

TM-0508

状況指向型 意味 表現言語 USSR

安川秀樹, 鈴木浩之(松下)

May, 1988

©1988, ICOT

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

状況指向型意味表現言語 USSR

安川秀樹 鈴木浩之

松下電器産業㈱ 技術本部 システム東京研究室

1. はじめに

現在、我々は契約書の作成支援を対象とする相談型の自然言語対話システムT o Rの研究開発を行っている。本論文はT o Rシステムの意味・文脈処理のための表現形式と推論体系を与える意味表現言語USSRについて述べる。USSRは状況理論[Barwise 85]に基づいたものであり、「もの」を中心に物事・概念を分類するものである。

T o Rシステムにおいて、意味表現が関係する処理は大別して、以下の4つである。

- (1) 自然言語文の意味を記述する。
- (2) 文脈(対話環境)を記述・管理する。
- (3) 対象世界の制約に基づき文の解釈・評価を行う。
- (4) システムの次の動作を決定し、ユーザへの応答を決定する。

これらの処理の各局面において必要とされる機能を提供することがUSSRの目的である。次章では、これらの機能の詳細化を行う。

2. 意味表現への機能要求

ここでは、前章で挙げた(1)~(4)の各局面に対応して、いくつかの観点から意味表現が提供すべき機能について述べる。

(1) 自然言語文の意味を記述する。

文の言語的意味(linguistic meaning)を記述するのではなく、与えられた文脈においてその解釈を決定するために必要な情報を適切な形式で表現する枠組みを提供することが基本的な目的である。ただし、言語的意味との対応が取りやすい表現形式であることが要求される。主に解析・生成での処理のために、以下のような機能を提供する必要がある。

- ①概念を構造として操作する。
- ②概念の意味的な分類の検索・照合を行う。
- ③語彙選択などのための指針を与える。

(2) 文脈(対話環境)を記述する。

対話の過程を通じて具体化され拡大されていく文脈(環境)や対象領域に関する知識・制約条件などを記述し、管理するための枠組を与える。言語的な意味での文脈だけでなく、関連する状況を記述し、操作するために以下のような機能を提供する。

- ①文脈(状況)の構造的な取扱い。

- ・文脈構造を記述する。

②部分的な記述とその操作。

- ・文脈中の一部分だけを切り出す。
- ・部分的な記述を文脈全体に対してあてはめる。

③視点(文脈の切り替え)の基本的な要素としての導入。

- ・文脈情報の場面に応じた柔軟な検索を可能とする。

(3) 対象世界における制約に基づく解釈と評価。

ユーザの入力文をその時点における文脈で解釈し、その文が文脈中で果たす役割を判断・評価する。

①対象領域における予測的知識との柔軟な照合。

- ・対象領域における知識と入力文によって表されるユーザの意図との対応のチェック。

②解釈結果による文脈の具体化・拡大を可能とする

- ・照合による文脈の具体化
- ・照合結果と制約条件による文脈の一貫性の確認
- ・照合結果と制約条件による文脈の有意味な拡大(対象世界に関する制約などから逆に得られる情報(inverse information)の具体化)

③文脈の一貫性を保持するための機能。

- ・制約条件の適用機構(推論機構)の導入
- ・矛盾検出機構

(4) 対話を制御する(話題選択)ための情報の管理。

- ①状況・事物間の関連性を記述する枠組みの提供
- ②ユーザの入力との照合による関連性の判定
- ③関連性に基づく次話題選択とその内容に対する表現の設定

以上のことから、意味表現に対して求められる機能は以下の4つに整理される。

- I 構造として取り扱える表現形式
- II 視点の導入による柔軟な表現の操作
- III 部分的な記述
- IV 記述・制約条件(規則)に関する推論のための形式的な体系

次章では、これらの要件を前提に意味表現言語の基本的な仕様について述べる。

3. 意味表現言語USSRの設計仕様

3.1 基本的要請

前章で議論したように、われわれは、意味表現言語は、下記の4つの要請を満足する必要があると考える。

- (1) 表現が構造として取扱える
- (2) 視点の機能が導入可能である
- (3) 部分を切り出して記述することが可能である
- (4) 記述・推論の形式的な体系が与えられる

3.2 従来の表現形式の問題点

それでは、上にあげた要請を念頭におき、従来から提案されてきた代表的な意味表現の枠組みであるフレーム・論理式とを概観してみる。

3.2.1 フレーム

フレームは、「形にはまった状況を表現するためのデータ構造」[Minsky 75]として提案されたものである。USSRもこの提案の枠を大きくはみ出すものではないが、ここでは、その後、構造的な知識表現手法として、リスプなどでインプリメントされた形の(狭義の)フレームとの比較を行なう。

フレームは、構造的な表現として計算機上で効率的に実現されている。しかし、その構造は固定的であるため次のような問題点がある。

(1) 視点

異なった視点から同じ対象を見た場合の記述が難しい。対象自身とデータ構造とが単純に同一視されているため、同じ状況が二つの状況のタイプを満たしている場合や同じ対象が複数の状況の構成要素となった場合の記述法がない。そのため、アクセスなどに関する管理をフレームシステムの外にユーザが手続きに行わねばならない。

(2) 推論と意味づけ

部分的な情報が与えられた場合にその情報が現在の状況とどう関わるのかの統一的な記述がない。

フレームシステムでは、各スロットにデフォルト値や値の制約関数の記述を書き加えることで、情報の予測や的確性の判断に用いている。また、スロット値が代入されたときに起動する関数を用いて、情報の伝搬を行なう。また、おおくのインプリメンテーションでは、同一のフレームに属するスロット間の関係の記述は不可能である。しかし、これらは本質的には、状況の構成要素間の関係として状況のタイプの定義に一本化して書かれるべきものである。

3.2.2 論理式

論理式による知識記述はその体系の形式性が高くて大きな特徴であろう。例えば、一階述語論理式は演繹体系などの非常に整った形式的記述体系を与えられている。

しかし、一階述語論理式自体は構造を持たない。そのため、ひとかたまりの知識や知識の構造といったものを素直に記述することはできない。

また、一階述語論理式で仮定されている世界は全体として透明である。すなわち、データが、いつでも、どこからでも、同じ様に、全て見える。したがって、視点や部分的な記述といった概念との親和性がよくない。また、自然言語の持つ基本的な性質である文脈依存性の記述ができない。

原則的には、前記の問題点の多くは、高階の論理体系(例えば、内包論理)を用いることで解消することは可能である。しかし、高階の論理体系でも、言語の文脈依存性を記述する上で問題があることが指摘されている。また、高階の論理体系に準拠した形で、推論などの処理を行うことは、計算量的に困難である。

3.2.3 フレームと論理式との融合

上記のように、フレームは構造的な表現力に優れ、論理式は形式的な推論を行なう能力に秀でている。そこで、両者の利点を生かす形で、両者の融合を図る研究が、KRYPTON[Brachman 84]などをはじめとして、行われてきた。しかしながら、部分記述性や視点の導入など、先に述べた我々の要求を満たすものではない。

3.3 USSRの仕様

これらの要素を考慮・検討し、状況理論をベースに意味表現言語を設計・開発した。上記の(1)~(4)の四つの基本要請は状況理論のつぎのような特徴を用いれば実現が可能である。

(a) 型多様な(polymorphic)タイプシステム

状況理論では、記述は実在の状況を分類するためのものであるとして、位置づけられている。したがって、一つの状況はいろいろな状況型になりうる。そこで、状況の記述をそうした状況型の合併で記述することで、構造的な記述を行なうことができる。また、視点を状況のタイプの集まりで表現し、タイプ間の上位下位関係を用いて実在の状況の分析を行なうことで、視点の機能を実現できる。

(b) 単一化

上記のように、状況を状況型で分類する形式をとるため、一つ一つの情報は一つもしくは複数の状況型となり、部分記述されている。しかし、部分的な記述をまとめて一つのものとする方法が必要である。この操作が単一化で実現できる。タイプ付きの論理変数の単一化がその基本となる。

(c) 関係・制約の評価

推論は、状況型の定義に書かれる関係の記述を評価することと、制約規則の実行により行なえる。関係の評価も制約条件の一種であるが、フレームというスロ

ットの値制約関数やいろいろな attached procedure にあたるものとして、別に扱うこととした。

これらの推論機能の形式的な体系は状況理論によって与えられる。

4. 状況理論

状況理論 (Situation Theory) は、状況意味論 (Situation Semantics) の土台の部分であり、言語が仮定している世界像を記述するための理論である。[Barwise 83,85]

状況理論では、関係・個体・場所・時間・事態・条件・タイプ・不確定項などといったものすべてが、一次的な要素、プリミティブ (primitive) である。すなわち、 n 項関係は個体の n 個の集合によって (外延的に) 定義されるものではなく、それ自身として、すでに存在するものである。そして、状況理論は、それらを用いて実際の状況を分類する手段を与えるものである。

以下、USSRの基本要素に対応する状況理論の基礎的な概念の説明を行なう。状況の理論より詳細な解説は、文献[向井 86]などを参照してほしい。

4.1 関係と事態

n 項関係 R には、それら n 個の変数位置 (argument place) の集合 $\text{Arg}(R)$ が付随している。それら変数位置に個体などのオブジェクトを対応づける関数 a を割当関数 (assignment) と呼ぶ。オブジェクトとして、不確定項 (indeterminate) を導入する。不確定項は、 $x \downarrow$, $y \downarrow$ などと表わす。

関係 R に対して、適切な割当関数 a および極性 p が与えられたとき、それらの組 $\sigma = \langle R, a : p \rangle$ を基本事態 (basic state of affairs) と呼ぶ。ここで、極性 p は値として 1 または 0 をとる。極性が 1 であることは、命題 $R(a)$ が成立することと同値であり、極性が 0 であることは、命題 $\sim R(a)$ が成り立つことと同値である。

これら事態の間に順序関係 $\triangleright$ (as strong as) が存在する。 $\sigma \triangleright \sigma'$ であるというのは、 σ が事実として成り立っていれば、必ず、 σ' も事実として成り立つという意味である。この順序 $\triangleright$ を用いて、関係の OR や AND に相当する事態間の和と積が定義される。基本事態の和や積の結果えられるものはすべて事態 (state of affairs, 以下 SOA と略記する) である。

4.2 パラメータ・条件・タイプ

事態の各変数位置にその変数位置にアクセスする手段としてラベルづけを行なう。関係の変数位置を定義域とする関数 l によってラベルづけられた事態をパラメータ化事態 (parametric SOA) と呼び、 L の値域の

各要素をハラメータと呼ぶ。

ラベル関数 l と割当関数 a とが与えられたときに、 $a = l \circ f$ となる関数 f をアンカー (anchor) と呼ぶ。アンカーはパラメータからオブジェクトへの関数である。 $\sigma[f]$ で、上記のような割当をもつハラメータ化事態を表わす。

ハラメータ化事態 σ が与えられたときに、 $\sigma[f]$ が事実であるとき、そのときに限って、条件 C がアンカー f に対して成立するという。

このとき、 x が二つのハラメータ化事態 σ と τ のハラメータであるときには、一方のハラメータ化事態 τ はもつ一方のハラメータ化事態 σ 中の x の値に制約条件を与えているとみることかできる。すなわち、条件 C が τ によって拘束されたハラメータ $x^T = x \upharpoonright C \tau$ が σ のハラメータとして用いられているとみなせる。こうしたハラメータを条件つきハラメータ (conditioned parameter) と呼ぶ。

条件つきハラメータの族 $(x \upharpoonright C \tau)$ は、割当関数 a と τ 中の変数と間の関係を与える。その関係は、 a に対応するアンカー f が条件 C を満たすとき、そのときにかぎり、成り立つような関係である。そのような関係をタイプ (type) と呼び、 $[x \upharpoonright C \tau]$ で表わす。

4.3 マージ

事態は実在の状況を分類するために導入したものであるが、実在の状況は、多くの事態によって分類される。そういった多面的な分類を可能にするには、事態間に新たな演算マージを導入する必要がある。

二つのハラメータ化事態 $\sigma = \langle R, a : i \rangle$, $\tau = \langle R, b : i \rangle$ が両立的であるというのは、割当関数 a と b とが関数として両立的であることである。そのときに、関数 a と b を貼合わせてできる関数 c によって決まる事態 $\langle R, c : i \rangle$ を $\sigma \oplus \tau$ で表わす。この操作をマージ (merge) または単一化 (unification) と呼ぶ。

マージ $\sigma \oplus \tau$ は、事態の積 $\sigma \wedge \tau$ とは異なる。一般に、 $\sigma \oplus \tau \triangleright \sigma \wedge \tau$ であるが、その逆は成り立たない。

4.1 制約

基本制約 (basic constraint) とは、関係 involves と二つの条件 C_1 と C_2 とでつくる基本事態

$$\langle \text{involves}, C_1, C_2 : 1 \rangle$$

のことである。ある事態 σ が基本制約 r と $\sigma \triangleright r$ であるとき、事態 σ を制約という。すなわち、制約は条件間の関係である。制約には、論理的制約 ($\triangleright$)・法制的制約・慣習的制約など多くのものが存在する。

5. 意味表現言語 USSR

USSRは、状況理論に基づく意味表現言語であり、関係(事態)により物事を分類(classify)することを基本とする。表現の基本的な枠組みは状況理論を受け継いでいる。計算機上での処理のために、若干の拡張とともに幾らかの制限が設けられている。以下では、USSRの基本的な要素について説明する。

5.1 USSRにおけるオブジェクト

状況理論と同様に、プリミティブもしくはいくつかのプリミティブの間で成り立つ関係に従って、物事や概念を分類するのが、USSRにおける意味表現の基本的な考え方である。その分類の基本となるのがタイプである。USSRでは、ある概念定義(タイプ)にしたがって、そのタイプのオブジェクト(USSRのデータ)を生成する。オブジェクトは、一般には、あるタイプのプリミティブに対応するものであり、タイプの制約が付けられた変数と考えられる。

USSRで操作の対象とするUSSRオブジェクトとしては、プリミティブに対応するインスタンス、プリミティブからなる集合、総称的な概念を表す総称インスタンス(generic instance)の3種類がある。これらの区別は、個々のUSSRオブジェクトが持つ種別(sort)によって表わされる。

また、固有概念(対象世界の特定のプリミティブを指すインスタンス、例えば、日本など、individualと呼ぶ)の定義を行うことが可能である。

5.2 関係(事態)

USSRでは、状況理論と同様に事態を物事・概念の分類の基礎を与える最も基本的な要素としている。USSRでは、区別の必要な時以外は事態を関係と呼ぶ。事態は以下に示すように構成子"&"と"or"(それぞれ状況理論の積と和に対応する)を用いて、複雑な関係を定義する。

`man(John) & woman(Mary) & ~marry(John, Mary)`

なお、USSRでは、否定記号"~"の有無が極性の0、1に対応する。

USSRでは、事態は下に示すようにタイプ定義中の"where"以下のところにだけ現れる。

```
愛する(主体:X, 対象:Y) =>
where
  X typeof 人 &
  Y typeof 人 &
  love(X, Y).
```

図1 USSRにおけるタイプ定義の例

USSRでは、処理系で特殊な解釈が行う事態をいくつか用意してある。以下にその代表的なものを示す。

• `X = Y`

XとYが(単一化の意味において)同一であることを示す。実際にXとYの単一化がおこなわれる。

`X = Y` は `merge(X, Y)`の略記である。

• `mergable(X, Y)`

XとYが単一化可能か否かを調べる。

• `same(X, Y)`, `X == Y`

XとYの値が厳密に同一である(strictly equal)か否かを調べる。現在の処理系では、Prologの"=="と同様に、呼ばれた時点におけるチェックとしてしか機能しない。

`X == Y` は `same(X, Y)`の略記である。

• `X typeof Y`

プリミティブXがYのタイプであることを示す。

あるタイプの定義中に現れた場合、そのタイプのオブジェクトの生成の際にXをYのタイプのオブジェクトとする。

これらの関係は、上記のような意味を持つものとしてUSSR処理系によって評価される。

5.3 ハラメータ

USSRでは、事態(関係)はタイプ定義において"where"以下のところにだけ現れ、その各引数に対するハラメータとしてのラベル付けはタイプ定義の左辺(タイプ表現)に示される。なお、ある事態に現れる引数が全て、ハラメータ化されている必要はない。したがって、状況のある事態に基づいて分類し、その関係に関わっているオブジェクトのうち、特に着目すべきものだけをハラメータとして定義し、アクセスすることができる。これは、N個のオブジェクト間に関係Rが成り立っている場合に、特にそのうちの例えば2個だけに着目して、その2つのオブジェクトの間の関係を定義する枠組みを与えるものであり、視点を導入する基礎となる。

5.4 概念(タイプ)定義

タイプはUSSRにおける物事や概念の分類・定義に用いられる最も基本的な要素である。状況理論では、タイプはハラメータ化事態の条件によって拘束されるハラメータの族として定義される。USSRでは、ハラメータ化事態が成立するか否かは単一化によって動的に検査されるため、タイプをハラメータ化事態と同様に取り扱う。ただし、USSRにおけるタイプは他のタイプ定義中に現れうるなど、ハラメータ化事態と異なった性格を持つものであり、その点、状況理論におけるタイプと同じである。USSRにおけるタイプ

は、大別して状況型とロール型の2種類に分類される。また、状況型はさらに単純型と複合型の2種類に細分される。

状況型とロール型の分類は主に、前者が対象世界におけるオブジェクト間の関係を定義するのに対して、後者は対象世界で成り立っている関係において何らかの役割を果たしているもの（条件付きパラメータに対応する）を定義する。状況型のタイプは全て状況タイプである（基本タイプsitの下位タイプである）。

単純型と複合型は状況型のタイプ定義のための枠組みである。単純型は前述の例のようなパラメータ化事態による状況の分類を行うものである。それに対し、複合型は他の状況型のタイプ定義を利用して、新たなタイプ定義を行うための枠組みである。複合型のタイプ定義はタイプ間に半順序関係を導入する。タイプ間の半順序関係は、意味的な下位分類の基準を与える。また、後述のように、タイプ間の半順序関係は単一化（マージ）の際の制約を与えると言う意味においても重要である。

なお、USSRでは、個体(ind)、状況(sit)、時間(time)、場所(place)、数(number)、整数(integer)、正の整数(positive_integer)、負の整数(negative_integer)、文字列(string)、および無型(notype)の10のタイプを基本的なタイプとして予め設定している。無型はタイプの半順序関係の上限(upper bound)であり、全てのタイプの上位タイプである。

5.4.1 状況型のタイプ

上で述べたように状況型には、単純型と複合型に分類される。すなわち、USSRでは、同じものを定義するのに、2通りの方法が許されることになる。すなわち、あるタイプを定義するのに、事態を用いて直接的に記述する方法とそのタイプの一部分となるような複数のタイプ（下位のタイプ）を組み合わせて記述する方法が可能となる。USSRでは、運用上、後者を望ましいものと考えている。後者の方式は、従来の意味表現言語における重要なファクタである概念継承を自然なかたちで導入するものであり、対象世界の記述を階層的に行うことを可能とする。

USSRにおける単純型のタイプの定義は、先に示した図1のようになる。一般的には、

```
<タイプ表現> '=>'
  'where'
  <事態>
```

ここで、タイプ表現は5.3で述べたように、パラメータ名と事態中に現れるオブジェクトとの対応を明示するリテラルである。USSRのタイプに関して、特に明記すべきは、上記のタイプ定義が単にオブジェク

トの構造を規定するだけでなく、あるオブジェクトがそのタイプであるかどうかを判断する基準を与えるものであるということである。つまり、あるオブジェクトがタイプTの定義に示される制約条件を満足するならば、そのオブジェクトはTのタイプであると結論できるわけで、単一化による状況の具体化・拡大の結果を、タイプを介してマクロに管理する枠組みを与える。

タイプ定義において、その関係中に現れる変数（オブジェクト）があるタイプであることを記述したい場合には、図1に示されるように関係typeofを用いて、その変数のタイプを指定する。また、事態中のオブジェクトが状況のタイプである場合には、下図に示すように事態中の他のオブジェクトと、その状況のタイプのインスタンスのパラメータとの関係を記述することができる。

```
製品の製造を許諾する(主体:X,相手:Y,製品:Z) =>
where
  X typeof 法人 &
  C typeof 製品を製造する(主体:Y,製品:Z) &
  許諾する(X,Y,C).
```

図2 オブジェクトへのタイプ制約の例

図2においては、Xが法人のタイプであること、Yが製品の製造主体であり、かつ、被許諾者であること、Zが製造される製品であることなどが記述されている。

USSRにおける複合型のタイプ定義は以下のような形式となる。

```
<タイプ表現> '=>'
  'composite'
  <タイプ表現> '*' ... <タイプ表現>
  'where'
  <事態>
```

複合型のタイプ定義の例を下図に示す。

```
製品の製造を許諾する(主体:X,相手:Y,製品:Z) =>
composite
  許諾する(主体:X,相手:Y,内容:C)
where
  C typeof 製品を製造する(主体:Y,製品:Z).
```

図3 複合型のタイプの定義の例

この例は、図2に示したのと同じタイプを、“許諾する”というタイプを利用して定義している。構文要素compositeに引続いて指定されるタイプを組み合わせてタイプを定義することを示す。複合型のタイプ定義により、タイプ間の半順序関係が指定される。この例の場合、“許諾する”は“製品の製造を許諾する”の上

位タイプとなる。このようなタイプ間の半順序関係によって、オブジェクトのタイプの多態性(polymorphism)が導入される。タイプ間の半順序関係は、いわゆる多重継承(multiple inheritance)に対応するものとみることができるが、図3に示すように複合された各タイプのパラメータ間の対応関係は、変数の対応関係として明示的に記述するため、より細かな記述が可能である。その反面、通常概念継承とは異なり、あるオブジェクトが複数のタイプとしてみることができても、パラメータはそれぞれのタイプで独立に定義されている必要がある。

5.4.2 ロール型のタイプ

状況型のタイプは、状況の分類を定義するもので、ある状況において何らかの役割を果たしているものを定義するものではない。従って、ある状況型のタイプのパラメータであることによって分類される概念(条件付きパラメータに対応する概念)をタイプとして記述する枠組みを以下のように提供する。

```
<タイプ定義> ::=  
  <タイプ名> '=>  
  <パラメータ名> 'of' <タイプ名>
```

このようなタイプをロール型のタイプと呼ぶ。以下にロール型のタイプの定義例を示す。

男 => 本体 of 男である。

男である(本体:λ) =>

```
where  
  λ typeof 人 &  
  male(λ).
```

人 => 本体 of 人である。

人である(本体:λ) =>

```
where  
  λ typeof ind &  
  human(λ).
```

図4 ロール型のタイプの定義例

図4のように“男”というタイプを定義すると、“男”のインスタンスは、男のタイプであり、人のタイプであり、かつ、ind(個体)のタイプである。このように、ロール型のタイプ定義によってもタイプの間の部分順序関係が規定される。

図4の“男”というタイプは、“男である”という状況型のタイプによって定義される。この時、“男である”を“男”の属性(プロパティ、property)と呼ぶ。

5.5 集合

以上で述べたタイプは、プリミティブやプリミティブの間に存在する制約条件を規定するもので、何等かの性質を持つものの集合を表わしているのではない。

一般的に集合の表現方法としては、内包的なもの(外延的なもの)がある。USSRでは、内包的、外延的定義の両者を許す。集合はUSSRのオブジェクト(データ型)として取り扱われ、内包的定義、外延的定義、濃度によって規定されるものである。集合に対して、生成、包含、要素の検索等の操作が許されている。

集合に関して考えられる問題点としては、タイプ定義において、あるパラメータにタイプによる制約が付けられている場合に、集合がそのアンカーとして認められるか否かということが挙げられる。すなわち、図5に示した“使う”というタイプのオブジェクトのパラメータ“主体”へのアンカーとして、例えば、太郎と次郎を要素とする“人”のインスタンスの集合が許されるか否かということである。USSRでは、タイプ定義中の変数に関するタイプの制約はアンカーに関する条件を示すものとして取扱う。つまり、その変数(オブジェクト)のタイプは決定されるが、種別(sort)は未定であり、集合の内包的定義およびその全ての要素が制約を満足するならば、集合もアンカーとして認められるという解釈を行う。

5.6 単一化

USSRにおけるオブジェクトは、タイプ定義に従って合成されるものだが、部分的な具体化を与えられたものである。従って、それらの部分的な記述を具体化するための手段、つまり、状況理論におけるアンカーに対応する手段が必要とされる。

USSRでは、アンカーに対応する値割当は単一化によっておこなわれる。すなわち、2つのオブジェクトの単一化により値の割り当てを保存する。USSRでは、単一化に関して2種類の手続きを用意している。1つはmergeであり、残りの1つはuniteである。

単一化は、2つのオブジェクトに対して両立的な(最小の)値割当を適用するものである。状況というものが本来部分的な記述であるため、違うタイプの2つのオブジェクトの単一化は両者の和をとることで成功してしまう。つまり、(偶然的に)特許のタイプのオブジェクトと国のタイプのオブジェクトの単一化が成功し、特許でも国でもあるものが合成されてしまう恐れがある。ところが、意味解析などで頻繁に行われるような概念の下位分類による適用可能性のチェックにおいては、このような単一化の定義では問題がある。そのため、上で述べた(一般的な意味での)単一化としてuniteを、さらに、概念間の半順序関係に従った強い意味での単一化をmergeとして提供している。

上記の例のような単一化を避けるためには、「特許であるものは同時に国であることはありえない」という対象世界に関する制約を記述する必要がある。しかし、このような制約を網羅的に記述することは多大な労力を必要とし、かつ、システムの見通しを悪くする。そこで、mergeでは、タイプ間の半順序関係を単一化に関する制約として利用する。すなわち、2つのオブジェクトがタイプの半順序関係によって規定される階層（ラティス）上で共通の下界(lower bound)を持つ場合にのみ単一化が成功する。タイプの階層上で明示的に示されていないことは、成り立たないと考えるのであるから、ある意味では、閉世界仮説に基づく制約が課せられた単一化がmergeであるといえる。

5.7 制約(constarints)

制約は、オブジェクト間の制約条件を記述するもので、推論規則として、あるいは、対象世界の一貫性を保持する規則として機能するものである。USSRでは、制約は、図5に示すようにタイプ表現をリテラルとする節形式で記述される。

```
no(arg1:X,arg2:Y) ->
  X typeof seihin &
  Y typeof kaisya &
  製造する(agent:X,object:Y,time:T),

製造する(agent:X,object:Y,time:T1) <-
  販売する(agent:X,object:Y,time:T2) &
  precedes(time1:T1,time2:T2).
```

図5 USSRにおける制約記述例

USSRでは、制約規則の適用方式として前向き(forward)と後向き(backward)の2種類を用意する。ただし、それぞれの規則がどちらの方式で適用されるかは制約の記述によって区別される。図5の最初の制約は、後向きに、2番目の制約は、前向きに解釈される。制約の適用に関して補足すると、現在のUSSR処理系では、導出原理(Resolution Principle)に基づく一般的な問題解決手法の導入は行っていない。部分的な記述に対して、そのような一般的な手法を適用した場合に効率的な実現が望めないことが、その理由である。

5.8 USSR処理系

USSR処理系は、本章でこれまでに説明したUSSRの意味表現の枠組みに従って、オブジェクトを生成し、操作する機能を提供する。現在、Quintus Prolog上にサブセットが実現されており、ToRシステムで利用されている。以下では、その処理系の概要について述べる。

5.8.1 USSRデータベースと中間的要素

USSRは意味表現の生成や意味表現に対する操作

とともに、計算機上に作り上げられていく対象世界（環境）の管理を行う。USSR処理系では、この環境の蓄積と管理のため、データベースを用いる。データベースはその存在・意味付けが固定されたオブジェクトを蓄積するものであり、ユーザとの対話によって徐々に拡大・拡張して行く。別の言葉で言えば、世界を表わすものである。基本的には、このデータベースがUSSRの様々な処理の基礎となる環境を与える。しかし、意味解析・文脈解析・文生成の過程において、実在しないオブジェクトが生成される可能性があるため、USSR処理系では、データベースに登録されない中間表現とデータベース表現の2種類のオブジェクトを用意している。中間表現は処理の過程で一時的に作られるものであり、タイプ定義に指定された情報や適用された操作の結果、得られる情報を保持する。ただし、データベースに登録されていないため、ある時点の環境全体において、どのような役割を果たしているかという情報は得られない。処理の過程において、中間表現で表されているオブジェクトが実在するものと判断された時点でデータベースに格納し、その存在・意味付けを固定する。

中間的表現は、タイプ定義をコンパイラによって実行可能なPrologプログラムに変換し、それを呼び出すことにより行われる。

データベースに関する操作の上で最も注意すべき点は、その一貫性の保持であり、今回の実現では、次の2点について、特に考慮した。

(1) 事態の成立に関する一貫性

各引数に関して同じ値割り当てをもち、極性が逆の事態がない事を保証する。

(2) 中間的要素とデータベース要素との対応

処理の途中で作られる中間的要素（の一部）に対して、データベース要素がマージされる場合、データベース上の値割り当てを直接変更する訳にはいかないので、データベース上の値割り当てをコピーし、中間的要素に与える。ただし、その場合、コピーされたものとデータベース上に保持されたものの間の一貫性を保つのがきわめて困難であり、問題が多く残されている。

5.8.2 USSRの基本操作用語

USSR処理系では意味表現の操作のために、さまざまな基本述語を提供している。以下では、その代表的なものを示す。

(1) オブジェクト生成

```
create
あらタイプのオブジェクトの生成
```

(2) オブジェクト関連情報の参照

parameter_of
オブジェクトのパラメータを参照する

property_of
オブジェクトの属性を参照する

typeof
オブジェクトのタイプを調べる

bound
オブジェクトが確定項かを調べる

unbound
オブジェクトが未確定項かを調べる

(3) 単一化関連

unite
2つのオブジェクトを単一化する

merge
2つのオブジェクトをマージする

mergable
2つのオブジェクトがマージ可能性を調べる

same
2つのオブジェクトの同一性を調べる

(4) データベース要素の登録

put_db
オブジェクトをデータベースに登録する

(5) データベース推論・検索

hold
あるタイプの成立を調べる

hold_rel
ある事態の成立を調べる

以上のほかに、全体で、約40の操作用述語が準備されている。

6. USSRの利用例

本章では、USSRの部分記述性・多態性を利用した自然言語処理法の例を示す。

6.1 文生成

USSRのオブジェクトから、それに対応する文の生成を行なう方式の概略を述べる。

発話したい内容は、一つのまとまりをなすので、それに対応するひとかたまりのUSSRの(中間)表現を生成する。そのときに、質問であるとか、陳述であるとかは、システムの状況であるとして記述しても構わないが、ここでは簡単のためそれらは別枠で表現されているものとして、発話内容のみがUSSRで記述されているものとする。

文の主動詞は、与えられた表現のタイプを調べ、主動詞の候補を選定することによって決定される。一般には、その際に、多態的なタイプをもちいているため、複数の動詞の候補が存在する。その場合には、今までの談話の流れの状況の記述などに基づいて、一つに決める。

つぎに、その動詞の必須格にはいる名詞類を、その

動詞に対応する状況タイプを視点として状況を見た場合に、その格に対応するパラメータから生成する。

(1) 固有概念に対応すれば、固有名詞とする

(2) 既知の対象であれば、以前にその個体を参照するために使われた表現または代名詞

(3) その個体のタイプに対応する普通名詞

という規則を適用し、選択する。

また、状況に付随している時に関する記述から時制やアスペクトの決定を行なう。

USSRを用いた対話における応答文生成の1方式が[野口 87]に述べられている。

6.2 話題選択

相談型の対話においては、システムが相談の領域について詳細を知っており、ユーザが述べる希望や現在の状況を基に、情勢を判断していく必要がある。このことをUSSRの用語を用いていうと、

構文・意味解析の結果である発話の意味表現のタイプを判定し、それに基づいて、その状況の構成要素や参加者に対して役割を付与する。また、その状況自身に対しても、状況の良否などの判断を行ない、それも状況として記憶する。

上記のような規則は、USSRの制約規則として記述可能である。

さて、このとき話題を転換する、すなわち、計算機の側が次話題を選択するという機能をも、上記の制約規則に基づいて行なうことができる。[本池 86] 以下、例をもとにその概略を説明する。

契約書作成支援システムを考えた場合、次のような制約が成り立つ。この制約は、現在のユーザの希望や現状を、契約上重要な状況の記述に対応づけるものとして用いられる。

「AがBの所有する特許Cを使っている」ならば「Aがライセンシー、Bがライセンサーである特許ライセンス契約でCをライセンス対象特許に含むものが結ばれる」

```
keiyaku(licensee:A,licensor:B,taishou_tokkyo:S)&
includes(S,C) /* 集合の包含関係 */
<- /* involves */
tsukan(agent:A,object:C,time:T0) &
C.typeof tokkyo &
shoyuu(agent:B,object:C,time:T1).
```

このとき、ユーザの発話内容が上の規則の前提部が示しているタイプの状況の一部であったとする。そのときには、すでにわかっている事実と発話内容とを併せて、上記制約の前提部に記述されているが十分であるかどうかを検討する。

- (1) 発話内容の当てはまったタイプに関する記述が不十分である場合は、不十分なハラメータをその発話内容の枠の中で書く。
- (2) 発話内容が当てはまるタイプの記述が十分であるが、前提部の他の記述が不十分な場合にはその不十分なところを質問する。
- (3) 上記の二つがともに存在しないときには結論部の側の関連性から、次に知りたいことを結論部に持つ制約を選択し、その前提部に書かれている状況の有無を尋ねる。

上記のように U S S R の特徴を生かした手法により関連性をもって話題を進めることが可能となる。

7. 他の状況指向型言語との比較

ここでは、U S S R と同じく状況理論に基づく言語である C I L [Mukai 85] との比較を試みる。

C I L は論理型言語 Prolog をベースとしたプログラミング言語であり、その特徴は以下の通りである。

- ・部分項の導入
- ・遅延実行制御の導入

このうち、遅延実行制御は「ある変数が具体化 (instantiate) された時点で、特定の手続きを実行するように指定できる機能」をもつプログラミング言語としての実行制御機構であり、U S S R への導入の可能性はあるものの意味表現という観点からは副次的なものである。一般に、U S S R は意味の表現に重点を置いているため、C I L に比べ手続き的な処理の記述については不足している。特に、制約の評価機構はナイーブであり、C I L のようにプログラミング言語的な制御構造を導入する必要がある。一方、意味表現の記述の枠組みに関して言えば、U S S R はタイプを本質的な要素として導入することによって、柔軟な概念定義、操作の機能を提供している。C I L の部分項はデータ構造として位置づけられるものであり、そのまま、意味の記述形式を与えるものではなく、多様なタイプシステムを提供するものではない。ただし、U S S R と C I L は、それぞれ、目指すところが違っており、単純な比較を行うことは適切ではなからう。U S S R のような状況理論に基づく意味表現言語を実現するための言語処理系として C I L を位置づけるのが適当であろう。

8. おわりに

自然言語理解、特に、自然言語による相談型対話を目的とするシステムにおける意味表現の果たすべき機能について検討を行い、状況理論をベースに意味表現言語 U S S R の設計とそのサブセットの実現を行った。

また、その応用例についても指針を示した。

文脈処理に関する研究が広く行われたしたのは確かだが、まだ、出発点に立っているというのが現状であろう。かすかす提唱されている文脈処理のためのモデルの中でも状況理論は有望なものといえる。ただし、状況理論自体は計算モデルは提示していないため、計算機上の実現のためには、数多くの検討事項が残されている。従って、U S S R は状況理論に基づいて設計されたものであるが、それにとらわれる必要はないと考えている。

契約書作成支援を対象とした自然言語インタフェースシステムでの試用を通じ、U S S R が 2 章で述べた意味表現に対する要請をかなり満足していることを確認できたと考えている。今後、残された課題を検討しつつ、文脈処理のための表現体系を整理して行きたい。

現在の U S S R の問題点としては、

(1) 表現上の問題

- ・個体の表現
関係中心の表現であるため、個体の表現が複雑になる場合がある。個体に対してもフレーム的な操作が可能となるようにすべきかもしれない。
- ・否定的知識の取扱
- ・限量子の取扱

(2) 実現上の問題

- ・制約の評価機構
推論機能の強化、記述力の強化が必要である。
- ・環境の持ちかた
中間表現とデータベース表現という 2 レベルの環境の持つ問題の解決。多世界的機構の導入によるモデル記述の枠組みの拡張。
- ・処理効率

などがあけられ、今後の課題となっている。

《謝辞》

本研究を行うにあたり、多くの方々にお教え頂いた。I C O T 第 2 研究室の方々、S S 輪講会のメンバーの方々との討論は大変参考になった。ここに感謝の意を表す。また、本研究の機会を与えて下さり、適切な指導・助言をいただいた山崎室長、小森田主担当に感謝する。

なお、本研究は第 5 世代コンピュータプロジェクトの一環として、(財)新世代コンピュータ技術開発機構からの委託により行ったものである。

《参考文献》

- [Barwise 83] J. Barwise and J. Perry :
Situations and Attitudes, MIT Press,
1983
- [Barwise 85] J. Barwise and R. Cooper : General-
ized Quantifiers in Situation
Semantics, draft, 1985
- [Brachman 83] R.J. Brachman, R.E. Fikes and
H.J. Levesque : KRYPTON: A func-
tional approach to knowledge
representation, IEEE Computer
16(10), 1983
- [Minsky 75] M. Minsky : A FRAMEWORK FOR
REPRESENTING KNOWLEDGE, in Winston,
P.H.(ed): "The Psychology of Com-
puter Vision", McGraw-Hill, 1975
(邦訳: 白井良明, 杉原厚吉訳: 「コン
ピュータービジョンの心理」, 産業図
書, 1979)
- [Mukai 85] K. Mukai and H. Yasukawa : Complex
Indeterminates in Prolog and its
Application to Discourse Models,
NEW GENERATION COMPUTING 3(4),
OHMSHA-Springer, 1985
- [野口 87] 野口直彦, 高橋雅則, 安川秀樹: 対話状況
を考慮した適切な応答文の生成, ロジック
プログラミングコンファレンス'87(投稿予
定), 1987
- [向井 86] 向井国昭: 状況意味論, in 淵一博監修:
「自然言語の基礎理論」, 第3章, 共立出
版, 1986
- [本池 86] 本池祥子, 鈴木浩之: 相談型会話における
話題の選択方式, 情報処理学会第33回全
国大会, 4L-1, 1986