余帰納的定義に基づく正則項の等価性と単一化

大島 心　青戸 等人

通常の述語論理や項書き換えでは有限の大きさの項を対象とするが，項書き換えの拡張体系では無限長の枝を持つような項（無限項）を考える場合がある．無限項は木領域から関数記号と変数の集合への写像として定義できるが，項の構成に関する規則に基づいて余帰納的にも定義できる．無限項のうちで有限種類の部分項しか持たない項を正則項と呼ぶ．正則項を対象とした書き換えとして正則項書き換えが知られているが，従来の正則項書き換えにおける多くの証明は前者の定義を用いており，余帰納的な定義に基づいた証明はあまり行われていない．しかし，前者の定義に基づく証明は複雑になりがちなため，余帰納法を用いたより見通しのよい証明が試みられている（伊井・青戸，2025）．本発表では，無限項の基本的な性質である等価性および正則項の単一化手続きの正しさについて，余帰納的な定義に基づく証明を報告する．

In ordinary predicate logic and term rewriting, terms are finite in size. However, in extended rewriting systems, one may consider terms with infinitely long branches, called infinite terms. Infinite terms can be defined as mappings from a tree domain to the set of function symbols and variables, and they can also be defined coinductively based on the rules for term construction. Among infinite terms, those having only finitely many distinct subterms are called rational terms. Rational term rewriting is known as a rewriting system for rational terms, but many existing proofs in this area are based on the former definition, and few proofs have been given based on coinductive definitions. Since proofs based on the former definition tend to be complicated, clearer proofs using coinduction have been attempted (Ii and Aoto, 2025). In this presentation, we report coinductive proofs of the equivalence of infinite terms and the correctness of a unification procedure for rational terms.

1　はじめに

等式の集合を用いて計算や推論の過程をモデル化する項書き換えシステムや，数学的な証明の基盤となる述語論理において扱われる項は，通常，有限の大きさを持つ木構造として定義される．一方，項書き換えの拡張体系などにおいては，無限長の枝を持つような木，すなわち無限項を対象として扱う場合がある．無限項の中でも，有限種類の部分項しか持たないものは正則項とよばれ，正則項書き換えの基礎となる重要な概念である．

無限項は，伝統的には木領域から関数記号と変数の集合への写像として定義されることが多いが，項の構成に関する規則に基づく余帰納的なアプローチによっても厳密に定義することが可能である．無限項では構造が無限に展開するため，通常の有限項に対する帰納的な議論だけでは扱いにくい場合があるが，このような性質を数学的に厳密に扱うために余帰納法が有用となる．従来，正則項書き換えに関する多くの証明は前者の写像による定義を用いて行われてきた．しかしながら，写像の定義に基づく証明は複雑になりがちであり，その理論的基盤の検証が困難になるという課題があった．そのため，近年では余帰納法を用いてより見通しのよい証明を行う試みがなされている [6].

本論文では，この余帰納的なアプローチに基づき，正則項における基本的な性質と操作について定式化し，その正しさを証明する．まず，無限項の基本的な性質である等価性について余帰納的な定義を与え，そ

Equivalence and Unification of Rational Terms Based on Coinductive Definitions
Shin Oshima, Takahito Aoto, 新潟大学大学院自然科学研究科, Graduate School of Science and Technology, Niigata University.

れが同値関係を満たすことを証明する．次に，正則項の等価性判定手続きを定式化し，その正当性を証明する．さらに，未知の変数に代入を行って二つの項を同じ形にする操作である単一化に着目し，正則項の単一化手続きを与え，その正しさを証明する．

本論文の構成は次のとおりである．第 2 節では本論文で通して用いる基本概念や記法について述べる．第 3 節では無限項の等価性を余帰納的に定義し，それが同値関係であることを示す．第 4 節では関数としての同一性を定義し，余帰納的等価性と同一性が一致することを示す．第 5 節では正則項の等価性判定手続きを定式化し，その正当性を示す．第 6 節では正則項の単一化手続きとその正しさの証明を与える．第 7 節は結論である．

2 準備

本節では，本論文を通して用いる項，木領域，および項の構成要素に関する基本概念と記法を定義する．

2.1 項と木領域

関数記号の集合を $\mathcal{F}$，変数の可算無限集合を $\mathcal{V}$ とし，

$$\mathcal{F} \cap \mathcal{V} = \emptyset$$

とする．各関数記号 $f \in \mathcal{F}$ には，その引数の個数を表す非負整数であるアリティ $\mathrm{arity}(f) \geq 0$ が一意に定まっているとする．アリティが 0 である関数記号は定数記号と呼ばれる．

項の位置を正確に指定するために，木領域（tree domain）およびパス（path）の概念を導入する．正の整数の集合を $\mathbb{N}_+$ とし，その有限列の集合を $\mathbb{N}_+^*$ と表す．空列を ϵ で表す．また，$i \in \mathbb{N}_+$ と $p \in \mathbb{N}_+^*$ に対し，$i \cdot p$ は，根から第 i 番目の子へ進み，その後にパス p に沿って進む位置を表す．特に，$p = \epsilon$ のとき，$i \cdot \epsilon$ は根の第 i 番目の子の位置を表す．

定義 2.1（木領域）

集合 $D \subseteq \mathbb{N}_+^*$ が以下の条件を満たすとき，D を木領域（tree domain）と呼ぶ．

(1) **空列を含む**

$$\epsilon \in D$$

が成立する．

(2) **子領域が木領域となる**

各 $i \in \mathbb{N}_+$ に対して，

$$D_i = \{\, p \in \mathbb{N}_+^* \mid i \cdot p \in D \,\}$$

とおく．このとき，D_i が空でないならば，D_i も木領域をなす．すなわち，ある位置が第 i 番目の子方向に存在するならば，その子を根とする部分構造も木領域となっている．

(3) **左木領域が木領域となる**

任意の $i \in \mathbb{N}_+$ に対して，D_i が空でないとき，任意の j について，$1 \leq j < i$ ならば D_j も空でない．すなわち，第 i 番目の子領域が存在するならば，それより前の第 $1, \ldots, i-1$ 番目の子領域も存在する．

木領域の要素 $p \in D$ をパスまたは位置（position）と呼ぶ．特に ϵ は根の位置を表す．

無限項は，木領域を定義域にもつ部分関数

$$\sigma : \mathbb{N}_+^* \rightharpoonup \mathcal{F} \cup \mathcal{V}$$

として形式化される．写像 σ が $\mathcal{F}$，$\mathcal{V}$ 上の無限項であるとは，以下の 3 条件を満たすときをいう．

1. $\mathrm{dom}(\sigma)$ が木領域である．
2. $\sigma(p) = f \in \mathcal{F}$ のとき：任意の $i \in \mathbb{N}_+$ に対し，
$$p \cdot i \in \mathrm{dom}(\sigma) \iff 1 \leq i \leq \mathrm{arity}(f)$$
が成立する．
3. $\sigma(p) = x \in \mathcal{V}$ のとき：任意の $i \in \mathbb{N}_+$ に対し，
$$p \cdot i \notin \mathrm{dom}(\sigma)$$
である．

このとき，各パス $p \in \mathrm{dom}(\sigma)$ に対して，その位置にある記号 $\sigma(p)$ が一意に割り当てられる．

項 σ が

$$\sigma(p) = \begin{cases} f & (p = \epsilon), \\ t_i(p') & (p = i \cdot p',\ 1 \leq i \leq n), \\ \text{未定義} & (\text{それ以外}) \end{cases}$$

となるとき，

$$\sigma = f(t_1, \ldots, t_n)$$

と記す．ここで $f \in \mathcal{F}$ はアリティ n の関数記号であ

り、$t_1, \ldots, t_n$ は項とする．

この定義により，根位置 $p = \epsilon$ においては関数記号 f が返される．また，第 i 番目の子ノード方向へ進むパス $p = i \cdot p'$ に対しては，第 i 番目の部分項 t_i のパス p' における記号 $t_i(p')$ へと展開される．一方，$1 \leq i \leq n$ を満たさない方向へのパスでは，$\sigma(p)$ は未定義となる．

また，変数 $x \in \mathcal{V}$ を項とみなす場合は，

$$x(p) = \begin{cases} x & (p = \epsilon), \\ \text{未定義} & (\text{それ以外}) \end{cases}$$

と定める．すなわち，変数項は根位置にのみ変数記号をもち，引数をもたない項である．

2.2 無限項と基本操作

$\mathcal{F}$，$\mathcal{V}$ 上の無限項全体の集合を

$$\mathcal{T}^\omega(\mathcal{F}, \mathcal{V})$$

と記す．文脈上明らかな場合には，これを単に $\mathcal{T}^\omega$ と書く．

無限項 $\sigma \in \mathcal{T}^\omega$ が

$$\sigma = f(t_1, \ldots, t_n)$$

の形で与えられたとき，その構造要素を取り出すための関数 root，arity，children，child を以下のように定義する．なお，本論文では，関数記号のアリティと項のアリティを同じ記号 arity で表すが，文脈から区別できるものとする．以下，本論文では無限項を項と呼ぶ．

定義 2.2（根記号とアリティ）

項 $\sigma \in \mathcal{T}^\omega$ に対し，根位置の記号を返す関数

$$\mathrm{root} : \mathcal{T}^\omega \to \mathcal{F} \cup \mathcal{V}$$

を

$$\mathrm{root}(\sigma) = \sigma(\epsilon)$$

により定義する．また，項のアリティを返す関数

$$\mathrm{arity} : \mathcal{T}^\omega \to \mathbb{N}$$

を次のように定義する．

$$\mathrm{arity}(\sigma) = \begin{cases} \mathrm{arity}(f) & (\mathrm{root}(\sigma) = f \in \mathcal{F}), \\ 0 & (\mathrm{root}(\sigma) \in \mathcal{V}). \end{cases}$$

したがって，$\sigma = f(t_1, \ldots, t_n)$ のとき，

$$\mathrm{root}(\sigma) = f, \qquad \mathrm{arity}(\sigma) = n$$

である．

定義 2.3（直下の子項列 children と特定位置の子項 child）

項 $\sigma \in \mathcal{T}^\omega$ に対し，引数の項へアクセスする 2 つの関数 children および child を次のように定義する．

まず，項 $\sigma = f(t_1, \ldots, t_n)$ の直下にあるすべての部分項の列を返す関数

$$\mathrm{children} : \mathcal{T}^\omega \to (\mathcal{T}^\omega)^*$$

を

$$\mathrm{children}(\sigma) = (t_1, t_2, \ldots, t_n)$$

により定義する．ただし，$n = \mathrm{arity}(\sigma)$ である．$\mathrm{arity}(\sigma) = 0$ の場合，すなわち σ が定数または変数である場合には，

$$\mathrm{children}(\sigma) = ()$$

と定める．引数列の長さを $|\mathrm{children}(\sigma)|$ と書くとき，

$$|\mathrm{children}(\sigma)| = \mathrm{arity}(\sigma)$$

が成立する．

次に，項 σ とインデックス $i \in \mathbb{N}_+$ を受け取り，第 i 番目の直下の部分項を返す部分関数

$$\mathrm{child} : \mathcal{T}^\omega \times \mathbb{N}_+ \rightharpoonup \mathcal{T}^\omega$$

を次のように定義する．

$$\mathrm{child}(\sigma, i) = \begin{cases} t_i & (1 \leq i \leq \mathrm{arity}(\sigma)), \\ \text{未定義} & (i > \mathrm{arity}(\sigma)). \end{cases}$$

関数 $\mathrm{children}(\sigma)$ が項 σ の直下の子構造全体の列を与えるのに対し，関数 $\mathrm{child}(\sigma, i)$ は，その第 i 番目に存在する個別の部分項を指定して取り出す．したがって，

$\mathrm{children}(\sigma) = (\mathrm{child}(\sigma, 1), \ldots, \mathrm{child}(\sigma, \mathrm{arity}(\sigma)))$

が成り立つ．

異なる部分項の個数が有限である無限項を正則項（rational term）と呼ぶ．正則項全体の集合を

$$\mathcal{T}^{\omega}_{\mathrm{rat}} \subseteq \mathcal{T}^{\omega}$$

と表す．項 $\sigma \in \mathcal{T}^{\omega}$ の各パスにおける部分木構造を厳密に指定するため，まず部分項（subterm）の概念を定義する．パス $p \in \mathrm{dom}(\sigma)$ における項 σ の部分項 $\sigma|_p$ は，以下のように定義される．

定義 2.4（部分項） パス $p \in \mathrm{dom}(\sigma)$ における項 σ の部分項 $\sigma|_p$ を以下のように再帰的に定める．

$$\begin{aligned} \sigma|_{\epsilon} &= \sigma \\ \sigma|_{i \cdot p'} &= \mathrm{child}(\sigma, i)|_{p'} \quad (\text{ただし,} i \cdot p' \in \mathrm{dom}(\sigma)) \end{aligned}$$

このとき，任意のパス $p \in \mathrm{dom}(\sigma)$ について，部分項 $\sigma|_p$ もまた $\mathcal{T}^{\omega}$ の要素（項）となる．

この部分項の厳密な定義に基づき，正則項（rational term）を以下のように定義する．

定義 2.5（正則項） 無限項 $\sigma \in \mathcal{T}^{\omega}$ が正則項であるとは，σ がもつすべての部分項からなる集合

$$\{\sigma|_p \mid p \in \mathrm{dom}(\sigma)\}$$

が有限集合であるときをいう [3]．正則項全体の集合を $\mathcal{T}^{\omega}_{\mathrm{rat}}$ と表す．

以下では，次の不動点定理を用いる．

命題 2.6 (不動点定理)

写像

$$\Phi : \mathcal{P}(T^{\omega} \times T^{\omega}) \to \mathcal{P}(T^{\omega} \times T^{\omega})$$

が

$$R \subseteq R' \implies \Phi(R) \subseteq \Phi(R')$$

を満たす（単調である）とする．このとき，Φ は次の性質をもつ．

(1) Φ は最大不動点 $\nu R.\Phi(R)$ をもち，

$$\nu R.\Phi(R) = \bigcup_{R \subseteq \Phi(R)} R$$

と表される．

(2) $R \subseteq \Phi(R)$ ならば，

$$R \subseteq \nu R.\Phi(R)$$

が成り立つ．

（証明）

(1) $\nu R.\Phi(R) = \bigcup_{R \subseteq \Phi(R)} R$ が Φ の不動点であり，かつ最大であることを以下 (i)，(ii) により示す．

(i) $R \subseteq \Phi(R)$ を満たす任意の R について，和集合の定義より

$$R \subseteq \bigcup_{R \subseteq \Phi(R)} R$$

が成り立つ．したがって，Φ の単調性より

$$R \subseteq \Phi(R) \subseteq \Phi\left(\bigcup_{R \subseteq \Phi(R)} R\right)$$

が得られる．これは $R \subseteq \Phi(R)$ を満たすすべての R について成り立つので，左辺の和集合をとっても

$$\bigcup_{R \subseteq \Phi(R)} R \subseteq \Phi\left(\bigcup_{R \subseteq \Phi(R)} R\right)$$

が成り立つ．

(ii) (i) の結果，すなわち

$$\nu R.\Phi(R) \subseteq \Phi(\nu R.\Phi(R))$$

に Φ の単調性を適用すると，

$$\Phi(\nu R.\Phi(R)) \subseteq \Phi(\Phi(\nu R.\Phi(R)))$$

が得られる．これは，$\Phi(\nu R.\Phi(R))$ 自身が $R \subseteq \Phi(R)$ の形の条件を満たす集合の一つであるから，$\nu R.\Phi(R)$ の定義より

$$\Phi(\nu R.\Phi(R)) \subseteq \bigcup_{R \subseteq \Phi(R)} R = \nu R.\Phi(R)$$

が成り立つ．

(i)，(ii) より

$$\nu R.\Phi(R) = \Phi(\nu R.\Phi(R))$$

であるから，$\nu R.\Phi(R)$ は Φ の不動点である．

さらに，Φ の任意の不動点 $R = \Phi(R)$ は，特に $R \subseteq \Phi(R)$ を満たすので，$\nu R.\Phi(R)$ の定義より

$$R \subseteq \bigcup_{R \subseteq \Phi(R)} R = \nu R.\Phi(R)$$

が成り立つ．したがって，$\nu R.\Phi(R)$ は Φ の不動点のうち最大のもの，すなわち最大不動点である．

(2) $R \subseteq \Phi(R)$ を満たす関係 R は，$\nu R.\Phi(R)$ を定義する和集合

$$\bigcup_{R \subseteq \Phi(R)} R$$

を構成する集合の一つである．したがって，和集合の定義より

$$R \subseteq \bigcup_{R \subseteq \Phi(R)} R = \nu R.\Phi(R)$$

が成り立つ． □

3 等価性の同値関係の証明

本節では，余帰納的アプローチに基づき，無限項の等価性を定義し，それが同値関係であることを最大不動点の性質を用いて証明する．

3.1 等価性の余帰納的定義

無限項に対して通常の有限項のような帰納的定義をそのまま適用することはできない．そこで，無限に続く項構造の挙動の等価性を数学的に厳密に扱うため，関係の上のオペレータを用いた余帰納的な定義を導入する．

定義 3.1（等価性オペレータ）

二項関係 $R \subseteq \mathcal{T}^\omega \times \mathcal{T}^\omega$ に対し，等価性オペレータ

$$T_\approx : \mathcal{P}(\mathcal{T}^\omega \times \mathcal{T}^\omega) \to \mathcal{P}(\mathcal{T}^\omega \times \mathcal{T}^\omega)$$

を次のように定義する：

$$T_\approx(R) = \left\{ (\sigma, \tau) \;\middle|\; \begin{array}{l} \mathrm{root}(\sigma) = \mathrm{root}(\tau), \\ \mathrm{arity}(\sigma) = \mathrm{arity}(\tau), \\ \forall i\,(1 \le i \le \mathrm{arity}(\sigma) \Rightarrow \\ \quad (\mathrm{child}(\sigma, i), \mathrm{child}(\tau, i)) \in R) \end{array} \right\}.$$

命題 2.6 を $\Phi = T_\approx$ として適用すると， $T_\approx$ の単調性より，最大不動点

$$\approx \; = \nu R.T_\approx(R)$$

の存在が保証され，「ある二項関係 R が $R \subseteq T_\approx(R)$ を満たせば $R \subseteq \approx$ が成り立つ」という余帰納法原理が命題 2.6 より直ちに得られる．

定義 3.2（無限項の等価性 $\approx$）

無限項の間の等価性

$$\approx \subseteq \mathcal{T}^\omega \times \mathcal{T}^\omega$$

を，等価性オペレータ $T_\approx$ の最大不動点として次のように定義する：

$$\approx = \nu R.T_\approx(R) = \bigcup \{\, R \subseteq \mathcal{T}^\omega \times \mathcal{T}^\omega \mid R \subseteq T_\approx(R) \,\}.$$

この定義から，ある二項関係 R が $R \subseteq T_\approx(R)$ を満たすことを示せば，余帰納法原理により $R \subseteq \approx$ が導かれる．また，$\approx$ 自身は不動点であるため，

$$\approx = T_\approx(\approx)$$

が成立する．したがって，任意の $(\sigma, \tau) \in \approx$ について，以下の 3 つの性質が成り立つ：

(1) $\mathrm{root}(\sigma) = \mathrm{root}(\tau)$

(2) $\mathrm{arity}(\sigma) = \mathrm{arity}(\tau)$

(3) $\forall i\,(1 \le i \le \mathrm{arity}(\sigma)),\ \mathrm{child}(\sigma, i) \approx \mathrm{child}(\tau, i)$

3.2 同値関係の証明

定義 3.2 で与えた等価性 $\approx$ が，反射性，対称性，推移性を満たすことを証明する．

補題 3.3（反射性）

任意の無限項 $\sigma \in \mathcal{T}^\omega$ について，$\sigma \approx \sigma$ が成り立つ．

（証明） 恒等関係 id を

$$id = \{\, (\sigma, \sigma) \mid \sigma \in \mathcal{T}^\omega \,\}$$

と定義する．任意の $(\sigma, \sigma) \in id$ について，$(\sigma, \sigma) \in T_\approx(id)$ を示す．

(i) $\mathrm{root}(\sigma) = \mathrm{root}(\sigma)$ は自明に成り立つ．

(ii) $\mathrm{arity}(\sigma) = \mathrm{arity}(\sigma)$ も自明に成り立つ．

(iii) 任意の $1 \le i \le \mathrm{arity}(\sigma)$ に対し，$\mathrm{child}(\sigma, i)$ 自身が同じ項であるから，

$$(\mathrm{child}(\sigma, i), \mathrm{child}(\sigma, i)) \in id$$

である．

したがって，

$$id \subseteq T_\approx(id)$$

が成り立つ．よって，$\approx$ が $T_\approx$ の最大不動点であることから，

$$id \subseteq \approx$$

となるため，任意の $\sigma \in \mathcal{T}^\omega$ について $\sigma \approx \sigma$ が成立する． □

補題 3.4（対称性）

任意の無限項 $\sigma, \tau \in \mathcal{T}^\omega$ について，$\sigma \approx \tau$ ならば $\tau \approx \sigma$ が成り立つ．

（証明） 関係 R_{sym} を

$$R_{\mathrm{sym}} = \{\, (\tau, \sigma) \mid \sigma \approx \tau \,\}$$

と定義する．任意の $(\tau, \sigma) \in R_{\mathrm{sym}}$ をとると，定義より $\sigma \approx \tau$ である．$\approx$ は $T_\approx$ の最大不動点であるから，

$$\approx \subseteq T_\approx(\approx)$$

が成り立つ．したがって，$(\sigma,\tau)\in T_{\approx}(\approx)$ であり，次の性質が成り立つ：

(1) $\mathrm{root}(\sigma)=\mathrm{root}(\tau)$

(2) $\mathrm{arity}(\sigma)=\mathrm{arity}(\tau)$

(3) 任意の $i\in\{1,\ldots,\mathrm{arity}(\sigma)\}$ について，$\big(\mathrm{child}(\sigma,i),\mathrm{child}(\tau,i)\big)\in\approx$

ここで，(1) より

$$\mathrm{root}(\tau)=\mathrm{root}(\sigma)$$

が成り立つ．また，(2) より

$$\mathrm{arity}(\tau)=\mathrm{arity}(\sigma)$$

が成り立つ．さらに，(3) より $\mathrm{child}(\sigma,i)\approx\mathrm{child}(\tau,i)$ であるから，R_{sym} の定義により，任意の $i\in\{1,\ldots,\mathrm{arity}(\sigma)\}$ について

$$\big(\mathrm{child}(\tau,i),\mathrm{child}(\sigma,i)\big)\in R_{\mathrm{sym}}$$

が成り立つ．したがって，

$$(\tau,\sigma)\in T_{\approx}(R_{\mathrm{sym}})$$

である．以上より

$$R_{\mathrm{sym}}\subseteq T_{\approx}(R_{\mathrm{sym}})$$

が示された．$\approx$ が $T_{\approx}$ の最大不動点であることから，

$$R_{\mathrm{sym}}\subseteq\approx$$

となるので，任意の $\sigma,\tau\in\mathcal{T}^{\omega}$ について

$$\sigma\approx\tau\implies\tau\approx\sigma$$

が成立する． □

補題 3.5（推移性）

任意の無限項 $\sigma,\rho,\tau\in\mathcal{T}^{\omega}$ について，$\sigma\approx\rho$ かつ $\rho\approx\tau$ ならば $\sigma\approx\tau$ が成り立つ．

（証明）関係 R_{trans} を

$$R_{\mathrm{trans}}=\{\,(\sigma,\tau)\mid\exists\rho\in\mathcal{T}^{\omega},\ \sigma\approx\rho\wedge\rho\approx\tau\,\}$$

と定義する．任意の $(\sigma,\tau)\in R_{\mathrm{trans}}$ に対し，ある $\rho\in\mathcal{T}^{\omega}$ が存在して $\sigma\approx\rho$ および $\rho\approx\tau$ が成り立つ．$\approx=T_{\approx}(\approx)$ より，$\sigma\approx\rho$ について

(1a) $\mathrm{root}(\sigma)=\mathrm{root}(\rho)$

(1b) $\mathrm{arity}(\sigma)=\mathrm{arity}(\rho)$

(1c) $\forall i\,(1\leq i\leq\mathrm{arity}(\sigma)),\ \mathrm{child}(\sigma,i)\approx\mathrm{child}(\rho,i)$

が成り立つ．同様に，$\rho\approx\tau$ について

(2a) $\mathrm{root}(\rho)=\mathrm{root}(\tau)$

(2b) $\mathrm{arity}(\rho)=\mathrm{arity}(\tau)$

(2c) $\forall i\,(1\leq i\leq\mathrm{arity}(\rho)),\ \mathrm{child}(\rho,i)\approx\mathrm{child}(\tau,i)$

が成り立つ．これらから $(\sigma,\tau)\in T_{\approx}(R_{\mathrm{trans}})$ を導く．

(i) $\mathrm{root}(\sigma)=\mathrm{root}(\rho)$ かつ $\mathrm{root}(\rho)=\mathrm{root}(\tau)$ より，

$$\mathrm{root}(\sigma)=\mathrm{root}(\tau)$$

が成立する．

(ii) $\mathrm{arity}(\sigma)=\mathrm{arity}(\rho)$ かつ $\mathrm{arity}(\rho)=\mathrm{arity}(\tau)$ より，

$$\mathrm{arity}(\sigma)=\mathrm{arity}(\tau)$$

が成立する．

(iii) 任意の $1\leq i\leq\mathrm{arity}(\sigma)$ について，$\mathrm{child}(\sigma,i)\approx\mathrm{child}(\rho,i)$ かつ $\mathrm{child}(\rho,i)\approx\mathrm{child}(\tau,i)$ である．したがって，R_{trans} の定義により

$$(\mathrm{child}(\sigma,i),\mathrm{child}(\tau,i))\in R_{\mathrm{trans}}$$

が成り立つ．

したがって，

$$R_{\mathrm{trans}}\subseteq T_{\approx}(R_{\mathrm{trans}})$$

が成立する．$\approx$ は $T_{\approx}$ の最大不動点であるから，

$$R_{\mathrm{trans}}\subseteq\approx$$

となり，推移性が得られる． □

定理 3.6（同値関係）

無限項の等価性 $\approx$ は同値関係である．

（証明）補題 3.3，補題 3.4，および補題 3.5 より，関係 $\approx$ は反射性，対称性，推移性を満たすため，同値関係である． □

4 等価性と同一性の一致

本節では，パス（位置）における記号の一致によって定義される無限項の同一性（$=$）を導入し，第 3 節で与えた余帰納的な等価性（equivalence，$\approx$）と一致すること，すなわち

$$\sigma\approx\tau\iff\sigma=\tau$$

が成り立つことを証明する．

4.1 同一性の定義

第 2 節で定義したように，無限項 $\sigma\in\mathcal{T}^{\omega}$ は木領域を定義域にもつ部分関数

$$\sigma:\mathbb{N}_{+}^{*}\rightharpoonup\mathcal{F}\cup\mathcal{V}$$

として形式化される．これに基づき，2 つの無限項が定義域が等しく定義域に含まれるすべての位置において同一記号を返す関数となっていることを表す同一性（$=$）を，次のように定義する．

定義 4.1（無限項の同一性）

任意の無限項 $\sigma, \tau \in \mathcal{T}^\omega$ に対し，同一性 $\sigma = \tau$ を，すべてのパス $p \in \mathbb{N}_+^*$ において，対応する位置の記号が一致することとして

$$\sigma = \tau \iff \forall p \in \mathbb{N}_+^*,\ \phi_\sigma(p) = \phi_\tau(p)$$

と定義する．ここで，$\phi_\sigma(p)$ は，

$$\phi_\sigma(p) = \begin{cases} \sigma(p) & (p \in \mathrm{dom}(\sigma) \text{ のとき}), \\ \bot & (p \notin \mathrm{dom}(\sigma) \text{ のとき}) \end{cases}$$

と定義する．ここで，$\bot$ は $\mathcal{F} \cup \mathcal{V}$ に現れない記号とする．

また，パスに沿って子ノード方向へ展開する際の部分項の挙動に関して，次の補題が成り立つ．

補題 4.2（パス展開と部分項）

任意の $\sigma \in \mathcal{T}^\omega$，$1 \le i \le \mathrm{arity}(\sigma)$，および $p' \in \mathbb{N}_+^*$ に対し，次が成立する：

$$\phi_\sigma(i \cdot p') = \phi_{\mathrm{child}(\sigma, i)}(p')$$

（証明）第 2 節における写像としての項の定義より，$\sigma = f(t_1, \ldots, t_n)$ のとき，

$$\phi_\sigma(i \cdot p') = \phi_{t_i}(p')$$

であり，また

$$\mathrm{child}(\sigma, i) = t_i$$

である．したがって，

$$\phi_\sigma(i \cdot p') = \phi_{\mathrm{child}(\sigma, i)}(p')$$

が従う． □

4.2 等価性と同一性の一致の証明

余帰納的に定義された等価性 $\approx$ と，記号の位置ごとの一致として定義された同一性 $=$ が一致することを示す．

補題 4.3（$\approx \subseteq =$）任意の無限項 $\sigma, \tau \in \mathcal{T}^\omega$ について，$\sigma \approx \tau$ ならば $\sigma = \tau$ が成り立つ．

（証明）背理法により証明する．$\sigma \approx \tau$ であるが $\sigma \neq \tau$ と仮定する．同一性の定義より，記号が一致しないパス $p \in \mathbb{N}_+^*$ が存在する．すなわち，

$$\phi_\sigma(p) \neq \phi_\tau(p)$$

である．このようなパスのうち，長さが最小であるものを

$$p = i_1 \cdot i_2 \cdots i_m$$

（$m \ge 0$）とする．

このとき，$\phi_\sigma(p) \neq \phi_\tau(p)$ より，$\phi_\sigma(p) \neq \bot$ かつ $\phi_\tau(p) \neq \bot$ が成立する．そこで一般性を失うことなく，$\phi_\sigma(p) \neq \bot$ を仮定する．$\mathrm{dom}(\sigma)$ は木領域であるから，$p \in \mathrm{dom}(\sigma)$ より，$\sigma_0, \ldots, \sigma_m$ を

$$\sigma_0 = \sigma, \qquad \sigma_j = \mathrm{child}(\sigma_{j-1}, i_j) \quad (1 \le j \le m)$$

により定めることができて，更に

$$1 \le i_j \le \mathrm{arity}(\sigma_{j-1}) \qquad (*)$$

が成立する．

次に，パスに沿って展開していく部分項のペアの列

$$(\sigma_0, \tau_0), (\sigma_1, \tau_1), \ldots, (\sigma_m, \tau_m)$$

を次のように定義する：

$$(\sigma_0, \tau_0) = (\sigma, \tau)$$

$$(\sigma_j, \tau_j) = \big(\mathrm{child}(\sigma_{j-1}, i_j), \mathrm{child}(\tau_{j-1}, i_j)\big) \qquad (1 \le j \le m).$$

仮定より $\sigma_0 \approx \tau_0$ である．等価性 $\approx$ は第 3 節で示したとおり最大不動点であるから，任意の $1 \le j \le m$ について，

$$\sigma_{j-1} \approx \tau_{j-1}$$

ならば

(1) $\mathrm{root}(\sigma_{j-1}) = \mathrm{root}(\tau_{j-1})$

(2) $\mathrm{arity}(\sigma_{j-1}) = \mathrm{arity}(\tau_{j-1})$

(3) 任意の i について，$1 \le i \le \mathrm{arity}(\sigma_{j-1})$ ならば

$$\mathrm{child}(\sigma_{j-1}, i) \approx \mathrm{child}(\tau_{j-1}, i)$$

が成り立つ．これと（$*$）を合わせて，すべての $0 \le j \le m$ に対して

$$\sigma_j \approx \tau_j$$

が成り立つ．特に，$j = m$ において

$$\sigma_m \approx \tau_m$$

であるから，

$$\mathrm{root}(\sigma_m) = \mathrm{root}(\tau_m)$$

が成立する．一方，補題 4.2 を繰り返し適用すると，$\phi_\sigma(p) = \phi_{\sigma_m}(\epsilon)$，$\phi_\tau(p) = \phi_{\tau_m}(\epsilon)$ であるから，これらを合わせて

$$\phi_{\sigma_m}(\epsilon) = \mathrm{root}(\sigma_m) = \mathrm{root}(\tau_m) = \phi_{\tau_m}(\epsilon)$$

が成り立つ．したがって

$$\phi_\sigma(p) = \phi_\tau(p)$$

となるが，これは最初の仮定

$$\phi_\sigma(p) \neq \phi_\tau(p)$$

に矛盾する．ゆえに仮定は誤りであり，

$$\sigma \approx \tau \implies \sigma = \tau$$

が成り立つ． □

補題 4.4（$=\subseteq\approx$）

任意の無限項 $\sigma, \tau \in \mathcal{T}^\omega$ について，$\sigma = \tau$ ならば $\sigma \approx \tau$ が成り立つ．

（証明）集合 R を

$$R = \{\, (\sigma, \tau) \in \mathcal{T}^\omega \times \mathcal{T}^\omega \mid \sigma = \tau \,\}$$

と定義する．任意の $(\sigma, \tau) \in R$ をについて，$(\sigma, \tau) \in T_\approx(R)$ を示す．

(i) 空パス $p = \epsilon$ を考えると，$\sigma = \tau$ より

$$\phi_\sigma(\epsilon) = \phi_\tau(\epsilon)$$

である．したがって，定義 2.2 より

$$\mathrm{root}(\sigma) = \phi_\sigma(\epsilon) = \phi_\tau(\epsilon) = \mathrm{root}(\tau)$$

が成立する．

(ii) (i) より根の記号が一致しているため，そのアリティも等しい．したがって，

$$\mathrm{arity}(\sigma) = \mathrm{arity}(\tau)$$

が成立する．

(iii) 任意の $1 \leq i \leq \mathrm{arity}(\sigma)$ に対し，$\mathrm{child}(\sigma, i)$ と $\mathrm{child}(\tau, i)$ の同一性を確認する．任意のパス $p' \in \mathbb{N}_+^*$ に対し，補題 4.2 より

$$\phi_{\mathrm{child}(\sigma,i)}(p') = \phi_\sigma(i \cdot p')$$
$$\phi_{\mathrm{child}(\tau,i)}(p') = \phi_\tau(i \cdot p')$$

である．$\sigma = \tau$ より，すべてのパスで記号が一致するため，特にパス $i \cdot p'$ においても

$$\phi_\sigma(i \cdot p') = \phi_\tau(i \cdot p')$$

が成立する．したがって，

$$\phi_{\mathrm{child}(\sigma,i)}(p') = \phi_{\mathrm{child}(\tau,i)}(p')$$

が任意の p' で成り立ち，定義より

$$\mathrm{child}(\sigma, i) = \mathrm{child}(\tau, i)$$

である．これは

$$(\mathrm{child}(\sigma, i), \mathrm{child}(\tau, i)) \in R$$

であることを意味する．

以上より，

$$R \subseteq T_\approx(R)$$

が示された．余帰納法原理より

$$R \subseteq \approx$$

となるため，

$$\sigma = \tau \implies \sigma \approx \tau$$

が成立する． □

定理 4.5（等価性と同一性の一致）

任意の無限項 $\sigma, \tau \in \mathcal{T}^\omega$ について，等価性 $\approx$ と同一性 $=$ は一致する．すなわち，

$$\sigma \approx \tau \iff \sigma = \tau$$

が成り立つ．

（証明）補題 4.3 および補題 4.4 より直ちに従う． □

5 正則項の等価性判定手続き

本節では，正則項の等価性を判定するアルゴリズムを定式化し，その手続きが正当であることを証明する．正則項の比較は葉まで到達できず無限ループに陥る可能性があるため，一度比較した等式を履歴として記録し，循環を検知する仕組みを導入する．

本章における等価性判定手続きを議論するため，まず，正則項の有限表現を与えるカノニカル等式系（カノニカルシステム）と，それを写像として表す「環境」，およびそこから実際の無限木（無限項）を展開する写像の定義を導入する．

定義 5.1（カノニカルシステムと環境）等式の有限集合 $\mathcal{E}$ を，カノニカル（正則）システム（canonical system）もしくはカノニカル等式系であるとは，以下の条件を満たすものという．

(1) $\mathcal{E}$ に含まれる等式の左辺が互いに異なる変数である．

(2) 各等式の右辺は関数記号の下に変数のみが引数として並ぶ形式，すなわち $f(x_1, \ldots, x_n)$ で，$x_1, \ldots, x_n (n = \mathrm{arity}(f))$ は $\mathcal{E}$ の左辺に現れる変数または左辺に現れない変数．

このカノニカルシステムは，各変数を関数記号と引数変数の形，または別の変数へと写す部分関数

$$\mathcal{E} : \mathcal{V} \rightharpoonup (\mathcal{F} \times \mathcal{V}^*) \cup \mathcal{V}$$

としてあたえることもでき，これを環境（environment）

と呼ぶ．$\mathcal{E}$ の左辺に現れる変数を $\mathcal{E}$ による束縛変数と呼び，それ以外の変数を自由変数と呼ぶ．

例えば，以下のような有限個の定義の集合 $\mathcal{E}$ は，カノニカルシステムをなしている：

$$\mathcal{E} = \begin{cases} x = f(x, y), \\ y = g(z), \\ z = w \end{cases}$$

このとき，$\mathcal{E}$ による束縛変数は x, y, z となる．

定義 5.2（環境による無限木への展開 $\mathcal{E}^*$）

与えられた環境 $\mathcal{E}$ の下で，各変数 $x \in \mathcal{V}$ から展開される無限木（無限項） $\mathcal{E}^*(x) \in \mathcal{T}^\omega$ は，部分関数 $\mathcal{E}^*(x) : \mathbb{N}_+^* \rightharpoonup \mathcal{F} \cup \mathcal{V}$ として，以下のように定義される．

1. $\mathcal{E}(x) = f(x_1, \ldots, x_n)$ （ただし $n = \mathrm{arity}(f)$）のとき：
$$\mathcal{E}^*(x)(p) = \begin{cases} f & (p = \epsilon), \\ \mathcal{E}^*(x_i)(p') & (p = i \cdot p',\ 1 \le i \le n), \\ \text{未定義} & (\text{それ以外}) \end{cases}$$
2. $\mathcal{E}(x) = y$ （ただし $y \in \mathcal{V}$）のとき：
$$\mathcal{E}^*(x)(p) = \mathcal{E}^*(y)(p)$$
3. $\mathcal{E}(x)$ が未定義のとき（すなわち x が自由変数のとき）：
$$\mathcal{E}^*(x)(p) = \begin{cases} x & (p = \epsilon \text{ のとき}), \\ \text{未定義} & (\text{それ以外}) \end{cases}$$

5.1 判定手続きの定式化

本アルゴリズムでは，判定対象となる二つの正則項を，同一のカノニカルシステムにおける変数 $x, y \in \mathcal{V}$ の等価性問題へと帰着する．すなわち，正則項 t を定義 5.2 で導入したように $t = \mathcal{E}^*(x)$ となるカノニカル等式系 $\mathcal{E}$ と変数 x により表現する．さらに，このカノニカルシステムを環境 $\mathcal{E}$ として参照することで，等式 $x \sim y$ から展開される無限木（無限項）の間の等価性 $\mathcal{E}^*(x) \approx \mathcal{E}^*(y)$ を判定する．

アルゴリズムの構成要素は以下の通りである．

- **入力**：背後にある環境 $\mathcal{E}$，および判定対象となる初期の変数対の集合 $\{x \sim y\}$
- **状態**：未処理の等式集合 $Conj$ と，処理済みの等式集合 $Done$ の組 $\langle Conj, Done \rangle$
- **出力**：成功なら処理済みの等式ペア集合 $Done$，失敗なら $\bot$

具体的には，比較対象となる二つの正則項 $s, t \in \mathcal{T}^\omega_{\mathrm{rat}}$ が，それぞれ変数集合上で定義された環境 $\mathcal{E}_s, \mathcal{E}_t$ の下で，特定の変数 $x, y \in \mathcal{V}$ から

$$s = \mathcal{E}_s^*(x), \quad t = \mathcal{E}_t^*(y)$$

と展開されるものとする．ここで，二つの環境で用いられる変数が衝突しないよう，$\mathrm{Dom}(\mathcal{E}_s) \cap \mathrm{Dom}(\mathcal{E}_t) = \emptyset$ であると仮定する．

このとき，二つの正則項の等価性（すなわち $s \approx t$）を判定するため，背景となる共通の環境 $\mathcal{E}$ を $\mathcal{E} = \mathcal{E}_s \cup \mathcal{E}_t$ として構成する．アルゴリズムは，判定対象となる初期変数対 $\{x \sim y\}$ のみを未処理集合 $Conj$ に設定し，初期状態 $\langle \{x \sim y\}, \emptyset \rangle$ から出発する．以降，各ステップにおいて背景環境 $\mathcal{E}$ を参照しながら，未処理集合 $Conj$ から等式を取り出し，手続きを進める．

1. **根記号が一致するなら分解する**：環境 $\mathcal{E}$ において $\mathcal{E}(x) = f(x_1, \ldots, x_n)$ および $\mathcal{E}(y) = f(y_1, \ldots, y_n)$ であり，根の関数記号 (アリティ) が一致する場合，新たな子ノードのペアの集合 $\{x_1 \sim y_1, \ldots, x_n \sim y_n\}$ を生成し，$x = y$ を $Done$ に追加する．
2. **分解した子ペアは $Done$ を見て重複を除く**：無限木の循環による無限ループを防ぐため，新たに生成されたペアのうち既に $Done$ に含まれているものは $Conj$ へ追加をしない．$Done$ にないものだけを $Conj$ に追加する．
3. **根記号が異なれば失敗する**：$\mathcal{E}(x) = f(\ldots)$ かつ $\mathcal{E}(y) = g(\ldots)$ のように根の関数記号が異なる（$f \neq g$）場合，等価ではないと判定し，失敗終了（$\bot$）する．

このうち，分解と重複排除を伴う正常な状態遷移は **Decompose**（分解）規則として，また根の関数記号が異なる場合の失敗終了（失敗）は **Clash**（衝突）規則として，それぞれ次のような推論規則の形式で定義される．

Decompose

$$\frac{\langle \{x \sim y\} \uplus Conj,\ Done \rangle}{\langle Conj \cup (\{x_i \sim y_i\}_{i=1}^n \setminus Done),\ Done \cup \{x \sim y\} \rangle}$$

$$\text{if } (\mathcal{E}(x) \sim f(x_1, \ldots, x_n) \text{ and } \mathcal{E}(y) \sim f(y_1, \ldots, y_n)) \text{ or } x = y$$

Clash

$$\frac{\langle \{x \sim y\} \uplus Conj,\ Done\rangle}{\bot}$$

$$\text{if } \mathrm{root}(\mathcal{E}(x)) \neq \mathrm{root}(\mathcal{E}(y)).$$

アルゴリズムは状態遷移を繰り返し，未処理集合 $Conj$ が空集合になった場合，すなわち初期状態 $\langle \{x \sim y\}, \emptyset\rangle$ から $\langle \emptyset, Done\rangle$ に到達した場合に成功として停止し，出力として $Done$ を返す．

5.2 判定手続きの停止性と正当性

本節では，5.1 節で定式化した等価性判定アルゴリズムの停止性と正当性を証明する．まず，アルゴリズムの実行状態で維持される不変量について述べる．環境 $\mathcal{E}$ に現れる変数の有限集合を V とおくとき，任意の実行状態 $\langle Conj, Done\rangle$ において常に以下が維持される．

(1) $Conj \cup Done \subseteq V \times V$

(2) $Conj \cap Done = \emptyset$

定理 5.3（判定手続きの停止性） 等価性判定アルゴリズムは，任意の初期状態 $\langle \{x \sim y\}, \emptyset\rangle$ から有限ステップの遷移の後に必ず停止状態 $\langle \emptyset, Done\rangle$ または $\bot$（異常終了）に到達する．

（証明） 実行状態に対する重みを $\Phi(\langle Conj, Done\rangle) = |(V \times V) \setminus Done|$ と定義する．$V \times V$ は有限集合であるため，この値は常に非負整数である．アルゴリズムの 1 ステップの遷移 $\langle Conj, Done\rangle \Rightarrow \langle Conj', Done'\rangle$ において：

- Clash 規則適用時は，直ちに $\bot$ へ遷移して停止する．
- Decompose 規則適用時は，不変量 (2) より，取り出された等式は $x = y \notin Done$ を満たす．したがって，$Done' = Done \cup \{x = y\}$ より $|Done'| = |Done| + 1$ が成り立ち，重みは $\Phi(\langle Conj', Done'\rangle) = \Phi(\langle Conj, Done\rangle) - 1$ と厳密に 1 減少する．

重みは非負整数であるため，Decompose 規則の連続適用回数は高々 $|V \times V|$ 回に制限される．したがって，アルゴリズムは無限に遷移を繰り返すことはなく，有限ステップの後に $Conj = \emptyset$（正常終了）または $\bot$（異常終了）となって必ず停止する． □

無限木の等価性を判定する有限のアルゴリズムが成功終了したとき，出力された等式ペア集合が第 2 節で定義した等価性 $\approx$ を満たすことを証明する．

成功終了時の処理済み集合（出力された $Done$）を C_∞ とおく．成功終了時には，集合 C_∞ は Decompose 規則に関して閉じており，以下の性質を満たす．

(i) 任意の $(x, y) \in C_\infty$ について，$root(\mathcal{E}(x)) = root(\mathcal{E}(y))$ が成立する．

(ii) $\mathcal{E}(x) \sim f(x_1, \ldots, x_n)$ かつ $\mathcal{E}(y) \sim f(y_1, \ldots, y_n)$ ならば，任意の $1 \leq i \leq n$ について $(x_i, y_i) \in C_\infty$ が成立する．

この閉包性を用いて，アルゴリズムの正当性を以下の定理として示す．

補題 5.4（判定手続きの正当性） 等価性判定アルゴリズムが成功終了したとき，初期状態の等式 $x \sim y$ について，$\mathcal{E}^*(x) \approx \mathcal{E}^*(y)$.

（証明） 終了時の処理済み集合 C_∞ の要素を用いて，無限木の間の二項関係 R を次のように構築する．

$$R = \{(\mathcal{E}^*(x), \mathcal{E}^*(y)) \mid (x, y) \in C_\infty\}.$$

等価性 $\approx$ は等価性オペレータ $T_\approx$ の最大不動点として定義されているため，関係 $R \subseteq T_\approx(R)$ が成立することを示せば，最大不動点の性質より $R \subseteq \approx$ が導かれる．

関係 R から任意のペア (s, t) を取り出す．R の定義より，ある $(x, y) \in C_\infty$ が存在して

$$s = \mathcal{E}^*(x), \qquad t = \mathcal{E}^*(y)$$

となっている．この s と t について，等価性オペレータ $T_\approx(R)$ の成立条件 (i)〜(iii) を検証する．

(i) C_∞ の閉包性より $root(\mathcal{E}(x)) = root(\mathcal{E}(y))$ であり，この根の記号を f とおく．木の展開の定義より，

$$root(s) = root(\mathcal{E}^*(x)) = root(\mathcal{E}(x)) = f$$

であり，同様に

$$root(t) = root(\mathcal{E}^*(y)) = root(\mathcal{E}(y)) = f$$

となる．したがって，$root(s) = root(t)$ が成立する．

(ii) (i) より根の関数記号または変数が一致しているため，そのアリティも等しくなり，

$$arity(s) = arity(t) = n$$

が成立する．

(iii) s および t の i 番目の子ノードをそれぞれ s_i, t_i とおく．木の展開の性質から，

$s_i = child(s,i) = \mathcal{E}^*(x_i), \qquad t_i = child(t,i) = \mathcal{E}^*(y_i)$

となる．ここで，成功終了時には C_∞ は Decompose 規則に関して閉じているため，親のペア $(x,y) \in C_\infty$ があればその子ノードの変数のペアも $(x_i, y_i) \in C_\infty$ に含まれる．したがって関係 R の定義より直接的に

$$(\mathcal{E}^*(x_i), \mathcal{E}^*(y_i)) \in R$$

すなわち

$$(s_i, t_i) \in R$$

が成立する．これにより，すべての子ノードの対が関係 R に含まれることが示された．

以上により，関係 R から取り出した任意の対 (s,t) は等価性オペレータ $T_\approx(R)$ の条件をすべて満たしている．したがって

$$R \subseteq T_\approx(R)$$

が示され，最大不動点の性質より

$$R \subseteq \approx$$

が成立する．よって，アルゴリズムの成功終了時において，初期状態のペア $(x,y) \in C_\infty$ より，

$$\mathcal{E}^*(x) \approx \mathcal{E}^*(y)$$

が成り立つ． □

定理 5.5（判定手続きの同値性） $\mathcal{E}_s$， $\mathcal{E}_t$ をカノニカルシステムとし，$s = \mathcal{E}_s^*(x)$， $t = \mathcal{E}_t^*(y)$ とする．このとき，二つの環境の定義域が交わらない（$\mathrm{Dom}(\mathcal{E}_s) \cap \mathrm{Dom}(\mathcal{E}_t) = \emptyset$）と仮定し，共通の環境を $\mathcal{E} = \mathcal{E}_s \cup \mathcal{E}_t$ とおく．等価性判定アルゴリズムが，背景環境 $\mathcal{E}$ の下で初期状態 $\langle \{x \sim y\}, \emptyset \rangle$ から開始して成功終了することと，$s = t$ であることは同値である．

（証明） ($\Longrightarrow$) アルゴリズムが初期状態 $\langle \{x \sim y\}, \emptyset \rangle$ から成功終了したと仮定する．このとき，補題 5.4（判定手続きの正当性）より，初期等式 $x \sim y$ の両辺から展開される無限木は余帰納的等価性 $\approx$ を満たす．すなわち，

$$\mathcal{E}^*(x) \approx \mathcal{E}^*(y)$$

が成り立つ．共通環境 $\mathcal{E}$ の構成において，変数 x の展開には $\mathcal{E}_s$ の定義のみが参照され，変数 y の展開には $\mathcal{E}_t$ の定義のみが参照されるため，$\mathcal{E}^*(x) = \mathcal{E}_s^*(x) = s$ および $\mathcal{E}^*(y) = \mathcal{E}_t^*(y) = t$ が成り立つ．よって $s \approx t$ が得られる．ここで，定理 4.5（等価性と同一性の一致）より $\approx$ と $=$ は一致するため，$s = t$ が成り立つ．

($\Longleftarrow$) $s = t$ と仮定する．定理 4.5 より $s \approx t$，すなわち $\mathcal{E}^*(x) \approx \mathcal{E}^*(y)$ が成り立つ．ここで，アルゴリズムが成功終了しないと仮定して矛盾を導く．定理 5.3（判定手続きの停止性）より，アルゴリズムは有限ステップで必ず停止するため，成功終了しない場合の終了状態は失敗終了 $\bot$ （Clash 規則の適用）のみである．すなわち，あるステップにおいて未処理等式集合 $Conj$ から等式 $u \sim v$ が取り出され，Clash 規則が適用されたことになる．Clash 規則の適用条件より，$root(\mathcal{E}(u)) \neq root(\mathcal{E}(v))$ ．しかし，初期等式 $x \sim y$ が $\mathcal{E}^*(x) \approx \mathcal{E}^*(y)$ を満たすとき，Decompose 規則によって展開されるすべての等式対 $u = v$ に対しても $\mathcal{E}^*(u) \approx \mathcal{E}^*(v)$ が維持される．余帰納的等価性 $\approx$ の定義より，等価な無限木の根記号は一致しなければならないため，

$$root(\mathcal{E}^*(u)) = root(\mathcal{E}^*(v))$$

が成り立つ．無限木への展開の定義より，$root(\mathcal{E}^*(u)) = root(\mathcal{E}(u))$，$root(\mathcal{E}^*(v)) = root(\mathcal{E}(v))$ であるから，$root(\mathcal{E}(u)) = root(\mathcal{E}(v))$ となり，$root(\mathcal{E}(u)) \neq root(\mathcal{E}(v))$ に矛盾する．したがって，アルゴリズムが異常終了することはないため，必ず成功終了する． □

6 正則項の単一化手続き

無限項における単一化は、古典的に研究されてきたが [2] [4]，本節では，単一化アルゴリズムを与え，その手続きが最汎単一化子を導出することを証明する．まず単一化可能および単一化子の定義を明確にする．通常の有限項における単一化では，$s\sigma = t\sigma$（構文的完全一致）となる代入 σ を単一化子とするが，本アルゴリズムでは無限木の同値性を扱うため，等式系に

よる無限木への展開操作（前節における $\mathcal{E}^*$ と同様の展開）を用いる．

定義 6.1（無限項における単一化および単一化子） 関数 $\theta : V \to T^\omega$ が正則代入であるとは，$\mathrm{dom}(\theta) = \{x \mid \theta(x) \neq x\}$ が有限であるときをいう．項 s と t が単一化可能であるとは，ある正則代入 θ が存在して $s\theta = t\theta$ となるときのことをいう．ある関数定義の等式集合 U の下で，項 s と項 t が単一化可能であるとは，$U^*(s)\sigma = U^*(t)\sigma$ となるような代入 σ が存在するときをいう．この条件を満たす代入 σ を，等式系 U における s と t の単一化子と呼ぶ．

6.1 実行状態の定義

本アルゴリズムは，あらかじめカノニカルシステムに変換された初期等式系 S_0 を入力とする．S_0 では左辺に現れる変数を束縛変数 (BV)，左辺に現れず右辺のみに現れる変数を自由変数 (FV) として区別する．すなわち，S_0 に含まれる各等式は以下のいずれかの形式をとる．

$$x \sim f(x_1, \ldots, x_n) \qquad \text{または} \qquad x \sim y$$

ここで，x および $x_1, \ldots, x_n$ は束縛変数，y は自由変数である．

アルゴリズムの実行状態は，$\langle U, D, \sigma \rangle$ の 3 つ組で表現される．各要素は以下のように定義される．

- U（未処理の等式集合）：現在処理を待っている等式制約の集合．
- D（展開済みの変数ペアの履歴）：すでに根の記号の一致が確認され，展開処理を通過した変数ペアの履歴集合（無限木の循環検知に使用）．
- σ（代入）：実行過程で逐次構築される単一化子．初期値は恒等代入 id とする．

アルゴリズムは初期状態 $\langle S_0, \emptyset, id \rangle$ から実行を開始する．項 s と項 t の単一化手続きでは，

$$s = \mathcal{E}_s^*(x), \qquad t = \mathcal{E}_t^*(y)$$

となる正則システム $\mathcal{E}_s, \mathcal{E}_t$，変数 x, y を入力とし，

$$S_0 = \mathcal{E}_s \cup \mathcal{E}_t \cup \{x \sim y\}$$

とする．なお，束縛変数の重複を避けるために，一般性を失うことなく，

$$V(\mathcal{E}_s) \cap V(\mathcal{E}_t) = \emptyset$$

としておく．また，代入 σ のドメインを

$$Dom(\sigma) = \{x \in V \mid \sigma(x) \neq x\}$$

と定義する．

また，導出の過程において，U は $x \sim f(x_1, \ldots, x_n)$ $(x_1, \ldots, x_n$ は束縛変数$)$，または $x \sim y$（x, y の片方または両方が束縛変数）の形の等式のみを含み，D は $x \sim y$（x, y の両方が束縛変数）の形の等式のみを含むものとする．$x_1 \sim x_2, x_2 \sim x_3, \ldots, x_n \sim t \in U$，$t \notin BV$，および $x_1, \ldots, x_n \in BV$ のとき，$x_1 \overset{*}{\underset{U}{\sim}} t$ と示す．

6.2 単一化の推論規則

アルゴリズムは，各ステップで未処理集合 U から等式 $x \sim y$ を 1 つ取り出し，推論規則に従って状態遷移を行う．規則の上部において未処理集合を $U \uplus \{x \sim y\}$（ただし $x \sim y \notin U$）と表し，下部において未処理集合に $x \sim y$ を含めないことで，取り出された対 $x \sim y$ が未処理集合から完全に除外されることを明示する．

Decompose（分解）：取り出した対 $x \sim y$ について x，y の根が同じ関数記号の場合に適用される．ペア $x \sim y$ が履歴集合 D に存在するかどうかで分岐する．
（未処理の場合：子のペアを展開し，現在の変数ペアを履歴 D に追加）

$$\frac{\langle U \uplus \{x \sim y\}, D, \sigma \rangle}{\langle U \cup \{x_1 \sim y_1, \ldots, x_n \sim y_n\}, D \cup \{x \sim y\}, \sigma \rangle}$$

$$x \overset{*}{\underset{U}{\sim}} f(x_1, \ldots, x_n),\ y \overset{*}{\underset{U}{\sim}} f(y_1, \ldots, y_n),\ (x \sim y) \notin D$$

（処理済みの場合：循環構造に到達したとみなし，展開をスキップ）

$$\frac{\langle U \uplus \{x \sim y\}, D, \sigma \rangle}{\langle U, D, \sigma \rangle} \quad \left(\begin{array}{l} x \overset{*}{\underset{U}{\sim}} f(\ldots) \\ y \overset{*}{\underset{U}{\sim}} f(\ldots) \\ (x \sim y) \in D \end{array} \right)$$

Bind（代入）：取り出したペアの片方 z が自由変数である場合，代入の更新と U 内の変数の置換を行う．

$$\frac{\langle U \uplus \{x \sim y\}, D, \sigma \rangle}{\langle U\{z \mapsto t\}, D, \{z \mapsto t\} \circ \sigma \rangle} \quad \left(\begin{array}{l} x \overset{*}{\underset{U}{\sim}} z \in FV \\ y \overset{*}{\underset{U}{\sim}} t \end{array} \right)$$

ここで $U\{z \mapsto t\}$ は，U 内のすべての等式の両辺に出現する変数 z を t に置換する操作を指す．

Clash（衝突）：両辺の変数の根の関数記号が異なる場合，単一化不能として異常終了（$\bot$）する．

$$\frac{\langle U \uplus \{x \sim y\}, D, \sigma\rangle}{\bot \ \text{(Failure)}} \quad \left(\begin{array}{l} x \overset{*}{\underset{U}{\sim}} f(\dots) \\ y \overset{*}{\underset{U}{\sim}} g(\dots) \\ f \neq g \end{array} \right)$$

Delete（削除）：取り出したペアが全く同じ変数の場合，自明な等式として削除する．

$$\frac{\langle U \uplus \{x \sim y\}, D, \sigma\rangle}{\langle U, D, \sigma\rangle} \quad (x = y)$$

Remove(移行)：分解と代入がすべて終わった後で D に $x \sim y$ を移動．

$$\frac{\langle U \uplus \{x \sim y\}, D, \sigma\rangle}{\langle U, D \cup \{x \sim y\}, \sigma\rangle} \quad \left(\begin{array}{l} x \overset{*}{\underset{U}{\sim}} t \Rightarrow t \notin FV \text{ かつ} \\ x \overset{*}{\underset{U}{\sim}} f(x_1, \dots, x_n) \\ \wedge y \overset{*}{\underset{U}{\sim}} g(y_1, \dots, y_m) \\ \Rightarrow f = g,\ n = m, \\ \forall i = 1, \dots, m. \\ \ x_i \sim y_i \in U \cup D \end{array} \right)$$

$$\text{or} \quad (\forall t.\ x \sim^*_U t \Rightarrow t \in FV) \text{ かつ}$$
$$\left(x \sim^*_U t,\ x \sim^*_U t' \Rightarrow t = t'\right)$$

アルゴリズムは，「変数 $\sim$ 変数」の等式制約がすべて解消され，$\langle U_{end}, D_{end}, \sigma_{end}\rangle$（ただし U_{end} には関数定義等式 $x \sim f(\dots)$ のみが残る）に到達した時点で成功終了する．

6.3 状態遷移における不変量

アルゴリズムが状態遷移を通して解の同値性を保ち続けることを示すために，任意の実行状態 $\langle U, D, \sigma\rangle$ に対する「解」を定義する．代入 θ が状態 $\langle U, D, \sigma\rangle$ の解であるとは，以下の 2 条件を満たすことをいう．

1. 任意の $(s \sim t) \in U \cup D$ について $s\theta = t\theta$ が成立する．これを θ が方程式系 $U \cup D$ の解であるという．
2. ある代入 δ が存在して $\theta = \delta \circ \sigma$ と表される．

これを $\sigma \leq \theta$ と記す．

本アルゴリズムの導出過程において，未処理集合 U および履歴集合 D に含まれる等式は常に特定の形式を保つ．この構造的な性質を以下の補題として示す．

補題 6.2（等式の構造的性質）
アルゴリズムの導出の任意のステップにおける実行状態 $\langle U, D, \sigma\rangle$ において，以下が成り立つ．

- U は，$x \sim f(x_1, \dots, x_n)$（$x_1, \dots, x_n$ は束縛変数），または $x \sim y$（x, y の片方または両方が束縛変数）の形の等式のみを含む．
- D は，$x \sim y$（x, y の両方が束縛変数）の形の等式のみを含む．

(証明) 初期状態 $\langle S_0, \emptyset, id\rangle$ において，カノニカルシステムの定義により $U(= S_0)$ は条件を満たし，$D = \emptyset$ であるため自明に成立する．以後の推論規則による遷移において，Decompose 規則は U に束縛変数同士の等式 $x_i \sim y_i$ を追加し，D に束縛変数同士の等式 $x \sim y$ を追加するのみであるため条件は保たれる．また，Bind 規則は自由変数への代入と消去のみを行うが，代入される変数は $x \sim z$（x は束縛変数）の形で出現しており，z には $f(x_1, \dots, x_n)$（$x_1, \dots, x_n$ は束縛変数）または自由変数が代入されるため，条件は保たれる．その他の規則は等式の削除のみを行うため，いかなる状態遷移後においてもこの構造的性質は常に保たれる． □

補題 6.3（変数の独立性）
遷移前の状態において $Var(U \cup D) \cap Dom(\sigma) = \emptyset$ が成立していると仮定すると，すべての推論規則による 1 ステップの遷移後の状態 $\langle U', D', \sigma'\rangle$ においても $Var(U' \cup D') \cap Dom(\sigma') = \emptyset$ が成立する．

(証明) Bind 規則において，自由変数 z が代入のドメインに追加されるが，置換 $U\{z \mapsto t\}$ により U' から z は消去される．また，z は自由変数であるため補題 6.2 より $z \notin Var(D)$ であり，$Var(U' \cup D') \cap \{z\} = \emptyset$ となる．他の規則では代入は変化せず，変数が減少するのみであるため，独立性は常に保たれる． □

補題 6.4（1 ステップの解の同値性）

任意の代入 τ が遷移前の状態の解であることと，遷移後の状態の解であることは同値である．

（証明）Decompose 規則では同値な制約の展開または重複削除が行われるのみで解は不変である．Bind 規則では，補題 6.2 より $z \notin Var(D)$ であるため $D\{z \mapsto t\} = D$ が成立し，代入の合成操作に対しても解の集合は保たれる． □

補題 6.5（複数ステップの解の同値性）
初期状態 $\langle S_0, \emptyset, id\rangle$ から最終状態に至る導出列において，ある代入が初期状態の解であることと，最終状態の解であることは同値である．

（証明）補題 6.4，補題 6.5 を用いて，導出のステップ数に関する数学的帰納法により示される． □

補題 6.6 (カノニカル等式系)

$$\langle S_0, \emptyset, id\rangle \to^* \langle U_i, D_i, \sigma_i\rangle$$

とする．

$$\widehat{U_i} := U_i \setminus (V \times V)$$

とおくと，$\widehat{U_i}$ はカノニカル等式系である．

（証明）$\langle S_0, \emptyset, id\rangle = \langle U_0, D_0, \sigma_0\rangle$ のときは明らか．Decompose 規則は $x \sim y \in V \times V$ しか追加しないこと，および Bind 規則は $x \sim y$ を $x \sim y'$ または $x \sim f(y_1, \ldots, y_n)$ の形にしか変えないことから明らかである． □

また、補題 6.6 から U_{end} がカノニカル等式系となることが従う．

6.4 単一化子の正当性と余帰納的等価性の成立

アルゴリズムが成功終了した際に出力される代入 σ_{end} が，初期方程式系 S_0 に対する単一化子であることを証明する．

補題 6.7 (代入適用の保存)

$$\langle U_i, D_i, \sigma_i\rangle \Rightarrow^* \langle U_{end}, D_{end}, \sigma_{end}\rangle$$

とする．このとき，任意の $x \in V$ について，

$$U^*_{end}(x) = \widehat{U}^*_i(x) U^*_{end}(\sigma_{end})$$

が成立する．ここで $U^*_{end}(\sigma_{end})$ は

$$U^*_{end}(\sigma_{end})(z) = U^*_{end}(\sigma_{end}(z))$$

より定まる代入を表す．

（証明）$\langle U_i, D_i, \sigma_i\rangle \Rightarrow^* \langle U_{end}, D_{end}, \sigma_{end}\rangle$ の長さに関する帰納法で示す．長さが 0 のときは明らかである．$\langle U_i, D_i, \sigma_i\rangle \Rightarrow \langle U_{i+1}, D_{i+1}, \sigma_{i+1}\rangle \Rightarrow^* \langle U_{end}, D_{end}, \sigma_{end}\rangle$ において，適用された推論規則で場合分けを行う．なお，Clash 規則は仮定より適用されていない．

- Decompose 規則および Delete 規則：$\widehat{U}_i = \widehat{U}_{i+1}$ となるため，帰納法の仮定より従う．
- Bind 規則：$\widehat{U}_{i+1} = \widehat{U}_i\{z \mapsto t\}$ となるが，

  $$\begin{aligned} U^*_{end}(\sigma_{end})(z) &= U^*_{end}(\sigma_{end}(z)) \\ &= U^*_{end}(\sigma_{end}(t)) \\ &= U^*_{end}(\sigma_{end})(t) \end{aligned}$$

  より，

  $$\widehat{U}^*_{i+1}(x) U^*_{end}(\sigma_{end}) = \widehat{U}^*_i(x) U^*_{end}(\sigma_{end})$$

  を得る．

□

補題 6.8（健全性と余帰納的等価性の成立）
D_{end} に含まれる任意のペア $u \sim v$ に対し，$U^*_{end}(u) = U^*_{end}(v)$ が成立する．

（証明）無限木の間の二項関係 R を

$$R = \{(U^*_{end}(u), U^*_{end}(v)) \mid (u \sim v) \in D_{end}\}$$

と定義する．任意の $(U^*_{end}(u), U^*_{end}(v)) \in R$ を取る．Decompose 規則が適用されたことにより，u と v の根の関数記号 f およびアリティ n は一致する．従って $U^*_{end}(u)$ と $U^*_{end}(v)$ の根記号も一致する．さらに，その子ノードのペア $(u_i \sim v_i)$ は，アルゴリズムの実行過程で D_{end} に追加されたか，Delete や Bind 規則によって同一化処理が完了しているため，閉包性が保たれている．したがって，すべての $i \in \{1, \ldots, n\}$ について子ノードのペアも $(U^*_{end}(u_i), U^*_{end}(v_i)) \in R$ となる．これにより関係 R は等価性オペレータの条件を満たし，$R \subseteq T_{\approx}(R)$ となる．最大不動点による余帰納法原理より $R \subseteq \approx$ が導かれ，D_{end} に登録されたすべての変数ペアについて $U^*_{end}(u) \approx U^*_{end}(v)$ が証明された． □

定理 6.9（単一化子の正当性）
アルゴリズムが成功終了したとき，出力された代入

$\mathcal{E}^*_{end}(\sigma_{end})$ は任意の $x \sim y \in S_0$ について $S^*_0(x)$ と $S^*_0(y)$ の単一化子である．

（証明） $x \sim y \in S_0$ とすると，$x \sim y \in D_{\mathrm{end}}$ なので，補題 6.8 より

$$U^*_{\mathrm{end}}(x) \approx U^*_{\mathrm{end}}(y).$$

よって補題より，

$$S^*_0(x)\,\mathcal{E}^*_{\mathrm{end}}(\sigma_{\mathrm{end}}) = \mathcal{E}^*_{\mathrm{end}}(x) \approx \mathcal{E}^*_{\mathrm{end}}(y) = S^*_0(y)\,\mathcal{E}^*_{\mathrm{end}}(\sigma_{\mathrm{end}}).$$

□

6.5 最汎単一化子の導出

$\mathcal{E}^*_{\mathrm{end}}(\sigma_{\mathrm{end}})$ が最汎単一化子であることを証明する．

定理 6.10（最汎単一化子の導出） アルゴリズムが成功終了して出力した代入 $U^*_{\mathrm{end}}(\sigma_{\mathrm{end}})$ は，等式 $\{S^*_0(x) = S^*_0(y) \mid x \sim y \in S_0\}$ の最汎単一化子である．

（証明） 最汎単一化子の定義に従い，

1. $U^*_{\mathrm{end}}(\sigma_{\mathrm{end}})$ が $\{S^*_0(x) = S^*_0(y) \mid x \sim y \in S_0\}$ の単一化子であること，
2. $\{S^*_0(x) = S^*_0(y) \mid x \sim y \in S_0\}$ の任意の単一化子 ρ に対して $U^*_{\mathrm{end}}(\sigma_{\mathrm{end}}) \leq \rho$ であること，

を示す．条件 (1) は定理 6.9 により既に証明されている．

条件 (2) について，$\{S^*_0(x) = S^*_0(y) \mid x \sim y \in S_0\}$ の任意の単一化子 ρ を取る．$\rho(x)$ の像は $\mathcal{E}^*_s(x), \mathcal{E}^*_t(y)$ の部分項であるため，ある $\hat{\rho} : FV \to V$ が存在して $\rho = S^*_0(\hat{\rho})$ を満たす．因子分解の成立を以下の 3 段階のステップにより示す．

1. $\hat{\rho}$ は S_0 のすべての等式を満たし，かつ $\hat{\rho} = \rho \circ id$（すなわち $id \leq \hat{\rho}$）が成立するため，$\hat{\rho}$ は初期状態 $\langle S_0, \emptyset, id\rangle$ の解である．
2. 補題 6.5（複数ステップの解の同値性）より，初期状態の解である $\hat{\rho}$ は，最終状態 $\langle U_{end}, D_{end}, \sigma_{end}\rangle$ の解でもある．
3. 最終状態の解の定義（第二条件）より，ある代入 δ が存在して

$$\hat{\rho} = \delta \circ \sigma_{end}$$

が成立する．$dom(\sigma_{end}) \subseteq FV$ であることから

$$\mathcal{E}^*_{end}(\hat{\rho}) = \mathcal{E}^*_{end}(S) \circ \mathcal{E}^*_{end}(\sigma_{end})$$

となる．よって

$$\mathcal{E}^*_{end}(\sigma_{end}) \subseteq \mathcal{E}^*_{end}(\hat{\rho}) = S^*_0(\hat{\rho})$$

以上の二条件より，出力された代入 $U^*_{\mathrm{end}}(\sigma_{\mathrm{end}})$ は等式 $\{S^*_0(x) = S^*_0(y) \mid x \sim y \in S_0\}$ に対する最汎単一化子である．

□

7 おわりに

本論文では，正則項における基本的な性質とアルゴリズムについて，余帰納的アプローチに基づく定式化とその正当性の証明を行った．まず，等価性オペレータの最大不動点として定義された無限項の等価性 $\approx$ が同値関係を満たすこと，およびパス関数によって定義される同一性 $=$ と一致することを示した．次に，正則項の等価性判定手続きを状態遷移系として定式化し，成功終了時に得られる等式ペアが等価性 $\approx$ を満たすことを証明した．さらに，等価性を考慮した正則項の単一化手続きを定式化し，そのアルゴリズムの出力が初期等式系に対する最汎単一化子となることを示した．

本研究の意義は，余帰納的定義に基づく無限項や正則項に関する性質の証明を見通しよく整理した点にある．正則項の単一化手続きは文献 [4] で与えられた [2] [3]．本論文で与えた手続きは基本的には同一のものである．これらの論文では余帰納的な定義ではなく，木領域から関数記号と変数への写像とする伝統的な無限項の定義に基づいている．文献 [5] では正則表現による正則項上の単一化手続きが正則項の単一化手続きとなることを厳密に証明しているが，これらも余帰納的な定義には基づいていない．これに対し，本研究では無限項の等価性を等価性オペレータの最大不動点として余帰納的に定式化し，その定式化の正しさを示すとともに，余帰納法原理を証明に直接適用した．これにより，位置に関する帰納法を必要とすることなく，等価性判定手続きおよび単一化手続きの正当性を簡潔かつ直接的に証明した．

今後の課題としては文献 [1] や [6] で示されている正則項や正則項書き換えに関する理論を余帰納的に形式化することなどが挙げられる．

参 考 文 献

[1] Aoto, T. and Ketema, J.: Rational term rewriting revisited: decidability and confluence, *In Proceedings of the 6th International Conference on Graph Transformation (ICGT 2012)*, Lecture Notes in Computer Science (LNCS), Vol. 7562, Springer, 2012, pp. 172–186.

[2] Colmerauer, A.: Equations and inequations on finite and infinite trees, *In Proceedings of the International Conference on Fifth Generation Computer Systems (FGCS 1984)*, 1984, pp. 85–99.

[3] Courcelle, B.: Fundamental properties of infinite trees, *Theoretical Computer Science*, Vol. 25(1983), pp. 95–169.

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

[5] Iwami, M. and Aoto, T.: Disproving Strong Head Normalization and General Productivity Automatically in Infinitary Term Rewriting Systems, コンピュータソフトウェア, Vol. 27, No. 1(2010), pp. 1–29.

[6] 伊井亮祐, 青戸等人: 重なりのある正則項書き換えシステムの合流性条件, 第 *27* 回プログラミングおよびプログラミング言語ワークショップ *(PPL 2025)* 論文集, 2025.