

特集 「現実テキストからの知識表現と推論の実現」

素性から組み上げられる文の論理構造

Logical Semantics Composed of Feature Structures

東条 敏
Satoshi Tojo

北陸先端科学技術大学院大学情報科学研究科
School of Information Science, Japan Advanced Institute of Science and Technology.
tojo@jaist.ac.jp

Keyword: feature structure, categorial grammar, HPSG, evolution and game theory.

1. はじめに

いきなり大げさな話から始めるが、近代科学の精神とはどんなものだったか？ いろいろな考え方があろうが、一番わかりやすい定義の一つが還元主義 (reductionism) である。すなわち、全体の性質は構成要素個々の性質の累積であるとする考え方である。最もわかりやすいのが物理におけるドルトン (J. Dalton, 1766-1844) の原子論であり、万物は等しく原子から構成されており、化学変化は原子の結合の仕方が変化するだけで新たに原子が生成されたり消滅することはないとするものである。これにより錬金術は否定され、金は合成できないものとなった。

言語学の歴史ではこれは構成性原理 (compositionality) あるいはフレーゲ (F. L. G. Frege, 1848-1925) の原理という形で現れる。すなわち文全体の意味は個々の単語から組み上げられたものによって決定されるとするものである。イデオムのような反例もあるが、これはあくまで反例であり、言語の大部分の意味は構成要素の意味に帰着され、かつイデオムにおいても個々の単語の例から著しく乖離するようなものは例外的と考えられる。

では個々の単語が原子であるというアナロジーは適切かというこれは問題で、物理学がのちに分子という考え方をを用いて物質の構成素材とし、それを元素と区別したように、単語でもそれをさらに内部に分割できるというなら形態素^{*1}がむしろ原子であり、語が分子であろう。原子として物理学の後の歴史では陽子だの電子だのと素粒子に分割され、そしてそれがクォークの存在によってさらに分割された。しかし素粒子のレベルでは、それらが共通に質量・電荷・スピンなどといった素性 (せせい; feature) をもつために特徴を比較することができる。

実用的な意味で言語の意味を還元するのに都合の良い素粒子は何だろうか？ オントロジーその他概念辞書の語彙項目の応用を考えると、意味を構成する単位は形態

素より単語と考えるほうが実用的である。すなわち言語の意味はすべて形態素に還元できるわけではなく、あくまで意味の構成要素は各単語であるとしておいて、さらに必要であればその中のいろいろな内部素性についても言及するとしたほうがオントロジー構築の実情に即している。

本稿ではまず2章で単語からボトムアップに文の意味を構成し、述語論理式を得るカテゴリー文法について述べる。次に単語の内部素性の例を3章で論じ、4章でカテゴリーを素性の構造とする文法 HPSG について解説する。次に素性の一致が働く例として5章で古典ギリシア語の語順の問題をあげ、またこのような一致が実際に曖昧性の解決に役立つことを6章で示す。

2. 文の意味の論理表現

2.1 カテゴリー文法

自然言語の生成規則の大部分は文脈自由文法 (context-free grammar) で書くことができる。以下は英語の一部である。

$$\begin{cases} S \rightarrow NP VP \\ VP \rightarrow TV NP \\ VP \rightarrow IV \end{cases}$$

ここで文 (S) は主語名詞句 (NP: Noun Phrase) と述語となる動詞句 (VP: verb phrase) から構成され、動詞句は他動詞 (TV: Transitive Verb) が目的語となる名詞句を伴うか、または単独の自動詞 (IV: Intransitive Verb) である。後者の場合、文は主語と自動詞のみであるから

$$S \rightarrow NP IV$$

のように書ける。発想を変えて、自動詞とは主語名詞句を喰って^{*2}文を返すものとも考えられる。喰う対象を ' (スラッシュ) の右側に、喰った結果を左側に書くと自動詞は 'S/NP' となり、以下のように書ける。

$$S \rightarrow NP S/NP$$

いま e を個体を表すカテゴリー、t を文を表すカテ

*1 形態素はインド=ヨーロッパ語族特有の概念ではない。漢字 (ideogram) は「へん」と「つくり」によって因数分解できる。

*2 「喰う」は bite の訳語。通常に入力の意味である。

リーであるとする、自動詞 IV はカテゴリ t/e となる。固有名詞は個体であるので e となる。

文	t
普通名詞	t/e
動詞句 (VP)	t/e
自動詞 (IV)	t/e
他動詞 (TV)	$(t/e)/e$
固有名詞	e
動詞につく副詞	$(t/e)/(t/e)$
文副詞	t/t
接続詞	$(t/t)/t$

このカテゴリづけで、動詞につく副詞は動詞句を喰って動詞句に、文副詞は文を喰って文に、接続詞は文二つを喰って文になるしくみは容易に理解されると思う。

2.2 タイプつきラムダ計算

前節で述べたカテゴリとは自然言語の構文の生成規則を記述するうえで現れた概念である。本節ではその自然言語表現に対応した論理表現 (= 形式言語表現) を導入する。前節の冒頭にも述べたとおり、この形式言語表現にはタイプ (型) というものがつけられ、それが自然言語表現のカテゴリと対応づけられる。よって、タイプづけされた形式言語表現は自然言語の文全体はもとより、その部分構造 (句構造) にも対応づけられる。そして文全体は形式言語の命題と考える。

自然言語表現	\Leftrightarrow	形式言語表現
カテゴリ	\Leftrightarrow	タイプ
文	\Leftrightarrow	命題

まずタイプとはどのようなものから始めよう。形式言語のタイプも自然言語のカテゴリに合わせて e と t から構成することにする。ただしタイプのほうは斜字体を用いる。

- e とは個体の集合 \mathcal{O} を考えたときにそのうちの一つの元に解釈されるタイプである。
- t とは形式言語の命題の真偽値に解釈されるタイプである。
- $\langle e, t \rangle$ とは e から t への関数のタイプのタイプである。

一階述語論理であれば、そこに現れる形式言語の部品のタイプは e (個体) と $\langle e, t \rangle$ (述語) だけであるが、高階の論理では e と t を任意に組み合わせてタイプを創り出すことができる。

さてラムダ表現 (lambda expression) とは、入力となるタイプの表現をギリシア文字 λ によって表示したものである。入力となる変数はラムダ変数といい、またこの変数はラムダ束縛されているという。形式言語表現のタイプは \cdot (コロン) を用いて指定される。 $\lambda x[p(x)]:\langle \alpha, \beta \rangle$ と書けば、 x がタイプ α であり、このタイプのものを喰ってタイプ β の形式言語表現を返すような関数である。

いま $a:\alpha$ であるとき

$$\lambda x[p(x)]a = p(a):\beta$$

は関数 $\lambda x[p(x)]$ が a を喰って $p(a)$ を返すという操作を表す。

このようにラムダ変数何かが定項を喰って λ の文字が消える変換は β -縮約 (beta contraction) と呼ばれる。ラムダ表現のこのような書換えに関する操作には、ほかに α -変換 (alpha conversion), η -変換 (eta reduction) がある。 α -変換はラムダで束縛された変数の名前のつけ換えによる等価変換 $\lambda x[p(x)] = \lambda y[p(y)]$ である。 η -変換とは、余分なラムダ変数を取り去る変換である。例えば論理式 φ に x が自由変数として含まれていなければ x をラムダ束縛しても $\lambda x[\varphi(x)] = \varphi$ となる [Hindley 86]。

これから自然言語の表現を逐語的に形式言語の表現に翻訳するが、その際形式言語表現はすべて斜字体とし、しかも小文字に限定することにする。まず、カテゴリとタイプを 1 対 1 に対応させるために、次のようなカテゴリからタイプをつくり出す (メタな) 関数を正式に定義しておく。

$$\begin{cases} f_0(e) = e \\ f_0(t) = t \\ f_0(A/B) = \langle f_0(B), f_0(A) \rangle \end{cases}$$

すると任意のカテゴリ、例えば $(t/(t/e))/(t/e)$ は

$$f_0((t/(t/e))/(t/e)) = \langle \langle e, t \rangle, \langle \langle e, t \rangle, t \rangle \rangle$$

のようにタイプへの変換ができる。

自然言語表現から形式言語表現へは通常の言語翻訳と同様に辞書があるものと仮定しよう。このとき自然言語表現のカテゴリは形式言語表現のタイプと同様 \cdot (コロン) で表示される。以下「翻訳」には \Rightarrow を用い

自然言語表現: カテゴリ \Rightarrow 形式言語表現: タイプ
のように表記する。

さて実際の翻訳例を見てみよう。自然言語の各単語にはそれに対応する形式言語の表現が対応づけられていて、これを辞書とする。"John walks." は以下のように辞書引きされるものとする。

$$\begin{cases} \text{john}: e \Rightarrow j:e \\ \text{walks}: t/e \Rightarrow \lambda x[\text{walk}(x)]: \langle e, t \rangle \end{cases}$$

このとき自然言語のカテゴリ組立てによる作文と全く平行に、形式言語のほうもラムダ変換を使って以下のように命題を生成できる。

$$\frac{j:e \quad \lambda x[\text{walk}(x)]: \langle e, t \rangle}{\text{walk}(j): t}$$

二変数の述語の場合ラムダ変数の並びによって喰う順序を規定し、外側のラムダ変数から順次入力を受け付けることにする。すると例文 "John loves Mary." に対しては次のように命題を生成することができる。ここで $\text{love}(y)(x)$ は $(\text{love}(y))(x)$ の意味であるが、しばしば順序を省略して $\text{love}(x, y)$ と書く。

$$\begin{cases} j:e \\ m:e \\ \lambda y \lambda x [\text{love}(y)(x)]:\langle e, \langle e, t \rangle \rangle \end{cases}$$

このとき、ラムダ変換を用いると

$$j:e \frac{\lambda y \lambda x [\text{love}(y)(x)]:\langle e, \langle e, t \rangle \rangle \quad m:e}{\lambda x [\text{love}(m)(x)]:\langle e, t \rangle} \quad \text{love}(j)(m):t$$

のように命題をつくることができる。

2.3 セマンティクス

形式言語のセマンティクスはそのまま一階述語論理のセマンティクスに従って与えられる。そのしくみは次の三点に要約される。

- 個体の集合を与え、タイプ e が一個体に解釈されること
- タイプ t (文) は命題であるから真偽値 T/F に解釈されること
- 命題の各部分の意味を合成していったものが全体の意味を構成すること

以下にこれらのことを順次説明していく。まず、個体 (individual) の集合 \mathcal{U} (universe) と解釈 $\llbracket \cdot \rrbracket$ が次のように与えられているものとする

- (1) タイプ e の解釈は \mathcal{U} の一つの元である。いま仮に $\mathcal{U} = \{\text{John}, \text{Mary}, \text{Paul}\}$ であったとすると

$$\llbracket j \rrbracket = \text{John}, \llbracket m \rrbracket = \text{Mary}, \llbracket p \rrbracket = \text{Paul}$$

ここで、 j だから John になるのではない。 j に Paul を割り当てることも可能である。

- (2) タイプ $\langle e, t \rangle$ はタイプ e のものを喰って t になる。すなわち $\langle e, t \rangle$ のものが解釈されると、個体を喰って T/F の真偽値を返す関数になる。例えば $\lambda x [\text{man}(x)]:\langle e, t \rangle$ であり、これを関数としての働きを明示するように入力・出力関係を ‘ \rightarrow ’ で結んで次のように表すことにする。以降、 $\llbracket \cdot \rrbracket$ の中ではラムダ変数の表示は適宜省略し、 $\llbracket \lambda x [p(x)] \rrbracket$ を単に $\llbracket p \rrbracket$ のように書くことを許す。

$$\llbracket \text{man} \rrbracket = \begin{bmatrix} \text{John} \rightarrow T \\ \text{Mary} \rightarrow F \\ \text{Paul} \rightarrow T \end{bmatrix}$$

という関数は関数らしく引数を () の中に表記すれば、値は

$$\llbracket \text{man} \rrbracket (\text{John}) = T, \llbracket \text{man} \rrbracket (\text{Mary}) = F, \dots$$

のようになる。

- (3) 部分部分の解釈から全体の解釈が合成される。例えば $\text{man}:\langle e, t \rangle$ と $j:e$ から構成される $\text{man}(j)$ の意味は、次のように構成される。

$$\begin{aligned} \llbracket \text{man}(j) \rrbracket \\ = \llbracket \text{man} \rrbracket (\llbracket j \rrbracket) \end{aligned}$$

$$\begin{aligned} &= \begin{bmatrix} \text{John} \rightarrow T \\ \text{Mary} \rightarrow F \\ \text{Paul} \rightarrow T \end{bmatrix} (\text{John}) \\ &= T \end{aligned}$$

二変数の高階の述語も合成性で考えると、最初の引数を喰ったときに一変数の述語になり、二つめの引数を喰ったときに真偽が判定できる命題になると考える。以下は $\lambda y \lambda x [\text{love}(x, y)]$ の解釈例である。 $\llbracket \lambda y \lambda x [\text{love}(x, y)] \rrbracket$ を略記してただ $\llbracket \text{love} \rrbracket$ と書くと

$$\llbracket \text{love} \rrbracket = \begin{bmatrix} \text{John} \rightarrow \begin{bmatrix} \text{John} \rightarrow T \\ \text{Mary} \rightarrow T \\ \text{Paul} \rightarrow F \end{bmatrix} \\ \text{Mary} \rightarrow \begin{bmatrix} \text{John} \rightarrow T \\ \text{Mary} \rightarrow T \\ \text{Paul} \rightarrow T \end{bmatrix} \\ \text{Paul} \rightarrow \begin{bmatrix} \text{John} \rightarrow T \\ \text{Mary} \rightarrow F \\ \text{Paul} \rightarrow T \end{bmatrix} \end{bmatrix}$$

このとき

$$\begin{aligned} \llbracket \text{love}(m)(j) \rrbracket \\ = \llbracket \text{love}(m) \rrbracket (\llbracket j \rrbracket) \\ = (\llbracket \text{love} \rrbracket (\llbracket m \rrbracket)) (\llbracket j \rrbracket) \\ \left(\llbracket \text{love} \rrbracket (\llbracket m \rrbracket) = \begin{bmatrix} \text{John} \rightarrow F \\ \text{Mary} \rightarrow T \\ \text{Paul} \rightarrow T \end{bmatrix} \right) \\ = F \end{aligned}$$

のように真偽値 (T/F) を返すことができる。

タイプの合成を行うには、その前段階として自然言語文の表層でもカテゴリーが隣接していることが要求される。いま自然言語の表現 P, Q の形式言語への翻訳が

$$\begin{aligned} \text{P:A/B} &\Rightarrow \lambda x [p(x)]:\langle \beta, \alpha \rangle \\ \text{Q:B} &\Rightarrow q:\beta \end{aligned}$$

のように与えられたとき、自然言語から形式言語への翻訳、形式言語の解釈は以下のような図式で与えられることになる。

$$\begin{aligned} &\frac{\text{P:A/B} \quad \text{Q:B}}{\text{PQ:A}} \\ &\quad \text{翻訳} \downarrow \\ &\frac{\lambda x [p(x)]:\langle \beta, \alpha \rangle \quad q:\beta}{p(q):\alpha} \\ &\quad \text{解釈} \downarrow \\ &\frac{\llbracket p \rrbracket \text{ の定義域に } \llbracket q \rrbracket \text{ が存在するならば}}{\llbracket p(q) \rrbracket = \llbracket p \rrbracket (\llbracket q \rrbracket)} \end{aligned}$$

最終的に文の意味とは、それに相当する形式言語を解釈したときの真偽値であると考える。

$$\text{自然言語表現} \quad \text{John loves Mary: } t$$

翻訳 ↓

$$\text{形式言語表現} \quad \text{love}(m)(j):t$$

解釈 ↓

$$\llbracket \llbracket \text{love}(m)(j) \rrbracket \rrbracket = T/F$$

さてこの先がモンタギュー意味論 (Montague semantics) である。この文法の正式名は PTQ (Proper Treatment of Quantification) であり、量化子 (∀ や ∃) の適切な取扱い (proper treatment) へと議論は展開する。さらに、様相論理を用いて内包と外延の区別、心的存在と客観的実在の区別、さらにはそれに対する可能世界意味論と、いわゆる内包論理へと進む。しかし本稿はここまでとし、この先モンタギュー意味論を理解するならば [Dowty 79, 東条 06] を参照されたい。

このあと本稿は、カテゴリー文法のような構成的な意味記述を中心に据えながら、同時にカテゴリーが単独のシンボルであると考えよりは内部素性をもつとしたほうが自然言語には便利であるとする考え方を展開する。このために、まず自然言語の素性の一致の問題を見てみよう。

3. 性・数・格の一致の問題

自然言語文には格 (case) というものがあって「～は」にあたる主格 (nominative), 「～の」にあたる属格 (genitive), 「～を」にあたる対格 (accusative), 「～に」にあたる与格 (dative) などを指す。

ドイツ語はこの四つの格システムをもつが、格の数は言語によって分類が異なるし、格の呼称も異なる。属格は主に所有を表すがそれだけではなく、いわゆる所有格 (possessive) と奪格^{*3}など複数の機能をもつことがある。対格は一般には目的格と考えられるが、対格と目的格が区別されることもある。ラテン語では呼びかけ時に用いる呼格 (vocative) を含めて主格・呼格・対格・属格・与格・奪格の6格を区別する。ロシア語では造格と前置格を加えて主格・生格・与格・対格・造格・前置格^{*4}の6格システムである。斜格 (oblique case) のように主格以外を一括して指す名前もあって格名は抽象度も異なる。日本語はこの「は」「を」「に」のように格マーカ (case marker) を名詞に膠着 (agglutination) させることで格を表し、名詞そのものの形は変わらない。日本語のほかには例えば韓国・朝鮮語がこのような格マーカをもつ。これに対してインド＝ヨーロッパ語族の多くは語形変化 (屈折; declension) によって格変化を表す [言語学大辞典 95]。

格変化と同様にインド＝ヨーロッパ語族の多くは性 (gender) ・数 (number) によっても屈折する。数の屈折とは英語でもおなじみの単数形・複数形の区別である。古典ギリシア語ではこれらに加えて双数形 (ペアを指す) があつた。性の区別は英語にはない。英語は歴史的にはドイツ語→オランダ語→フリージア語→英語と変移した

と考えられる。ドイツ語の名詞では男性・中性・女性の3種類が区別されるが、それがオランダ語に遷移すると男性・女性が融合して通性となり通性・中性の二性が区別され、英語に至っては性の区別はしない。一方ラテン語系の言語、フランス語・イタリア語・スペイン語では男性・女性の二性が区別されるが、オリジナルのラテン語は男性・女性・中性の三性があつた。

このような性・数・格変化は構文上の曖昧性を解決することができる。例えば、ドイツ語の男性単数の定冠詞は、主格 *der* / 属格 *des* / 与格 *dem* / 対格 *den*, のように活用する。すると、下記の二つの文

{ Ich habe dem Hund zu essen gekauft.
Ich habe den Hund zu essen gekauft.

{ Ich = 「私は」, Hund = 「犬」
habe ~ gekauft = kaufen 「買う」の現在完了形
zu essen = to eat 「食べる」

においては *dem Hund* 「犬に食べるものを買った」/ *den Hund* 「犬を食べるために買った」という区別ができる。

性・数・格における一致は特に人称代名詞において区別されるが、その一致の仕方は言語に依存する。英語で「彼の」は *his* であり「彼女の」は *her* である。よって「彼の車」は *his car* であり「彼女の車」は *her car* である。ところがフランス語では人称代名詞は次に伴う名詞の性に一致する。「彼の」「彼女の」も等しく *son* (次が男性名詞) / *sa* (次が女性名詞) / *ses* (次が複数) となる。よって *sa voiture* (車) だけでは「彼の車」か「彼女の車」かわからない。*son mari* (夫) と言ったら? 「彼の夫」は普通でないから *son* は一意に「彼女の」になる。

4. 素性構造の文法

4.1 カテゴリーから素性構造へ

さて素性の一致の問題の最初のステップとして、英語の数の一致を扱う。英語は述語動詞は主語名詞の単数・複数に一致して形を変える。これまでの議論では

(*) A man walk.

のような文は非文であるとする文法を書くのは煩わしい。2章の文脈自由文法の表記に戻れば、数の一致まで書くには

$S \rightarrow NP VP$

では不十分であり、おのおのを単数 (*sg*) ・複数 (*pl*) まで分けたカテゴリー名を設ける必要がある。

{ $S \rightarrow NP_{sg} VP_{sg}$
 $S \rightarrow NP_{pl} VP_{pl}$

これを冠詞と名詞の間の規則にも書かなければならない。英語ならまだしもドイツ語やフランス語のように性・数によってすべて一致の規則が変わる言語だとこの規則数はもっと増える。

そこで各カテゴリーには内部に変数があると考えたらどうだろう。すなわちカテゴリー *VP*, *NP* には *Num* と

*3 ablative; 「～から」

*4 生格は属格, 造格は具格「によって」, 前置格は処格「について」の斯拉ブ語系での名称。

いう素性があり、これは $sg \cdot pl$ という値を取ると考えるのである。すると規則は再び一つ

$$S \rightarrow NP[Num] VP[Num]$$

にまとめられる。この規則は多数の規則を一つにまとめただけではない。NP や VP の Num 素性は同一の値を取らなければならないことを積極的に述べているのである。

以下ではカテゴリーの内部を素性構造とした HPSG (Head-driven Phrase Structure Grammar) [Pollard 94, Sag 03] について概説する。

4.2 主辞, 下位範疇化, 型階層, 素性構造と単一化

素性構造とは素性 (属性) とその値のペアを縦に複数並べて角括弧でくくって表現したものである。値が複数の要素からなる可能性があるときは ' $\langle \dots \rangle$ ' によってリスト表現する。図1は自然言語文 "A man walks." の構文木を生成した例である。木の各ノードは素性構造になっている。各ノードでは素性 NUM (number) が値 sgl (単数; singular) であるという情報をもっており、この値が異なるものは木構造を構成し得ない。

木構造の中では上位構造を親、親を形成するいくつかの語や句を子と呼ぶ。名詞句や動詞句などあらゆる句にはその子の中に必ず中心的な役割を果たす句や語が存在し、これを主辞 (head) と呼ぶ。図1の中で 'A man' という句を形成するうえで主辞は名詞の 'man' である。HPSG では解析に必要なさまざまなカテゴリーを型 (sort) に分類する。おのおのの素性構造は型をもち、その名前は先頭行に配置され ' \cdot ' に先行されて表される*5。

すべての型は型階層においてその上下関係が定義される。その際、型階層宣言 \sqsubseteq の右辺は左辺より上位にあり、したがってより一般的な型を表す。素性の数が増えるとその値に対する制約を負うので、型は特殊化される。逆に、階層上上位の (より一般的な) の素性構造は下位の素性構造の部分構造となる。例えば ' $\sim decl$ ' は declinable (変化詞) の型であるとし、性・数・格の素性をもつものとする。このとき $\{\sim N, \sim Adj\} \sqsubseteq \sim decl \sqsubseteq$

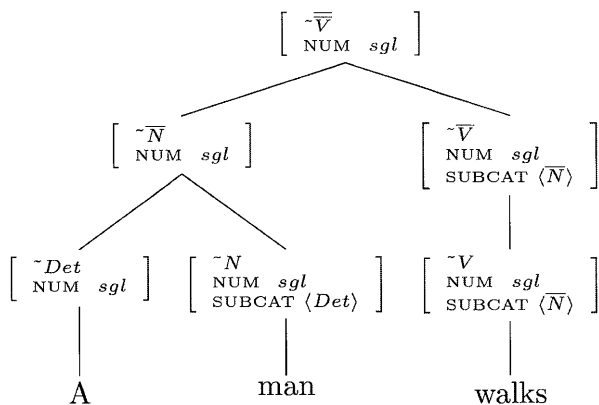


図1 自然言語文の木構造例

*5 以下、型名を特に表記する必要のない場合は省略する。

T, $\{\sim prep, \sim particle\} \sqsubseteq \sim indecl \sqsubseteq T$ のような型宣言が行われると、' $\sim N$ ' (名詞) も ' $\sim Adj$ ' (形容詞) も共通にこれらの素性を継承し、そのうえで独自の素性が付加される。一方 ' $\sim prep$ ' (前置詞) や ' $\sim particle$ ' (小辞) は変化しない語彙 ' $\sim indecl$ ' (indeclinable) である。

図中、主辞は型名の上にバーを付加して表現されている。すなわち 'man' の型は N であることにより、'a man' の型は N となっている。同様に、動詞句 (\bar{V}) を形成する主辞は (V) であるとし、文全体 (\bar{V}) は動詞句を主辞として形成されるとする。

また主辞である子がそうでない子と結びついて親を形成するプロセスを下位範疇化 (subcategorization) と呼ぶ。主辞 'man' が冠詞 'A' を結合する際、'walks' が 'A man' を結合する際がこれである。図1にあるように、ある型がどのような型を下位範疇化するかという情報を SUBCAT という素性にもっている。

4.3 文法規則

HPSG の文法規則は ID-スキーマとプリンシプル (原理) の二つから成り立っている。

ID-スキーマ (Schema) とは任意の二つの素性構造から親の素性構造を生成するという、いわゆる文法規則である。HPSG ではおのおののカテゴリーを素性構造で定義すると同時に、この規則自身も親と子を素性名とする一つの大きな素性構造によって表現するが、見やすさのため本稿では従来どおり ' \rightarrow ' を用いて表示する (5章の (1), (2) など)。この際、HEAD 素性には主辞の素性構造を、SUBCAT 素性にはそのカテゴリーが下位範疇化するカテゴリーの情報を、DTRS (daughters) 素性には主辞となる子 HEAD-DTR とほかの子 COMP-DTR (complement daughter) を格納する*6。

また、HPSG では ' \bar{i} ' のように記された四角囲みのインデックスによるポインタを素性構造に付加する (素性構造の角括弧の前に先行させて書く) ことによって、以降そのポインタがその素性構造全体を指すことができる。

ポインタを用いて異なる位置で同一の型や素性構造を共有することを構造共有と呼ぶ。つまりポインタ ' \bar{i} ' は構造共有のための変数とみなせる。

異なる二つの素性構造を一つの素性構造に統合する操作を単一化 (unification) と呼び、属性の値に矛盾のないときに限り可能である。例えば $\begin{bmatrix} f_1 & v_1 \\ f_2 & v_2 \end{bmatrix}$ と $\begin{bmatrix} f_2 & v_2 \\ f_3 & v_3 \end{bmatrix}$ は

単一化可能で結果は $\begin{bmatrix} f_1 & v_1 \\ f_2 & v_2 \\ f_3 & v_3 \end{bmatrix}$ となる。ところが、 $\begin{bmatrix} f_1 & v_1 \\ f_2 & v_2 \end{bmatrix}$

*6 本稿では素性構造を見やすく表示するために、直接関係のない素性は省略する。また、連続する素性の表示は ' \cdot ' を挟んで ' $F_1 \cdot F_2 \cdot F_3$ ' のように表記し、間に挟まれた素性を省略する際は ' $F_1 \cdot F_n$ '、最初の部分を省略する際は ' $\cdot F_n$ ' のように表記する。

と $\begin{bmatrix} f_1 & v_1 \\ f_2 & v_4 \end{bmatrix}$ の間では f_2 の値が異なるため単一化はできない。単一化は文法規則が辞書引きされた語に適用される際、あるいは別の文法規則が複数の句を結びつけてさらに大きな句構造を構成する際に用いられる。単一化の際は型どうしの単一化も求められるため、型階層の上下関係によって方向性がある。

プリンシプル (Principle) とは親の素性構造が満たさなければならないアприオリな制約のことである。プリンシプルもまた素性構造によって表現される。例えば、ヘッドフィーチャプリンシプルでは、親の HEAD 素性とその主辞である子の HEAD 素性は構造共有されなければならないということが表現されている。このプリンシプルによって主辞の子の重要な情報が親に伝えられる。

$$\left[\begin{array}{l} \text{HEAD } \boxed{1} \\ \text{DTRS} | \text{HEAD-DTR } \left[\begin{array}{l} \text{HEAD } \boxed{1} \end{array} \right] \end{array} \right]$$

また下位範疇化プリンシプルでは、親が下位範疇化すべきカテゴリーは主辞の子が下位範疇化すべきカテゴリーのうちすでに下位範疇化されたものを除いたものである、ということが表現されている。

$$\left[\begin{array}{l} \text{SUBCAT } \langle \boxed{2} \rangle \\ \text{DTRS} \left[\begin{array}{l} \text{HEAD-DTR } \left[\begin{array}{l} \text{SUBCAT } \langle \boxed{1}, \boxed{2} \rangle \end{array} \right] \\ \text{COMP-DTRS } \boxed{1} \end{array} \right] \end{array} \right]$$

このほかにもいくつかプリンシプルが存在し、これらの制約をもとに親の素性構造が生成される。

HPSG の構文解析とは、任意の二つの素性構造を空の素性構造の HEAD-DTR と COMP-DTRS 素性に格納し、それに対して ID-スキーマやプリンシプルを単一化し、親の素性構造を求めるプロセスである。構文解析プロセスでは隣り合う素性構造同士から階層的に親の素性構造を求める。このとき、親を求める順序や子のペアを指定する必要はない。ID-スキーマによって任意の子のペアから親を求めることが可能か順に調べられるため、解析の手順は自ずと決定される。

一番トップの親、文の素性構造が求められたときパーサは終了し、素性構造中に意味表現として論理表現 = 述語項構造 (Predicate-Argument Structure: PAS) が求められる。PAS そのものは図 1 の例でいえば walk(man) のような単純なものであるが、'walk' や 'man' という語彙項目が構文の中でどのような関係にあって求められたものであるかが文全体の素性構造によって同時に明示されていることに意義がある。また意味構造がこのような述語論理であるということは、テキスト処理の後の論理推論にそのままの形で適用できることになる。

5. ギリシア語の語順の問題

古典ギリシア語は、性・数・格の一致によって文構造が定まり、その結果語順に関する制約が緩くなっているという特徴がある。このような素性の一致を効率良く扱うために、HPSG を用いてみよう [中嶋 04, Tojo 05]。古典ギリシア語は基本は SOV 型言語であるが、この語順の制約は薄い。この語順の自由さにより、話者の注目するものを先へ先へともってくるという性質がある。この現象は動詞の位置について顕著である。その時点における話題の中でより重要なものが存在する場合、先行すべき補語が動詞の後ろに繰り下がってしまうことがある。また動詞の後に主語や補語を修飾する語が続く場合も往々にして存在する。また、通常補語に先行すると考えられる前置詞であってもその語順は固定されたものではなく、順序が逆転することがあるが、格の一致によって係り受けを定めることができる。

ある語が補語を取り句を形成することができるスキーマ Head complement schema は (1) のような形で記述される。

$$\left[\begin{array}{l} \text{phrase} \\ \text{COMPS } \langle \rangle \end{array} \right] \rightarrow H \left[\begin{array}{l} \text{word} \\ \text{COMPS } \langle \boxed{1} \rangle \end{array} \right] \boxed{1} \quad (1)$$

ここで COMPS (complements) は補語素性である。このスキーマは前置詞や動詞が補語を取る場合、形容詞が補語を取る場合*7などに用いられる。前置詞や一部の形容詞は基本的に右に補語を後置するが、動詞においては補語が左右どちらにも表れるためこのスキーマは 2 種類用意する必要がある。

句が特定の句を修飾することを定義するスキーマは Head modifier schema である。これは VALENCE 素性の中の MOD (modifier) 素性と被修飾句の HEAD 素性が単一化できるならば、修飾関係をつくることのできるという規則である。

$$\left[\begin{array}{l} \text{phrase} \\ \text{COMPS} \\ \text{MODS} \end{array} \right] \rightarrow H \boxed{1} \left[\begin{array}{l} \text{word} \\ \text{COMPS} \\ \text{HEAD } [\text{MODS } \boxed{1}] \end{array} \right] \quad (2)$$

このスキーマによって修飾関係をつくる例を図 2 に示す (ユークリッド『言論』第 8 巻, 命題 24 [Heath 25])。この図では数詞 δύο (two) が名詞 ἀριθμοί (numbers) を, πρὸς (to) によって率いられる前置詞句が名詞 λόγον (proportion) をそれぞれ修飾している*8。

同様に、分詞や形容詞が性・数・格の一致によって名詞に係ることや、副詞および接続詞によって導かれた文

*7 補語は文法書によっては「目的語」という言い方もするが、言語学一般の範疇として本稿では「補語」に統一する。

*8 この文は最後の動詞 ἔχουσιν (have) が接続法になっており, εἰάν (if) に続く節である。したがって正確には木のトップノードは文 S ではなく、節である。

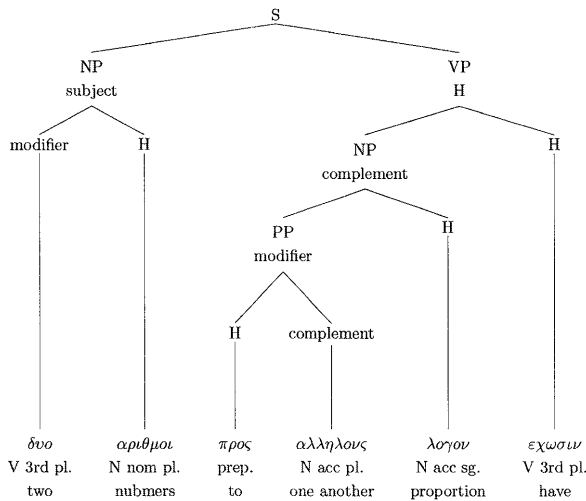


図2 形容詞が名詞を修飾する例

が動詞などに係ることもこのスキーマで扱われる。これらの修飾語句の係り方は以下のように多様である。

- (a) οἱ ἐλάχιστοι ἀριθμοί (the smallest number)
- (b) οἱ ἀριθμοὶ ἐλάχιστοι (the number smallest)
- (c) ἐλάχιστοι οἱ ἀριθμοί (smallest the number)

(a) は修飾語が属性的位置にあると呼ばれ、このように冠詞と名詞の間に修飾語が置かれる。このときは、被修飾語の SPR (specifier) 素性は飽和していない。一方 (b) と (c) はともに修飾語が述語的位置にあると呼ばれ、このように冠詞と名詞の組合せの外に修飾語が置かれる。このときには、被修飾語の SPR 素性は飽和している。ここで、冠詞は必ずしも出現する必要はないため SPR 素性の飽和・非飽和に関しては問わないこととするが、修飾語は修飾するときには自身は飽和している必要がある。

6. 進化ゲーム論的に見た素性一致の効用

[Jäger 03] は、能格 (ergative) 言語^{*9}の言語理解に対する格変化の効用をモデル化した。そこでは言語使用者の負担と伝達性の関係を効用関数を用いて定量化し、進化ゲーム理論による言語伝搬の解析を行っている。ここでは効用 (utility) を人称・数・格の一致まで広げて調べてみよう [Matoba 06, 的場 07]。話者は英語文か英語文の語順を入れ換えた文を話し、聞き手は素性の一致を考慮した文法か考慮しない文法で聞く状況を考える。

6.1 語順の入換え

人称・数・格の情報を失えば当然語順に対する制約が厳しくならなければならない。よって語順を入れ換える

ことによってこの効用を計測する。実験に適用した文は、コーパス中の文の語順を以下の手順で人工的に入れ換えたものである。語順の入換え方はランダムではなく次の2種類のルールで入れ換える。

- ルール 1：SVO 語順を SOV 語順に入れ換える
- ルール 2：SVO 語順を OSV 語順に入れ換える

ルール 1 はスペイン語やイタリア語などのラテン語から派生した言語に見られる。また、ドイツ語やオランダ語のような SOV-V2 語順を取るドイツ語派生の言語もこの語順を取る。ルール 2 は長距離離依存と呼ばれ、英語の関係詞節や疑問形などで取られる語順である。これら二つのルールに従い、主語と目的語を取る動詞 “love” による文 “[Subject] love [Object]” は “[Subject] [Object] love” や “[Object] [Subject] love” の語順に入れ換わる。また、これらの文を受理できるよう英文法用 HPSG の一部のスキーマを改良した文法を導入した。

ここで実験に使用したパーサは特徴として、任意の素性の単一化を無視でき、かつ語順を入れ換えた文に対応することができる。パーサの出力は述語項構造 (Predicate-Argument Structure: PAS) である。本研究においては、この述語項構造を出力することを文の理解と定義する。この述語項構造が1文に対して複数個出力されるとき、その文が曖昧であるとする。一つ目の特徴により、“she love me” や “her love me” といった数や格に誤りを含む文でもその素性を無視することによりパーサ可能となる。

例として、素性の単一化を考慮したときと無視したときに現れる解析結果の違いを示す。対象とする文は、“she loves me” を前章で説明した手順で入れ換えた “me she loves” とする。まず、文法の素性単一化を考慮した場合、“me” は対格、“she” は主格とわかるため文が表す意味は一意に定まり曖昧性はない。一方、一致素性の単一化を無視した場合、“I love her” または “she loves me” の二つの解釈ができるようになり曖昧性が発生する。

実験では、通常の英語文の集合 (Normal) と語順を入れ換えた文の集合 (Scrambled) の2種類を適用した。また、二つの文法を適用した。一つは人称、数、格を考慮した通常の英文法 (Person, Number, Case: PNC) であり、もう一つは格についての単一化を無視した文法 (Person, Number: PN) である。

6.2 効用値による人口流入

我々は、1文のもつ曖昧性と屈折に関わる素性の単一化にかかるコストから文の効用値を計算する関数をモデルに導入した。この効用値は Jäger が提案した効用に基づいている。1文のもつ効用値 U は

$$U = \frac{N}{|PAS|} - w \cdot C(unif)$$

$|PAS|$, N , $C(unif)$, w はそれぞれ PAS の数、文長、語の屈折に関わる素性に関する単一化の回数、 $C(unif)$ の

*9 自動詞・他動詞を問わず主語は主格で表され、他動詞の目的語は別途目的格で表示される言語は絶対格言語と言う。これに対して能格言語は自動詞の主語と他動詞の目的語が同じ形式で表示される。

ウェイトである。

PASの数は曖昧性を表しており、式(3)の第1項目は利得を示している。この値が大きければ曖昧性が小さいことを意味する。第2項目は語の屈折に関わる素性を単一化することによってかかるコストを表している。ウェイト w は正の係数で、第2項目の優先度を操作するパラメータである。

我々は、話者、聞き手がそれぞれの方策を取ったときにどの解に安定するかを計算した。本研究での話者の方策は、英語文を話すNormalと、語順入れ換え文を話すScrambledである。一方、聞き手の方策は、人称、数、格を考慮した文法で解析するPNCと、格を無視した文法で解析するPNである。本研究でのプレイヤーは話者と聞き手であり、利得行列は2行2列となる。また、各要素は、取り得る4種類の方策の組(Normal, PNC), (Normal, PN), (Scrambled, PNC), (Scrambled, PN)でコーパスを解析して得られた効用値の平均値である。

ウェイト $w=0$ は、聞き手が屈折に関わる素性を考慮するためのコストに無頓着な状態である。つまり、聞き手は屈折に係わる素性を通常から母語として使っており、これの使用になら負担を感じない状態にあると考えることができる。一方、ウェイト $w=0.1$ は、聞き手が屈折に関わる素性の単一化にコストを払わなければならない状態である。つまり、第二言語学習者のように聞き手がこのような素性を扱うことに不馴れな状態にあると考えることができる。

図3, 図4は、話者・聞き手それぞれの利得行列を用いたレプリケータ・ダイナミクスと呼ばれる方法

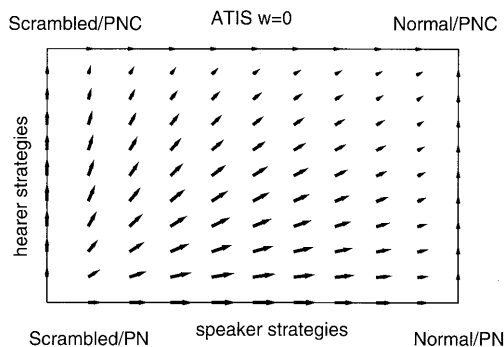


図3 $w=0$ での人口遷移

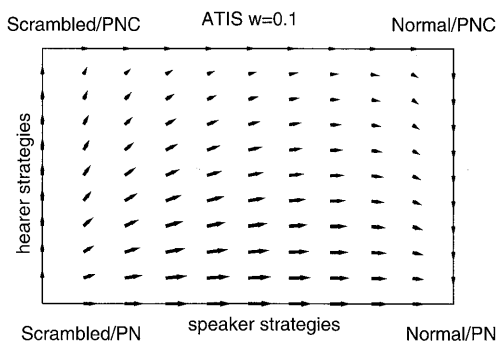


図4 $w=0.1$ での人口遷移

[Jäger 03]の位相図であり、 Δt ごとに変遷した話者と聞き手の人口を示している。ウェイト $w=0$ では収束点は(Normal, PNC)であるが、ウェイト $w=0.1$ では収束点が(Normal, PN)に移動したことが確認できる。

この実験より、語順を入れ換えた状態では語形変化に関する素性があったほうが効用値が高くなる。また第二言語話者のように素性一致にコストを考慮したモデルでは、聞き手は格変化を無視したほうが効用値が高くなり、異言語話者間での言語衝突時にはピジン(pidgin)などのように素性喪失が起こることが考えられる。

◇ 参考文献 ◇

- [Dowty 79] Dowty, D. R., Wall, R. E. and Peters, S.: *Introduction to Montague Semantics*, Synthese Language Library, D. Reidel (1979)
- [言語学大辞典 95] 亀井 孝ほか 編: 言語学大辞典第6巻—術語編—, 三省堂 (1995)
- [Heath 25] Heath, T. L.: *The Thirteen Books of Euclid's Elements*, Dover Publishing, Inc. (1925)
- [Hindley 86] Hindley, J. R. and Seldin, J. P.: *Introduction to Combinators and λ -Calculus*, Cambridge University Press (1986)
- [Jäger 03] Jäger, G.: Evolutionary games theory and typology: a case study, *Proc. 14th Amsterdam Colloquium*, pp. 108-117 (2003)
- [Matoba 06] Matoba, R., Nakamura, M. and Tojo, S.: Utility for communicability by profit and cost of agreement, P. Vogt et al. (eds.), *Symbol Grounding and Beyond*, LNAI 4211, Springer (2006)
- [的場 07] 的場隆一, 中村 誠, 東条 敏: 効用値からみた文法伝播: 屈折による曖昧性除去と適用コストのバランスから, 日本認知科学会第24回大会 (2007)
- [中嶋 04] 中嶋健一郎, 斎藤 憲, 東条 敏: HPSGを用いたユークリッド「原論」の解析, 自然言語処理, Vol. 11, No.1, pp.21-41 (2004)
- [Pollard 94] Pollard, C. and Sag, I.: *A Head-driven Phrase Structure Grammar*, University of Chicago Press (1994)
- [Sag 03] Sag, I., Wasow, T. and Bender, E.: *Syntactic Theory — A Formal Introduction —*, Second Edition, CSLI Publications (2003)
- [Tojo 05] Tojo, S. and Saito, K.: *Analysis of the Elements by HPSG*, PACLIC (2005)
- [東条 06] 東条 敏: 言語・知識・信念の論理, オーム社 (2006)

2007年6月20日 受理

著者紹介



東条 敏 (正会員)

1981年東京大学工学部計数工学科卒業, 1983年同大学院工学系研究科修士課程修了。1983~95年(株)三菱総合研究所。1995年北陸先端科学技術大学院大学情報科学研究科助教授, 2000年同教授。博士(工学)。自然言語の形式意味論および人工知能の論理の研究に従事。情報処理学会, 日本ソフトウェア科学会, 言語処理学会, 日本認知科学会, FOLLI各会員。