抽象高階書換え系におけるナローイング

奥居 哲[†] 鈴木太朗[††]

一階の項書換え系におけるナローイング手続きを高階の場合に拡張することは，関数・論理型プログラミング研究における重要な問題である．これまでに，Nipkow の高階項書換え系に対するナローイングの定式化が Prehofer により行われている．その一方，van Oostrom によって導入された，より一般的な枠組みである抽象高階書換え系に対しては，いまだ，その定式化すら進んでいない．本論文は，関数・論理型プログラムの計算機構であるナローイングの，抽象高階書換え系への一般化について考察するものである．ナローイングの 2 通りの定式化を新たに導入する．1 つは，制限のない抽象高階書換え系に対するもので，抽象的な重なりの概念に基づいた，可能な限り一般的な定式化である．もう 1 つは，パターンに制限された抽象高階書換え系に対するもので，従来の一階のナローイングと同じくナローイング可簡約項を明示的に扱うものである．これらの定式化を行うために必要となる高階文脈の変形に関する諸性質についても議論する．本論文の主要結果は，見かけ上まったく異なるこれらの定式化が，パターンに対しては一致することである．この主張の厳密な証明を与える．

Narrowing in Abstract Higher-order Rewrite Systems

Satoshi Okui[†] and Taro Suzuki[††]

Higher-order narrowing is an important issue in functional-logic programming. For higher-order term rewriting systems à la Nipkow, a formulation of narrowing has been so far given by Prehofer. On the other hand, nothing has ever been done for abstract higher-order rewrite systems, which is a more general framework introduced by van Oostrom. In this paper, we address higher-order narrowing in abstract higher-order rewrite systems. Two kind of formulations are considered: one is in arbitrary abstract higher-order rewrite systems, the other being restricted on so-called patterns. The former looks most unlike traditional first-order narrowing as it is based on an abstract notion of overlap, instead of usual narrowing redex. In contrast, the later is more akin to traditional one, but not for arbitrary terms. We compare those two formulations, showing their equivalence for patterns. A rigorous proof supporting our claim is given.

1. はじめに

等式の記号的な求解手法として広く知られるものに，ナローイング（narrowing）がある．1970 年代に Slagle[14]，Fay[6] らによって導入されて以来，今日に至るまで様々な改良が加えられてきた．ナローイングは，合流性を有する項書換え系によって与えられる等式理論に特化しており，求解のための探索空間が比較的小さい．このため，自動証明手続きとしてだけでなく，関数・論理型プログラミング言語の計算モデルとしても有効である．後者の観点から見たナローイングが，本研究の対象である．

関数・論理型プログラミング言語の計算モデルとしてのナローイングの問題点は，その適用範囲が，一階の項書換え系で表現可能な問題領域に限定されることである．これは，関数プログラミングにおいて有用性の認められた高階関数を扱ううえで，非常な困難をもたらす．したがって，ナローイングを高階の書換え系に対して一般化することが重要な課題である．

高階の書換え系には，大きく 2 通りの流儀がある．1 つは，Nipkow による高階項書換え系（Higher-0rder Rewrite System; HRS）[2),11)] である．これは，一階の書換え系の直接的な一般化である．もう 1 つは，van Oostrom によって導入された，抽象高階書換え系☆（Abstract Higer-Order Rewriting System;

† 中部大学工学部情報工学科
Department of Information Engineering, College of Engineering, Chubu University
†† 会津大学コンピュータ理工学部コンピュータソフトウェア学科
Department of Computer Software, The School of Computer Science and Engineering, The University of Aizu

☆ 文献 16)〜18) での正式な呼称は，Higer-Order Rewriting System（HORS）であるが，Niokow の HRS と非常に紛らわしいので，本論文においては，このように参照する．

AHRS)[16)~18)] という，より一般的な定式化である．後者に固有の特徴は，代入計算と呼ばれる仕組みを導入して，書換えを代入の局面と置換の局面に明確に分離している点である．このため，λ-計算のような束縛変数を有するシステムだけでなく，SK-コンビネータのようなシステムまで含めた広範囲のシステムを，AHRS の枠組みの中で，統一的に扱うことが可能である．これは，プログラミング言語の意味論から処理系実装までを統一的に扱う計算モデルにとって，望ましい特徴である．

高階の書換え系に 2 通りの定式化があることに対応して，高階のナローイングを定式化する方式も大きく 2 通り考えられることになる．HRS に基づく高階ナローイングに関しては，Prehofer による包括的な研究がある[12)]．一方，AHRS に基づいてナローイングを定式化する研究は，筆者らが知る限り，行われていない．

そこで，AHRS に基づいて，高階ナローイングの新たな定式化を行おうというのが，本研究である．

本論文で提示するナローイングの定式化は，2 通りある．1 つは，任意の高階項を扱うことが可能な，抽象度の高い定式化であり，もう 1 つは，対象をパターン[10)] に制限した場合の，より具体的な定式化である．両者は，一見，まったく異なる定式化である．しかしながら，前者が確かに後者の保存的（conservative）な一般化であり，パターンを仮定すれば，前者から後者が導出されることが明らかになる．これは，本論文の主張する主要結果であり，これについては，厳密な証明を与える．

本論文は，高階文脈を全面的に用いるという点で，従来のアプローチとはまったく異なっている．ここでいう高階文脈とは，AHRS の枠組みで用いられる文脈のことであり，HRS の枠組みにおける従来の文脈とは異なった性質を有している．高階文脈の性質については，これまで十分に明らかにされてこなかったので，ナローイングの定式化に先立ち，その性質についても考察する．

本論文は，以下のように構成される．まず 2 章で，記法等の準備を行った後，3 章で AHRS を導入する．続いて 4 章で，高階文脈の性質について論じる．それに基づき，5 章で，我々のナローイングの定式化を行う．ここで，一般の高階項に対する抽象的な定式化とパターンに対する具体的な定式化の 2 通りを考察し，抽象的定義において項をパターンに制限したものと，具体的定義とが，一致することを示す．最後に 6 章で，現時点での課題と，今後の研究の方向に言及する．

2. 準　　備

2.1 表　　記

本論文で用いる基本的な記法や概念は，一般的な用法に，ほぼ従う．項書換え系については文献 2)，7) 等に，λ-計算に関しては文献 4)，5) 等に，また単一化に関しては文献 2)，3)，13) 等に準じる．以下では，本論文で用いる記法について，簡単に断っておく．

通常の項書換え系の**項**（term）を指すのに $M, N, \ldots$ を用いる．これは $s, t, \ldots$ を後述の抽象高階書換え系の項を指すのに用いるためである．同様に，**書換え規則**（rewrite rule）を指すのに，$L \to R$ 等を用い，$\ell \to r$ 等は抽象高階書換え系の書換え規則を指すために用いる．M に出現する**自由変数**（free variable）の集合を $\mathcal{V}(M)$ と書く．項の中の記号の**出現位置**（occurrence）を指すには $p, q, \ldots$ を用いる．位置 p からみた q の相対的な位置を指すには $q\backslash p$ を用いる．たとえば $q = 1.2.3$，$p = 1$ のとき $q\backslash p = 2.3$ である．**代入**（substitution）σ の項 t への**作用**（application）を，後置記法で $t\sigma$ と書く．代入の**合成**（composition）は，並置によって表現される．すなわち，σ_1 と σ_2 を合成して得られる代入 $\sigma_1\sigma_2$ は，任意の項 M に対して $M(\sigma_1\sigma_2) = ((M\sigma_1)\sigma_2)$ のように作用するものとする．λ-項や SK-コンビネータ項を表記するのに，**適用的**（applicative）ではなく**関数的**（functional）な表記を用いる．すなわち $\lambda x.((fx)y)$ や $\lambda x.fxy$ と書く代わりに $\lambda x.f(x, y)$ のように書き，$\mathsf{S}xyz = xz(yz)$ のように書く代わりに $\mathsf{S}(x, y, z) = x(z, y(z))$ のように書く．λ-項のバインダは，その個々の変数を具体的に表記する必要がない場合，なるべく略記する．たとえば，$\lambda x_1 x_2 x_3.x_1$ を $\lambda\bar{x}.x_1(x_1 \in \{\bar{x}\})$ のように表記する．ここで $\{\bar{x}\}$ は，$\bar{x}$ に出現する変数の集合を意味する．本論文で取り扱う λ-項は，**単純型付き**（simply typed）である．型の明示的な表記は，多くの場合，省略される．

型 τ の**階数**（order）$\mathrm{Ord}(\tau)$ は通常どおり τ が基底型のとき 1，$\tau = \alpha \to \beta$ のとき $\max\{\mathrm{Ord}(\alpha) + 1, \mathrm{Ord}(\beta)\}$ を意味するものとする．

2.2 ナローイング

本論文の主題である抽象高階ナローイングは，従来のナローイングとはまったく異なる定式化に基づいている．比較のため，従来のナローイングの定義を以下に示す．

定義 1　ある書換え規則☆ $L \to R$ と，項 M のある

☆ $\mathcal{V}(M)$ と $\mathcal{V}(L)$ が排反になるように変数の名前変えをする．

$$\begin{aligned} eq(ap(u,v),c(a,c(b,nil))) &\rightsquigarrow_{1,(r_2)',\sigma_1,\sigma_2} eq(c(x',ap(y',v),c(a,c(b,nil)))\\ &\rightsquigarrow_{1.2,(r_1)'',\sigma_1',\sigma_2'} eq(c(x',v),c(a,c(b,nil)))\\ &\rightsquigarrow_{\varepsilon,(r_3)''',\sigma_1'',\sigma_2''} true \end{aligned}$$

図 1 ナローイング列
Fig. 1 A narrowing sequence.

非変数位置 p に対して $M|_p$ と L の最汎弱単一化代入組 (σ_1,σ_2) が存在するとする．このとき，項 M は項 $N \stackrel{\text{def}}{=} M\sigma_1[r\sigma_2]_p$ に（1 ステップで）**ナローイング**される（M narrows to N）といい，

$$M \rightsquigarrow_{p,L\to R,\sigma_1,\sigma_2} N$$

と表記する．$\rightsquigarrow$ の添字 $p, L \to R, \sigma_1, \sigma_2$ は適宜省略される．

ここで項の組 (M,N) の**最汎弱単一化代入組**（most general weak unifier）とは，$M\sigma_1 = N\sigma_2$ を満たす代入の組 (σ_1,σ_2) であって，$M\sigma_1' = N\sigma_2'$ を満たす任意の代入の組 (σ_1',σ_2') に対して，ある代入 τ が存在して $\sigma_1\tau = \sigma_1'$，$\sigma_2\tau = \sigma_2'$ を満足するものを意味する．

上の定義で書換え規則の左辺と単一化される部分項 $M|_p$ を**ナローイング可簡約項**（narrowing redex），あるいは，**ナレックス**（narex）と呼ぶ．

ナローイングのステップ $\rightsquigarrow$ を 0 個以上連結したものを $\stackrel{*}{\rightsquigarrow}_\sigma$ で表す．ここで σ は各ステップで用いた最汎弱単一化代入組の 1 番目の要素を順に合成したものである．たとえば，$M \stackrel{*}{\rightsquigarrow}_\sigma N = M \rightsquigarrow_{\sigma_1} \cdot \rightsquigarrow_{\sigma_2} \cdot \rightsquigarrow_{\sigma_3} N$ のとき，$\sigma = \sigma_1\sigma_2\sigma_3$ である．

例 1 **図 1** に，ナローイング列の例を示す．この例で用いているのは，書換え規則 (r_1) – (r_3) からなる書換え系である．

$$\mathcal{R} = \begin{cases} (r_1) & ap(nil,x) & \to & x \\ (r_2) & ap(c(x,y),z) & \to & c(x,ap(y,z)) \\ (r_3) & eq(x,x) & \to & true \end{cases}$$

$(r_2)'$ は (r_2) に出現する変数にダッシュ（′）を付加したものである．$(r_1)''$，$(r_3)'''$ についても同様である．各ステップで用いた最汎弱単一化代入組は，以下のとおりである．

$$\begin{aligned} \sigma_1 &= \{u \mapsto c(x',y')\} & \sigma_2 &= \{z' \mapsto v\}\\ \sigma_1' &= \{y' \mapsto nil\} & \sigma_2' &= \{x'' \mapsto v\}\\ \sigma_1'' &= \{x' \mapsto a, v \mapsto c(b,nil)\}\\ \sigma_2'' &= \{x''' \mapsto c(a,c(b,nil))\} \end{aligned}$$

$\sigma = \sigma_1\sigma_1'\sigma_1''$ と置けば，図 1 のナローイング列は以下のようにも記述可能である．

$$eq(ap(u,v),c(a,c(b,nil))) \stackrel{*}{\rightsquigarrow}_\sigma true$$

なお，

$$ap(u,v)\sigma \twoheadrightarrow_{\mathcal{R}} c(a,c(b,nil)$$

であるから，σ は，項 $eq(ap(u,v),c(a,c(b,nil)))$ を $\mathcal{R}$ を法とする等式 $ap(u,v) = c(a,c(b,nil))$ と見なした場合の「解」の 1 つを与えている．このことが，ナローイングが等式の求解手続きである所以である．

上記の定義は，ナローイングに関する文献 9) で一般的に用いられるものとは，最汎単一化代入（most general unifier）の代わりに最汎弱単一化代入組を用いている点が異なるが，本質的には同じものである．これは，単一化される項の組が共通する変数を持たないように書換え規則中の変数が名前変えされるからである☆．本論文では，後述の抽象高階ナローイングとの比較を容易にするために，上記の定義を採用する．

3. 抽象高階書換え系

本論文のナローイングの定式化は，van Oostrom の AHRS[16)~18)] の枠組みに基づいている．そこで，AHRS について必要最小限の事項を，予備知識を仮定せずに，導入する．

3.1 抽象高階書換え系（AHRS）とは

まず，通常の（一階の）項書換え系における書換えを分析することから始める．項書換え系 $\mathcal{R} = \{f(x) \to g(f(x))\}$ が与えられているとき，書換え $c(f(a),b) \to c(g(f(a)),b)$ が可能である．これは，文脈☆☆ $C = c(\Box,b)$ と代入 $\sigma = \{x \mapsto a\}$ を用いると，

$$\begin{aligned} c(f(a),b) &\stackrel{(1)}{=} C[f(x)\sigma]\\ &\stackrel{(2)}{\to} C[g(f(x))\sigma]\\ &\stackrel{(3)}{=} c(g(f(a)),b) \end{aligned}$$

と書ける．操作としては，(1)，(2)，(3) はそれぞれ以下のようなものと考えることができる．

☆ 最汎単一化代入と最汎弱単一化代入組の違いは，一般の E-単一化では単一化問題のタイプへの影響等微妙な問題をはらんでいるが，構文単一化に関する限りは問題は起きない[1)]．

☆☆ 文脈とは特別な記号 □（ホール）を含む項のことである．C が文脈のとき，$C[s]$ は C 中の □ を項 s で置き換えたものを表す．文脈の正確な定義は，後で与えられる．

(1) 項を，書換え規則の左辺と残りの部分（文脈 C と代入 σ）とに，分離する．
(2) 分離により現れた書換え規則の左辺を，右辺で置き換える．
(3) 書換え規則の右辺に文脈 C と代入 σ を適用して，項を生成する．

項書換え系では，これら 3 つの操作は暗黙のうちに行われるが，AHRS では，それらを明示的なものとして扱っている．これらの操作のうち，(2) は単なる置換であるが，(1) と (3) は代入と文脈に関する操作である．AHRS では，後者の操作を行う計算系を自由に選択可能である．(1) と (3) を行うために導入された計算系を**代入計算**（substitution calculus）と呼ぶ．

代入計算を交換することで，同じ書換え系の様々な表現が可能になる．例として，代入計算を λ-計算とした場合と，SK-コンビネータ論理とした場合に，上記の項書換え系がどのように表されるかを以下に示す．

代入計算が単純型付き λ-計算の場合には，代入や文脈に対する操作は β-簡約に基づき与えられる．これに対応して，AHRS は $\mathcal{R} = \{\lambda x.f(x) \to \lambda x.g(f(x))\ (\mathsf{f})\}$ のように，すべての変数が束縛された λ 項で表される．このとき，下図の (a)，(b)，(c) が (1)，(2)，(3) に対応することは，容易にみてとれる．(a) は β-展開（β-簡約の逆の操作）により，(c) は β-簡約により，実現される．

$$\begin{array}{ccc} c((\underline{\lambda x.f(x)})(a),b) & \xrightarrow[(b)]{(\mathsf{f})} & c((\underline{\lambda x.g(f(x))})(a),b) \\ *\downarrow\ \beta\ (a) & & *\downarrow\ \beta\ (c) \\ c(f(a),b) & \xrightarrow[\mathcal{R}]{} & c(g(f(a)),b) \end{array}$$

ここで $\to_{\mathcal{R}}$ は，(a)，(b)，(c) を結合したものであり，$\mathcal{R}$ での 1 ステップを表す．

次に，代入計算が SK-コンビネータ論理の場合には，代入や文脈に対する操作は簡約 $\to_{SK}$ に基づき与えられる．また，これに対応して，AHRS は $\mathcal{R}' = \{f \to S(K(g),f)\ (\mathsf{f}')\}$ のように，SK-コンビネータと関数記号だけを用いて表される．このとき，下図の (a')，(b')，(c') が (1)，(2)，(3) に対応することは，容易にみてとれる．(a') は SK-展開（SK-簡約の逆の操作）により，(c') は SK-簡約により，実現される．

$$\begin{array}{ccc} c(\underline{f}(a),b) & \xrightarrow[(b')]{(\mathsf{f}')} & c((\underline{S(K(g),f},a,b) \\ *\downarrow\ SK^{-1}\ (a') & & *\downarrow\ SK\ (c') \\ c(f(a),b) & \xrightarrow[\mathcal{R}]{} & c(g(f(a)),b) \end{array}$$

前述のように，AHRS は，代入のフェーズと書換え規則による置換のフェーズを明確に分離し，前者を代入計算という形でパラメータ化している．このことは，2 つの利点をもたらす．1 つは，上の 2 つの例に見られるように，同じ書換え系を異なる複数の形式で表せるという点である．これにより，ある書換え系の意味は変えずに，表現だけを状況に応じて望ましいものにすることが可能になる．たとえば，項書換え系に基づく言語処理系を設計・実装する際，抽象度の高い操作的意味は $\mathcal{R}$ のような単純型付き λ-計算で表現し，実装の際には $\mathcal{R}'$ のようなコンビネータで表現することが考えられる．もう 1 つは，広範囲に及ぶ書換え系が記述可能になるという点である．まったく異なるように見える様々な書換え系も，書換え規則による置換という操作では共通している．そこで代入計算の部分をパラメータとして抽象化することにより，様々な書換え系に関する性質を統一的に論じることが可能になると考えられる．

AHRS の定義

上で見た 2 つの例のように AHRS は (1) **指標**（signature）と (2) **代入計算**（substitution calculus）と (3) **書換え規則**（rewrite rule）を与えることで定義される．

指標は 2 項の記号 λ と $\cdot(\cdot)$ に加えて以下の 4 種類の定数記号から成る．

(1) **代入演算子** $\mathcal{R}$ にはない．$\mathcal{R}'$ では S,K.
(2) **代入変数（束縛変数）** $\mathcal{R}$ では x である．$\mathcal{R}'$ にはない．
(3) **演算子** $\mathcal{R}$ と $\mathcal{R}'$ に共通のもの（f,g,a,b）である．
(4) **ホール** 文脈を構成するための記号（後述）．

代入計算はこれらの記号からつくられる閉じた（自由変数の出現しない）λ-項の上の書換え系 SC によって与えられる．$\mathcal{R}$ では SC は型付き λ-計算であり，$\mathcal{R}'$ では型付きの SKI-コンビネータ論理である．SC の書換え関係を $\twoheadrightarrow_{SC}$（1 ステップの場合は $\to_{SC}$）で表す．以下，本論文ではことわらない限り SC はリダクション規則として β-簡約 $\to_\beta$ と η-展開 $\to_{\eta^{-1}}$ $(=\!\!\leftarrow_\eta)$ を持つ単純型付き λ-計算を意味する．したがって SC は完備である（すなわち，合流性と停止性を備えている）．η-簡約ではなく η-展開を使うのは，高階単一化において $\beta\eta^{-1}$-正規形のほうが $\beta\eta$-正規形よりも有用であり，好んで用いられるのと，同じ理由である[15]．

書換え規則は，同じ型の閉項の対によって定義される（書換え規則 (ℓ,r) を $\ell \to r$ と書く）．以下で，AHRS で項といったときには，つねに閉じた項を意味

していることに注意．また項は，特にことわらない限り，書換え $\twoheadrightarrow_{SC}$ に関する正規形（SC-正規形）を考える．

AHRS における**書換え関係**（rewrite relation）を定義するには**文脈**（context）が必要である．以下，指標には各々の型 τ 対して可算個の特別な定数記号（**ホール**と呼ぶ）$\Box_{\tau,0}, \Box_{\tau,1}, \ldots$ が含まれていると仮定する．添字は適宜省略する．文脈には同じホールはたかだか 1 カ所にしか出現しないとする．n（$n \geq 0$）個のホールを含む項を（n-項）**文脈**（n-ary context）と呼ぶ．今後は，項という用語を，もっぱらホールの出現しない項を指すためにだけ用いる（したがって，項は文脈の特別な場合である）．項 s と t に対して $C[s] =_{SC} t$ を満足する文脈 C があるとき，s は t **の中**の（文脈 C **における**）項であるという．これは，(SC として型付き λ-を考えている状況では)「項 s を具体化したものが t の部分項である」という状況を意味する．項書換え系における従来の文脈と異なり，2 階以上の型のホール記号が許容されるからである．このため AHRS における文脈は，従来の文脈の役割に加えて従来代入が担っていた役割も兼備することになる．AHRS の議論をする際に多用される重要な概念である．

文脈を用いて，書換えを次のように定義する．AHRS $\mathcal{R}$ の項 s，t にたいして，$\mathcal{R}$ のある書換え規則 $\ell \to r$ があって．ℓ が s **の中**の文脈 C における項であり r が t **の中**の文脈 C における項であるとき，s は t に（1 ステップで）**書換え可能**（rewritable）であるといい，$s \to_{\mathcal{R}} t$ と書く．図式的に表すと，以下のようになる．

$$\begin{array}{ccc} C[\ell] & \xrightarrow{\ell \to r} & C[r] \\ {\scriptstyle SC}\downarrow{\scriptstyle *} & & {\scriptstyle *}\downarrow{\scriptstyle SC} \\ s & \xrightarrow[\mathcal{R}]{} & t \end{array}$$

$\mathcal{R}$ の**書換え関係**$\twoheadrightarrow_{\mathcal{R}}$ は，$\to_{\mathcal{R}}$ の反射推移閉包（reflective-transitive closure）として定義される．

4. 高 階 文 脈

4.1 文脈の合成

文脈 C が $\Box_0, \ldots, \Box_n$ を含んでいるとき（これら以外にもホール記号を含んでいてよい），これらを同じ型の文脈 $C_0, \ldots, C_n$ で同時に置換したものの SC-正規形を，$C[C_0, \ldots, C_n]_{\Box_0,\ldots,\Box_n}$ で表す．置換するホールの対応が，前後の記述等から明らかな場合には，単に $C[C_0, \ldots, C_n]$ と表記する．

従来の文脈では，置換の結果，束縛変数のスコープが文脈の境界をまたぐような（すなわち，$C[t]$ において t の自由変数が C の中に現れるバインダで束縛されるような）状況も許されるが，AHRS における文脈はそうではないので注意が必要である．文脈も項も閉じているものだけを考えているので，t の変数が C で束縛されるような状況は生じない．

また，置換の結果，$C[C_0, \ldots, C_n]_{\Box_0,\ldots,\Box_n}$ において，同じホール記号が重複して出現する可能性がある．これを防ぐには，$C_0, \ldots, C_n$ におけるホール記号を，必要に応じて，名前変え（renaming）してから置換すればよい．以下では，このことを仮定する．

ブラケット（[ ]）が複数ある場合は，左結合であると約束する．たとえば $C[D]_{\Box_1}[E]_{\Box_2}$ は C の $\Box_1$ を D で置換してできる文脈の $\Box_2$ を E で置換してできる文脈の SC-正規形を意味する．一般にはブラケットの順序は交換できないことに注意．$\Box_2$ が D に由来しているかもしれないからである．

置換の順序は結合的，すなわち $C[D[E]] =_{SC} C[D][E]$ である．より一般的には

$$\begin{aligned} &C[D_1[E_{1,1}, \ldots, E_{1,m_1}], \ldots, \\ &\qquad D_n[E_{n,1}, \ldots, E_{n,m_n}]] =_{SC} \\ &\quad C[D_1, \ldots, D_n][E_{1,1}, \ldots, E_{1,m_1}, \\ &\qquad\qquad \ldots, E_{n,1}, \ldots, E_{n,m_n}] \end{aligned}$$

が成り立つ．

4.2 硬文脈と軟文脈

前節の例が示していたように高階の文脈は，「プレースホルダ」としての役割に加えて，従来の代入の役割をも担っている．このことが，AHRS における書換えの定式化に簡潔さをもたらす要因である．

その一方で，これは，高階の文脈を位置を特定するために利用できないということでもある．一階の場合には，$C_1[s] \equiv C_2[s]$ ならば，$C_1 \equiv C_2$ である．しかし，高階文脈の場合には，これが一般に成立しない．

例 2 文脈 $C_i \stackrel{\text{def}}{=} \Box(a_i)$ $(i = 1, 2; a_1 \neq a_2)$，と項 $s \stackrel{\text{def}}{=} \lambda x.c$ を考える．$C_1[s] =_{SC} C_2[s]$ であるが，$C_1 \neq_{SC} C_2$．

しかしながら，C，D を代入の作用を持たないような文脈に制限すれば，この性質が（このあとの**命題** 1 で示すように）回復する．本論文では，このような文脈を導入し，それを**硬文脈**（rigid context）と呼ぶ．

硬文脈を導入するにあたり，まず本論文における位置の定義を明確にすることから始める．

定義 2 文脈 $C = \lambda\bar{x}.a(C_1, \ldots, C_n)$ の**位置** p における**部分文脈** $C|_p$ と，位置 p における部分文脈の D による**置換** $C[D]_p$ を，それぞれ以下のように帰納的

に定義する．

$$C|_\varepsilon = C$$
$$C|_{i.p} = (\lambda\bar{x}.C_i)|_p$$
$$C[D]_\varepsilon = D$$
$$C[D]_{i.p} = \lambda\bar{x}.a(\ldots,(\lambda\bar{x}.C_i)[D]_p(\bar{x})_{\downarrow_{SC}},\ldots)$$

ここで，$(\lambda\bar{x}.C_i)[D]_p(\bar{x})_{\downarrow_{SC}}$ は $(\lambda\bar{x}.C_i)[D]_p(\bar{x})$ の SC-正規形を意味する．

$C|_p$ が**変数項**（$\lambda\bar{x}.x(\bar{s})(x \in \{\bar{x}\})$ という形の項）のときの位置 p を C の**変数位置**といい，そうでないときには**非変数位置**という．

例 3 文脈

$$C = \lambda x.f(\lambda y.\Box(x(y), \lambda z.x(z)))$$

の部分文脈（部分項）は，以下のようになる．

$$C|_1 = \lambda xy.\Box(x(y), \lambda z.x(z))$$
$$C|_{1.1} = \lambda xy.x(y)$$
$$C|_{1.2} = \lambda xyz.x(z)$$
$$C|_{1.1.1} = \lambda xy.x$$
$$C|_{1.1.2} = \lambda xy.y$$
$$C|_{1.2.1} = \lambda xyz.x$$
$$C|_{1.2.2} = \lambda xyz.z$$

また，C の部分項 $C|_{1.2}$ の，同じ型の項

$$\lambda uvw.g(w,v,u)$$

による置換 $C[\lambda uvw.g(w,v,u)]_{1.2}$ は，以下のようになる．

$$\lambda x.f(\lambda y.\Box(x(y), \lambda z.g(z,y,x)))$$

上記の定義では，部分文脈を考えるときに，もとの文脈のバインダを継承させている．これは，すでに述べたように，AHRS における文脈（項を含む）は閉じていなくてはならないからである．部分文脈の置換についても同様である．このことは，若干の煩雑さをもたらす一方で α-同値なものを同一視できるという利点をもたらす．

また，頭部の記号やバインダには位置が割り当てられていないことにも注意が必要である．これは，本論文では，すぐ後で導入する「のりしろ」という特別な部分項を考えるときにしか，位置を必要としないためである．

頭部記号がホールであるような部分文脈を，その文脈の，**のりしろ**と呼ぶ．上例で，$C|_1$ は，C の，のりしろである．のりしろは，含まれるホール記号ごとにある．以下では，「頭部がホール記号 □ であるような，のりしろ」のことを，短く「のりしろ □」と呼ぶ☆．

のりしろが，すべてホール記号の SC-正規形，本論文では代入計算として $\lambda_{\beta\eta^{-1}}$ を想定しているので $\lambda\bar{x}.\Box(\bar{x})$（厳密には，その長 η-正規形），であるような文脈を，**硬文脈**と呼ぶ．

例 4 文脈 $\lambda x.f(\lambda y.\Box(\lambda w.x(w), y))$ は，硬文脈である．ただし，$\lambda w.x(w)$ は x の長 η-正規形であり，y はすでに基底型であるとする．これに対して，$\lambda x.f(\lambda y.\Box(\lambda w.y, x(w)))$ は，硬文脈ではない．これはのりしろ □ における引数のならびと，のりしろ □ のバインダのならびの順序が異なるからである．また，$\lambda xy.\Box(x)$ も硬文脈ではない．y が，のりしろ □ のバインダに出現するにもかかわらず，のりしろ □ の引数に出現しないからである．

一階の場合には，硬文脈のホールの型は硬文脈自身の型と同じになるが，一般には，ホールの型のほうが階数が高くなる．これは項の頂点から部分文脈に至る途中に出現するバインダが付加されるからである．

上の定義に従えば，一般に C を硬文脈とするとき，ホールの置換と部分項の置換の間には，p を C ののりしろの位置とするとき $C[D]_p =_{SC} C[D]$，という関係がある．ここで p はホールそのものの位置ではなく，そのホールを頭部に持つ，のりしろの位置であることに注意☆☆．このことは，今後断りなく用いる．

次の命題は，硬文脈が，通常の一階の文脈と同じように，プレースホルダとして機能することを述べている．

命題 1 任意の硬文脈 C_1，C_2 と任意の項 s_1，s_2 が $C_1[s_1] =_{SC} C_2[s_2]$ を満足しているとき，以下が成り立つ．

$$C_1 =_{SC} C_2 \iff s_1 =_{SC} s_2$$

証明 ホールの位置の大きさに関する数学的帰納法．

□

硬文脈でない文脈を**軟文脈**（flex context）と呼ぶことにする．また，それ自身がのりしろ（の 1 つ）になっているような文脈を**代入文脈**という．代入文脈のうちで硬文脈でもあるものは $\lambda\bar{x}.\Box(\bar{y})$（$\{\bar{y}\} \subseteq \{\bar{x}\}$）（厳密には，その長 η-正規形）という形のものに限られる．

代入文脈が重要なのは，任意の文脈が，硬文脈と代入文脈の結合によって，一意に表せるからである．

命題 2 （セパレーション）任意の文脈は，硬文脈と

☆ キャップ文脈ののりしろの考え方には関数型言語の実装における lambda lifting と共通するものがある．

☆☆ そもそも，ホールそのものには（基底型である場合を除き）位置が割り当てられていない．

代入文脈の結合の形で，一意に表せる．

証明　多項文脈の場合も同じなので，簡単のため 1-項文脈の場合で考える．与えられた文脈 C の，のりしろの位置を p とすると，のりしろと同じ型の新しいホール記号 $\Box$（厳密には，その長 η-正規形）で，のりしろを置換して得られる $C' \stackrel{\text{def}}{=} C[\Box]_p$ は，定義より硬文脈である．また，$C|_p$ は定義より代入文脈である．部分項と置換の定義より $C'[C_p] = C$ を得る．一意性は，**命題** 1 に従う．□

以下で必要になるのは 1-項文脈のセパレーションだけである．そこで，これ以降では，この証明中にでてきた文脈を

$$\mathsf{cap}(C) \stackrel{\text{def}}{=} C[\Box]_p \quad \mathsf{sub}(C) \stackrel{\text{def}}{=} C|_p$$

と書いて参照する．

例 5　**例** 3 の文脈 C は，硬文脈

$$\mathsf{cap}(C) = \lambda x.f(\lambda y.\Box'(\lambda w.x(w), y))$$

と代入文脈

$$\mathsf{sub}(C) = \lambda xy.\Box(x(y), \lambda z.x(z))$$

とに分解できる．

4.3 パターン文脈

高階項の記号処理では，パターン[10] と呼ばれる特別な高階項が，考察の対象になる．

パターンにおいては，代入によって具体化された変数に対して逆に代入仕返すような変形が起こらない．このため，一階の項場合に成り立つ性質が，パターン項の場合にも成り立つことが多い．

AHRS においては，パターンも閉項で表現することになる．以下の定義は，van Oostrom [17] に従う．

定義 3　文脈 C を $\lambda\bar{x}.C'$（C' は基底型）と置く．すなわち $\{\bar{x}\}$ を C の最も外側のバインダに出現する束縛変数とする．このとき，任意の $x \in \{\bar{x}\}$ に対して，x を頭部に持つ部分項が $\lambda\bar{x}\bar{y}.x(\bar{z})$（$\{\bar{y}\} \cap \{\bar{x}\} = \emptyset$ であり，$\{\bar{z}\} \subseteq \{\bar{y}\}$）という形をしているならば，$C$ は**パターン文脈**と呼ばれる．

上の定義において，最も外側のバインダに出現する束縛変数 $\{\bar{x}\}$ を，以下では，C の**パターン変数**と呼んで参照する．

パターン文脈をパターン文脈で置換して得られる文脈も，パターン文脈である．また，パターン文脈に cap や sub を作用させたものも，パターン文脈である．

例 6　**例** 3 の文脈 C は，パターン文脈である．しかし，文脈 $\lambda xy.\Box(x(y))$ はパターン文脈ではない．パターン変数 x に，パターン変数 y が適用されているからである．また，文脈 $\lambda x.x(\Box)$ もパターン文脈ではない．パターン変数 x に，定数記号 $\Box$ が適用されているからである．

代入文脈 C と，それに含まれるホール $\Box$ と同じ型を持つ硬文脈 D を考える．すでに指摘したように，D のホールの型の階数は，C のホールの型の階数以上である．$\Box$ を D に含まれるホール $\Box'$ で置き換えたものの SC-正規形を $C\uparrow_{\Box'}$ と書く．D がホールを 1 つしか含まない場合は，$C\uparrow_D$ のようにも表記する．このとき以下の命題が成り立つ．

命題 3　（キャンセレーション）任意の文脈 C と，任意のパターン硬文脈 D に対して

$$\mathsf{sub}(C[D]) =_{SC} \mathsf{sub}(C)\uparrow_D$$

が成り立つ．

形式的な証明は，煩雑になるので省略するが，項を木構造で図示して考えれば，容易である．

D がパターンでないときには，上記の命題は成り立たない．たとえば，$C \stackrel{\text{def}}{=} \Box(\lambda xy.y)$，$D \stackrel{\text{def}}{=} \lambda x.x(\Box')$ を考えると $C[D] = \lambda y.y$ となり，そもそも左辺から $\Box'$ が消滅する．

なお，本論文では用いないが，上記の命題は D が任意のパターン文脈のときに一般化でき

$$\mathsf{sub}(C[D]) =_{SC} \mathsf{sub}(C)\uparrow_{\mathsf{cap}(D)} [\mathsf{sub}(D)]$$

が成立する．すなわち，sub をパターン文脈の集合の上の写像と見なすと，文脈の合成演算に関して一種の準同型写像になっている．

5. 2 通りの定式化と等価性

5.1 ナローイングの定式化

5.1.1 項の重なり

定義 4　ある項 t の中の，文脈 C_1，C_2 における項 t_1，t_2 を考える．このとき，フレッシュな☆ホール $\Box_1$，$\Box_2$ に対して

$$\begin{cases} C[\Box_1, t_2] & =_{SC} C_1[\Box_1] \\ C[t_1, \Box_2] & =_{SC} C_2[\Box_2] \end{cases}$$

を満たす文脈 C が存在するならば，t_1 と t_2 は項 t の中で（文脈 C_1，C_2 において）**両立している**（consistent, simultaneous）といい，$(C_1, t_1) \perp (C_2, t_2)$ と表記する．そのような C が存在しない場合は，**重なりがあ**

☆ C_1, C_2 にも t_1, t_2 にも出現しないの意味．

る（overlapping）といい，$(C_1, t_1) \not\perp (C_2, t_2)$ と表記する．

例 7

$$C_1 = \lambda z.\Box_1(z, \lambda y.g(y, \mathbf{t})) = \lambda zx.f(x(z))$$
$$C_2 = \lambda z.f(\Box_2(z, z)) \quad t_2 = \lambda zy.g(y, z)$$
$$t = \lambda z.f(g(z, z))$$

を考える．

$$C_1[t_1] =_{SC} t =_{SC} C_2[t_2]$$

であるから，t_1，t_2 は，それぞれ C_1，C_2 における，t の中の項である．今，

$$C = \lambda z.\Box_1(z, \Box_2(z))$$

を考えると

$$C[\Box_1, t_2] =_{SC} C_1[\Box_1]$$
$$C[t_1, \Box_2] =_{SC} C_2[\Box_2]$$

である．よって，これらは t で両立している．すわなち

$$(C_1, t_1) \perp (C_2, t_2)$$

である．

上の例が示すように，（パターンとは限らない）一般の項と一般の代入計算に対して，上記の定義が意味するところは把握しにくい．文脈のホールを置換した項の間で，代入計算による複雑な相互作用が生じるからである．しかし，項をパターンに制限し，代入計算を $\lambda_{\beta\eta^{-1}}$ に限定するならば，従来どおり項 t において t_1 と t_2 が重なることを意味する．このことは，形式的にはこの後の議論で正当化される．

次に述べる**補題** 1〜**補題** 3 はそれぞれ両立性が文脈の合成，硬文脈を除去する操作，部分項を取り出す操作で，閉じていることを述べており，この後で用いる．

補題 1 任意の文脈 D に対して，$(C_1, t_1) \perp (C_2, t_2)$ ならば $(D[C_1], t_1) \perp (D[C_2], t_2)$

証明 両立性の定義より明らか． □

補題 2 任意の硬文脈 D に対して，$(D[C_1], t_1) \perp (D[C_2], t_2)$ ならば $(C_1, t_1) \perp (C_2, t_2)$.

証明 両立性の定義より，ある文脈 C があって

$$\begin{cases} C[\Box_1, t_2] & =_{SC} D[C_1[\Box_1]] \\ C[t_1, \Box_2] & =_{SC} D[C_2[\Box_2]] \end{cases}$$

D は硬文脈であるので，そのホールの，のりしろの位置は，C の $\Box_1$，$\Box_2$ の，のりしろの位置より上にある．よって，ある文脈 E があって $C[\Box_1, \Box_2] = D[E[\Box_1, \Box_2]]$ となる．D は硬文脈であるから，**命題** 1 より

$$\begin{cases} E[\Box_1, t_2] & =_{SC} C_1[\Box_1] \\ E[t_1, \Box_2] & =_{SC} C_2[\Box_2] \end{cases}$$

を得る． □

補題 3 $(C_1, E_1[t_1]) \perp (C_2, E_2[t_2])$ であるならば $(C_1[E_1], t_1) \perp (C_2[E_2], t_2)$.

証明 両立性の定義より，ある文脈 C があって

$$\begin{cases} C[\Box_1, E_2[t_2]] & =_{SC} C_1[\Box_1] \\ C[E_1[t_1], \Box_2] & =_{SC} C_2[\Box_2] \end{cases}$$

そこで $F \stackrel{\text{def}}{=} C[E_1, E_2]$ とおくと t_1，t_2 と同じ型のフレッシュなホール記号 $\Box_1'$，$\Box_2'$ にたいして

$$\begin{cases} F[\Box_1', t_2] & =_{SC} C_1[E_1[\Box_1']] \\ F[t_1, \Box_2'] & =_{SC} C_2[E_2[\Box_2']] \end{cases}$$

を得る． □

補題 4 C_1，C_2 を代入文脈，s_1，s_2 を非変数項とし，これらが $C_1[s_1] =_{SC} C_2[s_2]$ を満たしているとする．このとき $(C_1, s_1) \not\perp (C_2, s_2)$.

証明 背理法による．仮に，$(C_1, s_1) \perp (C_2, s_2)$ であるとする．定義により，フレッシュなホール $\Box_1$，$\Box_2$ に対して

$$C[\Box_1, s_2]_{\Box_1', \Box_2'} =_{SC} C_1[\Box_1] \quad (1)$$
$$C[s_1, \Box_2]_{\Box_1', \Box_2'} =_{SC} C_2[\Box_2] \quad (2)$$

を満たす文脈 C が存在する．式 (1) と (2) の右辺は代入文脈であるから，左辺もそうでなければならない．よって，C は代入文脈でなければならない．$\Box_1$，$\Box_2$ はフレッシュであるから，C の頭部記号は $\Box_1'$ か $\Box_2'$ かのいずれかである．

C の頭部が $\Box_1'$ の場合を考える．s_1 は非変数項であり，フレッシュな $\Box_2$ を含まない．したがって式 (2) の左辺の頭部記号は $\Box_2$ にはならない．一方，式 (2) の右辺のホール記号は $\Box_2$ であるからこれは矛盾である．C の頭部が $\Box_1'$ の場合にも，同様にして，(1) 矛盾が導かれる．したがって，最初に C の存在を仮定したのが誤りであり，$(C_1, s_1) \not\perp (C_2, s_2)$ が結論できる． □

5.1.2 書換え規則

AHRS における**書換え規則**とは，同じ型を有する項 ℓ と r の組 (ℓ, r) である．ただし ℓ の頭部は演算子，すなわち代入計算に属さない関数記号（3 章参照），であるとする（最も，本論文では代入計算とし

て純粋な型付きラムダ計算 $\lambda_{\beta\eta^{-1}}$ だけを用いているので，関数記号といえば演算子のことである）．したがって，書換え規則の左辺 ℓ は非変数項である．項 ℓ と r が，ともにパターンであるような書換え規則を**パターン書換え規則**と呼ぶ．パターン書換え規則だけから成る AHRS を，**パターン書換え系**と呼ぶ．

書換え規則であることが容易に分かるように，組 (ℓ, r) のことを $\ell \to r$ と表記する．

従来の項書換え系では，書換え規則の両辺に共通して出現する変数が代入によって導入された項の受け渡しを担っていたが，AHRS の書換え規則では，両辺に共通するバインダがその役割を担っている．したがって，書換え規則の両辺が同じ型であることが重要である．たとえば，従来の項書換え系における $f(x,y,z) \to g(x,z)$ のような書換え規則は AHRS では $\lambda xyz.f(x,y,z) \to \lambda xyz.g(x,z)$ のように表現されるのであって，$\lambda xyz.f(x,y,z) \to \lambda xz.g(x,z)$ ではないことに注意が必要である．

5.1.3 抽象的定式化

本論文で提示する 2 通りの定式化のうち，まずは任意の高階項にたいする抽象的定義のほうを考える．

定義 5 項 s，t，書換え規則 $\ell \to r$，文脈 C に対して，ある文脈 D があって，以下を満たすとする．

(A0) C は代入文脈．
(A1) $C[s] =_{SC} D[\ell]$.
(A2) $(C,s) \not\perp (D,\ell)$.
(A3) $t =_{SC} D[r]$.

このとき s は C と $\ell \to r$ を用いて t に**抽象ナローイング可能**であるといい

$$s \looparrowright_{C,\ell\to r} t$$

と書く．添字は適宜省略する．

0 個以上の抽象ナローイングステップからなる列

$$s \looparrowright_{C_1} \ldots \looparrowright_{C_n} t$$

を $s \overset{*}{\looparrowright}_C t$ で表す．ここで $C = C_n[C_{n-1}[\ldots C_1 \ldots]]$ である．用いている AHRS $\mathcal{R}$ を明示して，$s \overset{*}{\looparrowright}_{C,\mathcal{R}} t$ のようにも書く．

5.1.4 ナレックスの存在

上の抽象ナローイングの定義は，従来のナローイングの定義からみると，大きく異なっている．最大の相違点は，従来のナローイングにおける中心概念であるナローイング可簡約項（ナレックス）が，より一般的な重なりの概念に置き換わっていることである．

これは，一般の高階項においては，代入された項に対して代入仕返す現象が起こるので，ナローイング可能性をとらえるのに，従来のナレックスの概念だけでは，不十分なためである．

しかしながら，パターンを仮定すれば，従来の一階のナローイングと同様に，ナレックスを用いて定式化できる．以下，このことを順に議論するが，まずはその鍵となる補題から始める．

次の補題は，パターンの仮定の下で，重なり（**定義** 4）からナレックスの存在を保証するものであり，次項のナローイングの具体的定式化の根拠となる．

補題 5 代入文脈 C，文脈 D，パターン項 s，非変数項 t が $C[s] =_{SC} D[t]$ を満たしているとする．このとき，もし $(C,s) \not\perp (D,t)$ ならば，$E[s_1] =_{SC} s$ を満たすパターン硬文脈 E と非変数パターン項 s_1 が存在して，$\mathsf{cap}(C[E]) =_{SC} \mathsf{cap}(D)$ と $C\!\uparrow_E [s_1] =_{SC} \mathsf{sub}(D)[t]$ が成り立つ．

証明 $\mathsf{cap}(\mathrm{D})$ の，のりしろの位置を p とする．このとき p は s の非変数位置であることを背理法で示す．仮に p は s の変数位置であるとする．$C[s] =_{SC} D[t]$ だから，p は $C[s]$ の位置であり，s のある変数位置 q に対して $q \le p$ となる．今 $\{\bar{x}\}$ を s のパターン変数の集合とすると，$s|_q$ は，$\lambda\bar{x}\bar{y}.y(\bar{u})$ という形をしているか，$\lambda\bar{x}\bar{y}.x_i(\bar{z})$ という形をしているか，どちらかである（ただし，ここで $x_i \in \{\bar{x}\}$，$y \in \{\bar{y}\}$，$\{\bar{y}\} \cap \{\bar{x}\} = \emptyset$，そして $\{\bar{z}\} \subseteq \{\bar{y}\}$ である）．しかし，前者はありえない．なぜなら，もし前者だとすると，y は s のパターン変数ではないので，代入文脈 C の作用を受けず，$C[s]|_q (=_{SC} \mathsf{sub}(D)[t])$ の頭部は y のままであるが，これでは t の頭部が変数であることになるからである．よって，後者である．

そこで，$s|_q = \lambda\bar{x}\bar{y}.x_i(\bar{z})$ の i を用いて $r = i.(p\backslash q)$ と置く．s がパターンであることと，C が代入文脈であることにとより，r は C における位置である．よって $C|_r$ を考えることができて，$C|_r =_{SC} C[s]|_p =_{SC} D[t]|_p =_{SC} \mathsf{sub}(D)[t]$．そこで

$$C' \overset{\mathrm{def}}{=} C[\mathsf{sub}(D)[\Box']]_r$$

なる 2 ホールの文脈☆を考えると

$$C'[\Box_1, t] =_{SC} C[\Box_1]$$

である．一方，

$$\begin{aligned} C'[s,\Box_2] &=_{SC} C[\mathsf{sub}(D)[\Box_2]]_r \\ &=_{SC} C[s][\mathsf{sub}(D)[\Box_2]]_p \\ &=_{SC} D[t][\mathsf{sub}(D)[\Box_2]]_p \\ &=_{SC} \mathsf{cap}(D)[\mathsf{sub}(D)[\Box_2]] \\ &=_{SC} D[\Box_2] \end{aligned}$$

☆ C のもともと t のあった部分が $\Box'$ で置き換わったから．

を得る．これは，しかし，$(C,s) \perp (D,t)$ を意味するので，仮定に反する．以上より，p は s の非変数位置である．そこで $\mathsf{cap}(D)$ ホール記号と同じ $\Box$ を用いて

$$E = s[\Box]_p, \quad s_1 = s|_p$$

と置くと，E はパターン硬文脈であり s は非変数パターン項である．

あとは，$\mathsf{cap}(C[E]) =_{SC} \mathsf{cap}(D)$ と $C{\uparrow_E}\,[s_1] =_{SC} \mathsf{sub}(D)[t]$ を示せばよい．E の定義より，$\mathsf{cap}(D)$ におけるのりしろ $\Box$ と，E におけるのりしろ $\Box$ の，位置は等しい．E がパターンであることに注意すれば，これは，さらに $\mathsf{cap}(C[E])$ におけるのりしろ $\Box$ の位置とも等しい．一方，キャンセレーションとセパレーションより，

$$\begin{aligned}
&\mathsf{cap}(C[E])[C{\uparrow_E}\,[s_1]]\\
&=_{SC} \mathsf{cap}(C[E])[\mathsf{sub}(C[E])[s_1]]\\
&=_{SC} C[E][s_1]\\
&=_{SC} C[s]\\
&=_{SC} D[t]\\
&=_{SC} \mathsf{cap}(D)[\mathsf{sub}(D)[t]]
\end{aligned}$$

がいえる．したがって**命題** 1 を適用することができて，$\mathsf{cap}(C[E]) =_{SC} \mathsf{cap}(D)$ と $C{\uparrow_E}\,[s_1] =_{SC} \mathsf{sub}(D)[t]$ とを得る． □

上の補題の証明で s がパターンであることは，不可欠である．事実，一般の高階項 s に対しては，この補題は成立しない．

5.1.5 具体的定式化

前項での議論に基づきナレックスを明示的に表現した，より具体的なナローイングの定式化を考察する．

定義 6 項 s，t，書換え規則 $\ell \to r$，文脈 C，硬文脈 E が与えられたとき，ある非変数項 s_1 と，ある代入文脈 D が存在して，以下を満たすとする．

(S0) C は代入文脈．
(S1) $s =_{SC} E[s_1]$．
(S2) $C{\uparrow_E}\,[s_1] =_{SC} D[\ell]$．
(S3) $t =_{SC} \mathsf{cap}(C[E])[D[r]]$．
(S4) s はパターン項．

このとき，s は C と $\ell \to r$ を用いて E において t に**ナローイング可能**であるといい

$$s \rightsquigarrow_{C,E,\ell\to r} t$$

と書く．添字は適宜省略する．

0 個以上のナローイングステップからなる列

$$s \rightsquigarrow_{C_1} \ldots \rightsquigarrow_{C_n} t$$

を $s \overset{*}{\rightsquigarrow}_C t$ で表す．ここで $C = C_n[C_{n-1}[\ldots C_1 \ldots]]$ である．用いている AHRS $\mathcal{R}$ を明示して，$s \overset{*}{\rightsquigarrow}_{C,\mathcal{R}} t$ のようにも書く．

5.2 等 価 性

2 通りの定式化の間の関係について考察する．

まず最初に，ナローイングの具体的定義は確かに抽象的定義の特別な場合であることを示す．

補題 6 項 s，t，書換え規則 $\ell \to r$，文脈 C，硬文脈 E が与えられたとする．このとき，もし $s \rightsquigarrow_{C,E,\ell\to r} t$ が成り立つならば，$s \looparrowright_{C,\ell\to r} t$ が成り立つ．

証明 条件 (S0)〜(S3) を満たす非変数項 s_1 と代入文脈 D を仮定する．$D' \stackrel{\text{def}}{=} \mathsf{cap}(C[E])[D]$ と置いたとき条件 (A0)〜(A3) の中の D を D' で置き換えたものが満たされることを示す．まず，(A0) は (S0) からただちにいえる．(A3) は，D' の定義と (S3) から，明らかである．

次に，D' の定義，(S1)，セパレーション，キャンセレーション，(S2) から，以下のようにして，(A1) がいえる．

$$\begin{aligned}
C[s] &=_{SC} C[E][s_1]\\
&=_{SC} \mathsf{cap}(C[E])[\mathsf{sub}(C[E])][s_1]\\
&=_{SC} \mathsf{cap}(C[E])[C{\uparrow_E}\,[s_1]]\\
&=_{SC} \mathsf{cap}(C[E])[D[\ell]]\\
&=_{SC} D'[\ell]
\end{aligned}$$

あと示すのは (A2) である．背理法を用いる．仮に (A2) が成立しないとする．すなわち，

$$(C,s) \perp (D',\ell)$$

を仮定する．これに (S1) と**補題** 3 を適用して

$$(C[E],s_1) \perp (D',\ell)$$

を得る．セパレーション，キャンセレーション，D' の定義を用いて，変形すると

$$(\mathsf{cap}(C[E])[C{\uparrow_E}], s_1) \perp (\mathsf{cap}(C[E])[D],\ell)$$

が得られる．さらに，**補題** 2 を適用して

$$(C{\uparrow_E}, s_1) \perp (D,\ell)$$

その一方，s_1 は非変数であり ℓ も書換え規則の左辺なので非変数であるから，(S2) に**補題** 4 が適用できて

$$(C{\uparrow_E}, s_1) \not\perp (D,\ell)$$

が得られ，矛盾が生じる．したがって (A2) が成立する． □

補題 7 パターン項 s，項 t，書換え規則 $\ell \to r$，文脈 C が与えられたとする．このとき，もし $s \looparrowright_{C,\ell\to r} t$ が成り立つならば，ある硬文脈 E が存在して $s \rightsquigarrow_{C,E,\ell\to r} t$ が成り立つ．

証明　$D' \stackrel{\mathrm{def}}{=} \mathsf{sub}(D)$ と置く．ある非変数項 s_1 が存在して，(S0)〜(S4) の中で D を D' で置き換えたものが満たされることを示す．まず，(S0) は，(A0) からただちにいえる．また，仮定より，(S4) が成り立つ．s がパターン項であるという仮定と，(A1)，(A2) とより，**補題** 5 の前提は満足される．したがって**補題** 5 より (S1) と (S2) を満たす E と非変数項 s_1 があることが分かる．最後に，**補題** 5 によって得られる等式

$$\mathsf{cap}(C[E]) = \mathsf{cap}(D)$$

に，D' の定義とセパレーションを適用し，(A3) から (S3) が得られる．□

上記の 2 つの補題は，次の定理にまとめられる．これは，本論文の主要結果である．

定理 1　パターン項 s，項 t，書換え規則 $\ell \to r$，文脈 C が与えられたとする．このとき，もし $s \looparrowright_{C,\ell\to r} t$ が成り立つならば，そして，そのときに限り，ある硬文脈 E が存在して $s \leadsto_{C,E,\ell\to r} t$ が成り立つ．□

文献 12) において，Nipkow の HRS に対して，Prehofer も，一般の高階項に対する抽象的なナローイングの定義と，パターンに制限した場合の定義の両方を与えている．しかし，両者の関係についてはまったく言及がなく，パターンを仮定すれば前者から後者が導出できるのか，といった疑問に答えていない．

一方，本論文においても，2 通りの定式化を議論しているが，両者の間の関係を厳密に与えているのが，上の定理である．

上の定理で，s がパターンであっても t はパターン項とは限らない．しかし，さらに C がパターンであり，$\ell \to r$ がパターン書換え規則である場合には，t もパターン項になる．よって，以下の系を得る．

系 1　パターン書換え系 $\mathcal{R}$，パターン項 s，項 t，パターン代入 C に対して，$s \stackrel{*}{\looparrowright}_{C,\mathcal{R}} t$ そして，そのときに限り $s \stackrel{*}{\leadsto}_{C,\mathcal{R}} t$ が成立する．□

6. おわりに

本論文では，抽象高階書換え系におけるナローイングの定式化を初めて提示した．さらに，任意の高階項に対する抽象的な（ナレックスを用いない）定式化とパターンに対する具体的な（ナレックスを用いる）定式化の 2 通りを論じ，それらが，等価になるための条件を明らかにした．

本研究の今後の課題として，まず以下の 3 つがあげられる．

1 つは，本論文で導入したナローイングによる求解の完全性が成立するための条件を明らかにすることである．従来のナローイングでカバーできない求解問題が，本論文で導入した抽象的なナローイングでいかに取り扱えるかを，具体的に明らかにしていく必要がある．

もう 1 つは，代入計算として $\lambda_{\beta\eta^{-1}}$ 以外の任意の計算系を用いる場合への一般化である．特にコンビネータ論理やグラフ書換え系は，関数プログラミング言語の実装方法として広く用いられているので，これらを代入計算として統一的に扱えるようにすることの意義は大きいと考えられる．

最後は，実際的な効率を考慮した改良である．効率を考慮した高階の遅延ナローイング手続き[8] の一般化や，更なる効率化を考察する際に，本研究の理論的結果がおおいに活用可能であると期待できる．

謝辞　貴重なコメントに対して査読者に感謝する．本研究は，科研費基盤研究（C）（2）13680388，15500014 と堀情報科学振興財団の助成を受けて行われた．

参　考　文　献

1) Baader, F.: Unification, Weak Unification, Upper Bound, LowerBound, and Generalization Problems, *4th RTA*, LNCS 488, pp.86–97, Springer (1991).

2) Baader, F. and Nipkow, T.: *Term Rewriting and All That*, Cambridge University Press (1998).

3) Baader, F. and Siekmann, J.H.: Unification Theory, *Handbook of Logic in Artifical Intelligence and Logic Programming*, Gabbay, D., et al. (Eds.), pp.41–125, Oxford University Press (1994).

4) Barendregt, H. P.: *The Lambda Calculus, Its Syntex and Semantics* (*revised editon*), North-Holland (1985).

5) Barendregt, H.P.: Lambda Calculi with Types, *Handbook of Logic in Computer Science, Volume 2*, Abramsky, S., et al. (Eds.), pp.117–309, Oxford University Press (1992).

6) Fay, M.: First-Order Unification in an Equational Theory, *CADE'79*, pp.161–167, Springer (1979).

7) Klop, J.W.: Term Rewriting Systems, *Handbook of Logic in Computer Science, Volume 2*, Abramsky, S., et al. (Eds.), pp.1–116, Oxford University Press (1992).

8) Marin, M., Ida, T. and Suzuki, T.: On Reducing the Search Space of Higher-Order Lazy

Narrowing, *FLOPS'99*, LNCS 1722, pp.319–334, Springer (1999).
9) Middeldorp, A. and Hamoen, E.: Completeness Results for Basic Narrowing, *Applicable Algebra in Engineering, Communication and Computing*, Vol.5, pp.213–253 (1994).
10) Miller, D.: A Logic Programming Language with Lambda-Abstraction, Function Variables, and Simple Unification, *Extentions of Logic Programming*, LNCS 475, pp.253–281, IEEE (1991).
11) Nipkow, T.: Higher-Order Critical Pairs, *LICS91*, pp.342–349, IEEE (1991).
12) Prehofer, C.: *Solving Higher-Order Equations*, Birkhauser (1997).
13) Siekmann, J.H.: Unification Theoty, *J. Symbolic Computation*, Vol.7, pp.207–274 (1989).
14) Slagle, J.R.: Automatic Theorem Proving in Theories with Simplifiers, Commutativity and Associativity, *JACM*, Vol.21, pp.622–642 (1974).
15) Snyder, W.: *A Proof Theory for General Unification*, Birkhäuser (1991).
16) van Oostrom, V.: Confluence for Abstract and Higher-Order Rewriting, Ph.D. Thesis, Vrije Universiteit, Amsterdam (1984).
17) van Oostrom, V.: Development Closed Critical Pairs, *HOA'95*, LNCS 1074, pp.185–200, Springer (1995).
18) van Oostrom, V.: Developping Developments, *Theoretical Computer Science*, Vol.175, No.1, pp.159–181 (1997).

(平成 15 年 5 月 20 日受付)
(平成 15 年 8 月 15 日採録)

奥居　哲（正会員）

昭和 42 年生．平成 7 年筑波大学大学院博士課程工学研究科電子・情報工学専攻修了．博士（工学）．平成 8 年三重大学工学部情報工学科助手．平成 11 年より講師．平成 13 年中部大学工学部情報工学科助教授．書換え計算系等に関する研究に従事．ソフトウェア科学会会員．

鈴木　太朗（正会員）

昭和 39 年生．博士（理学）．平成 7 年筑波大学電子・情報工学系助手．平成 10 年北陸先端科学技術大学院大学助手．平成 12 年東北大学電子通信研究所助手．平成 13 年より会津大学コンピュータ理工学部講師．関数・論理プログラミング言語の基礎理論と実装に関する研究に従事．ACM，ソフトウェア科学会各会員．