



Kobe Shoin Women's University Repository

Title	計算機上で動作する日本語文法の構築 Construction of a Working Japanese Grammar on Computers
Author(s)	橋本 力 (HASHIMOTO Chikara)
<i>Citation</i>	Theoretical and applied linguistics at Kobe Shoin, No.6 : 41-64
Issue Date	2003
Resource Type	Bulletin Paper / 紀要論文
Resource Version	
URL	
Right	
Additional Information	

計算機上で動作する日本語文法の構築*

橋本 力

Construction of a Working Japanese Grammar on Computers

HASHIMOTO Chikara

Abstract

There are two kinds of linguistics that seek models of we compute a language: Theoretical linguistics and Natural Language Processing. The former should be considered a scientific study, whereas the latter is an engineering problem or practice. Reflecting their characteristics, theoretical linguistics is not strictly speaking formalized nor does it have comprehensive coverage, which any system of linguistic computation is expected to have, and NLP fails to generalize to new phenomena like a linguistic theory. We are developing an HPSG-based Japanese grammar that is executable as a computer program as well as capable of expressing linguistic generalizations, so that it would constitute an interface between theoretical linguistics and NLP. We have currently implemented a grammar that is able to deal moderately well with basic phenomena, with some principles assumed within HPSG explicitly expressed. In this paper, we report on the details of this grammar and discuss its influence of it on theoretical linguistics and NLP.

言語の計算体系を研究する分野として理論言語学と自然言語処理があり、前者は科学としての言語学、後者は工学としての言語学と考えられる。それぞれの学問としての性格に対応して、理論言語学には計算体系としての厳密性、応用に耐えるだけの網羅性が、自然言語処理には言語理論としての妥当性、記述の一般性が欠けている。本研究では2つのアプローチの橋渡しとなれるような、計算機上で動作する言語学的に妥当な日本語文法をHPSGに基づいて構築する。現在までに我々は、基本的な文を扱えるだけの文法を、HPSGで想定されているいくつかの原理を明示的に抽象化した形で実装した。本稿ではこの文法の詳細を報告し、さらに、本研究が理論言語学と自然言語処理に与える影響を考察する。

Theoretical and Applied Linguistics at Kobe Shoin 6, 41-64, 2003.

© Kobe Shoin Institute for Linguistic Sciences.

*本稿で述べる文法は "<http://sifs.shoin.ac.jp/~chashi/talks/6>" から入手できる。

1. はじめに

1950年代チョムスキーにより始められた理論言語学は、自然言語の規則性を厳密な数学的道具立てを用いて記述していた。その成果は、当時機械翻訳研究で盛り上がっていた自然言語処理研究者達に受け入れられ、自然言語処理技術の一部として積極的に活用されていた。しかし、両者の関心や方法論が変わるにつれ、2つのアプローチの溝は大きくなっていった。理論言語学では、規則の体系としての文法を捨て言語獲得の事実の説明を可能にするより抽象的な原理の体系を構築し、自然言語処理にとって重要である処理の問題については全く注意が払われなくなった。その結果、かつてはその大きな特徴であったはずの厳密性と、当初持ち得ていた幅広い現象を扱える網羅性が失われることとなった。一方自然言語処理では、処理対象のデータに現れる言語現象を網羅するためには文法規則を人手で地道に作成するよりもコーパスを用いた統計的手法のほうが現実的で高性能であるとの認識から、ルールベースではなくコーパスベースの技術に関心を向け始め、言語を説明の対象ではなく完全に処理の対象と見なすようになった。つまり自然言語処理では文法の妥当性や一般性などは完全に考慮外となった。

2つのアプローチは、それでもなお、言語の計算体系を研究するという点ではなんら変わらず、説明と処理を両立させる研究が可能である。Head-driven Phrase Structure Grammar (HPSG) はタイプ付き素性構造に基づいて厳密に形式化された文法理論であり、すでに多くの言語の幅広い現象の分析がそれに基づいて行われている。本研究では HPSG に基づいて説明と処理を両立させる日本語文法を構築する。つまり、言語学的な妥当性一般性を維持しつつ、計算機上で動作可能なほどの厳密性と応用に耐えうる網羅性を備えた文法を構築する。現在のところ、基本的な文を扱えるだけの日本語文法を、主辞素性原理、意味的構成性原理などの HPSG で想定されているいくつかの原理を明示的に抽象化した形で実装し終えている。計算機上への文法の実装は、スタンフォード大学で開発されている文法開発環境 LKB (Copestake, 2002) を用いて行った。LKB は素性構造で記述された文法が入力されるとそれに基づいて構文解析、文生成を行う。

以下、2. 節で HPSG の概要に触れ、3. 節で本研究で構築した文法の詳細を報告し、4. 節でその文法の実装について述べる。さらに 5. 節では本研究が理論言語学と自然言語処理に与える影響を考察する。6. 節は結論である。

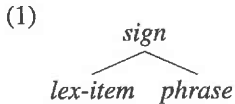
2. HPSG の概要

HPSG は制約に基づく語彙主義の文法であり、文の well-formedness は語彙項目の情報と、適格な言語記号(語や句)の構造に関する、素性構造で記述された制約群により規定される。変形や派生を用いないので語彙項目や制約群が膨大な情報を抱え込むが、それらの間に存在する一般性や個別性がタイプ階層と階層間の制約の継承(上位のタイプの制約が下位のタイプに引き継がれる)という仕組みにより言語学的に有意義な形で整理、抽象化されている。HPSG ではこのタイプ階層が文法の中核を担う。以下では Sag and Wasow (1999) に基づき本研究と関連のある部分にのみ焦点をあて英語のタイプ階層を概

観する。HPSG の詳細については Ginzburg and Sag (2000), Sag and Wasow (1999) を参照。

2.1 sign の階層

階層はまず言語記号 (*sign*) が語 (*lex-item*) と句 (*phrase*) から成ることを宣言する (1)。



sign では (2) の素性が導入される。PHON、SYN、SEM はそれぞれ記号の音、統語、意味情報を担う。

$$(2) \left[\textit{PHON list}, \textit{SYNSEM} \left[\textit{SYN gram-cat}, \textit{SEM sem-struct} \right] \right]$$

SYN に適切なタイプ *gram-cat* では (3) の素性が導入される。

$$(3) \left[\textit{HEAD pos}, \textit{SPR list (synsem)}, \textit{COMPS list (synsem)} \right]$$

SPR と COMPS にはそれぞれその記号の指定部と補語が (もしあれば) 入る。HEAD には品詞に相当するタイプ *pos* が割り当てられる。品詞は *verb*、*prep*、*noun*、*det*、*adv*、*adj*、*conj* が想定されている。各品詞にはそれぞれ特有の振舞いを表す素性の他に素性 MOD が導入されており、その品詞がどういった要素を修飾しうるのが規定されている。例えば *noun* の素性は (4) のようになる。

$$(4) \left[\textit{MOD none}, \textit{ANA} \left\{ +, - \right\}, \textit{CASE} \left\{ \textit{non}, \textit{acc} \right\} \right]$$

つまり *noun* は何も修飾せず、また、*noun* には先行詞を照応するものとしなないものがあり、主格か対格の格を持つ。

SEM に適切なタイプ *sem-struct* で導入される素性は (5) の通りである。

$$(5) \left[\textit{MODE} \left\{ \textit{prop}, \textit{ques}, \textit{dir}, \textit{ref}, \textit{none} \right\}, \textit{INDEX index}, \textit{RESTR list (predication)} \right]$$

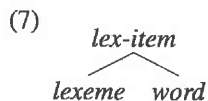
MODE は記号の意味的な機能 (命題、疑問、指示など) を、INDEX はその指示対象 (動詞などであれば状況、名詞などであれば個体など) を、RESTR はその意味の具体的内容を表す。RESTR は *predication* のリストを取るが、*predication* は (6) のような構造になっている。

$$(6) \left[\textit{RELN} \left\{ \textit{love}, \textit{walk}, \textit{give}, \dots \right\}, \textit{SIT} s, \textit{ARG1} \dots, \textit{ARG2} \dots, \dots \right]$$

RELN は関係を表し、ARG 群はその関係に関わる項を表す。SIT は状況を表す。

2.2 *lex-item* の階層

lex-item は *lexeme* と *word* から成る (7)。*word* においては “runs” “ran” “running” は区別されるが *lexeme* においては区別されない。

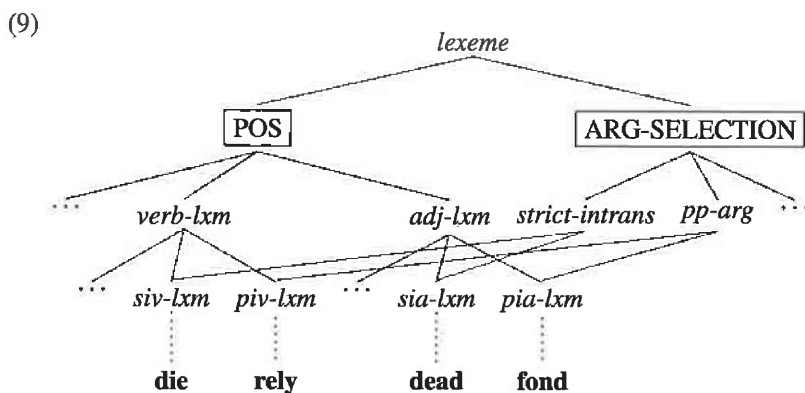


lex-item では項構造を表す素性 ARG-ST が導入される (8)。その値は *synsem* のリストである。

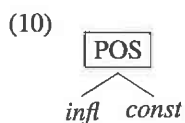


lexeme の階層

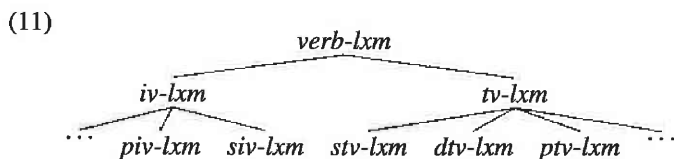
lexeme は品詞と項構造の観点からの多重継承階層により (9) のように表されている。



例えば *siv-lxm* (strict intransitive verb) は *verb-lxm* と *strict-intrans-lxm* の制約を継承している。さらに **POS** 以下を詳しく見ると屈折の可能性により (10) のように分類される。



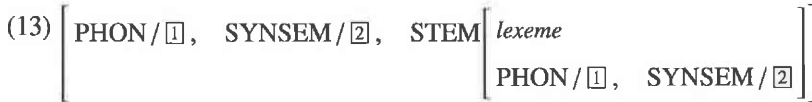
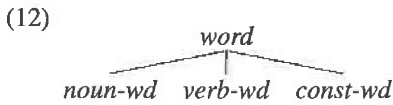
屈折するの (*infl*) は名詞と動詞である。動詞の階層をもう少し詳しく見てみると (11) のように自動詞他動詞の観点から分類されている。



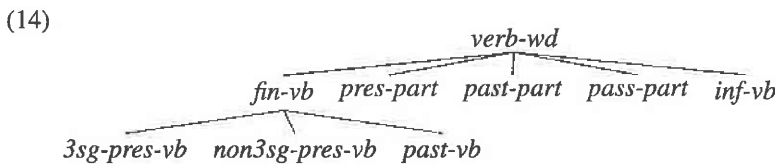
piv-lxm は prepositional intransitive verb、*stv-lxm* は strict transitive verb、*dtv-lxm* は ditransitive verb を表す。

word の階層

word の階層は (12) のようになっており、word には (13) の制約がかかる。



STEM はその word のもととなる lexeme を表し、“/” はデフォルト値であることを示す。つまり (13) はもとの lexeme の PHON 値 SYNSEM 値がデフォルトで word に引き継がれることを意味する。動詞の語階層を詳しく見てみると (14) のようになっている。



これらの動詞の下位タイプが (13) の制約のデフォルト値を上書きし、各下位タイプに適切な制約を生み出す。例えば past-vb は PHON 値を上書きし過去時制を表す形態素を付加するだろう。

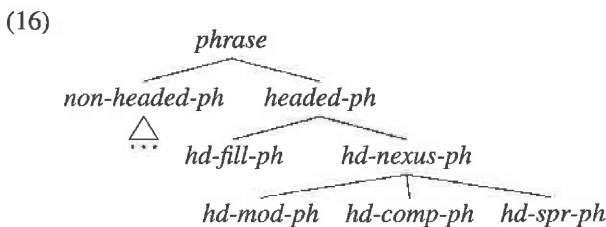
lexeme が word として文中に表出される段階で、ARG-ST 上の項は項具現化原理 (Argument Realization Principle, ARP) により統語的に実現される (15)。



SPR を取らず COMPS のみの場合には $\boxed{1}$ が空となることで対応する。

2.3 phrase の階層

phrase の階層は (16) のようになっている。



phrase の制約として $\left[\text{NHD-DTRS list (sign)} \right]$ が導入される。これは、phrase は全て主辞ではない daughter を取る、ということである。さらに意味的構成性原理 (17) が適用されるだろう。

$$(17) \left[\text{SEM} \boxplus \oplus \dots \oplus \boxminus, \text{DTRS} \left(\left[\text{SEM} \boxplus \right], \dots, \left[\text{SEM} \boxminus \right] \right) \right]$$

headed-phrase の制約として $\left[\text{HD-DTR sign} \right]$ があり、*headed-phrase* は主辞である daughter を取る、ということの意味する。*headed-phrase* にはさらに主辞素性原理 (18)、意味的継承原理 (19)、結合価原理 (20) が適用される。

$$(18) \left[\text{SYN} | \text{HEAD} \boxplus, \text{HD-DTR} \left[\text{SYN} | \text{HEAD} \boxplus \right] \right]$$

$$(19) \left[\text{SEM} | \text{INDEX} \boxplus, \text{HD-DTR} \left[\text{SEM} | \text{INDEX} \boxplus \right] \right]$$

$$(20) \left[\text{SYN} \left[\text{SPR} / \boxplus, \text{COMPS} / \boxminus \right], \text{HD-DTR} \left[\text{SYN} \left[\text{SPR} / \boxplus, \text{COMPS} / \boxminus \right] \right] \right]$$

(18)(19) はそれぞれ、HD-DTR の HEAD の値、INDEX の値が句全体の HEAD 値、INDEX 値になることを保証する。(20) は HD-DTR の SPR 値 COMPS 値がデフォルトで句全体の SPR 値 COMPS 値になることを保証する。これはデフォルトの制約なので下位の制約により上書きされうる。

hd-spr-ph、*hd-comp-ph*、*hd-mod-ph* はそれぞれ (21)(22)(23) のようになる。

$$(21) \left[\text{SYN} | \text{SPR} \langle \rangle, \text{HD-DTR} | \text{SYN} | \text{SPR} \boxplus, \text{NHD-DTR} \boxplus \right]$$

$$(22) \left[\text{SYN} | \text{COMPS} \langle \rangle, \text{HD-DTR} | \text{SYN} | \text{COMPS} \langle \boxplus, \dots, \boxminus \rangle, \text{NHD-DTR} \langle \boxplus, \dots, \boxminus \rangle \right]$$

$$(23) \left[\text{HD-DTR} \boxplus, \text{NHD-DTR} | \text{SYN} | \text{HEAD} | \text{MOD} \boxplus \right]$$

(21)(22) はそれぞれ、HD-DTR が NHD-DTR を指定部、補語として取ることを示す。(23) は修飾句である NHD-DTR の MOD 値が HD-DTR の SYNSEM 値と同じであることを示す。

文の開始記号は (24) にあるように定形動詞の投射で SPR と COMPS が飽和していなくてはならない。

$$(24) \left[\textit{phrase} \left[\text{SYN} \left[\text{HEAD} \left[\textit{verb} \right], \text{SPR} \langle \rangle, \text{COMPS} \langle \rangle \right], \text{FORM} \textit{fin} \right] \right]$$

2.4 自然言語処理から見た HPSG

以下では自然言語処理研究の視点から HPSG の特徴を概観する。

制約に基づく文法 制約に基づく文法では、変形もしくは派生といった操作的な概念を極力排除し、制約という非操作的な概念により文の well-formedness を規定する。理

論的な観点からは変形を用いるべきかどうかは議論の分かれるところかもしれないが、自然言語処理の観点からは変形を用いるべきでないことは明らかである。変形などの操作的な概念を取り入れた文法を理論に忠実に計算機に実装するのは容易ではない。また、文法を拡張していく際に文法内に矛盾を生じさせやすい。一方制約に基づく文法は多くの場合、理論にほぼ忠実に実装でき、文法のメンテナンスも容易である。そして変形や派生といった概念のない一つの層からなる文法は、言語の産出と理解を同じように扱える。

表層指向 HPSG では知覚可能な表層情報のみに基づいて文法が構成されている。つまり変形文法で仮定されているような空範疇あるいは空の語彙は HPSG では用いられない。これもまた理論的な観点からは一部の理論言語学研究者にとっては議論の余地のあるところかもしれないが、自然言語処理の観点からは大きな利点であることは明白である。知覚できる情報以外の存在を仮定した言語処理は、計算量的に非現実的なものになり、また処理方法も複雑なものとなる。

タイプ付き素性構造 HPSG の厳密な形式化はタイプ付き素性構造という数学的構造によって達成されている。これにより HPSG の計算機上への実装は十分可能なものとなる。初期の理論言語学で用いられていた単純な句構造規則は、自然言語の豊かな構造を表現するには不十分であるとして捨て去られたが、タイプ付き素性構造に基づく形式化はその厳密性において初期の単純な句構造規則と変わらず、かつ、その表現能力において変形文法に見劣りすることは決してない。

様々な言語情報の統一的記述 HPSG では形態音韻、統語意味、文脈に関する言語情報をタイプ付き素性構造により統一的並行的に扱う。これは変形文法などには観られない魅力である。理論的な利点もさることながら、自然言語処理においてもこのような様々な言語情報を同時に扱えることは有益である。例えば、会話の流れを考慮した音声機械翻訳システムを構築しようとする場合、統語情報の他に、明らかに音韻情報と文脈情報も必要になる。従来自然言語処理で使われてきたあまりに ad hoc な文法では音韻、文脈情報はおろか意味情報すら扱うのは困難である。

Constructionism 言語表現には確かに規則性が存在し、決してランダムな文字列ではない。しかし一方で、あらゆる言語表現を一般的な原理から導き出そうとすると必ずその原理からはずれてしまう construction に出くわす。しかしこれらを”peripheral”な現象として切り捨ててしまってもは应用到耐えうる文法は構築できない。様々な construction の共通性と個別性を整合的に扱う理論的枠組みが整備されており、最近の HPSG にもその要素が取り入れられている (Ginzburg & Sag, 2000)。

3. 日本語文法

本研究で構築した日本語文法の詳細について述べる。

3.1 *sign* の階層

タイプ階層最上部は (1) と、*sign* で導入される素性は (2) と、SYN に適切な素性 *gram-cat* は (3) と同じである。品詞は名詞 *n*、動詞 *v*、後置詞 *p*、形容詞 *adj*、副詞 *adv* を想定している。*adj* と *adv* のスーパータイプとして品詞 *m* を設定している。全ての品詞タイプには素性 MOD が設定される。現在 *n* と *v* は $\left[\text{MOD} \langle \rangle \right]$ であり、*adj* は $\left[\text{MOD } n \right]$ 、*adv* は $\left[\text{MOD } m \right]$ が $\left[\text{MOD } v \right]$ 、*p* は $\left[\text{MOD } v \right]$ である。*n* には MOD の他に以下のような素性が割り当てられる。

(25) $\left[\text{GR } gr, \text{ SFFX } sffx \right]$

GR はその名詞の文法関係を表し、SFFX は格を表す。

意味論の形式化は、現在のところ、理論的に動機づけられたものというより LKB による実装の簡易さを考慮したものとなっている。SEM に適切なタイプ *sem-struct* では Sag and Wasow (1999) のような MODE を用いない。INDEX は Sag and Wasow (1999) とほぼ同じで、動詞などであれば状況 (*event*) を、名詞などであれば個体 (*object*) を指示する (26)。

(26) $\left[\text{INDEX } index, \text{ RESTR } list (predication) \right]$

RESTR のリストの値となるタイプ *predication* は (6) とほぼ同じであるが、該当する RELN の項 (ARG) の数により *arg1-predication*、*arg1-2-predication*、*arg1-2-3-predication* があり、項をそれぞれ 1 個、2 個、3 個取る。*predication* には必ず ARG0 が存在し、必ず INDEX 値と同一視される。つまり ARG0 は動詞などであればその状況を、名詞などであればその指示対象である個体を値として持つ。意味論的枠組みは Minimal Recursion Semantics (MRS) に依拠しており、より詳しい情報は Copestake (2002), Copestake, Flickinger, Sag, and Pollard (1999) を参照。

3.2 *lex-item* の階層

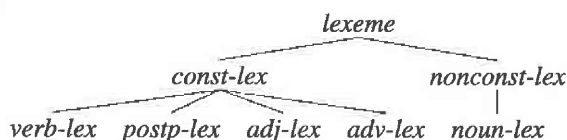
lex-item の階層は (7) と同じである。そして (8) と同じように *lex-item* において ARG-ST が導入される。しかし我々の ARG-ST はただのリストではなく外項 (EXT) と内項 (INT) の観点から構造化されたものである (27)。

(27) $\left[\text{EXT } list (synsem), \text{ INT1 } list (synsem), \text{ INT2 } list (synsem) \right]$

lexeme の階層

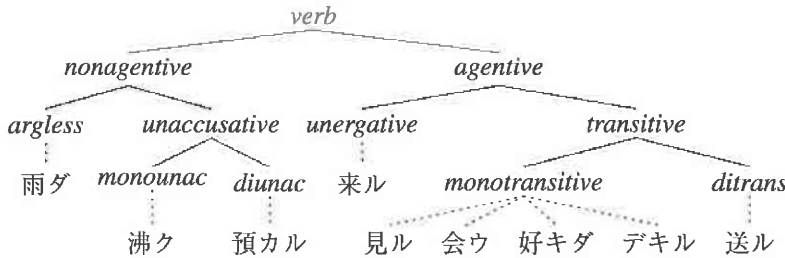
我々の *lexeme* 階層は (28) のように品詞の観点から分類されている。屈折するの (*non-const*) は名詞だけである。

(28)



動詞の階層 動詞の階層は郡司 (2000) に基づいており、(29) のように、項構造の観点から分類されている。

(29)



まず外項のある無しで分類され、その後内項のある無しで分類される。まず *verb* は *const-lex* のサブタイプであり、かつ、HEAD が *v* であること、INDEX が *event* であることが要求される。*nonagentive-verb-lex* では *verb* の制約を継承しさらに外項が空であることが要求される。*argless-verb-lex* では上位タイプの制約を継承した上でさらに2つの内項が共に空であることが要求される。*unaccusative-verb-lex* では1つめの内項が主格の主語名詞句で、かつ、その INDEX 値が ARG1 の値であることが要求される。*monounac-verb-lex* はさらに2つめの内項が空であることが要求される。*diunac-verb-lex* は2つめの内項が目的語名詞句で、かつ、その INDEX 値が ARG2 の値であることが要求される。*agentive-verb-lex* では外項が主格の主語名詞句で、かつ、その INDEX 値が ARG1 の値であることが要求される。*unergative-verb-lex* は *agentive-verb-lex* の制約を引き継ぎ、さらに2つの内項が空であることが要求される。*transitive-verb-lex* では1つめの内項が目的語名詞句で、かつ、その INDEX 値が ARG2 の値であることが要求される。*monotrans-verb-lex* は2つめの内項が空であることが要求される。*ditrans-verb-lex* は1つめの内項が与格、2つめの内項が対格の目的語名詞句、そして2つめの内項の INDEX 値が ARG3 の値であることが要求される。

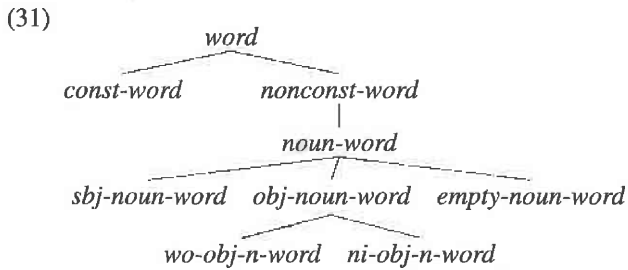
その他の *lexeme noun* は *nonconst* から制約を引き継ぎ、さらに HEAD が *n*、INDEX 値が *object* であることが要求される。項構造は外項内項ともに空である。*postp* は *const* から制約を引き継ぎ、さらに HEAD が *p* であることが要求される。項構造は1つめの内項に屈折してない名詞句を取る。*adj* は *const* から制約を引き継ぎ、さらに HEAD が *adj* であることが要求される。項構造は外項内項ともに空である *adv* は *const* 制約を引き継ぎ、さらに HEAD が *adv* であることが要求される。項構造は外項内項ともに空である。

本文法の項具現化原理は *lexeme* で適用される。(30) のように ARG-ST が外項を持つかどうかにより2つに分けられる。

- (30) a. $\left[\text{SUBJ} \boxed{1}, \text{COMP} \boxed{2}, \text{ARG-ST} \left[\text{EXT} \langle \rangle, \text{INT1} \boxed{1}, \text{INT2} \boxed{2} \right] \right]$
 b. $\left[\text{SUBJ} \boxed{1}, \text{COMP} \boxed{2} \oplus \boxed{3}, \text{ARG-ST} \left[\text{EXT} \boxed{1}, \text{INT1} \boxed{2}, \text{INT2} \boxed{3} \right] \right]$

word の階層

全ての *lexeme* は何らかの語彙規則の適用を受けて *word* となり文中に現れる。まず *word* の階層を (31) に示す。



word の制約は (32) のようになる。

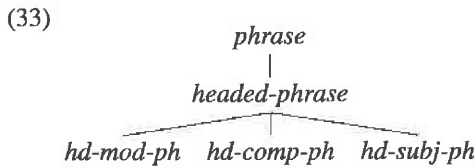
$$(32) \left[\text{SYN} \textcircled{1}, \text{SEM} \textcircled{2}, \text{ARG-ST} \textcircled{3}, \text{STEM} \left[\begin{array}{l} \textit{lexeme} \\ \text{SYN} \textcircled{1}, \text{SEM} \textcircled{2}, \text{ARG-ST} \textcircled{3} \end{array} \right] \right]$$

(32) で述べられているのは、対応する *word* と *lexeme* の間では SYN 値 SEM 値 ARG-ST 値が等しくなる、ということである。*nonconst-word* には屈折変化のある品詞が属し、屈折変化のない品詞は *const-word* に属する。*nonconst-word* に属するのは現在名詞だけである。

lexeme と *word* のマッピングは語彙規則により行われる。*const-lex* に属する *lexeme* とそれに対応する *word* のマッピングは全て定常語彙規則 *const-irule* により行われる。*const-irule* はカテゴリを *lexeme* から *word* に変えるだけで他は何もしない。*nonconst-lex* に属する唯一の *lexeme* である名詞とそれに対応する *word* のマッピングを担う語彙規則は、その名詞の格を表す suffix により主格名詞語彙規則 *ga-noun-irule*、対格名詞語彙規則 *wo-noun-irule*、与格名詞語彙規則 *ni-noun-irule*、無接辞名詞語彙規則 *empty-noun-irule* の4つに分けられる。これらは全て接辞付加にのみ関わり、それぞれ「が」「を」「に」「 ϕ 」を付加する。*ga-noun-irule* が適用された名詞は *sbj-noun-word* へ、*wo-noun-irule* が適用された名詞は *wo-obj-n-word* へ、*ni-noun-irule* が適用された名詞は *ni-obj-n-word* へ、*empty-noun-irule* が適用された名詞は *empty-noun-word* へとマッピングされる。*sbj-noun-word* では接辞が「が」であるだけでなく文法関係が主語であることが要求される。*obj-noun-word* では文法関係が目的語であることが要求され、そのサブタイプである *wo-obj-n-word* では接辞が「を」、*wo-obj-n-word* では接辞が「に」であることがさらに要求される。*empty-noun-word* では文法関係の指定はなく接辞は「 ϕ 」であることが要求される。

3.3 *phrase* の階層

本文法で扱う *phrase* は *hd-subj-ph*、*hd-comp-ph*、*hd-mod-ph* の3つであり階層は (33) のようになる。



headed-phrase のみなので *phrase* の段階で (34) のような素性 DTRS を導入する。

$$(34) \left[H \textit{ list (sign)}, \text{ NH1 } \textit{ list (sign)}, \text{ NH2 } \textit{ list (sign)} \right]$$

H は head daughter、NH1、NH2 は non head daughter を表す。また *phrase* には (17) とほぼ同じ意味的構成性原理が適用される (35)。

$$(35) \left[\text{SEM|RESTR } \boxed{1} \oplus \boxed{2} \oplus \boxed{3} \right. \\ \left. \text{DTR } \left[H|\text{SEM|RESTR } \boxed{1}, \text{ NH1}|\text{SEM|RESTR } \boxed{2}, \text{ NH2}|\text{SEM|RESTR } \boxed{3} \right] \right]$$

headed-phrase には主辞素性原理 (36) 意味的継承原理 (37) 結合価原理 (38) が適用される。

$$(36) \left[\text{SYN|HEAD } \boxed{1}, \text{ DTRS } \left[H|\text{SYN|HEAD } \boxed{1} \right] \right]$$

$$(37) \left[\text{SEM|INDEX } \boxed{1}, \text{ DTRS } \left[H|\text{SEM|INDEX } \boxed{1} \right] \right]$$

$$(38) \left[\text{SYN} \left[\text{SUBJ}/\boxed{1}, \text{ COMP}/\boxed{2} \right], \text{ DTRS} \left[H|\text{SYN} \left[\text{SUBJ}/\boxed{1}, \text{ COMP}/\boxed{2} \right] \right] \right]$$

hd-subj-ph、*hd-comp-ph*、*hd-mod-ph* はそれぞれ (39)(40)(41) のようになる。

$$(39) \left[\text{SYN|SUBJ} \langle \rangle, \text{ H|SYN|SUBJ } \boxed{1}, \text{ NH1 } \boxed{1} \right]$$

$$(40) \left[\text{SYN|COMP} \langle \rangle, \text{ H|SYN|COMP} \langle \boxed{1}, \boxed{2} \rangle, \text{ NH1 } \boxed{1}, \text{ NH2 } \boxed{2} \right]$$

$$(41) \left[\text{H } \boxed{1}, \text{ NH1|SYN|HEAD|MOD } \boxed{1} \right]$$

また、本研究の文法では文の開始記号は *phrase* のサブタイプとして (42) のように定義されている。

$$(42) \left[\text{SYN} \left[\text{HEAD } \nu, \text{ SUBJ} \langle \rangle, \text{ COMP} \langle \rangle \right] \right]$$

つまり文の開始記号は動詞の投射で、かつ、SUBJ と COMP が飽和していなくてはならない。

4. 日本語文法の実装

3. 節で述べた日本語文法を LKB 上に実装した。実装の際には LKB による制限のため文法の形式化をいくつか変更した。

4.1 文法開発環境 LKB

LKB (Linguistic Knowledge Building) はケンブリッジ大学で ACQUILEX プロジェクトの一環として開発され (その当時 LKB は Linguistic Knowledge Base の略)、現在スタンフォード大学 CSLI で LinGO (Linguistic Grammars Online) プロジェクトの一環として継続的にメンテナンスされている文法開発用ソフトウェアである。LKB は、図 1 に示したように、タイプ付き素性構造の形式に従った文法が入力されると、その文法にしたがって構文解析 (文字列 → 意味構造、統語構造)、文生成 (意味構造 → 文字列、統語構造) を行う。LKB に与える文法はタイプ付き素性構造の形式に従っていればよく、必ずしも HPSG である必要はない (LKB に付属のサンプル文法には Categorical Grammar の実装の例がある)。

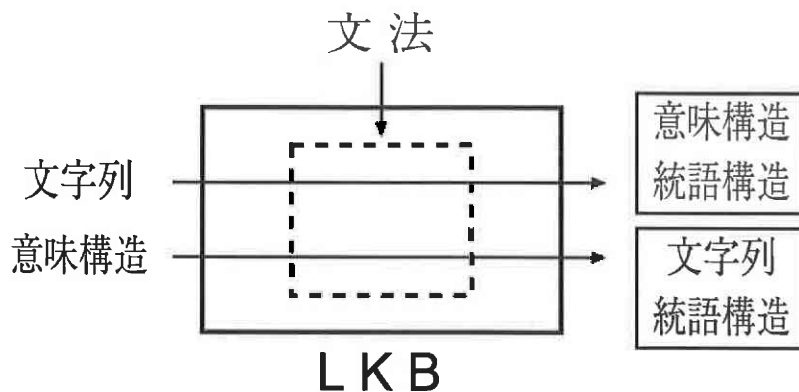


図 1: LKB とその機能

LKB で表示される統語構造は基本的にタイプ付き素性構造に従ったものだが、それに対応した木構造も表示可能である (図 2)。また LKB では MRS (Copestake et al., 1999) に基づく意味構造も出力できる (図 3)。これらの統語構造、意味構造は文「犬が寝(る)」 (“inuga ne”) に対応する表示である。

4.2 実装用日本語文法

以下では実装における文法の変更箇所について述べる。

sign の素性 まず *sign* で導入される素性に変更が加えられる (43)。

$$(43) \left[\text{ORTH } *dlist*, \text{ SYN } gram-cat, \text{ SEM } sem-struc, \text{ ARGS } *list* \right]$$

ORTH は PHON に対応する。ARGS は *phrase* と *word* で用いられる素性であり、*phrase* であればその構成素 (後述する *phrase* の制約により ARGS 内の要素が DTRS にマッピングされる) を、*word* であれば STEM、つまり基となる *lexeme* に対応する。

項具現化原理 LKB では (30) のような振舞いをそのまま実装することはできない。つまり外項を取る場合取らない場合だけでなく、内項を取る場合取らない場合、項を全く取

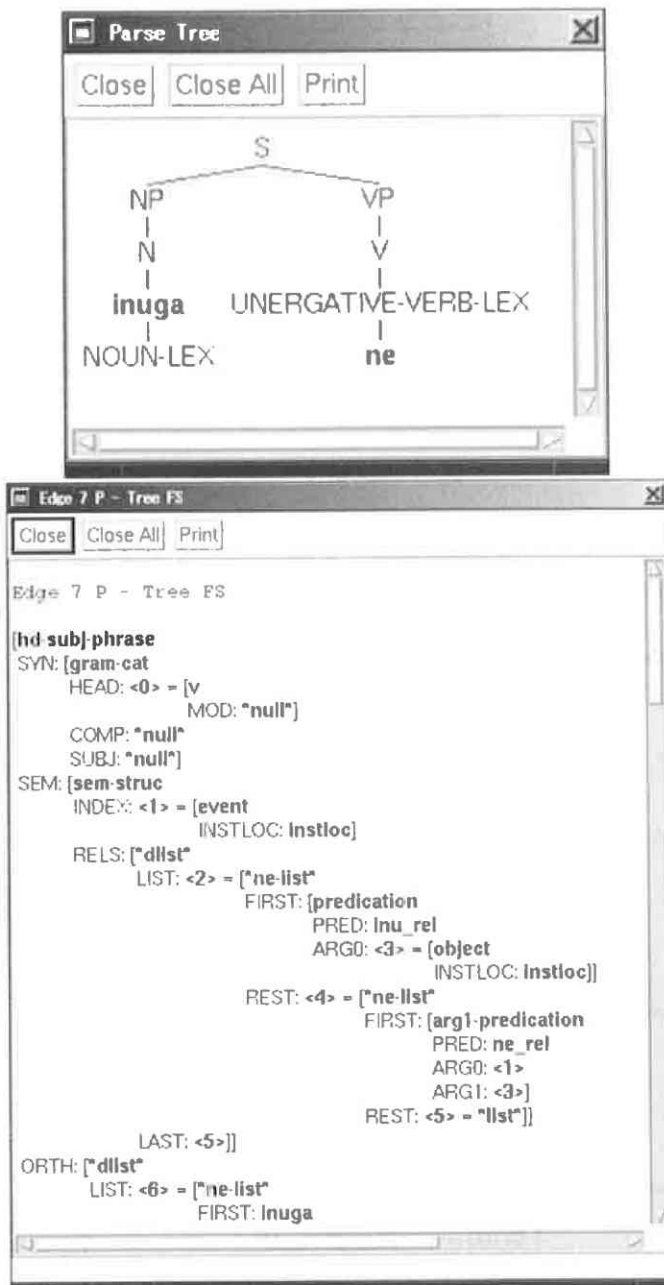


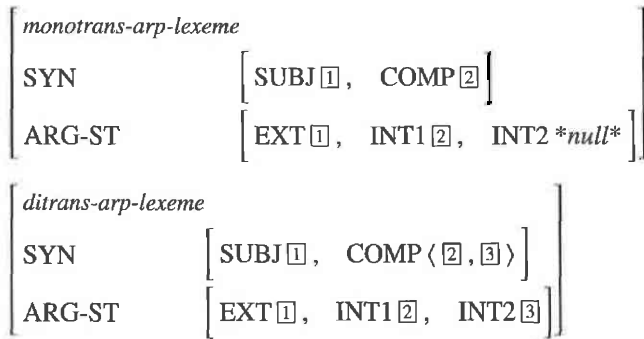
図 2: 統語構造の出力例



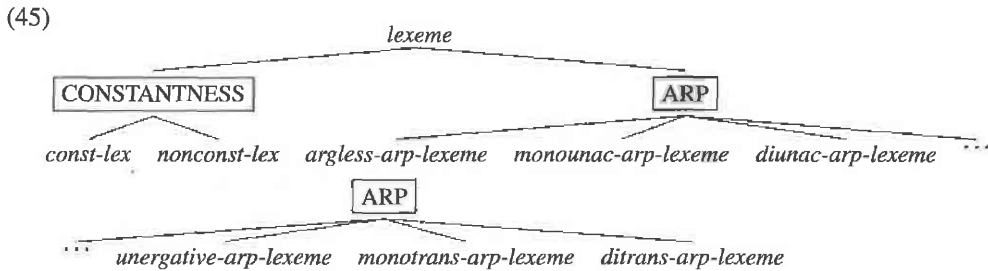
図 3: 意味構造の出力例

らない場合をそれぞれ別の項具現化原理により扱わなくてはならない。本文法では (29) に基づき項具現化原理を (44) のように規定した。

- (44) $\left[\begin{array}{l} \textit{argless-arp-lexeme} \\ \text{SYN} \quad \left[\text{SUBJ *null*, COMP *null*} \right] \\ \text{ARG-ST} \quad \left[\text{EXT *null*, INT1 *null*, INT2 *null*} \right] \end{array} \right]$
- $\left[\begin{array}{l} \textit{monounac-arp-lexeme} \\ \text{SYN} \quad \left[\text{SUBJ } \square, \text{ COMP *null*} \right] \\ \text{ARG-ST} \quad \left[\text{EXT *null*, INT1 } \square, \text{ INT2 *null*} \right] \end{array} \right]$
- $\left[\begin{array}{l} \textit{diunac-arp-lexeme} \\ \text{SYN} \quad \left[\text{SUBJ } \square, \text{ COMP } \square \right] \\ \text{ARG-ST} \quad \left[\text{EXT *null*, INT1 } \square, \text{ INT2 } \square \right] \end{array} \right]$
- $\left[\begin{array}{l} \textit{unergative-arp-lexeme} \\ \text{SYN} \quad \left[\text{SUBJ } \square, \text{ COMP *null*} \right] \\ \text{ARG-ST} \quad \left[\text{EXT } \square, \text{ INT1 *null*, INT2 *null*} \right] \end{array} \right]$

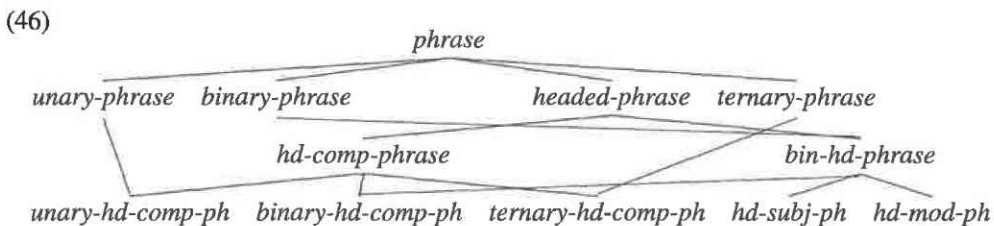


lexeme の階層 実装用文法の *lexeme* の階層はまず (44) の ARP による分類と屈折変化のある無しによる分類に従う。よって *lexeme* の最上部は (45) のようになる。



現在屈折変化のあるのは名詞だけである。一連の ARP 群は動詞だけに適用されるのではなく名詞、形容詞、副詞、後置詞にも汎用的に適用される。具体的には *noun-lex*、*adj-lex*、*adv-lex* には *argless-arp* が、*postp-lex* には *monotrans-arp* が適用される。

phrase の階層 3. 節の日本語文法において *phrase* は、その *daughter* 数に関わらず一律に規定されている。例えば *hd-comp-ph* は (40) にあるように NH1、NH2 がそれぞれある場合無い場合をまとめて規定されている。しかし LKB 上に実装する際には *daughter* が 1 つ (H のみ) の場合 (*unary-phrase*)、*daughter* が 2 つ (H と NH1) の場合 (*binary-phrase*)、*daughter* が 3 つ (H、NH1、NH2) の場合 (*ternary-phrase*) を別々に規定しなくてはならない (46)。



bin-hd-phrase のような中間のタイプを設定するのは *greatest lower bound* を一意に定めるためである。さらにこれに伴い、意味的構成性原理も (47) のように 3 つに分れる。

$$(47) \left[\begin{array}{l} \text{unary-phrase} \\ \text{SEM|RESTR } \boxed{1}, \text{ DTR} \left[\text{H|SEM|RESTR } \boxed{1} \right] \end{array} \right]$$

$$\left[\begin{array}{l} \text{binary-phrase} \\ \text{SEM|RESTR } \boxed{1} \oplus \boxed{2}, \text{ DTR} \left[\begin{array}{l} \text{H|SEM|RESTR } \boxed{1} \\ \text{NH1|SEM|RESTR } \boxed{2} \end{array} \right] \end{array} \right]$$

$$\left[\begin{array}{l} \text{ternary-phrase} \\ \text{SEM|RESTR } \boxed{1} \oplus \boxed{2} \oplus \boxed{3}, \text{ DTR} \left[\begin{array}{l} \text{H|SEM|RESTR } \boxed{1} \\ \text{NH1|SEM|RESTR } \boxed{2} \\ \text{NH2|SEM|RESTR } \boxed{3} \end{array} \right] \end{array} \right]$$

hd-comp-ph も (46) から分かるように3つに分かれる (48)。

$$(48) \left[\begin{array}{l} \text{unary-hd-comp-phrase} \\ \text{SYN|SUBJ } \boxed{1} \\ \text{DTRS} \left[\begin{array}{l} \text{H} \left[\begin{array}{l} \text{word} \\ \text{SYN} \left[\text{SUBJ } \boxed{1}, \text{ COMP } *null* \right] \end{array} \right] \end{array} \right] \end{array} \right]$$

$$\left[\begin{array}{l} \text{binary-hd-comp-phrase} \\ \text{SYN|SUBJ } \boxed{1} \\ \text{DTRS} \left[\begin{array}{l} \text{H} \left[\begin{array}{l} \text{word} \\ \text{SYN} \left[\text{SUBJ } \boxed{1}, \text{ COMP } \boxed{2} \right] \end{array} \right] \\ \text{NH1 } \boxed{2} \end{array} \right] \end{array} \right]$$

$$\left[\begin{array}{l} \text{ternary-hd-comp-phrase} \\ \text{SYN|SUBJ } \boxed{1} \\ \text{DTRS} \left[\begin{array}{l} \text{H} \left[\begin{array}{l} \text{word} \\ \text{SYN} \left[\text{SUBJ } \boxed{1}, \text{ COMP } \langle \boxed{2}, \boxed{3} \rangle \right] \end{array} \right] \\ \text{NH1 } \boxed{2} \\ \text{NH2 } \boxed{3} \end{array} \right] \end{array} \right]$$

デフォルトの扱い (48) では本来なら結合価原理より扱われるべき *mother* と *head daughter* の *SUBJ* の同一性がいちいち述べられている。これは *LKB* がデフォルトを扱えないためである。このことは *hd-subj-ph* (49) と *hd-mod-ph* (50) にもあてはまる。

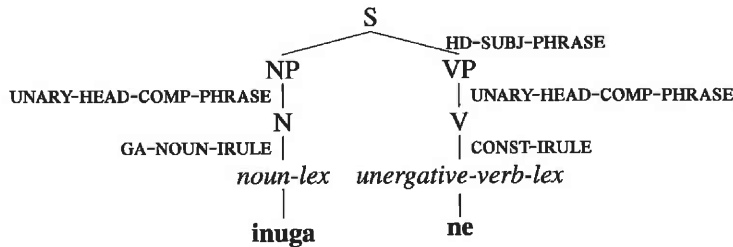
- (49) $\left[\begin{array}{l} \text{hd-subj-phrase} \\ \text{SYN} \quad \left[\text{SUBJ *null*}, \text{COMP *null*} \right] \\ \text{DTRS} \quad \left[\begin{array}{l} \text{NH1 } \square \\ \text{H} \quad \left[\text{SUBJ } \square, \text{COMP *null*} \right] \end{array} \right] \end{array} \right]$
- (50) $\left[\begin{array}{l} \text{hd-mod-phrase} \\ \text{SYN} \quad \left[\text{SUBJ } \square, \text{COMP } \square \right] \\ \text{DTRS} \quad \left[\begin{array}{l} \text{NH1} \left[\text{SYN | HEAD | MOD} \left[\begin{array}{l} \text{SYN | HEAD } \square \\ \text{SEM | INDEX } \square \end{array} \right] \right] \\ \text{H} \quad \left[\text{SYN} \quad \left[\text{HEAD } \square, \text{SUBJ } \square, \text{COMP } \square \right] \right] \\ \text{SEM | INDEX } \square \end{array} \right] \end{array} \right]$

4.3 解析例

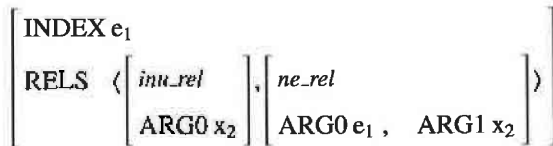
本節では、本研究で構築された文法によりいかにして文が解析されるのかを例示する。

犬が寝(る)

(51) 統語構造



意味構造



統語構造 まず *noun-lex*、**inu** が *ga-noun-irule* の適用を受けて *word* となり $\left[\begin{array}{l} \text{GR } \textit{sbj} \\ \text{SFFX } \textit{ga} \end{array} \right]$ のように文法関係と接尾辞が決まる。次に *word*、**inu** は *unary-head-comp-phrase* として認可され *phrase* になり、動詞句により主語として選択されることが可能となる。一方

unergative-verb-lex, **ne** は *const-irule* の適用を受けて *word* となった後、*unary-head-comp-phrase* として認可され *phrase* になる。最後に *inuga* と *ne* が *hd-subj-phrase* として認可される。

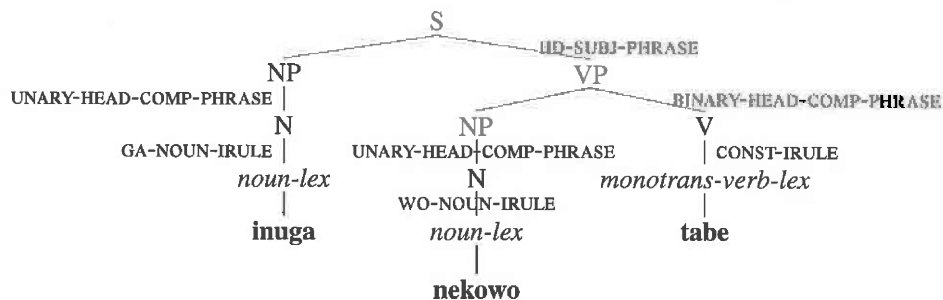
意味構造 **inuga** と **ne** の意味構造は (52) のようになっている。

$$(52) \text{ inuga : } \left[\begin{array}{l} \text{INDEX } \boxed{\text{object}} \\ \text{RELS } \left[\begin{array}{l} \text{predication} \\ \text{PRED } \text{ inu_rel, ARG0 } \boxed{} \end{array} \right] \end{array} \right]$$

$$\text{ne : } \left[\begin{array}{l} \text{INDEX } \boxed{\text{event}} \\ \text{RELS } \left[\begin{array}{l} \text{arg1-predication} \\ \text{PRED } \text{ ne_rel, ARG0 } \boxed{}, \text{ ARG1 } \text{object} \end{array} \right] \end{array} \right]$$

inuga は名詞なので INDEX の値として *object* をとり、一方 **ne** は動詞なので INDEX の値として *event* をとる。**ne** は *unergative* なので RELS の値として *arg1-predication* をとる。ARG1 が *unergative* の外項の意味に相当する。この2つが *hd-subj-phrase* として認可される際の一つになって (51) の意味構造が得られる。具体的には、*hd-subj-phrase* は *headed-phrase* の制約を継承している所以意味的継承原理の適用を受ける。従って主辞である **ne** の INDEX 値 (*event*, e_1) が *hd-subj-phrase* 全体の INDEX 値となる。また *hd-subj-phrase* は *binary-phrase* の制約も継承している所以意味的構成性原理の適用を受けて **inuga** と **ne** の RELS 値2つが *hd-subj-phrase* 全体の RELS 値となる。**ne** の ARG1 と **inuga** の ARG0 の値が単一化 (x_2) するのは **ne** が *agentive-verb-lex* の制約を継承しているためである。*agentive-verb-lex* は外項の名詞句の INDEX 値と ARG1 の値を同じにする。犬が猫を食べ(る)

(53) 統語構造



意味構造

$$\left[\begin{array}{l} \text{INDEX } e_1 \\ \text{RELS } \left\langle \left[\begin{array}{l} \text{inu_rel} \\ \text{ARG0 } x_2 \end{array} \right], \left[\begin{array}{l} \text{neko_rel} \\ \text{ARG0 } x_3 \end{array} \right], \left[\begin{array}{l} \text{tabe_rel} \\ \text{ARG0 } e_1, \text{ ARG1 } x_2, \text{ ARG2 } x_3 \end{array} \right] \right\rangle \end{array} \right]$$

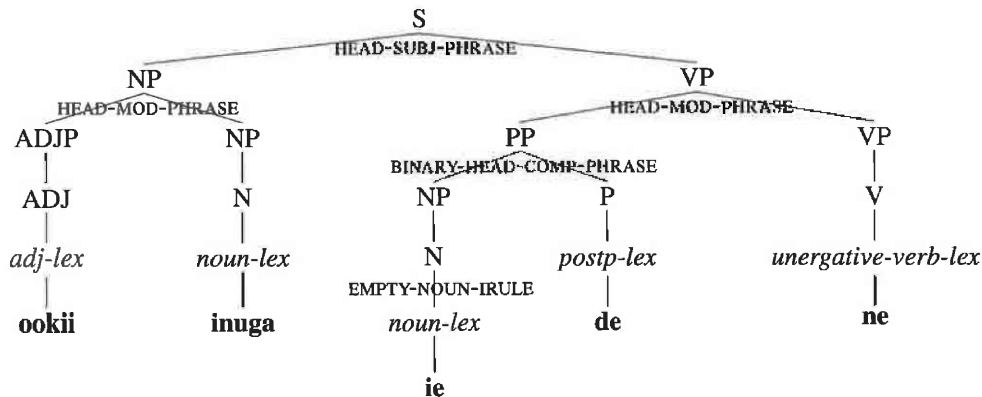
統語構造 **nekowo** は *noun-lex*、**neko** が *wo-noun-irule* の適用を受けて派生され、

$\left[\begin{array}{l} \text{GR } \textit{obj} \\ \text{SFFX } \textit{wo} \end{array} \right]$ が決まる。**tabe** は *const-irule* の適用を受けて *word* となり、その後 **nekowo** とともに *binary-head-comp-phrase* として認可される。主語の格は必ず主格となるようにしているが、目的語の格は動詞により対格であったり与格であったりする。動詞 **tabe** は対格の目的語を取るように語彙的に指定されている。

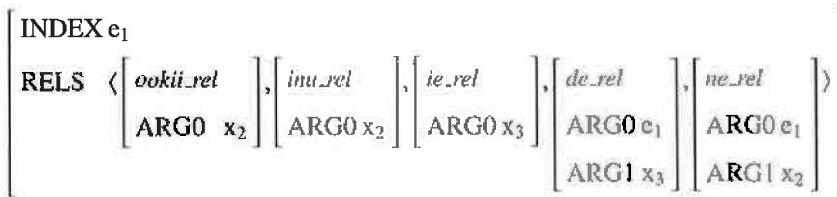
意味構造 **tabe** は *monotransitive* であり *arg1-2-predication* をとる。*monotransitive* は *agentive*、*transitive* から制約を引き継いでおり、その結果、ARG1 が外項の名詞句の INDEX 値 (ARG0) と、ARG2 が一番目の内項 (INT1) の INDEX 値 (ARG0) と同一視されることになる。

大きい犬が家で寝(る)

(54) 統語構造



意味構造



統語構造 *adj-lex*、**ookii** は *const-irule*、*unary-head-comp-phrase* の適用を受け、その後修飾先である名詞句 **inuga** とともに *head-mod-phrase* として認可される。**ie** は *empty-noun-irule* の適用を受けて *word* となり、後置詞句 **ie de** は後置詞 **de** を主辞とした *binary-head-comp-phrase* として認可され、その後修飾先である **ne** とともに *head-mod-phrase* として認可される。

意味構造 *adj-lex* の INDEX 値 (ARG0) は MOD 値の INDEX 値 (ARG0) と同一になるように制約されている。よって修飾先である名詞句 **inuga** の INDEX 値が **ookii** の INDEX

値となる。後置詞 **de** は補語は取るが主語は取らないように制約されており、RELS の値として *arg1-predication* を取る。ARG1 の値は補語の INDEX 値と同一視される。よって **de** の ARG1 の値が **ie** の ARG0 の値と同じになる。また **de** は動詞句を修飾するように制約されており、INDEX 値がその修飾先の INDEX 値と同一視される。

5. 計算機と言語理論

4. 節で計算機処理可能な文法について述べたが、このような文法を構築することの、理論言語学と自然言語処理に与える影響について考察する。

1. 節で述べたが、現在の自然言語処理で用いられている文法では言語学的な妥当性や一般性は全く考慮されていない。自然言語処理研究者は言語獲得の事実の説明には関心がほとんどないので、抽象的で計算機処理が困難な文法体系より単純で明示的な句構造文法を好む。また句構造文法用の処理効率の高い構文解析アルゴリズムが既に開発され、実用レベルに達しているのも理由の一つである。しかし 2. 節で述べたように、HPSG は言語学的な妥当性一般性を維持したまま自然言語処理の要求に応えることを可能にする：
i) そのタイプ階層を用いた文法体系は、応用に耐え得るほどの大規模文法を構築する際に伴うメンテナンスの困難さを軽減する。
ii) 制約に基づく単一の層からなる文法は構文解析と文生成を同様に扱うことができ、これも自然言語処理にとって大きな利点となる。
iii) 高度な一般性抽象性を持ちながらも、その表層指向性が計算量を現実的なレベルに押さえる。
iv) HPSG の形態音韻、統語意味、文脈などの情報を並行的に扱える記述能力は、自然言語処理の応用システムの幅を広げる可能性を持つ。
v) Constructionism を取り込んだ最近の HPSG は従来は例外的として除外されてきたような現象も理論的枠組み内で捉えることを可能にし、応用に耐えうる規模の文法を理論的に整合的に構築できる。一つの応用事例としてスタンフォード大学のものを挙げる。スタンフォード大学では HPSG に基づく大規模英文法 ERG を開発し、それを Verbmobil 話し言葉機械翻訳プロジェクト (Wahlster, 1997) の翻訳システムに組み込んだ。また ERG は言語障害者のための発話支援システムにも用いられている (Copestake, 1997)。これらのシステムに共通しているのは、話し言葉や障害者の発話に見られる断片的な発話に対して断片的な意味情報を生成し、それらを用いて完全な文の意味情報を構築する、という点である。これらのシステムでは統語と意味の密接な関係を同時・並行的に扱えるという HPSG の利点をうまく活用していると言える。

理論言語学で必要なものは母語話者の直観である。コーパス言語学のように大規模なデータの処理が要求されるような分野では計算機の必要性は明らかだが、理論言語学における計算機の役割はそれほど明白ではない。しかし計算機上に文法を実装することの理論的な観点からの利点はいくつかある：
i) 実装に際してその文法は計算機上への実装に耐えられるほどの厳密性明示性が要求される。この時点で曖昧な個所や不明瞭な個所は排除される。
ii) 文法の適用範囲が広がり規模が大きくなるにつれて生じ得る矛盾は、人間の目で検出するのは時として非常に困難だが、計算機にとっては容易である。矛盾が生じた時点で計算機はすぐにそれを検出する。
iii) 実装後、その文法に基づいてある程度

の規模のデータを実際に解析することで、その文法体系全体として当初の期待通りに振る舞うかを確認できる。iv) また現実の発話データの解析を通して、その文法理論の現実のデータに対する説明力、網羅性を検証できる。いかなる文法理論も、それがあつ個別言語の文法である限りは、その言語の現象を広範に扱えるべきである。v) いったん計算機上に実装され電子化された文法は、様々な媒体により携帯、保存が可能となる。そして、例えばネットワークなどにより多くの言語学者に共有されることにより、文法の共同開発が可能となる。ソフトウェアのオープンソース化による高度で急速な発展は例えば UNIX の世界を見れば明白である。

6. 結論

本稿では計算機上で動作する言語学的に妥当な日本語文法について述べた。4. 節で述べたように、本研究で用いた文法開発環境 LKB による制限のため、いくつかの点で理論と実装の間にわずかな乖離も見られたが、ほぼ理論に忠実に、そして理論が予測する通りの振舞いを示すように実装できた。また、近年その溝が深くなりつつある理論言語学と自然言語処理の関わりについて述べ、その2つのアプローチが HPSG を接点としてお互いにとって有益な形で相互作用し得ることについて論じた。

補遺: 語彙項目の例

(55) Unergative verb : 寝(る)

<i>unergative-verb-lex</i>													
ORTH	"ne"												
SYN	<table border="1"> <tr> <td>HEAD</td> <td><i>v</i></td> </tr> <tr> <td>SUBJ</td> <td>[1] <i>phrase</i></td> </tr> <tr> <td></td> <td> <table border="1"> <tr> <td>SYN HEAD</td> <td>[GR <i>sbj</i>, SFFX <i>ga</i>]</td> </tr> <tr> <td>SEM INDEX</td> <td>[2] <i>object</i></td> </tr> </table> </td> </tr> <tr> <td>COMP</td> <td>< ></td> </tr> </table>	HEAD	<i>v</i>	SUBJ	[1] <i>phrase</i>		<table border="1"> <tr> <td>SYN HEAD</td> <td>[GR <i>sbj</i>, SFFX <i>ga</i>]</td> </tr> <tr> <td>SEM INDEX</td> <td>[2] <i>object</i></td> </tr> </table>	SYN HEAD	[GR <i>sbj</i> , SFFX <i>ga</i>]	SEM INDEX	[2] <i>object</i>	COMP	< >
HEAD	<i>v</i>												
SUBJ	[1] <i>phrase</i>												
	<table border="1"> <tr> <td>SYN HEAD</td> <td>[GR <i>sbj</i>, SFFX <i>ga</i>]</td> </tr> <tr> <td>SEM INDEX</td> <td>[2] <i>object</i></td> </tr> </table>	SYN HEAD	[GR <i>sbj</i> , SFFX <i>ga</i>]	SEM INDEX	[2] <i>object</i>								
SYN HEAD	[GR <i>sbj</i> , SFFX <i>ga</i>]												
SEM INDEX	[2] <i>object</i>												
COMP	< >												
SEM	<table border="1"> <tr> <td>INDEX</td> <td>[3] <i>event</i></td> </tr> <tr> <td>RELS</td> <td> <table border="1"> <tr> <td colspan="2"><i>arg1-predication</i></td> </tr> <tr> <td>PRED</td> <td><i>ne_rel</i>, ARG0 [3], ARG1 [2]</td> </tr> </table> </td> </tr> </table>	INDEX	[3] <i>event</i>	RELS	<table border="1"> <tr> <td colspan="2"><i>arg1-predication</i></td> </tr> <tr> <td>PRED</td> <td><i>ne_rel</i>, ARG0 [3], ARG1 [2]</td> </tr> </table>	<i>arg1-predication</i>		PRED	<i>ne_rel</i> , ARG0 [3], ARG1 [2]				
INDEX	[3] <i>event</i>												
RELS	<table border="1"> <tr> <td colspan="2"><i>arg1-predication</i></td> </tr> <tr> <td>PRED</td> <td><i>ne_rel</i>, ARG0 [3], ARG1 [2]</td> </tr> </table>	<i>arg1-predication</i>		PRED	<i>ne_rel</i> , ARG0 [3], ARG1 [2]								
<i>arg1-predication</i>													
PRED	<i>ne_rel</i> , ARG0 [3], ARG1 [2]												
ARG-ST	[EXT [1], INT1 < >, INT2 < >]												

(56) Mono Transitive verb : 食べる

<i>monotrans-verb-lex</i>																					
ORTH	"tabe"																				
SYN	<table border="1"> <tr> <td>HEAD</td> <td><i>v</i></td> </tr> <tr> <td>SUBJ ①</td> <td><i>phrase</i></td> </tr> <tr> <td></td> <td> <table border="1"> <tr> <td>SYN HEAD</td> <td>[GR <i>sbj</i> , SFFX <i>ga</i>]</td> </tr> <tr> <td>SEM INDEX ②</td> <td><i>object</i></td> </tr> </table> </td> </tr> <tr> <td>COMP ③</td> <td> <table border="1"> <tr> <td>SYN HEAD</td> <td><i>phrase</i></td> </tr> <tr> <td></td> <td> <table border="1"> <tr> <td>GR</td> <td><i>obj</i> , SFFX <i>wo</i></td> </tr> </table> </td> </tr> <tr> <td>SEM INDEX ④</td> <td><i>object</i></td> </tr> </table> </td> </tr> </table>	HEAD	<i>v</i>	SUBJ ①	<i>phrase</i>		<table border="1"> <tr> <td>SYN HEAD</td> <td>[GR <i>sbj</i> , SFFX <i>ga</i>]</td> </tr> <tr> <td>SEM INDEX ②</td> <td><i>object</i></td> </tr> </table>	SYN HEAD	[GR <i>sbj</i> , SFFX <i>ga</i>]	SEM INDEX ②	<i>object</i>	COMP ③	<table border="1"> <tr> <td>SYN HEAD</td> <td><i>phrase</i></td> </tr> <tr> <td></td> <td> <table border="1"> <tr> <td>GR</td> <td><i>obj</i> , SFFX <i>wo</i></td> </tr> </table> </td> </tr> <tr> <td>SEM INDEX ④</td> <td><i>object</i></td> </tr> </table>	SYN HEAD	<i>phrase</i>		<table border="1"> <tr> <td>GR</td> <td><i>obj</i> , SFFX <i>wo</i></td> </tr> </table>	GR	<i>obj</i> , SFFX <i>wo</i>	SEM INDEX ④	<i>object</i>
HEAD	<i>v</i>																				
SUBJ ①	<i>phrase</i>																				
	<table border="1"> <tr> <td>SYN HEAD</td> <td>[GR <i>sbj</i> , SFFX <i>ga</i>]</td> </tr> <tr> <td>SEM INDEX ②</td> <td><i>object</i></td> </tr> </table>	SYN HEAD	[GR <i>sbj</i> , SFFX <i>ga</i>]	SEM INDEX ②	<i>object</i>																
SYN HEAD	[GR <i>sbj</i> , SFFX <i>ga</i>]																				
SEM INDEX ②	<i>object</i>																				
COMP ③	<table border="1"> <tr> <td>SYN HEAD</td> <td><i>phrase</i></td> </tr> <tr> <td></td> <td> <table border="1"> <tr> <td>GR</td> <td><i>obj</i> , SFFX <i>wo</i></td> </tr> </table> </td> </tr> <tr> <td>SEM INDEX ④</td> <td><i>object</i></td> </tr> </table>	SYN HEAD	<i>phrase</i>		<table border="1"> <tr> <td>GR</td> <td><i>obj</i> , SFFX <i>wo</i></td> </tr> </table>	GR	<i>obj</i> , SFFX <i>wo</i>	SEM INDEX ④	<i>object</i>												
SYN HEAD	<i>phrase</i>																				
	<table border="1"> <tr> <td>GR</td> <td><i>obj</i> , SFFX <i>wo</i></td> </tr> </table>	GR	<i>obj</i> , SFFX <i>wo</i>																		
GR	<i>obj</i> , SFFX <i>wo</i>																				
SEM INDEX ④	<i>object</i>																				
SEM	<table border="1"> <tr> <td>INDEX ⑤</td> <td><i>event</i></td> </tr> <tr> <td>RELS</td> <td><i>arg1-2-predication</i></td> </tr> <tr> <td>PRED</td> <td><i>tabe_rel</i></td> </tr> <tr> <td>ARG0</td> <td>⑤</td> </tr> <tr> <td>ARG1</td> <td>②</td> </tr> <tr> <td>ARG2</td> <td>③</td> </tr> </table>	INDEX ⑤	<i>event</i>	RELS	<i>arg1-2-predication</i>	PRED	<i>tabe_rel</i>	ARG0	⑤	ARG1	②	ARG2	③								
INDEX ⑤	<i>event</i>																				
RELS	<i>arg1-2-predication</i>																				
PRED	<i>tabe_rel</i>																				
ARG0	⑤																				
ARG1	②																				
ARG2	③																				
ARG-ST	[EXT ① , INT1 ③ , INT2 < >]																				

(57) Noun : 犬

<i>noun-lex</i>									
ORTH	"inu"								
SYN—HEAD	<table border="1"> <tr> <td><i>n</i></td> </tr> <tr> <td>GR <i>gr</i> , SFFX <i>sffx</i></td> </tr> </table>	<i>n</i>	GR <i>gr</i> , SFFX <i>sffx</i>						
<i>n</i>									
GR <i>gr</i> , SFFX <i>sffx</i>									
SEM	<table border="1"> <tr> <td>INDEX ①</td> <td><i>object</i></td> </tr> <tr> <td>RELS</td> <td><i>predication</i></td> </tr> <tr> <td></td> <td> <table border="1"> <tr> <td>PRED</td> <td><i>inu_rel</i> , ARG0 ①</td> </tr> </table> </td> </tr> </table>	INDEX ①	<i>object</i>	RELS	<i>predication</i>		<table border="1"> <tr> <td>PRED</td> <td><i>inu_rel</i> , ARG0 ①</td> </tr> </table>	PRED	<i>inu_rel</i> , ARG0 ①
INDEX ①	<i>object</i>								
RELS	<i>predication</i>								
	<table border="1"> <tr> <td>PRED</td> <td><i>inu_rel</i> , ARG0 ①</td> </tr> </table>	PRED	<i>inu_rel</i> , ARG0 ①						
PRED	<i>inu_rel</i> , ARG0 ①								
ARG-ST	[EXT < > , INT1 < > , INT2 < >]								

(58) **Postposition** : で

<i>postp-lex</i>	
ORTH	“de”
SYN	HEAD PFORM <i>de</i>
	MOD [HEAD ν SEM INDEX ① <i>event</i>]
SUBJ ②< >	
COMP ③	SYN HEAD [GR <i>gr</i> , SFFX <i>empty-sffx</i>]
	SEM INDEX ④ <i>object</i>
SEM	INDEX ①
	RELS [<i>arg1-relation</i> PRED <i>de_rel</i> , ARG0 ①, ARG1 ④]
ARG-ST [EXT ②, INT1 ③, INT2< >]	

(59) **Adjective** : 大きい

<i>adj-lex</i>	
ORTH	“ookii”
SYN HEAD MOD	SYN [GR <i>gr</i> , SFFX <i>sffx</i>]
	SEM INDEX ① <i>object</i>
SEM	INDEX ①
	RELS [<i>predication</i> PRED <i>ookii_rel</i> , ARG0 ①]
ARG-ST [EXT< >, INT1< >, INT2< >]	

参考文献

- Copestake, A. (1997). Augmented and alternative NLP techniques for augmentative and alternative communication. In *Proceedings of the ACL workshop on Natural Language Processing for Communication Aids*.
- Copestake, A. (2002). *Implementing Typed Feature Structure Grammars*. CSLI Publications.
- Copestake, A., Flickinger, D., Sag, I. A., & Pollard, C. (1999). Minimal Recursion Semantics: An introduction. Unpublished manuscript.

Ginzburg, J. & Sag, I. A. (2000). *Interrogative Investigations*. CSLI Publications.

郡司隆男 (2000). 「日本語の動詞の形態素とそれに基づくタグ付け」. 『制約に基づく文法の連続量の概念を取り入れた拡張の研究』, 平成 9~11 年度科学研究費補助金研究成果報告書.

Sag, I. A. & Wasow, T. (1999). *Syntactic Theory: A Formal Introduction*. CSLI Publications.

Wahlster, W. (1997). VerbMobil—Erkennung, Analyse, Transfer, Generierung und Synthese von Spontansprache. Tech. rep., VerbMobil Report 198. Deutsches Forschungszentrum für Künstliche Intelligenz GmbH, Saarbrücken, Germany.

Author's E-mail Address: chasi@sils.shoin.ac.jp

Author's web site: <http://sils.shoin.ac.jp/~chasi/>