Bounded ACh Unification

Ajay Kumar Eeralla[1] **and Christopher Lynch**[2]

1 **Department of Computer Science**
University of Missouri
Columbia, USA
`ae266@mail.missouri.edu`

2 **Department of Mathematics and Computer Science**
Clarkson University
Potsdam, USA
`clynch@clarkson.edu`

Abstract

We consider the problem of unification modulo an equational theory ACh, which consists of a function h which is homomorphic over an associative-commutative operator $+$. Unification modulo ACh is undecidable, so we define a bounded ACh unification problem. In this bounded version of ACh unification we essentially bound the number of times h can be recursively applied to a term, and only allow solutions that satisfy this bound. There is no bound on the number of occurrences of h in a term, and the $+$ symbol can be applied an unlimited number of times. We give inference rules for solving bounded ACh unification, and we prove that the rules are sound, complete and terminating. We have implemented the algorithm in Maude and give experimental results. We argue that this algorithm is useful in cryptographic protocol analysis.

Keywords and phrases homomorphism, splitting, bounded, unification

1 Introduction

Unification is a method to find a solution for a set of equations. For instance, consider an equation $x + y \stackrel{?}{=} a + b$, where x and y are variables, and a, and b are constants. If $+$ is an uninterpreted function symbol then the equation has one solution $\{x \mapsto a,\ y \mapsto b\}$, and this unification is called syntactic unification. If the function symbol $+$ has the property commutativity then the equation has two solutions: $\{x \mapsto a,\ y \mapsto b\}$ and $\{x \mapsto b,\ y \mapsto a\}$; and this is called unification modulo the commutativity theory.

Unification modulo equational theories plays a significant role in symbolic cryptographic protocol analysis [7]. An overview and references for some of the algorithms may be seen in [8, 6]. One such equational theory is the distributive axioms:

$$x \times (y + z) = (x \times y) + (x \times z)$$
$$(y + z) \times x = (y \times x) + (z \times x)$$

A decision algorithm is presented for unification modulo two-sided distributivity in [12].

A sub-problem of this, unification modulo one-sided distributivity, is in greater interest since many cryptographic protocol algorithms satisfy the one-sided distributivity. In their paper [13], Tiden and Arnborg presented an algorithm for unification modulo one-sided distributivity: $x \times (y + z) = (x \times y) + (x \times z)$, and also it has been shown that it is undecidable if we add the properties of associativity $x + (y + z) = (x + y) + z$ and a one-sided unit element $x \times 1 = x$. However, some counterexamples [11] have been presented showing that the complexity of the algorithm is exponential, although they thought it was polynomial-time bounded.

For practical purposes, one-sided distributivity can be viewed as the homomorphism theory: $h(x + y) = h(x) + h(y)$, where the unary operator h distributes over the binary operator $+$. Homomorphisms are highly used in cryptographic protocol analysis. In fact, homomorphism is a common property that many election voting protocols satisfy [9].

Our goal is to present a novel construction of an algorithm to solve unification modulo the homomorphism theory over a binary symbol $+$ that also has the properties the associativity and the commutativity (ACh), which is an undecidable unification problem [10]. Given that ACh unification is undecidable but necessary to analyze cryptographic protocols, we developed an approximation of ACh unification, which we show to be decidable.

In this paper, we present an algorithm to solve a modified general unification problem modulo the ACh theory, which we call *bounded ACh unification*. We define the *h-height* of a term to be basically the number of h symbols recursively applied to each other. We then only search for ACh unifiers of a bounded h-height. The number of occurrences of the $+$ symbol is not bounded. In order to accomplish this we define the *h-depth* of a variable, which is the number of h symbols on top of a variable. We develop a set of inference rules for ACh unification, but we keep track of the h-depth of variables. If the h-Depth of any term exceeds the bound κ then the algorithm terminates with no solution. Otherwise, it gives all the unifiers or solutions to the problem.

The remainder of the paper organized as follows. We give preliminary knowledge about unification modulo equational theories, in particular, the theory of homomorphism, and the h-Depth Set in section 2, an inference system consists of rules for splitting, decomposition and so forth in section 3, proofs of correctness in section 4, implementation and experimental results in section 5, and conclusion and future work in section 6.

2 Preliminaries

The following are some of the standard definitions in unification theory; most of them are from [3, 4].

2.1 Basic notation

A signature $\mathcal{F}$ is a finite or countably infinite set of function symbols with fixed arity. A constant symbol is a function symbol with arity 0.

▶ **Definition 2.1** (Terms)**.** Let $\mathcal{F}$ be a signature and $\mathcal{V}$ be a countably infinite set of variables. The set $\mathcal{T}(\mathcal{F}, \mathcal{V})$ of all terms over $\mathcal{V}$ and $\mathcal{F}$ is inductively defined as

- $V \subseteq \mathcal{T}(\mathcal{F}, \mathcal{V})$,
- for all $n \geq 0$, $f(t_1, \ldots, t_n) \in \mathcal{T}(\mathcal{F}, \mathcal{V})$, where $f \in \mathcal{F}$ with arity n and $t_1, \ldots, t_n \in \mathcal{T}(\mathcal{F}, \mathcal{V})$.

The set of variables occurring in a term t is denoted by $Var(t)$.

A term u is a subterm of t, denoted by $t[u]$, if either $t = u$ or if $t = f(t_1, \ldots, t_n)$ and u is a subterm of t_i for some i. If t and s are two terms, then we use $t[s]$ to denote that s occurs in t as a subterm. For example, x is a variable, $f(x, y)$, and $h(y)$ are terms, and the variable x is a subterm of $f(x, y)$ and y is a subterm of $h(y)$. The top symbol of a term t is f when t is of the form $f(t_1, \ldots, t_n)$, and x when it is a variable x.

A substitution $\sigma : \mathcal{V} \to \mathcal{T}(\mathcal{F}, \mathcal{V})$ is a mapping such that $\sigma(x) \neq x$ for only finitely many xs. For substitution σ, the domain is the set of variables which are not mapped to itself, i.e., $\mathcal{Dom}(\sigma) = \{x \mid \sigma(x) \neq x\}$, the range is defined as, $\mathcal{Range}(\sigma) = \cup_{x \in \mathcal{Dom}(\sigma)}\{\sigma(x)\}$. The set of variables occurring in $\mathcal{Range}(\sigma)$ is represented by $Var(\mathcal{Range}(\sigma))$. A substitution σ can be represented explicitly as a function by a set of bindings of variables in its domain, i.e., $\{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$, where σ maps x_i to t_i for $i = 1, \ldots, n$, and maps to itself otherwise.

The identity substitution is a substitution that maps every variable to itself and it is represented by Id. A substitution σ is idempotent if it satisfies $\sigma\sigma = \sigma$. It is easy to show that σ is idempotent if and only if $\mathcal{D}om(\sigma) \cap Var(\mathcal{R}ange(\sigma)) = \emptyset$. Without loss of generality, we assume that every substitution is idempotent.

Let σ be a substitution and t be a term. The application of σ to t, denoted by $t\sigma$, is defined by

$$t\sigma = \begin{cases} \sigma(x) & \text{if} \quad t = x \\ c & \text{if} \quad t = c \\ f(t_1\sigma, \ldots, t_n\sigma) & \text{if} \quad t = f(t_1, \ldots, t_n). \end{cases}$$

The composition of two substitutions σ and θ is written as $\sigma\theta$, and is defined by $t(\sigma\theta) = (t\sigma)\theta$. Two substitutions σ and θ are equal if $x\sigma = x\theta$ for all variables x, and is denoted as $\sigma = \theta$. The restriction of a substitution σ to a set of variables $\mathcal{V}$, denoted by $\sigma|\mathcal{V}$, is the substitution which is equal to the identity everywhere except over $\mathcal{V} \cap \mathcal{D}om(\sigma)$, where it is equal to σ.

▶ **Definition 2.2** (More General Substitution). A substitution σ is more general than substitution θ if there exists a substitution η such that $\theta = \sigma\eta$, denoted as $\sigma \lesssim \theta$. Note that the relation $\lesssim$ is a quasi-ordering, i.e., reflexive and transitive.

▶ **Definition 2.3** (Unifier, Most General Unifier). A substitution σ is a unifier or solution of two terms s and t if $s\sigma = t\sigma$; it is a most general unifier if for every unifier θ of s and t, $\sigma \lesssim \theta$. Moreover, a substitution σ is a solution of set of equations if it is a solution of each of the equations. If a substitution σ is a solution of a set of equations Γ, then it is denoted by $\sigma \models \Gamma$.

A set of identities E is a subset of $\mathcal{T}(\mathcal{F}, \mathcal{V}) \times \mathcal{T}(\mathcal{F}, \mathcal{V})$ and are represented in the form $s \approx t$. An equational theory $=_E$ is induced by a set of fixed identities E and it is the least congruence relation that is closed under substitution and contains E.

▶ **Definition 2.4** (E-Unification Problem, E-Unifier, E-Unifiable). Let $\mathcal{F}$ be a signature and E be an equational theory. An E-unification problem over $\mathcal{F}$ is a finite set of equations

$$\Gamma = \{s_1 \stackrel{?}{=}_E t_1, \ldots, s_n \stackrel{?}{=}_E t_n\}$$

between terms. An E-unifier or E-solution of two terms s and t is a substitution σ such that $s\sigma =_E t\sigma$. An E-unifier of Γ is a substitution σ such that $s_i\sigma =_E t_i\sigma$ for $i = 1, \ldots, n$. The set of all E-unifiers is denoted by $\mathcal{U}_E(\Gamma)$ and Γ is called E-unifiable if $\mathcal{U}_E(\Gamma) \neq \emptyset$. If $E = \emptyset$ then Γ is a syntactic unification problem.

Let $\Gamma = \{s_1 \stackrel{?}{=}_E t_1, \ldots, s_n \stackrel{?}{=}_E t_n\}$ be a set of equations, and let θ be a substitution. We say that θ satisfies Γ modulo equational theory E if θ is an E-solution of each equation in Γ, that is, $s_i\theta =_E t_i\theta$ for $i = 1, \ldots, n$. We write it as $\theta \models_E \Gamma$. Let $\sigma = \{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$ and θ be substitutions, and let E be an equational theory. We say that θ satisfies σ in the equational theory E if $x_i\theta =_E t_i\theta$ for $i = 1, \ldots, n$. We write it as $\theta \models_E \sigma$.

▶ **Definition 2.5** (Complete Set of E-Unifiers). A complete set of E-unifiers of an E-unification problem Γ is a set S of idempotent E-unifiers of Γ such that for each $\theta \in \mathcal{U}_E(\Gamma)$ in Γ there exists $\sigma \in S$ with $\sigma \lesssim_E \theta|Var(\Gamma)$, where $Var(\Gamma)$ is the set of variables in Γ.

A complete set S of E-unifiers is minimal if for two distinct unifiers σ and θ, one is not more general than the other; i.e., if $\sigma \lesssim_E \theta|Var(\Gamma)$ and $\sigma, \theta \in S$ then $\sigma = \theta$. A minimal complete set of unifiers for a syntactic unification problem Γ has only one element if it is not empty. It is denoted by $mgu(\Gamma)$ and can be called most general unifier of unification problem Γ.

2.2 ACh Theory

The equational theory we consider is the theory of a homomorphism over a binary function symbol $+$. The symbol $+$ has the properties associativity and commutativity. We abbreviate this theory as ACh. The signature $\mathcal{F}$ includes a unary symbol h, and a binary symbol $+$, and other uninterpreted function symbols with fixed-arity. The function symbols h and $+$ in the signature $\mathcal{F}$ satisfy the following identities:

- $x + (y + z) \approx (x + y) + z$ [Associativity, **A** for short]
- $x + y \approx y + x$ [Commutativity, **C** for short]
- $h(x + y) \approx h(x) + h(y)$ [Homomorphism, **h** for short]

2.3 h-Depth Set

For convenience, we assume that our unification problem is in *flattened* form, i.e., that every equation in the problem is in one of the following forms: $x \stackrel{?}{=} y$, $x \stackrel{?}{=} h(y)$, $x \stackrel{?}{=} y_1 + \cdots + y_n$, and $x \stackrel{?}{=} f(x_1, \ldots, x_n)$, where x, y, y_i, and x_i are variables, and f is a free symbol with $n \geq 0$. The first kind of equations are called *VarVar equations*. The second kind are called *h-equations*. The third kind are called *$+$-equations*. The fourth kind are called *free equations*. It is well-known that how to convert a unification problem into flattened form.

▶ **Definition 2.6** (Graph $\mathbb{G}(\Gamma)$)**.** Let Γ be a unification problem. We define a graph $\mathbb{G}(\Gamma)$ as a graph where each node represents a variable in Γ and each edge represents a function symbol in Γ. To be exact, if an equation $w \stackrel{?}{=} f(x_1, \ldots, x_n)$, where f is a symbol with $n \geq 1$, is in Γ then the graph $\mathbb{G}(\Gamma)$ contains n edges $w \xrightarrow{f} x_1, \ldots, w \xrightarrow{f} x_n$. For a constant symbol c, if an equation $w \stackrel{?}{=} c$ is in Γ then the graph $\mathbb{G}(\Gamma)$ contains a vertex w. Finally, the graph $\mathbb{G}(\Gamma)$ contains two vertices if an equation $w \stackrel{?}{=} y$ is in Γ.

▶ **Definition 2.7** (h-Depth)**.** Let Γ be a unification problem and let x be a variable that occurs in Γ. Let h be a unary symbol and let f be a symbol (distinct from h) with arity greater than or equal to 1 and occur in Γ. We define h-Depth of a variable x as the maximum number of h-symbols along a path to x in $\mathbb{G}(\Gamma)$, and it is denoted by $h_d(x)$.

In other words, $h_d(x) = \max\{h_{dh}, h_{df}, 0\}$, where $h_{dh} = \max\{1 + h_d(y) \mid y \xrightarrow{h} x\}$ and $h_{df} = \max\{h_d(y) \mid y \xrightarrow{f} x\}$.

▶ **Definition 2.8** (h-Height)**.** Let Γ be a unification problem and let t be a term that occurs in Γ. We define h-height of a term t as the following:

$$h_h(t) = \begin{cases} h_h(t) + 1 & \text{if} \quad t = h(t) \\ \max\{h_h(t_1), \ldots, h_h(t_n)\} & \text{if} \quad t = f(t_1, \ldots, t_n), f \neq h \\ 0 & \text{if} \quad t = x \,\text{or}\, c \end{cases}$$

where f is a function symbol with arity greater than or equal to 1.

Without loss of generality, we assume that h-Depth and h-height is not defined for a variable that occurs both sides of the equation. This is because the occur check rule—concludes the problem with no solution—presented in the next section has higher priority over the h-Depth updating rules.

▶ **Definition 2.9** (h-Depth Set)**.** Let Γ be a set of equations. Let $\mathcal{V}$ be a set of variables occurring in Γ. We define a set h-Depth Set of Γ whose elements are pairs of a variable from $\mathcal{V}$ and a non-negative integer. In other words, the elements in the h-Depth Set are of the form $(x,\ c)$, where x is a variable in $\mathcal{V}$ and c is a natural number representing the h-Depth of x.

Maximum value of h-Depth Set $\triangle$ is the maximum of all c values and it is denoted by $MaxVal(\triangle)$. In other words, $MaxVal(\triangle) = \max\{c \mid (x,\ c) \in \triangle\}$.

▶ **Definition 2.10** (Bounded E-Unification Problem, Bounded E-Unifier)**.** A κ bounded E-unification problem over $\mathcal{F}$ is a finite set of equations $\Gamma = \{s_1 \overset{?}{=}_E t_1, \ldots, s_n \overset{?}{=}_E t_n\}$, $s_i, t_i \in \mathcal{T}(\mathcal{F}, \mathcal{V})$, where E is an equational theory, and κ is a positive integer. A κ bounded E-unifier or κ bounded E-solution of Γ is a substitution σ such that $s_i\sigma =_E t_i\sigma$, $h_h(s_i\sigma) \leq \kappa$, and $h_h(t_i\sigma) \leq \kappa$ for all i.

3 Inference System $\mathfrak{I}_h$

3.1 Problem Format

An inference system is a set of inference rules that transforms an equational unification problem into other. In our inference procedure, we use a set triple $\Gamma||\triangle||\sigma$, where Γ is a unification problem modulo the ACh theory, $\triangle$ is an h-Depth Set, and σ is a substitution. Let $\kappa \in \mathbb{N}$ be a bound on the h-Depth of the variables. A substitution θ satisfies the set triple $\Gamma||\triangle||\sigma$ if θ satisfies every equation in Γ and σ, $MaxVal(\triangle) \leq \kappa$, and we write that relation as $\theta \models \Gamma||\triangle||\sigma$. We also use a special set triple *Fail* for no solution in the inference procedure. Generally, the inference procedure is based on priority of rules and also uses don't care determinism when there is no priority. i.e., any one rule applied from a set of rules without priority. Initially, Γ is the non-empty set of equations to solve and the substitution σ is the identity substitution. The inference rules are applied until either the set of equations is empty with most general unifier σ or *Fail* for no solution. Of course, the substitution σ is a κ bounded E-unifier of Γ.

An inference rule is written in the following form:

$$\frac{\Gamma||\triangle||\sigma}{\Gamma'||\triangle'||\sigma'}.$$

This means that if something matches the top of this rule, then it is to be replaced with the bottom of the rule. In the proofs we will write inference rules as follows: $\Gamma||\triangle||\sigma \Rightarrow_{\mathfrak{I}_h} \{\Gamma_1||\triangle_1||\sigma_1, \cdots \Gamma_n||\triangle_n|\sigma_n\}$ meaning to branch and replace the left hand side with one of the right hand sides in each branch. The only inference rule that has more than one branch is *AC Unification*. So we often just write inference rules as follows: $\Gamma||\triangle||\sigma \Rightarrow_{\mathfrak{I}_h} \Gamma'||\triangle'||\sigma'$. Let $\mathcal{OV}$ be the set of variables occurring in the unification problem Γ and let $\mathcal{NV}$ be a new set of variables such that $\mathcal{NV} = \mathcal{V} \setminus \mathcal{OV}$. Unless otherwise stated we assume that $x, x_1, \ldots, x_n$, and $y, y_1, \ldots, y_n, z$ are variables in $\mathcal{V}$, $v, v_1, \ldots, v_n$ are in $\mathcal{NV}$, and terms $t, t_1, \ldots, t_n, s, s_1, \ldots, s_n$ in $\mathcal{T}(\mathcal{F}, \mathcal{V})$, and f and g are uninterpreted function symbols. Recall that h is a unary, and the associativity and the commutativity operator $+$. A Fresh variable is a variable that is generated by the current inference rule and has never been used before.

Every equation is assumed to be in flattened form. Flattening inference rules are given in the appendix. All the other inference rules leave the problem in flattened form, so there is no need to perform these rules again later. The process of updating the h-depth set is straightforward, so we also give those inference rules in the appendix. We assume this is done after each inference rule.

3.2 Inference Rules

This rule takes the homomorphism theory into account. In this theory, we can not solve equation $h(y) \stackrel{?}{=} x_1 + x_2$ unless y can be written as the sum of two new variables $y = v_1 + v_2$, where v_1 and v_2 are in $\mathcal{NV}$. Without loss of generality we generalize it to n variables $x_1, \ldots, x_n$.

Splitting Rule

$$\frac{\{w \stackrel{?}{=} h(y), w \stackrel{?}{=} x_1 + \cdots + x_n\} \cup \Gamma||\triangle||\sigma}{\{w \stackrel{?}{=} h(y), y \stackrel{?}{=} v_1 + \cdots + v_n, x_1 \stackrel{?}{=} h(v_1), \ldots, x_n \stackrel{?}{=} h(v_n)\} \cup \Gamma||\triangle'||\sigma}$$

where $n > 1$, $y \neq w$, $\triangle' = \{(v_1,\ 0), \ldots, (v_n,\ 0)\} \cup \triangle\}$, and $v_1, \ldots, v_n$ are fresh variables in $\mathcal{NV}$.

▶ **Example 3.1.** Solve the unification problem $\{h(h(x)) \stackrel{?}{=} y_1 + y_2\ \}$.

Still we only consider pair $\Gamma||\triangle$, since rules modifying σ are not introduced yet.

$\{h(h(x)) \stackrel{?}{=} y_1 + y_2\ \}||\{(x,\ 0), (y_1,\ 0), (y_2,\ 0)\} \stackrel{(F.B.S)^*}{\Rightarrow}$

$\{v \stackrel{?}{=} h(v_1),\ v_1 \stackrel{?}{=} h(x),\ v \stackrel{?}{=} y_1 + y_2\ \}||\{(x,\ 0), (y_1,\ 0), (y_2,\ 0), (v,\ 0), (v_1,\ 0)\} \stackrel{(Update\ h)^*}{\Rightarrow}$

$\{v \stackrel{?}{=} h(v_1),\ v_1 \stackrel{?}{=} h(x),\ v \stackrel{?}{=} y_1 + y_2\ \}||\{(x,\ 2), (y_1,\ 0), (y_2,\ 0), (v,\ 0), (v_1,\ 1)\} \stackrel{Splitting}{\Rightarrow}$

$\{v \stackrel{?}{=} h(v_1),\ v_1 \stackrel{?}{=} v_{11} + v_{12},\ y_1 \stackrel{?}{=} h(v_{11}),\ y_2 \stackrel{?}{=} h(v_{12}),\ v_1 \stackrel{?}{=} h(x)\ \}||$
$\{(x,\ 2), (y_1,\ 0), (y_2,\ 0), (v,\ 0), (v_1,\ 1), (v_{11},\ 0), (v_{12},\ 0)\} \stackrel{(Update\ h)^*}{\Rightarrow}$

$\{v \stackrel{?}{=} h(v_1),\ v_1 \stackrel{?}{=} v_{11} + v_{12},\ y_1 \stackrel{?}{=} h(v_{11}),\ y_2 \stackrel{?}{=} h(v_{12}),\ v_1 \stackrel{?}{=} h(x)\ \}||$
$\{(x,\ 2), (y_1,\ 0), (y_2,\ 0), (v,\ 0), (v_1,\ 1), (v_{11},\ 1), (v_{12},\ 1)\} \stackrel{Splitting}{\Rightarrow}$

$\{v \stackrel{?}{=} h(v_1),\ y_1 \stackrel{?}{=} h(v_{11}),\ y_2 \stackrel{?}{=} h(v_{12}),\ v_1 \stackrel{?}{=} h(x),\ x \stackrel{?}{=} v_{13} + v_{14},\ v_{11} \stackrel{?}{=} h(v_{13}),$
$v_{12} \stackrel{?}{=} h(v_{14})\ \}||\{(x,\ 2), (y_1,\ 0), (y_2,\ 0), (v,\ 0), (v_1,\ 1), (v_{11},\ 1), (v_{12},\ 1), (v_{13},\ 0), (v_{14},\ 0)\} \stackrel{(Update\ h)^*}{\Rightarrow}$

$\{v \stackrel{?}{=} h(v_1),\ y_1 \stackrel{?}{=} h(v_{11}),\ y_2 \stackrel{?}{=} h(v_{12}),\ v_1 \stackrel{?}{=} h(x),\ x \stackrel{?}{=} v_{13} + v_{14},\ v_{11} \stackrel{?}{=} h(v_{13}),$
$v_{12} \stackrel{?}{=} h(v_{14})\ \}||\{(x,\ 2), (y_1,\ 0), (y_2,\ 0), (v,\ 0), (v_1,\ 1), (v_{11},\ 1), (v_{12},\ 1), (v_{13},\ 2), (v_{14},\ 2)\}$.

We pause this until other rules are introduced.

Trivial

The Trivial inference rule is to remove trivial equations in the given problem Γ.

Trivial

$$\frac{\{t \stackrel{?}{=} t\}\ \cup\ \Gamma||\triangle||\sigma}{\Gamma||\triangle||\sigma}$$

Variable Elimination

The Variable Elimination rule is to convert the equations into assignments. In other words, it is used to find the most general unifier. The rule V.E-2 is performed last after all other inference rules have been performed. The rule V.E-1 is performed eagerly.

Variable Elimination(V.E)

1.

$$\frac{\{x \overset{?}{=} y\} \ \cup \ \Gamma||\triangle||\sigma}{\Gamma\{x \mapsto y\}||\triangle||\sigma\{x \mapsto y\} \cup \{x \mapsto y\}}$$

2.

$$\frac{\{x \overset{?}{=} t\} \ \cup \ \Gamma||\triangle||\sigma}{\Gamma||\triangle||\sigma\{x \mapsto t\} \cup \{x \mapsto t\}} \qquad \text{if } t \notin \mathcal{V} \text{ and } x \text{ does not occur in } t$$

▶ **Example 3.2.** Solve unification problem $\{x \overset{?}{=} y,\ x \overset{?}{=} h(z)\}$.

$\{x \overset{?}{=} y,\ x \overset{?}{=} h(z)\}||\{(x,\ 0),\ (y,\ 0),\ (z,\ 0)\}||\emptyset \overset{Update\ h}{\Rightarrow}$
$\{x \overset{?}{=} y,\ x \overset{?}{=} h(z)\}||\{(x,\ 0),\ (y,\ 0),\ (z,\ 1)\}||\emptyset \overset{V.E-1}{\Rightarrow}$
$\{y \overset{?}{=} h(z)\}||\{(x,\ 0),\ (y,\ 0),\ (z,\ 1)\}||\{x \mapsto y\} \overset{V.E-2}{\Rightarrow}$
$\emptyset||\{(x,\ 0),\ (y,\ 0),\ (z,\ 1)\}||\{x \mapsto h(z),\ y \mapsto h(z)\}$.

The substitution $\{x \mapsto h(z),\ y \mapsto h(z)\}$ is the most general unifier of the given problem $\{x \overset{?}{=} y,\ x \overset{?}{=} h(z)\}$.

Now we can resume the inference procedures for the previous Example 3.1. And also note that we consider all the set triple from now onwards.

Example 3.1.
$\{v \overset{?}{=} h(v_1),\ y_1 \overset{?}{=} h(v_{11}),\ y_2 \overset{?}{=} h(v_{12}),\ v_1 \overset{?}{=} h(x),\ x \overset{?}{=} v_{13}+v_{14},\ v_{11} \overset{?}{=} h(v_{13}), v_{12} \overset{?}{=} h(v_{14})\}||$
$\{(x,\ 2),(y_1,\ 0),(y_2,\ 0),(v,\ 0),(v_1,\ 1),(v_{11},\ 1),(v_{12},\ 1),(v_{13},\ 2),(v_{14},\ 2)\}||\emptyset\} \overset{V.E-2}{\Rightarrow}$
$\{y_1 \overset{?}{=} h(v_{11}),\ y_2 \overset{?}{=} h(v_{12}),\ v_1 \overset{?}{=} h(x),\ x \overset{?}{=} v_{13}+v_{14},\ v_{11} \overset{?}{=} h(v_{13}), v_{12} \overset{?}{=} h(v_{14})\}||$
$\{(x,\ 2),(y_1,\ 0),(y_2,\ 0),(v,\ 0),(v_1,\ 1),(v_{11},\ 1),(v_{12},\ 1),(v_{13},\ 2),(v_{14},\ 2)\}||\{v \mapsto h(v_1)\} \overset{V.E-2}{\Rightarrow}$
$\{y_1 \overset{?}{=} h(v_{11}),\ y_2 \overset{?}{=} h(v_{12}),\ x \overset{?}{=} v_{13}+v_{14}, v_{11} \overset{?}{=} h(v_{13}), v_{12} \overset{?}{=} h(v_{14})\}||\{(x,\ 2),(y_1,\ 0),(y_2,\ 0),$
$(v,\ 0),(v_1,\ 1),(v_{11},\ 1),(v_{12},\ 1),(v_{13},\ 2),(v_{14},\ 2)\}||\{v \mapsto h(h(x)),\ v_1 \mapsto h(x)\} \overset{V.E-2}{\Rightarrow}$
$\{y_1 \overset{?}{=} h(v_{11}),\ y_2 \overset{?}{=} h(v_{12}), v_{11} \overset{?}{=} h(v_{13}),\ v_{12} \overset{?}{=} h(v_{14})\}||\{(x,\ 2),(y_1,\ 0),(y_2,\ 0),(v,\ 0),(v_1,\ 1),$
$(v_{11},\ 1),(v_{12},\ 1),(v_{13},\ 2),(v_{14},\ 2)\}||\{v \mapsto h(h(v_{13}+v_{14})),\ v_1 \mapsto h(v_{13}+v_{14}),$
$x \mapsto v_{13}+v_{14}\} \overset{V.E-2}{\Rightarrow} \{y_2 \overset{?}{=} h(v_{12}),\ v_{11} \overset{?}{=} h(v_{13}),\ v_{12} \overset{?}{=} h(v_{14})\ \}||\{(x,\ 2),(y_1,\ 0),(y_2,\ 0),$
$(v,\ 0),(v_1,\ 1),(v_{11},\ 1),(v_{12},\ 1),(v_{13},\ 2),(v_{14},\ 2)\}||\{v \mapsto h(h(v_{13}+v_{14})), v_1 \mapsto h(v_{13}+v_{14}),$
$x \mapsto v_{13}+v_{14},\ y_1 \mapsto h(v_{11})\} \overset{V.E-2}{\Rightarrow} \{v_{11} \overset{?}{=} h(v_{13}),\ v_{12} \overset{?}{=} h(v_{14})\ \}||\{(x,\ 2),(y_1,\ 0),(y_2,\ 0),(v,\ 0),$
$(v_1,\ 1),(v_{11},\ 1),(v_{12},\ 1),(v_{13},\ 2),(v_{14},\ 2)\}||\{v \mapsto h(h(v_{13}+v_{14})), v_1 \mapsto h(v_{13}+v_{14}),$
$x \mapsto v_{13}+v_{14}, y_1 \mapsto h(v_{11}),\ y_2 \mapsto h(v_{12})\} \overset{V.E-2}{\Rightarrow} \{v_{12} \overset{?}{=} h(v_{14})\ \}||\{(x,\ 2),(y_1,\ 0),(y_2,\ 0),(v,\ 0),$
$(v_1,\ 1),(v_{11},\ 1),(v_{12},\ 1),(v_{13},\ 2),(v_{14},\ 2)\}||\{v \mapsto h(h(v_{13}+v_{14})),\ v_1 \mapsto h(v_{13}+v_{14}),$
$x \mapsto v_{13}+v_{14},\ y_1 \mapsto h(h(v_{13})),\ y_2 \mapsto h(v_{12}), v_{11} \mapsto h(v_{13})\} \overset{V.E-2}{\Rightarrow} \emptyset||\{(x,\ 2),(y_1,\ 0),(y_2,\ 0),(v,\ 0),$
$(v_1,\ 1),(v_{11},\ 1),(v_{12},\ 1),(v_{13},\ 2),(v_{14},\ 2)\}||\{v \mapsto h(h(v_{13}+v_{14})),\ v_1 \mapsto h(v_{13}+v_{14}),$
$x \mapsto v_{13}+v_{14},\ y_1 \mapsto h(h(v_{13})), y_2 \mapsto h(h(v_{14})), v_{11} \mapsto h(h(v_{14})),\ v_{12} \mapsto h(v_{14})\}$.
So, the problem $\{h(h(x)) \overset{?}{=} y_1+y_2\}$ has the most general unifier$\{x \mapsto v_{13}+v_{14},$
$y_1 \mapsto h(h(v_{13})),\ y_2 \mapsto h(h(v_{14}))\}$.

Decomposition

The Decomposition rule decomposes an equation into several sub-equations if both sides top symbol matches.

Decomposition(Decomp)

$$\frac{\{x \overset{?}{=} f(s_1, \ldots, s_n), x \overset{?}{=} f(t_1, \ldots, t_n)\} \cup \Gamma||\triangle||\sigma}{\{x \overset{?}{=} f(t_1, \ldots, t_n),\ s_1 \overset{?}{=} t_1, \ldots, s_n \overset{?}{=} t_n\}\ \cup\ \Gamma||\triangle||\sigma} \qquad \text{if } f \neq +$$

▶ **Example 3.3.** Solve the unification problem $\{h(h(x)) \overset{?}{=} h(h(y))\}$.

$\{h(h(x)) \overset{?}{=} h(h(y))\}||\{(x,\ 0),\ (y,\ 0)\}||\emptyset \overset{(Flatten)^*}{\Rightarrow}$
$\{v \overset{?}{=} h(v_1),\ v_1 \overset{?}{=} h(x),\ v \overset{?}{=} h(v_2),\ v_2 \overset{?}{=} h(y)\}||\{(x,\ 0),\ (y,\ 0), (v,\ 0), (v_1,\ 0), (v_2,\ 0)\}||\emptyset \overset{(Update\ h)^*}{\Rightarrow}$
$\{v \overset{?}{=} h(v_1),\ v_1 \overset{?}{=} h(x),\ v \overset{?}{=} h(v_2),\ v_2 \overset{?}{=} h(y)\}||\{(x,\ 2),\ (y,\ 2), (v,\ 0), (v_1,\ 1), (v_2,\ 1)\}||\emptyset \overset{Dcomp}{\Rightarrow}$
$\{v \overset{?}{=} h(v_1),\ v_1 \overset{?}{=} v_2,\ v_1 \overset{?}{=} h(x),\ v_2 \overset{?}{=} h(y)\}||\{(x,\ 0),\ (y,\ 0), (v,\ 2), (v_1,\ 1), (v_2,\ 1)\}||\emptyset \overset{V.E-1}{\Rightarrow}$
$\{v \overset{?}{=} h(v_2),\ v_2 \overset{?}{=} h(x),\ v_2 \overset{?}{=} h(y)\}||\{(x,\ 2),\ (y,\ 2), (v,\ 0), (v_1,\ 1), (v_2,\ 1)\}||\{v_1 \mapsto v_2\} \overset{Dcomp}{\Rightarrow}$
$\{v \overset{?}{=} h(v_2),\ v_2 \overset{?}{=} h(x),\ x \overset{?}{=} y\}||\{(x,\ 2),\ (y,\ 2), (v,\ 0), (v_1,\ 1), (v_2,\ 1)\}||\{v_1 \mapsto v_2\} \overset{(V.E-2)^*}{\Rightarrow}$
$\emptyset||\{(x,\ 2),\ (y,\ 2), (v,\ 0), (v_1,\ 1), (v_2,\ 1)\}||\{v_1 \mapsto h(y),\ x \mapsto y,\ v \mapsto h(h(y)),\ v_2 \mapsto h(y)\}$,
where $\{x \mapsto y\}$ is the most general unifier of the problem $\{h(h(x)) \overset{?}{=} h(h(y))\}$.

AC Unification

The AC Unification rule calls an AC unification algorithm to unify the AC part of the problem. Notice that we apply AC unification only once when no other rule can apply. In this inference rule Ψ represents the set of all equations with the $+$ symbol on the right hand side. Γ represents the set of equations not containing a $+$ symbol. *Unify* is a function that returns one of the complete set of unifiers returned by the AC unification algorithm. *GETEqs* is a function that takes a substitution and returns the equational form of that substitution. In other words, $GETEqs([x_1 \mapsto t_1, \ldots, x_n \mapsto t_n]) = \{x_1 \overset{?}{=} t_1, \ldots, x_n \overset{?}{=} t_n\}$.

AC Unification

$$\frac{\Psi \cup \Gamma||\triangle||\sigma}{GETEqs(\text{unify } \Psi) \cup \Gamma||\triangle||\sigma}$$

Note that we have written the rule for one member of the complete set of AC unifiers of Ψ. This will branch on every member of the complete set of AC unifiers of Ψ.

Occur Check

It is to determine if a variable on the left hand side of an equation occurs on the other side of the equation. If it does, then there is no solution to the unification problem. This rule has the highest priority.

Occur Check(O.C)

$$\frac{\{x \overset{?}{=} f(t_1, \ldots, t_n)\} \cup \Gamma || \triangle || \sigma}{\mathit{Fail}} \qquad \text{If } x \in \mathcal{V}ar(f(t_1, \ldots, t_n)\sigma)$$

▶ **Example 3.4.** Solve the following unification problem: $\{\ x \overset{?}{=} y,\ y \overset{?}{=} z + x\ \}$.
$\{\ x \overset{?}{=} y,\ y \overset{?}{=} z + x\ \} || \{(x,\ 0), (y,\ 0), (z,\ 0)\} || \emptyset \overset{V.E-1}{\Rightarrow} \{y \overset{?}{=} z + y\ \} || \{(x,\ 0), (y,\ 0), (z,\ 0)\} ||$
$\{x \mapsto y\} \overset{O.C-I}{\Rightarrow}$ *Fail*. So, the problem $\{\ x \overset{?}{=} y,\ y \overset{?}{=} z + x\ \}$ has no solution.

Clash

This rule checks if the *top symbol* on both sides of an equation is the same. If not, then there is no solution to the problem, unless one of them is h and the other $+$.

Clash

$$\frac{\{x \overset{?}{=} f(s_1, \ldots, s_m),\ x \overset{?}{=} g(t_1, \ldots, t_n)\} \cup \Gamma || \triangle || \sigma}{\mathit{Fail}} \qquad \text{If } f \notin \{h,\ +\} \text{ or } g \notin \{h,\ +\}$$

▶ **Example 3.5.** Solve the unification problem: $\{f(x,\ y) \overset{?}{=} g(h(z))\ \}$.
$\{f(x,\ y) \overset{?}{=} g(h(z))\ \} || \{(x,\ 0), (y,\ 0), (z,\ 0)\} || \emptyset \overset{(Flatten)^*}{\Rightarrow}$
$\{v \overset{?}{=} f(x,\ y),\ v \overset{?}{=} g(v_1), v_1 \overset{?}{=} h(z)\ \} || \{(x,\ 0), (y,\ 0), (z,\ 0), (v,\ 0), (v_1,\ 0)\} || \emptyset \overset{(Update\ h)^*}{\Rightarrow}$
$\{v \overset{?}{=} f(x,\ y),\ v \overset{?}{=} h(v_1),\ v_1 \overset{?}{=} h(z)\ \} || \{(x,\ 0), (y,\ 0), (z,\ 1), (v,\ 0), (v_1,\ 0)\} || \emptyset \overset{Clash}{\Rightarrow}$ *Fail*.
Hence, the problem $\{f(x,\ y) \overset{?}{=} g(h(z))\ \}$ has no solution.

Bound Check

The Bound Check is to determine if a solution exists within the bound κ, a given maximum h-Depth of any variable in Γ. If one of the h-Depths in the h-Depth Set $\triangle$ exceeds the bound κ, then the problem has no solution.

Bound Check (B.C)

$$\frac{\Gamma || \triangle || \sigma}{\mathit{Fail}} \qquad \text{If } MaxVal(\triangle) > \kappa$$

▶ **Example 3.6.** Solve the following unification problem $\{h(y) \overset{?}{=} y + x\ \}$.
Let the bound be $\kappa = 2$.
$\{h(y) \overset{?}{=} y + x\ \} || \{(x,\ 0), (y,\ 0)\} || \emptyset \overset{F.B.S}{\Rightarrow}$
$\{v \overset{?}{=} h(y),\ v \overset{?}{=} y + x\ \} || \{(x,\ 0), (y,\ 0), (v,\ 0)\} || \emptyset \overset{Update\ h}{\Rightarrow}$
$\{v \overset{?}{=} h(y),\ v \overset{?}{=} y + x\ \} || \{(x,\ 0), (y,\ 1), (v,\ 0)\} || \emptyset \overset{Splitting}{\Rightarrow}$
$\{v \overset{?}{=} h(y),\ y \overset{?}{=} v_{11} + v_{12},\ y \overset{?}{=} h(v_{11}),\ x \overset{?}{=} h(v_{12}) || \{(x,\ 0), (y,\ 1), (v,\ 0), (v_{11},\ 0), (v_{12},\ 0)\} || \emptyset \overset{(Update\ h)^*}{\Rightarrow}$
$\{v \overset{?}{=} h(y),\ y \overset{?}{=} v_{11} + v_{12},\ y \overset{?}{=} h(v_{11}),\ x \overset{?}{=} h(v_{12}) || \{(x,\ 0), (y,\ 1), (v,\ 0), (v_{11},\ 2), (v_{12},\ 1)\} || \emptyset \overset{Splitting}{\Rightarrow}$
$\{v \overset{?}{=} h(y),\ v_{11} \overset{?}{=} v_{13} + v_{14},\ v_{11} \overset{?}{=} h(v_{13}),\ v_{12} \overset{?}{=} h(v_{14}),\ y \overset{?}{=} h(v_{11}),\ x \overset{?}{=} h(v_{12}) ||$
$\{(x,\ 0), (y,\ 1), (v,\ 0), (v_{11},\ 2), (v_{12},\ 1), (v_{13},\ 0), (v_{14},\ 0)\} || \emptyset \overset{(Update\ h)^*}{\Rightarrow}$
$\{v \overset{?}{=} h(y),\ v_{11} \overset{?}{=} v_{13} + v_{14},\ v_{11} \overset{?}{=} h(v_{13}),\ v_{12} \overset{?}{=} h(v_{14}),\ y \overset{?}{=} h(v_{11}),\ x \overset{?}{=} h(v_{12}) ||$

$\{(x,\ 0),(y,\ 1),(v,\ 0),(v_{11},\ 2),(v_{12},\ 1),(v_{13},\ 3),(v_{14},\ 2)\}||\emptyset \overset{B.C}{\Rightarrow}$ *Fail.*
Since $MaxVal(\triangle) = 3 > \kappa$, the problem $\{h(y) \overset{?}{=} y + x\ \}$ has no solution within the given bound.

4 Proof of Correctness

We give proof details for the termination, soundness, and completeness of our inference system.

4.1 Termination

We need to show that this process will eventually halt.

▶ **Lemma 4.1.** *There is no infinite sequence of inference rules*

Proof. First notice that at some point all the Decomp rules not involving h will eventually be performed. That is because when we perform Decomp on top symbol f one occurrence of f disappear, and no rule can make them come back. So from now on we assume that all such rules have been performed.

Let us call $x \overset{?}{=} h(y)$ an *h-rule of depth i* if $h_d(x) = i$. A splitting rule involving $x \overset{?}{=} h(y)$ and $x \overset{?}{=} v_1 + v_2$ is called a *splitting rule of depth i*. We say that the variable x is split. A Decomp rule involving $x \overset{?}{=} h(y)$ and $x \overset{?}{=} h(z)$ is called an *h-Decomp rule of depth i*. We say that x is h-decomposed.

We first show that at some point all splitting rules of depth 0 and h-Decomp rules of depth 0 will have been performed. We notice that after AC Unification is called for the first time, any variable appearing in a VarVar equation or a $+$-equation will either appear exactly once in the left hand side of one of those equations and never on the right hand side, or else it will never appear on the left hand side of one of those equations. This is because the AC unification rule creates equations from substitutions, which have this property. Also, the AC unification rule, the V.E-1 rule and the Trivial rule will not change this property. Therefore when a variable x is split, all of the occurrences of x in $+$ equations and VarVar equations disappear. If x has depth 0 then it cannot occur on the right hand side of an h-equation. So after this split, variable x cannot be split anymore. Also, h can then only be h-decomposed a finite number of times, because each time eliminates an h-equation with x on the left hand side, and no new ones of depth 0 can be created.

We want to show by induction that all splits and h-decomps will eventually be performed. Suppose that all of them at depth i have been performed at some point. We will show that at some point all of them at depth $i+1$ will be performed. Again we notice that after AC Unification is called for the first time, any variable appearing in a VarVar equation or a $+$-equation will either appear exactly once in the left hand side of one of those equations and never on the right hand side, or else it will never appear on the left hand side of one of those equations. This is because the AC unification rule creates equations from substitutions, which have this property. Also, the AC unification rule, the V.E-1 rule and the Trivial rule will not change this property. Therefore when a variable x is split, all of the occurrences of x in $+$ equations and VarVar equations disappear. If x has depth $i+1$ then it cannot occur on the right hand side of an h-equation that can possibly be split again. This is a result of our induction hypothesis. So after this split, variable x cannot be split anymore. Also, h can then only be h-decomposed a finite number of times, because each time eliminates an h-equation with x on the left hand side, and no new ones of depth $i+1$ can be created.

Because of our Bound Check rule, all splits and h-Decomp rules will eventually be performed. From now on we assume that they have all been performed. Assume all V.E-1 rules and Trivial rules that currently exist have been performed. Then suppose we perform AC Unification. This will not create any applications of Trivial or V.E-1. Therefore the process will be finished here. The only thing left is the performance of V.E-2 rules at the end, which trivially halts because they reduce the number of equations. ◀

4.2 Soundness

We prove that our inference system is truth-preserving.

▶ **Lemma 4.2.** *Let $\Gamma||\triangle||\sigma \Rightarrow_{\mathfrak{I}_h} \{\Gamma_1||\triangle_1||\sigma_1, \cdots \Gamma_n||\triangle_n|\sigma_n\}$ be an inference rule. Let θ be a substitution such that $\theta \models \Gamma_i||\triangle_i||\sigma_i$. Then $\theta \models \Gamma||\triangle||\sigma$.*

Proof. We prove this for each rule. Trivial: It is trivially true.
Splitting:

$$\frac{\{w \overset{?}{=} h(y), w \overset{?}{=} x_1 + x_2\} \cup \Gamma||\triangle||\sigma}{\{w \overset{?}{=} h(y), y \overset{?}{=} v_1 + v_2, x_1 \overset{?}{=} h(v_1), x_2 \overset{?}{=} h(v_2)\} \cup \Gamma||\triangle'||\sigma}$$

Let θ be a substitution. Assume that θ satisfies $\{w \overset{?}{=} h(y), y \overset{?}{=} v_1 + v_2, x_1 \overset{?}{=} h(v_1), x_2 \overset{?}{=} h(v_2)\} \cup \Gamma$. Then we have that $w\theta \overset{?}{=} h(y)\theta$, $y\theta \overset{?}{=} (v_1 + v_2)\theta$, $x_1\theta \overset{?}{=} h(v_1)\theta$ and $x_2\theta \overset{?}{=} h(v_2)\theta$. This implies that $w\theta \overset{?}{=} h(y\theta)$, $y\theta \overset{?}{=} v_1\theta + v_2\theta$, $x_1\theta \overset{?}{=} h(v_1\theta)$ and $x_2\theta \overset{?}{=} h(v_2\theta)$. In order to prove that θ satisfies $\{w \overset{?}{=} h(y), w \overset{?}{=} x_1 + x_2\}$, it is enough to prove θ satisfies the equation $w \overset{?}{=} x_1 + x_2$. By considering the right side term $x_1 + x_2$ and after applying the substitution, we get $(x_1 + x_2)\theta \overset{?}{=} x_1\theta + x_2\theta \overset{?}{=} h(v_1\theta) + h(v_2\theta)$. By the homomorphism theory, we write that $h(v_1\theta) + h(v_2\theta) \overset{?}{=} h(v_1\theta + v_2\theta)$. Then $h(v_1\theta + v_2\theta) \overset{?}{=} h(y\theta) \overset{?}{=} w\theta$. Hence, θ satisfies $w \overset{?}{=} x_1 + x_2$.
Variable Elimination:

1.
$$\frac{\{x \overset{?}{=} y\} \;\cup\; \Gamma||\triangle||\sigma}{\Gamma[x \mapsto y]||\triangle||\sigma[x \mapsto y] \cup \{x \mapsto y\}}$$
Assume that $\theta \models \Gamma[x \mapsto y]||\triangle||\sigma[x \mapsto y] \cup \{x \mapsto y\}$. This means that θ satisfies $\Gamma[x \mapsto y]$ and $\sigma[x \mapsto y] \cup \{x \mapsto y\}$. Now, we have to prove that θ satisfies $\{x \overset{?}{=} y\}, \Gamma$, and σ. But θ satisfies $x \mapsto y$ means that $x\theta \overset{?}{=} y\theta$. Γ is $\Gamma[x \mapsto y]$ but without replacing x with y. Since $y\theta \overset{?}{=} x\theta$, the substitution θ satisfies $y \mapsto x$. Hence, we conclude that θ satisfies Γ and σ .

2.
$$\frac{\{x \overset{?}{=} t\} \;\cup\; \Gamma||\triangle||\sigma}{\Gamma||\triangle||\sigma[x \mapsto t] \cup \{x \mapsto t\}}$$
We have that θ satisfies Γ and $\sigma[x \mapsto t] \cup \{x \mapsto t\}$. Now, we have to prove that θ satisfies $\{x \overset{?}{=} t\}$ and σ. By the definition of $\theta \models \Gamma$, we have $x\theta \overset{?}{=} t\theta$ and it is enough to prove that θ satisfies σ. Let $w \mapsto s[x]$ be an assignment in σ. After applying $x \mapsto t$ on σ, the assignment $w \mapsto s[x]$ becomes $w \mapsto s[t]$. We also know that θ satisfies $\sigma[x \mapsto t]$ implies that θ also satisfies $w \mapsto s[t]$. Then by the definition, we write that $w\theta \overset{?}{=} s[t\theta] \overset{?}{=} s[x\theta]$. This means that θ satisfies the assignment $w \mapsto s[x]$. Hence, θ satisfies σ.

Decomposition:

$$\frac{\{x \overset{?}{=} f(s_1, s_2, ..., s_n), x \overset{?}{=} f(t_1, t_2, ..., t_n)\} \cup \Gamma||\triangle||\sigma}{\{x \overset{?}{=} f(t_1, t_2, ..., t_n),\ s_1 \overset{?}{=} t_1, ..., s_n \overset{?}{=} t_n\}\ \cup\ \Gamma||\triangle||\sigma}$$

Assume that $\theta \models \{x \overset{?}{=} f(t_1, t_2, ..., t_n),\ s_1 \overset{?}{=} t_1, ..., s_n \overset{?}{=} t_n\}\ \cup\ \Gamma||\triangle||\sigma$. This means that θ satisfies $\{x \overset{?}{=} f(t_1, t_2, ..., t_n),\ s_1 \overset{?}{=} t_1, ..., s_n \overset{?}{=} t_n\} \cup \Gamma$. Now we have to prove that θ satisfies $\{x \overset{?}{=} f(s_1, s_2, ..., s_n), x \overset{?}{=} f(t_1, t_2, ..., t_n)\} \cup \Gamma$. Given that θ satisfies $x \overset{?}{=} f(t_1, t_2, ...t_n)$ and it is enough to show that θ also satisfies $x \overset{?}{=} f(s_1, s_2, ...s_n)$. We write $x\theta \overset{?}{=} f(t_1, t_2, ...t_n)\theta \overset{?}{=} f(t_1\theta, t_2\theta, ...t_n\theta) \overset{?}{=} f(s_1\theta, s_2\theta, ...s_n\theta)$ since $s_1\theta \overset{?}{=} t_1\theta,\ ...., s_n\theta \overset{?}{=} t_n\theta$. So, θ satisfies $x \overset{?}{=} f(t_1, t_2, ...t_n)$ and $x \overset{?}{=} f(s_1, s_2, ...s_n)$. Hence, $\theta \models \{x \overset{?}{=} f(s_1, s_2, ...s_n),\ x \overset{?}{=} f(t_1, t_2, ...t_n)\}$.
AC Unification:

$$\frac{\Psi \cup \Gamma||\triangle||\sigma}{GETEqs(\text{unify } \Psi) \cup \Gamma||\triangle||\sigma}$$

Given that $\theta \models GETEqs(\text{unify } \Psi) \cup \Gamma||\triangle||\sigma$. This means that θ satisfies $GETEqs(\text{unify } \Psi) \cup \Gamma$. Which implies that θ also satisfies Ψ.

◀

4.3 Completeness

Here we prove that our inference system never loses any solution. We recall formal definition of rewrite system from [3].

▶ **Definition 4.3.** A rewrite rule is an oriented pair $l \to r$, where $l \notin \mathcal{V}$ and $l, r \in \mathcal{T}(\mathcal{F}, \mathcal{V})$. A rewrite system R is a set of rewrite rules. We consider the two convergent rewrite systems R_1 and R_2 modulo $\mathcal{AC}$ where $R_1 = \{h(x+y) \to h(x)+h(y)\}$ and $R_2 = \{h(x)+h(y) \to h(x+y)\}$.

▶ **Lemma 4.4.** *Let $\Gamma||\triangle||\sigma$ be a set triple. Let $\Gamma||\triangle||\sigma \Rightarrow_{\mathfrak{I}_h} \{\Gamma_1||\triangle_1||\sigma_1, \cdots \Gamma_n||\triangle_n|\sigma_n\}$ be an inference rule. If $\theta \models \Gamma||\triangle||\sigma$, then there exists an i and a θ', whose domain is the variables in $Var(\Gamma_i) \setminus Var(\Gamma)$, such that $\theta\theta' \models \Gamma_i||\triangle_i||\sigma_i$.*

Proof. Trivial: It is trivially true.
Occur Check: In the homomorphism theory, no term can be equal to a subterm of itself. This is because the number of + symbols and h-Depth of each variable stay the same with the application of the homomorphism equation $h(x+y) \overset{?}{=} h(x)+h(y)$. So, the given problem has no solution in the homomorphism theory.
Bound Check: We see that there exists a variable y with the h-Depth $\kappa + 1$ in the graph, that is, there is a variable x above y with $\kappa + 1$ h-symbols below it. Let θ be a solution of the unification problem Γ. Then the term $x\theta$ has the h-height $\kappa + 1$, but the term $x\theta$ is also a subterm of some $s_i\theta$ or $t_i\theta$ in the original unification problem. Hence, the unification problem Γ has no solution within the given bound κ.
Variable Elimination: It is trivially true.
Clash: We don't have a rewrite rule that deals with the uninterpreted function symbols, i.e. the function symbols which are not in $\{h,\ +\}$. So the given problem has to have no solution.
Splitting: We have to make sure that we never lose any solution with this rule. Here we consider the rewrite system R_1 which has the rewrite rule $h(x + y) \to h(x) + h(y)$. In order to apply this rule the term under the h should be the sum of two variables. The problem $\{h(y) \overset{?}{=} x_1 + x_2\}$ is replaced by the set $\{h(v_1 + v_2) \overset{?}{=} x_1 + x_2\}$ with the substitution

$\{y \mapsto v_1 + v_2\}$. Then we have the equation with the reduced term in R_1 is the equation $h(v_1) + h(v_2) \overset{?}{=} x_1 + x_2$, and the substitution $\{y \mapsto v_1 + v_2, x_1 \mapsto h(v_1),\ x_2 \mapsto h(v_2)\}$. Hence, we never lose any solution here.
Decomposition: If f is the top symbol on both sides of an equation then there is no other rule to solve it except the Decomposition rule, where $f \neq h$ and $f \neq +$. So, we never lose any solution.

AC Unification: Assume that there are equations in Ψ that contains both h and $+$ symbols. Then the built-in unification algorithm on Ψ may loose solutions since h is considered as an uninterpreted symbol. However, the missing solutions are regained on the other branch of the unification problem.

To cover the case where the top symbol is h for the terms on both sides of an equation, we consider the rewrite system R_2 which has the rewrite rule $h(x)+h(y) \to h(x+y)$. In the homomorphism theory with the rewrite system R_2, we cannot reduce the term $h(t)$. So, we solve the equation of the form $h(t_1) \overset{?}{=} h(t_2)$ only with the Decomposition rule. Hence, we never lose any solution here too.

◀

5 Implementation

We have implemented the algorithm in Maude [5]. We chose the Maude language because the inference rules are very similar to the rules of Maude and an implementation will be integrated into the Maude-NPA tool at some time. The Maude-NPA tool is written in Maude. The system specifications are Ubuntu 14.04 LTS, Intel Core i5 3.20 GHz, and 8 GiB RAM with Maude 2.6.

We give a table to show some of our results. In the given table, we use five columns: Unification problem, Real Time, time to terminate the program in ms(milli seconds), Solution either Fail for no solution or Yes for solutions, # Sol. for number of solutions, and Bound κ. It makes sense that the real time keeps increasing as the given h-Depth κ increases for the first problem where the other problems give solutions, but in either case the program terminates.

6 Conclusion

We introduced a set of inference rules to solve the unification problem modulo the homomorphism theory h over an AC symbol $+$, by enforcing bound k on the h-Depth of any term. We proved that these inference rules are sound, complete, and terminating. Our algorithm finds all the solutions or unifiers within the given h-Depth k. We implemented the algorithm in Maude because the inference rules are easy to write in Maude and also we hope to incorporate them in the Maude-NPA tool, a protocol analyzer written in Maude. Our work on this topic actually came out of work on the Maude-NPA tool. Homomorphism is a property which is very common in cryptographic algorithms. So, it is important to analyze cryptographic protocols in the homomorphism theory. Some of the algorithms and details in this direction can be seen in [2, 6, 1]. However none of those results perform ACh unification because that is undecidable. One way around this is to assume that an identity and an inverse exist, but because of the way the Maude-NPA works it would still be necessary to unify modulo ACh. So an unification algorithm there becomes crucial. We

Table 1 Tested results with ACh-unification algorithm

Unification Problem	Real Time	Solution	# Sol.	Bound
$\{h(y) \overset{?}{=} y + x\}$	674ms	Fail	0	10
$\{h(y) \overset{?}{=} y + x\}$	15880ms	Fail	0	20
$\{h(y) \overset{?}{=} x_1 + x_2\}$	5m	Yes	1	10
$\{h(h(x)) \overset{?}{=} h(h(y))\}$	2ms	Yes	1	10
$\{x + y_1 \overset{?}{=} x + y_2\}$	3ms	Yes	1	10
$\{v \overset{?}{=} x + y, v \overset{?}{=} w + z, s \overset{?}{=} h(t)\}$	46ms	Yes	10	10
$\{v \overset{?}{=} x_1 + x_2, v \overset{?}{=} x_3 + x_4, x_1 \overset{?}{=} h(y), x_2 \overset{?}{=} h(y)\}$	100ms	Yes	6	10
$\{h(h(x)) \overset{?}{=} v + w + y + z\}$	224ms	Yes	1	10
$\{v \overset{?}{=} (h(x) + y), v \overset{?}{=} w + z\}$	55ms	Yes	7	10
$\{f(x, y) \overset{?}{=} h(x_1)\}$	0ms	Fail	0	10
$\{f(x_1, y_1) \overset{?}{=} f(x_2, y_2)\}$	1ms	Yes	1	10
$\{v \overset{?}{=} x_1 + x_2, v \overset{?}{=} x_3 + x_4\}$	17ms	Yes	7	10
$\{f(x_1, y_1) \overset{?}{=} g(x_2, y_2)\}$	0ms	Fail	0	10
$\{h(y) \overset{?}{=} x, y \overset{?}{=} h(x)\}$	0ms	Fail	0	10

believe our approximation is a good way to deal with it. We also tested some problems and the results are shown in Table 1.

References

1 S. Anantharaman, H. Lin, C. Lynch, P. Narendran, and M. Rusinowitch. Cap unification: application to protocol security modulo homomorphic encryption. In *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, pages 192–203. ACM.

2 S. Anantharaman, H. Lin, C. Lynch, P. Narendran, and M. Rusinowitch. Unification Modulo Homomorphic Encryption. In *booktitle of Automated Reasoning*, pages 135–158. Springer, 2012.

3 F. Baader and T. Nipkow. Term Rewriting and All that. Cambridge University Press, 1998.

4 F. Baader and W. Snyder. Unification Theory. In *Handbook of Automated Reasoning*, pages 447–533. Elsevier, 2001.

5 M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, and Carolyn L. Talcott. All About Maude - A High-Performance Logical Framework, How to Specify, Program and Verify Systems in Rewriting Logic. Springer, 2007.

6 S. Escobar, D. Kapur, C. Lynch, C. Meadows, J. Meseguer, P. Narendran, and R. Sasse. Protocol Analysis in Maude-NPA Using Unification Modulo Homomorphic Encryption. In *Proceedings of the 13th International ACM SIGPLAN Symposium on Principles and Practices of Declarative Programming*, pages 65–76. ACM, 2011.

7 S. Escobar, C. Meadows, and J. Meseguer. Maude-Npa: cryptographic protocol analysis modulo equational properties. In *Foundations of Security Analysis and Design V: FOSAD 2007/2008/2009 Tutorial Lectures*, pages 1–50. Springer, 2007.

8 D. Kapur, P. Narendran, and L. Wang. An E-unification Algorithm for Analyzing Protocols That Use Modular Exponentiation. In *Rewriting Techniques and Applications*, pages 165–179. Springer, 2003.

9 S. Kremer, M. Ryan, and B. Smyth. Election Verifiability in Electronic Voting Protocols. In *Computer Security – ESORICS*, pages 389–404. Springer, 2010.

10 P. Narendran. Solving Linear Equations over Polynomial Semirings. In *Proceedings 11th Annual IEEE Symposium on Logic in Computer Science*, pages 466–472. IEEE Computer Society, 1996.

11 P. Narendran, Andrew M. Marshall, and B. Mahapatra. On the Complexity of the Tiden-Arnborg Algorithm for Unification modulo One-Sided Distributivity. In *Proceedings 24th International Workshop on Unification*, pages 54–63. Open Publishing Association, 2010.

12 M. Schmidt-Schauß. A Decision Algorithm for Distributive Unification. In *Theoretical Computer Science*, pages 111–148. Elsevier, 1998.

13 E. Tidén and Stefan Arnborg. Unification Problems with One-Sided Distributivity. In *booktitle of Symbolic Computation*, pages 183–202. Springer, 1987.

Appendices

A Flattening

Here we present a set of inference rules for flattening.

Flatten Both Sides(F.B.S)

$$\frac{\{t_1 \overset{?}{=} t_2\} \ \cup\ \Gamma||\triangle||\sigma}{\{v \overset{?}{=} t_1,\ v \overset{?}{=} t_2\} \ \cup\ \Gamma||\{(v,\ 0)\} \cup \triangle||\sigma} \quad \text{if } t_1 \text{ and } t_2 \notin \mathcal{V}$$

where v is a fresh variable from $\mathcal{NV}$

Flatten Left $+$(L.$+$)

$$\frac{\{t \overset{?}{=} t_1 + t_2\} \ \cup\ \Gamma||\triangle||\sigma}{\{t \overset{?}{=} v + t_2,\ v \overset{?}{=} t_1\} \ \cup\ \Gamma||\{(v,\ 0)\} \cup \triangle||\sigma} \qquad \text{if } t_1 \notin \mathcal{V}$$

where v is a fresh variable from $\mathcal{NV}$

Flatten Right $+$(R.$+$)

$$\frac{\{t \overset{?}{=} t_1 + t_2\} \ \cup\ \Gamma||\triangle||\sigma}{\{t \overset{?}{=} t_1 + v,\ v \overset{?}{=} t_2\} \ \cup\ \Gamma||\{(v,\ 0)\} \cup \triangle||\sigma} \qquad \text{if } t_2 \notin \mathcal{V}$$

where v is a fresh variable from $\mathcal{NV}$

Flatten Under h(F.h)

$$\frac{\{t_1 \overset{?}{=} h(t)\} \ \cup\ \Gamma||\triangle||\sigma}{\{t_1 \overset{?}{=} h(v),\ v \overset{?}{=} t\} \ \cup\ \Gamma||\{(v,\ 0)\} \cup \triangle||\sigma} \qquad \text{if } t \notin \mathcal{V}$$

where v is a fresh variable from $\mathcal{NV}$

▶ **Example 1.1.** Solve the unification problem:$\{h(h(x)) \overset{?}{=} (s+w)+(y+z)\}$.
We only consider the set of equations Γ here, not the full triple.
$\{h(h(x)) \overset{?}{=} (s+w)+(y+z)\} \overset{F.B.S}{\Rightarrow}$
$\{v \overset{?}{=} h(h(x)),\ v \overset{?}{=} (s+w)+(y+z)\} \overset{L.+}{\Rightarrow}$
$\{v \overset{?}{=} h(h(x)),\ v \overset{?}{=} v_1+(y+z),\ v_1 \overset{?}{=} s+w\} \overset{R.+}{\Rightarrow}$

$\{v \overset{?}{=} h(h(x)),\ v \overset{?}{=} v_1 + v_2,\ v_1 \overset{?}{=} s + w,\ v_2 \overset{?}{=} y + z\} \overset{F.h}{\Rightarrow}$
$\{v \overset{?}{=} h(v_3),\ v_3 \overset{?}{=} h(x),\ v \overset{?}{=} v_1 + v_2,\ v_1 \overset{?}{=} s + w,\ v_2 \overset{?}{=} y + z\}$.
We see that each equation in the set $\{v \overset{?}{=} h(v_3),\ v_3 \overset{?}{=} h(x),\ v \overset{?}{=} v_1 + v_2, v_1 \overset{?}{=} s + w,$ $v_2 \overset{?}{=} y + z\}$ is in the flattened form.

B Update h-Depth Set

We also present a set of inference rules to update the h-Depth Set.

Update h

$$\frac{\{x \overset{?}{=} h(y)\}\ \cup\ \Gamma||\{(x,\ c_1),\ (y,\ c_2)\} \cup \triangle||\sigma}{\{x \overset{?}{=} h(y)\}\ \cup\ \Gamma||\{(x,\ c_1),\ (y,\ c_1+1)\} \cup \triangle||\sigma} \qquad \text{If } c_2 < (c_1+1)$$

▶ **Example 2.1.** Solve the unification problem: $\{x \overset{?}{=} h(h(h(y)))\}$.

We only consider the pair $\Gamma||\triangle$ since σ does not change at this step.
$\{x \overset{?}{=} h(h(h(y)))\}||\{(x,\ 0),\ (y,\ 0)\} \overset{(F.h)^*}{\Rightarrow}$
$\{x \overset{?}{=} h(v),\ v \overset{?}{=} h(v_1),\ v_1 \overset{?}{=} h(y)\}||\{(x,\ 0),\ (y,\ 0),\ (v,\ 0),\ (v_1,\ 0)\} \overset{Update\ h}{\Rightarrow}$
$\{x \overset{?}{=} h(v),\ v \overset{?}{=} h(v_1),\ v_1 \overset{?}{=} h(y)\}||\{(x,\ 0),\ (y,\ 1),\ (v,\ 0),\ (v_1,\ 0)\} \overset{Update\ h}{\Rightarrow}$
$\{x \overset{?}{=} h(v),\ v \overset{?}{=} h(v_1),\ v_1 \overset{?}{=} h(y)\}||\{(x,\ 0),\ (y,\ 1),\ (v,\ 0),\ (v_1,\ 1)\} \overset{Update\ h}{\Rightarrow}$
$\{x \overset{?}{=} h(v),\ v \overset{?}{=} h(v_1),\ v_1 \overset{?}{=} h(y)\}||\{(x,\ 0),\ (y,\ 2),\ (v,\ 0),\ (v_1,\ 1)\} \overset{Update\ h}{\Rightarrow}$
$\{x \overset{?}{=} h(v),\ v \overset{?}{=} h(v_1),\ v_1 \overset{?}{=} h(y)\}||\{(x,\ 0),\ (y,\ 2),\ (v,\ 1),\ (v_1,\ 1)\} \overset{Update\ h}{\Rightarrow}$
$\{x \overset{?}{=} h(v),\ v \overset{?}{=} h(v_1),\ v_1 \overset{?}{=} h(y)\}||\{(x,\ 0),\ (y,\ 2),\ (v,\ 1),\ (v_1,\ 2)\} \overset{Update\ h}{\Rightarrow}$
$\{x \overset{?}{=} h(v),\ v \overset{?}{=} h(v_1),\ v_1 \overset{?}{=} h(y)\}||\{(x,\ 0),\ (y,\ 3),\ (v,\ 1),\ (v_1,\ 2)\}$.
It is true that the h-Depth of y is 3 since there are three edges labeled h from x to y, in the graph $\mathbb{G}(\Gamma)$.

Update $+$
1.

$$\frac{\{x_1 \overset{?}{=} y_1 + y_2\}\ \cup\ \Gamma||\{(x_1,\ c_1),\ (y_1,\ c_2),\ (y_2,\ c_3)\} \cup \triangle||\sigma}{\{x_1 \overset{?}{=} y_1 + y_2\}\ \cup\ \Gamma||\{(x_1,\ c_1),\ (y_1,\ c_1),\ (y_2,\ c_3)\} \cup \triangle||\sigma} \qquad \text{If } c_2 < c_1$$

2.

$$\frac{\{x_1 \overset{?}{=} y_1 + y_2\}\ \cup\ \Gamma||\{(x_1,\ c_1),\ (y_1,\ c_2),\ (y_2,\ c_3)\} \cup \triangle||\sigma}{\{x_1 \overset{?}{=} y_1 + y_2\}\ \cup\ \Gamma||\{(x_1,\ c_1),\ (y_1,\ c_2),\ (y_2,\ c_1)\} \cup \triangle||\sigma} \qquad \text{If } c_3 < c_1$$

▶ **Example 2.2.** Solve the unification problem: $\{\ z \overset{?}{=} x + y,\ x_1 \overset{?}{=} h(h(z))\}$.
Similar to the last example, we only consider the pair $\Gamma||\triangle$,
$\{\ z \overset{?}{=} x + y,\ x_1 \overset{?}{=} h(h(z))\}||\{(x,\ 0), (y,\ 0), (z,\ 0), (x_1,\ 0)\} \overset{F.h}{\Rightarrow}$
$\{\ z \overset{?}{=} x + y,\ x_1 \overset{?}{=} h(v),\ v \overset{?}{=} h(z)\}||\{(x,\ 0), (y,\ 0), (z,\ 0), (x_1,\ 0), (v,\ 0)\} \overset{(Update\ h)^*}{\Rightarrow}$
$\{\ z \overset{?}{=} x + y,\ x_1 \overset{?}{=} h(v),\ v \overset{?}{=} h(z)\}||\{(x,\ 0), (y,\ 0), (z,\ 2), (x_1,\ 0), (v,\ 1)\} \overset{Update\ +}{\Rightarrow}$
$\{\ z \overset{?}{=} x + y,\ x_1 \overset{?}{=} h(v),\ v \overset{?}{=} h(z)\}||\{(x,\ 2), (y,\ 0), (z,\ 2), (x_1,\ 0), (v,\ 1)\} \overset{Update\ +}{\Rightarrow}$

$\{\ z \overset{?}{=} x + y,\ x_1 \overset{?}{=} h(v),\ v \overset{?}{=} h(z)\}||\{(x,\ 2), (y,\ 2), (z,\ 2), (x_1,\ 0), (v,\ 1)\}$.
Since there are two edges labeled h from x_1 to z in the graph $\mathbb{G}(\Gamma)$, the h-Depth of z is 2. The h-Depths of x and y are also updated accordingly.

Now, we resume the inference procedure for Example 1.1(Appendix A) and also we consider $\triangle$ because it will be updated at this step.
$\{v \overset{?}{=} h(v_3),\ v_3 \overset{?}{=} h(x),\ v \overset{?}{=} v_1 + v_2,\ v_1 \overset{?}{=} s + w,\ v_2 \overset{?}{=} y + z\}||$
$\{(x,\ 0), (y,\ 0), (z,\ 0), (s,\ 0), (w,\ 0), (v,\ 0), (v_1,\ 0), (v_2,\ 0), (v_3,\ 0)\} \overset{Update\ h}{\Rightarrow}$
$\{v \overset{?}{=} h(v_3),\ v_3 \overset{?}{=} h(x),\ v \overset{?}{=} v_1 + v_2,\ v_1 \overset{?}{=} s + w,\ v_2 \overset{?}{=} y + z\}||$
$\{(x,\ 1), (y,\ 0), (z,\ 0), (s,\ 0), (w,\ 0), (v,\ 0), (v_1,\ 0), (v_2,\ 0), (v_3,\ 0)\} \overset{Update\ h}{\Rightarrow}$
$\{v \overset{?}{=} h(v_3),\ v_3 \overset{?}{=} h(x),\ v \overset{?}{=} v_1 + v_2,\ v_1 \overset{?}{=} s + w,\ v_2 \overset{?}{=} y + z\}||$
$\{(x,\ 1), (y,\ 0), (z,\ 0), (s,\ 0), (w,\ 0), (v,\ 0), (v_1,\ 0), (v_2,\ 0), (v_3,\ 1)\} \overset{Update\ h}{\Rightarrow}$
$\{v \overset{?}{=} h(v_3),\ v_3 \overset{?}{=} h(x),\ v \overset{?}{=} v_1 + v_2,\ v_1 \overset{?}{=} s + w,\ v_2 \overset{?}{=} y + z\}||$
$\{(x,\ 2), (y,\ 0), (z,\ 0), (s,\ 0), (w,\ 0), (v,\ 0), (v_1,\ 0), (v_2,\ 0), (v_3,\ 1)\}$.

C Termination

Here we prove the termination of Flattening rules.

Consider a multi-set $\mathcal{F}(\Gamma)$ where each element of it is the number of function symbols of an equation in Γ. In other words, for every equation E in the unification problem Γ, there is a number k: total number of function symbols in E; in the multi-set $\mathcal{F}(\Gamma)$. We define a measure of $\Gamma||\triangle||\sigma$ as the multi-set ordering $\mathcal{F}(\Gamma)_{mul}$ on $\mathcal{F}(\Gamma)$. Note that $\mathcal{F}(\Gamma)_{mul}$ is well-founded since $\mathcal{F}(\Gamma)$ is a well-ordered set.

▶ **Lemma 3.1.** *Let* $\Gamma||\triangle||\sigma$ *and* $\Gamma'||\triangle'||\sigma'$ *be two triple sets such that* $\Gamma||\triangle||\sigma \overset{Flattening}{\Rightarrow} \Gamma'||\triangle'||\sigma'$. *Then,* $\mathcal{F}(\Gamma)_{mul} > \mathcal{F}(\Gamma')_{mul}$.

Proof. We have to prove that $\mathcal{F}(\Gamma)_{mul}$ is reducing in all the cases.
Flatten Both sides: Here the equation $\{t_1 \overset{?}{=} t_2\}$ is replaced by $\{v \overset{?}{=} t_1,\ v \overset{?}{=} t_2\}$, where t_1 and t_2 are terms but not variables. Let $k_1 \geq 0$ be the number of function symbols in t_1 and let $k_2 \geq 0$ be the number of function symbols in t_2. Now, the multi-set $\mathcal{F}(\{t_1 \overset{?}{=} t_2\}) = \{k_1 + k_2\}$ and the multi-set $\mathcal{F}(\{v \overset{?}{=} t_1,\ v \overset{?}{=} t_2\}) = \{k_1,\ k_2\}$. Hence, $\mathcal{F}(\Gamma)_{mul}$ is reduced since the number $(k_1 + k_2)$ is replaced by two smaller numbers k_1, and k_2.
Flatten Left +: The equation $\{t \overset{?}{=} t_1 + t_2\}$ is replaced by $\{t \overset{?}{=} v + t_2,\ v \overset{?}{=} t_1\}$, where t_1 is not a variable. Let k_1 be the number of function symbols in t, k_2 be the number of function symbols in t_1, and k_3 be the number of function symbols in t_2. Then $k_2 \geq 1$ as t_1 is not a variable. The set $\mathcal{F}(\{t \overset{?}{=} t_1 + t_2\}) = k_1 + k_2 + k_3 + 1$. But the set $\mathcal{F}(\{t \overset{?}{=} v + t_2,\ v \overset{?}{=} t_1\}) = \{k_1 + k_3,\ k_2\}$. Hence, $\mathcal{F}(\Gamma)_{mul} > \mathcal{F}(\Gamma')_{mul}$.
Flatten Right +: In this case, the equation $\{t \overset{?}{=} t_1 + t_2\}$ is replaced by $\{t \overset{?}{=} t_1 + v,\ v \overset{?}{=} t_2\}$, where t_2 is not a variable. The argument is same as above except t_2 is not a variable instead of t_1.
Flatten Under h: The equation $\{t \overset{?}{=} h(t_1)\}$ is replaced by $\{t \overset{?}{=} h(v),\ v \overset{?}{=} t_1\}$, where t_1 is not a variable. Let k_1 be the number of function symbols in t and let $k_2 \geq 1$ be the number of number of function symbols in t_1. Then the multi-set $\mathcal{F}(\{t \overset{?}{=} h(t_1)\}) = k_1 + k_2 + 1$ since there is h symbol on top of t_1, where the multi-set $\mathcal{F}(\{t \overset{?}{=} h(v),\ v \overset{?}{=} t_1\}) = \{k_1 + 1,\ k_2\}$. Hence, $\mathcal{F}(\{t \overset{?}{=} h(t_1)\})_{mul} > \mathcal{F}(\{t \overset{?}{=} h(v),\ v \overset{?}{=} t_1\})_{mul}$. ◀

D Soundness

Here we give soundness proof for the Flattening rules.

▶ **Lemma 4.1.** *Let $\Gamma||\triangle||\sigma \overset{Flattening}{\Rightarrow} \{\Gamma_1||\triangle_1||\sigma_1, \cdots \Gamma_n||\triangle_n|\sigma_n\}$ be an inference rule. Let θ be a substitution such that $\theta \models \Gamma_i||\triangle_i||\sigma_i$. Then $\theta \models \Gamma||\triangle||\sigma$.*

Proof. We have to prove that each of the Flattening rules are truth-preserving.
Flatten Both Sides:

$$\frac{\{t_1 \overset{?}{=} t_2\} \ \cup \ \Gamma||\triangle||\sigma}{\{v \overset{?}{=} t_1, \ v \overset{?}{=} t_2\} \ \cup \ \Gamma||\{(v, \ 0)\} \cup \triangle||\sigma}$$

Let θ be a substitution, such that $\theta \models \{v \overset{?}{=} t_1, \ v \overset{?}{=} t_2\} \cup \Gamma||\{(v, \ 0)\} \cup \triangle||\sigma$. Now we have to prove that θ also satisfies $\{t_1 \overset{?}{=} t_2\} \ \cup \ \Gamma||\triangle||\sigma$. In other words, $\theta \models \{t_1 \overset{?}{=} t_2\} \ \cup \ \Gamma||\triangle||\sigma$. To prove this it is enough to prove that $\theta \models \{t_1 \overset{?}{=} t_2\}$. Given that θ satisfies $\{v \overset{?}{=} t_1, \ v \overset{?}{=} t_2\}$ implies that $v\theta \overset{?}{=} t_1\theta$ and $v\theta \overset{?}{=} t_2\theta$. We can write from the above that $t_1\theta \overset{?}{=} v\theta \overset{?}{=} t_2\theta$. Then $t_1\theta \overset{?}{=} t_2\theta$. Hence, $\theta \models \{t_1 \overset{?}{=} t_2\}$.
Flatten Left + :

$$\frac{\{t \overset{?}{=} t_1 + t_2\} \ \cup \ \Gamma||\triangle||\sigma}{\{t \overset{?}{=} v + t_2, \ v \overset{?}{=} t_1\} \ \cup \ \Gamma||\{(v, \ 0)\} \cup \triangle||\sigma}$$

Assume that $\theta \models \{t \overset{?}{=} v + t_2, \ v \overset{?}{=} t_1\} \ \cup \ \Gamma||\{(v, \ 0)\} \cup \triangle||\sigma$. We have to prove that $\theta \models \{t \overset{?}{=} t_1 + t_2\}$. We have that $t\theta \overset{?}{=} (v + t_2)\theta \Rightarrow t\theta \overset{?}{=} v\theta + t_2\theta$ and $v\theta \overset{?}{=} t_1\theta$. From $v\theta \overset{?}{=} t_1\theta$, we can write $t\theta \overset{?}{=} v\theta + t_2\theta$ as $t\theta \overset{?}{=} t_1\theta + t_2\theta$. Hence, $\theta \models \{t \overset{?}{=} t_1 + t_2\}$.
Flatten Right + :

$$\frac{\{t \overset{?}{=} t_1 + t_2\} \ \cup \ \Gamma||\triangle||\sigma}{\{t \overset{?}{=} t_1 + v, \ v \overset{?}{=} t_2\} \ \cup \ \Gamma||\{(v, \ 0)\} \cup \triangle||\sigma}$$

This proof is very similar to Flatten Left +.
Flatten Under h :

$$\frac{\{t_1 \overset{?}{=} h(t)\} \ \cup \ \Gamma||\triangle||\sigma}{\{t_1 \overset{?}{=} h(v), \ v \overset{?}{=} t\} \ \cup \ \Gamma||\{(v, \ 0)\} \cup \triangle||\sigma}$$

Assume that $\theta \models \{t_1 \overset{?}{=} h(v), \ v \overset{?}{=} t\} \ \cup \ \Gamma||\{(v, \ 0)\} \cup \triangle||\sigma$. Now, we have to prove that $\theta \models \{t_1 \overset{?}{=} h(t)\} \ \cup \ \Gamma$. By the definition of $\theta \models \{t_1 \overset{?}{=} h(v), \ v \overset{?}{=} t\}$, we have that $t_1\theta \overset{?}{=} h(v)\theta$ and $v\theta \overset{?}{=} t\theta$. From the above two equations we can write that $t_1\theta \overset{?}{=} h(v)\theta \overset{?}{=} h(v\theta) \overset{?}{=} h(t\theta) \Rightarrow t_1\theta \overset{?}{=} h(t\theta)$. Hence, $\theta \models \{t_1 \overset{?}{=} h(t)\}$. ◀

E Completeness

Here we make sure that the Flattening rules never lose any equation.

▶ **Lemma 5.1.** *Let $\Gamma||\triangle||\sigma$ be a set triple. Let $\Gamma||\triangle||\sigma \overset{Flattening}{\Rightarrow} \{\Gamma_1||\triangle_1||\sigma_1, \cdots \Gamma_n||\triangle_n|\sigma_n\}$ be an inference rule. If $\theta \models \Gamma||\triangle||\sigma$, then there exists an i and a θ', whose domain is the variables in $Var(\Gamma_i) \setminus Var(\Gamma)$, such that $\theta\theta' \models \Gamma_i||\triangle_i||\sigma_i$.*

Proof. We have to prove that it is true in each of the Flattening rules.
Flatten Both Sides: If we have $\theta \models \{t_1 \overset{?}{=} t_2\}$, we prove that θ can be extended to θ' such that $\theta' \models \{v \overset{?}{=} t_1,\ v \overset{?}{=} t_2\}$. We extend θ to θ' with the addition of an assignment $v \mapsto t_1$.
Flatten Left $+$: We add $v \mapsto t_1$ to θ, and we get θ'.
Flatten Right $+$: It can be seen that we get θ' when we add $v \mapsto t_2$ to θ.
Flatten under h: Similarly, here we add $v \mapsto t$ to θ. ◀