限定継続を用いたフォーカスおよび inverse scope の分析と実装

叢悠悠　(指導教員：戸次大介)

1　はじめに

フォーカスや inverse scope といった言語現象は，その意味表示を与える際に特定の語彙項目に対するコンテキストが必要となる．意味表示とは文の真理条件を論理式で表現したものであるが，この中である項のまわりのコンテキストは，その項に対する残りの計算とみなすことができる．この「残りの計算」のことをプログラミングでは「継続」とよぶ．本研究では限定継続命令 `shift/reset` [3] を用いてフォーカスおよび inverse scope を含む文の意味表示を与える．また，これらの意味表示を OchaCaml [4] で実行し，正しく簡約されることを示す．

2　継続

継続とは，計算のある時点における残りの計算のことを指す．たとえば，$1+(2*3)$ という計算の $2*3$ の部分を実行しているときの継続は「現在実行している部分の結果に 1 を足す」という計算になる．この計算は $\lambda x.(1+x)$ という関数で表すことができる．

継続のうち，範囲の限られたものを限定継続とよぶ．`shift/reset` は限定継続を扱うためのオペレータであり，`shift` は継続を切り取る命令，`reset` は `shift` が切り取る継続の範囲を定める命令である．例として $1+\mathit{reset}\,(2*\mathit{shift}\,k.\,(3+k\ 4))$ という計算を考えると，`shift` によって切り取られる継続 k は，`shift` のまわりの計算のうち，`reset` で囲まれた範囲のもの，つまり $\lambda x.(2*x)$ という関数になる．この式を簡約すると 12 になる．

OchaCaml は `shift/reset` を使うことのできる ML 系言語である．本研究では OchaCaml 上で `shift/reset` を用いてフォーカスおよび inverse scope を含む文の意味表示を記述し，それらがどのように簡約されるかを観察する．

3　フォーカス

フォーカスとは，文の中で新しい，あるいは重要な情報となる部分のことである．*only* や *also* といった副詞は，フォーカスを伴う機能語として知られている ([6])．以下，*only* と *also* のフォーカスを $[\ ]_{\mathrm{F_o}}$, $[\ ]_{\mathrm{F_a}}$ で表すとし，例文 (1)-(4) を考える．

(1) Mary only introduced $[\mathrm{Bill}]_{\mathrm{F_o}}$ to Sue.
（Mary は Bill だけを Sue に紹介した.）

(2) Mary only introduced Bill to $[\mathrm{Sue}]_{\mathrm{F_o}}$.
（Mary は Bill を Sue だけに紹介した.）

(3) Mary also introduced $[\mathrm{Bill}]_{\mathrm{F_a}}$ to Sue.
（Mary は Sue に Bill も紹介した.）

(4) Mary also only introduced $[\mathrm{Bill}]_{\mathrm{F_o}}$ to $[\mathrm{Sue}]_{\mathrm{F_a}}$.
（Mary は Sue にも Bill だけを紹介した.）

(1) と (2) は *only* のフォーカスが Mary の紹介した人と，Mary が誰かを紹介した先の人のどちらになるかによって真理条件が異なっている．(3) では *also* によって「Bill 以外に Mary が Sue に紹介した人物が存在する」という前提がもたらされる．同様に，(4) は「Sue 以外に Mary が Bill だけを紹介した人物が存在する」という前提をもつ．

直感的に，*only* は「その命題を満たすのはフォーカスされたもののみである」という意味をもつが，これは *only* がフォーカスに対するコンテキスト，つまりフォーカスの部分が “hole” となった命題を要求し，hole にフォーカスされた語句を代入した命題は真，他の語句を代入したものは偽としていることを意味する．また，*also* は「フォーカス以外にその命題を満たすものが存在する」という前提を引き起こす．言い換えれば，*also* のフォーカスを他の語句に置き換えた命題の中で，真になるものが少なくとも 1 つ存在することを意味する．これらをふまえて，*only* のフォーカスに対する意味表示と *also* のフォーカスがもたらす前提を以下のように記述することができる ([2, 7])．

(5) $[M]_{\mathrm{F_o}} \stackrel{\mathrm{def}}{\equiv} \mathit{shift}\,k.\forall x\,(k\,x \leftrightarrow x = M)$

(6) $[M]_{\mathrm{F_a}} \stackrel{\mathrm{def}}{\equiv} \mathit{shift}\,k.\exists y\,(k\,y \wedge \neg(y = M))$

フォーカスを `shift` で表し，副詞のスコープを `reset` で囲むことにより，継続 k はフォーカス部分が hole となった命題になる．(5) は「hole の部分に何らかの語句を代入した命題が成り立つのは，それがフォーカスされた語句であるとき，かつそのときのみである」ということを意味する．また，(6) は「hole の部分に代入してその命題が成り立つような語句がフォーカスされた語句以外に存在する」ということを表す．

この定義に従って (1) および (4) の意味表示を記述し，OchaCaml で実行すると，それぞれ以下のように簡約される（なお，m, b, s は名前 Mary, Bill, Sue の指示対象である）．

(1) $\mathit{reset}\,(\mathit{introduce}\,(m, \mathit{shift}\,k.\forall x\,(k\,x \leftrightarrow x = b), s))$
$= \forall x\,(\mathit{introduce}\,(m, x, s) \leftrightarrow x = b)$

(4) $\mathit{reset}\,(\mathit{introduce}\,(m, \mathit{shift}\,k.\forall x\,(k\,x \leftrightarrow x = b),$
$\mathit{shift}\,k'.\exists y\,(k'\,y \wedge \neg(y = s))))$
$= \exists y(\forall x(\mathit{introduce}(m, x, y) \leftrightarrow x = b) \wedge \neg(y = s))$

さらに，1 つの語句が複数の副詞にフォーカスされている場合もある．以下がその例である．

(7) Sue has also thought that John only loved $[[\mathrm{Mary}]_{\mathrm{F_a}}]_{\mathrm{F_o}}$.

(7) は「Mary 以外に John が彼女だけを愛していたと Sue がかつて思っていた人が存在する」という読みをもつ．$[[\mathrm{Mary}]_{\mathrm{F_a}}]_{\mathrm{F_o}}$ がネストしたフォーカスになっており，外側が *only*，内側が *also* に対応している．ここで，*only* のスコープは “John loved x” という部分に限られるが，*also* のスコープは Sue の信念までを含んだ “Sue has thought that John only loved x” とい

う範囲である．つまり，(7) では 2 つの副詞のフォーカスに対するコンテキストが異なっている．これらのコンテキストを区別するためには，階層付けした 2 つの限定継続命令が必要になる．そこで，[8] は OchaCaml で通常の `shift/reset` にあたる `shift1/reset1` および 1 つ階層が上の `shift2/reset2` を定義し，(7) の意味表示を (8) のように与えた．

(8) $\mathit{reset2}\,(\mathit{think}\,(s,$
$\quad \mathit{reset1}\,(\mathit{love}\,(j, \mathit{shift1}\ k.\forall x\,(k\,x \leftrightarrow x =$
$\qquad \mathit{shift2}\ k'.\exists y\,(k'\,y \wedge \neg(y = m)))))))$
$= \exists y\,(\mathit{think}\,(s,$
$\qquad \forall x\,(\mathit{love}\,(j, x) \leftrightarrow x = y) \wedge \neg(y = m)))$

only と *also* のフォーカスをそれぞれ `shift1/reset1`, `shift2/reset2` で記述することで，*also* のフォーカスに含まれる `shift2` が `reset1` で囲まれた *only* のスコープを越えて `reset2` までの継続を切り取ることが可能になる．

4 Inverse scope

複数の量化表現を含む文においては，それぞれの量化表現のスコープ関係が表層的な順序と一致した surface scope reading と，順序が逆転した inverse scope reading をともにもつことがある．以下の例文を考える．

(9) Some woman loves every man.
（ある女性が全ての男性を愛している．）

a. $\exists x\,(\mathit{woman}\,(x) \wedge \forall y\,(\mathit{man}\,(y) \rightarrow \mathit{love}\,(x, y)))$
（ある特定の女性が全ての男性を愛している．）

b. $\forall y\,(\mathit{man}\,(y) \rightarrow \exists x\,(\mathit{woman}\,(x) \wedge \mathit{love}\,(x, y)))$
（全ての男性について，彼を愛する女性が存在する．）

(9a) は *some* と *every* のスコープが文中の順序と一致した surface scope reading である．一方，(9b) では *every* のスコープが *some* より上になっている．これが (9) に対する inverse scope reading である．

[5] は論理形式とよばれる表示を用いて文中に含まれる量化表現の相対的なスコープを表現している．たとえば，(9b) の読みは (9c) のような論理形式で表される（$[_{\mathrm{S}}\ ]$ と $[_{\mathrm{NP}}\ ]$ はそれぞれ [ ] 内が文および名詞句であることを表す）．

(9c) $[_{\mathrm{S}}\,[_{\mathrm{NP}}\,\text{every man}]_3\ [_{\mathrm{S}}\,[_{\mathrm{NP}}\,\text{some woman}]_2\ [_{\mathrm{S}}\,e_2\ \text{loves}\ e_3]]]$

(9c) の 2 つ目の S ノードと 3 つ目の S ノードはそれぞれ *every man*, *some woman* に対するコンテキストである．この場合，自身のコンテキストに他の量化表現を含んでいる *every man* が上のスコープをとる．

[1] は各語彙項目に対する継続を明示化し，文を構成する際の実行順序を指定することで surface/inverse scope reading を導出しているが，3 つの量化表現を含む文において，主語位置の量化表現のスコープが他の 2 つの間に入る読み（本稿では intermediate-inverse scope reading とよぶ）を導出できない．この問題を回避するために，[9] は inverse scope reading を導出するための INV オペレータを (10) のように定義した（e は個体，t は真偽値の型である）．

(10) $[f]_{\mathrm{INV}} \stackrel{\mathrm{def}}{\equiv} \mathit{shift}\,k.\,f\,k \quad (f : (e \rightarrow t) \rightarrow t)$

INV オペレータは量化名詞句 f を引数として受け取る．`shift` によって k に束縛される継続は，f のまわりのコンテキストである．このコンテキストが f に渡されることで，f に含まれる量化表現が上のスコープをとる表示となる．`shift` オペレータが含まれるため，実行する際は式全体を `reset` で囲む必要がある．

INV オペレータを用いると，量化表現を 3 つ含む文における intermediate-inverse scope reading を導出できる．以下がその例である（every_{pp} は前置詞句に含まれる *every* を表す）．

(11) Some teachers introduce [most students]$_{\mathrm{INV}}$ to every company.
$\mathit{reset}\,(\mathit{some\,teacher}\,(\mathit{every}_{pp}\,\mathit{company}$
$\qquad (\mathit{introduce}\,(\mathit{inv}\,(\mathit{most\,student}))))$
$= \mathit{most}\,z\,(\mathit{student}\,(z), \exists y\,(\mathit{teacher}\,(y) \wedge$
$\qquad \forall x\,(\mathit{company}\,(x) \rightarrow \mathit{introduce}\,(y, z, x))))$

5 おわりに

本研究ではフォーカスあるいは inverse scope のどちらか一方を含む文について，`shift/reset` で意味表示を与えた．今後はフォーカスと inverse scope をともに含む文における両者の相互作用について考察したいと考えている．

参考文献

[1] Barker, C.: Continuations and the Nature of Quantification, *Natural Language Semantics* 10(3), pp. 211–241 (2002).

[2] Bekki, D. and K. Asai: Representing Covert Movements by Delimited Continuations, In: K. Nakakoji, Y. Murakami, and E. McCready (eds.): *New Frontiers in Artificial Intelligence (JSAI-isAI 2009 Workshops, Tokyo, Japan, November 2009, Selected Papers from LENLS 6)*, Vol. LNAI 6284, pp. 161–180 (2010).

[3] Danvy, O. and Filinski, A.: Abstracting Control, In: *LFP90, the 1990 ACM Conference on Lisp and Functional Programming*, pp. 151–160 (1990).

[4] Masuko, M. and K. Asai: Caml Light + shift/reset = Caml Shift, Theory and Practice of Delimited Continuations (TPDC 2011), pp. 33–46 (2011).

[5] May, R.: *The Grammar of Quantification*, Doctoral dissertation, MIT, Cambridge (1977).

[6] Rooth, M.: Focus, *The Handbook of Contemporary Semantic Theory*, pp.271–298 (1997).

[7] 叢悠悠, 浅井健一, 戸次大介: 限定継続を用いたフォーカスの分析と実装に向けて, 情報処理学会第 214 回自然言語処理研究会 (2013).

[8] 叢悠悠, 浅井健一, 戸次大介: 限定継続を用いたフォーカスの分析と実装, 第 16 回プログラミングおよびプログラミング言語ワークショップ論文集掲載予定 (2014).

[9] 叢悠悠, 浅井健一, 戸次大介: 限定継続を用いた inverse scope の分析と実装, 言語処理学会第 20 回年次大会発表論文集掲載予定 (2014).