

経験科学としての生成文法を目指して

オーガナイザ：上山あゆみ
(九州大学)

0. ワークショップの趣旨説明

チョムスキーが生成文法を自然科学の一つ、すなわち、経験科学として位置づけたということは、比較的広く知られているかもしれない。しかし、一つ一つの研究をどのように行っていけば「経験」科学となるのかという具体的な点については、必ずしも衆目の一致するところではない。それどころか、そもそも理論言語学が経験科学として研究可能なのか、と懐疑的な意見すらよく耳にする。

言語研究の経験的基盤に対して疑問を抱かせる最も大きな要因としては、文法理論のデータとなるべきものが、提示された文に対する内省的判断という極めて主観的なものだということがあげられるだろう。実際、同じ文を提示しても、人によって判断が分かれたり、ときには、同一人物であっても聞き方やタイミングによって判断が異なる場合もある。このような状況に直面すると、生成文法研究の拠って立つところは何なのかということに本質的な疑問を抱いたとしても不思議はない。

本ワークショップでは、生成文法研究の現状の一端を紹介しつつ、この問題に対する意見を交換していきたい。提題者は、次の3人である。

上山あゆみ (九州大学) ... モデル全体の概説と、問題点の提起

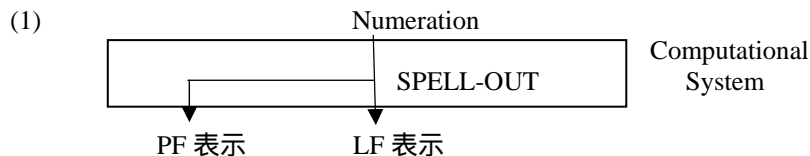
Hajime Hoji (南カリフォルニア大学) ... 統語論研究の典型例の紹介と方法論的議論

峯島宏次 (慶応大学大学院) ... 意味論・語用論研究とこのモデルとの関連について

以下に、上山の発表内容に関する資料を掲げておく。ワークショップでは、細かい点に立ち入る時間がないだろうと思われるので、なるべく要点だけにしぼって発表する予定である。

1. Computational system

生成文法の追究の対象となっているのは、「文法」という、その言語の話者が共有しているシステムである。もう少し具体的に言うと、生成文法を中心にあるのは、computational system と呼ばれる計算アルゴリズムである。これは、いくつかの単語の集合 (numeration) を入力とし、それらをつなげて1つの構成素にしたり (Merge)、語順を変えたり (Move)、組み合わせ可能かどうかをチェックしたり (Agree)、などの操作を行ったのちに、入力された単語がすべて組み込まれた構築物としての表示を2つ—PF表示とLF表示—出力するアルゴリズムである。このPF表示とLF表示は独立に作られるのではなく、Numeration から出発した作業の前半は共通で、SPELL-OUT と呼ばれるタイミングのあとで2つに分岐すると考えられている。



ことばには、もちろんのことながら、文法以外の側面もあり、その中には同じ言語の話者であっても共通でないものもあるだろう。極端な例を出すならば、詩をよんで感動するというのも明らかにことばの持つ働きの一つが関わっているだろうけれども、どのような詩をよめば感動するかということは、その言語の使い手に共通していることではない。したがって、「詩をよんで感動する」という現象そのものは、明らかに、文法というシステムに由来するものではない。このように、言語に関わる活動全体を視野に入れると、アルゴリズムとしてモデル化することが困難な部分がたくさんある。だからこそ、チョムスキーは、説明対象を(言語運用全般ではなく)(1)の計算システムに関わる部分に限ることによって、「科学としての言語学」という生成文法のプロジェクトを打ち立てた。チョムスキーは、文法というシステムからの出力があるかどうかを指す「文法性 (grammaticality)」という概念を導入した。定義上、これは、その言語の使い手に共通しているはずであり、それを説明するべく、

文法の仮説を構築し、それを検証していくのが生成文法研究である。

- (2) a. **文法性 (grammaticality) :**
computational system から出力される表示であるかどうか
b. **容認可能性 (acceptability) :**
その文が容認できると思うかどうか

しかし、実際の研究活動においてデータとなるのは、研究者自身の内省であれ、他人の反応の観察であれ、どちらにしても、それは言語運用の結果である。これがパラドキシカルな状況を作り出している。すなわち、容認可能性の判断という現実のデータと、文法性という理論的な概念との対応の問題である。

- (3) 生成文法の方法論上の問題：
容認可能性の分布をデータとして、computational system の仮説を構築し、検証していかなければならないということ

特に問題になるのは、文の容認可能性の判断が全員一致になる場合というのは、きわめて稀であるという事実である。容認可能性の判断が話者間で一致しないという問題点は、従来からよく知られていることである。この問題に対する典型的な返答は、「だからといって、文法というシステムの存在や、その中のメカニズムについての直接の反証となるわけではない」というものであった。確かに、その言明そのものは正しいと言わざるをえない。文を聞き、理解し、判断する、という作業の中には、計算体系としての文法以外のモジュールも多数関わっているため、様々なモジュールの働きのせいで容認可能性の判断に揺れが出ている可能性が十分に考えられるからである。しかし、このように言い放ってしまうことそのものが、経験科学としての生成文法の位置づけを最も危機的にしているという認識が必要であろう。もし、文法性という概念と経験的なデータとの関連が恣意的になってしまえば、そもそも文法というシステムの研究は反証可能でなくなり、文法を経験科学の対象にすることができないということになりかねない。

つまり、問題は、容認可能性の判断と文法性がどのように対応すると考えるべきか、その指針が必ずしも明確でないということである。

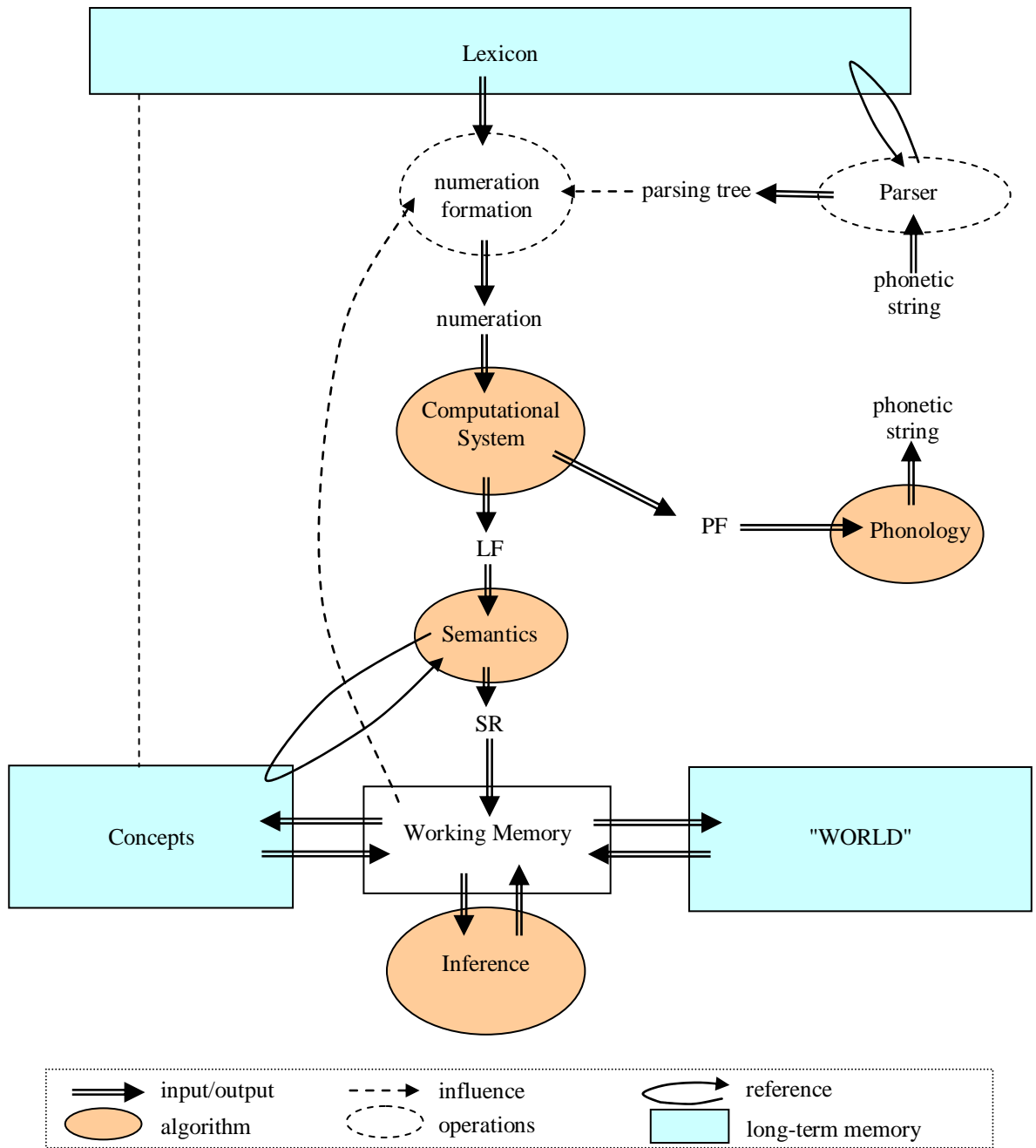
- (4) 課題：
容認可能性の判断という現実のデータと、文法性という理論的な概念とは、どのように対応していると考えられるべきなのか？

現状では、その判断は、個々の研究者にまかされている。研究者によっては、容認可能性と文法性の関係を、どのような根拠にもとづいて、どう結論づけたかを明らかにしている場合もあれば、その問題に関して、ほとんど述べられていない場合もある。後者は、いわば、実験の具体的な手法が明らかにされないまま、結果が述べられ、それに対する考察が展開されている状況であると言っても過言ではない。容認可能性というデータと文法性という説明対象となる概念との関連について、理論的基盤を整えることが非常に重要な課題なのである。

1つの可能性として、言語運用（特に、研究のデータとなる内省について）の全体的なモデルを提示し、計算システムがどこにどのように関わっているのかということを示しながら分析を進めていく、という方法を提案したい¹。言語現象の中には、明らかに、計算システムの外の仕組みが大きく関わっていると思われる側面もある。しかし、だからといって、そういう現象が計算システムの追究に無関係であるということにはならない。内省データが言語運用の結果であり、「計算システムのみが関わっているデータ」というものがありえない以上、計算システムの中で起こっていることに加えて、その外の仕組みも影響を与えているということが十分に考えられるからである。

資料の最後のページの図1がここで想定している言語運用のモデルである。図1のモデルのポイントを簡単に説明しておく。

¹ Chomsky (2000)からも、チョムスキーは、この計算システム（と同等の働きをする仕組み）が実際に私たちの脳の中で作動していると考えていることがうかがえる。しかし、このシステムが言語運用の中でどのように働いているのかということについては、必ずしも具体的なモデルが提案されているわけではない。



2. encoding 専用のアルゴリズム

自発的な文の生成の場合には、何らかの方法で Lexicon 中の語彙から numeration が形成されると考えておく²。

(5) numeration formation :

Lexicon から単語を選び出し、numeration を形成する操作

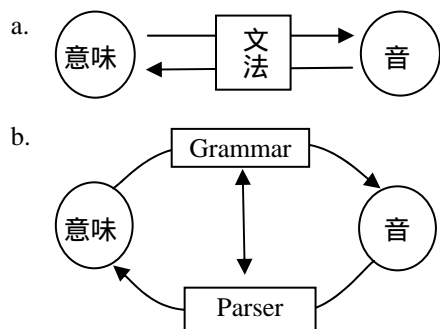
それに対して、例文判断のように、提示された音連鎖 (phonetic strings) がある場合には、その音連鎖に単語認識や統語解析などの作業を行い、その結果が numeration formation に影響を与えていると考えたい。ここでは便宜的に、その一連の操作の総体に対して Parser という名称を与え、その出力の結果をまとめたものを parsing tree と呼んでおく。

(6) Parser :

単語認識した結果に対して統語解析の作業を行い、numeration formation に対して情報を (わかった範囲で) 受け渡すもの

図1で表されている言語運用のモデルには、(1)の計算システムが直接組み込まれている。「文という音」のベースとなる PF 表示と「文の意味」のベースとなる LF 表示は、どちらも計算システムの出力であり、一方が他方の入力になっているわけではない。生成文法の言語観は、「言いたい意味」が「文という音」に変換されたり、「文という音」から「その文の意味」が伝わる、というような従来の言語観とは根本から異なっている。

(7) ここで仮定していない考え方 :



言語の研究においては、文を聞いたり読んだりする場合、numeration の形成を仮定しないモデルが想定されている場合も多い。つまり、統語解析器である Parser によって意味解釈が可能になる、という考え方である³。この考え方をつきつめると、文の生成に関わるメカニズムと、文の理解に関わるメカニズムは、まったく別のものであるということになる。確かに、文の生成と文の理解というのは、かなり異なった行為であるには違いないが、しかし、そこに関わる「文法の知識」は共通であると考えたい。しばしば、「Parser は文法を参照する」という仮定が言及される場合もあるが⁴、文法の主体が規則の集積であった標準理論 (Standard Theory) の場合ならばともかく、numeration を入力として LF と PF を出力するという動的なアルゴリズムとしての計算システムを仮定する以上、Parser がそれを「参照する」ことは不可能であろう。つまり、文法の中核をアルゴリズムとしてとらえる限り、この計算システムそのものが、文の生成にも文の理解にも関わっていると仮定せざるをえない。

parsing tree については、今後、より具体的に論じていく必要があるが、現時点では、認識された各

² この操作 (numeration formation) については、厳密なアルゴリズムとしてのモデル化は不可能であると考えている。

³ Parser というものは音連鎖を入力とするものであるから、このモジュールにおいて意味解釈が可能になるということは、いわば、PF から LF に変換するメカニズムの構築が可能だと主張することになるが、筆者は computational system 中の操作は可逆的なものではないと理解しているので、このような試みには無理があるのではないかと考えている。特に、LF 移動を仮定する必要がある文の解釈などは、いったい Parser がどのように対処しうるのが、きわめて難しい問題が提示されるだろうと理解している。

⁴ たとえば、坂本 (1998: 45-46) にその考え方が詳しく解説されている。

単語がどのような順番に並んでいるかということと、解析の結果得られた構造に関する情報とが表されているものを想定している。numeration においては、音として具現される単語だけでなく、少なからぬ数の機能範疇や抽象的な素性が必要となる。これらが numeration formation において単に恣意的に選択されるのならば、聞いたとおりの文が出力されるような numeration を形成するのは不可能に近いかもしれない。しかし、parsing tree において、どの単語がどこに係っているかという関係等が示されれば、numeration にどのような要素が必要かということ、かなりの精度でしぼりこむことができるはずである。

parsing tree が計算システムの出力としての表示と等しいものである必要はまったくない。LF 表示や PF 表示を出力するのは計算システムの役割なのであるから、統語解析においては、numeration を効率よく形成する役割が果たせれば十分である。そもそも、numeration formation というものは、自発的な文の生成の場合、Parser の力を借りることなく作業が完遂できるモジュールなのであるから、文を聞いたり読んだりした場合であっても、音連鎖から得られる情報に 100% 依拠していると考えする必要はない。Parser の出力に基づいても numeration が一意に決定されないという状態は、むしろ普通であろう。足りない情報は、numeration formation が自由に補って、numeration を形成するのである。

いったん、numeration が計算システムに入力されれば、そのアルゴリズムに従って PF 表示と LF 表示が得られる。そこで、出力された PF 表示から派生される音連鎖と、聞いたり読んだりした音連鎖の違いが気にならなければ、その LF 表示に基づいた解釈がその文の「意味」として知覚されると考えたい。つまり、私たちは、耳で聞いた文と同じ（だと思っている）文を作ることによって、元の文の内容をくみ取ろうとしているという仮説である。伝えたい内容を「音」という形で運ぶということは、一種の暗号化と考えてもよい。この表現を用いて上の考え方を言い換えるならば、私たちの頭の中にある言語のシステムは、暗号化（encoding）はできるが、直接その暗号を解読（decoding）することはできず、同じような暗号を自分で作ってみることによって、発話者の意図を推測する方式になっていることになる。

computational system の入力には numeration なのであるから、文法を機能させるためには、まず、numeration を作ることにどうしても必要である。生成文法の立場に立つならば、これは、自発的に文を生成する場合だけでなく、文を聞いたり読んだりした場合についても同様のはずであるが、このことは、生成文法理論の研究において、ほとんど意識されていないと言ってもいいだろう。しかし、生成文法の重要なデータが提示された文の容認可能性判断であり、それには、例外なく、聞いたり読んだりした文を脳内で表示するという作業が関わっている以上、その過程を慎重に考察することは非常に重要である。上で述べてきた仮説が正しいならば、提示された文 A を判断しているつもりでも、判断の感覚を生み出すものは、常に、自分の頭で生成しなおされた文 B でしかなく、A と B が同じ「文」とであるという保証は、極端に言えば、どこにもない。日常生活においては、相手が言った文 A に対して、自分が考えた文 B で相手の言ったことを「理解」し、その A と B が文としては大きく異なっている場合も、少なからずあるだろう。これに対して、文法研究における「文の判断」という作業では、その A と B が可能な限り同一であることが求められている。提示された文は、computational system にとっては、厳密な意味での「入力」ではなく、入力（すなわち numeration）に影響を与える「刺激」に過ぎないということを常に意識しなければならない。

このように考えれば、文の判断が「揺れる」ように感じられることがあることも納得できる。もし、提示された音連鎖がそのまま文法の「入力」になるのならば、一人の話者の中で異なった判断が出るというのは非常に不思議なことである。しかし、1つの刺激文 A に対して、頭の中で $B_1 \cdot B_2 \cdot B_3 \cdot B_4$ という、いろいろな文が生成されるとすれば、それぞれに対する感覚が異なっても当然である。その話者が、 $B_1 \cdot B_2 \cdot B_3 \cdot B_4$ のどれも A と「同じ文」とであると認識してしまえば、「A に対する判断」が「揺れる」ように知覚されるわけである。

3. 意味解釈

LF 表示とは、（PF 表示と同様）単語が構造化された表示である。図 1 では、この LF 表示が Semantics というモジュールの入力となっている。ここで想定している Semantics というアルゴリズムの動きは、次の通りである。

- (8) Semantics の働き：
 - i. LF 表示の構造はそのままに、それぞれの単語を、対応する概念に置き換える⁵。

⁵ この論文では、詳しいことについては述べられないが、すべての単語がいわゆる「概念」と結びついているわ

- ii. 姉妹関係になっている概念を合成する⁶。
- iii. それらの概念の組み合わせさせた構築物が意味表示 (SR) であり、これがいわゆる命題に相当する。

たいていの単語には、それに該当する概念がある。それらの概念は、図1の「Concepts」の中に蓄えられているとする。仮に、すべての概念に通し番号がついていると仮定しよう。しばしば、単語というものは、音韻素性と形式素性と意味素性の束である、という言い方がされるが、その通し番号がその場合の意味素性に相当すると考えておきたい。SRを形成するのは Semantics の働きであるが、その構造は LF 表示に基づいているので、SR の構造は間接的に計算システムが形成することになる。SR というものは、概念からできた構築物であり、全体で、命題もしくはある種の情報を表す⁷。これが、作業記憶領域 (Working Memory) に入って、その人が長期記憶として蓄えている情報と合わせて「理解」されたのちに、長期記憶へと運ばれていく。

この「理解」という作業の中には、たとえば次の(9i,ii)などが含まれる⁸。

- (9) i. その新しい情報を組み込むことによって、既存の情報を変更する必要があるかどうかを確認する。
- ii. 新しい情報と既存の情報との関連性をなるべく打ち立てる。

このような作業が可能になるためには、新しく入ってきた命題に関連する命題が長期記憶から作業記憶の中に呼び起こされなければならない。しかし、何から何を「連想」するか、ということに対するアルゴリズムが立てられるとは考えにくい。この作業をつかさどっているのは、神経ネットワークの複雑な所業であると考えざるをえないであろう。ただし、予測不可能なのは、どういう命題が呼び起こされるか、ということだけである。仮に、命題 P と命題 Q が呼び起こされた状態を想定すれば、例えばその2つが矛盾の関係にあるかどうか、といったことに関する私たちの判断は、予測可能な現象である。そこで、そのような命題間の計算は、アルゴリズムにしたがって明示的に行われていると考えたい。図1の Inference というモジュールは、いわゆる論理的な推論規則が蓄えられている部署であり、ここで、矛盾のチェックや推論などが行われると仮定している⁹。

さて、図1では、長期記憶として、「Lexicon」「Concepts」「"World"」の3つの領域が描かれている。これらは、それぞれ要素が異なるデータベースである。

- (10) Lexicon ... (厳密に言語としての) 単語とその特性に関するデータベース。
- (11) Concepts ... 部品となる概念のデータベース。ほとんどの場合、Lexicon 中の単語と対応関係がある。
- (12) "World" ... (その人が認識/想定している) 外の世界の(特定の)出来事/状態を、命題の集積として表現しているデータベース¹⁰。

けではない。たとえば、numeration の要素には、演算子に対応する要素もあると考えている。そういう場合も考慮にいれると、(8i)は「LFの構造はそのままに、それぞれの単語を、SR要素に置き換える」と述べたほうが正確であるが、本文では、わかりやすさを考慮した述べ方にしている。

⁶ 各概念を組み合わせる典型的な方法としては、いわゆる function application が考えられるが、それが唯一の方法であるとは仮定していない。(8ii)において、合成 (composition) という語は、もっとも広義のカバータームとして用いている。

⁷ 一般的には、Semantics とはその文の真理条件を定める仕組みであるという位置づけが多いが、ここでは、その考え方はとっていない。

⁸ このあたりの仕組みについても、この論文では詳しく述べることができないが、基本的に齊藤(2006)で展開されている知識モデルを想定している。齊藤(2006)は、ラシヤやヨウダの証拠推量の用法を、(9i, ii)の観点から考察したものである。

⁹ ただし、この Inference モジュールは、いわゆる「帰納的推論」を含まない。帰納的推論は、厳密にアルゴリズムで表現できるものではないと考えているからである。確かに、過去の状況をつづさに吟味し、反例がないことを確認した上で、帰納的に一般化を導き出すこともあるだろうが、極端な場合、特定の1つの出来事や経験に基づいて一般的な命題を「推論」することも、私たちはしばしば行ってしまうものである。

- (i) 傘を忘れたときに限って、雨が降る。
- (ii) フィンランド人は勤勉だ。

¹⁰ このデータベースには、一般に「知識 (knowledge)」と呼ばれるものと「単なる信念 (belief)」と呼ばれるものの両方が含まれている。knowledge と belief というものを二値的に区別することはできないと考えているからである。そのかわりにここでは、齊藤(2006)で提案されている通り、その情報の出処 (たとえば、直知によるものか、誰かから聞いたのか、もしくは、どれかの命題から自分が推量したのか、等) をそれぞれの命題にタグ

その SR が外界で起きた出来事 / 状態を述べたものであるならば、その情報は (9) の過程を経た後) 直接、"World" に登録される。しかし、私たちが文で表現することは、必ずしも外界における事象の報告だけとは限らない。"World" というものを (その人が認識 / 想定している) 外界の総体と位置づける以上、特定の出来事 / 状態を表現しない命題は、"World" には登録されない。その場合、その命題においてそれらの概念が組み合わされたということが、単に Concepts というデータベースのネットワークに吸収されるだけであると考えている¹¹。Concepts 中の概念どうしの「連想」関係の強度に影響を与えるが、"World" に書き込まれる情報とは異なるということである¹²。

4. まとめ：生成文法における「文法」という概念

初期の生成文法では、「すべての文法的な文を、そして、それだけを生成する仕組み」を「文法」と呼び、その仕組みの追究というものを生成文法研究の目標であると位置づけた。この時点での (特定言語の) 「文法」とは、句構造規則と変形規則の総体を指すと言っていいだろう。これは、いわば、その言語の「構文」を形づくるものであるから、一般的な意味での「文法知識」という概念とほぼ一致する。

言語によって、その「文法」は異なるわけであるから、もちろん、それらの句構造規則や変形規則は生得的ではなく、経験によって習得されるものである。これに対して、生得的な知識は普遍文法と呼ばれ、これが「文法知識」の発達の初期状態に相当すると考えられた。普遍文法は、句構造規則の一般型や変形に対する一般的制約を含んだものであるが、実際の規則は習得しなければならないものであるため、普遍文法だけでアルゴリズムができていたわけではなかった。

しかし、その後、この考え方を押し進めていこうとしている間に、いろいろと概念的な不具合が指摘されるようになり、結果的に、句構造規則という考え方が全廃され、(1) のような Computational System が提案された。現在の生成文法において中核の働きをしているのは、この Computational System であるから、このシステムを指して「文法」と呼ぶのは自然の成り行きかもしれない。しかし、Computational System と一般的な意味での「文法」との間には、かなり大きな乖離がある。

(13) (初期の生成文法における) 句構造規則 + 変形規則

- a. 経験によって習得しなければならないもの
- b. 言語によって異なっている
- c. どのような表示が文法的であるかが、比較的直接述べられている。

(14) (現代の生成文法における) Computational System

- a. 生得的なもの
- b. 言語普遍的
- c. それ自体は単なるアルゴリズムなので、結果的にどのような表示が出てくるのかは、実際に、そのアルゴリズムを動かしてみないとわからない。

そのかわりに、(13) のような働きをになうことになったのが Lexicon である。現代の生成文法では、いわゆる「構文的な知識」は、すべて Lexicon 中の何らかの語彙の特性としてとらえていくことになる。語彙の特性というものは、個々人が経験に基づいて習得するものなので、多少の個人差がありうる。したがって、computational system の研究においては、語彙の特性の個人差を捨象する方法が必要なのである。

として付しておくという形式を想定している。いわゆる knowledge としての命題 P と単なる belief としての命題 P の違いは、この出処タグの違いとして相対的に表示されることになる。

¹¹ 作業記憶に何が呼び出されるかということがアルゴリズムでとらえられないことであると考えているのと同様、Concepts 中の概念間のネットワークも非常に入り組んでおり、その関係の変化の仕方は、アルゴリズムで表現できるものではないと考えている。

¹² 脚注7でも述べたように、ここでは Semantics というモジュールを真理条件に関連づけていない。その理由の1つは、そもそも、真理条件が定めうる命題というのは、"World" に登録されるべき命題のみだと考えているからである。個人の頭の中の "World" に情報として蓄えられている「世界」と現実の「世界」との間に対応関係があつてほしいというのは、確かに人間の傾向として認められるので、その働きをになっているモジュールも存在していると考えているが、それは言語というものの働きとは独立のものなので、ここでは議論の対象にしていない。