One is all you need: Second-order Unification without First-order Variables *

David M. Cerna[1] and Julian Parsert[23]

[1] Czech Academy of Sciences, Prague, Czechia
dcerna@cs.cas.cz
[2] University of Oxford, United Kingdom
[3] University of Innsbruck, Austria
julian.parsert@gmail.com

Abstract

We consider the fragment of Second-Order unification, referred to as *Second-Order Ground Unification (SOGU)*, with the following properties: (i) only one second-order variable allowed, (ii) first-order variables do not occur. We show that Hilbert's 10^{th} problem is reducible to a *necessary condition* for SOGU unifiability if the signature contains a binary function symbol and two constants, thus proving undecidability. This generalizes known undecidability results, as either first-order variable occurrences or multiple second-order variables were required for the reductions. Furthermore, we show that adding the following restriction: (i) the second-order variable has arity 1, (ii) the signature is finite, and (iii) the problem has *bounded congruence*, results in a decidable fragment. The latter fragment is related to *bounded second-order unification* in the sense that the number of bound variable occurrences is a function of the problem structure. We conclude with a discussion concerning the removal of the *bounded congruence* restriction.

1 Introduction

In general, unification is the process of equating symbolic expressions. Second-order unification concerns symbolic expressions containing function variables, i.e. variables which take expressions as arguments. Such processes are fundamental to mathematics and computer science, and are central to formal methods, verification, automated reasoning, interactive theorem proving, and a variety of other areas. In addition, methods of formal verification based on *satisfiability modulo theories (SMT)* exploits various forms of unification within the underlying theories and their implementations.

Recent investigations have made efforts to increase the expressive power of SMT by adding higher-order features [4]. In some cases, synthesis techniques are required to find SMT models [17]. Furthermore, various function synthesis problems can be addressed using Syntax-Guided Synthesis (SyGuS) [3]. Observe that in many cases SyGuS can be considered as a form of equational second-order unification where (i) only one second-order variable allowed, (ii) first-order variables do not occur. Often, enumerative SyGuS solvers use Counterexample Guided Inductive Synthesis [1] to speed up the synthesis procedure by leveraging ground instances of the problem. In the synthesis domain of Programming-By-Example (PBE) the goal is to find functions that satisfy a given set of concrete input-output examples where no variables (other than the synthesis target) are present [8]. All of these developments combined motivate the investigation of second-order unification, including *ground* cases in the absence of first-order variables.

*Funded by Czech Science Foundation Grant No. 22-06414L and Cost Action CA20111 EuroProofNet and the Austrian Science Fund (FWF) project AUTOSARD (36623).

Already in the 1970s, Huet and Lucchesi proved the undecidability of higher-order logic [9, 15]. This result was generalized by Goldfarb [7] who proved that second-order unification is undecidable. Both results consider the general problem and thus led to the search for decidable fragments and honing of the undecidability results (See [12] for a comprehensive survey). Known decidable classes include *Monadic Second-order* [5, 13], *Second-order Linear* [11], *Bounded Second-order* [18], and *Context Unification* [10]. Undecidability of Second-order unification has been shown for the following fragments;

- Two second-order variables *without* first-order variables, [14]
- one second-order variable *with at least eight* first-order variables [6], and
- One second-order variable with only ground arguments and first-order variables [14]

Interestingly, Levy [12] notes that the number of second-order variables only plays a minor role in the decidability as Levy and Veanes [14] provide a reduction translating arbitrary second-order equations to equations containing only one second-order variable and *additional first-order variables*. These results immediately lead to the following question:

> How important are first-order variables for undecidability of second-order unification?

To address this question, we investigate *second-order ground unification* where only 1 second-order variable (arbitrary occurrences) is allowed and *no first-order variables* occur. To this end, we introduce two functions, the n-counter and the n-multiplier, that allow us to reason about the multiplicity of a given constant (monadic function symbol) post-substitution and of a non-variable symbol introduced by the substitution, respectively. These functions allow us to describe properties of the unification problem number-theoretically. As a result, we can reduce finding solutions to *Diophantine equations* to a *unification condition* involving the structure of the substitution and n-counter, and thus prove (un)decidability results. In particular, our contributions are:

- The n-counter and the n-multiplier and prove essential properties of both.
- We prove the undecidability of second-order unification with *one* function variable and *no* first-order variables.
- We describe a decidable fragment of Second-order ground unification.
- We discuss possible generalizations of this decidable fragment.
- We conjecture undecidability even when the function variable has arity 1.

Observe that through our encoding any decidable class of Diophantine equations provides a decidable fragment of the second-order unification problem presented in this work. Furthermore, our reduction uses a simple encoding which guarantees the equation presented in Lemma 3 directly reduces to $0 = p(\overline{x_n})$ where $p(\overline{x_n})$ denotes a polynomial with integer coefficients over the variables $x_1, \cdots, x_n$. There are likely more intricate encodings which map polynomials to more interesting unification problems. This may be especially relevant when reducing decidable classes of Diophantine equations to unification problems; a topic we plan to consider in future work.

2 preliminaries

We consider a finite *signature* $\Sigma = \{f_1, \cdots, f_n, c_1, \cdots, c_m\}$ where $n, m \geq 1$, for $1 \leq i \leq n$, the arity of f_i is denoted $arity(f_i) \geq 1$, and for all $1 \leq j \leq m$, the arity of c_j is denoted $arity(c_j) = 0$ (*constants*). Furthermore, let $\Sigma^{\leq 1} \subseteq \Sigma$ be the set of *base symbols* defined as $\Sigma^{\leq 1} = \{f \mid f \in \Sigma \wedge arity(f) \leq 1\}$.

By $\mathcal{V}$ we denote a countably infinite set of *variables*. Furthermore, let $\mathcal{V}_i, \mathcal{V}_f \subset \mathcal{V}$ such that $\mathcal{V}_i \cap \mathcal{V}_f = \emptyset$. We refer to members of $\mathcal{V}_i$ as *individual variables*, denoted by x,y,z, $\cdots$ and members of $\mathcal{V}_f$ as *function variables*, denoted by F,G,H, $\cdots$. Members of $\mathcal{V}_f$ have an arity ≥ 1 which we denote by $arity(F)$ where $F \in \mathcal{V}_f$. By $\mathcal{V}_f^n$, where $n \geq 1$, we denote the set of all function variables with arity n. We will use h to denote a symbol in $\mathcal{V} \cup \Sigma$ when doing so would not cause confusion.

We refer to members of the term algebra $\mathcal{T}(\Sigma, \mathcal{V})$, as *terms*. By $\mathcal{V}_i(t)$ and $\mathcal{V}_f(t)$ ($\mathcal{V}_f^n(t)$ for $n \geq 1$) we denote the set of individual variables and function variables (with arity $= n$) occurring in t, respectively. We refer to a term t as n-second-order ground (n-SOG) if $\mathcal{V}_i(t) = \emptyset$, $\mathcal{V}_f(t) \neq \emptyset$ with $\mathcal{V}_f(t) \subset \mathcal{V}_f^n$, first-order if $\mathcal{V}_f(t) = \emptyset$, and *ground* if t is first-order and $\mathcal{V}_i(t) = \emptyset$. The sets of n-SOG, first-order, and ground terms are denoted $\mathcal{T}_{SO}^n$, $\mathcal{T}_{FO}$, and $\mathcal{T}_G$, respectively. When possible, without causing confusion, we will abbreviate a sequence of terms $t_1, \cdots, t_n$ by $\overline{t_n}$ where $n \geq 0$.

The set of *positions* of a term t, denoted by $pos(t)$, is a set of strings of positive integers, defined as $pos(\mathsf{h}(t_1, \ldots, t_n)) = \{\epsilon\} \cup \bigcup_{i=1}^n \{i.p \mid p \in pos(t_i)\}$, $t_1, \ldots, t_n$ are terms, and ϵ denotes the empty string. For example, the term at position 1.1.2 of $g(f(x, a))$ is a. Given a term t and $p \in pos(t)$, then $t|_p$ denotes the subterm of t at position p. Given a term t and $p, q \in pos(t)$, we write $p \sqsubseteq q$ if $q = p.q'$ and $p \sqsubset q$ if $p \sqsubseteq q$ and $p \neq q$. The *set of subterms of a term* t is defined as $sub(t) = \{t|_p \mid p \in pos(t)\}$. The *head* of a term t is defined as $head(\mathsf{h}(t_1, \ldots, t_n)) = \mathsf{h}$, for $n \geq 0$. The number of occurrences of a term s in a term t is defined as $occ(s, t) = |\{p \mid s = t|_p \wedge p \in pos(t)\}|$. The number of occurrences of a symbol h in a term t is defined as $occ_\Sigma(\mathsf{h}, t) = |\{p \mid \mathsf{h} = head(t|_p) \wedge p \in pos(t)\}|$.

A *n-second-order ground (n-SOG) unification equation* has the form $u \stackrel{?}{=}_F v$ where u and v are n-SOG terms and $F \in \mathcal{V}_f^n$ such that $\mathcal{V}_f(u) = \{F\}$ and $\mathcal{V}_f(v) = \{F\}$. A *n-second-order ground unification problem* (n-SOGU problem) is a pair $(\mathcal{U}, F)$ where $\mathcal{U}$ is a set of n-SOG unification equations and $F \in \mathcal{V}_f^n$ such that for all $u \stackrel{?}{=}_G v \in \mathcal{U}$, $G = F$. Recall from the definition of n-SOG that $\mathcal{V}_i(u) = \mathcal{V}_i(v) = \emptyset$.

A *substitution* is set of bindings of the form $\{F_1 \mapsto \lambda\overline{y_{l_1}}.t_1, \cdots F_k \mapsto \lambda\overline{y_{l_k}}.t_k, x_1 \mapsto s_1, \cdots, x_w \mapsto s_w\}$ where $k, w \geq 0$, for all $1 \leq i \leq k$, t_i is first-order and $\mathcal{V}_i(t_i) \subseteq \{y_1, \cdots, y_{l_i}\}$, $arity(F_i) = l_i$, and for all $1 \leq i \leq w$, s_i is ground. Given a substitution σ, $dom_f(\sigma) = \{F \mid F \mapsto \lambda\overline{x_n}.t \in \sigma \wedge F \in \mathcal{V}_f^n\}$ and $dom_i(\sigma) = \{x \mid x \mapsto t \in \Sigma \wedge x \in \mathcal{V}_i\}$. We refer to a substitution σ as second-order when $dom_i(\sigma) = \emptyset$ and first-order when $dom_f(\sigma) = \emptyset$. We use postfix notation for substitution applications, writing $t\sigma$ instead of $\sigma(t)$. Substitutions are denoted by lowercase Greek letters. As usual, the application $t\sigma$ affects only the free variable occurrences of t whose free variable is found in $dom_i(\sigma)$ and $dom_f(\sigma)$. A substitution σ is a *unifier* of an n-SOGU problem $(\mathcal{U}, F)$, if $dom_f(\sigma) = \{F\}$, $dom_i(\sigma) = \emptyset$, and for all $u \stackrel{?}{=}_F v \in \mathcal{U}$, $u\sigma = v\sigma$.

We will use the following theorem due to Matiyasevich, Robinson, Davis, and Putnam, in later sections.

Theorem 1 (Hilbert's 10^{th} problem or Matiyasevich–Robinson–Davis–Putnam theorem [16])**.** *Given a polynomial $p(\overline{x})$ with integer coefficients, finding integer solutions to $p(\overline{x}) = 0$ is undecidable.*

3 n-Multipliers and n-Counters

In this section, we define and discuss the n-multiplier and n-counter functions, which allow us to encode number-theoretic problems in second-order unification. These functions are motivated by the following simple observation about n-SOGU.

Lemma 1. *Let $(\mathcal{U}, F)$ be a unifiable n-SOGU problem, and σ a unifier of $(\mathcal{U}, F)$. Then for all $c \in \Sigma^{\leq 1}$ and $u \stackrel{?}{=}_F v \in \mathcal{U}$, $occ_\Sigma(c, u\sigma) = occ_\Sigma(c, v\sigma)$.*

Proof. If there exists $u \stackrel{?}{=}_F v \in \mathcal{U}$ such that $occ_\Sigma(c, u\sigma) \neq occ_\Sigma(c, v\sigma)$, then there exists a position $p \in pos(u\sigma) \cap pos(u\sigma)$ such that $u\sigma|_p \neq v\sigma|_p$, i.e. σ is not a unifier. ■

With this observation, we now seek to relate the number of occurrences of a symbol in a term t and substitution σ with the number of occurrences of the same symbol in the term $t\sigma$. To this end, we define the *multiplier* function.

Definition 1 (n-Multiplier)**.** *Let t be a n-SOG term such that $\mathcal{V}_f(t) \subseteq \{F\}$ and $F \in \mathcal{V}_f^n$ and $h_1, \cdots, h_n \geq 0$. Then we define $mul(F, \overline{h_n}, t)$ recursively as follows:*

- *if $t = f(t_1, \cdots, t_l)$, then $mul(F, \overline{h_n}, t) = \sum_{j=1}^{l} mul(F, \overline{h_n}, t_j)$*
- *if $t = F(\overline{t_n})$, then $mul(F, \overline{h_n}, t) = 1 + \sum_{i=1}^{n} h_i \cdot mul(F, \overline{h_n}, t_i)$*

Furthermore, let $(\mathcal{U}, F)$ be an n-SOGU problem then,

$$mul_l(F, \overline{h_n}, \mathcal{U}) = \sum_{u \stackrel{?}{=}_F v \in \mathcal{U}} mul(F, \overline{h_n}, u) \qquad mul_r(F, \overline{h_n}, \mathcal{U}) = \sum_{u \stackrel{?}{=}_F v \in \mathcal{U}} mul(F, \overline{h_n}, v).$$

The *n-multiplier* captures the following property of a term: let t be a n-SOG term such that $\mathcal{V}_f(t) \subseteq \{F\}$, $f \in \Sigma$, and $\sigma = \{F \mapsto \lambda\overline{x_n}.s\}$ a substitution where $occ_\Sigma(f, s) \geq 0$, $\mathcal{V}_i(s) \subseteq \{\overline{x_n}\}$, and for all $1 \leq i \leq n$, $occ(x_i, s) = h_i$. Then $occ_\Sigma(f, t\sigma) \geq occ_\Sigma(f, s) \cdot mul(F, \overline{h_n}, t)$ where the $\overline{h_n}$ capture the duplication of the arguments to F. The following presents this idea using a concrete example.

Example 1. Consider the term $t = g(F(g(a, F(s(a)))), g(F(a), F(F(F(b)))))$. Then the n-multiplier of t is

$$mul(F, h, t) = mul(F, h, F(g(a, F(s(a))))) + mul(F, h, g(F(a), F(F(F(b))))) =$$
$$(1 + h) + (1 + (1 + h \cdot (1 + h))) = 3 + 2 \cdot h + h^2.$$

Thus, when $h = 2$ we get $mul(F, h, t) = 11$. Observe $occ_\Sigma(g', t\{F \mapsto \lambda x.g'(x, x)\}) = 11$. □

Next, we introduce the n-counter function. Informally, given an n-SOG term t such that $\mathcal{V}_f(t) \subseteq \{F\}$, a symbol $c \in \Sigma^{\leq 1}$, and a substitution σ with $dom_f(\sigma) = \{F\}$, the n-counter captures number of occurrences of c in $t\sigma$.

Definition 2 (n-Counter)**.** *Let $c \in \Sigma^{\leq 1}$, t be a n-SOG term such that $\mathcal{V}_f(t) \subseteq \{F\}$ and $F \in \mathcal{V}_f^n$, and $h_1, \cdots, h_n \geq 0$. Then we define $cnt(F, \overline{h_n}, c, t)$ recursively as follows:*

- *if $t = f(\overline{t_l})$ and $f \neq c$, then $cnt(F, \overline{h_n}, c, t) = \sum_{j=1}^{l} cnt(F, \overline{h_n}, c, t_j)$.*
- *if $t = c(t)$, then $cnt(F, \overline{h_n}, c, c(t)) = 1 + cnt(F, \overline{h_n}, c, t)$*

- *if* $t = c$*, then* $cnt(F, \overline{h_n}, c, c) = 1$
- *if* $t = F(\overline{t_n})$*, then* $cnt(F, \overline{h_n}, c, t) = \sum_{i=1}^{n} h_i \cdot cnt(F, \overline{h_n}, c, t_i)$

Furthermore, let $(\mathcal{U}, F)$ *be a n-SOGU problem them,*

$$cnt_l(F, \overline{h_n}, c, \mathcal{U}) = \sum_{u \stackrel{?}{=}_F v \in \mathcal{U}} cnt(F, \overline{h_n}, c, u) \qquad cnt_r(F, \overline{h_n}, c, \mathcal{U}) = \sum_{u \stackrel{?}{=}_F v \in \mathcal{U}} cnt(F, \overline{h_n}, c, v).$$

The n-counter captures how many occurrences of a given constant or monadic function symbol will occur in a term $t\sigma$ where $\mathcal{V}_f(t) \subseteq \{F\}$, $\sigma = \{F \mapsto \lambda \overline{x_n}.s\}$, $\mathcal{V}_i(s) \subseteq \{\overline{x_n}\}$, and for all $1 \leq i \leq n$, $occ(x_i, s) = h_i$ A concrete instance is presented in Example 2.

Example 2. Consider the term $t = g(g(a, a), g(F(g(a, F(g(a, a)))), g(F(a), F(F(F(b))))))$. The counter of t is $cnt(F, h, a, t) = cnt(F, h, a, g(a, a)) + cnt(F, h, a, F(g(a, F(g(a, a))))) + cnt(F, h, a, g(F(a), F(F(F(b))))) = 2 + (h + 2 \cdot h^2) + h = 2 + 2 \cdot h + 2 \cdot h^2$. Thus, when $h = 2$ we get $cnt(F, h, a, t) = 14$. Observe $occ_\Sigma(a, t\{F \mapsto \lambda x.g(x, x)\}) = 14$. □

The n-multiplier and n-counter functions differ in the following key aspects: the n-multiplier counts occurrences of a symbol occurring once in a given substitution with bound variable occurrences corresponding to $\overline{h_n}$, and the n-counter counts occurrences of a given symbol after applying the given substitution to a term.

Now we describe the relationship between the n-multiplier, n-counter, and the total occurrences of a given symbol.

Lemma 2. *Let* $c \in \Sigma^{\leq 1}$*,* t *be a n-SOG term such that* $\mathcal{V}_f(t) \subseteq \{F\}$*,* $h_1, \cdots, h_n \geq 0$*, and* $\sigma = \{F \mapsto \lambda \overline{x_n}.s\}$ *a substitution such that* $\mathcal{V}_i(s) \subseteq \{\overline{x_n}\}$ *and for all* $1 \leq i \leq n$ $occ(x_i, s) = h_i$*. Then* $occ(c, t\sigma) = occ(c, s) \cdot mul(F, \overline{h_n}, t) + cnt(F, \overline{h_n}, c, t)$*.*

Proof. We prove the lemma by induction on $dep(t)$. When $dep(t) = 1$, then t is a constant, $t = t\sigma$, and $mul(F, \overline{h_n}, t) = 0$. If $t = c$ then $cnt(F, \overline{h_n}, c, t) = 1$, otherwise $cnt(F, \overline{h_n}, c, t) = 0$. In either case we get $occ(c, t\sigma) = 0 + cnt(F, \overline{h_n}, c, t)$.

Now for the induction hypothesis we assume the lemma holds for all terms t such that $dep(t) < w + 1$ and prove the lemma for a term t' such that $dep(t) = w + 1$. Consider the following Three cases:

- $t = f(t_1, \cdots, t_k)$ for $f \in \Sigma$ and $f \neq c$. We know by the induction hypothesis that for $1 \leq i \leq k$, $occ(c, t_i\sigma) = occ(c, s) \cdot mul(F, \overline{h_n}, t_i) + cnt(F, \overline{h_n}, c, t_i)$. Thus,

$$occ(c, t\sigma) = \sum_{i=1}^{k} occ(c, t_i\sigma) = occ(c, s) \cdot \sum_{i=1}^{k} mul(F, \overline{h_n}, t_i) + \sum_{i=1}^{k} cnt(F, \overline{h_n}, c, t_i),$$

  and by the definition of n-multiplier and n-counter, $mul(F, \overline{h_n}, t) = \sum_{i=1}^{k} mul(F, \overline{h_n}, t_i)$, and $cnt(F, \overline{h_n}, c, t) = \sum_{i=1}^{k} cnt(F, \overline{h_n}, c, t_i)$. Thus, $occ(c, t\sigma) = occ(c, s) \cdot mul(F, \overline{h_n}, t) + cnt(F, \overline{h_n}, c, t)$.

- $t = c(t_1)$. Same as previous case except $cnt(F, \overline{h_n}, c, t) = 1 + cnt(F, \overline{h_n}, c, t_1)$
- $t = F(r_1, \cdots, r_n)$. By the induction hypothesis, we have that for all $1 \leq i \leq n$,

$$occ(c, r_i\sigma) = occ(c, s) \cdot mul(F, \overline{h_n}, r_i) + cnt(F, \overline{h_n}, c, r_i).$$

With this assumption we can derive the following equality and conclude the proof.

$$\begin{gathered}
occ(c,t\sigma) = occ(c,s) + \sum_{i=1}^{n} h_i \cdot occ(c,r_i\sigma) = \\
occ(c,s) + occ(c,s) \cdot \left(\sum_{i=1}^{n} h_i \cdot mul(F,\overline{h_n},r_i)\right) + \left(\sum_{i=1}^{n} h_i \cdot cnt(F,\overline{h_n},c,r_i)\right) = \\
occ(c,s) \cdot \left(1 + \sum_{i=1}^{n} h_i \cdot mul(F,\overline{h_n},r_i)\right) + \left(\sum_{i=1}^{n} h_i \cdot cnt(F,\overline{h_n},c,r_i)\right) = \\
occ(c,s) \cdot mul(F,\overline{h_n},F(\overline{r_n})) + cnt(F,\overline{h_n},c,F(\overline{r_n})) = \\
occ(c,s) \cdot mul(F,\overline{h_n},t) + cnt(F,\overline{h_n},c,t)
\end{gathered}$$

■

This lemma captures an essential property of the n-multiplier and n-counter. This is again shown in the following example.

Example 3. Consider the term $t = g(g(a,a), g(F(g(a,F(g(a,a)))), g(F(a), F(F(F(b))))))$ and substitution $\{F \mapsto \lambda x.g(a,g(x,x))\}$. The n-counter of t at 2 is $cnt(F,2,a,t) = 14$ and the n-multiplier of t at 2 is $mul(F,2,t) = 11$. Observe $occ_\Sigma(a, t\{F \mapsto \lambda x.g(a,g(x,x))\}) = 25$ and $occ(a,s) \cdot mul(F,2,t) + cnt(F,2,a,t) = 25$. □

Up until now we considered arbitrary terms and substitutions. We now apply these results to unification problems and their solutions. In particular, a corollary of Lemma 2 is that there is a direct relation between the n-multiplier and n-counter of a unifiable unification problem given a unifier of the problem. The following lemma describes this relation.

Lemma 3 (Unification Condition)**.** *Let $(\mathcal{U},F)$ be a unifiable n-SOGU problem such that $\mathcal{V}_f(\mathcal{U}) \subseteq \{F\}$, $h_1,\cdots,h_n \geq 0$, and $\sigma = \{F \mapsto \lambda\overline{x_n}.s\}$ a unifier of $(\mathcal{U},F)$ such that $\mathcal{V}_i(s) = \{\overline{x_n}\}$ and for all $1 \leq i \leq n, occ(x,s) = h_i$. Then for all $c \in \Sigma^{\leq 1}$,*

$$occ(c,s) \cdot (mul_l(F,\overline{h_n},\mathcal{U}) - mul_r(F,\overline{h_n},\mathcal{U})) = cnt_r(F,\overline{h_n},c,\mathcal{U}) - cnt_l(F,\overline{h_n},c,\mathcal{U}). \quad (1)$$

Proof. By Lemma 1, for any $c \in \Sigma^{\leq 1}$ and $u \stackrel{?}{=} v \in \mathcal{U}$, we have $occ(c,u\sigma) = occ(c,v\sigma)$ and by Lemma 2 we also have

$$\begin{gathered}
occ(c,u\sigma) = occ(c,s) \cdot mul(F,\overline{h_n},u) + cnt(F,\overline{h_n},c,u), \text{ and} \\
occ(c,v\sigma) = occ(c,s) \cdot mul(F,\overline{h_n},v) + cnt(F,\overline{h_n},c,v).
\end{gathered}$$

Thus, for any $c \in \Sigma^{\leq 1}$ and $u \stackrel{?}{=}_F v \in \mathcal{U}$,

$$occ(c,s) \cdot mul(F,\overline{h_n},u) + cnt(F,\overline{h_n},c,u) = occ(c,s) \cdot mul(F,\overline{h_n},v) + cnt(F,\overline{h_n},c,v).$$

From this equation we can derive:

$$occ(c,s) \cdot (mul(F,\overline{h_n},u) - mul(F,\overline{h_n},v)) = cnt(F,\overline{h_n},c,v) - cnt(F,\overline{h_n},c,u) \quad (2)$$

We can generalize this to $\mathcal{U}$ by computing Equation 1 for each $u \stackrel{?}{=}_F v \in \mathcal{U}$ adding the results together. The result is the following equation:

$$occ(c,s) \cdot ((mul_l(F,\overline{h_n},\mathcal{U}) - mul_r(F,\overline{h_n},\mathcal{U})) = cnt_r(F,\overline{h_n},c,\mathcal{U}) - cnt_l(F,\overline{h_n},c,\mathcal{U}).$$

■

The *unification condition* is at the heart of the undecidability proof presented in Section 4. Essentially, Equation 1 relates the left and right side of a unification equation giving a necessary condition for unification. The following example shows an instance of this property.

Example 4. Consider the 1-SOGU problem $F(g(a,a)) \stackrel{?}{=}_F g(F(a),F(a))$ and the unifier $\sigma = \{F \mapsto \lambda x.g(x,x)\}$. Observe that

$$occ(a,g(x,x)) \cdot ((mul_l(F,2,F(g(a,a))) - mul_r(F,2,g(F(a),F(a)))) = 0 \cdot (1-2) = 0$$

and for the right we get $cnt_r(F,h,a,g(F(a),F(a))) - cnt_l(F,h,a,F(g(a,a))) = 4-4 = 0$. □

4 Undecidability n-SOGU

In this section, we prove the main result of this paper. We do so by using the machinery we built in the previous section to encode Diophantine equations in unification problems. As a result, we are able to transfer undecidability results Diophantine equations to satisfying the following unification condition for n-SOGU: for a given $c \in \Sigma^{\leq 1}$ and n-SOGU problem $(\mathcal{U}, F)$, does there exist $\overline{h_n} \geq 0$ such that $cnt_r(F,\overline{h_n},c,\mathcal{U}) = cnt_l(F,\overline{h_n},c,\mathcal{U})$. This unification condition is a necessary condition for unifiability.

For the remainder of this section, we consider a finite signature Σ such that $\{g,a,b\} \subseteq \Sigma$, $arity(g) = 2$, and $arity(a) = arity(b) = 0$. By $p(\overline{x_n})$ we denote a polynomial with integer coefficients over the variables $x_1, \cdots, x_n$ ranging over the natural numbers.

First, we define a second-order term representation for arbitrary monomials as follows:

Definition 3 (n-Converter)**.** *Let $p(\overline{x_n}) = c \cdot x_1^{k_1} \cdots x_n^{k_n}$ be a monomial where $c \geq 0$, for all $1 \leq i \leq n$, $k_i \geq 0$, and $F \in \mathcal{V}_f^n$. Then we define the second-order term representation of $p(\overline{x_n})$, as $cvt_1(F,a,c,k_1,\cdots,k_n)$, where cvt_i is defined recursively as follows:*

- $cvt_n(F,a,c,\overbrace{0,\cdots,0}^{n}) = s$ *where* $s \in \mathcal{T}_G$ *such that* $occ_\Sigma(a,s) = c$.
- $cvt_i(F,a,c,\overline{k_n}) = s[cvt_{i+1}(F,a,1,\overline{k_n})]_p$ *where*
  - $1 \leq i \leq n$, $k_i = 0$, $1 \leq j < i$, $k_j = 0$,
  - $s \in \mathcal{T}_G$ *such that* $occ_\Sigma(a,s) = c$, $dep(s) \geq 1$, *and there exists* $q \in pos(s)$ *such that* $occ(a,s|_q) = 0$,
  - $p \in pos(s)$ *such that* $occ(a,s|_p) = 0$,
- $cvt_i(F,a,c,\overline{k_n}) = s[F(\overline{t_{i-1}}, cvt_i(F,a,c,\overline{k_{i-1}},k_i-1,k_{i+1},\cdots,k_n),t_{i+1},\cdots,t_n)]_p$ *where*
  - $1 \leq i \leq n$, $k_i \geq 1$, $1 \leq j < i$, $k_j = 0$,
  - $s \in \mathcal{T}_G$ *such that* $occ_\Sigma(a,s) = 0$ *and* $dep(s) > 1$,
  - $p \in pos(s)$, *and*
  - *for all* $1 \leq j \leq n$ *where* $j \neq i$, $t_j \in \mathcal{T}_G$ *such that* $occ_\Sigma(a,t_j) = 0$.

Intuitively, the n-converter takes a monomial in n variables with a positive coefficient and converts it into a n-SOG term containing the function variable F with a number of occurrences of F equivalent to the sum of the exponents in the monomial. When the exponent of a variable is 0 we still consider it, but do not introduce an F. Note that n-converter is defined primitive recursively and will always terminate. Defining the index to count upwards simplifies the proofs below. Example 5 illustrates the construction of a term from a simple monomial with 3 variables and Example 6 & 7 construct the n-multiplier and n-counter of the resulting term, respectively.

Example 5. Consider the monomial $3 \cdot x^3y^2z^3$. One possible term t that represents this monomial is $t = cvt_1(F, a, 3, 3, 2, 3)$ defined as follows:

$$\begin{aligned}
&cvt_1(F, a, 3, 3, 2, 3) = \\
&\qquad g(F(g(F(g(F(g(cvt_2(F, a, 1, 0, 2, 3), g(a, g(a, a)), b, b), b), b, b), b), b, b), b)\\
&cvt_2(F, a, 1, 0, 2, 3) = g(F(b, g(F(b, g(cvt_3(F, a, 1, 0, 0, 3), a), b), b), b), b)\\
&cvt_3(F, a, 1, 0, 0, 3) = g(F(b, b, g(F(b, b, F(b, b, a)), b)), b)
\end{aligned}$$

□

Example 6. Consider the term from Example 5. The n-multiplier is computed as follows:

$$\begin{aligned}
mul(F, h_1, h_2, h_3, cvt_1(F, a, 3, 3, 2, 3)) =& 1 + h_1 + h_1^2 + h_1^3 \cdot \\
& mul(F, h_1, h_2, h_3, cvt_2(F, a, 1, 0, 2, 3))\\
mul(F, h_1, h_2, h_3, cvt_2(F, a, 1, 0, 2, 3)) =& 1 + h_2 + h_2^2 \cdot mul(F, h_1, h_2, h_3, cvt_3(F, a, 1, 0, 0, 3))\\
mul(F, h_1, h_2, h_3, cvt_3(F, a, 1, 0, 0, 3)) =& 1 + h_3 + 2 \cdot h_3^2
\end{aligned}$$

Together, this results in $1 + h_1 + h_1^2 + h_1^3 \cdot (1 + h_2 + h_2^2 \cdot (1 + h_3 + 2 \cdot h_3^2)) = 1 + h_1 + h_1^2 + h_1^3 + h_1^3h_2 + h_1^3h_2^2 + h_1^3h_3h_2^2 + h_1^3h_3^2h_2^2$. □

Example 7. Consider the term from Example 5. The n-counter is computed as follows:

$$\begin{aligned}
cnt(F, h_1, h_2, h_3, a, cvt_1(F, a, 3, 3, 2, 3)) &= 3 \cdot h_1^3 \cdot cnt(F, h_1, h_2, h_3, cvt_2(F, a, 1, 0, 2, 3))\\
cnt(F, h_1, h_2, h_3, cvt_2(F, a, 1, 0, 2, 3)) &= h_2^2 \cdot cnt(F, h_1, h_2, h_3, cvt_3(F, a, 1, 0, 0, 3))\\
cnt(F, h_1, h_2, h_3, cvt_3(F, a, 1, 0, 0, 3)) &= h_3^3
\end{aligned}$$

Together we get $3 \cdot h_1^3h_2^2h_3^3$. Observe that this is precisely the polynomial From Example 5. □

Using the operator defined in Definition 3, we can transform a polynomial with integer coefficients into a n-SOGU problem by constructing a specific unification equation for each monomial. The next definition describes the process.

Definition 4. *Let $p(\overline{x_n})$ be a polynomial and $F \in \mathcal{V}_f^n$. Then the n-SOGU problem $(\mathcal{U}, F)$ induced by $p(\overline{x_n})$ contains precisely the unification problems defined as follows:*

- *For every monomial $c \cdot x_1^{k_1} \cdots x_n^{k_n}$ of $p(\overline{x_n})$ where $c < 0$,*

$$cvt_1(F, a, |c| + 1, \overline{k_n}) \stackrel{?}{=}_F cvt_1(F, a, 1, \overline{k_n}) \in \mathcal{U}.$$

- *For every monomial $c \cdot x_1^{k_1} \cdots x_n^{k_n}$ of $p(\overline{x_n})$ where $c > 0$,*

$$cvt_1(F, a, 1, \overline{k_n}) \stackrel{?}{=}_F cvt_1(F, a, c + 1, \overline{k_n}) \in \mathcal{U}.$$

Observe that in Definition 4, the left and right sides of the unification equations need not be precisely the same term modulo occurrences of a. The result of this translation is that the n-counter captures the structure of the polynomial and the n-multipliers cancel out.

Lemma 4. *Let $n \geq 1$, $p(\overline{x_n})$ be a polynomial, and $(\mathcal{U}, F)$ a n-SOGU problem induced by $p(\overline{x_n})$. Then*

$$p(\overline{x_n}) = cnt_r(F, \overline{x_n}, c, \mathcal{U}) - cnt_l(F, \overline{x_n}, c, \mathcal{U}) \quad \text{and} \quad 0 = mul_l(F, \overline{x_n}, \mathcal{U}) - mul_r(F, \overline{x_n}, \mathcal{U}).$$

Proof. We proceed by induction on n. Furthermore, we focus on $p(\overline{x_n})$ being a monomial as the general case is a simple corollary of the monomial case.
Base case: Let us consider for the base case that $p(x_1) = c \cdot x_1^{k_1}$ where w.l.o.g, $c > 0$. Observe that, when $c < 0$, the unification problem remains unchanged, but the order of the terms changes (Definition 4). This implies that $\mathcal{U}$ contains a single unification equation: $cvt_1(F, a, 1, k_1) \stackrel{?}{=}_F cvt_1(F, a, c+1, k_1)$ where

$$u = cvt_1(F, a, 1, k_1) = s_1^l[F(s_2^l[\cdots s_{k_1}^l[F(t_1^l)]_{p_{k_1}} \cdots)]_{p_2})]_{p_1},$$
$$v = cvt_1(F, a, c+1, k_1) = s_1^r[F(s_2^r[\cdots s_{k_1}^l[F(t_1^r)]_{p_{k_1}} \cdots)]_{p_2})]_{p_1},$$

with $occ(a, t_1^l) = 1$, and $occ(a, t_1^r) = c + 1$. Observe that $occ_\Sigma(F, u) = occ_\Sigma(F, v) = k_1$ and the occurrences are nested within each other. Thus by Definition 2 & 3, $cnt(F, x_1, a, u) = x_1^{k_1}$ and $cnt(F, x_1, a, v) = (c + 1) \cdot x_1^{k_1}$. Thus, we have,

$$cnt(F, x_1, a, v) - cnt(F, x_1, a, u) = (c + 1) \cdot x_1^{k_1} - x_1^{k_1} = c \cdot x_1^{k_1} = p(x_1).$$

By Definition 1 & 3, $mul(F, x_1, u) = \sum_{j=0}^{k_1-1} x_i^j$ and $mul(F, x_1, v) = \sum_{j=0}^{k_1-1} x_i^j$

$$mul(F, x_1, u) - mul(F, x_1, v) = \sum_{j=0}^{k_1-1} x_i^j - \sum_{j=0}^{k_1-1} x_i^j = 0.$$

Step: For the induction hypothesis, we assume the statement holds for $n' \leq n$ and show that it holds for $n + 1$. This implies that the induction hypothesis holds for all polynomials $p(x_1, \cdots, x_m) = c' \cdot x_1^{k_1} \cdot \ldots \cdot x_m^{k_m}$ where $c' \geq 0$ and $m \leq n$ and we are proving that the statement holds for polynomials $p(x_1, \cdots, x_n, x_{n+1}) = c \cdot x_1^{k_1} \cdot \ldots \cdot x_{n+1}^{k_{n+1}}$ where $c \geq 0$. Observe that $\mathcal{U}$ contains a single unification equation: $cvt_1(F, a, 1, \overline{k_{n+1}}) \stackrel{?}{=}_F cvt_1(F, a, c+1, \overline{k_{n+1}})$ where

$$\begin{aligned}
u &= cvt_1(F, a, 1, \overline{k_{n+1}}) = s_1^l[F(s_2^l[\cdots s_{k_1}^l[F(t_1^l, r_2^{k_1}, \cdots, r_{n+1}^{k_1})]_{p_{k_1}} \cdots)]_{p_2}, r_2^1, \cdots, r_{n+1}^1)]_{p_1},\\
v &= cvt_1(F, a, c+1, \overline{k_{n+1}})\\
&= s_1^r[F(s_2^r[\cdots s_{k_1}^l[F(t_1^r, w_2^{k_1}, \cdots, w_{n+1}^{k_1})]_{p_{k_1}} \cdots)]_{p_2}, , w_2^1, \cdots, w_{n+1}^1)]_{p_1},
\end{aligned}$$

and t_1^l and t_1^r abbreviate the rest of the construction. Observe that the first argument of F is ignored in t_1^l and t_1^r (set to an arbitrary term without occurrences of a), thus we can remove it generating the terms $u^* = t_1^l\{F \mapsto \lambda\overline{y_{n+1}}.G(y_2, \cdots, y_{n+1})\}$ and $v^* = t_1^r\{F \mapsto \lambda\overline{y_{n+1}}.G(y_2, \cdots, y_{n+1})\}$, i.e. dropping the first argument. By the induction hypothesis, the statement holds for the unification problem $u^* \stackrel{?}{=}_G v^*$. Notably, this substitution does not change the coefficients or the exponents of the variables $x_2, \cdots, x_{n+1}$ and that u^* and v^* represent the polynomials $x_2^{k_2} \cdot \ldots \cdot x_{n+1}^{k_{n+1}}$ and $(c+1) \cdot x_2^{k_2} \cdot \ldots \cdot x_{n+1}^{k_{n+1}}$, respectively. Furthermore,

$$\begin{aligned}
u' &= cvt_1(F, a, 1, k_1) = s_1^l[F(s_2^l[\cdots s_{k_1}^l[F(t_1^*, r_2^{k_1}, \cdots, r_{n+1}^{k_1})]_{p_{k_1}} \cdots)]_{p_2}, r_2^1, \cdots, r_{n+1}^1)]_{p_1}\\
v' &= cvt_1(F, a, c+1, k_1)\\
&= s_1^r[F(s_2^r[\cdots s_{k_1}^l[F(t_2^*, w_2^{k_1}, \cdots, w_{n+1}^{k_1})]_{p_{k_1}} \cdots)]_{p_2}, w_2^1, \cdots, w_{n+1}^1)]_{p_1}
\end{aligned}$$

where t_1^* and t_2^* are ground terms such that $\mathcal{V}_f(t_1^*) = \mathcal{V}_f(t_2^*) = \emptyset$, $occ(a, t_1^*) = 1$, and $occ(a, t_2^*) = 1$. Observe that u' and v' represent the polynomials x^{k_1} and x^{k_1}, respectively, and by the induction hypothesis the statement holds for the unification problem $u' \stackrel{?}{=}_F v'$.

Finally, let $u_2 = u'[t_1^l]_{q_1}$ and $v_2 = v'[t_1^r]_{q_2}$ where q_1 is the position of t_1^* in u' and q_2 is the position of an occurrence of t_2^* in v'. This amounts to multiplying the two polynomials. It follows that the statement holds for u and v by applying the substitution $\{G \mapsto \lambda\overline{y_n}.F(t', y_1, \cdots, y_{n+1})\}$ to u_2 and v_2 were t' is a term such that $occ_\Sigma(a, t') = 0$.

When $p(x_1, \cdots, x_n)$ consists of multiple monomials, observe that this amounts to adding equations of the form constructed above together and thus trivially follows. ∎

We now prove that the unification condition as introduced in Lemma 3 is equivalent to finding the solutions to polynomial equations. The following shows how a solution to a polynomial can be obtained from the unification condition and vice versa.

Lemma 5. *Let $p(\overline{x_n})$ be a polynomial and $(\mathcal{U}, F)$ the n-SOGU problem induced by $p(\overline{x_n})$ using the $c \in \Sigma^{\leq 1}$ (Definition 3). Then there exists $h_1, \cdots, h_n \geq 0$ such that $cnt_l(F, \overline{h_n}, c, \mathcal{U}) = cnt_r(F, \overline{h_n}, c, \mathcal{U})$ (*unification condition*) if and only if $\{x_i \mapsto h_i \mid 1 \leq i \leq n \wedge h_i \in \mathbb{N}\}$ is a solution to $p(\overline{x_n}) = 0$.*

Proof. Observe that, by lemma 4, $p(\overline{h_n}) = cnt_r(F, \overline{h_n}, c, \mathcal{U}) - cnt_l(F, \overline{h_n}, c, \mathcal{U})$. Thus, we can prove the two directions as follows:
$\Longrightarrow$: If $cnt_l(F, \overline{h_n}, c, \mathcal{U}) = cnt_r(F, \overline{h_n}, c, \mathcal{U})$ then $p(\overline{h_n}) = 0$.
$\Longleftarrow$: Any substitution of natural numbers $\overline{h_n}$ into $\overline{x_n}$ such that $p(\overline{h_n}) = 0$ would imply that $cnt_r(F, \overline{h_n}, c, \mathcal{U}) = cnt_l(F, \overline{h_n}, c, \mathcal{U})$. ∎

Using Lemma 5, we now show that finding $h_1, \cdots, h_n \geq 0$ such that the *unification condition* holds is undecidable by reducing solving $p(\overline{x_n}) = 0$ for arbitrary polynomials over $\mathbb{N}$ (Theorem 1) to finding $h_1, \cdots, h_n \geq 0$ which satisfy the *unification condition*.

Lemma 6 (Equalizer Problem)**.** *For a given n-SOGU problem, finding $h_1, \cdots, h_n \geq 0$ such that the* unification condition *(Lemma 3) holds is undecidable.*

Proof. Given a polynomial $p(\overline{x_n}) = 0$ we construct, with the help of Definition 4, an n-SOGU problem $(\mathcal{U}, F)$ using $c \in \Sigma^{\leq 1}$. Importantly, this process is terminating, finitely computable. We now prove that $h_1, \cdots, h_n \geq 0$ satisfies the unification condition (Lemma 3) over $(\mathcal{U}, F)$ using $c \in \Sigma^{\leq 1}$ if and only if $p(\overline{x_n}) = 0$ has solutions:
$\Longrightarrow$: Assume: $h_1, \cdots, h_n \geq 0$ satisfies the unification condition (Lemma 3) over $(\mathcal{U}, F)$ using $c \in \Sigma^{\leq 1}$, then by Lemma 4, $p(\overline{h_n}) = 0$ as

$$p(\overline{x_n}) = cnt_r(F, \overline{x_n}, c, \mathcal{U}) - cnt_l(F, \overline{x_n}, c, \mathcal{U}) \quad \text{and} \quad 0 = mul_l(F, \overline{x_n}, \mathcal{U}) - mul_r(F, \overline{x_n}, \mathcal{U}).$$

$\Longleftarrow$: Assume: $p(\overline{x_n}) = 0$ now has solutions. Then let $h_1, \cdots, h_n$ be a solution to $p(\overline{x_n}) = 0$. Observe that by Lemma 4,

$$p(\overline{x_n}) = cnt_r(F, \overline{x_n}, c, \mathcal{U}) - cnt_l(F, \overline{x_n}, c, \mathcal{U}) \quad \text{and} \quad 0 = mul_l(F, \overline{x_n}, \mathcal{U}) - mul_r(F, \overline{x_n}, \mathcal{U}).$$

Thus, $h_1, \cdots, h_n$ satisfy the unification condition (Lemma 3) over $(\mathcal{U}, F)$ using $c \in \Sigma^{\leq 1}$. By Theorem 1, we know that finding integer solutions to $p(\overline{x_n}) = 0$ is undecidable. Hence, solving for the unification condition is also undecidable. ∎

We have proven the equalizer problem to be undecidable by reducing Hilbert's 10th problem. Now, we use the equalizer problem to prove the main result of this paper.

Theorem 2. *There exists $n \geq 1$ such that n-SOGU is undecidable.*

Proof. We will prove this by contradiction. Hence, assume that n-SOGU is decidable and let $(\mathcal{U}, F)$ be an arbitrary n-SOGU problem. We will now consider the following two cases where a solution exists and where no solution exists:

Case 1: Assume $(\mathcal{U}, F)$ has at least one solution. Note that there might be infinitely many unifiers, but since by assumption n-SOGU is decidable we can compute a solution. We let $\sigma = \{F \mapsto \lambda\, \overline{y_n}.s\}$ be a solution. Now, for all $1 \leq i \leq n$, let $h_i = occ(y_i, s)$. Observe that for all $c \in \Sigma^{\leq 1}$,

$$occ(c, s) \cdot ((mul_l(F, \overline{h_n}, \mathcal{U}) - mul_r(F, \overline{h_n}, \mathcal{U})) = cnt_r(F, \overline{h_n}, c, \mathcal{U}) - cnt_l(F, \overline{h_n}, c, \mathcal{U}).$$

Observe that this choice of $h_1, \cdots, h_n$ is a solution to the *Equalizer problem*.
Case 2: Assume $(\mathcal{U}, F)$ has no solution. Then, by Lemma 3, for all $h_1, \cdots, h_n \geq 0$, there exists $c \in \Sigma^{\leq 1}$ such that for all $k \geq 0$,

$$k \cdot ((mul_l(F, \overline{h_n}, \mathcal{U}) - mul_r(F, \overline{h_n}, \mathcal{U})) \neq cnt_r(F, \overline{h_n}, c, \mathcal{U}) - cnt_l(F, \overline{h_n}, c, \mathcal{U}).$$

The above implies that there is no solution to the *Equalizer Problem*.

Note that computing $h_1, \cdots, h_n \geq 0$ in the first case is decidable and an answer in the second case is vacuously decidable by the assumption. However, this would imply that the *Equalizer Problem* is decidable, which is a contradiction. ■

Theorem 2 answers the question posed in Section 1 by proving that first-order variable occurrence does not impact the decidability of second-order unification. By combining the encoding used for the reduction as well as other number theoretic results, we can refine n to at most 9 using the encoding presented in [16]. We leave a more detailed investigation for future work.

5 Decidability of 1-SOGU with Bounded Congruence

In this section, we investigate further properties of the n-multiplier and n-counter functions. As a result, we obtain a decidable fragment of 1-SOGU based on a property we call *bounded congruence* (Definition 6). The following lemma highlights that the 1-counter and 1-multiplier are univariate polynomials. We will use this fact to define bounded congruence.

Lemma 7. *Let $(\mathcal{U}, F)$ be a 1-SOGU, $c \in \Sigma^{\leq 1}$, and $h \geq 0$. Then $cnt_l(F, h, c, \mathcal{U}) = q_l(h)$, $cnt_r(F, h, c, \mathcal{U}) = q_r(h)$, $mul_l(F, h, \mathcal{U}) = p_l(h)$ and $mul_r(F, h, \mathcal{U}) = p_r(h)$, where $q_l(x)$ and $q_r(x)$, $p_l(x)$ and $p_r(x)$ are polynomials with natural number coefficients and x ranges over $\mathbb{N}$.*

Proof. Derived by telescoping the recursive definition. ■

Furthermore, we denote the i^{th} coefficient of $cnt_l(F, h, c, \mathcal{U})$, $cnt_r(F, h, c, \mathcal{U})$, $mul_l(F, h, \mathcal{U})$, and $mul_r(F, h, \mathcal{U})$ by $cnt_l(F, h, c, \mathcal{U})_i$, $cnt_r(F, h, c, \mathcal{U})_i$, $mul_l(F, h, \mathcal{U})_i$ and $mul_r(F, h, \mathcal{U})_i$, respectively. Observe that, considering the 1-counter and 1-multiplier as univariate polynomials in h highlights the fact that only the 0^{th} coefficient is not multiplied by h. Thus, for h larger then the *Max-arg-multiplicity* we only need to consider the 0^{th} coefficients when considering divisibility modulo h.

Definition 5 (Max-arg-multiplicity)**.** *Let $(\mathcal{U}, F)$ be a 1-SOGU. Then we define $h_{\mathcal{U}} \geq 0$ as the minimal natural number such that $h_{\mathcal{U}} \geq mul_l(F, h, \mathcal{U})_0 - mul_r(F, h, \mathcal{U})_0 + 1$ and for all $c \in \Sigma^{\leq 1}$, and $h_{\mathcal{U}} \geq cnt_r(F, h_{\mathcal{U}}, c, \mathcal{U})_0 - cnt_l(F, h_{\mathcal{U}}, c, \mathcal{U})_0 + 1$.*

In the following lemma, we use max-arg-multiplicity to show that for large enough values of h we only have to consider the 0^{th} coefficient of the polynomial representations of the 1-multiplier and 1-counter.

Lemma 8. *Let $(\mathcal{U}, F)$ be a 1-SOGU problem. Then for all $c \in \Sigma^{\leq 1}$,*

$$mul_l(F, h_{\mathcal{U}}, \mathcal{U})_0 - mul_r(F, h_{\mathcal{U}}, c, \mathcal{U})_0 \equiv mul_l(F, h_{\mathcal{U}}, \mathcal{U}) - mul_r(F, h_{\mathcal{U}}, \mathcal{U}) \mod h_{\mathcal{U}}$$
$$cnt_r(F, h_{\mathcal{U}}, c, \mathcal{U})_0 - cnt_l(F, h_{\mathcal{U}}, c, \mathcal{U})_0 \equiv cnt_r(F, h_{\mathcal{U}}, c, \mathcal{U}) - cnt_l(F, h_{\mathcal{U}}, c, \mathcal{U}) \mod h_{\mathcal{U}}$$

Proof. The 0^{th} coefficient is the only coefficient not multiplied by $h_{\mathcal{U}}$. ∎

Below we define *bounded Congruence* and refer to this fragment of 1-SOGU as 1-SOGU with bounded congruence.

Definition 6 (Bounded Congruence)**.** *Let $(\mathcal{U}, F)$ be a 1-SOGU problem. We refer to $(\mathcal{U}, F)$ as having* bounded congruence *if there exists $c \in \Sigma^{\leq 1}$:*

- $mul_l(F, h_{\mathcal{U}}, \mathcal{U}) - mul_r(F, h_{\mathcal{U}}, \mathcal{U}) \not\equiv cnt_r(F, h_{\mathcal{U}}, c, \mathcal{U}) - cnt_l(F, h_{\mathcal{U}}, c, \mathcal{U}) \mod h_{\mathcal{U}}$
- $0 \not\equiv cnt_r(F, h_{\mathcal{U}}, c, \mathcal{U}) - cnt_l(F, h_{\mathcal{U}}, c, \mathcal{U}) \mod h_{\mathcal{U}}$

Observe that unification problems with bounded congruence are unifiable iff the number of occurrences of the bound variable is less than the Max-arg-multiplicity of the problem. The following lemma addresses this observation.

Lemma 9. *Let $(\mathcal{U}, F)$ be a 1-SOGU with bounded congruence. Then $\mathcal{U}$ is unifiable iff $\{F \mapsto \lambda x.t\}$ unifies $\mathcal{U}$ where $\mathcal{V}_i(t) = \{x\}$, and $occ(x, t) < h_{\mathcal{U}}$.*

Proof. The following holds (by Lemma 8) for all $h \geq h_{\mathcal{U}}$,
$mul_l(F, h_{\mathcal{U}}, \mathcal{U}) - mul_r(F, h_{\mathcal{U}}, \mathcal{U}) \mod h_{\mathcal{U}} = mul_l(F, h, \mathcal{U}) - mul_r(F, h, \mathcal{U}) \mod h$,
$cnt_r(F, h_{\mathcal{U}}, c, \mathcal{U}) - cnt_l(F, h_{\mathcal{U}}, c, \mathcal{U}) \mod h_{\mathcal{U}} = cnt_r(F, h, c, \mathcal{U}) - cnt_l(F, h, c, \mathcal{U}) \mod h$.
Thus, for all $h \geq h_{\mathcal{U}}$,

$$mul_l(F, h, \mathcal{U}) - mul_r(F, h, \mathcal{U}) \not\equiv cnt_r(F, h, c, \mathcal{U}) - cnt_l(F, h, c, \mathcal{U}) \mod h,$$

and $0 \not\equiv cnt_r(F, h, c, \mathcal{U}) - cnt_l(F, h, c, \mathcal{U}) \mod h$. In other words, for any unifier $\{F \mapsto \lambda x.t\}$ of $(\mathcal{U}, F)$, $occ(x, t) < h_{\mathcal{U}}$. ∎

Given a 1-SOGU problem with *bounded congruence*, a unifier can be found by checking all values less than or equal to the max-arg-multiplicity. By Lemma 3, we get the precise number of occurrences of every constant and monadic function symbol occurring in the problem. In the following lemma we provide the precise constraints needed for unifiability.

Lemma 10. *Let $(\mathcal{U}, F)$ be a 1-SOGU problem with* bounded-congruence. *Then $\mathcal{U}$ is unifiable iff there exists $1 \leq h' < h_{\mathcal{U}}$ such that for all $c \in \Sigma^{\leq 1}$, either*

$$mul_l(F, h', \mathcal{U}) - mul_r(F, h', \mathcal{U}) \mod h' \equiv cnt_r(F, h', c, \mathcal{U}) - cnt_l(F, h', c, \mathcal{U}) \mod h' \quad (3)$$

or Equation 3 does not hold, but

$$0 \equiv cnt_r(F, h', c, \mathcal{U}) - cnt_l(F, h', c, \mathcal{U}) \mod h' \quad (4)$$

holds and there exists a substitution σ such that σ unifies $\mathcal{U}$, $\sigma = \{F \mapsto \lambda x.t\}$, $\mathcal{V}_i(t) = \{x\}$, $occ(x, t) = h'$, for all $c \in \Sigma^{\leq 1}$, such that Equation 3 holds,

$$occ(c, t) = \frac{cnt_r(F, h', c, \mathcal{U}) - cnt_l(F, h', c, \mathcal{U})}{mul_l(F, h', \mathcal{U}) - mul_r(F, h', \mathcal{U})},$$

and for all $c \in \Sigma^{\leq 1}$, such that Equation 3 does not hold, $occ(c, t) = 0$.

Proof. The number of occurrences of $c \in \Sigma^{\leq 1}$ is fixed by Lemma 2. ∎

To emphasise this, we present the following example:

Example 8. Consider $\mathcal{U} = \{g(b, F(g(b, g(a, a)))) \stackrel{?}{=}_F g(b, g(b, g(F(a), F(a))))\}$. Observe that $mul_l(F, h, \mathcal{U}) = 1$, $mul_r(F, h, \mathcal{U}) = 2$, $cnt_l(F, h, a, \mathcal{U}) = 2 \cdot h$, $cnt_l(F, h, b, \mathcal{U}) = 1 + h$, $cnt_r(F, h, a, \mathcal{U}) = 2 \cdot h$, $cnt_r(F, h, b, \mathcal{U}) = 2$, and $h_{\mathcal{U}} = 3$. Thus,

$$cnt_r(F, h, a, \mathcal{U}) - cnt_l(F, h, a, \mathcal{U}) = 0$$
$$cnt_r(F, h, b, \mathcal{U}) - cnt_l(F, h, b, \mathcal{U}) = 1 - h$$
$$mul_l(F, h, \mathcal{U}) - mul_r(F, h, \mathcal{U}) = -1$$

Observe that $h = 3$,

$$mul_l(F, 2, \mathcal{U}) - mul_r(F, 2, \mathcal{U}) \not\equiv cnt_r(F, 2, b, \mathcal{U}) - cnt_l(F, 2, b, \mathcal{U}) \mod 3$$
$$0 \not\equiv cnt_r(F, 2, b, \mathcal{U}) - cnt_l(F, 2, b, \mathcal{U}) \mod 3$$

However, for $h = 2$,

$$mul_l(F, 2, \mathcal{U}) - mul_r(F, 2, \mathcal{U}) \not\equiv cnt_r(F, 2, a, \mathcal{U}) - cnt_l(F, 2, a, \mathcal{U}) \mod 2$$
$$0 \equiv cnt_r(F, 2, a, \mathcal{U}) - cnt_l(F, 2, a, \mathcal{U}) \mod 2$$
$$mul_l(F, 2, \mathcal{U}) - mul_r(F, 2, \mathcal{U}) \equiv cnt_r(F, 2, b, \mathcal{U}) - cnt_l(F, 2, b, \mathcal{U}) \mod 2$$

This implies that the unifier of $\mathcal{U}$ has 1 occurrence of b and no occurrences of a. Observe that $\{F \mapsto \lambda x.g(b, g(x, x))$ is a unifier of $\mathcal{U}$. □

In conclusion, we obtain the following result about problems with bounded congruence.

Theorem 3. *1-SOGU with bounded-congruence is decidable.*

Proof. Let $(\mathcal{U}, F)$ be a 1-SOGU with bounded-congruence. If $(\mathcal{U}, F)$ is unifiable, then there exists a unifier $\sigma = \{F \mapsto \lambda x.t\}$ such that $\mathcal{V}_i(t) \subseteq \{x\}$ and $occ(x, t) < h_{\mathcal{U}}$ (Lemma 9). Once we bound the occurrences of x we can use the decision procedure for bounded second-order unification [18] to find a unifier. ■

Essentially, 1-SOGU with bounded-congruence derives a bound on the number of occurrences of the bound variable in any unifier from the structure of the unification problem. Thus, the resulting unification problem, after deriving the max number of bound variable occurrences, is an instance of *bound-second-order unification* [18]. When the unification problem does not have bounded congruence, we cannot compute a max number of bound variable occurrences as above.

Conjecture 1. *1-SOGU is undecidable.*

Observe that the above construction reduces 1-SOGU to solving a system of integer polynomials with two unknowns (Lemma 3) each of which has the form $p(x) - y_i \cdot q(x) = 0$. Let use assume that $|\Sigma^{\leq 1}| \geq 2$ and the unification problem which induces the system of polynomials contains at least two symbols from $\Sigma^{\leq 1}$. Observe that $q(x)$ is derived from the 1-multiplier and will be the same for every polynomial in the system. Let the system be $p_1(x) = y_1 \cdot q(x), \cdots, p_n(x) = y_n \cdot q(x)$. If there exists $1 \leq i < j \leq n$ such that $p_i(x) \neq p_j(x)$, then we can build a new univariate polynomial $p_i(x) - p_j(x) = 0$ if we assume the same number of occurrences for both symbols in the unifier. We can use real root approximation methods [2] to find the integer solutions (of which there are finitely many), and thus, attempt unifier construction for all integer solutions as such polynomials have a maximum number of solutions dependent on the degree.

This leaves two cases, (i) $|\Sigma^{\leq 1}| = 1$ (or unification problem only uses one member of $\Sigma^{\leq 1}$), or (ii) for all $1 \leq i < j \leq n$, $p_i(x) = p_j(x)$. Observe that in both cases, either the system of equations has only 1 equation and two unknowns, or we cannot make the assumption that the symbols have the same number of occurrences. Though, if all coefficients of $q(x)$ are zero, then we are again in the univariate case. Furthermore, both cases are partially solved by *bounded congruence*. Though, it remains unclear if there is a general decision procedure for 1-SOGU in the particular cases outlined above. We conjecture that 1-SOGU is undecidable when the resulting system of equations is of size 1, does not have *bounded congruence*, and both $p(x)$ and $q(x)$ are non-trivial polynomials.

6 Conclusion and Future Work

We show that second-order ground unification is undecidable be reducing Hilbert's 10^{th} problem over $\mathbb{N}$ to a necessary condition for unification. The reduction required two novel occurrence counting functions and their relationship to the existence of a unifier. Furthermore, we show that restricting ourselves to arity 1 function variables results in a decidable fragment modulo so-called *bound-congruence*. For future work we plan to address Conjecture 1 and the possibly decidable fragments discussed at the end of the previous section.

References

[1] Alessandro Abate, Cristina David, Pascal Kesseli, Daniel Kroening, and Elizabeth Polgreen. Counterexample guided inductive synthesis modulo theories. In Hana Chockler and Georg Weissenbacher, editors, *Computer Aided Verification - 30th International Conference, CAV 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 14-17, 2018, Proceedings, Part I*, volume 10981 of *Lecture Notes in Computer Science*, pages 270–288. Springer, 2018.

[2] Oliver Aberth. Iteration methods for finding all zeros of a polynomial simultaneously. *Mathematics of Computation*, 27:339–344, 1973.

[3] Rajeev Alur, Rastislav Bodík, Eric Dallal, Dana Fisman, Pranav Garg, Garvit Juniwal, Hadas Kress-Gazit, P. Madhusudan, Milo M. K. Martin, Mukund Raghothaman, Shambwaditya Saha, Sanjit A. Seshia, Rishabh Singh, Armando Solar-Lezama, Emina Torlak, and Abhishek Udupa. Syntax-guided synthesis. In Maximilian Irlbeck, Doron A. Peled, and Alexander Pretschner, editors, *Dependable Software Systems Engineering*, volume 40 of *NATO Science for Peace and Security Series, D: Information and Communication Security*, pages 1–25. IOS Press, 2015.

[4] Haniel Barbosa, Andrew Reynolds, Daniel El Ouraoui, Cesare Tinelli, and Clark W. Barrett. Extending SMT solvers to higher-order logic. In Pascal Fontaine, editor, *Automated Deduction - CADE 27 - 27th International Conference on Automated Deduction, Natal, Brazil, August 27-30, 2019, Proceedings*, volume 11716 of *Lecture Notes in Computer Science*, pages 35–54. Springer, 2019.

[5] William M. Farmer. A unification algorithm for second-order monadic terms. *Ann. Pure Appl. Log.*, 39(2):131–174, 1988.

[6] Harald Ganzinger, Florent Jacquemard, and Margus Veanes. Rigid reachability. In Jieh Hsiang and Atsushi Ohori, editors, *Advances in Computing Science - ASIAN '98, 4th Asian Computing Science Conference, Manila, The Philippines, December 8-10, 1998, Proceedings*, volume 1538 of *Lecture Notes in Computer Science*, pages 4–21. Springer, 1998.

[7] Warren D. Goldfarb. The undecidability of the second-order unification problem. *Theor. Comput. Sci.*, 13:225–230, 1981.

[8] Sumit Gulwani, Oleksandr Polozov, and Rishabh Singh. Program synthesis. *Found. Trends Program. Lang.*, 4(1-2):1–119, 2017.

[9] Gérard P. Huet. The undecidability of unification in third order logic. *Inf. Control.*, 22(3):257–267, 1973.

[10] Artur Jez. Context unification is in PSPACE. In Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias, editors, *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part II*, volume 8573 of *Lecture Notes in Computer Science*, pages 244–255. Springer, 2014.

[11] Jordi Levy. Linear second-order unification. In Harald Ganzinger, editor, *Rewriting Techniques and Applications, 7th International Conference, RTA-96, New Brunswick, NJ, USA, July 27-30, 1996, Proceedings*, volume 1103 of *Lecture Notes in Computer Science*, pages 332–346. Springer, 1996.

[12] Jordi Levy. On the limits of second-order unification. In Temur Kutsia and Christophe Ringeissen, editors, *Proceedings of the 28th International Workshop on Unification, UNIF 2014, Vienna, Austria, July 13, 2014*, pages 5–14, 2014.

[13] Jordi Levy, Manfred Schmidt-Schauß, and Mateu Villaret. The complexity of monadic second-order unification. *SIAM J. Comput.*, 38(3):1113–1140, 2008.

[14] Jordi Levy and Margus Veanes. On the undecidability of second-order unification. *Inf. Comput.*, 159(1-2):125–150, 2000.

[15] Claudio L Lucchesi. The undecidability of the unification problem for third order languages. *Report CSRR*, 2059:129–198, 1972.

[16] Yuri V. Matiyasevich. *Hilbert's tenth problem.* MIT Press, Cambridge, MA, USA, 1993.

[17] Julian Parsert, Chad E. Brown, Mikolas Janota, and Cezary Kaliszyk. Experiments on infinite model finding in SMT solving. In Ruzica Piskac and Andrei Voronkov, editors, *LPAR 2023: Proceedings of 24th International Conference on Logic for Programming, Artificial Intelligence and Reasoning, Manizales, Colombia, 4-9th June 2023*, volume 94 of *EPiC Series in Computing*, pages 317–328. EasyChair, 2023.

[18] Manfred Schmidt-Schauß. Decidability of bounded second order unification. *Inf. Comput.*, 188(2):143–178, 2004.