TM-1275

演繹オブジェクト指向データベース Quixote の手続き的意味論

西岡　利博 (MRI)、小島　量、
津田　宏、横田　一正

ICOT
Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan
(03)3456-3191 ~ 5

Institute for New Generation Computer Technology

演繹オブジェクト指向データベース言語 *Quixote* の手続き的意味論

西岡 利博　　小島 量　津田 宏　横田 一正
(株) 三菱総合研究所　(財) 新世代コンピュータ技術開発機構

Quixote は、オブジェクトの概念の定式化を行なう、包摂関係に基づく制約を扱う論理型プログラミング言語である。知識情報処理に応用できる演繹オブジェクト指向データベース言語およびその処理系として ICOT で開発している。本稿では *Quixote* 言語に公理的意味論と手続き的意味論を与え、意味論面から考えられる今後の改良について考察する。

Procedural Semantics of a DOOD Programming Language *Quixote*

Toshihiro Nishioka　　Ryo Ojima　Hiroshi Tsuda　Kazumasa Yokota
Mitsubishi Research Institute, Inc.　Institute for New Generation Computer Technology

Quixote is a logic programming language with subsumption constraints, by which concepts of an object are formulated. ICOT has been developing the language and its experimental system in the framework of deductive object-oriented databases (DOOD) for knowledge information processing applications. This short paper gives both the axiomatic and the procedural semantics of the language, and consisder some improvements for the implementation from a semantic point of view.

1 はじめに

演繹オブジェクト指向データベースの枠組で知識を表現し、これを体系的に集積することにより、知識ベースとして機能させる研究を ICOT で行なっている。そのために、ある概念の持つ性質をファクトやルールとして記述する言語 QUIXOTE[8][7] を開発した。

これまでに、法的推論 [10]、自然言語処理 [5][9]、分子生物学データベース [4] などの応用例があり、 QUIXOTE の持つオブジェクトの表現能力、モジュール機能、多重継承機能、問い合わせ処理機能などを利用してそれぞれに成果をあげている。

しかし、個々の応用の要求に応えるための努力と比較して、 QUIXOTE の意味論に関する考察は十分であったとは言えない。本稿では QUIXOTE の意味論を検討することにより、現在の QUIXOTE 処理系の実装が持つ論理的意味を考察する基礎を与える。その前に、 2節で、 QUIXOTE の構文と直観的な意味について簡単に説明する。続いて 3節では、現在の QUIXOTE 処理系の実装が与える解を説明する公理的意味論を与える。これをもとに、 4 節では、 QUIXOTE の手続き的意味論を与え、これが公理的意味論によって証明されることを示す。また、解の極小性に関する定義を与えることで、 QUIXOTE 処理系の実装が出力すべき解について示唆を与える。最後に 5 節では、本稿で提示された意味論と現在の QUIXOTE 処理系の実装の対比を行なう。

2 QUIXOTE 言語

本節では、 QUIXOTE の言語仕様の概要について簡単に説明する。詳細は [7], [11] を参照されたい。

2.1 オブジェクト項

QUIXOTE では概念をオブジェクトとして表現する。オブジェクト項には**基本オブジェクト項**と、**複合オブジェクト項**がある。

リンゴ, 青, 塩素

これらは基本オブジェクト項である。典型的な応用では、これらはそれぞれ“リンゴ”、“青”、“塩素”という概念を表すのに使用される。

リンゴ[色 = 緑]
猫[産地 = ヒマラヤ, 性別 = オス]

これらは複合オブジェクト項である。元となる基本オブジェクト項を、特にそのオブジェクトに本質的な属性 (**固有属性**と言う) で修飾することで得られる。上の例は、それぞれ“青リンゴ”、“ヒマラヤ産のオス猫”を表す。固有属性の順序を入れ替えた複合オブジェクト項どうしは、同じ複合オブジェクト項である。

リンゴ, 猫 などの元となる基本オブジェクト項を、複合オブジェクト項の**頭部**と言う。 産地, 性別 などの、固有属性を特定するものを**ラベル**と言う。ラベルには基本オブジェクト項を用いることができる。“ヒマラヤ”、“オス”などは、対応するラベルに関する**属性値**と言う。属性値は複合オブジェクト項または基本オブジェクト項である。

2.2 包摂関係

QUIXOTE では、概念間の上下関係を基本オブジェクト項の半順序として定義する。この半順序を**包摂関係**と言い、上位のオブジェクトは下位のオブジェクトを**包摂**すると言う。記述例を示す。

リンゴ $\sqsubseteq$ 果物;;
青 $\sqsubseteq$ 色;;

それぞれ、リンゴが果物の一種であること、青が色の一種であることを表す。

包摂関係は半順序として定義されるが、意味論はこれを束として扱う。半順序から束を構成するために、一般にはいくつかの基本オブジェクト項と、それに対する包摂関係を追加する必要がある。意味論で扱う基本オブジェクト項とその集合上の束は、そのように拡張されたものである。半順序から束を構成する方法については [1] に述べられている。

包摂関係は複合オブジェクト項にも拡張される。二つの複合オブジェクト項 e, f があるとき、 e の頭部が f の頭部に包摂され、 f の固有属性に存在するラベルは必ず e にもあり、かつ、同じラベルの属性値は e のものが f のものに包摂されているなら、 e は f に包摂される。以下に例を示す。

猫[産地 = ヒマラヤ, 性別 = オス] $\sqsubseteq$ 動物[産地 = 高地]

(上の例では、 猫 $\sqsubseteq$ 動物, ヒマラヤ $\sqsubseteq$ 高地 という包摂関係が仮定されている。)

包摂関係が束であるならば、任意のオブジェクト項間に上限と下限が定義できなくてはならない。複合オブジェクト項の場合は次のように拡張される。二つのオブジェクト項 e, f があるとき、その上限 ($e \uparrow f$ と書いて表す) は、頭部が双方の頭部の上限であり、双方に共通な固有属性だけを持ち、その属性値はそれぞれの属性値の上限であるような複合オブジェクト項 (共通のラベルがなければ基本オブジェクト項) である。また、下限 ($e \downarrow f$ と書いて表す) は、頭部が双方の頭部の下限であり、少なくともどちらか一方に存在する固有属性すべてを持ち、その属性値は、共通な固有属性であれば双方の下限、一方だけに存在する固有属性であればその属性値そのものであるような複合オブジェクト項である。これが上の複合オブジェクト項間の包摂関係の定義に矛盾しないことは明らかである。

2.3 属性値

$o.l$ と書いてオブジェクト項 o のラベル l に関する属性値を参照できる。このようにオブジェクト項とラベルの間に "." を中置した項をドット項と言う。

QUIXOTE のオブジェクト項の属性値は、固有属性の属性値であるか、または固有属性として記述されていない場合、付帯属性の属性値である。付帯属性とは、ある概念の持つ付帯的な性質であって、次項で述べる制約による表記や、2.5 項で述べる属性項による表記で記述することができる。

2.4 制約

QUIXOTE では、オブジェクト項、ドット項、変数の間の制約を記述できる。制約には、オブジェクト項の包摂関係に基づく制約と、オブジェクト項の集合上の Hoare 順序[1]に基づく制約がある。制約記号とその意味を示す。

表記	意味
$x \sqsubseteq y$	x は y に包摂される。
$x \sqsupseteq y$	$y \sqsubseteq x$
$x \cong y$	$x \sqsubseteq y \wedge y \sqsubseteq x$
$x \sqsubseteq_H y$	x は y より Hoare 順序で小さい。
$x \sqsupseteq_H y$	$y \sqsubseteq_H x$
$x \cong_H y$	$x \sqsubseteq_H y \wedge y \sqsubseteq_H$

制約の記述例を示す。

$$
\begin{array}{l}
\{\text{リンゴ}.\text{色} \cong \text{赤}\} \\
\{X \sqsubseteq_H \{\text{猫}[\text{産地} = \text{ヒマラヤ}], \\
\qquad \text{犬}[\text{産地} = \text{アラスカ}]\}\} \\
\{\text{太郎}.\text{父} \cong Y, \\
\ Y.\text{ペット} \sqsubseteq \text{犬}[\text{産地} = \text{アラスカ}]\}
\end{array}
$$

上の二つはそれぞれ、リンゴの色は赤である、変数 X はヒマラヤ産の猫またはアラスカ産の犬などの集合である、という制約である。最後のは、太郎の父親がアラスカ産の犬を飼っていることを表している。

2.5 属性項

付帯属性に関する制約の略記法が用意されている。

オブジェクト項の後に / を置き、続いて $[\cdots]$ の中に *label op value* の列を記述する。*label* は付帯属性を特定するラベルであり、基礎オブジェクト項を用いることができる。*value* はオブジェクト項である。このような表記を属性項と言う。固有属性とは異なり、*label* と *value* とに中置される関係演算子を変えることで、属性値の制約を記述できる。*op* とその意味を示す。

表記	意味	表記	意味
$o/[l \rightarrow v]$	$o.l \sqsubseteq v$	$o/[l \rightarrow_{\mathbb{h}} v]$	$o.l \sqsubseteq_{\mathbb{h}} v$
$o/[l \leftarrow v]$	$o.l \sqsupseteq v$	$o/[l \leftarrow_{\mathbb{h}} v]$	$o.l \sqsupseteq_{\mathbb{h}} v$
$o/[l = v]$	$o.l \cong v$	$o/[l =_{\mathbb{h}} v]$	$o.l \cong_{\mathbb{h}} v$

2.6 ルール

H をオブジェクト項、Ch を制約の集合、B をオブジェクト項の列、Cb を制約の集合とするとき、QUIXOTE のルールは、$H/\!/Ch \Leftarrow B \parallel Cb;;$ と記述される。H を頭部、Ch を頭部制約、B を本体、Cb を本体制約と呼ぶ。直観的には、"B, Cb が満たされると H, Ch が成り立つ" ことを主張していると解釈される。;; はルールの終りを表す。ルールの記述例を示す。

$$
\begin{array}{l}
\text{太郎}/\!/\{\text{太郎}.\text{父} \cong \text{浩}\};; \\
X/\!/\{X.\text{取り引き} \cong \text{無税}\} \\
\quad \Leftarrow X \parallel \{X.\text{アルコール性} \cong \text{無し}, X \sqsubseteq \text{飲料}\};;
\end{array}
$$

最初のルールは 太郎 の父親が 浩 であることを表している。このような本体と本体制約のないルールをファクトと呼ぶ。二番めのルールは "ノン・アルコール飲料は無税" を表している。前項で述べたように、これらのルールは通常以下のように略記される。

$$
\begin{array}{l}
\text{太郎}/[\text{父} = \text{浩}];; \\
X/[\text{取り引き} = \text{無税}] \\
\quad \Leftarrow X/[\text{アルコール性} = \text{無し}] \parallel \{X \sqsubseteq \text{飲料}\};;
\end{array}
$$

2.7 制約のマージ

直観的に言って、あるオブジェクトの属性値は、そのオブジェクト項を頭部に持ち、本体と本体制約が満たされるすべてのルールの頭部制約の影響を総合したものになる。例えば、二つのファクト

$$
\begin{array}{l}
o/[l \rightarrow a, l_1 = a];; \\
o/[l \rightarrow b, l_2 = b];;
\end{array}
$$

があったなら、それは

$$
o/[l \rightarrow a \downarrow b, l_1 = a, l_2 = b];;
$$

という一つのファクトがあったのと同じことである (ここで、a, b はオブジェクト項)。この例の $o.l$ のように、複数のルールから得られる同じ属性値に関する制約が総合されることを特に制約のマージと言う。

2.8 属性継承

QUIXOTE の属性継承は、DOT [6] で行なわれているものと同様であり、次のような一般規則で表される。すなわ

[1]本来の Hoare 順序 $\sqsubseteq_H$ は、集合 R, S について、$R \sqsubseteq_H S \Leftrightarrow \forall r \in R, \exists s \in S.\ r \sqsubseteq s$ であるような集合間の順序である。定義から半順序ではないが、$r \sqsubseteq_H s \wedge s \sqsubseteq_H r \Leftrightarrow r \cong_H s$ と定義して、$\cong_H$ に関する同値類を同じ集合と考え、同値類上の関係と定義し直せば半順序と解釈できる。本稿ではこのように再定義された順序を Hoare 順序と呼んでいる。

ち、任意のオブジェクト項 o,p に対して、任意のラベル l について以下が成り立つ。

$$o \sqsubseteq p \Rightarrow o.l \sqsubseteq p.l$$

3 公理的意味論

前節で述べたような特徴を持つ *Quixote* に対し、本節では公理的意味論を与える。 sequent calculus LJ を元にし、そこに *Quixote* に固有の公理を加える方針で設計した。

本稿では厳密な定式化を行なうことよりも、むしろ意味論の骨子を述べることに焦点を当てる。より厳密な定式化については [12] を準備中である。

3.1 制限

3.1.1 集合

Quixote では、 Hoare 順序に基づく集合間の制約を扱うことができる。しかし意味論では二階の対象を扱わないようにするために、集合を集合としては扱わず、通常の項として意味を与えている。このため、通常のオブジェクト項と集合項との間の制約を生じさせないために記述がやや繁雑になる。本稿では、見通しを良くするため、集合間の制約に対する意味論については割愛した。

3.1.2 属性継承

複数のルールから得られる制約のマージに意味を与えるので、属性継承に対する意味論をこれの延長で与えることはやさしい。ただし記述が繁雑になるので、本稿では厳密にこれを扱うのは避け、本来それが要求されるところに簡単に補足するにとどめた。

3.1.3 モジュール

Quixote では、ルールをモジュールという単位に分割して記述することができる。モジュールは単なる知識の分類の枠組にとどまらない、モジュール間でルールの継承を行なうことができ、知識の階層的記述や試行錯誤的結合などに有効な機能である。しかし、モジュール間の演算により生じる影響は問い合わせの発行前に事前評価することができるので、本稿では触れない。

3.1.4 更新機能

Quixote ではルールの中にファクトの追加や削除を行なったり、一貫性制約の検査を行なうような制御子を記述することができるが、本稿では触れない。

3.2 プログラム

以下では、 *Quixote* のプログラムを、 (BO^*, L, R) の三項組であるとして考える。 BO^* は、基本オブジェクト項の有限集合 BO の上の束 $(BO, \preceq, \top, \bot)$ である。順序 $\preceq$ は基本オブジェクト項間の包摂関係を表す。 L は有限なラベルの集合である。 R は BO と L の要素を用いて作られたルールの集合である。

3.3 公理的意味論の文法

公理的意味論の持つ文法について、簡単に述べる。

項 オブジェクト項、ドット項、変数項は項である。

論理式 t_1, t_2 を項とするとき、 $t_1 \sqsubseteq t_2$、 $t_1 \cong t_2$ は制約論理式と呼ばれる論理式である。

t を項とするとき、 exist(t) は存在論理式と呼ばれる論理式である。以下ではこれを t と略して表記する。

A, B が論理式であるとき、 $A \wedge B$ と $A \supset B$ は論理式である。

自由変数 F を含む任意の論理式 $P(F)$ について、 $(\forall X)P(X)$, $(\exists X)P(X)$ は論理式である。

sequent Γ が論理式の有限列 (空でも良い) であって、 Θ が論理式(空でも良い) であるとき、 $\Gamma \rightarrow \Theta$ は sequent である。

3.4 公理

Quixote の公理的意味論には以下に述べるような公理がある[2]。

LJ の公理

A を論理式とすると、次の sequent は公理である。

$$A \rightarrow A$$

制約式の公理

t, t_1, t_2, t_3 を項とし、 $P(a)$ を、項 a を含む任意の論理式とすると、次の各 sequent は公理である。

$$\begin{aligned}
&\rightarrow t \cong t \\
&t_1 \cong t_2 \rightarrow t_2 \cong t_1 \\
&t_1 \cong t_2 \rightarrow t_1 \sqsubseteq t_2 \\
&t_1 \sqsubseteq t_2, t_2 \sqsubseteq t_1 \rightarrow t_1 \cong t_2 \\
&t_1 \sqsubseteq t_2, t_2 \sqsubseteq t_3 \rightarrow t_1 \sqsubseteq t_3 \\
&t \sqsubseteq t_1, t \sqsubseteq t_2 \rightarrow t \sqsubseteq t_1 \downarrow t_2 \\
&t_1 \sqsubseteq t, t_2 \sqsubseteq t \rightarrow t_1 \uparrow t_2 \sqsubseteq t \\
&t_1 \cong t_2 \wedge P(t_1) \rightarrow P(t_2)
\end{aligned}$$

[2]継承を扱う場合には、これらの他に継承の一般規則 $o \sqsubseteq p \rightarrow o.l \sqsubseteq p.l$ もここに含まれる。

包摂関係に関する公理

オブジェクト項 a, b が $a \sqsubseteq b$ の関係にあるならば、以下の sequent は公理である。

$$\rightarrow a \sqsubseteq b$$

オブジェクト項 a, b が $a \sqsubseteq b$ の関係にないならば、次の sequent は公理である。

$$a \sqsubseteq b \rightarrow$$

ルールの公理

B を存在論理式の連言、Cb を制約論理式の連言、H を存在論理式、Ch を制約論理式の連言、$X_1, \cdots, X_n$ を B, Cb, Ch, H に現れる変数とするとき、次の sequent が $Quixote$ のルールとして与えられ、それは公理である。

$$\rightarrow (\forall X_1) \cdots (\forall X_n)(B \wedge Cb \supset H \wedge Ch)$$

問い合わせの公理

B を存在論理式の連言、Cb を制約論理式の連言、$X_1, \cdots, X_n$ を B, Cb に現れる変数とするとき、次の sequent が $Quixote$ の問い合わせとして与えられ、それは公理である。

$$(\exists X_1) \cdots (\exists X_n)(B \wedge Cb) \rightarrow$$

3.5 推論規則

sequent calculus LJ の推論規則のうち、$\neg, \vee$ に関するものを除いた残りすべてが含まれる[3]。

3.6 問い合わせと解

公理から推論規則を用いて定理 $\Gamma \rightarrow$ を証明することが問い合わせに対する解となる。Γ は論理式の有限列で、この解の仮説と言う。問い合わせは一般に存在束縛された変数を含んでいるが、証明図では項 t を含む論理式 $P(t)$ から $(\exists X)P(X)$ を作ることでこれを証明することになる。このとき、X の値は t であると言う。

3.7 証明例

包摂関係から証明可能な例

$Quixote$	論理式
$a \sqsubseteq b$	$\rightarrow a \sqsubseteq b$
$o/[l \rightarrow a]$	$\rightarrow o \wedge o.l \sqsubseteq a$
?- $o/[l \rightarrow b]$.	$o \wedge o.l \sqsubseteq b \rightarrow$

[3]これは、$\neg, \vee$ を含まない一階述語論理と等価である。

証明図

$$\dfrac{\dfrac{\rightarrow a \sqsubseteq b \quad o.l \sqsubseteq a, a \sqsubseteq b \rightarrow o.l \sqsubseteq b}{o.l \sqsubseteq a \rightarrow o.l \sqsubseteq b} \quad o \sqsubseteq o}{o \wedge o.l \sqsubseteq a \rightarrow o \wedge o.l \sqsubseteq b \cdots (1)}$$

$$\dfrac{\dfrac{\rightarrow o \wedge o.l \sqsubseteq a \quad (1)}{\rightarrow o \wedge o.l \sqsubseteq b} \quad o \wedge o.l \sqsubseteq b \rightarrow}{\rightarrow}$$

属性値を求める例

$Quixote$	論理式
$o/[l = a];;$	$\rightarrow o \wedge o.l \cong a$
?- $o/[l = X]$.	$(\exists X)(o \wedge o.l \cong X) \rightarrow$

証明図

$$\dfrac{\dfrac{\rightarrow o \wedge o.l \cong a}{\rightarrow (\exists X)(o \wedge o.l \cong X)} \quad (\exists X)(o \wedge o.l \cong X) \rightarrow}{\rightarrow}$$

制約がマージされる例

$Quixote$	論理式
$c \sqsubseteq a, c \sqsubseteq b$	$\rightarrow c \sqsubseteq a \wedge c \sqsubseteq b$
$o/[l \rightarrow a];;$	$\rightarrow a \supset o \wedge o.l \sqsubseteq a$
$o/[l \rightarrow b];;$	$\rightarrow b \supset o \wedge o.l \sqsubseteq b$
?- $o/[l \rightarrow c]$.	$o \wedge o.l \sqsubseteq c \rightarrow$

証明図

$$\dfrac{\dfrac{\rightarrow o \wedge o.l \sqsubseteq a \quad \rightarrow o \wedge o.l \sqsubseteq b}{\rightarrow o \wedge o.l \sqsubseteq c} \quad o \wedge o.l \sqsubseteq c \rightarrow}{\rightarrow}$$

4 手続き的意味論

前節で述べた公理的意味論をもとに、$Quixote$ に手続き的意味論を与える。頭部制約がマージされる点や、本体制約が制約の検査的に働く点などが $Quixote$ を典型的な CLP として定式化することを妨げている。これに対しては、本体制約を制約集合間の関係を述べている制約のようなものと解釈して専用の処理系を用意することで定式化を試みている。また、異なるルールから得られる制約のマージを行なったり、異なるドット項の間の制約関係を、複数のルールを解くことで充足させたりすることから、通常の論理プログラミングの定式化のように、一度に一つのルールしか選択しない定式化ではうまくいかない。子ノードを生成する際に、複数のノードを一度に展開するような定式化を行なっている。

4.1 手続き的意味論の文法

公理的意味論と同様であるが、以下の点で異なる。

制約 公理的意味論で制約論理式と言っていた $t_1 \sqsubseteq t_2$, $t_1 \cong t_2$ を制約と言う。

関係 C_1, C_2 を制約の集合とするとき、 satisfied(C_1, C_2) を充足関係と言い、 C_2 から C_1 が証明可能であるという関係を表す。 $C_1 \backslash C_2$ と略記する。

4.2 導出

導出の目的は、五項組 $N_i = (G_i, K_i, S_i, A_i, C_i)$ をノードとする列 $N_1, N_2, \cdots, N_n$ (ただし、 G_n は $\emptyset$) を得ることである。 N_n から解を得る際には終了処理 (4.3.6項) を行なう。

ここで、 G_i はオブジェクト項の集合であり、ゴールと言う。 S_i は充足関係の集合である。 K_i は制約の集合であり、前提と言う。 A_i は制約の集合であり、仮説と言う。 C_i は制約の集合であり、結論と言う。

直観的には、前提はこれから証明すべき制約である。ゴール中のオブジェクト項を頭部に持つルールの頭部制約によって前提を証明するのだが、そのとき証明されるべき前提と、証明しようとする制約集合の組が充足関係として保持される。証明できなかった前提は仮説となる。結論には、仮説を含めて、その仮説のもとで成り立つ制約が蓄積される。

導出とは、 N_i から N_{i+1} を得る手続きであり、これを以下に定める。

1. $N_i = (G_i, K_i, S_i, A_i, C_i)$ がノードであるとする。
2. G_i の要素を一つ選び、 $G_i = \{g_i\} \cup G_i'$ とする。
3. K_i の部分集合を選び、 $K_i = Ks_i \cup K_i'$ とする。直観的には Ks_i はこの導出で証明されるべき前提である。
4. ルールの集合 Re を選び、後で述べる規則にしたがって $N_{i'}$ を生成する。直観的には Re は g と $\cong$ の関係にあるオブジェクト項を頭部に持つルールの集合であって、空であってはならない[4]。
5. 制約処理系が $N_{i'}$ を処理し、次のノード $N_{i+1} = (G_{i+1}, K_{i+1}, S_{i+1}, A_{i+1}, C_{i+1})$ を生成する。

途中、制約集合が矛盾して制約処理に失敗したならば、そのノードからはそれ以上の導出は行なわれない。

4.2.1 $N_{i'}$ の生成

ルールの集合が $Re = \{r_{e_1}, \cdots, r_{e_n}\}$ であって、 $U_i = \bigcup_j \{h_{e_j} \cong g_i\}$ とすると[5]、 $N_{i'} = (G_{i'}, K_{i'}, S_{i'}, A_{i'}, C_{i'})$ は以下の通りである。

$$N_{i'} = (G_i' \cup \bigcup_j B_{e_j}, K_i' \cup \bigcup_j Cb_{e_j}, S_i \cup \{Ks_i \backslash \bigcup_j Ch_{e_j}\}, A_i, C_i \cup \bigcup_j Ch_{e_j} \cup U_i)$$

4.2.2 問い合わせ

B をオブジェクト項の列、 Cb を制約の集合とするとき、

$$?\text{-}B \parallel Cb.$$

は QUIXOTE の問い合わせである。最後の "." は、問い合わせの終りを表す。問い合わせは、頭部と頭部制約のないルールと考えることもできる。問い合わせが発せられると、最初のノード $N_1 = (B, Cb, \emptyset, \emptyset, \emptyset)$ が生成され、以下、次々と導出が行なわれる。

4.2.3 解

ノード $N_n = (\emptyset, K_n, S_n, A_n, C_n)$ があるとき、二項組 $(\{A_n \cup K_n\}, C)$ を解と言う。特に $\{A_n \cup K_n\}$ を仮説部、 C を結論部と言う[6]。

4.3 制約処理

QUIXOTE の制約処理系は、ノード $N_{i'}$ を受け取って制約処理を施し、ノード N_{i+1} を生成する。制約処理は以下の手順で行なわれる。

1. 結論を正規化する。
2. 充足関係集合の各要素 $A_i \backslash B_i$ について、 A_i (または B_i) に出現する各変数に対する結論中の制約をすべて A_i (または B_i) に付加する。
3. 充足関係集合の各要素の解消処理 (4.3.5項) を行なう。

4.3.1 制約の可解性

QUIXOTE の制約には、束上の包摂関係に基づくものと、集合の Hoare 順序に基づくものがあるが、それらは区別されている。

束上の包摂関係に基づく制約については、限量子のない一階制約言語とみなすことができ、この解が求められることは Mukai [2] が証明している。

Hoare 順序に基づく制約も可解であることを示すのは容易である。

---

[4] 簡単のためにこうしているが厳密には、属性継承によって影響を及ぼすルールと、 Ks_i を証明するためだけに必要なルールもここで同時に選択しなければならない。

[5] 属性継承を扱う場合には、 U_i には継承によって影響を与えるルールの頭部のオブジェクト項と g との間の関係も含める。

[6] 結論部には導出に使われたさまざまな変数やドット項に関する制約が累積されているが、ユーザは問い合わせに書いた変数に対する制約以外には興味がないことが多いので、実際にユーザに提示される解は、 C から問い合わせ中の変数に対するものでない制約を取り除いたものが適当であろう。

4.3.2 正規化処理

S を有限な制約の集合とするとき、S の正規形 S^* とは、直観的には次のようなものである。

- ドット項どうしの制約で証明可能なものはすべて含まれる。
- ドット項がオブジェクト項に包摂される制約は、証明可能なものの中の極小元であるオブジェクト項についての制約が含まれる。包摂する場合も同様。

S^* は以下の手続きで求める。

1. 変数割当の適用
2. オブジェクト項の間の制約の評価
3. オブジェクト項の上限、下限の置換
4. 制約の飽和[7]

制約集合に変化が生じなくなるか、または S が矛盾していることが判明するまで、上の手順を繰り返す。手順中、S が矛盾していることが判明したなら、直ちに正規化は失敗する。

以下では、t を任意の項、a,b を互いに異なる基本オブジェクト項、e,f をそれぞれ複合オブジェクト項 $h_e[\cdots]$, $h_f[\cdots]$ とする。

変数割当の適用

P1: 変数項 F、オブジェクト項 o について、S 中に制約 $F \cong o$ または $o \cong F$ があるとき、その制約を除く S 中のすべての F の出現を o で置き換える。

オブジェクト項の間の制約の評価

P2: 以下のいずれかの場合 S が矛盾している。

$$a \cong b \in S$$
$a \sqsubseteq b \in S$であるのに$a \sqsubseteq b$でない
$$a \cong e \in S$$
$$e \cong a \in S$$
$$a \sqsubseteq e \in S$$
$e \sqsubseteq a \in S$であるのに$h_e \sqsubseteq a$でない

P3: $e \cong f \in S$ であるならば、e,f の頭部が同じで、固有属性のラベルも並べ替えの関係になっていなければ S は矛盾している。これらが満たされているなら、対応するラベルの属性値どうしの制約 $t_{e_i} \cong t_{f_i}$ の集合を S に付け加える[8]。

P4: $e \sqsubseteq f \in S$ であるならば、$h_e \sqsubseteq h_f$ であって、f にある固有属性のラベルは e にもなければならない。これらが満たされているなら、共通なラベルの属性値どうしの制約 $t_{e_i} \sqsubseteq t_{f_i}$ の集合を S に付け加える[9]。

オブジェクト項の上限、下限の置換

P5: t を S 中のドット項または変数項、t_1,t_2 をオブジェクト項とする。

このとき $\{t \sqsubseteq t_1, t \sqsubseteq t_2\} \subseteq S$ であれば、S に $\{t \sqsubseteq t_1 \sqcup t_2\}$ を加える。

また、$\{t_1 \sqsubseteq t, t_2 \sqsubseteq t\} \subseteq S$ であれば、S に $\{t_1 \sqcap t_2 \sqsubseteq t\}$ を加える。

t_1,t_2 が変数を含んでいる場合には、制約式を解くことによってその変数に対する制約を求め、S に加える。

制約の飽和

d_1,d_2 を S に現れる互いに異なるドット項、t を d_1,d_2 以外の任意の項とするとき、S が下表の $\Rightarrow$ の左辺の制約集合を含むなら、S に右辺の制約集合を加える。

P6: $\{d_1 \cong d_2\} \Rightarrow \{d_2 \cong d_1\}$

P7: $\{d_1 \cong d_2\} \Rightarrow \{d_1 \sqsubseteq d_2\}$

P8: $\{d_1 \sqsubseteq d_2, d_2 \sqsubseteq d_1\} \Rightarrow \{d_1 \cong d_2\}$

P9: $\{d_1 \sqsubseteq d_2, d_2 \sqsubseteq t\} \Rightarrow \{d_1 \sqsubseteq t\}$

P10: $\{t \sqsubseteq d_2, d_2 \sqsubseteq d_1\} \Rightarrow \{t \sqsubseteq d_1\}$

自明な制約の除去

t を任意の項とするとき、S に以下の制約が現れるなら、自明なので取り除く。

P11: $t \cong t$

P12: $t \sqsubseteq t$

4.3.3 正規化処理の合流性

任意の有限制約集合に対して、上の手順をどの順に適用しても、同じ正規系が得られることを示す。

多くの規則は、それまで制約集合中に存在した制約を取り除かないので、それらの規則だけが適用される場合の合流性は明らか。

例外は、P1 (変数割当の適用) と P11, P12 (自明な制約の除去) であるが、これらの規則と他の規則との間で適用順序が前後しても同じ結果が出ることは明らかであり、合流性がある[10]。

---

[7] 継承を扱う場合、この前に継承の一般規則の適用を行なう。

[8] ここでは簡単のためこう書いたが、e,f が無限項である場合はこの方法では停止しない。その場合、[2] にしたがって制約を解く必要がある。

[9] 前項に同じ。

[10] 証明は略す。

4.3.4 正規化処理の停止性

任意の有限制約集合に対して、上の手順が必ず停止することを示す。

S が有限なので、もともと制約集合中にあった項の種類は有限である。

項の種類が増えるような規則は、変数割り当てと、オブジェクト項の上限、下限の置換である。

変数項の数は有限であるので変数割り当ては有限回しか行なわれ得ない。

オブジェクト項の上限、下限の置換は、常にドット項または変数項に対する制約が対象となる。これらは有限であるから、少なくともすべてのドット項と変数項に対する制約について規則を適用してしまえば、それ以上項の種類は増えない。これは有限である。

項の種類が有限であるなら、上の手順に現れる規則は、いくつかの例外を除けば、すでに存在する項の間の制約を追加するものである。例外は変数の割り当て規則と自明な制約の除去規則であるが、これらは (上の議論から)、いずれも有限回しか適用されない。

以上から、少なくともすべての項の間に制約を生成しきった時点で停止するので、上の規則は停止性を持つ。

4.3.5 充足関係の解消処理

充足関係の解消処理とは、充足関係 $A\backslash B$ に対して以下の手続きを行なうことである。

1. A を正規形 A^* に、B を正規形 B^* に変換する。正規化に失敗するとその導出は失敗である。
2. A^* にゴール中に現れる変数が残っている間はここで解消処理を終了し、次の導出まで充足関係を保存する。
3. そのような変数がなくなったなら、A^* に現れる、少なくとも一方がドット項である制約を、下に述べる手続きで B^* の要素から証明できることを示す。

   すべてのそのような制約が証明できたなら、$A\backslash B$ は真である。A^* の要素をすべて結論に加え、充足関係を取り除き、解消処理は終了する。

   さもなければ偽である。その導出は失敗する。

証明は次の手続きで行なう。以下では d はドット項、x は変数項、t は任意の項である。

- $d \sqsubseteq t \in A^*$ であれば、$d \sqsubseteq t' \in B^*$ where $t' \sqsubseteq t$ であるなら $d \sqsubseteq t$ は証明される。$t' \sqsubseteq t$ は自明な包摂関係であっても良いし、B^* の要素であっても良い。
- 同様に $t \sqsubseteq d \in A^*$ であれば、$t' \sqsubseteq d \in B^*$ where $t \sqsubseteq t'$ であるなら $d \sqsubseteq t$ は証明される。
- $d \cong x \in A^*$ であって、x が変数項であるならば、それは直ちに証明される。
- $d \cong t \in A^*$ であれば、$d \cong t \in B^*$ であるなら $d \cong t$ は証明される。

この手続きで "証明される" と言っている内容が、実際に公理的意味論上で証明できることを示すのはやさしい。

4.3.6 終了処理

ゴールが空になり、結論の正規化を行なって変数制約を反映してもまだ変数の値が決まらずに処理ができない充足関係が存在する場合、以下の手続きを行なう。

充足関係 $A\backslash B$ について、A の要素で、少なくとも一方が変数項である制約は取り除いて結論に加え、それ以外の制約について、前項で述べた証明の手続きを行なう。

4.4 導出の例

包摂関係から証明可能な例

次のプログラムに対して問い合わせ ?- $o/[l \rightarrow b]$. を発したとする。

$$\begin{array}{ll} a \sqsubseteq b & \cdots \text{ (包摂関係の定義)} \\ o/[l \rightarrow a];; & \cdots \text{ (1)} \end{array}$$

ノード N_1 は $(\{o\}, \{o.l \sqsubseteq b\}, \emptyset, \emptyset, \emptyset)$ である。ゴールは一つだけなので g_1 としては o を選ぶ。前提も一つだけなので、Ks_1 は $\{o.l \sqsubseteq b\}$ とする。Re としてルールの集合 $\{(1)\}$ を選ぶと、

$$N_{1'} = (\emptyset, \emptyset, \{\{o.l \sqsubseteq b\} \backslash \{o.l \sqsubseteq a\}\}, \emptyset, \{o \cong o, o.l \sqsubseteq a\})$$

が得られる。

ここで制約が処理される。結論を正規化すると $o \cong o$ のような自明な制約は取り除かれ、$\{o.l \sqsubseteq a\}$ が残る。変数は現れないので続いて充足関係解消処理が行なわれる。A^* の唯一の要素 $o.l \sqsubseteq b$ に対して $o.l \sqsubseteq a \in B^*$ where $a \sqsubseteq b$ であるので証明できる。よって仮説は生成されない。$A^* = \{o.l \sqsubseteq b\}$ を結論に加えて正規化し、解 $(\emptyset, \{o.l \sqsubseteq a, o.l \sqsubseteq b\})$ が得られる。

属性値を求める例

次のプログラムに対して問い合わせ ?- $o/[l = X]$. を発したとする。

$$o/[l \rightarrow a];; \quad \cdots \quad (2)$$

問い合わせから $N_1 = (\{o\}, \{o.l \cong X\}, \emptyset, \emptyset, \emptyset)$ が生成される。

Re として $\{(2)\}$ を選択すると、

$$N_{1'} = (\emptyset, \emptyset, \{\{o.l \cong X\} \backslash \{o.l \sqsubseteq a\}\}, \emptyset, \{o.l \sqsubseteq a, o \cong o\})$$

が得られる。制約処理によって自明な制約 $o \cong o$ は除かれ、ゴールに現れない変数 X に関する制約 $o.l \cong X$ は

結論に入れられ、 $N_2 = (\emptyset, \emptyset, \emptyset, \emptyset, \{o.l \sqsubseteq a, o.l \cong X\})$ が得られる。 N_2 から、解 $(\emptyset, \{o.l \sqsubseteq a, o.l \cong X\})$ が得られる。

制約がマージされる例

次のプログラムに対して問い合わせ ?- $o/[l \to c]$. を発したとする。

$$c \sqsubseteq a, c \sqsubseteq b \quad \cdots \quad \text{(包摂関係の定義)}$$
$$o/[l \to a];; \quad \cdots \quad (3)$$
$$o/[l \to b];; \quad \cdots \quad (4)$$

問い合わせから $N_1 = (\{o\}, \{o.l \sqsubseteq c\}, \emptyset, \emptyset, \emptyset)$ が得られる。

Re として $\{(3), (4)\}$ を選択すると、

$$N_{1'} = (\emptyset, \emptyset, \{\{o.l \sqsubseteq c\} \backslash \{o.l \sqsubseteq a, o.l \sqsubseteq b\}\}, \emptyset, \{o.l \sqsubseteq a, o.l \sqsubseteq b, o \cong o\})$$

となるが、 $\{o.l \sqsubseteq a, o.l \sqsubseteq b\}$ は正規化によって $o.l \sqsubseteq c$ を生み出すので充足関係が解消され、 $N_2 = (\emptyset, \emptyset, \emptyset, \emptyset, \{o.l \sqsubseteq a, o.l \sqsubseteq b, o.l \sqsubseteq c\})$ となり、解 $(\emptyset, \{o.l \sqsubseteq a, o.l \sqsubseteq b, o.l \sqsubseteq c\})$ が得られる。

4.4.1 解の順序と極小性

次のプログラムに問い合わせ ?- $o/[l = X]$. を発したとする。

$$o/[l = a] \Leftarrow p;; \quad \cdots \quad (5)$$
$$o/[l \to a] \Leftarrow p/[l \to a];; \quad \cdots \quad (6)$$
$$p;;$$

直観的に言って、ルール (5) だけを使って求めた解はルール (6) だけを、あるいは両方を使って求めた解よりも望ましいように見える。これは解の順序で説明される。

解 $S_1 = (A_1, C_1), S_2 = (A_2, C_2)$ の間に $S_1 \models S_2$ の関係があるとは、 A_2 から A_1 が証明可能であり、かつ、 C_1 から C_2 が証明可能であることである。この関係 $\models$ が相互に成り立つ解を同じ解であるとみなし、 $\models$ をそのような同値類上の関係と定義し直すことによって $\models$ は半順序となる。

QUIXOTE の推論は、仮説を扱うことが可能なので、生成される解は一般に数が多い。 $S_1 \models S_2$ を、 S_1 が S_2 よりも小さいと読むことにすれば、 *QUIXOTE* の実装は、上で述べた半順序にしたがって極小値を求めるように構築することが望ましい。

4.5 導出の健全性

本節では、本稿で述べた手続き的意味論が公理的意味論で証明される解だけを出力することの証明の戦略について簡単に述べる。

ノード (G, K, S, A, C) を一つの sequent $\rightarrow G \wedge K \wedge S \wedge A \supset C$ と考え (集合は連言に直す)、これに対する手続きをすべて公理的意味論上の対応する証明操作に変換できることを示せば良い。

5 実装との関係

本節では、本稿の検討を通じて明らかになった、現在の実装[11]の仕様の定式化しにくい点や、意味論通りの単純な実装では効率が上がらない部分の実装上の工夫などをとりあげる。

5.1 ドット項の扱い

現在の実装のドット項の扱いは、必ずしも本稿で提示された意味論とは一致しない。

本稿の意味論では、 $o/[l = X];;$ というファクトを記述すると、 $o.l \cong a$ でも $o.l \cong b$ でも充足してしまい、一般に矛盾を生ずる。

現在の実装では、ファクト $o/[l = X];;$ の X は必ずしも全称束縛ではなく、直観的には "ある $o.l$ があって、その値は X である" というような意味を持つ。同様の現象は $o/[l \to X];;$ というファクトを記述した際にも現れる。

これは属性値がユニークなある値であることを強い意味で保証しようとした設計思想の現れであるが、いくつかの応用と、本稿で示した意味論の設計を通じ、再検討の必要性を感じている。

5.2 制約のマージ

一回の導出に複数のルールを用いるので、ルールの選択のバリエーションは、単純計算では Prolog を N とすれば 2^N に広がっている。しかし、一本だけを選択しても導出が失敗するルールは、他のルールと組み合わせてもやはり失敗するので、 2^N をすべて試さずとも、成功したルールどうしの組合せだけを試せば良い。それでも、その組合せを実際に再計算したのではやはり時間がかかるので、現在の実装では、一本のルールを用いた場合の結果から、それらを組合せた解を計算する方法を用いている。このとき、同時になるべく極小解をだけを残す処理を行ない、同じような計算を何度もすることを避けている。

5.3 再帰的ルールの扱い

同じオブジェクト項の付帯属性間の関係を記述するルールは自然に再帰的になるため、そのようなスタイルのプログラミングでは再帰的ルールが多くなる。特に、 *QUIXOTE* の実装は、次のようなプログラムに対する問い合わせ?-$o/[l_1 = X]$. を停止させることが期待されている。

$$o/[l_1 = a] \Leftarrow o/[l_2 = a];;$$
$$o/[l_2 = a];;$$

現在の実装は、最左ゴール優先かつ深さ優先の探索を基本とするが、必要に応じて OLDT [3] 風の表検索を用いる。即ち、それまでに選択したことのあるゴールと同じゴール

[11] *QUIXOTE* 第三版。

を選択しようとするとき、既に得られている解をコピーして用いる。ただし、導出木全体にわたってこれを行なうことは、常に表を参照しながら導出を行なうことになり、深さ優先の利点を損ねてしまう。現在の実装では、深さ方向で再帰的に展開されているゴールだけについて表検索を行なう。

6 今後の課題

6.1 実装の再検討

本稿で与えた意味論をもとに、記述実験などを通じて、現在の実装の評価と再検討を進めることは有益であろう。

6.2 解の極小性

解の極小性を定義したが、本稿で与えた手続き的意味論で求められる解よりも小さな解が存在する場合があることが分かっている[12]。しかしその種の解は、単一の充足関係から複数の仮説の可能性を導いてしまう[13]ので、現在の意味論を拡張しないと扱うことができない。極小性の定義の妥当性と併せて今後の検討を要する。

6.3 意味論の発展

手続き的意味論の公理的意味論に対する健全性は保証することができたが、完全性はまだである。また、手続き的意味論で必ず極小解が得られるという保証を与えることも有益である。公理的意味論と合わせて今後検討するに値する。

7 まとめ

Quixote に、LJ を基本とした公理的意味論と、CLP に近い手続き的意味論を与えた。検討を通じて、現在の実装上の特徴や問題点に対する意味づけができた部分があった。

Quixote は、論理型言語の立場から考えると、頭部の制約がマージされることなどから、CLP スキーマの拡張となっており、従来の定式化で意味論を与えることはできない。本稿では、一回の導出に複数のルールを用いるなどの拡張を行なって定式化した。

論理型言語にオブジェクト指向概念を組み込むためには、この拡張を考える必要があり、今後さらに理論的整備を行なう予定である。

[12] 例えば、$c \sqsubseteq a$, $c \sqsubseteq b$ という包摂関係があるときに、$o/[l \rightarrow b];;$ というプログラムに対して問い合わせ $?\text{-}o/[l \rightarrow c]$. を発すると、本稿の手続き的意味論で得られる最小の解は $(\{o.l \sqsubseteq c\}, \{o.l \sqsubseteq c\})$ であるが、公理的意味論ではより小さい $(\{o.l \sqsubseteq a\}, \{o.l \sqsubseteq c\})$ を証明できる。

[13] 先の脚注の例では、さらに $c \sqsubseteq d$ であった場合など。

謝辞

意味論の検討にあたって貴重な意見とコメントを頂いた *Quixote* グループのメンバーに謝意を表する。

参考文献

[1] H. Aït-Kaci. "An Algebraic Semantics Approach to the Effective Resolution of Type Equations". In *Theoritical Computer Science*, number 45, 1986.

[2] K. Mukai. "Coinductive Semantics of Horn Clauses with Compact Constraint". In G. P. Jon Barwise, Jean Mark Gawron, and S. Tutiya, editors, *Situation Theory and Its Applications*, volume 2. Stanford University, 1991.

[3] H. Tamaki and T. Sato. "OLD Resolution with Tabulation". In *Proc. 3rd Int. Conf. of Logic Programming*, 1984.

[4] H. Tanaka. "Integrated System for Protein Information Processing". In *Proc. Int. Conf. on Fifth Generation Computer Systems*, 1992.

[5] S. Tojo and H. Yasukawa. "Situated Inference of Temporal Information". In *Proc. of Fifth Generation Computer System*, 1992.

[6] M. Tsukamoto, S. Nishio, and M. Fujio. "DOT: A Term Representation Using DOT Algebra for Knowledge Representation". In *Proc. 2nd Int. on Deductive and Object-Oriented Databases (DOOD '91)*. Springer, 1991.

[7] H. Yasukawa, H. Tsuda, and K. Yokota. "Objects, Properties, and Modules in Quixote". In *Proc. of FGCS92*, 1992.

[8] K. Yokota and H. Yasukawa. "Towards an Integrated Knowledge Base Management System". In *Proc. Int. Conf. on Fifth Generation Computer Systems*, 1992.

[9] 淵田宏, 横田一正. "自然言語の制約ベース文法理論への DOOD 風アプローチ". 第 94 回 データベースシステム研究会, 1993.

[10] 柴崎真人, 山本展一郎, 近藤秀文, 横田一正. "法的推論システム TRIAL の開発". 情報処理学会第 46 回 (平成 5 年前期) 全国大会, 1993.

[11] g 新部裕. "*Quixote* (V3) 構文". a document file in Quixote (ICOT Free Software), 1993.

[12] 西岡利博, ほか. "*Quixote* の意味論". (準備中).