Proof Generalization in LK by Second Order Unifier Minimization

Thierry Boy de La Tour, Nicolas Peltier

▶ **To cite this version:**

Thierry Boy de La Tour, Nicolas Peltier. Proof Generalization in LK by Second Order Unifier Minimization. Journal of Automated Reasoning, 2016, 57 (3), pp.245-280. hal-01364000

HAL Id: hal-01364000
https://hal.science/hal-01364000

Submitted on 30 Aug 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Proof Generalization in LK by Second Order Unifier Minimization

Thierry Boy de la Tour Nicolas Peltier

Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG
38000 Grenoble, France
thierry.boy-de-la-tour@imag.fr nicolas.peltier@imag.fr

Abstract

We devise a method for generalizing proofs in Gentzen's sequent calculus LK, presented in a typed λ-calculus flavor. A constrained version LK^{c} of the calculus is introduced, aiming at collecting a second order constraint ensuring that all the inference steps occurring in a proof are syntactically correct. A semantics is provided for LK^{c}, extending the standard semantics of LK. It is then established that LK-proofs correspond to LK^{c}-proofs with valid constraint thanks to the use of eigenterms replacing LK's eigenvariables. Next, a lifting theorem shows how a valid LK^{c}-proof can be lifted to a most general proof, yielding a non-trivial constraint together with a solution. An algorithm is then provided that minimizes this solution of the constraint. The result, applied to the most general proof, yields a valid proof that translates to an LK-proof more general than the initial one. Finally, clues are given for extending this method to other logics with due care on proof lifting.

Keywords Sequent calculus Generalization Second order unification Constraints

Mathematical Subject Classification F.4.1

1 Introduction

Generalization, as the process of inducing general properties or patterns from particular cases, is a most essential aspect of human reasoning which is usually left aside by automated reasoning tools. In this paper, we are interested in generalizing proofs: given a proof $\boldsymbol{P}$ of some assertion (e.g., sequent) ϕ, we aim at computing a new proof $\boldsymbol{Q}$ of a new assertion ψ that is as general as possible and at least more general than ϕ in the sense that there exists a (second order) substitution θ such that $\psi\theta$ is identical to ϕ (up to usual properties such as $\alpha\beta\eta$-equivalence). Applications are numerous and obvious, i.e., proof reuse, proof structuring and compression (to detect repeating patterns in a proof and replace them by appropriate lemmata), program generalization (when applied on assertions stating properties of programs), proof by analogy, etc. This might also be useful for pruning the search space of theorem proving procedures (by generalizing previously generated subproofs to close other branches).

Our method can be informally summarized as follows. First, a constrained sequent calculus (called LK^{c}) is introduced. In this calculus, the conditions that allow for the application of the inference rules are not checked "statically" during the construction of the proof, but instead are "asserted" and collected as a conjunction of higher-order constraints attached to the proof. Any solution of the constraints gives rise to an LK-proof (under

some conditions). We provide a semantics for LK^{c}, extending the standard semantics of LK, then we show that LK^{c} is logically sound, and that every LK-proof can be transformed into an LK^{c}-proof with exactly the same size and structure. Next, we prove that for any LK^{c}-proof $\boldsymbol{P}$, there exists an "abstract" proof $\boldsymbol{P}^{\mathrm{a}}$ that is more general than $\boldsymbol{P}$, and also more general than any generalization of $\boldsymbol{P}$; $\boldsymbol{P}^{\mathrm{a}}$ can thus be viewed as the (unique) minimal generalization of $\boldsymbol{P}$ in LK^{c}. The proof $\boldsymbol{P}^{\mathrm{a}}$ is not an LK-proof because its constraints are not valid (but they are satisfiable). Finally, we show how to reconstruct a generalized LK-proof from $\boldsymbol{P}^{\mathrm{a}}$. In principle, this can be done by computing most general solutions of the constraints of $\boldsymbol{P}^{\mathrm{a}}$, however, since higher-order unification is undecidable, this is not always feasible. We therefore devise a minimization algorithm computing a solution that is always more general than the substitution corresponding to the initial LK-proof (the solution is however not necessarily a most general one). The principle of this algorithm is to use the latter substitution as a "witness" in order to guide the application of higher-order unification rules.

1.1 Related work

The idea of using constrained sequent or tableaux calculus is not new: it has been considered for instance in [9] to avoid backtracking in free-variable tableaux, and also in [24] to define proof procedures for reasoning with first order formulæ in Presburger arithmetic. The idea is similar – namely to collect constraints enabling the application of logical inference rules – but we apply it in a more general setting: in [9] or [24] the constraints only state properties of *terms* (elements of the domain of a first order interpretation or arithmetic terms respectively), whereas we use them to express properties of *formulæ*. Another essential difference is that [24] uses arithmetic quantifiers to take care of the eigenvariable condition (by introducing a universal quantifier for every eigenvariable), whereas we prefer to encode this condition by using a specific constructor ι (as explained in Section 3). The reason for this choice is that the handling of higher-order constraints with arbitrary quantifier alternation is difficult from a computational point of view (see [14] for undecidability results). The drawback is that LK^{c}-proofs cannot always be expressed in LK as we shall see in Section 5. The idea of separating the structure of the proof tree from the unification conditions has been used in other contexts, for instance in [11], for analyzing sequent proofs and reducing the complexity of the derivations, in [6] for reasoning modulo theories, or in [3, 22] for building models of clause sets.

The notion of abstract proof is also related to a result by Parikh (Lemma A in [21]) concerning so-called schematic systems and devoted to the study of Kreisel's Conjecture (or KC, a conjecture on proofs in Peano Arithmetic, see [4] for a survey). This result allows the representation of a family of proofs by a "proof analysis" (or "proof skeleton" in the terminology of [7]) together with a unification problem and a set of restrictions. Besides the fact that we can dispense with these restrictions thanks to our encoding of eigenvariables through ι, our abstract proofs carry slightly more information than proof skeletons, for the simple reason that we do not treat LK as a schematic system, thus following [13] or [2] (see Section 6 for more details). We show in Section 9 how LK could be treated as a schematic system in our setting. Other important differences with [21] are our use of λ-calculus which allows a more standard treatment of bound variables, and that we make no difference between formulæ and "concrete formulæ" since we may introduce second-order variables ("meta-notations" in [21]) in our generalized proofs. This departs from the works on KC which is very sensitive on the language used to represent PA (see [4]).

In [10], an algorithm is proposed to generalize statements in a typed λ-calculus aimed to represent inductive proofs. This language extends usual logical frameworks by arith-

metic variables and primitive recursive definitions. The approach in [10] is targeted at constructing, from a sequence of proofs of statements $P(0)$, $P(1)$, ..., an inductive proof of the statement $\forall x\, P(x)$. This is done by computing common patterns in the proofs of $P(i)$ (ignoring the difference between arithmetic terms) using a form of higher-order anti-unification [23] combined with an ad-hoc algorithm for generating suitable recursion steps from instances. This approach shares some similarities with ours, namely the use of λ-terms to represent proofs, but it is very different both in its scope and purposes. Our method focuses on the generalization of LK-proofs and only uses higher-order terms to express and compute such generalizations. Hagiya's approach focuses on the construction of iterated sequences of proofs from particular examples. In our context, induction could be performed by trying to generalize a single proof $P(i) \vdash P(i+1)$ (for a fixed $i \in \mathbb{N}$), into a proof of $P'(x) \vdash P'(x+1)$ (where x is a variable, and where P' is more general than $P(x)$). If such a generalization is feasible, then it is clear that an inductive proof of $\forall x P(x)$ can be constructed (if $P'(0)$ is provable).

Generalization is often handled in proof assistants by replaying tactics (see, e.g., [8, 12]): a tactic (sequence of proof steps) or proof term (describing the structure of the proof) corresponding to some assertion can be replayed on another one, and in case of success may yield a new proof with the same structure as the initial one. The advantage of our technique is that it computes an explicit description of the generalized theorems, using the same language as the original one.

Generalization has also been considered in the context of analogical reasoning (see e.g., [26, 15]). It is often performed by renaming different occurrences of the same symbol, in case these occurrences turn out to be unrelated in the context, i.e., if all logical inferences can be performed without assuming that these occurrences are equal. Our method has a wider scope, in the sense that it allows for richer transformations and produces more general proofs and formulæ, as evidenced by the following example.

Example 1. We consider the following LK-proof[1]

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{P(x_0,a)\vdash P(x_0,a)}\,(\text{Axiom})}{\forall x\, P(x,a)\vdash P(x_0,a)}\,(\forall\text{-L})}{\forall x\, P(x,a)\vdash P(x_0,a)\vee Q(x_0,a)}\,(\vee\text{-R1})}{\forall x\, P(x,a)\vdash \exists y\,(P(x_0,y)\vee Q(x_0,y))}\,(\exists\text{-R}) \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{Q(x_0,a)\vdash Q(x_0,a)}\,(\text{Axiom})}{\forall y\, Q(y,a)\vdash Q(x_0,a)}\,(\forall\text{-L})}{\forall y\, Q(y,a)\vdash P(x_0,a)\vee Q(x_0,a)}\,(\vee\text{-R1})}{\forall y\, Q(y,a)\vdash \exists y(P(x_0,y)\vee Q(x_0,y))}\,(\exists\text{-R})}{\forall x\, P(x,a)\vee\forall y\, Q(y,a)\vdash \exists y\,(P(x_0,y)\vee Q(x_0,y))}\,(\vee\text{-L})$$
$$\dfrac{\forall x\, P(x,a)\vee\forall y\, Q(y,a)\vdash \exists y\,(P(x_0,y)\vee Q(x_0,y))}{\forall x\, P(x,a)\vee\forall y\, Q(y,a)\vdash \forall x\exists y\,(P(x,y)\vee Q(x,y))}\,(\forall\text{-R})$$

It is intuitively clear that the fact that the two occurrences of a in the end-sequent are equal plays no role in the proof, hence the end-sequent can be safely replaced by, e.g., $\forall x\, P(x,a)\vee\forall y\, Q(y,b) \vdash \forall x\exists y\,(P(x,y)\vee Q(x,y))$ (replacing all occurrences of a in the right branch of the above proof by b). Using our approach, we can detect that a and b can actually be replaced by something that depends on the variables x and y respectively, and that this proof can be generalized to a proof of the sequent $\forall x\, p(x,f(x))\vee\forall y\, q(y,g(y))\vdash \forall x\exists y\,(p(x,y)\vee q(x,y))$, where p, q, f and g are second order variables. It is clear that the obtained sequent is strictly more general than the previous ones (in the sense that it matches more expressions).

[1]Formal definitions are given in Section 2, see in particular Figure 1 for the inference rules.

2 LK on Typed λ-Terms

We first give a definition of LK where terms and formulæ are members of the same language and are only distinguished by their types. Furthermore, bound variables are treated in a uniform way with the binder λ. We thus use a fragment of typed λ-calculus to represent terms and formulæ; hence term constructors are the standard abstraction and application constructors, but atoms are separated into constants (for logical symbols and the elements of a first order signature) and variables. We also use an extra term constructor ι for eigenterms (as explained in Section 1) though we will only use it in Section 3 and forth.

Definition 2. Let $\mathbf{\imath}$ and $\mathbf{o}$ be symbols denoting respectively the type of individuals (or terms) and the type of booleans (or formulæ). A *type* is inductively defined as either a *basic* type $\mathbf{\imath}$ or $\mathbf{o}$, or $\tau \to \tau'$ where τ and τ' are types. The *order* of types is inductively defined as $\mathrm{order}(\mathbf{o}) \stackrel{\text{def}}{=} \mathrm{order}(\mathbf{\imath}) \stackrel{\text{def}}{=} 1$ and $\mathrm{order}(\tau \to \tau') \stackrel{\text{def}}{=} \max(1 + \mathrm{order}(\tau), \mathrm{order}(\tau'))$.

For $n \in \mathbb{N}$ and $\tau', \tau_1, \ldots, \tau_n$ types, we write $\tau_1, \ldots, \tau_n \to \tau'$ for the type τ' if $n = 0$ and for $\tau_1 \to (\cdots (\tau_n \to \tau') \cdots)$ if $n > 0$. If $\tau_1 = \cdots = \tau_n = \tau$ then this type is written $\tau^n \to \tau'$.

Let $\mathcal{V}$ be a set of *variables* containing infinitely many variables of each type of the form $\mathbf{\imath}^n \to \mathbf{\imath}$ and $\mathbf{\imath}^n \to \mathbf{o}$ for $n \geq 0$ (which we call $\mathcal{V}$*-types*), and no other. Let $\mathcal{C}$ be a set of *constants* containing the *logical symbols* $\exists, \forall : (\mathbf{\imath} \to \mathbf{o}) \to \mathbf{o}$, $\neg : \mathbf{o} \to \mathbf{o}$, $\vee, \wedge, \Rightarrow : \mathbf{o}^2 \to \mathbf{o}$, $\simeq : \mathbf{\imath}^2 \to \mathbf{o}$ and a finite number of symbols of $\mathcal{V}$-types (this part corresponds to a first order signature).

The set $\mathcal{T}$ of *terms* together with their types and sets of free variables (denoted by $\mathrm{FV}(t)$) is inductively defined by:

- $\mathcal{V} \subseteq \mathcal{T}$, $\mathcal{C} \subseteq \mathcal{T}$, and as terms they have the same type as variables or constants; a variable is free in itself and a constant has no free variables;
- for all $x \in \mathcal{V}$ of type $\mathbf{\imath}$ and $t \in \mathcal{T}$ of type τ, $(\lambda x\, t) \in \mathcal{T}$ and has type $\mathbf{\imath} \to \tau$; $\mathrm{FV}((\lambda x\, t)) \stackrel{\text{def}}{=} \mathrm{FV}(t) \setminus \{x\}$;
- for all $s \in \mathcal{T}$ of type $\tau \to \tau'$ and $t \in \mathcal{T}$ of type τ, $(s\, t) \in \mathcal{T}$ and has type τ'; $\mathrm{FV}((s\, t)) \stackrel{\text{def}}{=} \mathrm{FV}(s) \cup \mathrm{FV}(t)$;
- for all $n \in \mathbb{N}$, $n > 0$, terms $\phi_1, \ldots, \phi_n \in \mathcal{T}$ of type $\mathbf{o}$, $(\iota\, \phi_1 \cdots \phi_n) \in \mathcal{T}$ is an *eigenterm* of type $\mathbf{\imath}$; $\mathrm{FV}((\iota\, \phi_1 \cdots \phi_n)) \stackrel{\text{def}}{=} \bigcup_{i=1}^{n} \mathrm{FV}(\phi_i)$.

An *atom* is an element of $\mathcal{V} \cup \mathcal{C}$. For any term $t \in \mathcal{T}$ we write $t : \tau$ if t has type τ. The *order* of a term of type τ is the order of τ. For any sequence $t_1, \ldots, t_n$ of terms in $\mathcal{T}$, let $\mathrm{FV}(t_1, \ldots, t_n) = \bigcup_{i=1}^{n} \mathrm{FV}(t_i)$.

The language of *standard* terms is the set $\mathcal{S}$ of terms in $\mathcal{T}$ without any occurrence of eigenterms. The language of *fair* terms is the set $\mathcal{F}$ of terms in $\mathcal{T}$ whose atoms are only variables and logical symbols.

For any language $\mathcal{L} \subseteq \mathcal{T}$ such that $\mathcal{V} \subseteq \mathcal{L}$, an $\mathcal{L}$*-substitution* (or *substitution* if $\mathcal{L} = \mathcal{T}$) is a type-preserving function σ from $\mathcal{V}$ to $\mathcal{L}$ whose *domain* $\mathrm{Dom}(\sigma) \stackrel{\text{def}}{=} \{x \in \mathcal{V} \mid \sigma(x) \neq x\}$ is finite. The identity on $\mathcal{V}$ is a substitution denoted by *id*. For $v \in \mathcal{V}$ and $t \in \mathcal{L}$ of the same type, $\sigma[v \mapsto t]$ is the $\mathcal{L}$-substitution identical to σ but on v where it yields t. The notion of free variables is extended to substitutions by $\mathrm{FV}(\sigma) \stackrel{\text{def}}{=} \mathrm{Dom}(\sigma) \cup \bigcup_{x \in \mathrm{Dom}(\sigma)} \mathrm{FV}(\sigma(x))$. A substitution σ of domain $\{x_1, \ldots, x_n\}$ and such that $\sigma(x_i) = t_i$ for $1 \leq i \leq n$ is written $\sigma = [t_1/x_1, \ldots, t_n/x_n]$.

Applying a substitution σ to a term $t \in \mathcal{T}$ yields a term $t\sigma \in \mathcal{T}$ inductively defined on t by

- $v\sigma \stackrel{\text{def}}{=} \sigma(v)$ for any $v \in \mathcal{V}$,
- $c\sigma \stackrel{\text{def}}{=} c$ for any $c \in \mathcal{C}$,
- $(s\ t)\sigma \stackrel{\text{def}}{=} (s\sigma\ t\sigma)$ for any $s, t \in \mathcal{L}$,
- $(\lambda x\, t)\sigma \stackrel{\text{def}}{=} (\lambda y\, t\sigma[x \mapsto y])$ for any $x \in \mathcal{V}$, $t \in \mathcal{L}$ and $y : \mathbf{1} \in \mathcal{V} \setminus \mathrm{FV}(t\sigma)$.
- $(\iota\, \phi_1 \cdots \phi_n)\sigma \stackrel{\text{def}}{=} (\iota\, \phi_1\sigma \cdots \phi_n\sigma)$ for any $n \in \mathbb{N}$, $n > 0$ and $\phi_1, \ldots, \phi_n \in \mathcal{T}$ of type $\mathbf{o}$.

A language $\mathcal{L}$ is *substitutive* if it is closed under $\mathcal{L}$-substitutions; the languages $\mathcal{T}$, $\mathcal{S}$ and $\mathcal{F}$ are substitutive. Any two $\mathcal{L}$-substitutions σ and μ can be composed by $\sigma\mu(x) \stackrel{\text{def}}{=} \sigma(x)\mu$ for all $x \in \mathcal{V}$, which is guaranteed to be an $\mathcal{L}$-substitution if $\mathcal{L}$ is substitutive.

We consider on $\mathcal{T}$ the equational theory generated by the following rules:

$$\begin{array}{rcl} (\lambda x\, l) & \to_\alpha & (\lambda y\, l[y/x]) \text{ if } y \notin \mathrm{FV}(l) \\ ((\lambda x\, l)\ t) & \to_\beta & l[t/x] \\ (\lambda x\, (l\ x)) & \to_\eta & l \text{ if } x \notin \mathrm{FV}(l). \end{array}$$

The restriction of this theory to $\mathcal{S}$ is obviously a fragment of typed λ-calculus which is well-known to be strongly normalizing, and since there is no rule for ι, it is obvious that $\beta\eta$-reduction is again strongly normalizing on $\mathcal{T}$, and every $t \in \mathcal{T}$ has a $\beta\eta$-normal form $t{\downarrow_{\beta\eta}} \in \mathcal{T}$. We write $\equiv_{\alpha\beta\eta}$ for $\alpha\beta\eta$-equivalence. Note that the languages $\mathcal{T}$, $\mathcal{S}$ and $\mathcal{F}$ are closed under $\alpha\beta\eta$-reduction.

As usual we often omit parentheses and associate to the left: $(l\ t_1 \ldots t_n)$ stands for $(\cdots (l\ t_1) \cdots\ t_n)$ if $n > 0$ and for l if $n = 0$. We write $t \Rightarrow s$, $\exists x\, t$,... for the applications $(\Rightarrow\ t\, s)$, $(\exists\ \lambda x\, t)$, ... respectively.

An *$\mathcal{L}$-sequent* (or *sequent* if $\mathcal{L} = \mathcal{T}$) is an expression of the form $\Gamma \vdash \Delta$ where Γ and Δ are sequences of terms in $\mathcal{L}$ of type $\mathbf{o}$. We write $\mathrm{FV}(\Gamma \vdash \Delta)$ for $\mathrm{FV}(\Gamma, \Delta)$. We apply substitutions to sequences or sets of terms in the standard way, and to sequents by $(\Gamma \vdash \Delta)\sigma \stackrel{\text{def}}{=} \Gamma\sigma \vdash \Delta\sigma$. If $\mathcal{L}$ is closed under $\alpha\beta\eta$-reduction the relation $\equiv_{\alpha\beta\eta}$ and $\beta\eta$-normalization are similarly extended in a standard way to sequences of $\mathcal{L}$-terms and to $\mathcal{L}$-sequents; we also extend $\equiv_{\alpha\beta\eta}$ to $\mathcal{L}$-substitutions σ, θ by $\sigma \equiv_{\alpha\beta\eta} \theta$ if $x\sigma \equiv_{\alpha\beta\eta} x\theta$ for all $x \in \mathcal{V}$, and we say that σ is in *$\beta\eta$-normal form* if for all $x \in \mathrm{Dom}(\sigma)$ the terms $x\sigma$ are reduced to $\beta\eta$-normal form. Note that $\sigma \equiv_{\alpha\beta\eta} \theta$ entails $t\sigma \equiv_{\alpha\beta\eta} t\theta$ for any term t, $\Gamma\sigma \equiv_{\alpha\beta\eta} \Gamma\theta$ for any sequence Γ of terms, etc.

The rules of LK are given in Figure 1. The meta-variables s, t denote terms of type $\mathbf{1}$, ϕ, ψ denote terms of type $\mathbf{o}$, ξ denotes a term of type $\mathbf{1} \to \mathbf{o}$, x denotes a variable of type $\mathbf{1}$ and Γ, Δ, Σ, Π denote sequences of terms of type $\mathbf{o}$. An inference between sequents holds by a rule (R) if these sequents are $\alpha\beta\eta$-equivalent to those obtained from the meta-sequents of rule (R) in Figure 1 by some instantiation of their meta-variables, and if the side condition holds with the same instantiation.

If $\mathcal{L}$ is closed under $\alpha\beta\eta$-reduction, an $\mathcal{L}$-LK-proof is an LK-proof built only with $\mathcal{L}$-sequents. A *standard* sequent (resp. LK-proof, substitution, etc.) is an $\mathcal{S}$-sequent (resp. $\mathcal{S}$-LK-proof, $\mathcal{S}$-substitution, etc.) and similarly a *fair* sequent, (resp. LK-proof, etc.) is an $\mathcal{F}$-sequent (resp. $\mathcal{F}$-LK-proof, etc.)

In order to handle proofs in a convenient way we adopt some simple notations. Proofs will be referred to by symbols derived from $\boldsymbol{P}$ and $\boldsymbol{Q}$. The notation $\boldsymbol{P} : s$ means that s is the end sequent of the proof $\boldsymbol{P}$ (hence that $\boldsymbol{P}$ is a proof *of* s). We will also decompose proofs into subproofs by writing

$$\boldsymbol{P} = \frac{\boldsymbol{P}_1 : s_1 \ \cdots\ \boldsymbol{P}_n : s_n}{s}(\mathrm{R}),$$

$$\frac{\Gamma \vdash \Delta}{\Gamma,\, \phi \vdash \Delta}\ (\text{W-L}) \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta,\, \phi}\ (\text{W-R})$$

$$\frac{\Gamma,\, \phi,\, \phi \vdash \Delta}{\Gamma,\, \phi \vdash \Delta}\ (\text{C-L}) \qquad \frac{\Gamma \vdash \Delta,\, \phi,\, \phi}{\Gamma \vdash \Delta,\, \phi}\ (\text{C-R})$$

$$\frac{\Gamma,\, \psi,\, \phi,\, \Sigma \vdash \Delta}{\Gamma,\, \phi,\, \psi,\, \Sigma \vdash \Delta}\ (\text{P-L}) \qquad \frac{\Gamma \vdash \Delta,\, \psi,\, \phi,\, \Pi}{\Gamma \vdash \Delta,\, \phi,\, \psi,\, \Pi}\ (\text{P-R})$$

$$\frac{}{\phi \vdash \phi}\ (\text{Axiom}) \qquad \frac{}{\vdash t \simeq t}\ (\text{Ref})$$

$$\frac{\Gamma,\, t \simeq s,\, (\xi\ t) \vdash \Delta}{\Gamma,\, t \simeq s,\, (\xi\ s) \vdash \Delta}\ (\text{Param-L}) \qquad \frac{\Gamma,\, t \simeq s \vdash \Delta,\, (\xi\ t)}{\Gamma,\, t \simeq s \vdash \Delta,\, (\xi\ s)}\ (\text{Param-R})$$

$$\frac{\Gamma,\, s \simeq t \vdash \Delta}{\Gamma,\, t \simeq s \vdash \Delta}\ (\text{Com-L}) \qquad \frac{\Gamma \vdash \Delta,\, \phi \quad \Sigma,\, \phi \vdash \Pi}{\Gamma,\, \Sigma \vdash \Delta,\, \Pi}\ (\text{Cut})$$

$$\frac{\Gamma,\, \phi \vdash \Delta}{\Gamma,\, \phi \wedge \psi \vdash \Delta}\ (\wedge\text{-L1}) \qquad \frac{\Gamma \vdash \Delta,\, \phi}{\Gamma \vdash \Delta,\, \phi \vee \psi}\ (\vee\text{-R1})$$

$$\frac{\Gamma,\, \psi \vdash \Delta}{\Gamma,\, \phi \wedge \psi \vdash \Delta}\ (\wedge\text{-L2}) \qquad \frac{\Gamma \vdash \Delta,\, \psi}{\Gamma \vdash \Delta,\, \phi \vee \psi}\ (\vee\text{-R2})$$

$$\frac{\Gamma,\, \phi \vdash \Delta \qquad \Sigma,\, \psi \vdash \Pi}{\Gamma,\, \Sigma,\, \phi \vee \psi \vdash \Delta,\, \Pi}\ (\vee\text{-L}) \qquad \frac{\Gamma \vdash \Delta,\, \phi \qquad \Sigma \vdash \Pi,\, \psi}{\Gamma,\, \Sigma \vdash \Delta,\, \Pi,\, \phi \wedge \psi}\ (\wedge\text{-R})$$

$$\frac{\Gamma \vdash \Delta,\, \phi \qquad \Sigma,\, \psi \vdash \Pi}{\Gamma,\, \Sigma,\, \phi \Rightarrow \psi \vdash \Delta,\, \Pi}\ (\Rightarrow\text{-L}) \qquad \frac{\Gamma,\, \phi \vdash \Delta,\, \psi}{\Gamma \vdash \Delta,\, \phi \Rightarrow \psi}\ (\Rightarrow\text{-R})$$

$$\frac{\Gamma \vdash \Delta,\, \phi}{\Gamma,\, \neg\phi \vdash \Delta}\ (\neg\text{-L}) \qquad \frac{\Gamma,\, \phi \vdash \Delta}{\Gamma \vdash \Delta,\, \neg\phi}\ (\neg\text{-R})$$

$$\frac{\Gamma,\, (\xi\ t) \vdash \Delta}{\Gamma,\, \forall\xi \vdash \Delta}\ (\forall\text{-L}) \qquad \frac{\Gamma \vdash \Delta,\, (\xi\ t)}{\Gamma \vdash \Delta,\, \exists\xi}\ (\exists\text{-R})$$

$$\frac{\Gamma,\, (\xi\ x) \vdash \Delta}{\Gamma,\, \exists\xi \vdash \Delta}\ (\exists\text{-L}) \qquad \frac{\Gamma \vdash \Delta,\, (\xi\ x)}{\Gamma \vdash \Delta,\, \forall\xi}\ (\forall\text{-R})$$

Where $x \notin \mathrm{FV}(\Gamma, \Delta, \xi)$ (*eigenvariable condition).*

Figure 1: The sequent calculus LK

meaning that $\boldsymbol{P}$ ends with the inference $\dfrac{s_1 \ \cdots \ s_n}{s}$ by rule (R). We write $\mathrm{FV}(\boldsymbol{P})$ for $\mathrm{FV}(s) \cup \bigcup_{i=1}^{n} \mathrm{FV}(\boldsymbol{P}_i)$.

Normalization of LK-proofs is defined inductively by

$$\boldsymbol{P}{\downarrow_{\beta\eta}} = \frac{\boldsymbol{P}_1{\downarrow_{\beta\eta}} \colon s_1{\downarrow_{\beta\eta}} \ \cdots \ \boldsymbol{P}_n{\downarrow_{\beta\eta}} \colon s_n{\downarrow_{\beta\eta}}}{s{\downarrow_{\beta\eta}}}(\mathrm{R}) \text{ if } \boldsymbol{P} = \frac{\boldsymbol{P}_1 : s_1 \ \cdots \ \boldsymbol{P}_n : s_n}{s}(\mathrm{R}).$$

Note that $\boldsymbol{P}{\downarrow_{\beta\eta}}$ is an LK-proof since the eigenvariable condition is preserved by $\beta\eta$-reduction ($\mathrm{FV}(t{\downarrow_{\beta\eta}}) \subseteq \mathrm{FV}(t)$ for every term t). The fact that inferences are defined only up to $\equiv_{\alpha\beta\eta}$-classes departs from standard practice and requires some explanations. It is made necessary by the fact that $(\xi\ t)$ corresponds to the replacement of a bound variable by t, as it should be according to Gentzen's calculus, only if ξ is instantiated by an abstraction $\lambda x\, \phi$ *and* the term $((\lambda x\, \phi)\ t)$ is rewritten to $\phi[t/x]$ by the β-rule.

For instance, an inference

$$\frac{(p\ y) \vdash \phi}{\exists(\lambda x\, (p\ x)) \vdash \phi}$$

(where $p : \mathbf{\iota} \to \mathbf{o}$ is a constant) holds by rule ($\exists$-L) if $y \notin \mathrm{FV}(\phi)$, since $((\lambda x\,(p\ x))\ y) \equiv_{\alpha\beta\eta} (p\ y)$. This inference exactly corresponds to the standard inference in Gentzen's calculus

$$\frac{p(y) \vdash \phi}{\exists x\, p(x) \vdash \phi}$$

It should thus be clear that proofs in Gentzen's calculus correspond to $\beta\eta$-normal $\mathcal{S}$-LK-proofs. We allow non-normal proofs for sake of simplicity, and also because we will consider $\mathcal{L}$-LK-proofs for other languages than $\mathcal{S}$ (provided such languages are stable under $\beta\eta$-reduction).

Example 3. We will use as a recurring illustrative example the following LK-proof $\boldsymbol{P}$.

$$\dfrac{\dfrac{\dfrac{}{(p\ a\ a) \vdash (p\ a\ a)}\,(\text{Axiom})}{\forall x\,(p\ x\ a) \vdash (p\ a\ a)}\,(\forall\text{-L})}{\forall x\,(p\ x\ a) \vdash \exists y\,(p\ y\ a)}\,(\exists\text{-R})$$

3 A Calculus with Explicit Constraints

We now wish to adapt the rules of Figure 1 in order to collect automatically all the syntactic constraints that make a proof valid, instead of checking them on the fly during proof construction. For most rules it is easy to express these constraints as a unification problem pertaining to the principal formulæ. However, eigenvariables are replaced by suitable eigenterms so that eigenvariable conditions may also be expressed as unification problems.

Definition 4. Let $\mathcal{L} \subseteq \mathcal{T}$ such that $\mathcal{V} \subseteq \mathcal{L}$. An *$\mathcal{L}$-constraint* (or *constraint* if $\mathcal{L} = \mathcal{T}$) is inductively defined as $\bot$, $\top$, $t \doteq s$ where t, s denote terms in $\mathcal{L}$ of the same type, or $\mathcal{X} \curlywedge \mathcal{Y}$, where $\mathcal{X}, \mathcal{Y}$ are two $\mathcal{L}$-constraints. A substitution σ is a *solution* of a constraint $\mathcal{X}$ if $\mathcal{X} = \top$ or ($\mathcal{X} = (t \doteq s)$ and $t\sigma \equiv_{\alpha\beta\eta} s\sigma$) or ($\mathcal{X} = (\mathcal{X}_1 \curlywedge \mathcal{X}_2)$ and σ is a solution of $\mathcal{X}_1$ and $\mathcal{X}_2$). The set of solutions of a constraint $\mathcal{X}$ is written $\mathrm{sol}(\mathcal{X})$. A constraint $\mathcal{X}$ is *satisfiable* if $\mathrm{sol}(\mathcal{X}) \neq \emptyset$; it is *valid* if every substitution belongs to $\mathrm{sol}(\mathcal{X})$ (equivalently, if $id \in \mathrm{sol}(\mathcal{X})$). We say that two constraints $\mathcal{X}$ and $\mathcal{Y}$ are *equivalent* and we write $\mathcal{X} \equiv \mathcal{Y}$ if $\mathrm{sol}(\mathcal{X}) = \mathrm{sol}(\mathcal{Y})$. $\mathcal{X}\sigma$ is obtained from $\mathcal{X}$ by replacing every $t \doteq s$ in it by $t\sigma \doteq s\sigma$. The equivalence $\equiv_{\alpha\beta\eta}$ is inductively extended to constraints by $\bot \equiv_{\alpha\beta\eta} \bot$, $\top \equiv_{\alpha\beta\eta} \top$, $(t \doteq s) \equiv_{\alpha\beta\eta} (t' \doteq s')$ if $t \equiv_{\alpha\beta\eta} t'$ and $s \equiv_{\alpha\beta\eta} s'$, $(\mathcal{X} \curlywedge \mathcal{Y}) \equiv_{\alpha\beta\eta} (\mathcal{X}' \curlywedge \mathcal{Y}')$ if $\mathcal{X} \equiv_{\alpha\beta\eta} \mathcal{X}'$ and $\mathcal{Y} \equiv_{\alpha\beta\eta} \mathcal{Y}'$. It is clear that $\mathcal{X} \equiv_{\alpha\beta\eta} \mathcal{Y} \Rightarrow \mathcal{X} \equiv \mathcal{Y}$.

A *constrained $\mathcal{L}$-sequent* is an expression of the form $\Gamma \vdash \Delta \mid \mathcal{X}$ where $\Gamma \vdash \Delta$ is an $\mathcal{L}$-sequent and $\mathcal{X}$ an $\mathcal{L}$-constraint. For any substitution σ, $(\Gamma \vdash \Delta \mid \mathcal{X})\sigma \stackrel{\text{def}}{=} \Gamma\sigma \vdash \Delta\sigma \mid \mathcal{X}\sigma$. The relation $\equiv_{\alpha\beta\eta}$ is extended to constrained sequents by $\Gamma \vdash \Delta \mid \mathcal{X} \equiv_{\alpha\beta\eta} \Gamma' \vdash \Delta' \mid \mathcal{X}'$ if $\Gamma \vdash \Delta \equiv_{\alpha\beta\eta} \Gamma' \vdash \Delta'$ and $\mathcal{X} \equiv_{\alpha\beta\eta} \mathcal{X}'$.

The rules of LK$^{\mathrm{c}}$ are given in Figure 2 in a condensed way; a rule of the form

$$\frac{\Gamma_1 \vdash \Delta_1 \cdots \Gamma_n \vdash \Delta_n}{\Gamma \vdash \Delta \mid \mathcal{X}} \text{ stands for } \frac{\Gamma_1 \vdash \Delta_1 \mid \mathcal{X}_1 \cdots \Gamma_n \vdash \Delta_n \mid \mathcal{X}_n}{\Gamma \vdash \Delta \mid \mathcal{X}_1 \curlywedge \cdots \curlywedge \mathcal{X}_n \curlywedge \mathcal{X}}.$$

$\Gamma, \Delta, \Sigma, \Pi$ are sequences of terms of type $\mathbf{o}$, $\chi, \phi, \psi, \varphi$ are terms of type $\mathbf{o}$, ξ is a term of type $\mathbf{\iota} \to \mathbf{o}$ and s, t are terms of type $\mathbf{\iota}$.

For any $\mathcal{L} \subseteq \mathcal{T}$ we call $\mathcal{L}$-LK$^{\mathrm{c}}$-proofs all proofs built with the rules of Figure 2 and containing only constrained $\mathcal{L}$-sequents. The *constraint of* an LK$^{\mathrm{c}}$-proof $\boldsymbol{P}$ is the constraint of its last constrained sequent. $\boldsymbol{P}$ is *valid* if its constraint is valid. For any substitution

$$\frac{\Gamma \vdash \Delta}{\Gamma,\, \phi \vdash \Delta \mid \top}\ (\text{W-L}^{\text{c}}) \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta,\, \phi \mid \top}\ (\text{W-R}^{\text{c}})$$

$$\frac{\Gamma,\, \phi,\, \psi \vdash \Delta}{\Gamma,\, \chi \vdash \Delta \mid \chi \doteq \phi \curlywedge \chi \doteq \psi}\ (\text{C-L}^{\text{c}}) \qquad \frac{\Gamma \vdash \Delta,\, \phi,\, \psi}{\Gamma \vdash \Delta,\, \chi \mid \chi \doteq \phi \curlywedge \chi \doteq \psi}\ (\text{C-R}^{\text{c}})$$

$$\frac{\Gamma,\, \psi,\, \phi,\, \Sigma \vdash \Delta}{\Gamma,\, \chi,\, \varphi,\, \Sigma \vdash \Delta \mid \chi \doteq \phi \curlywedge \varphi \doteq \psi}\ (\text{P-L}^{\text{c}}) \qquad \frac{\Gamma \vdash \Delta,\, \psi,\, \phi,\, \Pi}{\Gamma \vdash \Delta,\, \chi,\, \varphi,\, \Pi \mid \chi \doteq \phi \curlywedge \varphi \doteq \psi}\ (\text{P-R}^{\text{c}})$$

$$\frac{}{\phi \vdash \psi \mid \phi \doteq \psi}\ (\text{Axiom}^{\text{c}}) \qquad \frac{}{\vdash \phi \mid \phi \doteq t \simeq t}\ (\text{Ref}^{\text{c}})$$

$$\frac{\Gamma,\, \chi,\, \psi \vdash \Delta}{\Gamma,\, \chi,\, \phi \vdash \Delta \mid \chi \doteq t \simeq s \curlywedge \phi \doteq (\xi\ s) \curlywedge \psi \doteq (\xi\ t)}\ (\text{Param-L}^{\text{c}})$$

$$\frac{\Gamma,\, \chi \vdash \Delta,\, \psi}{\Gamma,\, \chi \vdash \Delta,\, \phi \mid \chi \doteq t \simeq s \curlywedge \phi \doteq (\xi\ s) \curlywedge \psi \doteq (\xi\ t)}\ (\text{Param-R}^{\text{c}})$$

$$\frac{\Gamma,\, \phi \vdash \Delta}{\Gamma,\, \chi \vdash \Delta \mid \phi \doteq s \simeq t \curlywedge \chi \doteq t \simeq s}\ (\text{Com-L}^{\text{c}}) \qquad \frac{\Gamma \vdash \Delta,\, \phi \quad \Sigma,\, \psi \vdash \Pi}{\Gamma,\, \Sigma \vdash \Delta,\, \Pi \mid \phi \doteq \psi}\ (\text{Cut}^{\text{c}})$$

$$\frac{\Gamma,\, \phi \vdash \Delta}{\Gamma,\, \chi \vdash \Delta \mid \chi \doteq \phi \wedge \psi}\ (\wedge\text{-L1}^{\text{c}}) \qquad \frac{\Gamma \vdash \Delta,\, \phi}{\Gamma \vdash \Delta,\, \chi \mid \chi \doteq \phi \vee \psi}\ (\vee\text{-R1}^{\text{c}})$$

$$\frac{\Gamma,\, \psi \vdash \Delta}{\Gamma,\, \chi \vdash \Delta \mid \chi \doteq \phi \wedge \psi}\ (\wedge\text{-L2}^{\text{c}}) \qquad \frac{\Gamma \vdash \Delta,\, \psi}{\Gamma \vdash \Delta,\, \chi \mid \chi \doteq \phi \vee \psi}\ (\vee\text{-R2}^{\text{c}})$$

$$\frac{\Gamma,\, \phi \vdash \Delta \qquad \Sigma,\, \psi \vdash \Pi}{\Gamma,\, \Sigma,\, \chi \vdash \Delta,\, \Pi \mid \chi \doteq \phi \vee \psi}\ (\vee\text{-L}^{\text{c}}) \qquad \frac{\Gamma \vdash \Delta,\, \phi \qquad \Sigma \vdash \Pi,\, \psi}{\Gamma,\, \Sigma \vdash \Delta,\, \Pi,\, \chi \mid \chi \doteq \phi \wedge \psi}\ (\wedge\text{-R}^{\text{c}})$$

$$\frac{\Gamma \vdash \Delta,\, \phi \qquad \Sigma,\, \psi \vdash \Pi}{\Gamma,\, \Sigma,\, \chi \vdash \Delta,\, \Pi \mid \chi \doteq \phi \Rightarrow \psi}\ (\Rightarrow\text{-L}^{\text{c}}) \qquad \frac{\Gamma,\, \phi \vdash \Delta,\, \psi}{\Gamma \vdash \Delta,\, \chi \mid \chi \doteq \phi \Rightarrow \psi}\ (\Rightarrow\text{-R}^{\text{c}})$$

$$\frac{\Gamma \vdash \Delta,\, \phi}{\Gamma,\, \chi \vdash \Delta \mid \chi \doteq \neg\phi}\ (\neg\text{-L}^{\text{c}}) \qquad \frac{\Gamma,\, \phi \vdash \Delta}{\Gamma \vdash \Delta,\, \chi \mid \chi \doteq \neg\phi}\ (\neg\text{-R}^{\text{c}})$$

$$\frac{\Gamma,\, \phi \vdash \Delta}{\Gamma,\, \chi \vdash \Delta \mid \chi \doteq \forall\xi \curlywedge \phi \doteq (\xi\ t)}\ (\forall\text{-L}^{\text{c}}) \qquad \frac{\Gamma \vdash \Delta,\, \phi}{\Gamma \vdash \Delta,\, \chi \mid \chi \doteq \exists\xi \curlywedge \phi \doteq (\xi\ t)}\ (\exists\text{-R}^{\text{c}})$$

$$\frac{\Gamma,\, \phi \vdash \Delta}{\Gamma, \chi \vdash \Delta \mid \chi \doteq \exists\xi \curlywedge \phi \doteq (\xi\ (\iota\, \Gamma\Delta\chi))}\ (\exists\text{-L}^{\text{c}})$$

$$\frac{\Gamma \vdash \Delta,\, \phi}{\Gamma \vdash \Delta, \chi \mid \chi \doteq \forall\xi \curlywedge \phi \doteq (\xi\ (\iota\, \Gamma\Delta\chi))}\ (\forall\text{-R}^{\text{c}})$$

Figure 2: The constrained sequent calculus LK^{c}

σ, $\boldsymbol{P}\sigma$ is obtained by applying σ to the constrained sequents in $\boldsymbol{P}$. The relation $\equiv_{\alpha\beta\eta}$ is inductively extended to LK^{c}-proofs by

$$\frac{\boldsymbol{P}_1 : s_1 \cdots \boldsymbol{P}_n : s_n}{s}(R) \equiv_{\alpha\beta\eta} \frac{\boldsymbol{Q}_1 : s'_1 \cdots \boldsymbol{Q}_n : s'_n}{s'}(R)$$

if $s \equiv_{\alpha\beta\eta} s'$ and $\boldsymbol{P}_i \equiv_{\alpha\beta\eta} \boldsymbol{Q}_i$ for all $1 \leq i \leq n$. The *length* of an LK^{c}- or LK-proof is the number of inferences occurring in it.

For any substitution σ and object O on which $O\sigma$ is defined, we write $O \preceq O\sigma$ and say that $O\sigma$ is an *instance* of O. The relation $\preceq$ is a quasiorder since $Oid = O$ and $O(\sigma\mu) = (O\sigma)\mu$ always hold.

If furthermore the relation $\equiv_{\alpha\beta\eta}$ is defined between objects O and O', we write $O \precsim O'$ (and say that O is *more general* than O'), if there is a substitution ρ such that $O\rho \equiv_{\alpha\beta\eta} O'$. The relation $\precsim$ is a quasiorder since, as is well known in λ-calculus, the relation $\equiv_{\alpha\beta\eta}$ is substitutive, i.e., $O \equiv_{\alpha\beta\eta} O'$ entails $O\rho \equiv_{\alpha\beta\eta} O'\rho$ for any substitution ρ. An object O that verifies some property P is *most general w.r.t.* P if all objects O' verifying P also verify $O \precsim O'$; and O is *minimal w.r.t.* P if all O' verifying P and $O' \precsim O$ also verify $O' \equiv_{\alpha\beta\eta} O$.

Example 5. The following is an LK$^{\mathrm{c}}$-proof $\boldsymbol{Q}$ with the same end-sequent as in Example 3 (discarding the constraints). Symbols u, v, ξ, ψ are variables of type $\mathbf{\imath}$, $\mathbf{\imath}$, $\mathbf{\imath} \to \mathbf{o}$ and $\mathbf{o}$ respectively.

$$\dfrac{\dfrac{\dfrac{}{\psi \vdash (p\ v\ a) \mid \mathcal{X}_1}\,(\mathrm{Axiom}^{\mathrm{c}})}{\forall x\,(p\ x\ a) \vdash (p\ v\ a) \mid \mathcal{X}_2}\,(\forall\text{-L}^{\mathrm{c}})}{\forall x\,(p\ x\ a) \vdash \exists y\,(p\ y\ a) \mid \mathcal{X}_3}\,(\exists\text{-R}^{\mathrm{c}})$$

where

$$\begin{array}{lcl} \mathcal{X}_1 & \stackrel{\mathrm{def}}{=} & \psi \doteq (p\ v\ a), \\ \mathcal{X}_2 & \stackrel{\mathrm{def}}{=} & \mathcal{X}_1 \curlywedge \forall x\,(p\ x\ a) \doteq \forall \xi \curlywedge \psi \doteq (\xi\ u), \\ \mathcal{X}_3 & \stackrel{\mathrm{def}}{=} & \mathcal{X}_2 \curlywedge \exists y\,(p\ y\ a) \doteq \exists y\,(p\ y\ a) \curlywedge (p\ v\ a) \doteq (\lambda y\,(p\ y\ a)\ v). \end{array}$$

It is clear that $\mathcal{X}_3 \equiv \psi \doteq (p\ v\ a) \curlywedge \xi \doteq \lambda x\,(p\ x\ a) \curlywedge u \doteq v$. Note that in the above LK$^{\mathrm{c}}$-proof, some of the meta-variables of Figure 2 are instantiated by usual formulæ or terms, i.e., built solely with symbols from the signature $\mathcal{C}$ and variables of type $\mathbf{\imath}$ (e.g, $\forall x\,(p\ x\ a)$), and others contain other variables ($\psi : \mathbf{o}$, $\xi : \mathbf{\imath} \to \mathbf{o}$). The proof is not valid: to get an LK-proof one has to instantiate all the variables in such a way that $\mathcal{X}_3$ is fulfilled, for instance with $\rho \stackrel{\mathrm{def}}{=} [a/u,\ a/v,\ \lambda x\,(p\ x\ a)/\xi,\ (p\ a\ a)/\psi]$.

Contrary to LK the rules are strictly syntactic; the constrained sequents occurring in a proof cannot be replaced by $\alpha\beta\eta$-equivalent constrained sequents. This is possible because $\alpha\beta\eta$-equivalence is taken care of in the constraints.

Another essential difference between LK and LK$^{\mathrm{c}}$ is that the latter introduces eigenterms instead of eigenvariables. An important though trivial consequence is substitutivity of LK$^{\mathrm{c}}$-proofs.

Proposition 6. *For any substitutive $\mathcal{L} \subseteq \mathcal{T}$, $\mathcal{L}$-LK$^{\mathrm{c}}$-proof $\boldsymbol{P}$ and $\mathcal{L}$-substitution σ, $\boldsymbol{P}\sigma$ is an $\mathcal{L}$-LK$^{\mathrm{c}}$-proof.*

Proof. The inference rules are all expressed with meta-variables and the only side conditions pertain to their type (including sequences of terms of type $\mathbf{o}$ and constraints). Since these types, as well as membership in $\mathcal{L}$, are preserved by $\mathcal{L}$-substitutions then so are the inference rules and therefore the proofs. □

Since the relation $\equiv_{\alpha\beta\eta}$ is defined on LK$^{\mathrm{c}}$-proofs we can now deduce that $\sigma \equiv_{\alpha\beta\eta} \theta$ entails $\boldsymbol{P}\sigma \equiv_{\alpha\beta\eta} \boldsymbol{P}\theta$ for any LK$^{\mathrm{c}}$-proof $\boldsymbol{P}$ and substitutions σ, θ. Proposition 6 would also hold on LK-proofs if substitutions were applied with a form of α-conversion of eigenvariables in order to avoid their "capture" (as in the definition of $(\lambda x\,t)\sigma$ above). This would allow a notion of more general proofs in LK, but we have chosen to dispense with such developments since they are not essential to this generalization method.

4 Semantics of LK^{c}

Example 5 shows that LK^{c} offers much freedom in building proofs, including proofs with unsatisfiable constraints which are meaningless. But even proofs with valid constraints may look strange compared to LK-proofs. In particular, the possibility in LK^{c} to paramodulate in eigenterms has no equivalence in LK, and it is not obvious that LK^{c} is sound. We prove in this section that this is the case by providing a semantics for sequents provable in LK^{c}. We first extend the usual definition of an interpretation. The main difference with the standard definition is that we need to interpret eigenterms. The constructor ι cannot be interpreted in the usual way, by mapping sequences of booleans to elements of the domain: it is clear that the value of $(\iota\,\phi_1 \cdots \phi_n)$ depends on the formulæ $\phi_1, \ldots, \phi_n$ and not only on their truth values.

Definition 7. Let D be any[2] non-empty set, we write $\mathcal{T}_D$ for the set of *D-terms*, which is inductively defined exactly as $\mathcal{T}$ except that elements of D are allowed as constants of type $\mathbf{\imath}$.

An *interpretation* I is defined by providing:

- a non-empty set D, called the *domain* of I,
- a function mapping every variable or constant f (distinct from $\simeq$) of type $\mathbf{\imath}^n \to \mathbf{\imath}$ (resp. $\mathbf{\imath}^n \to \mathbf{o}$) to a function f^I from D^n to D (resp. $\{\mathtt{true}, \mathtt{false}\}$),
- a function mapping every connective $\star \in \{\vee, \wedge, \Rightarrow, \neg, \forall, \exists, \simeq\}$ to the usual function $\star^I$ (for instance $x \vee^I y = \mathtt{true}$ iff $x = \mathtt{true}$ or $y = \mathtt{true}$; $\forall^I p = \mathtt{true}$ iff $p(x) = \mathtt{true}$ for all $x \in D$; etc.),
- a function ι^I mapping every non-empty sequence of D-terms of type $\mathbf{o}$ to an element of D.

Any interpretation I of domain D can be extended to an evaluation t^I of D-terms t in the following way:

- for $x \in D$, $x^I = x$,
- for $s, t \in \mathcal{T}_D$, where $s : \tau \to \tau'$ and $t : \tau$, $(s\,t)^I$ is the value of the function s^I at t^I,
- for $t \in \mathcal{T}_D$, $x \in \mathcal{V}$ where $t : \mathbf{\imath} \to \tau$ and $x : \mathbf{\imath}$, $(\lambda x\, t)^I$ is the function that maps every $e \in D$ to $(t[e/x])^I$,
- for $n \in \mathbb{N}$, $n > 0$, $\phi_1, \ldots, \phi_n \in (\mathcal{T}_D)^n$ where $\phi_i : \mathbf{o}$ for $1 \leq i \leq n$, $(\iota\,\phi_1 \cdots \phi_n)^I = \iota^I(\phi_1 \cdots \phi_n)$.

For every term $t \in \mathcal{T}_D$, $|t|$ denotes the *size* of t, inductively defined as follows: $|u| \stackrel{\text{def}}{=} 1$ if $u \in \mathcal{V} \cup \mathcal{C} \cup D$, $|\lambda x\, t| \stackrel{\text{def}}{=} 1 + |t|$, $|(l\ t)| \stackrel{\text{def}}{=} |l| + |t|$ and $|(\iota\,\phi_1 \cdots \phi_n)| = 1 + \sum_{i=1}^{n} |\phi_i|$. If $\Gamma = t_1 \cdots t_n$ is a sequence of terms, then $|\Gamma| \stackrel{\text{def}}{=} \max_{i=1}^{n} |t_i|$, with $|\Gamma| \stackrel{\text{def}}{=} 0$ if $n = 0$.

The usual notions of models, satisfiability, logical consequence $\models$,... can be extended accordingly. We write $\Gamma \models_{\mathrm{s}} \Delta$ if every model of the formulæ in Γ is a model of *some* formula in Δ.

[2] We assume w.l.o.g. that $D \cap (\mathcal{C} \cup \mathcal{V}) = \emptyset$ and that D contains no non-atomic terms in $\mathcal{T}_D$.

Note that this semantics coincides with the usual one for closed standard terms. The previous definition is similar to the interpretation of ϵ-terms in the ϵ-calculus (see, e.g., [19, 20]), which depends on the formulæ themselves[3] and not only on their interpretation. However, our definition departs from the semantics of the ϵ-calculus because the interpretation of the operator ι is arbitrary. This is due to the fact that the operator ι does not separate the considered existential formula from the other elements of the context. Furthermore, it is intended to be used only as a tool for writing proofs and not as a logical construct in the language.

Lemma 8. *Let I, J be two interpretations on the same domain D and identical on $\mathcal{V}$ and $\mathcal{C}$. For all $t \in \mathcal{T}_D$, if*

(*) for every non-empty sequence $\Gamma \in (\mathcal{T}_D)^+$ of D-terms of type $\mathbf{o}$ such that $|\Gamma| < |t|$, $\iota^I(\Gamma) = \iota^J(\Gamma)$,

then $t^I = t^J$.

Proof. By induction on t. This is trivial if $t \in \mathcal{V} \cup \mathcal{C} \cup D$, by the fact that I and J then coincide on t. Note that if we assume (*) on a term $(s\ t)$ then it holds both for s and t since $|s| < |(s\ t)|$ an $|t| < |(s\ t)|$, hence by induction hypothesis we get $s^I = s^J$ and $t^I = t^J$, from which $(s\ t)^I = (s\ t)^J$ is obvious. Similarly, if we assume (*) on $(\lambda x\, t)$ then for all $e \in D$, (*) holds on $t[e/x]$ since $|t[e/x]| = |t| < |(\lambda x\, t)|$, hence by induction hypothesis we get $(t[e/x])^I = (t[e/x])^J$, and therefore $(\lambda x\, t)^I = (\lambda x\, t)^J$. Finally, if $t = (\iota\, \phi_1 \cdots \phi_n)$ then $|\phi_1 \cdots \phi_n| < |t|$ hence by (*) we get $t^I = \iota^I(\phi_1 \cdots \phi_n) = \iota^J(\phi_1 \cdots \phi_n) = t^J$. □

The next theorem states that LK^{c} is sound.

Theorem 9. *Let $\boldsymbol{P}$ be an LK^{c}-proof of constrained end-sequent $\Gamma \vdash \Delta \mid \mathcal{X}$, then for every substitution $\sigma \in \mathrm{sol}(\mathcal{X})$, $\Gamma\sigma \models_{\mathrm{s}} \Delta\sigma$ holds.*

Proof. By induction on $\boldsymbol{P}$; we only consider one axiom and one inference rule.

If

$$\boldsymbol{P} = \frac{}{\vdash \phi \mid \phi \doteq t \simeq t}(\mathrm{Ref}^{\mathrm{c}})$$

and $\sigma \in \mathrm{sol}(\phi \doteq t \simeq t)$, then for any interpretation I, $(\phi\sigma)^I = (t\sigma \simeq t\sigma)^I = \mathtt{true}$, hence $\models_{\mathrm{s}} \phi\sigma$.

The proof is similar for all rules except the strong quantifier rules: let

$$\boldsymbol{P} = \frac{\boldsymbol{P}_1 : \Gamma, \phi \vdash \Delta \mid \mathcal{X}}{\Gamma, \chi \vdash \Delta \mid \mathcal{X} \curlywedge \chi \doteq \exists\xi \curlywedge \phi \doteq (\xi\ (\iota\, \Gamma\Delta\chi))}(\exists\text{-L}^{\mathrm{c}})$$

and $\sigma \in \mathrm{sol}(\mathcal{X} \curlywedge \chi \doteq \exists\xi \curlywedge \phi \doteq (\xi\ (\iota\, \Gamma\Delta\chi)))$. Let $\zeta = \xi\sigma$ and $\Pi = \Gamma\sigma\Delta\sigma\chi\sigma$. Assume that $\Gamma\sigma, \chi\sigma \not\models_{\mathrm{s}} \Delta\sigma$, then there is an interpretation I with domain D such that $I \models \Gamma\sigma$, $I \models \chi\sigma$ and $I \not\models_{\mathrm{s}} \Delta\sigma$. Since $\chi\sigma = \exists\zeta$, there exists an element $e \in D$ such that $\zeta^I(e) = \mathtt{true}$. Let J be the interpretation of domain D that is identical to I except that $\iota^J(\Pi) = e$. For all D-terms t such that $|t| \leq |\Pi|$, it is obvious by construction of J that $\iota^J(\Gamma') = \iota^I(\Gamma')$ for every $\Gamma' \in (\mathcal{T}_D)^+$ such that $|\Gamma'| < |t|$ (since then $\Gamma' \neq \Pi$), hence by Lemma 8 that $t^J = t^I$. In particular, $|\Gamma\sigma| \leq |\Pi|$ and $|\Delta\sigma| \leq |\Pi|$, hence $J \models \Gamma\sigma$ and $J \not\models_{\mathrm{s}} \Delta\sigma$. Furthermore, $|\zeta| < |\exists\zeta| = |\chi\sigma| \leq |\Pi|$, hence

$$(\phi\sigma)^J = (\zeta\ (\iota\, \Pi))^J = \zeta^J(e) = \zeta^I(e) = \mathtt{true},$$

so that $J \models \phi\sigma$. Hence $\Gamma\sigma, \phi\sigma \not\models_{\mathrm{s}} \Delta\sigma$, in contradiction with the induction hypothesis on $\boldsymbol{P}_1$. Therefore, $\Gamma\sigma, \chi\sigma \models_{\mathrm{s}} \Delta\sigma$ must hold. □

In particular, if $\Gamma \vdash \Delta$ is a standard sequent such that some $\Gamma \vdash \Delta \mid \mathcal{X}$ is provable in LK^{c} with $\mathcal{X} \equiv \top$, then $\Gamma \models_{\mathrm{s}} \Delta$ holds in the standard semantics of first order logic.

[3] More precisely on a special kind of formulæ called "matrices".

5 Proof Translations

It is important to notice that, although LK^{c} is sound, not all LK^{c}-proofs can be directly transformed into LK-proofs, as illustrated by the following example.

Example 10. Consider for instance the following LK^{c}-proof of a statement $a \simeq b, \exists x(p\, a\, x) \vee \exists x(p\, b\, x) \vdash \phi$.

$$\dfrac{\dfrac{\dfrac{\boldsymbol{P}}{a \simeq b, (p\ a\ sk_a) \vdash \phi \mid \top}\,(\text{W-L}^{\text{c}})}{a \simeq b, \exists x(p\, a\, x) \vdash \phi \mid \top}\,(\exists\text{-L}^{\text{c}}) \qquad \dfrac{\dfrac{\dfrac{\boldsymbol{P}}{a \simeq b, (p\ a\ sk_a) \vdash \phi \mid \top}\,(\text{W-L}^{\text{c}})}{a \simeq b, (p\ b\ sk_b) \vdash \phi \mid \top}\,(\text{Param-L}^{\text{c}})}{a \simeq b, \exists x(p\, b\, x) \vdash \phi \mid \top}\,(\exists\text{-L}^{\text{c}})}{a \simeq b, \exists x(p\, a\, x) \vee \exists x(p\, b\, x) \vdash \phi \mid \top}\,(\vee\text{-L}^{\text{c}})$$

The term sk_u (with $u = a, b$) denotes the eigenterm $(\iota\, a \simeq b\, \exists x(p\, u\, x)\, \phi)$. The symbol $\boldsymbol{P}$ denotes some proof of $(p\ a\ sk_a) \vdash \phi \mid \top$. For sake of conciseness valid constraints are abstracted as $\top$.

It is easy to see that this proof cannot be (directly) translated into an LK-proof, because of the paramodulation step occurring inside the eigenterm in the right branch (replacing $(p\ b\ sk_b)$ by $(p\ a\ sk_a)$, while sk_a and sk_b would be distinct eigenvariables in the corresponding LK-proof). In order to transform the above proof into an LK-proof one has to perform a non-trivial reorganization of the inferences, e.g., to apply the paramodulation step on the non-skolemized version of the formula (replacing $(\exists x(p\, b\, x)$ by $(\exists x(p\, a\, x))$. It is unclear whether such a transformation can be done in a systematic and purely automatic way, thus from a practical point of view we prefer to add restrictions that dismiss such inferences.

A straightforward solution to this problem would be to dismiss strong quantifiers entirely, by assuming that all statements are skolemized. However, this would significantly reduce the scope of our generalization algorithm, especially when applied to proofs produced by humans (who rarely consider skolemized statements). This remark is particularly relevant for proofs containing cuts, because in this case strong quantifiers may occur even if the end-sequent is skolemized. Worse, applying the generalization algorithm on a skolemized version of a proof may yield a statement that is not necessarily more general than the initial one (although it is of course more general than its skolemized version), since the quantifier structure is destroyed by the transformation.

A similar solution would be to use the ϵ-calculus (see, e.g., [19, 27]), in which quantifiers are replaced by a special operator ϵ_x mapping each formula $\phi(x)$ to an individual e such that $\phi(e)$ holds (if such an element exists, otherwise $\epsilon_x(\phi(x))$ is arbitrary). Again, the relation between generalized proofs in the ϵ-calculus and standard LK-proofs is unclear. Note that, unlike the ϵ-calculus or calculi operating on skolemized formulæ the LK^{c} calculus is not intended to be used to construct proofs (i.e., as an alternative to LK), but rather as a theoretical tool to generalize existing LK-proofs. Our operator ι shares obvious similarities with the ϵ_x operator, since a witness of some existential quantified formula is returned in both cases. However, ι has additional arguments encoding the context in which the existential formula occurs. Removing these arguments would make the calculus unsound (for instance the sequent $\exists x(p\ x) \vdash p(\iota(\exists x(p\ x)))$ would be provable although it is not valid) unless ι is interpreted as an existential witness. Furthermore, the translation of the obtained proofs into LK would be more problematic, because of the additional constraints induced by the quantifier structure. For instance, in order to prove the sequent $\forall x((p\ x) \wedge \exists y \neg(p\ y)) \vdash$, the variable x has to be instantiated twice in LK, once with an arbitrary term to get the

formula $\exists y \neg(p\ y)$, and once with the witness of $\exists y \neg(p\ y)$. Using ι-terms without context arguments, it would be possible to instantiate x directly with the witness of $\exists y \neg(p\ y)$.

In order to avoid this, we need to restrict LK$^{\mathrm{c}}$-proofs to a language that allow them to be translated into LK-proofs.

Definition 11. Let $\mathcal{K}$ be the set of terms $t \in \mathcal{T}$ in which the eigenterms $(\iota\, \phi_1 \cdots \phi_n)$ do not occur in the scope of λ-abstractions binding one of their free variables (in other words, every free occurrence of a variable in any ϕ_i is also free in t). We will refer to $\mathcal{K}$-sequents, $\mathcal{K}$-substitutions, $\mathcal{K}$-constraints, $\mathcal{K}$-LK-proofs and $\mathcal{K}$-LK$^{\mathrm{c}}$-proofs if they only contain terms in $\mathcal{K}$.

Remark 12. In the definition of $\mathcal{K}$ above, the restriction on the special symbol ι is essentially identical to the restriction introduced in [17] in order to ensure the soundness of the skolemization operation in higher order logic. Indeed, a direct use of the standard skolemization operator (as done in [1]) turns out to be unsound in second order logic, since the axiom of choice is not provable. In order to recover soundness, restrictions have been added in [16, 17, 18] to ensure that the Skolem symbols are used only to express dependencies between terms, and not as actual functions in the signature. While the restriction we use is similar, our purpose is different. Indeed, soundness is always guaranteed in our context, as it is shown by Theorem 9. However, the explicit use of the function ι, mapping formulæ to existential witnesses, allows one to express reasonings that cannot be expressed in LK, although they are sound at the meta-level, such as the fact that two formulæ that are equivalent can be associated with the same witness (see Example 10). In order to ensure that LK$^{\mathrm{c}}$-proofs can be translated in LK, we need to keep the witness symbol apart, so that it cannot be used to construct new terms, beside those corresponding to the eigenvariables.

We now investigate the properties of $\mathcal{K}$. Obviously, $\mathcal{S} \subset \mathcal{K} \subset \mathcal{T}$ and $\mathcal{K}$ is closed under $\rightarrow_\alpha$.

Lemma 13. *If $t, s \in \mathcal{K}$ then $t[s/x] \in \mathcal{K}$ for every variable x of appropriate type.*

Proof. Any occurrence of an eigenterm in $t[s/x]$ comes either from t or s. If it comes from s then it occurs there as a subterm $l = (\iota\, \phi_1 \cdots \phi_n)$ whose free variables are free in s; thus l is again a subterm of $t[s/x]$ and since the free variables of s are not captured in the substitution then the free variables of l are free in $t[s/x]$ as required.

If it comes from t then it occurs there as a subterm $(\iota\, \phi_1 \cdots \phi_n)$ whose free variables are free in t and must therefore be preserved in the substitution (not α-converted); thus the considered eigenterm is of the form $(\iota\, \phi_1[s/x] \ldots \phi_n[s/x])$. Its free variables are either free in some ϕ_i (but other than x) and therefore free in t and hence in $t[s/x]$, or they are free in s and then also free in $t[s/x]$ (by absence of capture). □

Proposition 14. *$\mathcal{K}$ is substitutive and closed under $\alpha\beta\eta$-reduction.*

Proof. It is obvious from Lemma 13 that $\mathcal{K}$ is substitutive, and from Definition 11 that if s is a subterm of $t \in \mathcal{K}$ then $s \in \mathcal{K}$, i.e., that $\mathcal{K}$ is closed for subterms. If $(\lambda x\, (l\ x)) \in \mathcal{K}$ and $x \notin \mathrm{FV}(l)$, then $l \in \mathcal{K}$, hence $\mathcal{K}$ is closed under $\rightarrow_\eta$. If $((\lambda x\, l)\ t) \in \mathcal{K}$, then $l, t \in \mathcal{K}$, hence $t[l/x] \in \mathcal{K}$ by Lemma 13 and $\mathcal{K}$ is therefore also closed under $\rightarrow_\beta$. □

This means that, for any $\mathcal{K}$-LK$^{\mathrm{c}}$-proof $\boldsymbol{P}$ and $\mathcal{K}$-substitution σ, $\boldsymbol{P}\sigma$ is a $\mathcal{K}$-LK$^{\mathrm{c}}$-proof by Propositions 6 and 14. We now prove that standard LK-proofs can be translated into valid $\mathcal{K}$-LK$^{\mathrm{c}}$-proofs.

Theorem 15. *For any standard* LK*-proof* $\boldsymbol{P} : \Gamma \vdash \Delta$ *there exists a valid* $\mathcal{K}$-LK$^{\mathrm{c}}$*-proof* $\boldsymbol{Q} : \Gamma \vdash \Delta \mid \mathcal{X}$ *of the same length as* $\boldsymbol{P}$*.*

Proof. We assume w.l.o.g. that $\boldsymbol{P}$ is in $\beta\eta$-normal form, and we proceed by induction on $\boldsymbol{P}$.

If

$$\boldsymbol{P} = \frac{}{\vdash t \simeq t}(\mathrm{Ref})$$

then $t \in \mathcal{S}$; let

$$\boldsymbol{Q} = \frac{}{\vdash t \simeq t \mid t \simeq t \doteq t \simeq t}(\mathrm{Ref^c})$$

which is clearly a valid $\mathcal{K}$-LK$^{\mathrm{c}}$-proof.

All other cases are similar, the only difficulty lies with the translation of eigenvariables into eigenterms, hence we examine one more rule.

If

$$\boldsymbol{P} = \frac{\boldsymbol{P}_1 : \Gamma, \phi \vdash \Delta}{\Gamma, \exists\xi \vdash \Delta}(\exists\text{-L})$$

then there is an eigenvariable y such that $\phi \equiv_{\alpha\beta\eta} (\xi\ y)$ and $y \notin \mathrm{FV}(\Gamma, \Delta, \xi)$, so that replacing ϕ by $(\xi\ y)$ in the last sequent of $\boldsymbol{P}_1$ yields an LK-proof $\boldsymbol{P}'_1 : \Gamma, (\xi\ y) \vdash \Delta$. By induction hypothesis there is a $\mathcal{K}$-LK$^{\mathrm{c}}$-proof $\boldsymbol{Q}_1 : \Gamma, (\xi\ y) \vdash \Delta \mid \mathcal{X}_1$ where $\mathcal{X}_1$ is a valid $\mathcal{K}$-LK$^{\mathrm{c}}$-proof. Let $\sigma = [(\iota\,\Gamma\Delta\exists\xi)/y]$, obviously Γ, Δ, ξ are fixpoints of σ. Let

$$\boldsymbol{Q} = \frac{\boldsymbol{Q}_1\sigma : \Gamma, (\xi\ (\iota\,\Gamma\Delta\exists\xi)) \vdash \Delta \mid \mathcal{X}_1\sigma}{\Gamma, \exists\xi \vdash \Delta \mid \mathcal{X}_1\sigma \curlywedge \exists\xi \doteq \exists\xi \curlywedge (\xi\ (\iota\,\Gamma\Delta\exists\xi)) \doteq (\xi\ (\iota\,\Gamma\Delta\exists\xi))}(\exists\text{-L}^{\mathrm{c}})$$

which is a $\mathcal{K}$-LK$^{\mathrm{c}}$-proof by Propositions 6 and 14 and is obviously valid. □

This translation provides the first step in our generalization procedure; we also need a reverse translation in its last step. In order to translate valid $\mathcal{K}$-LK$^{\mathrm{c}}$-proofs into standard LK-proofs, we first need to replace eigenterms $(\iota\,\phi_1 \cdots \phi_n)$ by eigenvariables x (of type $\mathbf{\iota}$). This replacement cannot take place within LK$^{\mathrm{c}}$ since eigenterms are mandatory in LK$^{\mathrm{c}}$-proofs, hence we will perform them within LK and therefore use $\mathcal{K}$-LK-proofs. This requires to investigate the behavior of such replacements with respect to $\equiv_{\alpha\beta\eta}$.

By abuse of notation, for any $t \in \mathcal{K}$ we write $t[x/(\iota\,\phi_1 \cdots \phi_n)]$ for the term obtained from t after every occurrence of $(\iota\,\phi_1 \cdots \phi_n)$ has been replaced by x; this is clearly not a substitution.

Lemma 16. *Let* $t = (\iota\,\phi_1 \cdots \phi_n) \in \mathcal{K}$ *in* β*-normal form, let* $l, m \in \mathcal{K}$ *such that* $l \to_\beta m$ *and let* $x : \mathbf{\iota}$ *be a variable that does not occur in* l *or* m*, then* $l[x/t] \to_\beta m[x/t]$*.*

Proof. By induction on l. Obviously l cannot be a variable, a constant or t since it must contain a β-redex. If $l = (\lambda y\, l')$ for some variable $y : \mathbf{\iota}$ and $l' \in \mathcal{K}$, then $m = (\lambda y\, m')$ with $l' \to_\beta m'$, hence by induction hypothesis $l'[x/t] \to_\beta m'[x/t]$, hence

$$l[x/t] = (\lambda y\, l'[x/t]) \to_\beta (\lambda y\, m'[x/t]) = m[x/t].$$

If $l = (\iota\,\psi_1 \cdots \psi_m)$ then there exist $1 \leq i \leq m$ and a term ψ_i' such that $\psi_i \to_\beta \psi_i'$ and $m = (\iota\,\psi_1 \cdots \psi_i' \cdots \psi_m)$. By induction hypothesis $\psi_i[x/t] \to_\beta \psi_i'[x/t]$, hence

$$l[x/t] = (\iota\,\psi_1[x/t] \cdots \psi_m[x/t]) \to_\beta (\iota\,\psi_1[x/t] \cdots \psi_i'[x/t] \cdots \psi_m[x/t]) = m[x/t].$$

If $l = (l'\ s)$ with $l', s \in \mathcal{K}$ and this is not the redex used to obtain m, then this redex must occur either in l' or in s, hence $l[x/t] = (l'[x/t]\ s[x/t])$ and we prove as above that $l[x/t] \to_\beta m[x/t]$.

Otherwise $l = (l'\ s)$ is the redex used to obtain m, in which case $l' = (\lambda y\, m')$ for some variable $y : \mathbf{1}$ distinct from x and $m' \in \mathcal{K}$, so that $m = m'[s/y]$. Obviously

$$l[x/t] = ((\lambda y\, m'[x/t])\ s[x/t]) \to_\beta m'[x/t][s[x/t]/y].$$

Assume this last term is different from $m[x/t] = m'[s/y][x/t]$, then there must be in $m'[x/t][s[x/t]/y]$ an occurrence of t left. This means that there is in $m'[x/t]$ a subterm t' other than t or y such that $t'[s[x/t]/y] = t$, hence t' is some $(\iota\,\phi_1' \cdots \phi_n')$ and contains a free occurrence of y, which is impossible since $(\lambda y\, m'[x/t]) = l'[x/t] \in \mathcal{K}$. Hence $l[x/t] \to_\beta m[x/t]$, which completes the induction. □

Lemma 17. *Let $t = (\iota\,\phi_1 \cdots \phi_n) \in \mathcal{K}$ in η-normal form, let $l, m \in \mathcal{K}$ such that $l \to_\eta m$ and let $x : \mathbf{1}$ be a variable that does not occur in l or m, then $l[x/t] \to_\eta m[x/t]$.*

Proof. We only consider the case where l is an η-redex $(\lambda y\,(m\ y))$ where $y \notin \mathrm{FV}(m)$, then $l[x/t] = (\lambda y\,(m[x/t]\ y)) \to_\eta m[x/t]$ since obviously $y \notin \mathrm{FV}(m[x/t])$. The other cases can be handled by a straightforward inductive decomposition as in the proof of Lemma 16. □

Corollary 18. *Let $t = (\iota\,\phi_1 \cdots \phi_n) \in \mathcal{K}$ in $\beta\eta$-normal form, let $l, m \in \mathcal{K}$ such that $l \equiv_{\alpha\beta\eta} m$ and let $x : \mathbf{1}$ be a variable that does not occur in l or m, then $l[x/t] \equiv_{\alpha\beta\eta} m[x/t]$.*

We can now start removing eigenterms from $\mathcal{K}$-LK-proofs, one at a time.

Lemma 19. *If $\boldsymbol{P}$ is a $\mathcal{K}$-LK-proof in $\beta\eta$-normal form, $(\iota\,\phi_1 \cdots \phi_n) \in \mathcal{K}$ is in $\beta\eta$-normal form and $x : \mathbf{1}$ is a variable that does not occur in $\boldsymbol{P}$, then $\boldsymbol{P}[x/(\iota\,\phi_1 \cdots \phi_n)]$ is a $\mathcal{K}$-LK-proof of the same length as $\boldsymbol{P}$.*

Proof. The proof is by induction on $\boldsymbol{P}$; we let $l = (\iota\,\phi_1 \cdots \phi_n)$ and only consider one axiom and two inference rules.

If

$$\boldsymbol{P} = \frac{}{\vdash t \simeq t}(\mathrm{Ref})$$

with $t \in \mathcal{K}$, since x does not occur in t we deduce that $t[x/l] \in \mathcal{K}$, and since $(t \simeq t)[x/l] = t[x/l] \simeq t[x/l]$, then obviously

$$\boldsymbol{P}[x/l] := \frac{}{\vdash t[x/l] \simeq t[x/l]}(\mathrm{Ref})$$

is a $\mathcal{K}$-LK-proof. All propositional rules can be handled similarly.

If

$$\boldsymbol{P} = \frac{\boldsymbol{P}_1 : \Gamma, t \simeq s, \phi \vdash \Delta}{\Gamma, t \simeq s, \psi \vdash \Delta}(\text{Param-L})$$

then there is a term $\xi : \mathbf{1} \to \mathbf{o}$ in $\mathcal{K}$ such that $\phi \equiv_{\alpha\beta\eta} (\xi\ t)$ and $\psi \equiv_{\alpha\beta\eta} (\xi\ s)$. By induction hypothesis

$$\boldsymbol{P}_1[x/l] : \Gamma[x/l], t[x/l] \simeq s[x/l], \phi[x/l] \vdash \Delta[x/l]$$

is a $\mathcal{K}$-LK-proof, and by Corollary 18 we know that $\phi[x/l] \equiv_{\alpha\beta\eta} (\xi\ t)[x/l] = (\xi[x/l]\ t[x/l])$ (since $(\xi\ t) \neq l$ by type mismatch) and $\psi[x/l] \equiv_{\alpha\beta\eta} (\xi[x/l]\ s[x/l])$, hence

$$\boldsymbol{P}[x/l] : \Gamma[x/l], t[x/l] \simeq s[x/l], \psi[x/l] \vdash \Delta[x/l]$$

is an LK-proof since its last inference is an instance of (Param-L). The other rules are similar but we still need to check whether the eigenvariable condition is preserved.

If

$$\boldsymbol{P} = \frac{\boldsymbol{P}_1 : \Gamma, \phi \vdash \Delta}{\Gamma, \exists\xi \vdash \Delta}(\exists\text{-L})$$

then there is a variable $y \notin \mathrm{FV}(\Gamma, \Delta, \xi)$ such that $\phi \equiv_{\alpha\beta\eta} (\xi\ y)$. Note that if $y \notin \mathrm{FV}(\phi)$ then $x = y$ is possible (since in this case y does not occur in $\boldsymbol{P}$), in which case y can be replaced by another suitable variable without affecting $\boldsymbol{P}$, so that we may assume $y \neq x$. By induction hypothesis

$$\boldsymbol{P}_1[x/l] : \Gamma[x/l], \phi[x/l] \vdash \Delta[x/l]$$

is a $\mathcal{K}$-LK-proof, and by Corollary 18 $\phi[x/l] \equiv_{\alpha\beta\eta} (\xi\ y)[x/l] = (\xi[x/l]\ y)$. Since obviously $y \notin \mathrm{FV}(\Gamma[x/l], \Delta[x/l], \xi[x/l])$ (and these terms are in $\beta\eta$-normal form) then

$$\boldsymbol{P}[x/l] : \Gamma[x/l], \exists\xi[x/l] \vdash \Delta[x/l]$$

is a $\mathcal{K}$-LK-proof. □

Lemma 20. *If $\boldsymbol{P} : \Gamma \vdash \Delta$ is a $\mathcal{K}$-LK-proof then there is a standard* LK*-proof $\boldsymbol{Q} : \Gamma' \vdash \Delta'$ of the same length as $\boldsymbol{P}$ such that $\Gamma' \vdash \Delta' \preceq \Gamma{\downarrow_{\beta\eta}} \vdash \Delta{\downarrow_{\beta\eta}}$. If furthermore $\boldsymbol{P}$ is fair then so is $\boldsymbol{Q}$.*

Proof. By Proposition 14 $\boldsymbol{P}{\downarrow_{\beta\eta}}$: $\Gamma{\downarrow_{\beta\eta}} \vdash \Delta{\downarrow_{\beta\eta}}$ is a $\mathcal{K}$-LK-proof. By Lemma 19 we can replace all eigenterms in $\boldsymbol{P}{\downarrow_{\beta\eta}}$ by new variables and thus obtain a standard LK-proof $\boldsymbol{Q} : \Gamma' \vdash \Delta'$ of the same length as $\boldsymbol{P}$, and it is obvious that there is a substitution σ, replacing these new variables by their corresponding eigenterms, such that $(\Gamma' \vdash \Delta')\sigma = \Gamma{\downarrow_{\beta\eta}} \vdash \Delta{\downarrow_{\beta\eta}}$. If $\boldsymbol{P}$ is fair then so is $\boldsymbol{P}{\downarrow_{\beta\eta}}$, and replacing eigenterms by variables preserves that property, hence $\boldsymbol{Q}$ is fair. □

This means that the presence of eigenterms has no influence on provability of standard sequents in LK, as long as the terms are in $\mathcal{K}$. There only remains to translate valid $\mathcal{K}$-LK$^{\mathrm{c}}$-proofs into $\mathcal{K}$-LK-proofs. This requires to replace eigenterms not simply by foreign variables as in Lemma 19 but by eigenvariables of the LK-proof, a task that validates the use of eigenterms.

Lemma 21. *For any valid $\mathcal{K}$-LK$^{\mathrm{c}}$-proof $\boldsymbol{P} : \Gamma \vdash \Delta \mid \mathcal{X}$ there exists a $\mathcal{K}$-LK-proof $\boldsymbol{Q} : \Gamma \vdash \Delta$ of the same length as $\boldsymbol{P}$. If $\boldsymbol{P}$ is fair then so is $\boldsymbol{Q}$.*

Proof. By induction on $\boldsymbol{P}$. If

$$\boldsymbol{P} = \frac{}{\vdash \phi \mid \phi \doteq t \simeq t}(\mathrm{Ref}^{\mathrm{c}})$$

and this constraint is valid, then $\phi \equiv_{\alpha\beta\eta} t \simeq t$ and $t \in \mathcal{K}$ (resp. $t \in \mathcal{K} \cap \mathcal{F}$ if $\boldsymbol{P}$ is fair), so that there is an obvious (resp. fair) $\mathcal{K}$-LK-proof of $\vdash \phi$.

If

$$\boldsymbol{P} = \frac{\boldsymbol{P}_1 : \Gamma, \phi \vdash \Delta \mid \mathcal{X}}{\Gamma, \chi \vdash \Delta \mid \mathcal{X} \curlywedge \chi \doteq \exists\xi \curlywedge \phi \doteq (\xi\ (\iota\,\Gamma\Delta\chi))}(\exists\text{-L}^{\mathrm{c}})$$

and this constraint is valid, hence so is $\mathcal{X}$ and $\chi \equiv_{\alpha\beta\eta} \exists\xi$, $\phi \equiv_{\alpha\beta\eta} (\xi\ (\iota\,\Gamma\Delta\chi))$. By induction hypothesis there is a $\mathcal{K}$-LK-proof $\boldsymbol{Q}_1 : \Gamma, \phi \vdash \Delta$. Let Γ', ξ', ϕ', Δ' and $\boldsymbol{Q}'_1$ be the respective

$\beta\eta$-normal forms of Γ, ξ, ϕ, Δ and $\boldsymbol{Q}_1$, so that $\boldsymbol{Q}'_1 : \Gamma', \phi' \vdash \Delta'$ is a $\mathcal{K}$-LK-proof. Let $t = (\iota\, \Gamma' \Delta' \exists \xi')$ be the $\beta\eta$-normal form of $(\iota\, \Gamma \Delta \chi)$, so that $\phi' \equiv_{\alpha\beta\eta} (\xi'\ t)$, and let $y : \mathbf{1}$ be a variable not occurring in $\boldsymbol{Q}'_1$ or ξ'. Since ξ' and the members of Γ' and Δ' are strict subterms of t, then t cannot be a subterm of ξ' or of any member of Γ' or Δ', hence $\xi'[y/t] = \xi'$, $\Gamma'[y/t] = \Gamma'$ and $\Delta'[y/t] = \Delta'$. We finally let

$$\boldsymbol{Q} = \frac{\boldsymbol{Q}'_1[y/t] : \Gamma', \phi'[y/t]' \vdash \Delta'}{\Gamma, \chi \vdash \Delta}(\exists\text{-L}).$$

By Lemma 19 $\boldsymbol{Q}'_1[y/t]$ is a $\mathcal{K}$-LK-proof, and by Corollary 18 $\phi'[y/t] \equiv_{\alpha\beta\eta} (\xi'\ t)[y/t] = (\xi'\ y)$. Since $\Gamma \equiv_{\alpha\beta\eta} \Gamma'$, $\chi \equiv_{\alpha\beta\eta} \exists \xi'$, $\Delta' \equiv_{\alpha\beta\eta} \Delta$ and y is not free in Γ', Δ' (since they occur in $\boldsymbol{Q}'_1$) or ξ', then this last inference is correct and therefore $\boldsymbol{Q}$ is a $\mathcal{K}$-LK-proof.

If $\boldsymbol{P}$ is fair then so are Γ, ξ, ϕ, Δ and $\boldsymbol{P}_1$, hence so is $\boldsymbol{Q}_1$ by induction hypothesis, as are their normal forms Γ', ξ', ϕ', Δ' and $\boldsymbol{Q}'_1$. Therefore $\boldsymbol{Q}'_1[y/t]$ and $\boldsymbol{Q}$ are fair.

All other rules can be treated similarly. □

Theorem 22. *For any valid $\mathcal{K}$-LK$^{\mathrm{c}}$-proof $\boldsymbol{P} : \Gamma \vdash \Delta \mid \mathcal{X}$ there exists a standard* LK*-proof $\boldsymbol{Q} : \Gamma' \vdash \Delta'$ of the same length as $\boldsymbol{P}$ such that $\Gamma' \vdash \Delta' \preceq \Gamma{\downarrow_{\beta\eta}} \vdash \Delta{\downarrow_{\beta\eta}}$. If $\boldsymbol{P}$ is fair then so is $\boldsymbol{Q}$.*

Proof. By Lemma 21 and 20. □

Together with Theorem 15, this shows that standard LK-proofs correspond to valid $\mathcal{K}$-LK$^{\mathrm{c}}$-proofs.

6 Abstract Proofs and Proof Lifting in LK$^{\mathrm{c}}$

We now prove a result in LK$^{\mathrm{c}}$ that is very similar to the well-known lifting lemma in resolution calculus: a proof of an instance is an instance of a "lifted" proof. One difference is that the initial proof does not need to be ground (propositional) and may involve quantifier inference rules. Another difference is that the lifted proof can use as many distinct variables as it may contain, i.e., up to a maximum defined below as a class of LK$^{\mathrm{c}}$-proofs, called *abstract*, in which, intuitively, all meta-variables are associated with distinct variables.

In the following, we treat sequences of pairwise distinct elements as sets, and using a sequence as a set always means that its elements are assumed to be pairwise distinct.

Definition 23. Let $n \in \mathbb{N}$ and

$$\boldsymbol{P} = \frac{\boldsymbol{P}_1 : \Gamma_1 \vdash \Delta_1 \mid \mathcal{X}_1 \ \cdots \ \boldsymbol{P}_n : \Gamma_n \vdash \Delta_n \mid \mathcal{X}_n}{\Gamma \vdash \Delta \mid \mathcal{X}_1 \curlywedge \cdots \curlywedge \mathcal{X}_n \curlywedge \mathcal{X}}(\mathrm{R})$$

be some LK$^{\mathrm{c}}$-proof, then $\boldsymbol{P}$ is an *abstract* proof if the following conditions hold:

- the subproofs $\boldsymbol{P}_i$ are abstract for all $1 \leq i \leq n$,
- (*local variables condition*, or *lvc*) the last inference is drawn by instantiating the meta-variables[4] of rule (R) by variables or sets of variables, and distinct meta-variables are instantiated by distinct variables or disjoint sets of variables (see Example 24 below),
- (*split variables condition*, or *svc*) $\mathrm{FV}(\boldsymbol{P}_i) \cap \mathrm{FV}(\boldsymbol{P}_j) = \emptyset$ for all $1 \leq i < j \leq n$, i.e., distinct subproofs have disjoint variables,

[4] note that the $\mathcal{X}_i$'s are not part of these meta-variables.

- (*constraint variables condition*, or *cvc*) $\mathrm{FV}(\mathcal{X}) \cap \mathrm{FV}(\boldsymbol{P}_i) \subseteq \Gamma_i \cup \Delta_i$ for all $1 \leq i \leq n$, i.e., the constraint $\mathcal{X}$ only applies to variables in each subproof that are local to this last inference.

Example 24.

$$\frac{\boldsymbol{P}_1 : \Gamma \vdash \Delta, u \mid \mathcal{X}_1 \quad \boldsymbol{P}_2 : \Sigma, v \vdash \Pi \mid \mathcal{X}_1}{\Gamma, \Sigma \vdash \Delta, \Pi \mid \mathcal{X}_1 \curlywedge \mathcal{X}_2 \curlywedge u \doteq v}(\mathrm{Cut^c})$$

is abstract if

- $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ are abstract,
- (lvc) u, v are distinct variables, Γ, Σ, Δ and Π are sequences of distinct variables and have no variable in common or equal to u or v.
- (svc) $\mathrm{FV}(\boldsymbol{P}_1) \cap \mathrm{FV}(\boldsymbol{P}_2) = \emptyset$,
- and (cvc) $\{u, v\} \cap \mathrm{FV}(\boldsymbol{P}_1) \subseteq \Gamma \cup \Delta \cup \{u\}$, i.e., $v \notin \mathrm{FV}(\boldsymbol{P}_1)$ and similarly $u \notin \mathrm{FV}(\boldsymbol{P}_2)$.

Here the cvc is a consequence of the svc, but this is not the case for (most) unary rules.

Note that the lvc entails that abstract proofs are fair $\mathcal{K}$-LK$^{\mathrm{c}}$-proofs. We now prove that every LK$^{\mathrm{c}}$-proof can be lifted to a more general abstract proof. The proof is constructive, i.e., it provides an algorithm for computing such an abstract proof together with a substitution that instantiates this abstract proof to the original one. Furthermore, this algorithm is deterministic (up to a renaming of the variables that it introduces).

Theorem 25 (proof lifting)**.** *For any* LK$^{\mathrm{c}}$*-proof* $\boldsymbol{P} : \Gamma \vdash \Delta \mid \mathcal{X}$*, for any* Γ', Δ', σ *and finite set* V *of variables such that*

$$\begin{cases} \Gamma'\sigma = \Gamma \text{ and } \Delta'\sigma = \Delta, \\ \mathrm{FV}(\Gamma', \Delta') \subseteq V, \end{cases}$$

there exist an LK$^{\mathrm{c}}$*-proof* $\boldsymbol{Q} : \Gamma' \vdash \Delta' \mid \mathcal{X}'$ *and a substitution* θ *such that*

1. θ *is equal to* σ *on* V*,*
2. $V \cap (\mathrm{FV}(\boldsymbol{Q}) \setminus \mathrm{FV}(\Gamma', \Delta')) = \emptyset$*,*
3. $\boldsymbol{Q}\theta = \boldsymbol{P}$*,*
4. *if* Γ' *and* Δ' *are disjoint sets of variables, then* $\boldsymbol{Q}$ *is abstract.*

Proof. By induction on $\boldsymbol{P}$. We examine in detail only three cases: one axiom, one unary and one binary inference rules.

1. We first assume
$$\boldsymbol{P} = \frac{}{\vdash \phi \mid \phi \doteq t \simeq t}(\mathrm{Ref^c})$$
and that ϕ', σ, V are such that $\mathrm{FV}(\phi') \subseteq V$ and $\phi'\sigma = \phi$. Let x be a variable of type $\mathbf{1}$ that does not belong to V, which always exists since V is finite. Let $\theta = \sigma[x \mapsto t]$ and
$$\boldsymbol{Q} = \frac{}{\vdash \phi' \mid \phi' \doteq x \simeq x}(\mathrm{Ref^c}),$$
then $\mathrm{FV}(\boldsymbol{Q}) = \mathrm{FV}(\phi') \uplus \{x\}$, so that
$$V \cap (\mathrm{FV}(\boldsymbol{Q}) \setminus \mathrm{FV}(\phi')) = V \cap \{x\} = \emptyset.$$
It is easy to check that $\boldsymbol{Q}\theta = \boldsymbol{P}$.
If ϕ' is a variable then the lvc holds and $\boldsymbol{Q}$ is thus abstract (the svc and cvc are trivially valid since there is no subproof).

2. Next, we assume that

$$\boldsymbol{P} = \frac{\boldsymbol{P}_1 : \Gamma, \phi \vdash \Delta \mid \mathcal{X}}{\Gamma, \chi \vdash \Delta \mid \mathcal{X} \curlywedge \chi \doteq \exists \xi \curlywedge \phi \doteq (\xi\ \iota \Gamma \Delta \chi)} (\exists\text{-L}^{\mathrm{c}})$$

and that $\Gamma', \chi', \Delta', \sigma, V$ are such that $\mathrm{FV}(\Gamma', \chi', \Delta') \subseteq V$, $\Gamma'\sigma = \Gamma$, $\chi'\sigma = \chi$ and $\Delta'\sigma = \Delta$. Let $u : \mathbf{o}$ and $f : \mathbf{\imath} \to \mathbf{o}$ be variables that do not belong to V, $V' = V \uplus \{u, f\}$ and $\sigma' = \sigma[u \mapsto \phi, f \mapsto \xi]$. Obviously $\mathrm{FV}(\Gamma', u, \Delta') \subseteq V'$, $\Gamma'\sigma' = \Gamma$, $u\sigma' = \phi$ and $\Delta'\sigma' = \Delta$, hence we obtain by the induction hypothesis on $\boldsymbol{P}_1$ that there exist a proof $\boldsymbol{Q}_1 : \Gamma', u \vdash \Delta' \mid \mathcal{X}'$ and a substitution θ equal to σ' on V' such that $V' \cap (\mathrm{FV}(\boldsymbol{Q}_1) \setminus \mathrm{FV}(\Gamma', u, \Delta')) = \emptyset$ and $\boldsymbol{Q}_1\theta = \boldsymbol{P}_1$ (in particular $\mathcal{X}'\theta = \mathcal{X}$). Let

$$\boldsymbol{Q} = \frac{\boldsymbol{Q}_1 : \Gamma', u \vdash \Delta' \mid \mathcal{X}'}{\Gamma', \chi' \vdash \Delta' \mid \mathcal{X}' \curlywedge \chi' \doteq \exists f \curlywedge u \doteq (f\ (\iota\, \Gamma' \Delta' \chi'))} (\exists\text{-L}^{\mathrm{c}}).$$

Obviously $\chi'\theta = \chi'\sigma = \chi$, $u\theta = \phi$ and $f\theta = \xi$, hence $\boldsymbol{Q}\theta = \boldsymbol{P}$. Since $\mathrm{FV}(\boldsymbol{Q}) = \mathrm{FV}(\boldsymbol{Q}_1, \chi', f)$ then

$$\begin{aligned} V \cap (\mathrm{FV}(\boldsymbol{Q}) \setminus \mathrm{FV}(\Gamma', \chi', \Delta')) &= V \cap (\mathrm{FV}(\boldsymbol{Q}_1, f) \setminus \mathrm{FV}(\Gamma', \Delta')) \\ &= V' \cap (\mathrm{FV}(\boldsymbol{Q}_1, f) \setminus \mathrm{FV}(\Gamma', \Delta', u, f)) = \emptyset. \end{aligned}$$

If $\Gamma'\chi'$ and Δ' are disjoint sets of variables, by definition $u \notin \Gamma' \cup \Delta'$, hence $\Gamma' u$ and Δ' are disjoint sets of variables and by induction hypothesis $\boldsymbol{Q}_1$ must be abstract. Obviously the lvc holds and the svc is trivial, hence there remains to check that $(\Gamma' \cup \Delta' \cup \{\chi', f, u\}) \cap \mathrm{FV}(\boldsymbol{Q}_1) \subseteq \Gamma' \cup \{u\} \cup \Delta'$, i.e., $\chi', f \notin \mathrm{FV}(\boldsymbol{Q}_1)$, which is true since $\chi', f \in V_1$. Hence the cvc condition holds and $\boldsymbol{Q}$ is abstract.

3. Finally, we assume that

$$\boldsymbol{P} = \frac{\boldsymbol{P}_1 : \Gamma \vdash \Delta, \phi \mid \mathcal{X}_1 \quad \boldsymbol{P}_2 : \Sigma, \psi \vdash \Pi \mid \mathcal{X}_2}{\Gamma, \Sigma \vdash \Delta, \Pi \mid \mathcal{X}_1 \curlywedge \mathcal{X}_2 \curlywedge \phi \doteq \psi} (\mathrm{Cut}^{\mathrm{c}})$$

with $\Gamma', \Sigma', \Delta', \Pi', \sigma, V$ such that $\mathrm{FV}(\Gamma', \Sigma', \Delta', \Pi') \subseteq V$, $\Gamma'\sigma = \Gamma$, $\Sigma'\sigma = \Sigma$, $\Delta'\sigma = \Delta$ and $\Pi'\sigma = \Pi$. Let $u : \mathbf{o}$ and $v : \mathbf{o}$ be distinct variables that do not belong to V, $V_1 = V \uplus \{u, v\}$ and $\sigma' = \sigma[u \mapsto \phi, v \mapsto \psi]$. Obviously $\mathrm{FV}(\Gamma', \Delta', u) \subseteq V_1$, $\Gamma'\sigma' = \Gamma$, $u\sigma' = \phi$ and $\Delta'\sigma' = \Delta$, hence we obtain by the induction hypothesis on $\boldsymbol{P}_1$ that there exist a proof $\boldsymbol{Q}_1 : \Gamma' \vdash \Delta', u \mid \mathcal{X}'_1$ and θ' equal to σ' on V_1 such that $V_1 \cap (\mathrm{FV}(\boldsymbol{Q}_1) \setminus \mathrm{FV}(\Gamma', \Delta', u)) = \emptyset$ and $\boldsymbol{Q}_1\theta' = \boldsymbol{P}_1$.

As above $\Delta'\theta' = \Delta$, $v\theta' = \psi$ and $\Pi'\theta' = \Pi$, hence with $V_2 = V_1 \cup \mathrm{FV}(Q_1)$, then $\mathrm{FV}(\Delta', \Pi', v) \subseteq V_2$ and we obtain by the induction hypothesis on $\boldsymbol{P}_2$ that there exist a proof $\boldsymbol{Q}_2 : \Sigma', v \vdash \Pi' \mid \mathcal{X}'_2$ and θ equal to θ' on V_2 (hence to σ on V, and $u\theta = \phi$, $v\theta = \psi$) such that $V_2 \cap (\mathrm{FV}(\boldsymbol{Q}_2) \setminus \mathrm{FV}(\Sigma', v, \Pi')) = \emptyset$ and $\boldsymbol{Q}_2\theta = \boldsymbol{P}_2$ (in particular $\mathcal{X}'_2\theta = \mathcal{X}_2$). Let

$$\boldsymbol{Q} = \frac{\boldsymbol{Q}_1 : \Gamma' \vdash \Delta', u \mid \mathcal{X}'_1 \quad \boldsymbol{Q}_2 : \Sigma', v \vdash \Pi' \mid \mathcal{X}'_2}{\Gamma', \Sigma' \vdash \Delta', \Pi' \mid \mathcal{X}'_1 \curlywedge \mathcal{X}'_2 \curlywedge u \doteq v} (\mathrm{Cut}^{\mathrm{c}}).$$

Since $\mathrm{FV}(\boldsymbol{Q}_1) \subseteq V_2$ then $\boldsymbol{Q}_1\theta = \boldsymbol{Q}_1\theta' = \boldsymbol{P}_1$, so that $\Gamma'\theta = \Gamma$, $\Delta'\theta = \Delta$ and $\mathcal{X}'_1\theta = \mathcal{X}_1$, hence $\boldsymbol{Q}\theta = \boldsymbol{P}$. Furthermore

$$\begin{array}{rcl} V \cap (\mathrm{FV}(\boldsymbol{Q}) \setminus \mathrm{FV}(\Gamma', \Sigma', \Delta', \Pi')) & & \\ & = & V \cap (\mathrm{FV}(\boldsymbol{Q}_1, \boldsymbol{Q}_2) \setminus \mathrm{FV}(\Gamma', \Sigma', \Delta', \Pi')) \\ & = & V_2 \cap (\mathrm{FV}(\boldsymbol{Q}_1, \boldsymbol{Q}_2) \setminus \mathrm{FV}(\Gamma', \Sigma', \Delta', \Pi', u, v, \boldsymbol{Q}_1)) \\ & = & V_2 \cap (\mathrm{FV}(\boldsymbol{Q}_2) \setminus \mathrm{FV}(\Gamma', \Sigma', \Delta', \Pi', u, v)) = \emptyset. \end{array}$$

If the sequences $\Gamma'\Sigma'$ and $\Delta'\Pi'$ are disjoint sets of variables, then so are Γ', Σ', Δ' and Π'; and by definition of u and v it is trivial that the lvc holds on $\boldsymbol{Q}$. Besides, Γ' and $\Delta' \uplus \{u\}$ are disjoint sets of variables, and similarly $\Sigma' \uplus \{v\}$ and Π', hence by induction hypothesis $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_2$ are abstract proofs. It is obvious that $\{u, v\} \cap \mathrm{FV}(\boldsymbol{Q}_1) \subseteq \Gamma' \cup \Delta' \cup \{u\}$ since $v \in V_1$ hence $v \notin \mathrm{FV}(\boldsymbol{Q}_1)$. Similarly $\{u, v\} \cap \mathrm{FV}(\boldsymbol{Q}_2) \subseteq \Sigma' \cup \{v\} \cup \Pi'$ holds since $u \in V_2$ hence $u \notin \mathrm{FV}(\boldsymbol{Q}_2)$, so that the cvc holds.

Finally, since $\mathrm{FV}(\boldsymbol{Q}_1) \subseteq V_2$, then $\mathrm{FV}(\boldsymbol{Q}_1) \cap (\mathrm{FV}(\boldsymbol{Q}_2) \setminus \Sigma' \cup \{v\} \cup \Pi') = \emptyset$, hence $\mathrm{FV}(\boldsymbol{Q}_1) \cap \mathrm{FV}(\boldsymbol{Q}_2) \subseteq \Sigma' \cup \{v\} \cup \Pi'$. But $v \notin \mathrm{FV}(\boldsymbol{Q}_1)$, and since $\Sigma' \cup \Pi' \subseteq V_1$, then $(\Sigma' \cup \Pi') \cap (\mathrm{FV}(\boldsymbol{Q}_1) \setminus \Gamma' \cup \Delta' \cup \{u\}) = \emptyset$, hence $(\Sigma' \cup \Pi') \cap \mathrm{FV}(\boldsymbol{Q}_1) = \emptyset$ (by the lvc), and therefore $\mathrm{FV}(\boldsymbol{Q}_1) \cap \mathrm{FV}(\boldsymbol{Q}_2) = \emptyset$. This establishes the svc and hence that $\boldsymbol{Q}$ is abstract.

□

Corollary 26. *Under the conditions of Theorem 25 and if furthermore $\boldsymbol{P}$ is valid then θ is a solution of $\mathcal{X}'$ and for every solution $\mu \precsim \theta$ of $\mathcal{X}'$, $\boldsymbol{Q}\mu$ is a valid proof more general than $\boldsymbol{P}$.*

Example 27. The following is an abstract LK^{c}-proof $\boldsymbol{Q}^{\mathrm{a}}$ obtained by lifting the proof $\boldsymbol{Q}$ of Example 5 with the end sequent $\phi_1 \vdash \phi_2$, the substitution $\sigma = [\forall x\,(p\ x\ a)/\phi_1,\ \exists y\,(p\ y\ a)/\phi_2]$ and $V = \{\phi_1, \phi_2\}$. Symbols z_1, z_2 are variables of type $\boldsymbol{\iota}$, $\phi_1, \phi_2, \phi_3, \phi_4$ are variables of type $\mathbf{o}$ and ξ_1, ξ_2 are variables of type $\boldsymbol{\iota} \to \mathbf{o}$.

$$\dfrac{\dfrac{\dfrac{}{\phi_4 \vdash \phi_3 \mid \mathcal{Y}_1}\,(\mathrm{Axiom}^{\mathrm{c}})}{\phi_1 \vdash \phi_3 \mid \mathcal{Y}_2}\,(\forall\text{-L}^{\mathrm{c}})}{\phi_1 \vdash \phi_2 \mid \mathcal{Y}_3}\,(\exists\text{-R}^{\mathrm{c}}) \qquad \text{where} \quad \begin{array}{rcl} \mathcal{Y}_1 & \stackrel{\mathrm{def}}{=} & \phi_4 \doteq \phi_3, \\ \mathcal{Y}_2 & \stackrel{\mathrm{def}}{=} & \mathcal{Y}_1 \curlywedge \phi_1 \doteq \forall \xi_2 \curlywedge \phi_4 \doteq (\xi_2\ z_2), \\ \mathcal{Y}_3 & \stackrel{\mathrm{def}}{=} & \mathcal{Y}_2 \curlywedge \phi_2 \doteq \exists \xi_1 \curlywedge \phi_3 \doteq (\xi_1\ z_1). \end{array}$$

It is easy to check that the proof $\boldsymbol{Q}$ of Example 3 is an instance of this proof, corresponding to the substitution:

$$\theta \stackrel{\mathrm{def}}{=} [\lambda y\,(p\ y\ a)/\xi_1,\ (p\ v\ a)/\phi_3,\ v/z_1,\ \xi/\xi_2,\ \psi/\phi_4,\ u/z_2]\sigma,$$

and in particular $\mathcal{Y}_3\theta = \mathcal{X}_3$.

Corollary 28. *Under the conditions of Theorem 25 and assuming that $\boldsymbol{Q}$ is abstract then any sequent $\Gamma'' \vdash \Delta''$ has a valid LK^{c}-proof less general than $\boldsymbol{Q}$ iff there is a solution θ of $\mathcal{X}'$ such that $(\Gamma' \vdash \Delta')\theta = \Gamma'' \vdash \Delta''$.*

This is a translation of Lemma A in [21] where the notion of proof skeleton (or proof analysis) has been replaced by that of abstract proof. This suggests that an abstract proof is nothing else than a proof skeleton, but this is not the case. More precisely, a *proof skeleton* is what remains of a proof when all formulæ have been removed: a tree labelled by inference rules. We can thus see it as a term in the signature of inference rules, e.g., the skeleton of the proof in Example 27 is $\exists\text{-R}^{\mathrm{c}}(\forall\text{-L}^{\mathrm{c}}(\mathrm{Axiom}^{\mathrm{c}}))$. But contrary to the schematic systems of [21] it is not generally possible to build the abstract constraint $\mathcal{X}'$ from the

proof skeleton. One problematic rule is (P-L$^{\mathrm{c}}$), because in order to generate a constraint on formulæ ϕ and ψ we need to know where exactly they occur in the left part of any given sequent. This is why in [13] an integer is attached to every occurrence of a (P-L) or (P-R) rule in a proof skeleton, in a rather ad-hoc way. This integer can of course be read out from an abstract proof which looks like

$$\frac{x_1, \ldots, x_n \vdash y_1, \ldots, y_m \mid \mathcal{X}_1}{x_1, \ldots, x_{i+1}, x_i, \ldots, x_n \vdash y_1, \ldots, y_m \mid \mathcal{X}_2}(\text{P-L}^{\mathrm{c}})$$

The binary rules we have chosen are also problematic since we need to know where both sides of the end sequent are split between the left and right premiss. Consequently in [2] a pair of integers is added to all occurrences of binary rules[5] in proof skeletons. It is easy to see that a skeleton enriched with this extra information is equivalent to an abstract proof in the sense that both allow to compute the other (including the lengths n and m at each inference). The notion of abstract proof naturally extends to other inference systems and meets the original notion of proof skeleton on schematic systems as illustrated in Section 9.

The next proposition states that an abstract proof is minimal w.r.t. the generalization ordering, i.e., it admits no strictly more general LK$^{\mathrm{c}}$-proof.

Proposition 29. *If $\boldsymbol{P}$, $\boldsymbol{P}^{\mathrm{a}}$ and $\boldsymbol{Q}$ are* LK$^{\mathrm{c}}$*-proofs such that $\boldsymbol{P}^{\mathrm{a}}$ is abstract, $\boldsymbol{P}^{\mathrm{a}} \precsim \boldsymbol{P}$ and $\boldsymbol{Q} \precsim \boldsymbol{P}$, then $\boldsymbol{P}^{\mathrm{a}} \preceq \boldsymbol{Q}$.*

Proof. By induction on $\boldsymbol{P}^{\mathrm{a}}$. There exist σ and μ such that $\boldsymbol{P}^{\mathrm{a}}\sigma \equiv_{\alpha\beta\eta} \boldsymbol{P}$ and $\boldsymbol{Q}\mu \equiv_{\alpha\beta\eta} \boldsymbol{P}$. We examine the details in two cases: one axiom and one binary inference rule.

1. If
$$\boldsymbol{P}^{\mathrm{a}} = \frac{}{\vdash u \mid u \doteq x \simeq x}(\text{Ref}^{\mathrm{c}})$$
where u and x are variables, then
$$\boldsymbol{P} \equiv_{\alpha\beta\eta} \boldsymbol{P}^{\mathrm{a}}\sigma = \frac{}{\vdash u\sigma \mid u\sigma \doteq x\sigma \simeq x\sigma}(\text{Ref}^{\mathrm{c}})$$
and since $\boldsymbol{Q}\mu \equiv_{\alpha\beta\eta} \boldsymbol{P}^{\mathrm{a}}\sigma$ then
$$\boldsymbol{Q} = \frac{}{\vdash \phi \mid \phi \doteq t \simeq t}(\text{Ref}^{\mathrm{c}})$$
where $\phi\mu \equiv_{\alpha\beta\eta} u\sigma$ and $t\mu \equiv_{\alpha\beta\eta} x\sigma$. We now define the substitution $\theta = [\phi/u, t/x]$, then obviously $\boldsymbol{P}^{\mathrm{a}}\theta = \boldsymbol{Q}$.

2. If $\boldsymbol{P}^{\mathrm{a}}$ is the proof (with variables u_i, $u'_{i'}$, v_j, $v'_{j'}$)
$$\frac{\boldsymbol{P}^{\mathrm{a}}_1 : u_1 \cdots u_n \vdash v_1 \cdots v_{m+1} \mid \mathcal{X}^{\mathrm{a}}_1 \quad \boldsymbol{P}^{\mathrm{a}}_2 : u'_1 \cdots u'_{n'+1} \vdash v'_1 \cdots v'_{m'} \mid \mathcal{X}^{\mathrm{a}}_2}{u_1 \cdots u_n u'_1 \cdots u'_{n'} \vdash v_1 \cdots v_m v'_1 \cdots v'_{m'} \mid \mathcal{X}^{\mathrm{a}}_1 \curlywedge \mathcal{X}^{\mathrm{a}}_2 \curlywedge v_{m+1} \doteq u'_{n'+1}}(\text{Cut}^{\mathrm{c}})$$
and since $\boldsymbol{Q}\mu \equiv_{\alpha\beta\eta} \boldsymbol{P}^{\mathrm{a}}\sigma$ then $\boldsymbol{Q}$ must be a proof
$$\frac{\boldsymbol{Q}_1 : \phi_1 \cdots \phi_n \vdash \psi_1 \cdots \psi_{m+1} \mid \mathcal{X}_1 \quad \boldsymbol{Q}_2 : \phi'_1 \cdots \phi'_{n'+1} \vdash \psi'_1 \cdots \psi'_{m'} \mid \mathcal{X}_2}{\phi_1 \cdots \phi_n \phi'_1 \cdots \phi'_{n'} \vdash \psi_1 \cdots \psi_m \psi'_1 \cdots \psi'_{m'} \mid \mathcal{X}_1 \curlywedge \mathcal{X}_2 \curlywedge \psi_{m+1} \doteq \phi'_{n'+1}}(\text{Cut}^{\mathrm{c}})$$
where $\boldsymbol{Q}_i\mu \equiv_{\alpha\beta\eta} \boldsymbol{P}^{\mathrm{a}}_i\sigma$ for $i = 1, 2$. Since $\boldsymbol{P}^{\mathrm{a}}_1$ and $\boldsymbol{P}^{\mathrm{a}}_2$ are abstract, by induction hypothesis there exist θ_1 and θ_2 such that $\boldsymbol{P}^{\mathrm{a}}_i\theta_i = \boldsymbol{Q}_i$ for $i = 1, 2$, and we may assume that $\mathrm{Dom}(\theta_i) \subseteq \mathrm{FV}(\boldsymbol{P}^{\mathrm{a}}_i)$. By the svc we deduce that $\mathrm{Dom}(\theta_1) \cap \mathrm{Dom}(\theta_2) = \emptyset$, hence we let θ be the substitution equal to θ_1 on $\mathrm{Dom}(\theta_1)$ and to θ_2 elsewhere, so that $\mathcal{X}^{\mathrm{a}}_1\theta = \mathcal{X}^{\mathrm{a}}_1\theta_1 = \mathcal{X}_1$, $\mathcal{X}^{\mathrm{a}}_2\theta = \mathcal{X}^{\mathrm{a}}_2\theta_2 = \mathcal{X}_2$, $v_{m+1}\theta = v_{m+1}\theta_1 = \psi_{m+1}$, $u'_{n'+1}\theta = u'_{n'+1}\theta_2 = \phi'_{n'+1}$, and similarly for the u_i's, $u'_{i'}$'s, v_j's and $v'_{j'}$'s, which finally yields $\boldsymbol{P}^{\mathrm{a}}\theta = \boldsymbol{Q}$.

[5] In all rigor the same should be done in [13] for the ($\Rightarrow$-L) rule from [25].

□

Note that this implies that, for any LK$^{\mathrm{c}}$-proofs $\boldsymbol{P}$ and $\boldsymbol{P}^{\mathrm{a}}$, if $\boldsymbol{P}^{\mathrm{a}}$ is abstract then $\boldsymbol{P}^{\mathrm{a}} \precsim \boldsymbol{P}$ is equivalent to $\boldsymbol{P}^{\mathrm{a}} \preceq \boldsymbol{P}$ (by taking $\boldsymbol{Q} = \boldsymbol{P}$). This theorem thus shows that the abstract proof $\boldsymbol{P}^{\mathrm{a}}$ obtained by lifting $\boldsymbol{P}$ is most general among the generalizations of $\boldsymbol{P}$. However, this is still an LK$^{\mathrm{c}}$-proof and our aim is to generalize LK-proofs, hence to obtain valid generalizations of valid LK$^{\mathrm{c}}$-proofs.

Proposition 30. *If $\boldsymbol{P}^{\mathrm{a}}$ is abstract, $\boldsymbol{P}$ and $\boldsymbol{Q}$ are valid, $\boldsymbol{P}^{\mathrm{a}} \preceq \boldsymbol{P}$, $\boldsymbol{Q} \precsim \boldsymbol{P}$ and σ is a most general solution of $\boldsymbol{P}^{\mathrm{a}}$'s constraint, then $\boldsymbol{P}^{\mathrm{a}}\sigma \precsim \boldsymbol{Q}$.*

Proof. By Proposition 29 there exists a substitution θ such that $\boldsymbol{P}^{\mathrm{a}}\theta = \boldsymbol{Q}$, and since $\boldsymbol{Q}$ is valid then θ is a solution of $\boldsymbol{P}^{\mathrm{a}}$'s constraint, hence there is a substitution ρ such that $\sigma\rho \equiv_{\alpha\beta\eta} \theta$, and therefore $\boldsymbol{P}^{\mathrm{a}}\sigma\rho \equiv_{\alpha\beta\eta} \boldsymbol{Q}$, that is $\boldsymbol{P}^{\mathrm{a}}\sigma \precsim \boldsymbol{Q}$. □

This means that in this case $\boldsymbol{P}^{\mathrm{a}}\sigma$ is most general among the valid generalizations of $\boldsymbol{P}$. However, there may not be a most general unifier of the abstract constraint since it may have second order variables. But we may still be able to find minimal unifiers and we can show that this property extends to the corresponding valid proofs.

Proposition 31. *If $\boldsymbol{P}^{\mathrm{a}}$ is abstract and σ is a minimal solution of $\boldsymbol{P}^{\mathrm{a}}$'s constraint then $\boldsymbol{P}^{\mathrm{a}}\sigma$ is a minimal valid proof.*

Proof. We may assume that $\mathrm{Dom}(\sigma) \subseteq \mathrm{FV}(\boldsymbol{P}^{\mathrm{a}})$. Suppose that there is a valid proof $\boldsymbol{Q} \precsim \boldsymbol{P}^{\mathrm{a}}\sigma$, then by Proposition 29 there is a θ such that $\boldsymbol{P}^{\mathrm{a}}\theta = \boldsymbol{Q}$ and we may assume that $\mathrm{Dom}(\theta) \subseteq \mathrm{FV}(\boldsymbol{P}^{\mathrm{a}})$. Hence $\boldsymbol{P}^{\mathrm{a}}\theta \precsim \boldsymbol{P}^{\mathrm{a}}\sigma$ and it is then easy to see that $\theta \precsim \sigma$. But θ is a solution of the abstract constraint (since $\boldsymbol{P}^{\mathrm{a}}\theta$ is valid), hence $\theta \equiv_{\alpha\beta\eta} \sigma$ by minimality of σ and therefore $\boldsymbol{Q} \equiv_{\alpha\beta\eta} \boldsymbol{P}^{\mathrm{a}}\sigma$. □

7 The Unifier Minimization Algorithm

Lifting a valid $\mathcal{K}$-LK$^{\mathrm{c}}$-proof $\boldsymbol{P}$ to an abstract proof $\boldsymbol{P}^{\mathrm{a}}$ yields a substitution θ such that $\boldsymbol{P}^{\mathrm{a}}\theta = \boldsymbol{P}$, hence θ is a solution of the constraint $\mathcal{X}$ of $\boldsymbol{P}^{\mathrm{a}}$ (which may not be valid) and is a $\mathcal{K}$-substitution. As explained in Section 1, we now need an algorithm to compute, given $\mathcal{X}$ and θ, a solution σ of $\mathcal{X}$ such that $\sigma \precsim \theta$, which according to Corollary 26 yields a valid proof $\boldsymbol{P}^{\mathrm{a}}\sigma \precsim \boldsymbol{P}$. Such a substitution always exists since θ fulfills the desired property; our goal is to find a solution that is as general as possible. In the present section we define and illustrate such an algorithm, and also prove a few lemmas to be used in Section 8 devoted to the basic properties of this algorithm. We first introduce the notion of a controlled unification problem, which is a second order unification problem (in the standard sense) associated with a particular solution:

Definition 32. A *unification problem* is a pair $(\mathcal{X}, \sigma)$ where:

- $\mathcal{X}$ is a $\mathcal{K}$-constraint.
- σ is an idempotent $\mathcal{K}$-substitution such that $\mathrm{Dom}(\sigma) \cap \mathrm{FV}(\mathcal{X}) = \emptyset$.

A $\mathcal{K}$-substitution γ is a *solution* of $(\mathcal{X}, \sigma)$ if $\sigma \precsim \gamma$ and γ is a solution of $\mathcal{X}$. Note that σ is a solution of $(\top, \sigma)$. The set of solutions of $(\mathcal{X}, \sigma)$ is denoted by $\mathrm{sol}(\mathcal{X}, \sigma)$.

A *controlled unification problem* is a triple $(\mathcal{X}, \sigma, \theta)$, where $(\mathcal{X}, \sigma)$ is a unification problem and θ is a solution of $(\mathcal{X}, \sigma)$. A substitution γ is a *solution* of $(\mathcal{X}, \sigma, \theta)$ if it is a solution of $(\mathcal{X}, \sigma)$ such that $\gamma \precsim \theta$. $(\mathcal{X}, \sigma, \theta)$ is *fair* if both $\mathcal{X}$ and σ are fair. The triple $(\mathcal{X}, \sigma, \theta)$ is *solved* if $\mathcal{X} = \top$ (and then σ is a solution).

The algorithm presented in this section consists in rewriting controlled unification problems $(\mathcal{X}, \sigma, \theta)$ by applying second order unification rules. The algorithm starts with $\sigma = id$ and instantiates it by solving equations in $\mathcal{X}$. It is sometimes necessary to introduce new variables in $\mathcal{X}$ and to extend θ accordingly. Since second order unification is undecidable, we use a specific strategy to preserve θ as a solution of $\mathcal{X}$, which ensures that all the obtained solutions will be more general than θ. Furthermore, we show in Section 8 that the unification rules eventually decrease the size of the image by θ of the set of variables occurring in the considered problem. This allows us to propose a terminating algorithm but not to guarantee that a minimal unifier is found. Yet we will ensure an elementary form of generality, i.e., that the algorithm always reaches a fair solution (even when θ is not fair, i.e., when it is "committed" to a particular signature).

As usual in this kind of algorithm (see [5]), we use slightly different representatives of $\beta\eta$-classes than the $\beta\eta$-normal forms. Obviously, any β-normal term can be uniquely written in the form $\lambda x_1 \cdots \lambda x_n \, (v \; t_1 \cdots t_m)$ where $n \geq 0$, $m \geq 0$, the t_i's are β-normal terms and v is either a β-normal eigenterm (in which case it has type $\mathbf{1}$ and hence $m = 0$) or $v \in \mathcal{V} \cup \mathcal{C}$. We now transform this term according to its type.

Definition 33. If $t = \lambda x_1 \cdots \lambda x_n \, (v \; t_1 \cdots t_m)$ is a term in β-normal form, with $n, m \geq 0$ and has type $\mathbf{1}^p \to \tau$ where $v \in \mathcal{V} \cup \mathcal{C}$ and τ is basic (hence $p \geq n$), let

$$t\downarrow \stackrel{\text{def}}{=} \lambda x_1 \cdots \lambda x_p (v \; t_1\downarrow \cdots t_m\downarrow \, x_{n+1} \cdots x_p)$$

where $x_{n+1}, \ldots, x_p$ are variables that do not occur in t.

If $t = \lambda x_1 \cdots \lambda x_n \, (\iota \;\; t_1 \cdots t_m)$ is a term in β-normal form with $n \geq 0, m > 0$, let

$$t\downarrow \stackrel{\text{def}}{=} \lambda x_1 \cdots \lambda x_n (\iota \, t_1\downarrow \cdots t_m\downarrow).$$

For any term t we write $t\downarrow \stackrel{\text{def}}{=} (t{\downarrow_{\beta\eta}})\downarrow$ and call it the *long normal form* (or *lnf*) of t. The *lnf* $\mathcal{X}\downarrow$ of a constraint $\mathcal{X}$ is obtained by transforming to lnf both members of every equation in $\mathcal{X}$.

Note that $t\downarrow \in \mathcal{K}$ whenever $t \in \mathcal{K}$, $t\downarrow \to_\eta{}^* t{\downarrow_{\beta\eta}}$ and $t\downarrow$ is only defined up to α-conversion. It is therefore clear that two terms s and t belong to the same $\alpha\beta\eta$-equivalence class if and only if $t\downarrow$ and $s\downarrow$ are α-equivalent. Furthermore, since free variables and constants are preserved by both η-reduction and its inverse, then $\mathrm{FV}(t\downarrow) \subseteq \mathrm{FV}(t)$ and if a constraint $\mathcal{X}$ is fair then so is $\mathcal{X}\downarrow$.

The following definition shows that every non atomic normal term of some $\mathcal{V}$-type can be written of the form $u\gamma$, where u is of some specific form. The motivation for such a definition is that, when solving a controlled unification problem $(\mathcal{X}, \gamma, \theta)$, one often has to instantiate a variable $x \in \mathrm{FV}(\mathcal{X})$ by a new non atomic term u to enable further decomposition or simplification steps. Since we must keep the solution θ during the solving process, u must be more general than the term $x\theta$, and since we want the solution to be as general as possible u must be $\preceq$-minimal (useless instantiations should be avoided). The next definition shows how u and γ can be computed from $t = x\theta$.

Definition 34. For every $\beta\eta$-reduced term $t \in \mathcal{K} \setminus (\mathcal{V} \cup \mathcal{C})$ of some $\mathcal{V}$-type, $\delta(t)$ denotes the pair (u, γ) defined as follows.

Case	t	$\delta(t) = (u\ ,\ \gamma)$
1	$\lambda x\, x$	$(\lambda x\, x\ ,\ id)$
2	$\lambda x\, a$	$(\lambda x\, y\ ,\ [a/y])$
3	$\lambda x\, (\iota\, \exists y\, (p\ y))$	$(\lambda x\, (\iota\, z)\ ,\ [\exists y\, (p\ y)/z])$
4	$\lambda x\, (\exists y\, (p\ x\ y))$	$(\lambda x\, (\exists (z\ x))\ ,\ [\lambda x \lambda y\, (p\ x\ y)/z])$
5	$\lambda x\, (p\ a\ x)$	$(\lambda x\, (y\ (z_1\ x)\ (z_2\ x))\ ,\ [p/y, \lambda x\, a/z_1, \lambda x\, x/z_2])$

Figure 3: Examples of application of Definition 34

1. If t is of the form $\lambda x_1 \cdots \lambda x_n\, x_i$ where $n \geq 1$ and $1 \leq i \leq n$ then: $(u, \gamma) \stackrel{\text{def}}{=} (t, id)$, where id is the identity substitution.

2. If $t = \lambda x_1 \cdots \lambda x_n\, v$, where $n \geq 1$ and $v \in \mathcal{C} \cup \mathcal{V} \setminus \{x_1, \ldots x_n\}$ then $(u, \gamma) \stackrel{\text{def}}{=} (\lambda x_1 \cdots \lambda x_n\, y, [v/y])$, where y is an arbitrarily chosen variable of the same type as v (which is a $\mathcal{V}$-type) distinct from $x_1, \ldots, x_n, v$.

3. If $t = \lambda x_1 \cdots \lambda x_n\, (\iota\, t_1 \cdots t_m)$ (with $m > 0$ and $n \geq 0$) then:

$$u \stackrel{\text{def}}{=} \lambda x_1 \cdots \lambda x_n\, (\iota\, y_1 \cdots y_m)$$

$$\gamma \stackrel{\text{def}}{=} [t_1/y_1, \ldots, t_m/y_m]$$

   where $y_1, \ldots, y_m$ are arbitrarily chosen distinct variables of type $\mathbf{o}$ not occurring in $\mathrm{FV}(t_1, \ldots, t_m, x_1, \ldots, x_n)$.

4. If $t = \lambda x_1 \cdots \lambda x_n\, (v\ t_1 \cdots t_m)$ with $m > 0$, $n \geq 0$ and $v \in \{\forall, \exists, \neg, \wedge, \vee, \Rightarrow\}$[6] then:

$$u \stackrel{\text{def}}{=} \lambda x_1 \cdots \lambda x_n\, (v\ (z_1\ x_1 \cdots x_n) \cdots (z_m\ x_1 \cdots x_n))$$

$$\gamma \stackrel{\text{def}}{=} [\lambda x_1 \cdots \lambda x_n\, t_1/z_1, \ldots, \lambda x_1 \cdots \lambda x_n\, t_m/z_m]$$

   where $z_1, \ldots, z_m$ are arbitrarily chosen distinct variables of the appropriate types[7], not occurring in $\mathrm{FV}(t_1, \ldots, t_m, x_1, \ldots, x_n)$.

5. If $t = \lambda x_1 \cdots \lambda x_n\, (v\ t_1 \cdots t_m)$ with $m > 0$, $n \geq 0$ and $v \notin \{\forall, \exists, \neg, \wedge, \vee, \Rightarrow\}$ then:

$$u \stackrel{\text{def}}{=} \lambda x_1 \cdots \lambda x_n\, (y\ (z_1\ x_1 \cdots x_n) \cdots (z_m\ x_1 \cdots x_n))$$

$$\gamma \stackrel{\text{def}}{=} [v/y, \lambda x_1 \cdots \lambda x_n\, t_1/z_1, \ldots, \lambda x_1 \cdots \lambda x_n\, t_m/z_m]$$

   where $y, z_1, \ldots, z_m$ are arbitrarily chosen distinct variables of the appropriate types, not occurring in $\mathrm{FV}(v, t_1, \ldots, t_m, x_1, \ldots, x_n)$.

Note that $\delta(t)$ is only defined up to a renaming of variables not free in t. See Figure 3 for an illustrating example of each case. The following propositions state immediate consequences of the definition.

Proposition 35. *The pair $\delta(t)$ is well-defined, for every $\beta\eta$-reduced term $t \in \mathcal{K} \setminus (\mathcal{V} \cup \mathcal{C})$ of some $\mathcal{V}$-type.*

[6]These constants cannot be generalized simply because there is no variable of the corresponding types. The equality predicate could be generalized if its specific properties are not used in the proof, i.e., if no paramodulation inference is applied on it. Of course, if no $\wedge$-rule is applied on a formula $(\wedge\, t_1\, t_2)$ then it can be generalized by a variable of type $\mathbf{o}$.

[7]In particular, if $v \in \{\forall, \exists\}$ then $m = 1$ and z_1 has type $\mathbf{\imath}^{n+1} \to \mathbf{o}$. If v is a binary connective then $m = 2$ since t has $\mathcal{V}$-type.

Proof. It is straightforward to check that the 5 items of Definition 34 do not overlap and cover all possible cases. Indeed, every $\beta\eta$-reduced term t must be of the form $\lambda x_1 \cdots \lambda x_n\,(v\ t_1 \dots t_m)$ (with possibly $n = 0$ and $m = 0$), for variables $x_1, \dots, x_n$ of type ι and terms $v, t_1, \dots, t_m$ such that $(v\ t_1 \dots t_m)$ has $\mathcal{V}$-type and v is either a variable, a constant or an eigenterm. If $m > 0$ then we are in Case 4 if $v \in \{\forall, \exists, \neg, \wedge, \vee, \Rightarrow\}$ and in Case 5 otherwise. If $m = 0$, then v has $\mathcal{V}$-type; it may be an eigenterm and we are in Case 3, or $v \in \mathcal{V} \cup \mathcal{C}$ and then $n \geq 1$ (otherwise $t = v$). If v is a constant then we are in Case 2, otherwise it is a variable and then we are in Case 1 if $v = x_i$ for some $1 \leq i \leq n$, and in Case 2 otherwise. □

Lemma 36. *Let $t \in \mathcal{K} \setminus (\mathcal{V} \cup \mathcal{C})$ be a $\beta\eta$-reduced term of some $\mathcal{V}$-type and let $(u, \gamma) = \delta(t)$. Then $u \in \mathcal{K}$, γ is a $\mathcal{K}$-substitution, $u\gamma{\downarrow_{\beta\eta}} = t$, $\mathrm{FV}(t) \cap \mathrm{FV}(u) = \emptyset$ and $|x\gamma| < |t|$ for every variable $x \in \mathrm{Dom}(\gamma)$.*

Proof. The proof is immediate in Case 1 since then $u = t$, $\gamma = id$, $\mathrm{FV}(t) = \emptyset$ and $\mathrm{Dom}(\gamma) = \emptyset$. In Case 2, $u\gamma = (\lambda x_1 \cdots \lambda x_n\, y)[v/y] = \lambda x_1 \cdots \lambda x_n\, v = t$, $\mathrm{FV}(u) = \{y\}$ and by definition y cannot occur in $\mathrm{FV}(t) \subseteq \{v\}$, thus $\mathrm{FV}(t) \cap \mathrm{FV}(u) = \emptyset$. Furthermore, $|y\gamma| = 1 < |t| = n+1$ (since $n \geq 1$, see Definition 7 for the definition of $|t|$).

In Case 3, $u \in \mathcal{K}$ is obvious by the restriction on the y_i's and since all t_i's are in $\mathcal{K}$ then γ is a $\mathcal{K}$-substitution. $u\gamma = t$ is obvious, $\mathrm{FV}(u) = \{y_1, \dots, y_m\}$ is disjoint from $\mathrm{FV}(t_1, \dots, t_m)$ by definition and $\mathrm{FV}(t_1, \dots, t_m) = \mathrm{FV}(t)$ since $t \in \mathcal{K}$, hence $\mathrm{FV}(u) \cap \mathrm{FV}(t) = \emptyset$. Furthermore, $y_i\gamma = t_i$ for all $1 \leq i \leq m$, and $|t_i| < 1 + \sum_{j=1}^{m} |t_j| = |t|$.

In Case 5, $u \in \mathcal{K}$ is obvious and as above $t \in \mathcal{K}$ entails $u \in \mathcal{K}$ and $\lambda x_1 \cdots \lambda x_n\, t_i \in \mathcal{K}$ for all $1 \leq i \leq m$, hence γ is a $\mathcal{K}$-substitution. Since $y\gamma = v$ and $(z_i\gamma\ x_1 \cdots x_n){\downarrow_{\beta\eta}} = ((\lambda x_1 \cdots \lambda x_n\, t_i)\ x_1 \cdots x_n){\downarrow_{\beta\eta}} = t_i$, then

$$\begin{aligned} u\gamma{\downarrow_{\beta\eta}} &= \lambda x_1 \cdots \lambda x_n\, (y\gamma\ (z_1\gamma\ x_1 \cdots x_n) \cdots (z_m\gamma\ x_1 \cdots x_n)){\downarrow_{\beta\eta}} \\ &= \lambda x_1 \cdots \lambda x_n\, (v\ t_1 \cdots t_m) = t. \end{aligned}$$

Obviously $\mathrm{FV}(u) = \{y, z_1, \dots, z_m\}$ and by definition $y, z_1, \dots, z_m \notin \mathrm{FV}(t)$, hence $\mathrm{FV}(t) \cap \mathrm{FV}(u) = \emptyset$. Finally, $|y\gamma| = |u| > 0$, $|z_i\gamma| = n + |t_i|$ and $|t| = n + |u| + \Sigma_{i=1}^{m} |t_i|$ with $m > 0$ and $|t_1| > 0$, thus $|y\gamma|, |z_1\gamma|, \dots, |z_m\gamma| < |t|$. Case 4 is similar. □

A first algorithm for finding solutions of fair controlled unification problems is defined by the set of rules $\mathfrak{R} \stackrel{\text{def}}{=} \{\rightarrow_i |\ 1 \leq i \leq 8\}$ given in Figure 4. The rules apply modulo the commutativity of $\doteq$, the associativity and commutativity of $\curlywedge$, the fact that $\top$ is the identity of $\curlywedge$, and modulo α-equivalence (i.e., bound variables can be renamed to allow for the application of the rules). The constraint is assumed to be in long normal form and the unifier θ to be $\beta\eta$-reduced. When applying Rule $\rightarrow_8$ we assume that $\delta(x\theta)$ is renamed so that u shares no variable with the considered problem. The notation $\overline{y}$ denotes possibly empty sequences of variables (with $\lambda\overline{y}\, t \stackrel{\text{def}}{=} t$ if $\overline{y}$ is empty and $\lambda\overline{y}\, t \stackrel{\text{def}}{=} \lambda y_1 \cdots \lambda y_n\, t$ if $\overline{y} = y_1 \cdots y_n$ with $n > 0$). We also consider an extended algorithm defined by the set of rules $\mathfrak{R}' \stackrel{\text{def}}{=} \mathfrak{R} \cup \{\rightarrow_9\}$. We write $t \rightarrow_{\mathfrak{R}} s$ (resp. $t \rightarrow_{\mathfrak{R}'} s$) if $t \rightarrow_i s$ for some $1 \leq i \leq 8$ (resp. $1 \leq i \leq 9$).

Rule $\rightarrow_1$ merely removes trivial equations. Rule $\rightarrow_2$ removes useless λ-abstractions. Rule $\rightarrow_3$ is the usual replacement rule of the standard unification algorithms, which simply replaces a variable x by its value once it is known. The value of the variable is stored into the second component of the problem to ensure that equivalence is preserved. Note that $\rightarrow_3$ applies also to second order variables, which, since we consider problems in lnf, always occur in the scope of a λ-abstraction (e.g., an equation $x = y$ is written $\lambda z\, (x\ z) = \lambda z\, (y\ z)$ if x, y are of type $\iota \rightarrow \iota$). Rules $\rightarrow_4$, $\rightarrow_5$, $\rightarrow_6$ and $\rightarrow_7$ are usual decomposition rules. Note that Rules $\rightarrow_1$, $\rightarrow_2$, $\rightarrow_3$ $\rightarrow_4$, $\rightarrow_5$ and $\rightarrow_7$ preserve the set of solutions of the problem (the

$(t \doteq t \curlywedge \mathcal{X}, \sigma, \theta) \quad \to_1 (\mathcal{X}, \sigma, \theta)$

$(\lambda y\, t \doteq \lambda y\, s \curlywedge \mathcal{X}, \sigma, \theta) \quad \to_2 (t \doteq s \curlywedge \mathcal{X}, \sigma, \theta)$
if $y \notin \mathrm{FV}(t, s)$,

$(\lambda \overline{y}\, (x\ \overline{y}) \doteq \lambda \overline{y}\, s \curlywedge \mathcal{X}, \sigma, \theta) \quad \to_3 (\mathcal{X}\gamma\lfloor, \sigma\gamma, \theta)$
if $x \in \mathcal{V}$, $x \notin \mathrm{FV}(s)$ and $\gamma = [\lambda \overline{y}\, s/x]$,

$((\iota\, t_1 \cdots t_n) \doteq (\iota\, s_1 \cdots s_n) \curlywedge \mathcal{X}, \sigma, \theta) \quad \to_4 (\curlywedge_{i=1}^{n} t_i \doteq s_i \curlywedge \mathcal{X}, \sigma, \theta)$

$(\lambda \overline{y}\, (c\ t_1 \cdots t_n) \doteq \lambda \overline{y}\, (c\ s_1 \cdots s_n) \curlywedge \mathcal{X}, \sigma, \theta) \quad \to_5 (\curlywedge_{i=1}^{n} \lambda \overline{y}\, t_i \doteq \lambda \overline{y}\, s_i \curlywedge \mathcal{X}, \sigma, \theta)$
if $n \geq 1$ and $c \in \mathcal{C}$,

$(\lambda \overline{y}\, (x\ t_1 \cdots t_n) \doteq \lambda \overline{y}\, (z\ s_1 \cdots s_n) \curlywedge \mathcal{X}, \sigma, \theta) \to_6 ((\curlywedge_{i=1}^{n} \lambda \overline{y}\, t_i \doteq \lambda \overline{y}\, s_i \curlywedge \mathcal{X})\gamma, \sigma\gamma, \theta)$
if $n \geq 1$, $x, z \in \mathcal{V}$, $x\theta, z\theta \in \mathcal{V} \cup \mathcal{C}$ and $\gamma = [x/z]$,

$(\lambda \overline{y}\, (x\ t_1\, t_2) \doteq \lambda \overline{y}\, (\simeq\ s_1\, s_2) \curlywedge \mathcal{X}, \sigma, \theta) \quad \to_7 ((\curlywedge_{i=1}^{2} \lambda \overline{y}\, t_i \doteq \lambda \overline{y}\, s_i \curlywedge \mathcal{X})\gamma, \sigma\gamma, \theta)$
if $x \in \mathcal{V}$, $x\theta$ is $\simeq$ and $\gamma = [\simeq /x]$,

$(\lambda \overline{y}\, (x\ t_1 \cdots t_n) \doteq \lambda \overline{y}\, s \curlywedge \mathcal{X}, \sigma, \theta) \quad \to_8 ((\lambda \overline{y}\, (x\ t_1 \cdots t_n) \doteq \lambda \overline{y}\, s \curlywedge \mathcal{X})\gamma'\lfloor, \sigma\gamma', \theta\gamma)$
if $n \geq 0$, $x \in \mathcal{V}$, $x\theta \notin \mathcal{V} \cup \mathcal{C}$, $(u, \gamma) = \delta(x\theta)$ and $\gamma' = [u/x]$,

$(\lambda \overline{y}\, (x\ t_1 \cdots t_n) \doteq \lambda \overline{y}\, (z\ s_1 \cdots s_n) \curlywedge \mathcal{X}, \sigma, \theta) \to_9 ((\curlywedge_{i=1}^{n} \lambda \overline{y}\, t_i \doteq \lambda \overline{y}\, s_i \curlywedge \mathcal{X})\gamma, \sigma\gamma, \theta)$
if $n \geq 1$, $x, z \in \mathcal{V}$, $x\theta \equiv_{\alpha\beta\eta} z\theta$, $(\lambda \overline{y}\, t_i)\theta \equiv_{\alpha\beta\eta} (\lambda \overline{y}\, s_i)\theta$ for all $1 \leq i \leq n$ and $\gamma = [x/z]$.

Figure 4: The unifier minimization rules

obtained problem is always equivalent to the initial one). However, this is not the case for Rules $\to_6$, $\to_8$ and $\to_9$, for which some of the solutions may be lost (e.g., in Rule $\to_6$, if x and z are constant functions the terms $(x\ t_1 \cdots t_n)$ and $(z\ s_1 \cdots s_n)$ may be equal even if t_i and s_i are distinct). However, the application conditions ensure that the specific solution θ associated with the controlled unification problem (third component) is preserved.

Rules $\to_8$ and $\to_9$ are different ways of using the solution θ as a guide toward a generalized solution of the constraint. In Rule $\to_8$ the choice is to make a minimal copy of the value $x\theta$ for a variable x (as explained before Definition 34). In Rule $\to_9$ the choice is to assert as constraints some equalities that happen to hold on θ (but the common value $x\theta{\downarrow_{\beta\eta}} = z\theta{\downarrow_{\beta\eta}}$ is not used as such). The same choice motivates Rule $\to_6$, but it is postponed until atomic values are reached for $x\theta$ and $z\theta$ (which implies that $x\theta \equiv_{\alpha\beta\eta} z\theta$ and therefore Rule $\to_6$ is a restriction of Rule $\to_9$), when Rule $\to_8$ can no longer be applied. In contrast, Rules $\to_8$ and $\to_9$ overlap and both may entail some loss of generality, since the set of solutions is not preserved, and it is easy to construct examples showing that neither of the rules is uniformly superior to the other. There does not seem to be any easy way to decide which rule should preferably be applied in any given context (other than computing and comparing all normal forms, but this is not practical). The algorithm $\mathfrak{R}$ circumvents the dilemma.

In the following example, we adopt algorithm $\mathfrak{R}$ and apply the rules that preserve equivalence with the highest priority.

Example 37. We consider the fair constraint $\mathcal{Y}_3$ of Example 27. In order to obtain a solution of $\mathcal{Y}_3$, since $\mathcal{Y}_3\mu = \mathcal{X}_3$ we may compose μ (obtained by lifting) with the solution ρ of $\mathcal{X}_3$ defined in Example 5; that is

$$\begin{aligned} \mu\rho \;=\; & [\lambda y\, (p\ y\ a)/\xi_1,\ (p\ v\ a)/\phi_3,\ v/z_1,\ \xi/\xi_2,\ \psi/\phi_4,\ u/z_2, \forall x\, (p\ x\ a)/\phi_1, \\ & \exists y\, (p\ y\ a)/\phi_2][a/u,\ a/v,\ \lambda x\, (p\ x\ a)/\xi,\ (p\ a\ a)/\psi], \end{aligned}$$

of which we discard the variables not occurring in $\mathcal{Y}_3$; with the suitable normalizations,

this yields

$$\begin{array}{rcl} \mathcal{Y}_3\downarrow & = & \phi_4 \doteq \phi_3 \curlywedge \phi_1 \doteq \forall x\,(\xi_2\ x) \curlywedge \phi_4 \doteq (\xi_2\ z_2) \curlywedge \phi_2 \doteq \exists x\,(\xi_1\ x) \curlywedge \\ & & \phi_3 \doteq (\xi_1\ z_1), \\ \theta & \stackrel{\text{def}}{=} & [\forall x\,(p\ x\ a)/\phi_1,\ \exists y\,(p\ y\ a)/\phi_2,\ (p\ a\ a)/\phi_3,\ (p\ a\ a)/\phi_4, \\ & & \lambda y\,(p\ y\ a)/\xi_1,\ \lambda x\,(p\ x\ a)/\xi_2,\ a/z_1,\ a/z_2]. \end{array}$$

We thus start the unifier minimization algorithm on the fair controlled unification problem $(\mathcal{Y}_3\downarrow, id, \theta)$. The first four equations can be discarded by applying Rule $\rightarrow_3$ four times, which yields the controlled unification problem

$$((\xi_2\ z_2) \doteq (\xi_1\ z_1), [\forall x\,(\xi_2\ x)/\phi_1, \exists x\,(\xi_1\ x)/\phi_2, (\xi_2\ z_2)/\phi_3, (\xi_2\ z_2)/\phi_4], \theta).$$

Since $\xi_1\theta \notin \mathcal{V} \cup \mathcal{C}$ only Rule $\rightarrow_8$ applies, for instance with $\delta(\xi_1\theta) = \delta(\lambda y\,(p\ y\ a)) = (u_1, \gamma_1)$, where (by Case 5 of Definition 34) $u_1 \stackrel{\text{def}}{=} \lambda y\,(\xi_3\ (f_1\ y)\ (f_2\ y))$, $\gamma_1 \stackrel{\text{def}}{=} [p/\xi_3, \lambda y\,y/f_1, \lambda y\,a/f_2]$ and $\xi_3 : \mathbf{1} \to \mathbf{1} \to \mathbf{o}$, $f_1, f_2 : \mathbf{1} \to \mathbf{1}$ are new variables. The result of applying the rule is obtained by using the substitutions $[u_1/\xi_1]$ (and normalizing) and γ_1, yielding $((\xi_2\ z_2) \doteq (\xi_3\ (f_1\ z_1)\ (f_2\ z_1)), \sigma_1, \theta_1)$, where σ_1 is

$$[\forall x\,(\xi_2\ x)/\phi_1, \exists x\,(\xi_1\ x)/\phi_2, (\xi_2\ z_2)/\phi_3, (\xi_2\ z_2)/\phi_4][\lambda y\,(\xi_3\ (f_1\ y)\ (f_2\ y))/\xi_1]$$

and $\theta_1 \stackrel{\text{def}}{=} \theta\gamma_1$. Again only Rule $\rightarrow_8$ applies, for instance with $\delta(\xi_2\theta_1) = \delta(\lambda x\,(p\ x\ a)) = (u_2, \gamma_2)$, where $u_2 \stackrel{\text{def}}{=} \lambda x\,(\xi_4\ (f_3\ x)\ (f_4\ x))$, $\gamma_2 \stackrel{\text{def}}{=} [p/\xi_4, \lambda x\,x/f_3, \lambda x\,a/f_4]$ and $\xi_4 : \mathbf{1} \to \mathbf{1} \to \mathbf{o}$, $f_3, f_4 : \mathbf{1} \to \mathbf{1}$ are new variables. Applying the rule yields

$$((\xi_4\ (f_3\ z_2)\ (f_4\ z_2)) \doteq (\xi_3\ (f_1\ z_1)\ (f_2\ z_1)), \sigma_2, \theta_2)$$

where $\sigma_2 \stackrel{\text{def}}{=} \sigma_1[\lambda x\,(\xi_4\ (f_3\ x)\ (f_4\ x))/\xi_2]$ and $\theta_2 \stackrel{\text{def}}{=} \theta_1\gamma_2$. Now $\xi_4\theta_2 = \xi_4\gamma_2 = p$ and $\xi_3\theta_2 = \xi_3\gamma_1 = p$ are both (necessarily the same) constant, hence Rule $\rightarrow_6$ applies, yielding

$$((f_3\ z_2) \doteq (f_1\ z_1) \curlywedge (f_4\ z_2) \doteq (f_2\ z_1), \sigma_2[\xi_4/\xi_3], \theta_2).$$

Rule $\rightarrow_6$ cannot be applied, we select the first equation to apply Rule $\rightarrow_8$ with $\delta(f_3\theta_2) = (\lambda x\,x, id)$ (by Case 1), thus applying $[\lambda x\,x/f_3]$ which yields

$$(z_2 \doteq (f_1\ z_1) \curlywedge (f_4\ z_2) \doteq (f_2\ z_1), \sigma_2[\xi_4/\xi_3][\lambda x\,x/f_3], \theta_2).$$

Now Rule $\rightarrow_3$ applies on the first equation, which yields

$$((f_4\ (f_1\ z_1)) \doteq (f_2\ z_1), \sigma_2[\xi_4/\xi_3][\lambda x\,x/f_3][(f_1\ z_1)/z_2], \theta_2).$$

Since $f_4\theta_2 = \lambda x\,a$ by Rule $\rightarrow_8$ with $\delta(f_4\theta_2) = (\lambda x\,z_3, [a/z_3])$ (by Case 2) we obtain

$$(z_3 \doteq (f_2\ z_1), \sigma_2[\xi_4/\xi_3][\lambda x\,x/f_3][(f_1\ z_1)/z_2][\lambda x\,z_3/f_4], \theta_3)$$

with $\theta_3 = \theta_2[a/z_3]$ and finally Rule $\rightarrow_3$ yields

$$(\top, \sigma_2[\xi_4/\xi_3][\lambda x\,x/f_3][(f_1\ z_1)/z_2][\lambda x\,z_3/f_4][(f_2\ z_1)/z_3], \theta_3).$$

The result is

$$\begin{array}{rcl} \sigma_3 & \stackrel{\text{def}}{=} & \sigma_2[\xi_4/\xi_3][\lambda x\,x/f_3][(f_1\ z_1)/z_2][\lambda x\,z_3/f_4][(f_2\ z_1)/z_3] \\ & = & [\forall x\,(\xi_4\ x\ (f_2\ z_1))/\phi_1, \exists x\,(\xi_4\ (f_1\ x)\ (f_2\ x))/\phi_2, (\xi_4\ (f_1\ z_1)\ (f_2\ z_1))/\phi_3, \\ & & (\xi_4\ (f_1\ z_1)\ (f_2\ z_1))/\phi_4, \lambda y\,(\xi_4\ (f_1\ y)\ (f_2\ y))/\xi_1, \lambda x\,(\xi_4\ x\ (f_2\ z_1))/\xi_2, \\ & & \xi_4/\xi_3, \lambda x\,x/f_3, (f_1\ z_1)/z_2, \lambda x\,(f_2\ z_1)/f_4, (f_2\ z_1)/z_3]. \end{array}$$

It is easy to check that σ_3 is a solution of $\mathcal{Y}_3$ and that it is more general than θ_4. Indeed, we find that $\sigma_3\gamma_1\gamma_2[a/z_1, a/z_2, a/z_3] \equiv_{\alpha\beta\eta} \theta_3$. Hence the restriction of σ_3 to $\mathrm{FV}(\mathcal{Y}_3)$ is more general than θ. Note that the solution σ_3 is fair even though the solution θ is not.

This algorithm is nondeterministic; the reader may check that if we change the orientation of the equation on which we first apply Rule $\to_8$, i.e., if we apply it to $(f_1\ z_1) \doteq (f_3\ z_2)$ we reach the following solution:

$$\begin{array}{rcl}\sigma_3' & = & [\forall x\,(\xi_4\ (f_3\ x)\ (f_4\ x))/\phi_1, \exists x\,(\xi_4\ x\ (f_4\ z_2))/\phi_2, (\xi_4\ (f_3\ z_2)\ (f_4\ z_2))/\phi_3,\\ & & (\xi_4\ (f_3\ z_2)\ (f_4\ z_2))/\phi_4, \lambda y\,(\xi_4\ y\ (f_4\ z_2))/\xi_1, \lambda x\,(\xi_4\ (f_3\ x)\ (f_4\ x))/\xi_2,\\ & & \xi_4/\xi_3, \lambda y\,y/f_1, (f_3\ z_2)/z_1, \lambda y\,(f_4\ z_2)/f_2, (f_4\ z_2)/z_3]\end{array}$$

and none of σ_3, σ_3' is more general than the other. The reader may check that Rule $\to_9$ yields extra solutions, but they are instances of those already found.

We next prove that these algorithms always produce a fair solution more general than the given unifier.

8 Properties and Consequences

We first state that the obtained triple is always a fair controlled unification problem and that all its solutions are also solutions of the original problem.

Lemma 38. *If $(\mathcal{Y}, \sigma, \theta)$ is a fair controlled unification problem, where $\mathcal{Y}$ is in lnf and θ is $\beta\eta$-reduced, and $(\mathcal{Y}, \sigma, \theta) \to_{\mathfrak{R}'} (\mathcal{Y}', \sigma', \theta')$ then $(\mathcal{Y}', \sigma', \theta')$ is a fair controlled unification problem, $\mathcal{Y}'$ is in lnf, θ' is a $\beta\eta$-reduced extension of θ and $\mathrm{sol}(\mathcal{Y}', \sigma') \subseteq \mathrm{sol}(\mathcal{Y}, \sigma)$.*

Proof.

- We consider Rule $\to_2$ by assuming that $\mathcal{Y} \equiv \lambda y\,t \doteq \lambda y\,s \curlywedge \mathcal{X}$, $\mathcal{Y}' \equiv t \doteq s \curlywedge \mathcal{X}$, with $y \notin \mathrm{FV}(t, s)$ and $\sigma' = \sigma$, $\theta' = \theta$. Since $\mathcal{Y}$ is a fair $\mathcal{K}$-constraint in lnf, then so is $\mathcal{Y}'$. Similarly σ' is a fair idempotent $\mathcal{K}$-substitution, and since $\mathrm{FV}(\mathcal{Y}') = \mathrm{FV}(t, s, \mathcal{X}) = \mathrm{FV}(\lambda y\,t, \lambda y\,s, \mathcal{X}) = \mathrm{FV}(\mathcal{Y})$, then $\mathrm{Dom}(\sigma') \cap \mathrm{FV}(\mathcal{Y}') = \mathrm{Dom}(\sigma) \cap \mathrm{FV}(\mathcal{Y}) = \emptyset$. $\sigma' \precsim \theta'$ is obvious, and since θ is a solution of $\mathcal{Y}$, then $(\lambda y\,t)\theta \equiv_{\alpha\beta\eta} (\lambda y\,s)\theta$. Let $x \in \mathcal{V} \setminus \mathrm{FV}(t\theta, s\theta)$ of type $\mathfrak{1}$, then $(\lambda y\,t)\theta = \lambda x\,t\theta[y \mapsto x] = \lambda x\,t\theta$ since $y \notin \mathrm{FV}(t)$, and similarly $(\lambda y\,t)\theta = \lambda x\,s\theta$, hence $\lambda x\,t\theta \equiv_{\alpha\beta\eta} \lambda x\,s\theta$ and therefore $t\theta \equiv_{\alpha\beta\eta} s\theta$. Thus θ' is a solution of $\mathcal{Y}'$, hence $(\mathcal{Y}', \sigma', \theta')$ is a fair controlled unification problem. θ' is obviously a $\beta\eta$-reduced extension of θ. Furthermore, it is clear that for every substitution μ, $t\mu \equiv_{\alpha\beta\eta} s\mu$ implies $(\lambda y\,t)\mu \equiv_{\alpha\beta\eta} (\lambda y\,s)\mu$ thus $\mathrm{sol}(\mathcal{Y}, \sigma) \subseteq \mathrm{sol}(\mathcal{Y}', \sigma)$.

- We examine Rule $\to_8$, hence we assume that $\mathcal{Y} \equiv \lambda \overline{y}\,(x\ t_1 \cdots t_n) \doteq \lambda \overline{y}\,s \curlywedge \mathcal{X}$ with $n \geq 1$, $x \in \mathcal{V}$, $x\theta \notin \mathcal{V} \cup \mathcal{C}$, $(u, \gamma) = \delta(x\theta)$, $\gamma' = [u/x]$, $\mathcal{Y}' = \mathcal{Y}\gamma'\lfloor$, $\sigma' = \sigma\gamma'$ and $\theta' = \theta\gamma$. By inspecting Definition 34 we see that only in Case 4 does u contain a constant, and it is a logical symbol; hence u is fair and therefore so are $\mathcal{Y}'$ and σ'. By the condition given on Page 25 for applying Rule $\to_8$ we also know that $\mathrm{FV}(u) \cap \mathrm{FV}(\mathcal{Y}, \sigma, \theta) = \emptyset$, and Definition 34 yields $\mathrm{Dom}(\gamma) = \mathrm{FV}(u)$. This first implies that $\mathrm{Dom}(\gamma) \cap \mathrm{FV}(\theta) = \emptyset$ and hence that θ' is an extension of θ. Since γ is clearly $\beta\eta$-reduced then so is θ'.

  By Lemma 36 $u\gamma \equiv_{\alpha\beta\eta} x\theta$, $u \in \mathcal{K}$, γ is a $\mathcal{K}$-substitution and so is γ', hence by Proposition 14 so are σ' and θ', and $\mathcal{Y}'$ is a $\mathcal{K}$-constraint. Since $\mathrm{FV}(u) \cap \mathrm{Dom}(\theta) = \emptyset$, then $x\gamma'\theta\gamma = u\theta\gamma = u\gamma \equiv_{\alpha\beta\eta} x\theta$. Besides, for any $z \in \mathrm{FV}(\mathcal{Y})$ other than x, $z\gamma'\theta\gamma = z\theta\gamma = z\theta$ since $\mathrm{FV}(z\theta) \subseteq \mathrm{FV}(\mathcal{Y}, \theta)$ is disjoint from $\mathrm{Dom}(\gamma)$. Hence $\mathcal{Y}'\theta' = \mathcal{Y}\gamma'\theta\gamma \equiv \mathcal{Y}\theta \equiv \top$, so that $\theta' \in \mathrm{sol}(\mathcal{Y}')$.

  Since $(\mathcal{Y}, \sigma, \theta)$ is a controlled unification problem then $\sigma^2 = \sigma$ and $\mathrm{Dom}(\sigma) \cap \mathrm{FV}(\mathcal{Y}) = \emptyset$. But $\mathrm{Dom}(\sigma') = \mathrm{Dom}(\sigma) \uplus \{x\}$ and $\mathrm{FV}(\mathcal{Y}') = (\mathrm{FV}(\mathcal{Y}) \cup \mathrm{FV}(u)) \setminus \{x\}$, hence

$\mathrm{Dom}(\sigma') \cap \mathrm{FV}(\mathcal{Y}') = \mathrm{Dom}(\sigma) \cap \mathrm{FV}(u) = \emptyset$. Hence $u\sigma' = u$ and since $x\sigma' = u$ then $x\sigma'^2 = u = x\sigma'$. Furthermore, for every $z \in \mathrm{Dom}(\sigma)$, $z\sigma'^2 = z\sigma\sigma' = z\sigma^2\gamma' = z\sigma'$. Therefore $\sigma'^2 = \sigma'$.

We also know that $\sigma \precsim \theta$, thus there exists a ρ such that, for every variable z, $z\sigma\rho \equiv_{\alpha\beta\eta} z\theta$. Let ρ' be the substitution coinciding with γ on $\mathrm{FV}(u)$ and with ρ elsewhere. If $z \in \mathrm{FV}(u)$ then $z\sigma'\rho' = z\rho' = z\gamma = z\theta'$. If $z \notin \mathrm{FV}(u)$ let $z' \in \mathrm{FV}(z\sigma)$, then $z' \notin \mathrm{FV}(u)$ and $z' \notin \mathrm{Dom}(\sigma)$ (since $\sigma^2 = \sigma$). If $z' \neq x$ then $z'\sigma'\rho' = z'\gamma\rho' = z'\rho' = z'\rho = z'\sigma\rho$. Furthermore, $x\sigma'\rho' = u\rho' = u\gamma \equiv_{\alpha\beta\eta} x\theta \equiv_{\alpha\beta\eta} x\sigma\rho$. This proves that $(z\sigma)\sigma'\rho' \equiv_{\alpha\beta\eta} (z\sigma)\sigma\rho$, but $z\sigma\sigma'\rho' = z\sigma'\rho'$ and $z\sigma^2\rho = z\sigma\rho \equiv_{\alpha\beta\eta} z\theta = z\theta'$, hence $z\sigma'\rho' \equiv_{\alpha\beta\eta} z\theta'$ holds for every variable z, thus $\sigma' \precsim \theta'$ and we have proved that $(\mathcal{Y}', \sigma', \theta')$ is a controlled unification problem.

For any $\mu \in \mathrm{sol}(\mathcal{Y}', \sigma')$ by definition $\mu \in \mathrm{sol}(\mathcal{Y}')$ and $\sigma' \precsim \mu$, hence there is a ρ such that $\mu \equiv_{\alpha\beta\eta} \sigma'\rho = \sigma\gamma'\rho$. Since $\mathrm{Dom}(\sigma) \cap \mathrm{FV}(\mathcal{Y}') \subseteq \mathrm{Dom}(\sigma) \cap \mathrm{FV}(\mathcal{Y}, u) = \emptyset$, then $\mathcal{Y}'\sigma = \mathcal{Y}'$ as well as $\mathcal{Y}\sigma = \mathcal{Y}$, and since $\gamma'^2 = \gamma'$ then $\mathcal{Y}'\mu \equiv \mathcal{Y}'\gamma'\rho \equiv \mathcal{Y}\gamma'\rho = \mathcal{Y}\sigma\gamma'\rho \equiv \mathcal{Y}\mu$. But $\mathcal{Y}'\mu \equiv \top$, which proves that $\mu \in \mathrm{sol}(\mathcal{Y}, \sigma)$.

- We now consider Rule $\rightarrow_9$, hence $\mathcal{Y} \equiv \lambda\overline{y}\,(x\ t_1 \cdots t_n) \doteq \lambda\overline{y}\,(z\ s_1 \cdots s_n) \curlywedge \mathcal{X}$, $n \geq 1$, $x, z \in \mathcal{V}$, $x\theta \equiv_{\alpha\beta\eta} z\theta$, $(\lambda\overline{y}\,t_i)\theta \equiv_{\alpha\beta\eta} (\lambda\overline{y}\,s_i)\theta$ for all $1 \leq i \leq n$, $\gamma = [x/z]$, $\mathcal{Y}' \equiv (\curlywedge_{i=1}^n \lambda\overline{y}\,t_i \doteq \lambda\overline{y}\,s_i \curlywedge \mathcal{X})\gamma$, $\sigma' = \sigma\gamma$ and $\theta' = \theta$. Obviously $\mathcal{Y}'$ is a fair $\mathcal{K}$-constraint in lnf and σ' is a fair $\mathcal{K}$-constraint. Since $\mathrm{Dom}(\sigma) \cap \mathrm{FV}(\mathcal{Y}) = \emptyset$ then $x, z \notin \mathrm{Dom}(\sigma)$, hence $\mathrm{Dom}(\sigma') = \mathrm{Dom}(\sigma) \cup \{z\}$, and since $\mathrm{FV}(\mathcal{Y}') = \mathrm{FV}(\mathcal{Y}) \setminus \{z\}$ then $\mathrm{Dom}(\sigma') \cap \mathrm{FV}(\mathcal{Y}') = \emptyset$. This also implies that $\gamma\sigma\gamma = \sigma\gamma$ and since σ is idempotent then so is σ'.

  Since $\sigma \precsim \theta$ there is a ρ such that $\sigma\rho \equiv_{\alpha\beta\eta} \theta$. Then

  $$z\sigma'\rho = z\sigma\gamma\rho = z\gamma\rho = x\rho = x\sigma\rho \equiv_{\alpha\beta\eta} x\theta \equiv_{\alpha\beta\eta} z\theta.$$

  Furthermore, for every variable $z' \neq z$, $z'\sigma'\rho = z'\sigma\rho \equiv_{\alpha\beta\eta} z'\theta$. Thus $\sigma' \precsim \theta$.

  Since $x\theta \equiv_{\alpha\beta\eta} z\theta$, then $t\gamma\theta \equiv_{\alpha\beta\eta} t\theta$ for any term t, hence $(\lambda\overline{y}\,t_i)\gamma\theta \equiv_{\alpha\beta\eta} (\lambda\overline{y}\,t_i)\theta$ for all $1 \leq i \leq n$, and similarly for s_i so that θ is clearly a solution of $(\lambda\overline{y}\,t_i)\gamma \doteq (\lambda\overline{y}\,s_i)\gamma$ and similarly of $\mathcal{X}\gamma$, hence $(\mathcal{Y}', \sigma', \theta')$ is a controlled unification problem.

  Finally we prove that $\mathrm{sol}(\mathcal{Y}', \sigma') \subseteq \mathrm{sol}(\mathcal{Y}, \sigma)$. Let $\mu \in \mathrm{sol}(\mathcal{Y}', \sigma')$, by definition there is a ρ such that $\mu \equiv_{\alpha\beta\eta} \sigma'\rho = \sigma\gamma\rho$ (hence $\sigma \precsim \mu$) and $(\lambda\overline{y}\,t_i)\gamma\mu \equiv_{\alpha\beta\eta} (\lambda\overline{y}\,s_i)\gamma\mu$ for all $1 \leq i \leq n$. But $z\mu \equiv_{\alpha\beta\eta} z\sigma\gamma\rho = x\rho = x\sigma\gamma\rho \equiv_{\alpha\beta\eta} x\mu$, hence $t\gamma\mu \equiv_{\alpha\beta\eta} t\mu$ for any term t; in particular $\mathcal{X}\mu \equiv \mathcal{X}\gamma\mu \equiv \top$. Let $t = (x\ t_1 \cdots t_n)$, $s = (z\ s_1 \cdots s_n)$ and $\overline{y}'$ be a sequence of variables of type $\mathbf{1}$ such that $\overline{y}' \cap \mathrm{FV}(t\gamma\mu, s\gamma\mu) = \emptyset$, then $(\lambda\overline{y}\,t_i)\gamma\mu = \lambda\overline{y}'\,t_i\gamma\mu[\overline{y} \mapsto \overline{y}']$ and similarly for s_i, hence $t_i\gamma\mu[\overline{y} \mapsto \overline{y}'] \equiv_{\alpha\beta\eta} s_i\gamma\mu[\overline{y} \mapsto \overline{y}']$ for all $1 \leq i \leq n$, and since $x\gamma = z\gamma$ then $t\gamma\mu[\overline{y} \mapsto \overline{y}'] \equiv_{\alpha\beta\eta} s\gamma\mu[\overline{y} \mapsto \overline{y}']$. This yields

  $$\begin{array}{lll} (\lambda\overline{y}\,t)\mu \equiv_{\alpha\beta\eta} (\lambda\overline{y}\,t)\gamma\mu & = \lambda\overline{y}'\,t\gamma\mu[\overline{y} \mapsto \overline{y}'] & \\ & \equiv_{\alpha\beta\eta} \lambda\overline{y}'\,s\gamma\mu[\overline{y} \mapsto \overline{y}'] & = (\lambda\overline{y}\,s)\gamma\mu \equiv_{\alpha\beta\eta} (\lambda\overline{y}\,s)\mu \end{array}$$

  hence $\mu \in \mathrm{sol}(\mathcal{Y}, \sigma)$.

- Rule $\rightarrow_1$ is trivial, Rule $\rightarrow_3$ is the standard replacement rule, Rules $\rightarrow_4$, $\rightarrow_5$ and $\rightarrow_7$ are standard decomposition rules and can be treated as above.

- We finally consider Rule $\rightarrow_6$, thus $\mathcal{Y} \equiv \lambda\overline{y}\,(x\ t_1 \cdots t_n) \doteq \lambda\overline{y}\,(z\ s_1 \cdots s_n) \curlywedge \mathcal{X}$, $n \geq 1$, $x, z \in \mathcal{V}$, $x\theta, z\theta \in \mathcal{V} \cup \mathcal{C}$, $\gamma = [x/z]$, $\mathcal{Y}' \equiv (\curlywedge_{i=1}^n \lambda\overline{y}\,t_i \doteq \lambda\overline{y}\,s_i \curlywedge \mathcal{X})\gamma$, $\sigma' = \sigma\gamma$ and $\theta' = \theta$. Let $t = (x\ t_1 \cdots t_n)$, $s = (z\ s_1 \cdots s_n)$ and $\overline{y}'$ be a sequence of variables of type

1 such that $\overline{y}' \cap \mathrm{FV}(t\theta, s\theta) = \emptyset$, then $(\lambda\overline{y}\, t)\theta = \lambda\overline{y}'\, t\theta[\overline{y} \mapsto \overline{y}']$ and similarly for s, and since θ is a solution of $\lambda\overline{y}\, t \doteq \lambda\overline{y}\, s$ then $t\theta[\overline{y} \mapsto \overline{y}'] \equiv_{\alpha\beta\eta} s\theta[\overline{y} \mapsto \overline{y}']$.

But $x\theta[\overline{y} \mapsto \overline{y}'] = x\theta \in \mathcal{V} \cup \mathcal{C}$, hence

$$t\theta[\overline{y} \mapsto \overline{y}']{\downarrow_{\beta\eta}} = (x\theta\ t_1\theta[\overline{y} \mapsto \overline{y}']{\downarrow_{\beta\eta}} \cdots t_n\theta[\overline{y} \mapsto \overline{y}']{\downarrow_{\beta\eta}})$$

and similarly for s, hence $x\theta = z\theta$ and $t_i\theta[\overline{y} \mapsto \overline{y}'] \equiv_{\alpha\beta\eta} s_i\theta[\overline{y} \mapsto \overline{y}']$ for all $1 \leq i \leq n$, so that $(\lambda\overline{y}\, t_i)\theta \equiv_{\alpha\beta\eta} (\lambda\overline{y}\, s_i)\theta$. This shows that the conditions of Rule $\to_9$ hold and we can therefore conclude as above.

□

The last item of the previous proof shows that Rule $\to_6$ is a restriction of Rule $\to_9$ and can therefore be removed from $\mathfrak{R}'$. Next, we prove that $\mathfrak{R}'$ terminates.

Proposition 39. *$\mathfrak{R}'$ is terminating.*

Proof. We consider the measure ℓ on controlled unification problems defined as follows. For every problem $\mathcal{P} = (\mathcal{X}, \sigma, \theta)$, $\ell(\mathcal{P}) \stackrel{\text{def}}{=} (\ell_1(\mathcal{P}), \ell_2(\mathcal{P}))$, where:

- $\ell_1(\mathcal{P}) \stackrel{\text{def}}{=} \{|x\theta| \mid x \in \mathrm{FV}(\mathcal{X})\}$.
- $\ell_2(\mathcal{P}) \stackrel{\text{def}}{=} \{\{|t|, |s|\} \mid t \doteq s \text{ occurs in } \mathcal{X}\}$.

The measures ℓ_1 and ℓ_2 are ordered by the multiset extensions of the standard ordering on natural numbers, and ℓ is ordered by the lexicographic extension of the measures on ℓ_1 and ℓ_2. We show that ℓ decreases strictly each time a rule in $\mathfrak{R}'$ is applied. Rule $\to_1$ cannot increase ℓ_1 and strictly decreases ℓ_2 (since one equation is removed). Rule $\to_2$ does not affect ℓ_1 (the set of free variables does not change) and strictly decreases ℓ_2. Rule $\to_3$ replaces a variable in $\mathrm{FV}(\mathcal{X})$ by a term $\lambda\overline{y}\, s$. Since $x \notin \mathrm{FV}(\lambda\overline{y}\, s)$, x does not occur freely in the obtained constraint, thus (since no new variable is introduced) ℓ_1 must be decreasing. Rule $\to_4$ does not affect ℓ_1 and replaces an equation $\lambda\overline{y}\, (\iota\, t_1 \cdots t_n) \doteq \lambda\overline{y}\, (\iota\, s_1 \cdots s_n)$ by n equations $\lambda\overline{y}\, t_i \doteq \lambda\overline{y}\, s_i$. Since $|t_1|, \ldots, |t_n| \leq \sum_{i=1}^{n} |t_i| < |(\iota\, t_1 \cdots t_n)|$ and similarly for the s_i's, then ℓ_2 decreases strictly. The proof for Rules $\to_5$, $\to_6$, $\to_7$ and $\to_9$ is similar since the substitution of a variable by another one does not affect ℓ_2 (ℓ_1 also decreases in the last two rules).

In Rule $\to_8$ a variable x is replaced by a term u in the constraint $\mathcal{X}$ before it is transformed to lnf, and θ is replaced by $\theta\gamma$. Note that u is not a variable and contains no variable in $\mathcal{X}$ (thus $z\theta\gamma = z\theta$ for every $z \in \mathrm{FV}(\mathcal{X})$). Furthermore, for every variable $y \in \mathrm{Dom}(\gamma)$, y occurs in u hence by strategy (last condition) $y \notin \mathrm{Dom}(\theta)$ and therefore $|y\theta\gamma| = |y\gamma| < |x\theta|$ by Lemma 36. Consequently, ℓ_1 must be strictly decreasing after the replacement of x by u, since the element $|x\theta|$ is deleted and replaced by strictly smaller elements $|y\theta\gamma|$. Then it is obvious that the reduction to lnf cannot increase ℓ_1 since it may remove (by β-reduction) but not add free variables to the constraint. □

The two previous results obviously hold also for $\mathfrak{R}$. The following lemma states that all $\mathfrak{R}$-irreducible fair problems are solved.

Lemma 40. *All $\mathfrak{R}$-irreducible fair controlled unification problems with a constraint in lnf and a $\beta\eta$-reduced solution θ are of the form $(\top, \sigma, \theta)$.*

Proof. Let $(\mathcal{Y}, \sigma, \theta)$ be an irreducible fair controlled unification problem where $\mathcal{Y}$ is in lnf and θ is $\beta\eta$-reduced. Since $\theta \in \mathrm{sol}(\mathcal{Y})$, $\mathcal{Y}$ cannot be $\bot$. Assume that $\mathcal{Y}$ is not $\top$, we show that $(\mathcal{Y}, \sigma, \theta)$ is not irreducible, in contradiction with the hypothesis. By suitable α-conversion we can assume that $\mathcal{Y}$ contains an equation $\lambda\overline{y}\, t \doteq \lambda\overline{y}\, s$ where $\overline{y}$ has no

common variable with $\mathrm{FV}(\theta)$ and, due to the lnf, t and s are terms of order 1. Thus $\lambda\overline{y}\,t\theta \equiv_{\alpha\beta\eta} (\lambda\overline{y}\,t)\theta \equiv_{\alpha\beta\eta} (\lambda\overline{y}\,s)\theta \equiv_{\alpha\beta\eta} \lambda\overline{y}\,s\theta$, hence $t\theta \equiv_{\alpha\beta\eta} s\theta$, and neither t nor s is an abstraction. Let $\mathcal{X}$ be the remaining constraint, i.e., such that $\mathcal{Y} \equiv \lambda\overline{y}\,t \doteq \lambda\overline{y}\,s \curlywedge \mathcal{X}$. Since $\mathcal{Y}$ is fair and there is no logical symbol of order 1, neither t nor s is a constant; hence they are either variables, applications or eigenterms.

Suppose t and s are both variables. If $\overline{y}$ is empty then either $t = s$ and then Rule $\to_1$ applies; or $t \neq s$ and then Rule $\to_3$ applies. If $\overline{y}$ is not empty and one of t or s, say t, belongs to $\overline{y}$ then $(\lambda\overline{y}\,t)\theta = (\lambda\overline{y}\,t) \equiv_{\alpha\beta\eta} (\lambda\overline{y}\,s)\theta$, which means that s must be the same variable as t, hence Rule $\to_1$ applies. If none belong to the non empty $\overline{y}$, then the first variable of $\overline{y}$ does not belong to $\mathrm{FV}(t, s) = \{t, s\}$ and Rule $\to_2$ applies. Since $(\mathcal{Y}, \sigma, \theta)$ is irreducible then t and s cannot both be variables.

Suppose one of t or s, say t, is a variable x; then s is not a variable. If $x\theta \notin \mathcal{V} \cup \mathcal{C}$ then Rule $\to_8$ applies, hence $x\theta \in \mathcal{V} \cup \mathcal{C}$. This means that $s\theta{\downarrow_{\beta\eta}}$ is equal to $x\theta$, hence $s\theta{\downarrow_{\beta\eta}} \in \mathcal{V} \cup \mathcal{C}$. But $s \notin \mathcal{V} \cup \mathcal{C}$ hence s must be an application $(z\ s_1 \cdots s_n)$ with $n \geq 1$ (which only can be eliminated by β-reduction) and $z\theta$ must be an abstraction. Since s is in lnf then z must be a variable, hence Rule $\to_8$ applies (to the same equation with reverse orientation). We conclude that neither t nor s can be a variable.

If one of t or s, say t, is an eigenterm $(\iota\, t_1 \cdots t_n)$, then so are $t\theta$ and $s\theta$. If s is an eigenterm $(\iota\, s_1 \cdots s_m)$, then $(\iota\, t_1\theta{\downarrow_{\beta\eta}} \cdots t_n\theta{\downarrow_{\beta\eta}}) = t\theta{\downarrow_{\beta\eta}} = s\theta{\downarrow_{\beta\eta}} = (\iota\, s_1\theta{\downarrow_{\beta\eta}} \cdots s_m\theta{\downarrow_{\beta\eta}})$, hence $n = m$. If $\overline{y}$ is not empty then its first variable does not belong to $\mathrm{FV}(t, s)$ by definition of $\mathcal{K}$ hence Rule $\to_2$ applies; otherwise $\overline{y}$ is empty and Rule $\to_4$ applies. Thus s cannot be an eigenterm or a variable, hence as above it must be an application $(z\ s_1 \cdots s_m)$ with $m \geq 1$, $z \in \mathcal{V}$ and $z\theta$ is an abstraction, hence Rule $\to_8$ applies (with reverse orientation). We conclude that neither t nor s can be an eigenterm.

Hence t and s must both be applications; let $(v\ t_1 \cdots t_n) = t$ and $(w\ s_1 \cdots s_m) = s$ where $n, m \geq 1$ and v, w are either variables or constants. If they are both constants, then $t\theta{\downarrow_{\beta\eta}} = (v\ t_1\theta{\downarrow_{\beta\eta}} \cdots t_n\theta{\downarrow_{\beta\eta}})$ and $s\theta{\downarrow_{\beta\eta}} = (w\ s_1\theta{\downarrow_{\beta\eta}} \cdots s_m\theta{\downarrow_{\beta\eta}})$ are equal, hence $n = m$, $v = w$ and Rule $\to_5$ applies. Hence one of v or w, say v, is a variable. If $v\theta \notin \mathcal{V} \cup \mathcal{C}$ then Rule $\to_8$ applies, hence $v\theta \in \mathcal{V} \cup \mathcal{C}$ and therefore $t\theta{\downarrow_{\beta\eta}} = (v\theta\ t_1\theta{\downarrow_{\beta\eta}} \cdots t_n\theta{\downarrow_{\beta\eta}})$. If w is also a variable then either $w\theta \notin \mathcal{V} \cup \mathcal{C}$ and Rule $\to_8$ applies (with reverse orientation), or $w\theta \in \mathcal{V} \cup \mathcal{C}$ and therefore $s\theta{\downarrow_{\beta\eta}} = (w\theta\ t_1\theta{\downarrow_{\beta\eta}} \cdots t_n\theta{\downarrow_{\beta\eta}})$, $n = m$ and Rule $\to_6$ applies. w must therefore be a constant, so that $s\theta{\downarrow_{\beta\eta}} = (w\ s_1\theta{\downarrow_{\beta\eta}} \cdots s_m\theta{\downarrow_{\beta\eta}})$, $n = m$ and $v\theta = w$. But $\mathcal{X}$ is fair, hence w is a logical symbol of $\mathcal{V}$-type, i.e., w is $\simeq$ and therefore Rule $\to_7$ applies, a contradiction. □

This entails that both $\mathfrak{R}$- and $\mathfrak{R}'$-normal forms are exactly the solved problems. This also means that, starting from a given controlled unification problem, $\mathfrak{R}'$ may reach more solutions than $\mathfrak{R}$; hence we adopt $\mathfrak{R}'$ as our minimization algorithm and we denote by $\mathcal{P}{\downarrow_{\mathfrak{R}'}}$ an arbitrarily chosen $\mathfrak{R}'$-normal form of $\mathcal{P}$. By putting together all the previous results, we immediately infer the correction of this minimization algorithm:

Theorem 41. *Let $\mathcal{X}$ be a fair $\mathcal{K}$-constraint in lnf, θ a $\beta\eta$-reduced $\mathcal{K}$-substitution such that $\theta \in \mathrm{sol}(\mathcal{X})$ and $(\mathcal{Y}, \sigma, \theta') = (\mathcal{X}, id, \theta){\downarrow_{\mathfrak{R}'}}$, then θ' is an extension of θ and σ is a fair $\mathcal{K}$-substitution such that $\sigma \in \mathrm{sol}(\mathcal{X})$ and $\sigma \precsim \theta'$.*

Proof. $(\mathcal{X}, id, \theta)$ is obviously a fair controlled unification problem, by Proposition 39 any $\mathfrak{R}'$-normal form $(\mathcal{Y}, \sigma, \theta')$ exists and by Lemma 38 it is a fair controlled unification problem (hence $\sigma \precsim \theta'$) such that θ' is an extension of θ and $\mathrm{sol}(\mathcal{Y}, \sigma) \subseteq \mathrm{sol}(\mathcal{X}, id)$, and $\mathcal{Y}$ is in lnf according to the strategy. But $\mathcal{Y} \equiv \top$ by Lemma 40, hence $\sigma \in \mathrm{sol}(\mathcal{Y}, \sigma)$ and therefore $\sigma \in \mathrm{sol}(\mathcal{X})$. □

Input: a standard LK-proof $\boldsymbol{P} : \Gamma \vdash \Delta$.

1. Translate $\boldsymbol{P}$ into a valid $\mathcal{K}$-LK$^{\mathrm{c}}$-proof $\boldsymbol{Q} : \Gamma \vdash \Delta \mid \mathcal{X}$.

2. Compute an abstract proof $\boldsymbol{Q}^{\mathrm{a}} : \Gamma^{\mathrm{a}} \vdash \Delta^{\mathrm{a}} \mid \mathcal{X}^{\mathrm{a}}$ and a $\mathcal{K}$-substitution θ such that $\boldsymbol{Q}^{\mathrm{a}}\theta = \boldsymbol{Q}$.

3. Compute an $\mathfrak{R}'$-normal form $(\top, \sigma, \theta')$ of the fair controlled unification problem $(\mathcal{X}^{\mathrm{a}}\restriction, id, \theta{\downarrow_{\beta\eta}})$.

4. Translate $\boldsymbol{Q}^{\mathrm{a}}\sigma$ into a standard LK-proof $\boldsymbol{P}' : \Gamma' \vdash \Delta'$ such that $\Gamma' \vdash \Delta' \preceq \Gamma^{\mathrm{a}}\sigma{\downarrow_{\beta\eta}} \vdash \Delta^{\mathrm{a}}\sigma{\downarrow_{\beta\eta}}$ and output $\boldsymbol{P}'$.

Figure 5: The generalization algorithm

Note that, if σ' is the restriction of σ to $\mathrm{FV}(\mathcal{X})$, σ' is still a solution of $\mathcal{X}$, $\sigma' \preceq \sigma$ and by construction θ' is equal to θ on $\mathrm{FV}(\mathcal{X})$, hence $\sigma' \precsim \theta$ and therefore σ' is a solution of the initial controlled unification problem $(\mathcal{X}, id, \theta)$.

We can now collect the results of the previous and present sections into an algorithm performing proof generalization in LK, see Figure 5.

Theorem 42. *Given any standard* LK*-proof* $\boldsymbol{P} : \Gamma \vdash \Delta$*, the generalization algorithm terminates and outputs a standard and fair* LK*-proof* $\boldsymbol{P}' : \Gamma' \vdash \Delta'$ *of the same length as* $\boldsymbol{P}$ *such that* $\Gamma' \vdash \Delta' \precsim \Gamma \vdash \Delta$*.*

Proof. The first step is performed according to Theorem 15, which provides the valid $\mathcal{K}$-LK$^{\mathrm{c}}$-proof $\boldsymbol{Q} : \Gamma \vdash \Delta \mid \mathcal{X}$ of the same length as $\boldsymbol{P}$. For the second step, we first build Γ^{a} and Δ^{a} as disjoint sequences of new variables (w.r.t. $\mathrm{FV}(\boldsymbol{Q})$) of the same length as Γ and Δ respectively. We let $V = \Gamma^{\mathrm{a}} \cup \Delta^{\mathrm{a}}$ and build the substitution ρ of domain V such that $\Gamma^{\mathrm{a}}\rho = \Gamma$ and $\Delta^{\mathrm{a}}\rho = \Delta$. Then Theorem 25 provides an abstract proof $\boldsymbol{Q}^{\mathrm{a}} : \Gamma^{\mathrm{a}} \vdash \Delta^{\mathrm{a}} \mid \mathcal{X}^{\mathrm{a}}$ and a substitution θ that is an extension of ρ such that $\boldsymbol{Q}^{\mathrm{a}}\theta = \boldsymbol{Q}$, hence $\Gamma^{\mathrm{a}}\theta = \Gamma$, $\Delta^{\mathrm{a}}\theta = \Delta$, $\Gamma^{\mathrm{a}} \cup \Delta^{\mathrm{a}} \subseteq \mathrm{Dom}(\theta)$ and $\boldsymbol{Q}^{\mathrm{a}}$ has the same length as $\boldsymbol{P}$. We may assume w.l.o.g. that $\mathrm{Dom}(\theta) \subseteq \mathrm{FV}(\boldsymbol{Q}^{\mathrm{a}})$. $\boldsymbol{Q}^{\mathrm{a}}$ is abstract hence is a fair $\mathcal{K}$-LK$^{\mathrm{c}}$-proof, and since $\boldsymbol{Q}^{\mathrm{a}}\theta$ is also a $\mathcal{K}$-LK$^{\mathrm{c}}$-proof then θ is a $\mathcal{K}$-substitution. Since $\boldsymbol{Q}^{\mathrm{a}}\theta$ is valid, θ is a solution of $\mathcal{X}^{\mathrm{a}}$ and therefore $(\mathcal{X}^{\mathrm{a}}\restriction, id, \theta{\downarrow_{\beta\eta}})$ is a fair controlled unification problem. By Proposition 39 an $\mathfrak{R}'$-normal form can be computed in finite time, and by Theorem 41 it must be some $(\top, \sigma, \theta')$ where θ' is an extension of $\theta{\downarrow_{\beta\eta}}$ and σ is a fair $\mathcal{K}$-substitution such that $\sigma \in \mathrm{sol}(\mathcal{X}^{\mathrm{a}})$ (hence $\boldsymbol{Q}^{\mathrm{a}}\sigma$ is fair and valid) and $\sigma \precsim \theta'$, which completes the third step. There is a substitution ρ' such that $\sigma\rho' \equiv_{\alpha\beta\eta} \theta'$, hence $\Gamma^{\mathrm{a}}\sigma\rho' \equiv_{\alpha\beta\eta} \Gamma^{\mathrm{a}}\theta' = \Gamma^{\mathrm{a}}\theta{\downarrow_{\beta\eta}} \equiv_{\alpha\beta\eta} \Gamma$, so that $\Gamma^{\mathrm{a}}\sigma \precsim \Gamma$ and similarly $\Delta^{\mathrm{a}}\sigma \precsim \Delta$. By Theorem 22 $\boldsymbol{Q}^{\mathrm{a}}\sigma$ can be translated into a standard and fair LK-proof $\boldsymbol{P}' : \Gamma' \vdash \Delta'$ of the same length as $\boldsymbol{P}$ such that $\Gamma' \vdash \Delta' \precsim \Gamma^{\mathrm{a}}\sigma \vdash \Delta^{\mathrm{a}}\sigma \precsim \Gamma \vdash \Delta$, which completes the fourth step. □

Example 43. It is easy to see that translating the proof $\boldsymbol{P}$ of Example 3 into a valid LK$^{\mathrm{c}}$-proof and then lifting this proof to an abstract proof yields a variant of $\boldsymbol{Q}^{\mathrm{a}}$ of Example 27 together with the substitution θ of Example 37. Translating the proof $\boldsymbol{Q}^{\mathrm{a}}\sigma_3$ into an LK-proof yields

$$\dfrac{\dfrac{\dfrac{}{(\xi_4\ (f_1\ z_1)\ (f_2\ z_1)) \vdash (\xi_4\ (f_1\ z_1)\ (f_2\ z_1))}\,(\text{Axiom})}{\forall x\,(\xi_4\ x\ (f_2\ z_1)) \vdash (\xi_4\ (f_1\ z_1)\ (f_2\ z_1))}\,(\forall\text{-L})}{\forall x\,(\xi_4\ x\ (f_2\ z_1)) \vdash \exists x\,(\xi_4\ (f_1\ x)\ (f_2\ x))}\,(\exists\text{-R})$$

Note that if p is replaced by the predicate $\simeq$ in $\boldsymbol{P}$ then the result is exactly the same, even though $\simeq$ is a logical symbol. The reason is that no equality rule is used in the proof and therefore the constant $\simeq$ does not occur in the constraint $\mathcal{Y}_3$, and therefore Rule $\rightarrow_7$ cannot be applied.

Similarly, translating the proof $\boldsymbol{Q}^{\mathrm{a}}\sigma'_3$ into an LK-proof yields

$$\dfrac{\dfrac{\dfrac{}{(\xi_4\ (f_3\ z_2)\,(f_4\ z_2)) \vdash (\xi_4\ (f_3\ z_2)\,(f_4\ z_2))}\,(\text{Axiom})}{\forall x\,(\xi_4\ (f_3\ x)\,(f_4\ x)) \vdash (\xi_4\ (f_3\ z_2)\,(f_4\ z_2))}\,(\forall\text{-L})}{\forall x\,(\xi_4\ (f_3\ x)\,(f_4\ x)) \vdash \exists x\,(\xi_4\ x\,(f_4\ z_2))}\,(\exists\text{-R})$$

which is another possible generalization of $\boldsymbol{P}$.

9 Conclusion

Most of the methods and results in this paper apply to other logics whose inference rules can be expressed with λ-terms, as long as the side conditions can themselves be expressed solely as constraints. This expression can be somewhat twisted as we have seen with the eigenvariable condition. The main point is of course to build a constrained inference system suitable for lifting any proof to an abstract proof and constraint.

One particularity of our version of LK is that formulæ can only be duplicated (from conclusion to premisses) by the contraction rules (C-L) and (C-R), and we have been careful in designing their constrained versions with *linear* premisses so that abstract proofs can always be obtained. Other versions of LK, especially those dispensing with the weakening rules, allow for duplications in binary rules. For instance, the ($\vee$-L) rule in [25] is:

$$\dfrac{\phi, \Gamma \vdash \Delta \quad \psi, \Gamma \vdash \Delta}{\phi \vee \psi, \Gamma \vdash \Delta}$$

where the formulæ in the end sequent are duplicated in each premiss. The constrained version of this rule cannot be

$$\dfrac{\phi, \Gamma \vdash \Delta \quad \psi, \Gamma \vdash \Delta}{\chi, \Gamma \vdash \Delta \mid \chi \doteq \phi \vee \psi}$$

because it would then be impossible to satisfy the split variable condition in abstract proofs. Hence a correct "liftable" constrained version of this rule could be

$$\dfrac{\phi, \Gamma \vdash \Delta \quad \psi, \Gamma' \vdash \Delta'}{\chi, \Gamma \vdash \Delta \mid \chi \doteq \phi \vee \psi \curlywedge \Gamma \doteq \Gamma' \curlywedge \Delta \doteq \Delta'}$$

where of course $\phi_1 \cdots \phi_n \doteq \psi_1 \cdots \psi_n$ stands for the constraint $\phi_1 \doteq \psi_1 \curlywedge \cdots \curlywedge \phi_n \doteq \psi_n$. Then the sequences Γ, Γ', Δ and Δ' can be disjoint sets of variables. It is therefore possible to transcribe our method to a version of LK where, except for the (P-L), (P-R) and cut rules, the premisses of an inference rule is always determined by its conclusion (as in [13]). As explained in Section 6 the structure of abstract proofs would consequently be simplified.

In order to simplify abstract proofs to the point that they become nothing more than proof skeletons, we need to treat LK as a schematic system in the sense of [21] and therefore treat the symbols $\Gamma, \Delta, \ldots$ as variables. An easy way to do this is to add two basic types, $*$ for sequences of formulæ and $\boldsymbol{s}$ for sequents, and three constants, the comma of type $\mathbf{o} \to * \to *$, the constant $\vdash$ of type $* \to * \to \boldsymbol{s}$ (both written in infix notation) and a constant of type $*$ for the empty sequence. The constrained version of the (P-L) rule would then be

$$\dfrac{s}{s' \mid s \doteq (\Gamma\ \phi, (\psi, \Sigma)) \vdash \Delta \curlywedge s' \doteq (\Gamma\ \psi, (\phi, \Sigma)) \vdash \Delta}\,(\text{P-L}^{\mathrm{c}})$$

where s, s' are variables of type $\boldsymbol{s}$, ϕ, ψ variables of type $\mathbf{o}$, Σ, Δ variables of type $*$ and Γ a variable of type $* \to *$. It is then obvious that an abstract proof is nothing more than a proof skeleton decorated with its resulting constraint (and then Proposition

29 becomes obvious). The reader has probably noticed that this permutation rule is not strictly equivalent to the original rule since it encompasses (when Γ is instantiated by a constant function) the identity inference, which is only derived in LK. We can see this as an extension of the general permutation rule, just as specifying a position for the permutation of formulæ is a restriction of the general rule. Such extensions could make proofs more general as they allow for more inferences at each rule.

As a rule of thumb we may presume that longer proofs mean more constraints and therefore less freedom for generalization, hence that only easy theorems (those with short proofs) are prone to generalization. It would be surprising that textbook theorems could be generalized in non-trivial ways; our method is probably more relevant to the context of interactive proofs of "real-world" conjectures. It is anyway sensible to adopt inference systems in which proofs are shorter.

One common way of shortening proofs is to use sequents in which the left and right parts of the turnstile are multisets of formulæ. This allows to dispense with the permutation rules (P-L) and (P-R) and therefore with the logistics of moving formulæ around in the right positions for the next move[8]. In this case, the comma in the rules should be interpreted as a multiset sum. The proof of Theorem 25 can then easily be adapted so that such LK-proofs can still be lifted to abstract proofs. However, there is a fundamental ambiguity in the process since any occurrence of a formula can be chosen as the principal formula (occurrence) of an inference. For instance, assuming the "weakening-less" axiom $\dfrac{}{\Gamma, \phi \vdash \Delta, \phi}$ and the cut rule we can build the following proof:

$$\dfrac{\dfrac{}{p \vdash p, q} \quad \dfrac{}{p, p \vdash p}}{p, p \vdash p, q}$$

There is an ambiguity as to which occurrence of p on the left of the sequent $p, p \vdash p$ is involved in the axiom and in the cut. If both involve the same occurrence, the abstract proof is

$$\dfrac{\dfrac{}{u \vdash v, w \mid u \doteq w} \quad \dfrac{}{u', v' \vdash w' \mid u' \doteq w'}}{u, v' \vdash w, w' \mid u \doteq w \curlywedge u' \doteq w' \curlywedge v \doteq u'}$$

If they involve distinct occurrences of p, then the abstract proof is

$$\dfrac{\dfrac{}{u \vdash v, w \mid u \doteq w} \quad \dfrac{}{u', v' \vdash w' \mid u' \doteq w'}}{u, v' \vdash w, w' \mid u \doteq w \curlywedge u' \doteq w' \curlywedge v \doteq v'}$$

and none of these abstract proofs is an instance of the other. In this case the lifting procedure is non-deterministic and Propositions 29, 30 and 31 do not hold. Of course the generalization algorithm can still be used (there is no guarantee that a most general proof can eventually be obtained anyway).

[8] Another way to do this is to allow principal formulæ to occur anywhere in the conclusions of the rules. For instance, the ($\neg$-L) rule would be $\dfrac{\Gamma, \Sigma \vdash \Delta, \phi}{\Gamma, \neg\phi, \Sigma \vdash \Delta}$.

9.1 Future work

The present work raises several interesting theoretical issues. The first one concerns the decidability of the satisfiability problem for LK^c-proofs: given a proof $\boldsymbol{P}$ in LK^c, is there a substitution σ such that $\boldsymbol{P}\sigma$ is valid? This amounts to checking whether the constraint $\mathcal{X}$ corresponding to $\boldsymbol{P}$ is satisfiable. It is easy to see that this problem is undecidable in general but particular cases deserve to be investigated, e.g., the case in which $\boldsymbol{P}$ is abstract (in the sense of Definition 23). Also, the possibility of computing a maximal solution for all LK^c-constraints is an open question (the fact that second order unification is undecidable is not sufficient to answer negatively since the generated constraints are more specific – for instance they are always satisfiable). The properties of the minimization algorithm need further investigations. From a more practical point of view, we also plan to implement the devised generalization procedure.

Acknowledgements: we thank the anonymous referees for their helpful comments and references.
Conflict of Interest: The authors declare that they have no conflict of interest.

References

[1] P. B. Andrews. Resolution in type theory. *The Journal of Symbolic Logic*, 36(3):414–432, 1971.

[2] M. Baaz and P. Wojtylak. Generalizing proofs in monadic languages. *Annals of Pure and Applied Logic*, 154(2):71 – 138, 2008.

[3] R. Caferra and N. Zabel. Building models by using tableaux extended by equational problems. *Journal of Logic and Computation*, 3:3–25, 1993.

[4] S. Cavagnetto. The lengths of proofs: Kreisel's conjecture and Gödel's speed-up theorem. *Journal of Mathematical Sciences*, 158(5):689–707, May 2009.

[5] G. Dowek. Higher-order unification and matching. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume II, chapter 16, pages 1009–1062. Elsevier Science, New York, 2001.

[6] G. Dowek, T. Hardin, and C. Kirchner. Theorem proving modulo. *J. Autom. Reason.*, 31(1):33–72, Oct. 2003.

[7] W. M. Farmer. A unification-theoretic method for investigating the k-provability problem. *Annals of Pure and Applied Logic*, 51(3):173–214, Mar. 1991.

[8] A. P. Felty and D. J. Howe. Generalization and reuse of tactic proofs. In F. Pfenning, editor, *LPAR*, volume 822 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 1994.

[9] M. Giese. Incremental Closure of Free Variable Tableaux. In R. Goré, A. Leitsch, and T. Nipkow, editors, *Proc. Intl. Joint Conf. on Automated Reasoning, Siena, Italy*, number 2083 in LNCS, pages 545–560. Springer-Verlag, 2001.

[10] M. Hagiya. A typed lambda-calculus for proving-by-example and bottom-up generalization procedure. *Theor. Comput. Sci.*, 137(1):3–23, 1995.

[11] S. Hetzl. A sequent calculus with implicit term representation. In A. Dawar and H. Veith, editors, *Computer Science Logic*, volume 6247 of *Lecture Notes in Computer Science*, pages 351–365. Springer Berlin Heidelberg, 2010.

[12] E. B. Johnsen and C. Lüth. Theorem reuse by proof term transformation. In K. Slind, A. Bunker, and G. Gopalakrishnan, editors, *TPHOLs*, volume 3223 of *Lecture Notes in Computer Science*, pages 152–167. Springer, 2004.

[13] J. Krajíček and P. Pudlák. The number of proof lines and the size of proofs in first order logic. *Archive of Mathematical Logic*, 27:69–84, 1988.

[14] D. Lugiez. Positive and negative results for higher-order disunification. *J. Symb. Comput.*, 20(4):431–470, 1995.

[15] E. Melis and J. Whittle. Analogy in inductive theorem proving. *J. Autom. Reasoning*, 22(2):117–147, 1999.

[16] D. Miller. *Proofs in Higher-Order Logic*. PhD thesis, Carnegie-Mellon University, 1983. Technical Report: MS-CIS-83-37.

[17] D. Miller. A compact representation of proofs. *Studia Logica*, 46(4):347–370, 1987.

[18] D. Miller. Unification under a mixed prefix. *Journal of Symbolic Computation*, 14(4):321 – 358, 1992.

[19] G. Moser and R. Zach. The epsilon calculus (tutorial). In M. Baaz and J. A. Makowsky, editors, *Computer Science Logic, 17th International Workshop, CSL 2003, 12th Annual Conference of the EACSL, and 8th Kurt Gödel Colloquium, KGC 2003, Vienna, Austria, August 25-30, 2003, Proceedings*, volume 2803 of *Lecture Notes in Computer Science*, page 455. Springer, 2003.

[20] G. Moser and R. Zach. The epsilon calculus and Herbrand complexity. *Studia Logica*, 82(1):133–155, 2006.

[21] R. J. Parikh. Some results on the length of proofs. *Transactions of the American Mathematical Society*, 177:29–36, Mar. 1973.

[22] N. Peltier. Pruning the search space and extracting more models in tableaux. *Logic Journal of the IGPL*, 7(2):217–251, 1999.

[23] F. Pfenning. Unification and anti-unification in the calculus of constructions. In *Logic in Computer Science, 1991. LICS '91., Proceedings of Sixth Annual IEEE Symposium on*, pages 74–85, July 1991.

[24] P. Rümmer. A constraint sequent calculus for first-order logic with linear integer arithmetic. In I. Cervesato, H. Veith, and A. Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning*, volume 5330 of *Lecture Notes in Computer Science*, pages 274–289. Springer Berlin Heidelberg, 2008.

[25] G. Takeuti. *Proof Theory*. North Holland, 2nd edition, 1987.

[26] C. Walther and T. Kolbe. Proving theorems by reuse. *Artif. Intell.*, 116(1-2):17–66, 2000.

[27] R. Zach. The practice of finitism: Epsilon calculus and consistency proofs in Hilbert's program. *Synthese*, 137(1-2):211–259, 2003.