An Extension of L_λ Unification [⋆]

Alwen Tiu

Computer Science and Engineering Department, 220 Pond Lab,
Pennsylvania State University Park, PA 16802-6106 USA
`tiu@cse.psu.edu`

Abstract. An extension of the language L_λ is given. The extension was mainly motivated by applications of higher-order abstract syntax in encoding generic judgments, where object-level signatures need to be coded explicitly. It is shown that this extension preserves the m.g.u property and the decidability of the unification problems.

1 Introduction

We extend the language of L_λ [Mil91] with the types *evs* (object-level eigen variables) and *nm* (names), and the constants $\hat{\rho}$: $evs \to evs$ and $\rho : evs \to nm$. We call the extended language L_λ^+. This extension is motivated by applications of higher-order abstract syntax in encoding generic judgments, where in certain cases a clear distinction of meta level signatures and object level signatures (i.e., the signatures of the system being encoded as generic judgments) must be made. The distinction means, operationally speaking, that no instantiation of object-level eigen variables can be made by the meta logic. This distinction is particularly useful for encoding transition systems such as π-calculus [MPW92] where there is a notion of fresh names. Fresh names can be naturally encoded as object-level eigen variables (for detailed discussions, see [Mil01, MM02]).

Object-level signatures are encoded as abstractions of type *evs*. That is, given a judgment

$$x_0, \ldots, x_n \vdash C x_0 \ldots x_n$$

of object logic, the corresponding atomic meta-level judgment would be

$$\triangleright \lambda l. C(\rho l) \ldots (\rho(\hat{\rho}^n l))$$

where $\triangleright : (evs \to o') \to o$. Here the type o is the type of meta-level formula, and o' is the type of object-level formula, and $\hat{\rho}^n l$ denotes n applications of $\hat{\rho}$ to l. The variable C above is a scheme variable that needs to be instantiated with approriate ground terms. We can encode object-level inference rules, for example, with Horn clauses by using such scheme variables. In the unification process, scheme variables are treated as existentially quantified, that is, if we see the variables in the unification pairs as quantified in a quantifier prefix [Mil92]. Therefore, the judgment above is clearly not in L_λ, because the variable C is

---

[⋆] Draft, dated September 16, 2002

applied to terms, instead of internally bound or universally quantified variables. However, we can see that intuitively, these $\rho(\hat{\rho}^k l)$ terms should not be too different from variables, since they are all distinct ground terms. Formally, we need to make certain restrictions on how the type *evs* and the constants ρ and $\hat{\rho}$ are used, precise enough to capture the intended meaning of these extensions.

In this paper, we show that the unification problem in L_λ^+ is decidable, and that it always gives the most general unifier if it is solvable. The algorithm and the proofs of main theorems follow the same outline as the ones for L_λ [Mil91], with an additional step involving the use of $\hat{\rho}$.

2 A Unification Algorithm for L_λ^+

The extension of L_λ is given in the following definition. We assume all terms are in $\beta\eta$ normal forms.

Definition 2.1. *Given a prefixed term $\mathcal{Q}t$, it is in L_λ^+ if the following hold.*

1. *Every subterm in t of the form $(xt_1 \ldots t_n)$, where $n \geq 0$ and x is existentially quantified in $\mathcal{Q}$, satisfies the following restrictions.*
   (a) *Each t_i, $1 \leq i \leq n$, is either a variable or a term of the form $\rho(\hat{\rho}^k l)$ or $\hat{\rho}^k l$. All variables in t_i are universally quantified to in the scope of x or internally bound by λ-abstractions.*
   (b) *For any two terms t_i, t_j, $i \neq j$, t_i is not a subterm of t_j and vice versa.*
2. *The constants ρ and $\hat{\rho}$ are applied only to terms of the form $\hat{\rho}^k l$, where $k \geq 0$ and $l : evs$ is a universally quantified or an internally bound variable.*

Remark 2.1. Item (2) is important to guarantee termination and mgu property of the unification. This is illustrated in the following example. Let $\exists M \forall l. M(\hat{\rho} l) = \hat{\rho}(Ml)$ be our unification problem. It is clear that any substitution of the form $\{M \mapsto \lambda l.\ \hat{\rho}^k l\}$, for $k \geq 0$ will be a solution, and each of these solutions is not comparable to the others. Hence we lost mgu property. From the algorithmic point of view, this is an instance of flexible-rigid case. If we apply Huet's algorithm [Hue75], in particular the projection and imitation steps, we will have an infinite reduction of this unification problem, i.e.,

$$\begin{aligned} M(\hat{\rho}\ l) = \hat{\rho}\ (Ml) &\Longrightarrow \hat{\rho}\ (H(\hat{\rho}\ l)) = \hat{\rho}\ (\hat{\rho}\ (Hl)) \\ &\Longrightarrow H(\hat{\rho}\ l) = \hat{\rho}\ (Hl) \\ &\Longrightarrow \ldots \end{aligned} \quad .$$

After seeing this example, readers might worry that this scheme of reduction can be generalized to the case where $\hat{\rho}$ is replaced with arbitrary rigid variables (i.e. universally quantified or internally bound) of the same type. In fact it can, but then it will fall outside the scope of our definition, more specifically, item (1.a.) above. This is also the reason why we need a special treatment of the constants ρ and (especially) $\hat{\rho}$.

We refer to the terms $\rho(\hat{\rho}^k l)$ and $\hat{\rho}^k l$ in the above definition as *ev*-term and *sig*-term (for signatures), respectively. A *sig*-subterm of t is a *maximal sig-subterm* if it is not a proper subterm of another *ev*- or *sig*-subterms of t. A *context* is a term with a hole, denoted by $t[]$. When we are interested in a particular occurence of a subterm s in a term t, we will write t as $t'[s]$, for some context $t'[]$. The term $t'[x]$ will then be the term obtained from t by replacing that particular occurence of s by x. We will also make use of multiple-holes context, when we are interested in several occurences of non-overlapping subterms. For example, given non-overlapping subterms $s_1, \dots, s_n$ in t, we can write it as $t'[s_1, \dots, s_n]$.

The unification algorithm we will describe here consists of the following one-step transitions on state formula: raising, ξ step, rigid-rigid, pruning, expansion, flex-rigid, and flex-flex. Raising, ξ-step and rigid-rigid steps are as in L_λ. All other steps are explained below. In the following explanation, we assume that a state formula $\mathcal{S}$ with quantifier prefix $\mathcal{Q}$ is given. Each step defines a substitution and a rewriting on the given state formula. If two state formula $\mathcal{S}$ and $\mathcal{S}'$ are related by a transition with substitution σ we write $\mathcal{S} \overset{\sigma}{\Longrightarrow} \mathcal{S}'$.

For most part of the algorithm, we will be working with $\beta\eta$-normal terms. One exception is when we consider termination problem. There we will use η-long β normal form instead. We refer to $\beta\eta$-normal form as λ-normal form (or λnorm for short), and η-long β normal form as λ-long normal form (or λlong). We study some simple properties of these normal forms, in our particular setting of L_λ^+.

Proposition 2.1. *Let t be a term in λ-normal (respectively, λ-long) form which contains a free occurrence of x. Let $\sigma = \{x \mapsto s\}$, where s is either y, $\rho(\hat{\rho}^k y)$ or $\hat{\rho}^k y$ and s is of the same type as x. Then $\sigma(t)$ is in λ-normal (λ-long) form.*

Proof. The proof is by induction on the structure of t. In the base cases, we have $t = x$ or $t = z$, where $x \neq z$, which is trivial. In the inductive step, we have $t = \lambda\bar{z}.ht_1 \dots t_n$. Then $\sigma(t) = \lambda\bar{z}.\sigma(h)\sigma(t_1) \dots \sigma(t_n)$. By inductive hypotheses each $\sigma(t_i)$ is in normal form. Substitution on the head h produces either y or $\rho(\hat{\rho}^k y)$ or $\hat{\rho}^k y$, and in each case, no β or η redex is introduced (in the case of λ-long form, no η-expansion is possible, since we do not change the type of the head). Hence, $\sigma(t)$ is in normal form. □

Proposition 2.2. *Let t be a λ-normal term in L_λ^+ with a prefix $\mathcal{Q}$ such that $t = w[t']$ where $t' = \lambda\bar{z}.(ut_1 \dots t_i \dots t_n)$, t' is not prefixed by an abstraction, u is existentially quantified in $\mathcal{Q}$ and t_i has an occurence of a universally quantified variable x in the scope of u. Let $\theta = \{u \mapsto \lambda x_1 \dots x_p.s\} \cup \theta'$ be a $\mathcal{Q}$-substitution and s is a λ-normal term that has an occurence of x_i. Then*

$$\lambda norm(\theta(t)) = r[\lambda norm(\theta(\lambda\bar{z}.ut_1 \dots t_n))],$$

for some context $r[]$.

Proof. By structural induction on t.

- $t = t'$, obvious.

- $t = \lambda\bar{y}.hw_1 \ldots w_j[t'] \ldots w_k$. Then

  $$\sigma(t) = \lambda\bar{y}.\theta(h)\theta(w_1) \ldots \theta(w_j[t']) \ldots \theta(w_k).$$

  By Definition 2.1, h must be a rigid head, so it does not change under substitution. By inductive hypothesis, we have

  $$\lambda\text{norm}(\theta(w_j[t'])) = r'[\lambda\text{norm}(\theta(t'))]$$

  for some context $r'[]$. We need only to make sure that $r'[\lambda\text{norm}(\theta(t'))]$ is not part of an η-redex. It is sufficient to show that $r'[\lambda\text{norm}(t')]$ is not a variable in $\bar{y}$. Since s is in normal form, we can apply Proposition 2.1, together with Definition 2.1, to show that $\lambda\text{norm}(\theta(t')) = \{x_1 \mapsto t_1, \ldots x_n \mapsto t_n\}s$. In particular, $\lambda\text{norm}(\theta(t'))$ will contain a universally quantified variable (i.e., the variable x in t_i), and therefore $r'[\lambda\text{norm}(\theta(t'))]$ cannot be a variable in $\bar{y}$.

□

Proposition 2.3. *Let t be a λ-long normal term in L^+_λ with a prefix $\mathcal{Q}$ such that $t = t'[\lambda\bar{z}.(ut_1 \ldots t_i \ldots t_n)]$ where u is existentially quantified in $\mathcal{Q}$. Let $\theta = \{u \mapsto \lambda x_1 \ldots x_n.s\} \cup \theta'$ be a $\mathcal{Q}$-substitution and s is a λ-long normal term. Then*

$$\lambda long\,(\theta(t)) = r[\lambda long\,(\theta(\lambda\bar{z}.ut_1 \ldots t_n))],$$

for some context $r[]$.

Proof. The proof is similar to the one in Proposition 2.2. The inductive case is simplified, that is, we need only to show that h is rigid. □

2.1 Expansion Step

Given the equation $vt_1 \ldots t_n = r$, let $s = \hat{\rho}^k l$ be a maximal *sig*-subterm of r, l is universally quantified in the scope of v and s does not have any subterm in $\{t_1 \ldots t_n\}$. If s does not occur as an argument of an existentially quantified variable, then replace this equation with $\bot$ and set the substitution σ to empty substitution. Otherwise, s occurs in a subterm $(us_1 \ldots s_i \ldots s_p)$ where $s_i = s$ and u is existentially quantified. We require that u and v be different, for proving termination of the algorithm later.

We define an m-expansion of the term $\hat{\rho}^k l$ as follows:

$$exp^m(\hat{\rho}^k l) = \{\rho(\hat{\rho}^k l), \rho(\hat{\rho}^{k+1} l), \ldots, \rho(\hat{\rho}^{m-1} l), \hat{\rho}^m l\}.$$

We then choose the smallest m that maximize the cardinality of the following set:

$$\mathcal{E} = \{t_1, \ldots, t_n\} \cap exp^m(\hat{\rho}^k l).$$

We call this set *the expansion set* of s_i. Let $w_1, \ldots, w_q$ be an enumeration of $\mathcal{E}$. Since all elements of the expansion set share the subterm $\hat{\rho}^k l$, we can write w_j as $w'_j[\hat{\rho}^k l]$. We then define the substitution σ as follows:

$$\sigma = \{u \mapsto \lambda x_1 \ldots \lambda x_p.u'x_1 \ldots x_{i-1}w'_1[x_i] \ldots w'_j[x_i]x_{i+1} \ldots x_p\},$$

where u' is not bound in $\mathcal{S}$. We then apply σ to $\mathcal{S}$ and replace u with u' in the prefix to obtain $\mathcal{S}'$. Note that if $\mathcal{E} = \emptyset$, the effect is like pruning. This can happen, for example, if $\{t_1, \ldots, t_n\}$ do not have any occurences of *evs* variables.

The maximal *sig*-subterm $\hat{\rho}^k l$ above will be referred to as *expandable subterm* of r. We now prove a local termination property of the expansion step, that is, the number of expandable subterms in the selected equation decreases.

Lemma 2.1. *Given a state formula $\mathcal{S}$, let $t = r$ be an equation in $\mathcal{S}$ subject to an expansion, and let $t' = r'$ be the corresponding equation after the expansion. Then the number of occurrences of expandable subterms in r' is strictly smaller than the number of occurrences of expandable subterms in r.*

Proof. The term t must be of the form $vt_1 \ldots t_n$. The notion of expandable subterm is dependent on the term t, therefore we first need to make sure that t is invariant under the substitution of this expansion step, which is the case here since $u \neq v$. We need to consider two cases. In the first case, the equation is replaced by $\bot$, and the lemma holds trivially. In the second case, let $\hat{\rho}^k l$ be the expandable subterm which occurs in the subterm $us_1 \ldots s_i \ldots s_p$, where $s_i = \hat{\rho}^k l$. After substitution, this subterm becomes $u's_1 \ldots w_1 \ldots w_q \ldots s_p$ where $\{w_1, \ldots, w_q\}$ is the expansion set. By the definition of the expansion set, each w_j is equal to some term $t_{\hat{j}}$, therefore no new expandable subterm is introduced. Since we also remove the expandable subterm s_i, clearly the number of expandable subterms decreases.

What is left to check is if the substitution introduces new expandable subterms in other subterms of the form $(uz_1 \ldots \hat{\rho}^{k'} l \ldots z_m)$ in r. After substitution, the *sig*-term $\hat{\rho}^{k'} l$ is expanded into a set of terms, and there can be at most one *sig*-term in the expansion set, which is of the form $\hat{\rho}^{k''} l$ where $k'' > k'$. If the *sig*-term $\hat{\rho}^{k'} l$ is not already an expandable subterm, which in this case means that it has a subterm in $\{t_1, \ldots, t_n\}$, obviously the substitute $\hat{\rho}^{k''} l$ is not an expandable subterm either. $\square$

The following lemma states the correctness of the transition, i.e., it preserves solvability.

Lemma 2.2. *Let $\mathcal{S} \overset{\sigma}{\Longrightarrow} \mathcal{S}'$ be an expansion step. Then σ is a factor for $\mathcal{S}$.*

Proof. We recall the definition of factor of unification [Mil92]: given a state formula (or unification problem), a substitution γ for $\mathcal{S}$ is a factor of $\mathcal{S}$ if for every solution θ of $\mathcal{S}$ there is a solution θ' for $\gamma(\mathcal{S})$ such that $\theta = \gamma \circ \theta'$. Let $vt_1 \ldots t_n = r$ be the equation under consideration. There are two possible cases. The first case is when this equation is replaced by $\bot$, and this can happen only if there is a maximal *sig*-subterm $\hat{\rho}^k l$ in r that has no subterm in $\{t_1, \ldots, t_n\}$. Then it must be the case that $\hat{\rho}^k l = r$ or it occurs as an argument of a rigid subterm of r, say, $h \ldots (\hat{\rho}^k l) \ldots$, where $h \neq \rho$ (by definition of maximal *sig*-subterm). In either case, $\hat{\rho}^k k$ remains after substitution. We argue that in this case $\mathcal{S}$ does not have a solution, and hence the lemma holds vacously. Suppose θ is a solution for $\mathcal{S}$. Then the λ normal term of $\theta(vt_1 \ldots t_n)$ must contain a maximal *sig*-subterm

$\hat{\rho}^k l$. Since l occurs in the scope of v, the only way to form the subterm $\hat{\rho}^k l$ is by projecting on the arguments of v. But then there must exist a t_i that is a subterm of $\hat{\rho}^k l$, contradiction.

The second case is when $r = r'[\lambda\bar{z}.us_1 \dots s_i \dots s_p]$ where $s_i = \hat{\rho}^k l$ and s_i does not have any subterm in $\{t_1, \dots, t_n\}$. Let $\mathcal{E} = \{w'_1[s_i], \dots, w'_q[s_i]\}$ be the expansion set of s_i. Let us recall the definition of σ:

$$\sigma = \{u \mapsto \lambda x_1 \dots \lambda x_p . u' x_1 \dots x_{i-1} w'_1[x_i] \dots w'_q[x_i] x_{i+1} \dots x_p\}.$$

We have to show that for every solution $\theta = \{u \mapsto \lambda x_1 \dots \lambda x_p . s\} \cup \gamma$ there exists a solution θ' for $\mathcal{S}'$ such that $\theta = \sigma \circ \theta'$. It is enough to show that we can decompose θ into $\sigma \circ \theta'$ since this implies that θ' is a solution for $\mathcal{S}$.

We do case analyses on the occurences of x_i in the term s. If x_i does not have any free occurences in s, then we define θ' as follows:

$$\theta' = \{u' \mapsto \lambda x_1 \dots \lambda x_{i-1} \lambda y_1 \dots \lambda y_q \lambda x_{i+1} \dots \lambda x_p . s\} \cup \gamma,$$

where $y_1, \dots, y_q$ are not free in s. It is easy to check that $\theta = \sigma \circ \theta'$ in this case.

If x_i have free occurences in s, we distinguish two cases. In the first case, x_i occurs in a subterm hx_i, where $h \neq \rho$ and $h \neq \hat{\rho}$. Then we have, by Proposition 2.2,

$$\begin{aligned} \lambda\mathrm{norm}(\theta(vt_1 \dots t_n)) &= \lambda\mathrm{norm}(\theta(r)) \\ &= r''[\lambda\mathrm{norm}(\theta(\lambda\bar{z}.us_1 \dots s_p))] \\ &= r''[\lambda\bar{z}.s'[(h\ \hat{\rho}^k l)]] \end{aligned}$$

for some contexts $r''[]$ and $s'[]$. Since l is in the scope of v, the term $\hat{\rho}^k l$ must be obtained by projection on $\{t_1, \dots, t_n\}$, which means that there is a term t_j that is subterm of $\hat{\rho}^k l$, contradiction.

Given the analyses above, the only possible occurence of x_i is in a subterm ρx_i or $\hat{\rho} x_i$. Any occurence of x_i will then be in a subterm w in s, of the form $\rho(\hat{\rho}^m x_i)$, where $m \geq 0$, or $\hat{\rho}^m l$ where $m > 0$. We claim that in this case, there exists a unique $w' \in \mathcal{E}$ such that w' is a subterm of $\{x_i \mapsto \hat{\rho}^k l\}w$. This can be checked by analyzing the structure of r and its normal form after substitution. If $r = \lambda\bar{z}.us_1 \dots s_p$, then, depending on w, $\lambda\mathrm{norm}(\theta(vt_1 \dots t_n)) = r_1[\rho(\hat{\rho}^{m+k} l)]$, or $r_2[\hat{\rho}^{m+k} l]$, for some contexts $r_1[]$ and $r_2[]$. In both cases, we must have a term t_j such that t_j is either $\rho(\hat{\rho}^{m+k} l)$ or $\hat{\rho}^{m'} l$ where $k < m' \leq m + k$. Therefore we take $w' = t_j$, since t_j must be in $\mathcal{E}$ by definition of $\mathcal{E}$. In the more general case, we have $r = r'[h \dots (\lambda\bar{z}.us_1 \dots s_p) \dots]$ for some context $r'[]$ and some variable h. By Definition 2.1, $h \neq \rho$ and $h \neq \hat{\rho}$. Therefore the argument in the previous case applies as well. The uniqueness property follows from the definiton of $\mathcal{E}$.

Having proven the above claim, we can assume that s is of the form

$$s = s''[w'_{\hat{1}}[x_i], \dots, w'_{\hat{j}}[x_i]]$$

where j is the number of occurences of x_i in s and $\hat{\ } : \{1, \dots, j\} \to \{1, \dots, q\}$. Now we define θ' as follows:

$$\theta' = \{u' \mapsto \lambda x_1 \dots \lambda x_{i-1} \lambda z_1 \dots \lambda z_q \lambda x_{i+1} \dots \lambda x_p . s''[z_{\hat{1}}, \dots, z_{\hat{j}}]\} \cup \gamma.$$

It remains to show that $\theta u = (\sigma \circ \theta')u$:

$$\begin{aligned}
\theta u &= \lambda x_1 \ldots \lambda x_p . s''[w'_{\hat{1}}[x_i], \ldots, w'_{\hat{j}}[x_i]] \\
&= \lambda x_1 \ldots \lambda x_p . (\theta' u') x_1 \ldots x_{i-1} w'_{\hat{1}}[x_i] \ldots w'_{\hat{j}}[x_i] x_{i+1} \ldots x_p \\
&= \theta'(\lambda x_1 \ldots \lambda x_p . u' x_1 \ldots x_{i-1} w'_{\hat{1}}[x_i] \ldots w'_{\hat{j}}[x_i] x_{i+1} \ldots x_p) \\
&= \theta'(\sigma u) \\
&= (\sigma \circ \theta')u
\end{aligned}$$

Hence, $\theta = (\sigma \circ \theta')$ $(mod\ \mathcal{S})$.

□

2.2 Pruning Step

Let $vt_1 \ldots t_n = r$ be an equation in $\mathcal{S}$ where v is existentially quantified. Let s be a variable or an *ev*-subterm in r. Suppose all variables in s are universally bound in the scope of v, and none of $\{t_1, \ldots, t_n\}$ is a subterm of s. If no such term occurs in r, then this step is not applicable. If s does not occur as an argument of some existentially quantified variable, replace the equation with $\bot$. Otherwise, s occurs in a subterm $uq_1 \ldots q_i s q_{i+1} \ldots q_j$, where $u \neq v$ is existentially quantified. We define a substitution σ as follows:

$$\sigma = \{u \mapsto \lambda x_1 \ldots \lambda x_j . u' x_1 \ldots x_{i-1} x_{i+1} \ldots x_j\},$$

where u' is not bound in $\mathcal{S}$. We then replace $\mathcal{S}$ by the formula $\mathcal{S}'$ obtained from replacing $\exists u$ with $\exists u'$ and applying the susbtitution σ to all judgments in $\mathcal{S}$.

We call the subterm s above as *prunable subterm* of r. As we did with expansion step, we prove a local termination property of pruning step.

Lemma 2.3. *Given an state formula $\mathcal{S}$, let $t = r$ be an equation in $\mathcal{S}$ subject to a pruning transition, and let $t' = r'$ be the corresponding equation after the expansion. Then the number of prunable subterms in r' is strictly smaller than the number of prunable subterms in r.*

Proof. We consider only the case where the equation is not replaced with $\bot$, otherwise it is trivial. Then, $t = vt_1 \ldots t_n$, and $r = r'[(us_1 \ldots s_i \ldots s_m)]$, where $u \neq v$. As in expansion step, it is important that t remains unchanged after substitution, which is the case here also because of the restriction $u \neq v$. Since the substitution σ in this step reduces the number of arguments of u, in particular, the prunable subterm s_i is discarded, the number of prunable subterms in r decreases after substitution. □

Lemma 2.4. *Let $\mathcal{S} \overset{\sigma}{\Longrightarrow} \mathcal{S}'$ be a pruning step. Then σ is a factor for $\mathcal{S}$.*

Proof. Let $vt_1 \ldots t_n = r$ be the equation subject to pruning. If this equation is replaced by $\bot$, then it must be the case there is a subterm s of r that has no subterm in $\{t_1, \ldots, t_n\}$, and either $r = s$ or $r = r'[(h \ldots s \ldots)]$, where h is a rigid head. By the same argument as in the proof of Lemma 2.2 we conclude that in this case $\mathcal{S}$ is not solvable, and therefore any substitution σ will serve as a factor. Otherwise, suppose s occurs as an argument of u, where u is existentially quantified. Then $r = r''[(us_1 \dots s_i \dots s_j)]$ where $s_i = s$. Recall the definition of σ:

$$\sigma = \{u \mapsto \lambda x_1 \dots \lambda x_j . u' x_1 \dots x_{i-1} x_{i+1} \dots x_j\}.$$

Suppose $\theta = \{u \mapsto \lambda x_1 \dots \lambda x_j q\} \cup \gamma$ is a solution for $\mathcal{S}$. We show that θ can be decomposed into $\sigma \circ \theta'$, for some θ'. We claim that q does not have any occurences of x_i, because otherwise the term θr will contain s as subterms, and consequently the λ normal form of $\theta(vt_1 \dots t_n)$ would have to contain s as subterms. Since all variables in s are universally quantified to the right of v, the subterm s can only be formed through projection on the arguments $\{t_1 \dots t_n\}$, but then s must have a subterm in $\{t_1 \dots t_n\}$, contradiction. Therefore we define

$$\theta' = \{u' \mapsto \lambda x_1 \dots \lambda x_{i-1} \lambda x_{i+1} \dots \lambda x_j . q\} \cup \gamma.$$

It is easy to see that $\theta = \sigma \circ \theta' (mod \mathcal{S})$. □

2.3 Flexible-Rigid Step

Let $vt_1 \dots t_n = r$ be the selected equation where r is a rigid term. If v occurs in r, then replace this equation with $\bot$. Otherwise, given raising, pruning and expansion steps, we assume that all universally quantified variables in r which are in the scope of v, all *ev*-subterms and maximal *sig*-subterms of r of the form $\rho(\hat{\rho}^k l)$ and $\hat{\rho}^m l$, where l is universally quantified in the scope of v, have subterms in $\{t_1, \dots, t_n\}$. Therefore r is λ-convertible to $(\lambda x_1 \dots \lambda x_n . r') t_1 \dots t_n$, where r' is a term obtained from r by replacing all subterms t_i with x_i. In other words, we abstracted the subterms t_i from r. Since the arguments of v are distinct terms and no two terms are subterms of one or another, the term r' is unique. We define the substitution σ as follows:

$$\sigma = \{v \mapsto \lambda x_1 \dots \lambda x_n . r'\},$$

and replace the equation with $\top$.

Definition 2.2. *Let t be a term in λ-long form, and let Γ be a set of variables that have free occurences in t. We define a measure on the term as follows:*

$$\begin{aligned} \#(x, \Gamma) &= \begin{cases} 1, & \textit{if } x \in \Gamma \ , \\ 0, & \textit{otherwise} \end{cases} \\ \#(\lambda x.r, \Gamma) &= \#(r, \Gamma \cup \{x\}) \\ \#((uv), \Gamma) &= \#(u, \Gamma) + \#(v, \Gamma) \end{aligned}$$

Lemma 2.5. *Let t be a term in λ-long normal form with an occurence of x, let Γ be a set of variables. Let $\sigma = \{x \mapsto s\}$ where s is in the form y, $\rho(\hat{\rho}^k y)$ or $\hat{\rho}^k y$ and either $y \in \Gamma$ or y is a new variable that is not free in t. Then $\#(t, \Gamma \cup \{x\}) = \#(\lambda long\,(\sigma(t)), \Gamma \cup \{y\})$.*

Proof. By Proposition 2.1, $\lambda\text{long}\ (\sigma(t)) = \sigma(t)$. The proof proceeds by induction on t. The base cases, namely $t = x$ and $t = z$ where $z \neq x$, holds trivially. In the inductive case, we have $t = \lambda\bar{z}.ht_1 \dots t_n$. Then

$$\#(t, \mathit{\Gamma} \cup \{x\}) = \#(h, \mathit{\Gamma} \cup \{x, \bar{z}\}) + \#(t_1, \mathit{\Gamma} \cup \{x, \bar{z}\}) + \dots + \#(t_n, \mathit{\Gamma} \cup \{x, \bar{z}\}).$$

By inductive hypothesis, we have $\#(h, \mathit{\Gamma} \cup \{x, \bar{z}\}) = \#(\sigma(h), \mathit{\Gamma} \cup \{y, \bar{z}\})$ and $\#(t_i, \mathit{\Gamma} \cup \{x, \bar{z}\}) = \#(\sigma(t_i), \mathit{\Gamma} \cup \{y, \bar{z}\})$, for all $i \in \{1, \dots, n\}$. Therefore,

$$\begin{aligned}\#(\sigma(t), \mathit{\Gamma} \cup \{y\}) &= \#(\lambda\bar{z}.\sigma(h)\sigma(t_1) \dots \sigma(t_n), \mathit{\Gamma} \cup \{y\}) \\ &= \#(\sigma(h), \mathit{\Gamma} \cup \{y, \bar{z}\}) + \textstyle\sum_{i=1}^{n} \#(\sigma(t_i, \mathit{\Gamma} \cup \{y, \bar{z}\}) \\ &= \#(t, \mathit{\Gamma} \cup \{x\}).\end{aligned}$$

□

Lemma 2.6. *Let $t = vt_1 \dots t_n$ and $s = vs_1 \dots s_p$ be two L_λ^+-terms with a prefix $\mathcal{Q}$, where v is existentially quantified in $\mathcal{Q}$ and $p \leq n$. Let $\theta = \{v \mapsto \lambda x_1 \dots \lambda x_n.r\} \cup \theta'$ be a $\mathcal{Q}$-substitution, where $\theta(v)$ is in λ-long normal form. Then*

$$\#(\lambda\mathit{long}\ (\theta(t)), \mathit{\Gamma} \cup \mathit{\Gamma}_t) = \#(\lambda\mathit{long}\ (\theta(s)), \mathit{\Gamma} \cup \mathit{\Gamma}_s)$$

where $\mathit{\Gamma}$ is the set of all universally quantified variables in $\mathcal{Q}$ and $\mathit{\Gamma}_t$ and $\mathit{\Gamma}_s$ are sets of variables in $\{t_1, \dots, t_n\}$ and $\{s_1, \dots, s_p\}$, respectively.

Proof. Let $\sigma = \{x_1 \mapsto t_1, \dots, x_n \mapsto t_n\}$ and $\gamma = \{x_1 \mapsto s_1, \dots, x_p \mapsto s_p\}$. By Proposition 2.1, $\lambda\text{long}\ (\theta(t)) = \sigma(r)$ and $\lambda\text{long}\ (\theta(s)) = \lambda x_{p+1} \dots \lambda x_n.\gamma(r)$. Applying Lemma 2.5, we have

$$\begin{aligned}\#(\lambda\text{long}\ (\theta(t)), \mathit{\Gamma} \cup \mathit{\Gamma}_t) &= \#(\sigma(r), \mathit{\Gamma} \cup \mathit{\Gamma}_t) \\ &= \#(r, \mathit{\Gamma} \cup \{x_1, \dots, x_n\}) \\ &= \#(\lambda x_{p+1} \dots \lambda x_n.r, \mathit{\Gamma} \cup \{x_1, \dots, x_p\}) \\ &= \#(\lambda x_{p+1} \dots \lambda x_n.\gamma(r), \mathit{\Gamma} \cup \mathit{\Gamma}_s) \\ &= \#(\theta(s), \mathit{\Gamma} \cup \mathit{\Gamma}_s).\end{aligned}$$

□

Note that the two lemmas above hold because we do not count the constants ρ and $\hat{\rho}$. Hence, Definition 2.1 plays an important role here.

Lemma 2.7. *Let $vt_1 \dots t_n = r$ be an equation in a state formula $\mathcal{S}$, where v is existentially quantified and r has a rigid head. If v occurs in r then $\mathcal{S}$ has no solution.*

Proof. Suppose v occurs in r in a subterm $h \dots (\lambda\bar{z}.vs_1 \dots s_m) \dots$, where h is a rigid variable, but $\mathcal{S}$ has a solution $\theta = \{v \mapsto \lambda x_1 \dots \lambda x_n.q\} \cup \theta'$. Note that in general m may not be equal to n. We consider here the λ-long normal form of terms after substitution. By Proposition 2.3,

$$\lambda\text{long}\ (\theta(r)) = r'[\lambda\bar{z}.\ \lambda\text{long}\ (\theta(vs_1 \dots s_m))].$$

Since λlong $(\theta(vt_1 \ldots t_n))$ =λlong $(\theta(r))$, we must have

$$\#(\lambda\text{long } (\theta(vt_1 \ldots t_n)), \mathit{\Gamma}) = \#(\lambda\text{long } (\theta(r)), \mathit{\Gamma})$$

where $\mathit{\Gamma}$ is the set of all universally quantified variables in $\mathcal{S}$. By Lemma 2.6, we have

$$\#(\lambda\text{long } (\theta(vs_1 \ldots s_m)), \mathit{\Gamma} \cup \mathit{\Gamma}') = \#(\lambda\text{long } (\theta(vt_1 \ldots t_n)), \mathit{\Gamma}),$$

where $\mathit{\Gamma}'$ contains free variables in $\{s_1 \ldots s_m\}$. Since the head of r is a universally quantified variable (it cannot be the constant ρ or $\hat{\rho}$ by Definition 2.1), we have

$$\#(\lambda\text{long } (\theta(r)), \mathit{\Gamma}) \geq \#(\theta(vs_1 \ldots s_m), \mathit{\Gamma} \cup \mathit{\Gamma}') + 1 > \#(\lambda\text{long } (\theta(vt_1 \ldots t_n)), \mathit{\Gamma}),$$

contradiction. Therefore, θ cannot be a solution for $\mathcal{S}$. ☐

Lemma 2.8. *Let $\mathcal{S} \overset{\sigma}{\Longrightarrow} \mathcal{S}'$ be a flex-rigid step. Then σ is a factor for $\mathcal{S}$.*

Proof. Let $vt_1 \ldots t_n = r$ be the selected equation in this step. Lemma 2.7 handles the case where the equation is replaced by $\perp$. We consider the other case, where $r =_\beta (\lambda x_1 \ldots \lambda x_n r')t_1 \ldots t_n$, and $\sigma = \{v \mapsto \lambda x_1 \ldots \lambda x_n.r'\}$. We refer to notations used in the beginning of this section. Again, as before, we show that given a solution θ of $\mathcal{S}$, we can find a substitution θ' such that $\theta = \sigma \circ \theta'$ $(mod\ \mathcal{S})$. Suppose $\theta = \{v \mapsto \lambda x_1 \ldots \lambda x_n.s\} \cup \gamma$. Then take $\theta' = \gamma$. Given α-conversion, we can assume that $x_1, \ldots, x_n$ are not free in θ'. We have $\theta(v) = \lambda x_1 \ldots \lambda x_n.s$ and

$$(\sigma \circ \theta')(v) = \theta'(\lambda x_1 \ldots \lambda x_n.r') = \lambda x_1 \ldots \lambda x_n.\theta'(r').$$

Since θ is a solution for $\mathcal{S}$, we have also:

$$\begin{aligned}
\theta(vt_1 \ldots t_n) &= \theta(r) \\
(\lambda x_1 \ldots \lambda x_n.s)t_1 \ldots t_n &= \theta'((\lambda x_1 \ldots \lambda x_n.r')t_1 \ldots t_n) \\
(\lambda x_1 \ldots \lambda x_n.s)t_1 \ldots t_n &= (\lambda x_1 \ldots \lambda x_n.\theta'(r'))t_1 \ldots t_n
\end{aligned}$$

hence, $\theta(v) = \lambda x_1 \ldots \lambda x_n.s = \lambda x_1 \ldots \lambda x_n.\theta'(r') = (\sigma \circ \theta')(v)$. Therefore, $\theta = \sigma \circ \theta'(mod\ \mathcal{S})$. ☐

2.4 Flexible-Flexible Step

Let $vt_1 \ldots t_n = us_1 \ldots s_p$ be the selected equation, where both u, v are existentially quantified. We distinguish two cases.

1. $u \neq v$. By raising, pruning and expansion, we can assume that every s_i has a unique $t_{\hat{i}}$ as subterm. Therefore, just like in flexible-rigid step, we have $vs_1 \ldots s_n =_\lambda (\lambda x_1 \ldots \lambda x_n.r')t_1 \ldots t_n$. We define a substitution

   $$\sigma = \{v \mapsto \lambda x_1 \ldots \lambda x_n.r')$$

   We replace the equation with $\top$ and apply θ to the rest of equations in $\mathcal{S}$ to obtain $\mathcal{S}'$.

2. $u = v$. Then we have $p = n$. Let $I = \{i \mid t_i = s_i, i \in \{1, \dots, n\}\}$ and let $i_1, \dots, i_m$ be an enumeration of I. We define the substitution

$$\sigma = \{v \mapsto \lambda x_1 \dots \lambda x_n . v' x_{i_1} \dots x_{i_m}\}$$

where v' is not bound in $\mathcal{S}$. We apply σ to $\mathcal{S}$, replace the equation with $\top$, and replace u in $\mathcal{Q}$ with v'.

Lemma 2.9. *Let $\mathcal{S} \overset{\sigma}{\Longrightarrow} \mathcal{S}'$ be a flexible-flexible step. Then σ is a factor for $\mathcal{S}$.*

Proof. The case where $u \neq v$ is similar to the proof in Lemma 2.8, so we consider here only the case where $u = v$, that is, $ut_1 \dots t_n = us_1 \dots s_n$, and σ is defined as

$$\sigma = \{v \mapsto \lambda x_1 \dots \lambda x_n . v' x_{i_1} \dots x_{i_m}\},$$

where $I = \{i \mid t_i = s_i, i \in \{1, \dots, n\}\} = \{i_1, \dots, i_m\}$.

Suppose $\theta = \{u \mapsto \lambda x_1 \dots \lambda x_n . s\} \cup \gamma$ is a solution for $\mathcal{S}$. Let $\phi = \{x_1 \mapsto t_1, \dots, x_n \mapsto t_n\}$ and $\phi' = \{x_1 \mapsto s_1, \dots, x_n \mapsto s_n\}$. Then $\lambda\text{norm}(\theta(vt_1 \dots t_n)) = \phi(s)$ and $\lambda\text{norm}(\theta(vs_1 \dots s_n)) = \phi'(s)$. It must be the case that $\phi(s) = \phi'(s)$. We show by induction that for every λ-normal term s, $\phi(s) \neq \phi'(s)$ if x_i has a free occurence in s. In the base case, suppose x_i occurs in s, then $s = \lambda \bar{z}.x_i$. But then $\phi(s) = \lambda \bar{z}.t_i \neq \lambda \bar{z}.s_i = \phi'(s)$. For the inductive case, we have $s = \lambda \bar{z}.hw_1 \dots w_k$. Then $\phi(s) = \lambda \bar{z}.\phi(h)\phi(w_1) \dots \phi(w_k)$, and $\phi'(s) = \lambda \bar{z}.\phi'(h)\phi'(w_1) \dots \phi'(w_k)$. Since $\phi(s) = \phi'(s)$, then it must be the case that $\phi(h) = \phi'(h)$ and $\phi(w_j) = \phi'(w_j)$, for all $1 \leq j \leq k$. By inductive hypothesis, none of h or w_j can have free occurences of x_i. Now we define θ' as follows:

$$\theta' = \{v' \mapsto \lambda x_{i_1} \dots \lambda x_{i_m} . s\} \cup \gamma,$$

where $\{i_1, \dots, i_m\} = I$. It is easy to see that $\theta = \sigma \circ \theta'$. □

2.5 Unification Algorithm

Given a state formula $\mathcal{S}_0$, we apply the following steps until there are no equation left or until $\bot$ appears in a state formula. This gives rise to a series of transitions

$$\mathcal{S}_0 \overset{\sigma_1}{\Longrightarrow} \dots \overset{\sigma_n}{\Longrightarrow} \mathcal{S}_n \ (n \geq 0).$$

The result of the unification algorithm is the pair $< \sigma_1 \circ \dots \circ \sigma_n, \mathcal{S}_n >$.

Step 1. Apply either the ξ or the rigid-rigid step to the first applicable equation found in a left-to-right transversal of the state formula. If neither of these steps applies, move to the next step.

Step 2. Select the first flexible-flexible or flexible rigid equation in a left-to-right order. Apply raising, expansion and pruning steps to that equation until these transitions can no longer be applied, and then move to the next step. The exact order in which the various raising steps, expansion steps or pruning steps are applied can be specified arbitrarily.

Step 3. Apply as appropriate either the flexible-flexible or flexible-rigid step to the resulting selected equation.

3 Correctness of the Unification Algorithm

We first show that the language L_λ^+ are closed with respect to the unification transitions. We then show termination and correctness of the transitions.

Theorem 3.1. *Let $\mathcal{S}$ be a state formula in L_λ^+ and $\mathcal{S} \overset{\sigma}{\Longrightarrow} \mathcal{S}'$ be a unification transition step. Then $\mathcal{S}'$ is in L_λ^+.*

Proof. We consider the case where σ is non-empty. In each transition step, a substitution $\sigma = \{u \mapsto \lambda \bar{z}.s\}$ is defined. It can be checked that s itself is an L_λ^+-term. Applying σ to $\mathcal{S}$ corresponds to replacing subterms of the form $ut_1 \ldots t_n$ with $\{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$, which is also an L_λ^+-term since we are replacing rigid variables with rigid variables, *ev*-terms or *sig*-terms (this is a bit simplified, but it holds as well in the general case where u is partially applied). Since this subterm $ut_1 \ldots t_n$ never appears as an argument of a flexible head, we can basically replace it with any L_λ^+-term of the same type and still get an L_λ^+-term. $\square$

Let t be a λ-normal term all of whose free tokens are quantified at the meta-level in $\mathcal{S}$, the measure $|t|$ counts the number of occurrences of abstractions and applications in t that are not in the scope of existentially quantified variables of $\mathcal{S}$. That is, $|t|$ is defined by

$$|\lambda x_1 \ldots \lambda x_k (h t_1 \ldots t_n)| = \begin{cases} k & h \text{ existentially quantified in } \mathcal{S} \\ k + n + \sum_{i=1}^{n} |t_i| & h \text{ universally quantified in } \mathcal{S} \end{cases}$$

where $(k, n \geq 0)$. (Of course, $|t|$ also has $\mathcal{S}$ as an argument, but its value will always be clear from context.) The *weight* of a meta-level, universal quantifier is the number of occurrences of meta-level, existential quantifiers in its scope.

Let $t_1 = s_1, \ldots, t_n = s_n$ be the list of equations that occur in $\mathcal{S}$ and let m be the number of existentially quantified variables in $\mathcal{S}$. The measure associated to $\mathcal{S}$ is defined by the quadruple

$$|\mathcal{S}| = \langle m, \sum_{i=1}^{n} |t_i| + |s_i|, n, w \rangle,$$

where w is the sum of the weights of all meta-level, universal quantifiers in $\mathcal{S}$, and n is the total number of equations in $\mathcal{S}$. Quadruples are ordered lexicographically.

This measure is adopted from the one for proving termination of L_λ unification algorithm [Mil91]. The difference with the measure used in [Mil91] is that we omit the part that measures the number of prunable variables; instead, pruning and expansion steps are dealt locally, i.e., by considering only the equation chosen at Step 2 in our algorithm. One reason is that we notice although pruning step reduces the number of occurrences of possibly prunable variable in one equation, it can possibly introduce new prunable variables in other equations. The same holds for expansion step. To see why this is the case, let us consider the following example. Let

$$\mathcal{S} = \forall a \exists u \exists v \exists w \forall a \forall b \forall z. uab = c(vaz)b \wedge vab = wab.$$

It is easy to check that this formula is typeable. This formula has one prunable variable, that is, z. Applying pruning to the first equation, we get

$$uab = c(v'a)b \wedge v'a = wab,$$

where v' is a new existential variable replacing v. We thus introduces a new prunable variable, namely, b. Other measures remains the same, so in this case the measure does not decrease.

We apply the measure above in the following way: we measure the resulting state formula after each pass of the unification algorithm. That is, instead of considering each unification transition step and showing that the measure decreases in each step, we consider a sequence of transitions steps and show that at the end of this sequence, the measure decreases. For this purpose, we introduce a local measure for equations, for Step 2 of the algorithm:

$$\begin{array}{ll} |\bot| & = 0, \\ |t = r| & = \langle |t|, e, p\rangle, \text{ if } t = vt_1 \dots t_n, \end{array}$$

where v is existentially quantified, e is the number of expandable terms in r and p is the number of prunable terms in r.

Theorem 3.2. *There is no infinite series of unification transitions.*

Proof. Let $\mathcal{S}$ be the given state formula. We show that after each step of the algorithm (not the unification transitions), the global measure decreases, with one exception of Step 2, where we show that the global measure does not change, but the local measure decreases, and hence this step will eventually terminate. In Step 1, we apply only ξ and rigid-rigid steps. Suppose the one-step transition $\mathcal{S} \overset{\sigma}{\Longrightarrow} \mathcal{S}'$ is the raising step, then the weight of at least one meta-level, universal quantifier in $\mathcal{S}$ decreases in $\mathcal{S}'$. Although the number of applications in the state formula may have increased, all new applications are in the scope of existentially quantified variables and are therefore not counted by the $| \ |$-measure. Since the number of equations and number of existentially quantified variables have not changed, $|\mathcal{S}'| < |\mathcal{S}|$. If the transition is the ξ step, the number of abstractions in equations decreases. If the transition is the rigid-rigid step, then either the number of applications decreases or the number of equations decreases. Thus in either of these cases, $|\mathcal{S}'| < |\mathcal{S}|$.

In Step 2, we apply raising, pruning and expansion. Let $t = r$ be the selected equation at this step. If the transition is the pruning step or expansion step, either the number of equations is reduced by one or all components of the global measure $|\mathcal{S}|$ are unchanged. Lemma 2.1 and Lemma 2.3 guarantee that components e and p decrease, and furthermore, these steps do not interfere with each other, that is, application of pruning does not introduce new expandable subterms and vice versa. Therefore, the local measure $|t = r|$ decreases. This means there can be only finitely many applications of pruning and expansions (and raising), and hence Step 2 terminates, without increasing the measure $|\mathcal{S}|$.

In Step 3, we apply flexible-flexible or flexible-rigid transition steps. In the first case of the flexible-flexible step, the number of existentially quantified variables decreases by one. Hence, the measure $|\mathcal{S}|$ decreases. In the second case, the number of existentially quantified variables and the number of occurrences of applications not in the scope of existentially quantified variables remain the same. Since the number of equations decreases, the measure $|\mathcal{S}|$ also decreases. Finally, if $\mathcal{S}'$ arises from $\mathcal{S}$ by applying the flexible-rigid case, the number of equations reduces by one and an existentially quantified variable from $\mathcal{S}$ may also be deleted. Thus, again $|\mathcal{S}'| < |\mathcal{S}|$.

It remains to check if the measure $|\mathcal{S}|$ always decreases in each pass of the algorithm. Since Step 1 and Step 3 always end up with a decrease in measure, we need only to check the case where only Step 2 is applicable but not Step 1 and Step 3. This happens only if the equation $t = r$ in Step 2 is eventually replaced by $\perp$, in which case the algorithm terminates. $\square$

The following lemma and propositions show that the unification transitions can be used to determine whether or not solutions exist and to characterize all of them if they do exist.

Lemma 3.1. *Assume that $\mathcal{S} \overset{\sigma}{\Longrightarrow} \mathcal{S}'$ is a unification transition. The solutions to $\mathcal{S}$ can be put into one-to-one correspondence with the solution to $\mathcal{S}'$ so that if the solution φ for $\mathcal{S}$ corresponds to the solution φ' for $\mathcal{S}'$ then $\rho \circ \varphi' = \varphi \pmod{\mathcal{S}}$.*

Proof. Assume that the transition is the raising step. That is, the state formula changed by lifting $\exists u$ up over the universally quantified variables in $\bar{w}$ to get the quantifier $\exists u'$ and $\rho = [u \mapsto u'\bar{w}]$ is applied to all judgements. The correspondence of solutions is given by either letting φ' be the result of replacing $u \mapsto s$ in φ with $u' \mapsto \lambda\bar{w}.s$, or conversely, letting φ be the result of replacing $u' \mapsto r$ in φ' with $u \mapsto \lambda\mathrm{norm}(r\bar{w})$. Since φ and φ' differ only on u and u' and since $(\rho \circ \varphi')u = \varphi'(u'\bar{w}) = (\lambda\bar{w}.s)\bar{w}\ \lambda conv\ s = \varphi u$, it follows that $\rho \circ \varphi' = \varphi$ $(\mathrm{mod}\ \mathcal{S})$. Notice that raising is a general transition for unification problems: it is dependent only on the scope of quantifiers and not on the judgements of the state formula. A fuller description of this transition is presented in [Mil92].

If the transition is the ξ step, the result follows immediately since ρ is the empty substitution and the set of solutions does not change.

Assume that the transition is the rigid-rigid step. If the equation replaced with this step is $ht_1 \ldots t_n = hs_1 \ldots s_n$, a substitution makes these terms λ-convertible if and only if it makes t_i λ-convertible s_i, for $i = 1, \ldots, n$. Thus, $\mathcal{S}$ and $\mathcal{S}'$ have the same solutions. If the equation replaced with this step is $ht_1 \ldots t_n = ks_1 \ldots s_m$, where h and k are different universally quantified variables in $\mathcal{S}$, then this equation cannot be made equal and $\mathcal{S}$ has no solutions. Neither does $\mathcal{S}'$ since it contains $\perp$.

The cases for pruning, expansion, flexible-rigid and flexible-flexible steps are covered by Lemma 2.4, 2.2, 2.8 and 2.9, respectively. $\square$

Theorem 3.3. *If the unification algorithm is applied to the state formula $\mathcal{S}$, it terminates with a result, say $\langle\theta, \mathcal{S}'\rangle$. If $\mathcal{S}'$ contains $\perp$, then $\mathcal{S}$ has no solutions. Otherwise, $\mathcal{S}'$ contains no equational judgements and the solutions to $\mathcal{S}$ and $\mathcal{S}'$ can be placed in one-to-one correspondence so that if the solution φ for $\mathcal{S}$ corresponds to the solution φ' for $\mathcal{S}'$ then $\theta \circ \varphi' = \varphi \pmod{\mathcal{S}}$.*

Proof. The fact that the unification algorithm terminates is an immediate consequence of Theorem 3.2. Assume that the unification algorithm makes the series of transitions

$$\mathcal{S} = \mathcal{S}_0 \overset{\sigma_1}{\Longrightarrow} \cdots \overset{\sigma_n}{\Longrightarrow} \mathcal{S}_n = \mathcal{S}' \quad (n \geq 0),$$

where $\theta = \rho_1 \circ \cdots \circ \rho_n$ (if $n = 0$ then θ is the empty substitution). Now $\mathcal{S}'$ either contains $\bot$ or contains no equations (that is, there is a unification transition available for every possible equation). In the first case, it follows immediately from Lemma 3.1 that none of the state formulas $\mathcal{S}_0, \ldots, \mathcal{S}_n$ can have a solution. In the second case, again using Lemma 3.1, it is possible to place solutions of $\mathcal{S}_i$ $(i = 0, \ldots, n)$ in one-to-one correspondence so that, if φ_i as a solution for $\mathcal{S}_i$ $(i = 0, \ldots, n)$ is in such a correspondence, we have

$$\sigma_1 \circ \varphi_1 = \varphi_0 \pmod{\mathcal{S}_0}, \ \ldots, \ \sigma_n \circ \varphi_n = \varphi_{n-1} \pmod{\mathcal{S}_n}.$$

Thus, $\sigma_1 \circ \cdots \circ \sigma_n \circ \varphi_n = \varphi_0 \pmod{\mathcal{S}}$. Therefore, solutions φ' to $\mathcal{S}'$ can be placed in one-to-one correspondence with solutions φ of $\mathcal{S}$ so that $\theta \circ \varphi' = \varphi$ $(\text{mod } \mathcal{S})$. □

A *unification problem* is a state formula that does not contain any sequent judgements. The following theorem follows immediately from the previous proposition.

Theorem 3.4. *Let $\mathcal{S}$ be a unification problem without the $\bot$ judgement. Assume that the unification algorithm returns $\langle \theta, \mathcal{S}' \rangle$ when applied to $\mathcal{S}$. Then $\mathcal{S}$ has no solution (i.e. unifier) if and only if $\mathcal{S}'$ contains $\bot$ or there are no $\mathcal{S}'$-substitutions. If $\mathcal{S}'$ does not contain $\bot$, the substitution θ represents the most general unifier of $\mathcal{S}$ in the sense that the set of solutions to $\mathcal{S}$ is exactly the set of substitutions $\theta \circ \varphi'$ where φ' ranges over $\mathcal{S}'$-substitutions.*

4 Conclusions and Future Works

We have shown that the unification problem for our particular extension of L_λ is decidable, and most importantly, there is an algorithm to compute the m.g.u if solutions exist. The extended language was intended to capture the encoding of object-level signatures. It is actually more general than necessary for this purpose, for example, we can have terms with more than one *evs* abstractions. One consideration for defining a broader class of language is that it simplifies the proofs of most lemmas and theorems, especially the one concerning closure property of the unification transitions.

The complexity issue is not addressed here. Our conjecture is that it is only slightly higher than L_λ, since the only additional step is the expansion step. Another issue is extending the algorithm to cover the multi-sorted object-level signatures, as readers might have noticed that we are currently assuming untyped object-level systems.

Acknowledgment I would like to thank Dale Miller for valuable discussions and suggestions.

References

[Hue75] Gérard Huet. A unification algorithm for typed λ-calculus. *Theoretical Computer Science*, 1:27–57, 1975.

[Mil91] Dale Miller. A logic programming language with lambda-abstraction, function variables, and simple unification. *Journal of Logic and Computation*, 1(4):497–536, 1991.

[Mil92] Dale Miller. Unification under a mixed prefix. *Journal of Symbolic Computation*, pages 321–358, 1992.

[Mil01] Dale Miller. Encoding generic judgments: Preliminary results. In R.L. Crole S.J. Ambler and A. Momigliano, editors, *Electronic Notes in Theoretical Computer Science*, volume 58. Elsevier Science Publishers, 2001. Proceedings of the MERLIN 2001 Workshop, Siena.

[MM02] Raymond McDowell and Dale Miller. Reasoning with higher-order abstract syntax in a logical framework. *ACM Transactions on Computational Logic*, 3(1):80–136, January 2002.

[MPW92] Robin Milner, Joachim Parrow, and David Walker. A calculus of mobile processes, Part II. *Information and Computation*, pages 41–77, 1992.