

# ワードグラマーの意味論の実現

中西克明

東大 言語情報 M2

k.naka@phiz.c.u.tokyo.ac.jp

## 1. はじめに

人工知能や認知諸科学の分野では、意味ネットワーク、特に多重継承とデフォルトに元づくそれ(以下 MDI ネットワーク)による知識表現が広く用いられている。また認知言語学の一般的な主張は、人間の能力のモジュール性を否定する。即ち、言語とその他の、或いは統語論と意味論の処理は明確な区別を持たないとされる。

従来、依存文法による英語の統語理論として有名であったワードグラマー[1](以下 WG)は、近年これらの議論を採用し、MDI ネットワークを表現手段として用いた、統語論に限らない(但し現状では統語論のほかには意味論の研究が殆どで、本稿でも意味論の解析を扱う)総合的な言語解析の理論へと発展してきている。MDI ネットワークで表わされた言語解析の例が与えられ、魅力的な言語学的妥当性を持つが、そのような解析を与える形式的手続きの議論が、基本的な方針の提示を除いて殆どなされていないという研究の現状は、二つの意味で残念だ。解析過程が計算機上で実現しないことによる実用上の意味と、形式的な再現性をもたない解析は客観性を欠いていると言う理論上の意味において、である。

本稿では、WG の解析過程を形式化するための理論上の問題点を指摘し、それらに対する解決案を提示する。また、それらの解決案に基づき与えられた形式化を、計算機上で実装し、実際に文を処理して見た結果を報告することで、本稿の提案の有効性を確認する。

## 2. WG とは何か?

WG は UCL の Richard Hudson による言語解析の理論であり、依存文法による英語の統語理論として有名である。ここでは MDI ネットワークを取り入れ、総合的な理論へと発展した現在の WG について概略を説明し、この理論に対する本稿の位置付けを明確にする。

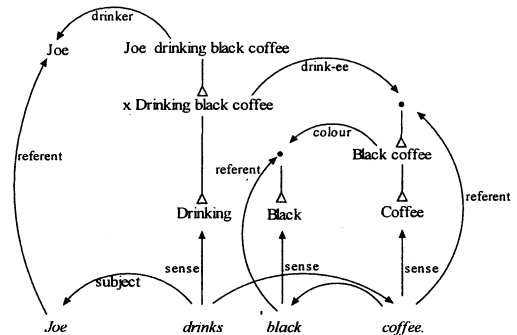


図1 (Hudson (2001) より引用)

### 2-1. WG のネットワーク意味論

WG で実際に提示された[2]解析の例を図1に引用した。WG で使われるネットワークは ISA-A, HAS-A による単純な MDI ネットワークである。名前付きの有向アーク(以下リンク)は HAS-A 関係を表わし、一方が三角形の一つの頂点に接続する線分(以下 IS-A リンク)は三角形のある方のノードをスーパークラスとする IS-A 関係を表わしている。図1のネットワークで最下部のノードの列は入力文の単語の並びを表わしており、それらノード間のリンクはヘッドからディペンデントへと伸びる単語間の依存関係に対応している。これは従来の WG の依存文法が MDI ネットワークの一部としてそのまま統合されたものである。他のノードはそれぞれが一つの意味を表わしている。単語からの sense リンクの先はその単語自体の持つ基本義を示し、referent リンクの先は、その単語のその文中での指示対象を表わす。その単語をヘッドとする最大の句の意味と考えてもよい。

ノードの担う意味は、そのノードから伸びるリンク、そのノードへと伸びる IS-A リンクにより決まる。例えば図1のネットワークで、ノード Joe drinking black coffee の担う意味は、それが x

Drinking black coffee の IS-A であるということ、その drinker が Joe であるということ定まる。

## 2-2. Semantic Phrasing

WG では基本的に、各単語の sense は語彙情報として指定され、referent は各文脈において Semantic Phrasing[3]という仕組みで決定されることになっている。この Semantic Phrasing とはフラットな依存構造である統語論に対し、意味論はある種の句構造を為すという主張で、次のような過程で各単語の referent が構成的に作られるとするものである。即ち、或る単語をヘッドとする句の意味は、その単語の基本義である sense から始まり、各ディペンデントと結合する度に、そのディペンデントの referent へと HAS-A 関係が伸びるなどにより一つ一つ特殊化され、そうして全てのディペンデントと結合して得られた意味の更に IS-A としてその単語の referent が作られる。

例えば図 1 のネットワークで *drinks* の意味は先ずその sense、即ち Drinking であり、*coffee* を目的語として取ることで、*coffee* の referent を drink-ee として持つ Drinking の特殊化、x Drinking black coffee となる、そして *Joe* を主語として取ることで *Joe* の referent である Joe を drinker として持つ x Drinking black coffee の特殊化、Joe drinking black coffee となっている。

## 2-3. 本稿の位置

WG の解析のうち依存文法による統語解析部は、従来の WG の統語解析を援用しており、それは既にパーザーが可能に形式化されている[4]。本稿は入力文に対する統語解析結果は文と共に入力として与えられるものとして考え、各単語に対してどのような語彙情報を与え、どのような形式的処理を施すことにより、WG で提示されているような意味ネットワークを生成できるのかを議論するものである。

## 3. 解析過程を形式化するにあたっての問題点

### 3-1. 語彙情報とその扱いにおける不備

語彙情報の中に変数ノードを許し、ディペンデントにより意味が特殊化される過程を一般的な形で記述しておく。そして、その語が出現した各文脈において、変数ノードがバインドされ具体的な意味の特殊化が導かれる。WG が意味解析過程として想定しているのはこうした処理である。例えば図 2 に引用したような語彙情報から図 3 に引用したような解析が導かれるとされる[5]。この議論には以下の問題点がある。まず、ディペン

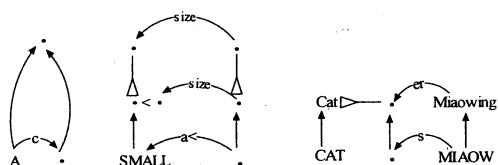


図 2 (Hudson (1998) より引用)

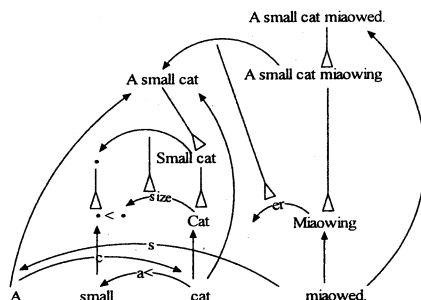


図 3 (Hudson (1998) より引用)

デントにより特殊化された単語の意味を、いずれかのノードに担わせることは出来ても、そのノードをその単語の特殊化された意味として指定する手段が考えられていないという問題がある。さらに、コンプリメントによる意味の特殊化はヘッドの語彙情報として書き得ても、アジヤントによるそれは定義上ヘッドの語彙情報としては書き得ず、またアジヤントの語彙情報の中でヘッドの意味の変更を指定する手段も与えられていないという問題がある。

具体例で説明しよう。図 2 の SMALL の語彙情報はヘッドの sense に対し、その size を指定した特殊化を与えているが、この特殊化のノードをヘッドの新しい意味として指定してはいないため、多数在るノードの内、どのノードに変更された意味が表現されているのかということは形式的には知り得ない。又、一般的には SMALL と結合する時点でヘッドが既に別のディペンデントで特殊化されているかも知れず、その場合、SMALL によって変更されるべきヘッドの意味はその sense では無く、別のノードであるはずだ。

### 3-2. その他の問題

また以下の問題も解決しなくてはいけない。WG においては言語知識と解析結果は同一のネットワーク内に表わされる、即ち、解析とは言語知識を表わす永続的ネットワークに対する一時的な拡張として定式化できるが、一般に同一の語が複数回

文中に出現し得るため、語彙項目のノードをそのまま各出現として使うことが出来ない、という問題である。

更に、次の問題もある。一つのノードから同じ名前のリンクが複数であるようなネットワークが WG の議論では意味論においても散見されるし、統語論においては一つの語が複数のアジャント(アジャントは全て a という名前のリンクで表わされる)を持つことは日常的であるが、そのことを認めてしまうと、名前と出発点のノードによるリンクの同定が出来ず、変数ノードのバインディングや、継承したリンクのオーバーライドの議論がそもそも無効になってしまうという問題である。

## 4. 解決案

### 4-1. before, after, modified\_sense

3-1 の問題に対し以下の解決を提案する。まず、単語の語彙情報の中で、単語から modified\_sense という名前のリンクが意味ノードへ伸びることを許す。このリンクはコンプリメントと結合することによって特殊化されたその単語の意味を指定する。これにより、例えば drinks の語彙情報は図 4 のように書ける。

又、アジャントの語彙情報の中で、before, after という名前の二つのリンクが意味ノードへ伸びることをゆるす。before リンクの先は変数ノードであり、アジャントとして機能する際のそのアジャントによる変更前のヘッドの意味を示す。また after のリンク先は、このアジャントによる変更後のヘッドの意味を指定する。従って、通常 after は before の IS-A となる。これにより例えば black の語彙情報は図 5 のように書ける。

これらのリンクの導入により、Semantic Phrasing の過程は次のように形式化できる。Semantic Phrasing とはディペンデントをもたない単語から文全体のヘッドへとボトムアップ的に単語の意味を「完了」させていく過程であり、全てのディペンデントの意味が「完了」しているような単語に対し、その意味を次のように「完了」する。即ち、まずその単語の語彙情報内の変数を実際のコンプリメントにバインドし、その単語の modified\_sense を求める。次に各アジャントに対し、

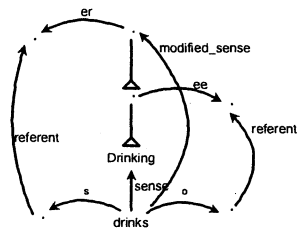


図 4

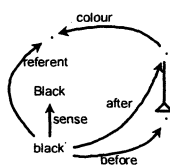


図 5

modified\_sense をそのアジャントの before にバインドし、そのアジャントの after をヘッドの新しい modified\_sense とする。これを全てのアジャントに対して行い、求められた modified\_sense の IS-A となるようなノードをその単語の referent とする。

### 4-2. 同一の語の複数回の出現

[6]などでは文中の各単語出現を w1, w2... という一時ノードとして導入し、各 wn を対応する語彙項目から継承させるという解析が与えられている。これは 3-2 前半の問題の解決として使える。しかし、その場合新たな問題が出てくる。語彙情報内の変数ノードは、それぞれの出現において異なるバインディングがなされなくてはいけない。例えば "A dog saw another dog." という文において二つの dog の出現の referent は異なるノードでなくてはならないが、この二つの dog の出現の referent はともに語彙項目 dog の referent を継承したものである為等しく、異なるノードになれないという問題である。

この新たな問題に対しては、単語の出現が語彙項目から継承する際、語彙情報内の変数をその出現に対して同等の効果をもつように複製するという解決を提案する。そうすれば各出現に対して異なる複製が使われるため、バインディングの干渉は起きない。

### 4-3. 同一のノードから同名のリンクが複数伸び得る問題

3-2 後半の問題に対して次のような解決を提案する。即ち、同一のノードから同名のリンクが複数伸びることは、統語解析における a リンク以外一切認めない。これは、リンクを出発点と名前により同定することは、MDI ネットワークの、そしてバインディングの基礎であり、これを変更することは出来ないからだ。ここで統語解析において、a リンクが同一の単語から複数伸びることを認めることは問題ないことに注意したい。何故なら a リンクの効果は実際の処理では before, after による前述の手法で処理される為、このリンクに対してバインディングなどの操作は一切行われなければならないからである。

## 5. 実装

以上の問題解決に基づき、WG の解析過程を形式化し、それを計算機上で実装した。実際に動作を確認し、本稿の議論の有効性を確認したので、その実装と動作について報告する。

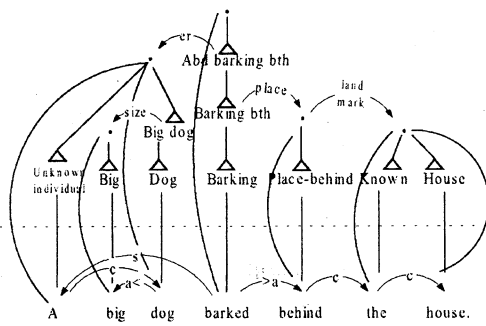


図 6 (Hudson (2000) より引用)

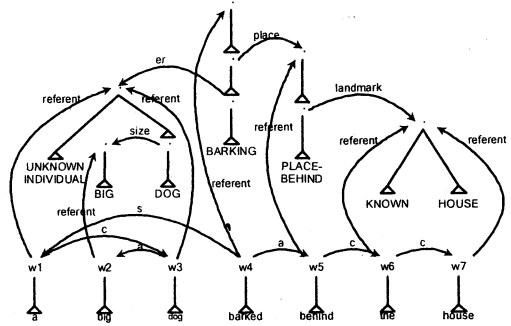


図 7

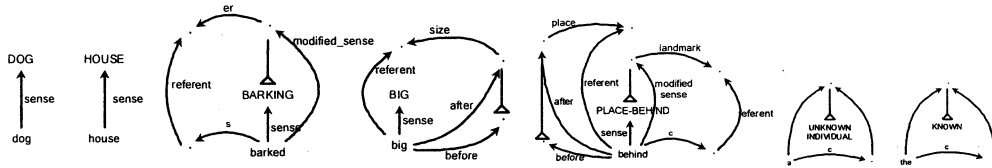


図 8

実装はオブジェクト指向型のコンピューター言語 Eiffel[7]を用いて行われた。語彙情報をデータとして与え、任意の文を統語解析と共に入力すると、その解析結果を出力する。ネットワークの入出力は、あるフォーマットに従いネットワークを表わすテキストデータで行われる。

小規模の語彙情報のネットワークを構築し、さまざまな文を成功裏に解析した。一例として“a dog barked behind the house.”の解析結果を報告する。

WG で提示されている解析例の中で最も大きなものの一つがこの文の解析例[8]であり、図 6 に引用した解析結果が与えられている。この文に出る単語について、自然に図 8 に示したような語彙情報が考えられる。それらとこの文の統語解析結果を入力したところ、図 7 に示したような解析結果が出力された。図 6 と図 7 は殆ど同一のものであり、本質的な違いは barked が Semantic Phrasing において結合するディペンデントの順序、即ち前者ではまず behind と結合して、次に主語と結合しているが、後者では逆であるという点だけである。

## 6. 今後の課題

今回動作を確認した限りでは処理時間の問題は生じなかったが、このシステムを本格運用する場合、言語知識を表わすネットワークが大規模なものになるにつれ、処理時間が指数関

数的に増すことが予想される。今後は今回無視した実装の効率化についても考える必要があるだろう。

また、今回の実装では Semantic Phrasing においてディペンデントと結合する順序は、まずコンプリメントで次にアジェンクトという順番に限定されてしまい、またアジェンクトが複数ある場合の処理順序も特に考慮されていないが、実際にはディペンデントとの結合の順序は重要であり、今後は妥当な結合順序について議論し、またそのような順序での意味生成を実現できるように形式化を改良したい。

## 参考文献

- [1] Hudson, Richard (1984) *Word Grammar*, Blackwell, Oxford.
- [2] Hudson, Richard (2001) “Constructions in Word Grammar” [ftp://ftp.phon.ucl.ac.uk/pub/Word-Grammar/constructions.zip](http://ftp.phon.ucl.ac.uk/pub/Word-Grammar/constructions.zip)
- [3] Hudson, Richard (1990) *English Word Grammar*, Blackwell, Oxford.
- [4] Fraser, Norman (1993) *Dependency Parsing*, PhD thesis, UCL, London.
- [5] Hudson, Richard (1998) “Word Grammar” [ftp://ftp.phon.ucl.ac.uk/pub/Word-Grammar/hcl.pdf](http://ftp.phon.ucl.ac.uk/pub/Word-Grammar/hcl.pdf)
- [6] Hudson, Richard (2002) “Word Grammar” [ftp://ftp.phon.ucl.ac.uk/pub/Word-Grammar/intro.zip](http://ftp.phon.ucl.ac.uk/pub/Word-Grammar/intro.zip)
- [7] Mayer, Bertrand (1997) *Object-Oriented Software Construction*, 2<sup>nd</sup> edition, Upper Saddle River, NJ.
- [8] Hudson, Richard (2000) “PLINS203 Word Grammar 2000-1” [ftp://ftp.phon.ucl.ac.uk/pub/Word-Grammar/lex2001.pdf](http://ftp.phon.ucl.ac.uk/pub/Word-Grammar/lex2001.pdf)