自然言語処理　54－7
（1986. 3. 28）

タプル論理による日本語文の意味表現と、Syllogism（三段論法）に基く推論法

森田憲一・江村恵里子
（大阪大学　基礎工学部）

1. まえがき

自然言語文の意味を数学的に厳密に構成された形式論理の枠組みの中で表現しようとする研究の1つの目的は、意味理解や推論のような高度で知的な情報処理が、どこまで、単純で機械的な「規則」の組合せとして把握できるのだろうか、という問の解を求めることであろう。もし、ある限られた範囲内の自然言語文に対してであったにせよ、そのような把握が可能なことが示されたならば、文の論理構造を数学的に非常に透明な眼でもって認識できるようになるだけでなく、文の理解や推論を計算機によって実際に機械的に遂行する道をも開くことになる。

Motague[8] によるいわゆるモンタギュー文法（MG）は、かなり広い範囲の英文に対して論理学的な把握が可能であることを示した研究として非常に重要である。MGでは、英文の意味は、内包論理と呼ばれる高階述語論理の論理式によって表す。これにより、“every ”や“a ”などの単語による限量表現、副詞句や前置詞句による修飾的な意味構造、that節を伴う複文構造などはもちろんのこと、内包と外延の区別を必要とするような微妙な言語表現をも、高度の論理的明晰さでもって記述することができる。

MGの基礎となっている内包論理は、このように言語学的にも論理学的にも非常に興味深く、また、すぐれた体系なのであるが、唯一の欠点は、非常に複雑だということである。論理が高階である上に内包と外延を扱えるようになっているために、統語論が込み入っているだけでなく、モデル理論的意味論もたいそう複雑である。言語の意味表現のためにこれほど複雑な論理体系が必要不可欠なのか、という問に解答を与えるのは難しい。しかし、論理をこんなにまで複雑にしてしまったのは、数学理論の記述に絶大な威力を発揮した述語論理を、その威光の故に安直に自然言語の意味記述にまで流用したためだと見ることは、あながち誤りだとは言い切れない。

述語論理は、ちょうど算術式が関数と引数とから構成されているように、命題（文）を、述語（真理値関数）と引数とから構成されているとみなそう、という着想[2] に基いて構築された論理である。さらに、引数は個体（entity）を表すもの、述語は個体の性質あるいは個体間の関係を表す数学的な関数だとみなすことにより、非常に簡潔な意味論を与えることが可能になる。数学理論においては個体の概念は完全に抽象化された形式で現れる（例えば、数の概念を見よ）ので、このような形式化は非常にふさわしいものである。しかし、一般的な知識を述べた自然言語文においては個体の概念が明確には現れない場合が多い。もちろん、個人とか個物とかの概念は頻繁に現れるが、それらを、いわば数学的な「点」でしかない抽象的な個体の概念に対応させてよいかどうかは自明なことではない。その上、個物として実際には存在しないようなものについても、何らかの意味のある陳述をすることが可能である。例えば、

“龍は非存在物である”

というような文がそうである。この文の意味を、個体を引数とする2つの述語“龍”と“非存在物”を用いて、

∀ x（龍（x ）→　非存在物（x ））

のような式で表すわけにはいかない。MGなどではこれらの述語は、個体そのものではなく、個体の内包を引数にとる述語だとして解決をはかっている。しかし、このように内包と外延の区別を持ち出さざるを得なくなったそもそもの原因は、述語論理の意味論が個体集合を基礎としているからに他ならず、それは結局、文を述語と引数に分解するという立場から生じたものである。

さらに、述語論理には次のような問題点もある。自然言語文の意味表現に述語論理を採用した場合、例えば英語の自動詞は、個体（または個体の内包）を引数とする1項述語に翻訳されるのが普通である。しかし、もしそうしたならば、自動詞句に係る副詞は、自動詞句を引数にとり自動詞句をその値とするような高階の述語（関数）にならざるを得ず、数学的に非常に複雑な関数になってしまう。また、自然言語文中にはthat節や引用などによる「高階風」の概念もよく現れるが、これらを数学的な意味での高階述語によって表現した場合、それらの対応関係はすっきりする反面、やはり意味論の方がどうしても複雑になってしまう。

著者は、文献 7）においてこれらの問題を検討し、自然言語的な知識の表現に適していると思われる新しい論理体系「タプル論理」（TL）を提案した。TLは、述語と引数の概念をなくし、文のすべての要素を数学的に同質の対象物として見るような論理体系である。そこでは、種々の概念間の結びつき（論理構造）を、記号の「組」（つまりtuple ）によって表現する点が特徴である（具体的にはリスト構造によって表す）。このため、従来は高階と考えられてきた諸概念を扱うのに高階述語を用いずに済み、論理体系が不必要に複雑になるのを避けることができる。実際、後にも示すように、単語や句の係り方の構造を括弧構造によって表したタプルそのものをその文の意味表現とするこ

とができる。また、英語の副詞などに対する意味公準も容易に書き下せる[9]。さらに、TLは述語と引数の区別を持たないので、表現に個体を対応させずに意味論を与えることができ、非常に単純である[7]。TLの意味論は、ちょうど述語論理におけるHerbrand 解釈のように、タプル表現それ自体だけを問題としている。これにより、内包と外延の区別さえ払拭してしまえる可能性がある。

ところで、TLそれ自体は拘束の非常に緩い体系であるので、論理的推論を行うためにTLに付加されるべき構造も、多種多様なものの中から選ぶことができる。もちろん、通常の述語論理で用いられているのと同じ論理記号 ～（否定）、→（含意）、∀（全称限量詞）と、それに対する推論規則と公理を与えることも可能であり、文献 7) ではそれによる演繹の完全性と無矛盾性を証明している。しかし、TLはこういった論理記号でさえ他の一般の記号と区別せずに扱おうとする体系であるので、例えば限量詞的な意味を表す“every ”や“a ”などの英単語を特別な∀や∃などの論理記号に翻訳せずに、そっくり残したままで論理的推論を行なえるように推論公理を付加する方法も考えられる[9]（但し、推論公理自体の表現のために論理記号を用いた記述が必要となる）。

さて本研究では、TLの枠組みの中で日本語文の意味を表す方法のうちで特に、なるべく文の原型に近い形のタプルのままで推論が行えるような表現法を与えることを目的としている。ここでは「文タプル」と呼ぶ、ある規格化された形式のタプルによる表現法を提案した。文タプルは、統語構造を示す括弧構造が付いてはいるが、原文に非常に近いような表現形式であり、特別な論理記号は含まない。そして、このような文タプルに対して論理的推論を実行するための構造として、Aristotle[1] のSyllogism（三段論法）を基礎とする方法を与えた。Syllogismは、元来、いくつかの文（命題）から別の文が論理的に導けるかどうかを、各々の文それ自体の形相（form）に基いて決定する推論法であり、特別な意味をもつ論理記号を基礎とした推論法ではないので、本研究の目的によく合致している。

次章以下の構成は次の通りである。2．では本稿で扱う日本語文の範囲を示す。3．では、文タプルの定義と、日本語文からそれへの翻訳方法の概要を述べる。4．では、Syllogismの解説を与えた後、それに基いて文タプル上での演繹の概念を定義する。さらに、推論を効率よく実行するために、Lukasiewiczによる妥当性判定法[4]を基礎とした推論手続きを与える。但し、本来のSyllogismは、普通名詞に相当する名辞間の関係を述べた「名詞文」（後述）だけに関する推論法であり、固有名詞や名詞文以外の文も扱えるようにする必要があるので、この拡張法も示す。5．では、以上の方法に基いてProlog 上にインプリメントした試作の推論システムについて述べる。

2．日本語文の範囲と構文

本研究で対象とする日本語文の基本カテゴリー（品詞）と構文規則を以下に示す。

2．1．基本カテゴリー

- 動詞　　　　：イク，…
- 形容詞　　　：カワイイ，…
- 形容動詞　　：シズカ，…
- 普通名詞　　：ショウネン，…
- 固有名詞　　：ハナコ，…
- 不定称代名詞：ナニ，ダレ，ドコ
- 限量詞　　　：アル，スベテノ
- 格助詞　　　：ガ，ヲ，ニ，カラ，ヘ
- 係助詞　　　：ハ
- 断定辞１　　：デアル，デス
- 断定辞２　　：ダ
- 疑問辞　　　：カ

〖注意 2.1.1〗　ここでいう形容動詞は、学校文法などでいう形容動詞の語幹である。

〖注意 2.1.2〗　ここでは連体詞「アル」と、名詞「スベテ」と格助詞「ノ」からなる句「スベテノ」を、それらの論理的な役割から、限量詞とした。

2．2．構文規則

本研究では、日本語文の範囲を以下に示す構文規則から生成される単文に限定する。また、否定の概念は扱わない。

日本語文の構造は、述語の前に種々の格の補語がいくつかついていると見る立場をとっている[3,5]。なお、ここでの「述語」は構文上の概念であり、述語論理における述語（真理値関数）とは区別すべきものであることに注意。

構文規則は文脈自由文法の形式で記述する。以下の規則では、“→”は書換えの矢印、“｜”は、書換えの選択枝を表す。

〈文〉	→〈平叙文〉｜〈単純疑問文〉｜〈不定詞疑問文〉
〈平叙文〉	→〈述節〉
〈述節〉	→〈述語〉｜〈補語〉〈述節〉
〈述語〉	→〈動詞〉｜〈形容詞〉｜〈形容動詞〉〈断定辞〉｜〈普通名詞句〉〈断定辞〉
〈補語〉	→〈単称名詞句〉〈格助詞〉｜〈単称名詞句〉〈係助詞〉｜〈普通名詞句〉〈格助詞〉｜〈普通名詞句〉〈係助詞〉
〈単称名詞句〉	→〈限量詞〉〈普通名詞句〉｜〈固有名詞〉
〈普通名詞句〉	→〈普通名詞〉｜〈形容詞〉〈普通名詞句〉
〈断定辞〉	→〈断定辞１〉｜〈断定辞２〉
〈単純疑問文〉	→〈疑問述節〉

〈疑問述節〉　→〈疑問述語〉|〈補語〉〈疑問述節〉
〈疑問述語〉　→〈動詞〉〈疑問辞〉|
　　　　　　　〈形容詞〉〈疑問辞〉|
　　　　　　　〈形容動詞〉〈疑問辞〉|
　　　　　　　〈形容動詞〉〈断定辞１〉〈疑問辞〉|
　　　　　　　〈普通名詞句〉〈疑問辞〉|
　　　　　　　〈普通名詞句〉〈断定辞１〉〈疑問辞〉
〈不定詞疑問文〉→〈不定詞疑問述節〉
〈不定詞疑問述節〉→〈不定詞補語〉〈疑問述節〉|
　　　　　　　　〈補語〉〈不定詞疑問述節〉
〈不定詞補語〉→〈不定称代名詞〉〈格助詞〉

〖注意 2.2.1〗　「ハイ」または「イイエ」で答えられる疑問文を単純疑問文、不定称代名詞を伴う疑問文を不定詞疑問文と呼ぶことにした。

３．タプル論理に基く意味表現

ここでは、前章で示した範囲の日本語文の意味をＴＬの枠組みの中で適切に表現するための方法を提示する。もちろん、表現法というのは推論法と組合されて初めて意味を持つ（意味論を与え得る）ものであり、ここでの表現法は、次章に示す推論法を暗に想定して定めている。

３．１．タプル論理（ＴＬ）

ＴＬに関する定義のうちで後で必要となるものを述べておく。なお、後で推論システムをProlog 上にインプリメントする関係で、タプルを書き表すためのリスト表現などはProlog の記法にならった。

（１）ＴＬで使う文字
　(i) 基本記号：　Ａ，…，Ｚ，a，…，z，0，…，9，ア，…，ン，…
　(ii) 論理記号：　～，→，∀
　(iii) 括弧とコンマ：　[，]，“，”

（２）定数と変数およびアトム
　(i) 定数：　英小文字またはカナで始まる基本記号列。
　(ii) 変数：　英大文字で始まる基本記号列。
　(iii) アトム：定数または変数。

（３）基本タプル
　(i) ωがアトムなら、ωは基本タプルである。
　(ii) ω_1，ω_2，…，ω_n　が基本タプルなら、
　　$[\omega_1, \omega_2, \dots, \omega_n]$ は基本タプルである。

（４）論理タプル
　(i) α_1，α_2，…，α_n　が基本タプルなら、
　　$[\alpha_1, \alpha_2, \dots, \alpha_n]$ は論理タプルである。
　(ii) α，β　が論理タプル、νが変数なら、
　　$[\sim, \alpha]$，$[\alpha, \rightarrow, \beta]$，$[\forall, \nu, \alpha]$
　　は論理タプルである。（但し、上のような表現は見づらいので、以下では便宜上、　$\sim\alpha$，　$\alpha\rightarrow\beta$，　$\forall\nu[\alpha]$　などの記法を許す。）

（５）論理記号に関する公理と推論規則
　論理記号～，→，∀に関する公理と推論規則は、一階述語計算におけるのと相似なものを与える[7]。

３．２．文タプルによる日本語文の意味表現

本研究では、日本語文の意味を表すのに文タプルと呼ぶ表現形式を用いる。これは、文を構成するいくつかの補語と述語とを同列に並べた形式の表現である。論理記号を用いないので元の文に非常に近い。以下にその定義を与える。

《文タプル》：：＝
　[《文の型》，[《補語》，…，《補語》]，《述語》]
《文の型》：：＝　d | k | kd | m
《補語》　：：＝[《限量詞》，《名詞句》，《格助詞》]
《限量詞》：：＝　アル|スベテノ|　p |－
《名詞句》：：＝　《固有名詞》|－|
　　　[[《形容詞》，…，《形容詞》]，《普通名詞》]
《述語》　：：＝《動詞》|《形容詞》|《形容動詞》|
　　　　　《名詞》

文タプルは、文の型，補語のリスト，述語の３つの要素からなるリストである。述語の種類としては、動詞，形容詞，形容動詞，名詞があり、それらを述語とする文をそれぞれ、動詞文，形容詞文，形容動詞文，名詞文と呼ぶ。文の型は、それらを d，k，kd，m で表したものである。補語のリストはその述語がとり得る格の補語をならべたものである。動詞文の場合、とり得る格は各々の動詞ごとに定まっている（文献 6）の付録を参考にした）。また、ここでは動詞文以外は、主格（ガ格）だけをとり得るとした。補語は、限量詞，名詞句，格助詞の３つの要素からなるリストである。限量詞は、「アル」，「スベテノ」以外に、名詞句が固有名詞である場合に“p”を用いる。名詞句には、固有名詞と普通名詞句がある。普通名詞句は、形容詞リストと普通名詞の２つからなるリストで表現される。

日本語文から文タプルへの具体的な翻訳方法については、以下の例によって示すにとどめる。

例日本語文：ハナコ　ハ　カワイイ　ショウジョ　デス。
　文タプル：[m，[[p，ハナコ，ガ]]，[[カワイイ]，ショウジョ]]

例日本語文：ショウネン　ハ　ワカイ　オトコ　デス。
　文タプル：[m，[[スベテノ，[[]，ショウネン]，ガ]]，[[ワカイ]，オトコ]]

例日本語文：オンナ　ガ　ハタラク。
　文タプル：[d，[[アル，[[]，オンナ]，ガ]]，ハタラク]

例日本語文：タロウ　ガ　ガッコウ　へ　イク。
　文タプル：[d，[[p，タロウ，ガ]，[－，－，カラ]，[アル，[[]，ガッコウ]，へ]]，イク]

【注意 3.2.1】 文中の述語がとり得る格の補語で、省略されているものは、［－，－，《格助詞》］で表す。

【注意 3.2.2】 原文の補語の名詞句が限量詞を伴わない普通名詞句の場合、その補語の助詞が係助詞のときには「スベテノ」を、それ以外は「アル」を補ったタプルに翻訳する。またここでは、係助詞「ハ」の働きについては、格助詞「ガ」の代用機能だけを扱った。

【注意 3.2.3】 形容動詞文，名詞文の述語は、断定辞を省略したもので表す。

4．Syllogismに基く推論法

ここでは、文タプルの形式に翻訳された日本語文に対してSyllogism（συλλογισμός）を適用する方法を考察し、それに基いた推論手続きを与える。Syllogismは、日本では「三段論法」という呼び名で広く知られているが、必ずしも正しく理解されていない場合が多いので、ここでは後で必要となることがらの定義と解説を４．１節に、またLukasiewiczによる形式化と妥当性判定手続きを４．２節に示した後、４．３節以下でそれを文タプルに対する推論法として用いる方法を述べる。これにより、従来の述語論理では限量記号∀、∃によって表現し、それらの記号に関する公理を用いて推論しなければならなかったような自然言語の限量表現が、非常に単純な枠組みの中で扱えるようになる。

4．1．AristotleのSyllogism

Aristotle[1]は、例えばa とb というような２つの名辞（ὅρος，term，普通名詞に相当するような単語または句）がそれぞれ主語と述語を構成しているような文を、命題（πρότασις，premiss，proposition）と呼び、それを次の４種類に分類した。（但し、ここでの「主語」は、ヨーロッパ語に対する用語であり、第２章の日本語文法では「主格補語」に対応する。）

全称肯定命題（Ａ） … すべてのa は bである
特称肯定命題（Ｉ） … あるa は bである
全称否定命題（Ｅ） … すべてのa は bでない
特称否定命題（Ｏ） … あるa は bでない

（Ａ，Ｉ，Ｅ，Ｏは、中世の論理学者が用いた略称）

ここではLukasiewicz[4]の記法に従って、名辞a とb を主語と述語に持つようなそれぞれの命題を Aab，Iab，Eab，Oab と書く。

Aristotleは、ある名辞a（大名辞と呼ぶ）が別の名辞c（小名辞）の述語となっているような命題（結論）が、第３の名辞b（中名辞）とa の関係を述べた命題（大前提）と、b とc の関係を述べた命題（小前提）とから、どのような場合に正しく導かれるのかを詳細に分析した。

大前提と小前提における各名辞の出現順序（主語、述語の順序）の組合せとして、次の４通りが考えられる。

	大前提	小前提	結 論
第１格	（b，a）	（c，b）	（c，a）
第２格	（a，b）	（c，b）	（c，a）
第３格	（b，a）	（b，c）	（c，a）
第４格	（a，b）	（b，c）	（c，a）

また、大前提、小前提、結論はそれぞれ、Ａ，Ｉ，Ｅ，Ｏの型の命題になり得るので、総計 $4 \times 4^3 = 256$ 通りの推論の形式（推論のムード）がある。Aristotleは、これらのうちで24通りのムードだけが妥当なものであることを示した（但し、実際にはこれとは多少違った述べ方をしている）。例えば、大前提 Aba と小前提 Acb とから結論 Aca を導く推論は正しい推論法である。これはＡＡＡの第１格であり、AAA_1 と書くことにする。

妥当な24個の推論のムードのうちで肯定命題だけからなるものは次の８個である（括弧内は、中世の論理学者がつけた、暗記用の呼び名）。

AAA_1（Barbara）	AII_1（Darii）
AAI_1（Barbari）	AAI_3（Darapti）
IAI_3（Disamis）	AII_3（Datisi）
IAI_4（Dimaris）	AAI_4（Bramantip）

4．2．Lukasiewiczによる形式化と妥当性判定手続き

Lukasiewicz[4]は、AristotleのSyllogismを記号論理の立場から公理的体系として形式化した。また、それに基いて、任意に与えられた推論形式が妥当か否かを決めるための判定手続きを与えた。以下にその概要を述べる。

x とy を任意の名辞変数とするとき、命題 Axy，Ixy，Exy，Oxy のいずれか１つだけからなる表現を、単純式（simple expression）と呼ぶ。但し、Ｅ命題とＯ命題は、Exy≡～Ixy，Oxy≡～Axy で定義されるものとする。単純式を元に、論理記号 ～ と → を用いて（命題論理におけるのと同様に）構成される表現を、意味のある式（significant expression）あるいは、単に式と呼ぶ（論理記号∨，∧，↔ は、～と→から定義できる）。

Lukasiewiczは、次の４つの式と、通常の命題計算の公理を、Syllogismのための公理として置いた。

(i) Aaa
(ii) Iaa
(iii) （Aba ∧ Acb）→ Aca
(iv) （Aba ∧ Ibc）→ Ica

公理(i)と(ii)は恒等法則（law of identity）と呼ばれる。また、(iii)はAAA_1，(iv)はAII_3である。

推論規則は次の２つである。

(a) 名辞変数への他の名辞変数の代入規則
(b) 分離法則（modus ponens）

これらの公理と推論規則から、AristotleのSyllogismにおける妥当な24の推論のムードだけでなく、すべての妥当な式が定理として導かれる。

【注意 4.2.1】 なお、以下では便宜上、2つの前提から1つの結論を導くような推論形式のことを、AristotleのSyllogismと呼び、Lukasiewiczによる、より一般的な扱いを単にSyllogismと呼ぶことにする。

Lukasiewiczはさらに、任意に与えられた式が妥当であるか否かを決定する問題についても考察し、そのための判定手続きを与えた。この手続きにより、式によって与えられた全く任意の形の推論形式に対する妥当性が判定できる。準備として、次の定理を述べておく。

【定理 4.2.1】[4] 任意の式は、基本式(elementary expression)と呼ぶ次の形の式の論理積として表現できる。

$$(\alpha_1 \rightarrow (\alpha_2 \rightarrow (\cdots (\alpha_{n-1} \rightarrow \alpha_n) \cdots)))$$

但し、各α_iは単純式である。

【注意 4.2.2】 上の基本式は、次式と等価である。

$$((\alpha_1 \wedge \alpha_2 \wedge \cdots \wedge \alpha_{n-1}) \rightarrow \alpha_n))$$

なお、基本式中のα_1, …, α_{n-1}を前件(antecedents)、α_nを後件(consequent)と呼ぶ。

この定理から、任意の基本式に対して妥当性判定ができれば、任意の式に対してもできることがわかる。基本式に対するLukasiewiczの判定手続きを要約すると次のようになる(詳しくは文献 4) pp.120-126参照)。

Lukasiewiczの判定手続き:

① 否定命題(ExyまたはOxy)を含むような基本式の妥当性判定は、否定命題を含まない基本式の妥当性判定に帰着できる(詳細は省略)。

② 肯定命題だけからなる基本式の妥当性を判定するには、前件から後件が導けるかどうかを次のようにして調べればよい。

(a) 後件α_nがA命題の場合:

(i) α_nがAaaの形ならばその基本式は妥当。

(ii) α_n=Aabと書けるとき($a \neq b$):
その基本式が妥当であるのは、前件の中から適当なものを選び出すことにより、次のような連鎖がつくれるとき、かつそのときに限る(図1(a))。

$$\mathrm{A}\ a\ c_1,\ \mathrm{A}\ c_1\ c_2,\ \cdots,\ \mathrm{A}\ c_j\ b \qquad (j = 0,1, \cdots) \qquad \text{(s1)}$$

但し、j=0 というのは連鎖の長さが 1、従って前件の中にAab自体が存在する場合を意味する。

(b) 後件α_nがI命題の場合:

(i) α_nがIaaの形ならばその基本式は妥当。

(ii) α_n=Iabと書けるとき($a \neq b$):
その基本式が妥当であるのは、前件の中から適当なものを選び出すことにより、次の2つの連鎖

$$\mathrm{A}\ c_j\ c_{j-1},\ \cdots\ \mathrm{A}\ c_2\ c_1,\ \mathrm{A}\ c_1\ a \qquad \text{(s2)}$$

$$\mathrm{A}\ d_k\ d_{k-1},\ \cdots\ \mathrm{A}\ d_2\ d_1,\ \mathrm{A}\ d_1\ b \qquad \text{(s3)}$$

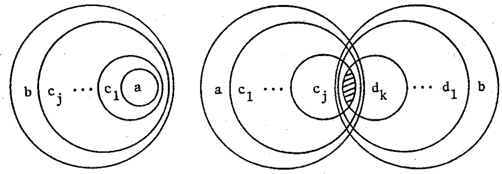
$$(j = 0,1, \cdots, k = 0,1, \cdots)$$

が作れ、かつ前件の中にそれらを中継する命題

$$\mathrm{I}\ c_j\ d_k,\quad \mathrm{I}\ d_k\ c_j,$$

$$\mathrm{A}\ c_j\ d_k,\quad \mathrm{A}\ d_k\ c_j$$

のいずれかが存在するとき、かつそのときに限る(図1(b))。但し、j=0(あるいは k=0)というのは、対応する連鎖の長さが 0、それゆえ c_j = a(d_k =b)であることを意味する。



(a) A命題の場合　　(b) I命題の場合

図1.後件を導くための連鎖のEuler図

4.3.Syllogismに基く、文タプル上での演繹

第3章では、日本語文の意味を文タプルで表す方法を示したが、ここではSyllogismを基礎として、文タプルに関する「演繹」の概念、つまり、ある1つのタプルが他の(いくつかの)タプルから「導かれる」ということの定義を与える。この節ではまず、主格補語が普通名詞句から構成されているような名詞文(これをタイプ1の名詞文と呼び、主格補語が固有名詞であるようなものをタイプ2の名詞文と呼ぶ)の扱いだけを考察する。それ以外の文の扱いについては4.5節以下で述べる。タイプ1の名詞文は、Syllogismで言うところの「命題」(特にA命題またはI命題)そのものであると考えられる(あるいは、ここではそのように見做す)ので、それらの翻訳である文タプルに対して演繹の概念を定義するのは容易である。つまり、4.2節のLukasiewiczの公理と推論規則をタプル表現に直したものを、TLにおける公理と推論規則として与えればよい。具体的には次のようになる。

(i) [m, [[スベテノ, A, ガ]], A]

(ii) [m, [[アル, A, ガ]], A]

(iii) ([m, [[スベテノ, B, ガ]], A]
　∧[m, [[スベテノ, C, ガ]], B])
　　→[m, [[スベテノ, C, ガ]], A]

(iv) ([m, [[スベテノ, B, ガ]], A]
　∧[m, [[アル, B, ガ]], C])
　　→[m, [[アル, C, ガ]], A]

なお、名辞変数への代入規則に対応する規則は不要である。なぜなら、上記公理中のA, B, CはTLの変数であるため、任意の普通名詞句に相当するタプルが代入できるからである(但し厳密には、普通名詞句に相当するタプルだけが代入できるようにするために、普通名詞句であることを示す目印をタプルに持たせるなどの工夫が必要である)。
また、命題計算の公理と推論規則は既にTLに含まれている(第3章参照)。

さて、演繹の概念を「定義」するだけなら、以上で一件落着なのだが、ある文タプルが別のいくつかの文タプルから導かれるか否かを実際に判定しようという目的から言えば、これは適切な方法ではない。上の公理は、述語論理でいうホーン節と同様の形になっているので、例えば、それらをそのまま埋め込んだようなProlog プログラムによって推論が実行できそうに見えるが、実際にはほとんどの場合にプログラムが停止せず、このままでは使い物にならない。この欠点は、Lukasiewiczの妥当性判定法を推論手続きとして与えてやることにことにより、うまく解消できる。

タイプ1の名詞文の翻訳である文タプルは、Syllogismにおける命題、つまり単純式に対応しているので、ある文タプルが別のいくつかの文タプルから導けるか否かを判定するには、基本式の妥当性判定だけができればよい。基本式に対するLukasiewiczの判定法は、Prolog プログラムとして非常に直接的かつ簡潔に書き下せる。また、結果として得られるプログラムは（実質上）導出原理における単位入力導出（unit input resolution）になっており、効率が良い。具体的なインプリメントは次節に示す。

4．4．Lukasiewicz法のインプリメンテーション

ここでは、4．2節の妥当性判定手続きをProlog 上にインプリメントする方法を述べる。本稿で扱っている日本語文は肯定文だけであるので、ここでは肯定命題しか含まないような基本式の妥当性判定だけを考える。

プログラムの説明のために、Syllogismにおいて証明されるいくつかの基本法則（但し、肯定命題に関するものだけ）を書いておく[4]。

(i) 大小対当の法則（law of subordination）
　Aab → Ｉab
(ii) 換位法則（law of conversion ）
　Aab → Ｉba
　Ｉab → Ｉba

さて、後件　Aab　または　Ｉab　が前件から導かれるか否かを判定するためのProlog の述語　a（A，B）とｉ（A，B）は、次のように定義される（“，”は論理積、“；”は論理和を示す）。

```
a（A，B）　←　a0（A，B）；a_（A，B）.
a_（A，B）←　a1（A，B）.
a_（A，B）←　a1（A，C），a_（C，B）.
a0　（A，A）.
a1　（A，B）←　ak（A，B）.
i（A，B）　←　i0（A，B）；i_（A，B）.
i_（A，B）←
        a（C，A），i1（C，D），a（D，B）.
i0　（A，A）.
i1　（A，B）←　ik（A，B）；ik（B，A）；
              ak（A，B）；ak（B，A）.
```

上で定義されたA命題に関する各々のProlog 述語の意味は、次の通りである。

　a（A，B）　…　命題Aabが前件から導かれる。
　a_（A，B）…　命題Aabが、恒等法則を用いずに前件から導かれる。
　a0　（A，B）…　命題Aabが恒等法則から導かれる。
　a1　（A，B）…　命題Aabが前件中にある。
　ak　（A，B）…　命題Aabは前件である。

Ｉ命題に関する各述語の意味もこれらと同様であるが、i1（A，B）だけが多少異なる。

　i1　（A，B）…　命題Ｉabが前件中にあるか、あるいは、ある1つの前件から大小対当または換位法則を使って導かれる。

以上のプログラムに加えて、すべての前件を宣言節

　ak（ u_i ,v_i ）.　または　ik（ u_i ,v_i ）.

として与えておき、後件が成立つか否かを問合せ節

　？－ a（a, b）.　または　？－ i（a, b）.

によって調べればよい。但し、第5章の推論システムでは、入力された平叙文の文タプルは、Prolog 述語“k ”が付けられ、宣言節の形で知識として登録されるようになっているので、“ak”と“ik”は次のように定義される。

ak（A，B）← k（［m，［［スベテノ，A，ガ］］，B］）.
ik（A，B）← k（［m，［［アル，A，ガ］］，B］）.

a（A，B）とｉ（A，B）が、4．2節の②における連鎖の存在をバックトラックによって調べるProlog 述語になっていることは、上のプログラムから明らかであり、「循環的な連鎖」が存在しない場合には、これによって基本式の妥当性が正しく判定でき、必ず停止する。循環的な連鎖とは、例えば前件の中にAbcとAcbがあった場合に、これらを交互に並べてできる無限の連鎖のことで、これによってプログラムがループする可能性がある。ループを防ぐための改善はそう困難ではないが、この問題は下の〖注意 4.4.2〗に述べる効率化の問題と合せて考えるべきだと思われるので、本稿では扱わなかった。

〖注意 4.4.1〗　Lukasiewiczは、否定命題を含む基本式の妥当性判定問題を、肯定命題だけからなる基本式の妥当性判定問題に還元する方法をとった（4．2節の①参照）が、これをそのままインプリメントしたのでは大変効率が悪い。実際には否定命題を扱うもっと良い方法があるのだが、これについては稿を改めて述べることにしたい。

〖注意 4.4.2〗　Lukasiewiczの判定手続きの実行効率をさらに上げようとするならば、4．2節の②における連鎖の探索問題を、有向グラフ上での道の探索問題に定式化し直し、グラフに関して知られている効率的アルゴリズムを応用すべきであろう。実際、各々の名辞を頂点に、各々の前件を（AまたはＩのラベルのついた）有向辺に対応づければ、連鎖の探索問題を2頂点間の道の探索問題に置換

えられる。その際には、Prolog ではなく、複雑なデータ構造を柔軟に扱えるようなプログラミング言語を用いるべきかも知れない。このようにすれば、先に述べた循環的な連鎖の問題も自動的に解決される。

4．5．タイプ2の名詞文への拡張

AristotleのSyllogismにおいて扱われている名辞は、基本的には、普通名詞等に相当するようなものだけであり、固有名詞等は除外されている。本稿では、固有名詞が述語の位置に来ている文は除外したが、以下では固有名詞が主語の位置に来ているような文に対する拡張法を与える。

主語の名辞が固有名 p、述語の名辞が普通名b である肯定命題「p は bである」をP命題と呼び、Ppbで表す。

Aristotle流のSyllogism（2つの前提から1つの結論を導く）をP命題にまで拡張した場合、固有名は（それが述語の位置には来ないという制約から）第1格と第2格の小名辞か、第3格の中名辞としてしか現れ得ない（4．1節参照）。そのような推論ムードのうち、肯定的なムードで表面上可能な形式は、APP_1，APP_2，PAA_3，IPP_1，IPP_2，PPI_3 の6つだけである。これらの中で直観的に妥当であると考えられるものは、APP_1，PPI_3 の2つである。ここでは4．2節のLukasiewiczの体系（“L”と呼ぶ）に、この2つを公理として付け加えた体系“LP”を定義し、それに基いてP命題に関する推論法を与えることにする。

【定義 4.5.1】 LukasiewiczのSyllogismの体系Lを次のように拡張してできる体系をLPと呼ぶ。

(a) 固有名変数が使えるようにする。

(b) 固有名変数に対する代入規則を付加する。

(c) P命題に関する次の公理を付加する。

(v) (Aba ∧ Ppb) → Ppa

(vi) (Ppa ∧ Ppc) → Ica

さて、αを肯定命題だけからなる任意のLPの式とする。α中のすべての固有名変数を、その式には現れていない新しい名辞変数に置換え（但し同一の（あるいは異なる）固有名変数は同一の（異なる）名辞変数に置換える）、さらにそれに伴って、P命題をすべてA命題に変えてできる式をαに対応するLの式と呼ぶ。

【定理 4.5.1】 αを肯定命題だけからなる任意のLPの基本式、βをαに対応するLの基本式とするとき、

αがLPの定理 ⟺ βがLの定理

が成立つ。

〖証明〗 公理(v)と(vi)に対応するLの式は、公理(iii)と、AAI_3（これはLの定理）である。αがLPで証明できたとする。αの証明中で、公理(v)が使われている箇所を公理(iii)に、公理(vi)をAAI_3 の証明に、それぞれ置換えて得られる証明は、明らかにLにおけるβの証明になっている。

逆にβがLで証明できたとする。そのときには、β中の前件を適当に並べることにより、4．2節のLukasiewiczの判定手続きにあるような連鎖がつくれるはずである。これらの連鎖中には、もともとαにおいて固有名変数であったような名辞変数が現れている場合がある。

(a) 連鎖（s1）では、名辞変数a だけがα中の固有名変数（p とする）に対応し得る。その場合、αの後件はPpbで、これは（s1）中のAac_1 をPpc_1 に置換えた連鎖を用いてLP中で導くことができる（APP_1 とAAA_1 を繰返し適用すればよい）。これを元に、命題計算の公理と推論規則を適当に使用すれば、αをLP中で証明できる。

(b) 連鎖（s2）と（s3）においては、c_j ,d_k のいずれか一方がそのような変数であり得る。c_j がそうである場合（d_k も同様）、αの後件Iabは（s2）の Ac_jc_{j-1} をPpc_{j-1}に変えた連鎖と（s3）と中継命題Ppd_k とを用いてLP中で導ける（PPI_3，AAA_1，AII_3 を適用する）。この場合もこれに基いてαを証明できる。 ■

この定理は、LPにおけるαの証明がいわばLの中で模倣できることを示している。特に、Lukasiewicz法を少し修正するだけで、LPにおけるαの妥当性判定ができることがわかる。これに基いて前節のインプリメンテーションを拡張するには、次のプログラムを付け加えればよい。

```
p(P, B)  ← p1(P, C), a(C, B).
p1 (P, B) ← pk(P, B).
i1 (A, B) ← pk(P, A), pk(P, B).
```

4．6．名詞文以外の文への拡張

ここでは、述語が、動詞、形容詞、形容動詞であるような文（便宜上「用言文」と呼ぶ）への拡張を示す。まず、主格補語だけを補語にとるような用言（形容詞、形容動詞の大部分と少数の動詞）が述語になっている文を考える。

名辞変数a を主格補語に、用言変数y を述語に持つような肯定的用言文は、その補語が全称的、特称的、固有名、のどれであるかに従って、A′，I′，P′ 命題と呼び、A′ay，I′ay，P′ay で表す。前節のP命題の場合と同様、用言変数が述語の位置にしか来ないという制約から、′付き命題を含み得る推論の格は第1，2，3格だけとなる。これらの格のすべての推論ムードのうちで妥当と考えられるのは次の8個である（これらはすべてのムードを調べることによって得られるものであるが、実際上は、′なしの命題と相似なムードだけが妥当と考えられる）。

$A'AA'_1$，$A'II'_1$，$A'AI'_1$，
$A'AI'_3$，$I'AI'_3$，$A'II'_3$，
$A'PP'_1$，$P'PI'_3$

【定義 4.6.1】 体系LPに、用言変数の概念と、それに関する代入規則、および上記の8つに対応する式を公理として付け加えた体系をLP′と呼ぶ（但し、8つの式の中には独立でないものもある）。

また、αを肯定命題だけからなるLP′の式とするとき、

前節と同様の方法で、<u>αに対応するLの式</u>が定義できる。

【定理 4.6.1】αを肯定命題だけからなる任意のLP′の基本式、βをαに対応するLの基本式とするとき、

αがLP′の定理 $\Longleftrightarrow$ βがLの定理

が成立つ。　(証明略)

この定理は、定理 4.5.1と同様の方法で証明できるので、LP′における基本式の妥当性判定も、Lukasiewicz法を少し修正するだけでできる。実際、各連鎖中で ′ 付きになり得る命題は、連鎖の終端 ((s1) のA c_j b と (s3) のA d_1 b) だけなので、判定法の修正も容易である。

最後に、主格補語以外にいくつかの補語をとるような用言の扱いを述べておく。ここでは次の2つの仮定を置き、それに従って推論手続きをインプリメントした。

① 主格以外の (ヲ格、ニ格などの) 補語に対しても、主格補語と同じ推論手続きが適用できる。
② 多くの補語を持つ用言文に関する推論は、各々の補語についての推論の帰着できる。

5. 推論システム

本研究では、前章までに示した方法に基いて、簡単な日本語質問応答システムをProlog 上に試作した。これは、日本語の平叙文が与えられたときにはそれを文タプルに翻訳して知識ベースに蓄え、疑問文のときには知識ベース中の文タプルを元に推論を行って答を出すようなシステムである。このシステムは次の5つの部分から成る。

(i) 構文解析と翻訳の部門
(ii) 辞書部門
(iii) 入出力と推論の制御の部門
(iv) 推論規則部門

推論規則部門は、4．4～4．6節に述べたSyllogismに関する推論規則を含むと共に、連体修飾語として用いられる形容詞に関する推論規則も含んでいる (後者については稿を改めて論じる予定である) 。

以下に実行例を示す。各々の入力文の次の行は文タプルへの翻訳、その次の行はシステムの応答である。

```
ｼｮｳﾈﾝ ﾊ ﾜｶｲ ｵﾄｺ ﾃﾞｽ｡
   k([m,[[ｽﾍﾞﾃﾉ,[[],ｼｮｳﾈﾝ],ｶﾞ]],[[ﾜｶｲ],ｵﾄｺ]]])
ﾜｶﾘﾏｼﾀ｡

ﾀﾛｳ ﾊ ｼｮｳﾈﾝ ﾃﾞｽ｡
   k([m,[[p,ﾀﾛｳ,ｶﾞ]],[[],ｼｮｳﾈﾝ]]])
ﾜｶﾘﾏｼﾀ｡

ﾀﾛｳ ﾊ ｶﾞｸｾｲ ﾃﾞｽ｡
   k([m,[[p,ﾀﾛｳ,ｶﾞ]],[[],ｶﾞｸｾｲ]]])
ﾜｶﾘﾏｼﾀ｡

ｶﾞｸｾｲ ﾊ ｶｼｺｲ｡
   k([k,[[ｽﾍﾞﾃﾉ,[[],ｶﾞｸｾｲ],ｶﾞ]],ｶｼｺｲ]])
ﾜｶﾘﾏｼﾀ｡


ｱﾙ ｵﾄｺ ﾊ ｶｼｺｲ ｶ｡
   q0(B_89):-q([k,[[ｱﾙ,[[],ｵﾄｺ],ｶﾞ]],ｶｼｺｲ]])
ﾊｲ !

ﾀﾞﾚ ｶﾞ ｵﾄｺ ﾃﾞｽ ｶ｡
   q0([H_91,G_91]):-
             q([m,[[H_91,G_91,ｶﾞ]],[[],ｵﾄｺ]]])
ｽﾍﾞﾃﾉ ｵﾄｺ , ｽﾍﾞﾃﾉ ｼｮｳﾈﾝ , ｱﾙ ｵﾄｺ , ｱﾙ ｼｮｳﾈﾝ ,
ｱﾙ ﾜｶｲ ｵﾄｺ , ｱﾙ ｶﾞｸｾｲ , ﾀﾛｳ ,  !
```

6. むすび

Lukasiewicz[4] は、AristotleのSyllogismはformal (形相的) な論理体系であるが formalistic (形式的) ではないと述べている。実際、AristotleによるA，I，E，Oの分類や推論法は、各命題の「内容的な意味形式」に対するものであって、命題を具体的に表現している「記号の表面的な配列形式」に対するものではない。Lukasiewiczの業績は、推論それ自体については厳密に形式的な、つまり記号操作的な扱いが可能であることを示した点にある。本研究は、タプル論理を用いて、文の分析や分類に相当するような部分をも形式的に扱う、つまりSyllogismを、記号列としての文そのものに対して形式的に適用する方法を探る研究だと見ることが可能である。もちろん、本稿で扱ったのはかなり限定された範囲の日本語文だけであり、また、古典的なSyllogismだけでは覆いきれない部分もかなりあると考えられるが、より包括的な扱いが可能になるようにこれらの点を徐々に改善していきながら、この方法の可能性を探るのが今後の課題である。

参考文献

1) Aristotle: Prior and Posterior Analytics, p. 266, J. M. Dent & Sons , London (1964) .
2) Frege, G. : Begriffsschrift — eine der arithmetischen nachgebildete Formelsprache des reinen Denkens, Halle (1879) .
3) 国立国語研究所：日本語の文法 (上) , p.107 (1978) .
4) Lukasiewicz, J. : Aristotle´s Syllogistic, p.141 , Oxford , London (1951) .
5) 三上章：現代語法序説, p.410 , くろしお出版, 東京 (1953, 復刊1972) .
6) 水谷・石綿・荻野・賀来・草薙・青山：文法と意味Ⅰ, p.294 , 朝倉書店, 東京 (1983) .
7) 森田憲一："Tuple Logic", 電子通信学会技術研究報告 AL 84-43 (1984) .
8) Montague , R. : The proper treatment of quantification in ordinary Engrish, in Formal Philosophy , ed. R. H. Thomason , pp.247-270, Yale Univ. Press (1974) .
9) 中山聡・森田憲一：述語を持たない論理体系 "Tuple Logic" とその知識表現への応用, 情報処理学会「知識情報処理」シンポジウム (1985) .

<u>謝辞</u>　本研究は、一部、文部省科研費補助金による。