

Analytic and Non-analytic Proofs

Frank Pfenning
Department of Mathematics
Carnegie-Mellon University
Pittsburgh, PA 15213

0. Abstract

In automated theorem proving different kinds of proof systems have been used. Traditional proof systems, such as Hilbert-style proofs or natural deduction we call *non-analytic*, while resolution or mating proof systems we call *analytic*. There are many good reasons to study the connections between analytic and non-analytic proofs. We would like a theorem prover to make efficient use of both analytic and non-analytic methods to get the best of both worlds.

In this paper we present an algorithm for translating from a particular non-analytic proof system to analytic proofs. Moreover, some results about the translation in the other direction are reformulated and known algorithms improved. Implementation of the algorithms presented for use in research and teaching logic is under way at Carnegie-Mellon University in the framework of TPS and its educational counterpart ETPS.

Finally we show how to obtain non-analytic proofs from resolution refutations. As an application, resolution refutations can be translated into comprehensible natural deduction proofs.

1. Introduction

In automated theorem proving different kinds of proof systems have been used. Traditional proof systems, such as Hilbert-style proofs or natural deduction we call *non-analytic*, while resolution or mating proof systems we call *analytic*. There are many good reasons to study the connections between analytic and non-analytic proofs. We would like a theorem prover to make efficient use of both analytic and non-analytic methods to get the best of both worlds.

The advantages of analytic proofs are well known. One of the most important advantage is that they seem to be ideally suited for an efficient automatic search for a proof on the computer.

On the other hand there is much to gain from the use of non-analytic proof systems in addition to analytic methods. Non-analytic proofs can be presented in a comprehensible and pleasing format. If we can translate, say, resolution refutations into legible non-analytic proofs, we can help the mathematician understand the automatically generated proof. Valuable work here has been done by Miller [10]. The natural deduction proofs obtained from mating refutations are often elegant and easy to understand and use such mathematically common concepts as proof by contradiction and case-analysis, and make use of intuitive operations such as backchaining. Better translations which are the object of current research would make this even more useful for a wider class of theorems.

The ability to freely translate between analytic and non-analytic proofs also gives us a tool for creating a more elegant natural deduction style proof from a given one. We would

translate a given proof into an analytic proof, possibly transform this analytic proof into a shorter one, and then build a new natural deduction style proof from it in a canonical fashion.

Good translation procedures can also serve as a valuable research tool. Heuristics and lemmas of use to a theorem prover can often be discovered and formulated naturally in some non-analytic proof style. The ability to translate these into an analytic format may help to incorporate them into a theorem prover. Moreover, if we can translate automatic proofs obtained with and without a certain heuristic, we may gain deeper insight into the nature and performance of the heuristics.

Another perhaps more immediately important application is in the use of these procedures in computer-aided instruction in logic. The student will attempt his proof in a deductive format, e.g. in a natural deduction style, on the computer. The analytic proof of the exercise can be found beforehand by an automated theorem prover employing resolution or a mating procedure, or even constructed from a sample natural deduction proof given by the teacher. This analytic proof can then be used to guide the student through his own attempts to prove the theorem by suggesting which inference rules may be appropriate when the student asks for help. Moreover, when the student is done, a "normalizing" procedure like the one described above can demonstrate to the student how he might have proven the theorem more elegantly or efficiently. A system called ETPS, which will contain all these features, is currently under development at Carnegie-Mellon University.

There is also a very good complexity-theoretic reason why a theorem prover may want to make use of non-analytic as well as analytic methods. A result by Statman [14] shows that there are theorems which have "short" non-analytic proofs, but no "short" analytic proofs whatsoever. He exhibits a sequence of theorems (from the theory of combinators) whose shortest possible analytic proof is $2^{2^{\dots^{2^l}}}$ }^d. (d is the number of connectives and quantifiers of a theorem X , and l the length of a non-analytic proof for X .) This lower bound is not Kalmár-elementary, and there are therefore theorems which cannot be practically proven by purely analytic methods which have short non-analytic proofs.

Let us now try to make more precise the distinction between analytic and non-analytic proof systems. The term "analytic" was introduced by Smullyan in [13] and conveys the idea that the proof (or refutation) procedure analyzes the given formula. An analytic proof has a very strong *subformula property*: Only subformulas of the theorem and their instances will appear in an analytic proof.

In the field of automated deduction the discovery of analytic proof systems such as resolution [12] went hand in hand with the beginning of research. The mating approach [3] and a similar method by Bibel [4] are other examples of analytic proof systems.

Examples of non-analytic methods in automated theorem proving can be found in Bledsoe's survey [6] of non-resolution theorem proving. This includes approaches like term-rewriting, built-in inequalities, forward-chaining, models, and even counterexamples. Some of these approaches may be called *non-analytic*, since they sometimes consider formulas not part of the proposed theorem. Many of the stimuli here come from mathematics rather than pure logic. Hilbert-style, Gentzen-style [7], or natural deduction style systems are all examples of traditional non-analytic proof systems. In general they do not obey the subformula property. Usually Cut or Modus Ponens is used to eliminate the helpful formulas, which are not part of the theorem, but substitutivity of equivalence or equality may be used as well. The use of Cut itself does not characterize non-analytic proof systems, as can be seen from the case of resolution, where the cut formulas are all subformulas of the given theorem.

Andrews has shown in [2] how to convert matings into natural deduction proofs. Miller [9] took this work further by generalizing it to higher-order logic and also addressing questions of style in these proofs. Some related work was also done by Bibel in [5]. An algorithm translating in the other direction is the main contribution of this paper. The ability to readily translate in either direction between analytic and non-analytic proofs (in the case of the implementation in TPS between expansion proofs and natural deduction style proofs) gives us all the aforementioned advantages.

As a representative of non-analytic proof systems we pick I^* , mainly for its conceptual clarity and simplicity of cut-elimination. I^* which is described in section 2 is closely related to the system LK of Gentzen [7] and a related system of Smullyan [13].

Following Miller in [9], who works in the setting of higher order logic, we define a purely analytic proof system in section 3. Expansion proofs, as they are called, are very natural and convenient and very concisely represent the information contained in an analytic proof.

In section 4 we give a new exposition of part of Miller's work in terms of our analytic and non-analytic first-order proof systems. This exposition provides the reader with a self-contained and unified treatment of the translations between the various proof styles. We also handle conjunction in a new way, thus creating stylistically different proofs.

As the main part of this paper, we give an explicit algorithm which translates I^* -proofs into expansion proofs in sections 5, 6, and 7. Expansion proofs are very much different from the kind of analytic proofs Gentzen or Smullyan considered, though some of their ideas, in particular for cut-elimination, are used. Our *merge* algorithm which deals with the inference rule *Contraction* is a significantly improved version of Miller's [9] MERGE, which generally produces much larger expansion trees.

Andrews in [1] has given an algorithm which computes a mating from a resolution refutation. In section 8 we state and prove the correctness of a different algorithm which translates resolution refutations into expansion proofs, which do not make use of Skolem-functions or conjunctive normal forms and satisfy a quite different acceptability criterion from Andrews'. We thus give a two-step procedure by which resolution refutations can be translated into I^* -proofs, or, in one more step, into natural deduction proofs.

Space does not permit to include here non-trivial examples illustrating the various algorithms. Detailed examples for all the translation procedures presented here are given by the author in [11].

2. The Systems I and I^*

Our non-analytic proof system is I^* , which builds upon similar systems of Gentzen [7] and Smullyan [13]. I^* is particularly well suited for the description of our algorithms. Notice, for instance, that any theorem derived in I^* is automatically in negation normal form. The work done here can easily be generalized to other superficially richer systems of first-order logic. To simplify some of our exposition we introduce a system I which is identical to I^* but does not contain the rule of *Mix* (a variant of *Cut*).

Our formulation of first-order logic includes the propositional connectives \vee , \wedge , \neg , the quantifiers \exists and \forall and an infinite number of individual variables and constants. Function constants of arbitrary finite arity are also permitted. An atomic formula is of the form $Pt_1 \dots t_n$ for an n -ary predicate P and terms t_1, \dots, t_n . A literal is of the form A or $\neg A$ for an atomic formula A . A formula is in negation normal form if the scope of each negation is

atomic. Each first-order formula has a classically equivalent formula in negation normal form, and we generally assume our formulas to be in negation normal form. $X[v/a]$ is our notation for the result of substituting a for the free occurrences of v in X . We write *nnformula* for a formula in negation normal form. We do not assume that formulas are alphabetically normal, except in section 8 where we talk about resolution refutations. Sometimes we write \times to indicate that an equation is valid for both conjunction and disjunction.

Nodes in a proof-tree in I we call *lines*. A line in I is a multi-set of formulas. This formulation is halfway between Gentzen's (sequents) and Smullyan's (sets). The reason for choosing this particular representation lies in the fact that contraction is an extremely powerful inference rule of our system. When we try to analyze how the effect of a contraction induces a change in an associated expansion tree, we will see that the transformation is really quite complicated. Thus we cannot leave contraction implicit, like Smullyan did, when he introduced sets of formulas as objects in the proof. Structural rules like exchange, however, have no impact on the logical contents of the formula or proof line. We therefore leave them implicit in the multi-set notation.

In general we let U and V stand for multi-sets of formulas, i.e. sets where we allow the same formula to appear more than once as a member. We often write U, X to mean $U \cup \{X\}$ if U is a multi-set.

The axioms of I are of the form

$$U, A, \neg A$$

where A is an atomic formula.

The inference rules can be divided into *structural rules*, *propositional rules*, and *quantificational rules*. The only structural rule in I is *contraction* (C). There is one propositional rule for each propositional connective: \vee -*introduction* ($\vee I$) and \wedge -*introduction* ($\wedge I$). There is also exactly one rule for the quantifiers: \exists -*introduction* ($\exists I$) and \forall -*introduction* ($\forall I$).

Structural rules

$$\text{Contraction: } \frac{U, X, X}{U, X} C$$

Propositional rules

$$\frac{U, X, Y}{U, X \vee Y} \vee I \qquad \frac{U, X \quad V, Y}{U, V, X \wedge Y} \wedge I$$

Quantificational rules

$$\frac{U, X[v/t]}{U, \exists v X} \exists I, \quad t \text{ a term free for } v \text{ in } X.$$

$$\frac{U, X[v/a]}{U, \forall v X} \forall I, \quad a \text{ not free in } U, \forall v X.$$

U, V contain the side-formulas of an inference rule. They may be empty. The propositional and quantificational inference rules correspond to Smullyan's [13] rules $\alpha, \beta, \gamma, \delta$.

System I is complete in the sense that we can derive the negation normal form of every valid formula in classical first order logic. This follows almost immediately from Smullyan's

form of the completeness result for Gentzen systems and we will not repeat the argument here.

We shall also use the system I^* which contains the rule of *Mix*:

$$\frac{U, X, \dots, X \quad V, \bar{X}, \dots, \bar{X}}{U, V} \text{Mix}, \quad X \notin U, \bar{X} \notin V$$

\bar{X} is the negation normal form of $\neg X$. There must be at least one occurrence of X , the *mix* formula, in the left premise and at least one occurrence of \bar{X} in the right premise. *Mix* was introduced by Gentzen and is a variant of the rule of *Cut*, and the two are easily shown to be equivalent.

3. Expansion Trees

Analytic proofs in this paper are presented as expansion trees. Expansion trees very concisely and naturally represent the information contained in an analytic proof, as we hope to show. They were first introduced by Miller [9] and are somewhat similar to Herbrand expansions [8]. Some redundancies can easily be eliminated for an actual implementation as done by Miller in the context of higher order logic. The shallow formula of an expansion tree will correspond to the theorem; the deep formula is akin to a Herbrand-expansion proving the theorem. Our formulation of expansion trees differs only trivially from Miller's in [10], if restricted to first-order logic. At several places it is convenient to allow n-ary conjunction and disjunction instead of treating them as binary operations.

3.1. Definition. We define Expansion Trees inductively. Simultaneously, we also define Q^D , the deep formula of an expansion tree, which is always quantifier-free, and Q^S , the shallow formula of an expansion tree. We furthermore place the restriction that no variable in an expansion tree may be selected more than once.

- (i) A literal l (signed atom) is an expansion tree. $Q^D(l) = Q^S(l) = l$. Literals form the leaves of expansion trees.
- (ii) If $Q_1, \dots, Q_n, n \geq 2$, are expansion trees, so is

$$Q = \begin{array}{c} \text{X} \\ \diagup \quad \diagdown \\ Q_1 \quad \dots \quad Q_n \end{array} \quad \begin{array}{l} \text{Then } Q^D = Q_1^D \text{X} \dots \text{X} Q_n^D, \\ \text{and } Q^S = Q_1^S \text{X} \dots \text{X} Q_n^S. \end{array}$$

- (iii) If Q_1, \dots, Q_n are expansion trees such that If $Q_1^S = S[v/t_1], \dots, Q_n^S = S[v/t_n], t_i$ a term free for v in S for $1 \leq i \leq n, n \geq 1$, then


$$Q = \begin{array}{c} \exists v S \\ \diagup \quad \diagdown \\ t_1 \quad \dots \quad t_n \\ Q_1 \quad \dots \quad Q_n \end{array} \quad \begin{array}{l} \text{is an expansion tree.} \\ \text{Then } Q^D = Q_1^D \vee \dots \vee Q_n^D, \\ \text{and } Q^S = \exists v S. \end{array}$$

$\exists v S$ is called an expansion node; v is the expanded variable; t_1, \dots, t_n are the expansion terms.

- (iv) If Q_0 is an expansion tree such that $Q_0^S = S[v/a]$ for a variable a , so is

$$Q = \begin{array}{c} \forall v S \\ | \\ a \\ | \\ Q_0 \end{array} \quad \text{Then } \begin{array}{l} Q^D = Q_0^D, \\ \text{and } Q^S = \forall v S. \end{array}$$

$\forall v S$ is called a **selection node**; a is the **variable selected** for this occurrence of v .

To improve legibility of our diagrams we will frequently draw  for an expansion tree with $Q^S = X$.

Since traditional proof systems do not contain Skolem-functions, we need a different mechanism to insure the soundness of our proofs. Following an idea of Bibel [4], which was picked up by Miller [9], we introduce a relation $<_Q$ on occurrences of expansion terms. The condition that $<_Q$ be acyclic replaces Skolemization in our analytic proof system. The reason for this definition will become clear in section 4. $<_Q$ is dual to $<$ defined in [10], and it is shown in [9] that they are equivalent. Later in section 8 we shall see how this relates to Skolemization.

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

We define clauses only for quantifier-free nnformulas, since this is the only case we will need.

3.3. Definition. Let X be a quantifier-free nnformula. A clause in X is a list of literal occurrences defined inductively by

- (i) $X = l$, a literal. Then $C = (l)$ is the only clause in X .
- (ii) $X = A \vee B$. Then for all clauses (a_1, \dots, a_n) in A and (b_1, \dots, b_m) in B , $C = (a_1, \dots, a_n, b_1, \dots, b_m)$ is a clause in $A \vee B$.
- (iii) $X = A \wedge B$. Then all clauses in A and all clauses in B are clauses in $A \wedge B$.

3.4. Definition. A relation on literal occurrences in a quantifier-free nnformula X is a **mating** M if $\neg l = k$ for every pair $(l, k) \in M$ and there is at least one clause in X containing both l and k . If $(l, k) \in M$, l and k are said to be M -mated.

3.5. Definition. A mating M is said to **span a clause** C if there are literals $l, k \in C$ such that $(l, k) \in M$. A mating M is said to be **clause-spanning** on a quantifier-free nnformula X if every clause in X is spanned by M .

The significance of this definition is of course that a quantifier-free nnformula X is tautologous iff there is a mating clause-spanning on X (see Andrews [3], [1], and Miller [9]).

3.6. Definition. A pair (Q, M) is called an **expansion tree proof** for a nnformula X if

- (i) $Q^S = X$.
- (ii) No selected variable is free in Q^S .
- (iii) $<_Q$ is acyclic.
- (iv) M , a **mating** on Q^D , is clause-spanning on Q^D .

Our translations establish soundness and completeness of expansion tree proofs with respect to nnformulas. We rely on the soundness and completeness of I^* , which is a simple consequence of results by Smullyan [13]. A similar, but necessarily less constructive argument was carried out by Miller [9] for expansion tree proofs in higher-order logic.

4. Building *I*-Proofs from Expansion Tree Proofs

The algorithm follows ideas of Miller [9], but we provide a different treatment of conjunction. Our algorithm results in shorter proofs than the more naive algorithm that always applies case (vii) below for a conjunction, but we do not achieve the full power of Miller's *focusing* method. In return, our method is computationally faster.

In the exposition below we sometimes assume that there is a unique correspondence between the formulas in a line and an associated expansion tree, even though we like to think of the line as a multi-set where several identical members are indistinguishable. In general it is sufficient to pick any correspondence between those multiple occurrences of a formula in a line and the unique subtrees of the associated expansion tree.

4.1. Definition. A pair (Q, M) is an expansion tree proof for a line $L = X_1, \dots, X_n$ in an *I*-proof iff (Q, M) is an expansion tree proof for $X_1 \vee \dots \vee X_n$.

4.2. Definition. Let (Q, M) be an expansion tree proof for a line L in an *I*-proof, and let X be a subformula of an element in L . Then $Q|_X$ is the part of the expansion tree Q representing X ($Q|_X^S = X$), and $M|_X$ is the restriction of M to pairs both of whose elements lie in $Q|_X^D$. We will sometimes talk about X^D instead of $Q|_X^D$, if the expansion tree Q is clear from the context.

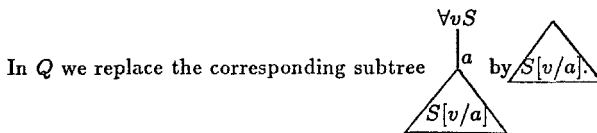
We shall describe an algorithm which constructs an *I*-proof from an expansion tree proof, starting with the formula to be proven and working upwards until every branch in the proof tree begins with an axiom. The cases given below can in principle be applied in any order. The ordering below will often, but not in general, result in the shortest proof that can be constructed with this algorithm.

If an $X \in L$ is such that $Q|_X^D$ has no literal in a pair in M , then X is to be ignored and can only be part of a side-formula in an inference above L .

Now assume L is a given line in an *I*-proof, and (Q, M) is an expansion tree proof for L .

- (i) $L = U, A, \neg A$. Then L is an axiom.
- (ii) $L = U, X \vee Y$. Infer L by $\frac{U, X, Y}{U, X \vee Y} \vee I$. (Q, M) is an expansion tree proof for U, X, Y .
- (iii) $L = U, \forall v S$. Infer L by $\frac{U, S[v/a]}{U, \forall v S} \forall I$,

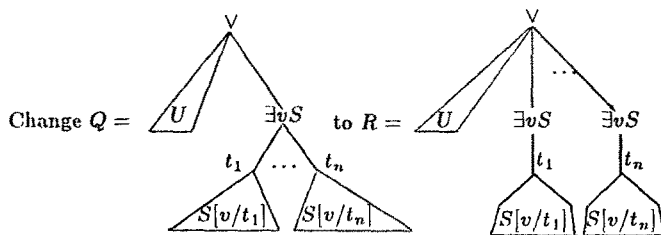
where a is the variable selected for this occurrence of $S[v/a]$.



By definition 3.6 and the inductive assumption that (Q, M) forms an expansion tree proof for $U, \forall v S$, a cannot be free in U or $\forall v S$, since a is a selected variable in Q .

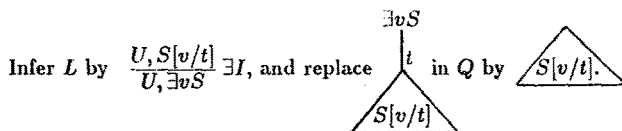
- (iv) $L = U, \exists v S$ and $\exists v S$ has $n, n \geq 2$ successors in Q .

$$\text{Infer } L \text{ by } \frac{U, \exists v S, \dots, \exists v S}{U, \exists v S} (n - 1) \times C.$$



Since $Q^D = R^D$, (R, M) is again an expansion tree proof for $U, \exists vS, \dots, \exists vS$.

- (v) $L = U, \exists vS$, and $\exists vS$ has exactly one successor $S[v/t]$, and no free variable in t is a variable to be selected in Q .



From the restriction on t it is clear that no variable to be selected will be free in $S[v/t]$, and therefore by inductive hypothesis in $U, S[v/t]$.

- (vi) $L = U, V, X \wedge Y$ such that $M = M|_{U,X} \cup M|_{V,Y}$, i.e. no literal in U^D or X^D is M -mated to any literal in V^D or Y^D .

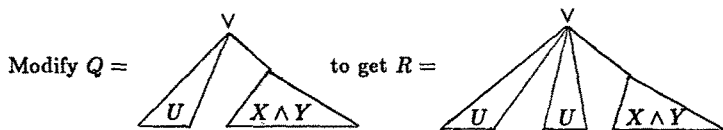
Here we have to consider three subcases.

- (a) $M|_U$ is clause-spanning for U^D . Then restrict the mating to $M = M|_U$. Then no literal in $V, X \wedge Y$ is involved in the mating and they will only appear as side formulas in any inference above L .
- (b) $M|_V$ is clause-spanning on V^D . This case is symmetric to case (a): Let $M = M|_V$.
- (c) Neither case (a) nor case (b) apply. Then infer L by $\frac{U, X \quad V, Y}{U, V, X \wedge Y} \wedge I$.

Since the problem is symmetric, we will simply show that $(Q|_{U,X}, M|_{U,X})$ is an expansion tree proof for U, X . It then follows analogously that $(Q|_{V,Y}, M|_{V,Y})$ is an expansion tree proof for V, Y . The only condition we have to test is whether $M|_{U,X}$ is clause-spanning on $Q|_{U,X}^D$. Let P be a clause in $Q|_{U,X}^D$. Since neither case (a) nor case (b) applies, there is a clause O in V^D not spanned by M . Let P' be the extension of P to a clause in Q^D such that $P'|_V = O$ and $P'|_{U,X} = P$. By inductive assumption, P' is spanned by $(l, k) \in M$. Not both l and k are in V^D , since M does not span O . We also assumed $M = M|_{U,X} \cup M|_{V,Y}$ and hence $(l, k) \in M|_{U,X}$.

- (vii) $L = U, X \wedge Y$ and case (vi) does not apply.

Then infer L by $\frac{U, U, X \wedge Y}{U, X \wedge Y} C$.



For every occurrence of a literal l in U , there are two occurrences of l in U, U . Call these l^1 and l^2 for the occurrences in the left and right copies of U , respectively. Let

$\mathcal{M}|_{U,X}^1 [\mathcal{M}|_{V,Y}^2]$ be the result of replacing every occurrence of a literal l from U^D in $\mathcal{M}|_{U,X}$ $[\mathcal{M}|_{V,Y}]$ by l^1 $[l^2]$. Then $\mathcal{N} = \mathcal{M}|_{U,X}^1 \cup \mathcal{M}|_{V,Y}^2$ spans every clause in R^D . To see this, let P be a clause in R^D . Then P contains literals from either X or Y , but not both. Without loss of generality, assume P contains literals in X , and let O be the clause in Q^D which agrees with P on X and contains a literal l in U^D iff l^1 is in P . By inductive assumption, O is closed by a pair $(k, m) \in \mathcal{M}$. But then also $(k^1, m) \in \mathcal{M}|_{U,X}^1 \subset \mathcal{N}$ (if m is in $Q|_X^D$), or $(k^1, m^1) \in \mathcal{M}|_{U,X}^1 \subset \mathcal{N}$ (if m is in $Q|_{U'}^D$). Thus P is spanned by \mathcal{N} . Since P was arbitrary, \mathcal{N} spans every clause in R^D .

Now the case (vi) can be applied immediately, thus reducing the complexity of $L = U, X \wedge Y$ to the complexities of the lines U, X and U, Y .

Since the size of connected subformulas of the unjustified lines in the I -proof is diminished in each step, all we need to show to prove correctness is that at least one of the cases always applies. One can see that only one problem may arise: all top-level unformulas are existentially quantified, each of them has just one substitution term, and all of the substitution terms contain a free variable which is still to be selected. Since $<_Q$ has no cycles, there is a term t such that for no $s, s <_Q t$. If t contained a free variable a , which were still to be selected, then the node where a is selected has to lie below one of the top-level existential quantifiers in Q . But if s is the substitution term for this node, then by definition 3.2, $s <_Q t$. This is a contradiction, since $<_Q$ is acyclic and therefore case (v) must apply for at least one of the quantifiers.

5. Building Expansion Tree Proofs from I -proofs

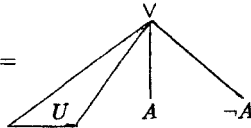
In this section we show how to construct an expansion tree proof from a proof in I . This translation plays an important role in giving a translation procedure from I^* into expansion tree proofs. Some ideas of Miller [9] are used, but we proceed entirely constructively. Also, the procedure for *merge* presented in case (vi) below results in much smaller expansion trees than the ones obtained by Miller's MERGE algorithm. Moreover, because of the way we set up I^* , a merge is necessary only for contraction and not inherently tied to any quantifier or logical connective. This allows a clearer exposition of the ideas which underly the translation from I -proofs into expansion tree proofs.

The construction proceeds by induction on the I -proof tree. Note that all cases except for *Contraction* are very simple. This supports our claim that the expansion tree proof induced by an I -proof corresponds to the I -proof "in a natural way". The basic "idea" underlying the original proof is retained.

We now assume we are given an inference (or axiom) in I , and we have already constructed expansion tree proofs for the premise. We shall call this expansion tree proof (Q, \mathcal{M}) $((Q_1, \mathcal{M}_1)$ and (Q_2, \mathcal{M}_2) in the case of $\wedge I$). The expansion tree proof for the conclusion will be (R, \mathcal{N}) .

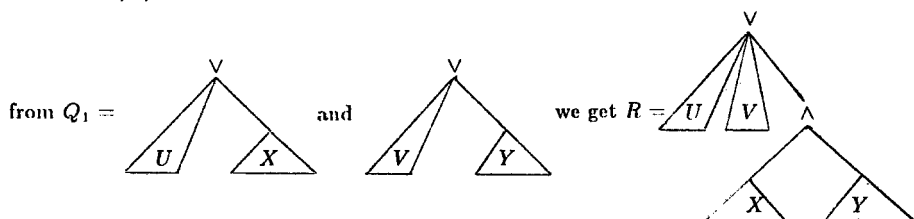
- (i) We have an axiom $U, A, \neg A$.

Then $\mathcal{N} = \{(-A, A)\}$ and $R =$



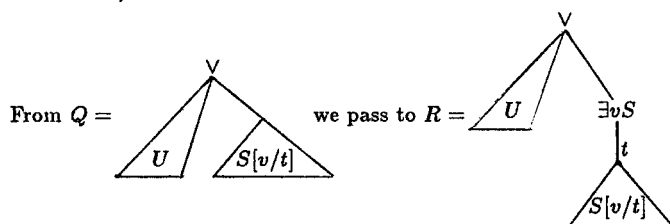
In $Q|_U$, let each existentially quantified variable expand to itself, and select a new unique variable for each universally quantified variable.

- (ii) $\forall I$: $\frac{U, X, Y}{U, X \forall Y}$. Here $(R, \mathcal{N}) = (Q, \mathcal{M})$.
- (iii) $\wedge I$: $\frac{U, X \quad V, Y}{U, V, X \wedge Y}$. Here $\mathcal{N} = \mathcal{M}_1 \cup \mathcal{M}_2$ and



In the new tree we may have to rename the selections for some universal variables, to make sure that no free or selected variable from one branch of the I -proof tree is selected in the other branch.

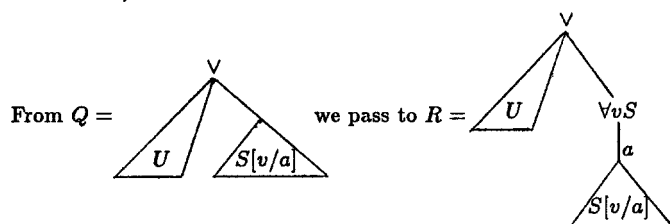
- (iv) $\exists I$: $\frac{U, S[v/t]}{U, \exists v S}$, t free for v in S .



If v does not appear in S , we pick a new variable a to be t , a not selected in Q and not free in U, S .

Since $R^D = Q^D$, we can take $\mathcal{N} = \mathcal{M}$. What remains to be shown in this case is that $<_R$ is acyclic. Let a be a variable selected below $\exists v S$ in R . There may be expansion terms $s_i, t <_R^0 s_i$, but there is no term s such that $s <_R^0 t$. If $s <_R^0 t$ would hold, there had to be a variable b selected in R , and b free in t . But then also b free in $S[v/t]$ (otherwise t was selected to be a new variable), and hence b free in Q^S which contradicts the assumption that (Q, \mathcal{M}) is an expansion tree proof for $U, S[v/t]$.

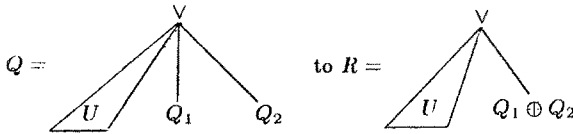
- (v) $\forall I$: $\frac{U, S[v/a]}{U, \forall v S}$, a a variable not free in U or $\forall v S$.



If v does not appear in S , we pick a new variable a not free in U or S or selected in Q . Since $R^D = Q^D$, we can take $\mathcal{N} = \mathcal{M}$. Moreover, since a is not free in $U, \forall v S$, a is a valid selection. Moreover, a could not have been selected in Q , since a occurs free in $S[v/a]$ or had been chosen not to be selected in Q . Thus a is selected in R only once.

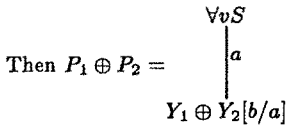
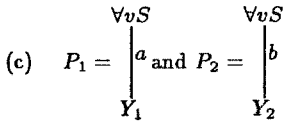
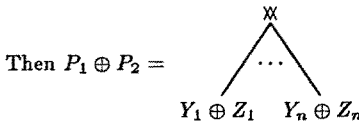
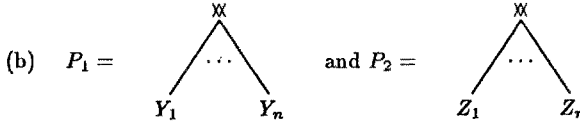
(vi) $C : \frac{U, X, X}{U, X}$

Let Q_1, Q_2 be the subtrees of Q with the root node being the left and right occurrences of X in the premise, respectively. We apply a recursive merging algorithm to obtain an expansion tree $Q_1 \oplus Q_2$ for the single occurrence of X in the conclusion. We will pass from

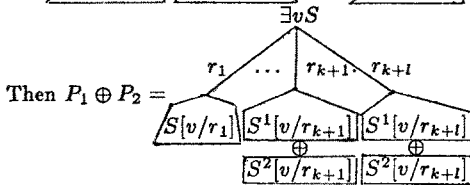
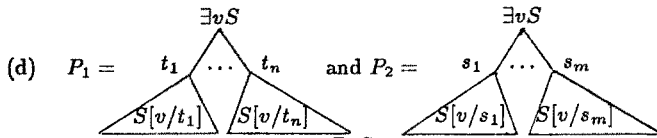


In order to apply \oplus to two expansion trees P_1, P_2 , we require $P_1^S = P_2^S$, which is certainly true of Q_1 and Q_2 .

(a) $P_1 = l_1 = l = l_2 = P_2$. Then $P_1 \oplus P_2 = l$. We say we identify the distinct occurrences of the literal l .



$Y_2[b/a]$ is the result of replacing every occurrence of b in the expansion tree Y_2 by a . But not only do we have to apply this change of names in Y_2 , but in the whole expansion tree in which our merge takes place.



Here r_1, \dots, r_k are the expansion terms which appear only in one of t_1, \dots, t_n and s_1, \dots, s_m ; r_{k+1}, \dots, r_{k+l} are the expansion terms which appear in both. $S^1 [S^2]$ stands for the occurrence of a subtree in $P_1 [P_2]$. If $r_{k+h} = t_i = s_j$ we say that r_{k+h} is the result of identifying the distinct occurrences of the expansion terms t_i and s_j .

We now show by induction on the number of identifications of expansion terms in $Q_1 \oplus Q_2$ that $<_R$ is acyclic. We define a sequence of relations, $<_Q = <^0, <^1, \dots, <^n = <_R$ such that each $<^i, 1 \leq i \leq n$, is acyclic.

Note first that $<^0$ is acyclic, since $<_Q$ is acyclic. If no two literal occurrences were identified during the merge, $<^0 = <_R$ and we are done. Otherwise let p_1, p_2, \dots, p_n be all the literal occurrences in R which result from identifying expansion terms in Q_1 and Q_2 ordered in such a way that $i < j$ whenever p_i is above p_j in R . Now assume we have already defined $<^i$. Let q_1 in Q_1 and q_2 in Q_2 be the expansion terms which were identified to form p_i . We define $t <^{i+1} s$ iff $t <^i s$ or $t = p_i$ and $q_2 <^i s$ or $q_1 <^i s$ bearing in mind that each variable selected below q_1 is also selected below q_2 after merging, since q_1 and q_2 are identified. This can only introduce a cycle into $<^{i+1}$ if $p_i <^{i+1} p_i$ which in turn can only happen if $q_1 <^i q_2$ or $q_2 <^i q_1$. But if for some $s, s <^i q_1$, then also $s <^i q_2$, since q_1 and q_2 have the same free variables. Thus this would mean $q_1 <^i q_1$ or $q_2 <^i q_2$, which is a contradiction to the inductive hypothesis that $<^i$ has no cycles.

One can finally see that $<_R = <^n$, since $t <_R s$ either since $t <_Q s$ or because of one of the identifications of distinct expansion term occurrences. The case where selected variables are being renamed and identified does not contribute any new pairs to $<_R$, since a selection is below a given expansion before identifying the selections iff it is below that expansion after identifying the selections.

To obtain \mathcal{N} on R from \mathcal{M} on Q , we simply identify in \mathcal{M} all literal occurrences which were identified to form one literal occurrence. Then \mathcal{N} spans every clause on R : Let l^\oplus be defined as $l \oplus k$, if l and k are literal occurrences which were identified using case (a) above when forming $Q_1 \oplus Q_2$, otherwise $l^\oplus = l$. Then $\mathcal{N} = \{(l^\oplus, k^\oplus) : (l, k) \in \mathcal{M}\}$. Now let C be a clause in R^D . Then there is a corresponding clause D in Q^D such that $l \in D$ iff $l^\oplus \in C$. D is spanned by a pair $(l, k) \in \mathcal{M}$. But then $(l^\oplus, k^\oplus) \in \mathcal{N}$ and consequently \mathcal{N} spans C .

6. Cut Elimination in I^*

Our cut elimination algorithm is based on similar algorithms of Gentzen [7] and Smullyan [13]. We reformulate these algorithms in terms of the system I^* in order to give a completely self-contained and unified treatment to all the translations between analytic and non-analytic proofs. If one wanted to write out the details of a procedure which computes an expansion tree proof for a formula B , given those for A and $\neg A \vee B$ directly in terms of expansion tree proofs, one could use the cases below in an inductive proof to show that such a direct procedure will result in the same expansion tree proof for B as the less direct procedure described in section 7.

The proof of termination relies on a double induction argument: At each step we transform one mix (which has no other mixes above it) into one or several mixes with lower degree, or, if the degree stays the same, with smaller rank. The degree of a mix is the number of quantifiers and connectives in the mix formula (the formula being eliminated). The left [right]

rank of a mix is the number of lines in the left [right] premise of a mix which contain the mix formulas. The rank of a mix is the sum of left and right rank.

For many of the following cases there is an obvious symmetric case which can be treated completely analogously. It is to be understood that there could be more occurrences of the mix formula in the premises of a mix, but we do not write this out to keep the diagrams as simple as possible. First we consider the case that one of the premises of the mix is an axiom.

- (i) The mix formula is the side-formula of the axiom. Then we eliminate the mix immediately:

$$\frac{U, A, \neg A, X}{U, V, A, \neg A} \frac{V, \bar{X}}{\text{Mix}} \Rightarrow U, V, A, \neg A$$

- (ii) The mix formula is not the side-formula of the axiom. Then we also eliminate the mix:

$$\frac{U, A}{U, V, A} \frac{V, A, \neg A}{\text{Mix}} \Rightarrow \frac{\text{Add } V \text{ as a side-} \\ \text{formula to every inference} \\ \text{above } U, A}{U, V, A}$$

We will now treat the case that the rank of the mix (which contains no other mix above it) is 2.

- (i) The mix formula is a literal A . Since the rank of the mix is 2, one of the previous two cases must apply.

- (ii) $C = X \vee Y, \bar{C} = \bar{X} \wedge \bar{Y}$.

$$\frac{\frac{U, X, Y}{U, \bar{X} \vee \bar{Y}} \forall I \quad \frac{V_1, \bar{X} \quad V_2, \bar{Y}}{V_1, V_2, \bar{X} \wedge \bar{Y}} \wedge I}{U, V_1, V_2} \text{Mix} \Rightarrow \frac{U, X, Y}{U, V_1, Y} \frac{V_1, \bar{X}}{\text{Mix}} \frac{V_2, \bar{Y}}{U, V_1, V_2} \text{Mix}$$

Each of the two new mixes has smaller degree.

- (iii) $C = \forall v S, \bar{C} = \exists v \bar{S}$.

$$\frac{\frac{U, S[v/a]}{U, \forall v S} \forall I \quad \frac{V, \bar{S}[v/t]}{V, \exists v \bar{S}} \exists I}{U, V} \text{Mix} \Rightarrow \frac{\vdots \\ \text{replace} \\ a \text{ by } t}{U, S[v/t]} \frac{V, \bar{S}[v/t]}{U, V} \text{Mix}$$

Note that t is free for v in \bar{X} , hence in X , and therefore replacing a by t is a legal operation, transforming one I -proof into another if we also rename some variables b which are free in t .

Now we consider the case where the rank is greater than 2. We treat the case where the left rank is greater than 1. The case where the right rank is greater than 1 can be treated analogously.

This case again breaks up into two subcases. The new formula on the left hand side of the premise may or may not be the same as the mix formula. First we show how to reduce a mix in case the new formula is not the same as the mix formula. Here we generally reduce the mix to a mix with the same degree but lower rank.

- (i)
$$\frac{\frac{U, A, B, X}{U, A \vee B, \bar{X}} \forall I \quad V, \bar{X}}{U, V, A \vee B} \text{Mix} \Rightarrow \frac{U, A, B, X}{U, V, A, B} \frac{V, \bar{X}}{U, V, A \vee B} \text{Mix}$$

$$(ii) \frac{\frac{U_1, A, X}{U_1, U_2, A \wedge B, \bar{X}, \bar{X}} \wedge I \quad \frac{U_2, B, X}{U_1, U_2, V, A \wedge B} \wedge I \quad V, \bar{X}}{U_1, U_2, V, A \wedge B} Mix \Rightarrow \frac{\frac{U_1, A, X}{U_1, V, A} V, \bar{X} Mix \quad \frac{U_2, B, X}{U_2, V, B} V, \bar{X} Mix}{\frac{U_1, U_2, V, V, A \wedge B}{U_1, U_2, V, A \wedge B} C} \wedge I$$

If X appears in only one premise of the $\wedge I$, this case simplifies in the obvious way.

$$(iii) \frac{\frac{U, A[v/t], X}{U, \exists v A, \bar{X}} \exists I \quad V, X}{U, V, \exists v A} Mix \Rightarrow \frac{\frac{U, A[v/t], X}{U, V, A[v/t]} V, \bar{X} Mix}{U, V, \exists v A} \exists I$$

$$(iv) \frac{\frac{U, A[v/a], X}{U, \forall v A, \bar{X}} \forall I \quad V, \bar{X}}{U, V, \forall v A} Mix \Rightarrow \frac{\frac{U, A[v/a], X}{U, V, A[v/a]} V, \bar{X} Mix}{U, V, \forall v A} \forall I$$

If a happens to be free in V , replace a by a new variable b everywhere above V, \bar{X} .

$$(v) \frac{\frac{U, A, A, X}{U, A, \bar{X}} C \quad V, \bar{X}}{U, V, A} Mix \Rightarrow \frac{\frac{U, A, A, X}{U, V, A, A} V, \bar{X} Mix}{U, V, A} C$$

The last case remaining occurs when the mix formula is also the formula introduced by the last inference rule on the left-hand side. The cases are analogous to the previous ones, except that one mix is now reduced to one mix of lower rank and another mix of left rank 1.

$$(i) \frac{\frac{U, A, B, A \vee B}{U, A \vee B, A \vee B} \vee I \quad V, \bar{A} \wedge \bar{B}}{U, V} Mix \Rightarrow \frac{\frac{U, A, B, A \vee B}{U, V, A, B} V, \bar{A} \wedge \bar{B} Mix}{\frac{U, V, A \vee B}{U, V, V} \vee I \quad V, \bar{A} \wedge \bar{B} Mix}{U, V} C$$

$$(ii) \frac{\frac{U_1, A, A \wedge B}{U_1, U_2, A \wedge B, A \wedge B, \bar{A} \wedge \bar{B}} \wedge I \quad \frac{U_2, B, A \wedge B}{U_1, U_2, V} \wedge I \quad V, \bar{A} \vee \bar{B}}{U_1, U_2, V} Mix \Rightarrow \frac{\frac{U_1, A, A \wedge B}{U_1, V, A} V, \bar{A} \vee \bar{B} Mix \quad \frac{U_2, B, A \wedge B}{U_2, V, B} V, \bar{A} \vee \bar{B} Mix}{\frac{U_1, U_2, V, V, A \wedge B}{U_1, U_2, V} \wedge I \quad V, \bar{A} \vee \bar{B} Mix}{\frac{U_1, U_2, V, V, V}{U_1, U_2, V} 2 \times C} Mix$$

This case simplifies if the mix formula does not appear in both premises of the $\wedge I$.

$$(iii) \frac{\frac{U, \exists v S, S[v/t]}{U, \exists v S, \exists v S} \exists I \quad V, \forall v \bar{S}}{U, V} Mix \Rightarrow \frac{\frac{U, \exists v S, S[v/t]}{U, V, S[v/t]} V, \forall v \bar{S} Mix}{\frac{U, V, \exists v S}{U, V} \exists I \quad V, \forall v \bar{S} Mix}{U, V} C$$

$$(iv) \frac{\frac{U, \forall v S, S[v/a]}{U, \forall v S, \forall v S} \forall I \quad V, \forall v \bar{S}}{U, V} Mix \Rightarrow \frac{\frac{U, \forall v S, S[v/a]}{U, V, S[v/a]} V, \forall v \bar{S} Mix}{\frac{U, V, \forall v S}{U, V} \forall I \quad V, \forall v \bar{S} Mix}{U, V} C$$

$$(v) \frac{\frac{U, X, X, X}{U, X, \bar{X}} C \quad V, \bar{X}}{U, V} Mix \Rightarrow \frac{\frac{U, X, X, X}{U, V} V, \bar{X} Mix}{U, V} C$$

7. Building Expansion Tree Proofs from I^* -proofs

Since we already showed how to construct expansion tree proofs from I -proofs we have only to show how to construct an expansion tree proof, given expansion tree proofs for the two premises of a mix. We emphasize the constructiveness of our approach. Of course we could simply use any theorem proving procedure and arrive at a proof, since we already know we are dealing with a theorem. Our goal, however, is to construct an expansion tree proof which most closely reflects the structure of the two given original proofs, and moreover can be explicitly obtained from them.

Here is our procedure: If we do not already have mix-free I -proofs for both premises, construct them with the algorithm described in section 4. Eliminate the mix from the resulting expansion proof in I^* to obtain a proof in I using the algorithm in section 6. Finally, construct an expansion tree proof from this I -proof using the procedure given in section 5.

In practice we do not have to explicitly construct these I -proofs. The procedure may be reformulated in terms of the expansion tree proofs themselves, but space does not permit to write out the rather laborious details here.

By looking at one of the critical cases, case (i) where a mix of rank 1 is eliminated, one can see the following: If d is the number of quantifiers and connectives in the mix formula (degree of the mix), l is the length of the proof (say, above the leftv premise), and $f(d, l)$ is a worst case lower bound of the length of the resulting mix-free proof, the following relation must hold: $f(d, l) \geq f(\frac{d}{2}, f(\frac{d}{2}, l))$. Thus we get $f(d, l) \geq 2^{2^{\dots^{2^l}}}$ } ^{d} .

Since an I -proof is at most exponentially bigger than a corresponding expansion tree proof, the lower bound remains non-Kalmar-elementary when the resulting I -proof is translated into an expansion tree proof. A result by Statman [14] mentioned in the introduction tells us that this can not be significantly improved. There cannot be a Kalmar-elementary translation from I^* -proofs into I -proofs.

In practice, however, the translation is often feasible and it is not clear which class of theorems will actually blow up the size of the proof by as much as $f(d, l)$.

8. Building Expansion Tree Proofs from Resolution Refutations

When describing the translation procedure from resolution refutations into expansion tree proofs care must be taken to avoid confusion between the different nformulas and the clauses in them. Resolution refutations are stated for the negation of a theorem; expansion tree proofs are defined for the theorem itself. In both cases clauses play a central role. Thus we will call clauses in an expansion tree paths, while clauses in a resolution refutation will be called clauses. We say a path intersects a clause if they have a literal occurrence in common. Notice that our definition of a clause is slightly different from the customary definition as a set. Since matings are relations on literal occurrences, we cannot afford to regard different occurrences of the same literal as identical. During a resolution of two clauses we delete all occurrences of the literal resolved upon. Generally in this section we will assume nformulas also to be $\alpha\beta$ -normal, i.e. no variable occurs both free and bound and each variable is bound at most once.

Andrews [1] described an algorithm which translates resolution refutation into matings, but the setting here is essentially different. We do not work with conjunctive normal forms or Skolem-terms in expansion tree proofs and the condition that matings in expansion tree

proofs must be clause-spanning is also quite different from Andrews' condition that every cycle in a mating must have a merge.

With the aid of this algorithm a resolution refutation can be translated into a non-analytic proof by first translating it into an expansion tree proof and then into a proof in I^* using the algorithm in section 5. This can be carried even further by translating the I^* -proof into a proof in natural deduction style. A procedure for this translation is given by Miller in [10]. This can help a mathematician understand a proof by a resolution theorem prover since he can study it in a familiar format. It may also be a valuable research tool as indicated in the introduction.

8.1. Definition. Let X be an $\alpha\beta$ -normal nnformula. Then X^* , the Skolem-form of X , is the result of replacing every subformula of the form $\exists vS$ by $S[v/f_v(w_1, \dots, w_n)]$, where w_1, \dots, w_n are all the universally quantified variables in whose scope $\exists vS$ lies, and then deleting all the universal quantifiers. $f_v(w_1, \dots, w_n)$ and instances thereof are called Skolem-terms, f_v the Skolem-function for v .

8.2. Definition. Let X be an $\alpha\beta$ -normal nnformula. A resolution refutation of X is a list of clauses c_1, \dots, c_n such that

- (i) $\exists m$ such that $\{c_j : 1 \leq j \leq m\}$ is a subset of the set of clauses of $\overline{X^*}$,
- (ii) for each $j > m$ either
 - (a) c_j is a substitution instance ϕc_i for some $i \leq j$,
 - (b) c_j is the resolvent of c_{a_j} and c_{b_j} , where $a_j, b_j \leq j$, and c_j is formed by appending the results of deleting all occurrences of a literal l_j from c_{a_j} and $\neg l_j$ from c_{b_j} .
 - (c) $c_n = \square$ (the empty clause).

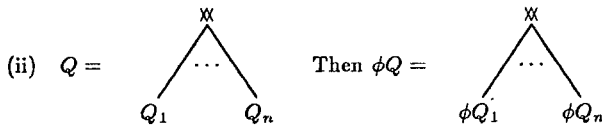
In our translation we will have to select unique variables for Skolem-functions and their arguments. In general, if $f(w_1, \dots, w_n)$ is a Skolem-term for arbitrary terms w_1, \dots, w_n , then $\overline{f(w_1, \dots, w_n)}$ is a unique corresponding variable. Note that this is just a notational convenience in our metalanguage. We must also occasionally model the effect of a substitution into a Skolem-term on the corresponding variables.

8.3. Definition. Let $\overline{f(w_1, \dots, w_n)}$ be a variable, ϕ a substitution for variables which do not come from Skolem-terms. We extend ϕ to terms and formulas in the usual way, but also extend it to act on variables which come from Skolem-terms. Recursively define $\phi \overline{f(w_1, \dots, w_n)} := \overline{f(\phi w_1, \dots, \phi w_n)}$.

We are now ready to define what it means to apply a substitution to an expansion tree. Note that $(\phi Q)^S = \phi(Q^S)$.

8.4. Definition. Let Q be an expansion tree. Then we define ϕQ inductively.

- (i) Q is a literal l . Then $\phi Q = \phi l$.



$$(iii) \quad Q = \begin{array}{c} \exists v S \\ \swarrow \quad \searrow \\ t_1 \quad \dots \quad t_n \\ \swarrow \quad \searrow \\ Q_1 \quad \quad \quad Q_n \end{array}$$

We leave the original expansions intact, and add all terms which change under the substitution as new expansion terms. Let t_{i_1}, \dots, t_{i_m} be all the expansions terms t_i such that $\phi t_i \neq t_i$. Then

$$\begin{array}{c} \exists v S \\ \swarrow \quad \dots \quad \searrow \\ t_1 \quad \dots \quad \phi t_{i_1} \quad \dots \quad \phi t_{i_m} \\ \swarrow \quad \searrow \quad \downarrow \quad \swarrow \quad \searrow \\ Q_1 \quad \quad \quad \phi Q_{i_1} \quad \quad \quad \phi Q_{i_m} \end{array}$$

$$(iv) \quad Q = \begin{array}{c} \forall v S \\ \hline f(w_1, \dots, w_n) \\ \downarrow \\ Q_0 \end{array} \quad \text{Then } \phi Q = \begin{array}{c} \forall v S \\ \hline \phi f(w_1, \dots, w_n) \\ \downarrow \\ \phi Q_0 \end{array}$$

During the translation from resolution refutations to expansion tree proofs we associate an expansion tree and a mating with each line in the resolution refutation. These expansion trees have to satisfy all of the conditions of expansion tree proofs except that the mating does not have to be clause-spanning. We therefore define:

8.5. Definition. A partial expansion tree proof (Q, M) for a nnformula X is an ordered pair consisting of an expansion tree Q and a mating M on Q^D such that

- (i) $Q^S = X$.
- (ii) No selected variable is free in Q^S .
- (iii) $<_Q$ is acyclic.

A particular partial expansion tree will correspond to the part of the resolution proof which is constructed solely from the clauses in the negated and Skolemized theorem.

8.6. Definition. Let X be an $\alpha\beta$ -normal nnformula. The initial expansion tree $\mathcal{Q}(X)$ for X is inductively defined for parts Y of X by

- (i) $Y = l$ for a literal l . Then $\mathcal{Q}(Y) = l$.

$$(ii) \quad Y = Y_1 X \dots X Y_n. \text{ Then } \mathcal{Q}(Y) = \begin{array}{c} X \\ \swarrow \quad \searrow \\ \mathcal{Q}(Y_1) \quad \dots \quad \mathcal{Q}(Y_n) \end{array}$$

$$(iii) \quad Y = \exists v S. \text{ Then } \mathcal{Q}(Y) = \begin{array}{c} \exists v S \\ \downarrow v \\ \mathcal{Q}(S) \end{array}$$

$$(iv) \quad Y = \forall v S. \text{ Then } Q(Y) = \frac{\forall v S}{\left| \frac{f_v(w_1, \dots, w_n)}{Q(S[v/f_v(w_1, \dots, w_n)])} \right.}$$

where $f_v(w_1, \dots, w_n)$ is the Skolem-term for v in \bar{X} .

Now we construct an expansion tree proof from a resolution refutation. Let a resolution refutation $c_1, \dots, c_m, c_{m+1}, \dots, c_n = \square$ be given. For each clause $c_j, j \geq m$ we will recursively construct a partial expansion tree proof (Q_j, M_j) with the following property:

- (*)_j Let $c_i, i \leq j$ be a clause in the resolution refutation. Then every path through Q_j^D which does not intersect c_i contains a pair of M_j -mated literals.

If we can show that (*)_j holds for all $m \leq j \leq n$, the correctness of our translation is proven, since $c_n = \square$ and therefore no path through Q_n^D intersects c_n by (*)_n. Hence every path through Q_n^D must be spanned by M_n and (Q_n^D, M_n) is an expansion tree proof for X .

Now we come to the construction of (Q_j, M_j) .

Let $(Q_m, M_m) = (Q(X), \{\})$. Since every path in $Q(X)^D$ intersects every clause in \bar{X}^* , (Q_m, M_m) is a partial expansion tree proof for X and satisfies (*)_m.

Now assume $(Q_m, M_m), \dots, (Q_{j-1}, M_{j-1})$ are partial expansion tree proofs for X and (*)_i is satisfied for $m \leq i \leq j-1$. We have to distinguish cases, since c_j could either be a substitution instance or a resolvent of earlier clauses.

- (i) Assume c_j is a substitution instance ϕc_i for some $1 \leq i \leq j-1$, ϕ a substitution for the free variables in c_i . If a variable is free in c_i it must be existentially quantified in X . Now we pass to a substitution θ such that θ agrees with ϕ if the substituent is not a Skolem-term, and $\theta v = \bar{f}(w_1, \dots, w_n)$ if $\phi v = f(w_1, \dots, w_n)$.

Let $Q_j = \theta Q_{j-1}$. (Q_j, M_j) is a partial expansion tree proof for X (M_j to be constructed later):

- (a) $Q_j^S = Q_{j-1}^S = X$ by inductive assumption.
 (b) From the way selections for universal variables in X are chosen and from the fact that X was $\alpha\beta$ -normal, it is clear that every variable is selected at most once and that no selected variable is free in Q_j^S .
 (c) $\langle Q_j \rangle$ is acyclic. Assume, to the contrary, that there is a cycle

$$t_1 <_{Q_j}^0 t_2 <_{Q_j}^0 \dots <_{Q_j}^0 t_n = t_1.$$

The first relation means that there is a variable selected below t_1 which is free in t_2 . Since the variable is selected below t_1 in the expansion tree, it has the form of a variable corresponding to a Skolem-term which contains t_1 . Thus t_2 contains a term of the form $\bar{f}_1(\dots, t_1, \dots)$. Hence in the Skolem-form ϕ of the substitution, t_1 is free in t_2 . The next relation would say that there is a variable selected below t_2 which is free in t_3 . Thus a term of the form $\bar{f}_2(\dots, t_2, \dots)$ is free in t_3 . Combined with the previous conclusion this gives us that t_1 is free in t_3 . Iterating this process we finally arrive at the conclusion that t_1 is free in $t_n = t_1$. But this would mean that the original substitution ϕ was not legal, which is a contradiction. Therefore $\langle Q_j \rangle$ must be acyclic.

Now we show how to construct M_j . First note that because of definition 8.4 any literal occurrence in Q_{j-1}^D is still present in Q_j^D . Each *new* literal occurrence in Q_j^D is of the form θl for some l in Q_{j-1}^D . Then we simply let $M_j = M_{j-1} \cup \{(\theta l, \theta k) : (l, k) \in M_{j-1}\}$.

- (a) Consider c_h , $h < j$, P a path through Q_j^D not intersecting c_h . Since paths in Q_j^D can only be longer than paths in Q_{j-1}^D , there is a projection P' of P in Q_{j-1}^D . P' may be obtained by deleting all the new literals from P . Then P' is spanned by M_{j-1} by inductive hypothesis and hence P by $M_j \supset M_{j-1}$.
 - (b) Consider c_j , P a path through Q_j^D not intersecting c_j . Construct a path P' through Q_{j-1}^D as follows: Every literal occurrence l in Q_{j-1}^D such that there is a *new* literal occurrence $\theta l \in P$ is included. Furthermore all literal occurrences such that there is no *new* literal occurrence θl in Q_j^D , but $l \in P$ are also included. Then P' does not intersect c_i and is therefore spanned by a pair $(l, k) \in M_{j-1}$. But then $\theta l, \theta k \in P$ (neither necessarily new) and $(\theta l, \theta k) \in M_j$. Hence P is spanned by M_j .
- (ii) Assume c_j is the resolvent of c_{a_j} and c_{b_j} upon the literal $l_j \in c_{a_j}$, $\neg l_j \in c_{b_j}$, where $a_j, b_j < j$. Define $Q_j = Q_{j-1}$ and let $M_j = M_{j-1} \cup \{(l, k) : l \text{ an occurrence of } l_j \text{ in } c_{a_j}, k \text{ an occurrence of } \neg l_j \text{ in } c_{b_j}\}$.

Since $Q_j = Q_{j-1}$, Q_j is a partial expansion tree proof for X . What remains to be shown is that M_j spans every path through Q_j^D which does not intersect c_i , for all $i \leq j$. For $i < j$ this is obvious by the inductive hypothesis and the fact that $M_j \supset M_{j-1}$.

Now consider a path P through Q_j not intersecting c_j . There are three cases:

- (a) P does not intersect c_{a_j} . By inductive hypothesis $M_{j-1} \subset M_j$ spans P .
- (b) P does not intersect c_{b_j} . By inductive hypothesis $M_{j-1} \subset M_j$ spans P .
- (c) P intersects both c_{a_j} and c_{b_j} . Since P does not intersect c_j , P must intersect c_{a_j} in one of the literal occurrences l_j resolved upon, and c_{b_j} in one of the literal occurrences $\neg l_j$. But then M_j spans P since $(l_j, \neg l_j) \in M_j$.

9. References

- [1] Peter B. Andrews, *Refutations by Matings*, IEEE Transactions on Computers C-25 (1976), 801-807.
- [2] Peter B. Andrews, Transforming Matings into Natural Deduction Proofs, in *5th Conference on Automated Deduction, Les Arcs, France*, edited by W. Bibel and R. Kowalski, Lecture Notes in Computer Science 87, Springer-Verlag, 1980, 281-292.
- [3] Peter B. Andrews, *Theorem Proving via General Matings*, Journal of the Association for Computing Machinery 28 (1981), 193-214.
- [4] Wolfgang Bibel, *Automatic Theorem Proving*, Vieweg, Braunschweig, 1982.
- [5] W. Bibel and J. Schreiber, *Proof search in a Gentzen-like system of first-order logic*, Proceedings of the International Computing Symposium, 1975, pp. 205-212.
- [6] W. W. Bledsoe, *Non-resolution Theorem Proving*, Artificial Intelligence 9 (1977), 1-35.

- [7] G. Gentzen, Investigations into Logical Deductions. In *The Collected Papers of Gerhard Gentzen*, M. E. Szabo, Ed., North-Holland Publishing Co., Amsterdam, 1969, pp. 68-131.
- [8] J. Herbrand, *Logical Writings*, Harvard University Press, 1972.
- [9] Dale A. Miller, *Proofs in Higher Order Logic*, Ph.D. Th., Carnegie-Mellon University, August 1983.
- [10] Dale A. Miller, *Expansion Tree Proofs and Their Conversion to Natural Deduction Proofs*, 7th Conference on Automated Deduction, Napa, May 1984.
- [11] Frank Pfenning. Conversions between Analytic and Non-analytic Proofs. Tech. Report, Carnegie-Mellon University, 1984. (to appear)
- [12] J. A. Robinson, *A machine-oriented logic based on the resolution principle*, Journal of the Association for Computing Machinery 12 (1965), 23-41.
- [13] R. M. Smullyan, *First-Order Logic*, Springer-Verlag, Berlin, 1968.
- [14] R. Statman, *Lower Bounds on Herbrand's Theorem*, Proceedings of the American Mathematical Society 75 (1979), 104-107.