数理解析研究所講究録
第2051巻 2017年 106-115

正則項上の単一化について

島根大学 総合理工学研究科 岩見 宗弘 (Munehiro Iwami)
Interdisciplinary Faculty of Science and Engineering,
Shimane University
e-mail: munehiro@cis.shimane-u.ac.jp

概要

有限項上の (一階の) 単一化アルゴリズムは，推論規則を用いて形式化され，その停止性，健全性と完全性が多くの文献で示されている．また，正則項上の単一化に対しても，推論規則を与えて形式化したいくつかの先行研究がある．しかしながら，それらの研究において，停止性，健全性と完全性がきちんと示されているものはあまりない．

そこで本論文では，正則項上における単一化の基礎理論を項書換えシステムの枠組みで整理することを目標とする．まず，正則項上の単一化を推論規則を用いて再定式化する．次に，その停止性，健全性と完全性の証明を与える．最後に，これらを用いて単一化可能であるならば，最汎単一化子である正則代入が存在することを示す．

1 はじめに

単一化は自動推論における最も重要な概念の1つである．単一化は導出原理や論理型言語における基本演算であり，関数型言語の型推論，項書換えシステムの合流性の証明や完備化手続き等に応用されている．有限項上の (一階の) 単一化に関しては，推論規則を用いてわかりやすく形式化され，その停止性，健全性と完全性が項書換えシステムの枠組みにおいて示されている [2, 3, 7, 10, 12].

有限項だけを対象とするのではなく，比較的性質のよい無限項である正則項上の単一化に関する研究も行われている．正則項とは，その部分項集合が有限である (無限) 項である．正則項上の単一化に関する最初の研究は文献 [8] で行われている[1]．文献 [6] では，無限木 (項) に関する基礎理論に関する研究がまとめられており，正則項上における単一化も代数的な手法により形式化されている．また，文献 [4, 5] では，正則項上の単一化を論理型言語 Prolog の処理系に実装するために，推論規則により形式化が試みられている．実際，Prolog II では文献 [5] で与えられた手法に基づき，正則項上の単一化が実装されている．現在でも，SWI-Prolog 等の多くの Prolog 処理系において，正則項上における単一化が実現されている．さらに，効率的な正則項上の単一化アルゴリズムも提案され，実装されている [9]．最近，このアルゴリズム ([9]) は，文献 [17] において無限項書換えシステムの強頭部正規化可能性および一般生成性に対する反証手続きに応用されている．

しかしながら，正則項上の単一化は，論理型言語の処理系等に実装されているにも関わらず，その理論的な研究はあまり進んでいるとはいえない．文献 [4] では，正則項上の単一化手続きを形式化するために推論規則が与えられているが，その停止性しか示されていない．文献 [5] では，新しい推論規則が与えられ，その停止性，健全性と完全性も示されているが，それらの証明の詳細はあまり正確には与えられていない．文献 [14] では，文献 [12] の手法に基づいた推論規則が与えられているが，その停止性しか示されていない．また，推論規則の定義が非常に複雑であり，そのために停止性の証明も複雑である．文献 [13] でも，文献 [12] の手法に基づいた正則項上の単一化アル

[1]文献 [16] でも，出現検査を取り除けば正則項上の単一化が実現できると具体例を用いて簡潔に述べられている．

ゴリズムが与えられている．これは，上記の先行研究 [4, 14] よりも推論規則は単純であり，文献 [4] より効率的であることが示されている．これらの研究は，論理型言語の研究として行われており，処理系の実装に重点がおかれている．また，効率性を重視しており，項ではなく DAG が基本的なデータ構造として使われている研究も多く，理論的に整理されつくされているとはいえない．

そこで本論文では，正則項上における単一化の基礎理論を項書換えシステムの枠組みで整理することを目標とする．まず，正則項上の単一化を推論規則を用いて再定式化する．次に，その停止性，健全性と完全性の証明を与える．最後に，これらを用いて単一化可能であるならば，最汎単一化子である正則代入が存在することを示す．

2 準備

本節では，本論文で使用する定義や記法を与える．本節の内容は主に文献 [17] に基づいている．詳細は文献 [2, 3, 6] などを参考にして頂きたい．

関数記号の集合を $\mathcal{F}$, **変数**の加算無限集合を $\mathcal{V}$ とする ($\mathcal{F} \cap \mathcal{V} = \emptyset$ とする). 0 引数の関数記号を**定数**とよび，n 引数の関数記号の集合を $\mathcal{F}_n$ と表す．$\mathbb{N}^+$ を正整数集合とし，正整数の有限列の集合を $\mathbb{N}^{+*}$ と表す．有限列 $p, q \in \mathbb{N}^{+*}$ の連結を $p.q$ と表す．部分関数 $t : \mathbb{N}^{+*} \to \mathcal{F} \cup \mathcal{V}$ に対して，以下の条件を満たすものを $\mathcal{F}, \mathcal{V}$ 上の**項**とよぶ: (1) $t(\epsilon) \in \mathcal{F} \cup \mathcal{V}$; (2) 任意の $p \in \mathbb{N}^{+*}$ に対して，$t(p.i) \in \mathcal{F} \cup \mathcal{V} \iff t(p) \in \mathcal{F}_n$ かつ $1 \leq i \leq n$. ここで，ϵ は空列を表す．$\mathcal{F}, \mathcal{V}$ 上の項の集合を $T_{inf}(\mathcal{F}, \mathcal{V})$ と表す．

項 $t \in T_{inf}(\mathcal{F}, \mathcal{V})$ の定義域 $Pos(t) = \{p \in \mathbb{N}^{+*} \mid t(p) \in \mathcal{F} \cup \mathcal{V}\}$ の要素を t における**位置**とよぶ．$p \notin Pos(t)$ なる $p \in \mathbb{N}^{+*}$ に対して，$t(p) = \bot$ と定義する．ただし，$\bot$ は $\bot \notin \mathcal{F} \cup \mathcal{V}$ なる定数とする．このとき，$s = t \iff \forall p \in Pos(s).\ s(p) = t(p)$ が成り立つ．項 t に出現する変数集合を $\mathcal{V}(t)$ と表す．位置集合が有限集合であるとき，項 t を**有限**であるという．有限項の集合を $T_{fin}(\mathcal{F}, \mathcal{V})$ と表す．有限項 t の**サイズ**を位置集合 $Pos(t)$ の要素の個数とし，$|\,t\,|$ と表す．有限項と区別するときには，項を**無限項**とよぶこともある．位置 $p \in Pos(t)$ における項 t の**部分項** $t\,|_p$ を $t_p(q) = t(p.q)$ により定義する．関数記号 $f \in \mathcal{F}_n$ と項 $t_1, \ldots, t_n \in T_{inf}(\mathcal{F}, \mathcal{V})$ に対して，次の条件により定義される項 t を $f(t_1, \ldots, t_n)$ と表す: (1) $t(\epsilon) = f$; (2) $t(i.p) = t_i(p)$ $(1 \leq i \leq n, p \in \mathbb{N}^{+*})$. 位置集合上の**接頭辞順序**を $p \preceq q \Leftrightarrow \exists r \in \mathbb{N}^{+*}.\ q = p.r$ により定義する．ある位置 q および正整数 $i < j$ が存在して，$q.i \preceq p_1, q.j \preceq p_2$ を満たすとき，p_1 は p_2 の**左に位置する**という．

写像 $\sigma : \mathcal{V} \to T_{inf}(\mathcal{F}, \mathcal{V})$ を**代入**とよぶ．代入 σ を項 $t \in T_{inf}(\mathcal{F}, \mathcal{V})$ に適用した結果 $\sigma(t)$ を次のように定義する: $\sigma(t)(p) = \sigma(t(p_0))(p_1)$ ($p = p_0.p_1$ かつ $t(p_0) \in \mathcal{V}$ を満たす $p_0, p_1 \in \mathbb{N}^{+*}$ が存在する場合); $\sigma(t)(p) = t(p)$ (それ以外の場合). $\sigma(t)$ は t に出現する変数 $x \in \mathcal{V}$ を $\sigma(x)$ により置き換えられた結果得られる項を表す．代入 σ の定義域 $\{x \in \mathcal{V} \mid \sigma(x) \neq x\}$ を $dom(\sigma)$ と表す．$dom(\sigma) = \{x_1, \ldots, x_n\}$ かつ $\sigma(x_i) = u_i$ である代入を $\{x_1 := u_1, \ldots, x_n := u_n\}$ とも表す．$dom(\sigma)$ が有限かつ任意の $x \in dom(\sigma)$ に対して $\sigma(x) \in T_{fin}(\mathcal{F}, \mathcal{V})$ を満たすとき，σ を**有限代入**とよぶ．$\sigma(t)$ を $t\sigma$ とも表す．代入の合成を $\circ$ により表す．すなわち，任意の変数 $x \in \mathcal{V}$ に対して，$(\sigma \circ \tau)(x) = \sigma(\tau(x))$.

$\Box \notin \mathcal{F} \cup \mathcal{V}$ なる定数 $\Box$(**ホール**とよぶ) を考える．$C \in T_{inf}(\mathcal{F} \cup \{\Box\}, \mathcal{V})$ に対して，$\{p \in \mathbb{N}^{+*} \mid C(p) = \Box\}$ が有限集合となるものを**文脈**とよぶ．文脈 C が $\{p \in \mathbb{N}^{+*} \mid C(p) = \Box\} = \{p_1, \ldots, p_n\}$ かつ任意の $i < j$ に対して p_i が p_j より左に位置するとき，C を $C[, \ldots,]_{p_1, \ldots, p_n}$ と表す．文脈 $C[, \ldots,]_{p_1, \ldots, p_n}$ と項 $t_1, \ldots, t_n \in T_{inf}(\mathcal{F}, \mathcal{V})$ に対して，項 $t = C[t_1, \ldots, t_n]_{p_1, \ldots, p_n}$ を次のように定義する: $\exists i, q.\ p = p_i.q$ のとき，$t(p) = t_i(q)$; それ以外のとき，$t(p) = C(p)$. すなわち，$C[t_1, \ldots, t_n]_{p_1, \ldots, p_n}$ は文脈 C に出現するホールを左から順に $t_1, \ldots, t_n$ に置き換えて得られる項を表す．また，ホールの出現が 1 つだけの文脈を $C[\,]_p$ と表す．

等式を $s \approx t$ と表す．ここでは $s,t \in T_{inf}(\mathcal{F},\mathcal{V})$ とする．等式 $s \approx t$ の左辺は s，右辺は t をさし，等式の左辺と右辺は区別する．代入 σ を等式集合 $\{x \approx \sigma(x) \mid x \in dom(\sigma)\}$ と同一視することがある．等式集合 E に出現する変数の集合を次のように定義する: $\mathcal{V}(E) = \bigcup_{(s\approx t)\in E}(\mathcal{V}(s) \cup \mathcal{V}(t))$.

等式集合 E の**単一化子**とは，任意の等式 $s \approx t \in E$ に対して，$s\sigma = t\sigma$ となる代入 σ をいう．等式集合 E の単一化子の集合を $\mathsf{Unif}_{inf}(E)$ と表す．$\mathsf{Unif}_{inf}(E) \neq \emptyset$ のとき，等式集合 E は**単一化可能**であるという．単一化可能性を判定する問題を，**単一化問題**という．代入上の擬順序 $\lesssim$ を $\theta \lesssim \eta \iff$ ある代入 ρ が存在して $\eta = \rho \circ \theta$ により定めるとき，単一化子のうち擬順序 $\lesssim$ に関して極小となる単一化子を**最汎単一化子**とよぶ．等式集合 E が単一化可能であるときには，最汎単一化子が存在し，同値関係 $\gtrsim \cap \lesssim$ に関して一意に定まる [6]．等式集合 E と代入 σ に対して，$E\sigma = \{s\sigma \approx t\sigma \mid s \approx t \in E\}$ と定義する．ここで，等式集合 E が**有限項上で単一化可能である**とは，有限代入であるような $\mathsf{Unif}_{inf}(E)$ の要素があるときをいう．特に区別する場合には，単一化可能であることを**無限項上で単一化可能である**ともいう．

3 正則項上の単一化と再帰式表現

本節では正則項上の単一化と再帰式表現の定義やそれらの基本的性質について述べる．

定義 1 *(正則項 [6])* 項 t が**正則** *(regular または rational)* であるとは，t の部分項集合が有限集合であるときをいう．

有限項は明らかに正則である．代入 θ が次の 2 つ条件を満たすとき**正則代入**とよぶ: $dom(\theta)$ が有限集合，かつ，任意の変数 $x \in dom(\theta)$ に対して $\theta(x)$ が正則項である．正則項は正則代入の適用に関して閉じており，代入の正則性は関数合成 $\circ$ により保存される．

命題 2 *(正則項の単一化 [6])* 正則項の無限項上での単一化問題は決定可能であり，単一化可能であるときに最汎単一化子を求めるアルゴリズムが存在する．また，最汎単一化子は正則代入となる．

以下では,「単一化問題を解く」とは単一化可能性を判定するとともに単一化可能であるときに最汎単一化子を求めることとする．また,「単一化手続き」とは単一化問題を解く手続きを指すものとする．

定義 3 *(再帰式表現 [17])* 有限代入 $\{x_1 := t_1, \ldots, x_n := t_n\}$ が以下の条件を満たすとき，これを**再帰式表現**とよび，$[x_1 := t_1, \ldots, x_n := t_n]$ と表す:

$$\neg\exists i_1, \ldots, i_k \in \{1, \ldots, n\}.\ (\forall 1 \leq j < k.\ t_{i_j} = x_{i_{j+1}}) \wedge t_{i_k} = x_{i_1} \quad (1)$$

定義 4 *(再帰式表現の解 [17])* $\theta = [x_1 := t_1, \ldots, x_n := t_n]$ を再帰式表現とし，任意の $1 \leq i \leq n$ に対して，$t_i = C_i[x_{i_1}, \ldots, x_{i_{k_i}}]$ とする．ただし，ここで，文脈 C_i には変数が出現しないとする．このとき，項 $\theta^\star(x_1), \ldots, \theta^\star(x_n)$ を以下のように相互再帰的に定義する．$\theta^\star(x_i)(p) =$
$$\begin{cases} C_i(p) & (p \in Pos(C_i) \wedge C_i\,|_p \neq \Box) \\ \theta^\star(x_{i_j})(q) & (\exists j, q.\ p = p_{i_j}.q) \end{cases}$$
項 $\theta^\star(x_1), \ldots, \theta^\star(x_n)$ を**再帰式表現の解**とよぶ．

再帰式表現の条件 (1) より，$\theta^\star(x_i)$ は定義として成立している (well-defined である) こと，また，$C_i\,|_{p_{i_j}} = \Box$ より項 $\theta^\star(x_1), \ldots, \theta^\star(x_n)$ は一意に定まる．さらに，定義より明らかに，$\theta^\star$ は $dom(\theta^\star) = \{x_1, \ldots, x_n\}$ なる正則代入である．

また，任意の再帰式表現 θ に対して，$\theta^\star = \delta^\star$ となる再帰式表現 δ で次の条件を満たすものが存在する: $\forall x \in dom(\delta).\ \delta(x) \notin dom(\delta)$.

例 5 *(再帰式表現 [17])* 有限代入 $\theta = \{x := \mathsf{f}(x), y := x\}$ は再帰式表現であり，有限代入 $\theta = \{x := \mathsf{f}(x), y := z, z := y\}$ は再帰式表現ではない．また，任意の $n \geq 0$ について $\theta^\star(x)(1^n) = \theta^\star(y)(1^n) = \mathsf{f}$ であるから，$\theta^\star(x) = \theta^\star(y) = \mathsf{f}(\mathsf{f}(\cdots))$ となる．$\delta = [x := \mathsf{f}(x), y := \mathsf{f}(x)]$ をとると，$\delta^\star = \theta^\star$ かつ $\forall x \in dom(\delta).\ \delta(x) \notin dom(\delta)$ が成立する．再帰式表現 $\eta = [x := \mathsf{f}(y), y := \mathsf{g}(x)]$ に対して，$\eta^\star(x) = \mathsf{f}(\mathsf{g}(\mathsf{f}(\mathsf{g}(\cdots)))), \eta^\star(y) = \mathsf{g}(\mathsf{f}(\mathsf{g}(\mathsf{f}(\cdots))))$ となる．

命題 6 *(正則項と再帰式表現 [5, 6])* 再帰式表現の解は正則項である．また，任意の正則項はある再帰式表現の解となる．

すなわち，任意の正則項は，再帰式表現 θ を用いて，$\theta^\star(x)$ の形で表すことができる．これを**正則項の再帰式表現**とよぶ．ここで，正則項の再帰式表現は必ずしも一通りとは限らない．

実際，次の補題は，再帰式表現 θ の表す正則代入 $\theta^\star$ は再帰式表現に対応する等式集合の最汎単一化子となっていることを示す．

以下では文脈から明らかな場合には，再帰式表現 $\theta = [x_1 := t_1, \ldots, x_n := t_n]$ を文脈に応じてそれに対応する等式集合 $\{x_1 \approx t_1, \ldots, x_n \approx t_n\}$ とみなす．

補題 7 *(再帰式表現の最汎単一化子 [17])* $\theta = [x_1 := t_1, \ldots, x_n := t_n]$ を再帰式表現とする．このとき，正則代入 $\theta^\star$ は再帰式表現 θ に対応する等式集合 $\{x_1 \approx t_1, \ldots, x_n \approx t_n\}$ の最汎単一化子である．

次に，再帰式表現の表す正則代入はべき等性を満たすことを示す．

補題 8 S が再帰式表現であるとき，$S^\star = S^\star \circ S^\star$.

(証明) $S = [x_1 := t_1, \ldots, x_n := t_n]$ とする．例 5 の直前に述べた事実から，一般性を失うことなく，$t_1, \ldots, t_n \notin \{x_1, \ldots, x_n\} = dom(S)$ としてよい．任意の i $(1 \leq i \leq n)$ に対して，$t_i = C_i[x_{i_1}, \ldots, x_{i_{k_i}}]_{p_{i_1}, \ldots, p_{i_{k_i}}}$ とおく．仮定 $t_1, \ldots, t_n \notin \{x_1, \ldots, x_n\}$ から，$C_i \neq \square$ であることに注意する．補題 7 から，正則代入 $S^\star$ は S の最汎単一化子である．ここで，一般性を失うことなく $\{x_1, \ldots, x_n\} \cap (\bigcup_{1 \leq i \leq n} \mathcal{V}(S^\star(x_i))) = \emptyset$ と仮定することができる．この理由を以下に示す．いま，$\{x_1, \ldots, x_n\} \cup (\bigcup_{1 \leq i \leq n} \mathcal{V}(S^\star(x_i)))$ に含まれない変数 $y_1, \ldots, y_n$ をとる．$S' = \{x_1 := y_1, \ldots, x_n := y_n\} \circ S^\star$ とすると，$\{x_1, \ldots, x_n\} \cap (\bigcup_{1 \leq i \leq n} \mathcal{V}(S'(x_i))) = \emptyset$ である．このとき，$S^\star \lesssim S'$．一方，$S^\star = \{y_1 := x_1, \ldots, y_n := x_n\} \circ S'$ より，$S^\star \gtrsim S'$．よって，$S^\star (\gtrsim \cap \lesssim) S'$．すなわち，$S'$ は $S^\star$ と同値である S の最汎単一化子であるからである．$\{x_1, \ldots, x_n\} \cap (\bigcup_{1 \leq i \leq n} \mathcal{V}(S^\star(x_i))) = \emptyset$ であれば，任意の $x \in \mathcal{V}$ に対して，$S^\star(x) = S^\star \circ S^\star(x)$ が成立する．

1. $x \in \{x_1, \ldots, x_n\}$; $x = x_j$ とする．$t_j = C_j[x_{j_1}, \ldots, x_{j_{k_j}}]_{p_{j_1}, \ldots, p_{j_{k_j}}}$ であるから，$S^\star(t_j) = C_j[S^\star(x_{j_1}), \ldots, S^\star(x_{j_{k_j}})]_{p_{j_1}, \ldots, p_{j_{k_j}}}$ かつ $S^\star(S^\star(t_j)) = C_j[S^\star(S^\star(x_{j_1})), \ldots, S^\star(S^\star(x_{j_{k_j}}))]_{p_{j_1}, \ldots, p_{j_{k_j}}}$．いま，$\{x_1, \ldots, x_n\} \cap (\bigcup_{1 \leq i \leq n} \mathcal{V}(S^\star(x_i))) = \emptyset$ より，$S^\star(x_{j_m}) = S^\star(S^\star(x_{j_m}))$ $(m = j_1, \ldots, j_{k_j})$．すなわち，$S^\star(x_j) = S^\star(S^\star(x_j))$．よって，$S^\star(x) = S^\star(t_j) = S^\star(S^\star(t_j)) = S^\star \circ S^\star(x)$.
2. $x \notin \{x_1, \ldots, x_n\}$; $S^\star(x) = x = S^\star \circ S^\star(x)$. $\square$

次の補題は次節において重要な役割を果たす．

補題 9 S を等式集合とする．S が再帰式表現の条件を満たすならば，任意の $\sigma \in \mathsf{Unif}_{inf}(S)$ に対して，$\sigma = \sigma \circ S^\star$.

(証明) $S = \{x_1 \approx t_1, \ldots, x_n \approx t_n\}$ とする．仮定より，$[x_1 := t_1, \ldots, x_n := t_n]$ は再帰式表現である．補題 7 より，正則代入 $S^\star$ は上記の再帰式表現に対応する等式集合 S の最汎単一化子である．よって，任意の $\sigma \in \mathsf{Unif}_{inf}(S)$ に対して，$\sigma = \rho \circ S^\star$ を満たす代入 ρ が存在する．

ここで，任意の $x \in \mathcal{V}$ に対して，$\sigma(x) = \sigma \circ S^\star(x)$ が成立することを示す．

1. $x \in \{x_1, \ldots, x_n\}$; $x = x_k$ とする．

このとき，$\sigma(x) = \sigma(t_k) = \rho(S^\star(t_k))$ かつ $\sigma \circ S^\star(x) = \sigma(S^\star(t_k)) = \rho(S^\star(S^\star(t_k)))$. 補題 8 より，$S^\star = S^\star \circ S^\star$. よって，$\sigma(x) = \sigma \circ S^\star(x)$ が成り立つ．

2. $x \notin \{x_1, \ldots, x_n\}$; $S^\star(x) = x$ より，$\sigma(x) = \sigma \circ S^\star(x)$. □

4 正則項上の単一化に対する推論規則

本節では正則項上の単一化に対する推論規則を与える．また，これらの規則を用いて，正則項における単一化を再定式化する．さらに，その停止性，健全性と完全性を示す．

次の正則項上の単一化に対する推論規則の定義は文献 [10, 4] に基づいている．ただし，以下では出現検査を行う Check∗ 規則 ([10]) は考えない．文献 [10] における有限項上の単一化に対する推論規則と同様な規則は文献 [7] においても与えられている．

以下では，解を持たない特別な単一化問題を $\perp$ で表す．

定義 10 *(正則項上の単一化に対する推論規則)* P を等式集合とする．

1. Delete : $\{s \approx s\} \uplus P \Longrightarrow P$,

2. Decompose : $f \in \mathcal{F}_n$ のとき，$\{f(s_1, \ldots, s_n) \approx f(t_1, \ldots, t_n)\} \uplus P \Longrightarrow \{s_1 \approx t_1, \ldots, s_n \approx t_n\} \cup P$,

3. Orient : $s \notin \mathcal{V}$ のとき，$\{s \approx x\} \uplus P \Longrightarrow \{x \approx s\} \cup P$,

4. Coalesce : $x \in \mathcal{V}(P)$ のとき，$\{x \approx y\} \uplus P \Longrightarrow \{x := y\}(P) \cup \{x \approx y\}$,

5. Merge : $s, t \notin \mathcal{V}$ かつ $\mid s \mid \leq \mid t \mid$ のとき, $\{x \approx s, x \approx t\} \uplus P \Longrightarrow \{x \approx s, s \approx t\} \cup P$,

6. Clash : $f, g \in \mathcal{F}$ かつ $f \neq g$ のとき，$\{f(s_1, \ldots, s_m) \approx g(t_1, \ldots, t_n)\} \uplus P \Longrightarrow \perp$.

以降では，正則項における単一化手続き RUnify を推論規則で次のように定義する．この関数は最汎単一化子が存在すればそれを返し，そうでなければ失敗を返す．ここで，S は等式集合とする．以下では，定義 10 の推論規則を任意に 1 回使用した推論を $\rightsquigarrow$ により表す．$\rightsquigarrow$ の反射推移的閉包を $\rightsquigarrow^*$ で表す．S が $\rightsquigarrow$ に関する**正規形**であるとは，$S \rightsquigarrow T$ を満たす T が存在しないときをいう．

定義 11 *(正則項における単一化手続き)*

RUnify(S) = *while* $\exists T$ *s.t.* $S \rightsquigarrow T$ *do* $S := T$;
if S が再帰式表現の条件を満たす
then return $S^\star$ *else* 失敗;

上記の手続きは非決定的であることに注意する．すなわち，2 つ以上の適用可能な規則が存在するとき，たとえば，$S \rightsquigarrow T_1$ かつ $S \rightsquigarrow T_2$ のとき，手続きは任意の規則を選んでよい．このように，RUnify の停止性は $\rightsquigarrow$ の停止性に依存する．

以下では，まず RUnify の停止性を示す．このために必要な補題をいくつか示す．

補題 12 任意の等式集合 P に対して，$P \rightsquigarrow^* S$ かつ S は推論 $\rightsquigarrow$ に関する正規形であるとする．このとき，S が再帰式表現の条件を満たすか否かを判定することは決定可能である．

(証明) いま，一般性を失うことなく $S=\{x_1 \approx t_1,\ldots,x_n \approx t_n\}$ とすることができる．このとき，$t_1,\ldots,t_n$ はすべて有限項であることに注意する．

1. S が有限代入の条件を満たすことを示す．

最初に，ある変数 $x \in \{x_1,\ldots,x_n\}$ に対して，$x \approx t_i, x \approx t_j \in S$ かつ $t_i \neq t_j$ を満たす等式が存在すると仮定する．

(1) $t_i, t_j \notin \mathcal{V}$ かつ $|t_i| \leq |t_j|$; このとき，$S=\{x \approx t_i, x \approx t_j\} \uplus S'$ とおける．S には Merge 規則が適用でき，S が正規形であることに矛盾する．

(2) $t_i \in \mathcal{V}, x \in \mathcal{V}(\{x \approx t_j\} \cup S'')$ かつ $x \neq t_i$; いま，$t_i = y$ とする．このとき，$S=\{x \approx y\} \uplus (\{x \approx t_j\} \cup S'')$ とおける．S には Coalesce 規則が適用でき，S が正規形であることに矛盾する．よって，$x_1,\ldots,x_n$ は相異なる変数である．

次に，仮定から，$x_1 \neq t_1,\ldots,x_n \neq t_n$. よって，$S(x_i) \neq x_i$ $(i=1,\ldots,n)$. すなわち，$dom(S)=\{x_1,\ldots,x_n\}$. さらに，$t_1,\ldots,t_n$ は有限項より，S は有限代入の条件を満たす．

2. S が再帰式表現の条件を満たすことを示す．

上記の 1 より，この等式集合 S に対応する有限代入は $S=\{x_1 := t_1,\ldots,x_n := t_n\}$ である．仮定から，$x_1 \neq t_1,\ldots,x_n \neq t_n$ が成り立つ．このとき，S に出現する変数 x_i と有限項 t_j が等しいかを，有限回確認すればよい．したがって，S が再帰式表現であるか否かを判定することは決定可能である．　□

定義 13 *([4])* 等式の *(*有限*)* 集合 P に対して，P の重みを次のように定義する*:*

$\| P \| = \sum_{(s \approx t) \in P} max\{|s|,|t|\}$

補題 14 すべての推論 $S \rightsquigarrow T$ に対して $\| S \| \geq \| T \|$ が成り立つ．また，Decompose 規則が適用されている推論 $S \rightsquigarrow T$ に対して，$\| S \| > \| T \|$ が成り立つ．

(証明) $\rightsquigarrow$ に適用されている推論規則の場合分けにより示す．

1. Delete 規則;

$\| \{s \approx s\} \uplus P \| = max\{|s|,|s|\} + \| P \| > \| P \|$.

2. Decompose 規則;

$\| \{f(s_1,\ldots,s_n) \approx f(t_1,\ldots,t_n)\} \uplus P \| = max\{|f(s_1,\ldots,s_n)|,|f(t_1,\ldots,t_n)|\} + \| P \| > max\{|s_1|,|t_1|\} + \ldots + max\{|s_n|,|t_n|\} + \| P \| = \| \{s_1 \approx t_1,\ldots,s_n \approx t_n\} \cup P \|$.

3. Orient 規則;

$\| \{s \approx x\} \uplus P \| = max\{|s|,|x|\} + \| P \| = \| \{x \approx s\} \uplus P \|$.

4. Coalesce 規則;

$\| \{x \approx y\} \uplus P \| = max\{|x|,|y|\} + \| P \| = \| \{x := y\}(P) \| + max\{|x|,|y|\} = \| \{x := y\}(P) \cup \{x \approx y\} \|$.

5. Merge 規則;

$\| \{x \approx s, x \approx t\} \uplus P \| = max\{|x|,|s|\} + max\{|x|,|t|\} + \| P \| = |s| + |t| + \| P \| = max\{|x|,|s|\} + max\{|s|,|t|\} + \| P \| = \| \{x \approx s, s \approx t\} \uplus P \|$.　□

次の補題の証明は文献 [4] の手法に基づいている．

補題 15 *(*RUnify(P) の停止性*)* 任意の等式集合 P に対して，RUnify(P) は停止する．

(証明) 等式集合 P_0 に対して，RUnify(P_0) が停止しないと仮定する．定義 11 と補題 12 から，次のような P_0 から始まる無限推論列が存在すると仮定して，矛盾を導く．

$$P_0 \rightsquigarrow P_1 \rightsquigarrow \ldots \rightsquigarrow P_i \rightsquigarrow P_{i+1} \rightsquigarrow \ldots$$

最初に，もし Decompose 規則が適用されなければ，このような推論列は有限であることを示す．推論列の最初の等式集合 P_0 中の $s \approx s$ または $x \approx y$ の形の等式の個数を n とする．また，P_0 中の $x \approx t$ または $t \approx x$ の形の等式の個数を m とする．ここで，P_0 は有限集合であることに注意する．このとき，Delete, Coalesce 規則の適用回数は n 以下である．また，Orient, Merge 規則の適用回数は m 以下である．よって，Decompose 規則が適用されなければ，推論列は有限である．

定義 13 の重みにより，上記の無限推論列から次のような自然数の無限列が得られる:

$$\| P_0 \|, \| P_1 \|, \ldots, \| P_i \|, \| P_{i+1} \|, \ldots$$

補題 14 から，すべての推論に対して $\| P_i \| \geq \| P_{i+1} \|$ が成り立ち，かつ，Decompose 規則が適用されている推論に対して，$\| P_i \| > \| P_{i+1} \|$ が成り立つ．P_0 からはじまる無限推論列が存在すると仮定しているので，Decompose 規則は無限回適用されている．このとき，次のような自然数の無限減少列 $\| P_{i_1} \| > \| P_{i_2} \| > \| P_{i_3} \| > \ldots$ が得られる．よって矛盾する． □

次の補題は推論過程において，単一化子の集合が保存されることを示す．これは RUnify の健全性と完全性の証明に必要である．

補題 16 S, T を等式集合とする．このとき，$S \rightsquigarrow T$ かつ $T \neq \perp$ ならば，$\mathsf{Unif}_{inf}(S) = \mathsf{Unif}_{inf}(T)$．

(証明) (1) Delete, Decompose, Orient 規則のときは自明である．

(2) Coalesce 規則のとき;

$\theta = \{x \approx y\}$ とする．θ は再帰式表現の条件を満たすから，補題 7 より $\theta^\star = \{x := y\}$．このとき，補題 9 より，任意の $\sigma \in \mathsf{Unif}_{inf}(\theta)$ に対して，$\sigma = \sigma \circ \theta^\star$．

$$\begin{aligned} & \sigma \in \mathsf{Unif}_{inf}(\{x \approx y\} \uplus P) \\ \Leftrightarrow\ & \sigma(x) = \sigma(y) \land \sigma \in \mathsf{Unif}_{inf}(P) \\ \Leftrightarrow\ & \sigma \circ \theta^\star \in \mathsf{Unif}_{inf}(P) \land \sigma(x) = \sigma(y) \\ \Leftrightarrow\ & \sigma \in \mathsf{Unif}_{inf}(\theta^\star(P)) \land \sigma(x) = \sigma(y) \\ \Leftrightarrow\ & \sigma \in \mathsf{Unif}_{inf}(\theta^\star(P) \cup \{x \approx y\}) \end{aligned}$$

(3) Merge 規則のとき;

$$\begin{aligned} & \sigma \in \mathsf{Unif}_{inf}(\{x \approx s, x \approx t\} \uplus P) \\ \Leftrightarrow\ & \sigma(x) = \sigma(s) \land \sigma(x) = \sigma(t) \land \sigma \in \mathsf{Unif}_{inf}(P) \\ \Leftrightarrow\ & \sigma(x) = \sigma(s) \land \sigma(s) = \sigma(t) \land \sigma \in \mathsf{Unif}_{inf}(P) \\ \Leftrightarrow\ & \sigma \in \mathsf{Unif}_{inf}(\{x \approx s, s \approx t\} \cup P) \end{aligned}$$

□

文献 [7] では，有限項上の単一化に対して停止性の証明が簡潔に述べられている．また，文献 [10] では，有限項上の単一化に対して健全性と完全性の証明が簡潔に述べられている．さらに，文献 [2, 3] では，有限項上の単一化に対して停止性，健全性と完全性の証明が詳しく与えられている．文献 [4] では，正則項上の単一化に対して停止性は証明されているが，健全性と完全性の証明は与えられていない．一方，文献 [5] では，正則項上の単一化に対して停止性，健全性や完全性の証明が与えられているが，その証明は十分ではない．以下では，本稿で示した補題を使用し，正則項上の単一化に関する健全性と完全性の詳しい証明を与える．

最初に，RUnify の健全性を示す．

補題 17 *(RUnify の健全性)* RUnify(P) が正則代入 σ を返すならば，σ は P の最汎単一化子である．

(証明) 仮定より，$P \rightsquigarrow P_1 \rightsquigarrow \ldots \rightsquigarrow P_n$ が存在する．このとき，P_n は再帰式表現の条件を満たす．よって，補題 7 から，$P_n^\star = \sigma$ が成立する．σ は再帰式表現に対応する等式集合 P_n の最汎単一化子である，すなわち，$\sigma \in \mathsf{Unif}_{inf}(P_n)$．補題 16 より，$\sigma \in \mathsf{Unif}_{inf}(P)$．一方，任意の $\theta \in \mathsf{Unif}_{inf}(P)$ に対して，補題 16 より，$\theta \in \mathsf{Unif}_{inf}(P_n)$．いま，$\sigma$ は P_n の最汎単一化子より，$\theta = \rho \circ \sigma$ を満たす代入 ρ が存在する．したがって，σ は P の最汎単一化子である． □

次に，RUnify の完全性を証明する．

補題 18 *(RUnify の完全性)* P が単一化可能ならば, RUnify(P) は失敗しない．

(証明) P が単一化可能であり，かつ，RUnify(P) が失敗すると仮定する．このとき，補題 15 から，有限の推論列 $P \rightsquigarrow P_1 \rightsquigarrow \ldots \rightsquigarrow P_n$ が存在し，P_n が推論 $\rightsquigarrow$ の正規形であり，かつ，P_n が再帰式表現の条件を満たさない．

1. $P_n = \bot$; P は単一化可能であるから，$\mathsf{Unif}_{inf}(P) \neq \emptyset$．補題 16 より，$\mathsf{Unif}_{inf}(P_n) \neq \emptyset$．いま，$\bot$ は解をもたない特別な単一化問題より，$\mathsf{Unif}_{inf}(P_n) = \emptyset$．よって，矛盾する．

2. $P_n \neq \bot$; $P_n = \{x_1 \approx t_1, \ldots, x_n \approx t_n\}$ とする．いま，P_n が再帰式表現の条件を満たさないから，$\exists i_1, \ldots, i_k \in \{1, \ldots, n\}.\ (\forall 1 \leq j < k.\ t_{i_j} = x_{i_{j+1}}) \land t_{i_k} = x_{i_1}$．すなわち，$t_{i_1} = x_{i_2}, t_{i_2} = x_{i_3}, \ldots, t_{i_{k-1}} = x_{i_k}, t_{i_k} = x_{i_1}$ が成り立つ．このとき，$P_n \supseteq \{x_{i_1} \approx x_{i_2}, x_{i_2} \approx x_{i_3}, \ldots, x_{i_{k-1}} \approx x_{i_k}, x_{i_k} \approx x_{i_1}\}$ より，P_n に対して，Coalesce 規則が適用できる．よって，P_n が $\rightsquigarrow$ の正規形であることに矛盾する． □

最後に，停止性，健全性と完全性から本論文の主定理を示す．

定理 19 P が単一化可能ならば，P は最汎単一化子である正則代入をもつ．

(証明) 補題 18 より，RUnify(P) は失敗しない．補題 15 より，RUnify(P) は停止する．よって，RUnify(P) は正則代入 σ を返す．補題 17 より，σ は P の最汎単一化子である． □

以下では，正則項上の単一化の例をいくつか与える．

例 20 いま，$E = \{\mathsf{f}(x,x) \approx \mathsf{f}(y,z)\}$ とする．

$E = \{\mathsf{f}(x,x) \approx \mathsf{f}(y,z)\} \rightsquigarrow_{\mathsf{Dec}} \{x \approx y, x \approx z\} \rightsquigarrow_{\mathsf{Coal}} \{x \approx y, y \approx z\} \rightsquigarrow_{\mathsf{Coal}} \{x \approx z, y \approx z\}$．このとき，$\{x \approx z, y \approx z\}$ が再帰式表現の条件を満たすから，RUnify(E) は正則代入 $\sigma = \{x := z, y := z\}$ を返す．補題 *17* より，σ は E の最汎単一化子である．

例 21 $E_0 = \{\mathsf{f}(x,x) \approx \mathsf{f}(z,\mathsf{g}(y))\} \rightsquigarrow_{\mathsf{Dec}} \{x \approx z, x \approx \mathsf{g}(y)\} \rightsquigarrow_{\mathsf{Merge}} \{x \approx z, z \approx \mathsf{g}(y)\} = E_2$．このとき，$E_2$ は再帰式表現であるから，RUnify(E_0) は正則代入として $E_2^\star = \{x := \mathsf{g}(y), z := \mathsf{g}(y)\}$ を返す．このとき，補題 *17* より，$E_2^\star$ は E_0 の最汎単一化子である．

例 22 *([2])* この例は有限項上の単一化では出現検査で失敗するが，正則項上の単一化では最汎単一化子として正則代入が求まる．

$$
\begin{array}{ll}
E_0 = & \{\mathsf{f}(x,x) \approx \mathsf{f}(y,\mathsf{g}(y))\} \\
\rightsquigarrow_{\mathsf{Dec}} & \{x \approx y, x \approx \mathsf{g}(y)\} = E_1 \\
\rightsquigarrow_{\mathsf{Coal}} & \{x \approx y, y \approx \mathsf{g}(y)\} = E_2
\end{array}
$$

このとき，等式 $x \approx y, x \approx \mathsf{g}(y) \in E_1$ には Merge 規則は適用できない．もし Coalesce 規則がなければ再帰式表現 E_2 を得ることはできない．よって，再帰式表現を得るためには Merge 規則だけでは不足であり，Coalesce 規則が必要である．E_2 は再帰式表現であるから，RUnify(E_0) は正則代入として $E_2^\star = \{x := \mathsf{g}^\omega, y := \mathsf{g}^\omega\}$ を返す．補題 *17* より，$E_2^\star$ は E_0 の最汎単一化子である．$E_2^\star(\mathsf{f}(x,x)) = \mathsf{f}(\mathsf{g}^\omega, \mathsf{g}^\omega) = \mathsf{f}(\mathsf{g}^\omega, \mathsf{g}(\mathsf{g}^\omega)) = E_2^\star(\mathsf{f}(y,\mathsf{g}(y)))$ より，実際に $E_2^\star$ は E の単一化子である．

5 むすび

本節ではまとめと今後の課題について述べる．

本論文では，正則項，単一化，再帰式表現の概念とそれらの基本的性質について述べた．また，正則項上における単一化を推論規則を用いて再定式化した．さらに，その停止性，健全性と完全性を示した．最後に，これらを用いて単一化可能であるならば，最汎単一化子である正則代入が存在することを示した．

今後の課題は，次の通りである．(i) 単一化の概念を拡張した一意半単一化に関する研究が行われている [11, 15]．有限項上の一意半単一化は文献 [1] において推論規則で形式化され，その停止性について議論されている．そこで，一意半単一化を正則項上へ拡張し，推論規則で形式化し，停止性，健全性と完全性を示す．この拡張により，文献 [17] における反証手続きを改良することを試みる．(ii) 交換律や結合律等の等式を法とした単一化 ([2, 3, 10]) を正則項上に拡張し，推論規則を用いて形式化し，それらの理論的な性質を示す．

参考文献

[1] Aoto, T. and Iwami, M.: Termination of Rule-Based Calculi for Uniform Semi-Unification, *Proc. the 7th International Conf. on Language and Automata Theory and Applications (LATA) 2013*, LNCS, Vol. 7810, Springer-Verlag, 2013, pp. 56–67.

[2] Baader, F. and Nipkow, T.: *Term Rewriting and All That*, Cambridge University Press, 1998.

[3] Baader, F. and Snyder, W.: *Unification Theory, pp.445–533, in Handbook of Automated Reasoning, vol.I*, Elsevier, 2001.

[4] Colmerauer, A.: *Prolog and Infinite Trees, pp.231–251, in Logic Programming*, Academic Press, 1982.

[5] Colmerauer, A.: Equations and inequations on finite and infinite trees, *Proc. of the International Conf. on Fifth Generation Computer Systems*, 1984, pp. 85–99.

[6] Courcelle, B.: Fundamental properties of infinite trees, *Theoretical Computer Science*, Vol. 25, No. 2(1983), pp. 95–169.

[7] Dershowitz, N. and Jouannaud, J.-P.: *Rewrite systems, pp.243–320, in Handbook of theoretical computer science, vol.B*, MIT Press/Elsevier, 1990.

[8] Huet, G.: *Résolution d'équations dans les languages d'ordre* $1, 2, \ldots, \omega$, PhD Thesis, University Paris-7, 1976.

[9] Jaffar, J.: Efficient unification over infinite terms, *New Generation Computing*, Vol. 2, No. 3(1984), pp. 207–219.

[10] Jouannaud, J.-P. and Kirchner, C.: *Solving equations in abstract algebras: A rule-based survey of unification, pp.257–321, in Computational Logic*, MIT Press, 1991.

[11] Kapur, D., Musser, D., Narendran, P., and Stillman, J.: Semi-unification, *Theoretical Computer Science*, Vol. 81, No. 2(1991), pp. 169–187.

[12] Martelli, A. and Montanari, U.: An efficient unification algorithm, *ACM Transactions on Programming Languages and Systems*, Vol. 4, No. 2(1982), pp. 258–282.

[13] Martelli, A. and Rossi, G.: Efficient unification with infinite trees in logic programming, *Proc. of the International Conf. on Fifth Generation Computer Systems*, 1984, pp. 202–209.

[14] Mukai, K.: A unification algorithm for infinite trees, *Proc. of IJCAI*, 1983, pp. 547–549.

[15] Oliart, A. and Snyder, W.: Fast algorithms for uniform semi-unification, *Journal of Symbolic Computation*, Vol. 37, No. 4(2004), pp. 455–484.

[16] Paterson, M. and Wegman, M.: Linear unification, *Journal of Computer and System Sciences*, Vol. 16, No. 2(1978), pp. 158–167.

[17] 岩見宗弘, 青戸等人: 無限項書き換えシステムにおける強頭部正規化可能性および一般生成性の自動反証, コンピュータソフトウェア, Vol. 29, No. 1(2012), pp. 211–239.