Universidade de Brasília

A detailed study of bounded ACh-unification

Guilherme Borges Brandão

Advisor: Dr. Daniele Nantes Sobrinho

Department of Mathematics
Universidade de Brasília

Dissertation submitted in partial fulfillment of the requirements for the degree of
Master in Mathematics

Brasília, March 6, 2024.

Ficha catalográfica elaborada automaticamente,
com os dados fornecidos pelo(a) autor(a)

BB817d Brandão, Guilherme B.
A detailed study of bounded ACh-unification / Guilherme B. Brandão; orientador Daniele Nantes Sobrinho. -- Brasília, 2024.
64 p.

Dissertação(Mestrado em Matemática) -- Universidade de Brasília, 2024.

1. Unificação. 2. Associatividade-Comutatividade. 3. Homomorfismo. 4. Decidibilidade. I. Sobrinho, Daniele Nantes, orient. II. Título.

Acknowledgements

First, I would like to acknowledge my advisor Prof. Daniele Nantes Sobrinho for her support, insights and guidance throughout this work. Her mentorship have been fundamental for giving the right direction for this project. I also want to thank Prof. Christopher Lynch, for answering our e-mails when we encountered questions about the paper.

I would also like to thank all the teachers that made part of my academic journey. Their passion for knowledge and research surely had a great part in inspiring me to follow this path.

To my dear friends, thank you for giving me encouragement, joy and advice. Your friendship gave a fundamental balance and lightness to my academic life. I will never forget the joyful memories that we made together.

Last but not least, I want to express my deepest love to my parents. They always encouraged me to study and were available when I needed the most. Their love, faith and understanding have been (and always will be) the basis for my character and my success.

> "Not a single one of us here today has done it alone. We are each a patchwork quilt of those who have loved us, those who have believed in our futures, those who showed us empathy and kindness or told us the truth even when it wasn't easy to hear. Those who told us we could do it when there was absolutely no proof of that."
>
> – Taylor Swift

Resumo

Título: Um estudo detalhado em ACh-unificação com limitantes

Esta dissertação trata do problema de unificação considerando a teoria equacional ACh, que consiste da teoria com um símbolo de função h que é homomorfismo sobre um operador associativo-comutativo. O problema de unificação módulo ACh busca em resolver equações do tipo $s \stackrel{?}{=}_{ACh} t$, para termos de primeira ordem s e t, encontrando uma substituição θ que faz com que ambos os termos quando instanciados por esta substituição sejam iguais módulo ACh, i.e., tal que $s\theta =_{ACh} t\theta$. Em geral, o problema de unificação módulo ACh é indecidível. Recentemente, Eeralla e Lynch definiram uma variação do problema chamada ACh-unificação com limitante que dá como entrada um limite na quantidade de símbolos de função de homomorfismo que são aplicados repetidamente, permitindo apenas soluções que não ultrapassem esse limite. Nosso objetivo é fornecer *um estudo detalhado em ACh-unificação com limitantes*, examinando cuidadosamente o algoritmo proposto para resolver o problema e verificando a prova de terminação, correção e completude.

Abstract

This master's thesis deals with the unification problem regarding the equational theory ACh, which consists of the theory with a function symbol h that is an homomorphism over an associative-commutative operator. The Unification problem modulo ACh seeks to solve equations of the type $s \stackrel{?}{=}_{ACh} t$, for first-order terms s and t, finding a substitution θ that makes both terms, when instantiated by this substitution, equal modulo ACh, i.e., such that $s\theta =_{ACh} t\theta$. In general, the problem of ACh Unification is undecidable. Recently, Eeralla and Lynch defined a variation of the problem called *Bounded ACh Unification*, which gives as an input a bound on the number of homomorphism function symbols that are applied repeatedly, allowing only solutions that do not surpass such bound. Our goal is to provide *a detailed study of Bounded ACh Unification* by carefully examining the algorithm designed to solve the problem and validating the proof of termination, soundness and completeness.

Table of contents

Introduction

In 1965, Robinson [Rob65] introduced the Unification concept as a fundamental operation within his resolution principle. Years later, in 1970, Knuth and Bendix [KB83] reinvented this concept and utilized it as a tool to test confluence in term rewriting systems through the computation of critical pairs [BN98]. Since then, the study of E-unification problems has become an expansive field of research.

Unification is a procedure designed to find solutions for a given set of equations involving terms. For example, consider the terms $t = f(a,X)$ and $s = f(Y,b)$, where f is a binary function symbol, and a,b are constants, while X and Y are variables. The goal is to substitute the variables X and Y in t and s with terms in a way that makes the two resulting terms identical. In this example, it is evident that a sufficient substitution involves replacing X with b and Y with a, resulting in both terms becoming syntactically equal to $f(a,b)$.

Equational unification (or simply E-unification), on the other hand, is concerned with making terms equivalent while taking a congruence induced by an equational theory E into consideration. For example, let $E = \{f(X,Y) \approx f(Y,X)\}$, that is f has the property of commutativity, then the problem $\{f(X,Y) \stackrel{?}{=} f(a,b)\}$ have two possible solutions: One of them being $X \mapsto b$, $Y \mapsto a$ and the other being $X \mapsto a$, $Y \mapsto b$.

Diophantine equations and Hilbert's tenth problem. A Diophantine equation is a polynomial equation with integer coefficients that only allows integer solutions. For example, the Pythagorean equation

$$x^2 + y^2 = z^2,\ x,y,z \in \mathbb{Z}$$

is a classic example of Diophantine equation.

A set S is said to be a Diophantine set if:

- $S \subset \mathbb{N}^n := \{(a_1,\ldots,a_n) \mid a_i \in \mathbb{N}\}$
- there exists a polynomial p with integer coefficients in $n+k$ variables such that $x \in S$ iff there exists $y \in \mathbb{N}^k$, such that $p(x,y)$

In 1900, Hilbert [Hil00] proposed 23 mathematical problems, the tenth problem of this list concerns the existence of an algorithm that can determine whether a Diophantine has integer solutions or not. In 1970, Matiyasevich [Mat70], managed to prove that every computably enumerable set is Diophantine. This and the fact that there exists a computably enumerable set that is not decidable is a proof that Hilbert's tenth problem is undecidable.

AC unification. One of the most important equational theories in mathematical formalism is the Associative-Commutative (AC) theory, since AC operations such as addition $(+)$ and multiplication $(\times)$ are very present in fundamental mathematics. In 1975, Stickel [Sti75] was the first to develop an algorithm to solve AC-unification problems. The technique consists in converting the AC-unification problem into a linear Diophantine equation, then taking a basis of solutions of said equation to obtain the complete set of AC-unifiers of the given problem.

His proof of termination, however, could not be applied to the general case. It was later on that Fages [Fag87] was able to fix the proof for this case. Recently, Ayala-Rincón et al. [AFSS22], using the PVS proof-assistant, provided a formalisation of termination, soundness and completeness of the Stickel's AC-unification algorithm.

ACh-unification. The addition of an homomorphism h acting over a binary function symbol that is associative and commutative (ACh) makes the unification problem intractable. In 1996, Narendran [Nar96] managed to prove that ACh-unification is an undecidable problem via a reduction from a modified version of Hilbert's tenth problem. To illustrate the idea, Narendran associated the polynomial

$$(x-1)Y = Z-1 \tag{1}$$

with the unification problem modulo ACh:

$$h(y)+a \stackrel{?}{=}_{ACh} y+z \tag{2}$$

Notice that

- $\{z \mapsto a\}$ is not a solution to (2). In fact, we would have $h(y)+a = y+a$ which would require $h(y) \stackrel{?}{=}_{ACh} y$, and this equality has no solution in ACh. Notice that it has a solution in AC1h, where there exists an extra identity $h(0)=0$.

- $\{z \mapsto h(a), y \mapsto a\}$ is a solution to (2). In fact, applying the solution, we have

$$h(a)+a = a+h(a).$$

Since we are in ACh theory, the equality holds.

- More generally, the solutions for (2) have the form

$$\{z \mapsto h^i(a), y \mapsto h^{i-1}(a) + h^{i-2}(a) + \ldots + a\}.$$

Narendran proved that the solutions to the polynomial (1) have the form

$$\langle Z = x^k, Y = x^{k-1} + x^{k-2} + \ldots + 1 \mid k \geq 1 \rangle.$$

Then, he used the similarities between the solutions to establish a reduction from one problem to the other, proving the undecidability result.

But, since such theory has many applications in cryptographic analysis, Eeralla and Lynch [EL20] provided an approximation of ACh-unification and investigated its decidability. We define the h-height of a term as the number of h symbols applied repeatedly on the term. Then, choosing a natural number to be our bound, we search only for the ACh-unifiers with a bounded h-height. To successfully accomplish that, it is required to define the h-depth of a variable as the number of h symbols on the top of a variable. With these concepts, they managed to create a set of inference rules for an ACh-unification algorithm and prove its correctness.

Goal. The primary objective of this dissertation is to investigate the ACh-unification algorithm presented in Eeralla and Lynch's article [EL20]. This involves a thorough exploration of its functionality through illustrative examples, coupled with a rigorous validation of its proof of correctness. In light of verifying such proof, our aim is to look meticulously into crucial moments and provide more details for easier comprehension. In short, this work proposes an accessible material for studying Bounded ACh-unification, providing a more complete resource for its study.

Contributions. The contributions on this work consist in the detailed presentation of the concepts and results regarding Bounded ACh-unification that has been previously established by [EL20]. Furthermore, we presented some original contributions listed below:

1. We defined the concept of AC-solved variable (Definition 3.2), to fix an inaccuracy on the proof of Lemma 3.2 that is used in the proof of termination.

2. We improved the termination measure that was initially given in [EL20], obtaining Definition 3.3 and Proposition 3.1, which are used to prove the termination of Algorithm 1 (Corollary 3.1);

3. We fixed the proof of Lemma 3.3 on the case of application of the rule Variable Elimination (VE) and provided more details to the other cases;

4. We provided the complete inductive proof of Theorem 3.2 that is required to prove the soundness of the Algorithm 1 (Corollary 3.2);

5. We provided the complete construction in Theorem 3.3 required to prove the completeness of the Algorithm 1 (Corollary 3.3);

The proofs in items 4 and 5 were not available in [EL20].

Organisation. This dissertation is organized as follows:

Chapter 1. Preliminary Notions. We presented the main definitions and properties about ACh-unification that are essential for the comprehension of this work. In Section 1.1, we presented the syntax of first-order terms, as well as the definitions regarding substitution. In Section 1.2, we presented the definitions of multisets, order of multisets and lexicographic orders, concepts that will be fundamental to prove the correctness of our algorithm. In Section 1.3, we presented the definitions of equational theory and unification modulo equational theories. Finally, in Section 1.4, we defined the core concepts of ACh-theory, as well as the h-depth and h-height of a variable, which will be necessary to solve a bounded ACh-unification problem.

Chapter 2. $\mathfrak{I}_{ACh}$: A rule-based algorithm for bounded ACh-unification. We presented a pseudo-algorithm to solve a bounded ACh-unification problem. In Section 2.1, we presented all the inference rules that will be used in the algorithm with some illustrative examples. In Section 2.2, we presented the pseudo-algorithm ($\mathsf{Unify}_{\mathsf{AC_h}}$) which uses these rules, in addition to an example.

Chapter 3. Correctness of the Algorithm. We presented some definitions and results regarding the correctness of the algorithm $\mathsf{Unify}_{\mathsf{AC_h}}$. In Section 3.1, we provided the notation that will be used alongside the chapter. In Section 3.2, we presented the proof that the algorithm always terminates. In Section 3.3, we presented the proof that the algorithm is truth-preserving. Finally, in Section 3.4, we studied the proof of completeness.

Conclusion. We conclude this work by summarizing the main results obtained in its scope and we also propose some directions for future research.

Remark. We draw the reader's attention to our decision not to present the proofs of already established results, especially those related to AC-unification. Our goal is to prioritize the proof of results related to bounded ACh-unification or those that have been reworked in a different manner.

Chapter 1

Preliminary Notions

The purpose of this chapter is to give the definitions required to understand our problem. It starts by defining the basic syntax, such as terms, substitutions and identities, etc. Then, it continues by defining equational theories and unification modulo an equational theory with some examples. Finally, it presents the important definitions to understand the theory of our interest – the ACh theory.

1.1 Terms and substitutions

In this section we will provide some basic notions regarding of what consists of a unification problem, these definitions can be found in [BN98] and [BS01].

Definition 1.1 (Signature). A *signature* $\mathcal{F}$ is a set of function symbols, where each $f \in \mathcal{F}$ is associated with an non negative integer n, called the *arity of* f. For $n \geq 0$, we denote by $\mathcal{F}^{(n)}$ the set of all n-ary functions. The elements of $\mathcal{F}^{(0)}$ are also called constant symbols.

Definition 1.2 (Terms). Let $\mathcal{F}$ be a signature and $\mathcal{V} = \{X, Y, Z, \ldots\}$ be a set of variables such as $\mathcal{F} \cap \mathcal{V} = \emptyset$. The set of $\mathcal{F}$*-terms over* $\mathcal{V}$, denoted by $\mathcal{T}(\mathcal{F}, \mathcal{V})$ is inductively defined as

- $\mathcal{V} \subseteq \mathcal{T}(\mathcal{F}, \mathcal{V})$, i.e. a variable is always a term;
- For all $n \geq 0$, $f \in \mathcal{F}^{(n)}$ and $t_1, \cdots, t_n \in \mathcal{T}(\mathcal{F}, \mathcal{V})$, we have $f(t_1, \cdots, t_n) \in \mathcal{T}(\mathcal{F}, \mathcal{V})$

Example 1.1. Let $\mathcal{F} = \{i, f, e\}$, where f is a binary symbol, i is unary and e is 0-ary. Then the following are examples of terms with the respective signature:

1. $f(i(X), e)$
2. $f(f(e, Y), Z)$
3. $i(f(X, i(e)))$

Remark.

(i) Notice that, since e is a constant, we can write it simply as e, instead of $e()$.

(ii) Some binary function symbols, such as $+$ or $\cdot$, can be written in infix form, that is, instead of writing $+(X,Y)$ or $\cdot(X,Y)$, we simply write it as $X+Y$ or $X \cdot Y$.

Definition 1.3 (Position). Let $\mathcal{F}$ be a signature, $\mathcal{V}$ be a set of variables disjoint from $\mathcal{F}$ and $s,t \in \mathcal{T}(\mathcal{F},\mathcal{V})$.

1. The set of *positions* of the term s is a set $\mathcal{P}os(s)$ of strings over the alphabet of positive integers. which is inductively defined as
   - If $s = X \in \mathcal{V}$, then $\mathcal{P}os(s) := \{\varepsilon\}$, where ε denotes the empty string;
   - If $s = f(s_1,...,s_n)$, then

   $$\mathcal{P}os(s) := \{\varepsilon\} \cup \bigcup_{i=1}^{n} \{ip \mid p \in \mathcal{P}os(s_i)\}.$$

   Notice that, if we have $s = a$, where $a \in \mathcal{F}^{(0)}$, then $\mathcal{P}os(s) = \{\varepsilon\}$.

2. Let $p,q \in \mathcal{P}os(t)$. Then, the *prefix order* is defined as

   $$p \leq q \text{ iff there exists } p' \text{ such that } pp' = q$$

   and is a partial order on position. We say that the positions p,q are *parallel* ($p \parallel q$) iff p and q are incomparable with respect to $\leq$. The position p is *above* q iff $p \leq q$.

3. For any $p \in \mathcal{P}os(t)$, $t|_p$ is the *subterm of t in the position p* and is defined by induction on the length of p:

   $$\begin{aligned} t|_\varepsilon &:= t \\ f(t_1,\dots,t_n)|_{iq} &:= t_i|_q \end{aligned}$$

4. For $p \in \mathcal{P}os(t)$, $t[s]_p$ is the term t in which $t|_p$ is replaced by s, i.e.

   $$\begin{aligned} t[s]_\varepsilon &:= s \\ f(t_1,\dots,t_n)[s]_{iq} &:= f(t_1,\dots,t_i[s]_q,\dots,t_n). \end{aligned}$$

Example 1.2. Let f be a binary function symbol and $t = f(f(X,Y),Z)$. Then, we have $t = f(t_1,Z)$, where $t_1 = f(X,Y)$ and

$$\begin{aligned}\mathcal{P}os(t_1) &= \{\varepsilon\} \cup \{1p \mid p \in \mathcal{P}os(X)\} \cup \{2p \mid p \in \mathcal{P}os(Y)\}\\ &= \{\varepsilon, 1, 2\}.\end{aligned}$$

Therefore,

$$\begin{aligned}\mathcal{P}os(t) &= \{\varepsilon\} \cup \{1p \mid p \in \mathcal{P}os(t_1)\} \cup \{2p \mid p \in \mathcal{P}os(Z)\}\\ &= \{\varepsilon\} \cup \{1, 11, 12\} \cup \{2\}\\ &= \{\varepsilon, 1, 2, 11, 12\}.\end{aligned}$$

Notice that, for example, $1 \leq 12$. We also have that, $t|_{12} = f(t_1,Z)|_{12} = t_1|_2 = Y$.

Definition 1.4 (Substitution). Let $\mathcal{F}$ be a signature and $\mathcal{V}$ be a countably infinite set of variables. A $\mathcal{T}(\mathcal{F},\mathcal{V})$-*substitution* (or simply substitution) is a function $\sigma\colon \mathcal{V} \to \mathcal{T}(\mathcal{F},\mathcal{V})$ such that $\sigma(X) \neq X$ for a finite number of X's.

We can extend the substitution σ to a mapping $\hat{\sigma}\colon \mathcal{T}(\mathcal{F},\mathcal{V}) \to \mathcal{T}(\mathcal{F},\mathcal{V})$ defined as:

- $\hat{\sigma}(X) := \sigma(X)$, if $X \in \mathcal{V}$ and
- if $t = f(t_1,\ldots,t_n)$, then $\hat{\sigma}(t) := f(\hat{\sigma}(t_1),\ldots,\hat{\sigma}(t_n))$

Definition 1.5 (Domain, Range and Variable Range). The set of variables which σ does not map to themselves is called the *domain* of σ and is denoted by $\mathcal{D}om(\sigma) := \{x \in \mathcal{V} \mid \sigma(X) \neq X\}$. If $\mathcal{D}om(\sigma) = \{X_1,...,X_n\}$, then we write σ as

$$\sigma = \{X_1 \mapsto \sigma(X_1),...,X_n \mapsto \sigma(X_n)\}$$

The *range of* σ is $\mathcal{R}an(\sigma) := \{\sigma(X) \mid X \in \mathcal{D}om(\sigma)\}$, and the *variable range of* σ consists of the variables occurring in $\mathcal{R}an(\sigma)$, i.e.

$$\mathcal{VR}an(\sigma) = \bigcup_{X\in\mathcal{D}om(\sigma)} \mathcal{V}ar(\sigma(X))$$

When we apply a substitution σ to a term, we simultaneously replace all the occurrences of variables by their respective image.

Remark. To simplify notation, sometimes we will simply write $t\sigma$ instead of $\sigma(t)$ to indicate the application of σ to the term t.

If s,t are terms and there exists a substitution σ such that $s\sigma = t$, then t is called an *instance* of s.

Example 1.3. Let $\mathcal{F}$ be the same signature as defined in Example 1.1 and let $\sigma := \{X_1 \mapsto f(i(Y),e), X_2 \mapsto Y\}$. Then, we have

- $\mathcal{D}om(\sigma) = \{X_1, X_2\}$
- $\mathcal{R}an(\sigma) = \{f(i(Y),e), Y\}$
- $\mathcal{VR}an(\sigma) = \{Y\}$

Notice that, for instance, $f(X_1,X_2)\sigma = f(f(i(Y),e),Y)$.

Definition 1.6 (Composition of substitutions). Let θ and σ be substitutions. Then, the *composition* $\theta\sigma$ of substitutions is defined as $X\theta\sigma := \hat{\sigma}(\theta(X))$

Definition 1.7 (More general substitution). Let σ and θ be substitutions. We say that σ is *more general than* θ if there exists a substitution η such that $\theta = \sigma\eta$. We denote it by $\sigma \lesssim \theta$.

Example 1.4. Let f be an unary function symbol and a be a constant. Define $\sigma = \{X \mapsto f(Y)\}$ and $\theta = \{X \mapsto f(a), Y \mapsto a\}$. Then $\sigma \lesssim \theta$. In fact, $\theta = \sigma\sigma'$, where $\sigma' = \{Y \mapsto a\}$ because $X\theta = f(a) = X\sigma\sigma'$, $Y\theta = a = Y\sigma\sigma'$ and $Z\theta = Z = Z\sigma\sigma'$.

1.2 Multisets and Lexicographic orders

To prove termination of any reduction system $(A, \rightarrow)$, it suffices to find another reduction system $(B, >)$, which we know it terminates, and a mapping $\phi \colon A \longrightarrow B$ such that, for every $x, y \in A$, if $x \rightarrow y$, then $\phi(x) > \phi(y)$. Such mapping is called a *measure function*. In this work we will use a measure function that requires some basic knowledge of multiset and lexicographic orders. The purpose of this section is to define these concepts.

Definition 1.8 (Multiset). A *multiset* M over a set A is a function $M \colon A \longrightarrow \mathbb{N}$. Intuitively $M(x)$ is the number of copies of $x \in A$ in M.

Example 1.5. Let $A = \{a,b,c\}$. Then, a multiset over A would be $M = \{a \mapsto 1, b \mapsto 2, c \mapsto 3\}$. We can also use a standard set notation to represent M, such as $M = \{a,b,b,c,c,c\}$

Definition 1.9 (Multiset order). Let $>$ be a strict order on a set A. Then, the corresponding *multiset order* $>_{mul}$ is defined as

$$\begin{aligned}
M >_{mul} N \text{ iff there exists } C, D \text{ such that}\\
&\emptyset \neq C \subset M \text{ and}\\
&N = (M - C) \cup D \text{ and}\\
&\text{for all } y \in D, \text{ there exists } x \in C \text{ such that } x > y.
\end{aligned}$$

Intuitively, what this definition is stating is that $M >_{mul} N$ iff we can get from M to N by removing elements of M and adding elements that are "smaller" than the greatest element we removed from M.

Example 1.6. Let $M = \{6,4,2,2\}$ and $N = \{5,4,4,2\}$. Consider $>$ as the usual order in $\mathbb{N}$. Then, if we have $X = \{6,2\}$ and $Y = \{5,4\}$, we obtain that $\underbrace{\{5,4,4,2\}}_{N} = (\underbrace{\{6,4,2,2\}}_{M} - \underbrace{\{6,2\}}_{X}) \cup \underbrace{\{5,4\}}_{Y}$. Therefore, $M >_{mul} N$.

Definition 1.10 (Lexicographic order). Let $(A,>_A)$ and $(B,>_B)$ be two strict orders. The *lexicographic order* $>_{A\times B}$ *(or* $>_{lex}$*)* is defined as

$$(x,y) >_{A\times B} (z,w) \text{ iff } (x >_A z) \vee (x = z \wedge y >_B w)$$

Notice that, by iteration, we can form lexicographic products of any number of orders $(A_i,>_i)$. In this case, we have

$$(x_1,\dots,x_n) >_{lex} (y_1,\dots,y_n) \text{ iff } \exists k \leq n.(\forall i < k.(x_i = y_i) \wedge x_k > y_k)$$

1.3 Identities, Equational theories and E-unification

In this section, we will present the notion of equational theories and define a unification problem modulo an equational theory.

Definition 1.11 (Identity). Let $\mathcal{F}$ be a signature and $\mathcal{V}$ a countably infinite set of variables such that $\mathcal{V} \cap \mathcal{F} = \emptyset$. An $\mathcal{F}$*-identity* (or simply identity) is a pair $(s,t) \in \mathcal{T}(\mathcal{F},\mathcal{V}) \times \mathcal{T}(\mathcal{F},\mathcal{V})$ and is denoted by $s \approx t$.

Identities can be used to transform terms into another ones by replacing instances of the left side with the corresponding instances of the right side and vice-versa.

Example 1.7. Take $\mathcal{F} = \{f,i,e\}$ the signature defined in Example 1.1 and define the identity $f(f(X,Y),Z) \approx f(X,f(Y,Z))$. Then we can transform $f(f(i(e),e),e)$ into $f(i(e),(e,e))$.

Definition 1.12 (Equational Theory). Let E be a set of identities. An *equational theory* $=_E$ is the least equivalence relation that is closed under substitutions and contains E.

Example 1.8. Again, using $\mathcal{F} = \{f,i,e\}$. We can define G as

$$G := \{f(f(X,Y),Z) \approx f(X,f(Y,Z)), f(e,X) \approx X, f(i(X),X) \approx e\}.$$

Thus, the equational theory of groups $=_G$ is defined as the least equivalence relation that is closed under substitutions and contains the identities in G.

Definition 1.13 (E-unification). 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.

Definition 1.14 (E-unifier). Let Γ be an E-unification problem. An E-*unifier* or E-*solution* of Γ is a substitution σ such that $s_i\sigma =_E t_i\sigma$ for all $i \in \{1, \ldots, n\}$. The set of all E-unifiers is denoted by $\mathcal{U}_E(\Gamma)$.

Definition 1.15 (Satisfability modulo E).

1. Let θ be a substitution and Γ be an E-unification problem. We say that θ *satisfies* Γ *in the equational theory* E if θ is an E-unifier of Γ and we denote it by $\theta \vDash_E \Gamma$.

2. Let $\sigma = \{X_1 \mapsto t_1, \ldots, X_n \mapsto t_n\}$ and θ be substitutions. We say that θ *satisfies* σ *in the equational theory* E, and denote as $\theta \vDash_E \sigma$, if $X_i\theta =_E t_i\theta$ for all $i \in \{1, \ldots, n\}$.

Definition 1.16. Let E be an equational theory and $\mathcal{X}$ be a set of variables. The substitution σ is *more general modulo* E *on* $\mathcal{X}$ than θ if there exists a substitution σ' such that $X\theta =_E X\sigma\sigma'$ for all $X \in \mathcal{X}$. We denote it by $\sigma \lesssim_E^{\mathcal{X}} \theta$.

Definition 1.17 (Complete Set of E-unifiers). Let Γ be an E-unification problem over $\mathcal{F}$ and $\mathcal{V}ar(\Gamma)$ be the set of variables occurring in Γ. A *complete set of* E-*unifiers of* Γ is a set S of substitutions such that each element of S is an E-unifier of Γ, i.e. $S \subseteq \mathcal{U}_E(\Gamma)$, and for each $\theta \in \mathcal{U}_E(\Gamma)$, there exists a $\sigma \in S$ such that $\sigma \lesssim_E^{\mathcal{V}ar(\Gamma)} \theta$.

Notice that, when $E = \emptyset$, our $\Gamma = \{s_1 \stackrel{?}{=} t_1, \ldots, s_n \stackrel{?}{=} t_n\}$ becomes a syntactic unification problem. In this case, solving this problem would be to simply find a substitution σ such that $s_i\sigma = t_i\sigma$, that is, $s_i\sigma$ and $t_i\sigma$ are syntactically identical.

Example 1.9. Let f be a unary function symbol and t be a term. Then,

- $f(X) \stackrel{?}{=} f(t)$ has exactly one unifier, which is $\{X \mapsto t\}$

- $X \stackrel{?}{=} f(Y)$ has infinitely many unifiers, such as $\{X \mapsto Y\}, \{X \mapsto f(t_i), Y \mapsto t_i\}$, where t_i is any term in our substitution. Notice that $\{X \mapsto Y\}$ is the most general unifier for this equation, since $\{X \mapsto f(t_i), Y \mapsto t_i\} = \{X \mapsto f(Y), Y \mapsto Y\}\{Y \mapsto t_i\}$.

1.4 Notions for Unification modulo ACh-Theory

In the previous section, we discussed about unification modulo a given theory E. The purpose of this section is to present the theory we are interested in this study, which is the ACh-theory, and give some definitions that will be important over the course of this work.

What is ACh Theory? Let $\mathcal{F} = \{+, h\}$ be our signature, where $+$ and h are a binary and unary function symbols, respectively. Then we have the following identities for ACh:

- $X + (Y + Z) \approx (X + Y) + Z$ (associativity)
- $X + Y \approx Y + X$ (commutativity)
- $h(X + Y) \approx h(X) + h(Y)$ (homomorphism)

For instance, $h((X + Y) + Z) =_{ACh} h(X + Y) + h(Z)$.

> We can also add other uninterpreted function symbols with fixed arity to our signature, but as it suggests, they do not have any influence in the theory itself.

A brief example for AC-unification. Since we will use the Stickel's algorithm [Sti75] to unify the AC part of our problem, we might as well illustrate how such algorithm works.

Let $+$ be our AC function symbol. Let us unify the set $\{X + Y \stackrel{?}{=} W + Z\}$. First, we associate problem with a Diophantine equation where each argument on the function is abstracted by one variable in the equation and the coefficients are the number of occurrences of the argument. Hence, we obtain:

$$X_1 + X_2 = Y_1 + Y_2$$

Here, X_1 is associated with X, X_2 is associated with Y, Y_1 is associated with W and Y_2 is associated with Z. Now, we search for a base of solutions to the equation and associate a new variable V_i to each solution, as you can see on table below

X_1	X_2	Y_1	Y_2	New Variables
1	0	1	0	V_1
1	0	0	1	V_2
0	1	0	1	V_3
0	1	1	0	V_4

Now, we relate the "old" variables with the "new" variables, obtaining

- $X_1 = V_1 + V_2$
- $X_2 = V_3 + V_4$
- $Y_1 = V_1 + V_4$
- $Y_2 = V_2 + V_3$

The next step is to decide if we will include or not the new variables in our unification problem, with the restriction that all the new variables must be different than zero (for example, if we exclude V_1, then we must include V_2, because otherwise X_1 would be 0). For instance, let us include V_1, V_3, V_4 and exclude V_2. In this case, we obtain

$$\{X_1 \stackrel{?}{=} V_1, X_2 \stackrel{?}{=} V_3 + V_4, Y_1 \stackrel{?}{=} V_1 + V_4, Y_2 \stackrel{?}{=} V_3\}$$

Finally, we replace the "old" variables by the original terms they were associated in the beginning, obtaining the following AC-unifiers:

$$\theta_1 = \{X \mapsto V_1, Y \mapsto V_3 + V_4, Z \mapsto V_1 + V_4, W \mapsto V_3\}$$

We execute the same procedure to solve the other cases, obtaining:

$$\theta_2 = \{X \mapsto V_2, Y \mapsto V_3 + V_4, V \mapsto V_4, W \mapsto V_2 + V_3\}$$

$$\theta_3 = \{X \mapsto V_1 + V_2, Y \mapsto V_3, V \mapsto V_1, Y_2 \mapsto V_2 + V_3\}$$

$$\theta_4 = \{X \mapsto V_1 + V_2, Y \mapsto V_4, V \mapsto V_1 + V_4, W \mapsto V_2\}$$

$$\theta_5 = \{X \mapsto V, Y \mapsto W\}$$

$$\theta_6 = \{X \mapsto W, Y \mapsto V\}$$

$$\theta_7 = \{X \mapsto V_1 + V_2, Y \mapsto V_3 + V_4, Y \mapsto V_1 + V_4, W \mapsto V_2 + V_3\}$$

.

We will repeat this example throughout this work.

***h*-depth set.** Now we shall define the *h-depth* of a variable occurring in our problem. Intuitively, it is the number of h symbols appearing on the top of said variable. Such concept would be of great importance to this work, since our main goal is to show that our problem has always a solution if we introduce a bound on h.

Definition 1.18. Let Γ be an ACh-unification problem, we say that Γ is in *flattened form* if every equation in Γ is in one of the following forms:

- $X \stackrel{?}{=} Y$
- $X \stackrel{?}{=} h(Y)$
- $X \stackrel{?}{=} X_1 + \ldots + X_m$
- $X \stackrel{?}{=} f(Y_1, \ldots, Y_n)$

where X and Y are variables, X_is and Y_js are pairwise distinct variables and f is a free function symbol with arity $n \geq 0$. The first kind is called *VarVar equations*, the second is called *h-equations*, the third $+$*-equations* and the fourth *free equations*.

> For convenience, through this section, we will assume that our problem is always in flattened form unless we state the opposite.

Definition 1.19 (Graph of Γ). Let Γ be a unification problem. We define the graph $\mathbb{G}(\Gamma)$ of Γ as a digraph where each node represents a variable Γ and each edge represents the function symbol that relates two variables in Γ. To be more precise, if we have f as a function symbol with arity $n \geq 0$ and $X \stackrel{?}{=} f(X_1, \cdots, X_n) \in \Gamma$, then the graph contains the edges $X \xrightarrow{f} X_1, \ldots, X \xrightarrow{f} X_n$. If c is a constant symbol, and $X \stackrel{?}{=} c \in \Gamma$, then the graph contains a vertex X. If $X \stackrel{?}{=} Y \in \Gamma$, where X, Y are variables, the the graph contains two disconnected vertices X and Y.

Example 1.10. Let

$$\Gamma = \{V_1 \stackrel{?}{=} h(V_2), V_1 \stackrel{?}{=} h(V_3), V_2 \stackrel{?}{=} V_4 + X_2, V_3 \stackrel{?}{=} Y_1 + V_5, V_4 \stackrel{?}{=} h(X_1), V_5 \stackrel{?}{=} h(Y_2)\}$$

Then, the graph of Γ is showed in Figure 1.1.

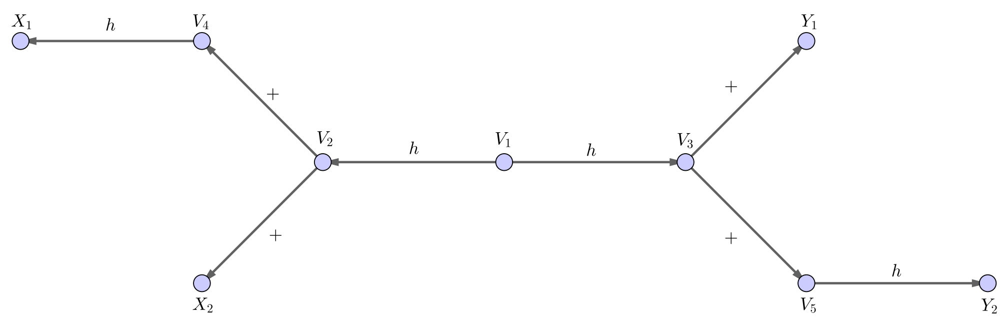

Fig. 1.1 Example of a graph of an ACh-unification Problem

Definition 1.20 (h-depth). Let Γ be a unification problem and let $X \in \mathcal{V}ar(\Gamma)$. Let h be a unary symbol and $f \neq h$ be a n-ary symbol, with $n \geq 1$ and occurring in Γ. We define the *h-depth of X*, denoted by $h_d(X,\Gamma)$, as the maximum number of h-symbols along in a path to X in $\mathbb{G}(\Gamma)$. That is,

$$h_d(X,\Gamma) := \max\{h_{dh}(X,\Gamma), h_{df}(X,\Gamma), 0\},$$

where

$$h_{dh}(X,\Gamma) := \max\{1 + h_d(Y,\Gamma) \mid Y \xrightarrow{h} X \text{ is an edge in } \mathbb{G}(\Gamma)\},$$

and

$$h_{df}(X,\Gamma) := \max\{h_d(Y,\Gamma) \mid \text{ there exists } f \neq h \text{ such that } Y \xrightarrow{f} X \text{ is in } \mathbb{G}(\Gamma)\}$$

Intuitively, what the definitions above are stating is that we can always represent the problem visually through a graph and count the number of h symbols in a path to the said variable to verify how "deep" it is into h symbols.

Example 1.11. Let us recall Example 1.10. As we can see by Figure 1.1,

- $h_d(V_1,\Gamma) = 0$
- $h_d(V_2,\Gamma) = h_d(V_3,\Gamma) = h_d(V_4,\Gamma) = h_d(V_5,\Gamma) = h_d(X_2,\Gamma) = h_d(Y_1,\Gamma) = 1$
- $h_d(X_1,\Gamma) = h_d(Y_2,\Gamma) = 2$

Definition 1.21 (h-height). Let Γ be a unification problem and t be a term in Γ. We define the *h-height of t* as it follows:

$$h_h(t) = \begin{cases} h_h(t') + 1 & \text{if } t = h(t') \\ \max\{h_h(t_1), \dots, h_h(t_n)\} & \text{if } t = f(t_1, \dots, t_n), f \neq h \\ 0 & \text{if } t = X \text{ or } t = c \end{cases}$$

where f is a function symbol with arity $n \geq 1$.

Example 1.12. Let $t = h(h(X_1 + h(X_2))$. Then, we have

$$\begin{aligned} h_h(t) &= h_h(h(h(X_1 + h(X_2)))) \\ &= h_h(h(X_1 + h(X_2))) + 1 \\ &= h_h(X_1 + h(X_2)) + 2 \\ &= h_h(h(X_2)) + 2 \\ &= 3 \end{aligned}$$

Definition 1.22 (h-depth set). Let Γ be a set of equations. The h-depth set of Γ is defined as $\Delta := \{(X, h_d(X,\Gamma)) \mid X \in \mathcal{V}ar(\Gamma)\}$. That is, the elements of Δ are pairs on the form (X,c), where X is a variable occurring in Γ and c is its respective h-depth. With that, we can also define the *maximum value of* Δ as the maximum value of all c values, that is $MaxVal(\Delta) := \max\{c \mid (X,c) \in \Delta\}$.

Example 1.13. Again, recalling Example 1.10, the h-depth set of Γ would be.

$$\Delta = \{(X_1,2),(X_2,1)(Y_1,1),(Y_2,2),(V_1,0),(V_2,1),(V_3,1),(V_4,1),(V_5,1)\}$$

Definition 1.23 (ACh-unification problem). Let $\mathcal{F}$ be a signature. An *ACh-unification problem* over $\mathcal{F}$ is a finite set of equations $\Gamma = \{s_1 \stackrel{?}{=}_{ACh} t_1, \ldots, s_n \stackrel{?}{=}_{ACh} t_n\}$, with $s_i, t_i \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ and ACh is the theory defined above.

Definition 1.24 (Bounded ACh unifier). Let $\Gamma = \{s_1 \stackrel{?}{=}_{ACh} t_1, \ldots, s_n \stackrel{?}{=}_{ACh} t_n\}$ be an ACh-unification problem. A κ *bounded ACh unifier/solution of* Γ is a substitution σ such that $t_i\sigma =_{ACh} s_i\sigma$ and $h_h(s_i\sigma), h_h(t_i\sigma) \leq \kappa$ for all $i \in \{1, \ldots, n\}$.

In the next chapter we will present an algorithm to find bounded ACh-unifiers to ACh-unification problems.

Chapter 2

$\mathfrak{I}_{ACh}$: A rule-based algorithm for bounded ACh-unification

The purpose of this chapter is to present the inference system $\mathfrak{I}_{ACh}$ to solve bounded ACh-unification problems. We will start this chapter by introducing the *Flattening* rules, which are the rules that put a problem in flattened form as we defined in the previous chapter. Then, we will provide some standard rules that are often used in syntactic unification theory and, finally, we present the specific rules for bounded ACh-unification.

2.1 The rules for $\mathfrak{I}_{ACh}$

For our inference system, denoted by $\mathfrak{I}_{ACh}$, we will use a set triple $\Gamma \parallel \Delta \parallel \sigma$, where Γ is the unification problem modulo the ACh theory, Δ is an h-depth set of Γ and σ is a substitution. We say that a substitution θ satisfies the triple $\Gamma \parallel \Delta \parallel \sigma$ when we have that $\theta \vDash \Gamma$, $\theta \vDash \sigma$ (cf. Definition 1.15) and $\mathcal{M}ax\mathcal{V}al(\Delta) \leq \kappa$, where $\kappa \in \mathbb{N}$ is a bound on the h-depth set of the variables. We denote it by $\theta \vDash \Gamma \parallel \Delta \parallel \sigma$.

Definition 2.1. Let $\Gamma \parallel \Delta \parallel \sigma$ be a set triple and $\kappa \in \mathbb{N}$ be a bound on the h-depth set of variables, then $\Gamma \parallel \Delta \parallel \sigma$ is said to be in *solved form* if $\Gamma = \emptyset$ and $\mathcal{M}ax\mathcal{V}al(\Delta) \leq \kappa$.

2.1.1 Flattening rules

These rules are responsible for putting all the equations $s \stackrel{?}{=} t$ in Γ in *flattened form*.

(FBS) Flatten Both Sides

$$(\textsc{fbs})\frac{\{t_1 \stackrel{?}{=} t_2\} \cup \Gamma \parallel \Delta \parallel \sigma}{\{V \stackrel{?}{=} t_1, V \stackrel{?}{=} t_2\} \cup \Gamma \parallel \Delta \cup \{(V,0)\} \parallel \sigma} \text{ IF } t_1, t_2 \notin \mathcal{V}$$

(FL) Flatten Left $+$

$$(\textsc{fl})\frac{\{t \stackrel{?}{=} t_1 + t_2\} \cup \Gamma \parallel \Delta \parallel \sigma}{\{t \stackrel{?}{=} V + t_2, V \stackrel{?}{=} t_1\} \cup \Gamma \parallel \Delta \cup \{(V,0)\} \parallel \sigma} \text{ IF } t_1 \notin \mathcal{V}$$

(FR) Flatten Right $+$

$$(\textsc{fr})\frac{\{t \stackrel{?}{=} t_1 + t_2\} \cup \Gamma \parallel \Delta \parallel \sigma}{\{t \stackrel{?}{=} t_1 + V, V \stackrel{?}{=} t_2\} \cup \Gamma \parallel \Delta \cup \{(V,0)\} \parallel \sigma} \text{ IF } t_2 \notin \mathcal{V}$$

(FU) Flatten under h

$$(\textsc{fu})\frac{\{t \stackrel{?}{=} h(t_1)\} \cup \Gamma \parallel \Delta \parallel \sigma}{\{t \stackrel{?}{=} h(V), V \stackrel{?}{=} t_1\} \cup \Gamma \parallel \Delta \cup \{(V,1)\} \parallel \sigma} \text{ IF } t_1 \notin \mathcal{V}$$

If we have f as an uninterpreted n-ary function symbol in our problem, we also have the following rule:

(FLFUN) Flatten under f

$$(\textsc{FlFun})\frac{\{t \stackrel{?}{=} f(t_1,\ldots,t_n)\} \cup \Gamma \parallel \Delta \parallel \sigma}{\{t \stackrel{?}{=} f(V_1,\ldots,V_n), V_1 \stackrel{?}{=} t_1, \ldots, V_n \stackrel{?}{=} t_n\} \cup \Gamma \parallel \Delta \cup \{(V_1,0),\ldots,(V_n,0)\} \parallel \sigma}$$

Intuitively, (FBS) abstracts both sides of the equation with the same variable, (FL/R) abstracts the left/right argument of $+$ with a fresh variable and (FU) abstracts the argument of the homomorphism h with a new variable. Such new variables can be any variable in $\mathcal{V}$ that did not occur in the set of equations in the previous step. Let's see in our example the applicability of these rules.

Notice that (FU) has been improved, compared to [EL20], to simultaneously update the h-depth of the new variable V, since it has a h symbol on top of it. In the original paper, we had

$$(\text{FU})\frac{\{t \stackrel{?}{=} h(t_1)\} \cup \Gamma \parallel \Delta \parallel \sigma}{\{t \stackrel{?}{=} h(V), V \stackrel{?}{=} t_1\} \cup \Gamma \parallel \Delta \cup \{(V,0)\} \parallel \sigma} \quad \text{IF } t_1 \notin \mathcal{V}$$

Example 2.1. Let $\Gamma = \{h(t_1) + t_2 \stackrel{?}{=} t_3 + t_4\}$, where $t_i \notin \mathcal{V}$, for all $i \in \{1,2,3,4\}$. Then, we have

$$\begin{aligned}
&\{h(t_1) + t_2 \stackrel{?}{=} t_3 + t_4\} \parallel \Delta \parallel \sigma \\
&\Longrightarrow_{(\text{FBS})} \\
&\{X \stackrel{?}{=} h(t_1) + t_2, X \stackrel{?}{=} t_3 + t_4\} \parallel \{(X,0)\} \cup \Delta \parallel \emptyset \\
&\Longrightarrow_{(\text{FL+})} \\
&\{X \stackrel{?}{=} Y_1 + t_2, X \stackrel{?}{=} t_3 + t_4, Y_1 \stackrel{?}{=} h(t_1)\} \parallel \{(X,0), (Y_1,0)\} \cup \Delta \parallel \emptyset \\
&\Longrightarrow_{(\text{FL+})} \\
&\{X \stackrel{?}{=} Y_1 + t_2, X \stackrel{?}{=} Y_3 + t_4, Y_1 \stackrel{?}{=} h(t_1), Y_3 \stackrel{?}{=} t_3\} \parallel \{(X,0), (Y_1,0), (Y_3,0)\} \cup \Delta \parallel \emptyset \\
&\Longrightarrow^2_{(\text{FR+})} \\
&\left\{\begin{array}{l} X \stackrel{?}{=} Y_1 + Y_2, X \stackrel{?}{=} Y_3 + Y_4, Y_1 \stackrel{?}{=} h(t_1), \\ Y_3 \stackrel{?}{=} t_3, Y_2 \stackrel{?}{=} t_2, Y_4 \stackrel{?}{=} t_4 \end{array}\right\} \parallel \{(X,0), (Y_1,0), (Y_3,0), (Y_2,0), (Y_4,0)\} \cup \Delta \parallel \emptyset \\
&\Longrightarrow_{(\text{FU})} \\
&\left\{\begin{array}{l} X \stackrel{?}{=} Y_1 + Y_2, X \stackrel{?}{=} Y_3 + Y_4, Y_1 \stackrel{?}{=} h(V), \\ Y_3 \stackrel{?}{=} t_3, Y_2 \stackrel{?}{=} t_2, Y_4 \stackrel{?}{=} t_4, V \stackrel{?}{=} t_1 \end{array}\right\} \parallel \{(X,0), (Y_1,0), (Y_3,0), (Y_2,0), (Y_4,0), (V,1)\} \cup \Delta \parallel \emptyset
\end{aligned}$$

Every equation in $\{X \stackrel{?}{=} Y_1 + Y_2, X \stackrel{?}{=} Y_3 + Y_4, Y_1 \stackrel{?}{=} h(V), Y_3 \stackrel{?}{=} t_3, Y_2 \stackrel{?}{=} t_2, Y_4 \stackrel{?}{=} t_4, V \stackrel{?}{=} t_1\}$ is in flattened form.

2.1.2 Update h-depth set rules

These rules are defined to update Δ, that is, to compute the h-depths of all the variables occurring in Γ.

(Uh) Update h

$$(\mathrm{U}h)\frac{\{X \stackrel{?}{=} h(Y)\} \cup \Gamma \parallel \{(X,k),(Y,l)\} \cup \Delta \parallel \sigma}{\{X \stackrel{?}{=} h(Y)\} \cup \Gamma \parallel \{(X,k),(Y,k+1)\} \cup \Delta \parallel \sigma} \quad \text{IF } l < k+1$$

Update $+$

1. **(UL) Update Left $+$**

$$(\mathrm{UL})\frac{\{X \stackrel{?}{=} Y_1+Y_2\} \cup \Gamma \parallel \{(X,k),(Y_1,l),(Y_2,m)\} \cup \Delta \parallel \sigma}{\{X \stackrel{?}{=} Y_1+Y_2\} \cup \Gamma \parallel \{(X,k),(Y_1,k),(Y_2,m)\} \cup \Delta \parallel \sigma} \quad \text{IF } l < k$$

2. **(UR) Update Right $+$**

$$(\mathrm{UR})\frac{\{X \stackrel{?}{=} Y_1+Y_2\} \cup \Gamma \parallel \{(X,k),(Y_1,l),(Y_2,m)\} \cup \Delta \parallel \sigma}{\{X \stackrel{?}{=} Y_1+Y_2\} \cup \Gamma \parallel \{(X,k),(Y_1,l),(Y_2,k)\} \cup \Delta \parallel \sigma} \quad \text{IF } m < k$$

Intuitively, (Uh) is applied when we have an equation of the form $X \stackrel{?}{=} h(Y) \in \Gamma$. In this case, it's clear that $h_d(Y,\Gamma) = h_d(X,\Gamma)+1$, since Y has an h symbol on top of it. **Update +** is applied when we have $X \stackrel{?}{=} X_1+X_2$. Notice that in this case, since X_1 and X_2 do not have an h symbol on top of them, $h_d(X_i,\Gamma) = h_d(X,\Gamma)$.

Example 2.2. Let $\Gamma = \{X \stackrel{?}{=} Y+Z, W \stackrel{?}{=} h(h(X))\}$, where $X,Y,Z,W \in \mathcal{V}$. Then, we have

$$\{X \stackrel{?}{=} Y+Z, W \stackrel{?}{=} h(h(X))\} \parallel \{(X,0),(Y,0),(Z,0),(W,0)\} \parallel \emptyset$$
$$\Longrightarrow_{(\mathrm{FU})}$$
$$\{X \stackrel{?}{=} Y+Z, W \stackrel{?}{=} h(V_1), V_1 \stackrel{?}{=} h(X)\} \parallel \{(X,0),(Y,0),(Z,0),(W,0),(V_1,1)\} \parallel \emptyset$$
$$\Longrightarrow_{(\mathrm{U}h)}$$
$$\{X \stackrel{?}{=} Y+Z, W \stackrel{?}{=} h(V_1), V_1 \stackrel{?}{=} h(X)\} \parallel \{(X,2),(Y,0),(Z,0),(W,0),(V_1,1)\} \parallel \emptyset$$
$$\Longrightarrow_{(\mathrm{UL})}$$
$$\{X \stackrel{?}{=} Y+Z, W \stackrel{?}{=} h(V_1), V_1 \stackrel{?}{=} h(X)\} \parallel \{(X,2),(Y,2),(Z,0),(W,0),(V_1,1)\} \parallel \emptyset$$
$$\Longrightarrow_{(\mathrm{UR})}$$
$$\{X \stackrel{?}{=} Y+Z, W \stackrel{?}{=} h(V_1), V_1 \stackrel{?}{=} h(X)\} \parallel \{(X,2),(Y,2),(Z,2),(W,0),(V_1,1)\} \parallel \emptyset$$

Notice that, if we look at the graph of our problem (see Figure 2.1), all the corresponding h-depths match with the ones given by our inference rule.

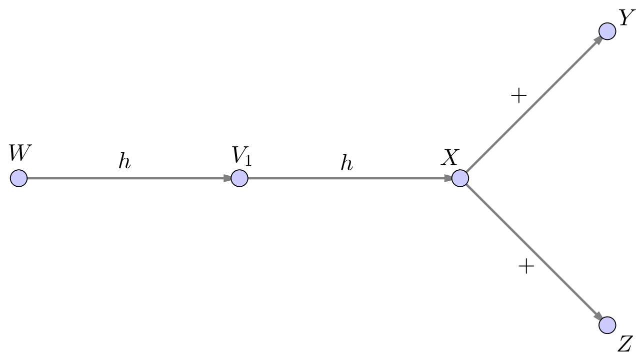

Fig. 2.1 Graph of $\{X \stackrel{?}{=} Y + Z, W \stackrel{?}{=} h(V_1), V_1 \stackrel{?}{=} h(X)\}$

2.1.3 Variable Elimination rules

These are the rules to find a unifier for the problem. They are responsible to transform the equations into assignments.

(VE1) Variable Elimination 1

$$(\text{VE1})\frac{\{X \stackrel{?}{=} Y\} \cup \Gamma \parallel \Delta \parallel \sigma}{\Gamma\{X \mapsto Y\} \parallel \Delta \parallel \sigma\{X \mapsto Y\} \cup \{X \mapsto Y\}} \quad \text{IF } X \neq Y$$

(VE2)Variable Elimination 2

$$(\text{VE2})\frac{\{X \stackrel{?}{=} t\} \cup \Gamma \parallel \Delta \parallel \sigma}{\Gamma\{X \mapsto t\} \parallel \Delta \parallel \sigma\{X \mapsto t\} \cup \{X \mapsto t\}} \quad \text{IF } X \notin \mathcal{V}ar(t)$$

Example 2.3. Let $\Gamma = \{X \stackrel{?}{=} Y, X \stackrel{?}{=} V_1 + V_2\}$. Then, we have

$$\{X \stackrel{?}{=} Y, X \stackrel{?}{=} V_1 + V_2\} \parallel \{(X,0),(Y,0),(V_1,0),(V_2,0)\} \parallel \emptyset$$
$$\Longrightarrow_{(\text{VE1})}$$
$$\{Y \stackrel{?}{=} V_1 + V_2\} \parallel \{(X,0),(Y,0),(V_1,0),(V_2,0)\} \parallel \{X \mapsto Y\}$$
$$\Longrightarrow_{(\text{VE2})}$$
$$\emptyset \parallel \{(X,0),(Y,0),(V_1,0),(V_2,0)\} \parallel \{X \mapsto V_1 + V_2, Y \mapsto V_1 + V_2\}$$

2.1.4 Basic rules

The following rules are the standard Martelli-Montanari unification rules found in [BN98]. The first rule is to remove trivial equations from our problem.

(TRIV) Trivial

$$(\text{TRIV})\frac{\{t \stackrel{?}{=} t\} \cup \Gamma \parallel \Delta \parallel \sigma}{\Gamma \parallel \Delta \parallel \sigma}$$

This rule swaps the left side of the equation with the right side and is applied when the left side is not a variable but the right side is.

(OR) Orient

$$(\text{OR})\frac{\{t \stackrel{?}{=} X\} \cup \Gamma \parallel \Delta \parallel \sigma}{\{X \stackrel{?}{=} t\} \cup \Gamma \parallel \Delta \parallel \sigma} \quad \text{IF } t \notin \mathcal{V}$$

The following rule decomposes an equation into sub-equations if the function symbols on both sides of the equation are equal, except if it's a $+$. For this case, we apply a different rule which we shall present later.

(DEC) Decomposition

$$(\text{DEC})\frac{\{X \stackrel{?}{=} f(X_1,\ldots,X_n), X \stackrel{?}{=} f(Y_1,\ldots,Y_n)\} \cup \Gamma \parallel \Delta \parallel \sigma}{\{X \stackrel{?}{=} f(X_1,\ldots,X_n), X_1 \stackrel{?}{=} Y_1,\ldots,X_n \stackrel{?}{=} Y_n\} \cup \Gamma \parallel \Delta \parallel \sigma} \quad \text{IF } f \neq +$$

Example 2.4. Let $\Gamma = \{h(h(X) \stackrel{?}{=} h(h(Y))\}$, where $X, Y \in \mathcal{V}$. Then, we have

$$\{h(h(X)) \stackrel{?}{=} h(h(Y))\} \parallel \{(X,0),(Y,0)\} \parallel \emptyset$$

$$\Longrightarrow_{(\text{FBS})}$$

$$\{V \stackrel{?}{=} h(h(X)), V \stackrel{?}{=} h(h(Y))\} \parallel \{(X,0),(Y,0),(V,0)\} \parallel \emptyset$$

$$\Longrightarrow^2_{(\text{FU})}$$

$$\{V \stackrel{?}{=} h(V_1), V \stackrel{?}{=} h(V_2), V_1 \stackrel{?}{=} h(X), V_2 \stackrel{?}{=} h(Y)\} \parallel \{(X,0),(Y,0),(V,0),(V_1,1),(V_2,1)\} \parallel \emptyset$$

$$\Longrightarrow^2_{(\text{U}h)}$$

$$\{V \stackrel{?}{=} h(V_1), V \stackrel{?}{=} h(V_2), V_1 \stackrel{?}{=} h(X), V_2 \stackrel{?}{=} h(Y)\} \parallel \{(X,2),(Y,2),(V,0),(V_1,1),(V_2,1)\} \parallel \emptyset$$

$$\Longrightarrow_{(\text{DEC})}$$

$$\{V \stackrel{?}{=} h(V_1), V_1 \stackrel{?}{=} V_2, V_1 \stackrel{?}{=} h(X), V_2 \stackrel{?}{=} h(Y)\} \parallel \underbrace{\{(X,2),(Y,2),(V,0),(V_1,1),(V_2,1)\}}_{\Delta} \parallel \emptyset$$

$$\Longrightarrow_{(\mathrm{VE1})}$$
$$\{V \stackrel{?}{=} h(V_2), V_2 \stackrel{?}{=} h(X), V_2 \stackrel{?}{=} h(Y)\} \parallel \Delta \parallel \{V_1 \mapsto V_2\}$$
$$\Longrightarrow_{(\mathrm{DEC})}$$
$$\{V \stackrel{?}{=} h(V_2), V_2 \stackrel{?}{=} h(X), X \stackrel{?}{=} Y\} \parallel \Delta \parallel \{V_1 \mapsto V_2\}$$
$$\Longrightarrow_{(\mathrm{VE1})}$$
$$\{V \stackrel{?}{=} h(V_2), V_2 \stackrel{?}{=} h(Y)\} \parallel \Delta \parallel \{V_1 \mapsto V_2, X \mapsto Y\}$$
$$\Longrightarrow_{(\mathrm{VE2})}$$
$$\{V_2 \stackrel{?}{=} h(Y)\} \parallel \Delta \parallel \{V_1 \mapsto V_2, X \mapsto Y, V \mapsto h(V_2)\}$$
$$\Longrightarrow_{(\mathrm{VE2})}$$
$$\emptyset \parallel \Delta \parallel \{V_1 \mapsto h(Y), X \mapsto Y, V \mapsto h(h(Y)), V_2 \mapsto h(Y)\}$$

Notice that $\{X \mapsto Y\}$ is the mgu of $\{h(h(X) \stackrel{?}{=} h(h(Y))\}$.

2.1.5 Checking rules

These rules are set to identify failure cases for an ACh unification problem.

(OC) Occur Check

$$(\mathrm{OC})\frac{\{X \stackrel{?}{=} f(t_1,\ldots,t_n)\} \cup \Gamma \parallel \Delta \parallel \sigma}{\bot} \quad \text{IF } X \in \mathcal{V}ar(f(t_1,\ldots,t_n)\sigma)$$

(CLASH) Clash

$$(\mathrm{CLASH})\frac{\{X \stackrel{?}{=} f(s_1,\ldots s_m), X \stackrel{?}{=} g(t_1,\ldots,t_n)\} \cup \Gamma \parallel \Delta \parallel \sigma}{\bot} \quad \text{IF } f \notin \{h,+\} \text{ OR } g \notin \{h,+\}$$

Notice that the two above are also the standard (OC) and (CLASH) rules found in [BN98]. Intuitively, if we have a variable on the left side of an equation occurring also on the right side, it never terminates. So, to avoid that, we indicate failure.

Also, as we know, it is not possible to solve an equation with different function symbols on the top of both sides of an equation, unless one of them is $+$ and the other is h – we will present a rule for this case later. Hence, we indicate failure if this happens as well.

Example 2.5.

1. **Occur Check**
   Let $\Gamma = \{X \stackrel{?}{=} Y, Y \stackrel{?}{=} h(Z+X)\}$, where $X,Y,Z \in \mathcal{V}$. Then, we have

$$\{X \stackrel{?}{=} Y, Y \stackrel{?}{=} h(Z+X)\} \parallel \{(X,0),(Y,0),(Z,0)\} \parallel \emptyset$$
$$\Longrightarrow_{(\text{FU})}$$
$$\{X \stackrel{?}{=} Y, Y \stackrel{?}{=} h(V), V \stackrel{?}{=} Z+X\} \parallel \{(X,0),(Y,0),(Z,0),(V,1)\} \parallel \emptyset$$
$$\Longrightarrow_{(\text{UL/R})}$$
$$\{X \stackrel{?}{=} Y, Y \stackrel{?}{=} h(V), V \stackrel{?}{=} Z+X\} \parallel \{(X,1),(Y,0),(Z,1),(V,1)\} \parallel \emptyset$$
$$\Longrightarrow_{(\text{VE1})}$$
$$\{Y \stackrel{?}{=} h(V), V \stackrel{?}{=} Z+Y\} \parallel \{(X,1),(Y,0),(Z,1),(V,1)\} \parallel \{X \mapsto Y\}$$
$$\Longrightarrow_{(\text{VE2})}$$
$$\{V \stackrel{?}{=} Z+h(V)\} \parallel \{(X,1),(Y,0),(Z,1),(V,1)\} \parallel \{X \mapsto h(V), Y \mapsto h(V)\}$$
$$\Longrightarrow_{(\text{OC})}$$
$$\perp$$

2. **Clash**
   Let f be a binary function symbol, g be a unary function symbol and $\Gamma = \{f(X,h(Y)) \stackrel{?}{=} g(Z)\}$. Then, we have

$$\{f(X,Y) \stackrel{?}{=} g(Z)\} \parallel \{(X,0)(Y,0)(Z,0)\} \parallel \emptyset$$
$$\Longrightarrow_{(\text{FBS})}$$
$$\{V \stackrel{?}{=} f(X,Y), V \stackrel{?}{=} g(Z)\} \parallel \{(X,0)(Y,0)(Z,0),(V,0)\} \parallel \emptyset$$
$$\Longrightarrow_{(\text{CLASH})}$$
$$\perp$$

The following rule determines if a solution exists within the given bound κ. It is one of the most important rules for $\mathfrak{I}_{ACh}$, since our main goal is to show that, putting a bound in the h-depth, our problem becomes decidable.

(BC) Bound Check

$$(\text{BC})\frac{\Gamma \parallel \Delta \parallel \sigma}{\perp} \quad \text{IF} \quad \mathcal{M}ax\mathcal{V}al(\Delta) > \kappa$$

The intuition behind this rule is straightforward, after computing the h-depths of all the variables in Γ, if there is at least one variable X such that $h_d(X,\Gamma) > \kappa$. Our problem cannot be solved. Let's see how this works in practice.

Example 2.6. Let $\kappa = 2$ be our bound and $\Gamma = \{Y \stackrel{?}{=} h(h(h(X)))\}$, where $X,Y \in \mathcal{V}$. Then, we have

$$
\begin{aligned}
&\{Y \stackrel{?}{=} h(h(h(X)))\} \parallel \{(X,0),(Y,0)\} \parallel \emptyset \\
&\Longrightarrow^{2}_{(\mathrm{FU})} \\
&\{Y \stackrel{?}{=} h(V_1), V_1 \stackrel{?}{=} h(V_2), V_2 \stackrel{?}{=} h(X)\} \parallel \{(X,0),(Y,0),(V_1,0),(V_2,0)\} \parallel \emptyset \\
&\Longrightarrow^{3}_{(\mathrm{U}h)} \\
&\{Y \stackrel{?}{=} h(V_1), V_1 \stackrel{?}{=} h(V_2), V_2 \stackrel{?}{=} h(X)\} \parallel \{(X,3),(Y,0),(V_1,1)(V_2,2)\} \parallel \emptyset \\
&\Longrightarrow_{(\mathrm{BC})} \\
&\perp
\end{aligned}
$$

2.1.6 Splitting rule

The following rule takes the homomorphism theory in consideration,

(SPLIT) Splitting

$$
(\textsc{Split})\,\frac{\{X \stackrel{?}{=} h(Y), X \stackrel{?}{=} X_1 + \ldots + X_n\} \cup \Gamma \parallel \Delta \parallel \sigma}{\{X \stackrel{?}{=} h(Y), Y \stackrel{?}{=} V_1 + \ldots + V_n, X_1 \stackrel{?}{=} h(V_1), \ldots, X_n \stackrel{?}{=} h(V_n)\} \cup \Gamma \parallel \Delta' \parallel \sigma}
$$

where $n > 1$, $X \neq Y$ and $X \neq X_i$ for all i, $\Delta' = \{(V_1,1),\ldots,(V_n,1)\} \cup \Delta$ and $V_1,\ldots,V_n$ are fresh variables.

Summarizing, we cannot solve an equation $h(Y) \stackrel{?}{=} X_1 + \ldots + X_n$ unless Y is also a sum. Hence, we create new variables $V_1,\ldots,V_n$, which did not occur anywhere in the problem, such that Y is the sum of these new variables and, recalling the definition of homomorphism, we must have that $X_i = h(V_i)$ for $i \in \{1,\ldots,n\}$.

Example 2.7. Let $\Gamma = \{h(X) = Y_1 + Y_2\}$ and $\kappa = 2$. Then, we have

$$\{h(X) \stackrel{?}{=} Y_1 + Y_2\} \parallel \{(X,0),(Y_1,0),(Y_2,0)\} \parallel \emptyset$$
$$\Longrightarrow_{(\text{FBS})}$$
$$\{V \stackrel{?}{=} h(X), V \stackrel{?}{=} Y_1 + Y_2\} \parallel \{(X,0),(Y_1,0),(Y_2,0),(V,0)\} \parallel \emptyset$$
$$\Longrightarrow_{(\text{U}h)}$$
$$\{V \stackrel{?}{=} h(X), V \stackrel{?}{=} Y_1 + Y_2\} \parallel \{(X,1),(Y_1,0),(Y_2,0),(V,0)\} \parallel \emptyset$$
$$\Longrightarrow_{(\text{SPLIT})}$$
$$\left\{V \stackrel{?}{=} h(X), X \stackrel{?}{=} V_1 + V_2, Y_1 \stackrel{?}{=} h(V_1), Y_2 \stackrel{?}{=} h(V_2)\right\} \parallel \{\underbrace{(X,1),(Y_1,0),(Y_2,0),(V,0),(V_1,1),(V_2,1)}_{\Delta}\} \parallel \emptyset$$
$$\Longrightarrow_{(\text{VE2})}$$
$$\left\{X \stackrel{?}{=} V_1 + V_2, Y_1 \stackrel{?}{=} h(V_1), Y_2 \stackrel{?}{=} h(V_2)\right\} \parallel \Delta \parallel \{V \mapsto h(X)\}$$
$$\Longrightarrow_{(\text{VE2})}$$
$$\{Y_1 \stackrel{?}{=} h(V_1), Y_2 \stackrel{?}{=} h(V_2)\} \parallel \Delta \parallel \{V \mapsto h(V_1 + V_2), X \mapsto V_1 + V_2\}$$
$$\Longrightarrow^2_{(\text{VE2})}$$
$$\emptyset \parallel \Delta \parallel \{V \mapsto h(V_1 + V_2), X \mapsto V_1 + V_2, Y_1 \mapsto h(V_1), Y_2 \mapsto h(V_2)\}$$

2.1.7 AC-Unification rule

This rule uses an established AC-unification algorithm to solve the AC part of the problem (as we discussed before, we are using the algorithm formalized by Gabriel Silva in [AFSS22]). Consider Ψ as the set of all the equations with the $+$ symbol on the right side and Γ as the set containing the other types of equations. Then,

AC-Unification

$$(\text{AC})\ \frac{\Psi \cup \Gamma \parallel \Delta \parallel \sigma}{GetEqs(\theta_1) \cup \Gamma \parallel \Delta_1 \parallel \sigma \vee \ldots \vee GetEqs(\theta_n) \cup \Gamma \parallel \Delta_n \parallel \sigma}$$

where $\text{Unify}(\Psi) = \{\theta_1, \ldots, \theta_n\}$

where Unify is a function which returns a complete set of AC-unifiers given by an AC-unification algorithm and *GetEqs* is a function that takes a substitution $\theta = \{X_1 \mapsto t_1, \ldots, X_n \mapsto t_n\}$ and returns its equational form, i.e., $GetEqs(\theta) = \{X_1 \stackrel{?}{=} t_1, \ldots, X_n \stackrel{?}{=} t_n\}$. Also, notice that, since the AC Unification algorithm introduces new variables to our problem, we must add such variables to our h-depth set.

For the next example, we will use the method to find solutions presented in [Sti75] and formalized by [AFSS22].

Example 2.8. Let $\Gamma = \{X_1 + X_2 \stackrel{?}{=} Y_1 + Y_2, X \stackrel{?}{=} h(Y)\}$. Then, we have

$$\{X_1 + X_2 \stackrel{?}{=} Y_1 + Y_2\} \cup \{X \stackrel{?}{=} h(Y)\} \parallel \{(X_1,0)(X_2,0)(Y_1,0)(Y_2,0)\} \parallel \emptyset$$
$$\Longrightarrow_{(\text{FBS})}$$
$$\{V \stackrel{?}{=} X_1 + X_2, V \stackrel{?}{=} Y_1 + Y_2\} \cup \{X \stackrel{?}{=} h(Y)\} \parallel \Delta \parallel \emptyset$$
$$\Longrightarrow_{(\text{AC})}$$
$$\{V \stackrel{?}{=} V_1 + V_2 + V_3 + V_4, X_1 \stackrel{?}{=} V_1 + V_2, X_2 \stackrel{?}{=} V_3 + V_4, Y_1 \stackrel{?}{=} V_1 + V_4, Y_2 \stackrel{?}{=} V_2 + V_3\}$$
$$\cup \{X \stackrel{?}{=} h(Y)\} \parallel \Delta \parallel \emptyset \vee$$
$$\vee \{V \stackrel{?}{=} V_1 + V_3 + V_4, X_1 \stackrel{?}{=} V_1, X_2 \stackrel{?}{=} V_3 + V_4, Y_1 \stackrel{?}{=} V_1 + V_4, Y_2 \stackrel{?}{=} V_3\} \cup \{X \stackrel{?}{=} h(Y)\} \parallel \Delta \parallel \emptyset$$
$$\vee \{V \stackrel{?}{=} V_1 + V_2 + V_4, X_1 \stackrel{?}{=} V_1 + V_2, X_2 \stackrel{?}{=} V_4, Y_1 \stackrel{?}{=} V_1 + V_4, Y_2 \stackrel{?}{=} V_2\} \cup \{X \stackrel{?}{=} h(Y)\} \parallel \Delta \parallel \emptyset$$
$$\vee \{V \stackrel{?}{=} V_1 + V_2 + V_3, X_1 \stackrel{?}{=} V_1 + V_2, X_2 \stackrel{?}{=} V_3, Y_1 \stackrel{?}{=} V_1, Y_2 \stackrel{?}{=} V_2 + V_3\} \cup \{X \stackrel{?}{=} h(Y)\} \parallel \Delta \parallel \emptyset$$
$$\vee \{V \stackrel{?}{=} V_2 + V_3 + V_4, X_1 \stackrel{?}{=} V_2, X_2 \stackrel{?}{=} V_3 + V_4, Y_1 \stackrel{?}{=} V_4, Y_2 \stackrel{?}{=} V_2 + V_3\} \cup \{X \stackrel{?}{=} h(Y)\} \parallel \Delta \parallel \emptyset$$
$$\vee \{V \stackrel{?}{=} X_1 + X_2, X_1 \stackrel{?}{=} Y_1, X_2 \stackrel{?}{=} Y_2\} \cup \{X \stackrel{?}{=} h(Y)\} \parallel \Delta \parallel \emptyset$$
$$\vee \{V \stackrel{?}{=} X_1 + X_2, X_1 \stackrel{?}{=} Y_2, X_2 \stackrel{?}{=} Y_1\} \cup \{X \stackrel{?}{=} h(Y)\} \parallel \Delta \parallel \emptyset$$

2.2 Pseudo Algorithms for Flattening and ACh-Unification

In this section, we will present the pseudo algorithm to solve a given ACh-Unification Problem. First, let us recap all the $\mathfrak{I}_{ACh}$ rules:

$$(\text{FBS})\frac{\{t_1 \stackrel{?}{=} t_2\} \cup \Gamma \parallel \Delta \parallel \sigma}{\{V \stackrel{?}{=} t_1, V \stackrel{?}{=} t_2\} \cup \Gamma \parallel \Delta \cup \{(V,0)\} \parallel \sigma} \text{ IF } t_1, t_2 \notin \mathcal{V} \qquad (\text{FL})\frac{\{t \stackrel{?}{=} t_1 + t_2\} \cup \Gamma \parallel \Delta \parallel \sigma}{\{t \stackrel{?}{=} V + t_2, V \stackrel{?}{=} t_1\} \cup \Gamma \parallel \Delta \cup \{(V,0)\} \parallel \sigma} \text{ IF } t_1 \notin \mathcal{V}$$

$$(\text{FR})\frac{\{t \stackrel{?}{=} t_1 + t_2\} \cup \Gamma \parallel \Delta \parallel \sigma}{\{t \stackrel{?}{=} t_1 + V, V \stackrel{?}{=} t_2\} \cup \Gamma \parallel \Delta \cup \{(V,0)\} \parallel \sigma} \text{ IF } t_2 \notin \mathcal{V} \qquad (\text{FU})\frac{\{t \stackrel{?}{=} h(t_1)\} \cup \Gamma \parallel \Delta \parallel \sigma}{\{t \stackrel{?}{=} h(V), V \stackrel{?}{=} t_1\} \cup \Gamma \parallel \Delta \cup \{(V,1)\} \parallel \sigma} \text{ IF } t_1 \notin \mathcal{V}$$

$$(\text{FLFUN})\frac{\{t \stackrel{?}{=} f(t_1, \ldots, t_n)\} \cup \Gamma \parallel \Delta \parallel \sigma}{\{t \stackrel{?}{=} f(V_1, \ldots, V_n), V_1 \stackrel{?}{=} t_1, \ldots, V_n \stackrel{?}{=} t_n\} \cup \Gamma \parallel \Delta \cup \{(V_1,0), \ldots, (V_n,0)\} \parallel \sigma}$$

$$(\text{U}h)\frac{\{X \stackrel{?}{=} h(Y)\} \cup \Gamma \parallel \{(X,k),(Y,l)\} \cup \Delta \parallel \sigma}{\{X \stackrel{?}{=} h(Y)\} \cup \Gamma \parallel \{(X,k),(Y,k+1)\} \cup \Delta \parallel \sigma} \text{ if } l < k+1$$

$$(\text{UL})\frac{\{X \stackrel{?}{=} Y_1 + Y_2\} \cup \Gamma \parallel \{(X,k),(Y_1,l),(Y_2,m)\} \cup \Delta \parallel \sigma}{\{X \stackrel{?}{=} Y_1 + Y_2\} \cup \Gamma \parallel \{(X,k),(Y_1,k),(Y_2,m)\} \cup \Delta \parallel \sigma} \text{ if } l < k \qquad (\text{UR})\frac{\{X \stackrel{?}{=} Y_1 + Y_2\} \cup \Gamma \parallel \{(X,k),(Y_1,l),(Y_2,m)\} \cup \Delta \parallel \sigma}{\{X \stackrel{?}{=} Y_1 + Y_2\} \cup \Gamma \parallel \{(X,k),(Y_1,l),(Y_2,k)\} \cup \Delta \parallel \sigma} \text{ if } m < k$$

$$(\text{VE1})\frac{\{X \stackrel{?}{=} Y\} \cup \Gamma \parallel \Delta \parallel \sigma}{\Gamma\{X \mapsto Y\} \parallel \Delta \parallel \sigma\{X \mapsto Y\} \cup \{X \mapsto Y\}} \text{ IF } X \neq Y \qquad (\text{VE2})\frac{\{X \stackrel{?}{=} t\} \cup \Gamma \parallel \Delta \parallel \sigma}{\Gamma\{X \mapsto t\} \parallel \Delta \parallel \sigma\{X \mapsto t\} \cup \{X \mapsto t\}} \text{ IF } X \notin \mathcal{V}ar(t)$$

$$(\text{DEC})\frac{\{X \stackrel{?}{=} f(X_1,\ldots,X_n), X \stackrel{?}{=} f(Y_1,\ldots,Y_n)\} \cup \Gamma \parallel \Delta \parallel \sigma}{\{X \stackrel{?}{=} f(X_1,\ldots,X_n), X_1 \stackrel{?}{=} Y_1, \ldots, X_n \stackrel{?}{=} Y_n\} \cup \Gamma \parallel \Delta \parallel \sigma} \text{ IF } f \neq + \qquad (\text{TRIV})\frac{\{t \stackrel{?}{=} t\} \cup \Gamma \parallel \Delta \parallel \sigma}{\Gamma \parallel \Delta \parallel \sigma}$$

$$(\text{SPLIT})\frac{\{X \stackrel{?}{=} h(Y), X \stackrel{?}{=} X_1 + \ldots + X_n\} \cup \Gamma \parallel \Delta \parallel \sigma}{\{X \stackrel{?}{=} h(Y), Y \stackrel{?}{=} V_1 + \ldots + V_n, X_1 \stackrel{?}{=} h(V_1), \ldots, X_n \stackrel{?}{=} h(V_n)\} \cup \Gamma \parallel \Delta' \parallel \sigma} \qquad (\text{OR})\frac{\{t \stackrel{?}{=} X\} \cup \Gamma \parallel \Delta \parallel \sigma}{\{X \stackrel{?}{=} t\} \cup \Gamma \parallel \Delta \parallel \sigma} \text{ IF } t \notin \mathcal{V}$$

$$(\text{AC})\frac{\Psi \cup \Gamma \parallel \Delta \parallel \sigma}{GetEqs(\theta_1) \cup \Gamma \parallel \Delta_1 \parallel \sigma \vee \ldots \vee GetEqs(\theta_n) \cup \Gamma \parallel \Delta_n \parallel \sigma}$$

$$(\text{OC})\frac{\{X \stackrel{?}{=} f(t_1,\ldots,t_n)\} \cup \Gamma \parallel \Delta \parallel \sigma}{\perp} \text{ IF } X \in \mathcal{V}ar(f(t_1,\ldots,t_n)\sigma) \qquad (\text{BC})\frac{\Gamma \parallel \Delta \parallel \sigma}{\perp} \text{ IF } \mathcal{M}ax\mathcal{V}al(\Delta) > \kappa$$

$$(\text{CLASH})\frac{\{X \stackrel{?}{=} f(s_1,\ldots s_m), X \stackrel{?}{=} g(t_1,\ldots,t_n)\} \cup \Gamma \parallel \Delta \parallel \sigma}{\perp} \text{ IF } f \notin \{h,+\} \text{ OR } g \notin \{h,+\}$$

Fig. 2.2 $\mathfrak{I}_{ACh}$ rules.

Algorithm 1: $\mathsf{Unify}_{\mathsf{AC_h}}$

Input: An equation set Γ, an empty h-depth set Δ, an empty set σ and a bound $\kappa \in \mathbb{N}$

Output: A complete set of κ-bounded ACh-unifiers $\{\sigma_1, ..., \sigma_n\}$ or $\perp$ indicating that the problem has no solution.

Begin

0. Compute Δ by adding all the variables in $\mathcal{V}ar(\Gamma)$ with initial h-depth zero;
1. Apply Algorithm 2 (Flattening) on Γ
2. **Repeat**
   (Apply (VE1) exhaustively after each of the following rule applications)
   (a) Apply (TRIV) exhaustively to eliminate equations of the form $t \stackrel{?}{=} t$;
   (b) Apply the (OC), i.e., **If** any variable on the left side occurs on the right **then** return $\perp$;
   (c) Apply the (BC), i.e., **If** $\mathcal{M}ax\mathcal{V}al(\Delta) > \kappa$ **then** return $\perp$;
   (d) **If** at least one of the h-depth update rules ((Uh), (UL) or (UR)) is applicable **then** apply the rule and go to (c) **else** go to next step;
   (e) Apply (OR) exhaustively;
   (f) **If** (SPLIT) is applicable **then** apply the rule and go to (a);
   (g) Apply (CLASH), i.e., **If** the top symbols of the left and right sides of an equation do not match **then** return $\perp$;
   (h) **If** (DEC) is applicable **then** apply the rule and go to (a);
   (i) **If** there is at least one variable X occurring left side in at least two equations of the form $X \stackrel{?}{=} Y_1 + ...Y_n$ and $X \stackrel{?}{=} Z_1 + ... + Z_n$, **then** apply the (AC) rule and go to (d) **else** go to next step;
   (j) Apply (VE2) exhaustively and return the output;

End

We shall explain how the algorithm works. First, we read all the variables $X \in \mathcal{V}ar(\Gamma)$ and add them to Δ initially as $(X, 0)$. Then, we call the Flattening algorithm (which is presented next) to put all the equations in flattened form. Then, we apply the other rules of $\mathfrak{I}_{ACh}$ following a specific strategy:

1. We can start by removing all the trivial equations, so we apply (TRIV).

2. Then, we apply (OC) and (BC) to see if it has an immediate failure.

3. Now, we update the h-depths of all the variables occurring in our problem. Then, we check if it has a variable with an h-depth greater than the given bound. If not, then we can move to the next procedure.

4. We orient all the equations. Then, we start looking for rules that might be applicable.

5. If we have $X \stackrel{?}{=} h(Y), X \stackrel{?}{=} X_1 + \ldots + X_n$ in our problem, we apply (SPLIT), since this rule creates new variables and new equations, we go back to the beginning to eliminate trivial equations, check failures and update the depths of the new variables. We repeat that until (SPLIT) is not applicable anymore.

6. Now we check if (CLASH) is applicable to see ultimately if it fails. If not, then we can apply (AC) to solve all the $+$-equations, if they exist. Since this rule creates new variables, we must update their h-depths, check if one of them surpasses the given bound and apply (OR), (SPLIT) and (CLASH) if necessary. If not, we continue this procedure until (AC) is not applicable anymore.

7. Finally, we apply (VE2) exhaustively to find a unifier to the problem.

Now, we shall present the Flattening algorithm, the intuition is very straightforward since a specific strategy to apply the Flattening rules is not required.

Algorithm 2: Flattening

Input: An equation set Γ

Output: An equation set Γ' where all of the equations are in *flattened form*.

```
while any of the flattening rules can be applied do
    Apply (FBS)
    Apply (FL)
    Apply (FR)
    Apply (FU)
    Apply (FLFUN)
```

Now let's see how the algorithm works in practice with an example:

Example 2.9. Let $\Gamma = \{h(h(X_1)+X_2) \stackrel{?}{=} h(Y_1+h(Y_2))\}$, $\Delta = \emptyset$, $\sigma = \emptyset$ and $\kappa = 3$.

0. We add all the variables in Γ, with initial depth equal zero, obtaining $\Delta = \{(X_1,0),(X_2,0)\}$.

1. Now, we apply Flattening.

   - Applying (FBS), we obtain:

$$\Gamma = \{V_1 \stackrel{?}{=} h(h(X_1)+X_2), V_1 \stackrel{?}{=} h(Y_1+h(Y_2))\}$$

$$\Delta = \{(X_1,0),(X_2,0),(Y_1,0),(Y_2,0),(V_1,0)\}$$

$$\sigma = \emptyset$$

   - Applying (FU) on each of the sub-terms highlighted before, we obtain:

$$\Gamma = \{V_1 \stackrel{?}{=} h(V_2), V_1 \stackrel{?}{=} h(V_3), V_2 \stackrel{?}{=} h(X_1)+X_2, V_3 \stackrel{?}{=} Y_1+h(Y_2)\}$$

$$\Delta = \{(X_1,0),(X_2,0),(Y_1,0),(Y_2,0),(V_1,0),(V_2,1),(V_3,1)\}$$

$$\sigma = \emptyset$$

   - Applying (FL) on the sub-term highlighted before, we obtain

$$\Gamma = \{V_1 \stackrel{?}{=} h(V_2), V_1 \stackrel{?}{=} h(V_3), V_2 \stackrel{?}{=} V_4+X_2, V_3 \stackrel{?}{=} Y_1+h(Y_2), V_4 \stackrel{?}{=} h(X_1)\}$$

$$\Delta = \{(X_1,0),(X_2,0),(Y_1,0),(Y_2,0),(V_1,0),(V_2,1),(V_3,1),(V_4,0)\}$$

$$\sigma = \emptyset$$

   - Finally, applying (FR) on the sub-term highlighted before, we obtain

$$\Gamma = \{V_1 \stackrel{?}{=} h(V_2), V_1 \stackrel{?}{=} h(V_3), V_2 \stackrel{?}{=} V_4+X_2, V_3 \stackrel{?}{=} Y_1+V_5, V_4 \stackrel{?}{=} h(X_1), V_5 \stackrel{?}{=} h(Y_2)\}$$

$$\Delta = \{(X_1,0),(X_2,0),(Y_1,0),(Y_2,0),(V_1,0),(V_2,1),(V_3,1),(V_4,0),(V_5,0)\}$$

$$\sigma = \emptyset$$

   Now that Γ is in flattened form, we can move on to the next step:

2. (a) (TRIV) is not applicable;

   (b) (OC) is not applicable;

(c) For now, we have

$$\Delta = \{(X_1,0),(X_2,0),(Y_1,0),(Y_2,0),(V_1,0),(V_2,1),(V_3,1),(V_4,0),(V_5,0)\}.$$

Hence, $\mathcal{M}ax\mathcal{V}al(\Delta) = 1 < 3 = \kappa$. Therefore, (BC) is not applicable;

(d) We have

$$\Gamma = \{V_1 \stackrel{?}{=} h(V_2), V_1 \stackrel{?}{=} h(V_3), V_2 \stackrel{?}{=} V_4 + X_2, V_3 \stackrel{?}{=} Y_1 + V_5, V_4 \stackrel{?}{=} h(X_1), V_5 \stackrel{?}{=} h(Y_2)\}$$

$$\Delta = \{(X_1,0),(X_2,0),(Y_1,0),(Y_2,0),(V_1,0),(V_2,1),(V_3,1),(V_4,0),(V_5,0)\}$$

$$\sigma = \emptyset$$

- Applying (Uh), we obtain:

$$\Gamma = \{V_1 \stackrel{?}{=} h(V_2), V_1 \stackrel{?}{=} h(V_3), V_2 \stackrel{?}{=} V_4 + X_2, V_3 \stackrel{?}{=} Y_1 + V_5, V_4 \stackrel{?}{=} h(X_1), V_5 \stackrel{?}{=} h(Y_2)\}$$

$$\Delta = \{(X_1,1),(X_2,0),(Y_1,0),(Y_2,1),(V_1,0),(V_2,1),(V_3,1),(V_4,0),(V_5,0)\}$$

$$\sigma = \emptyset$$

- Applying (UL) twice, we obtain:

$$\Gamma = \{V_1 \stackrel{?}{=} h(V_2), V_1 \stackrel{?}{=} h(V_3), V_2 \stackrel{?}{=} V_4 + X_2, V_3 \stackrel{?}{=} Y_1 + V_5, V_4 \stackrel{?}{=} h(X_1), V_5 \stackrel{?}{=} h(Y_2)\}$$

$$\Delta = \{(X_1,1),(X_2,0),(Y_1,1),(Y_2,1),(V_1,0),(V_2,1),(V_3,1),(V_4,1),(V_5,0)\}$$

$$\sigma = \emptyset$$

- Applying (UR) twice, we obtain:

$$\Gamma = \{V_1 \stackrel{?}{=} h(V_2), V_1 \stackrel{?}{=} h(V_3), V_2 \stackrel{?}{=} V_4 + X_2, V_3 \stackrel{?}{=} Y_1 + V_5, V_4 \stackrel{?}{=} h(X_1), V_5 \stackrel{?}{=} h(Y_2)\}$$

$$\Delta = \{(X_1,1),(X_2,1),(Y_1,1),(Y_2,1),(V_1,0),(V_2,1),(V_3,1),(V_4,1),(V_5,1)\}$$

$$\sigma = \emptyset$$

- Applying (Uh) twice, we obtain:

$$\Gamma = \{V_1 \stackrel{?}{=} h(V_2), V_1 \stackrel{?}{=} h(V_3), V_2 \stackrel{?}{=} V_4 + X_2, V_3 \stackrel{?}{=} Y_1 + V_5, V_4 \stackrel{?}{=} h(X_1), V_5 \stackrel{?}{=} h(Y_2)\}$$

$$\Delta = \{(X_1,2),(X_2,1),(Y_1,1),(Y_2,2),(V_1,0),(V_2,1),(V_3,1),(V_4,1),(V_5,1)\}$$

$$\sigma = \emptyset$$

(c) Now, we have $\mathcal{M}ax\mathcal{V}al(\Delta) = 2 < 3 = \kappa$. Therefore, (BC) is not applicable;

(d) Δ is already updated;

(e) All the equations are oriented. Therefore, (OR) is not applicable;

(f) (SPLIT) is not applicable;

(g) (CLASH) is not applicable;

(h) We have

$$\Gamma = \{V_1 \stackrel{?}{=} h(V_2), V_1 \stackrel{?}{=} h(V_3), V_2 \stackrel{?}{=} V_4 + X_2, V_3 \stackrel{?}{=} Y_1 + V_5, V_4 \stackrel{?}{=} h(X_1), V_5 \stackrel{?}{=} h(Y_2)\}$$

$$\Delta = \{(X_1,2),(X_2,1),(Y_1,1),(Y_2,2),(V_1,0),(V_2,1),(V_3,1),(V_4,1),(V_5,1)\}$$

$$\sigma = \emptyset$$

So, we can apply (DEC) on the highlighted equations in Γ, obtaining

$$\Gamma = \{V_1 \stackrel{?}{=} h(V_2), V_2 \stackrel{?}{=} V_3, V_2 \stackrel{?}{=} V_4 + X_2, V_3 \stackrel{?}{=} Y_1 + V_5, V_4 \stackrel{?}{=} h(X_1), V_5 \stackrel{?}{=} h(Y_2)\}$$

$$\Delta = \{(X_1,2),(X_2,1),(Y_1,1),(Y_2,2),(V_1,0),(V_2,1),(V_3,1),(V_4,1),(V_5,1)\}$$

$$\sigma = \emptyset$$

(VE1) is applicable ($V_2 \mapsto V_3$). Hence, we have

$$\Gamma = \{V_1 \stackrel{?}{=} h(V_3), V_3 \stackrel{?}{=} V_4 + X_2, V_3 \stackrel{?}{=} Y_1 + V_5, V_4 \stackrel{?}{=} h(X_1), V_5 \stackrel{?}{=} h(Y_2)\}$$

$$\Delta = \{(X_1,2),(X_2,1),(Y_1,1),(Y_2,2),(V_1,0),(V_2,1),(V_3,1),(V_4,1),(V_5,1)\}$$

$$\sigma = \{V_2 \mapsto V_3\}$$

Notice that no rules invoked by steps (a) to (h) is applicable anymore. Thus, we can move on to the next step.

(i) We have

$$\Gamma = \{V_1 \stackrel{?}{=} h(V_3), V_3 \stackrel{?}{=} V_4 + X_2, V_3 \stackrel{?}{=} Y_1 + V_5, V_4 \stackrel{?}{=} h(X_1), V_5 \stackrel{?}{=} h(Y_2)\}$$

$$\Delta = \{(X_1,2),(X_2,1),(Y_1,1),(Y_2,2),(V_1,0),(V_2,1),(V_3,1),(V_4,1),(V_5,1)\}$$

$$\sigma = \{V_2 \mapsto V_3\}$$

Applying (AC), we obtain 7 different possibilities for Γ, one of them is:

$$\Gamma_1 = \{V_3 \stackrel{?}{=} Y_1 + V_5, V_4 \stackrel{?}{=} Y_1, X_2 \stackrel{?}{=} V_5, V_1 \stackrel{?}{=} h(V_3), V_4 \stackrel{?}{=} h(X_1), V_5 \stackrel{?}{=} h(Y_2)\}$$

It is important to remind that we chose this specific possibility just for simplicity of the example. However, the algorithm itself continues to solve all the other problems in the disjunction simultaneously.

In this case,

$$\Delta_1 = \{(X_1,2),(X_2,1),(Y_1,1),(Y_2,2),(V_1,0),(V_2,1),(V_3,1),(V_4,1),(V_5,1)\}$$

So, Δ_1 is already updated and $\mathcal{M}ax\mathcal{V}al(\Delta) = 2 < 3 = \kappa$.
Notice that (VE1) is applicable ($X_2 \mapsto V_5$). Hence, we obtain

$$\Gamma_1 = \{V_3 \stackrel{?}{=} Y_1 + V_5, V_4 \stackrel{?}{=} Y_1, V_1 \stackrel{?}{=} h(V_3), V_4 \stackrel{?}{=} h(X_1), V_5 \stackrel{?}{=} h(Y_2)\}$$

$$\Delta_1 = \{(X_1,2),(X_2,1),(Y_1,1),(Y_2,2),(V_1,0),(V_2,1),(V_3,1),(V_4,1),(V_5,1)\}$$

$$\sigma_1 = \{V_2 \mapsto V_3, X_2 \mapsto V_5\}$$

We can apply (VE1) one more time ($V_4 \mapsto Y_1$), obtaining

$$\Gamma_1 = \{V_3 \stackrel{?}{=} Y_1 + V_5, V_1 \stackrel{?}{=} h(V_3), Y_1 \stackrel{?}{=} h(X_1), V_5 \stackrel{?}{=} h(Y_2)\}$$

$$\Delta_1 = \{(X_1,2),(X_2,1),(Y_1,1),(Y_2,2),(V_1,0),(V_2,1),(V_3,1),(V_4,1),(V_5,1)\}$$

$$\sigma_1 = \{V_2 \mapsto V_3, X_2 \mapsto V_5, V_4 \mapsto Y_1\}$$

(j) Now, we can apply (VE2) exhaustively

- Applying $V_1 \mapsto h(V_3)$, we obtain:

$$\Gamma_1 = \{V_3 \stackrel{?}{=} Y_1 + V_5, Y_1 \stackrel{?}{=} h(X_1), V_5 \stackrel{?}{=} h(Y_2)\}$$

$$\Delta_1 = \{(X_1,2),(X_2,1),(Y_1,1),(Y_2,2),(V_1,0),(V_2,1),(V_3,1),(V_4,1),(V_5,1)\}$$

$$\sigma_1 = \{V_2 \mapsto V_3, X_2 \mapsto V_5, V_4 \mapsto Y_1, V_1 \mapsto h(V_3)\}$$

- Applying $Y_1 \mapsto h(X_1)$, we obtain:

$$\Gamma_1 = \{V_3 \stackrel{?}{=} h(X_1) + V_5, V_5 \stackrel{?}{=} h(Y_2)\}$$

$$\Delta_1 = \{(X_1,2),(X_2,1),(Y_1,1),(Y_2,2),(V_1,0),(V_2,1),(V_3,1),(V_4,1),(V_5,1)\}$$

$$\sigma_1 = \{V_2 \mapsto V_3, X_2 \mapsto V_5, V_4 \mapsto h(X_1), V_1 \mapsto h(V_3), Y_1 \mapsto h(X_1)\}$$

- Applying $V_5 \mapsto h(Y_2)$, we obtain:

$$\Gamma_1 = \{V_3 \stackrel{?}{=} h(X_1) + h(Y_2)\}$$

$$\Delta_1 = \{(X_1,2),(X_2,1),(Y_1,1),(Y_2,2),(V_1,0),(V_2,1),(V_3,1),(V_4,1),(V_5,1)\}$$

$$\sigma_1 = \{V_2 \mapsto V_3, X_2 \mapsto h(Y_2), V_4 \mapsto h(X_1), V_1 \mapsto h(V_3), Y_1 \mapsto h(X_1), V_5 \mapsto h(Y_2)\}$$

- Applying $V_3 \mapsto h(X_1) + h(Y_2)$, we obtain:

$$\Gamma_1 = \emptyset$$

$$\Delta_1 = \{(X_1,2),(X_2,1),(Y_1,1),(Y_2,2),(V_1,0),(V_2,1),(V_3,1),(V_4,1),(V_5,1)\}$$

$$\begin{aligned}\sigma_1 = &\{V_2 \mapsto h(X_1) + h(Y_2), X_2 \mapsto h(Y_2), V_4 \mapsto h(X_1), V_1 \mapsto h(h(X_1) + h(Y_2)),\\ &Y_1 \mapsto h(X_1), V_5 \mapsto h(Y_2)\}\end{aligned}$$

Chapter 3

Correctness of the Algorithm

In this chapter, we will prove termination (Corollary 3.1) and correctness of the Algorithm $\mathsf{Unify}_{\mathsf{AC_h}}$ given in Chapter 2. Proving correctness consists in proving soundness (Corollary 3.2) and completeness (Corollary 3.3) of $\mathsf{Unify}_{\mathsf{AC_h}}$. These results guarantee that the algorithm $\mathsf{Unify}_{\mathsf{AC_h}}$ always terminates, it is truth-preserving and does not leave any solution behind, respectively.

3.1 Auxiliary Notions

Before presenting those proofs, we shall introduce some notations that will be used alongside this chapter. For two triples $\Gamma \parallel \Delta \parallel \sigma$ and $\Gamma' \parallel \Delta' \parallel \sigma'$,

- $\Gamma \parallel \Delta \parallel \sigma \Rightarrow_{\mathfrak{I}_{ACh}} \Gamma' \parallel \Delta' \parallel \sigma'$ means that $\Gamma' \parallel \Delta' \parallel \sigma'$ is obtained from $\Gamma \parallel \Delta \parallel \sigma$ after applying a rule from $\mathfrak{I}_{ACh}$ once. We call it *one step*.

- $\Gamma \parallel \Delta \parallel \sigma \Rightarrow_{Flat} \Gamma' \parallel \Delta' \parallel \sigma'$ means that $\Gamma' \parallel \Delta' \parallel \sigma'$ is obtained after applying a Flattening rule from $\mathfrak{I}_{ACh}$ once

- $\Gamma \parallel \Delta \parallel \sigma \Rightarrow^*_{\mathfrak{I}_{ACh}} \Gamma' \parallel \Delta' \parallel \sigma'$ means that $\Gamma' \parallel \Delta' \parallel \sigma'$ is obtained from $\Gamma \parallel \Delta \parallel \sigma$ by *zero or more steps*.

- $\Gamma \parallel \Delta \parallel \sigma \Rightarrow^+_{\mathfrak{I}_{ACh}} \Gamma' \parallel \Delta' \parallel \sigma'$ means that $\Gamma' \parallel \Delta' \parallel \sigma'$ is obtained from $\Gamma \parallel \Delta \parallel \sigma$ by *one or more steps*.

Notice that, since the AC unification rule divides our unification problem $\Gamma \parallel \Delta \parallel \sigma$ in $\Gamma_1 \parallel \Delta_1 \parallel \sigma_1, \ldots, \Gamma_n \parallel \Delta_n \parallel \sigma_n$, we have that, after applying some inference rules, the result is a disjunction of set triples $\bigvee_i (\Gamma_i \parallel \Delta_i \parallel \sigma_i)$. Hence, we present the following notation

– $\Gamma \parallel \Delta \parallel \sigma \Rightarrow_{\mathfrak{I}_{ACh}} \bigvee_{i\in I}(\Gamma_i \parallel \Delta_i \parallel \sigma_i)$ means that $\bigvee_{i\in I}(\Gamma_i \parallel \Delta_i \parallel \sigma_i)$ is obtained from $\Gamma \parallel \Delta \parallel \sigma$ after applying a rule from $\mathfrak{I}_{ACh}$ *one time*.

– $\Gamma \parallel \Delta \parallel \sigma \Rightarrow^*_{\mathfrak{I}_{ACh}} \bigvee_{i\in I}(\Gamma_i \parallel \Delta_i \parallel \sigma_i)$ means that $\bigvee_{i\in I}(\Gamma_i \parallel \Delta_i \parallel \sigma_i)$ is obtained from $\Gamma \parallel \Delta \parallel \sigma$ after applying a rule from $\mathfrak{I}_{ACh}$ *zero or more times*.

– $\Gamma \parallel \Delta \parallel \sigma \Rightarrow^+_{\mathfrak{I}_{ACh}} \bigvee_{i\in I}(\Gamma_i \parallel \Delta_i \parallel \sigma_i)$ means that $\bigvee_{i\in I}(\Gamma_i \parallel \Delta_i \parallel \sigma_i)$ is obtained from $\Gamma \parallel \Delta \parallel \sigma$ after applying a rule from $\mathfrak{I}_{ACh}$ *one or more times*.

3.2 Termination

As we discussed before (Section 1.2), proving termination of $\mathfrak{I}_{ACh}$ consists in finding a *measure function* and prove that it decreases after each step $\Rightarrow_{\mathfrak{I}_{ACh}}$. Since our algorithm is divided into two parts (Flattening and $\mathsf{Unify}_{\mathsf{AC_h}}$), we will also split the termination proof into two parts as well.

3.2.1 Termination of Flattening

First, consider a multiset $\mathcal{M}(\Gamma)$, whose the elements are the number of function symbols on each equation in Γ. For example, take

$$\Gamma = \{X_1 + X_2 \stackrel{?}{=} h(Y_1+Y_2), X_1 + h(Y_1) \stackrel{?}{=} Z_2 + Z_3, f(Z_1 + h(Z_2+Z_3)) \stackrel{?}{=} W_1 + W_2\}$$

Notice that $X_1 + X_2 \stackrel{?}{=} h(Y_1+Y_2)$, $X_1 + h(Y_1) \stackrel{?}{=} Z_2 + Z_3$ and $f(Z_1 + h(Z_2+Z_3)) \stackrel{?}{=} W_1 + W_2$ have 3, 3 and 5 function symbols, respectively, then $\mathcal{M}(\Gamma) = \{3,3,5\}$.

We define a measure on $\Gamma \parallel \Delta \parallel \sigma$ as the multiset ordering $>_{mul}$ on $\mathcal{M}(\Gamma)$. Below we write $\Rightarrow_{Flat}$ to denote one step of application of one of the flattening rules.

Lemma 3.1. Let $\Gamma \parallel \Delta \parallel \sigma$ and $\Gamma' \parallel \Delta' \parallel \sigma'$ be two set triples such that $\Gamma \parallel \Delta \parallel \sigma \Rightarrow_{Flat} \Gamma' \parallel \Delta' \parallel \sigma'$. Then, $\mathcal{M}(\Gamma) >_{mul} \mathcal{M}(\Gamma')$.

Proof. We will prove that, in each application of a flattening rule in Γ, $>_{mul}$ decreases, that is, $\mathcal{M}(\Gamma) >_{mul} \mathcal{M}(\Gamma')$. We proceed by analysing each rule:

(Flatten Both Sides) In this case, $\Gamma = \{t_1 \stackrel{?}{=} t_2\} \cup \overline{\Gamma}$, with $t_1, t_2 \notin \mathcal{V}$.

Applying (FBS), we obtain

$$\text{(FBS)}\frac{\{t_1 \stackrel{?}{=} t_2\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma}{\{V \stackrel{?}{=} t_1, V \stackrel{?}{=} t_2\} \cup \overline{\Gamma} \parallel \Delta \cup \{(V,0)\} \parallel \sigma}$$

Define $\Gamma' = \{V \stackrel{?}{=} t_1, V \stackrel{?}{=} t_2\} \cup \overline{\Gamma}$ and let n_i be the number of function symbols in t_i, with $i = 1,2$. Since $t_1, t_2 \notin \mathcal{V}$, we have that $n_1 \neq 0$ and $n_2 \neq 0$. Then, we have that

$$\begin{aligned}\mathcal{M}(\Gamma) &= \{n_1 + n_2\} \cup \mathcal{M}(\overline{\Gamma}) \\ &>_{mul} \{n_1, n_2\} \cup \mathcal{M}(\overline{\Gamma}) \\ &= \mathcal{M}(\Gamma')\end{aligned}$$

Notice that the result also holds if t_1, t_2 are constants since constants are function symbols with arity zero. In this case, we would have

$$\begin{aligned}\mathcal{M}(\Gamma) &= \{2\} \cup \mathcal{M}(\overline{\Gamma}) \\ &>_{mul} \{1,1\} \cup \mathcal{M}(\overline{\Gamma}) \\ &= \mathcal{M}(\Gamma')\end{aligned}$$

(Flatten Left/Right $+$) We will prove only for the left case since the argument for the right case is similar.
Here, $\Gamma = \{t \stackrel{?}{=} t_1 + t_2\}$, with $t_1 \notin \mathcal{V}$. Applying (FL), we obtain

$$\text{(FL)}\frac{\{t \stackrel{?}{=} t_1 + t_2\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma}{\{t \stackrel{?}{=} V + t_2, V \stackrel{?}{=} t_1\} \cup \overline{\Gamma} \parallel \Delta \cup \{(V,0)\} \parallel \sigma}$$

Define $\Gamma' = \{t \stackrel{?}{=} V + t_2, V \stackrel{?}{=} t_1\} \cup \overline{\Gamma}$ and let n be the number of function symbols in t and n_i be the number of function symbols in t_i, with $i = 1,2$. We have that $n_1 \geq 1$, since $t_1 \notin \mathcal{V}$. Notice that there is an $+$ symbol on top of t_1 and t_2.Then $\mathcal{M}(\Gamma) = \{n + n_1 + n_2 + 1\} \cup \mathcal{M}(\overline{\Gamma})$

$$\begin{aligned}\mathcal{M}(\Gamma) &= \{n + n_1 + n_2 + 1\} \cup \mathcal{M}(\overline{\Gamma}) \\ &>_{mul} \{n + n_2 + 1, n_1\} \cup \mathcal{M}(\overline{\Gamma}) \\ &= \mathcal{M}(\Gamma')\end{aligned}$$

Flatten Under h: In this case, $\Gamma = \{t \stackrel{?}{=} h(t_1)\}$, with $t_1 \notin \mathcal{V}$. Applying (FU), we obtain

$$(\mathrm{FU})\frac{\{t \stackrel{?}{=} h(t_1)\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma}{\{t \stackrel{?}{=} h(V), V \stackrel{?}{=} t_1\} \cup \overline{\Gamma} \parallel \Delta \cup \{(V,1)\} \parallel \sigma} \text{ if } t_1 \notin \mathcal{V}$$

Define $\Gamma' = \{t \stackrel{?}{=} h(V), V \stackrel{?}{=} t_1\} \cup \overline{\Gamma}$ and let n be the number of function symbols in t and n_1 be the number of function symbols in t_1. Notice that there is an h symbol on top of t_1, then $\mathcal{M}(\Gamma) = \{n + n_1 + 1\} \cup \mathcal{M}(\overline{\Gamma})$ and

$$\begin{aligned}\mathcal{M}(\Gamma) &= \{n + n_1 + 1\} \cup \mathcal{M}(\overline{\Gamma}) \\ &>_{mul} \{n+1, n_1\} \cup \mathcal{M}(\overline{\Gamma}) \\ &= \mathcal{M}(\Gamma')\end{aligned}$$

Flatten Under f: In this case, $\Gamma = \{t \stackrel{?}{=} h(t_1)\}$, with $t_1 \notin \mathcal{V}$. Applying (FLFUN), we obtain

$$(\mathrm{FLFUN})\frac{\{t \stackrel{?}{=} f(t_1, \ldots, t_m)\} \cup \Gamma \parallel \Delta \parallel \sigma}{\{t \stackrel{?}{=} f(V_1, \ldots, V_m), V_1 \stackrel{?}{=} t_1, \ldots, V_m \stackrel{?}{=} t_m\} \cup \Gamma \parallel \Delta \cup \{(V_1, 0), \ldots, (V_n, 0)\} \parallel \sigma}$$

$$\text{if } t_1, \ldots, t_n \notin \mathcal{V}$$

Define $\Gamma' = \{t \stackrel{?}{=} f(V_1, \ldots, V_m), V_1 \stackrel{?}{=} t_1, \ldots, V_m \stackrel{?}{=} t_m\} \cup \overline{\Gamma}$ and let n be the number of function symbols in t and n_i be the number of function symbols in t_i, for all $i = 1, \ldots, m$. Notice that there is an f symbol on top all t_i's, then $\mathcal{M}(\Gamma) = \{n + n_1 + \ldots + n_m + 1\} \cup \mathcal{M}(\overline{\Gamma})$ and

$$\begin{aligned}\mathcal{M}(\Gamma) &= \{n + n_1 + \ldots + n_m + 1\} \cup \mathcal{M}(\overline{\Gamma}) \\ &>_{mul} \{n+1, n_1, \ldots, n_m\} \cup \mathcal{M}(\overline{\Gamma}) \\ &= \mathcal{M}(\Gamma')\end{aligned}$$

□

From this point forward, for convenience, we will assume that our set of equations Γ is always in flattened form, unless we explicitly say otherwise. That assumption is possible because $\mathsf{Unify}_{\mathsf{AC_h}}$ invokes $\mathsf{Flattening}$ on its first step.

3.2.2 Termination of $\mathsf{Unify}_{\mathsf{AC_h}}$

Now, we prove that $\mathsf{Unify}_{\mathsf{AC_h}}$ terminates. Before giving the measure for termination, we will define some necessary concepts.

Definition 3.1 (Isolated Variable). Let Γ be a unification problem and $X \in \mathcal{V}ar(\Gamma)$. We say that X is an *isolated variable* if $X \stackrel{?}{=} t \in \Gamma$, with $X \notin \mathcal{V}ar(t)$.

In order to verify statements that will follow, we had to define what is a variable that is "solved for the AC theory". Intuitively, it would be a variable that does not invoke an application of the (AC) rule. Notice that this rule is invoked when we have a variable that occurs on the left side of $+$-equations more than once, that is, for example, $X \stackrel{?}{=} X_1 + \ldots + X_n$ and $X \stackrel{?}{=} Y_1 + \ldots + Y_n$ occurs on the same problem. Hence, we obtained the following definition:

Definition 3.2 (AC-solved variable). Let Γ be a unification problem and $X \in \mathcal{V}ar(\Gamma)$. We say that X is *AC-solved* if X is isolated and, after applying *Orient* (OR) exhaustively in Γ, X does not occur in $+$-equations with X on the left side more than once.

Example 3.1. Let $\Gamma = \{X = Y_1 + Y_2, X = V_1 + V_2\}$. Then X is not AC-solved, whereas for $\Gamma' = \{X = Y_1 + Y_2, Y = V_1 + V_2\}$, both X and Y are AC-solved.

It is obvious that, after applying the (AC) rule, all the variables that are not AC-solved in our problem become AC-solved. More than that, since this rule recalls an established AC-Unification algorithm (which is terminating, sound and complete), it is guaranteed that, after applying the rule, no AC-solved variable becomes non-solved.

It is also important to notice that, if X is AC-solved, there is no other rule in $\mathfrak{I}_{ACh}$, except for (VE) (see example below), that makes X not AC-solved again. It is very simple to verify this affirmation, it just requires to check rule by rule – except (VE) and (OR) since we assume that all the equations are already oriented by definition.

Example 3.2. Let $\Gamma = \{X \stackrel{?}{=} Y_1 + Y_2, Y \stackrel{?}{=} h(X), Y \stackrel{?}{=} h(Y_3 + Y_4)\}$. We have

$$\{X \stackrel{?}{=} Y_1 + Y_2, Y \stackrel{?}{=} h(X), Y \stackrel{?}{=} h(Y_3 + Y_4)\}$$
$$\Longrightarrow_{(\mathrm{FU})}$$
$$\{X \stackrel{?}{=} Y_1 + Y_2, Y \stackrel{?}{=} h(X), Y \stackrel{?}{=} h(V), V \stackrel{?}{=} Y_3 + Y_4\}$$
$$\Longrightarrow_{(\mathrm{DEC})}$$
$$\{X \stackrel{?}{=} Y_1 + Y_2, Y \stackrel{?}{=} h(X), X \stackrel{?}{=} V, V \stackrel{?}{=} Y_3 + Y_4\} \tag{*}$$
$$\Longrightarrow_{(\mathrm{VE1})}$$
$$\{V \stackrel{?}{=} Y_1 + Y_2, V \stackrel{?}{=} Y_3 + Y_4, Y \stackrel{?}{=} h(V)\}$$

Notice that in $(*)$, both X and V are AC-solved, according to our definition. But, after applying (VE1), V becomes non-solved.

Proposition 3.1. Let Γ be an ACh-unification problem and $X \in \mathcal{V}ar(\Gamma)$ an AC-solved variable. Then, for all $\Gamma' \parallel \Delta' \parallel \sigma'$ such that $\Gamma \parallel \Delta \parallel \sigma \Rightarrow_{\mathfrak{I}_{ACh}-(\text{VE1/OR})} \Gamma' \parallel \Delta' \parallel \sigma'$, if $X \in \mathcal{V}ar(\Gamma')$ then X is AC-solved.

Proof. The proof follows by analysing the rule applied in $\Gamma \parallel \Delta \parallel \sigma$, with Δ and σ arbitrary. For (AC), it is obvious, as we discussed previously. As for the checking rules (cf. Section 2.1.5), if we have an AC-solved variable, after applying one of these rules, despite indicating immediate failure, we did not make the variable non-solved, so we will skip their demonstrations as well.

Let $X \in \mathcal{V}$ be an AC-solved variable occurring in Γ.

(Trivial) In this case, we have $\Gamma = \{X \stackrel{?}{=} t\} \cup \{s \stackrel{?}{=} s\} \cup \overline{\Gamma}$. Then, applying (TRIV), we obtain

$$\{X \stackrel{?}{=} t\} \cup \{s \stackrel{?}{=} s\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma \Rightarrow_{\mathfrak{I}_{ACh}} \{X \stackrel{?}{=} t\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma.$$

Notice that X remains isolated and still does not occur in $+$-equations more than once. Therefore, X remains AC-solved.

(Splitting) In this case, we have $\Gamma = \{X \stackrel{?}{=} t\} \cup \{Y \stackrel{?}{=} h(Z), Y \stackrel{?}{=} X_1 + \ldots + X_n\} \cup \overline{\Gamma}$. After applying (SPLIT), we obtain

$$\begin{gathered}\{X \stackrel{?}{=} t\} \cup \{Y \stackrel{?}{=} h(Z), Y \stackrel{?}{=} X_1 + \ldots + X_n\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma \\ \Downarrow_{\mathfrak{I}_{ACh}} \\ \{X \stackrel{?}{=} t\} \cup \{Y \stackrel{?}{=} h(Z), Z \stackrel{?}{=} V_1 + \ldots + V_n, X_1 \stackrel{?}{=} h(V_1), \ldots, X_n \stackrel{?}{=} h(V_n)\} \cup \overline{\Gamma} \parallel \Delta' \parallel \sigma,\end{gathered}$$

where $\Delta' = \{(V_1, 1), \ldots, (V_n, 1)\} \cup \Delta$. Notice that X still does not occur on the left side of $+$-equations more than once. Hence, it is still AC-solved.

(Decomposition) In this case, we have

$$\Gamma = \{X \stackrel{?}{=} t\} \cup \{Y \stackrel{?}{=} f(X_n, \ldots, X_n), Y \stackrel{?}{=} f(Y_n, \ldots, Y_n)\} \cup \overline{\Gamma}.$$

After applying (DEC), we obtain

$$\{X \stackrel{?}{=} t\} \cup \{Y \stackrel{?}{=} f(X_n,\ldots,X_n), Y \stackrel{?}{=} f(Y_n,\ldots,Y_n)\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma$$
$$\Downarrow_{\mathfrak{I}_{ACh}}$$
$$\{X \stackrel{?}{=} t\} \cup \{Y \stackrel{?}{=} f(X_n,\ldots,X_n), X_1 \stackrel{?}{=} Y_1,\ldots,X_n \stackrel{?}{=} Y_n\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma,$$

Notice that X still does not occur on the left side of $+$-equations more than once. Therefore, it remains AC-solved.

□

Now, we shall define the following lexicographic measure for $\Gamma \parallel \Delta \parallel \sigma$ to prove termination of $\mathsf{Unify}_{\mathsf{AC_h}}$.

Definition 3.3 (Measure). Let $\Gamma \parallel \Delta \parallel \sigma$ be a triple. Consider the following measure for $\Gamma \parallel \Delta \parallel \sigma$:

$$\mathbb{M}_{\mathfrak{I}_{ACh}}(\Gamma,\Delta,\sigma,\kappa) := (\kappa - a, n_X, |\mathcal{S}ym(\Gamma)|, m, p, |\Gamma|, \overline{h_d}(\Delta)), \text{ where}$$

1. Let $\mathcal{S}ym(\Gamma)$ be a multiset of non-variable symbols occurring in Γ. Then $|\mathcal{S}ym(\Gamma)|$ is the standard ordering on the size of Γ based on the natural numbers.
2. Let κ be a given bound. Define the multiset

   $$\overline{h_d}(\Delta) := \{(\kappa+1) - h_d(X,\Gamma) \mid (X, h_d(X,\Gamma)) \in \Delta\}.$$

   Then we use the corresponding multiset order $>_{mul}$ for $\overline{h_d}(\Delta)$.
3. a is be the number of applications of the (AC) rule.
4. p is the number of isolated variables in Γ.
5. m is be the number of equations on the form $t \stackrel{?}{=} X \in \Gamma$, with $t \notin \mathcal{V}$.
6. Let $X \in \mathcal{V}ar(\Gamma)$. Then n_X is the number of occurrences of $+$-equations with the fixed X on the left side, that is, equations of the form $X \stackrel{?}{=} X_1 + \ldots + X_n$.

The measure above is different than the the measure in [EL20] in four ways:

i) We introduced the parameter κ in the measure, since it depends on κ as well;

ii) We swapped the positions of m and p in order to guarantee that the measure always decreases (see);

iii) We clarified that the entry n_X depends on a fixed variable X occurring in the problem;

iv) We changed the notation of the last parameter to $\overline{h_d}(\Delta)$. In the original work it was $\overline{h_d}(\Gamma)$, however, Γ is not altered when we change the depth of the variables.

Considering this measure, we can define a lexicographic order $>_{lex}$ on $\mathbb{M}_{\mathfrak{I}_{ACh}}(\Gamma, \Delta, \sigma, \kappa)$ induced by the product of orderings $>_{\mathbb{N}}$ over $\mathbb{N}$ and $>_{mul}$.

We have to guarantee that the first entry is always greater than zero, i.e. that the number of times we apply (AC) does not exceed the given bound.

Example 3.3. Let's recall Example 2.9. Before applying (AC), our triple consisted of the following sets

$$\Gamma = \{V_1 \stackrel{?}{=} h(V_3), V_3 \stackrel{?}{=} V_4 + X_2, V_3 \stackrel{?}{=} Y_1 + V_5, V_4 \stackrel{?}{=} h(X_1), V_5 \stackrel{?}{=} h(Y_2)\}$$

$$\Delta = \{(X_1,2),(X_2,1),(Y_1,1),(Y_2,2),(V_1,0),(V_2,1),(V_3,1),(V_4,1),(V_5,1)\}$$

$$\sigma = \{V_2 \mapsto V_3\}$$

Notice that V_3 is the variable with the lowest depth that is not AC-solved. After applying (AC), we obtained

$$\Gamma_1 = \{V_3 \stackrel{?}{=} Y_1 + V_5, V_4 \stackrel{?}{=} Y_1, X_2 \stackrel{?}{=} V_5, V_1 \stackrel{?}{=} h(V_3), V_4 \stackrel{?}{=} h(X_1), V_5 \stackrel{?}{=} h(Y_2)\}$$

$$\Delta_1 = \{(X_1,2),(X_2,1),(Y_1,1),(Y_2,2),(V_1,0),(V_2,1),(V_3,1),(V_4,1),(V_5,1)\}$$

$$\sigma_1 = \{V_2 \mapsto V_3, X_2 \mapsto V_5\}$$

as one of the solutions, notice that now V_3 becomes AC-solved. Then, we move to the variables with greater h-depth.

Hence, we have the following lemma.

Lemma 3.2. Let $\Gamma \parallel \Delta \parallel \sigma$ be a set triple and $\kappa \in \mathbb{N}$ be a bound given as an input to $\mathsf{Unify}_{\mathsf{AC_h}}$. The maximum number of times that the AC Unification is applied is κ.

Proof. Notice that we only apply (AC) if there is at least one variable that is not AC-solved in the problem. On each application of (AC), all the variables with the lowest h-depth becomes AC-solved. By Proposition 3.1, there is no other rule in $\mathfrak{I}_{ACh}$ that makes these variables not AC-solved again. Therefore, we do not surpass the given bound. □

Now, we prove that $\mathbb{M}_{\mathfrak{I}_{ACh}}(\Gamma, \Delta, \sigma, \kappa)$ is always decreasing. The proof follows by analysing the rule applied in $\mathfrak{I}_{ACh}$

Theorem 3.1. Let $\Gamma \parallel \Delta \parallel \sigma$ and $\Gamma' \parallel \Delta' \parallel \sigma'$ be two set triples that are already in flattened form and $\Gamma \parallel \Delta \parallel \sigma \Rightarrow_{\mathfrak{I}_{ACh}} \Gamma' \parallel \Delta' \parallel \sigma'$. Then $\mathbb{M}_{\mathfrak{I}_{ACh}}(\Gamma, \Delta, \sigma, \kappa) > \mathbb{M}_{\mathfrak{I}_{ACh}}(\Gamma', \Delta', \sigma', \kappa)$.

Proof. We have to prove that the measure decreases after each application of the inference rules.

(Trivial) In this case, $\Gamma = \{t \stackrel{?}{=} t\} \cup \overline{\Gamma}$. The reduction is as

$$\{t \stackrel{?}{=} t\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma \Rightarrow_{\mathfrak{I}_{ACh}} \overline{\Gamma} \parallel \Delta \parallel \sigma.$$

Notice that $|\{t \stackrel{?}{=} t\} \cup \overline{\Gamma}| > |\overline{\Gamma}| = |\{t \stackrel{?}{=} t\} \cup \overline{\Gamma}| - 1$. Hence, we have that

$$\begin{aligned} \mathbb{M}_{\mathfrak{I}_{ACh}}(\{t \stackrel{?}{=} t\} \cup \overline{\Gamma}, \Delta, \sigma, \kappa) &= (\kappa - a, n_X, |Sym(\{t \stackrel{?}{=} t\} \cup \overline{\Gamma})|, m, p, |\{t \stackrel{?}{=} t\} \cup \overline{\Gamma}|, \overline{h_d}(\Delta)) \\ &>_{lex} (\kappa - a, n_X, |Sym(\overline{\Gamma})|, m, p, |\overline{\Gamma}|, \overline{h_d}(\Delta)) \\ &= \mathbb{M}_{\mathfrak{I}_{ACh}}(\overline{\Gamma}, \Delta, \sigma, \kappa) \end{aligned}$$

(Orient) In this case, $\Gamma = \{t \stackrel{?}{=} X\} \cup \overline{\Gamma}$. The reduction is as

$$\{t \stackrel{?}{=} X\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma \Rightarrow_{\mathfrak{I}_{ACh}} \{X \stackrel{?}{=} t\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma.$$

We have that

$$\begin{aligned} \mathbb{M}_{\mathfrak{I}_{ACh}}(\{t \stackrel{?}{=} X\} \cup \overline{\Gamma}, \Delta, \sigma, \kappa) &= (\kappa - a, n_X, |Sym(\{t \stackrel{?}{=} X\} \cup \overline{\Gamma})|, m, p, |\Gamma|, \overline{h_d}(\Delta)) \\ &>_{lex} (\kappa - a, n_X, |Sym(\{X \stackrel{?}{=} t\} \cup \overline{\Gamma})|, m-1, p, |\Gamma|, \overline{h_d}(\Delta)) \\ &= \mathbb{M}_{\mathfrak{I}_{ACh}}(\{X \stackrel{?}{=} t\} \cup \overline{\Gamma}, \Delta, \sigma, \kappa) \end{aligned}$$

(Variable Elimination) In both cases of (VE1) and (VE2), $\Gamma = \{X \stackrel{?}{=} t\} \cup \overline{\Gamma}$. The reduction is as

$$\{X \stackrel{?}{=} t\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma \Rightarrow_{\mathfrak{I}_{ACh}} \overline{\Gamma}\{X \mapsto t\} \parallel \Delta \parallel \sigma\{X \mapsto t\}.$$

Hence, we have

$$\begin{aligned} \mathbb{M}_{\mathfrak{I}_{ACh}}(\{X \stackrel{?}{=} t\} \cup \overline{\Gamma}, \Delta, \sigma, \kappa) &= (\kappa - a, n_X, |Sym(\{X \stackrel{?}{=} t\} \cup \overline{\Gamma})|, m, p, |\{X \stackrel{?}{=} t\} \cup \overline{\Gamma}|, \overline{h_d}(\Delta)) \\ &>_{lex} (\kappa - a, n_X, |Sym(\overline{\Gamma}\{X \mapsto t\})|, m, p-1, |\overline{\Gamma}\{X \mapsto t\}|, \overline{h_d}(\Delta)) \\ &= \mathbb{M}_{\mathfrak{I}_{ACh}}(\overline{\Gamma}\{X \mapsto t\}, \Delta, \sigma\{X \mapsto t\}, \kappa) \end{aligned}$$

(Decomposition) In this case, $\Gamma = \{X \stackrel{?}{=} f(X_1,\ldots,X_n), X \stackrel{?}{=} f(Y_1,\ldots,Y_n)\} \cup \overline{\Gamma}$. The reduction is as

$$\begin{gathered}\{X \stackrel{?}{=} f(X_1,\ldots,X_n), X \stackrel{?}{=} f(Y_1,\ldots,Y_n)\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma \\ \Downarrow_{\mathfrak{I}_{ACh}} \\ \{X \stackrel{?}{=} f(X_1,\ldots,X_n), X_1 \stackrel{?}{=} Y_1,\ldots,X_n \stackrel{?}{=} Y_n\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma.\end{gathered}$$

Notice that the number of function symbols is decreased by 1, that is

$$\begin{aligned}|\mathcal{S}ym(\Gamma')| &= |\mathcal{S}ym(\{X \stackrel{?}{=} f(t_1,\ldots,t_n), s_1 \stackrel{?}{=} t_1,\ldots,s_n \stackrel{?}{=} t_n\} \cup \overline{\Gamma})| \\ &= |\{X \stackrel{?}{=} f(s_1,\ldots,s_n), X \stackrel{?}{=} f(t_1,\ldots,t_n)\} \cup \overline{\Gamma}| - 1 \\ &= |\mathcal{S}ym(\Gamma)| - 1.\end{aligned}$$

Hence, we have that

$$\begin{aligned}\mathbb{M}_{\mathfrak{I}_{ACh}}(\Gamma,\Delta,\sigma,\kappa) &= (\kappa - a, n_X, |\mathcal{S}ym(\Gamma)|, m, p, |\Gamma|, \overline{h_d}(\Delta)) \\ &>_{lex} (\kappa - a, n_X, |\mathcal{S}ym(\Gamma)| - 1, m, p, |\Gamma|, \overline{h_d}(\Delta)) \\ &= \mathbb{M}_{\mathfrak{I}_{ACh}}(\Gamma',\Delta,\sigma,\kappa)\end{aligned}$$

(Update h-depth Set) In this case, Γ remains unaltered, but we have $\Delta = \{(X,d)\} \cup \overline{\Delta}$ and the reduction is as

$$\Gamma \parallel \{(X,d)\} \cup \overline{\Delta} \parallel \sigma \Rightarrow_{\mathfrak{I}_{ACh}} \Gamma \parallel \{(X,d')\} \cup \overline{\Delta} \parallel \sigma,$$

where $d' > d$. Which implies that $(\kappa+1) - d > (\kappa+1) - d'$. Then, it follows that

$$\overline{h_d}(\{(X,d)\} \cup \overline{\Delta}) >_{mul} \overline{h_d}(\{(X,d')\} \cup \overline{\Delta}).$$

Hence, we have

$$\begin{aligned}\mathbb{M}_{\mathfrak{I}_{ACh}}(\Gamma,\{(X,d)\} \cup \overline{\Delta},\sigma,\kappa) &= (\kappa - a, n_X, |\mathcal{S}ym(\Gamma)|, m, p, |\Gamma|, \overline{h_d}(\{(X,d)\} \cup \overline{\Delta})) \\ &>_{lex} (\kappa - a, n_X, |\mathcal{S}ym(\Gamma)|, m, p, |\Gamma|, \overline{h_d}(\{(X,d')\} \cup \overline{\Delta})) \\ &= \mathbb{M}_{\mathfrak{I}_{ACh}}(\Gamma,\{(X,d')\} \cup \overline{\Delta},\sigma,\kappa)\end{aligned}$$

(AC Unification) In this case, $\Gamma = \Psi \cup \overline{\Gamma}$. The reduction is as

$$\Psi \cup \Gamma \parallel \Delta \parallel \sigma \Rightarrow_{\mathfrak{I}_{ACh}} \bigvee_i (GetEqs(\theta_i) \cup \overline{\Gamma} \parallel \Delta_i \parallel \sigma) = \Gamma' \parallel \Delta' \parallel \sigma'$$

And, we have that

$$\begin{aligned}\mathbb{M}_{\mathfrak{I}_{ACh}}(\Psi \cup \overline{\Gamma}, \Delta, \sigma, \kappa) =& (\kappa - a, n_X, |\mathcal{S}ym(\Gamma)|, m, p, |\Gamma|, \overline{h_d}(\Delta)) \\ &>_{lex} (\kappa - (a+1), n_X, |\mathcal{S}ym(\Gamma')|, m, p, |\Gamma'|, \overline{h_d}(\Delta')) \\ =& \mathbb{M}_{\mathfrak{I}_{ACh}}(\Gamma', \Delta', \sigma, \kappa)\end{aligned}$$

(Splitting) In this case, $\Gamma = \{X \stackrel{?}{=} h(Y), X \stackrel{?}{=} X_1 + \ldots + X_n\} \cup \overline{\Gamma}$. The reduction is as

$$\{X \stackrel{?}{=} h(Y), X \stackrel{?}{=} X_1 + \ldots + X_n\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma$$

$$\Downarrow_{\mathfrak{I}_{ACh}}$$

$$\underbrace{\{X \stackrel{?}{=} h(Y), Y \stackrel{?}{=} V_1 + \ldots + N_n, X_1 \stackrel{?}{=} h(V_1), \ldots, X_n \stackrel{?}{=} h(V_n)\} \cup \overline{\Gamma}}_{\Gamma'} \parallel \Delta' \parallel \sigma$$

□

Notice that, after reduction n_X decreases by 1. Hence, we have that

$$\begin{aligned}\mathbb{M}_{\mathfrak{I}_{ACh}}(\Gamma, \Delta, \sigma, \kappa) &= (\kappa - a, n_X, |\mathcal{S}ym(\Gamma)|, m, p, |\Gamma|, \overline{h_d}(\Delta)) \\ &>_{lex} (\kappa - a, n_X - 1, |\mathcal{S}ym(\Gamma')|, m, p, |\Gamma'|, \overline{h_d}(\Delta')) \\ &= \mathbb{M}_{\mathfrak{I}_{ACh}}(\Gamma', \Delta', \sigma, \kappa)\end{aligned}$$

Corollary 3.1 (Termination). For any set triple $\Gamma \parallel \Delta \parallel \sigma$, there is a disjunction of triples $\bigvee_{i \in I}(\Gamma_i \parallel \Delta_i \parallel \sigma_i)$ such that $\Gamma \parallel \Delta \parallel \sigma \Rightarrow^*_{\mathfrak{I}_{ACh}} \bigvee_{i \in I}(\Gamma_i \parallel \Delta_i \parallel \sigma_i)$ and none of the rules $\mathfrak{I}_{ACh}$ can be applied.

Proof. The measure $\mathbb{M}_{\mathfrak{I}_{ACh}}(\Gamma, \Delta, \sigma, \kappa)$ strictly decreases at each step. So, it follows immediately by the definition of termination. □

3.3 Soundness

This section is to prove that our inference system is truth preserving. Essentially, we want to prove that after applying exhaustively the rules from $\mathfrak{I}_{ACh}$, every solution is indeed a solution to our initial problem.

For the following results, we recall the definition of satisfiabilty of a triple stated in Section 2.1, that is, if $\theta \models \Gamma \parallel \Delta \parallel \sigma$ iff $\theta \models \Gamma$, $\theta \models \sigma$ and $\mathcal{M}ax\mathcal{V}al(\Delta) \leq \kappa$, where κ is the given bound on the h-depth set of variables.

Lemma 3.3. Let $\Gamma \parallel \Delta \parallel \sigma$ and $\Gamma' \parallel \Delta' \parallel \sigma'$ be two set triples such that $\Gamma \parallel \Delta \parallel \sigma \Rightarrow_{\mathfrak{I}_{ACh}} \Gamma' \parallel \Delta' \parallel \sigma'$ via all the rules in $\mathfrak{I}_{ACh}$ except for rule (AC). Let θ be a substitution such that $\theta \vDash \Gamma' \parallel \Delta' \parallel \sigma'$. Then, $\theta \vDash \Gamma \parallel \Delta \parallel \sigma$.

Proof. The proof is by analysing each application of a rule of $\mathfrak{I}_{ACh}$ on $\Gamma \parallel \Delta \parallel \sigma$.

(Trivial) In this case, $\Gamma = \{t \stackrel{?}{=} t\} \cup \overline{\Gamma}$. Applying the rule, we obtain

$$(\textsc{Triv})\frac{\{t \stackrel{?}{=} t\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma}{\overline{\Gamma} \parallel \Delta \parallel \sigma}$$

Let θ be a substitution such that $\theta \vDash \overline{\Gamma} \parallel \Delta \parallel \sigma$. Since $t\theta =_{ACh} t\theta$, we have that $\theta \vDash \{t \stackrel{?}{=} t\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma$ holds trivially.

(Splitting) In this case, $\Gamma = \{X \stackrel{?}{=} h(Y), X \stackrel{?}{=} X_1 + ...X_n\} \cup \overline{\Gamma}$. Applying the rule, we obtain:

$$(\textsc{Split})\frac{\{X \stackrel{?}{=} h(Y), X \stackrel{?}{=} X_1 + ... + X_n\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma}{\{X \stackrel{?}{=} h(Y), Y \stackrel{?}{=} V_1 + ... + V_n, X_1 \stackrel{?}{=} h(V_1), ..., X_n \stackrel{?}{=} h(V_n)\} \cup \overline{\Gamma} \parallel \Delta' \parallel \sigma}$$

Consider a substitution θ such that

$$\theta \vDash \{X \stackrel{?}{=} h(Y), Y \stackrel{?}{=} V_1 + ... + V_n, X_1 \stackrel{?}{=} h(V_1), ..., X_n \stackrel{?}{=} h(V_n)\} \cup \overline{\Gamma} \parallel \Delta' \parallel \sigma.$$

Then,

$$\theta \vDash \{X \stackrel{?}{=} h(Y), Y \stackrel{?}{=} V_1 + ... + V_n, X_1 \stackrel{?}{=} h(V_1), ..., X_n \stackrel{?}{=} h(V_n)\}. \tag{I}$$

By definition of satisfiabilty (Definition 1.15), this means that

(i) $X\theta =_{ACh} h(Y)\theta$

(ii) $Y\theta =_{ACh} (V_1 + ... + V_n)\theta$

(iii) $X_1\theta =_{ACh} h(V_1\theta), ..., X_n\theta = h(V_n\theta)$

We want to prove that $\theta \vDash \{X \stackrel{?}{=} h(Y), X \stackrel{?}{=} X_1 + ... + X_n\} \cup \overline{\Gamma} \parallel \Delta' \parallel \sigma$. For that, we need to prove that $\theta \vDash \{X \stackrel{?}{=} h(Y), X \stackrel{?}{=} X_1 + ... + X_n\} \cup \overline{\Gamma}$, $\theta \vDash \sigma$ and $\mathcal{M}ax\mathcal{V}al(\Delta) \leq \kappa$. By hypothesis (I), we have that $\theta \vDash \sigma$ and, since $\Delta \subseteq \Delta'$, then

$$\mathcal{M}ax\mathcal{V}al(\Delta) \leq \mathcal{M}ax\mathcal{V}al(\Delta') \leq \kappa.$$

It remains to show that $\theta \vDash \{X \stackrel{?}{=} h(Y), X \stackrel{?}{=} X_1 + ... + X_n\} \cup \overline{\Gamma}$. That is, that $\theta \vDash X \stackrel{?}{=} h(Y)$ and $X \stackrel{?}{=} X_1 + ... + X_n$. The first follows by (i). Now we will verify the latter.

Notice that

$$\begin{aligned}(X_1+\ldots+X_n)\theta &=_{ACh} X_1\theta+\ldots+X_n\theta\\ &=_{ACh} h(V_1\theta)+\ldots+h(V_n\theta)\\ &=_{ACh} h(V_1\theta+\ldots+V_n\theta)\\ &=_{ACh} h((V_1+\ldots+V_n)\theta)\end{aligned} \tag{II}$$

From (i) we have $X\theta = h(Y\theta)$, in addition to (II) we obtain

$$X\theta =_{ACh} h((V_1+\ldots V_n)\theta) =_{ACh} (X_1+\ldots+X_n)\theta$$

(Variable Elimination) There are two rules to consider:

- **VE1:** In this case, $\Gamma = \{X \stackrel{?}{=} Y\} \cup \overline{\Gamma}$ with $X \neq Y$. Applying the rule, we obtain:

$$(\mathrm{VE1})\frac{\{X \stackrel{?}{=} Y\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma}{\overline{\Gamma}\{X \mapsto Y\} \parallel \Delta \parallel \sigma\{X \mapsto Y\} \cup \{X \mapsto Y\}}$$

  Let θ be a substitution such that $\theta \vDash \overline{\Gamma}\{X \mapsto Y\} \parallel \Delta \parallel \theta \vDash \sigma\{X \mapsto Y\} \cup \{X \mapsto Y\}$. Now we have to prove that $\theta \vDash \{X \stackrel{?}{=} Y\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma$

  Notice that, by definition, $\theta \vDash \{X \mapsto Y\}$, then $X\theta = Y\theta$ $(*)$, which implies $\theta \vDash \{X \stackrel{?}{=} Y\}$.

  It remains to prove that $\theta \vDash \overline{\Gamma}$.

  Let $s_i \stackrel{?}{=} s_j[X]_p \in \overline{\Gamma}$. Since $\theta \vDash \overline{\Gamma}\{X \mapsto Y\}$, we have $\theta \vDash s_i \stackrel{?}{=} s_j[Y]_p$. Then,

$$s_i\theta =_{ACh} s_j[Y]_p\theta =_{ACh} (s_j\theta)[Y\theta]_p.$$

  Thus, by $(*)$, $s_i\theta =_{ACh} (s_j\theta)[X\theta]_p =_{ACh} s_j[X]_p\theta$. Hence $\theta \vDash \overline{\Gamma}$.

  Now, we are going to prove that $\theta \vDash \sigma$. Let $W \mapsto s[X]_p$ be an assignment in σ. Since $\theta \vDash \sigma\{X \mapsto Y\}$, we have

$$W\theta =_{ACh} (s\theta)[Y\theta]_p.$$

  Then, by $(*)$,

$$W\theta =_{ACh} (s\theta)[X\theta]_p =_{ACh} s[X]_p\theta.$$

  Hence, $\theta \vDash \{W \mapsto s[X]_p\}$. Therefore, $\theta \vDash \sigma$

- **VE2:** Let $\Gamma = \{X \stackrel{?}{=} t\} \cup \overline{\Gamma}$, with $X \notin \mathcal{V}ar(t)$. Applying the rule, we obtain

$$(\text{VE2})\frac{\{X \stackrel{?}{=} t\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma}{\overline{\Gamma}\{X \mapsto t\} \parallel \Delta \parallel \sigma\{X \mapsto t\} \cup \{X \mapsto t\}}$$

Let θ be a substitution such that $\theta \vDash \overline{\Gamma}\{X \mapsto t\} \parallel \Delta \parallel \theta \vDash \sigma\{X \mapsto t\} \cup \{X \mapsto t\}$. Now we have to prove that $\theta \vDash \{X \stackrel{?}{=} t\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma$

Notice that, by definition, $\theta \vDash \{X \mapsto t\}$, then $X\theta = t\theta$ $(**)$, which implies $\theta \vDash \{X \stackrel{?}{=} t\}$.

It remains to prove that $\theta \vDash \overline{\Gamma}$.

Let $s_i \stackrel{?}{=} s_j[X]_p \in \overline{\Gamma}$. Since $\theta \vDash \overline{\Gamma}\{X \mapsto t\}$, we have $\theta \vDash s_i \stackrel{?}{=} s_j[t]_p$. Then,

$$s_i\theta =_{ACh} s_j[t]_p\theta =_{ACh} (s_j\theta)[t\theta]_p.$$

Thus, by $(**)$, $s_i\theta =_{ACh} (s_j\theta)[X\theta]_p =_{ACh} s_j[X]_p\theta$. Hence $\theta \vDash \overline{\Gamma}$.

Now, we are going to prove that $\theta \vDash \sigma$. Let $W \mapsto s[X]_p$ be an assignment in σ. Since $\theta \vDash \sigma\{X \mapsto t\}$, we have

$$W\theta =_{ACh} (s\theta)[t\theta]_p.$$

Then, by $(**)$,

$$W\theta =_{ACh} (s\theta)[X\theta]_p =_{ACh} s[X]_p\theta.$$

Hence, $\theta \vDash \{W \mapsto s[X]_p\}$. Therefore, $\theta \vDash \sigma$.

(Decomposition) Let $\Gamma = \{X \stackrel{?}{=} f(X_1,...,X_n), X \stackrel{?}{=} f(Y_1,...,Y_n)\} \cup \overline{\Gamma}$, with $f \neq +$. Applying the rule, we obtain

$$(\text{DEC})\frac{\{X \stackrel{?}{=} f(X_1,...,X_n), X \stackrel{?}{=} f(Y_1,...,Y_n)\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma}{\{X \stackrel{?}{=} f(X_1,...,X_n), X_1 \stackrel{?}{=} Y_1,...,X_n \stackrel{?}{=} Y_n\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma}$$

Let θ be a substitution such that

$$\theta \vDash \{X \stackrel{?}{=} f(X_1,\dots,X_n), X_1 \stackrel{?}{=} Y_1,\dots,X_n \stackrel{?}{=} Y_n\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma.$$

We want to prove that

$$\theta \vDash \{X \stackrel{?}{=} f(X_1,\dots,X_n), X \stackrel{?}{=} f(Y_1,\dots,Y_n)\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma$$

Since $\theta \vDash X \stackrel{?}{=} f(X_1, ..., X_n)$, it suffices to show that $\theta \vDash X \stackrel{?}{=} f(Y_1, ..., Y_n)$. From $X_i\theta = Y_i\theta$, for all $i = 1, ..., n$, we have

$$\begin{aligned} X\theta &=_{ACh} f(X_1, ...X_n)\theta \\ &=_{ACh} f(X_1\theta, ...X_n\theta) \\ &=_{ACh} f(Y_1\theta, ..., Y_n\theta) \\ &=_{ACh} f(Y_1, ..., Y_n)\theta \end{aligned}$$

(Update h-depth Set) In this case $\Gamma \parallel \Delta \parallel \sigma = \Gamma \parallel \{(X,d)\} \cup \overline{\Delta} \parallel \sigma$. When we apply one of the update rules, we have

$$(\text{UPDATE } h\text{-DEPTH SET})\frac{\Gamma \parallel \{(X,d)\} \cup \overline{\Delta} \parallel \sigma}{\Gamma \parallel \{(X,d')\} \cup \overline{\Delta} \parallel \sigma}$$

where $d' > d$. Let θ be a substitution such that $\theta \vDash \Gamma \parallel \{(X,d')\} \cup \overline{\Delta} \parallel \sigma$. Then $\mathcal{M}ax\mathcal{V}al(\{(X,d')\} \cup \overline{\Delta}) \leq \kappa$, which means that $d' \leq \kappa$. Since $d < d'$, we also have $d < \kappa$. Thus, $\mathcal{M}ax\mathcal{V}al(\{(X,d)\} \cup \overline{\Delta}) \leq \kappa$. Hence $\theta \vDash \Gamma \parallel \Delta \parallel \sigma$.

□

Lemma 3.4. Let $\Gamma \parallel \Delta \parallel \sigma$ and $\bigvee_{i \in I}(\Gamma_i \parallel \Delta_i \parallel \sigma_i)$ be two ACh Unification problems such that $\Gamma \parallel \Delta \parallel \sigma \Rightarrow_{\mathfrak{I}_{ACh}} \bigvee_{i \in I}(\Gamma_i \parallel \Delta_i \parallel \sigma_i)$ via an application (AC). Let θ be a substitution such that $\theta \vDash \Gamma_i \parallel \Delta_i \parallel \sigma_i$ for some $i \in I$. Then $\theta \vDash \Gamma \parallel \Delta \parallel \sigma$.

Proof. The rule for AC unification is

$$(\text{AC})\frac{\Psi \cup \Gamma \parallel \Delta \parallel \sigma}{GetEqs(\theta_1) \cup \Gamma \parallel \Delta \parallel \sigma \vee ... \vee GetEqs(\theta_n) \cup \Gamma \parallel \Delta \parallel \sigma}$$

which each of the θ_i's is the unifier given by an AC unification algorithm. For this work, we are using Stickel's algorithm [Sti75], which the soundness is already proven (cf. [AFSS22]). So, given that $\theta \vDash GetEqs(\theta_1) \cup \Gamma \parallel \Delta \parallel \sigma \vee ... \vee GetEqs(\theta_n) \cup \Gamma \parallel \Delta \parallel \sigma$, we have that $\theta \vDash GetEqs(\theta_i) \cup \Gamma \parallel \Delta \parallel \sigma$ for all $i = 1, ..., n$, which implies that $\theta \vDash \Psi$. □

Combining the two previous Lemmas, we have immediately:

Lemma 3.5. Let $\Gamma \parallel \Delta \parallel \sigma$ and $\Gamma' \parallel \Delta' \parallel \sigma' = \bigvee_{i \in I}(\Gamma_i \parallel \Delta_i \parallel \sigma_i)$ be two ACh Unification problems such that $\Gamma \parallel \Delta \parallel \sigma \Rightarrow_{\mathfrak{I}_{ACh}} \bigvee_{i \in I}(\Gamma_i \parallel \Delta_i \parallel \sigma_i)$. Let θ be a substitution such that $\theta \vDash \Gamma_i \parallel \Delta_i \parallel \sigma_i$, for some $i \in I$. Then $\theta \vDash \Gamma \parallel \Delta \parallel \sigma$.

Theorem 3.2. Let $\Gamma \parallel \Delta \parallel \sigma$ and $\Gamma' \parallel \Delta' \parallel \sigma' = \bigvee_{i \in I}(\Gamma_i \parallel \Delta_i \parallel \sigma_i)$ be two ACh Unification problems such that $\Gamma \parallel \Delta \parallel \sigma \Rightarrow^*_{\mathfrak{I}_{ACh}} \Gamma' \parallel \Delta' \parallel \sigma'$. If θ is a substitution such that $\theta \vDash \Gamma_i \parallel \Delta_i \parallel \sigma_i$ for some $i \in I$, then $\theta \vDash \Gamma \parallel \Delta \parallel \sigma$.

Proof. The proof is by induction on the number n of steps in $\Gamma \parallel \Delta \parallel \sigma \stackrel{n}{\Longrightarrow}_{\mathfrak{I}_{ACh}} \Gamma' \parallel \Delta' \parallel \sigma'$.

(Base Case) $(n = 1)$ The base case follows by Lemma 3.5.

(Inductive Step) Suppose that for $\Gamma \parallel \Delta \parallel \sigma \Rightarrow^n_{\mathfrak{I}_{ACh}} \Gamma' \parallel \Delta' \parallel \sigma'$, follows for n, i.e. if θ is a substitution such that $\theta \vDash \Gamma_i \parallel \Delta_i \parallel \sigma_i$ for $i \in I$, then $\theta \vDash \Gamma \parallel \Delta \parallel \sigma$.

Now, will show that the result hold for derivations with $n+1$ steps.

Let $\Gamma'' \parallel \Delta'' \parallel \sigma''$ be a triple such that $\Gamma \parallel \Delta \parallel \sigma \Rightarrow^n_{\mathfrak{I}_{ACh}} \Gamma' \parallel \Delta' \parallel \sigma' \Rightarrow_{\mathfrak{I}_{ACh}} \Gamma'' \parallel \Delta'' \parallel \sigma''$ and θ be a substitution such that $\theta \vDash \Gamma'' \parallel \Delta'' \parallel \sigma''$. Again, by Lemma 3.5, we have that $\theta \vDash \Gamma' \parallel \Delta' \parallel \sigma'$. Then, by the induction hypothesis, we have that $\theta \vDash \Gamma \parallel \Delta \parallel \sigma$.

□

Corollary 3.2 (Soundness). Let Γ be a set of equations. Suppose that we get $\bigvee_{i \in I}(\Gamma_i \parallel \Delta_i \parallel \sigma_i)$ after exhaustively applying the rules from $\mathfrak{I}_{ACh}$ to $\Gamma \parallel \Delta \parallel \sigma$, that is, $\Gamma \parallel \Delta \parallel \sigma \Rightarrow^*_{\mathfrak{I}_{ACh}} \bigvee_{i \in I}(\Gamma_i \parallel \Delta_i \parallel \sigma_i)$, where for each i, there are no applicable rules to $\Gamma_i \parallel \Delta_i \parallel \sigma_i$. Let $\mathcal{S} = \{\sigma_i \mid \Gamma_i = \emptyset\}$. Then any element of $\mathcal{S}$ is an ACh Unifier of Γ.

Proof. Obviously, for all $\sigma_i \in \mathcal{S}$, we have that $\sigma_i \vDash \Gamma_i \parallel \Delta_i \parallel \sigma_i$. Hence, by Theorem 3.2, $\sigma_i \vDash \Gamma \parallel \Delta \parallel \sigma$. Therefore, σ_i is an ACh unifier of Γ. □

3.4 Completeness

In this section we shall prove that our system never leaves any solution behind. That is, after applying all the rules from $\mathfrak{I}_{ACh}$ exhaustively, for every possible solution to our initial problem, there exists an solution given by our inference system that is more general modulo ACh.

Lemma 3.6. Let $\Gamma \parallel \Delta \parallel \sigma$ be a set triple which is not in solved form (check Definition 2.1), and θ be a substitution such that $\theta \vDash \Gamma \parallel \Delta \parallel \sigma$. Then, there exists an inference

$$\Gamma \parallel \Delta \parallel \sigma \Rightarrow_{\mathfrak{I}_{ACh}} \bigvee_{i \in I}(\Gamma_i \parallel \Delta_i \parallel \sigma_i),$$

an i and θ_0 such that $\mathcal{D}om(\theta_0) \subset \mathcal{V}ar(\Gamma_i) \setminus \mathcal{V}ar(\Gamma)$ and $\theta\theta_0 \vDash \Gamma_i \parallel \Delta_i \parallel \sigma_i$.

Proof. We want to consider all the possible forms of Γ and show that there is an inference rule in $\mathfrak{I}_{ACh}$ that can be applied such that solution θ can be extended.

1. $\{t \stackrel{?}{=} t\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma$.

   In this case, we apply the rule (TRIV), which gives us the inference

   $$\{t \stackrel{?}{=} t\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma \Rightarrow_{\mathfrak{I}_{ACh}} \overline{\Gamma} \parallel \Delta \parallel \sigma.$$

   Notice that $\mathcal{V}ar(\overline{\Gamma}) \setminus \mathcal{V}ar(\{t \stackrel{?}{=} t\} \cup \overline{\Gamma}) = \emptyset$. Then, take $\theta_0 = id$ and we obtain $\theta id \vDash \overline{\Gamma} \parallel \Delta \parallel \sigma$ trivially.

2. $\{t \stackrel{?}{=} X\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma$.

   In this case, we apply the rule (OR), which gives us the inference

   $$\{t \stackrel{?}{=} X\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma \Rightarrow_{\mathfrak{I}_{ACh}} \{X \stackrel{?}{=} t\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma$$

   Notice that $\mathcal{V}ar(\{X \stackrel{?}{=} t\} \cup \overline{\Gamma}) \setminus \mathcal{V}ar(\{t \stackrel{?}{=} X\} \cup \overline{\Gamma}) = \emptyset$. Then, take $\theta_0 = id$ and the result follows.

3. $\{X \stackrel{?}{=} Y\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma$.
   In this case, we apply the rule (VE1), which gives us

   $$\{X \stackrel{?}{=} Y\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma \Rightarrow_{\mathfrak{I}_{ACh}} \overline{\Gamma}\{X \mapsto Y\} \parallel \Delta \parallel \sigma\{X \mapsto Y\} \cup \{X \mapsto Y\}.$$

   Again, we have that $\mathcal{V}ar(\overline{\Gamma}\{X \mapsto Y\}) \setminus \mathcal{V}ar(\{X \stackrel{?}{=} Y\} \cup \overline{\Gamma}) = \emptyset$. Taking $\theta_0 = id$, the result follows.

4. $\{X \stackrel{?}{=} t\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma$.
   In this case, we apply the rule (VE2) which gives us the inference

   $$\{X \stackrel{?}{=} t\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma \Rightarrow_{\mathfrak{I}_{ACh}} \overline{\Gamma}\{X \mapsto t\} \parallel \Delta \parallel \sigma\{X \mapsto t\} \cup \{X \mapsto t\}.$$

   Notice that, $\mathcal{V}ar(\overline{\Gamma}\{X \mapsto t\}) \setminus \mathcal{V}ar(\{X \stackrel{?}{=} t\} \cup \overline{\Gamma}) = \emptyset$. Taking $\theta_0 = id$, the result follows.

5. $\{X \stackrel{?}{=} f(X_1,...,X_n), X \stackrel{?}{=} f(Y_1,...,Y_n)\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma$, where $f \neq +$.
   In this case, we apply the rule (DEC) to get

$$\underbrace{\{X \stackrel{?}{=} f(X_1,...,X_n), X \stackrel{?}{=} f(Y_1,...,Y_n)\} \cup \overline{\Gamma}}_{\Gamma} \parallel \Delta \parallel \sigma$$

$$\Downarrow_{\mathfrak{I}_{ACh}}$$

$$\underbrace{\{X \stackrel{?}{=} f(X_1,...,X_n), X_1 \stackrel{?}{=} Y_1,...,X_n \stackrel{?}{=} Y_n\} \cup \overline{\Gamma}}_{\Gamma'} \parallel \Delta \parallel \sigma$$

   Notice that $\mathcal{V}ar(\Gamma) = \mathcal{V}ar(\Gamma')$. Hence, we can take $\theta_0 = id$ and the result follows.

6. $\{X \stackrel{?}{=} h(Y), X \stackrel{?}{=} X_1 + ... + X_n\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma$.
   For simplicity, suppose $n = 2$ and $\Gamma = \{X \stackrel{?}{=} h(Y), X \stackrel{?}{=} X_1 + X_2\} \cup \overline{\Gamma}$, the general case can be verified similarly.

   In this case, we apply the rule (SPLIT)

$$(\text{SPLIT})\frac{\{X \stackrel{?}{=} h(Y), X \stackrel{?}{=} X_1 + X_2\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma}{\{X \stackrel{?}{=} h(Y), Y \stackrel{?}{=} V_1 + V_2, X_1 \stackrel{?}{=} h(V_1), X_2 \stackrel{?}{=} h(V_2)\} \cup \overline{\Gamma} \parallel \Delta' \parallel \sigma}$$

   Let θ be a substitution such that $\theta \vDash \{X \stackrel{?}{=} h(Y), X \stackrel{?}{=} X_1 + X_2\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma$. Then, we have that

   (i) $X\theta =_{ACh} h(Y\theta)$

   (ii) $X\theta =_{ACh} X_1\theta + X_2\theta$, which implies

   (iii) $h(Y\theta) =_{ACh} X_1\theta + X_2\theta$.

   Since we are in ACh theory, we must have

$$X_1\theta =_{ACh} h(t_1) \text{ and } X_2\theta =_{ACh} h(t_2) \tag{3.1}$$

   for some terms t_1 and t_2. Therefore,

$$h(Y\theta) =_{ACh} h(t_1) + h(t_2) =_{ACh} h(t_1 + t_2) \text{ which implies } Y\theta =_{ACh} t_1 + t_2 \tag{3.2}$$

Applying (SPLIT), we obtain

$$\{X \stackrel{?}{=} h(Y), X \stackrel{?}{=} X_1 + X_2\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma$$
$$\Downarrow_{\mathfrak{I}_{ACh}}$$
$$\{X \stackrel{?}{=} h(Y), Y \stackrel{?}{=} V_1 + V_2, X_1 \stackrel{?}{=} h(V_1), X_2 \stackrel{?}{=} h(V_2)\} \cup \overline{\Gamma} \parallel \Delta' \parallel \sigma$$

Where V_1, V_2 are fresh variables. Define $\theta_0 = \{V_1 \mapsto t_1, V_2 \mapsto t_2\}$. Now, notice that

(i) $X\theta\theta_0 =_{ACh} (X_1\theta + X_2\theta)\theta_0$
$=_{ACh} (h(t_1) + h(t_2))\theta_0$ (by 3.1)
$=_{ACh} h(t_1) + h(t_2)$ ($\mathcal{D}om(\theta_0) = \{V_1, V_2\}$ which are new)

(ii) $h(Y\theta)\theta_0 =_{ACh} (X_1\theta + X_2\theta)\theta_0$
$=_{ACh} (h(t_1) + h(t_2))\theta_0$ (by 3.1)
$=_{ACh} h(t_1) + h(t_2)$

(iii) $Y\theta\theta_0 =_{ACh} (t_1 + t_2)\theta_0$ (by 3.2)
$=_{ACh} t_1 + t_2$
$=_{ACh} V_1\theta_0 + V_2\theta_0$ (by definition of θ_0)
$=_{ACh} (V_1 + V_2)\theta\theta_0$ since $V_1, V_2 \notin \mathcal{D}om(\theta)$

(iv) Notice that $X_1\theta\theta_0 =_{ACh} h(t_1)\theta_0 =_{ACh} h(t_1)$. But, by definition of θ_0, we have

$$X_1\theta\theta_0 =_{ACh} h(V_1\theta_0) =_{ACh} h(V_1)\theta\theta_0.$$

Similarly, $X_2\theta\theta_0 =_{ACh} h(V_2)\theta\theta_0$.

Combining (i) and (ii), we have $X\theta\theta_0 =_{ACh} h(Y)\theta\theta_0$. Hence,

$$\theta\theta_0 \vDash \{X \stackrel{?}{=} h(Y), Y \stackrel{?}{=} V_1 + V_2, X_1 \stackrel{?}{=} h(V_1), X_2 \stackrel{?}{=} h(V_2)\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma$$

7. $\{X \stackrel{?}{=} X_1 + \ldots + X_n, X \stackrel{?}{=} Y_1 + \ldots + Y_m\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma$.
In this case, we apply the (AC) rule to obtain the inference

$$\{X \stackrel{?}{=} X_1 + \ldots + X_n, X \stackrel{?}{=} Y_1 + \ldots + Y_m\} \cup \overline{\Gamma} \parallel \Delta \parallel \sigma$$
$$\Downarrow_{\mathfrak{I}_{ACh}}$$
$$\bigvee_{i=1}^{n} (\underbrace{GetEqs(\theta_i) \cup \overline{\Gamma}}_{\Gamma_i} \parallel \Delta_i \parallel \sigma)$$

As for the existence of the substitution θ_0 such that $\mathcal{D}om(\theta_0) \subset \mathcal{V}ar(\Gamma_i) \setminus \mathcal{V}ar(\Gamma)$ and $\theta\theta_0 \vDash \Gamma_i \parallel \Delta_i \parallel \sigma$, the proof can be found in [AFSS22].

$\square$

Remark. For the next theorem, we removed the affirmation in [EL20] that $\mathcal{D}om(\theta_0) = \mathcal{V}ar(\Gamma_i) \setminus \mathcal{V}ar(\Gamma)$. Such affirmation is true for one step, but, for more steps, the domain of θ_0 depends on all the previous steps taken before.

Theorem 3.3. Let $\Gamma \parallel \Delta \parallel \sigma$ be a triple which is not in solved form, and θ be a substitution such that $\theta \vDash \Gamma \parallel \Delta \parallel \sigma$. Then, there exists a sequence of inferences

$$\Gamma \parallel \Delta \parallel \sigma \Rightarrow^{+}_{\mathfrak{I}_{ACh}} \bigvee_{i \in I} (\Gamma_i \parallel \Delta_i \parallel \sigma_i),$$

and an i and θ_0 such that $\theta\theta_0 \vDash \Gamma_i \parallel \Delta_i \parallel \sigma_i$.

Proof. The proof is by construction. Let $\Gamma \parallel \Delta \parallel \sigma$ be a triple which is not in solved form and θ be a substitution such that $\theta \vDash \Gamma \parallel \Delta \parallel \sigma$. By Lemma 3.6, there exists an inference

$$\Gamma \parallel \Delta \parallel \sigma \Rightarrow_{\mathfrak{I}_{ACh}} \bigvee_{j} (\Gamma_j \parallel \Delta_j \parallel \sigma_j)$$

and j and θ_1 such that $\theta\theta_1 \vDash \Gamma_j \parallel \Delta_j \parallel \sigma_j)$ Assuming that $\Gamma_j \parallel \Delta_j \parallel \sigma_j$ is not in solved form, we have, again by Lemma 3.6, an inference

$$\Gamma_j \parallel \Delta_j \parallel \sigma_j \Rightarrow_{\mathfrak{I}_{ACh}} \bigvee_{k} (\Gamma_k \parallel \Delta_k \parallel \sigma_k)$$

an k and θ_2 such that $\theta\theta_1\theta_2 \vDash \Gamma_k \parallel \Delta_k \parallel \sigma_k$ By termination (Corollary 3.1), we repeat the same procedure a finite number of times, say m, to obtain

$$\theta\theta_1\theta_2 \dots \theta_m \vDash \Gamma_i \parallel \Delta_i \parallel \sigma_i$$

for some index i. Define $\theta_0 := \theta_1 \dots \theta_m$. Notice that we have the following sequence

$$\Gamma \parallel \Delta \parallel \sigma \Rightarrow_{\mathfrak{I}_{ACh}} \Gamma_j \parallel \Delta_j \parallel \sigma_j \Rightarrow^{+}_{\mathfrak{I}_{ACh}} \Gamma_i \parallel \Delta_i \parallel \sigma_i$$

Collecting all the branches, we have.

$$\Gamma \parallel \Delta \parallel \sigma \Rightarrow_{\mathfrak{I}_{ACh}} \bigvee_{i} (\Gamma_i \parallel \Delta_i \parallel \sigma_i).$$

$\square$

As a consequence, differently from [EL20], we obtain that set of computed solutions S which contains substitutions that are more general than the extension $\theta\theta_0$ of any solution θ of a problem $\Gamma \parallel \Delta \parallel \sigma$. Here, $\theta_0 = \theta_1 \dots \theta_m$, where each θ_i is computed in the proof of Theorem 3.3.

Corollary 3.3 (Completeness). Let $\Gamma \parallel \Delta \parallel \sigma$ be a triple. Suppose that

$$\Gamma \parallel \Delta \parallel \sigma \Rightarrow^*_{\mathfrak{I}_{ACh}} \bigvee_{i \in I} (\Gamma_i \parallel \Delta_i \parallel \sigma_i),$$

where, for each i, there are no rules left to be applied. Let $S = \{\sigma_i \mid \Gamma_i = \emptyset\}$. Then, for each ACh Unifier θ of Γ, there exists a $\sigma_j \in S$, and θ_0 such that $\sigma_j \lesssim^{\mathcal{V}ar(\Gamma)}_{ACh} \theta$.

Proof. Let θ be an ACh Unifier of Γ, then, by Theorem 3.3 we have that there exist inferences such that

$$\Gamma \parallel \Delta \parallel \sigma \Rightarrow^*_{\mathfrak{I}_{ACh}} \bigvee_{i \in I} (\Gamma_i \parallel \Delta_i \parallel \sigma_i),$$

and there exists $j \in I$, and θ_0 such that $\theta\theta_0 \vDash \Gamma_j \parallel \Delta_j \parallel \sigma_j$ and $\Gamma_j = \emptyset$.

We want to prove that $\sigma_j \lesssim^{\mathcal{V}ar(\Gamma)}_{ACh} \theta$, that is, there exists ρ such that $X\theta = X\theta\rho$, for all $X \in \mathcal{V}ar(\Gamma)$ (Definition 1.16). For any $X \in \mathcal{D}om(\sigma_j) \cap \mathcal{V}ar(\Gamma)$, consider that $X \mapsto t_X \in \sigma_j$, that is, $X\sigma_j = t_X (*)$. Since $\theta\theta_0 \vDash \sigma_j$, by Definition 1.15, we have

$$X\theta\theta_0 =_{ACh} t_X\theta\theta_0.$$

By $(*)$, we obtain

$$X\theta\theta_0 =_{ACh} X\sigma_j\theta\theta_0.$$

But notice that $\mathcal{D}om(\theta_0)$ consists on the new variables introduced by $(\textsc{Split})$ or (AC) and, since they are brand new, θ_0 does not affect the variables occurring in $X\theta$, in other words, $\mathcal{D}om(\theta_0) \cap Im(\theta) = \emptyset$. Therefore,

$$X\theta = X\theta\theta_0 =_{ACh} X\sigma_j \underbrace{\theta\theta_0}_{\rho}$$

which proves that $\sigma_j \lesssim^{\mathcal{V}ar(\Gamma)}_{ACh} \theta$.

□

Therefore, we have verified that the termination and correctness results follow.

Conclusion

In this dissertation, we studied the algorithm for bounded ACh-unification proposed by Lynch and Eeralla [EL20]. The general problem of ACh-unification was known to be undecidable, as proven by Narendran [Nar96]. However, Eeralla and Lynch presented a method to solve a decidable variant of this difficult unification problem by setting a bound κ to the number of nested occurrences of a homomorphic operator. While our primary objective was to verify the correctness of the algorithm, we discovered inaccuracies in some of the proofs and definitions that required more precision, fixes and polishing during the verification process. The table below summarizes our contribution:

<table>
<tr><th>Chapter 3</th><th>Results</th><th>Contribution</th></tr>
<tr><td rowspan="3">Termination</td><td>Lemma 3.2</td><td>Definition 3.2 for AC-solved variables created in order to prove the lemma.</td></tr>
<tr><td>Definition 3.3</td><td>Positions of m and p swapped and notation of n_X fixed.</td></tr>
<tr><td>Theorem 3.1</td><td>Detailed and expanded proof.</td></tr>
<tr><td rowspan="2">Soundness</td><td>Lemma 3.3</td><td>Detailed and expanded proof.</td></tr>
<tr><td>Theorem 3.2</td><td>Provided a complete proof; it was omitted in the paper.</td></tr>
<tr><td rowspan="3">Completeness</td><td>Lemma 3.6</td><td>Detailed and expanded proof.</td></tr>
<tr><td>Theorem 3.3</td><td>Provided a complete proof using a finite construction from the Lemma 3.6; it was omitted in the paper.</td></tr>
<tr><td>Corollary 3.3</td><td>Provided a complete proof; it was omitted in the paper.</td></tr>
</table>

After conducting this study, we have gained a better understanding of the problem and the method used to solve it through approximations. Moving forward, it would be worthwhile to explore what are the implications if we have multiple AC function symbols on our signature. Is it necessary to have a different homomorphism acting on each AC function symbol in this case? If so, is it possible solve a problem with multiple ACh identities?

It is also interesting to investigate whether this method is applicable to other equational theories or if it functions effectively with a combination of equational theories that possess the *finite variant property* [EEMR19]. This property allows for the reduction of an "E-unification problem" to syntactic unification by computing a finite number of variants of the unification problem.

Finally, it would be worth exploring the potential of using bounds in combining matching algorithms. One method, called the *hierarchical combination* [EMR22], creates combined matching algorithms for the union of regular theories that share a common constructor sub-theory.

References

[AFSS22] Mauricio Ayala-Rincón, Maribel Fernández, Gabriel Ferreira Silva, and Daniele Nantes Sobrinho. A Certified Algorithm for AC-Unification. In Amy P. Felty, editor, *7th FSCD 2022, August 2-5, 2022, Haifa, Israel*, volume 228 of *LIPIcs*, pages 8:1–8:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

[BN98] Franz Baader and Tobias Nipkow. *Term rewriting and all that*. Cambridge University Press, 1998.

[BS01] Franz Baader and Wayne Snyder. Unification Theory. In John Alan Robinson and Andrei Voronkov, editors, *Handbook of Automated Reasoning (in 2 volumes)*, pages 445–532. Elsevier and MIT Press, 2001.

[EEMR19] Ajay Kumar Eeralla, Serdar Erbatur, Andrew M. Marshall, and Christophe Ringeissen. Rule-Based Unification in Combined Theories and the Finite Variant Property. In Carlos Martín-Vide, Alexander Okhotin, and Dana Shapira, editors, *Language and Automata Theory and Applications - 13th International Conference, LATA 2019, St. Petersburg, Russia, March 26-29, 2019, Proceedings*, volume 11417 of *Lecture Notes in Computer Science*, pages 356–367. Springer, 2019.

[EL20] Ajay Kumar Eeralla and Christopher Lynch. Bounded ACh unification. *Math. Struct. Comput. Sci.*, 30(6):664–682, 2020.

[EMR22] Serdar Erbatur, Andrew M. Marshall, and Christophe Ringeissen. Combined Hierarchical Matching: the Regular Case. In Amy P. Felty, editor, *7th International Conference on Formal Structures for Computation and Deduction, FSCD 2022, August 2-5, 2022, Haifa, Israel*, volume 228 of *LIPIcs*, pages 6:1–6:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

[Fag87] François Fages. Associative-Commutative Unification. *J. Symb. Comput.*, 3(3):257–275, 1987.

[Hil00] David Hilbert. Mathematische probleme. *Nachrichten von der Koniglichen Gesellschaft der Wissenschaften zu Gottingen*, 1900.

[KB83] Donald E. Knuth and Peter B. Bendix. Simple word problems in universal algebras. *Automation of Reasoning: 2: Classical Papers on Computational Logic 1967–1970*, pages 342–376, 1983.

[Mat70] Yuri V. Matiyasevic. Enumerable sets are diophantine. In *Soviet Math. Dokl.*, volume 11, pages 354–358, 1970.

[Nar96] Paliath Narendran. Solving Linear Equations over Polynomial Semirings. In *Proceedings, 11th Annual IEEE Symposium on Logic in Computer Science, New Brunswick, New Jersey, USA, July 27-30, 1996*, pages 466–472. IEEE Computer Society, 1996.

[Rob65] John Alan Robinson. A Machine-Oriented Logic Based on the Resolution Principle. *J. ACM*, 12(1):23–41, 1965.

[Sti75] Mark E. Stickel. A Complete Unification Algorithm for Associative-Commutative Functions. In *Advance Papers of the Fourth International Joint Conference on Artificial Intelligence, Tbilisi, Georgia, USSR, September 3-8, 1975*, pages 71–76, 1975.