On the Dynamic Increase of Multiplicities in Matrix Proof Methods for Classical Higher-Order Logic

Serge Autexier

Saarland University & German Research Center for Artificial Intelligence (DFKI GmbH), Saarbrücken, Germany, autexier@ags.uni-sb.de

Abstract. A major source of the undecidability of a logic is the number of instances—the so-called multiplicities—of existentially quantified formulas that are required in a proof. We consider the problem in the context of matrix proof methods for classical higher-order logic and present a technique which improves the standard practice of iterative deepening over the multiplicities. We present a mechanism that allows to adjust multiplicities on demand during matrix-based proof search and not only preserves existing substitutions and connections, but additionally adapts them to the parts that result from the increase of the multiplicities.

1 Introduction

A major source of the undecidability of a logic is the number of instances—the so-called multiplicities—of existentially quantified formulas that are required in a proof. Specifying bounds for the *multiplicities* is impossible without sacrificing completeness. Proof procedures based on matrices [5, 2] address this problem by fixing the multiplicities of existential quantifiers, create an initial expansion tree (resp. indexed formula tree) with the fixed multiplicity and then search for an admissible substitution and a spanning set of connections. The standard overall procedure then consists of performing an iterative deepening over the multiplicities until eventually the kernel procedure is successful—in case the conjecture is valid. Restarting the kernel proof procedure implies that substitutions and connections found in previous runs must be established again. Hence, there is a need to integrate the adjustment of the multiplicities into the kernel matrix proof search procedure.

In this article we present a new technique to include the increase of multiplicities into the kernel matrix proof search procedure and to carry over any information about substitutions and connections to the new parts. This technique was developed for extensional higher-order logic as well as for some (first-order) modal logics with constant domains. For sake of brevity we present here only the version for higher-order logic and refer the interested reader to [3].

In order to be complete with respect to higher-order logic, the substitutions need to be applied on the expansion tree, since whole new expansion trees can result from instantiation of literals with head variables, e.g. $X(t)$. Aiming at a complete proof procedure requires that each time a variable is instantiated, the multiplicity of its quantifier must be increased. Increasing the multiplicity while adapting existing connections and substitutions may in turn require to increase the multiplicities of further quantifiers. We present both a mechanism to increase the multiplicities of quantifiers as well as a constructive mechanism to determine, for given a variable which shall be instantiated, the minimal set of quantifiers for which the multiplicities must be increased.

The paper is organized as follows: Sect. 2 recapitulates the basic definitions for classical higher-order logic and Smullyan's uniform notation. Sect. 3 presents a sound and complete extensional expansion proof technique, which integrates the formulation from [2, 9, 10] with *indexed formula trees* from [12]. The dynamic increase of multiplicities is presented in Sect. 4 and illustrated by an example.

2 Preliminaries

For the definition of higher-order logic, we use a simple higher-order type system $\mathcal{T}$ [4], composed of a base type ι for individuals, a type o for formulas, and where $\tau \rightarrow \tau'$ denotes the type of functions from τ to τ'. As usual, we assume that the functional type constructor $\rightarrow$ associates to the right. We annotate constants f_τ and variables x_τ with types τ from $\mathcal{T}$ to indicate their type. A higher-order signature $\Sigma = (\mathcal{T}, \mathcal{F}, \mathcal{V})$ consists of types $\mathcal{T}$, constants $\mathcal{F}$ and variables $\mathcal{V}$, both typed over $\mathcal{T}$. The typed λ-calculus is standard and defined as follows:

Definition 1 (λ-Terms). *Let $\Sigma = (\mathcal{T}, \mathcal{F}, \mathcal{V})$ be a higher-order signature. Then the typed λ-terms $\mathcal{T}_{\Sigma,\mathcal{V}}$ over Σ and $\mathcal{V}$ are: (Var) for all $x_\tau \in \mathcal{V}$, $x \in \mathcal{T}_{\Sigma,\mathcal{V}}$ is a* variable term *of type τ; (Const) for all $c_\tau \in \mathcal{C}$, $c \in \mathcal{T}_{\Sigma,\mathcal{V}}$ is a* constant term *of type τ; (App) if $t : \tau, t' : \tau \rightarrow \tau' \in \mathcal{T}_{\Sigma,\mathcal{V}}$ are typed terms, then $(t'\ t) \in \mathcal{T}_{\Sigma,\mathcal{V}}$ is an* application term *of type τ'; (Abs) if $x : \tau \in \mathcal{V}$ and $t : \tau' \in \mathcal{T}_{\Sigma,\mathcal{V}}$, then $\lambda x_\tau \,\textbf{.}\, t \in \mathcal{T}_{\Sigma,\mathcal{V}}$ is an* abstraction term *of type $\tau \rightarrow \tau'$.*

Definition 2 (Substitutions). *Let $\Sigma = (\mathcal{T}, \mathcal{F}, \mathcal{V})$ be a higher-order signature. A* substitution *is a type preserving function*[1] *$\sigma : \mathcal{V} \rightarrow \mathcal{T}_{\mathcal{F},\mathcal{V}}$ that is the identity function on $\mathcal{V}$ except for finitely many elements from $\mathcal{V}$. This allows for a finite representation of σ as $\{t_1/x_1, \ldots, t_n/x_n\}$ where $\sigma(y) = y$ if $\forall 1 \leq i \leq n, y \neq x_i$.*

As usual we do not distinguish between a substitution and its homomorphic extension. A substitution σ is *idempotent* if, and only if, its homomorphic extension to terms is idempotent, i.e. $\sigma(\sigma(t)) = \sigma(t)$ for all $t \in \mathcal{T}_{\Sigma,\mathcal{V}}$. Given a substitution σ we denote by $dom(\sigma)$ the set of all variables for which $\sigma(x) \neq x$, i.e. the *domain* of σ. Higher-order λ-terms usually come with a certain set of reduction and expansion rules. We use the β reduction rule and the η expansion rule (see [4]), which give rise to the $\beta\eta$ long normal form, which is unique up to renaming of bound variables (α-equal). Throughout the rest of this paper we assume substitutions are idempotent, all terms are in $\beta\eta$ long normal form, and substitution application includes $\beta\eta$-long normalisation.

[1] i.e. for all variables $x : \tau$, $\sigma(x)$ also has type τ.

α	α_0	α_1
$(\varphi \vee \psi)^+$	φ^+	ψ^+
$(\varphi \Rightarrow \psi)^+$	φ^-	ψ^+
$(\varphi \wedge \psi)^-$	φ^-	ψ^-
$(\neg\varphi)^+$	φ^-	$-$
$(\neg\varphi)^-$	φ^+	$-$

β	β_0	β_1
$(\varphi \wedge \psi)^+$	φ^+	ψ^+
$(\varphi \vee \psi)^-$	φ^-	ψ^-
$(\varphi \Rightarrow \psi)^-$	φ^+	ψ^-

γ	$\gamma_0(t)$
$(\forall x\,.\,\varphi)^-$	$(\varphi[x/t])^-$
$(\exists x\,.\,\varphi)^+$	$(\varphi[x/t])^+$

δ	$\delta_0(c)$
$(\forall x\,.\,\varphi)^+$	$(\varphi[x/c])^+$
$(\exists x\,.\,\varphi)^-$	$(\varphi[x/c])^-$

Fig. 1. Uniform Notation.

For the semantics of higher-order logic we use the extensional general models from [7] by taking into account the corrections from [1]. It is based on the notion of *frames* that is a τ-indexed family $\{\overline{D}_\tau\}_{\tau\in\mathcal{T}}$ of nonempty domains, such that $\overline{D}_o = \{\top,\bot\}$ and $\overline{D}_{\tau\to\tau'}$ is a collection of functions mapping $\overline{D}_\tau$ into $\overline{D}_{\tau'}$. The members of $\overline{D}_o$ are called *truth values* and the members of $\overline{D}_\iota$ are called *individuals*. Given a frame $\{\overline{D}_\tau\}_{\tau\in\mathcal{T}}$, an *assignment* is a function ρ on $\mathcal{V}$ such that for each variable x_τ holds $\rho(x_\tau) \in \overline{D}_\tau$. Given an assignment ρ, a variable x_τ and an element $e \in \overline{D}_\tau$ we denote by $\rho[e/x]$ that assignment ρ' such that $\rho'(x_\tau) = e$ and $\rho'(y_{\tau'}) = \rho(y_{\tau'})$, if $y_{\tau'} \neq x_\tau$.

Definition 3 (Satisfiability & Validity). *A formula φ is satisfiable if, and only if, there is a model M and an assignment ρ such that $M^\rho(\varphi) = \top$. φ is* true in a model M *if, and only if, for all variable assignments ρ holds $M^\rho(\varphi) = \top$. φ is valid if, and only if, it is true in all models.*

In order to ease the formulation of our technique for the modal logics from [12], we formulate expansion trees close to the representation used in [12]. [12] uses indexed formula trees, which rely on the concept of polarities and uniform notation [12, 6, 11]. Polarities are assigned to formulas and sub-formulas and are either positive (+) or negative (−). Intuitively, positive polarity of a sub-formula indicates that it occurs in the succedent of a sequent in a sequent calculus proof and negative polarity is for formulas occurring in the antecedent of a sequent.

Formulas annotated with polarities are called *signed formulas*. Uniform notation assigns *uniform types* to signed formulas which encode their "behavior" in a sequent calculus proof: there are two propositional uniform types α and β for signed formulas where the respective sequent calculus decomposition rule results in a split of the proof (β) or not (α). Furthermore there are two types γ and δ for quantification over object variables, respectively for signed formulas where in the respective sequent calculus quantifier elimination rule the *Eigenvariable*-condition must hold (δ) or not (γ). In Fig. 1 we give the list of signed formulas for each uniform type. The labels of the first column indicate the uniform type of the signed formula and the labels of the columns containing the signed subformulas are the so-called *secondary types* of the signed subformulas.

In the following we agree to denote by $\alpha^p(\alpha_1^{p_1}, \alpha_2^{p_2})$ a signed formula of polarity p, uniform type α, and sub-formulas α_i with respective polarities p_i according the tables in Fig. 1. We use an analogous notation for signed formulas of the other uniform types, i.e. for $\gamma^p x\,.\,F^p$ and $\delta^p x\,.\,F^p$. Furthermore we define $\overline{\alpha}^p(F^q) := \alpha^p(F^q)$ and for $n > 1$, $\overline{\alpha}^p(F_1^{p_1}, \ldots, F_n^{p_n}) := \alpha^p(F_1^{p_1}, \overline{\alpha}^{p_2}(F_2^{p_2}, \ldots, F_n^{p_n}))$.

In this paper we are mainly concerned with signed formulas. To ease the presentation we extend the notion of satisfiability to signed formulas. In order to motivate this definition consider a sequent $\psi_1, \ldots, \psi_n \vdash \varphi$. It represents the proof status that we have to prove φ from the ψ_i. In terms of polarities, all the ψ_i have negative polarity while φ has positive polarity. The ψ_i are the assumptions and thus we consider the models that satisfy those formulas and prove that those models also satisfy φ. Hence, we define that a model M satisfies a *negative* formula ψ_i^- if, and only if, M satisfies ψ_i. From there we derive the dual definition for positive formulas, namely that a model M satisfies a positive formula F^+, if, and only if, M *does not* satisfy F. Formally:

Definition 4 (Satisfiability of Signed Formulas). *Let F^p be a signed formula of polarity p, M a model and ρ an assignment. Then we define: $M^\rho \models F^+$* holds if, and only if, $M^\rho \not\models F$. *Conversely, we define $M^\rho \models F^-$* holds if, and only if, $M^\rho \models F$. *We extend that notion to sets of signed formulas $\mathcal{F}$ by: $M^\rho \models \mathcal{F}$, if, and only if, for all $F^p \in \mathcal{F}$, $M^\rho \models F^p$.*

Lemma 1. *Let M be a model, ρ a variable assignment, F^q, G^r signed formulas of polarities q, r and p also some polarity. Then $M^\rho \models \alpha^p(F^q, G^r)$ holds if, and only if, both $M^\rho \models F^q$ and $M^\rho \models G^r$ hold; $M^\rho \models \beta^p(F^q, G^r)$ holds if, and only if, either $M^\rho \models F^q$ or $M^\rho \models G^r$ holds; $M^\rho \models \gamma^q x\,.\,F^q$ holds if, and only if, for all $a \in M_\tau$, $M^{\rho[a/x]} \models F^q$ holds; $M^\rho \models \delta^q x\,.\,F^q$ holds if, and only if, there is an $a \in M_\tau$ such that $M^{\rho[a/x]} \models F^q$ holds.*

Remark 1 (Notational Conventions). We denote formulas by capital Latin letters $A, B, \ldots, F, G, \ldots$, and formulas with holes by Greek letters $\varphi(.)$, which denotes λ-abstractions $\lambda x\,.\,\varphi$ where x occurs *at least once* in φ.

Definition 5 (Literals in Signed Formulas). *Let $\varphi(F)^p$ be a signed formula of polarity p, and F occurs exactly once. We say that F is a* literal *in $\varphi(F)^p$ if using the rules from Fig. 1 we can assign a polarity to F, but not to its sub-terms.*

3 Extensional Expansion Proofs

This section introduces extensional expansion trees in the style of the indexed formula trees from [12]. First we define the extensional expansion tree obtained initially from a signed formula, which we denote by *initial extensional expansion tree*. In a second step we define the transformation rules that insert new subtrees to the subtree, perform instantiations, and draw connections. The first kind of rules are those dealing with the introduction of Leibniz' equality, extensionality introduction for equations and equivalences, expansion of positive equivalences as well as the introduction of cut.

Definition 6 (Initial Extensional Expansion Tree). *We denote extensional expansion trees (EET) for some signed formula F^p by $\triangle^{F^p}$; we say that F^p is the* label *of $\triangle^{F^p}$. In addition to the label of an EET $\triangle$, we define the* deep signed formula *$\mathsf{DF}(\triangle)$ contained in $\triangle$.*

– *Any literal F^p is a leaf node of an EET of polarity p and label F^p; it has no uniform type and $\mathsf{DF}(F^p) := F^p$.*

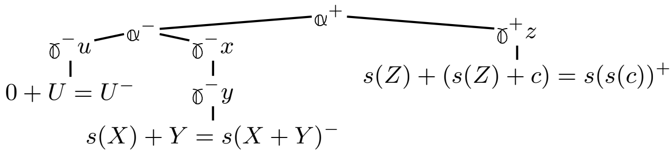


Fig. 2. Initial extensional expansion tree for the running example.

- *If $\triangle^{F^p}$ is an EET for F^p, then $\mathfrak{a}^{-p}(\triangle^{F^p})$ is an EET for $\alpha^{-p}(F^p)$ of polarity $-p$ and uniform type α. The secondary type of $\triangle^{F^p}$ is α_1 and $\mathit{DF}(\mathfrak{a}^{-p}(\triangle^{F^p}))$ $:= \alpha^{-p}(\mathit{DF}(\triangle^{F^p}))$.*
- *If $\triangle^{F^p}, \triangle^{G^q}$ are EETs respectively for F^p and G^q, then $\mathfrak{a}^r(\triangle^{F^p}, \triangle^{G^q})$ is an EET for $\alpha^r(F^p, G^q)$ of polarity r and uniform type α. The secondary types of $\triangle^{F^p}$ and $\triangle^{G^q}$ are α_1 and α_2, respectively, and $\mathit{DF}(\mathfrak{a}^r(\triangle^{F^p}, \triangle^{G^q})) := \alpha^r(\mathit{DF}(\triangle^{F^p}), \mathit{DF}(\triangle^{G^q}))$.*
- *If $\triangle^{F^p}, \triangle^{G^q}$ are EETs respectively for F^p and G^q, then $\mathfrak{b}^r(\triangle^{F^p}, \triangle^{G^q})$ is an EET for $\beta^r(F^p, G^q)$ of polarity r and uniform type β. The secondary types of $\triangle^{F^p}$ and $\triangle^{G^q}$ are β_1 and β_2, respectively, and $\mathit{DF}(\mathfrak{b}^r(\triangle^{F^p}, \triangle^{G^q})) := \beta^r(\mathit{DF}(\triangle^{F^p}), \mathit{DF}(\triangle^{G^q}))$.*
- *If $\triangle^{\varphi(X_i)^p}$, $1 \leq i \leq m$ are EETs for $\varphi(X_i)^p$, then $\mathfrak{g}^p x(\triangle^{\varphi(X_1)^p}, \ldots, \triangle^{\varphi(X_m)^p})$ is an EET for $\gamma^p x \,.\, \varphi(x)^p$ of polarity p, uniform type γ and multiplicity m. The secondary types of the $\triangle^{\varphi(X_i)^p}$ are γ_0 and $\mathit{DF}(\mathfrak{g}^p x(\triangle^{\varphi(X_1)^p}, \ldots, \triangle^{\varphi(X_m)^p})) := \overline{\alpha}^p(\mathit{DF}(\triangle^{\varphi(X_1)^p}), \ldots, \mathit{DF}(\triangle^{\varphi(X_m)^p}))$. We say that the X_i are γ-variables respectively bound on $\triangle^{\varphi(X_i)^p}$.*
- *If $\triangle^{F^p}$ is an EET for F^p, then $\mathfrak{d}^p x \triangle^{F^p}$ is an EET for $\delta^p x \,.\, F^p$ of polarity p and uniform type δ. The secondary type of $\triangle^{F^p}$ is δ_0 and $\mathit{DF}(\mathfrak{d}^p x \triangle^{F^p}) := \mathit{DF}(\triangle^{F^p})$. We say that x is a δ-variable.*

Remark 2. We denote by $\triangle_{\triangle^F}$ an extensional expansion tree which contains $\triangle^F$ as a subtree (not necessarily direct). We say that an extensional expansion tree has a *singular* multiplicity, if all its γ-type subtrees have multiplicity 1. Furthermore, for a given (initial) extensional expansion tree we denote the set of subtrees of secondary type γ_0 by Γ_0, and analogously we define Δ_0. Two subtrees $\triangle_1$ and $\triangle_2$ of some extensional expansion tree $\triangle$ are *α-related*, if, and only if, the smallest subtree of $\triangle$ which contains both $\triangle_1$ and $\triangle_2$ has uniform type α.

Example 1. As an example for an initial extensional expansion tree we consider the formula about natural numbers

$$((\forall u \,.\, 0 + u = u) \wedge \forall x \,.\, \forall y \,.\, s(x) + y = s(x + y)) \Rightarrow \exists z \,.\, s(z) + (s(z) + c) = s(s(c))$$

where 0_ι is the zero of natural numbers, $s_{\iota\to\iota}$ denote the successor and $+_{\iota\times\iota\to\iota}$ denotes the sum over natural numbers. Fig. 2 shows the initial extensional expansion tree with singular multiplicity for the positive formula.

During proof search the γ-variables bound on nodes of secondary type γ_0 are instantiated. Following [9, 10, 12] the admissibility of a substitution σ is realized as an acyclicity check of a directed graph obtained from the structure of the extensional expansion tree and dependencies between binding nodes of the instantiated γ-variable X and binding nodes of δ-variables occurring in $\sigma(X)$.

To this end we introduce the notions of *structural orderings* $\prec$ induced by the structure of an extensional expansion tree and of *quantifier orderings* $\prec$ induced by the substitution.

Definition 7 (Structural Ordering & Quantifier Ordering). *Let $\triangle$ be an extensional expansion tree and σ an (idempotent) substitution for γ-variables bound on γ_0-type nodes in $\triangle$ by terms containing only γ- and δ-variables also bound in $\triangle$. The* structural ordering $\prec$ *is a binary relation among the nodes in $\triangle$ defined by: $\triangle_1 \prec_\triangle \triangle_2$ iff $\triangle_1$ dominates $\triangle_2$ in $\triangle$. The* quantifier ordering $\prec^\sigma$ *induced by σ is the binary relation defined by: $\triangle_0 \prec^\sigma \triangle_1$ iff there is an $X \in dom(\sigma)$ bound on $\triangle_1$ and in $\sigma(X)$ occurs a δ-variable bound on $\triangle_0$.*

Definition 8 (Reduction-Relation & Admissible Substitutions). *Let $\triangle$ be an extensional expansion tree and σ a substitution. Then the* reduction relation $\lhd^\sigma_\triangle$ *is the transitive closure of the union of $\prec_\triangle$ and $\prec^\sigma$, i.e. $\lhd^\sigma_\triangle := (\prec_\triangle \cup \prec^\sigma)^+$. σ is* admissible *with respect to $\triangle$, if and only if $\lhd^\sigma_\triangle$ is irreflexive.*

The notion of admissibility of substitutions is equivalent to the notion defined in [12] if substitutions σ are idempotent. With respect to [9, 10] our notion corresponds to the *dependency relation* among the instances of γ-variables.

Following [12, 2] we define *(horizontal) paths* on extensional expansion trees.

Definition 9 (Paths). *Let $\triangle$ be an extensional expansion tree. Then a* path *in $\triangle$ is a sequence $\ll \triangle_1, \ldots, \triangle_n \gg$ of α-related subtrees in $\triangle$. The set of deep signed formulas of a path p is given by $\mathsf{DF}(p) := \{\mathsf{DF}(\triangle) \mid \triangle \in p\}$. The sets $\mathcal{P}(\triangle)$ of paths through $\triangle$ is the smallest set containing $\{\ll \triangle \gg\}$ and which is closed under the following operations: If $P \cup \{\ll \Gamma, \triangle' \gg\} \in \mathcal{P}(\triangle)$ then*

(α-Dec) *If $\triangle' := \mathbb{a}^p(\triangle_1, \triangle_2)$, then $P \cup \{\ll \Gamma, \triangle_1, \triangle_2 \gg\} \in \mathcal{P}(\triangle)$;*
(β-Dec) *If $\triangle' := \mathbb{b}^p(\triangle_1, \triangle_2)$, then $P \cup \{\ll \Gamma, \triangle_1 \gg, \ll \Gamma, \triangle_2 \gg\} \in \mathcal{P}(\triangle)$;*
(γ-Dec) *If $\triangle' := \mathbb{g}^p x(\triangle_1, \ldots, \triangle_n)$, then $P \cup \{\ll \Gamma, \triangle_1, \ldots, \triangle_n \gg\} \in \mathcal{P}(\triangle)$.*
(δ-Dec) *If $\triangle' := \mathbb{d}^p x(\triangle'')$, then $P \cup \{\ll \Gamma, \triangle'' \gg\} \in \mathcal{P}(\triangle)$.*

Definition 10 (Satisfiable & Unsatisfiable Ground Paths). *A path p is* ground *if all signed formulas in $\mathsf{DF}(p)$ are ground. A ground path p* is satisfiable *if there exists a model M such that $M \models \mathsf{DF}(p)$. Otherwise p is* unsatisfiable.

Corollary 1. *Let p be a path, such $\mathsf{DF}(p)$ contains either $\top^+$, or $\bot^-$ or signed formulas F^- and F^+. Then any ground path for p is unsatisfiable, and so is p.*

Note the close relationship between the decomposition rules for paths for the different types of trees and the relationship between the satisfiability of the main signed formulas with respect to its constituent signed formulas: a path containing an α-type tree is replaced by a path containing both subtrees, while a path containing a β-type tree is decomposed into two paths each containing one of the subtrees. Analogously the decomposition of γ- and δ-type trees corresponds to the relationship between the satisfiability of the signed formula of the respective type to its constituent signed formulas. This relationship together with the acyclicity of the reduction ordering $\lhd^\sigma_\triangle$ entail that whenever we have obtained a set of paths where all paths are unsatisfiable, then the initial conjecture is valid. Indeed, if all paths are unsatisfiable, we can apply the model satisfiability rules backwards from the constituent signed formulas to the main signed formulas and obtain that there is no model M satisfying φ^+, i.e. $\forall M \,.\, M \not\models \varphi^+$, and thus it holds by definition $\forall M \,.\, M \models \varphi$. Hence φ is valid.

Analyzing the decomposition rules and the respective satisfiability relations in more detail, we observe that they are equivalence transformations which is necessary for our soundness and safeness results. *Safeness* means intuitively that no possible refutation is lost by such a transformation. The decomposition rules have this property and this allows to switch freely between the granularity of the decomposition of paths. However, the definition of further transformations on extensional expansion trees requires a weaker property which only requires the preservation of satisfiable paths during the transformation, which is the general condition for *soundness*. Formally:

Definition 11 (Soundness & Safeness). *Let $\triangle$, $\triangle'$ be two extensional expansion trees with respective admissible substitutions σ and σ', and $\triangle'$ results from $\triangle$ by some transformation. The transformation is* sound *if, and only if, if there is a satisfiable path in $\triangle$ then so there is in $\triangle'$. The transformation is* safe *if, and only if, if there is a satisfiable path in $\triangle'$ then so there is one in $\triangle$.*

Definition 12 (Connections). *Let $\triangle$ be an extensional expansion tree and p a path through $\triangle$. A* connection *of p is a pair of subtrees $(\triangle^{F^+}, \triangle^{F^-})$ of opposite polarities, of the same label F (α-equal) and which are both in p. A set $\mathcal{C}$ of connections is* spanning *for $\triangle$, if, and only if, there is a set of paths P through $\triangle$, such that each path in P either contains a connection from $\mathcal{C}$ or $\top^+$ or $\bot^-$.*

We denote a proof state of an extensional expansion proof by $\triangle; \sigma \triangleright \mathcal{C}$: It consists of an extensional expansion tree $\triangle$, an admissible substitution σ and a set of connections $\mathcal{C}$. For a given closed formula we denote by $\triangle^{F^+,\mu}; id \triangleright \emptyset$ the initial proof state, where $\triangle^{F^+,\mu}$ is the initial extensional expansion tree for F^+, μ the multiplicities of the γ-type subtrees and id denotes the empty substitution.

The extensional expansion proof procedure is defined by a set of transformation rules $\frac{\pi'}{\pi}$, which are to be read bottom-up, i.e. the problem of finding a spanning set of connections for π is reduced to find a spanning set of connections for π'. A rule is sound (resp. safe), if the transformation on the extensional expansion tree together with the substitution is sound (resp. safe) and the set of connections are indeed connections for the respective expansion trees. The rules are given in Fig. 3 and consist of (1) an axiom rule, (2) a rule to insert a connection, (3) the introduction of Leibniz' equality for equations and equivalences, (4) two rules for functional and boolean extensionality introduction, (5) a rule to expand positive equivalences into a conjunction of implications, (6) a substitution rule, and (7) a rule to introduce cuts.

To describe the transformation rules, we write $\mathfrak{a}(\triangle, \triangle')$ to denote an expansion tree constructed from $\triangle$ and $\triangle'$ of uniform type α and *which has the same label and polarity than its first subtree* $\triangle$. For instance, $\mathfrak{a}(s = t^p, \triangle^{\forall P \,.\, P(s) \Rightarrow P(t)^p})$ is a subtree of uniform type α, *polarity* p and *label* $s = t^p$.

$$\frac{-}{\triangle; \sigma \triangleright \mathcal{C}}\ \mathit{Axiom}$$

If $\mathcal{C}$ is a spanning set of connections for $\triangle$

$$\frac{\triangle'; \sigma' \triangleright \{(\triangle^{F^-}, \triangle^{F^+})\} \cup \mathcal{C}}{\triangle; \sigma \triangleright \mathcal{C}}\ \mathit{Connect}$$

If $\triangle^{F^-}$ and $\triangle^{F^+}$ are α-related in $\triangle$

$$\frac{\triangle_{\mathfrak{a}(s=t^p, \triangle^{\forall q \,\blacksquare\, q(s) \Rightarrow q(t)^p})}; \sigma \triangleright \mathcal{C}}{\triangle_{s=t^p}; \sigma \triangleright \mathcal{C}}\ \mathit{Leibniz}_=$$

$$\frac{\triangle_{\mathfrak{a}(F \Leftrightarrow G^p, \triangle^{\forall q \,\blacksquare\, q(F) \Rightarrow q(G)^p})}; \sigma \triangleright \mathcal{C}}{\triangle_{F \Leftrightarrow G^p}; \sigma \triangleright \mathcal{C}}\ \mathit{Leibniz}_\Leftrightarrow$$

$$\frac{\triangle_{\mathfrak{a}((F \Leftrightarrow G)^+, \triangle_{(F \Rightarrow G) \wedge (G \Rightarrow F)^+})}; \sigma \triangleright \mathcal{C}}{\triangle_{(F \Leftrightarrow G)^+}; \sigma \triangleright \mathcal{C}}\ \Leftrightarrow\text{-}\mathit{Elim}$$

$$\frac{\sigma'(\triangle); \sigma' \circ \sigma \triangleright \sigma'(\mathcal{C})}{\triangle; \sigma \triangleright \mathcal{C}}\ \mathit{Subst}$$

if $\sigma' \circ \sigma$ admissible wrt. $\triangle$.

$$\frac{\triangle_{\mathfrak{a}(s=t^p, \lambda x \,\blacksquare\, s = \lambda x \,\blacksquare\, t)}; \sigma \triangleright \mathcal{C}}{\triangle_{s=t^p}; \sigma \triangleright \mathcal{C}}\ \mathit{f\text{-}Ext}$$

if x local to $s = t$ in $\triangle_{s=t^p}$.

$$\frac{\triangle_{\mathfrak{a}(A \Leftrightarrow B^p, \lambda x \,\blacksquare\, A = \lambda x \,\blacksquare\, B)}; \sigma \triangleright \mathcal{C}}{\triangle_{A \Leftrightarrow B^p}; \sigma \triangleright \mathcal{C}}\ \mathit{b\text{-}Ext}$$

if x local to $A \Leftrightarrow B$ in $\triangle_{A \Leftrightarrow B^p}$.

$$\frac{\triangle_{\mathfrak{a}(\triangle^{F^p}, \triangle^{\gamma^p \vec{x}' \,\blacksquare\, \beta^p(A^-, A^+)})}; \rho \circ \sigma \triangleright \mathcal{C}}{\triangle_{\triangle^{F^p}}; \sigma \triangleright \mathcal{C}}\ \mathit{Cut}\ A$$

where $\vec{x}'$ are new variables for the free γ- and δ-variables of A not bound above $\triangle^{F^p}$ and $\rho := [\vec{x}'/\vec{x}]$

Fig. 3. The Extensional Expansion Proof Rules $\mathcal{EEP}$

Axiom. The axiom rule terminates a proof, if the derived set of connections $\mathcal{C}$ is spanning for $\triangle$. The soundness is a direct consequence of Definition 12 and Corollary 1 and the admissibility of the substitution σ.

Insertion of Connections. Assume $\triangle^{F^+}$ and $\triangle^{F^-}$ are α-related subtrees in $\triangle$. Then there is a path through $\triangle$ which contains both $\triangle^{F^+}$ and $\triangle^{F^-}$, and hence we can make a connection between them. A connection between non-leaf trees $\triangle^{F^+}$ and $\triangle^{F^-}$ can always be propagated to connections between the subtrees. This possibly requires to increase the multiplicity of some γ-type node and applying a substitution in order to represent the connection, as is illustrated by the following two EETs for the same formula $\forall x \,\blacksquare\, F(x)$, but with opposite polarities: $\mathfrak{g}^- x(F(a)^-)$ and $\mathfrak{d}^+ z(F(z)^+)$. Note that we have already instantiated the γ-variable x in the first tree by some a. Assuming the trees are α-related, we can make a connection between them[2]. In order to propagate the connection into the subtrees, we have to add an additional child to the first tree and instantiate it with the δ-variable z from the second tree, which results in $\mathfrak{g}^- x(F(a)^-, F(z)^-)$. The new inserted subtree is maximal with respect to the $\lhd$ and remains maximal after subtitution, since its root–the binding position of the substituted variable–must be greater than the binding position of the substituted Eigenvariable. Hence the new substitution is admissible with respect to the new EET. For the special nodes $\mathfrak{a}(s = t, \triangle')$ the connections are propagated into $\triangle'$ [3] . The connection

[2] Note that whenever on one side we have a quantifer of type γ, the counter-part is a δ-type formula.

[3] This may require to add Leibniz' definition of equality for leaf nodes.

rule we use always propagates connections to the leaf-nodes and therefore may change $\triangle$ and extend the overall substitution. The rule is both sound and safe.

$\Leftrightarrow$-Elimination, Introduction of Leibniz' Equality. These rules are only necessary since we the logical symbols "=" and "$\Leftrightarrow$" are part of your logical signature. The first replaces an equivalence $A \Leftrightarrow B$ by $(A \Rightarrow B) \wedge (B \Rightarrow A)$ and the latter replace equations $s = t$ and equivalences $A \Leftrightarrow B$ by Leibniz Definition of equality, i.e. $\forall q \,.\, q(s) \Rightarrow q(t)$ and $\forall q \,.\, q(A) \Rightarrow q(B)$. Due to the presence of the latter rules, the $\Leftrightarrow$-elimination rule can be safely restricted to positive formulas.

Functional and Boolean Extensionality. The functional and Boolean extensionality rules *f-Ext* and *b-Ext* require the variable x to be local with respect to the equation (resp. equivalence). This intuitively means that if x is a γ-variable, then it does not occur in a part which is β-related to the equation (resp. equivalence). If x is a δ-variable, then it does not occur in a part which is α-related[4].

The functional extensionality rule creates for a literal leaf node $s = t^p$, for which a variable x is local, a leaf node for $\lambda x \,.\, s = \lambda x \,.\, t^p$ and inserts it to $s = t^p$ by some node of uniform type α and label $s = t^p$. The Boolean extensionality rule is analogous for some literal leaf node $F \Leftrightarrow G^p$. The substitution remains admissible and the set of connections is still valid for the new expansion tree. Any path p which contained $s = t^p$ in the old expansion tree becomes $\ll p, \lambda x \,.\, s = \lambda x \,.\, t^p \gg$. It is easy to see that p is satisfiable if, and only if, $\ll p, \lambda x \,.\, s = \lambda x \,.\, t^p \gg$ is satisfiable.

Cut. The rule allows to insert a cut over some formula A at some arbitrary subtree $\triangle^{F^p}$ of the expansion tree. Assume $\overrightarrow{x}$ are the γ- and δ-variables that occur free in A and which are not bound by some binding node above $\triangle^{F^p}$, $\overrightarrow{x}'$ some new variables and $\rho := \{\overrightarrow{x}'/\overrightarrow{x}\}$. It creates an initial expansion tree for the signed formula $\gamma^p \overrightarrow{x}' \,.\, \beta^p(A^-, A^+)$ and inserts it on $\triangle^{F^p}$ by some α-type node of polarity p and label F^p. The admissibility of the substitution is trivially preserved since ρ creates a dependency only to subtrees adjacent to $\triangle^{F^p}$. Furthermore, any path p which contained F^p is split into two paths $\ll p, A^- \gg$ and $\ll p, A^+ \gg$. Hence, p is satisfiable, if, and only if, either $\ll p, A^- \gg$ or $\ll p, A^+ \gg$ is.

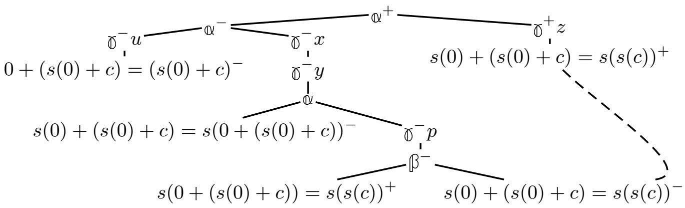
Remark 3 (Cut Elimination). Despite the admissibility of cut for the extensional expansion proofs from [10] on which we build upon, we present a set of rules which includes cut. The cut rule is only required in order to simulate the extensionality rule from [10]. The reason why we prefer the rules with cut is that the extensionality rule from [10] is less intuitive than our rules which are closer to the way extensionality is typically defined. We not only keep our rules for readability, but also because we use the extensional expansion trees presented here in the context of interactive theorem (see [3]), where intuitiveness of the rules matters much more than in automatic theorem proving. However, it seems to be straightforward to adapt the techniques presented in this paper to the extensional expansion trees from [10].

[4] For a complete formalization we refer to [3].

We denote the transformation rules shown in Fig. 3 by $\mathcal{EEP}$. Each application of a rule of name R is an *extensional expansion proof step* $\pi \vdash_R \pi'$, and $\vdash^*$ denotes its reflexive transitive closure. An extensional expansion proof with respect to $\mathcal{EEP}$ for some proof state π is a sequence of proof steps starting from π using rules from $\mathcal{EEP}$ and terminated with the *Axiom*-rule, i.e. $\pi \vdash^* \ldots \vdash_{Axiom}$. From [10] we can obtain the following soundness and completeness result:

Theorem 1 (Soundness & Completeness of $\mathcal{EEP}$). *F is a valid formula if, and only if, there exists a multiplicity μ for which there exists an extensional expansion proof with respect to $\mathcal{EEP}$ for $\triangle^{F^+,\mu}; id \triangleright \emptyset$.*

Proof (Sketch). We proved that each rule preserves the existence of satisfiable paths, which proves the soundness of $\mathcal{EEP}$. Completeness of $\mathcal{EEP}$ follows from [10]: in contrast to [10] "$=$" and "$\Leftrightarrow$" are part of our basic set of logical symbols, while [10] replaces them by Leibniz' definition of equality resp. by a conjunction of implications. Therefore we need the rules *Leibniz*$_=$, *Leibniz*$_\Leftrightarrow$, and $\Leftrightarrow$ *-Elim*. Our axiom rule is like in [10] and our connection rule is a macro for introducing many connections at once between leaf nodes. The functional and Boolean extensionality rules are different from those used in [10], but the later are admissible using our extensionality rules in combination with cut. Hence $\mathcal{EEP}$ is complete. □



Example 2. We apply the $\mathcal{EEP}$-calculus on the initial extensional expansion tree from Fig. 2.

Using the rules (1) *Subst* $\sigma := \{0/X, s(0) + c/U, 0/Z, s(0) + c/Y\}$ (2) *Leibniz*$_=$ on $\sigma(s(X)+Y = s(X+Y))$, (3) instantiating the set variable—introduced in 2—by $\lambda n \,.\, n = s(s(c))$ and (4) drawing a connection—indicated by the dashed line—between $s(0) + (s(0) + c) = s(s(c))^-$ and $s(0) + (s(0) + c) = s(s(c))^-$ results in the EET shown above. The proof is blocked in this situation since the initial multiplicities were insufficient.

4 Increase of Multiplicities

Proof search in extensional expansion proofs ([10]) proceeds by fixing the multiplicity of nodes of primary type γ, and subsequently searching for an admissible substitution and a spanning set of connections. To overcome the problem of guessing the right multiplicity beforehand there is a need to dynamically adjust multiplicities on the fly. An example is the instantiation of some γ-variable X bound on some γ_0-type subtree $\triangle_X$ of parent node $\triangle_\gamma$ of label $\gamma^p x \,.\, \varphi(x)$: in order to design a complete proof procedure, we must be able to "copy" that γ-variable before instantiating it. Furthermore, adding a new initial indexed formula tree $\triangle'_{X'}$ for $\varphi(X')$ for some new γ-variable X' and attaching it to $\triangle_\gamma$, prevents to carry over all proof information onto $\triangle'_{X'}$: instantiations, elimination of positive equivalences, Leibniz' equality introductions, extensionality introductions, and cuts are lost, as well as all connections that involved a subtree of $\triangle_X$ are not present for sub-nodes in $\triangle'_{X'}$. Finally, substitution information is not carried over onto $\triangle'_{X'}$. For instance, consider some γ-variable Y different from X which has been instantiated with a term containing X: there is no copy of Y which could be instantiated with the equivalent term containing X'. Thus, the proof information already established for $\triangle_X$ would have to be redone for $\triangle'_{X'}$.

In the following we present a mechanism to dynamically increase the multiplicity of some subtree which carries over all proof information to the new subtrees. First, note that the increase of multiplicity of some node may entail the increase of multiplicity of some other node. This is the case for example if a γ-variable to be copied occurs in the instance of some other γ-variable. Intuitively we need a notion of a *self contained set of subtrees with respect to a substitution*, in the sense that all γ- and δ-variables x which occur in the instance of some γ-variable Y, the set contains both the subtrees binding x and Y.

We formalize this intuition by the notion of a *convex set of subtrees with respect to some admissible substitution* σ.

Definition 13 (Convex Set of Subtrees). *Let $\triangle$ be an extensional expansion tree, σ an admissible substitution, and $\mathcal{K}$ a set of independent*[5] *subtrees of $\triangle$. $\mathcal{K}$ is* convex with respect to σ *if, and only if, for all $\triangle' \in \mathcal{K}$ and for all γ- and δ-variables x bound in $\triangle'$ we have: if x occurs in some instance $\sigma(Y)$ for some Y, then there exists some $\triangle'' \in \mathcal{K}$ in which Y is bound.*

A trivial example for such a set is the set that consists of the whole extensional expansion tree $\triangle$.

A convex set $\mathcal{K}$ of subtrees from some extensional expansion tree $\triangle$ has the property that it is not smaller with respect to $\lhd^{\sigma}_{\triangle}$ than any other part of $\triangle$ which is not in $\mathcal{K}$. In other words, there is no γ-variable bound outside $\mathcal{K}$ which is instantiated with some variable bound in $\mathcal{K}$. Consider the restriction $\sigma_{\mathcal{K}}$ of σ to those γ-variables that are bound in $\mathcal{K}$. Copying the subtrees from $\mathcal{K}$ yields a new set of subtrees $\mathcal{K}'$ and a renaming ρ of all γ- and δ-variables bound in $\mathcal{K}$. Then the renamed substitution $\sigma_{\mathcal{K}'} := \{\rho(\sigma_{\mathcal{K}}(x))/\rho(x) \mid x \in dom(\sigma_{\mathcal{K}})\}$ does not introduce any additional dependencies to parts not in $\mathcal{K}'$. Furthermore, if the original substitution was admissible, then so is $\sigma_{\mathcal{K}'} \circ \sigma$. In the following lemma we formalize this observation:

Lemma 2 (Maximality of Convex Sets of Subtrees). *Let $\triangle$ be an extensional expansion tree, σ an admissible substitution, and $\mathcal{K}$ a convex set of subtrees of $\triangle$. For all $x \in dom(\sigma)$, if x is not bound in $\mathcal{K}$, then all variables that occur in $\sigma(x)$ are also not bound in $\mathcal{K}$.*

Proof. Assume some x not bound in $\mathcal{K}$ and y bound in $\mathcal{K}$ that occurs in $\sigma(x)$. Then, by Definition 13, x should be bound in $\mathcal{K}$, which is a contradiction. $\square$

[5] i.e. no nested subtrees.

Given a convex set of subtrees from Γ_0, we increase their parents multiplicities by first copying these subtrees and thereby renaming the variables if necessary. Note that we cannot just create extensional expansion trees, since we want to preserve the applications of rules, for instance Leibniz' equality introductions and extensionality introductions, during the copying process. From this copying process we obtain a renaming ρ of the copied γ- and δ-variables and an isomorphic function ι between the original subtrees and their copies. We agree that ρ is a total function, which is the identity function for all variables not bound in the copied subtrees, and ι is a total function which is the identity function on all nodes, except those occurring in the copied subtrees.

The new subtrees are also of secondary type γ_0 and are inserted as further children on the respective parent node of primary type γ, which increases their multiplicities. The renaming ρ and the node mapping ι are used in order to carry over the substitution information by enlarging the substitution σ by defining $\sigma' := \{\rho(\sigma(x))/\rho(x) \mid x \in dom(\rho)\}$ and use $\sigma' \circ \sigma$ as the new overall substitution.

Finally, the information about established connections in $\mathcal{C}$ is carried over by enlarging $\mathcal{C}$ using ι, i.e. we apply ι to the following connections $\mathcal{C}$ as follows: $\iota(\mathcal{C}) := \{(\iota(c), \iota(c')) \mid (c, c') \in \mathcal{C}, c \text{ or } c' \in dom(\iota)\}$.

Definition 14 (Multiplicity Increase). *Let $\triangle; \sigma \triangleright \mathcal{C}$ be an extensional expansion proof state and $\triangle_{X_1}, \ldots, \triangle_{X_n} \in \Gamma_0$ such that $\{\triangle_{X_1}, \ldots, \triangle_{X_n}\}$ is a convex set with respect to σ. Then the increase of multiplicities of their parent subtrees $\mathfrak{T}^p x_i(\ldots, \triangle_{X_i}, \ldots), 1 \leq i \leq n$ is defined by the rule*

$$\frac{\triangle_{\mathfrak{T}^p x_1(\ldots,\triangle_{X_1},\triangle'_{X'_1},\ldots),\ldots,\mathfrak{T}^p x_n(\ldots,\triangle_{X_n},\triangle'_{X'_n},\ldots)}; \sigma' \circ \sigma \triangleright \mathcal{C} \cup \iota(\mathcal{C})}{\triangle_{\mathfrak{T}^p x_1(\ldots,\triangle_{X_1},\ldots),\ldots,\mathfrak{T}^p x_n(\ldots,\triangle_{X_n},\ldots)}; \sigma \triangleright \mathcal{C}} \; \mu\text{-}Inc$$

where (1) $\triangle'_{X'_i}$ are the copies of $\triangle_{X_i}$ and attached as additional subtrees to the respective γ-type parents, (2) ρ and ι are respectively the re-namings and the mapping of subtrees obtained from the copying, and (3) $\sigma' := [\rho(\sigma(X))/\rho(X) \mid$ for all γ-variables $X \in dom(\rho)]$.

We now introduce a constructive mechanism to determine the minimal set of nodes whose multiplicities need to be increased when increasing the multiplicity of some given node. Intuitively, if we have to copy a subtree $\triangle_m$ which contains the binding subtree of some γ-variable that occurs in the instance of some further γ-variable bound on some $\triangle'$, then we must copy $\triangle'$ as well.

Definition 15 (Determining Nodes to Increase Multiplicities). *Let $\triangle$ be an extensional expansion tree, σ an admissible substitution, and $\triangle_m$ a subtree of $\triangle$ of secondary type γ_0. The subtrees to copy in order to increase the multiplicity of $\triangle_m$'s parent are given by the $\mu(\triangle_m)$ that is inductively defined:*

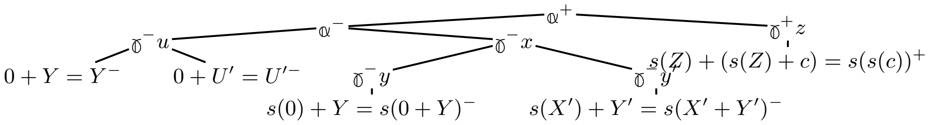
$$\mu(\triangle_m) = \{\triangle_m\} \cup \left(\bigcup_{\triangle' \mid \triangle_m \prec_\triangle \triangle'} \mu(\triangle')\right) \cup \left(\bigcup_{\triangle' \in Inst_\triangle(\triangle_m)} \mu(\triangle')\right)$$

where $Inst_\triangle(\triangle_m)$ is the set of binding nodes of γ-variables X, such that y occurs in $\sigma(X)$ and y is bound on $\triangle_m$. If $\triangle_m$ is not a binding node, then $Inst_\triangle(\triangle_m) = \emptyset$. We denote by $\mu(\triangle_m)_{min}$ the minimal subtrees of $\mu(\triangle_m)$ with respect to $\lhd^\sigma_\triangle$.

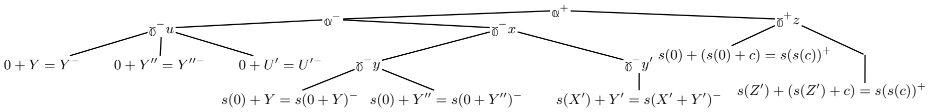
Lemma 3. *Let $\triangle$ be an extensional expansion tree of secondary type γ_0 and σ an admissible substitution. Then $\mu(\triangle)_{min}$ (1) is a convex set of subtrees with respect to σ and (2) it contains only subtrees of secondary type γ_0.*

Proof (Sketch). For (1) we prove that $\mu(\triangle)$ is always a convex set of *dependent* subtrees by deriving a contradiction from assuming there is some $\triangle_x \in \mu(\triangle)$ binding some γ- or δ-variable x, x occurs in $\sigma(y)$, and y is bound on $\triangle_y \notin \mu(\triangle)$. Then $\mu(\triangle_m)_{min}$ is a convex set of subtrees is an easy consequence. (2) The elements of $\mu(\triangle_m)$ are all subtrees of $\triangle_m$ and all subtrees of trees binding some γ-variable Y. Since $\triangle_m$ has type γ_0, all minimal subtrees are of type γ_0. □

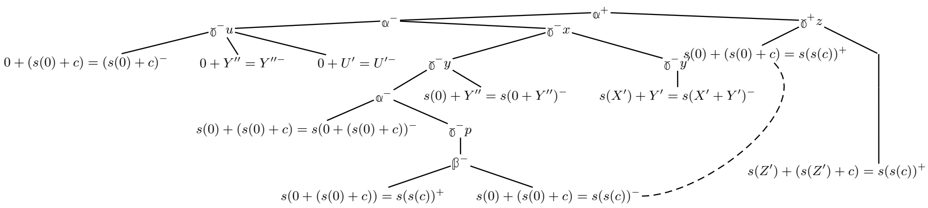
Example 3. We illustrate the dynamic increase of multiplicities using our running example. We redo the same proof as before, but always increase the multiplicities of the quantifiers of the substituted variables. We use the function from Definition 15 to determine the set of convex subtrees in each case. Using the rules (1) Increase of multiplicities for $\mathfrak{g}^- u$ and $\mathfrak{g}^- x$ and (2) *Subst* $\sigma := \{0/X, Y/U\}$ results in the EET



In order to apply the substitution $\sigma' := \{s(0) + c/Y, 0/Z\}$ in a safe way, we have to increase the multiplicities of the parents of Y and Z. However, $s(0) + Y = s(0 + Y)^-$ is the binding node of Y and $\mu(s(0) + Y = s(0 + Y)^-)_{min} := \{s(0) + Y = s(0 + Y)^-, 0 + Y = Y^-\}$. This indicates that if we want to increase the multiplicity of $\mathfrak{g}^- y$ we also have to increase the multiplicity of $\mathfrak{g}^- u$, since U has been substituted by Y already. Increasing the multiplicities of $\mathfrak{g}^- y, \mathfrak{g}^- u$ and $\mathfrak{g}^- z$ and applying the renaming $\rho := \{U''/U, Y''/Y, Z'/Z\}$ and only then applying σ' results in the extensional expansion tree



The overall substitution now is $\{0/X, s(0) + c/U, Y''/U'', s(0) + c/Y, 0/Z\}$. Applying, as before, *Leibniz*$_=$, then increase the multiplicity of $\mathfrak{g}^- p$ before substituting P with $\lambda n \,.\, n = s(s(c))$ and drawing the same connection we obtain



Now the proof is not blocked and can be continued. However, due to lack of space we do not present the extensional expansion tree of the final proof state.

Lemma 4 (Soundness & Safeness of Multiplicity Increase). *The increase of multiplicities is sound and safe.*

Proof. We first prove that the increase of multiplicities preserves the admissibility of the substitution and then that the resulting EET contains satisfiable paths, if, and only if, the former EET does.

(1) By Lemma 2 the convex set $\{\triangle_{x_1}, \ldots, \triangle_{x_n}\}$ is maximal with respect to $\lhd_\sigma^\triangle$. Thus there are disjoint substitutions σ_1, σ_2 such that the overall substitution $\sigma := \sigma_1 \uplus \sigma_2$ and $dom(\sigma_2)$ contains exactly all γ-variables from $dom(\sigma)$ which are bound in some $\triangle_{X_i}$. If σ is irreflexive, then $\lhd^{\sigma_2}_{\triangle_{X_i}, 1 \leq n}$ is irreflexive and for any $x \in dom(\sigma_2)$, $\sigma_2(x)$ contains only γ- and δ-variables bound in some $\triangle_{X_i}$. Furthermore, $\sigma' := \{\rho(\sigma_2(x))/\rho(x) \mid x \in dom(\sigma_2)\}$ and $\lhd^{\sigma'}_{\triangle_{X'_i}, 1 \leq n}$ is irreflexive since $\triangle_{X_i}$ and $\triangle'_{X'_i}$ are isomorphic. Since σ_2 was disjoint from σ_1 and ρ is a renaming of all γ- and δ-variables in σ_2, so is σ' from both σ_1 and σ_2. Hence $\lhd^{\sigma' \circ \sigma}_\triangle$ is irreflexive wrt. the new EET.
(2) For each $\triangle_{X_i}$ the increase of multiplicities adds its renamed copy $\triangle'_{X'_i}$ to each path that contained $\triangle_{X_i}$. Thus, if $\mathsf{DF}(\triangle_{X_i}) = F^p$, then $\mathsf{DF}(\triangle'_{X'_i}) = \rho(F^p)$. Since ρ is simply a renaming of variables, it trivially holds that F^p is satisfiable if, and only if, $\rho(F^p)$ is satisfiable. □

Theorem 2 (Soundness & Completeness of $\mathcal{EEP} + \mu$-*Inc*). *F is a valid formula if, and only if, there exists an extensional expansion proof using $\mathcal{EEP}$ augmented by μ-Inc for $\triangle^{F^+, \mu_1}; id \rhd \emptyset$, where μ_1 denotes the singular multiplicity.*

Proof (Sketch). The proof is based on the fact that the only rule which is unsafe is the substitution rule. Combining the instantiation of some γ-variable X by some term t with the increase of the multiplicities of the subtrees determined by $\mu(\triangle_X)_{min}$ before application of the substitution we can make it a safe transformation. Let $\triangle$ be some $\triangle \in \mu(\triangle_X)_{min}$; any path $\ll \Gamma, \triangle \gg$ has the deep signed formulas $\{\varphi(X)\} \cup \mathsf{DF}(\Gamma)$. After increase of the multiplicities and application of the substitution, this path becomes $\ll \Gamma', \Gamma, \triangle, \triangle' \gg$, where $\triangle'$ is the "copy" of $\triangle$ and Γ' results from increasing the multiplicities of the other elements of $\triangle \in \mu(\triangle_X)_{min}$. The deep signed formulas now are $\{\varphi(t), \varphi(X')\} \cup \mathsf{DF}(\Gamma) \cup \mathsf{DF}(\Gamma')$, where X' is the new γ-variable for X. If this set of formulas is satisfiable, then so is $\{\varphi(X)\} \cup \mathsf{DF}(\Gamma)$. Hence the transformation is safe. □

5 Related Work & Conclusion

Designing a complete matrix proof search procedure requires to search for the right multiplicities. We presented a mechanism to increase the multiplicities which includes the search for the multiplicities into the kernel matrix proof search procedure. It supports a demand-driven increase of the multiplicities which not only preserves existing substitutions and connections, but moreover inherits this information to the new subtrees that result from the increase of multiplicities.

As an easy consequence the transformation rules that result from the addition of the multiplicity increasing rule are sound and complete. Although the technique was developed also for some first-order classical modal logics with constant domains, this paper described the technique only for higher-order logic matrix calculi. The definition of the technique for the first-order modal logics is based on indexed formula trees from [12] and essentially consists of extending the notion of convex set of subtrees to take into account the modal quantifiers and treat them analogous to the variable quantifiers. Details of it can be found in [3].

Issar [8] presented a respective procedure for the JForms introduced in [2], but restricts it to the first-order fragment. It increases the multiplicities on demand for some problematic path for which no connection can be found. The increase of multiplicities is realized by copying and renaming a part of the JForm, but restricting the effect to the problematic path in order to control the increase of the number of new paths. However, any information about substitutions and connections that existed for the copied parts are not adapted during copying. The procedure is complete as substitutions are not applied to the JForm, which is probably a major reason why the technique is restricted to first-order logic.

References

1. Peter B. Andrews. General models, descriptions, and choice in type theory. *The Journal of Symbolic Logic*, 37(2):385–397, June 1972.
2. Peter B. Andrews. Theorem proving via general matings. *Journal of the Association for Computing Machinery*, 28(2):193–214, April 1981.
3. Serge Autexier. *Hierarchical Contextual Reasoning.* PhD thesis, Computer Science Department, Saarland University, Saarbrücken, Germany, 2003.
4. Henk P. Barendregt. *The Lambda Calculus – Its Syntax and Semantics.* North Holland, 1984.
5. Wolfgang Bibel. On matrices with connections. *Journal of the Association for Computing Machinery*, 28(4):633–645, October 1981.
6. Melvin Fitting. Tableau methods of proof for modal logics. *Notre Dame Journal of Formal Logic*, XIII:237–247, 1972.
7. Leon Henkin. Completeness in the theory of types. *The Journal of Symbolic Logic*, 15:81–91, 1950.
8. Sunil Issar. Path-focused duplication: A search procedure for general matings. In T. S. W. Dietterich, editor, *Proceedings of the 8th National Conference on Artificial Intelligence (AAAI 90)*, volume 1, pages 221–226, Menlo Park - Cambridge - London, July 1990. AAAI Press / MIT Press.
9. Dale A. Miller. *Proofs in Higher-Order Logic.* Phd thesis, Carnegie Mellon University, 1983.
10. Frank Pfenning. *Proof Transformation in Higher-Order Logic.* Phd thesis, Carnegie Mellon University, 1987.
11. R. M. Smullyan. *First-Order Logic*, volume 43 of *Ergebnisse der Mathematik.* Springer-Verlag, Berlin, 1968.
12. Lincoln Wallen. *Automated proof search in non-classical logics: efficient matrix proof methods for modal and intuitionistic logics.* MIT Press series in artificial intelligence, 1990.