

Title	作用項書換えのための多重集合経路順序
Author(s)	谷内, 亘
Citation	
Issue Date	2025-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/19818
Rights	
Description	Supervisor: 廣川 直, 先端科学技術研究科, 修士 (情報科学)

修士論文

作用項書換えのための多重集合経路順序

2310165 谷内 亘

主指導教員	廣川	直
審査委員主査	廣川	直
審査委員	石井	大輔
	緒方	和博
	高木	翼

北陸先端科学技術大学院大学
先端科学技術研究科
(情報科学)

令和 7 年 2 月

Abstract

In this thesis, we introduce a higher-order version of the multiset path order where λ -abstractions and types are absent. *Reduction orders* are well-founded orders on terms. It can show termination of term rewriting systems. For example, the rewrite system $\mathcal{R}_1$

$$\begin{array}{ll} x + 0 \rightarrow x & x \times 0 \rightarrow 0 \\ x + s(y) \rightarrow s(x + y) & x \times s(y) \rightarrow x + (x \times y) \end{array}$$

is terminating if there exists some reduction order $>$ such that the following inequalities hold:

$$\begin{array}{ll} x + 0 > x & x \times 0 > 0 \\ x + s(y) > s(x + y) & x \times s(y) > x + (x \times y). \end{array}$$

One of the most popular reduction orders is the multiset path order, introduced by Dershowitz (1982). This order takes an order on function symbols, called precedence. Since the multiset path order with the precedence $\times > + > s$ satisfies the above inequality constraints, the termination of $\mathcal{R}_1$ is concluded.

Applicative terms represent higher-order functions without λ -abstraction. For instance, the higher-order function `map` used in functional programming is modeled by rewrite system $\mathcal{R}_2$

$$\begin{array}{ll} \text{add } x \ 0 \rightarrow x & \text{app nil } ys \rightarrow ys \\ \text{add } x \ (s \ y) \rightarrow s \ (\text{add } x \ y) & \text{app (cons } x \ xs) \ ys \rightarrow \text{cons } x \ (\text{app } xs \ ys) \\ \text{mul } x \ 0 \rightarrow 0 & \text{map } f \ \text{nil} \rightarrow \text{nil} \\ \text{mul } x \ (s \ y) \rightarrow \text{add } x \ (\text{mul } x \ y) & \text{map } f \ (\text{cons } x \ xs) \rightarrow \text{cons } (f \ x) \ (\text{map } f \ xs). \end{array}$$

Unfortunately, the multiset path order cannot orient the all rules in $\mathcal{R}_2$, and it even fails to orient the curried version $\text{mul } x \ (s \ y) \rightarrow \text{add } x \ (\text{mul } x \ y)$ of the definition of multiplication $x \times s(y) > x + (x \times y)$. If we have a reduction order that can orient the all rules in $\mathcal{R}_2$ then we can show that the following equation

$$\text{map } f \ (\text{app } xs \ ys) = \text{app } (\text{map } f \ xs) \ (\text{map } f \ ys)$$

holds for all ground constructor terms consisting of 0, s, nil, and cons.

Remarkable character of the multiset path order is that it can show termination of term rewriting systems that represent primitive recursive functions. However, we are confronted with some problems when adapting the multiset path order to applicative terms. Firstly, variables at head positions do not preserve the direction when they are substituted. For instance, consider the precedence $f > g$. Thus, $x\ y > f\ y$ holds but if we substitute g into x then $g\ y \not> f\ y$ since $f > g$. Secondly, the subterm property also breaks the direction when we attach arguments to both hand sides. For example, we obtain $g\ f > f$ by the subterm property, but $g\ f\ a \not> f\ a$ due to the precedence $f > g$. Finally, proving well-foundedness of reduction orders are difficult. The proof method of well-foundedness of orders that have the subterm property are provided. But since our order lacks this property, we need another proof technique.

Here we summarize the problems as follows:

1. stability of head variable terms,
2. compatibility with contexts, and
3. well-foundedness proof of orders without the subterm property.

In this thesis we give the following solutions for the above problems:

1. *uncurrying* and Tanaka's method,
2. introducing an *arity assignment* to restrict the comparing of terms, and
3. borrowing the proof technique of well-foundedness from complexity analysis

Then we obtain a first path ordering on applicative terms that can prove the termination of $\mathcal{R}_2$.

目次

第 1 章	はじめに	1
1.1	研究目的	1
1.2	既存手法	4
1.3	成果	5
1.4	構成	5
第 2 章	準備	6
2.1	順序	6
2.2	抽象書換え系	7
2.3	項書換え系	8
2.4	簡約順序	10
2.5	作用項書換え系	11
第 3 章	多重集合経路順序	12
3.1	定義	12
3.2	原始帰納的関数との関係	13
3.3	部分項性と作用項との関係	14
第 4 章	作用型多重集合経路順序	16
4.1	定義	16
4.2	整礎順序	18
4.3	文脈について閉じた順序の構成	25
4.4	代入について閉じた順序の構成	28
4.5	$\triangleright_{\text{ampo}}$ と $\sqsubset_{\text{ampo}}$ の関係	30
第 5 章	自動化	33

5.1	漸増性	33
5.2	制約式	34
第 6 章	評価	37
6.1	例	37
6.2	実験	41
6.3	実験の評価	41
6.4	グレースフルネス	44
第 7 章	結論	47
7.1	uncurrying 変換との対応	47
7.2	作用項上の簡約順序	48
7.3	自動定理証明への応用	49
7.4	今後の課題	51
参考文献		52

第 1 章

はじめに

大規模なソフトウェアについて、その正しさの検証の全てを人手によって行うことは時間的に困難であり、検証を自動化するシステムが必要になる。項書換えはそのような検証システムの理論的基礎となる計算モデルである。簡約順序は項書換えにおける重要な概念であり、停止性の証明手法のみに留まらず、検証システムに応用可能な自動定理証明手法の構成要素にもなる。

本章の構成は以下の通りになる。1.1 節では掛け算を例に取り、本研究の対象である簡約順序の具体的な役割や応用例を紹介する。続けてより実用的な例であるプログラム検証についての応用を見る。1.2 節では既存研究を取り上げ、1.3 節では本研究における成果をまとめる。そして最後に論文全体の構成について説明する。

1.1 研究目的

次の等式系 $\mathcal{E}_1$ を考える。

$$\begin{array}{ll} x + 0 = x & x \times 0 = 0 \\ x + s(y) = s(x + y) & x \times s(y) = x + (x \times y) \end{array}$$

この等式系はペアノ数上の掛け算の定義を表している。表記の簡単化のために $\bar{n} = s^n(0)$ とする。 $\bar{2} \times \bar{2} = \bar{4}$ は $\mathcal{E}_1$ のもとで以下のように導出される。

$$\begin{aligned}
\bar{2} \times \bar{2} &= \bar{2} + (\bar{2} \times \bar{1}) \\
&= \bar{2} + (\bar{2} + (\bar{2} \times \bar{0})) \\
&= \bar{2} + (\bar{2} + \bar{0}) \\
&= \bar{2} + \bar{2} \\
&= s(\bar{2} + \bar{1}) \\
&= s(s(\bar{2} + \bar{0})) \\
&= \bar{4}
\end{aligned}$$

次にこの導出を自動化する方法を考える。導出において $\bar{2} \times \bar{2}$ の変形は $\mathcal{E}_1$ の等式の左辺から右辺への変形のみが用いられる。そのため導出を自動化するために $\mathcal{E}_1$ の等式を左辺から右辺への変形に限定する。このとき $\mathcal{E}_1$ の各等式は書換え規則としてみなすことができる。このような書換え規則の集合を項書換え系と呼び、 $\mathcal{E}_1$ から得られる項書換え系 $\mathcal{R}_1$ は次のようになる。

$$\begin{array}{ll}
x + \bar{0} \rightarrow x & x \times \bar{0} \rightarrow \bar{0} \\
x + s(y) \rightarrow s(x + y) & x \times s(y) \rightarrow x + (x \times y)
\end{array}$$

$\mathcal{R}_1$ を用いると $\bar{2} \times \bar{2}$ は次のように書換えられる。

$$\begin{aligned}
\bar{2} \times \bar{2} &\rightarrow_{\mathcal{R}_1} \bar{2} + (\bar{2} \times \bar{1}) \\
&\rightarrow_{\mathcal{R}_1} \bar{2} + (\bar{2} + (\bar{2} \times \bar{0})) \\
&\rightarrow_{\mathcal{R}_1} \bar{2} + (\bar{2} + \bar{0}) \\
&\rightarrow_{\mathcal{R}_1} \bar{2} + \bar{2} \\
&\rightarrow_{\mathcal{R}_1} s(\bar{2} + \bar{1}) \\
&\rightarrow_{\mathcal{R}_1} s(s(\bar{2} + \bar{0})) \\
&\rightarrow_{\mathcal{R}_1} \bar{4}
\end{aligned}$$

ここで $\bar{4}$ はこれ以上 $\mathcal{R}_1$ のどの規則を用いても書換えることができない。このような項を正規形という。

書換え規則次第では無限の書換えが起こる場合がある。 $\mathcal{E}_1$ に掛け算の交換律

$$x \times y = y \times x$$

を追加した $\mathcal{E}'_1$ を考える。このとき $\mathcal{E}'_1$ を表す項書換え系 $\mathcal{R}'_1$ は $x \times y \rightarrow y \times x$ という書換え規則を $\mathcal{R}_1$ に加えることによって得られる。すると $\mathcal{R}'_1$ による $\bar{2} \times \bar{1}$ の書換えは

$$\bar{2} \times \bar{1} \rightarrow_{\mathcal{R}'_1} \bar{1} \times \bar{2} \rightarrow_{\mathcal{R}'_1} \bar{2} \times \bar{1} \rightarrow_{\mathcal{R}'_1} \cdots$$

となって停止しない。項書換え系 $\mathcal{R}$ が無限の書換え列を持たないとき $\mathcal{R}$ は停止性を持つという。自動定理証明においては、等式系から項書換え系を得る際に停止性を持つよう書換え規則を向き付けることが望ましい。このとき用いられるのが簡約順序である。項書換え系 $\mathcal{R}$ の各規則 $\ell \rightarrow r$ について $\ell > r$ となる簡約順序 $>$ が存在すれば $\mathcal{R}$ は停止性を持つ。代表的な簡約順序としては Dershowitz の 多重集合経路順序 [7] がある。これは優先順序と呼ばれる関数記号上の順序に基づく簡約順序であり、任意の原始帰納的関数に対してそれを表現する項書換え系の停止性を証明することができる [9]。優先順序 $\times > + > s > 0$ を用いれば $\mathcal{R}_1$ の停止性が多重集合経路順序 $>_{\text{mpo}}$ によって証明される。

簡約順序は停止性の証明のみならず、等式系の上で成り立つ命題の証明にも用いることができる。例えば $\mathcal{E}_1$ の下で、自然数を表す項の集合 $\{\bar{n} \mid n \in \mathbb{N}\}$ 上で成り立つの足し算の交換律 $x + y = y + x$ は $>_{\text{mpo}}$ による整礎帰納法によって次のように示すことができる。ただし証明では2つの補題 $0 + x = x, s(x) + y = s(x + y)$ を用いている。

命題 1.1. $x + y = y + x$ が全ての $x, y \in \{\bar{n} \mid n \in \mathbb{N}\}$ について成り立つ。

証明. 優先順序を $+ > s > 0$ とする。命題を $>_{\text{mpo}}$ の多重集合拡張（詳しい定義は2章で述べる） $>_{\text{mpo}}^{\text{mul}}$ に関する $\{x + y, y + x\}$ 上の整礎帰納法で示す。 y についての場合分けを行う。 $y = 0$ のとき $x + 0 = x = 0 + x$ となり題意が成り立つ。次に $y = s(y')$ のとき $x + s(y') = s(x + y')$ かつ $s(y') + x = s(y' + x)$ となる。また

$$\{x + s(y'), s(y') + x\} >_{\text{mpo}}^{\text{mul}} \{x + y', y' + x\}$$

が成り立つことより帰納法の仮定から $x + y' = y' + x$ を得る。したがって次を得る。

$$x + y = x + s(y') = s(x + y') = s(y' + x) = s(y') + x = y + x \quad \square$$

このように帰納法を用いる証明を自動的に行う手法として書換え帰納法 [20] がある。書換え帰納法では等式系を与えられた簡約順序から得られる項書換え系として扱う。MPO と書換え帰納法を用いると先に証明した命題 $x + y = y + x$ を2つの補題 $0 + x = x, s(x) + y = s(x + y)$ を含めて自動的に証明することができる。

Haskell のような関数型言語によるプログラムを等式系とみなせば、プログラムについての性質を等式として表すことができる。例えば次の等式系 $\mathcal{E}_{\text{map}}$

$$\begin{aligned} \text{map } f \text{ nil} &= \text{nil} & \text{app nil } ys &= ys \\ \text{map } f (\text{cons } x \text{ } xs) &= \text{cons } (f \ x) (\text{map } f \ xs) & \text{cons } x (\text{app } xs \ ys) &= \text{cons } x (\text{app } xs \ ys) \end{aligned}$$

は高階関数 map とリストの結合 app の定義であり、等式

$$\text{map } f (\text{app } xs \ ys) = \text{app } (\text{map } f \ xs) (\text{map } f \ ys)$$

は map と app の上で成り立つ性質である。先に示した足し算の交換律の証明と同様に、この等式を証明することを考える。そのために $\mathcal{E}_1$ から $\mathcal{R}_1$ を求めたように $\mathcal{E}_{\text{map}}$ から項書換え系を求めることにする。しかし一階の項書換えでは map のような高階関数を直接表現することは難しい。これは $f(x)$ のような変数に対する適用は表現できないためである。そのため項書換えで高階関数を扱うための枠組みが提案されており、その一つに作用項書換えがある。作用項は変数、定数項、2 引数関数である関数適用記号 $\circ$ のみからなる項である。作用項では関数適用 $f(x_1, x_2, \dots, x_n)$ は $((f \circ x_1) \circ x_2) \circ \dots \circ x_n$ と表される。作用項を用いると $\mathcal{E}_{\text{map}}$ の 2 つ目の等式は次のように表現される。

$$(\text{map} \circ f) \circ ((\text{cons} \circ x) \circ xs) = (\text{cons} \circ (f \circ x)) \circ ((\text{map} \circ f) \circ xs)$$

表記の簡単化のために並置記法 $((t_1 \circ t_2) \circ t_3) \circ \dots \circ t_n = t_1 \ t_2 \ t_3 \ \dots \ t_n$ を用いれば $\mathcal{E}_{\text{map}}$ は次のようになる。

$$\begin{aligned} \text{map } f \text{ nil} &= \text{nil} & \text{app nil } ys &= ys \\ \text{map } f (\text{cons } x \ xs) &= \text{cons } (f \ x) (\text{map } f \ xs) & \text{app } (\text{cons } x \ xs) \ ys &= \text{cons } x (\text{app } xs \ ys) \end{aligned}$$

$\mathcal{E}_{\text{map}}$ を $>_{\text{mpo}}$ で向き付けられるならば、掛け算の場合と同様に $\text{map } f(\text{app } xs \ ys) = \text{app } (\text{map } f \ xs) (\text{map } f \ ys)$ を自動で証明することが可能になる。しかし $>_{\text{map}}$ は $\mathcal{R}_{\text{map}}$ 向き付けることができない。

またこの順序はカーリー化した項については $>_{\text{mpo}}$ と同様の向き付けが保たれることが望ましい。なおここで言うカーリー化とは、例えば次の一階の項 $\text{add}(x, s(y))$ から作用項 $\text{add } x \ (s \ y)$ を得る変換である。

1.2 既存手法

一階の簡約順序としては再帰経路順序と呼ばれる Dershowitz の多重経路順序、Kamin と Lévy の辞書式経路順序が知られている。高階の項書換えに用いる簡約順序を得るアプローチとして、この再帰経路順序を高階版へ拡張する試みが為されてきた。Jouannaud と Rubio の高階再帰経路順序 [14] は高階書換えという型有りの体系における再帰経路順序の拡張である。しかし型無しの体系である作用項書換えのための新たな簡約順序は長らく提案されてこなかった。これは部分項性に起因する問題から、作用項上で満足に機能する再帰経路順序の拡張が困難であったためである。lambda-free 再帰経路順序 [3] は Blanchette らによる再帰経路順序の作用項に対する拡張である。lambda-free 再帰経路順序はカーリー化した項については再帰経路順序による元の項の向き付けと一致する。しかし

この順序は文脈について閉じていない。例えば $f(g\ a) > g$ となるが両辺に引数 b を追加したとき $f(g\ a)\ b \not> g\ b$ になってしまう。そのため lambda-free 再帰経路順序は簡約順序ではない。しかし 2021 年に Bentkamp が lambda-free RPO のアイデアをもとに作用項上の簡約順序である 埋め込み経路順序 (embedding path order, EPO) [2] を提案する。EPO は作用項上の簡約順序ではあるが `map` のような高階関数について向き付けを得ることができない。また EPO は再帰経路順序を基にした順序ではあるが、カーリー化した項についても $\text{mul } x\ (s\ y) > \text{add } x\ (\text{mul } x\ y)$ というような期待した向き付けが得られない。従って作用項に対する再帰経路順序の拡張は未だ果たされていない。一方で多項式解釈順序 [16] という再帰経路順序とは異なる簡約順序に基づく作用型解釈順序 [26] が 2023 年に田中によって提案された。これは多項式解釈順序 [16] と uncurrying 変換 [13] という手法を組み合わせたものである。しかし多項式解釈順序は適切な解釈の発見の自動化が困難であり `map` の向き付けは可能であるが掛け算の向き付けに失敗する。

1.3 成果

本論文では uncurrying 変換と多重集合経路順序 (MPO) に基づく作用項上の新たな簡約順序である作用型多重集合経路順序 (AMPO) を提案する。AMPO はカーリー化した項について、元の項の MPO による向き付けと一致する。つまり次の順序関係

$$x \times s(y) >_{\text{mpo}} x + (x \times y)$$

が得られるならば、カーリー化した $\text{mul } x\ (s\ y) \rightarrow \text{add } x\ (\text{mul } x\ y)$ について AMPO 上の順序関係

$$\text{mul } x\ (s\ y) > \text{add } x\ (\text{mul } x\ y)$$

が成り立つ。

1.4 構成

本論文の構成は次の通りである。2 章では項書換えについての基本的な定義と性質を述べる。3 章では提案する順序の基礎となる MPO について述べる。4 章では AMPO の定義と簡約順序であることの証明を行う。5 章では AMPO を用いた停止性証明の自動化について述べる。6 章では 5 章の手法に基づく実装の実験結果より AMPO と他の簡約順序との比較と AMPO の有効性について議論する。

第 2 章

準備

本章では以降において用いる概念の定義と性質について説明する。また項書換えにおける基本的な定義と性質については [1] に基づいている。

2.1 順序

定義 2.1. X を集合とする。 X 上の二項関係 R とは直積集合 $X \times X$ の部分集合である。このとき $(x, y) \in R$ を $x R y$ と表す。 X 上の二項関係 R, S の合成 $R \cdot S$ を次に定める。

$$R \cdot S = \{(x, z) \mid \exists y \in X. x R y \wedge y S z\}$$

定義 2.2. 集合 X 上の二項関係 R についてそれぞれ

- 全ての $x \in X$ について $x R x$ となるとき**反射的**
- 全ての $x \in X$ について $\neg x R x$ となるとき**非反射的**
- 全ての $x, y, z \in X$ について $x R y$ かつ $y R z$ ならば $x R z$ となるとき**推移的**
- 無限降下例 $x_1 R x_2 R x_3 R \cdots$ を持たないとき**整礎**

という。非反射的かつ推移的な二項関係 $>$ を**狭義半順序**といい、整礎な狭義半順序を**整礎順序**という。反射的かつ推移的な二項関係 $\succeq$ を**擬順序**という。集合 X 上の擬順序 $\succeq$ について $x \succeq y$ かつ $y \succeq x$ であるとき $x \sim y$ とする。

例えば自然数上の大小関係 $>$ は非反射的かつ推移的であるため狭義半順序となる。また $0 > n$ となる自然数 n は存在しないため $>$ は整礎順序である。

次に狭義半順序のいくつかの拡張を定義する。

定義 2.3. $>$ を集合 X 上の狭義半順序とする。 $>$ の辞書式拡張 $>^{\text{lex}}$ とは 集合 X^* 上の順序で次のように定義されるものである。 $(x_1, x_2, \dots, x_m) >^{\text{lex}} (y_1, y_2, \dots, y_n)$ が成り立つとは $x_1 > y_1$ または $(x_2, \dots, x_m) >^{\text{lex}} (y_2, \dots, y_n)$ となることである。

定義 2.4. X を集合とする。写像 $M : X \rightarrow \mathbb{N}$ について $M(x) > 0$ となる x が有限個であるとき M を X 上の **多重集合** といい、 X 上の全ての多重集合を $\mathcal{M}(X)$ で表す。 $M(x) > 0$ ならば $x \in M$ とし、空の多重集合を $\emptyset$ で表す。 $M_1, M_2 \in \mathcal{M}(X)$ について合併 $M_1 \uplus M_2$ を $(M_1 \uplus M_2)(x) = M_1(x) + M_2(x)$ と定め、差 $M_1 - M_2$ を次のように定める。

$$(M_1 - M_2)(x) = \begin{cases} M_1(x) - M_2(x) & \text{if } M_1(x) \geq M_2(x) \\ 0 & \text{otherwise} \end{cases}$$

また 全ての $x \in X$ について $M_1(x) \leq M_2(x)$ のとき $M_1 \subseteq M_2$ と表す。

定義 2.5. $>$ を 集合 X 上の狭義半順序とする。 $>$ の**多重集合拡張** (multiset extension) $>^{\text{mul}}$ は次のように定義される $\mathcal{M}(X)$ 上の二項関係である。 $M_1 >^{\text{mul}} M_2$ であるとは $M_2 = (M_1 - X) \uplus Y$ となる $X, Y \in \mathcal{M}(X)$ が存在し、かつ条件

1. $\emptyset \neq X \subseteq M_1$
2. 全ての $y \in Y$ について $x > y$ となる $x \in X$ が存在する

を満たすことである。

定理 2.6 ([10]). $>$ が集合 X 上の整礎順序ならば $>^{\text{mul}}$ は $\mathcal{M}(X)$ 上の整礎順序である。

2.2 抽象書換え系

定義 2.7. 集合 A と A 上の二項関係 $\rightarrow$ の組 $(A, \rightarrow)$ を**抽象書換え系** (abstract reduction system) という。また次の記法を [1] に基づいて定める。

- $\rightarrow^0 = \{(a, a) \mid a \in A\}$
- $\rightarrow^n = \rightarrow^{n-1} \cdot \rightarrow$ (n 回の合成)
- $\rightarrow^+ = \bigcup_{n>0} \rightarrow^n$ (推移閉包)
- $\rightarrow^* = \rightarrow^0 \cup \rightarrow^+$ (反射推移閉包)

また $a \in A$ に対して $a \rightarrow b$ となる $b \in A$ が存在しないとき a を**正規形** (normal form) という。

2.3 項書換え系

定義 2.8. シグネチャ $\mathcal{F}$ は関数記号 $f^{(n)}$ の集合である。このとき $n \in \mathbb{N}$ は項数 (arity) という。シグネチャ $\mathcal{F}$ と $\mathcal{F} \cap \mathcal{V} = \emptyset$ となる変数の集合 $\mathcal{V}$ について、 $\mathcal{T}(\mathcal{F}, \mathcal{V})$ を $\mathcal{F}$ と $\mathcal{V}$ からなる項の全体集合とし、項を次のように帰納的に定める。

- 全ての変数 $v \in \mathcal{V}$ は項である。
- $t_1, \dots, t_n \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ かつ $f^{(n)} \in \mathcal{F}$ ならば $f(t_1, \dots, t_n)$ は項である。

また項 t の大きさ $|t|$ を次のように定める。

$$|t| = \begin{cases} 1 & \text{if } t \in \mathcal{V} \\ 1 + \sum_{i=1}^n |t_i| & \text{if } t = f(t_1, \dots, t_n) \end{cases}$$

例 2.9. シグネチャを $\mathcal{F} = \{f^{(2)}, g^{(1)}\}$ とする。このとき $x, g(y), f(x, g(y))$ は項である。また $f(x, g(y))$ の大きさは

$$|f(x, g(y))| = 1 + |x| + |g(y)| = 1 + 1 + 1 + |y| = 1 + 1 + 1 + 1 = 4$$

となる。

定義 2.10. 項 t' が $t = f(t_1, \dots, t_n)$ の部分項 (subterm) であるとは $t_i = t'$ または t' は t_i の部分項であることがいずれかの $1 \leq i \leq n$ について成り立つことである。またこのとき $t \triangleright t'$ とする。

例 2.11. 項 $x + s(x \times s(y))$ の部分項は $x, y, s(y), x \times s(y), s(x \times s(y))$ となる。またこのとき関係

$$x + s(x \times s(y)) \triangleright s(x \times s(y)) \triangleright x \times s(y) \triangleright s(y) \triangleright y$$

が成り立つ。

項上の関係 $\triangleright$ は整礎である。

定義 2.12. 代入 (substitution) σ は有限個の $x \in \mathcal{V}$ についてのみ $\sigma(x) \neq x$ となる $\mathcal{V}$ から $\mathcal{T}(\mathcal{F}, \mathcal{V})$ への写像である。また代入 σ の項 t への適用 $t\sigma$ を次のように定める。

$$t\sigma = \begin{cases} \sigma(t) & \text{if } t \in \mathcal{V} \\ f(t_1\sigma, \dots, t_n\sigma) & \text{if } t = f(t_1, \dots, t_n) \end{cases}$$

また代入 σ, τ の**合成** $\sigma\tau$ を次のように定義する。

$$\sigma\tau = \{x \mapsto (x\sigma)\tau \mid x \in \mathcal{V}\}$$

例 2.13. 次の代入について

$$\sigma = \{x \mapsto s(x), y \mapsto x\} \quad \tau = \{x \mapsto s(0), y \mapsto 0\}$$

項 $x + y$ への σ と $\sigma\tau$ の適用は

$$(x + y)\sigma = s(x) + x \quad (x + y)\sigma\tau = s(s(0)) + s(0)$$

となる。

$s(s(0))$ のような変数を含まない項を**基底項** (ground term) という。集合 $X \subseteq \mathcal{V}$ の全ての要素 x について $\sigma(x)$ が基底項となる代入を**基底代入** (ground substitution) という。ここでは $\tau, \sigma\tau$ は基底代入となる。

また二つの代入 σ, τ とその合成 $\sigma\tau$ について次が成り立つ。

定理 2.14. 任意の代入 σ, τ について $t(\sigma\tau) = (t\sigma)\tau$ が任意の項 t において成り立つ。

定義 2.15. 穴 (hole) $\square$ は $\{\square\} \cap (\mathcal{F} \cup \mathcal{V}) = \emptyset$ となる記号である。 $\square$ を唯一つ含む項を**文脈**という。文脈 C と項 t に対して、 C 中の $\square$ を t で置き換えた項 $C[t]$ を次のように帰納的に定義する。

$$C[t] = \begin{cases} t & \text{if } C = \square \\ f(t_1, \dots, C'[t], \dots, t_n) & \text{if } C = f(t_1, \dots, C', \dots, t_n) \end{cases}$$

定義 2.16. 書換え規則 $\ell \rightarrow r$ は $\ell \notin \mathcal{V}$ かつ $\text{Var}(r) \subseteq \text{Var}(\ell)$ を満たす項の組 (ℓ, r) である。**項書換え系** (term rewriting system, TRS) とは書換え規則の集合である。項書換え系 $\mathcal{R}$ と項 s, t について $s = C[\ell\sigma]$ かつ $t = C[r\sigma]$ となる書換え規則 $\ell \rightarrow r \in \mathcal{R}$ 代入 σ 文脈 C が存在するとき $s \rightarrow_{\mathcal{R}} t$ と表す。

例 2.17. シグネチャ $\mathcal{F} = \{\times^{(2)}, +^{(2)}, s^{(1)}, 0^{(0)}\}$ とする。項書換え系 $\mathcal{R}_1$

$$\begin{array}{ll} x + 0 \rightarrow x & x \times 0 \rightarrow 0 \\ x + s(y) \rightarrow s(x + y) & x \times s(y) \rightarrow x + (x \times y) \end{array}$$

のもとで 項 $s(s(0)) \times s(s(0))$ は正規形 $s(s(s(s(0))))$ を持つ。

項書換え系 $\mathcal{R}$ が無限の書換え列 $t_0 \rightarrow_{\mathcal{R}} t_1 \rightarrow_{\mathcal{R}} t_2 \rightarrow_{\mathcal{R}} \cdots$ を持たないとき $\mathcal{R}$ は**停止性を持つ** という。 $\mathcal{R}_1$ は停止性を持つが、停止性を持たない項書換え系も存在する。例えば次の項書換え系 $\mathcal{R}_2$

$$f(x) \rightarrow f(f(x))$$

は無限の書換え列 $f(x) \rightarrow_{\mathcal{R}_2} f(f(x)) \rightarrow_{\mathcal{R}_2} f(f(f(x))) \rightarrow_{\mathcal{R}_2} \cdots$ を持つ。よって $\mathcal{R}_2$ は停止性を持たない。

2.4 簡約順序

定義 2.18. R を項上の二項関係とし、 s, t を任意の項とする。 R が**代入について閉じている**とは、 $s R t$ ならば $s\sigma R t\sigma$ が全ての代入 σ について成立することである。または R が**文脈について閉じている**とは、 $s R t$ ならば $C[s] R C[t]$ が全ての文脈 C について成立することである。**書換え順序** (rewrite order) とは項上の狭義半順序で代入と文脈について閉じたものである。**簡約順序** (reduction order) とは整礎な書換え順序である。

項書換え系 $\mathcal{R}$ の停止性と簡約順序に関して次の定理が知られている。

定理 2.19. $\mathcal{R}$ が停止性を持ちかつそのときに限り、全ての $\ell \rightarrow r \in \mathcal{R}$ について $\ell > r$ となる簡約順序 $>$ が存在する。

簡約順序の構成方法の一つとして**単純化順序** (simplification order) が知られている。これは部分項性と呼ばれる次の性質を持つ書換え順序である。

定義 2.20 ([17]). R を項上の関係とする。任意の $C \neq \square$ となる文脈 C と項 t について $C[t] R t$ が成立するとき R は**部分項性** (subterm property) を持つという。

単純化順序はシグネチャ $\mathcal{F}$ が有限のとき簡約順序となる。また単純化順序は次に定める項上の関係 $\triangleright_{\text{emb}}$ を包含する。

定義 2.21. $\mathcal{F}$ をシグネチャとする。項書換え系 $\mathcal{Emb}(\mathcal{F})$ を次のように定める。

$$\mathcal{Emb}(\mathcal{F}) = \{f(x_1, \dots, x_i, \dots, x_n) \rightarrow x_i \mid f^{(n)} \in \mathcal{F}, 1 \leq i \leq n\}$$

このとき**埋め込み** (embedding) $s \triangleright_{\text{emb}} t$ を $s \rightarrow_{\mathcal{Emb}(\mathcal{F})}^+ t$ ならば $s \triangleright_{\text{emb}} t$ として定める。

命題 2.22. 書換え順序 $>$ が部分項性を持つとき、かつそのときに限り $\triangleright_{\text{emb}} \subseteq >$ となる。

2.5 作用項書換え系

最後に本研究の対象となる作用項を定義する。

定義 2.23. 項数が 2 である関数記号がただ一つであり、他の全ての関数記号の項数が 0 であるようなシグネチャを**作用型シグネチャ** (applicative signature) という。またこのとき項数が 2 の関数記号を**関数適用記号** (application symbol) 呼ぶ。**作用項** (applicative term) は作用型シグネチャと変数上の項であり、作用項上の項書換え系を**作用項書換え系** (applicative term rewriting system, ATRS) という。並置記法 $t_1 t_2 t_3 \cdots t_n$ は $t_1, \dots, t_n$ を作用項として、次の作用項 $((t_1 \circ t_2) \circ t_3) \circ \cdots \circ t_n$ を表す。ここで $\circ$ は関数適用記号である。

例 2.24. 次のシグネチャ $\Sigma = \{\text{map}^{(0)}, \text{cons}^{(0)}, \text{nil}^{(0)}, \circ^{(2)}\}$ は作用型シグネチャである。このとき $\circ$ は関数適用記号である。次の $x, \text{map}, f \circ x, \text{cons} \circ x, (\text{map} \circ f) \circ ((\text{cons} \circ x) \circ xs)$ は全て Σ 上の作用項である。作用項 $(\text{map} \circ f) \circ ((\text{cons} \circ x) \circ xs)$ は並置記法を用いると $\text{map } f (\text{cons } x xs)$ と表すことができる。また以下は Σ 上の作用項書換え系である。

$$\begin{aligned} \text{map } f \text{ nil} &\rightarrow \text{nil} \\ \text{map } f (\text{cons } x xs) &\rightarrow \text{cons } (f x) (\text{map } f xs) \end{aligned}$$

第 3 章

多重集合経路順序

多重経路順序 [7] は 2 つの関数適用を比較するとき、優先順序と呼ばれるシグネチャ上の擬順序 $\succeq$ に基づいて再帰的に比較を行う。例えば $f(s_1, \dots, s_m), g(t_1, \dots, t_n)$ に対して $f > g$ ならば $f(s_1, \dots, s_m)$ と全ての g の引数 t_j をそれぞれ比較し、 $f \sim g$ ならばその引数同士を多重集合と見なして比較を行う。なお $f > g$ は $f \succeq g$ かつ $g \not\succeq f$ であるときに成り立ち、また $f \sim g$ は $f \succeq g$ かつ $g \succeq f$ であるときに成立する。

3.1 定義

定義 3.1. $\succeq$ を優先順序とする。項上の関係 $>_{\text{mpo}}$ を次のように帰納的に定義する。

$$\begin{array}{c} \frac{s_i >_{\text{mpo}}^= t}{f(s_1, \dots, s_i, \dots, s_m) >_{\text{mpo}} t} \quad \mathbf{1} \\[10pt] \frac{f > g \quad \forall j. s >_{\text{mpo}} t_j}{s = f(s_1, \dots, s_m) >_{\text{mpo}} g(t_1, \dots, t_n)} \quad \mathbf{2} \\[10pt] \frac{f \sim g \quad \{s_1, \dots, s_m\} >_{\text{mpo}}^{\text{mul}} \{t_1, \dots, t_n\}}{f(s_1, \dots, s_m) >_{\text{mpo}} g(t_1, \dots, t_n)} \quad \mathbf{3} \end{array}$$

定理 3.2 ([7]). 優先順序 $\succeq$ の狭義半順序部分が整礎であるとき $>_{\text{mpo}}$ は簡約順序である。

例 3.3. 以下の項書換え系 $\mathcal{R}_1$ と優先順序 $\text{mul} > \text{add} > \text{s} > 0$ を考える。

$$\begin{array}{ll} \text{add}(x, 0) \rightarrow x & \text{mul}(x, 0) \rightarrow 0 \\ \text{add}(x, \text{s}(y)) \rightarrow \text{s}(\text{add}(x, y)) & \text{mul}(x, \text{s}(y)) \rightarrow \text{add}(\text{mul}(x, y), x) \end{array}$$

例えば規則 $\text{mul}(x, s(y)) \rightarrow \text{add}(\text{mul}(x, y), x)$ に対する順序関係は次のように導出される。

$$\begin{array}{c}
 \frac{y >_{\text{mpo}} y}{x = x \quad s(y) >_{\text{mpo}} y} \mathbf{1} \\
 \frac{\{x, s(y)\} >_{\text{mpo}}^{\text{mul}} \{x, y\}}{\text{mul} > \text{add} \quad \ell >_{\text{mpo}} \text{mul}(x, y)} \mathbf{3} \quad \frac{x >_{\text{mpo}} x}{\ell >_{\text{mpo}} x} \mathbf{1} \\
 \hline
 \ell = \text{mul}(x, s(y)) >_{\text{mpo}} \text{add}(\text{mul}(x, y), x) \mathbf{2}
 \end{array}$$

また $\mathcal{R}_1$ の各規則 $\ell \rightarrow r$ は $\ell >_{\text{mpo}} r$ を満たす。よって定理 2.19 から $\mathcal{R}_1$ は停止性を持つことが証明される。

3.2 原始帰納的関数との関係

MPO は原始帰納的関数で定義される関数全てを向きつけることができる。

定義 3.4. 原始帰納的関数は次のように帰納的に定義される自然数上の関数である。

1. 零関数 $\text{zero} = 0$ 、後者関数 $s(x) = x + 1$ 、射影関数 $p_i^n(x_1, \dots, x_i, \dots, x_n) = x_i$ は原始帰納的関数である。
2. $g, h_1, \dots, h_k$ が原始帰納的関数であるとき

$$f(\bar{x}) = g(h_1(\bar{x}), \dots, h_k(\bar{x}))$$

で定義される f は原始帰納的関数である。

3. g, h が原始帰納的関数であるとき

$$\begin{aligned}
 f(0, \bar{x}) &= g(\bar{x}) \\
 f(s(y), \bar{x}) &= h(f(y, \bar{x}), y, \bar{x})
 \end{aligned}$$

で定義される f は原始帰納的関数である。

全ての原始帰納的関数について、その原始帰納的関数を表現する項書換え系が存在し、かつその停止性は MPO を用いて証明できることが知られている [8, 19]。

例 3.5. 足し算 add は次のように定義される原始帰納的関数である。

$$\begin{aligned}
 \text{add}(0, x) &= p_1^1(x) \\
 \text{add}(s(y), x) &= h(\text{add}(y, x), y, x) \\
 h(x, y, z) &= s(p_3^3(x, y, z))
 \end{aligned}$$

add を表現する項書換え系 $\mathcal{R}$ は次のようになる。

$$\begin{array}{ll} p_1^1(x) \rightarrow x & \text{add}(0, x) \rightarrow p_1^1(x) \\ p_3^3(x, y, z) \rightarrow z & \text{add}(s(y), x) \rightarrow h(\text{add}(y, x), y, x) \\ & h(x, y, z) \rightarrow s(p_3^3(x, y, z)) \end{array}$$

このとき優先順序 $\text{add} > h > s > p_3^3 > p_1^1$ のもとで

$$\begin{array}{ll} p_1^1(x) >_{\text{mpo}} x & \text{add}(0, x) >_{\text{mpo}} p_1^1(x) \\ p_3^3(x, y, z) >_{\text{mpo}} z & \text{add}(s(y), x) >_{\text{mpo}} h(\text{add}(y, x), y, x) \\ & h(x, y, z) >_{\text{mpo}} s(p_3^3(x, y, z)) \end{array}$$

となり $\mathcal{R}$ は停止性を持つ。

3.3 部分項性と作用項との関係

多重集合経路順序は単純化順序の一つである。単純化順序が作用項に対して有効ではないことを示す。

例 3.6. $\mathcal{R}_1$ をカーリー化した作用項書き換え系 $\mathcal{R}_2$ を考える。

$$\begin{array}{ll} \text{add } x \ 0 \rightarrow x & \text{mul } x \ 0 \rightarrow 0 \\ \text{add } x \ (s \ y) \rightarrow s \ (\text{add } x \ y) & \text{mul } x \ (s \ y) \rightarrow \text{add} \ (\text{mul } x \ y) \ x \end{array}$$

このとき $\mathcal{R}_2$ の停止性は単純化順序 $>$ によって示すことができない。このことを $\mathcal{R}_2$ の最後の規則が $>$ によって向き付かないことを証明することによって示す。

命題 3.7. 任意の単純化順序 $>$ について $\text{mul } x \ (s \ y) \not> \text{add} \ (\text{mul } x \ y) \ x$ が成り立つ。

証明. 背理法を用いる。 $\ell = \text{mul } x \ (s \ y) > \text{add} \ (\text{mul } x \ y) \ x = r$ を仮定する。 $>$ は代入について閉じているため $\sigma = \{x \mapsto s \ y\}$ について

$$\ell\sigma = \text{mul} \ (s \ y) \ (s \ y) > \text{add} \ (\text{mul} \ (s \ y) \ y) \ (s \ y) = r\sigma$$

となる。 $\mathcal{E}_{\text{mb}} = \{x \ y \rightarrow x, x \ y \rightarrow y\}$ を用いて右辺を書換えていくと

$$\begin{aligned} \text{add} \ (\text{mul} \ (s \ y) \ y) \ (s \ y) &\rightarrow_{\mathcal{E}_{\text{mb}}} \text{mul} \ (s \ y) \ y \ (s \ y) \\ &\rightarrow_{\mathcal{E}_{\text{mb}}} \text{mul} \ (s \ y) \ (s \ y) \end{aligned}$$

となる。よって

$$\text{add} (\text{mul} (s\ y)\ y) (s\ y) \triangleright_{\text{emb}} \text{mul} (s\ y) (s\ y)$$

となる。 $>$ は部分項性を持つため、命題 2.22 から

$$\text{add} (\text{mul} (s\ y)\ y) (s\ y) > \text{mul} (s\ y) (s\ y)$$

を得る。 $>$ は推移性を持つことから

$$\text{mul} (s\ y) (s\ y) > \text{mul} (s\ y) (s\ y)$$

が成り立つ。しかしこれは $>$ が非反射性を持つことに矛盾する。 $\square$

単純化順序 $>$ が作用項に対して有効ではない理由は埋め込み規則によってカーリー化前の関数適用の形から関数記号を消去するような操作にも向き付けが許されてしまうことにある。先の証明中でも埋め込み規則のために $\text{add} (\text{mul} (s\ y)\ y) (s\ y)$ から左辺を得ることができてしまう。そして部分項性を持つ単純化順序は命題 2.22 に従い、この操作の全てについて向き付けなければならない。よってこの問題を解決するために、部分項性を持たない簡約順序を構成する必要がある。

第 4 章

作用型多重集合経路順序

本章では多重集合経路順序に基づく新たな作用項上の簡約順序を提案する。

まず 4.1 節で関係 $\triangleright_{\text{ampo}}$ を定義し 4.2 節にて $\triangleright_{\text{ampo}}$ が作用項上の整礎順序であることを示す。しかし $\triangleright_{\text{ampo}}$ は文脈と代入について閉じていない。そこで 4.3 節において $\triangleright_{\text{ampo}}$ を元に文脈について閉じた順序 $>_{\text{ampo}}$ を構成し、最後に 4.4 節にて $>_{\text{ampo}}$ を元に簡約順序である $\sqsubset_{\text{ampo}}$ を構成する。

4.1 定義

まずはじめに作用項に関する次の記法を定める。 Σ を作用型シグネチャとし、また $\circ$ を Σ の二項演算子とする。

定義 4.1. 作用項の列 $(t_1, \dots, t_n)$ を $\bar{t}$ と表し、 $\{t\} = \{t_1, \dots, t_n\}$ とする。また $|(t_1, \dots, t_n)| = n$ とする。 f が変数または定数のとき $f \bar{t}$ は $f t_1 \cdots t_n$ を表し、このとき f を**頭部** (head) と呼ぶ。

次に関係 $\triangleright_{\text{ampo}}$ を定義していく。 $\triangleright_{\text{ampo}}$ は Σ に関数適用記号の代わりとなる関数記号 $\bullet^{(0)}$ を加えた新たな作用型シグネチャ $\Sigma_\bullet$ 上の優先順序 $\succsim$ と関数の引数の数を表す項数割当て α をパラメータとして取る。

定義 4.2. Σ から誘導される新たな作用型シグネチャ $\Sigma_\bullet$ を $\Sigma_\bullet = \Sigma \uplus \{\bullet\}$ とする。写像 $\alpha : \Sigma_\bullet \setminus \{\bullet\} \rightarrow \mathbb{N}$ を**項数割当て** (arity assignment) と呼ぶ。また次のように 項数割当て α を変数に対して拡張する。

$$\alpha_{\mathcal{V}}(f) = \begin{cases} \alpha(f) & \text{if } f \in \Sigma_\bullet \\ 0 & \text{if } f \in \mathcal{V} \end{cases}$$

α を項数割当て $\succsim$ を $\Sigma_\bullet$ 上の優先順序とする。また $\bullet$ を $\succsim$ 上の極大元とし $\alpha(\bullet) = 2$ とする。作用項上の関係 $\triangleright_{\text{ampo}}$ を次のように帰納的に定める。

$$\begin{array}{c}
\frac{|\bar{s}| = \alpha(f) \quad s_i \triangleright_{\text{ampo}}^= t}{f \bar{s} \triangleright_{\text{ampo}} t} \quad \mathbf{1} \\
\\
\frac{|\bar{s}| = \alpha(f) \quad |\bar{t}| = \alpha(g) \quad f > g \quad \forall j. s \triangleright_{\text{ampo}} t_j}{s = f \bar{s} \triangleright_{\text{ampo}} g \bar{t}} \quad \mathbf{2} \\
\\
\frac{|\bar{s}| = \alpha(f) \quad |\bar{t}| = \alpha(g) \quad f \sim g \quad \{\bar{s}\} \triangleright_{\text{ampo}}^{\text{mul}} \{\bar{t}\}}{f \bar{s} \triangleright_{\text{ampo}} g \bar{t}} \quad \mathbf{3} \\
\\
\frac{|\bar{s} s'| \neq \alpha_V(f) \quad \bullet (f \bar{s}) s' \triangleright_{\text{ampo}} t}{f \bar{s} s' \triangleright_{\text{ampo}} t} \quad \mathbf{4} \\
\\
\frac{|\bar{s}| = \alpha(f) \quad g \neq \bullet \quad f \bar{s} \triangleright_{\text{ampo}} \bullet (g \bar{t}) t'}{f \bar{s} \triangleright_{\text{ampo}} g \bar{t} t'} \quad \mathbf{5}
\end{array}$$

1, 2, 3 は MPO の定義に従ったものである。**4, 5** は部分適用あるいは引数が過剰に適用された場合に向き付けを行えるよう導入された。これはつまり $\text{add } x \triangleright_{\text{ampo}} x$ や $\text{add } x \ 0 \ y \triangleright_{\text{ampo}} 0 \ y$ という向き付けのためである。また **5** は右辺に頭部変数項を許している。この規則によって map のような高階関数の向き付けが可能になる。

4 にある条件 $|\bar{s} s'| \neq \alpha(f)$ は作用項 $f \bar{s}$ とその任意の部分項同士の比較を不可能にするためのものである。これによって $\triangleright_{\text{ampo}}$ は部分項性を持たない。また **5** にある条件 $g \neq \bullet$ がない場合、次のような無限の導出を許してしまう。

$$\begin{array}{c}
\vdots \\
\frac{}{f \ x \ y \triangleright_{\text{ampo}} \bullet (\bullet (\bullet x)) \ y} \quad \mathbf{5} \\
\frac{}{f \ x \ y \triangleright_{\text{ampo}} \bullet (\bullet x) \ y} \quad \mathbf{5} \\
\frac{}{f \ x \ y \triangleright_{\text{ampo}} \bullet x \ y} \quad \mathbf{5} \\
\frac{}{f \ x \ y \triangleright_{\text{ampo}} x \ y} \quad \mathbf{5}
\end{array}$$

例 4.3. 作用型シグネチャ $\Sigma = \{\text{map}, \text{cons}, \text{nil}, \circ\}$ に対して、項数割当て α を次のように定める。

$$\alpha(\bullet) = 2 \qquad \alpha(\text{map}) = 2 \qquad \alpha(\text{cons}) = 2 \qquad \alpha(\text{nil}) = 0$$

$\text{map } f \text{ nil} \triangleright_{\text{ampo}} \text{nil}$ は次のように導出される。

$$\frac{2 = \alpha(\text{map}) \quad \text{nil} = \text{nil}}{\text{map } f \text{ nil} \triangleright_{\text{ampo}} \text{nil}} \quad \mathbf{1}$$

優先順序 $\bullet \sim \text{map} > \text{cons}$ のもとで $\text{map } f (\text{cons } x \text{ } xs) \triangleright_{\text{ampo}} f \text{ } x$ は次のように導出される。

$$\frac{2 = \alpha(\text{map}) \quad f \neq \bullet \quad \frac{\frac{2 = \alpha(\text{map}) \quad 2 = \alpha(\bullet) \quad \text{map} \sim \bullet \quad \{f, \text{cons } x \text{ } xs\} \triangleright_{\text{ampo}}^{\text{mul}} \{f, x\}}{\text{map } f (\text{cons } x \text{ } xs) \triangleright_{\text{ampo}} \bullet f \text{ } x}} \quad \mathbf{3}}{\text{map } f (\text{cons } x \text{ } xs) \triangleright_{\text{ampo}} f \text{ } x} \quad \mathbf{5}$$

また $\text{map } f (\text{cons } x \text{ } xs) \triangleright_{\text{ampo}} \text{map } f \text{ } xs$ は次のように導出される。

$$\frac{2 = \alpha(\text{map}) \quad 2 = \alpha(\text{map}) \quad \text{map} \sim \text{map} \quad \frac{\frac{2 = \alpha(\text{cons}) \quad xs = xs}{\text{cons } x \text{ } xs \triangleright_{\text{ampo}} xs} \quad \mathbf{1}}{\{f, \text{cons } x \text{ } xs\} \triangleright_{\text{ampo}}^{\text{mul}} \{f, xs\}} \quad \mathbf{3}}{\text{map } f (\text{cons } x \text{ } xs) \triangleright_{\text{ampo}} \text{map } f \text{ } xs}$$

そして $\text{map } f (\text{cons } x \text{ } xs) \triangleright_{\text{ampo}} \text{cons } (f \text{ } x) (\text{map } f \text{ } xs)$ は次のように導出される。

$$\frac{2 = \alpha(\text{map}) \quad 2 = \alpha(\text{cons}) \quad \text{map} > \text{cons} \quad s \triangleright_{\text{ampo}} f \text{ } x \quad s \triangleright_{\text{ampo}} \text{map } f \text{ } xs}{s = \text{map } f (\text{cons } x \text{ } xs) \triangleright_{\text{ampo}} \text{cons } (f \text{ } x) (\text{map } f \text{ } xs)} \quad \mathbf{2}$$

また $\text{map } x \text{ } y \triangleright_{\text{emb}} \text{map}$ が成り立つが $\text{map } x \text{ } y \not\triangleright_{\text{ampo}} \text{map}$ となる。このことから $\triangleright_{\text{ampo}}$ が部分項性を持たないことが確認できる。

4.2 整礎順序

簡約順序は整礎性と推移性、非反射性を持つ。本節では $\triangleright_{\text{ampo}}$ が推移性と整礎性を持つ関係であることを証明する。なお非反射性は整礎性から直ちに導かれる。

以降は作用型シグネチャ Σ の作用項を考える。まず準備として $\bullet$ に関する補題を証明する。

補題 4.4. 作用項 $t = f \text{ } t_1 \cdots t_n$ とする。 $n < \alpha_V(f)$ ならば、全ての $1 \leq i \leq n$ について $f \text{ } t_1 \cdots t_n \triangleright_{\text{ampo}} t_i$ が成り立つ。

証明. n についての帰納法を行う。まずはじめに $n = 1$ のとき次の導出を得る。

$$\frac{i \neq \alpha_V(f) \quad \frac{2 = \alpha(\bullet) \quad t_1 = t_1}{\bullet f t_1 \triangleright_{\text{ampo}} t_1} \textcolor{red}{1}}{f t_1 \triangleright_{\text{ampo}} t_1} \textcolor{red}{4}$$

$n = n' + 1$ のとき i についての場合分けを行う。 $1 \leq i \leq n'$ のとき帰納法の仮定より $f t_1 \cdots t_{n'} \triangleright_{\text{ampo}} t_i$ が成り立つ。よって次の導出を得る。

$$\frac{n \neq \alpha_V(f) \quad \frac{2 = \alpha(\bullet) \quad f t_1 \cdots t_{n'} \triangleright_{\text{ampo}} t_i}{\bullet (f t_1 \cdots t_{n'}) t_n \triangleright_{\text{ampo}} t_i} \textcolor{red}{1}}{f t_1 \cdots t_{n'} t_n \triangleright_{\text{ampo}} t_i} \textcolor{red}{4}$$

$i = n$ のとき次の導出を得る。

$$\frac{n \neq \alpha_V(f) \quad \frac{2 = \alpha(\bullet) \quad t_n = t_n}{\bullet (f t_1 \cdots t_{n'}) t_n \triangleright_{\text{ampo}} t_n} \textcolor{red}{1}}{f t_1 \cdots t_{n'} t_n \triangleright_{\text{ampo}} t_n} \textcolor{red}{4}$$

□

補題 4.5. 作用項 $t = f t_1 \cdots t_n$ について、 $f \in \Sigma$ かつ $n = \alpha(f)$ ならば $\bullet (f t_1 \cdots t_{n-1}) t_n \triangleright_{\text{ampo}} f t_1 \cdots t_n$ が **2** または **3** から成り立つ。

証明. f についての場合分けを行う。 $\bullet > f$ のとき次の導出を得る。

$$\frac{\bullet > f \quad \frac{\frac{\forall i < n. f t_1 \cdots t_{n-1} \triangleright_{\text{ampo}} t_i \quad \text{補題 4.4}}{\bullet (f t_1 \cdots t_{n-1}) t_n \triangleright_{\text{ampo}} t_i} \quad \frac{2 = \alpha(\bullet) \quad t_n = t_n}{\bullet (f t_1 \cdots t_{n-1}) t_n \triangleright_{\text{ampo}} t_n} \textcolor{red}{1}}{\bullet (f t_1 \cdots t_{n-1}) t_n \triangleright_{\text{ampo}} f t_1 \cdots t_n} \textcolor{red}{2}$$

$\bullet \sim f$ のとき補題 4.4 から $f t_1 \cdots t_{n-1} \triangleright_{\text{ampo}} t_i$ が全ての $1 \leq i < n$ について成り立つ。よって $\{f t_1 \cdots t_{n-1}, t_n\} \triangleright_{\text{ampo}}^{\text{mul}} \{t_1, \dots, t_n\}$ が得られる。従って次の導出を得る。

$$\frac{\bullet \sim f \quad \{f t_1 \cdots t_{n-1}, t_n\} \triangleright_{\text{ampo}}^{\text{mul}} \{t_1, \dots, t_n\}}{\bullet (f t_1 \cdots t_{n-1}) t_n \triangleright_{\text{ampo}} f t_1 \cdots t_n} \textcolor{red}{3}$$

□

次に $\triangleright_{\text{ampo}}$ の推移性を証明する。**1, 2, 3** はもととなる MPO の規則 **1, 2, 3** をカーリー化したものと見なせ、その推移性は MPO のものに従う。そのためまず $\triangleright_{\text{ampo}}$ が **1, 2, 3** による導出間で推移性を持つことを示し、ついで **4, 5** を含む場合の推移性を証明する。

補題 4.6. $s \triangleright_{\text{ampo}} t, t \triangleright_{\text{ampo}} u$ が共に **1, 2, 3** のいずれかから得られるとき $s \triangleright_{\text{ampo}} u$ が **1, 2, 3** のいずれかから導出される。

証明. $|s| + |t| + |u|$ に関する帰納法で示す。 $s \triangleright_{\text{ampo}} t$ および $t \triangleright_{\text{ampo}} u$ の導出について場合分けを行う。

$s \triangleright_{\text{ampo}} t$ が **1** から得られるとき $s = f \bar{s}$ となり $s_i \triangleright_{\text{ampo}}^= t \triangleright_{\text{ampo}} u$ となる $1 \leq i \leq |\bar{s}|$ が存在する。従って帰納法の仮定より $s_i \triangleright_{\text{ampo}}^= u$ となる。よって次の導出を得る。

$$\frac{|\bar{s}| = \alpha(f) \quad s_i \triangleright_{\text{ampo}}^= u}{s \triangleright_{\text{ampo}} u} \quad \mathbf{1}$$

$s \triangleright_{\text{ampo}} t$ が **2** から、 $t \triangleright_{\text{ampo}} u$ が **1** から得られるとき $s = f \bar{s}, t = g \bar{t}$ となり、 $s \triangleright_{\text{ampo}} t_j \triangleright_{\text{ampo}} u$ となる $1 \leq j \leq |\bar{t}|$ が存在する。よって帰納法の仮定より $s \triangleright_{\text{ampo}} u$ となる。

$s \triangleright_{\text{ampo}} t$ が **2** または **3** から、 $t \triangleright_{\text{ampo}} u$ が **2** から得られるとき、 $s = f \bar{s}, t = g \bar{t}, u = h \bar{u}$ かつ $f \gtrsim g > h$ となる。 $t \triangleright_{\text{ampo}} u_k$ が全ての $k \leq |\bar{u}|$ について成り立つ。従って帰納法の仮定より $s \triangleright_{\text{ampo}} u_k$ となる。よって次の導出を得る。

$$\frac{|\bar{s}| = \alpha(f) \quad |\bar{u}| = \alpha(h) \quad f > h \quad \forall k. s \triangleright_{\text{ampo}} u_k}{s \triangleright_{\text{ampo}} u} \quad \mathbf{2}$$

$s \triangleright_{\text{ampo}} t$ が **2** から、 $t \triangleright_{\text{ampo}} u$ が **3** から得られるとき $s = f \bar{s}, t = g \bar{t}, u = h \bar{u}$ となり $f > g \gtrsim h, \{\bar{t}\} \triangleright_{\text{ampo}}^{\text{mul}} \{\bar{u}\}$ が成立する。 $\{\bar{t}\} \triangleright_{\text{ampo}}^{\text{mul}} \{\bar{u}\}$ から、全ての $1 \leq k \leq |\bar{u}|$ について $t_j \triangleright_{\text{ampo}}^= u_k$ となる $1 \leq j \leq |\bar{t}|$ が存在する。また全ての j について $s \triangleright_{\text{ampo}} t_j$ となり、従って帰納法の仮定より $s \triangleright_{\text{ampo}} u_k$ を得る。よって次の導出を得る。

$$\frac{l = \alpha(f) \quad n = \alpha(h) \quad f > h \quad \forall k. s \triangleright_{\text{ampo}} u_k}{s \triangleright_{\text{ampo}} u} \quad \mathbf{2}$$

$s \triangleright_{\text{ampo}} t$ が **3** から、 $t \triangleright_{\text{ampo}} u$ が **1** から得られるとき $s = f \bar{s}, t = g \bar{t}, \{\bar{s}\} \triangleright_{\text{ampo}}^{\text{mul}} \{\bar{t}\}$ となり $t_j \triangleright_{\text{ampo}}^= u$ となる $1 \leq j \leq |\bar{t}|$ が存在する。 $\{\bar{s}\} \triangleright_{\text{ampo}}^{\text{mul}} \{\bar{t}\}$ から t_j について $s_i \triangleright_{\text{ampo}}^{\text{mul}} t_j$ となる $1 \leq i \leq |\bar{s}|$ が存在する。従って帰納法の仮定より $s_i \triangleright_{\text{ampo}} u$ が成り立ち次の導出を得る。

$$\frac{|\bar{s}| = \alpha(f) \quad s_i \triangleright_{\text{ampo}}^= u}{s \triangleright_{\text{ampo}} u} \quad \mathbf{1}$$

$s \triangleright_{\text{ampo}} t$ と $t \triangleright_{\text{ampo}} u$ がどちらも **3** から得られるとき $s = f \bar{s}, t = g \bar{t}, u = h \bar{u}, f \sim g \sim h, \{\bar{s}\} \triangleright_{\text{ampo}}^{\text{mul}} \{\bar{t}\} \triangleright_{\text{ampo}}^{\text{mul}} \{\bar{u}\}$ となる。推移性を持つ順序の多重集合拡張は推移性を持つ

つことと帰納法の仮定より $\{\bar{s}\} \triangleright_{\text{ampo}}^{\text{mul}} \{\bar{u}\}$ が成立する。よって次の導出を得る。

$$\frac{|\bar{s}| = \alpha(f) \quad |\bar{u}| = \alpha(h) \quad f \sim h \quad \{\bar{s}\} \triangleright_{\text{ampo}}^{\text{mul}} \{\bar{u}\}}{s \triangleright_{\text{ampo}} u} \quad \mathbf{3}$$

□

補題 4.7. $s \triangleright_{\text{ampo}} t$ が **5** から得られかつ $t \triangleright_{\text{ampo}} u$ のとき $s \triangleright_{\text{ampo}} u$ が成り立つ。

証明. $s \triangleright_{\text{ampo}} t$ が **5** から得られることから $s = f \bar{s}, t = g \bar{t} t', f \bar{s} \triangleright_{\text{ampo}} \bullet (g \bar{t}) t'$ が成り立ち、また **4, 5** の条件から $f \bar{s} \triangleright_{\text{ampo}} \bullet (g \bar{t}) t'$ は **1, 2, 3** のいずれかから導出される。 $t \triangleright_{\text{ampo}} u$ の導出について場合分けを行う。

$t \triangleright_{\text{ampo}} u$ が **1, 2, 3** のいずれかから得られるとき補題 4.5 から $\bullet (g \bar{t}) t' \triangleright_{\text{ampo}} g \bar{t} t' \triangleright_{\text{ampo}} u$ が **2** または **3** から得られる。よって補題 4.6 より $f \bar{s} \triangleright_{\text{ampo}} \bullet (g \bar{t}) t' \triangleright_{\text{ampo}} u$ となり、更に補題 4.6 より $f \bar{s} \triangleright_{\text{ampo}} u$ となる。

$t \triangleright_{\text{ampo}} u$ が **4** から得られるとき $f \bar{s} \triangleright_{\text{ampo}} \bullet (g \bar{t}) t' \triangleright_{\text{ampo}} u$ が成り立つ。 $\bullet (g \bar{t}) t' \triangleright_{\text{ampo}} u$ が **1, 2, 3** のいずれかから得られるとき、補題 4.6 より $s = f \bar{s} \triangleright_{\text{ampo}} u$ が成り立つ。 $\bullet (g \bar{t}) t' \triangleright_{\text{ampo}} u$ が **5** から得られるとき $u = h \bar{u} u'$ かつ $\bullet (g \bar{t}) t' \triangleright_{\text{ampo}} \bullet (h \bar{u}) u'$ が成り立つ。 $\bullet (g \bar{t}) t' \triangleright_{\text{ampo}} \bullet (h \bar{u}) u'$ は **1, 2, 3** のいずれからのみ導出されるので $f \bar{s} \triangleright_{\text{ampo}} \bullet (g \bar{t}) t' \triangleright_{\text{ampo}} \bullet (h \bar{u}) u'$ と補題 4.6 より $f \bar{s} \triangleright_{\text{ampo}} \bullet (h \bar{u}) u'$ が得られる。よって次の導出を得る。

$$\frac{|\bar{s}| = \alpha(f) \quad h \neq \bullet \quad f \bar{s} \triangleright_{\text{ampo}} \bullet (h \bar{u}) u'}{s = f \bar{s} \triangleright_{\text{ampo}} h \bar{u} u' = u} \quad \mathbf{5}$$

$t \triangleright_{\text{ampo}} u$ が **5** から得られるとき $|\bar{t} t'| = \alpha(g), u = h \bar{u} u'$ となり、 $g \bar{t} t' \triangleright_{\text{ampo}} \bullet (h \bar{u}) u'$ が **1, 2, 3** のいずれかから成り立つ。 $g \in \Sigma$ かつ $|\bar{t} t'| = \alpha(g)$ と補題 4.5 より $\bullet (g \bar{t}) t' \triangleright_{\text{ampo}} g \bar{t} t'$ が **2, 3** のいずれかから得られる。従って補題 4.6 と $f \bar{s} \triangleright_{\text{ampo}} \bullet (g \bar{t}) t' \triangleright_{\text{ampo}} g \bar{t} t'$ から $f \bar{s} \triangleright_{\text{ampo}} g \bar{t} t'$ が得られる。よって更に補題 4.6 より $f \bar{s} \triangleright_{\text{ampo}} \bullet (h \bar{u}) u'$ が得られる。従って **5** から $f \bar{s} \triangleright_{\text{ampo}} h \bar{u} u'$ を得る。 □

補題 4.8. $s \triangleright_{\text{ampo}} t$ であり、また $t \triangleright_{\text{ampo}} u$ が **5** から得られるとき $s \triangleright_{\text{ampo}} u$ が成り立つ。

証明. 仮定から $t = g \bar{t} \triangleright_{\text{ampo}} \bullet (h \bar{u}) u'$ が成立するが、左辺の項数割当ての制約と右辺の頭部が $\bullet$ であることから **4** ないし **5** による導出は不可能である。よって **1, 2, 3** のいずれかから導出される。 $s \triangleright_{\text{ampo}} t$ の導出について場合分けを行う。

$s \triangleright_{\text{ampo}} t$ が **1, 2, 3, 5** のいずれかから導出されるとき $s = f \bar{s}$ かつ $|\bar{s}| = \alpha(f)$ となる。補題 4.6 と補題 4.7 より $f \bar{s} \triangleright_{\text{ampo}} \bullet (h \bar{u}) u'$ が得られる。よって次の導出を得る。

$$\frac{|\bar{s}| = \alpha(f) \quad h \neq \bullet \quad f \bar{s} \triangleright_{\text{ampo}} \bullet (h \bar{u}) u'}{f \bar{s} \triangleright_{\text{ampo}} h \bar{u} u'} \quad \mathbf{5}$$

$s \triangleright_{\text{ampo}} t$ が **4** から得られるとき $s = f \bar{s} s', |\bar{s}| \neq \alpha_V(f)$ かつ $\bullet (f \bar{s}) s' \triangleright_{\text{ampo}} t$ となる $\bullet (f \bar{s}) s' \triangleright_{\text{ampo}} t$ は **1, 2, 3, 5** のいずれかからのみ得られるため、補題 4.6 と補題 4.7 より $\bullet (f \bar{s}) s' \triangleright_{\text{ampo}} \bullet (h \bar{u}) u'$ を得る。よって次の導出を得る。

$$\frac{|\bar{s}| \neq \alpha_V(f) \quad \frac{2 = \alpha(\bullet) \quad h \neq \bullet \quad \bullet (f \bar{s}) s' \triangleright_{\text{ampo}} \bullet (h \bar{u}) u'}{\bullet (f \bar{s}) s' \triangleright_{\text{ampo}} h \bar{u} u'} \quad \mathbf{5}}{f \bar{s} s' \triangleright_{\text{ampo}} h \bar{u} u'} \quad \mathbf{4}$$

□

補題 4.9. $s \triangleright_{\text{ampo}} t$ が **4** から得られ、また $t \triangleright_{\text{ampo}} u$ であるとき $s \triangleright_{\text{ampo}} u$ が成り立つ。

証明. 仮定より $s = f \bar{s} s'$ となる。また $\bullet (f \bar{s}) s' \triangleright_{\text{ampo}} t$ が成り立ち、これは **1, 2, 3, 5** のいずれかから導出される。 $\bullet (f \bar{s}) s' \triangleright_{\text{ampo}} t$ と $t \triangleright_{\text{ampo}} u$ の導出に関して場合分けを行う。

$\bullet (f \bar{s}) s' \triangleright_{\text{ampo}} t$ と $t \triangleright_{\text{ampo}} u$ が **1, 2, 3, 5** のいずれかから得られるとき補題 4.6、補題 4.7、補題 4.8 より $\bullet (f \bar{s}) s' \triangleright_{\text{ampo}} u$ が成り立つ。よって次の導出を得る。

$$\frac{|\bar{s} s'| \neq \alpha_V(f) \quad \bullet (f \bar{s}) s' \triangleright_{\text{ampo}} u}{f \bar{s} s' \triangleright_{\text{ampo}} u} \quad \mathbf{4}$$

$\bullet (f \bar{s}) s' \triangleright_{\text{ampo}} t$ と $t \triangleright_{\text{ampo}} u$ が **4** から得られるとき $t = g \bar{t} t'$ と $\bullet (f \bar{s}) s' \triangleright_{\text{ampo}} \bullet (g \bar{t}) t' \triangleright_{\text{ampo}} u$ が成り立つ。 $\bullet (f \bar{s}) s' \triangleright_{\text{ampo}} \bullet (g \bar{t}) t'$ は **1, 2, 3** のいずれかからのみ得られる。 $\bullet (g \bar{t}) t' \triangleright_{\text{ampo}} u$ は **1, 2, 3, 5** のいずれかからのみ得られる。よって補題 4.6 と補題 4.8 から $\bullet (f \bar{s}) s' \triangleright_{\text{ampo}} u$ が成り立つ。よって次の導出を得る。

$$\frac{|\bar{s} s'| \neq \alpha_V(f) \quad \bullet (f \bar{s}) s' \triangleright_{\text{ampo}} u}{f \bar{s} s' \triangleright_{\text{ampo}} u} \quad \mathbf{4}$$

□

補題 4.10. $s \triangleright_{\text{ampo}} t$ かつ $t \triangleright_{\text{ampo}} u$ が **4** から得られるとき $s \triangleright_{\text{ampo}} u$ が成り立つ。

証明. $|s|$ についての帰納法で示す。仮定より $t = g \bar{t} t', |\bar{t} t'| \neq \alpha(g)$ と $\bullet (g \bar{t}) t' \triangleright_{\text{ampo}} u$ が **1, 2, 3, 5** のいずれかから得られる。 $|\bar{t} t'| \neq \alpha(g)$ より $s \triangleright_{\text{ampo}} t$ の導出は **1, 4, 5** のいずれかによる。 $s \triangleright_{\text{ampo}} t$ の導出について場合分けを行う。

$s \triangleright_{\text{ampo}} t$ が **1** から得られるとき $s = f \bar{s}, |\bar{s}| = \alpha(f)$ となり $s_i \triangleright_{\text{ampo}}^= t$ となる $1 \leq i \leq |\bar{s}|$ が存在する。帰納法の仮定より $s_i \triangleright_{\text{ampo}} u$ となる。よって **1** より $s = f \bar{s} \triangleright_{\text{ampo}} u$ が得られる。

$s \triangleright_{\text{ampo}} t$ が **4** から得られるとき補題 4.9 から $s \triangleright_{\text{ampo}} u$ を得る。

$s \triangleright_{\text{ampo}} t$ が **5** から得られるとき $s = f \bar{s}, |\bar{s}| = \alpha(f)$ となり $f \bar{s} \triangleright_{\text{ampo}} g \bar{t} t'$ が **1, 2, 3** のいずれかから成り立つ。よって補題 4.6 と補題 4.8 から $s = f \bar{s} \triangleright_{\text{ampo}} u$ となる。 $\square$

そして補題 4.6、補題 4.7、補題 4.8、補題 4.9、補題 4.10 から次の補題が導かれる。

補題 4.11. $\triangleright_{\text{ampo}}$ は推移的である。

次に $\triangleright_{\text{ampo}}$ が整礎であることを示す。整礎性の証明手法として Buchholz [4] の手法を用いる。ここで作用項 t について t から始まる無限降下列 $t \triangleright_{\text{ampo}} t_1 \triangleright_{\text{ampo}} t_2 \triangleright_{\text{ampo}} \cdots$ が存在しないとき t は強正規化性 (strong normalization) を持つと言い、その全体集合を SN で表す。

補題 4.12. 作用項 $t = f \bar{t} t'$ について $\bullet (f \bar{t}) t' \in \text{SN}$ ならば $t \in \text{SN}$ となる。

証明. t が無限降下列 $t \triangleright_{\text{ampo}} t_1 \triangleright_{\text{ampo}} t_2 \triangleright_{\text{ampo}} \cdots$ を持つと仮定し、 $\bullet (f \bar{t}) t' \notin \text{SN}$ であることを示す。 f についての場合分けを行う。 $f \in \Sigma$ かつ $|\bar{t} t'| = \alpha(f)$ のとき、補題 4.5 から $\bullet (f \bar{t}) t' \triangleright_{\text{ampo}} t$ を得る。よって

$$\bullet (f \bar{t}) t' \triangleright_{\text{ampo}} t \triangleright_{\text{ampo}} t_1 \triangleright_{\text{ampo}} t_2 \triangleright_{\text{ampo}} \cdots$$

となり $\bullet (f \bar{t}) t' \notin \text{SN}$ が成り立つ。 $|\bar{t} t'| \neq \alpha_V(f)$ のとき、 $t \triangleright_{\text{ampo}} t_1$ は **4** からのみ導かれる。従って $\bullet (g \bar{t}) t' \triangleright_{\text{ampo}} t_1$ が成り立つ。よって

$$\bullet (f \bar{t}) t' \triangleright_{\text{ampo}} t_1 \triangleright_{\text{ampo}} t_2 \triangleright_{\text{ampo}} \cdots$$

となり、 $\bullet (f \bar{t}) t' \notin \text{SN}$ となる。 $\square$

補題 4.13. $s = f \bar{s} \triangleright_{\text{ampo}} t$ とする。任意の $1 \leq i \leq |\bar{s}|$ について $s_i \in \text{SN}$ ならば $t \in \text{SN}$ となる。

順序 $>_1, \dots, >_n$ の辞書式順序積 $>$ とは組 $A_1 \times \cdots \times A_n$ 上の順序で次のように定義されるものである。 $(a_1, \dots, a_n) > (b_1, \dots, b_n)$ であるとは $a_i >_i b_i$ かつ全ての $1 \leq j < i$

について $a_j = a_j$ となる i が存在することである。このとき $>_1, \dots, >_n$ が整礎であればその辞書式順序積は整礎となる [17]。

証明. 優先順序 $\succsim$ の狭義半順序部分 $>$ が整礎であると仮定する。よって $>, \triangleright_{\text{ampo}}^{\text{mul}}, \triangleright$ の辞書式順序積は整礎となるので組 $(f, \{\bar{s}\}, t)$ についての帰納法を用いて題意を示す。 $s \triangleright_{\text{ampo}} t$ が **1** から得られるならば、ある $1 \leq i \leq |\bar{s}|$ について $s_i \triangleright_{\text{ampo}} t$ となる。仮定より $s_i \in \text{SN}$ となり、よって $t \in \text{SN}$ が成立する。次に $s \triangleright_{\text{ampo}} t$ が **2** から得られるならば、 $t = g \bar{t}$ かつ $f > g$ かつ任意の $1 \leq j \leq |\bar{t}|$ について $s \triangleright_{\text{ampo}} t_j$ が成り立つ。第三成分から $(f, \{\bar{s}\}, t) > (f, \{\bar{s}\}, t_j)$ が成り立ち、帰納法の仮定より $t_j \in \text{SN}$ となる。ここで、 $t \triangleright_{\text{ampo}} u$ となる u を仮定する。第一成分から $(f, \{\bar{s}\}, t) > (g, \{\bar{t}\}, u)$ となり、帰納法の仮定より $u \in \text{SN}$ となる。よって $t \in \text{SN}$ が成立する。 $s \triangleright_{\text{ampo}} t$ が **3** から得られるならば、 $t = g \bar{t}$ かつ $f \sim g$ かつ $\{\bar{s}\} \triangleright_{\text{ampo}}^{\text{mul}} \{\bar{t}\}$ が成り立つ。 $\{\bar{s}\} \triangleright_{\text{ampo}}^{\text{mul}} \{\bar{t}\}$ より、任意の $1 \leq j \leq |\bar{t}|$ について、 $s_i \triangleright_{\text{ampo}} t_j$ となる i が存在する。よって仮定から $t_j \in \text{SN}$ となる。ここで $t \triangleright_{\text{ampo}} u$ となる u を仮定する。第二成分から $(f, \{\bar{s}\}, t) > (f, \{\bar{t}\}, u)$ となり、帰納法の仮定より $u \in \text{SN}$ となる。従って $t \in \text{SN}$ が成立する。 $s \triangleright_{\text{ampo}} t$ が **4** から得られるならば $s = f \bar{s} s'$ かつ $\bullet (f \bar{s}) s' \triangleright_{\text{ampo}} t$ が成り立つ。 $\bullet (f \bar{s}) s' \triangleright_{\text{ampo}} t$ が **1, 2, 3** のいずれかから得られるならば、先の場合の通り $t \in \text{SN}$ となる。 $\alpha(\bullet) = 2$ より $\bullet (f \bar{s}) s' \triangleright_{\text{ampo}} t$ が **4** から得られることはない。 $\bullet (f \bar{s}) s' \triangleright_{\text{ampo}} t$ が **5** から得られるとき $t = \bar{t} t'$ となり $\bullet (f \bar{s}) s' \triangleright_{\text{ampo}} \bullet \bar{t} t'$ が成り立つ。よって $\{f \bar{s}, s'\} \triangleright_{\text{ampo}}^{\text{mul}} \{\bar{t}, t'\}$ が得られる。したがって、次の4つの場合を考える。 $f \bar{s} \triangleright_{\text{ampo}} \bar{t}$ のとき、

- $f \bar{s} \triangleright_{\text{ampo}} \bar{t}$ のとき、第三成分から $(f, \{\bar{s} s'\}, t) > (f, \{\bar{s}\}, \bar{t})$ となる。よって帰納法の仮定より、 $\bar{t} \in \text{SN}$ となる。
- $f \bar{s} \triangleright_{\text{ampo}} t'$ のとき、第三成分から $(f, \{\bar{s} s'\}, t) > (f, \{\bar{s}\}, t')$ となる。よって帰納法の仮定より、 $t' \in \text{SN}$ となる。
- $s' \triangleright_{\text{ampo}} \bar{t}$ のとき、仮定から $s' \in \text{SN}$ となり、従って $\bar{t} \in \text{SN}$ となる。
- $s' \triangleright_{\text{ampo}} t'$ のとき、仮定から $t' \in \text{SN}$ となる。

よって $\bar{t}, t' \in \text{SN}$ となる。ここで $\bullet \bar{t} t' \triangleright_{\text{ampo}} u$ となる u を考える。第二成分より $(\bullet, \{f \bar{s}, s'\}, t) > (\bullet, \{\bar{t}, t'\}, u)$ となる。よって帰納法の仮定から $u \in \text{SN}$ を得る。従って $\bullet \bar{t} t' \in \text{SN}$ となり、補題 4.12 から $t \in \text{SN}$ となる。最後に $s \triangleright_{\text{ampo}} t$ が **5** から得られるならば、 $f \bar{s} \triangleright_{\text{ampo}} \bullet \bar{t} t'$ が成り立つ。 $|\bar{s}| = \alpha(f)$ と $\bullet \bar{t} t'$ より $f \bar{s} \triangleright_{\text{ampo}} \bullet \bar{t} t'$ は **1, 2** または **3** からのみ得られる。よって先の場合から $\bullet \bar{t} t' \in \text{SN}$ となる。補題 4.12 から $t \in \text{SN}$ となる。□

補題 4.14. 任意の作用項 s について、 $s \in \text{SN}$ が成り立つ。

証明. $s = f \bar{s}$ に関する構造帰納法で示す。 $s \triangleright_{\text{ampo}} t$ を仮定する。帰納法の仮定より、 $s_i \in \text{SN}$ が任意の i について成り立つ。従って補題 4.13 より、 $t \in \text{SN}$ を得る。よって $s \in \text{SN}$ が成立する。 $\square$

補題 4.15. $\succsim$ の狭義半順序部分が整礎ならば $\triangleright_{\text{ampo}}$ は整礎である。

そして補題 4.15 から次が導かれる。

補題 4.16. $\triangleright_{\text{ampo}}$ は非反射的である。

よって最後に次の補題を得る。

補題 4.17. $\triangleright_{\text{ampo}}$ は整礎順序である。

4.3 文脈について閉じた順序の構成

$\triangleright_{\text{ampo}}$ は作用項上の整礎順序だが、文脈について閉じていないため簡約順序ではない。

例 4.18. 作用型シグネチャ $\Sigma = \{f, a, b, o\}$ に対して項数割当て α を次のように定める。

$$\alpha(f) = 2 \qquad \alpha(a) = 0 \qquad \alpha(b) = 0$$

優先順序 $\bullet > f > a > b$ の下で $f a \triangleright_{\text{ampo}} b$ は次のように導出される。

$$\frac{\frac{1 \neq \alpha(f)}{\frac{2 = \alpha(\bullet)}{\frac{0 = \alpha(a) \quad 0 = \alpha(b) \quad a > b}{a \triangleright_{\text{ampo}} b} \text{ 2}} \bullet f a \triangleright_{\text{ampo}} b \text{ 1}}{f a \triangleright_{\text{ampo}} b} \text{ 4}}$$

ここで文脈 $C = \square x$ について $C[f a] = f a x \triangleright_{\text{ampo}} b x = C[b]$ は成立しない。 $f a x$ は与えられた項数と引数の数が一致しているため、**5** から導出されることになる。しかし $\bullet > f$ から $f a x \triangleright_{\text{ampo}} \bullet b x$ が成り立たず **5** による導出はできない。ただし優先順序が $\bullet \sim f$ を満たせば $f a \triangleright_{\text{ampo}} b$ と $f a x \triangleright_{\text{ampo}} b x$ の両方が成立する。

この問題は左辺の引数の数が f の項数に対して不足している場合に生じる。比較の際は左辺の引数の数が項数と等しい場合にのみ行うようにすれば、問題を回避できる。これをもとに文脈について閉じた順序 $>_{\text{ampo}}$ を定義する。

定義 4.19. s, t を作用項とし $s = f \bar{s}$ とする。 $s >_{\text{ampo}} t$ が成り立つとは $s \triangleright_{\text{ampo}} t$ かつ、 $|\bar{s}| < \alpha_V(f)$ ならば $s x >_{\text{ampo}} t x$ となることである。ここで x は s と t の両方に現れない新規変数である。

$>_{\text{ampo}}$ が文脈について閉じていることを示す。そのためにまず特定の場合においてのみ $>_{\text{ampo}}$ が代入について閉じていることを示す。

補題 4.20. s, t を作用項、 x を変数とする。代入 $\sigma = \{x \mapsto t\}$ について $s \triangleright_{\text{ampo}} x$ ならば $s\sigma \triangleright_{\text{ampo}} t$ が成り立つ。

証明. $|s|$ に関する帰納法で示す。 $s \triangleright_{\text{ampo}} x$ の導出について場合分けを行う。 $s \triangleright_{\text{ampo}} x$ が **1** から導出されるとき $s = f \bar{s}$ かつ $s_i = x$ となる $1 \leq i \leq |\bar{s}|$ が存在する。よって $s_i\sigma = t$ となり $s\sigma \triangleright_{\text{ampo}} t$ を **1** から得る。 $s \triangleright_{\text{ampo}} x$ が **4** から導出されるとき $s = f \bar{s} s', |\bar{s} s'| \neq \alpha(f)$ となり $\bullet (f \bar{s}) s' \triangleright_{\text{ampo}} x$ が **1** から成り立つ。 $s' \triangleright_{\text{ampo}} x$ のとき帰納法の仮定から $s'\sigma \triangleright_{\text{ampo}} t$ となる。 $f \bar{s} \triangleright_{\text{ampo}} x$ のとき同様に帰納法の仮定から $(f \bar{s})\sigma \triangleright_{\text{ampo}} t$ を得る。よって $(\bullet (f \bar{s}) s')\sigma \triangleright_{\text{ampo}} t$ となり、**4** から $(f \bar{s} s')\sigma \triangleright_{\text{ampo}} t$ が得られる。□

補題 4.21. s, t, u を作用項、 x を s, t 中に現れない新規変数とする。 $s x \triangleright_{\text{ampo}} t x$ ならば $s u \triangleright_{\text{ampo}} t u$ が成り立つ。

証明. 代入 $\sigma = \{x \mapsto u\}$ とする。 $s x \triangleright_{\text{ampo}} t x$ の導出について場合分けを行う。 $s x \triangleright_{\text{ampo}} t x$ は x が新規変数より **1** からは導出されない。次に $s x \triangleright_{\text{ampo}} t x$ が **2** から得られるとき $s = f \bar{s} x, t = g t_1 \cdots t_n x, f > g$ となり、全ての $1 \leq j \leq n$ について $s x \triangleright_{\text{ampo}} t_j$ となる。また $s x \triangleright_{\text{ampo}} x$ を得る。補題 4.20 より $(s x)\sigma \triangleright_{\text{ampo}} u$ を得る。 x が新規変数であり s 中に出現しないことから $s u \triangleright_{\text{ampo}} t_j$ となる。よって $s u \triangleright_{\text{ampo}} t u$ を **2** から得る。 $s x \triangleright_{\text{ampo}} t x$ が **3** から得られるとき $s x = f s_1 \cdots s_m, s_m = x, t x = g t_1 \cdots t_n, t_n = x, \{s_1, \dots, s_m\} \triangleright_{\text{ampo}}^{\text{mul}} \{t_1, \dots, t_n\}$ となる。 $\{s_1, \dots, s_m\} \triangleright_{\text{ampo}}^{\text{mul}} \{t_1, \dots, t_n\}$ から全ての $1 \leq j \leq n$ について $s_i \triangleright_{\text{ampo}} t_j$ となる $1 \leq i \leq m$ が存在する。 t_n について x が新規変数から $s_m = x = t_n$ となる。よって $\{s_1, \dots, s_{m-1}, u\} \triangleright_{\text{ampo}}^{\text{mul}} \{t_1, \dots, t_{n-1}, u\}$ となり **3** より $s u \triangleright_{\text{ampo}} t u$ を得る。 $s x \triangleright_{\text{ampo}} t x$ が **5** から得られるとき $s x \triangleright_{\text{ampo}} \bullet t x$ が成り立つ。 x が新規変数より $s x \triangleright_{\text{ampo}} \bullet t x$ は **2** または **3** からのみ得られる。よって先の場合から $s u \triangleright_{\text{ampo}} \bullet t u$ が成り立ち **5** から $s u \triangleright_{\text{ampo}} t u$ を得る。最後に $s x \triangleright_{\text{ampo}} t x$ が **4** から得られるとき $\bullet s x \triangleright_{\text{ampo}} t x$ が成り立つ **2, 3, 5** のいずれかからのみ得られる。よって先の場合と同

様に $\bullet s u \triangleright_{\text{ampo}} t u$ が成り立ち $s u \triangleright_{\text{ampo}} t u$ が 4 から成り立つ。 $\square$

補題 4.22. 任意の作用項 s, t, u について、 $s >_{\text{ampo}} t$ ならば $s u >_{\text{ampo}} t u$ が成り立つ。

証明. $s = f \bar{s}$ として f についての場合分けを行う。 $|\bar{s}| \geq \alpha_V(f)$ のとき $t = g \bar{t}$ となる。 $s >_{\text{ampo}} t$ から $s \triangleright_{\text{ampo}} t$ となる。よって次のように $s u \triangleright_{\text{ampo}} t u$ を得る。

$$\frac{\frac{|\bar{s}| + 1 \neq \alpha_V(f)}{\frac{2 = \alpha(\bullet) \quad g \neq \bullet}{\bullet s u \triangleright_{\text{ampo}} t u} \quad 4}}{s u \triangleright_{\text{ampo}} t u} \quad \frac{\frac{2 = \alpha(\bullet) \quad \bullet \sim \bullet \quad \frac{s \triangleright_{\text{ampo}} t}{\{s, u\} \triangleright_{\text{ampo}}^{\text{mul}} \{t, u\}} \quad 3}{\bullet s u \triangleright_{\text{ampo}} \bullet t u} \quad 5}{\bullet s u \triangleright_{\text{ampo}} t u} \quad 5$$

$|\bar{s}| + 1 \leq \alpha_V(f)$ のとき $N = \alpha_V(f) - |\bar{s}| + 1 \geq 0$ についての帰納法で示す。

- $N = 0$ のとき $s >_{\text{ampo}} t$ から $s \triangleright_{\text{ampo}} t$ かつ新規変数 x について $s x >_{\text{ampo}} t x$ を得る。 $s x >_{\text{ampo}} t x$ から $s x \triangleright_{\text{ampo}} t x$ を得る。補題 4.21 より $s u \triangleright_{\text{ampo}} t x$ を得る。 $|\bar{s}| + 1 = \alpha_V(f)$ から $s u >_{\text{ampo}} t u$ となる。
- $N \geq 1$ のとき $f \bar{s} x x' >_{\text{ampo}} g \bar{t} x x'$ を得る。ここで x' は新規変数である。帰納法の仮定より $f \bar{s} u x' >_{\text{ampo}} g \bar{t} u x'$ となる。また補題 4.21 より $f \bar{s} u \triangleright_{\text{ampo}} g \bar{t} u$ を得る。よって $f \bar{s} u >_{\text{ampo}} g \bar{t} u$ を得る。

$\square$

補題 4.23. 任意の作用項 s, t, u について、 $s >_{\text{ampo}} t$ ならば $u s >_{\text{ampo}} u t$ が成り立つ。

証明. $s >_{\text{ampo}} t$ から $s \triangleright_{\text{ampo}} t$ を得る。 u に関する場合分けを行う。 $u = h \bar{u}$ かつ $|\bar{u}| \geq \alpha_V(h)$ のとき次の導出を得る。

$$\frac{\frac{|\bar{u}| + 1 \neq \alpha_V(h)}{\frac{2 = \alpha(\bullet) \quad h \neq \bullet}{\bullet (h \bar{u}) s \triangleright_{\text{ampo}} h \bar{u} t} \quad 4}}{h \bar{u} s \triangleright_{\text{ampo}} h \bar{u} t} \quad \frac{\frac{2 = \alpha(\bullet) \quad \bullet \sim \bullet \quad \frac{s \triangleright_{\text{ampo}} t}{\{h \bar{u}, s\} \triangleright_{\text{ampo}}^{\text{mul}} \{h \bar{u}, t\}} \quad 3}{\bullet (h \bar{u}) s \triangleright_{\text{ampo}} \bullet (h \bar{u}) t} \quad 5}{\bullet (h \bar{u}) s \triangleright_{\text{ampo}} h \bar{u} t} \quad 5$$

$|\bar{u}| + 1 > \alpha_V(h)$ より $h \bar{u} s >_{\text{ampo}} h \bar{u} t$ を得る。同様にして $u = h \bar{u}$ かつ $h \in \text{Var}$ のときも $u s >_{\text{ampo}} u t$ が成り立つ。次に $u = h \bar{u}$ かつ $|\bar{u}| + 1 \leq \alpha_V(h)$ のとき $N = \alpha(f) - |\bar{u}| + 1$ についての帰納法を行う。

- $N = 0$ のとき次の導出を得る。

$$\frac{| \bar{u} | + 1 = \alpha(h) \quad h \sim h \quad \frac{s \triangleright_{\text{ampo}} t}{\{ \bar{u} s \} \triangleright_{\text{ampo}}^{\text{mul}} \{ \bar{u} t \}}}{h \bar{u} s \triangleright_{\text{ampo}} h \bar{u} t} \quad \mathbf{3}$$

$| \bar{u} | + 1 = \alpha(h)$ から $h \bar{u} s >_{\text{ampo}} h \bar{u} t$ を得る。

- $N \geq 1$ のとき次のように $h \bar{u} s \triangleright_{\text{ampo}} h \bar{u} t$ を得る。

$$\frac{\frac{2 = \alpha(\bullet) \quad h \neq \bullet \quad \frac{\frac{s \triangleright_{\text{ampo}} t}{\{ h \bar{u}, s \} \triangleright_{\text{ampo}}^{\text{mul}} \{ h \bar{u}, t \}}}{\bullet (h \bar{u}) s \triangleright_{\text{ampo}} \bullet (h \bar{u}) t}}{| \bar{u} | + 1 \neq \alpha_V(h) \quad \bullet (h \bar{u}) s \triangleright_{\text{ampo}} h \bar{u} t}}{h \bar{u} s \triangleright_{\text{ampo}} h \bar{u} t} \quad \mathbf{4}$$

$\mathbf{5}$

帰納法の仮定から $h \bar{u} s x >_{\text{ampo}} h \bar{u} t x$ が成り立つ。ここで x は新規変数である。よって $h \bar{u} s >_{\text{ampo}} h \bar{u} t$ となる。 $\square$

補題 4.24. 任意の作用項 s, t と文脈 C について、 $s >_{\text{ampo}} t$ ならば $C[s] >_{\text{ampo}} C[t]$ が成り立つ。

証明. C に関する構造帰納法で示す。 $C = \square$ のとき明らかに $C[s] = s >_{\text{ampo}} t = C[t]$ が成り立つ。その他のとき $C = C' u$ または $C = u C'$ となる二つの場合について考える。ここで C' は文脈、 u は作用項である。

- $C = C' u$ のとき帰納法の仮定から $C'[s] >_{\text{ampo}} C'[t]$ が成り立つ。補題 4.22 より $C'[s] u >_{\text{ampo}} C'[t] u$ を得る。よって $C'[s] u >_{\text{ampo}} C'[t] u$ となり $C[s] >_{\text{ampo}} C[t]$ が成立する。
- $C = u C'$ のとき帰納法の仮定と補題 4.23 から

$$C[s] = u C'[s] >_{\text{ampo}} u C'[t] = C[t]$$

を得る。よって $C[s] >_{\text{ampo}} C[t]$ が成立する。 $\square$

4.4 代入について閉じた順序の構成

前節では $>_{\text{ampo}}$ が文脈について閉じていることを示した。しかし $>_{\text{ampo}}$ は代入について閉じていないため簡約順序ではない。

例 4.25. 作用項シグネチャ $\Sigma = \{f, g, \circ\}$ に対して項数割当て α を次のように定める。

$$\alpha(f) = 1 \qquad \alpha(g) = 1$$

優先順序 $\bullet > f > g$ の下で $x y \triangleright_{\text{ampo}} f y$ は次のように導出される。

$$\frac{1 \neq \alpha_V(x) \quad \frac{\frac{2 = \alpha(\bullet) \quad 1 = \alpha(f) \quad \bullet > f \quad \frac{2 = \alpha(\bullet) \quad y = y}{\bullet x y \triangleright_{\text{ampo}} y} \text{ 1}}{\bullet x y \triangleright_{\text{ampo}} f y} \text{ 2}}{x y \triangleright_{\text{ampo}} f y} \text{ 4}$$

$1 > \alpha_V(x)$ より $x y >_{\text{ampo}} f y$ が成り立つ。ここで代入 $\sigma = \{x \mapsto g\}$ を考える。すると $(x a)\sigma = g y \triangleright_{\text{ampo}} f y$ は $\bullet > f > g$ から成立しない。

これは $x y$ のような変数が項の頭部に現れる場合に発生する問題である。よって基底項のみを考えればこの問題を回避できる。 $>_{\text{ampo}}$ に対して基底代入を導入した順序を定義する。

定義 4.26. 二つの作用項 s と t について関係 $s \sqsubset_{\text{ampo}} t$ が成り立つとは、任意の基底代入 σ について $s\sigma >_{\text{ampo}} t\sigma$ が成立することである。

関係 $\sqsubset_{\text{ampo}}$ が文脈について閉じた整礎順序であることは補題 4.17 と補題 4.24 から得られる。次に $\sqsubset_{\text{ampo}}$ が文脈について閉じていることを示す。

補題 4.27. 任意の作用項 s, t について $s \sqsubset_{\text{ampo}} t$ ならば、 $s\sigma \sqsubset_{\text{ampo}} t\sigma$ が任意の代入 σ について成り立つ。

証明. $s \sqsubset_{\text{ampo}} t$ を仮定する。 σ を代入、 τ を基底代入とすると、この二つの合成 $\sigma\tau$ もまた基底代入となる。 $s \sqsubset_{\text{ampo}} t$ より、 $s(\sigma\tau) >_{\text{ampo}} t(\sigma\tau)$ が成り立つ。従って $(s\sigma)\tau >_{\text{ampo}} (t\sigma)\tau$ が成り立ち、よって $s\sigma \sqsubset_{\text{ampo}} t\sigma$ が成立する。□

そして補題 4.17、補題 4.24、補題 4.27 から次の定理が得られる。

定理 4.28. 優先順序の狭義半順序部分が整礎かつ基底項が存在するとき、関係 $\sqsubset_{\text{ampo}}$ は簡約順序となる。

定理 4.28 において基底項の条件は必須である。基底項の存在しない次の作用型シグネチャ $\Sigma = \{\circ\}$ について $x \sqsubset_{\text{ampo}} x \sqsubset_{\text{ampo}} x \sqsubset_{\text{ampo}} x \sqsubset_{\text{ampo}} \dots$ となる。

4.5 $\triangleright_{\text{ampo}}$ と $\sqsubset_{\text{ampo}}$ の関係

$\sqsubset_{\text{ampo}}$ の定義は基底代入に関する全称量子子を含んでいる。そのため、作用項 s, t に対して $s \sqsubset_{\text{ampo}} t$ が成立するかどうかを自動的に判定することは困難になる。しかし $s \triangleright_{\text{ampo}} t$ が成り立つかどうかを自動的に判定することは可能である。本節では次に示す作用項について $\triangleright_{\text{ampo}}$ が $\sqsubset_{\text{ampo}}$ を代替できることを示す。

定義 4.29. 作用項 $f \bar{t}$ について $f \in \mathcal{V}$ かつ $|\bar{t}| \geq 1$ のとき $f \bar{t}$ を**頭部変数項**という。作用項 t の部分項中に頭部変数項が現れないとき t は**無頭部変数**であるという。また作用項書換え系 $\mathcal{R}$ が**左辺無頭部変数**であるとは、 $\mathcal{R}$ の全ての規則 $\ell \rightarrow r \in \mathcal{R}$ について左辺 ℓ が無頭部変数であることをいう。

補題 4.30. $s \triangleright_{\text{ampo}} t$ かつ s が無頭部変数ならば $s\tau \triangleright_{\text{ampo}} t\tau$ が任意の基底代入 τ について成り立つ。

証明. τ を基底代入とする。 $s\tau \triangleright_{\text{ampo}} t\tau$ が成立することを $s \triangleright_{\text{ampo}} t$ の導出に関する帰納法で示す。 $s \triangleright_{\text{ampo}} t$ が 次の導出

$$\frac{|\bar{s}| = \alpha(f) \quad s_i = t}{s = f \bar{s} \triangleright_{\text{ampo}} t} \quad \mathbf{1}$$

から得られるとき $s_i = t$ から $s_i\tau = t\tau$ となる。よって次の導出を得る。

$$\frac{|\bar{s}\tau| = \alpha(f) \quad s_i\tau = t\tau}{s\tau = f \bar{s}\tau \triangleright_{\text{ampo}} t\tau} \quad \mathbf{1}$$

$s \triangleright_{\text{ampo}} t$ が $\mathbf{1}$ から得られるとき $s = f \bar{s}, |\bar{s}| = \alpha(f)$ となり $s_i \triangleright_{\text{ampo}} t$ となる $i (1 \leq i \leq |\bar{s}|)$ が存在する。 s が無頭部変数項であることから s_i は無頭部変数項である。従って帰納法の仮定から $s_i\tau \triangleright_{\text{ampo}} t\tau$ を得る。よって次の導出を得る。

$$\frac{|\bar{s}\tau| = \alpha(f) \quad s_i\tau \triangleright_{\text{ampo}} t\tau}{f \bar{s}\tau \triangleright_{\text{ampo}} t\tau} \quad \mathbf{1}$$

$s \triangleright_{\text{ampo}} t$ が $\mathbf{2}$ から得られるとき $s = f \bar{s}, |\bar{s}| = \alpha(f), t = g \bar{t}, |\bar{t}| = \alpha(g), f > g$ となり $s \triangleright_{\text{ampo}} t_j$ が全ての $j (1 \leq j \leq |\bar{t}|)$ について成り立つ。 s は無頭部変数項であるから帰納法の仮定より $s\tau \triangleright_{\text{ampo}} t_j\tau$ となる。よって次の導出を得る。

$$\frac{|\bar{s}\tau| = \alpha(f) \quad |\bar{t}\tau| = \alpha(g) \quad f > g \quad \forall j. s\tau \triangleright_{\text{ampo}} t_j\tau}{f \bar{s}\tau \triangleright_{\text{ampo}} g \bar{t}\tau} \quad \mathbf{2}$$

$s \triangleright_{\text{ampo}} t$ が 3 から得られるとき $s = f \bar{s}, |\bar{s}| = \alpha(f), t = g \bar{t}, |\bar{t}| = \alpha(g), f \sim g$ となり $\{\bar{s}\} \triangleright_{\text{ampo}}^{\text{mul}} \{\bar{t}\}$ となる。帰納法の仮定から $\{\bar{s}\tau\} \triangleright_{\text{ampo}}^{\text{mul}} \{\bar{t}\tau\}$ を得る。よって次の導出を得る。

$$\frac{|\bar{s}\tau| = \alpha(f) \quad |\bar{t}\tau| = \alpha(g) \quad f > g \quad \{\bar{s}\tau\} \triangleright_{\text{ampo}}^{\text{mul}} \{\bar{t}\tau\}}{f \bar{s}\tau \triangleright_{\text{ampo}} g \bar{t}\tau} \quad 2$$

$s \triangleright_{\text{ampo}} t$ が 4 から得られるとき $s = f \bar{s} s', |\bar{s} s'| \neq \alpha_V(f)$ かつ $\bullet (f \bar{s}) s' \triangleright_{\text{ampo}} t$ となる。 s が無頭部変数より $\bullet (f \bar{s}) s'$ は無頭部変数項になる。従って帰納法の仮定より $\bullet ((f \bar{s})\tau) s'\tau \triangleright_{\text{ampo}} t\tau$ となる。よって次の導出を得る。

$$\frac{|\bar{s} s'\tau| \neq \alpha_V(f) \quad \bullet ((f \bar{s})\tau) s'\tau \triangleright_{\text{ampo}} t\tau}{(f \bar{s} s')\tau \triangleright_{\text{ampo}} t\tau} \quad 4$$

$s \triangleright_{\text{ampo}} t$ が 5 から得られるとき $s = f \bar{s}, |\bar{s}| = \alpha(f), t = g \bar{t} t', g \neq \bullet$ となり $f \bar{s} \triangleright_{\text{ampo}} \bullet (g \bar{t}) t'$ となる。 s が無頭部変数と帰納法の仮定より $(f \bar{s})\tau \triangleright_{\text{ampo}} \bullet ((g \bar{t})\tau) t'\tau$ を得る。よって次の導出を得る。

$$\frac{|\bar{s}\tau| = \alpha(f) \quad g \neq \bullet \quad (f \bar{s})\tau \triangleright_{\text{ampo}} \bullet ((g \bar{t})\tau) t'\tau}{(f \bar{s})\tau \triangleright_{\text{ampo}} (g \bar{t} t')\tau} \quad 5$$

□

定理 4.31. 任意の作用項 s, t について s が無頭部変数かつ $s >_{\text{ampo}} t$ が成り立つとき $s \sqsubset_{\text{ampo}} t$ が成り立つ。

証明. τ を基底代入とする。 $s = f \bar{s}$ として場合分けを行う。 $|\bar{s}| \geq \alpha_V(f)$ のとき $s >_{\text{ampo}} t$ から $s \triangleright_{\text{ampo}} t$ となる。 s が無頭部変数と補題 4.30 より $s\tau \triangleright_{\text{ampo}} t$ を得る。よって $s\tau >_{\text{ampo}} t\tau$ となる。 τ は任意の基底代入のため $s \sqsubset_{\text{ampo}} t$ を得る。 $|\bar{s}| < \alpha_V(f)$ のとき $N = \alpha_V(f) - |\bar{s}|$ として N についての帰納法を行う。

- $N = 1$ のとき $s \triangleright_{\text{ampo}} t$ かつ 新規変数 x に対して $s x >_{\text{ampo}} t x$ となる。 $s x >_{\text{ampo}} t x$ から $s \triangleright_{\text{ampo}} x t x$ を得る。 s が無頭部変数であるから $s x$ は無頭部変数項になる。従って補題 4.30 より $(s x)\tau \triangleright_{\text{ampo}} (t x)\tau$ となる。 $|\bar{s}| + 1 = \alpha_V(f)$ より $(s x)\tau >_{\text{ampo}} (t x)\tau$ が成り立つ。また $s \triangleright_{\text{ampo}} t$ と補題 4.30 から $s\tau \triangleright_{\text{ampo}} t\tau$ となる。よって $s\tau >_{\text{ampo}} t\tau$ から $s \sqsubset_{\text{ampo}} t$ を得る。
- $N > 1$ のとき帰納法の仮定から $s x \sqsubset_{\text{ampo}} t x$ を得る。よって $(s x)\tau >_{\text{ampo}} (t x)\tau$ となる。補題 4.30 より $s\tau \triangleright_{\text{ampo}} t\tau$ となる。従って $s\tau >_{\text{ampo}} t\tau$ を得る。よって $s \sqsubset_{\text{ampo}} t$ が成立する。

□

例 4.32. 作用型シグネチャ $\Sigma = \{\text{map}, \text{cons}, \text{nil}, \circ\}$ と次の項数割当てを考える。

$$\alpha(\text{map}) = 2 \qquad \alpha(\text{cons}) = 2 \qquad \alpha(\text{nil}) = 0$$

このとき Σ 上の作用項書換え系 $\mathcal{R}$ は

$$\begin{aligned} \text{map } f \text{ nil} &\rightarrow \text{nil} \\ \text{map } f (\text{cons } x \text{ } xs) &\rightarrow \text{cons } (f \ x) (\text{map } f \ xs) \end{aligned}$$

優先順序 $\bullet \sim \text{map} > \text{cons}$ のもとで

$$\begin{aligned} \text{map } f \text{ nil} &>_{\text{ampo}} \text{nil} \\ \text{map } f (\text{cons } x \text{ } xs) &>_{\text{ampo}} \text{cons } (f \ x) (\text{map } f \ xs) \end{aligned}$$

となる。また $\mathcal{R}$ は左辺無頭部変数であるから定理 4.31 より

$$\begin{aligned} \text{map } f \text{ nil} &\sqsubset_{\text{ampo}} \text{nil} \\ \text{map } f (\text{cons } x \text{ } xs) &\sqsubset_{\text{ampo}} \text{cons } (f \ x) (\text{map } f \ xs) \end{aligned}$$

となる。よって定理 2.19 から $\mathcal{R}$ の停止性が証明される。

第 5 章

自動化

本章では AMPO により停止性を証明することが可能な作用項書き換え系に対して、適切な優先順序の発見を自動化する手法について述べる。

5.1 漸増性

作用項 $s \triangleright_{\text{ampo}} t$ が優先順序 $\succsim$ の下で成り立っているとする。自動化の際は $\succsim \subseteq \succsim'$ となる別の優先順序 $\succsim'$ の下で $s \triangleright'_{\text{ampo}} t$ が成り立っていると仮定し、適切な優先順序 $\succsim'$ の発見を行うことになる。これは優先順序 $\succsim$ を制約としてエンコードするとき、整数上の全順序に埋め込むためである。そしてこの自動化の際の仮定が常に成り立つことを次の定理が保証する。

定理 5.1. 任意の優先順序 $\succsim, \succsim'$ について $\succsim \subseteq \succsim'$ ならば $\triangleright_{\text{ampo}} \subseteq \triangleright'_{\text{ampo}}$ が成り立つ。ここで $\triangleright_{\text{ampo}}$ は優先順序として $\succsim$ を用い $\triangleright'_{\text{ampo}}$ は $\succsim'$ を用いるものである。

証明. $\succsim \subseteq \succsim'$ と仮定する。 $s \triangleright_{\text{ampo}} t$ として $s \triangleright'_{\text{ampo}} t$ が成り立つことを、 $s \triangleright_{\text{ampo}} t$ の導出に関する帰納法で示す。 $s \triangleright_{\text{ampo}} t$ が **1** によるならば $s = f s_1 \cdots s_i \cdots s_m$ かつ $s_i \triangleright_{\text{ampo}}^= t$ となる。よって帰納法の仮定より $s_i \triangleright'_{\text{ampo}}^= t$ となる。従って次の導出から $s \triangleright'_{\text{ampo}} t$ を得る。

$$\frac{m = \alpha(f) \quad s_i \triangleright'_{\text{ampo}}^= t}{s = f s_1 \cdots s_i \cdots s_m \triangleright'_{\text{ampo}} t} \mathbf{1}$$

$s \triangleright_{\text{ampo}} t$ が **2** から得られるとき $s = f s_1 \cdots s_m, t = g t_1 \cdots t_n$ かつ $f > g$ となる。また全ての $1 \leq j \leq n$ について $s \triangleright_{\text{ampo}} t_j$ となる。仮定より $f >' g$ となる。帰納法の仮

定より $s \triangleright'_{\text{ampo}} t_j$ を得る。よって次の導出から $s \triangleright'_{\text{ampo}} t$ が成り立つ。

$$\frac{m = \alpha(f) \quad n = \alpha(g) \quad f >' g \quad \forall j. s \triangleright'_{\text{ampo}} t_j}{s = f s_1 \cdots s_m \triangleright'_{\text{ampo}} g t_1 \cdots t_n = t} \quad \mathbf{2}$$

$s \triangleright_{\text{ampo}} t$ が **3** から得られるとき $s = f \bar{s}, t = g \bar{t}, f \sim g$ かつ $\{\bar{s}\} \triangleright_{\text{ampo}}^{\text{mul}} \{\bar{t}\}$ となる。よって仮定より $f \sim' g$ を得る。また帰納法の仮定から $\{\bar{s}\} \triangleright_{\text{ampo}}^{\text{mul}} \{\bar{t}\}$ となる。従って次の導出から $s \triangleright'_{\text{ampo}} t$ を得る。

$$\frac{|\bar{s}| = \alpha(f) \quad |\bar{t}| = \alpha(g) \quad f \sim' g \quad \{\bar{s}\} \triangleright_{\text{ampo}}^{\text{mul}} \{\bar{t}\}}{s = f \bar{s} \triangleright'_{\text{ampo}} g \bar{t} = t} \quad \mathbf{3}$$

$s \triangleright_{\text{ampo}} t$ が **4** から得られるとき $s = f \bar{s} s'$ となり $\bullet (f \bar{s}) s' \triangleright_{\text{ampo}} t$ が成り立つ。よって帰納法の仮定から $\bullet (f \bar{s}) s' \triangleright'_{\text{ampo}} t$ を得る。従って次の導出から $s \triangleright'_{\text{ampo}} t$ を得る。

$$\frac{|\bar{s} s'| \neq \alpha_V(f) \quad \bullet (f \bar{s}) s' \triangleright_{\text{ampo}} t}{f \bar{s} s' \triangleright'_{\text{ampo}} t} \quad \mathbf{4}$$

$s \triangleright_{\text{ampo}} t$ が **5** から得られるとき $s = f \bar{s}, t = g \bar{t} t', g \neq \bullet f \bar{s} \triangleright_{\text{ampo}} \bullet (g \bar{t}) t'$ となる。よって帰納法の仮定から $f \bar{s} \triangleright'_{\text{ampo}} \bullet (g \bar{t}) t'$ を得る。従って次の導出から $s \triangleright'_{\text{ampo}} t$ を得る。

$$\frac{|\bar{s}| = \alpha(f) \quad g \neq \bullet \quad f \bar{s} \triangleright'_{\text{ampo}} \bullet (g \bar{t}) t'}{s = f \bar{s} \triangleright'_{\text{ampo}} g \bar{t} t' = t} \quad \mathbf{5}$$

□

5.2 制約式

制約解消系への入力となる制約式は、論理演算子と不等号から構成される。制約式を次のように定義する。

$$\begin{aligned} \langle \text{Expr} \rangle ::= & \top \mid \perp \mid p_f > p_g \mid p_f = p_g \mid p_f \geq p_g \mid \langle \text{Expr} \rangle \wedge \langle \text{Expr} \rangle \mid \\ & \langle \text{Expr} \rangle \vee \langle \text{Expr} \rangle \mid \langle \text{Expr} \rangle \Rightarrow \langle \text{Expr} \rangle \mid \neg \langle \text{Expr} \rangle \end{aligned}$$

ここで p_f は関数記号 f に対応する整数上の変数を表している。 $\top, \perp$ は真理値、 $=, >, \geq$ は整数の等号と不等号、 $\wedge, \vee, \Rightarrow, \neg$ はそれぞれ論理積、論理和、論理包含、論理否定を意味している。

次に $\lceil s \triangleright_{\text{ampo}} t \rceil$ に対応する制約式を定義する。なお多重集合についての制約式は [5] を参考にした。まず AMPO の各規則に対応する制約式を定義する。**1** に対応する制約式は $s = f \bar{s}$ として

$$\lceil s \triangleright_{\text{ampo}} t \rceil_{\mathbf{1}} = \begin{cases} \bigvee_{i=1}^{|\bar{s}|} \lceil s_i \triangleright_{\text{ampo}} t \rceil & \text{if } |\bar{s}| = \alpha(f) \\ \perp & \text{otherwise} \end{cases}$$

となる。**2** と **3** に対応する制約式は $s = f \bar{s}$ かつ $g = g \bar{t}$ として

$$\begin{aligned} \lceil s \triangleright_{\text{ampo}} t \rceil_{\mathbf{2}} &= \begin{cases} p_f > p_g \wedge \bigwedge_{j=1}^n \lceil s \triangleright_{\text{ampo}} t_j \rceil & \text{if } |\bar{s}| = \alpha(f) \text{ and } |\bar{t}| = \alpha(g) \\ \perp & \text{otherwise} \end{cases} \\ \lceil s \triangleright_{\text{ampo}} t \rceil_{\mathbf{3}} &= \begin{cases} p_f = p_g \wedge \lceil \{\bar{s}\} \triangleright_{\text{ampo}}^{\text{mul}} \{\bar{t}\} \rceil & \text{if } |\bar{s}| = \alpha(f) \text{ and } |\bar{t}| = \alpha(g) \\ \perp & \text{otherwise} \end{cases} \end{aligned}$$

となる。**4** に対応する制約式は $s = f \bar{s} s'$ として

$$\lceil s \triangleright_{\text{ampo}} t \rceil_{\mathbf{4}} = \begin{cases} \lceil \bullet (f \bar{s}) s' \triangleright_{\text{ampo}} t \rceil & \text{if } |\bar{s} s'| \neq \alpha_V(f) \\ \perp & \text{otherwise} \end{cases}$$

となる。そして **5** に対応する制約式は $s = f \bar{s}, t = g \bar{t} t'$ として

$$\lceil s \triangleright_{\text{ampo}} t \rceil_{\mathbf{5}} = \begin{cases} \lceil s \triangleright_{\text{ampo}} \bullet (g \bar{t}) t' \rceil & \text{if } |\bar{s}| = \alpha(f) \text{ and } g \neq \bullet \\ \perp & \text{otherwise} \end{cases}$$

となる。 $s \triangleright_{\text{ampo}}^{\text{mul}} t$ に対応する制約式は次のように定義される。

$$\lceil s \triangleright_{\text{ampo}}^{\text{mul}} t \rceil = \begin{cases} \top & \text{if } s = t \\ \lceil s \triangleright_{\text{ampo}} t \rceil & \text{if } s \neq t \end{cases}$$

$\lceil \{\bar{s}\} \triangleright_{\text{ampo}}^{\text{mul}} \{\bar{t}\} \rceil$ は次のように定義される。

$$\begin{aligned} \lceil \{\bar{s}\} \triangleright_{\text{ampo}}^{\text{mul}} \{\bar{t}\} \rceil &= \\ \lceil (\gamma, \varepsilon) \rceil \wedge \bigwedge_{\substack{1 \leq i \leq |\bar{s}|, \\ 1 \leq j \leq |\bar{t}|}} \left(\gamma_{i,j} \Rightarrow \left(\begin{array}{l} \varepsilon_i \Rightarrow \lceil s_i = t_j \rceil \\ \neg \varepsilon_i \Rightarrow \lceil s_i \triangleright_{\text{ampo}} t_j \rceil \end{array} \right) \right) \wedge \neg \bigwedge_{1 \leq i \leq m} \varepsilon_i \end{aligned}$$

ε_i と $\gamma_{i,j}$ は真理値の変数である。ここで $\lceil s = t \rceil$ は次のように定義される。

$$\lceil s = t \rceil = \begin{cases} \top & \text{if } s = t \\ \perp & \text{if } s \neq t \end{cases}$$

また $\lceil(\gamma, \varepsilon)\rceil$ は次のように定義される。なお総和を計算する際は $\gamma_{i,j}$ を $\top$ のときは 1、 $\perp$ のときは 0 と解釈する。

$$\lceil(\gamma, \varepsilon)\rceil = \bigwedge_{1 \leq j \leq n} \left(\sum_{i=1}^m \gamma_{i,j} = 1 \right) \wedge \bigwedge_{1 \leq i \leq m} \left(\varepsilon_i \Rightarrow \left(\sum_{j=1}^m \gamma_{i,j} = 1 \right) \right)$$

そして、 $\lceil s >_{\text{ampo}} t \rceil$ に対応する制約式を次のようになる。

$$\begin{aligned} \lceil s >_{\text{ampo}} t \rceil = & \lceil s >_{\text{ampo}} t \rceil_{\mathbf{1}} \vee \lceil s >_{\text{ampo}} t \rceil_{\mathbf{2}} \vee \lceil s >_{\text{ampo}} t \rceil_{\mathbf{3}} \\ & \vee \lceil s >_{\text{ampo}} t \rceil_{\mathbf{4}} \vee \lceil s >_{\text{ampo}} t \rceil_{\mathbf{5}} \end{aligned}$$

例 5.2. $\ell = \text{mul } x (s \ y) \triangleright_{\text{ampo}} \text{add } (\text{mul } x \ y) \ x = r$ に対する制約式を求める。**1** に対応する制約式は次のようになる。

$$\begin{aligned} & \lceil x >_{\text{ampo}} \text{add } (\text{mul } x \ y) \ x \rceil \vee \lceil s \ y >_{\text{ampo}} \text{add } (\text{mul } x \ y) \ x \rceil \\ = & \perp \vee (p_s > p_{\text{add}} \wedge \lceil s \ y >_{\text{ampo}} \text{mul } x \ y \rceil \wedge \lceil s \ y >_{\text{ampo}} x \rceil) \\ = & \perp \vee (p_s > p_{\text{add}} \wedge \lceil s \ y >_{\text{ampo}} \text{mul } x \ y \rceil \wedge \perp) \\ = & \perp \vee \perp \\ = & \perp \end{aligned}$$

2 に対応する制約式は次のようになる。

$$\begin{aligned} & p_{\text{mul}} > p_{\text{add}} \wedge \lceil \ell >_{\text{ampo}} \text{mul } x \ y \rceil \wedge \lceil \ell >_{\text{ampo}} x \rceil \\ = & p_{\text{mul}} > p_{\text{add}} \wedge \lceil \ell >_{\text{ampo}} \text{mul } x \ y \rceil \wedge \top \\ = & p_{\text{mul}} > p_{\text{add}} \wedge \lceil s \ y >_{\text{ampo}} y \rceil \wedge \top \\ = & p_{\text{mul}} > p_{\text{add}} \wedge \top \wedge \top \\ = & p_{\text{mul}} > p_{\text{add}} \end{aligned}$$

3, 4, 5 に対応する制約式は $\perp$ になる。最終的に制約式は以下の通りになる。

$$\perp \vee p_{\text{mul}} > p_{\text{add}} \vee \perp \vee \perp \vee \perp$$

$p_{\text{mul}} = 1$ と $p_{\text{add}} = 0$ とすれば制約式は成り立つ。よって優先順序 $\text{mul} > \text{add}$ を得る。

第 6 章

評価

本章では AMPO を評価する一つの指標として作用項書換え系の停止性の証明能力を取り上げる。まず停止性問題を用いる実験から AMPO が停止性を証明することができるあるいは証明が不可能な作用項書換え系を明らかにする。次に AMPO と MPO の関係から、AMPO が停止性を証明できる作用項書換え系のクラスについて議論する。

実験のために 5 章での手法と項数割当てを定める方法として [13] による作用型アリティを用いて、作用項書換え系の停止性を自動的に証明するツールを実装した。このツールは与えられた作用項書換え系 $\mathcal{R}$ が左辺無頭部変数かつ全ての規則 $\ell \rightarrow r \in \mathcal{R}$ について $\ell >_{\text{ampo}} r$ が成り立つとき、これを満たす優先順序を出力する。停止性証明ツールの実装にはプログラミング言語 Haskell を用い、また制約解消系は Z3 4.12.2 [6] を使用した。

6.1 例

停止性問題とは入力として与えられた項書換え系の停止性を証明あるいは反証する問題である。以下の例は掛け算、冪乗を表す作用項書換え系と Haskell, ML といった関数型言語で利用されている標準的な関数の定義を表現する作用項書換え系である。それぞれの例において、実装した AMPO による停止性証明ツールがその停止性の証明に成功するものは優先順序を合わせて示す。

例 6.1 (掛け算). 掛け算は再帰に用いる引数の位置の違いから 2 つの例を示す。

$$\begin{array}{ll} \text{add } x \ 0 \rightarrow x & \text{mul } x \ 0 \rightarrow 0 \\ \text{add } x \ (\text{s } y) \rightarrow \text{s } (\text{add } x \ y) & \text{mul } x \ (\text{s } y) \rightarrow \text{add } x \ (\text{mul } x \ y) \end{array}$$

$$\begin{array}{ll}
\text{add } 0 \ y \rightarrow y & \text{mul } 0 \ y \rightarrow 0 \\
\text{add } (s \ x) \ y \rightarrow s \ (\text{add } x \ y) & \text{mul } (s \ x) \ y \rightarrow \text{add } (\text{mul } x \ y) \ y
\end{array}$$

どちらも優先順序 $\bullet > \text{mul} > \text{add} > s$ を用いて停止性を証明できる。

例 6.2 (冪乗).

$$\begin{array}{ll}
\text{add } x \ 0 \rightarrow x & \text{pow } x \ 0 \rightarrow s \ 0 \\
\text{add } x \ (s \ y) \rightarrow s \ (\text{add } x \ y) & \text{pow } x \ (s \ y) \rightarrow \text{mul } x \ (\text{pow } x \ y) \\
\text{mul } x \ 0 \rightarrow 0 & \\
\text{mul } x \ (s \ y) \rightarrow \text{add } x \ (\text{mul } x \ y) &
\end{array}$$

優先順序 $\bullet \sim \text{pow} > \text{mul} > \text{add} > s$ を用いて停止性を証明できる。

例 6.3 (reverse). 関数 reverse はリストの反転を行う関数である。次の作用項書換え系は reverse の定義を表したものである。

$$\begin{array}{l}
\text{append nil } ys \rightarrow ys \\
\text{append } (\text{cons } x \ xs) \ ys \rightarrow \text{cons } x \ (\text{append } xs \ ys) \\
\text{reverse nil} \rightarrow \text{nil} \\
\text{reverse } (\text{cons } x \ xs) \rightarrow \text{append } (\text{reverse } xs) \ (\text{cons } x \ \text{nil})
\end{array}$$

優先順序 $\bullet > \text{rev} > \text{append} > \text{cons}$ を用いて停止性を証明できる。

例 6.4 (map). 高階関数 map は引数として関数とリストを受け取り、その関数をリストのの要素に適用していくものである。次の作用項書換え系は map の定義を表したものである。

$$\begin{array}{l}
\text{map } f \ \text{nil} \rightarrow \text{nil} \\
\text{map } f \ (\text{cons } x \ xs) \rightarrow \text{cons } (f \ x) \ (\text{map } f \ xs)
\end{array}$$

優先順序 $\bullet \sim \text{map} > \text{cons}$ を用いて停止性を証明できる。

例 6.5 (filter). 次の作用項書換え系はリストから述語を満たす要素以外のものを除く高階関数 filter の定義を表したものである。

$$\begin{array}{ll}
\text{if true } x \ y \rightarrow x & \text{filter } p \ \text{nil} \rightarrow \text{nil} \\
\text{if false } x \ y \rightarrow y & \text{filter } p \ (\text{cons } x \ xs) \rightarrow \text{if } (p \ x) \ (\text{cons } x \ (\text{filter } p \ xs)) \ (\text{filter } p \ xs)
\end{array}$$

優先順序 $\bullet \sim \text{filter} > \text{if} \sim \text{cons}$ を用いて停止性を証明できる。

例 6.6 (concatMap). 次の作用項書換え系は map を適用した後にリストの要素を全て結合する高階関数である concatMap の定義を表したものである。

$$\begin{aligned}\text{concatMap } f \text{ nil} &\rightarrow \text{nil} \\ \text{concatMap } f (\text{cons } x \text{ xs}) &\rightarrow \text{append } (f \ x) (\text{concatMap } f \text{ xs})\end{aligned}$$

優先順序 • ~ concatMap > append を用いて停止性を証明できる。

例 6.7 (find). 次の作用項書換え系はリスト内の述語を満たす要素を検索する高階関数 find の定義を表したものである。

$$\begin{aligned}\text{find } p \text{ nil} &\rightarrow \text{Nothing} \\ \text{find } p (\text{cons } x \text{ xs}) &\rightarrow \text{if } (p \ x) (\text{Just } x) (\text{find } p \text{ xs})\end{aligned}$$

優先順序 • ~ find ~ Just > if ~ Nothing を用いて証明できる。

例 6.8 (all, any). 次の作用項書換え系はリストの要素が全て述語を満たすか判定する高階関数 all の定義を表したものである。

$$\begin{aligned}\&\& \text{ true true} &\rightarrow \text{true} & \text{all } p \text{ nil} &\rightarrow \text{true} \\ \&\& \text{ false } y &\rightarrow \text{false} & \text{all } p (\text{cons } x \text{ xs}) &\rightarrow \&\& (p \ x) (\text{all } p \text{ xs}) \\ \&\& \text{ } x \text{ false} &\rightarrow \text{false}\end{aligned}$$

優先順序 • ~ all > && ~ true ~ false を用いて停止性を証明できる。また次の作用項書換え系はリスト中に述語を満たす要素が存在するかを判定する高階関数 any の定義を表したものである。

$$\begin{aligned}\parallel \text{ false false} &\rightarrow \text{false} & \text{any } p \text{ nil} &\rightarrow \text{false} \\ \parallel \text{ true } y &\rightarrow \text{true} & \text{any } p (\text{cons } x \text{ xs}) &\rightarrow \parallel (p \ x) (\text{any } p \text{ xs}) \\ \parallel \text{ } x \text{ true} &\rightarrow \text{true}\end{aligned}$$

優先順序 • ~ all > || ~ true ~ false を用いて停止性を証明できる。

例 6.9 (reverseMap). 高階関数 reverseMap はリストの反転を同時に行う map である。次の作用項書換え系はその定義を表したものである。

$$\begin{aligned}\text{reverseMap}' f \ e \text{ nil} &\rightarrow e \\ \text{reverseMap}' f \ e (\text{cons } x \text{ xs}) &\rightarrow \text{reverseMap}' f (\text{cons } (f \ x) \ e) \text{ xs} \\ \text{reverseMap } f \text{ xs} &\rightarrow \text{reverseMap}' f \text{ nil xs}\end{aligned}$$

AMPO による停止性の証明はできない。

例 6.10 (zipWith). 高階関数 zipWith は例えば

$$\text{zipWith } (+) [1, 2, 3] [4, 5, 6] = [5, 7, 9]$$

となるような高階関数である。次の作用項書換え系は zipWith の定義を表したものである。

$$\begin{aligned}\text{zipWith } f \text{ nil } ys &\rightarrow \text{nil} \\ \text{zipWith } f \text{ xs nil} &\rightarrow \text{nil} \\ \text{zipWith } f (\text{cons } x \text{ xs}) (\text{cons } y \text{ ys}) &\rightarrow \text{cons } (f \ x \ y) (\text{zipWith } f \text{ xs } ys)\end{aligned}$$

AMPO による停止性の証明はできない。

例 6.11 (foldl, foldr). foldl と foldr は例えば

$$\begin{aligned}\text{foldl } + \ 0 \ [1, 2, 3] &= ((0 + 1) + 2) + 3 \\ \text{foldr } + \ 0 \ [1, 2, 3] &= 1 + (2 + (3 + 0))\end{aligned}$$

となるような高階関数である。次の作用項書換え系は foldl の定義を表したものである。

$$\begin{aligned}\text{foldl } f \ e \ \text{nil} &\rightarrow e \\ \text{foldl } f \ e \ (\text{cons } x \text{ xs}) &\rightarrow \text{foldl } f \ (f \ e \ x) \text{ xs}\end{aligned}$$

次の作用項書換え系は foldr の定義を表したものである。

$$\begin{aligned}\text{foldr } f \ e \ \text{nil} &\rightarrow e \\ \text{foldr } f \ e \ (\text{cons } x \text{ xs}) &\rightarrow f \ x \ (\text{foldr } f \ e \ xs)\end{aligned}$$

どちらも AMPO による停止性の証明はできない。

例 6.12 (recursor). recursor は引数として受け取った関数を指定された回数分だけ再帰的に適用する高階関数である。次の作用項書換え系は recursor の定義を表したものである。

$$\begin{aligned}\text{rec } f \ x \ 0 &\rightarrow x \\ \text{rec } f \ x \ (\text{s } y) &\rightarrow f \ (\text{s } y) (\text{rec } f \ x \ y)\end{aligned}$$

AMPO による停止性の証明はできない。

例 6.13 (ack). 次の作用項書換え系はアッカーマン関数の定義を表したものである。

$$\begin{aligned}\text{ack } 0 \ y &\rightarrow \text{s } y \\ \text{ack } (\text{s } x) \ 0 &\rightarrow \text{ack } x \ (\text{s } 0) \\ \text{ack } (\text{s } x) \ (\text{s } y) &\rightarrow \text{ack } x \ (\text{ack } (\text{s } x) \ y)\end{aligned}$$

AMPO による停止性の証明はできない。

6.2 実験

AMPO が停止性を証明することができる作用項書換え系のクラスを評価するために、停止性問題集を用いて実験を行う。Termination Problem Database (TPDB) [23] は国際的な停止性証明ツールの競技会である Termination Competition において使用されている問題集である。実験では TPDB Version 11.3 のうち一階の項書換え系の問題集である TRS_Standard から、作用項書換え系であるものを抽出し利用した。このとき作用項書換え系の定義に従い、定数記号とただ一つの 2 引数関数記号から構成されるものを作用項書換え系とした。抽出した作用項書換え系の問題数は 201 問で、その内左辺無頭部変数である問題の数は 149 問である。実験は TPDB からの問題 (201 問) と先に上げた例題 (16 問) を合わせた 217 問を用いて行った。実験に使用したコンピュータは Intel Core i7-1065G7 1.30GHz プロセッサと 16GB のメモリを搭載したものである。表 6.2 に実験の結果を示す。AMPO, APOL, EPO, MPO は それぞれ作用型多重集合経路順序、作用型解釈順序 [26]、埋め込み経路順序 [2]、多重集合経路順序の実装である。また比較のために Termination Competition 2024 [22] に参加した停止性証明ツールのうち入手可能であった NaTT [24], AProVE [11], NTI [18] を用いた。ツールの実行時間は 1 問につき 60 秒とした。以下の表において、タイムアウトは 60 秒以内にツールが停止性の証明と反証のどちらもできなかった問題数、証明成功・反証成功はそれぞれ停止性の証明・反証に成功した問題数、証明失敗は 60 秒以内ではあるがツールが停止性の証明と反証のどちらにも失敗した問題数である。また各項目下の合計時間はツールの処理時間の合計である。なお実験結果と実験に使用したツールは以下の Web ページにある。

<https://www.jaist.ac.jp/~hirokawa/software/yachi/>

6.3 実験の評価

先に上げた例題について個別にその証明の成否をまとめたものを表 6.3 に示す。表 6.3 中では停止性の証明に成功したものに ✓ を、失敗したものに ✗ を、タイムアウトだったものに ? を付している。

表 6.1 停止性証明問題を用いた実験の結果

	AMPO	APOL	EPO	MPO	NaTT	AProVE	NTI
証明成功	20	45	6	3	159	189	17
合計時間 (秒)	<i>0.56</i>	<i>24.30</i>	<i>0.11</i>	<i>0.05</i>	<i>79.44</i>	<i>256.91</i>	<i>25.97</i>
反証成功	0	0	0	0	9	14	18
合計時間 (秒)	<i>0</i>	<i>0</i>	<i>0</i>	<i>0</i>	<i>6.53</i>	<i>18.23</i>	<i>2.97</i>
証明失敗	191	162	211	214	48	0	43
合計時間 (秒)	<i>6.74</i>	<i>178.32</i>	<i>3.76</i>	<i>26.57</i>	<i>57.57</i>	<i>0</i>	<i>8.15</i>
タイムアウト	1	8	0	0	0	23	139

表 6.2 例題についての結果

	AMPO	APOL	EPO	MPO	NaTT	AProVE	NTI
mul1	✓	×	×	×	✓	✓	✓
mul2	✓	×	×	×	✓	✓	?
pow	✓	×	×	×	✓	✓	?
reverse	✓	✓	×	×	✓	✓	?
map	✓	✓	×	×	✓	✓	?
filter	✓	×	×	×	✓	✓	?
concatMap	✓	✓	×	×	✓	✓	?
find	✓	✓	×	×	✓	✓	?
all	✓	✓	×	×	✓	✓	?
any	✓	✓	×	×	✓	✓	?
reverseMap	×	✓	×	×	✓	✓	?
zipWith	×	✓	×	×	✓	✓	?
foldl	×	×	×	×	×	?	?
foldr	×	×	×	×	×	?	?
recursor	×	×	×	×	×	?	?
ack	×	×	×	×	✓	✓	✓

表 6.2 において AMPO が証明に成功した問題よりも多くの問題に APOL が証明成功している。これは AMPO が $\text{add } x (\text{add } y z) \rightarrow \text{add } (\text{add } x y) z$ のような演算子の結合

律を表すような規則について向き付けを行えないが、作用型解釈順序では可能であるためである。また表 6.3 において AMPO は zipWith の証明に失敗しているが APOL は成功している。これは zipWith 中の規則

$$\text{zipWith } f (\text{cons } x \text{ } xs) (\text{cons } y \text{ } ys) \rightarrow \text{cons } (f \ x \ y) (\text{zipWith } f \ xs \ ys)$$

の向き付けに AMPO は失敗するためである。これは $\bullet$ が極大元であるという制約に起因している。先の例において向き付けが失敗する要因となるのは次の比較である。

$$\text{zipWith } f (\text{cons } x \text{ } xs) (\text{cons } y \text{ } ys) \triangleright_{\text{ampo}} f \ x \ y$$

ここで簡略化したものを考える。

$$\text{zipWith } f (\text{c } x) (\text{c } y) \triangleright_{\text{ampo}} f \ x \ y$$

右辺からこの比較は

$$\text{zipWith } f (\text{c } x) (\text{c } y) \triangleright_{\text{ampo}} \bullet (f \ x) \ y$$

に取って代わる。 $\bullet$ は優先順序において極大元であるから $\text{zipWith} \sim \bullet$ となる。従って **3** において引数同士の比較

$$\{f, \text{c } x, \text{c } y\} \triangleright_{\text{ampo}}^{\text{mul}} \{f \ x, y\}$$

が行われる。しかし $f \ x$ は左辺中のどの項とも望む方向には向き付かない。よってこの比較は失敗する。この問題は **5** においても同様に発生する。一方で、作用型解釈順序においては再帰的な部分項の比較が生じないため、向き付けに成功する。

AMPO が停止性の証明に失敗するが EPO は証明に成功する問題として、次のような単一の変換規則からなるものがる。

$$\mid x \ y \rightarrow x \ y$$

これは関数型言語においては関数適用演算子として知られているものである。 $\alpha(l) = 2$ とすると $\mid x \ y \triangleright_{\text{ampo}}^{\text{mul}} \bullet \ x \ y$ の向き付けは失敗する。 $\mid \sim \bullet$ から **3** において引数同士の比較を行うが、これは $\{x, y\} \not\triangleright_{\text{ampo}} \{x, y\}$ となって失敗する。一方で EPO は部分項性を持つため $\mid x \ y \triangleright_{\text{emb}} x \ y$ から向き付けが必ず成功する。ただし AMPO においても $\alpha(l) = 1$

とすると次の導出から $\vdash x\ y \triangleright_{\text{ampo}} x\ y$ が得られる。

$$\begin{array}{c}
\frac{1 = \alpha(\text{!}) \quad x = x}{\vdash x \triangleright_{\text{ampo}} x} \text{1} \\
\frac{2 = \alpha(\bullet) \quad \bullet \sim \bullet \quad \frac{\vdash x, y \triangleright_{\text{ampo}}^{\text{mul}} \{x, y\}}{\vdash x \triangleright_{\text{ampo}} x} \text{3}}{\vdash x \triangleright_{\text{ampo}} x} \text{2} \\
\frac{2 \neq \alpha_V(\text{!}) \quad \frac{2 = \alpha(\bullet) \quad \bullet (\vdash x) \ y \triangleright_{\text{ampo}} \bullet \ x \ y}{\vdash x \triangleright_{\text{ampo}} x \ y} \text{5}}{\vdash x \ y \triangleright_{\text{ampo}} x \ y} \text{4}
\end{array}$$

そして $\vdash x\ y$ は無頭部変数のため $\vdash x\ y \sqsubset_{\text{ampo}} x\ y$ が成り立つ。

APOL, EPO が共に停止性の証明に失敗し、AMPO が成功している問題として filter がある。map と filter の向き付けが可能であることは、プログラム検証においてはリスト内包表記を扱うことが可能であること、数学定理の証明においては集合の内包表記が扱えることを意味している。

表 6.3 において NaTT の結果は AMPO と APOL の結果を包含しているように見える。NaTT [24] では作用項書換え系に対する停止性証明を uncurrying 変換の後に重み付き経路順序 (WPO) [25] という簡約順序を用いて行っている。WPO は解釈順序と MPO を含む再帰経路順序の一般化である。そのため AMPO と APOL によって停止性が証明される作用項書換えのクラスの合併は NaTT が停止性を証明可能なクラスに包含されると予想される。

NaTT 以外の停止性ツールについても見ていく。NTI は依存対手法 [12] を用いて停止性の証明を行っている。しかし NTI は停止性の反証に特化したツールのため、他のツールと比べて停止性の証明に成功した問題数が少なくなっている。AProVE はほとんどの問題について証明に成功している。これは AProVE が複数の停止性証明手法を用いて停止性、あるいは非停止性の証明を試みるからである。

6.4 グレースフルネス

前節では高階関数について AMPO が停止性を証明できるものと証明できないものを示した。ここでは高階関数ではない一階の関数に注目し、AMPO が停止性を証明できる作用項書換え系のクラスについて議論する。

AMPO を MPO の作用項に対する一般化を目指して定義した。一階の項 s, t について必ず $s >_{\text{mpo}} t$ となるならば、そのカーリー化した項 s', t' について $s' \sqsubset_{\text{ampo}} t'$ となること望ましい。この性質をグレースフルネスという [3]。本節では AMPO が MPO に対し

てこの条件を満たすグレースフル^{*1} な一般化であることを示す。

まず一階の項に対応する作用項である、カーリー化された項を定義する。

定義 6.14. 項 t に対して カーリー化された項 $\hat{t}$ を次のように定める。

$$\hat{t} = \begin{cases} v & \text{if } t = v \in \mathcal{V} \\ f \hat{t}_1 \cdots \hat{t}_n & \text{if } t = f(t_1, \dots, t_n) \end{cases}$$

ある項 s, t が優先順序 $\succeq$ の下で $s >_{\text{mpo}} t$ となるならば、 s, t のカーリー化された項 $\hat{s}, \hat{t}$ について $\hat{s} \triangleright_{\text{ampo}} \hat{t}$ となる項数割当てと優先順序が存在することを示す。このとき項数割当てはカーリー化される前の項の関数記号が持つ項数をそのまま利用する。また優先順序も $\succeq$ に極大元 $\bullet$ を加えたものを用いる。

補題 6.15. 項 $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ について $s >_{\text{mpo}} t$ ならば $\hat{s} \triangleright_{\text{ampo}} \hat{t}$ が成り立つ。

証明. $s >_{\text{mpo}} t$ が 優先順序 $\succeq$ の下で成り立つとする。シグネチャ $\mathcal{F}$ の関数記号を全て定数記号と見なした作用型シグネチャ $\Sigma = \{f^{(0)} \mid f^{(n)} \in \mathcal{F}\} \cup \{o\}$ を考え、 $\succeq'$ を $f \succeq g$ ならば $f \succeq' g$ を満たす $\Sigma_\bullet$ 上の優先順序とする。また Σ に対する項数割当ては $\alpha = \{(f, n) \mid f^{(n)} \in \mathcal{F}\}$ とし、 $\mathcal{F}$ における項数をそのままに使用する。題意を $s >_{\text{mpo}} t$ の導出に関する帰納法で示す。 $s >_{\text{mpo}} t$ が **1** から導出されるとき $s = f(s_1, \dots, s_i, \dots, s_m)$ かつ $s_i >_{\text{mpo}} t$ となる。よって次の2つの場合を考える。 $s_i = t$ のとき $s >_{\text{mpo}} t$ の導出は

$$\frac{s_i = t}{s = f(s_1, \dots, s_i, \dots, s_m) >_{\text{mpo}} t} \quad \mathbf{1}$$

となる。 $\hat{s}_i = \hat{t}$ から

$$\frac{m = \alpha(f) \quad \hat{s}_i = \hat{t}}{\hat{s} = f \hat{s}_1 \cdots \hat{s}_i \cdots \hat{s}_m \triangleright_{\text{ampo}} \hat{t}} \quad \mathbf{1}$$

となり、 $\hat{s} \triangleright_{\text{ampo}} \hat{t}$ が成り立つ。 $s_i >_{\text{mpo}} t$ のとき $s >_{\text{mpo}} t$ は次のように導出される。

$$\frac{s_i >_{\text{mpo}} t}{s = f(s_1, \dots, s_i, \dots, s_m) >_{\text{mpo}} t} \quad \mathbf{1}$$

帰納法の仮定から $\hat{s}_i \triangleright_{\text{ampo}} \hat{t}$ が成り立ち、よって $\hat{s} \triangleright_{\text{ampo}} \hat{t}$ を次の導出から得る。

$$\frac{m = \alpha(f) \quad \hat{s}_i \triangleright_{\text{ampo}} \hat{t}}{\hat{s} = f \hat{s}_1 \cdots \hat{s}_i \cdots \hat{s}_m \triangleright_{\text{ampo}} \hat{t}} \quad \mathbf{1}$$

^{*1} デジタル通信の分野ではグレースフルデグラデーションという術語が、異なるシステム間における通信性能の劣化が下位互換である方のシステムの最高性能を下回らないという性質を意味するものとして使われている [21]。本性質名はこの術語からの借用である。

次に $s >_{\text{mpo}} t$ が **2** から得られるときを考える。

$$\frac{f > g \quad \forall j. s >_{\text{mpo}} t_j}{s = f(s_1, \dots, s_m) >_{\text{mpo}} g(t_1, \dots, t_n) = t} \quad \mathbf{2}$$

$f > g$ から $f >' g$ を得る。また帰納法の仮定から全ての $j, 1 \leq j \leq n$ に対して $\hat{s} \triangleright_{\text{ampo}} \hat{t}_j$ が成り立つ。よって次の導出から $\hat{s} \triangleright_{\text{ampo}} \hat{t}$ を得る。

$$\frac{m = \alpha(f) \quad n = \alpha(g) \quad f >' g \quad \forall j. \hat{s} >_{\text{mpo}} \hat{t}_j}{\hat{s} = f \hat{s}_1 \cdots \hat{s}_m \triangleright_{\text{ampo}} g \hat{t}_1 \cdots \hat{t}_n = \hat{t}} \quad \mathbf{2}$$

最後に $s >_{\text{mpo}} t$ が **3** から得られるときを考える。

$$\frac{f \sim g \quad \{s_1, \dots, s_m\} >_{\text{mpo}}^{\text{mul}} \{t_1, \dots, t_n\}}{s = f(s_1, \dots, s_m) >_{\text{mpo}} g(t_1, \dots, t_n) = t} \quad \mathbf{3}$$

帰納法の仮定より $\{\hat{s}_1, \dots, \hat{s}_m\} \triangleright_{\text{ampo}}^{\text{mul}} \{\hat{t}_1, \dots, \hat{t}_n\}$ が成り立つ。よって次の導出から $\hat{s} \triangleright_{\text{ampo}} \hat{t}$ を得る。

$$\frac{m = \alpha(f) \quad n = \alpha(g) \quad f \sim g \quad \{\hat{s}_1, \dots, \hat{s}_m\} \triangleright_{\text{ampo}}^{\text{mul}} \{\hat{t}_1, \dots, \hat{t}_n\}}{\hat{s} = f \hat{s}_1 \cdots \hat{s}_m \triangleright_{\text{ampo}} \hat{g} \hat{t}_1 \cdots \hat{t}_n = \hat{t}} \quad \mathbf{3}$$

□

カーリー化された項は明らかに頭部変数項を含まない。よって次の定理を得る。

定理 6.16. 項 $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ について $s >_{\text{mpo}} t$ ならば $\hat{s} \sqsupset_{\text{ampo}} \hat{t}$ が成り立つ。

3 章で述べたように、MPO は全ての原始帰納的関数を表現する項書換え系の停止性を証明することができる。よって定理 6.16 から AMPO が停止性を証明できる作用項書換え系のクラスは、原始帰納的関数を表現する作用項書換え系のクラスを包含する。

第 7 章

結論

本論文では多重集合経路順序（MPO）に基づく作用項上の新たな簡約順序である作用型多重集合経路順序（AMPO）と AMPO に基づく作用項書換え系の自動停止性証明の手法を提案した。本章では関連する研究を紹介し、最後に今後の課題について述べる。

7.1 uncurrying 変換との対応

作用項書換え系の停止性証明手法に uncurrying 変換 [13] が知られている。uncurrying 変換によって作用項書換え系 $\mathcal{R}$ から得らる項書換え系 $\mathcal{R}'$ について、 $\mathcal{R}$ が左辺無頭部変数であるとき $\mathcal{R}$ と $\mathcal{R}'$ の停止性は同値となる。例として次の作用項書換え系 $\mathcal{R}$ の停止性は

$$\begin{aligned}\text{map } f \text{ nil} &\rightarrow \text{nil} \\ \text{map } f (\text{cons } x \text{ } xs) &\rightarrow \text{cons } (f \ x) (\text{map } f \ xs)\end{aligned}$$

uncurrying 変換によって得られる項書換え系 $\mathcal{R}'$

$$\begin{aligned}\text{map}_2(f, \text{nil}_0) &\rightarrow \text{nil}_0 & \text{map}_0 \bullet x &\rightarrow \text{map}_1(x) \\ \text{map}_2(f, \text{cons}_2(x, xs)) &\rightarrow \text{cons}_2(f \bullet x, \text{map}_2(f \ xs)) & \text{map}_1(x) \bullet y &\rightarrow \text{map}_2(x, y) \\ & & \text{cons}_0 \bullet x &\rightarrow \text{cons}_1(x) \\ & & \text{cons}_1(x) \bullet y &\rightarrow \text{cons}_2(x, y)\end{aligned}$$

と同値になる。ここで $\mathcal{R}'$ の停止性は優先順序

$$\bullet \sim \text{map}_2 \sim \text{map}_1 \sim \text{map}_0 > \text{cons}_2 \sim \text{cons}_1 \sim \text{cons}_0$$

の下で MPO によって証明される。反対に $\mathcal{R}$ の停止性は AMPO によって次の優先順序

$$\bullet \sim \text{map} > \text{cons}$$

の下で証明できる。

7.2 作用項上の簡約順序

埋め込み経路順序 (embedding path order, EPO) [2] は作用項上の簡約順序として初めて提案された順序である。

定義 7.1. $>$ を $\Sigma \setminus \{\circ\}$ 上の優先順序とする。作用項 s, t に対して $s >_{\text{epo}} t$ を次のように帰納的に定義する。

$$\begin{array}{c} \frac{|\bar{s}| > 0 \quad \text{chop}(s) >_{\text{epo}}^= t}{s = f \bar{s} >_{\text{epo}} t} \quad \mathbf{1} \\[10pt] \frac{f > g \quad |\bar{t}| = 0 \vee s >_{\text{epo}} \text{chop}(t)}{s = f \bar{s} >_{\text{epo}} g \bar{t} = t} \quad \mathbf{2} \\[10pt] \frac{f \in \Sigma \quad \bar{s} \gg_{\text{epo}}^{\text{lex}} \bar{t} \quad |\bar{t}| = 0 \vee s >_{\text{epo}} \text{chop}(t)}{s = f \bar{s} >_{\text{epo}} f \bar{t} = t} \quad \mathbf{3} \\[10pt] \frac{f \in \mathcal{V} \quad |\bar{s}| > 0 \quad \bar{s} \gg_{\text{epo}}^{\text{lex}} \bar{t} \quad |\bar{t}| = 0 \vee \text{chop}(s) >_{\text{epo}} \text{chop}(t)}{s = f \bar{s} >_{\text{epo}} f \bar{t} = t} \quad \mathbf{4} \end{array}$$

ここで chop は $\text{chop}(f t_1 \cdots t_n) = t_1 \cdots t_n$ となる部分関数である。また 2 つの組 $(s_1, \dots, s_m), (t_1, \dots, t_n)$ に対して $(s_1, \dots, s_m) \gg^{\text{lex}} (t_1, \dots, t_n)$ は $m > n$ あるいは、 $m = n$ かつ $(s_1, \dots, s_m) >^{\text{lex}} (t_1, \dots, t_n)$ のときに成り立つ。

EPO は部分項性を持つ。これは chop が $\mathcal{E}_{\text{mb}}$ による頭部の切り落としに対応しているためである。例として 6.3 節で議論した $|x y \rightarrow x y$ を振り返る。 $|x y >_{\text{epo}} x y$ は次のように導出される。

$$\frac{|x y| > 0 \quad \text{chop}(|x y) = x y = x y}{|x y >_{\text{epo}} x y} \quad \mathbf{1}$$

しかし AMPO では 6.3 節で示したように $|x y \not>_{\text{ampo}} x y$ となる。一方で部分項性を持つことで、3 章で議論したように mul や map の向き付けを行うことができない。

作用型解釈順序 [26] は解釈順序と uncurrying 変換に基づく簡約順序である。例として作用型シグネチャ $\Sigma = \{\text{map}, \text{nil}, \text{cons}, \circ\}$ 上の作用項書換え系 $\mathcal{R}$ を考える。

$$\begin{aligned}\text{map } f \text{ nil} &\rightarrow \text{nil} \\ \text{map } f (\text{cons } x \ y) &\rightarrow \text{cons } (f \ x) (\text{map } f \ y)\end{aligned}$$

作用型解釈順序では Σ の関数記号に対して解釈と呼ばれる自然数上の関数を定める。

$$\begin{aligned}\text{map}_{2\mathcal{A}}(x, y) &= xy + x + 3y + 1 & \text{cons}_{2\mathcal{A}}(x, y) &= x + y + 1 & \text{nil}_{0\mathcal{A}} &= 0 \\ \text{map}_{1\mathcal{A}}(x) &= x + 2 & \text{cons}_{1\mathcal{A}}(x) &= x + 1 & \circ_{\mathcal{A}}(x, y) &= xy + x + y + 1 \\ \text{map}_{0\mathcal{A}} &= 2 & \text{cons}_{0\mathcal{A}} &= 1\end{aligned}$$

このとき任意の $f, x, y \in \mathbb{N}$ について次の不等式が成り立つ。

$$\begin{aligned}\text{map}_{2\mathcal{A}}(f, \text{nil}_{0\mathcal{A}}) &> \text{nil}_{0\mathcal{A}} \\ \text{map}_{2\mathcal{A}}(f, \text{cons}_{2\mathcal{A}}(x, y)) &> \text{cons}_{2\mathcal{A}}(\circ_{\mathcal{A}}(f, x), \text{map}_{2\mathcal{A}}(f, xs)) \\ \circ_{\mathcal{A}}(\text{map}_{0\mathcal{A}}, x) &\geq \text{map}_{1\mathcal{A}}(x) \\ \circ_{\mathcal{A}}(\text{map}_{1\mathcal{A}}(x), y) &\geq \text{map}_{2\mathcal{A}}(x, y) \\ \circ_{\mathcal{A}}(\text{cons}_{0\mathcal{A}}, x) &\geq \text{cons}_{1\mathcal{A}}(x) \\ \circ_{\mathcal{A}}(\text{cons}_{1\mathcal{A}}(x), y) &\geq \text{cons}_{2\mathcal{A}}(x, y)\end{aligned}$$

よって

$$\begin{aligned}\text{map } f \text{ nil} &>_{\mathcal{A}} \text{nil} \\ \text{map } f (\text{cons } x \ y) &>_{\mathcal{A}} \text{cons } (f \ x) (\text{map } f \ y)\end{aligned}$$

となり、 $\mathcal{R}$ の停止性が証明される。

7.3 自動定理証明への応用

作用項上の等式系 $\mathcal{E}$ を考える。

$$\begin{aligned}\text{add } x \ 0 &= x & \text{mul } x \ 0 &= 0 \\ \text{add } x \ (\text{s } y) &= \text{s } (\text{add } x \ y) & \text{mul } x \ (\text{s } y) &= \text{add } x (\text{mul } x \ y) \\ \text{nil} \ \text{++ } ys &= ys & \text{map } f \ \text{nil} &= \text{nil} \\ (x : xs) \ \text{++ } ys &= x : (xs \ \text{++ } ys) & \text{map } f \ (x : xs) &= (f \ x) : (\text{map } f \ xs)\end{aligned}$$

優先順序 $\bullet \sim \text{map} > ++ > : > \text{nil} > \text{mul} > \text{add} > \text{s} > 0$ を用いると AMPO によって $\mathcal{E}$ の等式は全て向き付き、以下の作用項書換え系 $\mathcal{R}$ を得る。

$$\begin{array}{ll}
\text{add } x \ 0 \rightarrow x & \text{mul } x \ 0 \rightarrow 0 \\
\text{add } x \ (\text{s } y) \rightarrow \text{s } (\text{add } x \ y) & \text{mul } x \ (\text{s } y) \rightarrow \text{add } x (\text{mul } x \ y) \\
\text{nil } ++ \ y \rightarrow y & \text{map } f \ \text{nil} \rightarrow \text{nil} \\
(x : xs) ++ \ y \rightarrow x : (xs ++ \ y) & \text{map } f \ (x : xs) \rightarrow (f \ x) : (\text{map } f \ xs)
\end{array}$$

このとき次の等式

$$\text{map } f \ (xs ++ \ y) = (\text{map } f \ xs) ++ (\text{map } f \ y)$$

が自然数のリストを表す基底項の上で成り立つことを $\mathcal{R}$ を用いて示す。

証明. 題意を $\sqsubset_{\text{ampo}}^{\text{mul}}$ に関する多重集合 $\{s, t\}$ 上の整礎帰納法で示す。 $xs = \text{nil}$ のとき左辺 $\text{map } f \ (\text{nil} ++ \ y)$ は $\mathcal{R}$ を用いて $\text{map } f \ y$ に書換えられる。また右辺は

$$(\text{map } f \ \text{nil}) ++ (\text{map } f \ y) \rightarrow_{\mathcal{R}} \text{nil} ++ (\text{map } f \ y) \rightarrow_{\mathcal{R}} \text{map } f \ y$$

となり、両辺は一致する。 $xs = x : xs'$ のとき左辺 $\text{map } f \ ((x : xs') ++ \ y)$ は

$$\begin{aligned}
\text{map } f \ ((x : xs') ++ \ y) &\rightarrow_{\mathcal{R}} \text{map } f \ (x : (xs' ++ \ y)) \\
&\rightarrow_{\mathcal{R}} (f \ x) : (\text{map } f \ (xs' ++ \ y))
\end{aligned}$$

となり、右辺 $(\text{map } f \ (x : xs')) ++ (\text{map } f \ y)$ は

$$\begin{aligned}
(\text{map } f \ (x : xs')) ++ (\text{map } f \ y) &\rightarrow_{\mathcal{R}} ((f \ x) : (\text{map } f \ xs')) ++ (\text{map } f \ y) \\
&\rightarrow_{\mathcal{R}} (f \ x) : ((\text{map } f \ xs') ++ (\text{map } f \ y))
\end{aligned}$$

となる。このとき

$$\left\{ \begin{array}{l} \text{map } f \ ((x : xs') ++ \ y), \\ (\text{map } f \ (x : xs')) ++ (\text{map } f \ y) \end{array} \right\} \sqsubset_{\text{ampo}}^{\text{mul}} \left\{ \begin{array}{l} \text{map } f \ (xs' ++ \ y), \\ (\text{map } f \ xs') ++ (\text{map } f \ y) \end{array} \right\}$$

が成り立つため、帰納法の仮定より

$$\text{map } f \ (xs' ++ \ y) = (\text{map } f \ xs') ++ (\text{map } f \ y)$$

が得られる。またこの等式について次の向き付け

$$\text{map } f \ (xs' ++ \ y) \sqsubset_{\text{ampo}} (\text{map } f \ xs') ++ (\text{map } f \ y)$$

が得られることから帰納法の仮定を書換え規則

$$\text{map } f (xs' ++ ys) \rightarrow (\text{map } f xs') ++ (\text{map } f ys)$$

とし、 $\mathcal{R}$ に加えたものを $\mathcal{R}'$ とする。よって

$$\begin{aligned} \text{map } f ((x : xs') ++ ys) &\rightarrow_{\mathcal{R}'}^* (f x) : ((\text{map } f xs') ++ (\text{map } f ys)) \\ (\text{map } f (x : xs')) ++ (\text{map } f ys) &\rightarrow_{\mathcal{R}'}^* (f x) : ((\text{map } f xs') ++ (\text{map } f ys)) \end{aligned}$$

となり、両辺は一致する。 □

このとき EPO は map の規則を向き付けられず、また表 6.3 から作用型解釈順序は $\mathcal{R}$ を向き付ける解釈の発見に失敗する。しかし AMPO は帰納法の仮定も含める $\mathcal{R}'$ を向き付けることができる。

7.4 今後の課題

本論文では多重集合経路順序に基づく新たな作用項上の順序を構成した。作用型多重集合経路順序を構成したアプローチは他の簡約順序にも同様に適用できると考えられる。

辞書式経路順序 (lexicographic path order, LPO) [15] は MPO と同じ再帰経路順序の一つであり、MPO の **3** を次の規則に替えたものである。ここで $>^{\text{lex}}$ は $>$ の辞書式拡張である。

$$\frac{f \sim g \quad (s_1, \dots, s_m) >_{\text{lpo}}^{\text{lex}} (t_1, \dots, t_n) \quad \forall j. s >_l \text{pot}_j}{s = f(s_1, \dots, s_m) >_{\text{lpo}} g(t_1, \dots, t_n)}$$

LPO と MPO の差異はこの規則のみである。よって次の規則

$$\frac{m = \alpha(f) \quad n = \alpha(g) \quad f \sim g \quad (s_1, \dots, s_m) \triangleright_{\text{alpo}}^{\text{lex}} (t_1, \dots, t_n) \quad \forall j. s \triangleright_{\text{alpo}} t_j}{s = f s_1 \cdots s_m \triangleright_{\text{alpo}} g t_1 \cdots t_n}$$

を AMPO の **3** と置き換えることで作用項に対する LPO である ALPO を構成することは可能であると考えられる。

重み付き経路順序 [25] は解釈順序と再帰経路順序を包含した強力な順序である。この重み付き経路順序について、作用型解釈順序と作用型多重集合経路順序、あるいは作用型辞書式経路順序から作用項に対する拡張である作用型重み付き経路順序を構成することは可能であると考えられる。

■謝辞 本研究に際し廣川直准教授からは多大なご指導を頂きました。ここに深く感謝申し上げます。また研究活動を支えてくださった研究室の皆様にも感謝いたします。

参考文献

- [1] F. Baader and T. Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998.
- [2] A. Bentkamp. The embedding path order for lambda-free higher-order terms. *Journal of Applied Logics*, 8(10):2447–2470, 2021.
- [3] J. C. Blanchette, U. Waldmann, and D. Wand. A lambda-free higher-order recursive path order. In *Proc. 20th International Conference on Foundations of Software Science and Computation Structures*, volume 10203 of *Lecture Notes in Computer Science*, pages 461–479, 2017.
- [4] W. Buchholz. Proof-theoretic analysis of termination proofs. *Annals of Pure and Applied Logic*, 75(1-2):57–65, 1995.
- [5] M. Codish, J. Giesl, P. Schneider-Kamp, and R. Thiemann. SAT solving for termination proofs with recursive path orders and dependency pairs. *Journal of Automated Reasoning*, 49(1):53–93, 2012.
- [6] L. M. de Moura and N. S. Bjørner. Z3: an efficient SMT solver. In *Proc. 14th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, volume 4963 of *Lecture Notes in Computer Science*, pages 337–340, 2008.
- [7] N. Dershowitz. Orderings for term-rewriting systems. *Theoretical Computer Science*, 17:279–301, 1982.
- [8] N. Dershowitz. Termination of rewriting. *Journal of Symbolic Computation*, 3(1/2):69–116, 1987.
- [9] N. Dershowitz. Primitive rewriting. In *Processes, Terms and Cycles: Steps on the Road to Infinity, Essays Dedicated to Jan Willem Klop, on the Occasion of His 60th Birthday*, volume 3838 of *Lecture Notes in Computer Science*, pages

- 127–147. Springer, 2005.
- [10] N. Dershowitz and Z. Manna. Proving termination with multiset orderings. In *Proc. 6th Colloquium on Automata, Languages and Programming*, volume 71 of *Lecture Notes in Computer Science*, pages 188–202. Springer, 1979.
 - [11] J. Giesl, M. Brockschmidt, F. Emmes, F. Frohu., C. Fuhs, C. Otto, M. Plücker, P. Schneider-Kamp, T. Ströder, S. Swiderski, and R. Thiemann. Proving termination of programs automatically with AProVE. In *Proc. 7th International Joint Conference on Automated Reasoning*, volume 8562 of *Lecture Notes in Computer Science*, pages 184–191. Springer, 2014.
 - [12] N. Hirokawa and A. Middeldorp. Dependency pairs revisited. In *Proc. 15th International Conference on Rewriting Techniques and Applications*, volume 3091 of *Lecture Notes in Computer Science*, pages 249–268. Springer, 2004.
 - [13] N. Hirokawa, A. Middeldorp, and H. Zankl. Uncurrying for termination and complexity. *Journal of Automated Reasoning*, 50(3):279–315, 2013.
 - [14] J.-P. Jouannaud and A. Rubio. The higher-order recursive path ordering. In *Proc. 14th Symposium on Logic in Computer Science*, pages 402–411. IEEE Computer Society, 1999.
 - [15] S. Kamin and J.-J Lévy. Two generalizations of the recursive path ordering. unpublished note, 1980.
 - [16] D. Lankford. On proving term rewriting systems are noetherian. Technical report, Louisiana Technical University, 1979.
 - [17] A. Middeldorp. Term rewriting. Lecture notes (unpublished manuscript), 2023.
 - [18] É. Payet. NTI (Non-Termination Inference). <http://lim.univ-reunion.fr/staff/epayet/Research/NTI/NTI.html>. Accessed on 23/01/2025.
 - [19] D. A. Plaisted. A recursively defined ordering for proving termination of systems of rewrite rules. Report R-78-943, Department of Computer Science, University of Illinois, Urbana, IL, 1978.
 - [20] U. S. Reddy. Term rewriting induction. In *Proc. 10th International Conference on Automated Deduction*, volume 449 of *Lecture Notes in Computer Science*, pages 162–177. Springer, 1990.
 - [21] G. R. Redinbo. Optimum soft decision decoding with graceful degradation. *Information and Control*, 41(2):165–185, 1979.
 - [22] Termination Community. Termination Competition 2024. <https://>

- `termination-portal.org/wiki/Termination_Competition_2024`. Accessed on 20/01/2025.
- [23] Termination Community. The Termination Problem Database (TPDB). <https://github.com/TermCOMP/TPDB>. Accessed on 17/12/2024.
 - [24] A. Yamada, K. Kusakari, and T. Sakabe. Nagoya termination tool. In *Proc. Joint International Conference on Rewriting and Typed Lambda Calculi*, volume 8560 of *Lecture Notes in Computer Science*, pages 466–475. Springer, 2014.
 - [25] A. Yamada, K. Kusakari, and T. Sakabe. A unified ordering for termination proving. *Science of Computer Programming*, 111:110–134, 2015.
 - [26] 田中 哲平. 作用項書換えのための解釈順序. 修士論文, 北陸先端科学技術大学院大学, 2023.