On the elimination of quantifiers through descriptors in predicate logic*

Domenico Cantone[1], Marianna Nicolosi Asmundo[1], and Eugenio G. Omodeo[2]

[1] Dipartimento di Matematica e Informatica, Università di Catania
email: `cantone@dmi.unict.it, nicolosi@dmi.unict.it`
[2] Dipartimento di Matematica e Geoscienze, Università di Trieste
email: `eomodeo@units.it`

Abstract. We present a variant of the Davis–Fechter's technique for eliminating quantifiers in first-order logic, aimed at reducing the incidence of irrelevant dependencies in the construction of Skolem terms.
The basic idea behind this contribution is to treat as a single syntactic unit every maximal 'quantifier batch', i.e., group of contiguous alike quantifiers, whose internal order (which has no significance) is thereby prevented from entangling the final result.
Through concrete cross-translations, our version of the free-variable predicate calculus turns out to be equipollent—in means of expression and of proof—to the one originally proposed by Davis and Fechter and hence to a relatively conventional version of quantified first-order logic.

Keywords: Hilbert's ε-descriptor, global Skolemization, quantifier-elimination.

1 Introduction

Elimination of quantifiers from formulae of first-order logic is a process that has important implications for the automated deduction field [12, 2, 5] and in foundational issues [10, 13].

When no particular theory is focussed upon, quantifiers are usually eliminated by adopting as tool either Skolemization or Hilbert's ε-descriptor.

Traditionally, Skolemization and the ε-symbol have been exploited differently, in somewhat complementary rôles.

Skolemization is the most basic and widespread technique used to expunge existential quantifiers from automated proofs [12], one's rationale for effacing them being the relative ease with which Skolem terms (namely, terms whose lead functor is a Skolem symbol) can be manipulated in deductions. These can in fact be treated, both at the syntactic and at the semantic level, very much like the terms of the initial language.

Skolem terms are generally forgetful of the formulae from which they stem and of the context in which they are introduced. This may be of hindrance,

* Work partially supported by the INdAM/GNCS 2012 project *"Specifiche insiemistiche eseguibili e loro verifica formale"*.

though, because it is plausible that information potentially crucial for the automatic discovery of shorter proofs dwells in relationships among Skolem terms.

On the other hand, descriptions assembled by means of the ε-operator, the so called ε-terms, remain tied to the formulae which are the scopes of their descriptors: this is why they are so useful in investigations about foundational issues [10, 11], non-classical logics [8], and linguistics [14]. However, ε-terms are less well-suited for automation because of their elaborate structure.

In [6], Davis and Fechter propose a quantifier-elimination technique which bridges Skolemization with the ε-operator. It comes along with a free-variable calculus, provably equipollent to the standard predicate calculus; unlike in standard predicate calculus, in [6] quantifiers have the mere status of abbreviations, as they are reducible to suitable Skolem terms. These terms, offering a (functional) definition of the ε-terms, are constructed in such a way that structurally similar formulae share the same Skolem symbol. More precisely, Davis–Fechter's technique singles out certain formulae of the language, called *key formulae*, to each of which it associates a function symbol *uniquely*. Any other formula of the language owns a specific key formula, from which it inherits a Skolem symbol.

Through key formulae, [6] succeeds in providing a simultaneous translation of all formulae φ of a language $\mathcal{L}_\Sigma$ of standard predicate logic into free-variable formulae φ^* over a rich signature $\Sigma_{\mathbf{sko}}$. This signature results from a countable infinity of Skolem augmentations of the original signature Σ, and from an overall final amalgamation step. Thanks to the plenitude of the amalgamated language, the quantifiers of any input formula φ can get counterpart Skolem terms inside the translated formula φ^*, whose Skolem symbols are enacted to faithfully represent them in all regards by suitable axioms, named *ε-formulae*.

In this paper we present a variant of the Davis–Fechter's technique, allowing one to treat contiguous alike quantifiers as a single 'batch'. This version is based on a generalized notion of key formula, enabling one to associate *several* Skolem functions, independent of one another, with the same key formula. The resulting Skolem terms may be less deep and give rise to shorter proofs. We illustrate with an example the results of applying Skolemization, ϵ-operator, the Davis-Fechter technique, and our variant to a valid formula.

Our variant free-variable calculus can be proved equipollent to Davis and Fechter's original one, and hence to standard predicate logic.

2 Preliminaries

We recall here some basic notions, notation and terminology used in the paper.

Let $\mathcal{L}_\Sigma$ be the first-order language based on: a *signature* $\Sigma = (\mathcal{P}, \mathcal{F})$, where $\mathcal{P}$ (non-void) and $\mathcal{F}$ are finite or denumerable collections of predicate symbols and function symbols (each endowed of a fixed *arity*), respectively; a fixed countable collection *Vars* of individual variables; the propositional connectives (say & , $\vee$, $\neg$, and $\supset$, for specificity); the (infinitely many) quantifiers $\exists\, x$ and $\forall\, x$ corresponding to the individual variables $x \in \mathit{Vars}$.

Later on, various other first-order languages $\mathcal{L}_{\Sigma'}$ will enter into our discussion, whose formation rules will be the same as for the $\mathcal{L}_\Sigma$ just outlined. We will designate by $\mathcal{L}_{\Sigma'}^{\breve{}}$ the sublanguage resulting from each one of these by withdrawal of the quantifiers formation rule.

Throughout, we use the Greek letters $\varphi, \psi, \chi, \kappa$ as *metavariables* ranging over the formulae of $\mathcal{L}_\Sigma$; Q, P, R and g, f as metavariables ranging over predicate and function symbols, respectively; t, s, d, θ as metavariables ranging over first-order terms; and x, y, z as metavariables ranging over the individual variables.

Terms and formulae are constructed according to the standard rules; likewise, the notions of syntax tree of a *well-formed expression* (i.e., a term or a formula; in short, *wfe*), of atomic formulae, of literals, of (immediate) subformulae of a given formula, of free and bound variables,, of closed formulae (sentences), of lead symbol, and so on, are as usual. Precise definitions can be found in [9, 7].

Occurrences and positions. A *wfe* E *occurs within a wfe* F *at position* ν, where ν is a node in the syntax tree T_F of F, if the subtree of T_F rooted at the node ν is identical to the syntax tree of E. In such a case, we also say that the node ν is an *occurrence* of E (and also an occurrence of the lead symbol of E) in F and that the path from the root of T_F to ν is its OCCURRENCE PATH.

An occurrence of E within F can be conveniently coded by a sequence over the set $\mathbb{N}_+$ of all positive integers,[3] representing the positions within its siblings of each node in the occurrence path. Specifically, the set $Pos(F)$ of the POSITIONS in (the syntax tree of) any *wfe* F can be defined recursively as follows:

1. the empty word λ is in $Pos(F)$;
2. let f be a function symbol of arity n in $\mathcal{F}$, and $t_1, \ldots, t_n$ be terms: if $F = f(t_1, \ldots, t_n)$ and $\pi \in Pos(t_i)$, for some $i \in \{1, \ldots, n\}$, then $i \cdot \pi \in Pos(F)$;[4]
3. let R be a predicate symbol of arity n in $\mathcal{P}$, and $t_1, \ldots, t_n$ be terms: if $F = R(t_1, \ldots, t_n)$ and $\pi \in Pos(t_i)$, for some $i \in \{1, \ldots, n\}$, then $i \cdot \pi \in Pos(F)$;
4. if $F = \varphi_1 \circ \varphi_2$, where $\circ \in \{\ \&\ , \vee, \supset\}$, and $\pi \in Pos(\varphi_i)$, for some $i \in \{1, 2\}$, then $i \cdot \pi \in Pos(F)$;
5. if $F = \neg\psi$, or $F = (\exists x)\psi$, or $F = (\forall x)\psi$, and $\pi \in Pos(\psi)$, then $1 \cdot \pi \in Pos(F)$.

Given any *wfe* F, the occurrences at given positions of subformulae or subterms of F in F are determined as follows. We put $F|_\lambda = F$. In case F is a term $f(t_1, \ldots, t_n)$ or an atomic formula $R(t_1, \ldots, t_n)$, we put $F|_{i\cdot\pi} = t_i|_\pi$ for $i \in \{1, \ldots, n\}$. If $F = \varphi_1 \circ \varphi_2$, with $\circ \in \{\ \&\ , \vee, \supset\}$, we put $F|_{i\cdot\pi} = \varphi_i|_\pi$, for $i \in \{1, 2\}$. Finally, if $F = \neg\psi$, or $F = (\forall x)\psi$, or $F = (\exists x)\psi$, then $F|_{1\cdot\pi} = \psi|_\pi$. Thus the label of a node ν at position π in the syntax tree T_F of a *wfe* F (denoted $lbl(F, \pi)$) is the lead symbol of $F|_\pi$. For example, the positions in the formula $\varphi = P(x, y)\ \&\ Q(f(w, w))$ form the set $Pos(\varphi) = \{\lambda, 1, 2, 1{\cdot}1, 1{\cdot}2, 2{\cdot}1, 2{\cdot}1{\cdot}1, 2{\cdot}1{\cdot}2\}$. The occurrence of φ within itself is coded by λ. $\varphi|_1$ denotes $P(x, y)$, $\varphi|_{2\cdot1}$ denotes $f(w, w)$, and $\varphi|_{2\cdot1\cdot1}$, $\varphi|_{2\cdot1\cdot2}$ denote two different occurrences of the same term w.

[3] The set of all words over the 'alphabet' $\mathbb{N}_+$ will be designated $(\mathbb{N}_+)^*$.

[4] The expression $\pi_1 \cdot \pi_2$ denotes the concatenation of the sequences π_1 and π_2.

Given two positions $\pi_1, \pi_2 \in Pos(F)$ in a *wfe* F, we write $\pi_1 \sqsubseteq \pi_2$ to indicate that π_1 is a PREFIX of π_2, i.e. $\pi_2 = \pi_1 \cdot \eta$ for some $\eta \in (\mathbb{N}_+)^*$. Similarly, we write $\pi_1 \sqsubset \pi_2$ when π_1 is a PROPER PREFIX of π_2, i.e. $\pi_1 \sqsubseteq \pi_2$ and $\pi_1 \neq \pi_2$. If π_1 is a (proper) prefix of π_2, then $F|_{\pi_2}$ is a (proper) subexpression of $F|_{\pi_1}$.

We denote by Π_E^F the collection of all positions $\pi \in Pos(F)$ such that $F|_\pi = E$. If $\left|\Pi_E^F\right| = 1$, where $|S|$ denotes the cardinality of any set S, we may use π_E^F to denote the position of the unique occurrence of E in F. We write Π_E and π_E in case the corresponding F is clear from the context. For a collection T of *wfes*, we put $\Pi_T^F =_{\mathrm{Def}} \bigcup_{E \in T} \Pi_E^F$.

It is possible to establish a lexicographic ordering $\prec$ over the set $Pos(\varphi)$ of positions in a formula φ such that for any $\pi_1, \pi_2 \in Pos(\varphi)$, if $\pi_1 \sqsubset \pi_2$, then $\pi_1 \prec \pi_2$; if $\pi \cdot n \sqsubseteq \pi_1$ and $\pi \cdot n' \sqsubseteq \pi_2$, for some $\pi \in (\mathbb{N}_+)^*$ and $n < n'$, then $\pi_1 \prec \pi_2$. By the syntactic structure of formulae and terms, $\prec$ is a well-ordering. Therefore, we can define an operation 'min' which selects from any nonempty subset X of $Pos(\varphi)$ the minimum of X relative to the ordering $\prec$. We also write $\pi_1 \preceq \pi_2$ to indicate that either $\pi_1 \prec \pi_2$ or $\pi_1 = \pi_2$.

Let φ be a formula and let us suppose, without loss of generality, that φ contains as propositional connectives only $\neg$, $\&$, and $\vee$. An occurrence ν of a *wfe* E within a formula φ is *positive* if the negation symbol $\neg$ occurs an even number of times in the occurrence path of ν deprived of its last component. Otherwise, the occurrence is said to be *negative*.

Let F be a *wfe*, π a position in F, and E a *wfe* of the same type as $F|_\pi$ (that is, a formula if $F|_\pi$ is a formula, and a term otherwise). We indicate with $F[\pi/E]$ the *wfe* obtained by substituting the occurrence of $F|_\pi$ in F at position π by E, so that we have $F[\pi/E]|_\pi = E$.

Given two *wfes* E and F, we write $F = F(E)$ to stress that the occurrences of E in F play a significant rôle. Thus, for instance, if E' is another *wfe* of the same type as E, by $F(E')$ we denote the *wfe* resulting from F when each occurrence of E within F is replaced by a copy of E'.

For any formula φ in the language $\mathcal{L}_\Sigma$, the collection of all variables that occur free in φ is denoted by $Free(\varphi)$, whereas the collection of variables that occur bound in φ is denoted by $Bound(\varphi)$. We further denote by $Vars(\varphi)$ the set of *all* variables appearing in φ.

Substitutions. A *(variable-)*SUBSTITUTION is a mapping $x \overset{\sigma}{\mapsto} x\sigma$ from *Vars* to the collection of all terms over $(\Sigma, Vars)$ such that $x\sigma = x$ holds for all but a finite number of variables x. We indicate with $\mathrm{Supp}(\sigma)$ the SUPPORT of σ, namely the collection of all $x \in Vars$ such that $x\sigma \neq x$. Also, given the terms $t_1, \ldots, t_n$, we denote by $\{x_1/t_1, \ldots, x_n/t_n\}$ the substitution σ such that $\mathrm{Supp}(\sigma) \subseteq \{x_1, \ldots, x_n\}$ and $x_i\sigma = t_i$, for $i = 1, \ldots, n$, where $x_1, \ldots, x_n$ are distinct variables.

A substitution is a VARIABLE RENAMING if it has the form $\{x_1/y_1, \ldots, x_n/y_n\}$, with $y_1, \ldots, y_n$ pairwise distinct variables. If $\sigma = \{x_1/y_1, \ldots, x_n/y_n\}$ is a variable renaming, $\{y_1/x_1, \ldots, y_n/x_n\}$ is the INVERSE SUBSTITUTION of σ.

Given a substitution σ and a set of variables $V \subseteq Vars$, the restriction of σ to V is the substitution $\sigma_{|V}$ s.t. $x\sigma_{|V} =_{\mathrm{Def}} x\sigma$ if $x \in V$, $x\sigma_{|V} =_{\mathrm{Def}} x$ otherwise.

The action of a substitution can be extended into a mapping $E \mapsto E\sigma$, recursively defined over the *wfes* of $\mathcal{L}_\Sigma$ as usual. A substitution σ is *free* for a formula φ, if φ and $\varphi\sigma$ have exactly the same occurrences of bound variables.

3 A variant of Davis–Fechter's language

Davis and Fechter [6] proposed a way of eliminating bound variables from first-order predicate logic by successive enlargements of the signature of the language, based on the well-known Skolemization technique, which is wholesale applied to a denumerable infinity of so-called *key formulae*. In what follows, we will present a variant of Davis–Fechter's calculus, where by resorting to a revised definition of key formula, we can lower the complexity of the quantifier-elimination process.

The crucial remark inspiring our proposal is that in standard first-order logic contiguous alike quantifiers can be treated as a single syntactic unit: a 'batch'. E.g., in view of their logical equivalence, the formulae $\varphi_1 = (\exists x)(\exists y)Q(x, y)$ and $\varphi_2 = (\exists y)(\exists x)Q(x, y)$ can be replaced by $(\exists x, y)Q(x, y)$, so as to make the independency between x and y explicit, and to stress the immateriality of the order of the quantifiers $(\exists x)$ and $(\exists y)$ occurring in φ_1 and φ_2.

This concealment of syntactic features of only apparent significance is particularly rewarding when Skolemization comes into play, because the reduction of irrelevant dependencies among quantified variables leads to the construction of less intricate Skolem terms, as well as to shorter proofs [2].

Skolemization of formulae adopting *generalized* quantifiers like $(\exists x_1, \ldots, x_n)$ can be defined in the flavour of [12] as follows.

Definition 1 (Generalized Skolemization).
Let χ be a formula and let $\chi|_\pi$ be a positive (resp., negative) occurrence of $(\exists x_1, \ldots, x_n)\varphi(x_1, \ldots, x_n)$ (resp., $(\forall x_1, \ldots, x_n)\varphi(x_1, \ldots, x_n)$) in χ, with π position of $Pos(\chi)$. Moreover, let $y_1, \ldots, y_m$ be the free variables in $\chi|_\pi$, $h_1, \ldots, h_n$ new m-ary function symbols, and $\varphi(x_1/h_1(y_1, \ldots, y_m), \ldots, x_n/h_n(y_1, \ldots, y_m))$ the formula resulting from the replacement in φ of each x_i by the corresponding $h_i(y_1, \ldots, y_m)$. Then, $\chi[\pi/\varphi(x_1/h_1(y_1, \ldots, y_m), \ldots, x_n/h_n(y_1, \ldots, y_m))]$ is obtained from χ by a Skolemization step.

Skolemization of a formula is carried out by repeatedly performing Skolemization steps till all the existential (resp., universal) quantifiers occurring positively (resp., negatively) in the formula are eliminated. While performing Skolemization of a formula through an inside-out visit, one may sometimes associate with the innermost quantifier a function symbol whose arity exceeds the need; in fact, unless the quantifiers forming a single batch are treated simultaneously, one may fail to see at the inner level that certain variables are bound at an outer level. For example, the above-displayed formulae φ_1 and φ_2 could be brought to quantifier-free form through Skolemization in the following way (constants being regarded as 0–ary function symbols): φ_1 becomes $Q(f_x(), f_y(f_x()))$ and φ_2 becomes $Q(g_x(g_y()), g_y())$. These ground sentences would actually result from the

Skolemization technique proposed by Davis and Fechter; an improved Skolemization, treating x and y simultaneously and on a par, and disregarding distinctions between φ_1 and φ_2, would produce the simpler result $Q(\ell_x(), \ell_y())$. Def. 1 authorizes us to proceed this way, thereby leading to terms that are less deep and reflect more closely the genuine dependencies across quantified variables.

We will manage things so that whenever a new function symbol arises from Skolemization, it gets drawn from an infinite repository $\mathcal{F}_{\mathbf{sko}}$, disjoint from $\mathcal{F}$ and consisting of countably many distinct function symbols of any arity. The augmented signature, to be designated by $\Sigma_{\mathbf{sko}} = (\mathcal{P}, \mathcal{F} \cup \mathcal{F}_{\mathbf{sko}})$, will originate from the *quantifier-free* language $\mathcal{L}^{\smile}_{\Sigma}$ through successive enlargement steps culminating in the language $\mathcal{L}^{\smile}_{\Sigma_{\mathbf{sko}}}$.

3.1 Generalized key formulae

We will now single out a collection of special '*key*' formulae, which will serve us in the planned formation of $\mathcal{F}_{\mathbf{sko}}$. Our policy will be to associate Skolem function symbols *exclusively* to these formulae, so as to avoid excessive proliferation of symbols in $\mathcal{F}_{\mathbf{sko}}$. Key formulae must hence have a very restrained syntactic form: among others, they will be devoid of quantifiers; moreover, we will class the variables of each key formula κ into two disjoint groups:

- 'key ' variables, whose overall number n indicates how many function symbols $h_1 = h_{\kappa,1}, \dots, h_n = h_{\kappa,n}$ are associated to κ, and which (in analogy to the existential variables x_i of Def. 1) indicate where to 'graft' the corresponding Skolem terms $h_i(\underbrace{\_, \dots, \_}_{m})$; and
- 'anonymous' variables, occurring only once inside κ, whose overall number m will determine the common arity of the Skolem symbols h_i.

We will also specify at the end of this section how to extract from any given formula φ its key formula, from which φ will inherit its Skolem symbols and whose anonymous variables must be replaced, in order to get φ, by terms acting—so-to-speak—as the 'actual parameters' of φ.

For this purpose it is convenient to assume that the individual variables *Vars* are arranged in a sequence $\langle \dots, \mathsf{x}_{-2}, \mathsf{x}_{-1}, \mathsf{x}_0, \mathsf{x}_1, \mathsf{x}_2, \mathsf{x}_3, \dots \rangle$, and that the two subsequences $\mathit{Vars}^- = \langle \dots, \mathsf{x}_{-2}, \mathsf{x}_{-1}, \mathsf{x}_0 \rangle$, and $\mathit{Vars}^+ = \langle \mathsf{x}_1, \mathsf{x}_2, \mathsf{x}_3, \dots \rangle$ are adopted to denote key variables and anonymous variables, respectively. When it is not necessary to insist on such a convention, we will just use the meta-variables x, y, z (possibly subscripted with natural numbers) that, as mentioned above, stand for generic variables in *Vars*.

Most of the notions introduced in Sec. 2 for the languages of standard predicate logic can be referred to *quantifier-free* languages as well. However, in absence of bound variables, $\mathit{Free}(\psi) = \mathit{Vars}(\psi)$ holds for every formula ψ.

Since variables in Vars^- play a key rôle which will be highlighted in the rest of this section, we define the DEGREE $\mathit{deg}(\varphi)$ of a formula φ to be the cardinality $\left| \mathit{Vars}(\varphi) \cap \mathit{Vars}^- \right|$.

As mentioned earlier, in standard first-order logic a substitution σ is free for a formula ψ if it does not change the occurrences of bound variables in ψ. Since quantified variables are not subject to substitutions, one only has to make sure that when a variable x is replaced by a term, no variable y of the term is such that x occurs free within the scope of a quantifier binding y within ψ.

An analogous concept can be given in the free-variable language, by introducing the notion of *substitution σ free for ψ relative to a set $B \subseteq Vars$*, where the variables in B are treated as bound variables. More precisely, we have the following definition.

Definition 2. *Given a quantifier-free formula ψ, a substitution σ is said to be* FREE FOR ψ RELATIVE TO $B \subseteq Vars$ *if* $\mathrm{Supp}(\sigma) \cap B = \emptyset$ *and for each $x \in \mathrm{Supp}(\sigma) \cap Vars(\psi)$ the term $x\sigma$ contains none of the variables in B.*

Next we give the notion of *most general formula* relative to a set of variables.

Definition 3. *Let ψ and φ be quantifier-free formulae and let $B \subseteq Vars$. We say that ψ is* MORE GENERAL THAN φ RELATIVE TO B, *if $\varphi = \psi\sigma$ for some substitution σ free for ψ relative to B.*

ψ is MOST GENERAL RELATIVE TO B *if whenever a formula ψ' is more general than ψ relative to B, then ψ is also more general than ψ' relative to B.*

Most general formulae can be characterized in merely syntactical terms as stated by the next lemma, whose proof can be found in [4].

Lemma 1. *A formula ψ is most general relative to a set of variables B if and only if it satisfies the conditions:*

1. *any variable in $Vars(\psi) \setminus B$ has exactly one occurrence in ψ;*
2. *any term occurring in ψ either contains a variable of B or is itself a variable, not necessarily in B.*

It can be checked that when two formulae ψ and φ are most general relative to a common set of variables B, and each one is more general than the other (relative to B), then they solely differ by a renaming of their variables not belonging to B. The latter remark yields a criterion for grouping most general formulae into equivalence classes. To set also a criterion for choosing a representative member from each of such classes, we give the following definition.

Definition 4. *A formula κ is a* KEY FORMULA *if the following holds:*

1. *κ is most general relative to $\{\mathsf{x}_0, \mathsf{x}_{-1}, \dots, \mathsf{x}_{1-n}\} \subseteq Vars(\kappa)$, for some $n \geqslant 1$, and $\min(\Pi_{\mathsf{x}_0}) \prec \dots \prec \min(\Pi_{\mathsf{x}_{1-n}})$;*
2. *$Vars(\kappa) \setminus \{\mathsf{x}_0, \dots, \mathsf{x}_{1-n}\} = \{\mathsf{x}_1, \dots, \mathsf{x}_m\}$, for some $m \geqslant 0$, $|\Pi_{\mathsf{x}_i}| = 1$, for $i = 1, \dots, m$, and $\pi_{\mathsf{x}_1} \prec \dots \prec \pi_{\mathsf{x}_m}$.*

Consider the formula $\varphi_1 = P(\mathsf{x}_0, \mathsf{x}_{-1}, \mathsf{x}_1, \mathsf{x}_2)$. Clearly it satisfies points 1 and 2 of Def. 4 and, therefore, it is a key formula. On the other hand, $\varphi_2 = P(\mathsf{x}_0, \mathsf{x}_{-1}, \mathsf{x}_1, \mathsf{x}_1)$ and $\varphi_3 = P(\mathsf{x}_0, \mathsf{x}_{-1}, f(\mathsf{x}_1), \mathsf{x}_1)$ are not key formulae because they are not most general relative to $\{\mathsf{x}_0, \mathsf{x}_{-1}\}$ and they do not satisfy point 2 of

Def. 4. Note that our key formulae of degree $n = 1$ coincide with key formulae as defined in [6]. Now that key formulae—the only formulae associated with Skolem symbols—have been characterized, we need to relate them to the other formulae of the language in order to be able to Skolemize any quantified formula.

The idea is that any formula $(\exists x_1, \ldots, x_k)\varphi$ of standard predicate logic can be related to a key formula ψ and borrow its Skolem symbols in order to determine Skolem terms suitably replacing the quantified variables $x_1, \ldots, x_k$. Next we show how to relate key formulae with other formulae of the same language. Specifically, we show how, given a quantifier-free formula φ and a set of terms T (in principle we can consider any type of term, but for our purposes we will focus on variables and Skolem terms) we are able to effectively determine a key formula ψ and a substitution σ such that $\varphi = \psi\sigma$. This result turns out to be useful several times in this paper: for example in Sec. 3.2, where we define quantifier batches, and in Sec. 4, where we introduce translation maps from standard predicate logic to the considered quantifier-free predicate logics.

Let φ be a quantifier-free formula of a first-order language $\mathcal{L}_\Sigma$ and let T be a nonempty set of terms occurring in φ. We start by singling out the collection $\mathcal{K}_T^\varphi$ of the positions of the $\sqsubseteq$-minimal occurrences of the terms of T in (the syntax tree T_φ) of φ. More precisely, $\mathcal{K}_T^\varphi$ is the collection of the $\sqsubseteq$-minimal positions in Π_T^φ, i.e., $\mathcal{K}_T^\varphi = \{\pi \in \Pi_T^\varphi \mid \text{there is no } \pi' \in \Pi_T^\varphi \text{ such that } \pi' \sqsubset \pi\}$. In the following, terms at such positions will be substituted with key variables whose indices depend on a rank function $\iota : \mathcal{K}_T^\varphi \to \mathbb{N}$ (which we will soon extend to other positions of φ as well), where, for $\pi \in \mathcal{K}_T^\varphi$, $\iota(\pi)$ is the number of distinct terms in T having some $\sqsubseteq$-minimal occurrence which precede, with respect to $\preceq$, the first occurrence of the term $\varphi_{|\pi}$; in symbols, $\iota(\pi) =_{\mathrm{Def}} \left|\left\{\varphi_{|\pi'} : \pi' \in \mathcal{K}_T^\varphi \mid \pi' \preceq \pi^*\right\}\right|$, where $\pi^* =_{\mathrm{Def}} \min\{\pi'' \in \mathcal{K}_T^\varphi \mid \varphi_{|\pi''} = \varphi_{|\pi}\}$. We put $t_{\iota(\pi)} =_{\mathrm{Def}} \varphi_{|\pi}$; then our understanding is that the term $t_{\iota(\pi)}$ at position $\pi \in \mathcal{K}_T^\varphi$ in φ is to be substituted with the key variable $\mathsf{x}_{1-\iota(\pi)}$. Let $n = \left|\{\varphi_{|\pi} : \pi \in \mathcal{K}_T^\varphi\}\right|$. Then we plainly have $\{t_1, \ldots, t_n\} \subseteq T$. When $\{t_1, \ldots, t_n\} = T$, we say that T is INDEPENDENT w.r.t. φ; in such a case, any term $t \in T$ must have at least one occurrence in φ which is not contained in any occurrence in φ of any other term $t' \in T$.

Next, let $\mathcal{A}_T^\varphi$ be the set of the $\sqsubseteq$-minimal positions of subterms in φ which are not $\sqsubseteq$-comparable with any of the positions in Π_T^φ, where two positions π and π' in φ are $\sqsubseteq$-comparable if either $\pi \sqsubseteq \pi'$ or $\pi' \sqsubseteq \pi$ holds, and let $m = |\mathcal{A}_T^\varphi|$. Plainly, $\mathcal{A}_T^\varphi \cap \mathcal{K}_T^\varphi = \emptyset$. We extend the rank function ι on any $\pi \in \mathcal{A}_T^\varphi$ by putting $\iota(\pi) =_{\mathrm{Def}} \left|\{\pi' \in \mathcal{A}_T^\varphi \mid \pi' \preceq \pi\}\right|$, i.e., $\iota(\pi)$ is the rank of π, within $\mathcal{A}_T^\varphi$, in a left-to-right scan of the syntax tree of φ. Much as before, we also put $t'_{\iota(\pi)} =_{\mathrm{Def}} \varphi_{|\pi}$, for $\pi \in \mathcal{A}_T^\varphi$. Then our understanding is that the term $t'_{\iota(\pi)}$ at position π in φ is to be substituted by the anonymous variable $\mathsf{x}_{\iota(\pi)}$.

Thus, let ψ be the formula obtained by simultaneously substituting in φ each term $t_{\iota(\pi)}$ at position $\pi \in \mathcal{K}_T^\varphi$ in φ with the key variable $\mathsf{x}_{1-\iota(\pi)}$ and each term $t'_{\iota(\pi')}$ at position $\pi' \in \mathcal{A}_T^\varphi$ in φ with the anonymous variable $\mathsf{x}_{\iota(\pi)}$, i.e., $\psi =_{\mathrm{Def}} \varphi[\pi/\mathsf{x}_{1-\iota(\pi)}]_{\pi \in \mathcal{K}_T^\varphi}[\pi/\mathsf{x}_{\iota(\pi')}]_{\pi' \in \mathcal{A}_T^\varphi}$. It can easily be checked that the positions in ψ containing variable occurrences are exactly those in $\mathcal{K}_T^\varphi \cup \mathcal{A}_T^\varphi$.

More precisely, we have

$$\mathit{\Pi}^{\psi}_{\mathsf{x}_{1-\iota(\pi)}} = \mathit{\Pi}^{\varphi}_{t_{\iota(\pi)}} \cap \mathcal{K}^{\varphi}_{T}\,, \quad \text{for } \pi \in \mathcal{K}^{\varphi}_{T}\,, \tag{1}$$

$$\mathit{\Pi}^{\psi}_{\mathsf{x}_{\iota(\pi')}} = \{\pi'\}\,, \qquad\qquad \text{for } \pi' \in \mathcal{A}^{\varphi}_{T}\,. \tag{2}$$

In view of Lemma 1 and the definition of the rank function ι over the positions in $\mathcal{K}^{\varphi}_{T}$, (1) and (2) imply that

$$\psi \text{ is a key formula of degree } n = \left|\{\varphi|_{\pi} : \pi \in \mathcal{K}^{\varphi}_{T}\}\right|\,,[5] \tag{3}$$

which we call the KEY FORMULA OF φ RELATIVE TO T and denote by $\mathit{Key}(\varphi, T)$.

Additionally, if we put $\sigma =_{\mathrm{Def}} \{\mathsf{x}_0/t_1, \mathsf{x}_{-1}/t_2, \ldots, \mathsf{x}_{1-n}/t_n, \mathsf{x}_1/t'_1, \ldots, \mathsf{x}_m/t'_m\}$, then, by construction, we have immediately that

$$\varphi = \psi\sigma\,. \tag{4}$$

Conditions (1)-(4) characterize the key formula of φ relative to T. In fact, it can easily be shown that if ψ' is a key formula of degree $\left|\{\varphi|_{\pi} : \pi \in \mathcal{K}^{\varphi}_{T}\}\right|$ such that

$$\begin{aligned} \mathit{\Pi}^{\psi'}_{\mathsf{x}_{1-\iota(\pi)}} &= \mathit{\Pi}^{\varphi}_{t_{\iota(\pi)}} \cap \mathcal{K}^{\varphi}_{T}\,, && \text{for } \pi \in \mathcal{K}^{\varphi}_{T}\,, \text{ and} \\ \mathit{\Pi}^{\psi'}_{\mathsf{x}_{\iota(\pi')}} &= \{\pi'\}\,, && \text{for } \pi' \in \mathcal{A}^{\varphi}_{T}\,. \end{aligned}$$

and σ' is a substitution such that $\varphi = \psi'\sigma'$, then ψ' is the key formula of φ relative to T, i.e., ψ' coincides with $\mathit{Key}(\varphi, T)$.

3.2 Definition of quantifier batches

We are now ready to define quantifiers in a way which generalizes their definition as given in [6]. When viewed as a single syntactic unit $(Q\,x_1, \ldots, x_n)$, a tuple $Q\,x_1 \cdots Q\,x_n$ of consecutive alike quantifiers (where $Q \in \{\exists, \forall\}$) will be called a QUANTIFIER BATCH.

Let φ be a quantifier-free formula and let $\{x_1, \ldots, x_n\} \subseteq \mathit{Vars}(\varphi)$. The quantified forms $(\exists x_1, \ldots, x_n)\varphi$ and $(\forall x_1, \ldots, x_n)\varphi$ are rendered in the quantifier-free language as follows.

Let $\psi(\mathsf{x}_0, \ldots, \mathsf{x}_{1-n},\, \mathsf{x}_1, \ldots, \mathsf{x}_m) =_{\mathrm{Def}} \mathit{Key}(\varphi, \{x_1, \ldots, x_n\})$ be the key formula of φ relative to $\{x_1, \ldots, x_n\}$ such that $\mathit{Vars}(\psi) = \{\mathsf{x}_0, \ldots, \mathsf{x}_{1-n},\, \mathsf{x}_1, \ldots, \mathsf{x}_m\}$, and let σ be a companion substitution such that $\varphi = \psi\sigma$. We define the quantifiers $\exists$ and $\forall$ as the following shorthands

$$\begin{aligned} (\exists x_1, \ldots, x_n)\varphi &\equiv_{\mathrm{Def}} \psi(h_{\psi,1}(\mathsf{x}_1, \ldots, \mathsf{x}_m)\sigma, \ldots, h_{\psi,n}(\mathsf{x}_1, \ldots, \mathsf{x}_m)\sigma, \mathsf{x}_1\sigma, \ldots, \mathsf{x}_m\sigma), \\ (\forall x_1, \ldots, x_n)\varphi &\equiv_{\mathrm{Def}} \psi(h_{\neg\psi,1}(\mathsf{x}_1, \ldots, \mathsf{x}_m)\sigma, \ldots, h_{\neg\psi,n}(\mathsf{x}_1, \ldots, \mathsf{x}_m)\sigma, \mathsf{x}_1\sigma, \ldots, \mathsf{x}_m\sigma), \end{aligned}$$

where the $h_{\psi,i}$'s and the $h_{\neg\psi,i}$'s are the distinct symbols uniquely associated with ψ and $\neg\psi$, respectively, as hinted at the beginning of Sec. 3.1 (further clarifications are postponed to Sec. 3.3).

[5] In particular, when T is independent w.r.t. φ, we have $n = |T|$.

Let us momentarily denote by $\varphi_{\exists}$ the *definiens* (just given) of $(\exists x_1, \ldots, x_n)\varphi$ and by $\varphi_{\forall}$ the analogous *definiens* of $(\forall x_1, \ldots, x_n)\varphi$; so we can say that the quantifier batches $(\exists x_1, \ldots, x_n)$ and $(\forall x_1, \ldots, x_n)$ appearing in the original φ have 'concretized' into the n-tuples

- $\langle h_{\psi,1}(\mathsf{x}_1, \ldots, \mathsf{x}_m)\sigma, \ldots, h_{\psi,n}(\mathsf{x}_1, \ldots, \mathsf{x}_m)\sigma\rangle$ and
- $\langle h_{\neg\psi,1}(\mathsf{x}_1, \ldots, \mathsf{x}_m)\sigma, \ldots, h_{\neg\psi,n}(\mathsf{x}_1, \ldots, \mathsf{x}_m)\sigma\rangle$

of Skolem terms appearing in $\varphi_{\exists}$ and in $\varphi_{\forall}$, respectively. Looking at each Skolem term more closely, we can say that the term $h_{\psi,j}(\mathsf{x}_1, \ldots, \mathsf{x}_m)\sigma$ concretizes in $\varphi_{\exists}$ that one component x_k of $(\exists x_1, \ldots, x_n)$ for which $\mathsf{x}_{1-j}\sigma = x_k$ holds. Likewise, the Skolem term $h_{\neg\psi,j}(\mathsf{x}_1, \ldots, \mathsf{x}_m)\sigma$ concretizes in $\varphi_{\forall}$ the component x_k of $(\forall x_1, \ldots, x_n)$ for which $\mathsf{x}_{1-j}\sigma = x_k$ holds.

Example 1. The free-variable rendering of $(\exists x, y)Q(x, y)$ is $Q(h_{Q,1}(), h_{Q,2}())$, with $h_{Q,1}()$ and $h_{Q,2}()$ the 0-ary Skolem functions associated with the key formula $Q(\mathsf{x}_0, \mathsf{x}_{-1})$ of degree 2 of $Q(x, y)$ relative to $T = \{x, y\}$. Compare this with the quantifier-free translations of φ_1 and φ_2 discussed after Def. 1: $h_{Q,1}()$ and $h_{Q,2}()$ match the third and best rendering of $(\exists x, y)Q(x, y)$ examined there.

When ψ is a key formula of degree 1, we often prefer to denote its corresponding Skolem function symbol with h_{ψ} rather than with $h_{\psi,1}$.

3.3 The augmented signatures and their amalgamation

In the light of the notions introduced in Sec. 3, we are ready to define the completed signature $\Sigma_{\mathbf{sko}}$, on which our quantifier-free language $\mathcal{L}_{\Sigma_{\mathbf{sko}}}^{\smile}$ is based.

Starting with the initial signature $\Sigma = (\mathcal{P}, \mathcal{F})$ of our language, we define the following hierarchy of augmented signatures:

- $\mathcal{F}_0 =_{\mathrm{Def}} \mathcal{F}$ and $\Sigma_0 =_{\mathrm{Def}} \Sigma$.
- For $i \geqslant 1$,
  - $\mathcal{F}_i$ is the set comprising exactly one symbol $h_{\psi,j}$ new to Σ_{i-1}, for each pair (ψ, j) with $1 \leqslant j \leqslant n$ and ψ a key formula of $\mathcal{L}_{\Sigma_{i-1}}^{\smile}$ of degree $n \geqslant 1$ not belonging to any $\mathcal{L}_{\Sigma_{i'}}^{\smile}$ for which $i' < i - 1$. The arity of $h_{\psi,j}$ equals $|\mathit{Vars}(\psi)| - n$;
  - $\Sigma_i =_{\mathrm{Def}} (\mathcal{P}, \mathcal{F}_i)$.
- To end, put $\mathcal{F}_{\mathbf{sko}} =_{\mathrm{Def}} \bigcup_{i=1}^{\infty} \mathcal{F}_i$ and $\Sigma_{\mathbf{sko}} =_{\mathrm{Def}} (\mathcal{P}, \bigcup_{i=0}^{\infty} \mathcal{F}_i)$.

We also define the notion of rank of a *wfe* of $\mathcal{L}_{\Sigma_{\mathbf{sko}}}^{\smile}$, $\mathit{Skrank}(\cdot)$, as follows:

$$\mathit{Skrank}(E) =_{Def} \min\{k \in \mathbb{N} : E \text{ is in the language } \mathcal{L}_{\Sigma_k}^{\smile}\}.$$

4 Cross-translation tools

In considering now the quantified version $\mathcal{L}_{\Sigma_{\mathbf{sko}}}$ of the $\mathcal{L}_{\Sigma_{\mathbf{sko}}}^{\smile}$ constructed so far, we will assume, for the sake of simplicity and without loss of generality, that all quantifiers are maximally generalized in the sense that $\mathcal{L}_{\Sigma_{\mathbf{sko}}}$ contains no formulae like $(\exists x, y)(\exists z)P(x, y, z)$, but only formulae like $(\exists x, y, z)P(x, y, z)$.

First translation of the standard predicate language $\mathcal{L}_{\Sigma_{\mathbf{sko}}}$ into the free-variable language $\mathcal{L}_{\Sigma_{\mathbf{sko}}}^{\smile}$. In [6], Davis and Fechter defined a translation map $(\cdot)^* : \mathcal{L}_{\Sigma_{\mathbf{sko}}} \to \mathcal{L}_{\Sigma_{\mathbf{sko}}}^{\smile}$, leading from standard predicate logic to free-variable languages via the exploitation of key formulae of degree 1 for defining quantifiers. The translation map $(\cdot)^*$ preserves the structure of *wfe*s and acts only on formulae $\chi = (Qx_1, \ldots, x_n)\varphi$ with $Q \in \{\exists, \forall\}$. It is recursively defined as follows:

$$\chi^* = \begin{cases} \varphi^*\{x_n/h_{\pm\psi_n}(\overrightarrow{s_n})\} & \text{if } n = 1, \\ \Big((Qx_1, \ldots, x_{n-1})\varphi^*\{x_n/h_{\pm\psi_n}(\overrightarrow{s_n})\}\Big)^* & \text{otherwise,} \end{cases}$$

where

- $\psi_n = \mathit{Key}(\varphi^*, \{x_n\})$,
- $\pm\psi_n =$ **if** $(Q = \exists)$ **then** ψ_n **else** $\neg\psi_n$ **fi**,
- σ_n is such that $\varphi^* = \psi_n\sigma_n$,
- $\overrightarrow{s_n} = \overrightarrow{\mathsf{x}_n}\sigma_n$,
- $\overrightarrow{\mathsf{x}_n} = \langle \mathsf{x}_1, \ldots, \mathsf{x}_{m_n}\rangle$, with $\mathsf{x}_1, \ldots, \mathsf{x}_{m_n}$ the variables in $\mathit{Vars}(\psi_n) \setminus \{\mathsf{x}_0\}$.

We indicate with $[\mathcal{L}_{\Sigma_{\mathbf{sko}}}^{\smile}]^*$ the set of the images of the map $(\cdot)^*$. The following example illustrates how a formula of standard predicate logic can be translated into quantifier-free form via $(\cdot)^*$.

Example 2. Let $\chi = (\exists x, y, z)P(x, y, z)$ and put $\varphi_4 = P(x, y, z)$. Then by a first recursion step we obtain

$$\chi^* = \Big((\exists x, y)\varphi_4^*\{z/h_{\psi_3}(\overrightarrow{s_3})\}\Big)^* = \Big((\exists x, y)P(x, y, h_{\psi_3}(x, y))\Big)^*,$$

where $\psi_3 = \mathit{Key}(\varphi_4, \{z\}) = P(\mathsf{x}_1, \mathsf{x}_2, \mathsf{x}_0)$, $\overrightarrow{s_3} = \overrightarrow{\mathsf{x}_3}\sigma_3 = \langle \mathsf{x}_1, \mathsf{x}_2\rangle\{\mathsf{x}_0/z, \mathsf{x}_1/x, \mathsf{x}_2/y\} = \langle x, y\rangle$. Next, put $\varphi_3 = P(x, y, h_{\psi_3}(x, y))$. By a second recursion step, we obtain

$$\chi^* = \Big((\exists x)\varphi_3^*\{y/h_{\psi_2}(\overrightarrow{s_2})\}\Big)^* = \Big((\exists x)P(x, h_{\psi_2}(x, x), h_{\psi_3}(x, h_{\psi_2}(x, x)))\Big)^*,$$

where

– $\psi_2 = \mathit{Key}(\varphi_3, \{y\}) = P(\mathsf{x}_1, \mathsf{x}_0, h_{\psi_3}(\mathsf{x}_2, \mathsf{x}_0))$,
– $\overrightarrow{s_2} = \overrightarrow{\mathsf{x}_2}\sigma_2 = \langle \mathsf{x}_1, \mathsf{x}_2\rangle\{\mathsf{x}_0/y, \mathsf{x}_1/x, \mathsf{x}_2/x\} = \langle x, x\rangle$.

Finally, let us put $\varphi_2 = P(x, h_{\psi_2}(x, x), h_{\psi_3}(x, h_{\psi_2}(x, x)))$. Then by a last application of a recursion step we obtain
$\chi^* = \varphi_2^*\{x/h_{\psi_1}(\overrightarrow{s_1})\} = P(h_{\psi_1}(), h_{\psi_2}(h_{\psi_1}(), h_{\psi_1}()), h_{\psi_3}(h_{\psi_1}(), h_{\psi_2}(h_{\psi_1}(), h_{\psi_1}())))$, where $\psi_1 = \mathit{Key}(\varphi_2, \{x\}) = P(\mathsf{x}_0, h_{\psi_2}(\mathsf{x}_0, \mathsf{x}_0), h_{\psi_3}(\mathsf{x}_0, h_{\psi_2}(\mathsf{x}_0, \mathsf{x}_0)))$, and $\overrightarrow{s_1} = \overrightarrow{\mathsf{x}_1}\sigma_1 = \langle\ \rangle\{\mathsf{x}_0/x\} = \langle\ \rangle$.

Second translation of the standard predicate language $\mathcal{L}_{\Sigma_{\mathbf{sko}}}$ into the free-variable language $\mathcal{L}_{\Sigma_{\mathbf{sko}}}^{\smile}$. We will now describe our proposed map for translating standard predicate languages into free-variable languages, where quantifiers are defined by means of generalized key formulae.

As $(\cdot)^*$, $(\cdot)^g : \mathcal{L}_{\Sigma_{\mathbf{sko}}} \to \mathcal{L}_{\Sigma_{\mathbf{sko}}}^{\smile}$ is a map preserving the structure of *wfe*s acting only on formulae $\chi = (Qx_1, \ldots, x_n)\varphi$ with $Q \in \{\exists, \forall\}$. Specifically we have that $\chi^g = \big((Qx_1, \ldots, x_n)\varphi\big)^g = \psi\sigma'$, where

- $\psi = Key(\varphi^g, \{x_1, \ldots, x_n\})$,
- $\sigma'(x) = \begin{cases} h_{\pm\psi,j}(\overrightarrow{\mathsf{x}})\sigma & x = \mathsf{x}_{1-j}, \text{ for } 1 \leqslant j \leqslant n \\ \sigma(x) & \text{otherwise} \end{cases}$
- σ is such that $\varphi^g = \psi\sigma$,
- $\pm\psi = \textbf{if } (Q = \exists) \textbf{ then } \psi \textbf{ else } \neg\psi \textbf{ fi}$,
- $\overrightarrow{\mathsf{x}} = \langle \mathsf{x}_1, \ldots, \mathsf{x}_m \rangle$, where $\mathsf{x}_1, \ldots, \mathsf{x}_m$ are the variables in $Vars(\psi) \setminus \{\mathsf{x}_0, \ldots, \mathsf{x}_{1-n}\}$.

Example 3. Let us consider again the formula $\chi = (\exists x, y, z)P(x, y, z)$ from Example 2. Let us put $\varphi = P(x, y, z)$. Then we have $\varphi^g = P(x, y, z)$. Let $\psi = Key(\varphi^g, \{x, y, z\}) = P(\mathsf{x}_0, \mathsf{x}_{-1}, \mathsf{x}_{-2})$ and let $\sigma = \{\mathsf{x}_0/x, \mathsf{x}_{-1}/y, \mathsf{x}_{-2}/z\}$. We have $\varphi^g = \psi\sigma$. Since ψ does not contain any anonymous variable, we have $\overrightarrow{\mathsf{x}} = \langle\ \rangle$ and $\sigma' = \{\mathsf{x}_0/h_{\psi,1}(), \mathsf{x}_{-1}/h_{\psi,2}(), \mathsf{x}_{-2}/h_{\psi,3}()\}$. Therefore

$$\begin{aligned} \chi^g = \psi\sigma' &= P(\mathsf{x}_0, \mathsf{x}_{-1}, \mathsf{x}_{-2})\{\mathsf{x}_0/h_{\psi,1}(), \mathsf{x}_{-1}/h_{\psi,2}(), \mathsf{x}_{-2}/h_{\psi,3}()\} \\ &= P(h_{\psi,1}(), h_{\psi,2}(), h_{\psi,3}()) . \end{aligned}$$

Notice that, much like in Example 1, no nesting of function symbols has arisen.

The set of all images of the map $(\cdot)^g$ is denote with $[\mathcal{L}^{\smile}_{\Sigma_{\mathbf{sko}}}]^g$.

5 The free-variable calculus, revisited

Our version of the Davis–Fechter's free-variable calculus, here denoted GV_Σ, is based on the language $\mathcal{L}^{\smile}_{\Sigma_{\mathbf{sko}}}$. Its only inference rule is *modus ponens* and its logical axioms are the formula schemes of $\mathcal{L}^{\smile}_{\Sigma_{\mathbf{sko}}}$ listed here:

- *tautologies*;
- *identity axioms* of the form $t = t$;
- *congruence axioms*, of the two types:

$$t_0 = s_0 \ \& \ \cdots \ \& \ t_n = s_n \supset g(t_0, \ldots, t_n) = g(s_0, \ldots, s_n) ,$$
$$t_0 = s_0 \ \& \ \cdots \ \& \ t_n = s_n \supset \big(Q(t_0, \ldots, t_n) \supset Q(s_0, \ldots, s_n)\big) ;$$

- ε*-formulae*, of the form

$$\varphi\Big(t_1, \ldots, t_n, d_1, \ldots, d_m\Big) \supset \varphi\Big(h_{\varphi,1}(d_1, \ldots, d_m), \ldots, h_{\varphi,n}(d_1, \ldots, d_m), d_1, \ldots, d_m\Big) ,$$

where t_is, s_js, and d_ks stand for arbitrary terms, g for a function symbol of non-null arity in $\mathcal{F} \cup \mathcal{F}_{\mathbf{sko}}$, Q for a predicate symbol, φ for a key formula of degree n, and m equals $|Vars(\varphi)| - n$.

Note that all of these are universally valid formulae, save the ε-formulae, which are nevertheless essential to reflect the intended meaning of the Skolem function symbols. They enter into the notion of derivability introduced below.

Let φ and Γ be, respectively, a formula and a collection of formulae of $\mathcal{L}^{\smile}_{\Sigma_{\mathbf{sko}}}$. We write $\Gamma \vdash_{GV_\Sigma} \varphi$ (φ is *derivable* from Γ), to indicate that there exists a *derivation* D of φ from Γ in GV_Σ. D is a sequence $\varphi_1, \ldots, \varphi_n$ of formulae of $\mathcal{L}^{\smile}_{\Sigma_{\mathbf{sko}}}$ such that $\varphi_n \equiv \varphi$ and such that each φ_i is either a tautology, or an identity or congruence axiom, or an ε-formula, or it is obtained by means of modus ponens from φ_j, $\varphi_k \equiv \varphi_j \supset \varphi_i$, where $j, k < i$.

Notice that the only difference between the above formulation GV_Σ of the calculus and the very similar Davis–Fechter's formalization V_Σ (cf. [6]) lies in the languages underlying the calculi: $\mathcal{L}_{\Sigma_{\mathbf{sko}}}^{\smile}$ for GV_Σ and $[\mathcal{L}_{\Sigma_{\mathbf{sko}}}^{\smile}]^*$ for V_Σ. This forces us to introduce generalized ε-formulae among the axioms of GV_Σ.

Let T_Σ be a standard predicate calculus over $\mathcal{L}_\Sigma$ formalized as in [13]. It can be shown that for every formula $\varphi \in \mathcal{L}_\Sigma$, $\vdash_{T_\Sigma} \varphi$ holds if and only if $\vdash_{GV_\Sigma} \varphi^g$. For space reasons we refrain from presenting the whole proof and just give some hints on how to proceed. The interested reader can find the details in [4].

The proof is worked out by relating GV_Σ with V_Σ and exploiting the equivalence result between V_Σ and T_Σ presented in [6].

To begin with, it can be shown that GV_Σ is a conservative extension of V_Σ. This means that introducing new function symbols (generated by the generalized key formulae) and new axioms (the generalized ε-formulae) in V_Σ does not increase its deductive power. The proof can be carried out by showing that any proof of φ in GV_Σ, for a given formula φ of $[\mathcal{L}_{\Sigma_{\mathbf{sko}}}^{\smile}]^*$, can be translated into a proof of φ in V_Σ, by induction on the length of the proof of φ in GV_Σ.

Next it can be proved that for every φ of $\mathcal{L}_\Sigma$, $\vdash_{V_\Sigma} \varphi^*$ holds if and only if $\vdash_{GV_\Sigma} \varphi^g$. To do this, one can first show that $\vdash_{GV_\Sigma} \varphi^*$ if and only if $\vdash_{GV_\Sigma} \varphi^g$ and then, since GV_Σ is a conservative extension of V_Σ, the thesis follows. The first part of the proof can be carried out by using an appropriate function to translate any formula of $\mathcal{L}_{\Sigma_{\mathbf{sko}}}^{\smile}$ into a formula of $[\mathcal{L}_{\Sigma_{\mathbf{sko}}}^{\smile}]^*$ and a function to translate any formula of $\mathcal{L}_{\Sigma_{\mathbf{sko}}}^{\smile}$ into a formula of $[\mathcal{L}_{\Sigma_{\mathbf{sko}}}^{\smile}]^g$. The latter function is constructed using a suitable tool, called *Skolem nest*, to detect Skolem terms representing contiguous alike quantifiers inside formulae of $\mathcal{L}_{\Sigma_{\mathbf{sko}}}^{\smile}$. For a description of Skolem nest and related technical details the reader is referred to [4].

Using the above result jointly with [6, Theorem 6.2], one can prove that T_Σ and GV_Σ are equivalent.

Our variant of the free-variable calculus does not add deductive power to Davis–Fechter's original version and, therefore, it does not add deductive power to standard predicate logic either.

On the other hand, some differences may be noticed in the length of the resulting proofs. In fact, as the constructive proofs of the statements introduced in this section suggest, our variant may lead to shorter proofs. We illustrate this by means of a simple example.

Example 4. Consider the valid formula $\varphi = (\exists x, y)Q(x, y) \supset (\exists y, x)Q(x, y)$ of $\mathcal{L}_\Sigma$. The formulae obtained from φ by applying Skolemization, the ε-operator, the Davis-Fechter's technique, and our variant of Davis-Fechter's technique, are:

- Skolemization: $\mathbf{Skol}(\varphi) \equiv (\exists x, y)Q(x, y) \supset Q(h_1(), h_2())$.
  To prove the validity of the formula $\mathbf{Skol}(\varphi)$ one must instantiate the universally quantified variables x and y as $h_1()$ and $h_2()$, respectively.
- ε-operator: $\mathbf{Eps}(\varphi) \equiv Q(\varepsilon x.Q(x, \varepsilon y.Q(x, y)), \varepsilon y.Q(\varepsilon x.Q(x, \varepsilon y.Q(x, y)), y))$
  $\supset Q(\varepsilon x.Q(x, \varepsilon y.Q(\varepsilon x.Q(x, y), y)), \varepsilon y.Q(\varepsilon x.Q(x, y), y))$.

Clearly the formula $\mathbf{Eps}(\varphi)$ looks neither easy to read nor to manipulate.

– Davis-Fechter's technique:
  $\varphi^* \equiv Q(h_{Q_2}(), h_{Q_1}(h_{Q_2}())) \supset Q(h_{Q_3}(h_{Q_4}()), h_{Q_4}())$.
  Validity of φ^* can be proved by applying the syllogism rule to the ε-formulae:
  $\varepsilon_1 \equiv Q(h_{Q_2}(), h_{Q_1}(h_{Q_2}())) \supset Q(h_{Q_3}(h_{Q_1}(h_{Q_2}())), h_{Q_1}(h_{Q_2}()))$,
  $\varepsilon_2 \equiv Q(h_{Q_3}(h_{Q_1}(h_{Q_2}())), h_{Q_1}(h_{Q_2}())) \supset Q(h_{Q_3}(h_{Q_4}()), h_{Q_4}())$;
– our variant of Davis-Fechter's technique:
  $\varphi^g \equiv Q(h_{Q_{1,1}}(), h_{Q_{1,2}}()) \supset Q(h_{Q_{1,1}}(), h_{Q_{1,2}}())$.
  Notice that φ^g is a tautology of $\mathcal{L}_{\Sigma_{\mathbf{sko}}}^{\smile}$.

6 Conclusions and future work

We have defined a variant of Davis–Fechter's technique for eliminating quantifiers in first-order predicate logic, that enables treatment of contiguous alike quantifiers as a single syntactic unit.

Our variant does not add deductive power to the original formalism presented in [6]. Nonetheless it makes the system more efficient, because by constructing less complex and less deep Skolem terms it allows one to obtain shorter proofs, as said in the ending remarks and example here above.

Davis–Fechter's technique has already been employed in the context of tableau systems for classical first-order logic. For instance, in [1] it has been used to define an optimized version of the delta expansion rule. Such variant, called $\boldsymbol{\delta}^{*^*}$-rule, has the advantage of leading to non elementary speed-ups in proof length with respect to other delta-rule versions in the literature. It also yields Skolem terms that reflect in a more proper way the meaning of the instantiation. The construction of the $\boldsymbol{\delta}^{*^*}$-rule relies on a variant of the notion of key-formula that allows one to deal with quantifiers, inevitably present in an outer Skolemization process such as the one adopted in tableau systems by the $\boldsymbol{\delta}$-rule.

Davis–Fechter's technique also played a central rôle in [2], where a generic $\boldsymbol{\delta}$-rule based on the notion of key formula is presented, which is so designed as to ensure the correctness of any variant $\boldsymbol{\delta}$-rule to which it can be instantiated. Such a unified framework not only provides a schema apt to prove the correctness of versions of the $\boldsymbol{\delta}$-rule present in the literature, but it also enables comparisons between them that yield information on their efficiency and naturalness.

In [3], the framework introduced in [2] has been applied to map in the context of standard $\boldsymbol{\delta}$-rules, a version of the $\boldsymbol{\delta}$-rule, called $\boldsymbol{\delta}^{\varepsilon}$-rule, that adopts ε-terms (instead of Skolem terms) as syntactical objects to expand existentially quantified formulae.

The results that we present in this paper are an initial contribution to the process of analysis and improvement of the Davis–Fechter's technique.

We plan to add further refinements to the main procedure, like for instance the possibility of using, in validity (unsatisfiability) proofs, only ε-formulae relative to positive (negative) occurrences of existential quantifiers in the formula under consideration.

We also intend to compare Davis–Fechter's Skolem terms with ε-terms, investigating their exploitations both in theoretical issues and in automated deduction.

A further development of our research concerns the application of Davis–Fechter's technique to mathematical theories such as set theory. This will call for an adaptation of the original notion of key formula that makes it sensitive to the axioms of the theories being considered.

Acknowledgments

The authors wish to thank the referees for their helpful suggestions and comments.

References

1. D. Cantone and M. Nicolosi Asmundo. A Further and Effective Liberalization of the δ-Rule in Free Variable Semantic Tableaux. In R. Caferra and G. Salzer, editors, *Automated Deduction in Classical and Non Classical Logic - Selected papers*, volume 1761 of *Lecture Notes in Artificial Intelligence*, pages 109–125. Springer Verlag Berlin Heidelberg, 2000.
2. D. Cantone and M. Nicolosi Asmundo. A sound framework for $\boldsymbol{\delta}$-rule variants in free variable semantic tableaux. *J. Autom. Reasoning*, 38(1-3):31–56, 2007.
3. D. Cantone and M. Nicolosi Asmundo. Skolem functions and Hilbert's ϵ-terms in Free Variable Tableau Systems. In *Proceedings of CILC'06 - Italian Conference on Computational Logic*, Bari, Italia, 26-27 Giugno, 2006.
4. D. Cantone, M. Nicolosi Asmundo, and E. G. Omodeo. On the elimination of quantifiers through descriptors in predicate logic (extended version), 2012. http://www.dmi.unict.it/∼nicolosi/QuantElim.pdf.
5. D. C. Cooper. Theorem proving in arithmetic without multiplication. In B. Meltzer and D. N. Michie, editors, *Machine Intelligence*, volume 7, pages 91–99. Edinburgh University Press, 1972.
6. M.D. Davis and R. Fechter. A free variable version of the first-order predicate calculus. *J. of Logic and Computation*, 1(4):431–451, 1991.
7. N. Dershowitz and J.-P. Jouannaud. Rewrite systems. In *Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics (B)*, pages 243–320. 1990.
8. M. C. Fitting. A modal logic ε-calculus. *Notre Dame J. of Formal Logic*, 16:1–16, 1975.
9. M. C. Fitting. *First-Order Logic and Automated Theorem Proving*. Graduate Texts in Computer Science. Springer-Verlag, New York, 1996. second edition.
10. A.C. Leisenring. *Mathematical Logic and Hilbert's ε-symbol*. MacDonald & Co., 1969.
11. G. Moser and R. Zach. The epsilon calculus and Herbrand complexity. *Studia Logica*, 82:133–155, 2006.
12. A. Nonnengart and C. Weidenbach. Computing small clause normal forms. In *Handbook Automated Reasoning*, pages 335–367. 2001.
13. J. R. Shoenfield. *Mathematical Logic*. Series in Logic. Addison-Wesley, 1967.
14. K. von Heusinger. Definite descriptions and choice functions. In S. Akama, editor, *Logic, Language and Computation*, pages 61–91, Dordrecht, 1997. Kluwer.