合字化と文字列化に基づく書換えシステムの停止性解析

山﨑 健人

平成 28 年 3 月 23 日

目 次

第1章　はじめに

1.1　研究の目的と背景

研究目的　現代社会は、原子力発電所、プラント、電気・通信・交通のインフラシステムから身近な携帯電話や自動車まで、あらゆるものがソフトウェアに立脚している。多くの場合、プログラム実行の非停止性はソフトウェアの暴走を意味し、停止性を検証する技術は安全性の検証に不可欠である。項書換えは等式を左辺から右辺へと向きを持つ書換え規則とみなせる計算モデルである。また、書換えは宣言型言語 (CafeOBJ, OCaml 等) の基盤として用いられる計算モデルであり、文字列書換えは計算の対象を文字列に限定した項書換えである。項書換えシステムや文字列書換えシステムは無限の書換えの列を持たない時に停止性をもつ。停止性を持つ書換えシステムは何らかの答えが導出することが決まっているため停止性は重要な性質である。しかし、書換えシステムの停止性は決定できないため書換えシステムが停止性を持つことを研究することが盛んに行われている。本研究では書換え、特に文字列書換えの停止性解析の研究を行う。文字列書換えは書換え分野においては、応用に適した項書換えに先立ち、文字列書換えで萌芽的な研究を行うアプローチがしばしば用いられる。行列順序 [9] やマッチバウンド [6] はその一例である。文字列書換えシステムの自動化に適した簡単な停止性証明の手法の開発および文字列書換えシステムの停止性証明手法を項書換えシステムにおいても使用可能なことを示す。

研究背景　古典的な停止性証明手法は簡約順序と呼ばれる整礎順序で書換え規則を順序付ける方法である [4] 。もし、ある簡約順序である書き換え規則が順序付くなら、$\mathcal{R}$ は停止する。簡約順序には定理証明システムで標準的に用いられている辞書式経路順序 [10] 、Knuth-Bendix 順序 (KBO) [11] や、停止性ツールで多用されている多項式解釈順序 (polynomial interpretations) [4, 15] など様々な種類がある。この方法では実用上、前者の方が停止性の (自動) 証明に適していることが経験的に知られている。2000 年に依存対と呼ばれる変換手法が導入された [3]。これは簡約対と呼ばれる整礎順序と擬順序の対を用いて停止性を証明する。簡約対は簡約順序をベースに構成されるが、依存対の下では逆に多項式順序が極めて有効であることが知られている [14, 15]。

本研究では新たな簡約順序として合字順序を提案し、依存対との組み合わせを試す。依存対を使用する際の簡約対を設計する手法として、引数切り落とし法 [3] を定義する。また、文字列化により文字列書換えシステムの停止性技法を項書換えシステムで使用できることを示す。これら手法を実装評価する。

1.2　成果

本論文では、合字順序、文字列化順序という停止性解析技法を提案する。一つ目として合字化に基づく合字順序を提案する。

まずは合字順序の動機となった例を紹介したい。停止性解析の手法として辞書式経路順序、Knuth-Bendix 順序や polynomial interpretations が知られている。これら有名な手法が停止性を判定できない例として組紐問題がある。組紐理論とは三つ編みを考えたようなものである。次に出てくる ab、ba 等は三つ編みをあむ手順を表している。

図 1.1: 組紐の表記法

図 1.1 を使った組紐の例を図 1.2 に示す

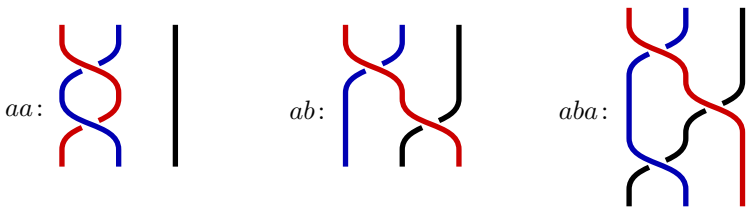

図 1.2: 組紐の例

あむ手順が違っても最終的に同じ三つ編みができる場合がある。違う手順で同じ三つ編みになるか判定するのがこのシステムである。

図 1.3 の aba と bab は同値である。aba と bab はどちらも赤い線は三つの線の一番上を、青い線は真ん中を、黒い線は一番下を通っており紐が同じ行

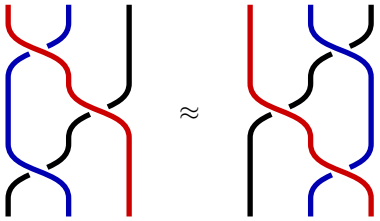

図 1.3: 同値な組紐 aba と bab

き先となっている。組紐問題である次の書換え規則の集合を考える。

$$\mathcal{R} = \left\{ \begin{array}{rcll} aaba & \to & abab & (1) \\ bbab & \to & baba & (2) \\ ca & \to & ac & (3) \\ caba & \to & bacb & (4) \end{array} \right\}$$

この $\mathcal{R}$ を用いると、二つの文字列 s, t が組紐として同じであるかを cs と ct を $\mathcal{R}$ に従って書き換えていくことで判定できる。もし、書換えられなくなるまで書き換えた結果が同一であれば組紐の構造は同じであり、そうでなければ異なる。この $\mathcal{R}$ は停止性を持つがその証明は自明ではない。この書換え集合の (1) と (2) に着目すると左辺の隣り合った aa と bb が右辺ではなくなっているのがわかる。この問題から隣り合った文字同士を何らかの方法で一緒に比べるという着想を得た。そこで、安易ではあるが隣合った文字同士を合体させる合字化の着想を得た。合字化は隣同士の文字を全て連結させることである。合字化すると次のような文字列書換えシステムができる。順序の一つ目は (1)、一つ目は (2) の規則と対応している。

$$\underline{aa}\ ab\ ba \to \underline{ab}\ ba\ ab$$
$$\underline{bb}\ ba\ ab \to \underline{ba}\ ab\ ba$$

この文字列書換えシステムにおいて aa が ab、bb が ba より優先順位が高いとすると下線の部分で左辺が大きくなる。しかし、(3、4) は単純に合字化しただけでは簡約順序にはならない。必要な条件を 3 章で説明する。さらに、合字化と長さ辞書式順序を組み合わせた長さ辞書式合字順序という手法を提案する。長さ辞書式順序は左辺が右辺より長ければ文字列書換えシステムの停止性を判定できる。長さ辞書式合字順序は、長さ辞書式順序の特性より左辺が右辺より長いもしくは同じ場合のみにしか停止性を判定できない。そのため、右辺が左辺より長い時でも停止性を判定可能にするために引数切り落としを導入した。引数切り落としを導入することで右辺が左辺より長い場合の文字列書換えシステムのときに停止性を判定できる場合があることが示す。その際の例として $\mathcal{R} = \{ab \to ba, ba \to acb\}$ がある。二つ目の規則の c よ

り後ろの文字を考えないとすると長さ辞書式合字順序で停止性を判定できる。詳しい説明は 3 章に記述する。

また Ngo Thi Bao Tran [18] によって文字列化と呼ばれる項を文字列の集合に変換する方法が提案された。Ngo Thi Bao Tran はこの方法を用いて arctic interpretation [12] を用いて停止性判定するためのコンパクトな制約式への変換を与えた。文字列化を次の項書換えシステムの例を用いて説明する。

$$\begin{aligned} 1\colon &\quad \mathsf{len}(\mathsf{nil}) \to \mathsf{0} \\ 2\colon &\quad \mathsf{len}(\mathsf{cons}(x, y)) \to \mathsf{s}(\mathsf{len}(y)) \end{aligned}$$

依存対により次の規則が追加される。

$$3\colon \quad \mathsf{len}^\sharp(\mathsf{cons}(x, y)) \to \mathsf{len}^\sharp(y)$$

次の式に登場する $\succsim^{\mathsf{H}}$ はべき集合においての関係である。この関係は定義 31 を参照されたい。

$$\begin{aligned} 1\colon &\quad \left\{ \begin{matrix} \mathsf{len} \\ \mathsf{len}_1\mathsf{nil} \end{matrix} \right\} \succsim^{\mathsf{H}} \varnothing \\ 2\colon &\left\{ \begin{matrix} \mathsf{len} \\ \mathsf{len}_1\mathsf{cons} \end{matrix} \right\} \succsim^{\mathsf{H}} \left\{ \begin{matrix} \mathsf{s} \\ \mathsf{s}_1\mathsf{len} \end{matrix} \right\} \quad \{\mathsf{len}_1\mathsf{cons}_2\} \succsim^{\mathsf{H}} \{\mathsf{s}_1\mathsf{len}_1\} \\ 3\colon &\left\{ \begin{matrix} \mathsf{len}^\sharp \\ \mathsf{len}^\sharp_1\mathsf{cons} \end{matrix} \right\} \succ^{\mathsf{H}} \{\mathsf{len}^\sharp\} \qquad \{\mathsf{len}^\sharp_1\mathsf{cons}_2\} \succ^{\mathsf{H}} \{\mathsf{len}^\sharp_1\} \end{aligned}$$

集合に登場している添字は何引数目かを表している。文字列化をすると項がアルファベット

$$\{\mathsf{len}, \mathsf{len}_1, \mathsf{len}^\sharp, \mathsf{cons}, \mathsf{cons}_2, \mathsf{s}, \mathsf{s}_1, \mathsf{nil}, \mathsf{0}\}$$

上の文字列の集合になるが、上手く比較すると文字列書換えの手法で停止性を判定できる。その一般的な枠組みとして文字列化順序を提案する。

本論文は次のような構成となっている。2 章では文字列、項書換えシステムや停止性について説明を行う。3 章 では私達が新たに提唱する合字化について説明する。さらに、長さ辞書式合字順序における引数切り落としや、符号化についても紹介する。4 章では、文字列化順序について説明する。文字列化順序は文字列書換えの停止性技法を、項書換えの場合に適用できるようにした手法である。5 章には 3、4 章で紹介した合字化と文字列化の手法を実装したツールの停止性解析の結果を載せた。6 章には本研究の成果の概要と今後の課題を述べる。

第2章　書換えシステム

この章では、項書換え、文字列書換えや停止性についての表記法や定義について紹介する。定義記法は [4, 5] に従う。

2.1　文字列書換え

この節では文字列書換えについて定義する。文字列書換えは項書換えの一種であり、応用に適した項書換えに先立ち、文字列書換えで萌芽的な研究を行うアプローチがしばしば用いられる。最初に、書換えシステムにおいて重要な二項関係の表記法を定義する。

定義 1. A を集合とする。二項関係または A 上の関係は、直積集合 $A \times A$ の部分集合である。 A 上の関係 $\to$ は次のように呼ばれる。

- **反射的**とは、全ての $a \in A$ に対して $a \to a$ のときである。
- **非反射的**とは、全ての $a \in A$ に対して $a \to a$ でないときである。
- **対称的**とは、$a \to b$ ならば $b \to a$ が成り立つときである。
- **推移的**とは、$\to \cdot \to \subseteq \to$ のときである。
- **同値関係**とは、反射的で対称的で推移的なときである。
- **厳格順序**とは、非反射的で推移的なときである。
- **半順序**とは、対称的でなく反射的かつ推移的なときである。
- **擬順序**とは、反射的で推移的なときである。
- **無限下降列が存在しない**とは、$a_1 \to a_2 \to a_3 \to \cdots$ のような無限列が存在しないことである。

文字列について定義する。

定義 2. 文字の有限集合 Σ を考える。Σ 上の文字の有限列を**文字列** (string) と呼ぶ。また、0 個以上の文字の集合を Σ^*、1 個以上の文字の集合を Σ^+ である。**空文字列**を ϵ で表記する。文字列 x の文字数は $|x|$ で表記する。

次に文字列書換えシステムについて定義する。

定義 3. $\Sigma^+ \times \Sigma^*$ の部分集合 $\mathcal{R}$ を**文字列書換えシステム** (string rewrite system, SRS) と呼ぶ。関係 $\{(u\ell v, urv) \mid \ell \to r \in \mathcal{R}$ かつ $u, v \in \Sigma^*\}$ を $\to_{\mathcal{R}}$ で表記する。

例 4. 文字列書換えシステムは次のように表記する。

$$\mathcal{R}_1 = \{\ aa \ \to \ aba\ \}$$

$$\mathcal{R}_2 = \left\{ \begin{array}{rcl} aa & \to & aba \\ b & \to & \epsilon \end{array} \right\}$$

今後例を用いる際特別な記述がない場合は今回の文字列書換えシステム $\mathcal{R}_1$ または $\mathcal{R}_2$ のこととする。

定義 5. 文字列書換えシステム $\mathcal{R}$ が停止性を持つとは $\to_{\mathcal{R}}$ に関する無限の書換え列が存在しないことを言う。

例 6. 文字列書換えシステム $\mathcal{R}_1$ は停止性を持ち $\mathcal{R}_2$ は停止性を持たない。

$$aaa \to_{\mathcal{R}_1} abaa \to_{\mathcal{R}_1} ababa$$

$$aa \to_{\mathcal{R}_2} aba \to_{\mathcal{R}_2} aa \to_{\mathcal{R}_2} \cdots$$

定義 7. 文字列上の関係 R に対し以下の性質を定義する。

- 左の文脈について閉じているとは sRt ならば $usRut$ が成立することである。
- 右の文脈について閉じているとは sRt ならば $suRtu$ が成立することである。

定義 8. 文字列を考えた際に全ての文字列 s, t, u, v に対して $s > t$ ならば $usv > utv$ が成り立つ整礎な順序 $>$ を簡約順序と呼ぶ。

定理 9. 全ての書換え規則 $\ell \to r \in \mathcal{R}$ に対し、 $\ell > r$ となる簡約順序 $>$ が存在するとき文字列書換えシステム $\mathcal{R}$ は停止性を持つ。

2.2　項書換え

この節では項書換えの基本的な性質を定義する。まず、項書換えの項を定義する。

定義 10. 関数記号の集合を $\mathcal{F}$ と表記する。各関数記号 f は固有の**アリティ** (引数の数) を持ち、 $f^{(n)}$ という表記は f が n 引数取ることを表す。変数の集合を $\mathcal{V}$ と表記し $\mathcal{F} \cap \mathcal{V} = \varnothing$ と仮定する。**項**全体の集合 $\mathcal{T}(\mathcal{F}, \mathcal{V})$ を以下のように帰納的に定義する。

- もし $x \in \mathcal{V}$ ならば $x \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ である。
- もし $f^{(n)} \in \mathcal{F}$ かつ、全ての $1 \leqslant i \leqslant n$ において $t_i \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ ならば、$f(t_1, \cdots, t_n) \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ である。

項について操作を定義する。

定義 11. $t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ とする。

- t の変数全体の集合 $\mathcal{V}\mathsf{ar}(t)$ は以下のように定義される。

$$\mathcal{V}\mathsf{ar}(t) = \begin{cases} \{t\} & t \text{ が変数のとき} \\ \mathcal{V}\mathsf{ar}(t_1) \cup \cdots \cup \mathcal{V}\mathsf{ar}(t_n) & t = f(t_1, \cdots, t_n) \text{ のとき} \end{cases}$$

- t の関数記号全体の集合 $\mathcal{F}\mathsf{un}(t)$ は以下のように定義される。

$$\mathcal{F}\mathsf{un}(t) = \begin{cases} \varnothing & t \text{ が変数のとき} \\ \{f\} \cup \mathcal{F}\mathsf{un}(t_1) \cup \cdots \cup \mathcal{F}\mathsf{un}(t_n) & t = f(t_1, \cdots, t_n) \text{ のとき} \end{cases}$$

- t の位置全体の集合 $\mathcal{P}\mathsf{os}(t)$ は以下のように定義される。

$$\mathcal{P}\mathsf{os}(t) = \begin{cases} \{\epsilon\} & t \text{ が変数のとき} \\ \{\epsilon\} \cup \bigcup_{i=1}^{n} i \cdot \mathcal{P}\mathsf{os}(t_i) & t = f(t_1, \cdots, t_n) \text{ のとき} \end{cases}$$

  ここで、ϵ は空文字列を表し根位置と呼ぶ。また $i \cdot \mathcal{P}$ は $\{ip \mid p \in \mathcal{P}\}$ を表す。

- **根**記号 $\mathsf{root}(t)$ は以下のように定義される。

$$\mathsf{root}(t) = \begin{cases} t & t \text{ が変数のとき} \\ f & t = f(t_1, \cdots, t_n) \text{ のとき} \end{cases}$$

- t の**サイズ**は以下のように定義される。

$$\mathsf{size}(t) = \begin{cases} 1 & t \text{ が変数のとき} \\ 1 + \sum_{i=1}^{n} \mathsf{size}(t_i) & t = f(t_1, \cdots, t_n) \text{ のとき} \end{cases}$$

定義 12. 項として s と t を考える。t の位置 p における**部分項** (subterm) を以下のように定義する。

$$t|_p = \begin{cases} t & p = \epsilon \text{ のとき} \\ t_i|_q & t = f(t_1, \cdots, t_n) \text{ かつ } p = iq \text{ のとき} \end{cases}$$

$t[s]_p$ という表記法は、t の位置 p における部分項を s に置き換えることで得られる項を表し、以下のように定義される。

$$t[s]_p = \begin{cases} s & p = \epsilon \text{ のとき} \\ f(t_1, \cdots, t_i[s]_p, \cdots, t_n) & t = f(t_1, \cdots, t_n) \text{ かつ } p = iq \text{ のとき} \end{cases}$$

例 13. 項 $t = \mathsf{plus}(\mathsf{s}(\mathsf{0}), \mathsf{s}(x))$ について考えると、$\mathcal{V}\mathsf{ar}(t) = \{x\}$, $\mathcal{F}\mathsf{un}(t) = \{\mathsf{s}, \mathsf{0}, \mathsf{plus}\}$ となり、項 t の部分項は、$\mathsf{plus}(\mathsf{s}(\mathsf{0}), \mathsf{s}(x)), \mathsf{s}(\mathsf{0}), \mathsf{0}, \mathsf{s}(x), x$ となる。

定義 14. ホールと呼ばれる定数記号 $\Box$ を一つ定める。**文脈** (context) とは、正確に一つホール $\Box$ が含まれている $\mathcal{T}(\mathcal{F} \uplus \{\Box\}, \mathcal{V})$ 項である。文脈 C と t が与えられると、$C[t]$ は次のように帰納的に定義される。

$$C[t] = \begin{cases} t & C = \Box \text{ のとき} \\ f(t_1, \cdots, C'[s], \cdots, t_n) & C = f(t_1, \cdots, C', \cdots, t_n) \text{ のとき} \end{cases}$$

ただし C' は文脈である。

定義 15. 代入 (substitution) とは、$\mathcal{V}$ から $\mathcal{T}(\mathcal{F}, \mathcal{V})$ への写像 σ であり、かつ $\mathcal{D}\mathsf{om}(\sigma)$ が有限であるものを言う。ここで $\mathcal{D}\mathsf{om}(\sigma)$ は集合 $\{x \in \mathcal{V} \mid \sigma(x) \neq x\}$ を表す。項 t への代入 $t\sigma$ は以下のように定義される。

$$t\sigma = \begin{cases} \sigma(t) & t \text{ が変数のとき} \\ f(t_1\sigma, \cdots, t_n\sigma) & t = f(t_1, \cdots, t_n) \text{ のとき} \end{cases}$$

2 つの σ と τ の代入である**合成** (composition) $\sigma\tau$ は $\{x \mapsto (x\sigma)\tau \mid x \in \mathcal{V}\}$ という代入である。

定義 16. 項上の関係 $\to$ に対し以下の性質を定義する。

- **文脈について閉じている** (closed under contexts) とは $s \to t$ かつ C が文脈のときに $C[s] \to C[t]$ が成立することを言う。
- **代入について閉じている** (closed under substitutions) とは $s \to t$ かつ σ が代入のときに $s\sigma \to t\sigma$ が成立することを言う。

書換え関係 (rewrite relation) とは文脈と代入について閉じている関係である。

項書換えシステムについて記述する。

定義 17. 書換え規則 (rewrite rule) とは $\ell \notin \mathcal{V}$ と $\mathcal{V}\mathsf{ar}(r) \subseteq \mathcal{V}\mathsf{ar}(\ell)$ を満たす項の対 $\ell \to r$ のことである。**項書換えシステム** (term rewrite system, TRS) とは書換え規則の集合 $\mathcal{R}$ である。項書換えシステムの $\mathcal{R}$ を含む最小の書換え関係を $\to_{\mathcal{R}}$ と表記する。$\to_{\mathcal{R}}$ は $\mathcal{R}$ の**書換え関係**と呼ばれる。

次のように文字列は項とみなすことができる。変数 x を固定し、各文字 a を 1 引数の関数記号とみなす。文字列 $a_1a_2\cdots a_n$ を $a_1(a_2(\cdots(a_n(x))\cdots))$ と表せる。たとえば文字列書換えシステム $\mathcal{R}_1$ を考えると $a(a(x)) \to a(b(a(x)))$ と表記できる。

項書換えシステムが停止性を持つかは決定出来ない。そのため、項書換えシステムが停止性を持つか調査することは重要である。書換えシステムにおいて重要な停止性を定義する。

定義 18. 項書換えシステム $\mathcal{R}$ が**停止性** (termination) を持つとは $\to_{\mathcal{R}}$ による無限の書換え列が存在しないことを言う。

例 19. 次の項書換えシステム $\mathcal{R}_3$ を考える。len はリストの長さを測るシステムである。0、1、2、... をそれぞれ 0、s(0)、s(s(0))、... と表記する。

$$\mathcal{R}_3 = \left\{ \begin{array}{rcl} \mathsf{len}(\mathsf{nil}) & \to & \mathsf{0} \\ \mathsf{len}(\mathsf{cons}(x,y)) & \to & \mathsf{s}(\mathsf{len}(y)) \end{array} \right\}$$

項 $\mathsf{len}(\mathsf{cons}(\mathsf{a},\mathsf{nil}))$ は次のように書換えられる。

$$\mathsf{len}(\mathsf{cons}(\mathsf{a},\mathsf{nil})) \to_{\mathcal{R}_3} \mathsf{s}(\mathsf{len}(\mathsf{nil})) \to_{\mathcal{R}_3} \mathsf{s}(\mathsf{0})$$

これ以上書き換えることが出来ないのでこの書換えは終了する。

書換えシステムの停止性を判定するためにしばしば用いられる簡約順序を定める。

定義 20. 関係 $>$ が**整礎** (well-founded) とは、無限下降列 $s_1 > s_2 > \cdots$ が存在しないことを言う。文脈と代入について閉じている整礎な厳格順序を**簡約順序** (reduction order) と呼ぶ。

定理 21. 文脈と代入について閉じている整礎な全ての書換え規則 $\ell \to r \in \mathcal{R}$ に対し $\ell > r$ が成立するならば項書換えシステム $\mathcal{R}$ は停止性を持つ。

項書換えシステムの停止性を解析できる依存対という手法がある [3]。依存対は停止性を証明する手法として強力なことが知られており、有名な停止性ツールである TTT2 [13] や AProVE [7] にもこの手法が使われている。

定義 22. 項書換えシステム $\mathcal{R}$ を考える。**被定義記号** (defined symbol) とは $\{\mathsf{root}(\ell) \mid \ell \to r \in \mathcal{R}\}$ の要素のことである。被定義記号全体の集合を $\mathcal{D}_{\mathcal{R}}$ で表記する。$f \in \mathcal{D}_{\mathcal{R}}$ のそれぞれに対し $f^\sharp$ という記号を用いる。$f^\sharp$ を $\mathcal{F}$ に属さない同じアリティの関数記号とする。項 $t^\sharp$ を $f^\sharp(t_1,\cdots,t_n)$ と書く。項の対 $l^\sharp \to u^\sharp$ が $\mathcal{R}$ の**依存対** (dependency pair) であるとは全ての書換え規則 $\ell \to r \in \mathcal{R}$ に対して、u が r の部分項で $\mathsf{root}(u) \in \mathcal{D}_{\mathcal{R}}$ が成立し、かつ ℓ の真部分項ではないときである。依存対全体の集合を $\mathsf{DP}(\mathcal{R})$ で表記する。

例 23. 次の項書換えシステム $\mathcal{R}_4$ を考え依存対を構成する。

$$\mathcal{R}_4 = \left\{ \begin{array}{rcl} \mathsf{0}-\mathsf{y} & \to & \mathsf{0} \\ x-\mathsf{y} & \to & x \\ \mathsf{s}(x)-\mathsf{s}(y) & \to & x-y \\ \mathsf{0}\div\mathsf{s}(y) & \to & \mathsf{0} \\ \mathsf{s}(x)\div\mathsf{s}(y) & \to & \mathsf{s}((x-y)\div\mathsf{s}(y)) \end{array} \right\}$$

被定義記号全体の集合 $\mathcal{D}_\mathcal{R}$ が $\{-, \div\}$ となることから依存対全体の集合 $\mathsf{DP}(\mathcal{R})$ は次のようになる。

$$\mathsf{DP}(\mathcal{R}_4) = \left\{ \begin{array}{rcl} \mathsf{s}(x)-^\sharp\mathsf{s}(y) & \to & x-^\sharp y \\ \mathsf{s}(x)\div^\sharp\mathsf{s}(y) & \to & (x-y)\div^\sharp\mathsf{s}(y) \\ \mathsf{s}(x)\div^\sharp\mathsf{s}(y) & \to & (x-^\sharp y) \end{array} \right\}$$

書換えシステムは有限の依存対問題 $(\mathcal{P}, \mathcal{R})$ とみなせる。

定義 24. 項書換えの依存対問題は $(\mathcal{P}, \mathcal{R})$ の対である。$\mathcal{P}$ は書換え規則の右辺が変数ではなく被定義記号は $\mathcal{R}$ に現れない関数記号である。s と t の任意の真部分項は $s \to t \in \mathcal{P}$ とはならない。各 t_i が $\mathcal{R}$ に対して停止性を持つ無限列

$$t_1 \to_\mathcal{R}^* t_2 \xrightarrow{\epsilon}_\mathcal{P} t_3 \to_\mathcal{R}^* t_4 \xrightarrow{\epsilon}_\mathcal{P} \cdots$$

が存在しないとき依存対問題 $(\mathcal{P}, \mathcal{R})$ は有限であるという。

項書換えシステム $\mathcal{R}$ を $a(a(x)) \to a(b(a(x)))$ とみなしたとき依存対 $\mathsf{DP}(\mathcal{R})$ は $a^\sharp(a(x)) \to a^\sharp(b(a(x)))$ であり、この $\mathsf{DP}(\mathcal{R})$ を文字列書換えシステムの表記に戻すと $a^\sharp a \to a^\sharp ba$ となる。そのため以降の項書換えシステムに対する定義にも文字列書換えシステムは対応する。

次の定理は Arts and Gisel [3] による。

定理 25. 依存対問題 $(\mathsf{DP}(\mathcal{R}), \mathcal{R})$ が有限ならば $\mathcal{R}$ は停止性をもつ。

依存対において重要な簡約対について定義する。

定義 26. 順序 $\succ$ と、擬順序 $\succsim$ の対 $(\succ, \succsim)$ は以下の条件を満たすとき、**簡約対**と呼ばれる。

- $\succ$ が整礎かつ代入について閉じている。
- $\succsim$ が文脈と代入について閉じている。
- $\succ \cdot \succsim \subseteq \succ$ または $\succsim \cdot \succ \subseteq \succ$ が成り立つ。

定理 27 ([8])**.** 簡約対 $(\succ, \succsim)$ が依存対問題 $(\mathcal{P}, \mathcal{R})$ に対し $\mathcal{P} \cup \mathcal{R} \subseteq \succsim$ を満たすとする。このとき、$(\mathcal{P}, \mathcal{R})$ の有限性と $(\mathcal{P} \setminus \succ, \mathcal{R})$ の有限性は同値である。

簡約対を作成する際に用いる引数切り落としを定義する。

定義 28. 引数切り落とし π (argument filtering, AF) は各 $f^{(n)} \in \mathcal{F}$ に対し以下の条件を満たす写像である:

- $\pi(f) \in \{1, \cdots, n\}$ または
- $\pi(f) \in \{[i_1, \cdots, i_m] \mid 1 \leqslant i_1 < \cdots < i_m \leqslant n\}$

t が変数の時 $\hat{\pi}(t) = t$ であり $t = f(t_1, \cdots, t_n)$ とする引数切り落とし $\hat{\pi}$ は以下のように項へ拡張される。

$$\hat{\pi}(t) = \begin{cases} \hat{\pi}(t_i) & \pi(f) = i \text{ のとき} \\ f(\hat{\pi}(t_{i_1}), \cdots, \hat{\pi}(t_{i_m})) & \pi(f) = [i_1, \cdots, i_m] \text{ のとき} \end{cases}$$

$\hat{\pi}(s)\ R\ \hat{\pi}(t)$ を $s\ R^{\pi}\ t$ と表記する。

定理 29. $(\succ, \succsim)$ が簡約対であるならば $(\succ^{\pi}, \succsim^{\pi})$ も簡約対である。

例 30. 項書換えシステム $\mathcal{R}_4$ を考えたとき引数切り落とし π を次のようにとる。

$$\pi(-) = \pi(\div) = \pi(-^{\sharp}) = \pi(\div^{\sharp}) = 1, \pi(\mathsf{0}) = [\,], \pi(s) = [1]$$

すると次の順序が得られる。

$$\begin{array}{ccc} \mathsf{0} \succsim \mathsf{0} & x \succsim x & \mathsf{s}(x) \succsim x \\ \mathsf{0} \succsim \mathsf{0} & \mathsf{s}(x) \succsim \mathsf{s}(x) & \\ \mathsf{s}(x) \succ x & \mathsf{s}(x) \succ x & \mathsf{s}(x) \succ x \end{array}$$

順序のホーア拡張について定義する。

定義 31. 集合 A 上の順序 $>$ 及び、擬順序 $\gtrsim$ を考える。集合 A 上のべき集合において、$>^{\mathsf{H}}$ 及び $\gtrsim^{\mathsf{H}}$ が次の条件を満たす関係を**ホーア拡張**と呼ぶ。

- $M \neq \varnothing$ 及び、任意の $y \in N$ に対して $x > y$ となる $x \in M$ が存在するとき $M >^{\mathsf{H}} N$ が成り立つ。
- 任意の $y \in N$ に対して $x \gtrsim y$ となる $x \in M$ が存在するとき $M \gtrsim^{\mathsf{H}} N$ が成り立つ。

もし $> \cdot \gtrsim \subseteq \gtrsim \cdot >$ または $> \cdot \gtrsim \subseteq \gtrsim \cdot >$ が成り立つならば、順序 $>$ 及び、擬順序 $\gtrsim$ は **互換性を持つ** と呼ぶ。

補題 32. $>$ と、 $\gtrsim$ を集合 A 上の順序および擬順序とすると、次の条件をそれぞれ満たす。

1. $>^{\mathsf{H}}$ は推移的かつ非反射的な順序である。
2. $\gtrsim^{\mathsf{H}}$ は擬順序である。
3. $>$ が無限下降列を持たなければ $>^{\mathsf{H}}$ は無限下降列を持たない。
4. $>$ と $\gtrsim$ が互換性を持つとき、$>^{\mathsf{H}}$ と $\gtrsim^{\mathsf{H}}$ もまた互換性を持つ。

第3章　合字化

3.1　長さ辞書式合字順序

まず長さ辞書式順序の定義を振り返る。Σ を有限のアルファベットとする。

定義 33. アルファベット Σ 上の擬順序 $\gtrsim$ を**優先順序**という。優先順序 $\gtrsim$ に対し、Σ^* 上の**長さ辞書式順序** (length-lexicographic order) $>_{\mathrm{ll}}$ は次の三つの規則によって帰納的に定義される。

$$\frac{|x| > |y|}{x >_{\mathrm{ll}} y} \qquad \frac{|ax| = |by| \quad a > b}{ax >_{\mathrm{ll}} by} \qquad \frac{|ax| = |by| \quad a \sim b \quad x >_{\mathrm{ll}} y}{ax >_{\mathrm{ll}} by}$$

但し、$> = \gtrsim \setminus \lesssim$ かつ $\sim = \gtrsim \cap \lesssim$ とする。同様に擬順序 $\gtrsim_{\mathrm{ll}}$ は次の４つの規則によって帰納的に定義される。

$$\frac{}{\epsilon \gtrsim_{\mathrm{ll}} \epsilon} \qquad \frac{|x| > |y|}{x \gtrsim_{\mathrm{ll}} y} \qquad \frac{|ax| = |by| \quad a > b}{ax \gtrsim_{\mathrm{ll}} by} \qquad \frac{|ax| = |by| \quad a \sim b \quad x \gtrsim_{\mathrm{ll}} y}{ax \gtrsim_{\mathrm{ll}} by}$$

長さ辞書式順序は簡約順序になる。その整礎性の証明に Higman の補題を用いる。

定義 34. 整列擬順序 (well-quasi-order) とはアルファベット Σ 上の順序 $\gtrsim$ に対して無限列 $a_1, a_2, \cdots$ が、$a_i \gtrsim a_j$ でも $a_j \gtrsim a_i$ でもないことを満たす $1 \lesssim i < j$ が存在しないことである。整列擬順序 $\gtrsim$ が半順序であるとき、部分整列順序と呼ぶ。

定理 35. 任意の整列擬順序に対し $>_{\mathrm{ll}}$ は簡約順序になる。

Higman の補題とは次のような補題である。

補題 36. $\gtrsim$ を整列擬順序とする。任意の Σ^* 上の無限列 $(s_0, s_1, s_2, \cdots)$ に対してある $i < j$ なる添字が存在し $s_i \lesssim_{\mathsf{emb}} s_j$ を満たす。

定理 37. $>_{\mathrm{ll}}$ は整礎である。

証明. 以下を満たすとき $>_{\mathrm{ll}}$ は整礎である。

1. $>_{\mathrm{ll}}$ が書換え順序である。
2. 全ての $a \in \Sigma$ に対して $a >_{\mathrm{ll}} \epsilon$ である。
3. $a \gtrsim b$ の時に $a >_{\mathrm{ll}} b$ が成立する。

これら $1-3$ を仮定し、また $>_{\mathrm{II}}$ は無限下降列が存在しないと仮定すると $s_0 >_{\mathrm{II}} s_1 >_{\mathrm{II}} s_2 >_{\mathrm{II}} \cdots$ の形の無限列が存在する。Higman の補題から、$i_0 \lesssim i_1 \lesssim i_2 \lesssim \cdots$ の関係をもつ $i_0, i_1, i_2, \cdots$ が存在し $s_{i_0} \lesssim_{\mathsf{emb}} s_{i_1} \lesssim_{\mathsf{emb}} s_{i_2} \lesssim_{\mathsf{emb}} \cdots$ から $s_{i_0} <_{\mathrm{II}} s_{i_1}$ または、$s_{i_0} \approx s_{i_1}$ かつ $s_{i_0} >_{\mathrm{II}} s_{i_1}$ のため、矛盾が生じる。 □

定義 38. $[as]_2 >_{\mathrm{II}} [at]_2$ であるときに限り $s >_{\mathrm{III}} t$ である。任意の文字 a はアルファベット Σ 上にある。

証明. 擬順序 $\gtrsim$ による関係を $aa \gtrsim ba$ かつ $bb \gtrsim ba$ とする。$aa >^{\mathrm{III}} ba$ は成り立つが $baa >^{\mathrm{III}} bba$ は成り立たない。 □

定義 39. $[a_n]_2$ は次の形に定義する。

$$[a_1a_2a_3\cdots a_n]_2 = \begin{cases} \epsilon & n \leqslant 1 \text{ のとき} \\ a_1a_2 \;\; a_2a_3 \;\; \cdots \;\; a_{n-1}a_n & n > 1 \text{ のとき} \end{cases}$$

補題 40. $>^{\mathrm{III}}$ は簡約順序である。

以下、$>$ を合字 Σ^2 上の整礎な優先順序とする。アルファベットが空でないとき、$>_{\mathrm{III}}$ が Σ^* 上の簡約順序になることを示す。まず $>_{\mathrm{III}}$ が整礎順序であることを示す。

補題 41. Σ が空でないとき $>_{\mathrm{III}}$ は非反射性を持つ。

証明. 条件より Σ の要素が存在する。これを a とする。$>_{\mathrm{II}}$ は非反射性をもつことから $[as]_2 >_{\mathrm{II}} [as]_2$ が成立せず、よって $s >_{\mathrm{III}} s$ は成立しない。 □

補題 42. $>_{\mathrm{III}}$ は推移的である。

証明. 条件より Σ の要素が存在する。これを a とし、また $s >_{\mathrm{III}} t >_{\mathrm{III}} u$ ならば $s >_{\mathrm{III}} u$ を仮定する。任意の $a \in \Sigma$ に対し $[as]_2 >_{\mathrm{II}} [au]_2$ である。$s >_{\mathrm{III}} t >_{\mathrm{III}} u$ ならば $s >_{\mathrm{III}} u$ から $[as]_2 >_{\mathrm{II}} [at]_2 >_{\mathrm{II}} [au]_2$ が $>_{\mathrm{II}}$ の推移性から $[as]_2 >_{\mathrm{II}} [au]_2$ が成り立つことにより $s >_{\mathrm{III}} u$ は成立する。 □

補題 43. Σ が空でないとき $>_{\mathrm{III}}$ は整礎である。

証明. 背理法を用いる。仮に $>_{\mathrm{III}}$ が無限下降列 $s_1 >_{\mathrm{III}} s_2 >_{\mathrm{III}} s_3 >_{\mathrm{III}} \cdots$ を持つとする。また、条件より Σ の要素が存在しこれを a とする。仮定より $[as_1]_2 >_{\mathrm{II}} [as_2]_2 >_{\mathrm{II}} [as_3]_2 >_{\mathrm{III}} \cdots$ が得られる。よって $>_{\mathrm{II}}$ が整礎であることに反し矛盾が生じる。 □

次に $>_{\mathrm{III}}$ が書換え関係であることを示す。

補題 44. $a \in \Sigma$ とする。 $s >_{\mathrm{III}} t$ ならば $as >_{\mathrm{III}} at$ である。

証明. $s >_{\text{III}} t$ を仮定すると任意の b に対して $[bs]_2 >_{\text{II}} [bt]_2$ が成り立つ。

$as >_{\text{III}} at$ は任意の b に対して $[bas]_2 >_{\text{II}} [bat]_2$ と表せる。両辺をそれぞれ展開すると、$ba[as]_2 >_{\text{II}} ba[at]_2$ となる。$[as]_2 >_{\text{II}} [at]_2$ が前提より成り立つことから $ba[as]_2 >_{\text{II}} ba[at]_2$ が成り立ちしたがって $as >_{\text{III}} at$ は成り立つ。 □

補題 45. $>_{\text{III}}$ は左の文脈について閉じている。

証明. $s >_{\text{III}} t$ とする。 文字列 u についての帰納法により $us >_{\text{III}} ut$ を示す。

1. $u = \epsilon$ のとき。(左辺) $= us = s$ かつ (右辺) $= ut = t$ より $s >_{\text{III}} t$ が成り立つとき $us >_{\text{III}} ut$ は成り立つ。

2. $u = au'$ かつ $a \in \Sigma$ のとき。(左辺) $= au's$ かつ (右辺) $= au't$ となる。$au's >_{\text{III}} au't$ は補題 44 より $u's >_{\text{III}} u't$ が成り立つ。

$1, 2$ より $us >_{\text{III}} ut$ が成り立つことが証明された。 □

補題 46. $|[s]_2| = \mathsf{max}\{|s| - 1,\ 0\}$ である。

補題 47. $a \in \Sigma$ とする。$[s]_2 >_{\text{II}} [t]_2$ ならば $[sc]_2 >_{\text{II}} [tc]_2$ である。

証明. $[s]_2 >_{\text{II}} [t]_2$ と仮定する。s についての帰納法により $[sc]_2 >_{\text{II}} [tc]_2$ を示す。

- $s = \epsilon$ または $s \in \Sigma$ のとき。$[s]_2 >_{\text{II}} [t]_2$ は成り立たない。つまり、前提が偽のため $[sc]_2 >_{\text{II}} [tc]_2$ は成り立つ。

- $s = aa's'$ かつ、 $t = \epsilon$ または $t \in \Sigma$ のとき。$[aa's']_2 >_{\text{II}} \epsilon$ は成り立つため $[sc]_2 >_{\text{II}} [tc]_2$ は成り立つ。

- $s = aa's'$ かつ $t = bb't'$ のとき次の一つでも成り立つことを示す。。

  - $|[s]_2| > |[t]_2|$ のとき。補題 46 より $|[s]_2| = |s| - 1 = |t| - 1 = |[t]_2|$ は成立する。

  - $|[s]_2| = |[t]_2|$、$aa' \approx bb'$、$[a's']_2 >_{\text{II}} [b't']$ これら三つ全てが成り立つとき。補題より $|[sc]_2| = |sc| - 1 = |tc| - 1 = |[tc]_2|$ は成立する。$aa' \approx bb'$ は先頭の文字のため成立する。$[a's']_2 >_{\text{II}} [b't']$ から帰納法により $[a's'c]_2 >_{\text{II}} [b't'c]$ は成立する。

  - $|[s]_2| = |[t]_2|$ かつ $aa' \gtrsim bb'$ のとき。上述より $|[s]_2| = |[t]_2|$ は成り立つ。$aa' \gtrsim bb'$ も成り立つ。

  よって、$[sc]_2 >_{\text{II}} [tc]_2$ は成り立つことが証明された。

□

補題 48. $>_{\text{III}}$ は右の文脈について閉じている。

証明. $s >_{\mathrm{III}} t$ とする。 文字列 u についての帰納法により $su >_{\mathrm{III}} tu$ を示す。

1. $u = \epsilon$ のとき。(左辺) $= su = s$ かつ (右辺) $= tu = t$ より $s >_{\mathrm{III}} t$ が成り立つとき $su >_{\mathrm{III}} tu$ は成り立つ。

2. $u = au'$ かつ $a \in \Sigma$ のとき。(左辺) $= sau'$ かつ (右辺) $= tau'$ となる。$sau' >_{\mathrm{III}} tau'$ は補題 47 より $su' >_{\mathrm{III}} tu'$ が成り立つ。

1, 2 より $us >_{\mathrm{III}} ut$ が成り立つことが証明された。 □

以上より前述の主張が結論される。

定理 49. $\Sigma \neq \varnothing$ とする。Σ^2 上の部分整列順序である優先順序 $\geqslant$ に対し $>_{\mathrm{III}}$ は Σ^* 上の簡約順序になる。

例 50. 長さ辞書式合字順序を用いて文字列書換えシステムの停止性証明を行う。$\mathcal{R}$ を次の書換え規則の集合とする。

$$\mathcal{R} = \left\{ \begin{array}{rcll} aaba & \to & abab & (1) \\ bbab & \to & baba & (2) \end{array} \right\}$$

$\gtrsim$ を擬順序とする関係 $aa \gtrsim ab$ と $bb \gtrsim ba$ が成り立つとする。次の順序の一つ目と二つ目は (1)、三つ目と四つ目は (2) の規則と対応している。

$$aa\ \underline{aa}\ ab\ ba >_{\mathrm{II}} aa\ \underline{ab}\ ba\ ab$$
$$ba\ \underline{aa}\ ab\ ba >_{\mathrm{II}} ba\ \underline{ab}\ ba\ ab$$
$$ab\ \underline{bb}\ ba\ ab >_{\mathrm{II}} ab\ \underline{ba}\ ab\ ba$$
$$bb\ \underline{bb}\ ba\ ab >_{\mathrm{II}} bb\ \underline{ba}\ ab\ ba$$

下線は $\gtrsim$ の関係が成り立つ場所である。それぞれ順序がつくことから $\mathcal{R} \subseteq >_{\mathrm{III}}$ となる。

例 51. 次の停止性を持たない文字列書換えシステム $\mathcal{R}$ を考える。

$$\mathcal{R} = \left\{ \begin{array}{rcl} a & \to & b \\ abb & \to & aaa \end{array} \right\}$$

$\gtrsim$ を順序とした関係を $ba \approx bb > aa \approx ab$ とする。$\mathcal{R}$ は停止性を持たない。なぜなら、$\mathcal{R}$ は次のような無限列を持つからである。

$$\underline{abb} \to_{\mathcal{R}} a\underline{a}a \to_{\mathcal{R}} ab\underline{a} \to_{\mathcal{R}} \underline{abb} \to_{\mathcal{R}} \cdots$$

下線は $\mathcal{R}$ の書換え規則を適用する部分である。この書換え列を合字化し順序付けると次の形となる。

$$ab\ bb \gtrsim_{\mathrm{II}} aa\ aa \gtrsim_{\mathrm{II}} ab\ aa \not\gtrsim_{\mathrm{II}} ab\ ba$$

最後の関係が不成立である。事実、逆向きの $ab\ ba \gtrsim_{\mathrm{II}} ab\ aa$ が成立する。つまり、$\gtrsim_{\mathrm{II}}$ が左の文脈について閉じていない。

3.2 引数切り落とし

依存対を使用する際の簡約対を設計する手法として、引数切り落とし法がある。Arts と Gisel は 2000 年に項書換えシステムにおいての引数切り落とし法を提案した [3]。引数切り落としは強力な停止性判定ツールであることが知られている TTT2 [13] や AProVE [7] にも採用されている。次に、文字列に対する引数切り落としについて記述する。

定義 52. アルファベット Σ に対する**引数切り落とし**とは Σ から $\{\epsilon, [\,], 1, [1]\}$ への写像を指す。引数切り落とし π が与えられたとき、文字列 s に対する引数切り落とし $\hat{\pi}(s)$ は次のように定義される。

$$\hat{\pi}(s) = \begin{cases} \epsilon & s = at \text{ と } \pi(a) = \epsilon \text{ または } s = \epsilon \text{ のとき} \\ a & s = at \text{ と } \pi(a) = [\,] \text{ のとき} \\ \hat{\pi}(t) & s = at \text{ と } \pi(a) = 1 \text{ のとき} \\ a\hat{\pi}(t) & s = at \text{ と } \pi(a) = [1] \text{ のとき} \end{cases}$$

文字列上の二項関係 R と文字列 $s = a_1 \cdots a_m$ と $t = b_1 \cdots b_n$ を考える。$\pi(b_1) \wedge \cdots \wedge \pi(b_n) \in \{1, [1]\}$ ならば $\pi(a_1) \wedge \cdots \wedge \pi(a_m) \in \{1, [1]\}$ が成立し、かつ $\hat{\pi}(s)\ R\ \hat{\pi}(t)$ ならば $s\ R^{\pi}\ t$ と書く。

$\pi(b_1) \wedge \cdots \wedge \pi(b_n) \in \{1, [1]\}$ ならば $\pi(a_1) \wedge \cdots \wedge \pi(a_m) \in \{1, [1]\}$ が成立するという条件は t において引数切り落としが行われると、s においても引数切り落としが行われることを意味する。

例 53. $\pi(b_1) \wedge \cdots \wedge \pi(b_n) \in \{1, [1]\}$ ならば $\pi(a_1) \wedge \cdots \pi(a_m) \in \{1, [1]\}$ が成立するという条件を外した引数切り落としを、停止性がない文字列書換えシステム $\mathcal{R}_2$ に用いる。

$$\mathcal{R}_2 = \left\{ \begin{array}{rcl} aa & \to & aba \\ b & \to & \epsilon \end{array} \right\}$$

依存対は次の集合となる。

$$\mathsf{DP}(\mathcal{R}_2) = \left\{ \begin{array}{rcl} a^\sharp a & \to & a^\sharp ba \\ a^\sharp a & \to & b^\sharp a \end{array} \right\}$$

引数切り落とし π を $\pi(a^\sharp) = \pi(a) = [1], \pi(b) = \pi(b^\sharp) = \epsilon$ とすると、$\mathcal{R}_2 \subseteq \gtrsim^{\pi}$ は次の順序関係に対応する。

$$\begin{array}{rr} \hat{\pi}(aa) = aa & \hat{\pi}(aba) = \epsilon \\ \hat{\pi}(b) = \epsilon & \hat{\pi}(\epsilon) = \epsilon \end{array}$$

$\mathsf{DP}(\mathcal{R}_2) \subseteq >^{\pi}$ は次の順序関係に対応する。

$$\begin{array}{ll} \hat{\pi}(a^\sharp a) = a^\sharp a & \hat{\pi}(a^\sharp ba) = \epsilon \\ \hat{\pi}(a^\sharp a) = a^\sharp a & \hat{\pi}(b^\sharp a) = \epsilon \end{array}$$

となり、順序が長さ辞書式順序でついてしまうが、この文字列書換えシステム $\mathcal{R}_2$ は停止性がない。つまり、条件を外すと停止性がない文字列書換えシステムも停止してしまう可能性が出るため条件を入れる。

簡約対 $(\succ, \succsim)$ に対し、$(\succ^\pi, \succsim^\pi)$ が簡約対になることを示す。以下、R を文字列上の関係、π を引数切り落としとする。まず R が (擬) 順序であるとき、R^π も (擬) 順序となることを示す。

補題 54. R が反射性を持つとき R^π も反射性を持つ。

証明. 任意の文字列 s に対し、R の反射性より $\hat{\pi}(s)\ R\ \hat{\pi}(s)$ が成立する。 □

補題 55. R が非反射性を持つとき R^π も非反射性を持つ。

証明. 任意の文字列 s に対し、R の非反射性より $(\hat{\pi}(s)\ R\ \hat{\pi}(s))$ は成立しない。 □

補題 56. R が推移的であるとき R^π も推移的である。

証明. $s\ R^\pi\ t\ R^\pi\ u$ とする。R^π の定義より $\hat{\pi}(s)\ R\ \hat{\pi}(t)\ R\ \hat{\pi}(u)$ である。$\mathcal{R}$ が推移的であることから $\hat{\pi}(s)\ R\ \hat{\pi}(u)$ が成り立つ。よって、$s\ R^\pi\ u$ は成立する。 □

補題 57. R が反射的かつ左の文脈について閉じているとき、R^π は左の文脈について閉じる。

証明. $s\ R^\pi\ t$ ならば $us\ R^\pi\ ut$ を u に関する帰納法で示す。

- $u = \epsilon$ のとき $us = s$ また $ut = t$ のため $s\ R^\pi\ t$ が成り立つため $us\ R^\pi\ ut$ は成り立つ。
- $u = au'$ かつ $\pi(a) = \epsilon$ のとき

  $$\hat{\pi}(us) = \epsilon = \hat{\pi}(ut)$$

  であるため $us\ R^\pi\ ut$ は成り立つ。
- $u = au'$ かつ $\pi(a) = 1$ のとき

  $$\hat{\pi}(au's) = \hat{\pi}(u's) \qquad \hat{\pi}(au't) = \hat{\pi}(u't)$$

  帰納法の仮定より $u's\ R^\pi\ u't$ であるため $\hat{\pi}(us)\ R\ \hat{\pi}(ut)$ は成り立つ。
- $u = au'$ かつ $\pi(a) = [\,]$ のとき

  $$\hat{\pi}(au's) = a \qquad \hat{\pi}(au't) = a$$

  R は反射的かつ左の文脈について閉じているため $\hat{\pi}(us)\ R\ \hat{\pi}(ut)$ は成り立つ。

- $u = au'$ かつ $\pi(a) = [1]$ のとき

$$\hat{\pi}(au's) = a\hat{\pi}(u's) \qquad \hat{\pi}(au't) = a\hat{\pi}(u't)$$

帰納法の仮定と R は左の文脈について閉じていることから

$$\begin{aligned} \hat{\pi}(au's) &= a\hat{\pi}(u's) & \hat{\pi}(au't) &= a\hat{\pi}(u't) \\ &= \hat{\pi}(us) & &= \hat{\pi}(ut) \end{aligned}$$

のため $\hat{\pi}(us)\ R\ \hat{\pi}(ut)$ は成り立つ。 □

次の補題において、文字列 s 中の全ての文字 a が $\pi(a) \in \{1, [1]\}$ を満たすとき $B(s)$ と書く。

補題 58. 任意の文字列 s, t に対し $\hat{\pi}(st) = \hat{\pi}(s)u$ が成立する。ここで u は以下のように与えられる。

$$u = \begin{cases} \hat{\pi}(t) & B(s) \text{ であるとき} \\ \epsilon & B(s) \text{ ではないとき} \end{cases}$$

証明. 文字列 s に関する帰納法による。

- $s = \epsilon$ のとき $B(s)$ であるため

$$\hat{\pi}(st) = u = \hat{\pi}(s)u$$

が成り立つ。よって、$\hat{\pi}(st) = \hat{\pi}(s)u$ と成り立つ。

- $s = as'$ かつ $\pi(a) = \epsilon$ のとき $B(s)$ ではないため

$$\hat{\pi}(st) = \epsilon = \hat{\pi}(s)u$$

が成り立つ。よって、$\hat{\pi}(st) = \hat{\pi}(s)u$ は成り立つ。

- $s = as'$ かつ $\pi(a) = 1$ のとき $B(s)$ であるため

$$\hat{\pi}(st) = \hat{\pi}(s't) = \hat{\pi}(s)u$$

が成り立つ。よって、帰納法の仮定より $\hat{\pi}(st) = \hat{\pi}(s)u$ は成り立つ。

- $s = as'$ かつ $\pi(a) = [\,]$ のとき $B(s)$ ではないため

$$\hat{\pi}(st) = a = \hat{\pi}(s)u$$

が成り立つ。よって、$\hat{\pi}(st) = \hat{\pi}(s)u$ は成り立つ。

- $s = as'$ かつ $\pi(a) = [1]$ のとき $B(s)$ であるため

$$\begin{aligned} \hat{\pi}(st) &= a(s't) & \hat{\pi}(s)u &= a(s'u) \\ &= a\hat{\pi}(s't) & &= a\hat{\pi}(s't) \end{aligned}$$

が成り立つ。よって、$\hat{\pi}(st) = \hat{\pi}(s)u$ は成り立つ。

これらから、$\hat{\pi}(st) = \hat{\pi}(s)u$ は成り立つ。 □

補題 59. R が右の文脈について閉じているならば R^π も右の文脈について閉じている。

証明. $s\ R^\pi\ t$ ならば $us\ R^\pi\ ut$ を u に関する帰納法で示す。

- $u = \epsilon$ のとき $su = s$ また $tu = t$ のため $s\ R^\pi\ t$ が成り立つため $su\ R^\pi\ tu$ は成り立つ。
- $u = au'$ かつ $\pi(a) = \epsilon$ のとき

  $$\hat{\pi}(sau') = \epsilon = \hat{\pi}(tau')$$

  であるため $su\ R^\pi\ tu$ は成り立つ。
- $u = au'$ かつ $\pi(a) = 1$ であり、補題 59 を用いたとき

  $$\hat{\pi}(sau') = \hat{\pi}(s)\hat{\pi}(u') \qquad \hat{\pi}(tau') = \hat{\pi}(t)\hat{\pi}(u')$$

  帰納法の仮定より $su'\ R^\pi\ tu'$ であるため $\hat{\pi}(su)\ R\ \hat{\pi}(tu)$ は成り立つ。
- $u = au'$ かつ $\pi(a) = [\,]$ のとき

  $$\hat{\pi}(sau') = \hat{\pi}(s)a \qquad \hat{\pi}(tau') = \hat{\pi}(t)a$$

  R は仮定より右の文脈について閉じているため $\hat{\pi}(su)\ R\ \hat{\pi}(tu)$ は成り立つ。
- $u = au'$ かつ $\pi(a) = [1]$ のとき

  $$\hat{\pi}(sau') = \hat{\pi}(s)\hat{\pi}(au') \qquad \hat{\pi}(tau') = \hat{\pi}(t)\hat{\pi}(au')$$

  帰納法の仮定と R は右の文脈について閉じていることから

  $$\begin{aligned}\hat{\pi}(sau') &= \hat{\pi}(s)\hat{\pi}(au') & \hat{\pi}(tau') &= \hat{\pi}(t)\hat{\pi}(au')\\ &= \hat{\pi}(su) & &= \hat{\pi}(tu)\end{aligned}$$

  のため $\hat{\pi}(us)\ R\ \hat{\pi}(ut)$ は成り立つ。 □

定理 60. 写像 π を引数切り落としとする。$(\succ, \gtrsim)$ が簡約対であるならば $(\succ^\pi, \gtrsim^\pi)$ も簡約対である。

。

例 61. 合字順序と引数切り落としを利用した停止性の証明を行う。$\mathcal{R}$ を次の書換え規則の集合とする。

$$\mathcal{R} = \{\ aa \quad \to \quad aba\ \}$$

依存対は次の集合となる。

$$\mathsf{DP}(\mathcal{R}) = \{\ a^\sharp a \quad \to \quad a^\sharp b a\ \}$$

引数切り落とし π を $\pi(a^\sharp) = \pi(a) = [1], \pi(b) = [\,]$ とする。また、$>$ を順序とした関係を $a^\sharp a > a^\sharp b > aa > ab$ とする。$\mathcal{R} \subseteq \gtrsim^\pi$ は次の順序関係に対応する。

$$\hat{\pi}(\text{左辺}) = aa \qquad \hat{\pi}(\text{右辺}) = ab$$

$\mathsf{DP}(\mathcal{R}) \subseteq >^\pi$ は次の順序関係に対応する。

$$\hat{\pi}(\text{左辺}) = a^\sharp a \qquad \hat{\pi}(\text{右辺}) = a^\sharp b$$

これらより $\mathcal{R} \subseteq \gtrsim_{\mathsf{II}}$ かつ $\mathsf{DP}(\mathcal{R}) \subseteq >_{\mathsf{II}}$ から $\mathcal{R}$ は停止性を持つ。

3.3 符号化

SMT ソルバーを用いて、適切な引数切り落とし π と優先順序 $>$ を発見するための依存対問題から制約式への変換手法を紹介する。ここでの制約式は線形算術と命題論理式からなるものである。より正確には次の文法で与えられる式である。p、n、x はそれぞれ命題変数、整数、整数上の変数である。

- $c ::= e = e \mid e \geqslant e \mid e > e \mid p \mid c \wedge c \mid c \vee c \mid \neg c$
- $e ::= n \mid e + e \mid x \mid \text{if } c \text{ then } e \text{ else } e$

π を引数切り落としとする。$A(a)$ が $\pi(a) \in \{[\,], [1]\}$ を、また $B(a)$ が $\pi(a) \in \{1, [1]\}$ を表すこととする。$\ell \gtrsim^\pi r$ の制約式への符号化において自明ではない箇所は

$$[\pi(a\ell)]_2 \gtrsim_{\mathsf{II}} [\pi(ar)]_2$$

の形の条件であるが、それは

$$\bigwedge_{a \in \Sigma} (A(a) \wedge B(a) \implies [a\pi(\ell)]_2 \gtrsim_{\mathsf{II}} [a\pi(r)]_2)$$

と同値である。上式の不等式は以下の補題を用いて再帰的に制約式に変換できる。$\ell >^\pi_{\mathsf{III}} r$ についても同様である。

補題 62. $\ell = a'\ell'$ と $r = b'r'$ とする。ただし $a, a', b, b' \in \Sigma$ である。このとき、次の性質が成立する。

- $[a\pi(\epsilon)]_2 \not\gtrsim_{\mathsf{II}} [b\pi(r)]_2$
- $[a\pi(\ell)]_2 >_{\mathsf{II}} [b\pi(b)]_2 \iff |[a\pi(\ell)]_2| > |[\pi(b)]_2|$

- $[a\pi(\ell)]_2 >_{\shortparallel} [b\pi(r)]_2 \iff |[\pi(a\ell)]_2| > |[\pi(br)]_2|$ または次の条件のうち一つと $|[\pi(a\ell)]_2| \geqslant |[\pi(br)]_2|$ が成立する。
  - $aa' \geqslant bb' \wedge A(a') \wedge A(b')$
  - $\neg A(a') \wedge B(a') \wedge [a\pi(\ell')]_2 >_{\shortparallel} [b\pi(r)]_2$
  - $\neg A(b') \wedge B(b') \wedge [a\pi(\ell)]_2 >_{\shortparallel} [b\pi(r')]_2$
  - $A(a') \wedge B(a') \wedge A(b') \wedge \neg B(b') \wedge aa' = bb' \wedge [\pi(a'\ell')]_2 >_{\shortparallel} [\pi(b'r')]_2$

さらに次の関係を満たす。

- $[a\pi(\ell)]_2 \geqslant_{\shortparallel} [b\pi(\epsilon)]_2$
- $[a\pi(\epsilon)]_2 \geqslant_{\shortparallel} [b\pi(r)]_2 \iff |[\pi(a)]_2| \geqslant |[\pi(br)]_2|$
- $[a\pi(\ell)]_2 \geqslant_{\shortparallel} [b\pi(r)]_2 \iff |[\pi(a\ell)]_2| > |[\pi(br)]_2|$ または次の条件のうち一つと $|[\pi(a\ell)]_2| \geqslant |[\pi(br)]_2|$ が成立する。
  - $aa' \geqslant bb' \wedge A(a') \wedge A(b')$
  - $\neg A(a') \wedge B(a') \wedge [a\pi(\ell')]_2 \geqslant_{\shortparallel} [b\pi(r)]_2$
  - $\neg A(b') \wedge B(b') \wedge [a\pi(\ell)]_2 \geqslant_{\shortparallel} [b\pi(r')]_2$
  - $A(a') \wedge B(a') \wedge A(b') \neg B(b') \wedge aa' = bb' \wedge [\pi(a'\ell')]_2 \geqslant_{\shortparallel} [\pi(b'r')]_2$

補題 63. 引数切り落としを行い合字化した際の文字列の長さの比較は次の式となる。

$$|[\pi(s)]_2| > |[\pi(t)]_2|$$

補題 63 は $A(a)$ と $B(a)$ を命題変数とみなせば、先述の線形算術に関する制約式になっている。ただし、$|[\pi(s)]_2| > |[\pi(t)]_2|$ については次の考察が必要である。補題 46 より

$$|[\pi(s)]_2| = \mathsf{max}\{|\pi(s) - 1|, 0\} > \mathsf{max}\{|\pi(t) - 1|, 0\} = |[\pi(t)]_2|$$

ここで

$$|[\pi(s)]| = \begin{cases} 0 & s = \epsilon \text{のとき} \\ \alpha + \beta & s = as' \text{のとき} \end{cases}$$

ただし α と β は次の値となる。

- $\alpha = A(a)$ のとき 1 それ以外は 0
- $\beta = B(a)$ のとき $|\pi(s')|$ それ以外は 0

3.4 関連研究

この節では合字化に関連した研究である ルートラベリング [17] や、freezing [19]、Arts と Giesl [3] の引数切り落としについて説明する。その際に文字列書換えシステムの

$$\mathcal{R} = \{\ aa \ \ \to \ \ aba\ \}$$

を用いる。合字による比較は次のようになる。

$$\left\{ \begin{array}{lll} aa\ aa & >_{\mathrm{ll}} & aa\ ab\ ba \\ ba\ aa & >_{\mathrm{ll}} & ba\ ab\ ba \end{array} \right\}$$

- ルートラベリング

  ルートラベリングと呼ばれる合字順序とよく似た手法を Middeldorp と Sternagel が 2007 年に提案した。この手法は全ての書換え規則の両辺の先頭に書換え規則に出てきた全ての文字をつけ、書換え規則のそれぞれの文字の後ろに後ろの文字の情報を付加する手法である。末尾には書換え規則に出てきた全ての文字の一つを付加する。文字列書換えシステム $\mathcal{R}$ を考えると次のような規則が作られる。

  $$\left\{ \begin{array}{lll} a_a a_a a_a & \to & a_a a_b b_a a_a \\ a_a a_a a_b & \to & a_a a_b b_a a_b \\ b_a a_a a_a & \to & b_a a_b b_a a_a \\ b_a a_a a_b & \to & b_a a_b b_a a_b \end{array} \right\}$$

  この書換え規則を見てわかるようにとても合字順序と似ている。合字順序と比較した際、rootlabeling は合字順序より強力であるが合字順序の倍の規則を作ることになる。

- unavoidable pattern と freezing

  freezing は Xi が 1998 に提案した。この手法は Puel の unavoidable pattern [16] から派生した手法の一つである。この手法は適当な隣り合った文字同士を一つにまとめる。文字列書換えシステム $\mathcal{R}$ を考えると次のような規則が作られる。

  $$\left\{ \begin{array}{lll} a\ a & \to & ab\ a \\ a\ ab & \to & ab\ ab \end{array} \right\}$$

  freezing は健全性のために補助規則が必要になる。Puel は再起経路順序を unavoidable pattern の手法を用いて部分文字列を優先順序の比較に用いられるように拡張した。合字順序と比較した際 freezing は二つの文字がひとつの合字に置き換わるが、合字順序では各文字がその両側の文字と合字化される。

- 引数切り落とし

  一般の引数切り落とし方法は定義 28 を、今回用いた引数切り落としについては定義 52 に記述している。この二つの違いは一般の引数切り落としに $\pi(a) = \epsilon$ の形は使われていないことである。これは項に空文字列に対応するものが存在するとは限らないからである。

第4章　文字列化

Ngo Thi Bao Tran [18] によって文字列化と呼ばれる項を文字列の集合に変換する方法が提案された。Ngo Thi Bao Tran はこの方法を用いて arctic interpretation [12] の SMT ソルバーを用いた実装方法を与えた。本章ではこのアイデアを発展させ、与えられた任意の文字列上の簡約対を項上の簡約対へ持ち上げる枠組みを構築する。

4.1　文字列化順序

文字列化では、文字列の集合に項を変換する。

定義 64. シグネチャ $\mathcal{F}$ 上の項 t を考える。項 t に対する **文字列化** は以下のように定義される：

$$\mathcal{S}(t) = \begin{cases} \{t\} & t \text{ が変数のとき} \\ \{f\} \cup \{f_i w \mid 1 \leqslant i \leqslant n \text{ かつ } w \in \mathcal{S}(t_i)\} & t = f(t_1, \ldots, t_n) \text{ のとき} \end{cases}$$

さらに, 2つの別の形も定義する。任意の変数 x に対して

$$\mathcal{S}_x(t) = \{w \mid wx \in \mathcal{S}(t)\} \qquad \mathcal{S}_{\mathcal{F}}(t) = \{wf \mid wf \in \mathcal{S}(t) \text{ かつ } f \in \mathcal{F}\}$$

と定める。

定義 65. 文字列上で順序 $>$ と、擬順序 $\gtrsim$ を考える。項上の関係 $>^{\mathcal{S}}$ と $\gtrsim^{\mathcal{S}}$ を以下のように定義する。

- $\mathcal{S}_{\mathcal{F}}(s) >^{\mathsf{H}} \mathcal{S}_{\mathcal{F}}(t)$ と任意の変数 $x \in \mathcal{V}\mathsf{ar}(t)$ に対して $\mathcal{S}_x(s) >^{\mathsf{H}} \mathcal{S}_x(t)$ が成立するとき $s >^{\mathcal{S}} t$ とする。
- $\mathcal{S}_{\mathcal{F}}(s) \gtrsim^{\mathsf{H}} \mathcal{S}_{\mathcal{F}}(t)$ と任意の変数 $x \in \mathcal{V}\mathsf{ar}(t)$ に対して $\mathcal{S}_x(s) \gtrsim^{\mathsf{H}} \mathcal{S}_x(t)$ が成立するとき $s \gtrsim^{\mathcal{S}} t$ とする。

$>^{\mathcal{S}}$ と $\gtrsim^{\mathcal{S}}$ を文字列化順序と呼ぶ。次節で証明を行うが簡約対 $(>, \gtrsim)$ に対して $(>^{\mathcal{S}}, \gtrsim^{\mathcal{S}})$ が簡約対になる。

例 66. 次の項書換えシステム $\mathcal{R}$ に文字列化を行い長さ辞書式順序を用いて停止性を証明する。

$$\begin{aligned} 1\colon &\quad \mathsf{len}(\mathsf{nil}) \to 0 \\ 2\colon &\quad \mathsf{len}(\mathsf{cons}(x,y)) \to \mathsf{s}(\mathsf{len}(y)) \end{aligned}$$

$\mathcal{R}$ の依存対は次の集合となる。

$$3\colon \quad \mathsf{len}^\sharp(\mathsf{cons}(x,y)) \to \mathsf{len}^\sharp(y)$$

$>^\pi_{\mathsf{lll}}$ と $\geqslant^\pi_{\mathsf{lll}}$ を $\succ$ と $\succsim$ と略記する。

$$\begin{aligned} 1\colon &\quad \left\{\begin{array}{c}\mathsf{len}\\ \mathsf{len}_1\mathsf{nil}\end{array}\right\} \succsim^{\mathsf{H}} \varnothing \\ 2\colon &\left\{\begin{array}{c}\mathsf{len}\\ \mathsf{len}_1\mathsf{cons}\end{array}\right\} \succsim^{\mathsf{H}} \left\{\begin{array}{c}\mathsf{s}\\ \mathsf{s}_1\mathsf{len}\end{array}\right\} \quad \{\mathsf{len}_1\mathsf{cons}_2\} \succsim^{\mathsf{H}} \{\mathsf{s}_1\mathsf{len}_1\} \\ 3\colon &\left\{\begin{array}{c}\mathsf{len}^\sharp\\ \mathsf{len}^\sharp_1\mathsf{cons}\end{array}\right\} \succ^{\mathsf{H}} \{\mathsf{len}^\sharp\} \qquad \{\mathsf{len}^\sharp_1\mathsf{cons}_2\} \succ^{\mathsf{H}} \{\mathsf{len}^\sharp_1\} \end{aligned}$$

$\mathsf{len} > \mathsf{s}_1$ とすると上記の全ての順序が向き付くため、この TRS は停止性を持つ。

例 67. 次の項書換えシステム $\mathcal{R}$ に文字列化を行い長さ辞書式順序を用いて停止性を持つか検証する。

$$\begin{aligned} 1\colon &\quad \mathsf{plus}(0,x) \to x \\ 2\colon &\quad \mathsf{plus}(\mathsf{s}(x),y) \to \mathsf{s}(\mathsf{plus}(x,y)) \end{aligned}$$

項書換えシステムの依存対は次の集合となる。

$$3\colon \quad \mathsf{plus}^\sharp(\mathsf{s}(x),y) \to \mathsf{plus}^\sharp(x,y)$$

$>^\pi_{\mathsf{lll}}$ と $\geqslant^\pi_{\mathsf{lll}}$ を $\succ$ と $\succsim$ と略記する。

$$\begin{aligned} 1\colon &\left\{\begin{array}{c}\mathsf{plus}\\ \mathsf{plus}_1\mathsf{0}\end{array}\right\} \succsim^{\mathsf{H}} \varnothing \qquad\qquad \{\mathsf{plus}_2\} \succsim^{\mathsf{H}} \varnothing \\ 2\colon &\left\{\begin{array}{c}\mathsf{plus}\\ \mathsf{plus}_1\mathsf{s}\end{array}\right\} \succsim^{\mathsf{H}} \left\{\begin{array}{c}\mathsf{s}\\ \mathsf{s}_1\mathsf{plus}\end{array}\right\} \quad \{\mathsf{plus}_1\mathsf{s}_1\} \succsim^{\mathsf{H}} \{\mathsf{s}_1\mathsf{plus}_2\} \quad \{\mathsf{plus}_2\} \succsim^{\mathsf{H}} \{\mathsf{s}_1\mathsf{plus}_2\} \\ 3\colon &\left\{\begin{array}{c}\mathsf{plus}^\sharp_1\\ \mathsf{plus}^\sharp_1\mathsf{s}\end{array}\right\} \succ^{\mathsf{H}} \{\mathsf{plus}^\sharp\} \qquad \{\mathsf{plus}^\sharp_1\mathsf{s}_1\} \succ^{\mathsf{H}} \{\mathsf{plus}^\sharp_1\} \qquad \{\mathsf{plus}^\sharp_2\} \succ^{\mathsf{H}} \{\mathsf{plus}^\sharp_2\} \end{aligned}$$

規則 3 において左辺が右辺と比べて長くはない。引数切り落としを用いても左辺が長くはならない。同じ変数が両辺で同じ位置に出現する。一般に、$f(\cdots x_i \cdots) \to f(\cdots x_i \cdots)$ のような規則を文字列化順序で向き付けることができない。そのため、引数切り落としを用いた長さ辞書式順序では、この TRS は停止性を持つことは証明できない。

4.2 文字列順序の証明

前節で述べたように、任意の文字列上の簡約順序 $>$ 及び 書換え擬順序 $\gtrsim$ に対して、$(>^{\mathcal{S}}, \gtrsim^{\mathcal{S}})$ が項上の簡約対となることを証明する。

補題 68. 次の等式が成り立つ。

(1) $\mathcal{S}_{\mathcal{F}}(t\sigma) = \mathcal{S}_{\mathcal{F}}(t) \cup \{uv \mid x \in \mathcal{V}$ が存在し $u \in \mathcal{S}_x(t)$ かつ $v \in \mathcal{S}_{\mathcal{F}}(x\sigma)\}$

(2) $\mathcal{S}_x(t\sigma) = \{uv \mid y \in \mathcal{V}$ が存在し $u \in \mathcal{S}_y(t)$ かつ $v \in \mathcal{S}_x(y\sigma)\}$

証明. (1) t の帰納法によって証明する。

- t が変数の時

$$\begin{aligned}(\text{右辺}) &= \mathcal{S}_{\mathcal{F}}(t) \cup \{uv \mid x \in \mathcal{V} \text{ が存在し } u \in \mathcal{S}_x(t) \text{ かつ } v \in \mathcal{S}_{\mathcal{F}}(x\sigma)\} \\ &= \varnothing \cup \mathcal{S}_{\mathcal{F}}(t\sigma) \\ &= (\text{左辺})\end{aligned}$$

- $t = f(t_1, \ldots, t_n)$ の時

$$\begin{aligned}(\text{左辺}) &= \{f\} \cup \{f_i w \mid \ w \in \mathcal{S}_{\mathcal{F}}(t_i\sigma)\} \\ &= \{f\} \cup \{f_i w \mid \ w \in \mathcal{S}_{\mathcal{F}}(t_i)\} \cup \\ &\quad \{f_i uv \mid \text{ある } x \in \mathcal{V} \text{ に対して } u \in \mathcal{S}_x(t_i) \text{ かつ } v \in \mathcal{S}_{\mathcal{F}}(x\sigma)\} \\ &= \mathcal{S}_{\mathcal{F}}(t) \cup \\ &\quad \{uv \mid \text{ある } x \in \mathcal{V} \text{ に対して } u \in \mathcal{S}_x(t) \text{ かつ } v \in \mathcal{S}_{\mathcal{F}}(x\sigma)\} \\ &= (\text{右辺})\end{aligned}$$

$1 \leqslant i \leqslant n$ である。ここで 一つ目の等式は t_i に対する帰納法の仮定による。

(2) t の帰納法によって証明する。

- t が変数の時

$$\begin{aligned}(\text{右辺}) &= \{uv \mid y \in \mathcal{V} \text{ が存在し } u \in \mathcal{S}_y(t) \text{ かつ } v \in \mathcal{S}_x(y\sigma)\} \\ &= \mathcal{S}_x(t\sigma) \\ &= (\text{左辺})\end{aligned}$$

- $t = f(t_1, \cdots, t_n)$ の時

$$\begin{aligned}(\text{左辺}) &= \{f_i w \mid \ w \in \mathcal{S}_x(t_i\sigma)\} \\ &= \{f_i uv \mid \text{ある } y \in \mathcal{V} \text{ に対して } u \in \mathcal{S}_y(t_i) \text{ かつ } v \in \mathcal{S}_x(y\sigma)\} \\ &= \{uv \mid \text{ある } y \in \mathcal{V} \text{ に対して } u \in \mathcal{S}_y(t) \text{ かつ } v \in \mathcal{S}_x(y\sigma)\} \\ &= (\text{右辺})\end{aligned}$$

$1 \leqslant i \leqslant n$ である。ここで一つ目の等式は t_i に対する帰納法の仮定による。

□

補題 69. 以下の等式は成り立つ。

(1) $\mathcal{S}_{\mathcal{F}}(C[t]) = \mathcal{S}_{\mathcal{F}}(C) \cup \{uv \mid \text{ある } x \in \mathcal{V} \text{ に対し } u \in \mathcal{S}_x(C) \text{ かつ } v \in \mathcal{S}_{\mathcal{F}}(t)\}$

(2) $\mathcal{S}_x(C[t]) = \mathcal{S}_x(C) \cup \{uv \mid u \in \mathcal{S}_x(C) \text{ かつ } v \in \mathcal{S}_x(t)\ \}$

証明. (1) を C の帰納法によって証明する。

- $C = \Box$ の時

$$\begin{aligned}(\text{右辺}) &= \mathcal{S}_{\mathcal{F}}(C) \cup \{uv \mid \text{ある } x \in \mathcal{V} \text{ に対し } u \in \mathcal{S}_x(C) \text{ かつ } v \in \mathcal{S}_{\mathcal{F}}(t)\} \\ &= (\text{左辺})\end{aligned}$$

- $C = f(t_1, \cdots, C', \cdots t_n)$ ただし C' は文脈

$$\begin{aligned}(\text{左辺}) &= \{f\} \cup \{f_i w \mid 1 \leqslant i \leqslant n \text{ かつ } w \in \mathcal{S}_{\mathcal{F}}(C'[t])\} \\ &= \{f\} \cup \{f_i w \mid 1 \leqslant i \leqslant n \text{ かつ } w \in \mathcal{S}_{\mathcal{F}}(C')\} \cup \\ &\quad \{f_i uv \mid 1 \leqslant i \leqslant n \text{ かつ } \exists x \in \mathcal{V}.\ u \in \mathcal{S}_x(C'[t]) \text{ かつ } v \in \mathcal{S}_{\mathcal{F}}(t)\} \\ &= \mathcal{S}_{\mathcal{F}}(t) \cup \{uv \mid \exists x \in \mathcal{V}.\ u \in \mathcal{S}_x(C) \text{ かつ } v \in \mathcal{S}_{\mathcal{F}}(t)\} \\ &= (\text{右辺})\end{aligned}$$

ここで一つ目の等式は C' に対する帰納法の仮定による。 □

証明. (2) を C の帰納法によって証明する。

- $C = \Box$ の時

$$\begin{aligned}(\text{右辺}) &= \mathcal{S}_x(C) \cup \{uv \mid u \in \mathcal{S}_x(C) \text{ かつ } v \in \mathcal{S}_x(t)\} \\ &= (\text{左辺})\end{aligned}$$

- $C = f(t_1, \cdots, C', \cdots t_n)$ ただし C' は文脈

$$\begin{aligned}(\text{左辺}) &= \{f_i w \mid 1 \leqslant i \leqslant n \text{ かつ } w \in \mathcal{S}_x(C'[t])\} \\ &= \{f_i w \mid 1 \leqslant i \leqslant n \text{ かつ } w \in \mathcal{S}_x(C')\} \cup \\ &\quad \{f_i uv \mid 1 \leqslant i \leqslant n \text{ かつ } u \in \mathcal{S}_x(C'[t]) \text{ かつ } v \in \mathcal{S}_x(t)\} \\ &= \{uv \mid\ u \in \mathcal{S}_x(C) \text{ かつ } v \in \mathcal{S}_x(t)\} \\ &= (\text{右辺})\end{aligned}$$

ここで一つ目の等式は C' に対する帰納法の仮定による。 □

補題 70. $\mathcal{S}_x(t) \neq \varnothing$ が成り立つとき $x \in \mathcal{V}\mathsf{ar}(t)$ も成り立つ

次に、 $>^{\mathcal{S}}$ 及び $\gtrsim^{\mathcal{S}}$ の性質について調査する。

補題 71. $s >^{\mathcal{S}} t$ または $s \gtrsim^{\mathcal{S}} t$ ならば $\mathcal{V}\mathsf{ar}(s) \supseteq \mathcal{V}\mathsf{ar}(t)$ である。

証明. まず $s >^{\mathcal{S}} t$ ならば $\mathcal{V}\mathsf{ar}(s) \supseteq \mathcal{V}\mathsf{ar}(t)$ を証明する。$s >^{\mathcal{S}} t$ と仮定する。次に任意の $x \in \mathcal{V}\mathsf{ar}(t)$ を考えると補題 70 より、$\mathcal{S}_x(t) \neq \varnothing$ が成り立つ。$x \in \mathcal{V}\mathsf{ar}(t)$ と $s >^{\mathcal{S}} t$ の定義より、$\mathcal{S}_x(s) >^{\mathsf{H}} \mathcal{S}_x(t)$ となる。$\mathcal{S}_x(t) \neq \varnothing$ と $>^{\mathsf{H}}$ の定義より $\mathcal{S}_x(s) \neq \varnothing$ が導出される。補題 70 より、$s \gtrsim^{\mathcal{S}} t$ の場合も同様に導かれる。 □

補題 72. 関係 $>^{\mathcal{S}}$ と $\gtrsim^{\mathcal{S}}$ はそれぞれ順序、擬順序である。

証明. まず関係 $>^{\mathcal{S}}$ が順序であることを示す。

- 推移性を示す。$s >^{\mathcal{S}} t >^{\mathcal{S}} u$ より、$\mathcal{S}_{\mathcal{F}}(s) >^{\mathsf{H}} \mathcal{S}_{\mathcal{F}}(t) >^{\mathsf{H}} \mathcal{S}_{\mathcal{F}}(u)$ かつ全ての $x \in \mathcal{V}\mathsf{ar}(t)$ に対して $\mathcal{S}_x(s) >^{\mathsf{H}} \mathcal{S}_x(t)$ また全ての $x \in \mathcal{V}\mathsf{ar}(u)$ に対して $\mathcal{S}_x(t) >^{\mathsf{H}} \mathcal{S}_x(u)$ が成り立つ。補題 71 より、$t >^{\mathcal{S}} u$ が成り立つことから $\mathcal{S}_{\mathcal{F}}(s) >^{\mathsf{H}} \mathcal{S}_{\mathcal{F}}(t) >^{\mathsf{H}} \mathcal{S}_{\mathcal{F}}(u)$ かつ全ての $x \in \mathcal{V}\mathsf{ar}(u)$ に対して $\mathcal{S}_x(s) >^{\mathsf{H}} \mathcal{S}_x(t) >^{\mathsf{H}} \mathcal{S}_x(u)$ となる。補題 32 より $>^{\mathsf{H}}$ は推移性を持つため $>^{\mathcal{S}}$ は推移性を持つ。
- 非反射的であることを示す。t を任意の項とする。仮定より $>$ は非反射である。ゆえに補題 32 から $>^{\mathsf{H}}$ も非反射である。$\mathcal{S}_{\mathcal{F}}(t) >^{\mathsf{H}} \mathcal{S}_{\mathcal{F}}(t)$ は成立しない。つまり $>^{\mathcal{S}}$ は非反射的となる。

次に $\gtrsim^{\mathcal{S}}$ が擬順序であることを示す。順序は推移的かつ反射的であるからその両方の性質を持つことを示す。推移的であることは $>^{\mathcal{S}}$ と同様に導かれるため反射的であることをここで示す。

- t を任意の項とする。仮定より $\gtrsim$ は反射的である。ゆえに補題 32 から $\gtrsim^{\mathsf{H}}$ も反射的である。よって、 $\mathcal{S}_{\mathcal{F}}(t) \gtrsim^{\mathsf{H}} \mathcal{S}_{\mathcal{F}}(t)$ と $\mathcal{S}_x(s) \gtrsim^{\mathsf{H}} \mathcal{S}_x(t)$ は成り立つ。よって、$\gtrsim^{\mathcal{S}}$ は反射的である。

□

補題 73. $>$ が文字列上の右の文脈に閉じた順序ならば $>^{\mathcal{S}}$ は項上の代入に閉じた順序である。

証明. $>^{\mathcal{S}}$ が代入に閉じていることを示す。$s >^{\mathcal{S}} t$ を仮定する。これより、$\mathcal{S}_{\mathcal{F}}(s\sigma) >^{\mathsf{H}} \mathcal{S}_{\mathcal{F}}(t\sigma)$ と任意の変数 $x \in \mathcal{V}\mathsf{ar}(t\sigma)$ に対して $\mathcal{S}_x(s\sigma) >^{\mathsf{H}} \mathcal{S}_x(t\sigma)$ が成立する。$\mathcal{S}_{\mathcal{F}}(s\sigma) >^{\mathsf{H}} \mathcal{S}_{\mathcal{F}}(t\sigma)$ の定義を展開すると

$$\mathcal{S}_{\mathcal{F}}(s) \cup \{uv \mid x \in \mathcal{V} \text{ が存在し } u \in \mathcal{S}_x(s) \text{ かつ } v \in \mathcal{S}_{\mathcal{F}}(x\sigma)\}$$
$$>^{\mathsf{H}} \mathcal{S}_{\mathcal{F}}(t) \cup \{uv \mid x \in \mathcal{V} \text{ が存在し } u \in \mathcal{S}_x(t) \text{ かつ } v \in \mathcal{S}_{\mathcal{F}}(x\sigma)\}$$

となる。これを証明する。w を右辺の要素とする。

- $w \in \mathcal{S}_{\mathcal{F}}(t)$ のとき。仮定より $\mathcal{S}_{\mathcal{F}}(s) >^{\mathsf{H}} \mathcal{S}_{\mathcal{F}}(t)$ であるため成り立つ。
- $w = uv$ かつ変数 $x \in \mathcal{V}$ が存在し $u \in \mathcal{S}_x(t)$ かつ $v \in \mathcal{S}_{\mathcal{F}}(x\sigma)$ のとき仮定より $\mathcal{S}_{\mathcal{F}}(s) >^{\mathsf{H}} \mathcal{S}_{\mathcal{F}}(t)$ であるため、ある $v' \in \mathcal{S}_{\mathcal{F}}(s)$ に対し $v' > v$ である。よって、$>$ は右の文脈について閉じているため $uv' > uv$ である。$uv' \in$ (左辺) であるため成り立つ。

次に変数 x を $\mathcal{V}\mathsf{ar}(t\sigma)$ の要素とする。$\mathcal{S}_x(s\sigma) >^{\mathsf{H}} \mathcal{S}_x(t\sigma)$ は

$$\{uv \mid y \in \mathcal{V} \text{ が存在し } u \in \mathcal{S}_y(s) \text{ かつ } v \in \mathcal{S}_x(y\sigma)\}$$
$$>^{\mathsf{H}} \{uv \mid y \in \mathcal{V} \text{ が存在し } u \in \mathcal{S}_y(t) \text{ かつ } v \in \mathcal{S}_x(y\sigma)\}$$

と展開しこれを証明する。w を右辺の要素とする。$w = uv$ かつ変数 $y \in \mathcal{V}$ が存在し $u \in \mathcal{S}_y(t)$ かつ $v \in \mathcal{S}_x(y\sigma)$ のとき仮定より $\mathcal{S}_x(s\sigma) >^{\mathsf{H}} \mathcal{S}_x(t\sigma)$ なため、ある $v' \in \mathcal{S}_x(s)$ に対し $v' > v$ である。よって、$>$ は右の代入について閉じているため $uv' > uv$ である。$uv' \in$ (左辺) のため成り立つ。$>^{\mathcal{S}}$ が代入について閉じていることが判明したため先述の主張が導かれる。 □

補題 74. $\gtrsim$ が文字列上の書換え擬順序ならば $\gtrsim^{\mathcal{S}}$ は項上の書換え擬順序である。

証明. 補題 73 と同じように $\gtrsim^{\mathcal{S}}$ は代入について閉じていることが導かれる。次に $\gtrsim^{\mathcal{S}}$ が文脈が閉じていることを示す。$s \gtrsim^{\mathcal{S}} t$ を仮定し $C[s] \gtrsim^{\mathcal{S}} C[t]$ が成り立つとする。これより、$\mathcal{S}_{\mathcal{F}}(C[s]) \gtrsim^{\mathsf{H}} \mathcal{S}_{\mathcal{F}}(C[t])$ と任意の変数 $x \in \mathcal{V}\mathsf{ar}(C[t])$ に対して $\mathcal{S}_x(C[s]) \gtrsim^{\mathsf{H}} \mathcal{S}_x(C[t])$ が成立する。$\mathcal{S}_{\mathcal{F}}(C[s]) \gtrsim^{\mathsf{H}} \mathcal{S}_{\mathcal{F}}(C[t])$ を

$$\mathcal{S}_{\mathcal{F}}(C) \cup \{uv \mid x \in \mathcal{V} \text{ が存在し } u \in \mathcal{S}_x(C) \text{ かつ } v \in \mathcal{S}_{\mathcal{F}}(s)\}$$
$$>^{\mathsf{H}} \mathcal{S}_{\mathcal{F}}(C) \cup \{uv \mid x \in \mathcal{V} \text{ が存在し } u \in \mathcal{S}_x(C) \text{ かつ } v \in \mathcal{S}_{\mathcal{F}}(t)\}$$

と展開しこれを証明する。w を右辺の要素とする。

- $w \in \mathcal{S}_{\mathcal{F}}(t)$ のとき。w は左辺に属すことは明らかである。
- $w = uv$ かつ変数 $x \in \mathcal{V}$ が存在し $u \in \mathcal{S}_x(C)$ かつ $v \in \mathcal{S}_{\mathcal{F}}(t)$ のとき仮定より $\mathcal{S}_{\mathcal{F}}(s) \gtrsim^{\mathsf{H}} \mathcal{S}_{\mathcal{F}}(t)$ なため、ある $v' \in \mathcal{S}_{\mathcal{F}}(s)$ に対し $v' \gtrsim v$ である。よって、$\gtrsim$ は左の文脈について閉じているため $uv' \gtrsim uv$ である。$uv' \in$ (左辺) のため成り立つ。

次に変数 x を $\mathcal{V}\mathsf{ar}(C[t])$ の要素とする。$\mathcal{S}_x(C[s]) \gtrsim^{\mathsf{H}} \mathcal{S}_x(C[t])$ は

$$\mathcal{S}_x(C) \cup \{uv \mid u \in \mathcal{S}_x(C) \text{ かつ } v \in \mathcal{S}_x(s)\}$$
$$>^{\mathsf{H}} \mathcal{S}_x(C) \cup \{uv \mid u \in \mathcal{S}_x(C) \text{ かつ } v \in \mathcal{S}_x(t)\}$$

と展開しこれを証明する。w を右辺の要素とする。

- $w \in \mathcal{S}_x(t)$ のとき。w は左辺に属すことは明らかである。
- $w = uv$ と $u \in \mathcal{S}_x(C)$ かつ $v \in \mathcal{S}_x(t)$ のとき仮定より $\mathcal{S}_x(s) \gtrsim^{\mathsf{H}} \mathcal{S}_x(t)$ なため、ある $v' \in \mathcal{S}_x(s)$ に対し $v' \gtrsim v$ である。よって、$\gtrsim$ は左の文脈について閉じているため $uv' \gtrsim uv$ である。$uv' \in$ (左辺) のため成り立つ。

$\gtrsim^{\mathcal{S}}$ が代入と文脈について閉じていることが判明したため先述の主張が導かれる。 □

定理 75. 簡約対 $(>, \gtrsim)$ に対して $(>^{\mathcal{S}}, \gtrsim^{\mathcal{S}})$ が簡約対になる。

4.3 関連研究

文字列化順序の関連研究について述べる。

- 文字列化

  文字列化は Ngo Thi Bao Tran が提案した [18]。行列順序と arctic interpretation [12] による停止性解析の自動化を行うために導入された。本来の文字列化の狙いは行列順序と arctic interpretation を項書換え上でコンパクトな制約式への変換を与えるためであった。arctic interpretation は以下のような定義となる。

  $$f_A(x_1, \ldots, x_n) = max\{x_1 + f_1, \ldots, x_n + f_1, f_0\}$$

  ここで $f_0, \cdots f_n$ は自然数である。arctic interpretation は項書換えを文字列化し文字列化された文字に重みを与え両辺の大きさを比べる手法である。w を文字から自然数への写像とする。文字列上の重み付け順序 $>_w$ は次のように定義される。

  $$s >_w t \text{ と} \hat{w}(s) > \hat{w}(t)$$

  ここで $w(a_1 \cdots a_n)$ は $w(a_1) + \cdots + w(a_n)$ で定義される。$>_A$ を arctic interpretation から導出される解釈順序とすると

  $$\ell >_A r \iff \ell(>_{\mathsf{w}})^{\mathcal{S}} r$$

  が成立する。例 66 の項書換えシステムを Arctic interpretation を用いて停止性を証明する。重さはそれぞれ次のようになる。$w(\mathsf{nil}) = 0, w(\mathsf{len}) = w(\mathsf{len}_1^\sharp) = w(\mathsf{len}^\sharp) = w(\mathsf{len}_1) = w(\mathsf{s}) = w(\mathsf{s}_1) = w(\mathsf{m}_2) = 1$ かつ $w(\mathsf{cons}_2) = w(\mathsf{cons}) = 2$ とする。この重みによって順序がついたため停止性が証明できる。

  文字列化は停止性解析の自動化を行うために導入されたが、私達は文字列書換えの順序付けをそのまま項書換えの順序付けする方法として使用した。

- 辞書式経路順序

  辞書式経路順序は文字列書換え上では勝ち抜き順序と呼ばれ、長さ辞書式合字順序と似ている簡約順序である。文字列化によりこの勝ち抜き順序を項上で比較しても長さ辞書式合字順序とは停止性の領域が異なる。勝ち抜き順序と長さ辞書式合字順序の簡約順序において停止性を判定できる領域が異なるからである。

第5章　実験評価

5.1　実験

ここまで議論した合字化と文字列化の実装および実験を行った。実装には関数型プログラミング言語 OCaml を用い、長さ辞書式合字順序は約 200 行、長さ辞書式合字順序を文字列化した順序は長さ辞書式合字順序のプラグラムに約 20 行加えたコードからなる。合字化、文字列化どちらも停止性の条件を満たしているかは SMT ソルバー である Yices バージョン 2.4.2 を用いて判断している。実験には Termination Problem Data Base (TPDB) バージョン 10.3 [1] の停止性の問題集を用いた。文字列書換えシステム、項書換えシステムの問題はそれぞれ 1315 問、1498 問である。実験には Intel Core i7 − 2640M 2.80GHz プロセッサおよび $8G$ バイトのメモリを搭載した PC を使用した。今回一つの問題に対する最大の計算時間は 60 秒とした。そのため、表のタイムアウトは 60 秒以内に停止性を判定できなかった個数であり、表の証明成功は時間以内に判定できた問題数である。計算時間は全ての問題を判定するためにかけた時間である。表 5.1 − 5.4 には長さ簡約順序である長さ辞書式順序 (ll)、長さ辞書式合字順序 (lll)、また比較のため簡約順序として広く用いられている辞書式経路順序 (LPO) [10]、Knuth-Bendix 順序 (KBO) [11]、行列順序 [9] さらに TTT2 ver. 1.15 [13]、AProVE [7]、Mint ver. 1.5 [18]、NaTT ver. 1.3 [20] という停止性判定ツールの結果を併せて記載した。TTT2 と AProVE は文字列書換えシステムと項書換えシステムに対応している。NATT と Mint は項書換えシステムに対応している。辞書式経路順序は関数記号の優先順序によって項上の順序を決定する手法である。適当な優先順序を用いることで書き換えシステムの停止性を判定できる。Knuth-Bendix 順序は関数記号の優先順序と、関数記号と変数に重みを適当に割り当てることで順序を決定する手法である。行列順序は両辺の関数記号と変数にそれぞれ行列を適当につけることで順序を決定する手法である。TTT2、AProVE、NATT は停止性判定ツールであり、辞書式経路順序、Knuth-Bendix 順序、行列順序や依存対法等の様々な手法を基に作られている。

文字列書換えシステムの TPDB には全ての書換え規則の左辺より右辺が長いか両辺が同じ長さの問題が 272 個ある。そのうちの 10 問が長さ辞書式順序で、30 問が長さ辞書式合字順序で停止性を判定できた数である。長さ辞書式順序で停止性を判定出来た問題は全て長さ辞書式合字順序でも判定できた。

手法 ・ツール	証明成功 (問)	時間 (秒)	証明失敗 (問)	時間 (秒)	時間切れ (問)
ll	10	0.07	1305	10.77	0
lll	30	2.00	1285	65.31	0
LPO	10	5.54	1242	2055.26	63
KBO	13	90.15	1265	679.63	37
3 次行列順序	28	280.39	1104	5257.57	183
TTT2	558	5534.90	279	16600.01	448
AProVE	693	8693.52	0	0	534

表 5.1: 簡約順序による停止性解析の結果

手法 ・ツール	証明成功 (問)	時間 (秒)	証明失敗 (問)	時間 (秒)	時間切れ (問)
ll	10	0.07	20	0.21	0
lll	30	2.00	0	0	0
LPO	3	0.33	27	19.37	0
KBO	10	1.52	20	7.11	0
3 次行列順序	9	93.52	21	129.88	0
TTT2	12	51.89	18	1102.46	0
AProVE	22	2324.98	0	0	8

表 5.2: lll の結果を基準とした簡約順序による停止性解析の結果 (計算 1 時間)

表 5.1 を参照する。長さ辞書式合字順序が他の手法より停止性証明が成功している。さらに、タイムアウトは 0 であり、計算時間も 230 秒と長さ辞書式合字順序以外の全ての手法より計算時間が短い。これは、長さ辞書式順序が長さを左辺と右辺で比べるだけという非常に簡単な手法であり長さ辞書式合字順序はその手法を用いているためである。また、名前は似ているが辞書式経路順序と長さ辞書式順序は停止性を判定できる領域が異なる。Zantema_04_z026 の問題 $\mathcal{R} = \{abb \to baa, aab \to bba\}$ は長さ辞書式順序で停止性を判定できるが辞書式経路順序では判定できない。長さ辞書式合字順序では $aa > ab$ かつ $aa > ab$ とした優先順序を定めれば順序はつくが辞書式経路順序では判定できない。

表 5.2 は長さ辞書式合字順序が停止性を判定出来た問題のみを用いて実験をした結果である。長さ辞書式合字順序が停止性を判定できる範囲は LPO、KBO、3 次の行列順序でも判定できない問題がある。特に、ICFP の問題群は長さ辞書式合字順序が証明成功する問題数が多く TTT2、AProVE でもタイムアウトになる問題は全て ICFP の問題である。なぜなら ICFP の問題は

書換えシステムを構成する文字数が他の問題と比べて非常に長いためである。他の問題は高々多くて 20 文字ほどだが ICFP の問題は長さ辞書式合字順序が停止性を判定できる問題で、少なくても 20 文字、多いと 2500 文字のシステムを処理しなければならない。長さ辞書式合字順序は長さ辞書式順序が基であるため処理が軽く、合字化により優先順序のとり方が多くなったためだと思われる。TPDB はプログラミングコンテスト ICFP contest 2010 [2] において作成された文字列書換えの停止性問題を 595 問含んでいる。

手法 ・ツール	証明成功 (問)	時間 (秒)	証明失敗 (問)	時間 (秒)	時間切れ (問)
ll	13	166.07	698	1256.07	604
lll	13	251.47	703	1085.84	599
LPO	29	15.51	827	1844.64	459
KBO	31	40.91	896	3758.01	388
3 次行列順序	113	733.44	982	10310.95	220
TTT2	558	5534.90	279	16600.01	448
AProVE	693	8693.52	0	0	534

表 5.3: 引数切り落としと簡約対による停止性解析の結果

表 5.3 を参照する。長さ辞書式合字順序は他の手法に比べて停止性を判定できる量が一番少なかった。また、タイムアウトの量も一番多い。簡約順序から簡約対に拡張すると他の手法のように停止性を判定できる量が多くなると予想されたが、結果は減少している。簡約順序 TPDB の ICFP の問題群のように非常に多くの文字で書換え規則を構成しているような問題は長さ辞書式合字順序で停止性を判定しようとすると時間がかかるためタイムアウトや計算時間が多い。長さ辞書式合字順序の簡約順序版では ICFP の問題群から 24 問停止性証明が成功している。しかし、簡約対版では 2 個のみである。Zantema _04_z007、Zantema_04_z012 と Zantema_04_z066 の問題を簡約対を用いた長さ辞書式順序では停止性を判定できたのに対して、長さ辞書式合字順序では判定できなかった。これは、長さ辞書式順序が優先順序において $\gtrsim$ を使用できたのに対して長さ辞書式合字順序では優先順序で $\gtrsim$ を使用できなく $\geqslant$ を用いているためとみられる。長さ辞書式合字順序の簡約順序で停止性を判定できた問題は理論上簡約対版でも判定できるが解けなかった問題は全てタイムアウトになっている。引数切り落とし π を利用できる時と出来ない時の差を比べた。すると、$\pi = \epsilon$ を用いた場合で解けた問題でも $\pi = \epsilon$ を使ってない場合ではタイムアウトになる問題が二つあった。Zantema _04_z092 と ICFP_2010_264370 である。

表 5.4 を参照する。この表は TPDB の項書換えシステムの問題集を用いた時の停止性解析の結果である。長さ辞書式順序と長さ辞書式合字順序では

手法・ツール	証明成功 (問)	時間 (秒)	証明失敗 (問)	時間 (秒)	時間切れ (問)
ll	99	1001.52	439	5828.28	900
lll	95	931.55	467	6570.54	876
Mint	762	170.90	590	586.89	24
NaTT	847	443.37	459	988.60	20
TTT2	763	1268.41	455	9872.00	88
AProVE	1014	3352.26	0	0	218

表 5.4: 文字列化順序による停止性解析の結果

長さ辞書式順序が SK90_4.33 と Rubio_04_$mfp90b$ の問題分多く解けた。長さ辞書式順序の簡約対で多く停止性を判定しているからだろう。文字列化による停止性解析の結果より、文字列化によって文字列書換えの手法が項上の順序で停止性が判定できることが実験の上でも確認された。

第6章　結論

本研究では、合字順序、文字列化順序という停止性解析技法を提案した。従来の長さ辞書式順序で停止性を証明できなかった文字列書換えシステムを長さ辞書式合字順序を用いて停止性を証明した。また、簡約対を用いた長さ辞書式合字順序では簡約順序では解けなかった問題が解けたが計算に時間がかかりすぎることが明らかとなった。長さ辞書式合字順序は優先順序を付ける際に擬順序 $\gtrsim$ を使用することが出来なかったため、簡約対を用いた際に $\gtrsim$ を使用できれば停止性判定が強力になるだろう。

他には今までされてこなかった文字列書換えシステムの停止性解析技法をそのまま項書換えシステムの停止性解析に使用できることが示された。今後は様々な文字列書換えの手法を文字列化を用いて解析してみることが必要だろう。可能な拡張をして引数切り落とし π を $\mathcal{S}(t)$ に組み込む方法が考えられる。つまり、変数に対し引数切り落としの定義を $\hat{\pi}(x) = x$ とし、

$$\mathcal{S}^{\pi}(t) = \{\pi(s) \mid s \in \mathcal{S}(t)\}$$

と定義する。そして、$>^{\mathcal{S}}$ と $\gtrsim^{\mathcal{S}}$ の定義中の $\mathcal{S}(t)$ を全て $\mathcal{S}^{\pi}(t)$ に置き換える。これが簡約対になると予想する。この証明と評価は今後の課題である。

第7章　謝辞

本研究を遂行するにあたり多大なご指導をしてくださった廣川直准教授に深く感謝いたします。寺内先生、小川先生にはセミナー等で助言、指導して頂きありがとうございました。また、研究室のメンバーや他研究室の皆様にも感謝を申し上げます。

参考文献

[1] Termination Problem Data Base version. 10.3. http://cl2-informatik.uibk.ac.at/mercurial.cgi/TPDB.

[2] The 15th ACM SIGPLAN International Conference on Functional Programming. http://cl-informatik.uibk.ac.at/workspace/ajsw10/slides/BF.pdf.

[3] T. Arts and J. Giesl. Termination of term rewriting using dependency pairs. *Journal of Theoretical Computer Science*, 236:133–178, 2000.

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

[5] V. Book and F. Otto. *String-Rewriting Systems.* Springer, 1993.

[6] A. Geser D. Hofbauer and J. Waldmann. Match-bounded string rewriting systems. In *Proc. 28th MFCS*, volume 2747, pages 449–459, 2003.

[7] J. Giesl, M. Brockschmidt, F. Emmes, F. Frohn, 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 IJCAR*, volume 8562 of *Lecture Notes in Computer Science*, pages 184–191. Springer, 2014.

[8] N. Hirokawa and A. Middeldorp. Automating the dependency pair method. *Information and Computation*, 199(1,2):172–199, 2005.

[9] D. Hofbauer and J. Waldmann. Termination of string rewriting with matrix interpretations. In *Proc. 17th RTA*, volume 4098, pages 328–342, 2006.

[10] S. kaminand and J. Lévy. Two generalizations of the recursive path ordering. *Unpublished manuscript*, 1980.

[11] D. Knuth and P. Bendix. Simple word problems in universal algebra. In *Computational Problems in Abstract Algebra*, pages 263–297, 1970.

[12] A. Koprowski and J. Waldmann. Arctic termination ...below zero. In *Proc. 19th RTA*, volume 5117, pages 202–216, 2008.

[13] M. Korp, C. Sternagel, H. Zankl, and A. Middeldorp. Tyrolean Termination Tool 2. In *Proc. 9th RTA*, volume 5595 of *Lecture Notes in Computer Science*, pages 295–304. Springer, 2009.

[14] D. Lankford. On proving term rewrite systems are noetherian. *Thechnical Report MTP-3*, 1979.

[15] É. Contejeanand C. Marché, A.P. Tomás, and X. Urbain. Mechanically proving termination using polynomial interpretations. *Journal of Automated Reasoning*, 34:325–363, 2005.

[16] L. Puel. Using unavoidable set of trees to generalize Kruskal's theorem. *Journal of Symbolic Computation*, 8:335–382, 1989.

[17] C. Sternagel and A. Middeldorp. Root-labeling. In *Proc. 19th RTA*, volume 5117, pages 336–350, 2008.

[18] Ngo Thi Bao Tran. Lightweight termination analysis for term rewriting. Master's thesis, JAIST, 2015.

[19] H. Xi. Towards automated termination proofs through "freezing". In *Proc. 9th RTA*, volume 1379, pages 271–285, 1998.

[20] A. Yamada, K. Kusakari, and T. Sakabe. Nagoya termination tool. In *Rewritting and Typed Lambda Calculi*, volume 8560 of *Lecture Notes in Computer Science*, pages 466–475. Springer, 2014.