

TR-0890

言語情報処理の枠組としてのQuixote

東条 敏 (MRI)、津田 宏、
安川 秀樹 (松下)、横田 一正、
森田 幸伯 (沖)

September, 1994

© Copyright 1994-8-20 ICOT, JAPAN ALL RIGHTS RESERVED

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

言語情報処理の枠組としての *QUIXOTE*

東条 敏, 津田 宏

安川 秀樹, 横田 一正, 森田 幸伯

(財) 新世代コンピュータ技術開発機構

〒108 東京都港区三田 1-4-28 三田国際ビル 21 階

(phone) +81-3-3456-3069 (e-mail) tojo@mri.co.jp, tsuda@icot.or.jp

概要

われわれは自然言語処理における複雑な情報の流れを扱うツールとして、言語 *QUIXOTE* を開発した。*QUIXOTE* は演繹オブジェクト指向データベースと制約論理プログラミング言語を融合した言語である。*QUIXOTE* の特徴は、オブジェクト同一性のようなオブジェクト指向の考え方と、大規模知識ベースを分節できるモジュールという考え方を組み合わせたことにある。しかも、この言語の演繹システムは、制限付きのアブダクションができるよう拡張されている。われわれはまず最初に、この言語を制約ベースの文法の型付き素性構造の記述に適用する。次に、状況に依存した意味記述に *QUIXOTE* を適用し、まず、アブダクションを用いて背後知識を明らかにする例、次にモジュールを状況と見立てて、名詞の参照に複数の可能性を捜し出す例を示す。

1 はじめに

自然言語処理の過程は、非常に複雑な情報の流れを含んでいるため、単純な手続き的言語で記述するのは一般に困難である。このような複雑な処理過程をできるだけ自然に表現できるような枠組があることが望ましい。具体的には、近年の自然言語処理では (1) 制約ベースの文法を記述でき、かつ (2) 意味の状況依存性を明示できる 枠組みが求められている。

このために、最初われわれは制約論理型言語 (Constraint Logic Programming Language; CLP) *cu-Prolog* [Tsuda 94] を開発し、制約ベースの文法の記述においてその有効性を検証した。しかしながら、*cu-Prolog* は宣言的には Horn 節論理と同等であるため、文脈に依存する情報の記述には向いていなかった。この時点で *cu-Prolog* に求められた要件は、制約ベースの論理推論機構に加えて、

- ソート付き素性構造 (sorted feature structure) を処理できる機構を持つこと、
- 状況に依存した情報の表現ができること

の二点であった。

本稿では、このような要件を実現するために開発された言語 *QUIXOTE* [Yasukawa 92, Yokota 92] を紹介する。*QUIXOTE* は、CLP 言語の特徴をもった演繹オブジェクト指向データベース (Deductive Object-Oriented Database; DOOD) として設計された¹。*QUIXOTE* の自然言語処理への有効性は次の二つの側面から述べることができる。一つは、*QUIXOTE* は様々な言語現象や意味論で述べられている内容を統一的に表現できるという静的な側面。もう一つは、*QUIXOTE* の有しているアブダクションを含んだ推論メカニズムは、言語現象の処理にも有効であるという動的な側面である。

本稿の構成は以下のようになっている。まず第2節では、*QUIXOTE* の基本的な仕様について述べる。続く第3、4節では、*QUIXOTE* が上述の二点の要件、すなわちソート付き素性構造と状況意味論についてどのような有効性を持つかを例証する。まず第3節では HPSG (Head-driven Phrase Structure Grammar)[Pollard 94] のソート付き素性構造の記述を試みる。続く第4節では、*QUIXOTE* がどのように状況に依存した意味論を記述できるかについて述べ、モジュールを状況と見立て、状況に依存した表現と推論の例を示す。最後の第5節では、関連研究とそれらに対する *QUIXOTE* の特徴をまとめ、将来の研究課題について述べる。

2 QUIXOTE の概要

本節では、自然言語情報を表現し実現する枠組という観点から、*QUIXOTE* の主要な機能を紹介する。*QUIXOTE* は DOOD 言語または包摂関係に基づく記号的な制約を扱う CLP 言語と分類され、以下のような特徴を持つ。

- オブジェクト同一性 (object identity) のようなオブジェクト指向概念
- 属性と値の対による複雑な情報、制約の表現
- モジュール及び継承機構による、情報の効率的な記述
- 仮説生成 (アブダクション)、仮説推論

一般に *QUIXOTE* のプログラムは次の3つの部分からなる。

- 基礎オブジェクト項間の包摂関係の定義
- サブモジュール関係の定義
- ルール定義

包摂関係については2.1節で、サブモジュール関係は2.3節にて説明する。

QUIXOTE のルールは、以下の形の節で構成される。

$$\underbrace{m_0 :: H}_{\text{頭部}} \mid \underbrace{HC}_{\text{頭部制約}} \Leftarrow \underbrace{m_1 : B_1, \dots, m_n : B_n}_{\text{本体}} \parallel \underbrace{BC}_{\text{本体制約}} ; ; \quad (1)$$

¹ 実際この言語は自然言語の時間的意味曖昧性の記述 [Tojo 92] など自然言語分野での応用例に加え、データベースとしても法的推論や分子生物学などに適用例がある。

ここで H および B_i はオブジェクト項 もしくはそれに属性指定を加えた属性項, HC および BC は制約 (包摂関係に基づく制約式の集合) である. m_i はモジュールを表している. なお, 本体, 制約, モジュールは省略可能である.

上のルールは直観的には「もし 各 B_i がモジュール m_i において制約 BC のもとで成り立つならば, モジュール m_0 において H および制約 HC が成り立つ」という m_0 における知識を意味する².

2.1 オブジェクト項

オブジェクト項 (object term) には, 基礎オブジェクト項 (basic object term) および複合オブジェクト項 (complex object term), 変数 (Variable) がある. $Obj, Bobj, Cobj, Var$ をそれぞれ, オブジェクト項, 基礎オブジェクト項, 複合オブジェクト項, 変数の集合とする. $Obj = Bobj \cup Cobj \cup Var$ である.

2.1.1 基礎オブジェクト項

$Bobj$ は原始記号の有限集合で, その要素を基礎オブジェクト項と呼ぶ.

包摂関係 (subsumption) $\sqsubseteq$ は, 基礎オブジェクト項の間の半順序であり, ユーザが *QUINOTE* のプログラムにおいて定義する. $a \sqsubseteq b (a, b \in Bobj)$ は, 「 a は b の下位概念である」または「 a は b である (ISA 関係)」ことを意味する. 特別な基礎オブジェクト項として, 最小元 $\perp$ と最大元 $\top$ があり, $\forall o \in Bobj, \perp \sqsubseteq o, o \sqsubseteq \top$ により定義される. なお, $\langle Bobj, \sqsubseteq \rangle$ から完備束を構成することができる.

以下は, 基礎オブジェクト項とそれらの包摂関係の例である.

`cat  $\sqsubseteq$  animal, animal  $\sqsubseteq$  creature`

2.1.2 複合オブジェクト項

複合オブジェクト項は, $o[l_1 = v_1, l_2 = v_2, \dots]$ という形の項で表現される. ここで $o \in Bobj, \forall i, l_i \in Bobj, v_i \in Obj$, ただし $l_i \neq l_j (i \neq j)$ である. 各ラベル l_i は固有属性 (intrinsic attribute) と呼ばれる. 複合オブジェクト項は, データベースの複合オブジェクトの概念と関連がある.

複合オブジェクト項に関する包摂関係は, 以下のように基礎オブジェクト項の包摂関係から定義される.

$$\begin{aligned} o[l_1 = u_1, \dots] \sqsubseteq p[m_1 = v_1, \dots] &\iff o \sqsubseteq p, \forall i \exists j, l_j = m_i, u_j \sqsubseteq v_i \\ o[l_1 = u_1, \dots] \sqsubseteq p &\iff o \sqsubseteq p \end{aligned}$$

例えば, `cat  $\sqsubseteq$  animal, human  $\sqsubseteq$  animal` の場合に,

`cat[color=white]  $\sqsubseteq$  animal`
`human[age=20, sex=male]  $\sqsubseteq$  animal[age=20]`

のような関係が成り立つ.

複合オブジェクト項同士のミートやジョインも, 包摂関係上の最大下界 (greatest lower bound) もしくは最小上界 (least upper bound) として定義される.

2.2 属性項と属性継承

属性項 (attribute term) は, オブジェクト項に属性の指定を加えた項の記述方法であり, 次のような形式である.

$$o/[l_1 op_1 v_1, l_2 op_2 v_2, \dots]$$

ここで $o \in Obj, \forall i, l_i \in (Bobj \cup Cobj), op_i \in \{=, \leftarrow, \rightarrow\}, v_i$ は オブジェクト項もしくは, 変数を含まないオブジェクト項を要素とする集合である. ラベル l_i を, 特に 付帯属性 (extrinsic attribute) と呼ぶ.

属性項の記法は一種のマクロであり, 以下の規則によりオブジェクト項とその付帯属性についての制約とに変換される.

$$\begin{aligned} o/[l = X] &\iff o \mid \{o.l \cong X\} \\ o/[m \rightarrow U] &\iff o \mid \{o.m \sqsubseteq U\} \\ o/[n \leftarrow V] &\iff o \mid \{V \sqsubseteq o.n\} \end{aligned}$$

²モジュールの間の関係には, この例のような外部モジュール参照の他に, サブモジュール関係 (2.3参照) もある.

$o \mid C$ において、 o はオブジェクト項、 C はその制約を表す。 $o.l$ の形の項は、ドット項 (*dotted term*) と呼ばれ、オブジェクト項 o の属性 l の値を示す。制約は、オブジェクト項もしくはドット項を包摂関係で結んだ制約式の集合である。各制約式は、 $\langle term \rangle \langle op \rangle \langle term \rangle$ という形をしている。ここで $\langle term \rangle$ はオブジェクト項またはドット項であり、 $\langle op \rangle \in \{\equiv, \sqsubseteq, \supseteq, \&in\}$ である。 $\&in$ はオブジェクト項とオブジェクト項の集合の間に成り立つメンバシップ関係である³。

オブジェクト項の付帯属性の値は、そのオブジェクト項と包摂関係にあるオブジェクト項に以下のような一般規則に従って継承される。これを属性継承と呼ぶ。

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

例えば、`apple  $\sqsubseteq$  macintosh` かつ `apple.color  $\equiv$  red` であれば、`red  $\sqsubseteq$  macintosh.color` というように `color` 属性が `macintosh` に継承される。

複合オブジェクト項の場合、固有属性の値は、継承される付帯属性の値に優先されるため、上の規則はあてはまらない。固有属性を用いると属性継承の例外を記述することができる。

例えば、`apple[color = green]  $\sqsubseteq$  apple` は常に成り立つ。しかし、`apple.color  $\equiv$  red` であっても、やはり `apple[color=green].color  $\equiv$  green` となり `color` 属性については固有属性値が優先される。

2.3 モジュールとオブジェクト同一性

2.3.1 サブモジュール関係とルール継承

QUIXOTE のプログラムはモジュール (*module*) と呼ばれる単位で分割して記述することができる。各モジュールは、ルールの集合からなり、モジュール識別子 (*module identifier*) と呼ばれるオブジェクト項により識別される。多くのプログラム言語と同様に、上位のモジュールのルールを、下位のモジュールに継承させることで、効率良く知識を記述することができる。この継承をルール継承と呼ぶ。

サブモジュール関係 $\sqsubseteq_S$ は、モジュール識別子の間の半順序であり、ルール継承の方向を規定する⁴。例えば `john  $\sqsubseteq_S$  common.knowledge` が成り立っている場合、モジュール `common.knowledge` 内の全てのルールは、原則としてモジュール `john` に継承される。この時 `john` は `common.knowledge` のサブモジュールまた `common.knowledge` は `john` のスーパーモジュールという。

ルール継承においても例外が記述できる。各ルールには、継承モードとして `o`、`l`、`ol` のいずれかを付加することができる。`o` (*override*) のついたルールは、スーパーモジュールから継承してきたルールのうち、同一の頭部オブジェクト項を持つものを上書きする。`l` (*local*) のついたルールは、サブモジュールに継承されない。モード `ol` は `o` と `l` の組合せである。

以下のプログラム例は「ヨーロッパでは一般に車は右側通行である。ただし、イギリスでは左側通行である。」という知識を、ルール継承および継承の例外を使って記述したものである。`england` モジュールのルールには `o` の指定があるため、`europe` の `car` を頭部とするルールは継承されない。

```
england  $\sqsubseteq_S$  europe, france  $\sqsubseteq_S$  europe
europe :: car/[drive = right];;
england :: (o)car/[drive = left];;
```

2.3.2 オブジェクト同一性

複数のオブジェクト項が同一であるとは、ラベルの順序を除いて構文的に等しいオブジェクト項にバインドしていることである。例えば X が a にバインドしている場合、 $o[l=X, m=b]$ と $o[m=b, l=a]$ は同一である。

一つのモジュール内では、同一のオブジェクト項の同一属性の値は等しくなければならない。しかし、サブモジュール関係にない異なるモジュールでは、同一のオブジェクト項は異なった値を持つことができる。

例えば、次のプログラムでは、モジュール `sit_1` の2本のルールにおいて、同一のオブジェクト `john` の `age` 属性値が20と30で異なっているため、矛盾である。

```
sit_1 :: john/[age=20];;
sit_1 :: john/[age=30];;
```

³ さらに表現力を高めるため、否定制約を含めることを検討中である。非等式 $\neq$ による制約は論理的には問題はない。ただし、複合項の間の $\neq$ 制約により、制約の選言が生じるため計算量が増大する恐れがある。

⁴ 現在の枠組では、モジュール識別子のオブジェクト項としての包摂関係と、サブモジュール関係とは別ものだと仮定している。

しかし, `sit_1` と `sit_2` とがサブモジュール関係にないのであれば, 次のプログラムは矛盾ではない.

```
sit_1 :: john/[age=20];;
sit_2 :: john/[age=30];;
```

2.4 仮説付きの答

QUIXOTE に対する質問文は,

$$? - m_1 : G_1, \dots, m_n : G_n \parallel C. \quad (2)$$

という形で与えられる. m_i はモジュール識別子, G_i はオブジェクト項 / 属性項, C は制約である. 質問に対する答として, *QUIXOTE* は, 質問中の変数の代入値や制約と, 仮説 (assumption) と呼ばれるドット項間の制約を返す. とは, 導出過程で充足されなかった制約の集合であり, プログラム自体に欠けていた情報と見なすことができる. 仮説をつけて解を返すのは, 一種のアブダクションと見ることができ, 事例ベース推論 (case-based reasoning) や, 状況推論 (4 節) 等の応用で重要な役割を果たす.

QUIXOTE ルールの頭部制約は, 従来の CLP 言語 [Jaffer 87] と同様の動きをする. ルールの本体制約のうち, 頭部制約によって充足されないものは, 仮説として蓄積されていく. (詳細は [西岡 93] を参照のこと) この点から, *QUIXOTE* は部分情報データベースを扱うアブダクティブなプログラミング言語と見られることもできる.

以下のプログラム例を見てみよう. これは, ある本が存在して, それがハードカバーであれば輸送費は 500 円, ソフトカバーであれば 300 円ということを意味する.

```
sit::book;;
sit::shipping[fee=500] <= book/[cover->hard];;
sit::shipping[fee=300] <= book/[cover->soft];;
```

最初のルールは, オブジェクト `book` の存在だけを示し, その属性については何も言っていない. 次のルールは, もし `book` が `sit` に存在し, その `cover` 属性値が `hard` に包摂されるならば, `sit` にて `shipping[fee=500]` が成り立つことを意味している. 最後のルールも同様な知識を表している. さて, このプログラムに, 本の輸送費を尋ねる次の質問を与えてみよう.

```
?-sit:shipping[fee=Yen].
```

という質問に対しては, *QUIXOTE* は次のように二つの独立な解を返す.

```
IF sit:book.cover=<hard THEN Yen=500
IF sit:book.cover=<soft THEN Yen=300
```

それぞれの解は, 各ルールの本体から出た `book` の `cover` 属性値に対する制約⁵を, 仮説として持っている. いずれの仮説も, 導出時に頭部制約 (この場合は空) では充足されなかったものである.

3 *QUIXOTE* による素性構造の記述

本節では, 自然言語文法理論で使われる, (ソート付き) 素性構造の *QUIXOTE* による記述を示す.

3.1 素性構造

素性構造 (feature structure) は, 単一化ベースまたは制約ベースの自然言語文法理論 [Shieber 86] で, 自然言語の部分的な情報を扱うためにしばしば使用されるデータ構造である. 素性構造は, 素性から値への部分関数であり, 素性と値の対の集合である AVM (attribute-value matrix) 記法により表現されることが多い. 次の例 (3) は, 素性 `number` から `singular` へ, `person` から `third` への関数を表している [Shieber 86].

$$\left[\begin{array}{l} \text{number : singular} \\ \text{person : third} \end{array} \right] \quad (3)$$

⁵*QUIXOTE* 処理系の構文では $\sqsubseteq$ は $=<$ で表現される.

素性構造は、素性をリンク、値をノードとする、DAG(directed acyclic graph) 構造としても定式化される [Shieber 86]。

CIL 言語 [Mukai 85] の基本的なデータ構造である PST(partially specified term) は、情報の部分的表現を行なうもので、素性構造と関係が深い。例えば (3) は、{number/singular, person/third} という PST により表現される。

さて、情報の部分表現を表現するデータ構造として、QUIXOTE には属性項がある。例えば (3) は、属性項

f1/[number=singular, person=third]

により表現される。ただし、QUIXOTE ではさらに制約も記述することができるため、PST に比べて記述力はまさっている。素性構造の単一化は属性項の等号制約解消と、素性構造の包摂関係は属性項の包摂制約の解消とそれぞれ等価である。QUIXOTE の包摂制約は、[Dörre 91] で述べている素性制約のうち weak subsumption に基づく制約なので、多項式時間で決定可能である。

実際の自然言語処理への応用を考えると、処理で必要となる全ての素性名があらかじめ決まっていることがしばしばある。それに対しては、アリティの決まったデータ構造を用いて処理速度の向上を計ることが考えられる。CIL における TST(totally specified term) と呼ばれるデータ構造は、PST と同じ構文で表されるが、実行時には Prolog の項に変換され高速な単一化が可能になる。QUIXOTE における複合オブジェクト項は、ラベルつきグラフ構造 (labeled graph structure) として定式化されており [安川 90]、固定アリティの素性構造と自然に対応づけできる。例えば (3) は、fs[number=singular, person=third] という複合オブジェクト項により表すことができる。この場合、素性構造の包摂関係は QUIXOTE の複合オブジェクト項の包摂関係 (2.1.2 節) に対応し、⁶素性構造の単一化は複合オブジェクト項のミート操作に対応する。

3.2 ソート付き素性構造

次に、素性構造の拡張であるソート付き素性構造 (sorted feature structure [Pollard 94])⁷ を考えよう。ソート付き素性構造は、各ノードがソート記号によりラベルづけされている素性構造で、HPSG(Head-driven Phrase Structure Grammar) の最新の定式化 [Pollard 94] で用いられている。ソート付き素性構造の利点の一つは、ソートに包摂関係があり、下位のソートが上位のソートの情報を継承することで、効率良く自然言語の記述ができることである。

QUIXOTE では基礎オブジェクト項をソート記号とみなすことで、ソート付き素性構造を表現することができる。QUIXOTE における基礎オブジェクト項と、その包摂関係 $\langle \text{Bobj}, \sqsubseteq \rangle$ により、ソートおよびそれらの継承階層が実現される。ソート付き素性構造自体は、素性構造と同様に QUIXOTE の属性項で表現できる。QUIXOTE における属性継承は、まさにソート付き素性構造間の情報の継承に対応する。

最初の例として、ソート付き素性構造 (4) を取り上げる。これは、HPSG 風に単語 “run” を記述したものである。(4) のタイプは word であり、CAT は文法カテゴリー、PH は音素、SUBCAT はこの動詞が取る補語 (の集合) を表す。

$$\left[\begin{array}{l} \text{word} \\ \text{CAT} : [\text{vp}] \\ \text{PH} : [\text{run}] \\ \text{SUBCAT} : \left[\begin{array}{l} \text{phrase} \\ \text{CAT} : [\text{np}] \end{array} \right] \end{array} \right] \quad (4)$$

一番外側の素性構造を w1, SUBCAT の引数の素性構造を p1 により表すと、(4) は次のように包摂関係の定義と、属性項により表現できる。

```
w1=<word, p1=<phrase
w1/[cat->vp, ph->run, subcat=p1]
p1/[cat=np]
```

ここで、例えば word に他の属性が定義されている場合には、オブジェクト間の属性継承機構により、その属性は自然に w1 に継承される。

簡単な例として、[Carpenter 92] で紹介されているソート付き素性構造を利用して解くパズルを取り上げる。

⁶[Shieber 86] とは関係の向きは逆である

⁷[Carpenter 92] では、型付き素性構造 (typed feature structure) とも呼ばれる。

Smith, John, Tucker の3人の車は、メーカーおよびモデルが互いに異なっている。メーカーは Buick, Ford, Toyota のいずれかであり、モデルはスポーツ車、クーペ、ワゴンのいずれかである。Tucker の車は Buick または Ford 製のワゴンかスポーツ車、Smith の車は Ford か Toyota 製のスポーツ車、John の車は Buick 製のクーペかワゴンである。

第2文は以下のような、*QUIXOTE* の基礎オブジェクト項間の包摂関係の記述に相当する。

```
car_maker >= {ford, buick, toyota}
style >= {sport, wagon, coupe}
```

第3文は、以下のように *QUIXOTE* のルールにより表現される。各人の車は `car[owner=smith]` のような複合オブジェクト項で表しているが、`car[owner=smith] =< car` という自然な包摂関係があるために、`car` の属性が各人の車の属性に継承される。

```
%% 車というタイプの定義
car/[maker->car_maker, type->style];;
%% 複合項により各人の車を定義
car[owner=smith]/[maker &in {ford,toyota},type=sport];;
car[owner=jones]/[maker=buick, type &in {coupe,wagon}];;
car[owner=tucker]/[maker &in {buick,ford}, type &in {wagon,sport}];;
```

第1文は質問の形で記述してみる。その一部として、Tucker の車が John の車 (buick) とメーカーが異なるという制約を付け加えてみよう。ただし、現在の制約の枠組では非等式を扱えないので、`car[owner=tucker].maker ≠ buick` という制約を、`car[owner=tucker].maker &in {ford,toyota}` により表現する。

```
?- car[owner=tucker]/[maker=X] | {X &in {ford,toyota}}.
```

QUIXOTE は上の質問に対し `X=ford` と答えることになる。これは Tucker の車の make 属性の値は ford であることを示す。

4 *QUIXOTE* による状況推論

4.1 状況依存性の表現

状況推論という用語はさまざまな意味に解釈できるが、その中でも、いかに環境や状態の影響が推論規則に及ぶかという点を考察するのが Barwise の定めた状況推論の定義である ([Barwise 89], pp.157-160)。自然言語の多くの問題はこの状況依存性に帰着できる。この場合、状況ということばは、文脈、スコープ、時空間、知識、信念などを表すメタ・レベルの用語である。

自然言語の意味の状況依存性を形式化する試みはこれまでにいくつかあり、可能世界という考え方 [Dowty 81] 以降、状況 [Barwise 83]、メンタル・スペース [Fauconnier 84]、DRT (Discourse Representation Theory) [Kamp 81] などが提案されている。もちろんこれらの目的は異なり、背後の哲学も違うものであるが、それらの間に共通性を指摘することも容易である。すなわち、これらの理論に共通する、状況依存性およびそこで参照されるオブジェクトという側面のみを単純にとらえると、これら諸理論の概念は *QUIXOTE* によって次のように実現可能である。まず、メンタル・スペース理論においては、次のような対応関係が考えられる。

メンタル・スペース理論		<i>QUIXOTE</i>
メンタル・スペース	⇒	モジュール
トリガおよびターゲット	⇒	オブジェクト
コネクタ	⇒	モジュール間の制約

同様に、DRT については、以下の対応が考えられる。

DRT 理論		<i>QUIXOTE</i>
DRS	⇒	モジュール
レファランス・マーカの対象	⇒	オブジェクト
DRS 間の従属関係	⇒	モジュール間の継承または制約関係

ここで、DRT との対応関係について、もう少し詳しく、例を挙げる。意味論的な問題として有名なロバ文 (donkey sentence) ([Geach 62], 以降, [Cooper 79, Evans 77, Evans 88, Heim 90, Kamp 81] 他多数) を例として、*QUIXOTE* と DRT の対応を考えてみる。ロバ文に関する問題とは、以下に示す文に見られる、照応と限量化に関するパズルである。

Every farmer who owns a donkey beats it. (5)

ここで問題とされるのは、従来の意味論的分析においては、

- 代名詞 'it' は名詞句 'a donkey' の指示対象を指すというのが、自然な読みであるにもかかわらず、'it' が 'a donkey' のスコープの外にあるため、'it' を 'a donkey' に束縛することができないこと、
- 'a donkey' は普遍限量化されているというのが、自然な読みであるにもかかわらず、存在限量化されていると解釈されること、

の 2 点である。DRT においては、この問題点を、

- (1) 名詞句 NP の談話表示 (DRS) を K_{NP} 、動詞句 VP の DRS を K_{VP} とする時、文 'every NP VP' に対して

$$K_{NP} \Rightarrow K_{VP}$$

なる DRS を割り当て、 K_{NP} を満たす全ての割り当てが K_{VP} をも満たす時、その時のみ、真という解釈規則を与えること、

- (2) K_{VP} が K_{NP} に従属する (K_{VP} からは K_{NP} 中の指示標識を参照できる) ものとするこ、

により解決している。(1) は、'a' を普遍限量化として解釈するためのものであり、結果として、'every' に対して、'if then' と同じ解釈規則を適用していることになる。(2) は、限量化のスコープに関わるものである。先に示したように、DRT と *QUIXOTE* のエンティティを対応させ、ロバ文に対して *QUIXOTE* の記述を与えることにする。その際、(2) を *QUIXOTE* で表すために、サブモジュール関係を利用することにより、簡潔な記述を与えることができる。

NP の談話表示を表すモジュールを knp、VP の談話表示を表すモジュールを kvp、文全体の談話表示を表すモジュールを ks とする。それぞれの談話表示に設定されるレファランス・マーカを保持するモジュールとして、 $rm[dr=knp]$ 、 $rm[dr=kvp]$ 、 $rm[dr=ks]$ を導入する。また、指示対象の同一性を表すため、レファランス・マーカを表すオブジェクトに対して refers という属性を導入すると以下ようになる。

$x.refers = y.refers, x.refers = 'Pedro', etc.$

以下に示すのが、*QUIXOTE* による、ロバ文 (5) に対する記述である。

```
kvp >- knp ;;
kvp >- rm[dr=kvp];; knp >- rm[dr=knp];;
ks::if_then[ante=knp, cons=kvp] ;;
knp::farmer[obj=x] ;;
knp::donkey[obj=y] ;;
knp::owns[agt=x,obj=y] ;;
kvp::beats[agt=x,obj=z] ;;
rm[dr=knp]::x ;;
rm[dr=knp]::y ;;
rm[dr=kvp]::z/[refers=X] <= x || {x.refers=X};;
```

続いて、状況理論と *QUIXOTE* の対応関係について示すことにする。本稿において主に扱う状況理論の概念は、次の表のように置換することができる。

状況理論	<i>QUIXOTE</i>
状況	$\Rightarrow$ モジュール
インフオン	$\Rightarrow$ 複合オブジェクト項
ロール	$\Rightarrow$ ラベル
支持関係 ($\models$)	$\Rightarrow$ モジュール-ルールの包含関係 ($:$)

状況理論の優位性は、自然言語のさまざまな現象を似たような格好で表現できることである。すなわち、ある文 σ に相当する表現の意味が状況 s に依存する場合、状況理論ではこれを

$$s \models \sigma$$

と書く。したがって、状況推論の規則とは以下のような形をしたものであるとする。

$$s_0 \models \sigma_0 \Leftarrow s_1 \models \sigma_1, s_2 \models \sigma_2, s_3 \models \sigma_3, \dots, s_n \models \sigma_n \parallel C. \quad (6)$$

この規則は次のように解釈される。もし s_1 が σ_1 を、 s_2 が σ_2 を、以下同様に s_n が σ_n を支持するとき、 C という制約のもとで、 s_0 が σ_0 を支持すると推論できる。このような推論規則は Prolog の規則のように後向き連鎖を行なうものと考えられる。ここで最も重要なのは、*QUIXOTE* の規則そのものがそのまま状況推論の規則となるようになっていることである。すなわち (6) と (7) の類似性が重要である。

$$m_0 :: \sigma_0 \Leftarrow m_1 : \sigma_1, m_2 : \sigma_2, \dots, m_n : \sigma_n \parallel C;; \quad (7)$$

以上のように、いくつかの状況依存性に関わる理論は *QUIXOTE* により統一的に表現でき、かつまた状況に依存した形式を明示したまま推論を進めることができる。以下に、実際の自然言語文に対してこのことを適用した例を示す。

4.2 背後知識の推論

自然言語処理の目的の一つは、表層から隠された情報を明らかにすることである。Barwise は、この要件を状況推論のひとつの目的であるとした [Barwise 89] (p. 159)。本稿では、状況理論における推論システムを定式化し、ある条件下において背後にある情報を推論する例を紹介する。

まず、条件付き推論の例文として、以下の A、B 二人の人間による発話を考察する ([Quine 59] (p. 15) および [Barwise 89] (p. 105))⁸。

A: "If Bizet and Verdi are compatriots, Bizet is Italian." (8)

B: "If Bizet and Verdi are compatriots, Verdi is French." (9)

この例文 (8), (9) のおもしろいところは、両者ともに同じ if-文を設定しているのに、異なった結論が引き出されていることである。これは発話を行なった二人の人間の知識がそれぞれ部分的であり、一致していないためであり、またこの知識の偏りが陽に述べられていないためである。以下に紹介するプログラム例は、Verdi および Bizet の国籍を尋ねられたときに、背後の知識を補って推論を行ない、それを答えるものである。

(8) および (9) に相当する自然言語表現は、*QUIXOTE* により以下のように直接的に翻訳される。

```
hypothesis_a ::
  bizet/[nationality = italy] <=
    compatriots[per1=bizet,per2=verdi] ;;
hypothesis_b ::
  verdi/[nationality = france] <=
    compatriots[per1=bizet,per2=verdi] ;;
```

最初の規則は、A という人の仮想の中で、「Bizet、Verdi という名の二人の人間が同国人である」ということから「Bizet の国籍がイタリアである」ことを導くものである。二番目の規則は、B という人の仮想の中で、「Bizet、Verdi という名の二人の人間が同国人である」ということから「Verdi の国籍がフランスである」ことを導くものである。これらの推論規則の連鎖の終端として、同国人および Bizet、Verdi の定義も以下のように必要である。

```
world ::
  compatriots[per1=X,per2=Y] <=
    X/[nationality=N1],
    Y/[nationality=N2] ||
    {N1=<nation,N2=<nation,N1==N2} ;;
world :: bizet;;
world :: verdi;;
```

⁸ 本来の例文では現在時刻 'are' の代わりに仮定法の 'would have been' が用いられ、仮想世界であることがよりはっきり明示されている。本稿では *QUIXOTE* 上で時刻や仮定法の表現を省く意図で単純な現在時刻とした。

これらの定義はどの状況でも通用するものであり、したがって最も一般的な世界 *world* で定義されたものとみなすことができる。すなわち、状況 (モジュール) 間の継承関係は以下になる⁹。

```
hypothesis_a >- world ;;
hypothesis_b >- world ;;
```

「同国人」の定義は次のように解釈される。もし、ふたつのオブジェクト *X*, *Y* が存在して、それらの *nationality* という属性の値が *nation* というオブジェクトに包摂される上、これが等しいとき、それら *X* と *Y* は *compatriots* という関係にある。また基礎オブジェクトの包摂関係は次のように定義されている。

```
nation >= italy ;;
nation >= france ;;
```

以上が規則のすべてである。A の知識、または B の知識というものがプログラム中どこにも明示されていないことに注意されたい。

次に *QUIXOTE* の推論がどのように働くかを紹介する。まず Bizet の国籍を尋ねる。

```
?-hypothesis_a:bizet/[nationality=N].
```

推論結果は以下になる。

```
** 2 answers exist **
** Answer 1 **
IF hypothesis_a:verdi.nationality == italy
THEN
  N == italy
** Answer 2 **
  N == Unbound
```

システムは二つの回答をしている。後者はより簡単な方で、このような推論規則の連鎖からは Bizet の国籍は推論できないとするものである。これは Prolog における推論と同様なものである。*QUIXOTE* の推論の特徴は前者のような回答も得られることである。すなわち、与えられた包摂関係の中で解を満たすモデルを見つけ、そのためにどのような条件が必要かを付加して返すというものである。この付加条件は回答中、if で書かれてある。同様な推論が Verdi の国籍についても可能で、B はもともと Bizet がフランス人であることを知っていた、ということを知ることができる。

4.3 仮想世界におけるオブジェクトの同一性

4.3.1 名詞句の参照の問題

本節ではいろいろな場面で議論となる名詞の参照の問題を検討する。この問題は不透明性 (*opaqueness*; *de re* と *de dicto* の区別)、照応のスコープ、比喩と換喩、ロールと属性・値の混乱などを含む。これらの例として以下のような文を考察する¹⁰。

Last year, the president was forty-six. (10)

Hitchcock saw himself in that movie. (11)

Hitchcock saw a unicorn in that movie. (12)

The mushroom omelet left without paying the bill. (13)

John believes that Verdi is French. (14)

最初の文 (10) は次の二通りに解釈することが可能である。

⁹ *QUIXOTE* の具象構文では、 $\sqsupset$ は $\supset$ である。

¹⁰ 本稿では基本的に状況理論 [Barwise 89] の記法を中心に記述するが、メンタル・スペースにも共通な問題を扱うため、例文も多くは [Fauconnier 84] から取る。

- 現在の大統領は昨年は46歳であった。
- 昨人大統領をしていた者は当時46歳であった。

同様に文(11)は次のように解釈の多様性がある。

- Hitchcock は映画の中で何かの役割を演じている自分を見た。
- Hitchcock は映画の中で誰かが自分 (Hitchcock という人) の役をやっているのを見た¹¹。

例文(12)は実在しないオブジェクトへ言及している。例文(13)は言及される対象がその名のものそのものではなく、それによって代わりに表現される人間を指している。換喩の例である。例文(14)は誰かの思い込みという仮想的な世界で指されているオブジェクトの属性の真偽が逆転されている例である。

4.3.2 ロール、属性、およびオブジェクト

最初に例文(10)の二通りの解釈について考察する。まず、最初の解釈を次に与える。

$$s_{last} \models \ll is-46, x \uparrow \ll president, x \gg \gg. \quad (15)$$

ここで、 $x \uparrow s \models \ll rel, x \gg$ という形の表記は制限付きパラメータと呼ばれ、「状況 s において、 rel という関係にある x 」を意味している。(15)では、 $'s \models'$ の部分が省略されているため、ここには全体の状況 ' $s_{last} \models$ ' があるものとする。すなわち、(15)では、大統領職にある者 x が状況 s_{last} で特化されている。これに対して、現在 s_{now} という状況を導入して、 x に別の制限を与えたのが(16)である。(16)は、メンタル・スペース理論風に言えば、 s_{now} における大統領ロールによってトリガされた、 s_{last} におけるターゲットを表現しているということができる。

$$s_{last} \models \ll is-46, x \uparrow s_{now} \models \ll president, x \gg \gg. \quad (16)$$

以上のような読みは、*QUIXOTE* によって次のように書くことができる。最初は(15)に対応するものである。

$$s_{last} :: o/[age = 46] \leq o/[role = president];; \quad (17)$$

ただしこの含意は *QUIXOTE* の記述としては必要ではなく、単純に

$$s_{last} :: o/[age = 46, role = president];; \quad (18)$$

と書いても同様なオブジェクトを指すことができる。(16)の意味を反映した *QUIXOTE* の記述は(19)のようになる。

$$s_{last} :: o/[age = 46] \leq s_{now} :: o/[role = president];; \quad (19)$$

次は例文(11)について考察する。

Hitchcock saw himself in that movie.

この文の曖昧性はロールとその値、および映画と現実世界という二つの状況を区別することによって形式化することができる。以下、状況理論の記法で書くかわりに直接 *QUIXOTE* により推論規則を書くことにする¹²。

```
real::see[agt=hitchcock,obj=X]
  <= movie:cast[name=hitchcock,mname=life_of_hitchcock,mact=X];;
real::hitchcock;;
movie::cast[name=X,mname=M,mact=X] <= M:X;;
movie::cast[name=X,mname=M,mact=Y] <= M:Y/[actor=X],real:X;;
life_of_hitchcock::man[place=bus_stop]/[actor=hitchcock];;
life_of_hitchcock::hitchcock/[actor=orson_welles];;
```

上のプログラムで、第1,2行は *hitchcock* と称される誰かが、*life_of_hitchcock* (Hitchcockの生涯) と呼ばれる映画の中で、*X* という俳優を見ていたことを記述している。第3行は実世界 *world* の中で *hitchcock* という人間が実在することを述べている。第4行と第5行は、*M* という映画の中で、次の二つの可能性を述べている。

¹¹ さらには、両方を組み合わせて Hitchcock 自身が Hitchcock という人の役をやっているのを見た、という読みもある。

¹² 本文ならば、自然言語によって指示されるオブジェクトというのはオントロジカルに存在する物理的実体に限定されるべきで、この意味で名前を直接オブジェクトとして書くことは避けるべきである。したがって *hitchcock*/[...] は悪い例であり、*o*/[name=hitchcock] と書くべきである。

- X という名前は映画の中の役の名前を言及する。
- X という名前は映画の中で X の役を演じている俳優 Y を言及する。

以上のようなプログラムに対して、次のような簡単な質問を行なう。

```
?-movie:cast[name=hitchcock,mname=life_of_hitchcock,mact=X].
```

これは、「映画の中の Hitchcock」に相当するオブジェクトにどんなものがあるかを問う質問である。QUIXOTE は次の二つの答を返す。

```
** 2 answers exist **
** Answer 1 **
    X == man[place=bus_stop]
** Answer 2 **
    X == hitchcock
```

また、Hitchcock が実世界で見たものを問うことにすると、

```
?-real:see[obj=X,agt=hitchcock].
```

QUIXOTE はやはり同様に上記二つの可能性があることを示す。

4.3.3 信念と換喩の解釈

文 (11) は、'Hitchcock' という名詞句に二通りの解釈があったが、一方の解釈しかない文もある。まず 最初に文 (12) について説明する。

Hitchcock saw a unicorn in that movie.

この文の場合には、実世界において unicorn に相当するオブジェクトが存在しない。したがってこの場合の可能な解釈はひとつに限られる。すなわち、unicorn は必然的に仮想世界のみに存在するという解釈である。つまり、前章の cast の例で言えば、前者の答しかない場合に相当する。

一方、換喩の例 (13) を考える。

The mushroom omelet left without paying the bill.

メンタルスペースでは、コネクタはトリガーとターゲットと呼ばれるオブジェクトの間の関係として定義される。この例で、料理からそれを注文した客を対応付けるコネクタは、QUIXOTE では次のように定義される。ここで、モジュール restaurant はこの文の発せられた状況 (レストラン) であるとする。

```
restaurant::role[trigger=X,target=X] <= X;;
restaurant::role[trigger=X,target=Y]
    <= Y/[order=X], real:X;;
```

例文 (13) においては、mushroom omelet に相当するオブジェクト (客) はこの状況には存在しない。より正確に言えば、mushroom_omelet というオブジェクト (料理) は存在しても、それは動詞 left と共存できないため、それは必然的に別のオブジェクトとなる。したがってここでは後者の読みのみが可能であり、それは換喩の解釈となる。

最後の例文 (14) は信念の問題である。

John believes that Verdi is French.

この文を解釈するためにはふたつの状況が必要である。すなわち、実世界と John の頭の中という仮想世界である。実世界では Verdi はイタリア人であり、したがって verdi の nationality という属性の値は italy になる。しかしながら、John の頭の中では、この属性の値が french になっている。QUIXOTE においてはオブジェクトの付帯属性の値はモジュール内のみで有効なため、以下のように記述することができる。(もし s_real というのが s_john の上位モジュールであったならば、このプログラムは規則の継承という意味で、矛盾を生ずる。)

```
s_real::verdi/[nationality=italy];;
s_john::verdi/[nationality=french];;
```

5 おわりに

本稿では、自然言語の複雑な現象を扱う言語として *QUIXOTE* を紹介した。伝統的な CLP 言語 [Jaffer 87] と比較すると、本言語はモジュールを持ち、また推論でもアブダクションが行なえる点で拡張的である。またシンボル制約の領域を持ち、自然言語の記述に向いた設計になっている。 *cu-Prolog* [Tsuda 94] の後継者として、*QUIXOTE* はいろいろな文法原理や語彙の曖昧さなどを制約として扱うことができるようになっている。また、状況推論の先駆者という意味では PROSIT [Nakashima 91, Nakashima 88] が提案されている。PROSIT では、各々のルールはひとつの状況の中でアサートされ、状況は階層構造をなし、ルールは状況間で継承される。*QUIXOTE* は基本的にはこれと全く同様な機構を有する。しかしながら PROSIT の中の簡単な形のインフォンは、複合オブジェクト項として素性構造を含んだ形で書くことができるよう拡張されている。このような複合オブジェクト項の概念は、最初 CIL [Mukai 85] の PST (partially specified term) から引き継いだものである。

QUIXOTE の新しい特徴は以下のように要約できる。

- オブジェクト同一性のようなオブジェクト指向の考え方が論理プログラミング言語に導入されている。
- モジュールが導入され、大規模知識ベースの中で局所的な定義ができるようになっている。
- 論理推論機構がアブダクションを行なえるよう拡張されている。

これらは自然言語処理用のツールとして次のような意義を生む。まず、近年の自然言語処理の最も重要なパラダイムのひとつである制約ベースの文法を記述でき、かつ制約論理規則として動的に運用できる。次に、これも近年の意味論で著しく重要な要件である意味の状況依存性を、多々ある状況依存意味論に対して統一的に記述ができ、かつまた状況推論規則として運用ができる。

本稿での適用例としては、まず第3節において、制約ベースの文法でのソート付き素性構造の記述を行なった。次に第4節において、状況に依存した意味記述ができることを示した。その例としては、まず隠れた知識に対して仮説推論を行なうシステムを提示した。そこでは与えられたオブジェクトの包摂関係をからモデルを見い出すことにより解が求められている。また、次の例として、名詞の参照のシステムを提示した。そこではモジュールを状況とみなし、オブジェクト同一性の概念を用いて、異なる状況で異なる対象を指すしくみを実現し、メンタル・スペースのコネクタの問題、知識と信念の問題、換喩の問題などが、状況理論の観点から同様な形式で記述できることを示した。

参考文献

- [Barwise 83] J. Barwise and J. Perry. *Situations and Attitudes*. The MIT Press, 1983.
- [Barwise 89] J. Barwise. *The Situation in Logic*. CSLI Lecture Notes No.17. Stanford:CSLI, 1989.
- [Carpenter 92] B. Carpenter. *The Logic of Typed Feature Structure*. Cambridge University Press, 1992.
- [Cooper 79] R. Cooper. The interpretation of pronouns. In F. Henry and H. Schnelle, editors, *Syntax and Semantics 10*. Academic Press, New York, 1979.
- [Dörre 91] J. Dörre. Feature Logic with Weak Subsumption Constraints. In *Proc. 19th ACL Annual Meeting*, pages 256-263, June 1991.
- [Dowty 81] D. Dowty, R. Wall, and S. Peters. *Introduction to Montague Semantics*. D. Reidel, 1981.
- [Evans 77] G. Evans. Pronouns, quantifiers, and relative clauses (1). *Canadian Journal of Philosophy*, 7, 1977.
- [Evans 88] G. Evans. Pronouns. *Linguistic Inquiry*, 11, 1980.
- [Fauconnier 84] G. Fauconnier. *Espaces Mentaux*. Editions de Minuit, 1984. (邦訳: メンタルスペース, 坂原, 水光, 田窪, 三藤訳, 白水社, 1987.)
- [Heim 90] I. Heim. E-type pronouns and donkey anaphora. *Linguistics and Philosophy*, 13:137-177, 1990.
- [Nakashima 88] H. Nakashima, H. Suzuki, P. Halversen, and S. Peters. Towards a computational interpretation of situation theory. In *Proc. FGCS '88*, pages 489-498, 1988.
- [Nakashima 91] H. Nakashima, S. Peters, and H. Shutze. Communication and inference through situations. In *Proc. IJCAI'91*, pages 76-81, 1991.

- [Jaffer 87] J. Jaffer and J.-L. Lassez. Constraint Logic Programming. In *Proc. the 14th ACM POPL*, pages 111–119, Munich, 1987.
- [Kamp 81] H. Kamp. A theory of truth and semantic representation. In J. Groenendijk, T. Jansson, and M. Stockhof, editors, *Methods in the Study of Language Representation*. Math Carter, Amsterdam, 1981.
- [Mukai 85] K. Mukai and H. Yasukawa. Complex Indeterminates in Prolog and its Application to Discourse Models. *New Generation Computing*, 3(4):441–466, 1985.
- [Geach 62] P. Geach. *Reference and Generality*. Cornell, Ithaca, New York, 1962.
- [Pollard 94] C. Pollard and I. A. Sag. *Head-Driven Phrase Structure Grammar*. University of Chicago Press and CSLI Publications, 1994.
- [Shieber 86] S. M. Shieber. *An Introduction to Unification-Based Approach to Grammar*. CSLI Lecture Notes No. 4. Stanford: CSLI, 1986.
- [Tojo 92] S. Tojo and H. Yasukawa. Situated inference of temporal information. In *Proc. FGCS '92*, volume 1, pages 395–404, 1992.
- [Tsuda 94] H. Tsuda. cu-Prolog for Constraint-Based Natural Language Processing. *IEICE Transactions on Information and Systems*, E77-D(2):171–180, February 1994.
(システムは <ftp://ftp.icot.ot.jp/ifs/kbms-clp/unix/cuprolog.tar.Z>)
- [Quine 59] W. V. Quine. *Methods of Logic (revised edition)*. New York: Holt, Rinehart and Winston, 1959.
- [Yasukawa 92] H. Yasukawa, H. Tsuda, and K. Yokota. Objects, Properties, and Modules in *QUIXOTE*. In *Proc. FGCS92*, volume 1, pages 257–268, 1992.
- [Yokota 92] K. Yokota and H. Yasukawa. Towards an Integrated Knowledge-Base Management System. In *Proc. FGCS92*, volume 1, pages 89–112, 1992.
- [西岡 93] 西岡 利博, 小島 量, 津田 宏, 横田 一正. 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. 第 94 回情報処理学会データベースシステム研究会, pages 1–10, July 1993.
- [安川 90] 安川秀樹, 横田一正. ラベルつきグラフに基づくオブジェクトの意味論. 情報処理学会データベースシステム・人工知能研究会, pages 119–127, November 1990.