数理解析研究所講究録
第556巻 1985年 294-305

知識の表現のための「述語」を持たない述語(?)論理"Tuple Logic"の提案

森田 憲一
Kenichi MORITA
(大阪大学 基礎工学部)

あらまし : 現在の標準的な論理体系として用いられている「述語論理」は、自然言語の文を「関数」(今でいう「述語」)と「引数」に分解するという着想を元にFregeが提案したものである。述語論理は、数学理論の記述のための体系としては大変すぐれたものであるが、自然言語の意味を詳細に表現するための体系としては適切でない面を持っている。本稿では、一階や高階の述語論理が、自然言語で述べられたような「知識」を表現するための体系としてなぜ不適切であるかを議論し、それに基いて、新しい論理体系"Tuple Logic"(TL)を提案した。TLにおいては、「関数」という考え方を排し、すべての知識(つまり、概念間の結びつき)を項目の『組』(tuple)によって表現する。これにより、自然言語表現などに含まれる「高階風」の概念や知識をtupleの構造パターンとして簡潔に表現できる。ここでは、TLと一階述語計算が、ある関係によって密接に対応づけられることを示し、それを元にTLの完全性と無矛盾性を証明する。この結果、TLが一階述語論理と同様の「扱いやすさ」を持つ推論体系であることがわかる。

1. まえがき

「知識」というものをどのように表せば便利か、という知識の表現法の問題は、知識を計算機によって扱おうとする場合だけではなく、知識の論理構造を理論的に分析、研究する場合においても非常に重要な問題である。例えば人工知能の分野においては、「意味ネットワーク」や「フレーム」など、種々の表現法が考案されており、計算機による実際的な処理の側面からみて、あるいは人間の知識構造に関する心理的な側面からみて、それぞれ興味ある特徴を持っている。

Fregeが今から100年余り前に提案した「述語論理」[3]も、ある意味ではこれらのものと同様、知識表現のための人工言語だと考えられるだろう。もちろん、Fregeが述語論理を考案したときには計算機械(推論機械)のことなどは全く念頭になかったはずであるが、近年ではPrologのような、述語論理を基礎としたプログラミング言語が開発されたこともあり、述語論理式がほぼそのままの形で計算機上での知識の表現や推論に使える、といった状況にある(あるいは、なりつつある)。

また近年では、人間が扱っている知識の内でも特に重要な自然言語によって表現された知識を、述語論理を基礎とした枠組みの中でなるべく正確に表現しようという試みも多くなされるようになった。「内包論理」と呼ぶ複雑な高階述語論理の体系を用いて、英語の意味論を精密に記述しようとしたMontagueの研究[6]は、そのような試みの中で一つの規範となる重要な研究である。

自然言語の意味論がこのような方法論によって研究されるようになってきたのは、ひとえに、述語論理体系それ自体の意味論が数学的に厳密に定められており、従って理論的な扱いに大変適している、という理由による。もちろん、述語論理に関して現在までに得られている豊富な理論的成果が活用できる、という点も重要であろう。

しかし、述語論理の生い立ちをもう一度ふり返って調べてみると、人間の扱っているような知識を表現する言語として、果して(高階の論理も含めて)述語論理が本当に適当なものであるのか、非常に疑問に思える点が出てくるのである。なぜそうなのかは第2章でもう少し詳しく論じることにするが、要点は次の通りである。

Fregeは、自然言語の文(あるいは命題)を、「関数」(今でいう「述語」)と「引数」に分解するという着想を元に述語論理を提案した。この着想は非常にすぐれたものであり、文の論理構造の第一近似として大変有用である。とりわけ、自然数論のような数学理論を記述するための言語として適している。というのは、述語論理の意味論においては、「引数」は「個体」(対象)を表すものとして、「述語」は個体の「性質」あるいは個体間の「関係」を表すものとして解釈される。一方、自然数論というのも、1,2,…などの数が「個体」として存在することを認めた上で、それら個体に関する性質や個体間の関係を論じる理論だからである。つまり数学理論においては、完全に抽象化された形で「個体」という概念が存在しており、「述べられるもの」(つまり対象個体)と「述べるもの」(つまり述語)が、はっきりと区別できる状況にあるのである。

ところが、数学理論ではない一般的な知識を表現した自然言語の文を観察すると、「個体」の概念が非常に不明確にしか現れない場合が多いことに気づく。自然言語表現のうちで、

「個体」に抵抗なく結びつけられるようなものは、固有名詞ぐらいのものであろう（深く考えると、それすらあやしくなってくる）。その結果、「個体」の概念が明確ではない場合にも、何かを個体にでっちあげて引数のところに持ってこないといけない羽目になる。

また、自然言語の文の意味を述語論理式によって表現しようとした場合、文中で述語と考えた単語や句が、同時に別の述語の引数にもなっている、といった「高階の概念」が随所に現れることもわかる。高階の概念を扱うために考えられた高階述語論理は数学的には興味深い枠組みである。しかし、「述語」を数学的な「関数」と解釈するため、高階述語が、関数の関数の…の関数、といったようなものになってしまい、意味論が非常に複雑となる。自然言語の文の意味を表すためにMontague のように高階論理を用いるのは、数学的、哲学的には面白いにしても、本当にそれほど複雑な関数の概念を必要とするのだろうか、という素朴な疑問が生じる。

結局、これらの問題点はすべて、文を「関数」（述語）と「引数」という、数学的にかなり異質な物に分解したことに起因している。

本研究では、文のある種の要素（単語や句）を「関数」と見なし、別の要素を「引数」と見なすという従来の考え方を排し、文のすべての要素を同質の「対象物」として見るような新しい論理体系“Tuple　Logic”を提案する。この論理体系は、種々の概念間の結びつき（論理構造）を対象物の「組」（つまり“tuple ”）として表す点が特徴である。そして高階の概念は、tuple の構造パターンとして表現される。すなわち、この論理体系においては、「関数」の概念も「個体」の概念もなく、あるのは文字列としての“tuple ”だけである。

今述べたように、“Tuple　Logic”は、「個体」の概念が必ずしも明確ではない場合や、高階の概念を簡便に扱いたい場合に実用的にも有用であろう、と筆者は推測している。もちろん、この論理体系が本当に実用的なものとなり得るかどうかは、人工言語であるこの論理体系をいろいろな場面で実際に使ってみてからでなければ結論は出せない。その結果、“Tuple　Logic”よりもやはり従来の（高階の）述語論理の方がよかった、という結論が出る可能性もある。しかし、述語論理は、そもそも、かなり大胆な仮定のもとに自然言語の文を近似したものであった。それゆえ、単に数学理論の記述に大きな成功を収めたという理由だけで、この仮定を吟味することなく、それよりもずっと微妙な知識を表現するための言語として、述語論理を無批判かつ安易に利用する、ということは大変危険であると考えている。つまり、本論文の“Tuple　Logic”は、述語論理とは異なる観点あるいは仮定を元に、知識の表現と推論のための体系を構築しようとする1つの試みである。

本論文の第２章では、Ｆregeにまで立ち戻って述語論理を再検討し、“Tuple　Logic”の提案の意義を論じる。第3章では、“Tuple　Logic”の統語論（syntax）と意味論（semantics ）の厳密な定義を与える。第４章では、一階述語計算との関係を明らかにし、それを元に“Tuple　Logic”の完全性を証明する。

２．なぜ“Tuple　Logic”か？（述語論理の再検討）

２．１．述語論理のとっている立場と仮定、およびその検討

現代論理学の出発点である歴史的名著“Begriffsschrift”（『概念記法』）［３］の中でＦregeは、自然言語の文の論理構造を詳細に分析し、その結果、今日いうところの「述語論理」の体系を提案した。述語論理は、それまでのアリストテレス的な論理におけるように、文を「主語」と「述語」に分け、それらの形態に着目して文（命題）の論理構造を分類するのではなく、数学における式のように、文を「関数」と「引数」に分解する、という着想に基いて構築されている点が特徴である（但し現在では、Ｆregeのいう「関数」を「述語」と呼んでいる）。この着想が大変すぐれたものであることは、Ｆrege自身の説明もさることながら、彼以後の非常に多数の研究者が述語論理を標準的な論理体系として多くの場面で使用していることからも、うかがい知ることができる。また、述語論理が自然数論などの数学理論の記述のための言語として特に有用であり、これに基いて多くの興味深い結果が導かれていることも、よく知られている通りである。

さて、文を「関数」と「引数」に分解する、ということから必然的に、「述べるもの」と「述べられるもの」の区別が生まれてくる（もっとも、Ｆrege自身はその区別をあまりはっきりさせていなかったため、Russel によって矛盾を指摘されることになる）。

「述べられるもの」の概念を抽象化していくと、次第に、何かある「まとまりのあるもの」という概念が浮んでくる。なぜなら、まとまり（あるいは、境界とか区別）のないようなものは、それについて「述べよう」と思っても、その範囲が定まらないからである。このような抽象化を押し進めていくと最終的には、「述べられるもの」の概念が「個体」という不可分な対象の概念に対応づけられることになる。またその結果、「述べるもの」は、個体の性質や個体間の関係を叙述するようなものとなる。現在採用されている述語論理体系に対する「解釈」（interpretation）は、まさにこのような立場を定式化したものである。

述語論理体系のとっているこの立場が要求する仮定を書き出してみると、次のようになるだろう。

- 個体および個体集合の存在を認める。
- 個体には種々の「性質」が、個体間には種々の「関係」が、それぞれ付随している、ということを認める。
- 文の中で「引数」と見なされる表現は「個体」に、「関数」と見なされる表現は「性質」または「関係」に、それぞれ対応させる、ということを認める。

これらの仮定は、人間が世界を認識するあり方のモデルとして、単純でしかも非常にすぐれた１つの見方である。しかしながら、これは絶対的なものではない。

「個体」の概念は、「述べられるもの」という概念があった上で、さらにそれを抽象化した結果初めて得られるものである。確かに、日本語や英語においては「主語」と「述語」という概念があり、それらについてはほぼ「述べられるもの」と「述べるもの」に対応づけられるが、前者が常に「個体」の概念に対応づけられているかは疑問である。実際、人間が頭の中でいろいろな知識を扱うときに、いつもこのように徹底した抽象化を行っているとは考えにくい（少なくとも、そのようにしているという証拠はない）。また、「関数」と「引数」の考え方は、文の修飾的な構造（例えば副詞句、前置詞句など）を説明するものではない。

述語論理をきちんと定式化しようとした場合には、どんな概念についても徹底した抽象化をしておかないと数学的に厳密な定義にならない、という事情があるので、「述べられるもの」という概念を出した以上、それを「個体」の概念にまで抽象化せざるを得なくなってしまう。第１章で述べたように、数学理論が述語論理と相性が良いのは、単にどちらも抽象化が徹底しているというだけでなく、元々数学者であったＦregeが、「述べられるもの」と「述べるもの」の区別が明確になされている「数学」を手本として述語論理を作ったのであるから、当然といえば当然のことである（“Begriffs-schrift ”の副題が『算術を模して作られた純粋思考のための式言語』となっていることに注意）。

人は日常的な思考においても常に「個体」という抽象的な概念を基礎として思考しているのか？という問題は大変難しいので、そう簡単に結論は下せない。しかし、人間が感覚器官を通して世界（外界）を観測したときに一次情報として直接に得るものは、観測したものが個体に対応するものであれ性質に対応するものであれ、いずれの場合も神経インパルスの時空パターンという、完全に同質な情報のはずである。従って、もし「個体」という概念が生じたとすれば、これらの一次情報を元に、頭の中の高度な抽象化のプロセスを経ることによってのみ得られるのだと考えられる。つまり、「個体」というものは、外界に普遍的に存在するものではなく、抽象化によって作り出されるものだ、ということである。

それゆえ、こういった抽象化のプロセスをも一種の思考、推論の過程として記述できるような体系であれば、「個体」の概念を完全に排除してしまうという立場も（役に立つかどうかは別として）不合理ではないと考えられる。

２．２．高階述語論理の検討

対象個体（を表す記号）のみが述語の引数となり得る「一階の述語論理」は、定式化が簡単であるためよく研究されている。その性質として最も重要なのは、「完全性」が成立つことである。もう一つの重要な性質は、一階の論理の「解釈」が高階の論理の解釈に比べて簡単であるということである。特に、論理式の恒真性が問題となる場合には、対象個体の領域として任意の領域を考える代りに、（可算個の要素からなる）Herbrand 領域を考えるだけでよい、ということが知られている。

（一階の）述語を引数とするような二階の述語、それをまた引数とするような三階の述語、などを許す高階述語論理は、これに比べると扱いにくい体系である。まず、有限的に記述された公理と推論規則からすべての恒真式が証明できない、という意味で完全ではない。また、対象領域として非可算の領域を考える必要のある場合が存在する。対象領域として可算領域を考えるだけでよい場合でも、一階述語の解釈となり得る関数の集合が非可算になるので、二階以上の述語の解釈は、非可算個の関数集合を定義域とする関数となる。

このように厄介な性質を持つにもかかわらず、高階論理は数学理論の記述のための体系としては有用であり、興味深い結果が多数導かれている（例えば［２］参照）。

ところで、人が扱っているような知識の中にも、高階風の概念は随所に現れる。実際我々は、文について述べることもできれば、数学理論について論じることもできる。もちろん、そのために上に述べたような高階論理が必要かどうかは別問題であり、これについてはすぐあとで論じる。しかし、自然言語の文の意味を、述語論理の立場に即して「述語」と「引数」に分けて表現しようとすると、そのとたんに高階の述語（高階の関数）を必要とするような事例にぶつかる。

例えば、英語の自動詞を１引数の真理値関数（述語）に翻訳したとすると、自動詞に係る副詞（句）は、真理値関数から真理値関数への関数、副詞に係る副詞は、［真理値関数から真理値関数への関数］から［真理値関数から真理値関数への関数］への関数、と解釈せざるを得ない。というのは、例えば“walk very slowly”という句を考えると、副詞“very”は副詞“slowly”に係って、副詞句である新しい概念“very slowly”を生み出し、さらにそれは自動詞“walk”に係って、自動詞句である新しい概念“walk very slowly”を生み出している、と考えるのが自然だからである。

これは、「カテゴリー文法」（例えば［１］参照）における統語論的な各カテゴリーの「意味」として、そのカテゴリーの構成法にちょうど相似な関数を対応させる、という考え方である。これに「内包」という概念をつけ加え、数学的に洗練された形にまとめたものが、Montague による、いわゆる「モンタギュー文法」である［６］。

このような考え方は確かに数学的には大変面白いものであるが、例えば副詞をなぜこれほど複雑な関数と考えねばならないのかという、その必然性が見当らない（但し、だからだめだ、と言っているのではなく、このような考え方が絶対的なものではない、ということだけを主張している）。

また、このような考え方を仮に認めたとしても、なお、従来の高階論理の枠組みは知識表現のための人工言語として強力すぎる。これは次の理由による。まず、対象個体の領域は

可算であるとする（我々は自然数についても論じたいので）。そうすると、先に述べたように高階の関数は非可算集合を定義域とする関数となる。しかし、例えば副詞は、言語表現として実際に存在する自動詞句（これは可算）に対してだけ作用できれば十分であるので、副詞を非可算集合を定義域とする関数と見なすのはあまりにも無駄が多すぎるのである。

このように、自然言語によって表されるような知識の表現のためには、従来の高階の論理は強力すぎるが、しかし、現実には先に述べたような「高階風」の知識が頻繁に現れるので、それらも表現できる体系でなければならない。

Gödelが有名な「不完全性定理」を証明したとき、いわゆる「Gödel数」という符号化によって、超数学的言明を１つの数で表した。これに類する方法を用いると、数あるいは数に相当するものさえ扱えれば、ある種の高階の概念については、一階の述語論理でも、それを符号化して表現することができる。（もちろん、本当の高階論理でないと扱えない、真の高階の概念も存在する。これについては［４］参照）。そして実際、人間の頭の中ではどんな概念でも神経インパルスによって符号表現されているはずだから、高階の概念といっても、そのような符号化によって表現できる、疑似的な高階の概念だけに限定してもよいだろう。

一階述語論理で間に合うのならば、それを使えばよいではないか、と思えるかも知れないが、一階述語論理は本来そのようなものとして設計されていないので、「疑似高階」の概念を表現しようとすると、ある種の符号化が必要となり、直接的に表現するのが困難である。

従って結局の所、高階風の概念をもっと直接的かつ簡潔に表現できるような論理体系が望まれる（但し、扱える範囲は一階述語論理と同程度でよい）。

２．３．“Tuple Logic”の提案とその特徴

本論文では、以上の検討をふまえた上で、述語論理とは異なった立場から、新しい論理体系“Tuple Logic”（ＴＬ）の定式化を試みた（厳密な定義は第3章に与える）。ＴＬは、

- 「関数」と「引数」の概念をなくす（従って、「個体」の概念もなくなる）。
- 「高階風」の概念を簡便に扱えるようにする。

の２点を満し、しかも自然言語の文がすなおに翻訳できるような、そういう論理体系を作ろうという意図を具体化したものである。

ＴＬの基本構成要素は、「アトム」と呼ぶ文字列と「括弧」である。アトムには定数と変数があり、変数は“＊Ｘ”のように先頭に“＊”がついた文字列で表す。「タプル」というのは、１個以上のアトムあるいは他のタプルを空白をはさんで並べ、全体を括弧でくくった文字列である。（形式上、１個のアトムもタプルと呼ぶ。また、次章では、「基本タプル」と「論理タプル」の区別を設けている。）タプルは見かけ上、Ｌisp のS式と同じ形をしている。しかし、Ｌisp はあくまでプログラミング言語であり、しかも「関数」の概念を基礎としているのに対し、ＴＬは論理体系であり、しかも「関数」の概念を排除している点が根本的に異なる。

以下では、いくつかの英文を例にとり、それらがＴＬでどのように表現できるかを示す。但し、ここではＴＬの特徴を述べることを目的としているので、種々の文型をどのようなタプルに翻訳すればよいか、各単語の意味をどのような意味公準として与えればよいか、推論規則はどうすればよいか、などの問題は扱わない。

“John walks”という英文を考えてみる。これは、一階の述語論理の枠内では通常、“walk”を述語と見なして、walk（John） という式に翻訳するが、ＴＬでは単純に、（John walk）と書き表せる。つまり、“walk”をいわば引数に「格下げ」して、述語をなくしてしまったわけである。従って、アトムを並べる順序も英語の語順と同じでよい。

タプル （John walk） は、直観的には、“John ”と“walk”という記号が、単にそういう順序で並べられるような関係にある、ということを意味しているだけであり、決して、何か外界に存在する“John ”という個体が“walk”するという性質を持っている、ということを意味しているのではない。ＴＬにおいては、存在が無条件に認められているのは文字や文字列だけである。従って、ＴＬの意味論も、これら文字列を外界の「もの」や「真理値」に対応させたりはしない（人間の頭の中や計算機の中でも、外界にあるものを直接扱っているわけではないので）。このため、後に示すように、ＴＬの意味論（解釈）が非常に簡単に定義できる。

一階の論理では表現しにくい“John walks very slowly ”という文は、 （John（walk（very slowly））） というタプルで表せる。副詞は、他の単語や句に「係る」ことによって新しい意味を生み出すものであり、このタプルは、この「係り方」の構造を「括弧」によってすなおに表したものに他ならない。

ＴＬは、従来の論理体系と同様に、“～”（否定）、“→”（含意）、“∀”（全称限量詞）、のような論理記号も持っている。これを用いると、“every ”や“a ”などの限量表現も、従来の体系と同様に表すことができる。

例えば、“Every man walks slowly”という文は、

（∀ ＊x （（＊x man）→（＊x （walk slowly））））

というタプルで表現できるだろう。但し、変数＊x は、従来の論理のように個体集合上を動くのではなく、（論理記号と変数を含まない）すべてのタプルの集合上を動く。また、このタプルと、（（every man）（walk slowly））とが等価になるように意味公準と推論規則が定められれば、より好都合であろう（但し、ここではそれについては論じない）。

また、（∀ ＊x （（Mary ＊x ）→（John ＊x ）））というタプルによって、『Mary の持つ性質はすべてJohn が持っている』に近い意味を表現できる。つまり、一階述語論理において「述語」に相当していたものに対しても、気がねなく変数を導入でき、限量化できる。

３．論理体系“Tuple Logic”

３．１．Tuple Logic（ＴＬ）の定義

（１）ＴＬで使う文字

(i) 基本記号

Ａ，Ｂ，…，Ｚ，a，b，…，z，０，１，…９，

␣（空白），＊，…

(ii) 論理記号

～（否定），→（含意），∀（全称限量詞）

(iii) 括弧

（，）

（２）定数と変数

(i) 定数

＊で始まらず、␣を含まない任意の基本記号列（有限列）を定数と呼ぶ。すべての定数の集合をＣと記す。

(ii) 変数

＊で始まり、␣を含まない任意の基本記号列を変数と呼ぶ。すべての変数の集合をＶと記す。

（３）アトム（atomic symbol）

定数または変数をアトムという。すべてのアトムの集合をＡと記す（従って、Ａ＝Ｃ∪Ｖ）。

（４）基本タプル（basic tuple）

(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、

（ω_1 ␣ω_2 ␣…␣ω_n）は基本タプルである。（但し、n＝0,1,…）

(iii) (i)と(ii)によって定められるものだけが基本タプルである。

すべての基本タプルの集合をΩと記す。（Ａ⊂Ωであることに注意。）

（５）グラウンド・タプル（ground tuple）

基本タプルのうちで、変数を含まないタプルをグラウンド・タプルという。すべてのグラウンド・タプルの集合をΔと記す。（Ｃ⊂Δ⊂Ωであることに注意。）

（６）論理タプル（logical tuple）

(i) α_1，α_2，…，α_n が基本タプルなら、

（α_1 ␣α_2 ␣…␣α_n）は論理タプルである。（但し、n＝0,1,…）

(ii) α，βが論理タプル、νが変数なら、

（～␣α），（α␣→␣β），（∀␣ν␣α）

は論理タプルである。

(iii) (i)～(ii)によって定められるものだけが論理タプルである。

すべての論理タプルの集合をΛと記す。（（Ω－Ａ）⊂Λであることに注意。）

また、以下では論理タプルのことを単にタプルと呼ぶことにする。

３．２．自由変数と束縛変数、および代入

【定義　３．２．１】　［自由変数と束縛変数］

(i) 限量詞∀を全く含まないタプルαの中に現れている変数はどれも、自由変数として生起しているという。

(ii) タプルα，βにおいて自由変数として生起している変数は、

（～　α）　および　（α　→　β）

というタプルにおいても自由変数として生起しているという。

(iii) αをタプル、νを変数とするとき、

（∀　ν　α）

というタプルの中に現れている変数νはどれも、束縛変数として生起しているという。また、αにおいて自由変数として生起しているν以外の変数は、このタプルにおいても自由変数として生起しているという。

【定義　３．２．２】　［代入］

Ｖに属する各変数をΩ中の適当な要素（基本タプル）に置換えることを代入という。つまり、代入σは、

σ：　Ｖ　→　Ω

なる写像である。特に、Ｖ中のある有限個の変数ν_1，…，ν_k を基本タプル　ω_1，…，ω_k に置換え、　ν_1，…，ν_k 以外の変数は置換えずにそのままにしておくような代入σを

$\sigma \equiv [\omega_1 / \nu_1, \ \cdots \ , \omega_k / \nu_k]$

で表現する。

また、代入のうちで特に、各変数をグラウンド・タプルに置換えるような代入、つまり、

σ：　Ｖ　→　Δ

なる写像をグラウンド代入と呼ぶ。

σを任意の代入とするとき、タプルα中の変数のうちで自由変数として生起しているものをすべて、代入σによって置換えて得られるタプルを　$\alpha\sigma$　で表す。

３．３．公理的体系としてのＴＬ

ＴＬの公理と推論規則を次のように定める。

（１）公理

α，β，$\gamma \in \Lambda$、$\omega \in \Omega$、$\nu \in V$　とするとき、以下のタプルはすべて公理である。

①（α　→（β　→　α））

②（（α→（β→γ））→（（α→β）→（α→γ）））

③（（（～α）→（～β））→（β→α））

④（（∀　ν　α）→　α［ω／ν］）

⑤（（∀　ν　（α→β））→（α→（∀　ν　β）））

但し⑤では、νはαにおいて自由変数としては現れていないとする。

（2）推論規則

推論規則は次の2つである。（α，$\beta \in \Lambda$、$\nu \in V$）

① 分離法則（modus ponens）

α と （$\alpha \rightarrow \beta$） とから、β を導く。

② 一般化

α から （$\forall \ \nu \ \alpha$） を導く。

【定義 3．3．1】［TLの定理］

公理を元に、推論規則を有限回適用して得られるようなタプルをTLの定理と呼ぶ。$\alpha \in \Lambda$が定理であることを

$\vdash \alpha$

と書く。また、αが定理でないことを

$\nvdash \alpha$

と書く。

3．4．TLの意味論

【定義 3．4．1】［TLの解釈（interpretation）］

TLの解釈とは、以下の条件を満すような（任意の）タプルの集合$\mathscr{I}$（$\subset \Lambda$）のことである。

(i) 自由変数を含まない任意の$\alpha \in \Lambda$に対して、
$(\sim \alpha) \in \mathscr{I}$ iff $\alpha \notin \mathscr{I}$

(ii) 自由変数を含まない任意のα，$\beta \in \Lambda$に対して、
$(\alpha \rightarrow \beta) \in \mathscr{I}$ iff $\alpha \notin \mathscr{I}$ または、$\beta \in \mathscr{I}$

(iii) 任意の$\alpha \in \Lambda$に対して、
$\alpha \in \mathscr{I}$ iff すべてのグラウンド代入σに対して$\alpha\sigma \in \mathscr{I}$となる

(iv) 任意の$\nu \in V$と、任意の $\alpha \in \Lambda$に対して、
$(\forall \ \nu \ \alpha) \in \mathscr{I}$ iff $\alpha \in \mathscr{I}$

【定義 4．3．2】［TLの解釈の基底］

$\mathscr{I}$をTLの任意の解釈とする。このとき、$\mathscr{B} = \mathscr{I} \cap \Delta$ なる$\mathscr{B}$を、解釈$\mathscr{I}$の基底と呼ぶ。

【定義 3．4．3】［Validity］

いかなる解釈$\mathscr{I}$にも属するようなタプルをvalid なタプルと呼ぶ。

【定義 3．4．4】［モデル］

ある解釈$\mathscr{I}$が、あるタプルの集合S（$\subset \Lambda$）のモデルであるとは、 $S \subset \mathscr{I}$ となることをいう。

4．一階述語計算との関係、および完全性

この章では、TLの体系と、ある種の一階述語計算の体系（FPC）とが、後に示す2つの写像θとϕによって、統語論的および意味論的に密接に関係づけられることを示す。θはTLの基本タプルをFPCの項に、ϕはTLの（論理）タプルをFPCの式に、それぞれ対応づける写像である。従って、TLの「基本タプル」（但し、アトムは除く）は、FPCの「項」に相当するものであるとも「式」に相当するものであるとも、どちらともとれるような性格を持つと考えられる。TLは、「述語」を持たないという点でFPCと非常に異なっているのだが、写像θとϕによって対応づけると「一階述語計算とそれほど大きくは違っていない体系」であることがわかる。従って、一階述語計算と同様の「扱いやすさ」を持つ体系であることがわかる。この章の最後では、この性質を使ってTLの完全性を示す。

4．1．一階述語計算の体系FPC

ここでは、一階述語計算（first-order predicate calculus）の体系として次のような体系FPCを考える（FPCの構成に際しては文献［5］を参考にした）。FPCが、通常の一階述語計算の体系の定式化と異なっているのは次の点である。

・ n 項関数記号と n項述語記号が、各 n（n ＝0,1,…）に対してちょうど1つしか存在しないこと。

（1）FPCの記号

(i) 定数
c_1，c_2，…

(ii) 変数
x_1，x_2，…
すべての変数の集合をXと記す。

(iii) 関数記号
f^0，f^1，f^2，…
（但し、f^n（n ＝0,1,…）は、n 引数の関数記号である。）

(iv) 述語記号
p^0，p^1，p^2，…
（但し、p^n（n ＝0,1,…）は、n 引数の述語記号である。）

(v) 論理記号
～（否定），→（含意），∀（全称限量詞）

(vi) 括弧
（，）

（2）項

(i) 定数 c_i および変数 x_i は項である（i ＝1,2,…）。

(ii) t_1，…，t_n が項なら、f^n（t_1，…，t_n）も項である（n ＝0,1,…）。

(iii) (i)と(ii)で定められるものだけが項である。

FPCのすべての項の集合をΘと記す。

（3）式

(i) $t_1, \cdots, t_n$ が項なら、$p^n(t_1, \cdots, t_n)$ は式である（$n=0,1,\cdots$）。

(ii) ζ, η が式なら、

$(\sim\zeta)$, $(\zeta\to\eta)$, $(\forall x_i\ \zeta)$

も式である（$i=1,2,\cdots$）。

(iii) (i)〜(ii)で定められるものだけが式である。

ＦＰＣのすべての式の集合をΦと記す。

（4）自由変数と束縛変数

自由変数と束縛変数の概念の定義は通常の一階述語計算の場合と同様である。

４．２．ＦＰＣの公理と推論規則

ＦＰＣの公理と推論規則はＴＬと同様のものである。また、ＦＰＣの「定理」も、これら公理と推論規則を元に、通常の方法で定義される。

（1）公理

ζ, η, ξ を式、t を項、x を変数とするとき、以下の形の式はすべて公理である。

(i) $(\zeta\to(\eta\to\zeta))$

(ii) $((\zeta\to(\eta\to\xi))\to((\zeta\to\eta)\to(\zeta\to\xi)))$

(iii) $(((\sim\zeta)\to(\sim\eta))\to(\eta\to\zeta))$

(iv) $((\forall x\ \zeta)\to\ \zeta')$

但し ζ' は、ζ 中で自由変数として現れているすべての x を項 t によって置換えて得られる式である。

(v) $((\forall x(\zeta\to\eta))\to(\zeta\to(\forall x\ \eta)))$

但し、x は ζ において自由変数としては現れていないとする。

（2）推論規則

推論規則は次の2つである。（ζ, η は式、x は変数）

(i) 分離法則（modus ponens）

ζ と $(\zeta\to\eta)$ とから、η を導く。

(ii) 一般化

ζ から $(\forall x\ \zeta)$ を導く。

４．３．ＦＰＣの意味論

ＦＰＣの意味論、特に「解釈」、「割当て」、「式の真理値」などの概念は、一階述語計算の体系に対して普通なされている定義（例えば［5］参照）と同様である。但し、ここでは解釈として「Herbrand 解釈」だけを考えることにする。

【定義　４．３．１】［ＦＰＣの解釈］

ＦＰＣの解釈は、

$\mathscr{I}=(D, \Psi)$

によって与えられる。

Ｄは領域であるが、ここではＤをＦＰＣの「Herbrand 領域」にとる。つまり、Ｄは次の(i), (ii)を満す最小の集合である。（$D\subset\Theta$ であることに注意。）

(i) $c_i \in D$　（$i=1,2,\cdots$）

(ii) $a_1, \cdots, a_n \in D$ ならば $f^n(a_1, \cdots, a_n)\in D$　（$n=0,1,\cdots$）

また、Ψは、変数を含まないＦＰＣの項にはＤの要素を、n 項述語記号 p^n にはＤ上の n項関係を（但し、p^0 には真理値｛0，1｝を）、それぞれ対応させるような、付値である。従って、

- t が変数を含まない項ならば $\Psi(t)\in D$
- $\Psi(p^n)\subset D^n$　　（$n=1,2,\cdots$）
- $\Psi(p^0)\in\{0, 1\}$

である。但し、$\mathscr{I}$ はHerbrand 解釈であるので、Ψは変数を含まない任意の項t をそれ自体に対応させる。つまり、

$\Psi(t)=t$

である。なお、以下では

$(t_1, \cdots, t_n)\in\Psi(p^n)$

$(t_1, \cdots, t_n)\notin\Psi(p^n)$

と書く代りに、それぞれ、

$\Psi(p^n(t_1, \cdots, t_n))=1$

$\Psi(p^n(t_1, \cdots, t_n))=0$

と書いてもよいことにする（$(t_1, \cdots, t_n)\in D^n$）。

【定義　４．３．２】［割当て］

割当てとは、

$s : X \to D$

なる写像 s のことである。つまり、各々の変数にＤの元を割当てる写像である。

また、

$s^* : \Theta\cup\Phi \to \Theta\cup\Phi$

なる s の拡張写像を次のように定義する。

(i) $\zeta\in\Theta$、つまり ζ が項であるとき：

$s^*(\zeta)$ は、ζ の中の各変数 x_i を $s(x_i)$ に置換えて得られる項である。

(ii) $\zeta\in\Phi$、つまり ζ が式であるとき：

$s^*(\zeta)$ は、ζ 中に自由変数として現れている各変数 x_i を $s(x_i)$ に置換えて得られる式である。

【定義　４．３．３】［式の充足性、真理値、恒真性］

（Ｉ）式の充足性

ＦＰＣの式 ζ が解釈 $\mathscr{I}=(D, \Psi)$ と割当て s のもとで充足される（これを $\mathscr{I}, s\models\zeta$ と書く）ということを次のように定義する。（なお、ζ が $\mathscr{I}$ とs のもとで充足されないことを、$\mathscr{I}, s\not\models\zeta$ と書く。）

(i) ある項 $t_1, \cdots, t_n$（$n=0,1,\cdots$）に対して

$\zeta=p^n(t_1, \cdots, t_n)$ と書けるならば、

$\mathscr{I}, s\models\zeta$ iff $\Psi(s^*(\zeta))=1$

(ii) ある式ηに対して$\zeta=(\sim\eta)$と書けるならば、

$\mathscr{J}, s \models \zeta$ iff $\mathscr{J}, s \not\models \eta$

(iii) ある式η，ξに対して$\zeta=(\eta\rightarrow\xi)$と書けるならば、

$\mathscr{J}, s \models \zeta$ iff $\mathscr{J}, s \not\models \eta$ であるか、または $\mathscr{J}, s \models \xi$

(iv) ある式ηとある変数 x_i に対して$\zeta=(\forall\ x_i\ \eta)$と書けるならば、

$\mathscr{J}, s \models \zeta$ iff 高々、変数 x_i に対する割当てのみがs と異なるようなすべての割当て s' に対して $\mathscr{J}, s' \models \eta$

（II）式の真理値

ＦＰＣの式ζが、解釈$\mathscr{J}=(D, \Psi)$のもとで<u>真</u>である（これを $\mathscr{J}\models\zeta$ と書く）ということを次のように定義する。

$\mathscr{J}\models\zeta$ iff すべての割当てs に対し、$\mathscr{J}, s \models \zeta$

（III）式の恒真性

ＦＰＣの式ζが<u>恒真</u>である（これを $\models\zeta$ と書く）ということを次のように定義する。

$\models\zeta$ iff すべての解釈$\mathscr{J}$に対して $\mathscr{J}\models\zeta$

<u>４．４．ＴＬのタプルと、ＦＰＣの項および式との対応づけ</u>

ＴＬの基本タプルにＦＰＣの項を、ＴＬの（論理）タプルにＦＰＣの式を、それぞれある特定の方法で対応づける２つの写像

$$\theta\ :\ \Omega\ \rightarrow\ \Theta$$
$$\phi\ :\ \Lambda\ \rightarrow\ \Phi$$

を以下に定める。

ここでは、ＴＬの定数と変数はそれぞれ、適当な方法であらかじめ番号づけがなされていると仮定し、i 番目の定数をε_i、i 番目の変数をν_i と書くことにする。

【定義 ４．４．１】 ［θとϕの定義］

(i) αが i番目の変数ν_i ならば、

$\theta(\alpha)=x_i$

(ii) αが i番目の定数ε_i ならば、

$\theta(\alpha)=c_i$

(iii) ある$\alpha_1, \cdots, \alpha_n \in \Omega$ （$n=0,1,\cdots$）に対して、$\alpha=(\alpha_1\ \cdots\ \alpha_n)$と書けるならば、

$\theta(\alpha)=f^n(\theta(\alpha_1), \cdots, \theta(\alpha_n))$

$\phi(\alpha)=p^n(\theta(\alpha_1), \cdots, \theta(\alpha_n))$

(iv) ある$\beta\in\Lambda$に対して、$\alpha=(\sim\ \beta)$と書けるならば、

$\phi(\alpha)=(\sim\phi(\beta))$

(v) あるβ，$\gamma\in\Lambda$に対して、$\alpha=(\beta\ \rightarrow\ \gamma)$と書けるならば、

$\phi(\alpha)=(\phi(\beta)\rightarrow\phi(\gamma))$

(vi) ある$\beta\in\Lambda$とある$\nu_i\in V$に対して、$\alpha=(\forall\ \ \nu_i\ \ \beta)$と書けるならば、

$\phi(\alpha)=(\forall\ x_i\ \phi(\beta))$

このようにして定められたθとϕは、共に全単射になることが示せる（証明は容易なので省略）。

【補題 ４．４．１】 θとϕは共に全単射である。

<u>４．５．θとϕのもとでのＴＬとＦＰＣの相同性</u>

前節で定義したθとϕを仲介としてＴＬとＦＰＣを結びつけてやると、これらの論理体系は、統語的構造（特に「定理」の概念）についても意味的構造についても、共にうまく対応がつく。以下ではこれらを示す。

まず、統語的構造については次の定理が成立つ。

【<u>定理 ４．５．１</u>】 ［統語的構造の相同性定理］

任意の$\alpha\in\Lambda$に対し、αがＴＬの定理であるとき、かつそのときに限り、$\phi(\alpha)$はＦＰＣの定理となる。

〘証明〙 これを示すためには、

(a) αがＴＬの公理であるとき、かつそのときに限り$\phi(\alpha)$はＦＰＣの公理である。

(b) βとγからαが［あるいは、βからαが］ＴＬの推論規則①［②］によって導かれるとき、かつそのときに限り、$\phi(\beta)$と $\phi(\gamma)$から $\phi(\alpha)$が［$\phi(\beta)$から $\phi(\alpha)$が］ＦＰＣの推論規則(i)［(ii)］によって導かれる。

を言えばよい。(a)、(b)により、

(c) タプルの列$\alpha_1, \alpha_2, \cdots, \alpha_{j-1}, \alpha$ がＴＬにおけるαの証明になっているとき、かつそのときに限り、式の列 $\phi(\alpha_1), \phi(\alpha_2), \cdots, \phi(\alpha_{j-1}), \phi(\alpha)$がＦＰＣにおける $\phi(\alpha)$の証明になっている。

が成立ち、従って、この定理が結論される。

(a)の証明： ϕの定義と、ＴＬおよびＦＰＣの公理の形から明らか。

(b)の証明： βとγからαが［βからαが］ＴＬの推論規則①［②］によって導かれるならば、$\gamma=(\beta\ \rightarrow\ \alpha)$［ある変数$\nu_i\in V$に対して、$\alpha=(\forall\ \ \nu_i\ \beta)$］と書けるはずである。従って、$\phi(\gamma)=\phi((\beta\ \rightarrow\ \alpha))=(\phi(\beta)\rightarrow\phi(\alpha))$［$\phi(\alpha)=\phi((\forall\ \ \nu_i\ \beta))=(\forall\ x_i\ \phi(\beta))$］となり、ＦＰＣにおいても、推論規則(i)［(ii)］により $\phi(\beta)$と $\phi(\gamma)$から $\phi(\alpha)$が［$\phi(\beta)$から $\phi(\alpha)$が］導かれる。この逆の証明も容易である。 ■

意味的構造の相同性を示すために次の定義を与える。

【定義　4．5．1】［TLの解釈とFPCの解釈の対応］
$\mathscr{I}$をTLの解釈、$\mathscr{J}=(D, \Psi)$をFPCの解釈とする。もし、すべての$\alpha \in \Lambda$に対して、

$\alpha \in \mathscr{I}$　iff　$\mathscr{J} \models \phi(\alpha)$

が成立つならば、$\mathscr{I}$と$\mathscr{J}$は互いに対応する解釈であるという。特に、$\mathscr{I}$は$\mathscr{J}$に対応するTLの解釈、$\mathscr{J}$は$\mathscr{I}$に対応するFPCの解釈であるという。

【定理　4．5．2】［意味的構造の相同性定理］
（Ⅰ）　任意のTLの解釈$\mathscr{I}$に対し、それに対応するFPCの解釈$\mathscr{J}$がちょうど1つ存在する。
（Ⅱ）　任意のFPCの解釈$\mathscr{J}$に対し、それに対応するTLの解釈$\mathscr{I}$がちょうど1つ存在する。

この定理を証明するために、2つの補題を示しておく。

【補題　4．5．1】
(i)　任意のグラウンド代入$\sigma: V \to \Delta$　に対し、すべての$\alpha \in \Lambda$について

$\phi(\alpha\sigma) = s^*(\phi(\alpha))$

となるような割当て $s: X \to D$　が存在する。
(ii)　任意の割当て $s: X \to D$　に対し、すべての$\zeta \in \Phi$について

$\phi^{-1}(s^*(\zeta)) = \phi^{-1}(\zeta)\sigma$

となるようなグラウンド代入$\sigma: V \to \Delta$　が存在する。

〖証明〗 (i)　与えられた代入$\sigma: V \to \Delta$　を元に、割当て $s: X \to D$を次のように定める：　σが変数ν_iに$\delta_i \in \Delta$を代入するならば、sは変数x_iに$\theta(\delta_i) \in D$を割当てる。このように定めたsが、すべての　$\omega \in \Omega$　に対して$\theta(\omega\sigma) = s^*(\theta(\omega))$となることは、$\omega$の構成法に関する帰納法で容易に証明できる。これを用いることにより、$\phi(\alpha\sigma) = s^*(\phi(\alpha))$となることも、$\alpha$の構成法に関する帰納法により簡単に確かめられる。
(ii)　与えられた割当て $s: X \to D$を元に、代入$\sigma: V \to \Delta$を次のように定める：　sが変数x_iに$d_i \in D$を割当てるならば、σは変数ν_iに$\theta^{-1}(d_i) \in \Delta$を代入する。このように定めた$\sigma$が　$\phi^{-1}(s^*(\zeta)) = \phi^{-1}(\zeta)\sigma$　を満すことは、(i)と同様にして簡単に証明できる。　■

【補題　4．5．2】　$\mathscr{J}$をFPCの任意の解釈、sを任意の割当てとする。このとき、すべての式$\zeta \in \Phi$に対して、

$\mathscr{J}, s \models \zeta$　iff　$\mathscr{J} \models s^*(\zeta)$

が成立つ。

〖証明〗　ζに含まれる論理記号の個数kに関する帰納法で証明する。
(i)　k=0 の場合：
容易に証明できるので省略。
(ii)　k=m の場合：
m−1 個以下の論理記号を含むようなすべての$\eta \in \Phi$に対して、$\mathscr{J}, s \models \eta$　iff　$\mathscr{J} \models s^*(\eta)$　が成立つと仮定し、m 個の論理記号を含む任意の$\zeta \in \Phi$についてもこれが成立つことを示す。

①　m 個の論理記号を含む$\zeta \in \Phi$が、ある$\eta \in \Phi$に対して、$\zeta = (\sim\eta)$と書けるとき：

$\mathscr{J}, s \models \zeta$
⇔　$\mathscr{J}, s \models (\sim\eta)$
⇔　$\mathscr{J}, s \not\models \eta$　（∵FPCの充足性の定義）
⇔　$\mathscr{J} \not\models s^*(\eta)$　（∵帰納法の仮定）
⇔　すべての割当てrに対して、$\mathscr{J} \not\models r^*(s^*(\eta))$
（∵ $s^*(\eta) = r^*(s^*(\eta))$ ）
⇔　すべての割当てrに対して、$\mathscr{J}, r \not\models s^*(\eta)$
（∵帰納法の仮定）
⇔　すべての割当てrに対して、
$\mathscr{J}, r \models (\sim s^*(\eta))$
（∵FPCの充足性の定義）
⇔　$\mathscr{J} \models (\sim s^*(\eta))$　（∵FPCの真理値の定義）
⇔　$\mathscr{J} \models s^*((\sim\eta))$　（∵ s^* の定義）
⇔　$\mathscr{J} \models s^*(\zeta)$

②　m 個の論理記号を含む$\zeta \in \Phi$が、ある$\eta, \xi \in \Phi$に対して、$\zeta = (\eta \to \xi)$と書けるとき：

$\mathscr{J}, s \models \zeta$
⇔　$\mathscr{J}, s \models (\eta \to \xi)$
⇔　$\mathscr{J}, s \not\models \eta$　または　$\mathscr{J}, s \models \xi$
⇔　$\mathscr{J} \not\models s^*(\eta)$または　$\mathscr{J} \models s^*(\xi)$
⇔　すべての割当てrに対して、
$\mathscr{J} \not\models r^*(s^*(\eta))$または
$\mathscr{J} \models r^*(s^*(\xi))$
⇔　すべての割当てrに対して、
$\mathscr{J}, r \not\models s^*(\eta)$または　$\mathscr{J}, r \models s^*(\xi)$
⇔　すべての割当てrに対して、
$\mathscr{J}, r \models (s^*(\eta) \to s^*(\xi))$
⇔　$\mathscr{J} \models (s^*(\eta) \to s^*(\xi))$
⇔　$\mathscr{J} \models s^*((\eta \to \xi))$
⇔　$\mathscr{J} \models s^*(\zeta)$

③　m 個の論理記号を含む$\zeta \in \Phi$が、ある$\eta \in \Phi$と$x_i \in X$に対して、$\zeta = (\forall x_i\ \eta)$と書けるとき：
変数$x_i \in X$に対してだけは、Dの要素を割当てずにそのままにしておき、x_i以外の変数に対してはsと同じ要素を割当てる割当てを、$s_i: X - \{x_i\} \to D$と記す。

$\mathscr{J} \models s^*(\zeta)$
⇔　$\mathscr{J} \models s^*((\forall x_i\ \eta))$
⇔　$\mathscr{J} \models (\forall x_i\ s_i{}^*(\eta))$
⇔　すべての割当てrに対して、$\mathscr{J}, r^* \models s_i{}^*(\eta)$
⇔　すべての割当てrに対して、
$\mathscr{J} \models r^*(s_i{}^*(\eta))$

⇔ 高々、変数 x_i に対する割当てのみがs と異なるような、すべての割当て s' に対し、
$\mathscr{I} \models s'^*(\eta)$

⇔ 高々、変数 x_i に対する割当てのみがs と異なるような、すべての割当て s' に対し、
$\mathscr{I}, s' \models \eta$

⇔ $\mathscr{I}, s \models (\forall x_i\ \eta)$

⇔ $\mathscr{I}, s \models \zeta$ ■

【定理4．5．2の証明】

まずＴＬの解釈に対応するＦＰＣの解釈が高々１つしか存在しないことを示す。

あるＴＬの解釈$\mathscr{I}$に対し、それに対応する２つの異なるＦＰＣの解釈 $\mathscr{I}_1 = (D, \Psi_1)$，$\mathscr{I}_2 = (D, \Psi_2)$ があったと仮定して矛盾を導こう。$\mathscr{I}_1$ と$\mathscr{I}_2$ は異なる解釈であるので、付値 Ψ_1 と Ψ_2 は異なっているはずである。述語記号 p^n に対する付値が異なっていたとする。この場合、$\Psi_1(p^n(t_1, \cdots, t_n)) = 1$ かつ$\Psi_2(p^n(t_1, \cdots, t_n)) = 0$（あるいはその逆の付値）となるような $(t_1, \cdots, t_n) \in D^n$ が存在することになる。これから、$\mathscr{I}_1 \models p^n(t_1, \cdots, t_n)$ かつ$\mathscr{I}_2 \not\models p^n(t_1, \cdots, t_n)$ が導かれ、$\phi^{-1}(p^n(t_1, \cdots, t_n))$ が$\mathscr{I}$に属するとしても属さないとしても矛盾が生じる。

次に、ＦＰＣの解釈に対応するＴＬの解釈が高々１つしか存在しないことを示す。

あるＦＰＣの解釈$\mathscr{I}$に対し、それに対応する２つの異なるＴＬの解釈$\mathscr{I}_1$，$\mathscr{I}_2$ があったと仮定して矛盾を導く。$\mathscr{I}_1$ と$\mathscr{I}_2$ は異なるので、$\alpha \in \mathscr{I}_1$ かつ $\alpha \notin \mathscr{I}_2$（あるいはその逆）になるような$\alpha \in \Lambda$が存在するはずである。しかし、$\mathscr{I}_1$ と$\mathscr{I}_2$ が$\mathscr{I}$に対応する解釈であることから、それぞれ、$\mathscr{I} \models \phi(\alpha)$，$\mathscr{I} \not\models \phi(\alpha)$ が導かれ、矛盾が生じる。

以上により、あとは、任意のＴＬの解釈$\mathscr{I}$［あるいは、ＦＰＣの解釈$\mathscr{I}$］に対し、それに対応するＦＰＣ［ＴＬ］の解釈が１つ以上存在することを言えばよい。

（Ｉ）の証明： 任意に与えられたＴＬの解釈$\mathscr{I}$を元にして、ＦＰＣの解釈$\mathscr{I} = (D, \Psi)$ を次のように定める。

各々のグラウンド・タプル$\alpha \in \Delta$に対して：

$\alpha \in \mathscr{I}$ならば、$\Psi(\phi(\alpha)) = 1$ とする。

$\alpha \notin \mathscr{I}$ならば、$\Psi(\phi(\alpha)) = 0$ とする。

（Ψは$\mathscr{I}$の基底だけによって定められていることに注意。）

このように定めた$\mathscr{I}$が$\mathscr{I}$に対応するＰＬの解釈になっていること、つまり、$\alpha \in \mathscr{I}$ iff $\mathscr{I} \models \phi(\alpha)$ であることを示そう。αに含まれる論理記号（～，→，∀）の個数 kに関する帰納法によって証明する。

(i) $k=0$ の場合：

$k=0$ というのは、αが$\Omega - \Lambda$の要素である場合である。このとき、

$\alpha \in \mathscr{I}$

⇔ すべてのグラウンド代入σに対して、
$\alpha\sigma \in \mathscr{I}$ （∵ＴＬの解釈の定義）

⇔ すべてのグラウンド代入σに対して、
$\Psi(\phi(\alpha\sigma)) = 1$ （∵$\mathscr{I}$の定め方）

⇔ すべての割当てs に対して、
$\Psi(s^*(\phi(\alpha))) = 1$ （∵補題4．5．1）

⇔ すべての割当てs に対して、
$\mathscr{I}, s \models \phi(\alpha)$ （∵ＦＰＣの充足性の定義）

⇔ $\mathscr{I} \models \phi(\alpha)$ （∵ＦＰＣの真理値の定義）

(ii) $k=m$ の場合：

$m-1$ 個以下の論理記号を含むようなすべての $\beta \in \Lambda$ に対して $\beta \in \mathscr{I}$ iff $\mathscr{I} \models \phi(\beta)$ が成立つと仮定する。

① m 個の論理記号を含む$\alpha \in \Lambda$が、ある$\beta \in \Lambda$に対して、$\alpha = (\sim\ \beta)$ と書けるとき：

$\alpha \in \mathscr{I}$

⇔ すべてのグラウンド代入σに対して、
$(\sim\ \beta)\sigma \in \mathscr{I}$ （∵ＴＬの解釈の定義）

⇔ すべてのグラウンド代入σに対して、
$(\sim\ \beta\sigma) \in \mathscr{I}$ （∵σの定義）

⇔ すべてのグラウンド代入σに対して、
$\beta\sigma \notin \mathscr{I}$ （∵ＴＬの解釈の定義）

⇔ すべてのグラウンド代入σに対して、
$\mathscr{I} \not\models \phi(\beta\sigma)$ （∵帰納法の仮定）

⇔ すべての割当てs に対し、
$\mathscr{I} \not\models s^*(\phi(\beta))$ （∵補題4．5．1）

⇔ すべての割当てs に対し、
$\mathscr{I}, s \not\models \phi(\beta)$ （∵補題4．5．2）

⇔ すべての割当てs に対し、
$\mathscr{I}, s \models (\sim \phi(\beta))$ （∵ＦＰＣの充足性の定義）

⇔ $\mathscr{I} \models (\sim \phi(\beta))$ （∵ＦＰＣの真理値の定義）

⇔ $\mathscr{I} \models \phi((\sim\ \beta))$ （∵ϕの定義）

⇔ $\mathscr{I} \models \phi(\alpha)$

② m 個の論理記号を含む$\alpha \in \Lambda$が、ある$\beta, \gamma \in \Lambda$に対して、$\alpha = (\beta \rightarrow \gamma)$ と書けるとき：

$\alpha \in \mathscr{I}$

⇔ すべてのグラウンド代入σに対して、
$(\beta \rightarrow \gamma)\sigma \in \mathscr{I}$ （∵ＴＬの解釈の定義）

⇔ すべてのグラウンド代入σに対して、
$(\beta\sigma \rightarrow \gamma\sigma) \in \mathscr{I}$ （∵σの定義）

⇔ すべてのグラウンド代入σに対して、
$\beta\sigma \notin \mathscr{I}$ または $\gamma\sigma \in \mathscr{I}$（∵ＴＬの解釈の定義）

⇔ すべてのグラウンド代入σに対して、
$\mathscr{I} \not\models \phi(\beta\sigma)$ または $\mathscr{I} \models \phi(\gamma\sigma)$
（∵帰納法の仮定）

⇔ すべての割当てs に対し、
$\mathscr{I} \not\models s^*(\phi(\beta))$ または $\mathscr{I} \models s^*(\phi(\gamma))$
（∵補題4．5．1）

⇔ すべての割当てs に対し、

$\mathscr{I}$, s $\not\models \phi$ (β) または $\mathscr{I}$, s $\models \phi$ (γ)

(∵補題4．5．2)

⇔ すべての割当てs に対し、

$\mathscr{I}$, s $\models$ (φ (β) → φ (γ))

(∵ＦＰＣの充足性の定義)

⇔ $\mathscr{I} \models$ (φ (β) → φ (γ))

(∵ＦＰＣの真理値の定義)

⇔ $\mathscr{I} \models \phi$ ((β → γ)) (∵φの定義)

⇔ $\mathscr{I} \models \phi$ (α)

③ m 個の論理記号を含むα∈Λが、あるβ∈Λとν_i∈Vに対して、α＝(∀ ν_i β) と書けるとき：

α∈$\mathscr{I}$

⇔ β∈$\mathscr{I}$ (∵ＴＬの解釈の定義)

⇔ すべてのグラウンド代入σに対して、

βσ∈$\mathscr{I}$ (∵ＴＬの解釈の定義)

⇔ すべてのグラウンド代入σに対して、

$\mathscr{I} \models \phi$ (βσ) (∵帰納法の仮定)

⇔ すべての割当てs に対し、

$\mathscr{I} \models$ s^* (φ (β)) (∵補題4．5．1)

⇔ すべての割当てs に対し、

$\mathscr{I}$, s $\models \phi$ (β) (∵補題4．5．2)

⇔ すべての割当てs に対し、

高々、変数 x_i に対する割当てのみがs と異なるような、すべての割当て s′ に対して、

$\mathscr{I}$, s′ $\models \phi$ (β)

⇔ すべての割当てs に対し、

$\mathscr{I}$, s $\models$ (∀ x_i φ (β))

(∵ＦＰＣの充足性の定義)

⇔ $\mathscr{I} \models$ (∀ x_i φ (β)) (∵ＦＰＣの真理値の定義)

⇔ $\mathscr{I} \models \phi$ ((∀ ν_i β)) (∵φの定義)

⇔ $\mathscr{I} \models \phi$ (α)

(Ⅱ) の証明： 与えられたＦＰＣの解釈$\mathscr{I}$＝(D, Ψ) を元にして次のように定められる集合$\mathscr{I}$⊂Λを考える：

任意の式ζ∈Φに対して、

$\mathscr{I} \models$ ζ ならば ϕ^{-1}(ζ) ∈$\mathscr{I}$ とする。

$\mathscr{I} \not\models$ ζ ならば ϕ^{-1}(ζ) ∉$\mathscr{I}$ とする。

明らかに、この集合$\mathscr{I}$は

α∈$\mathscr{I}$ iff $\mathscr{I} \models \phi$ (α)

を満す。従って、あとは$\mathscr{I}$がＴＬの解釈になっていること、つまり、定義3．4．1の条件(i)～(iv)を満すことを示せばよい。それには、次の①～④を証明すればよい。

① 自由変数を含まないすべてのα∈Λに対して、

$\mathscr{I} \models \phi$ ((～ α)) iff $\mathscr{I} \not\models \phi$ (α)

② 自由変数を含まないすべてのα, β∈Λに対して、

$\mathscr{I} \models \phi$ ((α → β)) iff $\mathscr{I} \not\models \phi$ (α) または

$\mathscr{I} \models \phi$ (β)

③ すべてのα∈Λに対して、

$\mathscr{I} \models \phi$ (α) iff すべてのグラウンド代入σに対して、

$\mathscr{I} \models$ (ασ)

④ すべてのα∈Λとν_i∈Vに対して、

$\mathscr{I} \models \phi$ ((∀ ν_i α)) iff $\mathscr{I} \models \phi$ (α)

なぜなら、α∈$\mathscr{I}$ iff $\mathscr{I} \models \phi$ (α) であるので、例えば①から、

(～ α) ∈$\mathscr{I}$ iff α∉$\mathscr{I}$

つまり、定義3．4．1の条件(i)が導かれる。

①の証明：

$\mathscr{I} \models \phi$ ((～ α))

⇔ すべてのs に対し、$\mathscr{I}$, s $\models \phi$ ((～ α))

⇔ すべてのs に対し、$\mathscr{I}$, s $\models$ (～φ (α))

⇔ すべてのs に対し、$\mathscr{I}$, s $\not\models \phi$ (α)

⇔ すべてのs に対し、$\mathscr{I} \not\models$ s^* (φ (α))

⇔ $\mathscr{I} \not\models \phi$ (α)

(∵αが自由変数を含まないので、φ (α) も自由変数を含まない。従って、どのようなs に対しても s^* (φ (α)) ＝φ (α) となる。)

②の証明：

$\mathscr{I} \models \phi$ ((α → β))

⇔ すべてのs に対し、$\mathscr{I}$, s $\models \phi$ ((α → β))

⇔ すべてのs に対し、$\mathscr{I}$, s $\models$ (φ (α) → φ (β))

⇔ すべてのs に対し、

$\mathscr{I}$, s $\not\models \phi$ (α) または $\mathscr{I}$, s $\models \phi$ (β)

⇔ すべてのs に対し、

$\mathscr{I} \not\models$ s^* (φ (α)) または $\mathscr{I} \models$ s^* (φ (β))

⇔ $\mathscr{I} \not\models \phi$ (α) または $\mathscr{I} \models \phi$ (β)

③の証明：

$\mathscr{I} \models \phi$ (α)

⇔ すべてのs に対し、$\mathscr{I}$, s $\models \phi$ (α)

⇔ すべてのs に対し、$\mathscr{I} \models$ s^* (φ (α))

⇔ すべてのグラウンド代入σに対し、$\mathscr{I} \models \phi$ (ασ)

④の証明：

$\mathscr{I} \models \phi$ ((∀ ν_i α))

⇔ すべてのs に対し、$\mathscr{I}$, s $\models \phi$ ((∀ ν_i α))

⇔ すべてのs に対し、$\mathscr{I}$, s $\models$ (∀ x_i φ (α))

⇔ すべてのs に対し、

高々、変数 x_i に対する割当てのみがs と異なっているような、すべての割当て s′ に対して、

$\mathscr{I}$, s′ $\models \phi$ (α)

⇔ すべてのs に対し、$\mathscr{I}$, s $\models \phi$ (α)

⇔ $\mathscr{I} \models \phi$ (α) ■

4．6．ＴＬの完全性

前節の結果を用いると、ＴＬの完全性、つまり『タプルα

がＴＬの定理であるとき、かつそのときに限り、αは valid である』を導くのは容易である。これからまた、ＴＬの無矛盾性も保証されることになる。

通常の一階述語計算の体系に関しては、『Gödelの完全性定理』としてよく知られている結果がある（ＦＰＣと同じ公理系と推論規則を持つ一階述語計算の完全性の証明は、例えば文献［５］参照）。ＦＰＣと通常の一階述語計算の体系との間には、この章の始めに指摘したような点での違いはあるが、これは完全性の証明には全く影響しないので、ＦＰＣについても完全性定理が成立つ。

【Gödelの完全性定理】

$\zeta \in \Phi$をＦＰＣの任意の式とするとき、

ζがＦＰＣの定理 iff $\models \zeta$

という関係が成立つ。

ＴＬの完全性は、この定理と、前節に示した、ＴＬとＦＰＣの間の相同性定理を用いて証明できる。

【定理 ４．６．１】 ［ＴＬの完全性定理］

$\alpha \in \Lambda$をＴＬの任意のタプルとするとき、

$\vdash \alpha$ iff αが validである

という関係が成立つ。

〖証明〗

$\vdash \alpha$

$\Leftrightarrow$ $\phi(\alpha)$がＦＰＣの定理 （∵定理４．５．１）

$\Leftrightarrow$ $\models \phi(\alpha)$ （∵Gödelの完全性定理）

$\Leftrightarrow$ ＦＰＣの任意の解釈$\mathscr{I}$に対し、$\mathscr{I} \models \phi(\alpha)$ （∵恒真性の定義）

$\Leftrightarrow$ ＴＬのすべての解釈$\mathscr{I}$に対し、$\alpha \in \mathscr{I}$ （∵定理４．５．２）

$\Leftrightarrow$ αは valid （∵validityの定義）

■

【定理 ４．６．２】 ［無矛盾性定理］

ＴＬにおいては、$\vdash \beta$ かつ $\vdash (\sim \beta)$ となるような$\beta \in \Lambda$は存在しない。

〖証明〗 そのようなβが存在したと仮定する。完全性定理より、βと$(\sim \beta)$は共にvalid となる。従って、任意のＴＬの解釈$\mathscr{I}$に対して、$\beta \in \mathscr{I}$ かつ $(\sim \beta) \in \mathscr{I}$ となる。ＴＬの解釈の定義から、任意のグラウンド代入σに対して、$\beta\sigma \in \mathscr{I}$ かつ $(\sim \beta)\sigma \in \mathscr{I}$、すなわち、$\beta\sigma \in \mathscr{I}$ かつ $(\sim \beta\sigma) \in \mathscr{I}$ が成立つ。しかし、$\beta\sigma$は自由変数を含まない（∵σはグラウンド代入）ので、ＴＬの解釈の定義の条件(i)：$(\sim \alpha) \in \mathscr{I}$ iff $\alpha \notin \mathscr{I}$ と矛盾する。 ■

５． むすび

本稿では、人が扱っているような知識、とりわけ、自然言語で表されるような知識を表現するのに適していると思われる、新しい論理体系“Tuple Logic”を提案した。ここでは、それの完全性、無矛盾性、一階述語計算との関係、を示した。

ＴＬを用いると、モンタギュー文法などで高階の関数に対応づけられていたような統語カテゴリー（例えば、副詞（句）前置詞（句）など）に属する単語や句、あるいは複文などが、タプルの構造パターンとして簡潔に表現できる。しかも、解釈（意味論）も非常に簡単である。

しかし、日本文や英文などの実際の自然言語文の意味を、タプルによって具体的にどのように表現すればよいか、という問題が残されている。文の意味を述語論理式で表そうとして場合でも、その表現法には大きな自由度があるが、タプルの場合には、述語論理式よりもさらに表現法の自由度が大きいように思える。しかも、表現法の良し悪しによって推論のしやすさが左右される。

紙面も尽きたので、この問題も含めたＴＬの具体的応用については、稿を改めて議論することにしたい。

参考文献

［１］Allwood, J., Andersson, L. G., and Dahl, Ö.：“Logic in Linguistics”, Cambridge Univ. Press (1977)（公平，野家 訳：“日常言語の論理学”，産業図書，1979）

［２］Boolos, G. S. and Jeffrey, R. C.：“Computability and Logic”, Cambridge Univ. Press (1974)

［３］Frege, G.：“Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens”, Halle (1879)

［４］Gödel, K.：“On undecidable propositions of formal mathematical systems”, Note by S. C. Kleene and B. Rosser on lectures at the Institute for Advanced Study, Princeton (1934)

［５］Mendelson, E.：“Introduction to Mathematical Logic (2nd ed.)”, D. Van Nostrand (1979)

［６］Montague, R.：“The proper treatment of quantification in ordinary English”, in Formal Philosophy, ed. R. H. Thomason, pp.247-270, Yale Univ. Press (1974)