structure of D. $\mathcal{C}_D$ is the set of clauses in D while $\mathcal{V}_D$ is the set of "dual" clauses in D. Dual clauses have been called vertical paths by Andrews (see [Andrews81]).

(1) If D is a b-atom, then $\mathcal{C}_D := \{\{D\}\}$ and $\mathcal{V}_D := \{\{D\}\}$.

(2) If $D = {\sim}D_1$ then $\mathcal{C}_D := \mathcal{V}_{D_1}$ and $\mathcal{V}_D := \mathcal{C}_{D_1}$.

(3) If $D = D_1 \vee D_2$ then $\mathcal{C}_D := \mathcal{C}_{D_1} \uplus \mathcal{C}_{D_2}$ and $\mathcal{V}_D := \mathcal{V}_{D_1} \cup \mathcal{V}_{D_2}$.

(4) If $D = D_1 \wedge D_2$ then $\mathcal{C}_D := \mathcal{C}_{D_1} \cup \mathcal{C}_{D_2}$ and $\mathcal{V}_D := \mathcal{V}_{D_1} \uplus \mathcal{V}_{D_2}$.

(5) If $D = D_1 \supset D_2$ then $\mathcal{C}_D := \mathcal{V}_{D_1} \uplus \mathcal{C}_{D_2}$ and $\mathcal{V}_D := \mathcal{C}_{D_1} \cup \mathcal{V}_{D_2}$. ∎

If $\mathcal{B}$ is a finite set, we write $|\,\mathcal{B}\,|$ to denote the cardinality of $\mathcal{B}$.

The number $cl(\mathcal{O})$, defined below, can be thought of as the maximum number of subproofs (sequents) which must be examined in $\mathcal{O}$ before we have given justifications to all the proof lines in $\mathcal{O}$. The naive algorithm for constructing ND-proofs from outlines will in fact generate all of these subproofs before it terminates. In general, many of these subproofs are redundant and/or trivial. In the next section we use a criterion called *focusing* to recognize and avoid some of these subproofs.

5.5.20. Definition. Let A be a formula$_o$, and let A' be a λ-normal form of A. We define $cl(A)$ to be the number of clauses in A', *i.e.* $cl(A) :=|\, \mathcal{C}_{A'}\,|$. (Clearly, the choice of A' does not affect this value.) Let $\mathcal{O} = \langle L, \Sigma, \{R_l\}\rangle$ be an outline with active lines. We define

$$cl(\mathcal{O}) := \sum_{\sigma\in\Sigma} cl(Fm(Q_\sigma)).$$

Let R be a list representation such that either $\langle R, A\rangle \in \mathcal{E}$ or $\langle {\sim}R, {\sim}A\rangle \in \mathcal{E}$ for some formula A. The *measure* of R, $\#R$, is defined to be the number of occurrences of $\wedge$, $\vee$, $\supset$, SEL, and expansion terms in R. We now define the *measure of the outline* $\mathcal{O}$ to be the ordinal number

$$\#\mathcal{O} := \omega \cdot (cl(\mathcal{O}) - |\,\Sigma\,|) + \sum_l \#R_l$$

where the sum is over active lines in $\mathcal{O}$ and ω is the order type of the natural numbers. ∎

5.5.21. Example. If R is the list structure

$$(\text{EXP}\ \ (u\ (\text{SEL}\ \ v\ \ (\supset\ Pv\ Pu)))(v\ (\text{SEL}\ \ w\ \ (\supset\ Pw\ Pv))))$$

then $\#R = 6$. If $\mathcal{O}$ is the outline in Example 5.10, then $\#R_2 = 2$, $\#R_3 = 3$, $\#R_{16} = 2$, $cl(\mathcal{O}) = cl({\sim}Pzy \vee {\sim}[Pzy \supset Pz.c.Pz] \vee Pz.c.Pz) = 2$, and $\#\mathcal{O} = \omega\cdot 1 + 7$. ∎

5.5.22. **Proposition.** *If $\mathcal{O}$ is an outline and $\mathcal{O}'$ is the result of applying a D- or P- transformation to $\mathcal{O}$, then $cl(\mathcal{O}) = cl(\mathcal{O}')$.*

Proof. We shall prove this by showing that (a) if $\sigma\in\Sigma$ gives rise to one sequent $\sigma'\in\Sigma'$ then $cl(Fm(Q_\sigma)) = cl(Fm(Q_{\sigma'}))$, or (b) if σ gives rise to two sequents σ' and σ'' then $cl(Fm(Q_\sigma)) = cl(Fm(Q_{\sigma'}))+cl(Fm(Q_{\sigma''}))$.

If the transformation applied was either D-Lambda or P-Lambda (case (a)), $Q_\sigma = Q_{\sigma'}$, so we clearly have $cl(Fm(Q_\sigma)) = cl(Fm(Q_{\sigma'}))$. If the transformation applied was either D-Neg or P-Neg when they only remove double negations (case (a)), then $Q_{\sigma'}$ is the result of removing a double negation from Q_σ. $Fm(Q_{\sigma'})$ is then the result of dropping double negations from $Fm(Q_\sigma)$ and, clearly, $cl(Fm(Q_\sigma)) = cl(Fm(Q_{\sigma'}))$.

If $Q_{\sigma'}$ is the result of eliminating a top-level universal or existential node from Q_σ (modulo double negations), it is easy to show that $cl(Fm(Q_\sigma)) = cl(Fm(Q_{\sigma'}))$. Hence, this Proposition holds for the cases where the transformation is either D-All, D-NotExists, D-Exists, P-All, P-Exists, or P-NotAll, which all fall under case (b).

If the transformation applied was D-Conj (case (a)) then Q_σ is $[\bigvee_{c\in\Gamma} \sim Q_c] \vee \sim[Q'_b \wedge Q'_c] \vee Q_z$ and $Q_{\sigma'}$ is $[\bigvee_{c\in\Gamma} \sim Q_c] \vee \sim Q'_b \vee \sim Q'_c \vee Q_z$. Clearly, we then have $cl(Fm(Q_\sigma)) = cl(Fm(Q_{\sigma'}))$.

If the transformation applied was P-Disj1 then Q_σ is $[\bigvee_{c\in\Gamma_z} \sim Q_c] \vee [Q'_a \vee Q'_x]$ and $Q_{\sigma'}$ is $[\bigvee_{c\in\Gamma_z} \sim Q_c] \vee \sim\sim Q'_a \vee Q'_x$. Again, we then have $cl(Fm(Q_\sigma)) = cl(Fm(Q_{\sigma'}))$.

If the transformation applied was P-Imp then Q_σ is $[\bigvee_{c\in\Gamma_z} \sim Q_c] \vee [Q'_a \supset Q'_x]$ and $Q_{\sigma'}$ is $[\bigvee_{c\in\Gamma_z} \sim Q_c] \vee \sim Q'_a \vee Q'_x$. Again, we then have $cl(Fm(Q_\sigma)) = cl(Fm(Q_{\sigma'}))$.

If the transformation was P-Conj (case (b)) then Q_σ is $[\bigvee_{c\in\Gamma_z} \sim Q_c] \vee [Q'_m \wedge Q'_y]$, $Q_{\sigma'}$ is $[\bigvee_{c\in\Gamma_z} \sim Q_c] \vee Q'_m$, and $Q_{\sigma''}$ is $[\bigvee_{c\in\Gamma_z} \sim Q_c] \vee Q'_y$. It is easy to now verify that $cl(Fm(Q_\sigma)) = cl(Fm(Q_{\sigma'})) + cl(Fm(Q_{\sigma''}))$.

If the transformation was D-Disj then Q_σ is $[\bigvee_{c\in\Gamma} \sim Q_c] \vee \sim[Q'_b \vee Q'_n] \vee Q_z$, $Q_{\sigma'}$ is $[[\bigvee_{c\in\Gamma} \sim Q_c] \vee \sim Q'_b \vee Q_z]$, and $Q_{\sigma''}$ is $[[\bigvee_{c\in\Gamma} \sim Q_c] \vee \sim Q'_n \vee Q_z]$. Again, it is easy to now verify that $cl(Fm(Q_\sigma)) = cl(Fm(Q_{\sigma'})) + cl(Fm(Q_{\sigma''}))$.

The only transformations left to consider are D-Imp and both D-Neg and P-Neg when they do not simply remove double negations. But their application reduces to one of the transformation already considered.

Q.E.D.

5.5.23. **Proposition.** *Let $\mathcal{O}$ be an outline and $\mathcal{O}'$ be the result of applying a D- or P- transformation to $\mathcal{O}$. If the transformation was either P-Lambda or D-Lambda, or if it was P-Neg or D-Neg, when they only remove double negations, then $\#\mathcal{O}' = \#\mathcal{O}$. In all other cases, $\#\mathcal{O}' < \#\mathcal{O}$.*

Proof. First assume that the transformation was either D-Lambda or P-Lambda. Then $|\Sigma| = |\Sigma'|$ and the expansion trees associated with active lines in $\mathcal{O}$ are the same as those associated with active lines in $\mathcal{O}'$. Hence, $\#\mathcal{O}' = \#\mathcal{O}$. If the transformation was either D-Neg or P-Neg where only double negations were removed, then again $|\Sigma| = |\Sigma'|$. Also the #-value of the expansion trees associated with active lines do not change since the number of negations in expansion trees are not counted. Thus $\#\mathcal{O}' = \#\mathcal{O}$.

If the transformation applied was either D-Disj or P-Conj, then $|\Sigma'| > |\Sigma|$ and $\#\mathcal{O}' < \#\mathcal{O}$. If the transformation was any one of the remaining ones, then $|\Sigma| = |\Sigma'|$ and some $\wedge$, $\vee$, $\supset$, SEL, or expansion term is removed from an expansion tree associated with some active line. Hence, again we must have $\#\mathcal{O}' < \#\mathcal{O}$ since $\sum_l \#R'_l < \sum_l \#R_l$. **Q.E.D.**

5.5.24. **Theorem.** *Algorithm 5.18 terminates when applied to any proof outline. Hence, if R represents an ET-proof of some formula$_o$ A, then Algorithm 5.18 will construct an ND-proof for A from the trivial outline for A based on R. Similarly, a cut-free LKH-proof can be effectively constructed for the sequent $\rightarrow A$.*

Proof. We first consider the termination of this algorithm. The loop consisting of step (2) and the loop consisting of step (3) must terminate. (By Proposition 5.23, the value of $\#\mathcal{O}$ is not changed in these loops.) Let us now look at the loop termination condition in line (4). Assume that $\mathcal{O}$ is an outline whose active lines are all in λ-normal form and none of which are double negations, *i.e.* $\mathcal{O}$ has been processed by steps (2) and (3). Now, an active lines, l, in $\mathcal{O}$ assert only signed atoms if and only if $\#R_l = 0$. This latter condition is also equivalent to $cl(Fm(Q_\sigma)) = 1$ for each $\sigma \in \Sigma$. Hence, all the active lines in $\mathcal{O}$ assert signed atoms if and only if $\#\mathcal{O} = 0$.

Now, each time we do step (5), the measure of $\mathcal{O}$ decreases (Proposition 5.23). Hence, we must eventually enter line (4) with $\#\mathcal{O} = 0$, which is equivalent to the termination condition.

After the remaining sequents in $\mathcal{O}$ are processed by either RuleP1 or RuleP2 (guaranteed by Proposition 5.17), the final outline, since it contains no active lines, will be a completed ND-proof. If we had started this algorithm on a trivial proof outline of A based on R, the resulting outline would then be an ND-proof for A reflecting some of the structure of the ET-proof R. If we were to use Proposition 5.15, each time a transformation (other than RuleP1 or RuleP2) was applied, we could build a LKH-proof figure (without the cut inference) which has as leaves the sequents in Σ and has as a root the sequent $\rightarrow A$. A complete LKH-proof could then be constructed if we produce a cut-free LKH-proof of any sequent to which we have applied either RuleP1 or RuleP2. In the case of RuleP1, we have two complementary, signed atoms A_1 and A_2 which are the assertions of active lines, a_1 and a_2. We may assume that A_2 conv-I $\sim A_1$. Then the sequent, $\Gamma, a_1, a_2 \longrightarrow z$, where $\Gamma := \Gamma_z \setminus \{a_1, a_2\}$, has the following cut-free proof.

$$\dfrac{\dfrac{\dfrac{A_1 \longrightarrow A_1}{A_1, \sim A_1 \longrightarrow}\ \sim\text{–IA}}{\Gamma, A_1, \sim A_1 \longrightarrow z}\ \text{several Thinnings}}{\Gamma, A_1, A_2 \longrightarrow z}\ \lambda$$

In the case of RuleP2, we have the following cut-free proof for $\Gamma, a \longrightarrow z$, where $\Gamma := \Gamma_z \setminus \{a\}$. Here, the assertion in line a is an alphabetic variant of the assertion in line z.

$$\dfrac{\dfrac{a \longrightarrow a}{\Gamma, a \longrightarrow a}\ \text{several Thinnings}}{\Gamma, a \longrightarrow z}\ \lambda$$

Using these in two inference figures and the rules for constructing new inference figures given in Proposition 5.15, we can construct a cut-free LKH-proof for the sequent $\longrightarrow A$.

Notice that we now have a proof of Takahashi's cut-elimination theorem ([Takahashi67]) for $\mathcal{T}$, which is a generalization of Gentzen's *Hauptsatz* for $\mathcal{T}$. If $\vdash_{\mathcal{T}} A$, then A has an ET-proof which can be converted to a cut-free LKH-proof of A. **Q.E.D.**

Notice that if we have an outline with a sponsoring line of the form $A \supset B$, it is a complete procedure to try to prove B from the hypothesis A. That is, we do not need to have the contrapositive inference available to build ND-proofs. Use of the contrapositive rule may, however, make an ND-proof more readable.

At this point we can add to our list of transformations four more transformations. They are useful to note here since their use can result in shorter final ND-proofs. For example, the following transformation is obviously valid.

RuleP

Let lines $a_1, \ldots, a_n$ be some of the supports of z, such that $[A_1 \wedge \ldots \wedge A_n] \supset A$ is tautologous. Then we can change the justification of line z from NJ to $RuleP : a_1, \ldots, a_n$. Σ' is Σ_0.

(a_1)	$\mathcal{H}_1$	$\vdash$	A_1	$RuleX$
	$\vdots$		$\vdots$	$\vdots$
(a_n)	$\mathcal{H}_n$	$\vdash$	A_n	$RuleX$
(z)	$\mathcal{H}$	$\vdash$	A	NJ

$\Longrightarrow$

(z)	$\mathcal{H}$	$\vdash$	A	$RuleP : a_1, \ldots, a_n$

5.5.25. Definition. A function, T, on outlines is called a *safe transformation* if whenever $\mathcal{O}$ is an outline and $\mathcal{O}'$ is the result of applying T to $\mathcal{O}$, $\mathcal{O}'$ is an outline for the same formula as $\mathcal{O}$ and $\#\mathcal{O}' < \#\mathcal{O}$. ∎

All the transformations described to this point, except D-Lambda, P-Lambda and D-Neg, P-Neg when these simply remove double negations, are safe transformation. Any safe transformation can be added to the set of transformations selected in line (5) of Algorithm 5.18 without upsetting the termination or correctness of that algorithm. One easy way to show that a transformation is safe is to show that it is the composition of several other safe transformations. Hence, we make the following definition.

5.5.26. Definition. A transformation for outlines, say T, is *closely derived* from other transformations, $T_1, \dots, T_n$ if for any outline $\mathcal{O}$ for which T is applicable, the outlines $\mathcal{O}_1 := T(\mathcal{O})$ and $\mathcal{O}_2 := T_n(T_{n-1} \dots (T_1(\mathcal{O}))\dots)$ are such that the sets $\{Q_{\sigma_1} \mid \sigma_1 \in \Sigma_1\}$ and $\{Q_{\sigma_2} \mid \sigma_2 \in \Sigma_2\}$ are the same modulo adding or dropping double negations. The outlines $\mathcal{O}_1$ and $\mathcal{O}_2$ will generally differ in the non-active lines they contain. Notice, that any clause in any subproof, Q_{σ_1}, of $\mathcal{O}_1$ corresponds to a clause in some subproof, Q_{σ_2} of $\mathcal{O}_2$, and vice versa. ∎

If T is closely derived from D-Lambda, P-Lambda, D-Neg, P-Neg (when they only remove double negations) and at least one safe transformations, T is safe.

Since there was nothing special about using the negation of the first disjunct as a hypothesis to prove the second disjunct in P-Disj1, the following transformation behaves essentially the same.

P-Disj2

Set $\Sigma' := \Sigma_0 \cup \{\Gamma_z, a \to y\}$. Since $\langle R_z, A_1 \vee A_2\rangle \in \mathcal{E}$, $R_z = (\vee\ R_1\ R_2)$. Set $R'_a := {\sim}R_2$ and $R'_y := R_1$.

(z)	$\mathcal{H}$	$\vdash$	$A_1 \vee A_2$	NJ

$\Longrightarrow$

(a)	a	$\vdash$	${\sim}A_2$	Hyp
(x)	$\mathcal{H}, a$	$\vdash$	A_1	NJ
(y)	$\mathcal{H}$	$\vdash$	${\sim}A_2 \supset A_1$	$Deduct : x$
(z)	$\mathcal{H}$	$\vdash$	$A_1 \vee A_2$	$RuleP : y$

We can also introduce the contrapositive transformation. This can be used whenever P-Imp is used.

P-Contrapositive

Set $\Sigma' := \Sigma \cup \{\Gamma_z, a \to x\}$ and $R_a := {\sim}R_2$, $R_x := {\sim}R_1$, where $R_z := (\supset R_1\ R_2)$.

(z)	$\mathcal{H}$	$\vdash$	$A_1 \supset A_2$	NJ

$\Longrightarrow$

(a)	a	$\vdash$	${\sim}A_2$	Hyp
(x)	$\mathcal{H}, a$	$\vdash$	${\sim}A_1$	NJ
(y)	$\mathcal{H}$	$\vdash$	${\sim}A_2 \supset {\sim}A_1$	$Deduct : x$
(z)	$\mathcal{H}$	$\vdash$	$A_1 \supset A_2$	$RuleP : y$

Notice that the active lines left after applying P-Contrapositive are those left after applying P-Disj2 to the equivalent disjunctive form of line z. Thus, P-Contrapositive is closely derived from P-Disj2. Proposition 5.15 is still true if we add in the above three transformations. This is obvious in the case of RuleP and P-Disj2. For the P-Contrapositive case, the following inference figure will work.

$$\cfrac{\cfrac{\cfrac{\Gamma, {\sim}A_2 \longrightarrow {\sim}A_1}{\Gamma, {\sim}A_2, A_1 \longrightarrow}\ {\sim}{-}\mathrm{ES}}{\Gamma, A_1 \longrightarrow A_2}\ {\sim}{-}\mathrm{EA}}{\Gamma \longrightarrow A_1 \supset A_2}\ \supset{-}\mathrm{IS}$$

We can now describe how the ND-proof in Example 5.7 could be constructed by appling Algorithm 5.18 to the outline in Example 5.10. Below we list a possible sequence of transformations which this algorithm could apply to the outline. Along with the transformation, we list the sequents which are present in the outline after that transformation was applied.

Transformation	Sequents
P-Exists	$2,3 \rightarrow 15$
P-All	$2,3 \rightarrow 14$
D-All	$3,4 \rightarrow 14$
D-Exists	$3,5 \rightarrow 13$
D-All	$5,6 \rightarrow 13$
D-Imp	$5,8 \rightarrow 10 \quad 5,11 \rightarrow 12$
D-NotExists	$5,9 \rightarrow 10 \quad 5,11 \rightarrow 12$
RuleP1	$5,11 \rightarrow 12$
RuleP2	–

Section 5.6: Focused Construction of Proof Outlines

The outlines which are generated by the algorithm presented in the previous section are generally very redundant. For example, assume that lines 1, 50, and 100 (shown below) are the active lines of an outline.

(1)	$\mathcal{H}$	$\vdash$	$\forall x\,.Px \lor Mx$	*RuleX*
(50)	$\mathcal{H}_1$	$\vdash$	A_1	NJ
(100)	$\mathcal{H}_2$	$\vdash$	A_2	NJ

where R_1, the expansion tree for line 1, has the list representation (EXP $(t'\ R')\ (t''\ R'')$) for some formulas t', t'' and list structures R', R'', and line 1 supports *both* lines 50 and 100. In order to process line 1, we would apply D-All to it twice (assuming that t' and t'' are admissible in this outline). The new active lines would then be:

(2)	$\mathcal{H}$	$\vdash$	$Pt' \lor Mt'$	$\forall I : 2$
(3)	$\mathcal{H}$	$\vdash$	$Pt'' \lor Mt''$	$\forall I : 2$
(50)	$\mathcal{H}_1$	$\vdash$	A_1	NJ
(100)	$\mathcal{H}_2$	$\vdash$	A_2	NJ

where the expansion trees for line 2 and 3 are $R_2 = R'$ and $R_3 = R''$. Here, lines 2 and 3 support both lines 50 and 100. As is often the case, both instances of line 1 need not be needed to support the sponsors of line 1. For example, assume that line 2 is needed to finish proving line 50 but is not necessary to prove line 100, while line 3 is needed to finish proving line 100 but not line 50. In this case, we would like to remove line 3 as a support from line 50 and remove line 2 as a support of line 100. This would result in focusing the outline building process. The resulting ND-proof would be much easier to read. Also, once an outline becomes unfocused in this sense, it can become crowded with unnecessary lines. For example, in the above outline fragment, if we applied D-Disj to line 2 twice (in order to "deactivate" it) while it supports both lines 50 and 100, we would get an outline whose active lines are shown below.

(3)	$\mathcal{H}$	$\vdash$	$Pt'' \lor Mt''$	$\forall I : 3$
(4)	4	$\vdash$	Pt'	Hyp
(25)	$\mathcal{H}_1, 4$	$\vdash$	A_1	NJ
(26)	26	$\vdash$	Mt'	Hyp
(49)	$\mathcal{H}_1, 26$	$\vdash$	A_1	NJ
(51)	51	$\vdash$	Pt'	Hyp

(75)	$\mathcal{H}_2, 51$	$\vdash$	A_2	NJ
(76)	76	$\vdash$	Mt'	Hyp
(99)	$\mathcal{H}_2, 76$	$\vdash$	A_2	NJ

where the justification for lines 50 and 100 are changed to be $Cases : 2, 25, 49$ and $Cases : 2, 75, 99$, resp. If we could have determined that line 2 was not needed to prove line 100, then we would not have added lines 49,51,75,76, or 99 when we applied D-Disj. Also notice that line 3, in the above outline, is a support for lines 25, 49, 75, and 99, and hence, if we were to apply D-Disj with respect to these lines, we would get 8 new sponsoring lines and 8 new supporting lines. After these applications of D-Disj, we would have a total of 24 new lines added to our outline. If we could have simplified the the supports prior to doing these transformations, we would have only entered 8 new lines.

To use a disjunction in a deduction, we generally have to argue by cases and much of the case analysis in the above example is often unavoidable. However, if we wish to use an implication in a deduction, we seldom use the Rule of Cases on the equivalent disjunctive form of this implication. Unfortunately, this is how our naive algorithm will use such a support line. Much more appropriate would, of course, be to use such a line in conjunction with *modus ponens*. But again, this requires us to have some ability to look ahead to see exactly when *modus ponens* is correct. As it turns out, a solution to the problem of determining which support lines are truly needed to prove their sponsoring line also provides us with the tools needed to determine when *modus ponens* can correctly be used with an implicational support line. Below we introduce several concepts which will allow us to solve these problems.

5.6.27. Definition. Let D be a λ-normal formula$_o$. Let $\mathcal{M}$ be a set of unordered pairs of b-atom subformula occurrences of D, such that if $\{H, K\} \in \mathcal{M}$, then H conv-I K, and either H occurs positively and K occurs negatively in D, or H occurs negatively and K occurs positively in D. Such a set $\mathcal{M}$ is called a *mating* for D. If $\{H, K\} \in \mathcal{M}$ we say that H and K are $\mathcal{M}$-*mated*, or simply *mated* if the mating can be determined from context. If it is also the case that for any $\xi \in \mathcal{C}_D$ there is a $\{H, K\} \in \mathcal{M}$ such that $\{H, K\} \subset \xi$, then we say that $\mathcal{M}$ is a *clause-spanning mating* (cs-mating, for short) for D. In this case, we shall also say that $\mathcal{M}$ *spans* D. If $\mathcal{D}$ is a set of λ-normal formulas$_o$, we say that $\mathcal{M}$ is a mating (cs-mating) for $\mathcal{D}$ if $\mathcal{M}$ is a mating (cs-mating) for $\vee\mathcal{D}$. Here, the order by which the disjunction $\vee\mathcal{D}$ is constructed is taken to be arbitrary but fixed. ∎

The notion of a mating used by Andrews in [Andrews81] is a bit more general than the one we have defined here. In that paper, a mating, $\mathcal{M}$, was a set of pairs, $\langle H, K\rangle$, such that there was a substitution θ which made them complementary, *i.e.* $\theta K = \sim\theta H$. Except for this difference, the notion of a cs-mating corresponds very closely to his notion of a p-acceptable mating. Bibel in [Bibel81] also exploits matings for various theorem proving and metatheoretical application.

5.6.28. Definition. Let H be a b-atom subformula occurrence in D. Let $\hat{H}$ be equal to H if H occurs positively in D and $\sim H$ otherwise. (Notice that although $\hat{H}$ is defined with respect to D, we drop this reference to D for notational convention.) We also define

$$\hat{\mathcal{C}}_D := \bigwedge_{\xi\in\mathcal{C}_D} \bigvee_{H\in\xi} \hat{H} \quad \text{and} \quad \hat{\mathcal{V}}_D := \bigvee_{\xi\in\mathcal{V}_D} \bigwedge_{H\in\xi} \hat{H}.$$

Here, $\hat{\mathcal{C}}_D$ is a conjunctive normal form of D and $\hat{\mathcal{V}}_D$ is a disjunctive normal form of D. It is straightforward to show that $D \equiv \hat{\mathcal{C}}_D$ and $D \equiv \hat{\mathcal{V}}_D$. ∎

5.6.29. **Proposition.** *Let D be in λ-normal form. D is tautologous if and only if D has a cs-mating.*

Proof. Since $D \equiv \hat{\mathcal{C}}_D$, D is tautologous if and only if every $\bigvee_{H\in\xi} \hat{H}$ is tautologous for each $\xi \in \mathcal{C}_D$. But $\bigvee_{H\in\mathcal{C}_D} \hat{H}$ is tautologous if and only if ξ contains two b-atom subformulas occurrences H and K of D such that one occurs positively and the other occurs negatively in D and H conv-I K. This is precisely the criterion by which D has a cs-mating. **Q.E.D.**

5.6.30. Definition. Let $\mathcal{D}$ be a finite, nonempty set of formulas$_o$, and let $\mathcal{M}$ be a mating for $\mathcal{D}$. With respect to $\mathcal{D}$ and $\mathcal{M}$, define $\approx^0$ to be the binary relation on $\mathcal{D}$ such that when $D_1, D_2 \in \mathcal{D}$, $D_1 \approx^0 \mathcal{D}_2$ if D_1 contains a b-atom subformula occurrence H and D_2 contains a b-atom subformula occurrence K such that $\{H, K\} \in \mathcal{M}$. Let $\approx$ be the reflexive, transitive closure of $\approx^0$. Clearly $\approx$ is an equivalence relation on $\mathcal{D}$. If $D \in \mathcal{D}$, we shall write $[D]_\approx$ to denote the equivalence class (partition) of $\mathcal{D}$ which contains D. ∎

5.6.31. **Theorem.** *Let $\mathcal{D}$ be a finite, nonempty set of formulas$_o$. $\mathcal{M}$ is a cs-mating for $\mathcal{D}$ if and only if $\mathcal{M}$ spans at least one of the $\approx$-partitions of $\mathcal{D}$.*

Proof. Assume that while $\mathcal{M}$ is a cs-mating for $\mathcal{D}$, $\mathcal{M}$ does not span any $\approx$-partition of $\mathcal{D}$. Let $\mathcal{P}_1, \ldots, \mathcal{P}_n$ be the $\approx$-partitions of $\mathcal{D}$. Hence, for each $i = 1, \ldots, n$, there is a clause in $\vee\mathcal{P}_i$, $\xi_i \in \mathcal{C}_{\vee\mathcal{P}_i}$, which does contain a mated pair. Let $\xi := \xi_1 \cup \ldots \cup \xi_n$. ξ is a clause in $\vee\mathcal{D}$ and hence, must have a mated pair H and K, and these are such that there are distinct integers i, j such that $1li, jln$ and $H \in \xi_i$ and $K \in \xi_j$. This implies that there is some formula, $E \in \mathcal{P}_i$, and some formula $F \in \mathcal{P}_j$ such that $E \approx^0 F$. This contradicts the fact that $\mathcal{P}_i$ and $\mathcal{P}_j$ are distinct $\approx$-partitions. Hence, $\mathcal{M}$ must span some $\approx$-partition of $\mathcal{D}$.

The converse is trivially true. **Q.E.D.**

Below is our definition of such a criterion. The information necessary in making this determination is contained in the connection information of a cs-mating.

5.6.32. Definition. Let $\mathcal{O} = \langle L, \Sigma, \{R_l\}\rangle$ be an outline and for each $\sigma \in \Sigma$, let $\mathcal{M}_\sigma$ be a cs-mating for $Fm(Q_\sigma)$. Set $\mathcal{M} := \bigcup_{\sigma\in\Sigma} \mathcal{M}_\sigma$. $\mathcal{M}$ is called a *cs-mating for* $\mathcal{O}$. (Notice that $\mathcal{M}$ is also a cs-mating for each $Fm(Q_\sigma)$.) Let D_l be the formula $Fm(Q_l)$ if l is a sponsoring line or $Fm({\sim}Q_l)$ if l is a supporting line. Let $\sigma \in \Sigma$ be a sequent $\Gamma_z \rightarrow z$, and let $\mathcal{D}_\sigma$ be the set of formulas $\mathcal{D}_\sigma := \{D_l \mid l \in \Gamma_z\} \cup \{D_z\}$ if z does not assert $\perp$ or $\mathcal{D}_\sigma := \{D_l \mid l \in \Gamma_z\}$ if z does assert $\perp$. Notice, that $Q_\sigma = \bigvee_l D_l$ where the disjunction is taken over the (active) lines in σ. We say that $\mathcal{O}$ is $\mathcal{M}$*-focused* if for each $\sigma \in \Sigma$, $\mathcal{D}_\sigma$ is composed of exactly one $\approx$-partition. ∎

Let $\mathcal{O} = \langle L, \Sigma, \{R_l\}\rangle$ be an outline and let $\mathcal{M}$ be a cs-mating for $Fm(Q_\sigma)$. If $\mathcal{O}$ is not $\mathcal{M}$-focused, then there must be a $\sigma \in \Sigma$ such that $\mathcal{D}_\sigma$ has too many members, *i.e.* there are at least two $\approx$-partitions of $\mathcal{D}_\sigma$. The above theorem tells us that we really only require one partition. What we need is a transformation which will permit us to remove elements of $\mathcal{D}_\sigma$. Since this may be done by either removing a support from a sponsoring line or by changing the assertion of the sponsoring line to $\perp$, we need the following two outline transformations.

P-Thinning
We may replace line z with line y as a sponsoring line, provided that what we get is still an outline. In this case, set $\Sigma' := \Sigma_0 \cup \{\Gamma_z \rightarrow y\}$, where $\Sigma_0 := \Sigma \setminus \{\Gamma_z \rightarrow z\}$.

(z)	$\mathcal{H}$	$\vdash$	A	NJ

$\Longrightarrow$

(y)	$\mathcal{H}$	$\vdash$	$\perp$	NJ
(z)	$\mathcal{H}$	$\vdash$	A	$RuleP : y$

D-Thinning

If line a supports line z, then we can drop line a as a support of line z, provided that what we get is still an outline.

Notice that Proposition 5.22 is not true of these transformations. In particular, if $\mathcal{O}'$ results from applying either P-Thinning or D-Thinning to an outline $\mathcal{O}$, $cl(\mathcal{O}')lcl(\mathcal{O})$. This helps account for the fact that using these transformations will generally shorten the final ND-proofs derived from them, since the maximum number of subproofs we need to examine has been reduced.

Although the provisos for these two transformations are very strong, we shall still be able to focus an outline which is not focused by applying these two transformations. The following algorithm will preform this task.

5.6.33. Algorithm. Let the outline $\mathcal{O} = \langle L, \Sigma, \{R_l\}\rangle$ and the cs-mating $\mathcal{M}$ for $\mathcal{O}$ be given. For each $\sigma \in \Sigma$, do steps (1) and (2) below.

(1) Let $\mathcal{P}$ be a $\approx$-partition of $\mathcal{D}_\sigma$ which is spanned by $\mathcal{M}$. Let $\mathcal{B} := \Gamma_z \cup \{z\}$ if z does not assert $\perp$ and $\mathcal{B} := \Gamma_z$, otherwise. Hence, $\mathcal{D}_\sigma = \{D_l \mid l \in \mathcal{B}\}$. Let $\mathcal{B}'$ be the set $\mathcal{B}$ less those lines l such that $D_l \in \mathcal{P}$.

(2) If $\mathcal{B}'$ is not empty, then for each $l \in \mathcal{B}'$ do either step (2.1) or step (2.2).

(2.1) If $l = z$ then apply P-Thinning to remove z as a sponsoring line.

(2.1) If $l \in \Gamma_z$ then apply D-Thinning to remove l as a support of line z. ∎

5.6.34. **Theorem.** *Let $\mathcal{O} = \langle L, \Sigma, \{R_l\}\rangle$ be an outline, and let $\mathcal{M}$ be a cs-mating for $\mathcal{O}$. If $\mathcal{O}$ is not $\mathcal{M}$-focused, then Algorithm 5.33, when applied to $\mathcal{O}$ and $\mathcal{M}$, will produce an outline which is $\mathcal{M}$-focused.*

Proof. Let $\mathcal{O}' = \langle L', \Sigma', \{R'_l\}\rangle$ be the structure which results from applying either P- or D-Thinning to an outline $\mathcal{O}$ as determined by the algorithm above. Of the five conditions to check to verify that $\mathcal{O}'$ is an outline, only the last one needs to be looked at closely here. Now, let $\sigma \in \Sigma$ be the sequent affected by the application of the P- or D-Thinning transformation, and let σ' be the sequent that derives from σ. In either of the thinning transformations, $\mathcal{D}_{\sigma'} \subset \mathcal{D}_\sigma$, so $<_{Q'_\sigma}$ is a subrelation of $<_{Q_\sigma}$ and, therefore, is acyclic. Also, since $\mathcal{D}_{\sigma'}$ contains a $\approx$-partition which is spanned by $\mathcal{M}$, $\vee\mathcal{D}'$, and therefore $Fm(Q_{\sigma'})$, is tautologous, and $\mathcal{O}'$ is an outline. Thus each application of a thinning transformation is valid. It is easy to see that when the algorithm halts, the resulting outline must be $\mathcal{M}$-focused.

Notice that Proposition 5.15 is still true if we were to permit the thinning transformations to be used along with the transformations listed in Definition 5.12. If we have a cut-free LKH-proof of the sequent σ' then we get one for σ by simply adding the Thinning inference to this LKH-proof figure. **Q.E.D.**

Now that we know how to focus an outline, we look at how to modify some of our transformations so that a focused outline can often be transformed directly to another focused outline without applying either the P- or D-Thinning transformations explicitly. Consider the following transformations.

P-DropDisj$_i$
Let $i = 1, 2$, and set $j := 3 - i$. Let $\mathcal{M}$ be a cs-mating for Q_σ, let $Q_1 \vee Q_2 := Q_z$, and let $\mathcal{D} := \{Fm(\sim Q_l) \mid l \in \Gamma_z\}$. If there is a $\approx$-partition of $\mathcal{D} \cup \{Fm(Q_i)\}$ which is spanned by $\mathcal{M}$, then we can drop the disjunct A_j, as below. Here, $R'_y := Q_i$ and $\Sigma' := \Sigma_0 \cup \{\Gamma_z \rightarrow y\}$.

(z)	$\mathcal{H}$	$\vdash$	$A_1 \vee A_2$	NJ

$\Longrightarrow$

(y)	$\mathcal{H}$	$\vdash$	A_i	NJ
(z)	$\mathcal{H}$	$\vdash$	$A_1 \vee A_2$	$RuleP : y$

Notice that P-DropDisj$_1$ is a closely derived transformation since we would have the essentially the same effect (*i.e.* an outline with the same active lines) if we had first applied P-Disj1 and then applied D-Thinning to remove the hypothesis line a from its support line. P-DropDisj$_2$ is related to P-Disj2 in the same fashion.

D-BackChain$_i$
Let $i = 1, 2$ and set $j := 3 - i$. Let a be a disjunctive support line which is supported by z, and let σ be the sequent $\Gamma, a \rightarrow z$. Let $\mathcal{M}$ be a cs-mating for $Fm(Q_\sigma)$, $(\vee\ R_1\ R_2) := R_a$, $Q_1 := \text{rep}[\![R_1, A_1]\!]$, $Q_2 := \text{rep}[\![R_2, A_2]\!]$, and $\mathcal{D} := \{Fm(\sim Q_i)\} \cup \{Fm(\sim Q_l) \mid l \in \Gamma\}$. If $\mathcal{D}$ is spanned by $\mathcal{M}$ then add the lines below to the outline. If we let $\Sigma_0 := \Sigma \setminus \{\Gamma_z \rightarrow z\}$ then $\Sigma' := \Sigma_0 \cup \{\Gamma \rightarrow m,\ \Gamma, n \rightarrow x\}$. Also, set $R'_m := \sim R_i$, $R'_n := R_j$, and $R'_x := R_z$.

(a)	$\mathcal{H}$	$\vdash$	$A_1 \vee A_2$	$RuleX$
(z)	$\mathcal{H}_1$	$\vdash$	C	NJ

$\Longrightarrow$

(m)	$\mathcal{H}_1$	$\vdash$	$\sim A_i$	NJ
(n)	n	$\vdash$	A_j	Hyp
(x)	$\mathcal{H}_1, n$	$\vdash$	C	NJ
(y)	$\mathcal{H}_1$	$\vdash$	$A_j \supset C$	$Deduct : x$
(z)	$\mathcal{H}_1$	$\vdash$	C	$RuleP : a, m, y$

In order to justify this transformation, we shall show that D-BackChain$_i$ is closely derived from D-Disj and P-Thinning. We do this by showing that if $\mathcal{O}$ is an outline for which D-BackChain$_i$ can be applied to the disjunctive support line a to get the outline $\mathcal{O}_1 = \langle L_1, \Sigma_1, \{R^1_l\}\rangle$, then D-Disj and P-Thinning can be applied to $\mathcal{O}$ in such a fashion that a third outline $\mathcal{O}_2 = \langle L_2, \Sigma_2, \{R^2_l\}\rangle$ is constructed such that $\{Q_\sigma \mid \sigma \in \Sigma_1\} = \{Q_\sigma \mid \sigma \in \Sigma_2\}$. Let σ be a sequent in $\mathcal{O}$ of the form $\Gamma, A_1 \vee A_2 \rightarrow z$, such that the proviso that $\mathcal{D}$ is spanned by $\mathcal{M}$ is satisfied. After applying D-BackChain, σ gives rise to the two sequents $\sigma_1 := \Gamma \rightarrow \sim A_i$ and $\sigma_2 := \Gamma, A_j \rightarrow z$. If we were to apply D-Disj to line a in $\mathcal{O}$, σ gives rise to the two sequents $\sigma_3 := \Gamma, A_i \rightarrow z$ and $\sigma_4 := \Gamma, A_j \rightarrow z$. Since $\mathcal{D} = \mathcal{D}_{\sigma_3} \setminus \{Fm(Q_z)\}$ and it is spanned by $\mathcal{M}$, line z would be one of the lines our thinning Algorithm 5.33 would remove as a sponsoring line. Hence, the structure, say $\mathcal{O}_2$, which results from applying P-Thinning to line z is thus an outline, with the sequents $\sigma_5 := \Gamma, A_1 \rightarrow$ and σ_4 replacing σ in $\mathcal{O}$. But since $Q_{\sigma_1} = Q_{\sigma_5}$ and $Q_{\sigma_2} = Q_{\sigma_4}$, $\{Q_\sigma \mid \sigma \in \Sigma_1\} = \{Q_\sigma \mid \sigma \in \Sigma_2\}$. Hence, D-BackChain is closely derived from D-Disj and P-Thinning.

Since implication can be thought of disjunctively (in $\mathcal{T}$, anyway), the following two transformations can be thought of as two variants of D-BackChain$_i$ (which is the reason the D-BackChain transformation was given its name).

D-ModusPonens
Let $\mathcal{M}$ be a cs-mating for $Fm(Q_\sigma)$. Let a be an implicational support line, which is sponsored by line z and let σ be the sequent $\Gamma, a \rightarrow z$, $(\supset\ R_1\ R_2) := R_a$, $Q_1 := \text{rep}[\![R_1, A_1]\!]$, $Q_2 := \text{rep}[\![R_2, A_2]\!]$, and $\mathcal{D} := \{Fm(Q_1)\} \cup \{Fm(\sim Q_l) \mid l \in \Gamma\}$. If $\mathcal{D}$ is spanned by $\mathcal{M}$ then add the lines below to the outline, otherwise this transformation cannot be applied. If we set $\Sigma_0 := \Sigma \setminus \{\Gamma_z \rightarrow z\}$ then $\Sigma' := \Sigma_0 \cup \{\Gamma \rightarrow m,\ \Gamma, n \rightarrow x\}$. Also, set $R'_m := R_1$, $R'_n := R_2$, and $R'_x := R_z$.

$$\boxed{\begin{array}{llll} (a) & \mathcal{H} & \vdash & A_1 \supset A_2 \qquad\qquad RuleX \\ (z) & \mathcal{H}_1 & \vdash & C \qquad\qquad\qquad\quad NJ \end{array}} \quad \Longrightarrow \quad \boxed{\begin{array}{lllll} (m) & \mathcal{H}_1 & \vdash & A_1 & NJ \\ (n) & n & \vdash & A_2 & Hyp \\ (x) & \mathcal{H}_1, n & \vdash & C & NJ \\ (y) & \mathcal{H}_1 & \vdash & A_2 \supset C & Deduct : x \\ (z) & \mathcal{H}_1 & \vdash & C & RuleP : a, m, y \end{array}}$$

D-ModusTollens

This is a same as above, except that $\mathcal{D} := \{Fm(\sim Q_2)\} \cup \{Fm(\sim Q_l) \mid l \in \Gamma\}$ and $R'_m := \sim R_2$, $R'_n := \sim R_1$, and $R'_x := R_z$.

$$\boxed{\begin{array}{llll} (a) & \mathcal{H} & \vdash & A_1 \supset A_2 \qquad\qquad RuleX \\ (z) & \mathcal{H}_1 & \vdash & C \qquad\qquad\qquad\quad NJ \end{array}} \quad \Longrightarrow \quad \boxed{\begin{array}{lllll} (m) & \mathcal{H}_1 & \vdash & \sim A_2 & NJ \\ (n) & n & \vdash & \sim A_1 & Hyp \\ (x) & \mathcal{H}_1, n & \vdash & C & NJ \\ (y) & \mathcal{H}_1 & \vdash & \sim A_1 \supset C & Deduct : x \\ (z) & \mathcal{H}_1 & \vdash & C & RuleP : a, m, y \end{array}}$$

If A_2 conv-I C in D-ModusPonens, we clearly do not need to add lines n, x, y to the outline if we also give line z the justification $RuleP : a, m$. This way we could avoid proving an obvious subproof. This strengthed form of D-ModusPonens is not technically a closely derived transformation in the strong sense of 5.26, but this modification is clearly valid. The same comments apply to D-ModusTollens if A_1 and C are complementary.

Now we show how we can construct a much more readable ND-proof from the proof outline in Example 5.10. If to that outline we applied P-Exists, P-All, and apply D-All to line 2 and then line 3, we have the following outline.

$$\begin{array}{llll r} (1) & 1 & \vdash & \exists c\, \forall p\, .[\exists u\, .pu] \supset .p.cp & Hyp \\ (2) & 2 & \vdash & \forall x\, \exists y\, .Pxy & Hyp \\ (3) & 3 & \vdash & \forall p\, .[\exists u\, .pu] \supset .p.cp & Hyp \\ (4) & 2 & \vdash & \exists y\, .Pzy & \forall I\ : z, 2 \\ (5) & 3 & \vdash & [\exists u\, .Pzu] \supset .Pz.c.Pz & \forall I\ : Pz, 3 \\ (14) & 2,3 & \vdash & Pz.c.Pz & NJ \\ (15) & 2,3 & \vdash & \forall z\, .Pz.c.Pz & \forall G\ : 14 \\ (16) & 2,3 & \vdash & \exists f\, \forall z\, .Pz.fz & \exists G\ : [\lambda v.c.Pv], 15 \\ (17) & 1,2 & \vdash & \exists f\, \forall z\, .Pz.fz & RuleC : 1, 16 \\ (18) & 1 & \vdash & [\forall x\, \exists y\, .Pxy] \supset .\exists f\, \forall z\, .Pz.fz & Deduct : 17 \\ (19) & & \vdash & [\exists c\, \forall p\, .[\exists u\, .pu] \supset .p.cp] \supset & \\ & & & \quad [\forall x\, \exists y\, .Pxy] \supset \exists f\, \forall z\, .Pz.fz & Deduct : 18 \end{array}$$

The active lines are 4, 5, 14 and the expansion trees associated with them are

$$\begin{aligned} R_4 &= (\text{SEL}\ \ y\ \ Pzy) \\ R_5 &= (\supset (\text{EXP}\ \ (y\ Pzy))\ Pz.c.Pz) \\ R_{14} &= Pz.c.Pz. \end{aligned}$$

These actives line form the sequent $4, 5 \to 14$. Let Q_4, Q_5, Q_{14} be the trees represented by R_4, R_5, R_{14}, resp. Then we have

$$D_4 := \sim Fm(Q_4) = \sim Pzy = A1$$

$$D_5 := \sim Fm(Q_5) = \sim .Pzy \supset Pz.c.Pz = \sim .A2 \supset A3$$
$$D_{14} := Fm(Q_{14}) = Pz.c.Pz = A4$$

where $A1, A2, A3, A4$ are used to name b-atom occurrences. If we set $\mathcal{M} := \{(A1, A2), (A3, A4)\}$, it is easy to verify that $\mathcal{M}$ is a cs-mating for the set $\{D_4, D_5, D_{14}\}$ and, therefore, for the entire outline. To check if the proviso on D-ModusPonens holds with respect to line 5, we first build the set $\mathcal{D} := \{A2, \sim A1\}$ and then check whether $\mathcal{M}$ is a cs-mating for this set. Since, this is indeed the case, we can apply D-ModusPonens to this outline. If we use the strong form described above, we would add the new line

$$(6) \quad 3 \quad \vdash \quad \exists u\, Pzu \qquad\qquad NJ$$

and change the justification of line 14 to be $RuleP : 5, 6$. The RuleP transformation could be applied to the resulting outline to give the justification of $RuleP : 4$ to the new line 6. The use of *modus ponens* in this case is quite obvious. In fact, even the strong form of the our D-ModusPonens transformation over looked the fact that we did not need to enter the new line 6, since it was essentially already present as line 4. In an implementation of the D- and P- transformations, when the schemes listed above mention that a new proof line should be entered, it should only be entered if it in fact is not already present.

Appendix 2 contains a more involved example where the use of D-ModusPonens is not so obvious.

APPENDIX A

Two Independence Results

Section A.1: Introduction

If we restrict our attention to formulas of $\mathcal{T}$ which are first-order in nature, ET-proofs would essentially satisfy Gentzen's "subformula property" (see [Gentzen35]). Proof structures with this property can provide valuable information concerning the metatheory of the logic being investigated. For example, from the completeness of cut-free LK derivations, Gentzen was able to conclude, using the subformula property of these derivations, that first-order logic is consistent. Because the nature of substitution in HOL is much more complex than in the first-order case, it would be unreasonable to expect to find proof structures for $\mathcal{T}$ with Gentzen's formulation of the subformula property. Instead, let us say that a proof structure for a formula A has the "generalized subformula property" if that structure is composed of subformulas of A or of subformulas of substitution instances of formulas in the structure. Clearly, ET-proofs have the generalized subformula property. So too are the two natural deduction-style proof structures which we presented in Chapter 5. It turns out that completeness results for such proof structures can also provide valuable information concerning the metatheory of $\mathcal{T}$. For example, the refutation system described in [Andrews74] has the generalized subformula property and this was used by Andrews to demonstrate that equality in $\mathcal{T}$ was rather weak and nonextensional. In Section 5.2, we shall repeat his demonstration using ET-proofs rather than refutations to prove these same results. The availability of list representations will make these proofs straightforward. In Section 5.3 we shall use this same property of ET-proofs to show that the Axioms of Choice and Descriptions are not derivable in $\mathcal{T}$. These proofs are very much different than the frame-semantic proofs of these independence results provided by Andrews in [Andrews72a]. The proofs presented there, however, are a bit stronger since he was working in a formulation of HOL which was extensional.

Smullyan's term *analytic*, for describing certain types of proof systems (see [Smullyan68]), is a much better term than the more vague "generalized subformula property." ET-proofs are clearly analytic in Smullyan's sense.

Section A.2: Equality is Nonextensional in $\mathcal{T}$

As is well known for $\mathcal{T}$, the equality relation, $E_{o\alpha\alpha}$, for any type α, can be defined by the formula $[\lambda x_\alpha \lambda y_\alpha \forall U_{o\alpha} \,.Ux \supset Uy]$. We shall generally write the formula $E_{o\alpha\alpha}AB$ in the more familiar infix notation, $A = B$. After we define our notion of a substitution for a list of variables, we shall then prove, using ET-proofs, Theorem 2 of [Andrews74].

A.2.1. Definition. A *substitution* θ for a finite list of variables $x_1, \ldots, x_n$ $(n \geq 0)$ is a set of pairs $\{(x_1, t_1), \ldots, (x_n, t_n)\}$ such that for each $i = 1, \ldots, n$, x_i and t_i have the same type. The *application* of θ to A is the formula $\theta A := \rho([\lambda x_1 \ldots \lambda x_n.A]t_1 \ldots t_n)$. Notice that if $\theta = \{(x, t)\}$ then θA and $\rho(\mathbf{S}^x_t A)$ are the same formula, where the operator $\mathbf{S}^x_t$ was defined in Chapter 2. ∎

A.2.2. **Theorem.** *Let T be the formula $\exists x_1 \ldots \exists x_n \,.A = B$ where A and B are formulas$_\alpha$. $\vdash_{\mathcal{T}} T$ if and only if there is a substitution, θ, for the variables $x_1, \ldots, x_n$ such that θA and θB are the same formula.*

Proof. We can assume that all the variables, $x_1, \ldots, x_n$ are distinct, since we could easily remove the vacuous occurrences of a duplicated variables. First assume that such a substitution $\theta = \{(x_1, t_1), \ldots, (x_n, t_n)\}$ exists. Let R be the list structure

$$(\text{EXP} \;\; (t_1 \;\ldots (\text{EXP} \;\; (t_n \;(\text{SEL} \;\; U \;\; (\supset \; U[\theta A] \; U[\theta B])))) \ldots))$$

where U is a variable$_{o\alpha}$ different from $x_1, \ldots, x_n$ and not free in $t_1, \ldots, t_n, A, B$. Clearly, R represents an expansion tree which is an ET-proof for T.

Now assume that $\vdash_{\mathcal{T}} T$. For the sake of readable, assume that $n = 2$. How to proceed when n is other than 2 will be clear. T must then have an ET-proof Q with a list representation given by

$$\begin{array}{l} (\text{EXP } (t_1 \; (\text{EXP } (s^1_1 \; (\text{SEL } U^1_1 \; R^1_1)) \ldots (s^1_{p_1} \; (\text{SEL } U^1_{p_1} \; R^1_{p_1})))) \\ \qquad \vdots \\ \qquad (t_q \; (\text{EXP } (s^q_1 \; (\text{SEL } U^q_1 \; R^q_1)) \ldots (s^q_{p_q} \; (\text{SEL } U^q_{p_q} \; R^q_{p_q}))))) \end{array}$$

where $t_1, \ldots, t_q$ are formulas of the same type as x_1, $s^1_1, \ldots, s^q_{p_q}$ are formulas of the same type as x_2, $U^1_1, \ldots, U^q_{p_q}$ are distinct variables$_{o\alpha}$ such that U^i_j is not free in either t_i or s^i_j, and if $\theta^i_j := \{(x_1, t_i), (x_2, s^i_j)\}$ then R^i_j is the list structure $(\supset \; A^i_j \; B^i_j)$ where $A^i_j = \theta^i_j(U^i_j A)$ and $B^i_j = \theta^i_j(U^i_j B)$. Now $Fm(Q)$ is truth-functionally equivalent to

$$[A^1_1 \supset B^1_1] \vee \ldots \vee [A^q_{p_q} \supset B^q_{p_q}].$$

Since this must be tautologous there must be integers i, j, k, l such that $1li, klq, 1ljlp_i, 1lllp_k$, and A^i_j and B^k_l are the same formula. Since the head of A^i_j is U^i_j and for B^k_l it is U^k_l, $i = k$ and $j = l$. But this implies that $\theta^i_j A$ and $\theta^i_j B$ are the same formula. Hence, the desired substitution is θ^i_j. **Q.E.D.**

A.2.3. **Corollary.** *The η-rule scheme formulas*

$$(\eta^{\alpha\beta}) \qquad \forall f_{\alpha\beta} \, [\lambda x.fx] = f$$

and, therefore, the Axiom of Extensionality scheme formulas

$$(EX^{\alpha\beta}) \qquad \forall f_{\alpha\beta} \, \forall g_{\alpha\beta} \,.\forall x_\beta \, [fx = gx] \supset f = g$$

are not derivable in $\mathcal{T}$.

Proof. If $(\eta^{\alpha\beta})$ were derivable, then for any particular variable$_{\alpha\beta}$ f, $\vdash_{\mathcal{T}} [\lambda x.fx] = f$. But this contradicts the preceding theorem (when $n = 0$). Also, it is easy to show that $\vdash_{\mathcal{T}} \forall x \,.[\lambda x.fx]x = fx$. Hence, if $EX^{\alpha\beta}$ were provable in $\mathcal{T}$ then we would be able to conclude that $\vdash_{\mathcal{T}} [\lambda x.fx] = f$. Hence, $EX^{\alpha\beta}$ is not derivable in $\mathcal{T}$. **Q.E.D.**

Section A.3: The Axioms of Choice and Descriptions are Not Derivable in $\mathcal{T}$

As in [Andrews72a], let the Axiom of Choice be given by the formula scheme

$$(AC^{\alpha}) \qquad \exists c_{\alpha(o\alpha)}\, \forall p_{o\alpha}\, .[\exists x_{\alpha}\; px] \supset p.cp$$

and let the Axiom of Descriptions be given by the formula scheme

$$(D^{\alpha}) \qquad \exists \iota_{\alpha(o\alpha)}\, \forall x_{\alpha}\, .x = \iota[E_{o\alpha\alpha}x].$$

The fact that $\vdash_{\mathcal{T}} AC^{\alpha} \supset D^{\alpha}$ is easy to verify by providing the following ET-proof. Notice that this ET-proof is not grounded since two of its leaves are labeled with $y = c.Ey$, which is an abbreviation of a universally quantified formula. Also notice that the formula $E_{o\alpha\alpha}y_{\alpha}$ represents the singleton set containing y.

$$\begin{array}{l} (\supset \text{(SEL } \; c \;\; \text{(EXP } \;\; (Ey\; (\supset \text{(EXP } \;\; (y\; \text{(SEL } \;\; U_{o\alpha} \;\; (\supset \; Uy\; Uy)))) \; y = c.Ey)))) \\ \quad \text{(EXP } \;\; (c\; \text{(SEL } \;\; y \;\; y = c.Ey)))) \end{array}$$

If we prove that D^{α} is not derivable, it will immediately follow that AC^{α} is not derivable. Andrews showed that AC^{ι} and D^{ι} were not derivable in a system of higher-order logic which was extensional. His proofs were based on the use of frame semantics. Below we shall prove that AC^{α} and D^{α} are not derivable in $\mathcal{T}$ by a simpler argument which uses the completeness result for ET-proofs. Since $\mathcal{T}$ is nonextensional, Andrews results are actually a bit stronger than the ones presented below.

A.3.4. **Theorem.** *D^{α} is not derivable in $\mathcal{T}$.*

Proof. Let α be any type symbol, and assume that $\vdash_{\mathcal{T}} D^{\alpha}$. Thus D^{α} must have an ET-proof Q with a list representation of the form

$$\begin{array}{l} \text{(EXP } (\iota_1 \; \text{(SEL } x_1 \; \text{(SEL } R_1 \; (\supset \; R_1[\iota_1.Ex_1] \; R_1x_1)))) \\ \qquad\quad \vdots \\ \qquad (\iota_n \; \text{(SEL } x_n \; \text{(SEL } R_n \; (\supset \; R_n[\iota_n.Ex_n] \; R_nx_n))))) \end{array}$$

where $n \geq 1$, $x_1, \ldots, x_n$ are distinct variables$_{\alpha}$, $R_1, \ldots, R_n$ are distinct variables$_{o\alpha}$, and $\iota_1, \ldots, \iota_n$ are formulas$_{\alpha(o\alpha)}$. $Fm(Q)$ is then truth-functionally equivalent to a λ-normal form of

$$[R_1[\iota_1.Ex] \supset R_1x_1] \vee \ldots \vee [R_n[\iota_n.Ex] \supset R_nx_n].$$

But this is tautologous only if there are two integers i, j such that $1li, jln$ and R_ix_i conv $R_j.\iota_j.Ex_j$. This is only possible if $i = j$ and x_i conv $\iota_i.Ex_i$. Using Huet's unification algorithm [Huet75], we can then say that the disagreement pair $\langle x_i, \iota.Ex_i\rangle$, where ι is a some variable$_{\alpha(o\alpha)}$ must have a unifier, $\theta = \{(\iota, \iota_i)\}$. However, Huet's MATCH procedure yields as the only possible unifier the formula $\iota_i = [\lambda w_{o\alpha}.x_i]$. But this implies that the dependency relation $<_Q$ is cyclic — x_i is free in ι_i and is selected in the scope of the ι_i expansion — which contradicts the fact that Q is an ET-proof. Hence, D^{α} has no ET-proof and is, therefore, not derivable in $\mathcal{T}$. **Q.E.D.**

As we mentioned above, the fact that AC^{α} is not derivable in $\mathcal{T}$ follows immediately. A direct proof of this fact, however, could be done in a fashion identical to the above proof.

An Example

If the function $f_{\iota\iota}$ has an iterate $g_{\iota\iota} = f \circ \ldots \circ f$ (where the composition is done $n \geq 1$ times) which has a unique fixed point, then f must have a fixed point. This theorem is easily stated in HOL. First, let $J_{o(\iota\iota)(\iota\iota)}$ represent the relation among functions from individuals to individuals such that Jfg is true if g is an iterate of f. J can be defined as the following λ-term.

$$\lambda f \lambda g \forall p_{o(\iota\iota)} .[pf \wedge \forall h_{\iota\iota} .ph \supset p.\lambda t_\iota.f.ht] \supset pg$$

Also, let the expression $\exists_1 x.Px$ be an abbreviation for the formula

$$\exists x .Px \wedge \forall y .Py \supset y = x.$$

Equality is an abbreviation for the formula $\lambda x \lambda y \forall U .Ux \supset Uy$, as described in the previous appendix. Now let Ψ be the following HOL formula. (See [Andrews71] for a refutation of $\sim\Psi$.)

$$\forall f .[\exists g .Jfg \wedge \exists_1 x .gx = x] \supset \exists y .fy = y$$

In order to deal with abbreviations within proof structures, we shall adopt the following simplistic approach to them. If B is a formula containing abbreviations, and B' is the result of replacing all abbreviations with the formulas which stand for them, then if Q is an expansion tree for B', we shall also say that Q is an expansion tree for B. In the context of outlines, we introduce the Def inference rule which permits us to infer the result of introducing or removing an abbreviation. We also allow the following two transformation rules.

D-Def
Let a be a supporting line with assertion A. If A is a top-level abbreviations, let B be the result of removing that abbreviation from A. Set $R'_b := R_a$ and construct Σ' by replacing line a with the line b in each sequent of Σ.

$\boxed{(a) \quad \mathcal{H} \quad \vdash \quad A \qquad\qquad RuleX}$ $\quad\Longrightarrow\quad$ $\boxed{(b) \quad \mathcal{H} \quad \vdash \quad B \qquad\qquad Def : a}$

P-Def

Let z be a sponsoring line with assertion A. If A is a top-level abbreviation, let B be the result of removing that abbreviation from A. Set $\Sigma' := \Sigma_0 \cup \{\Gamma_z \rightarrow y\}$ and $R'_y := R_z$. ($\Sigma_0 := \Sigma \setminus \{\Gamma_z \rightarrow z\}$.)

(z)	$\mathcal{H}$	$\vdash$	A	NJ

$\Longrightarrow$

(y)	$\mathcal{H}$	$\vdash$	B	NJ
(z)	$\mathcal{H}$	$\vdash$	A	$Def : y$

Now to return to our theorem Ψ. In order to specify an ET-proof for Ψ, we first define some smaller trees.

$$
\begin{aligned}
R_{50} &= (\wedge\ (\text{SEL } U\ (\supset\ U[f.fx]\ U[f.fx]))\\
&\qquad\quad (\text{SEL } h\ (\supset\ (\text{EXP } (\lambda x.V[fx]\ (\supset\ V[f.h.fx]\ V[f.f.hx])))\\
&\qquad\qquad\qquad (\text{SEL } V\ (\supset\ V[f.h.fx]\ V[f.f.hx])))))
\end{aligned}
$$

$$R_7 = (\text{EXP } (\lambda x.U[fx]\ (\supset\ U[f.gx]\ U[fx])))$$

$$R_{51} = (\text{EXP } (U\ (\supset\ U[g.fx]\ U[f.gx])))$$

$$R_{92} = (\text{SEL } U\ (\supset\ U[g.fx]\ U[fx]))$$

$$
\begin{aligned}
R_{100} = (\text{SEL } f\ (\supset\ &(\text{SEL } g\ (\wedge\ (\text{EXP } (\lambda k.k[fx] = f[kx]\ (\supset\ R_{50}\ R_{51})))\\
&\qquad\qquad (\text{SEL } x\ (\wedge\ R_7\\
&\qquad\qquad\qquad (\text{EXP } ([fx]\ (\supset\ R_{92}\ (\text{EXP } (\text{W } (\supset\ W[fx]\ Wx)))))))))\\
&(\text{EXP } (x\ (\text{SEL } w\ (\supset\ W[fx]\ Wx))))))
\end{aligned}
$$

It is easy to checked that R_{100} is an ET-proof for Ψ. We now show how an outline for Ψ can be built by applying outline transformation rules. Assume that we start with the trival outline for Ψ based on R_{100}, where the sole line in this outline is 100. We can then apply the following transformations. Listed with the transformation is the set of sequents associated with the outline which results after the transformation is applied. We shall freely use the λRules without explicitly specifying them.

Transformation	**Sequents**
P-All	$\rightarrow 99$
P-Imp	$1 \rightarrow 98$
D-Exists	$2 \rightarrow 97$
D-Conj	$3, 4 \rightarrow 97$
D-Def	$3, 5 \rightarrow 97$
D-Exists	$3, 6 \rightarrow 96$
P-Exists	$3, 6 \rightarrow 95$
D-Conj	$3, 7, 8 \rightarrow 95$
D-Def	$7, 8, 9 \rightarrow 95$

The lines in the resulting outline are the following.

(1)	1	$\vdash$	$\exists g\ .Jfg \wedge \exists_1 x\ .gx = x$	Hyp
(2)	2	$\vdash$	$Jfg \wedge \exists_1 x\ .gx = x$	Hyp
(3)	2	$\vdash$	Jfg	$RuleP : 2$
(4)	2	$\vdash$	$\exists_1 x.gx = x$	$RuleP : 2$
(5)	2	$\vdash$	$\exists x\ .gx = x \wedge \forall z\ .gz = z \supset z = x$	$Def : 4$
(6)	6	$\vdash$	$gx = x \wedge \forall z\ .gz = z \supset z = x$	Hyp
(7)	6	$\vdash$	$gx = x$	$RuleP : 6$

(8)	6	$\vdash$	$\forall z\ .gz = z \supset z = x$	$RuleP : 6$
(9)	2	$\vdash$	$\forall p\ .[pf \wedge \forall h\ .ph \supset p.\lambda t.f.ht] \supset pg$	$Def : 3$
(95)	$1,2,6$	$\vdash$	$fx = x$	NJ
(96)	$1,2,6$	$\vdash$	$\exists y\ .fy = y$	$\exists G : 95$
(97)	$1,2$	$\vdash$	$\exists y\ .fy = y$	$RuleC : 5,96$
(98)	1	$\vdash$	$\exists y\ .fy = y$	$RuleC : 1,97$
(99)		$\vdash$	$[\exists g\ .Jfg \wedge \exists_1 x\ .gx = x] \supset \exists y\ .fy = y$	$Deduct : 98$
(100)		$\vdash$	$\forall f\ .[\exists g\ .Jfg \wedge \exists_1 x\ .gx = x] \supset \exists y\ .fy = y$	$\forall G : 99$

We now apply the following transformations to this outline.

Transformation	**Sequents**
D-All	$7,9,10 \rightarrow 95$
D-ModusPonens	$7,9 \rightarrow 94$
D-All	$7,11 \rightarrow 94$
D-ModusPonens	$7 \rightarrow 50 \quad 7,51 \rightarrow 92$
D-Thinning	$\rightarrow 50 \quad 7,51 \rightarrow 92$

The resulting outline contains the following additional lines.

(10)	6	$\vdash$	$g[fx] = fx \supset fx = x$	$\forall I : fx, 8$
(11)	2	$\vdash$	$[f[fx] = f[fx] \wedge \forall h\ .h[fx] = f[hx] \supset f[h[fx]] = f[f[hx]]] \supset$ $g[fx] = f[gx]$	$\forall I : \lambda k.k[fx] = f[kx], 9$
(50)	$1,2,6$	$\vdash$	$f[fx] = f[fx] \wedge \forall h\ .h[fx] = f[hx] \supset f[h[fx]] = f[f[hx]]$	NJ
(51)	51	$\vdash$	$g[fx] = f[gx]$	Hyp
(92)	$1,2,6,51$	$\vdash$	$g[fx] = fx$	NJ
(93)	$1,2,6$	$\vdash$	$g[fx] = f[gx] \supset g[fx] = fx$	$Deduct : 92$
(94)	$1,2,6$	$\vdash$	$g[fx] = fx$	RuleP:11,50,93

Let us look closer at the last two transformations. Prior to the application of D-ModusPonens the only sequent in the outline was $7,11 \longrightarrow 94$. The list representations associated with line 7 is given above. The list representations for lines 11 and 94 are given below.

$$R_{11} = (\supset\ R_{50}\ R_{51})$$
$$R_{94} = R_{92}$$

We now must define the following formulas$_o$ in order to determine how D-ModusPonens was used.

$$\begin{aligned}
D_7 &:= \sim Fm(Q_7) = \sim .U[f.gx] \supset U[fx] \\
D_{11} &:= \sim Fm(Q_{11}) = \sim .\quad [[U[f.fx] \supset U[f.fx]] \wedge \\
&\qquad .[V[f.h.fx] \supset V[f.f.hx]] \supset .V[f.h.fx] \supset V[f.f.hx]] \\
&\qquad \supset .U[g.fx] \supset U[f.gx] \\
D' &:= [U[f.fx] \supset U[f.fx]] \wedge .[V[f.h.fx] \supset V[f.f.hx]] \supset .V[f.h.fx] \supset V[f.f.hx] \\
D'' &:= \sim .U[g.f.x] \supset U[f.gx] \\
D_{94} &:= Fm(Q_{94}) = U[g.fx] \supset U[fx]
\end{aligned}$$

Notice that $D_{11} \equiv .D' \wedge D''$. If we rewrite these formulas using names to replace b-atom occurrence within these formulas, we would have something like the following.

$$
\begin{aligned}
D_7 &:= {\sim}Fm(Q_7) = {\sim}.A1 \supset A2 \\
D_{11} &:= {\sim}Fm(Q_{11}) = {\sim}. \quad [[A3 \supset A4] \wedge .[A5 \supset A6] \supset .A7 \supset A8] \\
&\qquad\qquad\qquad\qquad \supset .A9 \supset A10 \\
D' &:= [A3 \supset A4] \wedge .[A5 \supset A6] \supset .A7 \supset A8 \\
D'' &:= {\sim}.A9 \supset A10 \\
D_{94} &:= Fm(Q_{94}) = A11 \supset A12
\end{aligned}
$$

Set $\mathcal{D} := \{D_7, D_{11}, D_{94}\}$. A cs-mating for $\vee\mathcal{D}$ would be

$$\mathcal{M} := \{(A1, A10), (A2, A12), (A3, A4), (A5, A7), (A6, A8), (A9, A11)\}.$$

Notice that $\mathcal{M}$ spans the set $\{D_7, D'\}$. Hence, we can call D-ModusPonens as was specified above. The subsequent call to D-Thinning is caused by the fact that $\mathcal{M}$ actually spans a subset of this set, *i.e.* the set $\{D'\}$. Hence, line 7 is not needed in proving line 50.

At this point in building an outline for Ψ, we have reduced its proof to some simple theorems about equality. None of the transformations permitted to this point do anything special with equality, except for D-Def and P-Def which simply replace it with the formula it stands for. Although this is not a very sophisticated use of equality, it is complete for $\mathcal{T}$. We shall leave the completion of this example to the reader.

Bibliography

[Andrews71] Peter B. Andrews, *Resolution in Type Theory*, Journal of Symbolic Logic 36, 414-432 (1971).

[Andrews72a] Peter B. Andrews, *General Models, Descriptions, and Choice in Type Theory*, Journal of Symbolic Logic 37, 385-394 (1972).

[Andrews72b] Peter B. Andrews, *General Models and Extensionality*, Journal of Symbolic Logic, 37, 395-397, (1972).

[Andrews73] Peter B. Andrews, Letter addressed to Roger Hindley, January 22, 1973.

[Andrews74] Peter B. Andrews, *Provability in Elementary Type Theory*, Zeitschrift für Mathematische Logik und Grundlagen der Mathematik, 20, 411-418 (1974).

[Andrews80] Peter B. Andrews, *Transforming Matings into Natural Deduction Proofs*, in "The 5th Conference on Automated Deduction, Les Arcs, France", edited by W. Bibel and R. Kowalski. LNCS 87 281-292, (1980).

[Andrews81] Peter B. Andrews, *Theorem Proving via General Matings*, Journal of the Association for Computing Machinery, 28, 193-214 (1981).

[Bibel81] Wolfgang Bibel, *On Matrices with Connections*, Journal of the Association of Computing Machinery, 28(4), 633-645 (1981).

[Church40] Alonzo Church, *A Formulation of the Simple Theory of Types*, Journal of Symbolic Logic, 5, 56-68 (1940).

[Church41] Alonzo Church, *The Calculi of Lambda-Conversion.* Princeton University Press, 1941.

[Gentzen35] Gerhard Gentzen, *Investigations into Logical Deductions*, in "The Collected Papers of Gerhard Gentzen", edited. by M. E. Szabo. North-Holland Publishing Co., 1969.

[Henkin50] Leon Henkin, *Completeness in the theory of types*, Journal of Symbolic Logic 15, 81-91 (1950).

[Huet75] Gérard P. Huet, *A Unification Algorithm for Typed λ-Calculus*, Theoretical Computer Science 1, 27-57 (1975).

[Miller82] Dale A. Miller, Eve Longini Cohen, Peter B. Andrews, *A Look at TPS*, in "The 6th Conference on Automated Deduction" edited by Donald W. Loveland. LNCS 138, 50-69 (June 1982).

[Prawitz68] Dag Prawitz, *Hauptsatz for higher order logic*, Journal of Symbolic Logic 33, 452-457 (1968).

[Smullyan68] Raymond M. Smullyan, *First-Order Logic.* Springer-Verlag, New York, 1968.

[Takahashi67] Moto-o-Takahashi, *A proof of cut-elimination theorem in simple type-theory*, Journal of the Mathematical Society of Japan, 19, 399-410 (1967).