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 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 this fragment if the signature contains a binary function symbol and two constants. This generalizes known undecidability results. [1]

1 Introduction

In the 2014 addition of the unification workshop Levy [1] provided a comprehensive survey of decidability and undecidability results for second-order unification. While second-order unification without first-order variables was considered [2], 2 second-order variables were required to show undecidability. Furthermore, investigations proving undecidability of second-order unification with 1 second-order variable required first-order variables [2]. We generalize these result by showing one second-order variable is enough undecidability (no first-order variables). Proofs of all significant lemmas and theorems may be found in the *arxiv version* of the paper arxiv.org/abs/2404.10616.

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 $\Sigma^{\leq 1} = \{c \mid c \in \Sigma \wedge arity(c) \leq 1\}$.

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 $\mathcal{V}_i(t) = \emptyset$,

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

[1] Full Results and proofs in Arxiv paper arxiv.org/abs/2404.10616.

$\mathcal{V}_f(t) \neq \emptyset$ 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 $u \stackrel{?}{=}_F v \in \mathcal{U}$, $u\sigma =_{\alpha\beta} v\sigma$.

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

Theorem 2.1 (Hilberts 10^{th} problem or Matiyasevich–Robinson–Davis–Putnam theorem [3])**.** 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 3.1. 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 $u \stackrel{?}{=}_F v \in \mathcal{U}$, $occ_\Sigma(c, u\sigma) = occ_\Sigma(c, v\sigma)$.

Definition 3.1 (n-Mutiplier)**.** 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 $t = b$ and $arity(b) = 0$, then $mul(F, \overline{h_n}, t) = 0$.
- 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, $mul_l(F, \overline{h_n}, \mathcal{U}) = \sum_{u \stackrel{?}{=}_F v \in \mathcal{U}} mul(F, \overline{h_n}, u)$ and $mul_r(F, \overline{h_n}, \mathcal{U}) = \sum_{u \stackrel{?}{=}_F v \in \mathcal{U}} mul(F, \overline{h_n}, v)$.

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 3.1. 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 3.2 (n-Counter)**.** Let $c \in \Sigma^{\leq 1}$, t be a n-SOG term such that $\mathcal{V}_f(t) = \{F\}$ 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 $t = b$, $arity(b) = 0$, and $b \neq c$, then $cnt(F, \overline{h_n}, c, t) = 0$.
- 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 $t = F(\overline{t_n})$, 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, $cnt_l(F, \overline{h_n}, c, \mathcal{U}) = \sum_{u \stackrel{?}{=}_F v \in \mathcal{U}} cnt(F, \overline{h_n}, c, u)$ and $cnt_r(F, \overline{h_n}, c, \mathcal{U}) = \sum_{u \stackrel{?}{=}_F v \in \mathcal{U}} cnt(F, \overline{h_n}, c, v)$.

The n-counter captures how many occurrences of a given constant or monadic function symbol will occur in a term $t\sigma$ where $\mathcal{V}_f(t) = \{F\}$, $\sigma = \{F \mapsto \lambda \overline{x_n}.s\}$, $\mathcal{V}_i(s) \subseteq \{\overline{x_n}\}$, and for all $1 \leq i \leq n$, $occ(x_i, s) = h_i$ A concrete instance is presented in Example 3.2.

Example 3.2. 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 3.2. Let $c \in \Sigma^{\leq 1}$, t be a n-SOG term such that $\mathcal{V}_f(t) = \{F\}$, $h_1, \cdots, h_n \geq 0$, 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 $occ(c, t\sigma) = occ(c, s) \cdot mul(F, \overline{h_n}, t) + cnt(F, \overline{h_n}, c, t)$.

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

Example 3.3. 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 3.2 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 3.3 (Unification Condition)**.** Let $(\mathcal{U}, F)$ be a unifiable n-SOGU problem such that $\mathcal{V}_f(\mathcal{U}) = \{F\}$, $h_1, \cdots, h_n \geq 0$, 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 $c \in \Sigma^{\leq 1}$,

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

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 3.4. 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 $occ(a, g(x,x)) \cdot ((mul_l(F, 2, F(g(a,a))) - mul_r(F, 2, g(F(a), F(a)))) = 0 \cdot (1-2) = 0$ and $cnt_r(F, h, a, g(F(a), F(a))) - cnt_l(F, h, a, F(g(a,a))) = 4 - 4 = 0$.

4 Undecidability n-SOGU

We now use the ideas from 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 exists $\overline{h_n} \geq 0$ 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 and by $mono(p(\overline{x_n}))$ we denote the set of monomials of $p(\overline{x_n})$. Given a polynomial $p(\overline{x_n})$ and $1 \leq i \leq n$, if for all $m \in mono(p(\overline{x_n}))$, there exists a monomial m' such that $m = x_i \cdot m'$ then we say $div(p(\overline{x_n}), x_i)$. Furthermore, $deg(p(\overline{x}_n)) = \max\{k \mid k \geq 0 \wedge m = x_i^k \cdot q(\overline{x_n}) \wedge 1 \leq i \leq n \wedge m \in mono(p(\overline{x_n}))\}$. Given a polynomial $p(\overline{x_n})$, a polynomial $p'(\overline{x_n})$ is a sub-polynomial of $p(\overline{x_n})$ if $mono(p'(\overline{x_n})) \subseteq mono(p(\overline{x_n}))$. Using the above definition we define distinct sub-polynomials based on divisibility by one of the input unknowns.

Definition 4.1 (monomial groupings)**.** Let $p(\overline{x_n}) = q(\overline{x_n}) + c$ be a polynomial where $c \in \mathbb{Z}$, $0 \leq j \leq n$, and $S_j = \{m \mid m \in mono(p(\overline{x_n})) \wedge \forall i (1 \leq i < j \Rightarrow \neg div(m, x_i))\}$. Then

- $p(\overline{x_n})_0 = c$,
- $p(\overline{x_n})_j = 0$ if there does not exists $m \in S_j$ such that $div(m, x_j)$,

- otherwise, $p(\overline{x_n})_j = p'(\overline{x_n})$, where $p'(\overline{x_n})$ is the sub-polynomial of $p(\overline{x_n})$ such that $mono(p'(\overline{x_n})) = \{m \mid m \in S_j \wedge div(m, x_j)\}$.

Furthermore, let $p(\overline{x_n})_j = x_j \cdot p'(\overline{x_n})$. Then $p(\overline{x_n})_j \downarrow = p'(\overline{x_n})$.

We now define a second-order term representation for arbitrary polynomials as follows:

Definition 4.2 (n-Converter)**.** Let $p(\overline{x_n})$ be a polynomial and $F \in \mathcal{V}_f^n$. Then we define the positive (negative) second-order term representation of $p(\overline{x_n})$, as $cvt^+(F, p(\overline{x_n}))(cvt^-(F, p(\overline{x_n})))$, where cvt^+ (cvt^-) is defined recursively as follows:

- if $p(\overline{x_n}) = p(\overline{x_n})_0 = 0$, then $cvt^+(F, p(\overline{x_n})) = cvt^-(F, p(\overline{x_n})) = b$
- if $p(\overline{x_n}) = p(\overline{x_n})_0 = c \geq 1$, then
  - $cvt^+(F, p(\overline{x_n})) = t$ where $occ_\Sigma(a, t) = |p(\overline{x_n})_0| + 1$ and t is ground.
  - $cvt^-(F, p(\overline{x_n})) = t$ where $occ_\Sigma(a, t) = 1$ and t is ground.
- if $p(\overline{x_n}) = p(\overline{x_n})_0 < 0$, then
  - $cvt^-(F, p(\overline{x_n})) = t$ where $occ_\Sigma(a, t) = |p(\overline{x_n})_0| + 1$ and t is ground.
  - $cvt^+(F, p(\overline{x_n})) = t$ where $occ_\Sigma(a, t) = 1$ and t is ground.
- if $p(\overline{x_n}) \neq p(\overline{x_n})_0$and $p(\overline{x_n})_0 = 0$, then for all $\star \in \{+, -\}$,

$$cvt^\star(F, p(\overline{x_n})) = F(cvt^\star(F, p(\overline{x_n})_1 \downarrow), \cdots, cvt^\star(F, p(\overline{x_n})_n \downarrow))$$

- if $p(\overline{x_n}) \neq p(\overline{x_n})_0$and $p(\overline{x_n})_0 \geq 1$, then
  - $cvt^+(F, p(\overline{x_n})) = g(t, F(cvt^+(F, p(\overline{x_n})_1 \downarrow), \cdots, cvt^+(F, p(\overline{x_n})_n \downarrow))$ where $occ_\Sigma(a, t) = p(\overline{x_n})_0$ and t is ground.
  - $cvt^-(F, p(\overline{x_n})) = F(cvt^-(F, p(\overline{x_n})_1 \downarrow), \cdots, cvt^-(F, p(\overline{x_n})_n \downarrow))$
- if $p(\overline{x_n}) \neq p(\overline{x_n})_0$, and $p(\overline{x_n})_0 < 0$, then
  - $cvt^-(F, p(\overline{x_n})) = g(t, F(cvt^-(F, p(\overline{x_n})_1 \downarrow), \cdots, cvt^-(F, p(\overline{x_n})_n \downarrow))$ where $occ_\Sigma(a, t) = p(\overline{x_n})_0$ and t is ground.
  - $cvt^+(F, p(\overline{x_n})) = F(cvt^+(F, p(\overline{x_n})_1 \downarrow), \cdots, cvt^+(F, p(\overline{x_n})_n \downarrow))$

Intuitively, the n-converter takes a polynomial in n unknowns separates it into $n+1$ variable disjoint subpolynomials. Each of these subpolynomials is assigned to one of the arguments of the second-order variable (except the subpolynomial representing an integer constant) and the n-converter is called recursively on these subpolynomials. The process stops when all the subpolynomials are integers. Example 4.1 illustrates the construction of a term from a polynomial. Example 4.2 & 4.3 construct the n-multiplier and n-counter of the resulting term, respectively.

Example 4.1. Consider the polynomial $p(x, y) = 3 \cdot x^3 + xy - 2 \cdot y^2 - 2$. The positive and negative terms representing this polynomial are as follows:

$$\begin{aligned} cvt^+(F, 3 \cdot x^3 + xy - 2 \cdot y^2 - 2) =& F(F(F(g(g(a, a), g(a, a)), b), g(a, a)), F(b, a)) \\ cvt^-(F, 3 \cdot x^3 + xy - 2 \cdot y^2 - 2) =& g(g(a, a), F(F(F(a, b), a), F(b, g(a, g(a, a)))) \end{aligned}$$

Observe that the n-converter will always produce a flex-rigid unification equation as long as the input polynomial is of the form $p(\overline{x_n}) = p'(\overline{x_n}) + c$ where $c \in \mathbb{Z}$. When $c = 0$, we get a flex-flex unification equation and there is always a solution.

Example 4.2. Consider the term from Example 4.1. The n-multiplier is as follows:
Thus, $mul(F, x, y, cvt^+(F, 3 \cdot x^3 + xy - 2 \cdot y^2 - 2)) = mul(F, x, y, cvt^-(F, 3 \cdot x^3 + xy - 2 \cdot y^2 - 2)) = 1 + x^2 + y$.

Example 4.3. Consider the term from Example 4.1. The n-counter is as follows:

$$\begin{aligned} cnt(F, x, y, a, cvt^+(F, 3 \cdot x^3 + xy - 2 \cdot y^2 - 2)) &= 4 \cdot x^3 + 2 \cdot xy + y^2 \\ cnt(F, x, y, a, cvt^-(F, 3 \cdot x^3 + xy - 2 \cdot y^2 - 2)) &= x^3 + xy + 3 \cdot y^2 + 2 \\ cnt(F, x, y, a, cvt^+(F, p(x, y))) - cnt(F, x, y, a, cvt^-(F, p(x, y))) &= 3x^3 + xy - 2 \cdot y^2 - 2 \end{aligned}$$

Using the operator defined in Definition 4.2, we can transform a polynomial with integer coefficients into a n-SOGU problem. The next definition describes the process:

Definition 4.3. Let $p(\overline{x_n})$ be a polynomial and $F \in \mathcal{V}_f^n$. Then $(\mathcal{U}, F)$ is the n-SOGU problem induced by $p(\overline{x_n})$ where $\mathcal{U} = \{cvt^-(F, p(\overline{x_n})) \stackrel{?}{=}_F cvt^+(F, p(\overline{x_n}))\}$.

The result of this translation is that the n-counter captures the structure of the polynomial and the n-multipliers cancel out.

Lemma 4.1. Let $n \geq 1$, $p(\overline{x_n})$ be a polynomial, and $(\mathcal{U}, F)$ an n-SOGU problem induced by $p(\overline{x_n})$ where $\mathcal{U} = \{cvt^-(F, p(\overline{x_n})) \stackrel{?}{=}_F cvt^+(F, p(\overline{x_n}))\}$. Then

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

A simply corollary of Lemma 4.1 concerns commutativity of unification equations:

Corollary 4.1. Let $n \geq 1$, $p(\overline{x_n})$ be a polynomial, and $(\{s \stackrel{?}{=} t\}, F)$ an n-SOGU problem induced by $p(\overline{x_n})$. Then $-p(\overline{x_n}) = cnt_r(F, \overline{x_n}, a, \{t \stackrel{?}{=} s\}) - cnt_l(F, \overline{x_n}, a, \{t \stackrel{?}{=} s\})$.

Both $p(\overline{x_n})$ and $-p(\overline{x_n})$ have the same roots and the induced unification problem cannot be further reduced without substituting into F, thus the induced unification problem uniquely captures the polynomial $p(\overline{x_n})$. We now prove that the unification condition as introduced in Lemma 3.3 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 4.2. 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 4.2). 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 $p(\overline{x_n}) = 0$.

Using Lemma 4.2, 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 2.1) to finding $h_1, \cdots, h_n \geq 0$ which satisfy the *unification condition*.

Lemma 4.3 (Equalizer Problem)**.** For a given n-SOGU problem, finding $h_1, \cdots, h_n \geq 0$ such that the *unification condition* (Lemma 3.3) holds is undecidable.

Theorem 4.1. There exists $n \geq 1$ such that n-SOGU is undecidable.

We prove Theorem 4.1 by assuming n-SOGU is decidable and using this assumption to show that the Equalizer Problem must be decidable, thus resulting in a contradiction.

In particular, we answer the question posed in Section 1 by proving that first-order variables occurrence does not impact the decidability of second-order unification.

References

[1] 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.

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

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