

知能ソフトウェア特論

Intelligent Software

項書換え系 (2)

停止性

---

Term Rewriting Systems(2)

Termination

# 1. 項書換え系の基本用語と記法 (1/4)

(Terminology and notation for term rewriting systems)

基本的な記号

- 変数  $x, y, z, \dots$
- 定数  $0, 1, \dots, a, b, c, \dots$
- 関数記号  $f, g, h, \dots$

項 変数, 定数, または  $f(t_1, \dots, t_n)$   
( $t_1, \dots, t_n$  は項)

$F$ : 関数記号(定数を含む)の集合

$V$ : 変数の集合

$T(F, V)$ :  $F$ と $V$ から作られる項の集合

書換え規則 2つの項の順序対  $l \rightarrow r$

項書換え系 書換え規則の集合  $R$ .

Atomic symbols are classified into variables ( $x, y, z, \dots$ ), constants ( $0, 1, \dots, a, b, c, \dots$ ), and function symbols ( $f, g, h, \dots$ ).

A **term** is either a variable, a constant, or  $f(t_1, \dots, t_n)$ , if  $f$  is a function symbol and  $t_1, \dots, t_n$  are terms.

$F$  is a set of function symbols and constants.

$V$  is a set of variables.

$T(F, V)$  is a set of terms constructed from  $F$  and  $V$ .

A **rewrite rule** is an ordered pair  $l \rightarrow r$  of terms

A **term rewriting system (TRS)**  $R$  is a set of rewrite rules.

# 1. 項書換え系の基本用語と記法 (2/4)

代入  $\sigma:V \rightarrow T(F,V)$

各変数  $x \in V$  に項  $\sigma(x) \in T(F,V)$  を割り当てる写像. つぎの形の集合で表現する.

$$\sigma = \{x_1 \mapsto \sigma(x_1), \dots, x_n \mapsto \sigma(x_n)\}$$

A **substitution**  $\sigma:V \rightarrow T(F,V)$  is a mapping that assigns each variable  $x$  in  $V$  with a term  $\sigma(x)$  in  $T(F,V)$ . It is represented as a set in the form shown here.

Example:  $\sigma = \{x \mapsto a, y \mapsto f(x, a)\}$

代入の適用  $\sigma(t)$

項  $t$  に出現している各変数  $x$  をすべて項  $\sigma(x)$  で置き換えた項を  $\sigma(t)$  と書く.

An **application**  $\sigma(t)$  of a substitution  $\sigma$  to a term  $t$  is a term obtained from  $t$  by replacing all variables  $x$  in  $t$  by the term  $\sigma(x)$ .

Example: If  $\sigma = \{x \mapsto a, y \mapsto f(b)\}$ , then

$$\sigma(f(g(x, y))) = f(g(a, f(b)))$$

# 1. 項書換え系の基本用語と記法 (3/4)

## インスタンス

$\sigma(t) = s$  となる代入  $\sigma$  が存在するとき,  
 $s$  は  $t$  のインスタンスであるという.

A term  $s$  is an **instance** of a term  $t$  if there exists a substitution  $\sigma$  such that  $\sigma(t) = s$ .

Example:  $f(g(a))$  is an instance of  $f(x)$ .

# 1. 項書換え系の基本用語と記法 (4/4)

簡約関係  $\rightarrow_R$

ある書換え規則  $l \rightarrow r \in R$  の  
左辺のインスタンス  $\sigma(l)$  が  
項  $s$  の部分項として含まれていて、  
 $\sigma(l)$  を  $\sigma(r)$  に置き換えたときに、  
 $s$  が  $t$  になるとき、 $s \rightarrow_R t$  と書き、  
 $s$  は  $t$  に簡約可能という。

**Reduction relation**  $\rightarrow_R$

A term  $s$  is reducible to  $t$ ,  
notation  $s \rightarrow_R t$ , if there exists a  
rewrite rule  $l \rightarrow r$  in  $R$  such that  
 $t$  is obtained from  $s$  by replacing  
an instance  $\sigma(l)$  (contained in  $s$   
as a subterm) by the  
corresponding instance  $\sigma(r)$ .

Example: the rewrite rule  $f(x) \rightarrow g(x)$

the reduction relation  $h(f(a)) \rightarrow_R h(g(a))$

## 2. 停止性の決定不能性 (1/2)

(Undecidability of termination)

### 停止性

どんな初期項  $t_0$  からでも、書換えの無限列  $t_0 \rightarrow_R t_1 \rightarrow_R t_2 \rightarrow_R \dots$  が存在しないこと。

#### (例 1)

$$R = \begin{cases} f(0,1,x) \rightarrow f(x,x,x) \\ g(x,y) \rightarrow x \\ g(x,y) \rightarrow y \end{cases}$$

は停止性をもたない。

$$\begin{aligned} & f(g(0,1), g(0,1), g(0,1)) \\ & \rightarrow_R \cdot \rightarrow_R f(0,1, g(0,1)) \\ & \rightarrow_R f(g(0,1), g(0,1), g(0,1)) \\ & \rightarrow_R \dots \end{aligned}$$

### Termination

A TRS  $R$  is terminating, if there exists no infinite rewrite sequence  $t_0 \rightarrow_R t_1 \rightarrow_R t_2 \rightarrow_R \dots$  from any initial term  $t_0$ .

#### (Example 1)

The TRS  $R$  shown here is NOT terminating, because there is an infinite rewrite sequence as follows.

## 2. 停止性の決定不能性 (2/2)

### 定理 1 (停止性は決定不能)

項書換え系の停止性は決定不能である。

すなわち、任意の項書換え系  $R$  を入力として受け取り、それが停止性をもつかどうかを有限時間内で判定するアルゴリズムは存在しない。

### Theorem 1 (undecidability of termination)

Termination of TRSs is undecidable.  
In other words, there exists no algorithm that accepts a TRS as an input and decides in a finite time whether it is terminating or not.



### 工学的な研究目標:

できるだけ広い範囲の項書換え系の停止性を検証できるアルゴリズムの開発

**Engineering research goal:**  
development of algorithms that can verify the termination of broader classes of TRSs.

### 3. 簡約順序 (1/6) (Reduction order)

#### 二項関係

2つの要素間の関係の概念を抽象化したもの

集合  $S$  上の二項関係  $R$  は、直積  $S \times S$  の部分集合.

$$R \subseteq S \times S = \{ (x, y) \mid x \in S, y \in S \}$$

$(x, y) \in R$  のとき、 $xRy$  と書く.

**Binary relation** is an abstraction of a relationship between two objects.

Formally, a **binary relation**  $R$  on a set  $S$  is a subset of the Cartesian product

$$S \times S = \{ (x, y) \mid x \in S, y \in S \}.$$

When  $(x, y) \in R$ , we write  $xRy$ .

Example:  $S = \{a, b, c\}$ ,  $R = \{(a, b), (a, c), (b, c)\}$ ,  $aRb$

この授業では  $R$  のかわりに不等号  $>$  などを使う

In this lecture, we often use the symbol  $>$  instead of  $R$ .



### 3. 簡約順序 (2/6)

#### 厳格半順序

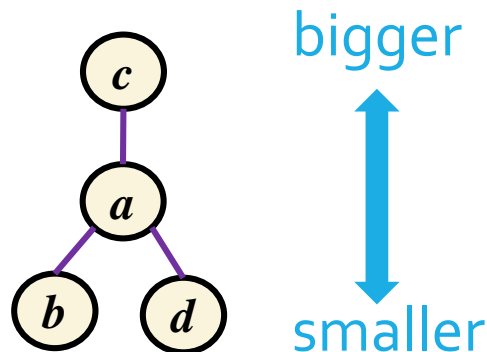
2つの要素間の大小関係を抽象化したもの

推移性 ( $x \succ y \wedge y \succ z \Rightarrow x \succ z$ ) と

非反射性 ( $\neg(x \succ x)$ ) を満たす二項関係  $\succ$  .

Example:  $S = \{a, b, c, d\}$ ,

$\succ = \{(a,b), (a,d), (c,a), (c,b), (c,d)\}$ .



**Strict partial order** is an abstraction of “*is greater than*” relationship.

Formally, a **strict partial order** is a binary relation  $\succ$  that satisfies *transitivity*: if  $x \succ y$  and  $y \succ z$ , then  $x \succ z$

and

*irreflexivity*:  $\neg(x \succ x)$  for any  $x$ .

### 3. 簡約順序 (3/6)

#### 定理 2 (停止性の必要十分条件)

項書換え系  $R$  が停止性をもつ必要十分条件は、  
すべての書換え規則  $l \rightarrow r \in R$  について  $l \succ r$   
となる簡約順序  $\succ$  が項の集合の上に存在すること。

**Theorem 2 (necessary and sufficient condition for termination)**

A TRS  $R$  is terminating if, and only if,  
 $l \succ r$  for all rewrite rules  $l \rightarrow r$  of  $R$   
in some reduction order  $\succ$  on the set of terms.

例 :  $R = \begin{cases} \text{not}(\text{and}(x, y)) \rightarrow \text{or}(\text{not}(x), \text{not}(y)) \\ \text{not}(\text{not}(x)) \rightarrow x \end{cases}$

この  $R$  が停止性をもつことを証明するには、  
ある簡約順序  $\succ$  を用いて、次のことを示せばよい。

$\text{not}(\text{and}(x, y)) \succ \text{or}(\text{not}(x), \text{not}(y))$   
 $\text{not}(\text{not}(x)) \succ x$

**Example:** In order to prove the termination of the TRS  $R$  shown here, we need to prove the two inequalities given below, using some reduction order  $\succ$ .

### 3. 簡約順序 (4/6)

簡約順序 整礎な書換え順序のこと.

- 書換え順序 項の集合の上で定義された単調で安定な厳格半順序  $\succ$ .

単調性 : 局所的な減少が大域的な減少となる.  
すなわち, 任意の項  $s, t$  について :

$$s \succ t \text{ ならば } f(\dots, s, \dots) \succ f(\dots, t, \dots)$$

安定性 : 代入により順序が変化しない. すなわち,  
任意の項  $s, t$  と代入  $\sigma$  について :

$$s \succ t \text{ ならば } \sigma(s) \succ \sigma(t)$$

- 整礎 無限下降列  $a_1 \succ a_2 \succ a_3 \succ \dots$  が存在しないこと.

A **reduction order** is a *well-founded rewrite order*.

A **rewrite order** is a *monotone and stable* strict partial order  $\succ$  defined on a set of terms.

A **monotone** order ensures that local decrease implies global decrease, i.e.,  $s \succ t$  implies  $f(\dots, s, \dots) \succ f(\dots, t, \dots)$ .

A **stable** order ensures that the order is preserved by substitution, i.e.,  $s \succ t$  implies  $\sigma(s) \succ \sigma(t)$ .

A **well-founded** order ensures that there is no infinitely decreasing sequence  $a_1 \succ a_2 \succ a_3 \succ \dots$

### 3. 簡約順序 (5/6)

#### (例 2)

項  $t$  のサイズ (含まれる変数, 定数, 関数記号の数) を  $|t|$ , 項  $t$  に変数  $x$  が現れている回数を  $|t|_x$  と書く. つぎの式で定義される  $\succ$  は簡約順序である.

$$s \succ t \iff |s| > |t| \text{ and } |s|_x \geq |t|_x \text{ for all } x \in V$$

証明は省略 (自明ではないが, 難しくもない)

したがって, つぎの  $R$  は停止性をもつ.

$$R = \begin{cases} nand(x, x) \rightarrow not(x) \\ not(nand(x, y)) \rightarrow and(x, y) \end{cases}$$

$$nand(x, x) \succ not(x)$$

$$3 > 2$$

$$not(nand(x, y)) \succ and(x, y)$$

$$4 > 3$$

#### (Example 2)

Let  $|t|$  be the number of symbols (i.e., variables, constants and function symbols) in  $t$ , and  $|t|_x$  be the number of occurrences of  $x$  in  $t$ . Then the binary relation defined here is a reduction order. (Proof omitted.)

Using this fact, we see that the TRS  $R$  given here is terminating.

### 3. 簡約順序 (6/6)

#### (例 3)

変数に重み 1, 関数記号  $f$  に正の整数の重み  $w_f$  を与え, 項  $t$  の重みの総和を  $w(t)$  と書く.  
つぎの  $\succ$  は簡約順序である.

$$s \succ t \Leftrightarrow w(s) > w(t) \text{ and } |s|_x \geq |t|_x \text{ for all } x \in V$$

したがって, つぎの  $R$  は停止性をもつ.

$$R = \begin{cases} \text{not}(\text{and}(x, y)) \rightarrow \text{or}(\text{not}(x), \text{not}(y)) \\ \text{not}(\text{not}(x)) \rightarrow x \end{cases}$$

$$w(\text{not}) = w(\text{or}) = 1, w(\text{and}) = 3$$

$$\text{not}(\text{and}(x, y)) \succ \text{or}(\text{not}(x), \text{not}(y))$$

$$\text{not}(\text{not}(x)) \succ x$$

$$6 > 5$$

$$3 > 1$$

#### (Example 3)

Give a weight 1 to each variable and a positive integer  $w_f$  to each function symbol  $f$ . Let  $w(t)$  be the sum of the weights of all symbols in a term  $t$ . Then the binary relation defined here is a reduction order.

For example, the TRS  $R$  given here is terminating.

# 【参考】多項式解釈への動機付け(1/2)

(Motivation for polynomial interpretation)

$$s \succ t \iff |s| > |t| \text{ and } |s|_x \geq |t|_x \text{ for all } x \in V$$
$$s \succ t \iff w(s) > w(t) \text{ and } |s|_x \geq |t|_x \text{ for all } x \in V$$

■例2・例3では,

$$|s|_x \geq |t|_x \text{ for all } x \in V$$

の(表現力を制限する)条件が不可欠.  
なぜなら, この条件がなければ

$$d(x) > plus(x, x)$$

$$4 > 3$$

$$w(d)=3$$

$$w(plus)=1$$

$$d(d(2)) < plus(d(2), d(2))$$

$$7 < 9$$

$$w(2)=1$$

→ ※安定性が失われる

This condition (with yellow background) in Examples 2 and 3 **undesirably** restricts the expressiveness of rewrite rules, but unfortunately, it is necessary, because without it, the strict partial order would **lose its stability** as shown in this example.

## 【参考】多項式解釈への動機付け(2/2)

$$\text{and}(x, \text{or}(y, z)) > \text{or}(\text{and}(x, y), \text{and}(x, z))$$

■この例では, どんな重みを付けても,  
左辺 > 右辺  
とすることはできない

→ 多項式解釈なら  
左辺 > 右辺  
とすることができる

In this example, we can never get the left-hand side greater than the right-hand side with any weighting.

With polynomial interpretation you study in the next section, however, you will get LHS greater than RHS.

# 4. 多項式解釈に基づく停止性証明 (1/6)

(Termination proof by polynomial interpretation)

## 単調関数による解釈

$A \subseteq \{1, 2, 3, \dots\}$  を正の整数の集合とし、  
各関数記号  $f \in F$  に  $A$  の上で定義された関数

$$f_A : A \times \dots \times A \rightarrow A$$

を対応させる。

$f_A$  は次式で定義される **単調** なものとする。

$$a > b \Rightarrow f_A(\dots, a, \dots) > f_A(\dots, b, \dots)$$

項  $t$  に現れているすべての変数  $x \in V$  を  
 $A$  の上を動く変数  $x \in A$  と解釈し、  
各関数記号  $f$  を  $A$  の上の単調な関数  $f_A$  と解釈  
してできる数式を  $\varphi(t)$  とする。

$f_A$  と  $\varphi(t)$  をそれぞれ  $f$  と  $t$  の **解釈** という。

(定数記号は、引数無し関数記号として扱う)

## Interpretation by monotonic functions

Let  $A \subseteq \{1, 2, 3, \dots\}$  be a set of positive integers. We associate each function symbol  $f$  with a function  $f_A$  defined on  $A$ , where  $f_A$  is a **monotone** function satisfying

$$a > b \Rightarrow f_A(\dots, a, \dots) > f_A(\dots, b, \dots).$$

Let  $\varphi(t)$  be a mathematical formula obtained from the term  $t$  by interpreting each function symbol  $f$  and each variable  $x$  in  $t$  as a monotone function  $f_A$  on  $A$  and a variable  $x$  moving on  $A$ , respectively.  $f_A$  and  $\varphi(t)$  are called an **interpretation** for  $f$  and  $t$ . Constant symbols are regarded as function symbols with no arguments.



## 4. 多項式解釈に基づく停止性証明 (2/6)

(例 4)

(Example 4)

$$t = g(x, f(y, z))$$

$$f_A(x, y) = 2x + y + 1, \quad g_A(x, y) = xy.$$

$$\begin{aligned} \varphi(g(x, f(y, z))) &= g_A(x, f_A(y, z)) \\ &= g_A(x, 2y + z + 1) \\ &= x(2y + z + 1) \\ &= 2xy + xz + x \end{aligned}$$

## 4. 多項式解釈に基づく停止性証明 (3/6)

### 解釈に基づく半順序

項の集合の上の半順序  $\succ_A$  を  
つぎの式で定義する.

$$s \succ_A t \iff \varphi(s) > \varphi(t)$$

ただし, 不等式  $\varphi(s) > \varphi(t)$  は,  
含まれる変数が  $A$  のどの値を  
とっても成り立つことを表す.

### Partial order induced by interpretation

We define  $\succ_A$  to be a partial order on the set of terms, specified by

$$s \succ_A t \iff \varphi(s) > \varphi(t)$$

where  $\varphi(s) > \varphi(t)$  means that this inequality holds universally (for all values of  $A$  taken by the variables in  $s$  and  $t$ ).

## 4. 多項式解釈に基づく停止性証明 (4/6)

**定理 3** すべての関数記号  $f$  の解釈  $f_A$  が単調ならば,  
 $>_A$  は項の集合の上の簡約順序である.

**Theorem 3 (monotone interpretation induces a reduction order)**

If the interpretations  $f_A$  for all  $f$ 's are monotone, then  $>_A$  is a reduction order on the set of terms.

**系 4 (単調な解釈に基づく停止性証明)**

すべての書換え規則  $l \rightarrow r \in R$  について  $\varphi(l) > \varphi(r)$   
となる単調な解釈が存在すれば,  $R$  は停止性をもつ.

**Corollary 4 (termination proof by monotone interpretation)**

If there exists a monotone interpretation  $\varphi$  such that  $\varphi(l) > \varphi(r)$  for all rewrite rules  $l \rightarrow r$  of  $R$ , then  $R$  is terminating.

## 4. 多項式解釈に基づく停止性証明 (5/6)

### 多項式解釈

$f_A$  が多項式のときの解釈を**多項式解釈**という.

多項式の**すべての係数を正の整数**とし, かつ,  
式の値が**すべての変数に依存**するようにすれば,  
その多項式は**単調**.

### Polynomial interpretation

$f_A$  is a polynomial interpretation if it is a polynomial.

A polynomial is monotonic if

- all of its coefficients are positive integers and
- its value is dependent on all variables.

## 4. 多項式解釈に基づく停止性証明 (6/6)

(例 5)

(Example 5)

$$R = \begin{cases} \text{and}(x, \text{or}(y, z)) \rightarrow \text{or}(\text{and}(x, y), \text{and}(x, z)) \\ \text{or}(\text{or}(x, y), z) \rightarrow \text{or}(x, \text{or}(y, z)) \end{cases}$$

(停止性の証明)

(Proof of termination)

$$A = \{2, 3, 4, \dots\}, \quad \text{or}_A(x, y) = 2x + y + 1, \quad \text{and}_A(x, y) = xy.$$

$$\varphi(\text{and}(x, \text{or}(y, z)))$$

$$= x(2y + z + 1)$$

$$= 2xy + xz + x$$

$$> 2xy + xz + 1$$

$$= \varphi(\text{or}(\text{and}(x, y), \text{and}(x, z)))$$

$$\varphi(\text{or}(\text{or}(x, y), z))$$

$$= 2(2x + y + 1) + z + 1$$

$$= 4x + 2y + z + 3$$

$$> 2x + 2y + z + 2$$

$$= \varphi(\text{or}(x, \text{or}(y, z)))$$

→ このような多項式解釈を見つけるソフトウェアが開発されている

# 演習問題6

多項式解釈

$$A = \{2, 3, 4, \dots\}$$

$$\text{plus}_A(x, y) = x + 2y,$$

$$s_A(x) = x + 1$$

$$\text{double}_A(x) = 3x$$

$$0_A = 2$$

を用いて、つぎの項書換え系の停止性を証明せよ。

$$R = \begin{cases} \text{plus}(x, 0) \rightarrow x \\ \text{plus}(x, s(y)) \rightarrow s(\text{plus}(x, y)) \\ \text{double}(0) \rightarrow 0 \\ \text{double}(s(x)) \rightarrow s(\text{double}(x)) \end{cases}$$

【注意】定数記号は引数無し関数記号として扱う。

定数記号  $0$  はその本来の意味(ゼロ)にかかわらず、ここでは停止性証明の目的で、多項式(定数)  $2$  として解釈されている。

# Exercise 6

Prove the termination of the TRS  $R$  given here by using the given polynomial interpretation.

Constant symbols are treated as function symbols with no arguments. In particular here, the constant symbol  $0$  should be interpreted as a polynomial (constant)  $2$  for the purpose of termination proof, although its natural meaning is clearly ‘zero’.