JAIST Repository
https://dspace.jaist.ac.jp/

Title	項書き換えシステムにおける可簡約演算子とその応用
Author(s)	中村，正樹；緒方，和博；二木，厚吉
Citation	情報処理学会論文誌，46(SIG 6 (PRO25))：47-59
Issue Date	2005-04-15
Type	Journal Article
Text version	publisher
URL	http://hdl.handle.net/10119/4584
Rights	社団法人　情報処理学会，中村正樹，緒方和博，二木厚吉，情報処理学会論文誌，46(SIG 6 (PRO25))，2005，47-59．　ここに掲載した著作物の利用に関する注意：本著作物の著作権は（社）情報処理学会に帰属します。本著作物は著作権者である情報処理学会の許可のもとに掲載するものです。ご利用に当たっては「著作権法」ならびに「情報処理学会倫理綱領」に従うことをお願いいたします。 Notice for the use of this material: The copyright of this material is retained by the Information Processing Society of Japan (IPSJ). This material is published on this web site with the agreement of the author (s) and the IPSJ. Please be complied with Copyright Law of Japan and the Code of Ethics of the IPSJ if any users wish to reproduce, make derivative work, distribute or make available to the public any part or whole thereof. All Rights Reserved, Copyright (C) Information Processing Society of Japan.
Description	

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

項書き換えシステムにおける可簡約演算子とその応用

中村正樹[†] 緒方和博[†,††] 二木厚吉[†]

項書き換えシステムにおける演算子は，その意味から関数としての役割を持つ演算子と台集合の要素を構成する構成子としての演算子の 2 つに分けられる．このとき項書き換えシステムには，任意の正規形が構成子からのみなるという性質が期待される．我々はそのような性質を持つ演算子として可簡約演算子の概念を提案する．すなわち可簡約演算子を含む任意の項は正規形ではない．通常の項，基底項（変数を含まない項），あるソートに属する項をそれぞれ簡約対象とした場合の可簡約演算子の定義を与え，その性質を示す．これらの可簡約演算子は，単に演算子が正規形に現れないという性質を持つのみでなく，効率的な書き換え関係を得る助けにもなる．また代数仕様への応用として，振舞仕様における重要な性質であるコヒーレント性の十分条件を与える．

Reducible Operation Symbols for the Term Rewriting System and Their Applications

Masaki Nakamura,[†] Kazuhiro Ogata[†,††]
and Kokichi Futatsugi[†]

In a term rewriting system, operation symbols can be classified into two divisions from its denotational semantics: the operation symbols as functions and the operation symbols as constructors. Each term constructed by constructors corresponds to an element of the carrier set. Then the term rewriting system is expected to have the property that each normal form is constructed by only constructors. We propose the notion of the reducible operation symbols as any term having a reducible function symbol is not a normalform. We discuss about reducible operation symbols for ordinary terms, ground terms and terms of some sorts. Not only the reducible operation symbol has the property that it does not appear in each normal form but also it gives us an efficient rewrite relation. Moreover as an application to algebraic specifications we show a sufficient condition for the behavioral coherence, which is one of the most important properties of the behavioral specification.

1. 序

項書き換えシステムは，項のパターンを別のパターンに置き換える書き換え規則の集合によって与えられる．書き換え規則による項の書き換えにおいては，各演算子は単に項を構成する要素にすぎず演算子間に区別はない．しかしながら，何らかの目的で項書き換えシステムを記述する者は，ある演算子は関数として，またある演算子は構成子として定義する．たとえば，自然数を定義する項書き換えシステムに対しては，0 や $s(\,)$ は自然数を構成する演算子であり，$+$ や $\times$ は自然数上の関数を表す演算子である．構成子として定義された演算子のみからなる項を構成子項と呼ぶとき，関数を表す演算子はすべての構成子項に対して定義されるべきである．正規形とは，書き換え規則が適用できない，すなわちどの書き換え規則の左辺のインスタンスも含まない項である．正規形でない項は可簡約な項と呼ばれる．関数として定義された演算子は正規形に現れないことが期待される．本研究では可簡約の概念を演算子に適用する．**可簡約演算子**とは，それを含む任意の項が可簡約となる演算子であり，関数として定義された演算子が満たすべき性質といえる．書き換える項の種類に応じていくつかの種類の可簡約性を提案する．その応用として，代数仕様の一種である振舞仕様に対し，適切な可簡約演算子を与え，それが振舞仕様の重要な性質であるコヒーレント性の十分条件を与えることを示す☆．

† 北陸先端科学技術大学院大学
Japan Advanced Institute of Science and Technology
†† NEC ソフトウェア北陸
NEC Software Hokuriku, Ltd.

☆ 本研究は論文「項書換えシステムにおける可簡約演算子とその応用」情報科学技術レターズ 第 3 巻，pp.13–14，2004，として発表された内容を発展させたものである．

2. 準　　備

準備として，項書き換えシステム（TRS）[14] の諸定義・用語・性質を与える．

2.1 項書き換えシステム

シグネチャ Σ は演算子の有限集合で与えられる．ただし各演算子 $f \in \Sigma$ はそれぞれ引数の数 $ar(f) \in \mathcal{N}$ を持つ．ここで $\mathcal{N}$ は自然数の集合を表す．変数の可算集合 V は，シグネチャと互いに素であるとする（$\Sigma \cap V = \emptyset$）．項は演算子をノードに持ち変数を葉に持つ木構造である．シグネチャ Σ と変数の集合 V に対する項の集合 $T(\Sigma, V)$（混乱がないときは T で略記）は以下を満たす最小の集合である：$V \subset T(\Sigma, V)$ かつ任意の項 $\overrightarrow{t_i} \in T(\Sigma, V)$ および演算子 $f \in \Sigma$ に対して $f(\overrightarrow{t_i}) \in T(\Sigma, V)$．ここで $\overrightarrow{t_i}$ は $t_1, \ldots, t_n$ の略記とする．以降，同様の列の略記を用いる．f を演算子，x，y，i，j，$\overrightarrow{x_i}$ を変数として断りなく用いることがある．項 t のルート記号を $root(t)$ で記述する．引数の数が 0 の演算子を定数と呼び，定数項 $c()$ を単に c と書く．変数を含まない項を基底項と呼び，その集合 $T(\Sigma, \emptyset)$ を $T(\Sigma)$ と記述する．項のポジションは正数の列で表される．項 t のポジション集合 $O(t) \subset \mathcal{N}_+^*$ は以下で定義される．ただし $\mathcal{N}_+$ は正数の集合，A^* は A の要素からなる列の集合である．空列は ε とする．

$$O(t) = \begin{cases} \{\varepsilon\} & \text{if } t \in V \\ \{\varepsilon\} \cup \bigcup_{i=1}^{n} \bigcup_{p \in O(t_i)} \{i.p\} & \text{if } t = f(\overrightarrow{t_i}). \end{cases}$$

$p = q.q'$ となる q' が存在するとき $p \geq q$ と書き，$p \neq q$ のとき $p > q$ と書く（e.g. $1.2.3.4 > 1.2$）．項 t のポジション $p \in O(t)$ の部分項 $t|_p$ を $t|_\varepsilon = t$ かつ $f(\overrightarrow{t_i})|_{i.p} = t_i|_p$ と定義する．項 $s, t \in T$ に対して，t のポジション $p \in O(t)$ の部分項 $t|_p$ を s で置き換えた項 $t[s]_p$ を $t[s]_\varepsilon = s$ かつ $f(\overrightarrow{t_i})[s]_{i.p} = f(t_1, \ldots, t_i[s]_p, \ldots, t_n)$ と定義する．シグネチャ Σ に含まれない特別な定数 $\Box$ をただ 1 つ含む項 $t[\Box]_p$ を文脈と呼ぶ．写像 $\theta : V \to T$ を代入と呼ぶ．項 $t \in T$ の各変数 x を項 $\theta(x)$ に置き換えた項を $t\theta$ と記述する．$t = s\theta$ となる代入 θ が存在するとき，t は s のインスタンスという．

項書き換えシステム（TRS）はシグネチャと書き換え規則の有限集合の対 (Σ, R) からなる．書き換え規則 $l \to r$ は以下を満たす項の対である．左辺 l は変数でなく，右辺 r に含まれる変数はすべて左辺 l にも含まれる．TRS (Σ, R) を単に R と書いたときは，Σ は R に現れるすべての演算子の集合とする．TRS R，ポジション p に対して，書き換え関係 $\to_p$ および $\to_R$ を以下で定義する．

$$\begin{aligned} s \to_p t \overset{\text{def}}{\iff} \quad & \exists l \to r \in R, \theta : V \to T. \\ & s|_p = l\theta, t = s[r\theta]_p, \\ s \to_R t \overset{\text{def}}{\iff} \quad & \exists p \in O(s). s \to_p t. \end{aligned}$$

二項関係 $\circ$ に対して，その反射的推移的閉包を $\circ^*$ と書く．また，$t \circ u$ となる u が存在しないとき，t を $\circ$-正規形と呼ぶ．$\to_R$-正規形を **R-正規形**と呼ぶ．ある書き換え規則 $l \to r$ の左辺のインスタンス $l\theta$ をリデックスと呼ぶ．TRS による項の簡約とは，リデックスを対応する右辺のインスタンス $r\theta$ に順次置き換えていく作業である．リデックスを含む項を**可簡約項**と呼ぶ．t が R-正規形であることと，t が可簡約でないことは同値である．R-正規形の集合を NF_R で記述する．また，リデックスの集合を $Red(R) = \{l\theta \in T \mid l \to r \in R, \theta \in V \to T\}$，左辺の集合を $L(R) = \{l \in T \mid l \to r \in R\}$ と定義する．

例 2.1 以下の TRS を考える．

$$R_n = \begin{cases} +(x, 0) \to x \\ +(x, s(y)) \to s(+(x, y)) \end{cases}$$

規則 $+(x, 0) \to x \in R$，代入 $\theta(x) = s(0)$ が存在するため，項 $+(s(0), 0)$ はリデックスである．よって $+(s(s(0)), +(s(0), 0)) \to_2 +(s(s(0)), s(0))$ が成り立つ．正規形まで簡約を行うと $s(s(s(0)))$ が得られる．これは $2 + (1 + 0) = 3$ の計算に相当する． □

3. 可簡約演算子とその性質

可簡約演算子をそれを含む任意の項は可簡約であると定義する．関連研究に，Sufficient completeness [6]，基底可簡約性 [5] などの研究がある．本論文では演算子の視点からこれらの概念を再考し，有用な性質を導く．TRS $R = \{id(x) \to x\}$ に対して，演算子 id は可簡約である．可簡約演算子はその引数の簡約が制限できるという有用な性質を持つ．たとえば，項 $id(id(0))$ は，最内戦略 [14] で $id(\underline{id(0)}) \to_R \underline{id(0)} \to_R 0$（下線部がリデックス）と簡約できるが，もし可簡約演算子 id の引数の簡約を制限したとしても，$\underline{id(id(0))} \to_R \underline{id(0)} \to_R 0$ のように同じく正規形が得られる．

3.1 文脈依存書き換え

文脈依存書き換え（CSR）は書き換えの対象となるリデックスが制限された書き換え関係である [7]．書き換え可能なリデックスは，**置換写像** $\mu : \Sigma \to \mathcal{P}(\mathcal{N}_+)$（ただし任意の $f \in \Sigma$ に対して $\mu(f) \subset \{1, 2, \ldots, ar(f)\}$）

によって形式化される．置換写像 μ に対して，項 t の書き換え可能なポジションの集合 $O_\mu(t) \subset O(t)$ は以下で定義される．

$$O_\mu(t) = \begin{cases} \{\varepsilon\} & \text{if } t \in V, \\ \{\varepsilon\} \cup \bigcup_{i \in \mu(f)} \bigcup_{p \in O_\mu(t_i)} \{i.p\} & \text{if } t = f(\overrightarrow{t_i}). \end{cases}$$

文脈依存書き換え $\to_\mu$ を以下で定義する．

$$s \to_\mu t \overset{\text{def}}{\iff} \exists p \in O_\mu(s), s \to_p t$$

$\to_\mu$-正規形を **μ-正規形**，μ-正規形の集合を NF_μ とする．R-正規形は μ-正規形である（$NF_R \subset NF_\mu$）．

例 3.1 $\Sigma_l = \{cons, hd, tl, zero, 0\}$ とする．項 $cons(x, y)$ を $x{:}y$ と略記する．TRS R_l を与える．

$$R_l = \begin{cases} hd(x{:}y) \to x \\ tl(x{:}y) \to y \\ zero \to 0{:}zero \end{cases}$$

演算子 $cons$ はリストの構成子である．hd はリストの先頭要素を返す関数，tl は残りの先頭以外のリストを返す関数を表す．$zero$ は無限リスト $0{:}0{:}0{:}\cdots$ を表す．置換写像 μ を $\mu(cons) = \mu(hd) = \mu(tl) = \{1\}$ とする．このとき $2 \notin \mu(cons)$ であるため，$2 \notin O_\mu(0 : zero)$．よって $0 : zero \to_\mu 0 : (0 : zero)$ は成り立たない．この置換写像 μ で，3 番目の規則 $zero \to 0 : zero$ の無限の適用をうまく避けて，（CSR の）最内戦略で項 $hd(tl(zero))$ を R-正規形に簡約できる．$hd(tl(\underline{zero})) \to_\mu hd(\underline{tl(0{:}zero)}) \to_\mu hd(\underline{zero}) \to_\mu \underline{hd(0{:}zero)} \to_\mu 0$． □

簡約ポジションの制限のため，μ-正規形は R-正規形とならない可能性がある（$NF_R \neq NF_\mu$）．たとえば，可簡約項 $+(0, +(0, 0))$ は $\mu(+) = \emptyset$ のとき μ-正規形である．よって CSR では，停止性，合流性などの書き換え関係における重要な性質とともに μ-正規形が R-正規形であるという性質が重要となる．本論文ではそのような性質を**正当性**と呼び，可簡約演算子による正当性の十分条件を与える．

3.2 可簡約演算子

$NF_\mu = NF_R$ を満たすとき**置換写像は正当である**という．$\mu(f) = \{1, 2, \ldots, ar(f)\}$ のとき f を（μ に関して）**自明な演算子**といい，任意の演算子が自明のとき，μ を**自明な置換写像**と呼ぶ．自明な μ に対しては，$\to_R = \to_\mu$ であるため明らかに μ-正規形は R-正規形である．本研究の目的の 1 つは，可簡約演算子を用いて自明でない μ を含む正当性の十分条件を与えることである．

定義 3.2 置換写像 μ は**正当**である $\overset{\text{def}}{\iff} NF_\mu = NF_R$． □

定義 3.3 演算子 $f \in \Sigma$ は**可簡約**である $\overset{\text{def}}{\iff} f$ を含む任意の項が可簡約である． □

可簡約演算子の必要十分条件を与える．

定理 3.4 f は可簡約演算子 $\iff f(\overrightarrow{x_i}) \in L(R)$ が存在する．ただし $x_i \in V$ は互いに異なる変数．

証明． 自明． □

可簡約演算子による正当性の十分条件を与える．

定理 3.5 任意の $f \in \Sigma$ が可簡約または自明のとき，μ は正当である．

証明． μ-正規形に可簡約演算子が含まれないことを示す．$t \in T$ が可簡約演算子を含むと仮定する．p を $root(t|_p)$ が可簡約演算子となる最小のポジションとする．すなわち $q < p$ に対して $root(t|_q)$ は可簡約でない．$root(t|_p) = f$ とする．可簡約演算子以外は自明であるため，$p \in O_\mu(t)$ である．定理 3.4 より $f(\overrightarrow{x_i}) \in L(R)$ が存在するので，$t|_p$ はリデックスである．CSR の定義から t は μ-正規形ではない．よって任意の $t \in NF_\mu$ に対して $O_\mu(t) = O(t)$ が成り立ち，$NF_\mu = NF_R$． □

例 3.6 すべての左辺が $f(\overrightarrow{x_i})$ の形をしている TRS を RPS（recursive program scheme）と呼ぶ．定理 3.5 より，任意の RPS R に対して，任意の置換写像 μ は正当である． □

定理 3.4 の可簡約演算子の必要十分条件はあまり有用ではない．なぜなら TRS の強みの 1 つは，関数をパターンごとに定義できる点だからである．たとえば，例 2.1 の TRS R_n の演算子 $+$ は，$(x, 0)$ と $(x, s(y))$ のパターンについて定義されている．例 3.1 の TRS R_l の hd，tl はパターン $cons(x, y)$ について定義されている．また別の角度から見ると，演算子 $cons$ は $hd(\square)$ と $tl(\square)$ という文脈のパターンに対して定義されていると見ることもできる．以下では，パターンによって定義された実際的な TRS を含む可簡約演算子の概念を与える．

3.3 基底可簡約演算子

項 t の任意の基底インスタンスが可簡約であるとき**基底可簡約**という[5]．可簡約項は明らかに基底可簡約である．例 2.1 の TRS R_n に対して，項 $+(x, y)$ は可簡約でないが基底可簡約である．演算子 f に対して，項 $f(\overrightarrow{x_i})$ が基底可簡約項のとき，f を**基底可簡約演算子**と定義する．しかしこのままでは有用な正当性の条件が得られない．さらに強い基底可簡約演算子の定義を与える．基底可簡約項 $+(x, y)$ は，2 引数目の変数 y のみに基底項を代入すればリデックスとなる．すなわち $+(t, t')$ は，部分項 t' が基底項であれば，部

分項 t が変数を含んでいても，可簡約項である．このような演算子を $\{2\}$ 基底可簡約であると定義する．I 基底可簡約演算子を用いて，集合 I に含まれない部分項の簡約を制限した μ が正当であることを示す．

定義 3.7 置換写像 μ は**基底正当**である $\stackrel{\text{def}}{\Longleftrightarrow}$ $NF_\mu \cap T(\Sigma) \subset NF_R$. □

定義 3.8 集合 I ($\subset \mathcal{N}$) に対し f は I **基底可簡約**である $\stackrel{\text{def}}{\Longleftrightarrow}$ $i \in I$ で t_i が基底項である部分項 $f(\overrightarrow{t_i})$ を持つ任意の項は可簡約である． □

I 基底可簡約演算子の必要十分条件を与える．

定理 3.9 $f \in \Sigma$ が I 基底可簡約演算子である $\Longleftrightarrow$ 任意の $i \in I$ に対し t_i が基底 R-正規形である $f(\overrightarrow{t_i})$ はリデックスである．

証明. ($\Rightarrow$:) 項 s_i $(1 \le i \le n)$ を $i \in I$ に対して $s_i = t_i$ かつ $i \notin I$ に対して $s_i = x_i \in V$ と定義する．ただし x_i は t_i に出現しない変数とする．f は I 基底可簡約であるので項 $f(\overrightarrow{s_i})$ は可簡約である．各 s_i は R-正規形であるため項 $f(\overrightarrow{s_i})$ はリデックスである．$l \in L(R)$ に対して $f(\overrightarrow{s_i}) = l\theta$ と仮定する．代入 θ' を $i \in I$ に対して $\theta'(x_i) = t_i$ かつそれ以外の変数 $x \notin \{x_i \mid i \in I\}$ に対して $\theta'(x) = \theta(x)$ と定義する．$f(\overrightarrow{t_i}) = l\theta'$ より $f(\overrightarrow{t_i})$ はリデックス．

($\Leftarrow$:) $i \in I$ で t_i が基底項である部分項 $t|_p = f(\overrightarrow{t_i})$ を持つ任意の項 t を考える．ある t_i が可簡約項ならば t は可簡約項である．どの t_i も可簡約項でないならば，仮定より $t|_p$ が可簡約項であるので t は可簡約項である． □

I 基底可簡約演算子による正当性の十分条件を与える．

補題 3.10 項 t が μ-正規形でかつ $p \in O_\mu(t)$ ならば $t|_p$ は μ-正規形．

証明. 定義より $q \in O_\mu(t|_p)$ ならば $p.q \in O_\mu(t)$ である．t は μ-正規形なので，任意の $q \in O_\mu(t|_p)$ に対して $(t|_p)|_q = t|_{p.q}$ はリデックスでない．よって $t|_p$ は μ-正規形である． □

定理 3.11 任意の $f \in \Sigma$ に対して f が $\mu(f)$ 基底可簡約または自明のとき，μ は基底正当である．

証明. 任意の基底 μ-正規形が R-正規形であることを項の構造帰納法で示す．任意の基底 μ-正規形が $\mu(f)$ 基底可簡約演算子を含まないことを示せば十分である．$t \in T(\Sigma)$ が $\mu(f)$ 基底可簡約演算子を持つと仮定する．p を $root(t|_p)$ が $\mu(f)$ 基底可簡約演算子 f となる最小のポジションとすると，$p \in O_\mu(t)$ である．定義より $i \in \mu(f)$ ならば $p.i \in O_\mu(t)$ なので，補題 3.10 より任意の $i \in \mu(f)$ に対して $t|_{p.i}$ は μ-正規形である．帰納法の仮定より $t|_{p.i}$ は R-正規形である．定理 3.9 より $t|_p$ はリデックス．$t \in NF_\mu$ と矛盾． □

例 3.12 例 2.1 の TRS R_n を再び考える．演算子 $+$ は $\{2\}$ 基底可簡約演算子である．$\mu(+) = \{2\}$, $\mu(s) = \{1\}$ ならば定理 3.11 より μ は基底正当． □

3.4 ソート可簡約演算子

この節では，あるソートの集合に属する文脈のもとでの可簡約演算子を提案する．多ソート TRS の諸定義・表記を与える．ソートの集合は有限集合とする．たとえば例 2.1 の TRS R_n に対する $S_n = \{Nat\}$，例 3.1 の TRS R_l に対する $S_l = \{Elt, List\}$ はソートの集合の例である．ソートの集合 S に対して S 上の演算子は，$w \in S^*$ かつ $s \in S$ に対し，$f : w \to s$ のように記述される．定数 $c : \varepsilon \to s$ は，単に $c : s$ と書く．変数の集合は $V = \bigcup_{s \in S} V_s$ で与えられる．各 V_s はソート s を持つ変数の可算集合である．$x \in V_s$ のとき $x : s$ と書く．ソート $s \in S$ を持つ項の集合 $T(\Sigma, V)_s$（または単に T_s）は以下を満たす最小の集合である：$V_s \subset T_s$ かつ任意の演算子 $f : \overrightarrow{s_i} \to s \in \Sigma$ および項 $t_1 \in T_{s_1}, \ldots, t_n \in T_{s_n}$ に対して $f(\overrightarrow{t_i}) \in T_s$．ソートの集合 $S' \subset S$ に対し，$T(\Sigma, V)_{S'}$（または $T_{S'}$）を $T(\Sigma, V)_{S'} = \bigcup_{s \in S'} T(\Sigma, V)_s$ と定義する．$t \in T_s$ のとき，$t : s$ と書く．$sort(t)$ で t のソートを表す．多ソート TRS は，ソートの集合 S，S 上の演算子の集合 Σ，$sort(l) = sort(r)$ を満たす書き換え規則 $l \to r$ の集合 R の 3 つ組 (S, Σ, R) である．代入などの諸定義は TRS の定義から自然に拡張される．以下では多ソート TRS (S, Σ, R) を R と書いたり単に TRS と呼ぶことがある☆．

例 3.13 $S_z = \{Nat, Bool\}$

$$\Sigma_z = \begin{cases} 0 : Nat \\ s : Nat \to Nat \\ + : Nat, Nat \to Nat \\ true, false : Bool \\ and : Bool, Bool \to Bool \\ zero : Nat \to Bool \end{cases}$$

$$R_z = R_n \cup \begin{cases} and(true, x) \to x \\ and(false, x) \to false \\ zero(0) \to true \\ zero(s(x)) \to false \\ zero(+(x, y)) \to and(zero(x), zero(y)) \end{cases}$$

☆ 本論文の結果は順序ソート TRS に対しても自然に適用できる．

このとき (S_z, Σ_z, R_z) は多ソート TRS である． □

TRS R_z において演算子 $zero$ は基底可簡約である．一方，$Bool$ ソートの項のみを書き換え対象の項と限定すると演算子 0，s，$+$ が可簡約となる．項 0，$s(t)$，$+(t,t')$ 自体はリデックスではないが，文脈 $c : Nat \to Bool$ に対して項 $c[0]$，$c[s(t)]$，$c[+(t,t')]$ はリデックスを含む．ここで文脈 $c : s_1 \to s_2$ とは，$c : s_2$ かつ $\Box : s_1$ となる文脈である．実際にソート $Bool$ の項中に 0，s，$+$ があれば，必ず $zero(0)$，$zero(s(x))$，$zero(+(x,y))$ のいずれかのリデックスパターンが存在する．このとき $\mu(0) = \mu(s) = \mu(+) = \emptyset$ としてもソート $Bool$ の項に限り，R-正規形と μ-正規形の集合は一致する．

定義 3.14 μ が $S'(\subset S)$ **のもとで正当**である $\overset{\text{def}}{\iff} NF_\mu \cap T_{S'} \subset NF_R$． □

定義 3.15 集合 $S'(\subset S)$ に対し f は S' **可簡約**である $\overset{\text{def}}{\iff}$ f を含む $sort(t) \in S'$ の任意の項 t が可簡約である． □

項 t を受け取り書き換え可能でないポジションをそれぞれ異なる変数に置き換える関数 cut_μ を定義する．変数 x に対して $cut'_\mu(x) = x$ かつ $cut'_\mu(f(\overrightarrow{t_i})) = f(\overrightarrow{t'_i})$ ただし t'_i は $i \in \mu(f)$ ならば $cut(t_i)$，$i \notin \mu(f)$ ならば定数 □ とする．$cut_\mu(t)$ は $cut'_\mu(t)$ のすべての □ を相異なる変数で置き換えた項である．

補題 3.16 t が μ-正規形ならば $cut_\mu(t)$ は R-正規形である．

証明. $t' = cut_\mu(t)$ が R-正規形でないと仮定する．定義より $t'\theta = t$ を満たす θ が存在する．ポジション p を項 t' のリデックスのポジションとする．すなわちある代入 θ' と規則 $l \to r \in R$ が存在して $t'|_p = l\theta'$ である．すべての書き換え可能でないポジションは変数に置き換えられており，変数はリデックスにはなりえないので $p \in O_\mu(t')$ である．以下の等式が成り立つ．$t|_p = t'\theta|_p = (t'|_p)\theta = (l\theta')\theta = l(\theta;\theta')$．代入の合成は代入であり，代入の適用は書き換え可能ポジションを保存するので，t はポジション $p \in O_\mu(t)$ でリデックスを持つ，すなわち μ-正規形でない． □

定理 3.17 任意の $f \in \Sigma$ に対して f が S' 可簡約または自明のとき μ は S' のもとで正当である．

証明. 任意のソート $s \in S'$ の μ-正規形が S' 可簡約演算子を含まないことを示せば十分である．$t \in T(\Sigma, V)_{S'}$ が S' 可簡約演算子を持つと仮定する．p を $root(t|_p)$ が S' 可簡約演算子 f となる最小のポジションとすると，$p \in O_\mu(t)$ である．関数 cut_μ の定義より $cut_\mu(t)$ は f を持つ．$sort(cut_\mu(t)) \in S'$ なので $cut_\mu(t)$ は可簡約項である．補題 3.16 より t は μ-正規形ではない． □

例 3.18 TRS R_z において演算子 0，s，$+$ は $\{Bool\}$ 可簡約である．$\mu(0) = \mu(s) = \mu(+) = \emptyset$ かつそれ以外の演算子 f が自明ならば，定理 3.17 より μ は $\{Bool\}$ のもとで正当である．また $zero$，and は $\{1\}$ 基底可簡約である．したがって $Bool$ ソートの基底正規形にはこれらの演算子が現れない．すなわち $true$ または $false$ である． □

例 3.19 例 3.1 の R_l，$S_l = \{Nat, List\}$，$\Sigma_l = \{hd : List \to Nat,\ tl : List \to List,\ cons : Nat\ List \to List,\ zero : List,\ 0 : Nat\}$ に対して (S_l, Σ_l, R_l) は多ソート TRS である．演算子 $cons$ は $\{Nat\}$ 可簡約である．$\mu(cons) = \emptyset$ かつ hd，tl が自明ならば，定理 3.17 より μ は $\{Nat\}$ のもとで正当である．TRS R_l はこの μ のもとで μ-停止性を持つ[7]．すなわち無限簡約列 $t_1 \to_\mu t_2 \to_\mu \cdots$ を持たない．本結果と合わせると，TRS R_l で任意のソート Nat の項の R-正規形が有限時間内に得られることが保証される． □

3.5 基底ソート可簡約演算子

I 基底可簡約と S' 可簡約を合わせた I 基底 S' 可簡約演算子を定義する．

定義 3.20 μ が $S'(\subset S)$ **のもとで基底正当**である $\overset{\text{def}}{\iff} NF_\mu \cap T(\Sigma) \cap T_{S'} \subset NF_R$． □

定義 3.21 集合 I $(\subset \mathcal{N})$，$S'(\subset S)$ に対し f は I **基底** S' **可簡約**である $\overset{\text{def}}{\iff}$ $i \in I$ で t_i が基底項である部分項 $f(\overrightarrow{t_i})$ を持つ任意の $sort(t) \in S'$ の項は可簡約である． □

定理 3.22 任意の $f \in \Sigma$ に対して f が I 基底 S' 可簡約または自明のとき μ は S' のもとで基底正当である．

証明. 定理 3.11 および 3.17 の証明と同様． □

例 3.23 自然数を保持するセルを表す TRS を考える．put により自然数をセルに挿入し，$zero$ によりセルに保持されている自然数が 0 かどうかをチェックする．$S_c = \{Nat, Cell, Bool\}$，$\Sigma_c = \{0 : Nat,\ s : Nat \to Nat,\ empty : Cell,\ put : Nat, Cell \to Cell,\ zero : Cell \to Bool\}$，

$$R_c = \begin{cases} zero(empty) \to false \\ zero(put(0,x)) \to true \\ zero(put(s(x),y)) \to false \end{cases}$$

put は基底可簡約でも $\{Bool\}$ 可簡約でもないが，$\{1\}$ 基底 $\{Bool\}$ 可簡約である．よって定理 3.22 より $\mu(put) = \{1\}$ で他の演算子が自明のとき，μ は $\{Bool\}$ のもとで基底正当である． □

3.6 可簡約性の判定法

演算子の可簡約性の判定可能な必要十分条件を与える．アルゴリズムを定義するためにいくつかの表記および関数を用意する．

定義 3.24 項 t の高さ $h(t)$ を変数 x に対して $h(x) = 1$ かつ定数 c に対して $h(c) = 1$ かつ $h(f(\overrightarrow{t_i})) = 1 + max\{h(t_i) \mid 1 \leq i \leq n\}$ と定義する．TRS R の高さ h_R を $max\{h(l) \mid l \in L(R)\}$ で定義する．ポジション p の長さを $|p|$ で表す $(h(t)=max\{|p| \in \mathcal{N} \mid p \in O(t)\}+1)$． □

基底項を $i+1$ の高さに枝刈りする関数 cut_i を定義する．切り口は異なる変数である．たとえば $t = s(s(+(s(s(0)), s(0))))$ に対して $cut_4(t) = s(s(+(s(x), s(y))))$.

定義 3.25 関数 cut_i を以下のように定義する．

$$\begin{aligned} cut'_0(t) &= \Box, \\ cut'_{i+1}(t) &= t && \text{if } h(t) \leq i+1, \\ cut'_{i+1}(f(\overrightarrow{t_j})) &= f(\overrightarrow{cut_i(t_j)}) && \text{if } h(t) > i+1. \end{aligned}$$

$cut_i(t)$ は $cut'_i(t)$ に現れる □ を互いに異なる変数で置き換えた項である． □

定理 3.26 多ソート TRS (S, Σ, R)，ソートの部分集合 $S' \subset S$，置換写像 μ とする．
演算子の部分集合 $D \subset \Sigma$ （$\overline{D} = \Sigma \setminus D$ と書く）に対して項の部分集合 T' を以下で定義する．

$$A \stackrel{def}{=} \left\{ cut_{h_R}(t) \in T(\overline{D}, V) \mid t \in T(\overline{D}) \cap NF_R \right\}$$

$$B \stackrel{def}{=} \left\{ \begin{array}{l} t[\Box]_p \in \\ T(\overline{D} \cup \{\Box\}) \end{array} \middle| \begin{array}{c} t \in NF_R, p \in O(t) \\ sort(t) \in S' \end{array} \right\}$$

$$C \stackrel{def}{=} \left\{ \begin{array}{l} \\ cut_{h_R}(t') \in \\ T(\overline{D} \cup \{\Box\}) \\ \\ \end{array} \middle| \begin{array}{c} t[\Box]_p \in B, \text{ and} \\ 1.\ |p|+1 < h_R, t' = t, \text{ or} \\ 2.\ p = q.q', |q'|+1 = h_R, \\ t' = t|_q[\Box]_{q'} \end{array} \right\}$$

$$T' \stackrel{def}{=} \left\{ t[f(\overrightarrow{t_i})] \in T \middle| \begin{array}{c} t[\Box] \in C, f \in D, \\ \forall i \in \mu(f).t_i \in A, \\ \forall i \notin \mu(f).t_i = x_i \in V \end{array} \right\}$$

このとき，任意の $t \in T'$ は可簡約項である $\Longleftrightarrow$ 任意の $f \in D$ は $\mu(f)$ 基底 S' 可簡約演算子である．

証明． $(\Rightarrow:)$ 任意の $t[\Box] \in T(\Sigma \cup \{\Box\} \setminus D, V)$, $\overrightarrow{t_i}$ に対して，$sort(t) \in S'$ かつ任意の $i \in \mu(f)$ で t_i が基底項ならば，$s = t[f(\overrightarrow{t_i})]$ は可簡約項であることを示せば，f は $\mu(f)$ 基底 S' 可簡約演算子である．集合 B から C を得るように $t[\Box]$ から部分項 $t'[\Box]$ を得る．項 s_i を $i \in \mu(f)$ に対して $s_i = cut_{h_R}(t_i)$，かつ $i \notin \mu(f)$ に対して $s_i = x_i$ と定義する．T' の作り方から $s' = t'[f(\overrightarrow{s_i})] \in T'$ である．s' は可簡約項であるので，ある規則 $l \to r \in R$ が存在し，$l\theta = s'|_p$ である．明らかに $s|_p$ も l のインスタンスである．よって s は可簡約項である．
$(\Leftarrow:)$ $t[f(\overrightarrow{t_i})]_p \in T'$ とする $(f \in D)$．すべての変数にある基底正規形を割り当てる代入を θ とする．ただし割当てはそれぞれ十分な異なる高さを持つ基底正規形とする（異なる x と y に対し $|h(\theta(x)) - h(\theta(y))| > h_R$）．$C$ より $sort(t') \in S'$ で $t'[t[\Box]_p]$ を R-正規形とする文脈 $t' \in T(\overline{D} \cup \Box)$ が存在する（C の定義内の表記で，1. のときは □，2. のときは $t[\Box]_q$）．f の可簡約性より $t'[t[f(\overrightarrow{t_i})\theta]]$ はリデックスを持つ．$t'[t[\Box]_p]$ は R-正規形であることからリデックスは f を含まなければならない．$t' = \Box$（C の 1）の場合は自明に，$t' \neq \Box$（C の 2）の場合は $t[\Box]_p$ に対して $|p|+1 = h_R$ が成り立つため，$t[f(\overrightarrow{t_i})\theta]$ はリデックスを持つ．A より各 t_i に現れる変数のポジションの長さは h_R より大きい．よって θ のとり方から代入前の $t[f(\overrightarrow{t_i})]$ もリデックスを持つ． □

ソートおよび演算子の数を有限と仮定する．部分集合 $D \subset \Sigma$，置換写像 μ の数は有限であるため，集合 T' は有限集合である．したがってある演算子の集合に対して各要素 f が $\mu(f)$ 基底 S' 可簡約演算子であるかどうかは決定可能である．

4. 代数仕様への応用

TRS は等式推論を効果的に行うための基礎理論である．TRS R は各書き換え規則 $l \to r$ を等式 $l = r$ と見ることで代数仕様（等式仕様）と見なすことができる．OBJ 言語族に基づく代数仕様の意味論について簡単に説明する．詳細は文献 2), 9) を参照．

仕様 $SP = (S, \Sigma, R)$ のモデルとは，各ソート $s \in S$ に対し台集合 M_s，各定数 $c : s \in \Sigma$ に対し要素 $M_c \in M_s$，各演算子 $f : w \to s \in \Sigma$ に対し関数 $M_f \in M_w \to M_s$ が存在し，任意の $l = r \in R$ で両辺が同じ要素（または関数）に解釈される代数である．基本仕様には 2 通りの意味論がある．仕様が**始代数意味論**を持つときその仕様の表示的意味 $[SP]$ は始代数すべての集合である．そうでないときは，すべてのモデルの集合が $[SP]$ となる．始代数とはその仕様のすべてのモデルへのユニークな射が存在するモデルである．**項代数**は代表的な始代数である．項の集合に対して，$(\to_R \cup \leftarrow_R)^*$ に関する同値類 $[t]$ の（商）集合を台集合とする代数が項代数である．始代数ではすべての要素に対して対応する基底項が存在するため，

検証に帰納法や場合分けなどのテクニックを用いることができる．始代数どうしは同型である．以下，始代数の集合を 1 つの始代数で代表して議論することがある．例 2.1 の TRS R_n の始代数は自然数（およびその同型）である．規則 $+(x,0) \to x$ の両辺は恒等関数に，規則 $+(x,s(y)) \to s(+(x,y))$ の両辺は関数 $f(x,y) = x+y+1$ に解釈される．始代数以外のモデルとして整数やブール代数なども考えられる．たとえば整数をモデルとした場合は，負の値に対応する項は R_n には存在しない．

輸入は，仕様の再利用，ライブラリの利用などを可能にする構造的仕様記述に有用な概念である．仕様 $SP = (S, \Sigma, R)$ が仕様 $SP' = (S', \Sigma', R')$ を輸入するとは，$S' \subset S$，$\Sigma' \subset \Sigma$，$R' \subset R$ の関係があることである．**輸入が保護されている**とき輸入した仕様の表示的意味 $[SP]$ は，以下を満たす SP のモデル M の集合である．ある $M' \in [SP']$ が存在し各ソート $s \in S'$ に対応する台集合 M'_s が M_s と一致する．SP が始代数を持つときは，それらのモデルのうち他のすべてのモデルへユニークな射が存在するモデルの集合である．たとえば，例 3.13 における R_z は R_n を輸入している．もし R_n が始代数意味論を持つならば，ソート Nat に対応するモデル $M' \in [R_n]$ の台集合 M'_{Nat} は自然数の集合であり，もし輸入が保護輸入ならば，R_z のモデル $M \in [R_z]$ において $M_{Nat} = M'_{Nat}$ である．

4.1 振舞仕様

この節では**振舞仕様**と呼ばれる代数仕様の一種を紹介する[2),4)]．振舞仕様では，システムの内部構造を隠し，振舞いのみを記述する．振舞仕様では**隠蔽ソート**と呼ばれる特殊なソートが存在する．これは記述するシステムの状態空間に相当する．隠蔽でないソートは**可視ソート**と呼ばれる．$\mathcal{H}$ と $\mathcal{V}$ でそれぞれ隠蔽ソートと可視ソートの集合を表す（$S = \mathcal{H} \cup \mathcal{V}$）．隠蔽ソートの項を**隠蔽項**，可視ソートの項を**可視項**と呼ぶ．**振舞演算**と呼ばれる特殊な演算子が存在する．振舞演算は引数にただ 1 つだけの隠蔽ソートを持たなければならない（$f : \overrightarrow{s_i} \to s$ が振舞演算 $\Rightarrow \exists! i \in \mathcal{N}.s_i \in \mathcal{H}$）[☆]．振舞演算の集合を $\mathcal{B}(\subset \Sigma)$ と記述する．振舞演算の集合 $\mathcal{B}$ はさらに**観測演算**と**操作演算**の 2 種類に分けられる．$f : \overrightarrow{s_i} \to s \in \mathcal{B}$ に対し，$s \in \mathcal{V}$ のとき観測演算，$s \in \mathcal{H}$ のとき操作演算という．システムの状態は観測演算によってのみ見ることができ，操作演算によってのみ変更できる．これらはそれぞれオブジェクト指向の用語における属性とメソッドに対応する．引数に隠蔽ソートを持つ演算子を**隠蔽演算子**と呼ぶ．振舞演算以外の隠蔽演算子を**隠蔽構成子**と呼び，その集合を $\mathcal{HC}$ で記述する．したがって $\mathcal{B} \cup \mathcal{HC}$ は隠蔽演算子すべての集合である．

振舞仕様 SP において，保護輸入され，かつ始代数意味論を持つ仕様 $SP' = \{S', \Sigma', R'\}$ に属する各ソート $s \in S'$ を**保護可視ソート**と呼び，その集合を $\mathcal{PV}$ と書く．振舞仕様は，その振舞いを満たすすべての実装の仕様であるため始代数意味論を持たない．よって隠蔽ソートは保護可視ソートではない（$\mathcal{PV} \subset \mathcal{V}$）．通常，振舞仕様で利用するデータ型（自然数，ブール値）は始代数意味論を持ち，保護輸入されることが多い．

4.2 振舞等価

ルートから □ までの間の演算子がすべて振舞演算子である文脈を**振舞文脈**と呼び，その集合を $\mathcal{BC}$ で書く．すなわち $\mathcal{BC}$ は，任意の観測演算 f に対する文脈 $o = f(\overrightarrow{s_j}, \Box, \overrightarrow{t_k})$，任意の操作演算 f に対する文脈 $a_i = f(\overrightarrow{s_j}, \Box, \overrightarrow{t_k})$ に対して $t[\Box]_p = o[a_1[a_2 \cdots a_n[\Box]]]$ と書ける文脈の集合である．ただし各項 s_j，t_k は，保護可視ソートのときは（保護輸入された仕様における）任意の基底項，そうでないときは変数である．振舞仕様における等価関係として**振舞等価関係**がある．振舞仕様 SP のモデル $M \in [SP]$ で，任意の $bc \in \mathcal{BC}$ に対し $M_{bc}(a) = M_{bc}(b)$ のとき，$a, b \in M_s$ は振舞等価であるといい，$a \sim b$ と書く[☆☆]．

振舞等価性は文脈に対して閉じていないので振舞等価のための等式推論を行うためには通常とは異なる書き換え関係を定義する必要がある．振舞仕様のための書き換え関係 $\hookrightarrow_R$ を以下で定義する．

$$s \hookrightarrow_R t \overset{\text{def}}{\iff} \exists p \in O(s).s \to_p t \text{ and,} \begin{cases} s|_p \in T_{\mathcal{V}} \text{ or} \\ s|_p \in T_{\mathcal{H}}, bc \in \mathcal{BC}, s = s'[bc[s|_p]] \end{cases}$$

書き換え関係 $\hookrightarrow_R$ に関する正規形の集合を BNF_R と書く．また振舞仕様のための文脈依存書き換えは $\hookrightarrow_\mu = \hookrightarrow_R \cap \to_\mu$ で定義され，その正規形の集合を BNF_μ と書く．$s =_R t \overset{\text{def}}{\iff} s(\hookrightarrow_R \cup \hookleftarrow_R)^* t$ とする．任意の振舞文脈 $bc \in \mathcal{BC}$ で $bc(t_1) =_R bc(t_2)$ ならば任意の $M \in [SP]$ で M_{t_1}，M_{t_2} は振舞等価である．

例 4.1 隠蔽ソート $\mathcal{H}_{nl} = \{List\}$，可視ソート $\mathcal{V}_{nl} = \{Nat\}$，振舞演算の集合 $\mathcal{B}_{nl} = \{hd, tl, cons\}$ とすると例 3.19 の TRS R_l は振舞仕様である．

[☆] 唯一の隠蔽ソート引数を持っていても振舞演算である必要はない．

[☆☆] $\mathcal{BC}$ は可視空文脈 □ を含むので $\sim$ は通常の等価関係を含む．

文脈 $hd(cons(s(0), tl(cons(0, \Box))))$ は振舞文脈である．□

例 4.2 自然数を要素とする無限長の配列の振舞仕様を与える．システムの仕様の前に自然数などの抽象データ型を以下で改めて定義する．自然数の仕様 $\mathcal{S}_n = (\{Nat\}, \{0 : Nat, s : Nat \to Nat\}, \emptyset)$，ブール値の仕様 $\mathcal{S}_b = (\{Bool\}, \{true : Bool, false : Bool\}, \emptyset)$ は，それぞれ始代数意味論を持つとする．自然数，ブール値におけるいくつかの関数を取り入れた抽象データ型の仕様 $\mathcal{S}_{ne} = (S_{ne}, \Sigma_{ne}, R_{ne})$ を与える．$S_{ne} = S_n \cup S_b$, $\Sigma_{ne} = \Sigma_n \cup \Sigma_b \cup \{eq : Nat, Nat \to Bool,\ if : Bool, Nat, Nat \to Nat\}$,

$$R_{ne} = \begin{cases} eq(0,0) \to true \\ eq(s(x),0) \to false \\ eq(0,s(x)) \to false \\ eq(s(x),s(y)) \to eq(x,y) \\ if(true,x,y) \to x \\ if(false,x,y) \to y \end{cases}$$

ここで $\mathcal{S}_{ne}$ は $\mathcal{S}_n$ および $\mathcal{S}_b$ を保護輸入し，始代数意味論を持つとする．Nat の基底項 s，t に対して $eq(s,t) \to_R^* true \Longleftrightarrow s =_R t$ が成り立つ．

仕様 $\mathcal{S}_{ne}$ を保護輸入した配列の振舞仕様を与える．可視ソート $\mathcal{V}_{ar} = S_{ne}$，隠蔽ソート $\mathcal{H}_{ar} = \{Array\}$，振舞演算の集合 $\mathcal{B}_{ar} = \{val : Nat, Array \to Nat,\ put : Nat, Nat, Array \to Array\}$，演算子の集合 $\Sigma_{ar} = \mathcal{B}_{ar} \cup \Sigma_{ne} \cup \{init : Array\}$,

$$R_{ar} = R_{ne} \cup \begin{cases} val(i, init) = 0 \\ val(i, put(j,x,y)) = \\ if(eq(i,j), x, val(i,y)). \end{cases}$$

R_{ar} における保護可視ソートの集合は $\mathcal{PV} = \{Bool, Nat\}$ である．隠蔽ソート $Array$ の項は可算個のセルを持つ配列に対応する．各セルは自然数 $0, 1, 2, \ldots$ でインデックス付けされている．配列 A に対して，$val(i, A)$ は配列 A の i 番目のセルの要素（自然数）を意味する．$put(i, x, A)$ は i 番目のセルに自然数 x を入れる操作をした後の配列を意味する．$init$ は要素すべてが 0 の配列である．

ここで $t = put(0, 0, init)$ とすると $t =_R init$ ではないが $t \sim init$ が成り立つ．実際，振舞文脈 $c = val(i, put(j_n, x_n, \cdots put(j_1, x_1, \Box)))$ の n に関する帰納法により $c(t) =_R c(init)$ が導ける．

基底段階：$i =_R 0$ および $i =_R s(t')$ のどちらの場合も $val(i,t) =_R 0 =_R val(i, init)$ ☆．

帰納段階：$i =_R j_n$ のとき，$c(t) =_R x_n =_R c(init)$．$i \neq_R j_n$ のとき，

$c(t) =_R val(i, put(j_{n-1}, \cdots put(j_1, x_1, t)))$
$=_R val(i, put(j_{n-1}, \cdots put(j_1, x_1, init)))$
$=_R c(init)$．□

4.3 振舞可簡約演算子

振舞仕様における隠蔽演算子（振舞演算あるいは隠蔽構成子）の可簡約を考える．振舞仕様では隠蔽項同士の等価性は振舞等価を用いて比較される．振舞等価の判定は任意の振舞文脈に対して行われる．振舞文脈は可視項である．よって振舞仕様における TRS による等式推論では，書き換えの対象となる項は可視項であるといえる．次に隠蔽演算子の引数について考える．任意の保護可視ソートの要素に対して対応する基底項が存在する．したがって振舞仕様における演算子が関数として定義されているためには，保護可視ソートに対してはすべての基底項上でそれ以外の隠蔽ソートを含むソートに対してはすべての項（変数）上で定義されている必要がある．以上をふまえて振舞可簡約演算子を定義する．

注 4.3 振舞仕様について議論するときは，置換写像 μ は任意の隠蔽演算子 $f : \overrightarrow{s_i} \to s$ に対して $i \in \mu(f)$ ならば $s_i \in \mathcal{PV}$ を満たすとする．□

定義 4.4 f は**振舞可簡約**である $\overset{\text{def}}{\Longleftrightarrow}$ f は $\mu(f)$ 基底 $\mathcal{V}$ 可簡約である．□

定理 3.22 より振舞可簡約演算子以外の演算子 f が自明ならば μ は可視ソートのもとで基底正当である．しかし振舞仕様における検証では，通常は任意のシステムに対する性質を示すため，隠蔽ソートの変数（あるいは新たに宣言された任意の要素を表す定数）を含む項が書き換え対象となることが多い．振舞等価を示すための振舞文脈を考える場合，保護可視ソートの引数に対してはすべての基底項を考えれば十分だが，隠蔽ソートを含む非保護可視ソートに対しては変数を考えなければならない．以上より，振舞仕様では書き換えの対象となる項として保護可視ソート以外の変数を持つ項を考える必要がある．振舞仕様における正当性を以下のように定義する．

定義 4.5 振舞仕様 $(\mathcal{H} \cup \mathcal{V}, \Sigma, R)$，置換写像 μ に対して $BNF_\mu \cap T(\Sigma, \cup_{s \notin \mathcal{PV}} V_s) \cap T_{\mathcal{V}} \subset NF_R$ のとき μ を**振舞正当**であるという．□

定理 3.11 および 3.17 と同様に，振舞可簡約演算

☆ i は Nat ソートの項であり，Nat は保護可視ソートである（保護輸入は推移的）．よって，$i = 0$ または $i = s(t')$ の 2 通りとしてよい．一般に $t =_R u$ と $t \neq_R u$ の場合分けを行える．

子以外の演算子が自明ならば μ は振舞正当であるという性質を示したい．しかしこの定理は一般には成り立たない．定理 3.11 および 3.17 の証明のメインは，可簡約演算子が μ-正規形に含まれないことの証明であったが，隠蔽ソートの変数のみを持つ可視ソートの μ-正規形で振舞可簡約演算子を持つ項が存在する．

反例 4.6 自然数を保持するシステムの振舞仕様を考える．$\mathcal{V}_{add} = S_n$，$\mathcal{H}_{add} = \{H\}$，$\mathcal{B}_{add} = \{val : H \to Nat,\ add : Nat, H \to H\}$，$\Sigma_{add} = \mathcal{B}_{add} \cup \{0 : Nat, s : Nat \to Nat\}$，$R_{add} = \{val(add(0,x)) \to val(x),\ val(add(s(i),x)) \to s(val(add(i,x)))\}$．このとき add は $\{1\}$ 基底 $\{Nat\}$ 可簡約なので振舞可簡約であるが，隠蔽変数 $x, y \in V_H$ のみを持つ可視正規形 $val(add(val(x), y))$ に出現する． □

振舞正当性の十分条件を与えるため，隠蔽ソートの入れ子という概念を定義する．

定義 4.7 項に現れる隠蔽演算子の保護可視ソートの引数以下に非保護可視ソートの部分項が現れないとき，（**保護可視ソートの**）**入れ子がない**という．すなわち $t = t[f(\overrightarrow{t_i})]$ かつ $f : \overrightarrow{s_i} \to s \in \mathcal{B} \cup \mathcal{HC}$ かつ $s_i \in \mathcal{PV}$ ならば $\forall p \in O(t_i).t_i|_p \in T_{\mathcal{PV}}$ である．TRS の各両辺が入れ子なしの項であり，右辺が非保護可視ソートの部分項を持つならば左辺も非保護可視ソートの部分項を持つとき☆，TRS に入れ子がないと定義する．以下「**入れ子なしの仮定**」というときは書き換えようとする項および TRS に入れ子がないことを仮定する☆☆． □

上記の反例 $val(add(val(x), y))$ では集合 $\{val, add\}$ が隠蔽演算子の集合（振舞演算の集合と一致）である．add の 1 引数目は保護可視ソート Nat であるが部分項 $val(x)$ は保護可視ソートでない部分項 x を持つ．よってこの項には入れ子がある．入れ子なしの仮定は振舞仕様において不自然な仮定ではない．多くの実用的な振舞仕様は入れ子なしの仮定を満たしている．

定理 4.8 隠蔽置換写像 μ に対して，入れ子なしの仮定のもとで，(1) 任意の隠蔽構成子 $f \in \mathcal{HC}$ が振舞可簡約で (2) それ以外の演算子 $f \in \Sigma \setminus \mathcal{HC}$ が振舞可簡約または自明のとき μ は振舞正当である．

証明． 定理 3.11 および 3.17 の証明と同様．振舞可簡約演算子が $BNF_\mu \cap T(\Sigma, \cup_{s \notin \mathcal{PV}} V_s) \cap T_{\mathcal{V}}$ の項に現れないことを示す際，保護可視ソート以外の変数の出現と振舞文脈書き換え $\hookrightarrow_\mu$ の正規形に注意する．振舞正当の証明の際にチェックする項は入れ子なしの仮定より隠蔽演算子の保護可視ソートの下に保護可視ソート以外の部分項は現れない，すなわち変数は現れない．証明は，振舞可簡約演算子の最小ポジションの出現を考える．隠蔽構成子はすべて振舞可簡約演算子という仮定より，最小ポジションの振舞可簡約演算子は振舞文脈下にあるはずである．また最終的に振舞可簡約演算子がないことが示せると項に隠蔽構成子が出現しないことになる．よってすべての隠蔽ソートの部分項は振舞文脈下に存在し，通常の正規形 NF_R と振舞書き換えの正規形 BNF_R は一致する． □

4.4 合同関係とコヒーレント

定義より，等価関係 $\sim$ は反射律 $a \sim a$，対称律 $a \sim b \Rightarrow b \sim a$，推移律 $a \sim b, b \sim c \Rightarrow a \sim c$ を満たす．しかし合同関係 $\overrightarrow{a_i \sim b_i} \Rightarrow M_f(\overrightarrow{a_i}) \sim M_f(\overrightarrow{b_i})$ は，一般には成り立たない．このため $\to_R$ と $\hookrightarrow_R$ に差が生じる．隠蔽ソートの集合 $\mathcal{H} = \{H\}$，可視ソートの集合 $\mathcal{V} = \{V\}$，振舞演算の集合 $\mathcal{B} = \{val : H \to V\}$，$\Sigma = \mathcal{B} \cup \{f : H \to H, a : H, b : H\}$，$R = \{a \to b\}$ という振舞仕様を考える．振舞演算は観測演算 val のみである．項 $f(a)$ はリデックスを含み $f(a) \to_R f(b)$ だが $f(a) \hookrightarrow_R f(b)$ は成り立たない．$\hookrightarrow_R$ は書き換え規則の適用の際，リデックスが振舞文脈下にあることを確かめなければならず，効率的な実装が容易でない．

演算子 f がモデル $M \in [SP]$ において $\overrightarrow{a_i \sim b_i}$ ならば $M_f(\overrightarrow{a_i}) \sim M_f(\overrightarrow{b_i})$ を満たすときモデル $M \in [SP]$ において**コヒーレント**であるという[4]．すべてのモデルでコヒーレントであるとき，単にコヒーレントであるという．すべての演算子がコヒーレントならば合同関係が成り立つ．隠蔽ソートを引数に持たない演算子および振舞演算は定義から明らかにコヒーレントである．したがってコヒーレントかどうかをチェックする対象は，振舞演算以外の隠蔽演算子である隠蔽構成子となる．文献 3) には，すべての隠蔽構成子は意味的も操作的にもコヒーレントであるべきであるという言及がある．コヒーレントでない演算子の存在は，$\to_R$ による等式推論ができないだけでなく，通常は意味のあるモデルを持たない．すなわちコヒーレントでない演算子の出現は仕様記述のミスであることが多い．よってコヒーレント証明の目的は，正しい（意図どおりの）仕様が書けているかどうかのチェックでもある．またもう 1 つの目的は，振舞等価性の証明作業の軽減である．

例 4.9 例 4.2 の TRS R_{ar} に 2 つの配列の対応するセルの要素を足し合わせた配列を作成する演算子 $U : Array, Array \to Array$ を定義する．この演算子は引数に 2 つ以上の隠蔽ソートを持つため振舞演算に

☆ 書き換えによって入れ子が出現することを防ぐため．
☆☆ よって任意の書き換え列に入れ子のある項は出現しない．

はなれない．したがって合同関係のためには U のコヒーレントを証明する必要がある．U を定義する書き換え規則を $val(i,U(x,y)) \to +(val(i,x),val(i,y))$ と定義するとコヒーレントとなる． □

例 4.10 再び TRS R_{ar} を考える．今 $\mathcal{B}=\{val\}$ とし，put は振舞演算でないとする．すると，考えなければいけなかった振舞文脈 $val(i,put(i_1,\ldots put(i_n,\Box)))$ が $val(i,\Box)$ のみなるため，振舞等価性の証明は容易になる．put は隠蔽構成子なので適切な仕様であるためにはコヒーレントを証明する必要がある． □

4.5 コヒーレントの十分条件

振舞仕様 $\mathcal{SP}=(S,\Sigma,R)$ に対して演算子 $f: V,H \to H$ のコヒーレントの証明，すなわち $\forall x,y \in M_H.\forall n \in M_V.\ x \sim y \Rightarrow M_f(n,x) \sim M_f(n,y)$ の証明を TRS を用いて行う手法を紹介する[4]．まず前提部分の $a \sim b$ を仮定した証明用の振舞仕様を定義する．Σ に現れない 2 つの隠蔽ソートの定数を加えた $\Sigma'=\Sigma\cup\{a,b\}$，それらが振舞等価であるという公理を加えた $R'=R\cup\{a\to b\}$ に対する振舞仕様 $\mathcal{SP}'=(S,\Sigma',R')$ を定義する．すると任意のモデル $M \in [SP]$，$x \sim y$ を満たす任意の $x,y \in M_H$ に対して，$\forall s \in S.M'_s=M_s$ かつ $M'_a=x$ かつ $M'_b=y$ を満たすモデル $M' \in [SP']$ が存在する．次に任意の振舞文脈 $bc \in \mathcal{BC}$ および V が保護可視ソートのときは任意の基底項 t，そうでないときは変数 t に対して $bc[f(t,a)] =_{R'} bc[f(t,b)]$ を示す．これは任意のモデル $M' \in [SP']$ において $M'_f(M'_t,M'_a)=M'_{f(t,a)} \sim M'_{f(t,b)}=M'_f(M'_t,M'_b)$ が成り立つことを保証する．t は任意の基底項（あるいは変数）なので，任意の要素 $n \in M'_V(=M_V)$ に対して $M'_t=n$ を満たす t が存在する．以上より，$\forall x,y \in M_H.\forall n \in M_v.\ x \sim y \Rightarrow M_f(n,x) \sim M_f(n,y)$ の証明が行われたことになり，f はコヒーレントである．詳しくは文献 2)〜4) などを参照．以下の命題はこの手法の一般形である．

命題 4.11 振舞仕様 (S,Σ,R) に対して $f: \overrightarrow{s_i},\overrightarrow{s'_j},\overrightarrow{s''_k} \to s \in \Sigma$ とする．一般性を失わず $\overrightarrow{s_i \in \mathcal{PV}}$，$\overrightarrow{s'_j \in \mathcal{V}\setminus\mathcal{PV}}$，$\overrightarrow{s''_k \in \mathcal{H}}$ とする．各隠蔽ソート s''_k に対して（Σ に現れない）2 つの定数記号 $a_k:s''_k$ と $b_k:s''_k$ を用意し，$\Sigma'=\Sigma\cup\{\overrightarrow{a_k},\overrightarrow{b_k}\}$ および $R'=R\cup\{\overrightarrow{a_k\to b_k}\}$ とする．こうして定義された振舞仕様 (S,Σ',R') において，任意の振舞文脈 $bc \in \mathcal{BC}$，基底項 $\overrightarrow{t_i \in \mathcal{T}_{s_i}}$，変数 $\overrightarrow{x_j}$ に対して，$bc[f(\overrightarrow{t_i},\overrightarrow{x_j},\overrightarrow{a_k})] =_{R'} bc[f(\overrightarrow{t_i},\overrightarrow{x_j},\overrightarrow{b_k})]$ ならば，f はコヒーレントである． □

命題 4.11 に基づくコヒーレント証明は TRS による書き換えをベースに行われる．可簡約演算子を用いて，書き換えを用いないコヒーレントの十分条件を得る．

TRS が μ 振舞弱正規とは，すべての項 t に対して $t \hookrightarrow^*_\mu u$ となる μ-振舞正規形 $u \in BNF_\mu$ が存在することである．関数の「すべての定義域の要素を値域のある要素に対応させる」という性質の「すべての定義域の要素」の部分を可簡約演算子が，「ある要素に対応させる」の部分を弱正規性が担う．たとえば，$R=\{f(x)\to f(x)\}$ において演算子 f は可簡約であるが $f(x)$ の値は決まらないので関数であるとはいえない．もし弱正規性があれば，任意の項を可簡約演算子のない項へと書き換えることができる．

定理 4.12 振舞仕様 (S,Σ,R) に対し，TRS R が置換写像 μ に対し μ 振舞弱正規とする．入れ子なしの仮定のもとで，(1) 任意の隠蔽構成子 $f \in \mathcal{HC}$ が振舞可簡約で (2) それ以外の演算子 $f \in \Sigma\setminus\mathcal{HC}$ が振舞可簡約または自明のとき，すべての演算子はコヒーレントである．

証明. 命題 4.11 に基づいて証明する．任意の保護可視ソートの項 $\overrightarrow{t_i}$，ソート $s \in \mathcal{V}\setminus\mathcal{PV}$ の変数 $\overrightarrow{x_j}$，隠蔽ソートの変数 $\overrightarrow{y_k}$ に対し，項 $t=bc[f(\overrightarrow{t_i},\overrightarrow{x_j},\overrightarrow{y_k})]$ を考える．振舞文脈の定義および各 x_j，y_k が非保護可視ソートの変数であることから，項 t は保護可視ソートの変数は持たない．μ 振舞弱正規性より t は $t \hookrightarrow^*_\mu u$ となる正規形 $u \in BNF_\mu$ を持ち，定理 4.8 の証明より u には隠蔽構成子は現れない．項 $u[\overrightarrow{u_k}]$ を項 u に現れる隠蔽ソートの変数 y_k をそれぞれ項 u_k で置き換えた項とする．すると (1) $bc[f(\overrightarrow{t_i},\overrightarrow{x_j},\overrightarrow{a_k})] \hookrightarrow^*_\mu u[\overrightarrow{a_k}]$ および (2) $bc[f(\overrightarrow{t_i},\overrightarrow{x_j},\overrightarrow{b_k})] \hookrightarrow^*_\mu u[\overrightarrow{b_k}]$ が成り立つ．項 u は隠蔽構成子を持たず可視ソートの項であることから，項 $u[\overrightarrow{a_k}]$ の各 a_k は振舞文脈の下にある（$\forall k.\exists bc \in \mathcal{BC}.\ u[\overrightarrow{a_k}]=u'[bc[a_k]]$）．よって $a_k \to b_k \in R'$ より (3) $u[\overrightarrow{a_k}] \hookrightarrow^*_\mu u[\overrightarrow{b_k}]$ が成り立つ．以上 (1)，(2)，(3) より，$bc[f(\overrightarrow{t_i},\overrightarrow{x_j},\overrightarrow{a_k})] =_{R'} bc[f(\overrightarrow{t_i}\overrightarrow{x_j},\overrightarrow{b_k})]$ が成り立ち，命題 4.11 より f はコヒーレントである． □

例 4.13 例 4.9 の U および例 4.10 の put はいずれも振舞可簡約演算子である．なぜなら可視ソートの項のルートから U または put の出現の間には必ず val がパターン $val(i,put(\ldots))$ あるいは $val(i,U(\ldots))$ のリデックスの形で出現する．明らかに入れ子なしの仮定は満たしている．またこの TRS は停止性（強正規性）を持つ．無限の書き換え系列が存在しないため明らかに（μ 振舞）弱正規性である．よって定理 4.12 よりコヒーレントである． □

例 4.14 TRS R_{ar} に以下のような隠蔽構成子 $slide: Array \to Array$ を追加したい．配列 $slide(a)$ のあるセルの要素は配列 a における 1 つ前のセルの要素である．単純に書き換え規則 $val(s(x),slide(a)) \to$

$val(x,a)$ のみを与えたとする．すると $slide$ はコヒーレントではない．なぜなら $slide(0,a)$ の値が不定だからである．$a \sim b$ であるが $M_{slide}(0,a) \not\sim M_{slide}(0,b)$ となるモデル $M \in [SP]$ がありうる．$slide(0,a) \to 0$ という書き換え規則を追加すれば振舞可簡約演算子となり，入れ子なしの仮定および停止性を満たすので定理 4.12 よりコヒーレントである． □

5. 関連研究

5.1 文脈依存書き換え

可簡約演算子の性質として，CSR の正当性の十分条件を与えた．μ-正規形が頭部正規形となるための十分条件が提案されている[7]．頭部正規形とは，簡約によってリデックスになることがない項であり，R-正規形を求める際のサブゴールと見なすことができる（t が頭部正規形 $\overset{\text{def}}{\Longleftrightarrow} \nexists u \in Red(R).t \to_R^* u$）．これまで頭部正規形の条件は与えられていたが正当性に関する研究はあまり行われていなかった．正当性は TRS による等式推論を行う際に重要である．停止性および正規形の一意性☆を持つ TRS においては，2 項の等価性判定が決定可能であるという性質がよく知られている[14]．正当性は，停止性を持たない TRS における決定可能な等価性判定法を与える．TRS R，置換写像 μ に対し，μ が正当性を持ち，R が μ-停止性かつ正規形の一意性を持つと仮定する．μ-停止性より任意の二項 t，t' から $t \to_\mu^* u$，$t' \to_\mu^* u'$ となる μ-正規形 $u, u' \in NF_\mu$ が得られる．正当性より $u, u' \in NF_R$ である．もし $u = u'$ ならば $\to_\mu \subset \to_R$ より明らかに $t =_R t'$ である．もし $u \neq u'$ ならば，正規形の一意性より $u \neq_R u'$ よって $t \neq_R t'$ である．正当性の条件を外すことはできない．$\mu(f) = \mu(a) = \mu(b) = \emptyset$ で，TRS $R = \{a \to b\}$ のとき，項 $f(a)$，$f(b)$ はともに μ 正規形であるが，$f(a) =_R f(b)$ である．例 3.19 の多ソート TRS (S_l, Σ_l, R_l) は $\mu(cons) = \emptyset$ かつ hd, tl が自明のとき μ-停止性を持ち，$\{Nat\}$ のもとで正当であった．TRS R_l は明らかに停止性を持っていない．R_l は通常の項書き換え $\to_R$ による等価性判定に適さないが，Nat ソートの項であれば CSR の等価性判定を行うことができる．代数仕様言語 CafeOBJ [2] などで採用されている評価戦略は CSR の実装と見なすことができる[8]．正当性の十分条件により評価戦略が R-正規形を返すための十分条件が得られる．

5.2 コヒーレント

振舞仕様におけるコヒーレントの十分条件として**観測完全定義**（observer complete definitions）が提案されている[1]．演算子 f の観測完全定義とは，以下を満たす等式の集合 $\{c_1[f(\overrightarrow{x_i})] = r_1, c_2[f(\overrightarrow{x_i})] = r_2, \ldots, c_k[f(\overrightarrow{x_i})] = r_k\}$ である：(1) $\{c_1, \ldots, c_k\}$ は完全観測文脈である☆☆．(2) 各 x_i は変数．(3) 各 r_j は隠蔽演算子として f か観測演算のみを持つ．(4) 単調で無限下降列のない文脈上の順序 $>$ が存在し任意の r_j の部分項 $c[f(\overrightarrow{t_i})]$ に対して $c_i > c$ である☆☆☆．

仕様が演算子 f の観測完全定義を含むならば，f はコヒーレントであるという性質が示されている[1]．

本研究のコヒーレントの十分条件と比較すると，条件 (1)，(2) は振舞可簡約演算子，条件 (3)，(4) は μ 振舞弱正規性に対応する．観測完全定義はシンタクティックな制約が大きい．顕著な差は，左辺の f の引数が変数に限られることと，右辺に関する制限である．書き換え規則（等式）$val(0, slide(x)) \to 0$ と $val(s(x), slide(y)) \to val(x, y)$ は引数に基底項があるため観測完全定義ではない．また $put12(x) \to put(1, put(2, x))$ などのような既存の隠蔽演算子を組み合わせた新たな演算子の定義も，右辺に観測および $put12$ 自身以外の隠蔽演算子が現れるため，観測完全定義ではない．定理 4.12 の μ 振舞弱正規性の条件は，条件 (3)，(4) を完全には含まないが，本質的には包含する．実際，定理 4.12 の条件を条件 (3)，(4) を含むように弱めることは可能である（任意の振舞文脈 $bc \in \mathcal{BC}$ に対して項 $bc(f(\ldots))$ は $\hookrightarrow_\mu$-正規形を持つ，など）．よって観測完全定義は本提案のコヒーレント判定条件の特殊ケースといえる．

5.3 観測遷移機械

観測遷移機械（OTS）は振舞仕様の特殊形である．無限状態を持つ遷移機械を記述するのに優れており，相互排他アルゴリズムの記述と検証[10),11] やセキュアプロトコルの記述と検証[12),13] など多くの分散システムへの応用で成果をあげている．OTS は，振舞演算として観測演算のみを持ち，すべての隠蔽演算子の隠蔽ソート引数はただ 1 つであり，演算子がすべてコヒーレントである振舞仕様として定義できる．たとえば，R_{ar} で put を隠蔽構成子とした仕様やそれに $slide$ を加えた仕様などは OTS である．したがって OTS を記述する際には，まず振舞演算として観測演算のみを記述し，それ以外の演算子のコヒーレント，実際には遷移規則を定義するための隠蔽構成子（put，$slide$ など）のコヒーレントを示す必要がある．入れ子なしの

☆ $u, u' \in NF_R, u =_R u' \Rightarrow u = u'$ という性質

☆☆ C が完全観測文脈 $\Longleftrightarrow \forall bc \in \mathcal{BC}.\ \exists c_i \in C.\ bc = c[c_i]$

☆☆☆ $>$ が単調 $\Longleftrightarrow c_1 > c_2$ ならば $c[c_1] > c[c_2]$

仮定および振舞可簡約演算子かどうかのチェックは決定可能であるので，停止性の証明器[15),16)]と組み合わせることで振舞仕様が OTS であることの証明を行うツールを作成することが可能である．

6. まとめ

μ 振舞弱正規性以外の定理 4.12 の仮定はすべて決定可能である．μ 停止性は，μ 振舞弱正規性の十分条件の 1 つであり数多くの停止性判定手法が提案されている．μ 停止性判定手法と我々の可簡約演算子の検査手法（定理 3.26）を組み合わせることで，我々は，演算子が（全域）関数として適切に定義されているかどうか，演算子がコヒーレントかどうか，振舞仕様が OTS であるかどうかを確かめる判定手法を得ることができる．本研究では，書き換え対象の項を制限することでいくつかの有用な性質を得たが，これらの制限に共通の性質は書き換えによって保存されることである．基底項やあるソートに属する項，あるソートに属する変数のみを持つ項は書き換えによって保存される．すなわち基底項を書き換えても基底項であり，ソート s の項を書き換えた項はソート s を持ち，新たな変数は書き換えによって現れない．したがって書き換えによって保存される性質であればその制限のもとでの正当性を定義し，その十分条件となる可簡約演算子を定義することが可能である．興味深い応用分野を持つそのような性質の発見は今後の課題である．

参考文献

1) Bidoit, M. and Hennicker, R.: Observer complete definitions are behaviorally coherent, *Proc. OBJ/CafeOBJ/Maude Workshop at Formal Methods'99*, Toulouse, France, pp.83–94 (1999).
2) Diaconescu, R. and Futatsugi, K.: CafeOBJ report, *World Scientific* (1998).
3) Diaconescu, R., Futatsugi, K. and Iida, S.: Component-based algebraic specification an verification in CafeOBJ, *World Congress on Formal Methods in the Development of Computing Systems*, LNCS 1708, pp.1644–1663 (1999).
4) Diaconescu, R. and Futatsugi, K.: Behavioral coherence in object-oriented algebraic specification, *Journal of Universal Computer Science*, Vol.6, No.1, pp.74–96 (2000).
5) Jouannaud, J.-P. and Kounalis, E.: Automatic proofs by induction in equational theories without constructors, *Proc. Symposium on Logic in Computer Science*, Cambridge, Massachusetts, pp.358–366, IEEE Computer Society (1986).
6) Kapur, D., Narendran, P. and Zhang, H.: On sufficient completeness and related properties of term rewriting systems, *Acta Informatica*, Vol.24, No.4, pp.395–415 (1987).
7) Lucas, S.: Context-sensitive computations in functional and functional logic programs, *Journal of Functional and Logic Programming*, 1998(1), pp.1–61 (1998).
8) Nakamura, M.: Evaluation strategy for term rewriting systems, Ph.D. thesis, Japan Advanced Institute of Science and Technology (2002).
9) 中村正樹，二木厚吉：構造的代数仕様のための等価述語の提案と実装，ソフトウェア工学の基礎 XI（FOSE2004），pp.117–128, 日本ソフトウェア科学会 (2004).
10) Ogata, K. and Futatsugi, K.: Formally modeling and verifying Ricart & Agrawala distributed mutual exclusion algorithm, *APAQS'01*, pp.357–366, IEEE CS Press (2001).
11) Ogata, K. and Futatsugi, K.: Formal analysis of Suzuki & Kasami distributed mutual exclusion algorithm, *FMOODS 2002*, pp.181–195, Kluwer Academic Publishers (2002).
12) Ogata, K. and Futatsugi, K.: Formal analysis of the *i*KP electronic payment protocols, *ISSS 2002*, LNCS 2609, pp.441–460 (2003).
13) Ogata, K. and Futatsugi, K.: Formal verification of the Horn-Preneel micropayment protocol, *VMCAI 2003*, LNCS 2575, pp.238–252 (2003).
14) TERESE: Term rewriting systems, Cambridge University Press (2003).
15) The C*i*ME Rewrite Tool. URL: cime.lri.fr
16) MU-TERM. URL: www.dsic.upv.es/~slucas/csr/termination/muterm/

(平成 16 年 9 月 30 日受付)
(平成 17 年 1 月 12 日採録)

中村　正樹（正会員）

2002 年北陸先端科学技術大学院大学情報科学研究科博士後期課程修了．博士（情報科学）．同年同大学院大学助手に就任，現在に至る．プログラム言語の基礎理論，項書き換えシステム，フォーマルメソッドに関心を持つ．日本ソフトウェア科学会会員．

緒方 和博（正会員）

1995 年慶應義塾大学大学院理工学研究科博士課程修了．博士（工学）．北陸先端科学技術大学院大学助手，SRA 先端技術研究所研究員を経て，2002 年（株）NEC ソフトウェア北陸研究エキスパート，北陸先端科学技術大学院大学客員研究員．並列・分散プログラミング（言語）およびシステム，それらの解析，解析のための基礎技術および支援ツールの開発等に関心がある．

二木 厚吉（正会員）

1975 年東北大学大学院工学研究科博士課程修了．工学博士．通商産業省工業技術院電子技術総合研究員，室長，首席研究員を経て，1993 年北陸先端科学技術大学院大学教授．1983 年から 1984 年にかけてスタンフォード研究所（SRI International）客員研究員．形式仕様言語に基づくフォーマルメソッド（Formal Methods）の研究を進め，HISP，OBJ，CafeOBJ といった先進的なシステム開発法の産業界への浸透に力を注ぐとともに，システム設計論の基礎となるべき言語設計学の確立を目指している．最近は，要求仕様や設計仕様の CafeOBJ を用いた検証法の研究を進めている．