社団法人　情報処理学会　研究報告
IPSJ SIG Technical Reports

2005－ＮＬ－168（19）
2005／7／23

頑健なHPSGパーザの出力からTDL意味表現への変換手法

佐藤　学*　戸次　大介**　宮尾　祐介†　辻井　潤一†‡

*東京大学大学院情報理工学系研究科コンピュータ科学専攻　〒113-0033 東京都文京区本郷 7-3-1
**東京大学 21 世紀 COE「心とことば－進化認知科学的展開－」
〒153-8902 東京都目黒区駒場 3-8-1
† 東京大学大学院情報学環　〒113-0033 東京都文京区本郷 7-3-1
‡ School of Informatics, University of Manchester　POBox 88, Sackville St, MANCHESTER M60 1QD, UK
E-mail: {sa-ma, becky, yusuke, tsujii}@is.s.u-tokyo.ac.jp

あらまし　本論文では、実テキストから高い被覆率で表現力の高い意味表現を得る手法として、頑健なHPSGパーザの出力からTDL意味表現を得る変換規則を提案する。また、実テキストからTDL意味表現を導出する実験について報告し、変換規則の実装の進捗状況と今後の見通しについて述べる。

キーワード　計算意味論、頑健なパーザ、HPSG、TDL、動的意味論

Method of transformation from the output of a robust HPSG parser to TDL semantic representation

Manabu SATO*　Daisuke BEKKI**　Yusuke MIYAO†　and　Jun'ichi TSUJII†‡

*Department of Computer Science, University of Tokyo　Hongo 7-3-1, Bunkyo-ku, Tokyo
**Center for Evolutionary Cognitive Sciences at the University of Tokyo
† Interfaculty Initiative in Information Studies, University of Tokyo　Hongo 7-3-1, Bunkyo-ku, Tokyo
‡ School of Informatics, University of Manchester　POBox 88, Sackville St, MANCHESTER M60 1QD, UK
E-mail:　{sa-ma, becky, yusuke, tsujii}@is.s.u-tokyo.ac.jp

Abstract　In this paper we propose a set of rules which map outputs of a robust HPSG parser to TDL semantic representations, as a method to acquire rich semantic representations from real texts with high coverage. We will present the progress of our implementation of the method by reporting the experiments to derive TDL semantic representations from real texts, and make some remarks on the future development.

Keyword　computational semantics, robust parser, HPSG, TDL, dynamic semantics

1. はじめに

タグ付きコーパスを利用して頑健で高精度な構文解析器を作成する研究が進んでおり、様々な文法理論に基づく構文解析器が既に開発されている[8][9][13]。これらの構文解析器は、実テキストを高い精度で構文解析することに成功している。しかしながら、これらの文法理論は言語学的には意味論的解析を視野に入れているものの、実装された解析器の出力は、主に構文木のような統語的情報であり、意味論的情報は述語項構造にとどまっている。

そこで、高い被覆率で、より高度な意味論的解析結果を得るために、Bosらは、コーパスより学習した頑健なCCGパーザの出力から、一階述語論理に基づく意味表現を導出する手法を提案した[3]。その手法は、CCGパーザの語彙項目ごとに、対応する意味表現を与える規則と、意味表現の合成規則を定義し、パーザの出力する構文木に沿って意味表現を合成していく、というものである。この手法はPenn Treebankコーパスにおいて、CCGパーザが解析に成功した文の97%以上に対して一階述語論理の論理式を割り当てることに成功し、実テキストに対して高い被覆率で意味表現を導出することが可能であることを示した。しかし、一階述語論理に基づく意味論は、照応、量化等の言語現象に関する十分な表現力を持たないことが知られている。自然言語に対してより高度な意味解析を施すためには、古典的述語論理に基づく意味論ではなく、照応、量化等を扱うことのできる動的意味論を導出することが求められている。

本研究では、Bosらの手法に準じて、Miyaoらの開発した頑健なHPSGパーザ[9]の出力から、型付き動的論理(Typed Dynamic Logic: TDL)[1][2]に基づく意味表現を導出する手法を提案する。具体的には、文のTDL意味表現を導出するために、HPSG語彙項目をTDL意味表現に

対応させる規則と、HPSG 構文木に基づいて TDL 意味表現を合成する規則を定義する。また、現在、この手法の計算機への実装を進めており、実際に実テキストから TDL 意味表現を導出する実験を通して、その進捗と見通しを示す。

TDL は動的意味論の一種であり、照応や量化の問題を統一的に扱うことのできる論理系である。このため、自然言語の意味の構造をよりきめ細かに捉えることができ、高度な推論等への応用が期待できる。また、重要な性質として、TDL は構成性(compositionality)を持つことが挙げられる。語彙化文法の語彙項目に TDL 意味表現を持たせ、各語の TDL 意味表現を合成することにより、文全体、文章全体の TDL 意味表現を構成することが可能になる。更に、TDL では意味表現を導出するための統語理論が提示されていることも、本研究にとって利点である。しかしながら、TDL の統語理論に基づくタグ付きコーパスが存在しないために、現状では、既存の頑健な構文解析器の作成手法に則って、TDL の構文解析器を作成することは不可能である。また、新たにタグ付きコーパスを作成するには莫大な費用が必要である。そこで、本研究においては、TDL 意味表現を頑健な HPSG パーザの出力から導出する。

HPSG[10]は、少数の合成規則からなる語彙化文法であるため、TDL に関する合成規則を定義する際にも、少数の合成規則で済むという利点がある。更に、Miyao らのパーザは、高い精度と被覆率を持つ上に、その語彙項目は、屈折の情報や、詳細な一致制約の情報等の、言語学的な情報量に富んでいる。また、TDL 意味表現は統語表現よりも詳細な情報を持つため、TDL 意味表現を統語表現から導出するにあたっては、できるだけ多くの言語学的情報を必要とする。この点においても、複雑で情報量の大きい語彙項目を持つ HPSG パーザは有効である。以上の点から、HPSG パーザは、高い被覆率と精度での TDL 意味表現の導出を達成するのに適した選択肢であると考えられる。

一方、コーパスを利用せずに、人手で構築された文法による構文解析器では、意味論的解析に取り組んでいるものが存在する。例えば、LinGO プロジェクト[7](http://lingo.stanford.edu)で開発されている HPSG パーザは、統語的情報とともに、MRS 意味表現[4]を出力する。しかし、これらのパーザは頑健性に劣り、また MRS 意味表現は、そのままでは動的意味論に拡張することが難しいため、実テキストから高い被覆率で表現力の高い意味表現を出力するには至っていない。

2. 背景

2.1. 動的意味論と型付き動的論理(TDL)

2.1.1. 動的意味論の必要性

自然言語には、動的でない古典的述語論理では記述できない現象が存在することが知られている。Evans によって指摘された E-type 照応もその一つである[5][6]。

(1) Few boys fell. They died.

(1)の文では、They が、前の文に現れる Few boys と照応関係を持っている。この文の意味は、自然言語ではインフォーマルに「ごく少数の少年が落ちた。落ちた少年達は死んだ」のように言い換えられよう。ところが、古典論理ではこの文の意味を表現できない。

古典的述語論理は、文単位の意味を表現することを念頭に置いたものであり、基本的に量化詞のスコープは文境界を越えられない。そこで(1)の文の意味を古典的述語論理で(1')のように表現してしまうと、(1')では They の意味に当たる変数 x(die(x)中の x)が量化詞 Few のスコープの外に出現してしまっているために、They と Few boys の間に照応関係があることが表現されないのである。

(1') Few x.(boy(x)∧fall(x))∧die(x)

では上述の問題を解決するために、単純に量化詞 Few のスコープを広げてみてはどうだろうか。

(1'') Few x.(boy(x)∧fall(x)∧die(x))

しかし、今度はこの記述と(1)の表現する意味が合致しないという問題が生じる。(1'')は自然言語を用いて言い換えると、「落ち、なおかつ死んだ少年は少数だった」となる。これは、例えば「多くの少年が落ちたが、そのうち死んだ者は少数であった」ような状況でも真になる。しかし、(1)の文はこの状況では偽となる。このように、古典的述語論理を用いると、量化詞のスコープを複数の文にまたがって広げたとしても、(1)の照応関係を捉えることができないのである。

一方、動的意味論の一つである TDL では(1)の意味を以下のように表現する。

(1''') few(x)[boy(x)][fall(x)]∧ref(x)[ ][die(x)]

TDL では、命題(例えば、文)は、指標とそれに対応する要素の結びつきを受け取り、その中から、自らの持つ制約を充足する指標と要素の結びつきを出力する関数だと考える。この例においては、第一文において、「ごく少数の少年が落ちた。」が示す要素と指標 x が結び付けられ、第二文ではこの x を参照することができる。この方式により、文境界を越えた変数も指し示すことができる。このように、照応や量化に関する問題を含む文の意味を記述するには古典的述語論理では不十分であり、動的意味論が必要であることが知られている。

2.1.2. TDL 意味表現を導出する文法

[2]では、TDL 意味表現を導出する文法が、CCG[11][12]に基づいて定義されている。例として、"runs" の語彙項目が図 1に示されている。語彙項目中の要素は上から順に、音声、範疇、TDL 意味表現を示す。TDL 意味表現は、型付きラムダ計算によって表される。TDL 意味表現同士の合成には関数適用、関数合成等いくつかの方法があり、適用可能な合成規則は範疇によって定められる。

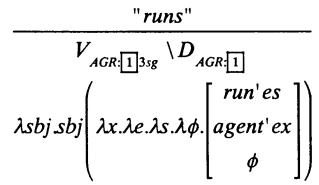

図 1　TDL 文法の語彙項目

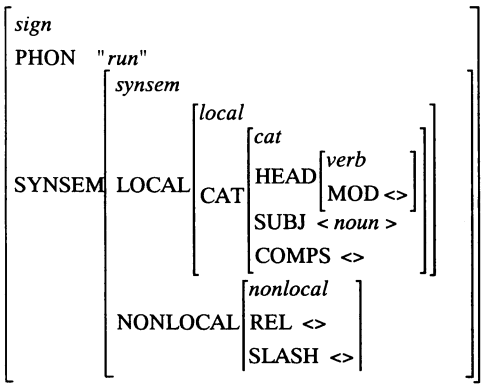

図 3　HPSG の語彙項目

詳しい方式は[2]を参照されたい。図 2は、"John"と"runs"の TDL 意味表現の合成の例である。二つの語彙項目の範疇の関係から、ここでは後ろ向きの関数適用が行われる。その結果にβ簡約が施されることにより、図の下の、合成後の TDL 意味表現が得られる。

2.2. Head-driven Phrase Structure Grammar

HPSG は、語彙化文法の一種であり、スキーマと呼ばれる少数の合成規則と、多数の語彙項目からなる。スキーマと語彙項目は、いずれも型付き素性構造と、それらの間に定義されている単一化という操作によって記述される。図 3は HPSG 語彙項目である。HEAD 素性は語句中の主辞の文法的性質を示す。MOD 素性の値にはこの語彙項目が修飾する語句の制約が記述される。SUBJ 素性、COMPS 素性にはそれぞれ、この語彙項目の左の項、右の項の制約が記述される。また、REL 素性、SLASH 素性には、関係節による制約や、長距離依存する語句の文法的制約が記述される。図 4はスキーマの例である。同じ番号のエントリーは構造共有している。このような合成規則により、構文木が作られる。

パーザによる構文木の導出の例が図 5に示されている。まず、各語("John", "runs")の音声表現から、対応する語彙項目が割り当てられる。次に、"John"に対応する語彙項目と"runs"に対応する語彙項目が Subject-Head スキーマによって合成される。大きい木の場合も、この操作を繰り返すことによって、最終的な構文木が合成される。

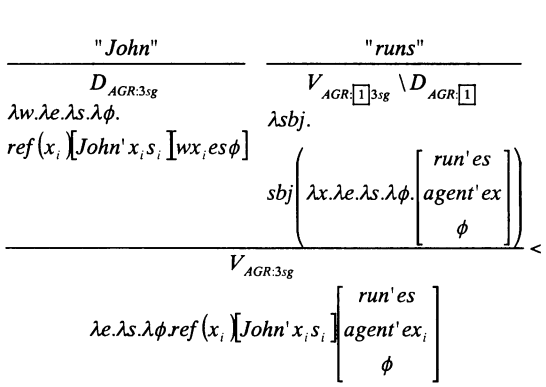

図 2　TDL 意味表現の合成

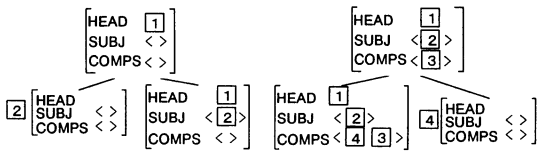

図 4　スキーマの例：Subject-Head スキーマ(左)と Head-Complement スキーマ(右)

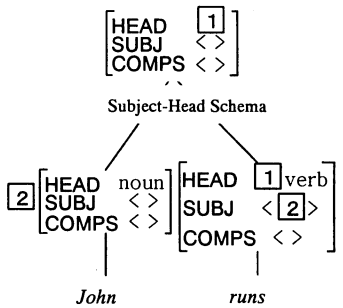

図 5　HPSG 構文木

3. HPSG 構文木からの TDL 意味表現導出方法

HPSG 構文木から TDL 意味表現を導出する手法は、Bos らに倣う。すなわち、HPSG 構文木のノードに TDL 意味表現を割り当て、それらを合成していく。しかしながら、TDL 文法は HPSG と異なる文法理論であるため、TDL 文法と同様の合成規則では、HPSG 構文木に沿って TDL 意味表現を合成し、新たに適切な TDL 意味表現を構成することができない。そのため、HPSG 構文木に沿った TDL 意味表現の合成規則を定義する必要がある。この合成規則は、HPSG と TDL 文法の差異を吸収するものであるが、その際、TDL 意味表現と HPSG 項目以外の情報が必要となる。したがって、それらの情報も HPSG 構文木のノードに与えなければならない。TDL 意味表現と、合成のために必要な追加情報をまとめて、TDL 構造(TDLS)とよぶ。TDLS については、3.1

$$\begin{bmatrix}\text{Syntactic Cat}\\ \text{Logical Form}\\ \text{NonLocal}\\ \text{Pred}\end{bmatrix}\begin{bmatrix}N\\ \lambda x.\lambda s.\lambda\phi.\begin{bmatrix}student'\,xs\\ \phi\\ [\,]\end{bmatrix}\\ \text{"}student\text{"}\end{bmatrix}$$

図 6 TDLS の例

節で詳しく述べる。

これらをまとめると、HPSG 構文木から TDL 意味表現を導出するためには、まず、以下の規則を定義する。

1. HPSG 語彙項目へ TDLS を割り当てる規則
2. HPSG 構文木に沿って TDLS を合成する規則

そして、これらの規則を用いて、再帰的に HPSG 構文木のノードに TDLS を割り当てていく。すなわち、HPSG 構文木の根ノードから始まって、

1. そのノードが中間ノードであるならば、そのノードの子ノードの TDLS を合成規則によって合成し、それをそのノードの TDLS とする。
2. そのノードが終端ノードであるならば、そのノードの HPSG 語彙項目に対応する TDLS を割り当てる。

この手順を繰り返して、構文木の頂点に対応する TDLS を求め、文の TDL 意味表現を得る。

以下の節では、手法の詳細について述べる。3.1 節では、TDLS に関して説明する。3.2 節、3.3 節では、TDLS の割り当て規則と合成規則を説明する。以後の節では、特殊な処理が必要な諸問題について述べ、解決法を提示する。3.4 節では、HPSG には存在していない TDL の型変換規則を処理する方法について述べる。3.5節では、HPSG 文法と TDL 文法で扱い方の異なる、長距離依存現象を取り上げる。最後に、3.6節では、TDL では説明対象となっていない語形成の問題を扱う。

3.1. TDL 構造(TDLS)

例として、名詞"student"に対応する TDLS を図 6に示す。TDLS は以下の4つの要素よりなる。

- Syntactic Category : CCG に基づく範疇。基礎範疇(N(名詞)、D(限定詞)、V(動詞)、C(節)、S(文):[2]による)、及び、これらの合成によって得られる複合範疇からなる。他に、本研究では、特殊な範疇として[word modifier]、[identical modifier]を用いる。二つの TDLS の Syntactic Category の関係により、適用される合成規則が決定される。また、型変換が施されるかどうか、どのような型変換が施されるかを決定するためにも用いられる。
- Logical Form : TDL 意味表現を表す。型付きラムダ式によって表現される。
- Nonlocal Abstraction : Logical Form 内で抽象化されている変数のうち、HPSG では SLASH 素性に格納される項に対応する変数が格納される。このスロットにより、HPSG と TDL 文法における長距離依存現象の扱いの差を吸収する(3.5節参照)。
- Predicate Name : 基本的には、TDLS に対応する語句の主辞の名前に当たる。TDL 意味表現中に、この Predicate Name と同じ名前を持つ命題が存在し、その命題名と構造共有している。語形成を処理するために用いられる(3.6節参照)。

$$\begin{bmatrix}head\ \text{H}\\ \text{SUBJ}\ <S>\\ \text{COMPS}\ <C_1\cdots C_n>\end{bmatrix}\Rightarrow H\backslash S/C_1/\cdots/C_n$$

$$\begin{bmatrix}head\ \text{H}\\ \text{MOD}\ M\\ \text{SUBJ}\ <S>\\ \text{COMPS}\ <C_1\cdots C_n>\end{bmatrix}\Rightarrow\begin{cases}M(/or\backslash)M\backslash S/C_1/\cdots/C_n\\ or\\ H\backslash S/C_1/\cdots/C_n\ \text{and}\\ H\Rightarrow M(/or\backslash)M\end{cases}$$

図 7 統語的範疇の対応

3.2. HPSG 語彙項目への TDLS の割り当て

[2]で示された TDL 文法の語彙項目と、HPSG の語彙項目との対応関係に基づいて、HPSG 語彙項目への TDLS 割り当て規則を記述していく。ここで、HPSG と TDL 文法において、統語的にまったく考え方の異なる語彙項目は少ない。ほとんどの場合は、HPSG の統語範疇と TDL 文法の範疇の間に以下のような対応関係が認められる(図 7)。H、S、C、M を統語的範疇とすると、

- HPSG 語彙項目が、head:H、SUBJ:<S>、COMPS:<C_1, ..., C_n> のとき、TDLS の Syntactic Category は、H\S/C_1/.../C_n となる。
- HPSG 語彙項目が、head:H、MOD:M、SUBJ:<S>、COMPS:<C_1, ..., C_n> のとき、TDLS の Syntactic Category は、M(\or/)M\S/C_1/.../C_n となるか、もしくは、H\S/C_1/.../C_n となり、かつ、H ⇒ M (\or/)M への型変換規則が存在する。

オープンクラスの語彙項目では、HPSG 語彙項目と TDLS の対応関係は半自動的に得られる。一方、クローズドクラスの語彙項目では、同じ統語構造を持っていたとしても、異なる TDL 意味表現を持つことがありうる。たとえば、不定冠詞"a"と、定冠詞"the"は、統語構造は等しいものの、意味的には完全に区別される(図 8)。よって、これらの語は、人手により逐一対応を記述する必要がある。

$$\left[\begin{array}{c} D/N \\ \lambda n.\lambda w.\lambda e.\lambda s.\lambda\phi.nx_i s((\Delta x_i)[wx_i es\phi]) \\ [\,] \\ "a" \end{array}\right]$$

$$\left[\begin{array}{c} D/N \\ \lambda n.\lambda w.\lambda e.\lambda s.\lambda\phi.ref(x_i)[nx_i s_i(id)][wx_i es\phi] \\ [\,] \\ "the" \end{array}\right]$$

図 8 統語的範疇が等しく、意味表現が異なる例

$$\left[\begin{array}{c} D \\ \lambda w.\lambda e.\lambda s.\lambda\phi.ref(x_i)[year'x_i s_i][wx_i es\phi] \\ [\,] \\ "year" \end{array}\right]$$

$$\Downarrow$$

$$\left[\begin{array}{c} V\backslash V \\ \lambda v.\lambda e.\lambda s.\lambda\phi.ref(x_i)[year'x_i s_i]\left[ves\left[\begin{array}{c} mod'ex_i \\ \phi \end{array}\right]\right] \\ [\,] \\ "year" \end{array}\right]$$

図 10 型変換の例

3.3. 合成の方法

基本的には、Syntactic Category と Logical Form を参照し、図 9のような合成をする。これは、CCG における functional application にあたる。ここで、*head(L, R)* は、L, R のうち、HPSG 構文木において主辞である構成素の Predicate Name を返す関数である。また、*xor(U, V)*は、U,V のうちどちらかが値を持つとき、その値を返し、両者とも値が入っていなければ空を返す関数である。両者ともに値を持つ場合は帰り値を定義しない。これは、HPSG における SLASH 素性の扱いに準じたものである。

合成される TDLS のうち、どちらかの Syntactic Category が[word modifier]である場合は、語形成にかかわる合成である。TDL では語形成は扱わないため、語形成を処理するための特殊な規則を適用する(3.6節で述べる)。また、どちらかの Syntactic Category が[identical modifier]である場合は、もう片方の TDLS を合成後の TDLS とする。すなわち、Syntactic Category が[identical modifier]である語句は、統語的にも意味的にも無視される。読点や記号の一部がこれに該当する。

3.4. 型変換規則

TDL 文法では、その導出において、語句が空範疇(音声を持たないが、統語的・意味的な作用を持つ範疇)と合成さ

$$\frac{\left[\begin{array}{c} X/Y \\ f \\ U \\ L \end{array}\right]\left[\begin{array}{c} Y \\ a \\ V \\ R \end{array}\right]}{\left[\begin{array}{c} X \\ fa \\ xor(U,V) \\ head(L,R) \end{array}\right]}> \qquad \frac{\left[\begin{array}{c} Y \\ a \\ U \\ L \end{array}\right]\left[\begin{array}{c} X\backslash Y \\ f \\ V \\ R \end{array}\right]}{\left[\begin{array}{c} X \\ fa \\ xor(U,V) \\ head(L,R) \end{array}\right]}<$$

図 9 Functional application

れることにより新たな句を構成することがある。このような合成を型変換とよぶことにする。型変換の例が図 10に示されている。TDL 文法では、動詞を修飾する句"this year"は、始めは名詞であり、それが空範疇と合成されることにより動詞を修飾する句になる、と考えられる。一方、HPSG では空範疇ではなく MOD 素性を用いる。"this year"の"year"は MOD 素性の値が動詞であるため、主辞素性が名詞であるにも関わらず、動詞を修飾することができるのである。このように、多くの型変換規則は HPSG 文法では存在しないため、TDLS を合成する際に、HPSG 構文木とは独立に TDL の型変換規則が適用できるかどうか試みる。型変換規則が適用されるかどうかは、Syntactic Category によって決定される。

3.5. 長距離依存現象

目的格の項が wh 移動している場合を考える。HPSG では、移動する語句を SLASH 素性の値とし、filler-gap 規則を導入することで、これを処理している。すなわち、長距離依存現象のための語彙規則及び合成規則を導入することで処理する。一方、TDL 文法では CCG と同様、語彙規則は用いずに、通常の組合せ規則で処理する。このとき、関数合成規則(functional composition)が用いられることが多い。HPSG 構文木から TDL 意味表現を導出するにあたり、長距離依存現象を語彙規則及び合成規則の追加とみなすか、組合せ規則の一種であるとみなすか、という文法間の差異を吸収する必要がある。この問題を解決するにあたり、HPSG 的な方式を採る。TDLS に Nonlocal Abstraction というスロットを定義し、HPSG 項目の Nonlocal 素性の値が示す語句の TDL 意味表現を格納するようにした。

図 11は、動詞"love"の例であり、目的語の情報を Nonlocal Abstraction に移すように語彙項目を変換する規則を示している。Nonlocal Abstraction に収められた変数は、Logical Form 中では自由変数となり、関係節化演算子(which, that 等)と合成される時に、この変数に値が代入される(図 12)。これは、元の TDL 文法において、wh 移動した語句の TDL 意味表現が代入されるべき変数は、関係節化演算子と合成されるまで値が代入されないことと等しい。

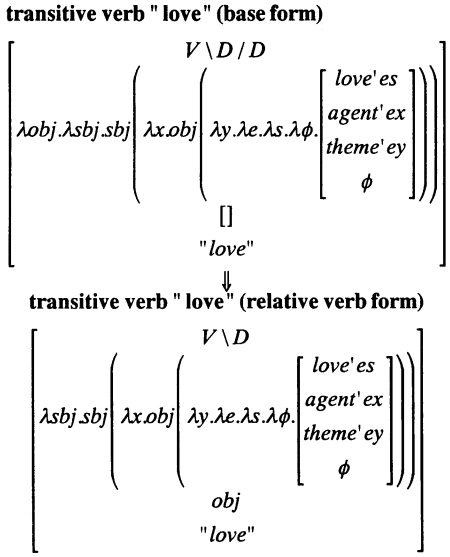

図 11　長距離依存現象のための語彙項目変換

3.6. 語形成

例えば“orange juice”という語句の意味を述語論理で表すとき、orange(x) ∧ juice(x) とすべきではない。“orange juice”は“orange”ではないからである。このような、複合名詞や動詞+不変化詞の生成を語形成という。TDL では、語形成は扱わずに、“orange juice”で一語であると考える。ところが、多くのパーザの出力では、“orange”、“juice”が別々の語として出現する。そのため、TDL の合成規則とは別に、語形成のための合成規則(図 13)を用意する必要がある。複合語を構成する語のうち、主辞でない語を語修飾詞と定め、Syntactic Category を[word modifier]とする。合成される TDLS のどちらかの Syntactic Category が[word modifier]であるとき、語形成規則が適用される。語形成規則は、修飾詞の Predicate Name を被修飾詞の Predicate Name に結合し、それ以外の要素は被修飾詞のものを保存する合成規則である。現在は、「名詞を修飾する名詞」、「不変化詞」を[word modifier]と定めている。

4. 実装状況

現在、前節に示した手法の実装を進めており、今までに HPSG 語彙項目から TDLS の変換規則 27 個、TDLS の型変換規則 4 個が実装されている。本節では、実際にコーパスの文から TDL 意味表現を取り出し、その被覆率を計測することによって、実装の進捗状況と今後の見通しを示す。対象となるコーパスには、Penn Treebank 第22節を採用した。まず、HPSG パーザを用いて文を構文解析して HPSG 構文木を取得し、次にその HPSG 構文木から TDL 意味表現への変換を試み、被覆率を計測した。結果を表 1に示す。文

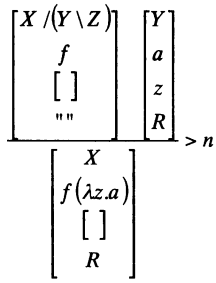

図 12　Filler-gap 規則

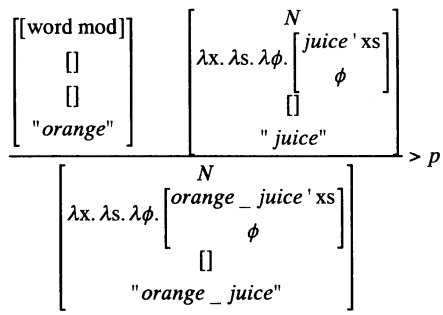

図 13　語形成規則

全文数	1,515
被覆文数	332
文被覆率	21.9%
全単語被覆文数	360
全単語被覆文に対する被覆率	92.2%
全単語数	31,297
被覆単語数	28,732
単語被覆率	91.8%

表 1　Penn Treebank 第 2 2 節に対する被覆率

被覆率は 21.9%と低い水準にとどまっている一方、全ての単語に TDLS を割り当てることに成功した文に対する被覆率は、92.2%を達成している。このことから、文被覆率の低さの原因は、主に単語被覆率に起因するものであることがわかる。この実験により、単語被覆率を上昇させることが、文被覆率を上げることに直結することが判明した。

次に、TDLS の割り当てが成功しなかった単語に関して、その原因を調べた。原因には以下の三つが考えられる。

- TDL で研究継続中の言語現象:TDL は現在も研究・改良が進められている文法理論であり、分析が定まっていない言語現象が存在する。等位接続等がその例である[14]。これらの言語現象を発生させる単語は、現時点では実装することはできない。

TDL で研究中の語彙項目	69
実装されていない語彙項目	21
非言語的な語彙項目	10

表 2　TDLS の割り当ての失敗の原因

- 実装されていない語彙項目：TDL 及び TDL 文法で分析されている単語であるが、HPSG 語彙項目から TDLS への変換規則が対応していないもの。
- 頑健なパーザによる非言語的な語彙項目の割り当て：実テキストに情報を付加したコーパスから取得した文法は、実テキスト中の非言語的な用例、情報の付加の誤り、文法獲得の失敗という理由により、言語学的に想定されない語彙項目を持つ。今回の実装はこのような語彙項目を想定していない。

これらの原因を判定するために、TDLS を割り当てることに失敗した単語を100語、無作為に抽出し、目視によって原因を判定した。表 2に結果が示されている。割り当ての失敗の原因の大部分は TDL で研究中の語彙項目であった。しかし、TDL は現在精力的に研究が進められており、近い将来にこれらの語彙項目を扱うことができるようになることが期待される。一方、TDL で解析されてはいるが、実装がなされていない語彙項目も存在する。これらを処理するためには、更にエラー解析を行い、適切な変換規則を記述・実装していくことが必要であるが、理論的に扱うことが不可能であるわけではない。今後の実装の進展により解決できると考えられる。最後に、少数ながら、頑健なパーザに特有の、非言語的な語彙項目が見られた。これらの語彙項目に TDL を割り当てるには、非言語的な語彙項目に対して、「正しい」語彙項目を推測し、その TDLS を割り当てる操作が必要になる。この操作を実現することが今後の課題である。

5. まとめと今後

実テキストから高い被覆率で統語解析結果を得ること、そして、統語解析結果から高い表現力を持つ意味表現を得ることは、従来両立が難しいタスクであった。本研究では、出力の言語学的情報量に富む、頑健な HPSG パーザと、高い意味論的表現能力を持ち、統語理論も示されている TDL 意味表現を組み合わせることによって、被覆率と表現力を両立させることを試みた。

そこで問題となったのが、異なる文法理論である HPSG と TDL との差違の吸収である。本研究では、HPSG 語彙項目から TDL 意味表現への変換規則と、HPSG 構文木に沿った TDL 意味表現の合成規則を定義する手法を提示し、その中で

- HPSG には存在しない、TDL の型変換規則への対応
- HPSG と TDL で分析の方向性が異なる長距離依存現象の処理
- 実テキストにおける語形成を構成する為の手法

という問題に取り組んだ。

更に、提案した手法の実装を進めており、実際に実テキストから TDL 意味表現を得て、その被覆率を計測することにより、実装の進捗を示した。結果は、文被覆率は低いものの、全ての単語に TDL 意味表現が割り当てられた文においては、高い被覆率を示した。このことから、単語被覆率が文被覆率に大きく寄与することが示唆された。

次に、TDL 意味表現が割り当てられなかった単語について、その原因を調べた。多くの単語については、TDL 文法理論においてなお研究が継続中であり、分析が定まっていないことが原因であったが、これらは、TDL の研究の進展により、意味表現を割り当てることが可能になると考えられる。一方、少数ながら、頑健なパーザにより、非言語的な語彙項目が割り当てられたために、TDL 意味表現の割り当てに失敗した単語が確認された。これは、頑健なパーザを用いることに起因する問題であり、このような単語に適当な TDL 意味表現を割り当てる手法を確立することが、今後の課題である。

文　献

[1] D. Bekki, "Typed Dynamic Logic for Compositional Grammar", Doctoral Dissertation, Faculty of Science, Department of Information Science, The University of Tokyo, 2000.

[2] D. Bekki, "Typed Dynamic Logic and Grammar: the Introduction", manuscript, The University of Tokyo, 2005.

[3] J. Bos, S. Clark, M. Steedman, J. R. Curran and J. Hockenmaier, "Wide-Coverage Semantic Representations from a CCG Parser", Proceedings of the 20th International Conference on Computational Linguistics (COLING '04), Geneva, Switzerland, 2004.

[4] A. Copestake, D. Flickinger, I. A. Sag and C. Pollard, "Minimal Recursion Semantics: An introduction", manuscript, 1999.

[5] G. Evans, "Pronouns, Quantifiers and Relative Clauses", in Canadian Journal of Philosophy 7, pp.467-536, 1977.

[6] G. Evans, "Pronouns", in Linguistic Inquiry 11, pp.337-362, 1980.

[7] D. Flickinger, A. Copestake and I. A. Sag "HPSG analysis of English" in W. Wahlster and R. Karger (ed.), Verbmobil: Foundations of Speech-to-Speech Translation, pp.254-263, Springer-Verlag, Berlin, Heidelberg and New York, 2000.

[8] J. Hockenmaier and M. Steedman, "Acquiring Compact Lexicalized Grammars from a Cleaner Treebank", Proceedings of Third International Conference on Language Resources and Evaluation, Las Palmas, 2002.

[9] Y. Miyao, T. Ninomiya and J. Tsujii, "Corpus-oriented Grammar Development for Acquiring a Head-driven Phrase Structure Grammar from the Penn Treebank", in KY. Su, J. Tsujii, JH. Lee and OY. Kwong (ed.), Natural Language Processing - IJCNLP 2004, LNAI3248, pp.684-693, Springer-Verlag, 2005.

[10] C. Pollard and I. A. Sag, "Head-Driven Phrase Structure Grammar", Studies in Contemporary Linguistics, University of Chicago Press, Chicago, London, 1994.

[11] M. Steedman, "Surface Structure and Interpretation", The MIT Press, 1996.

[12] M. Steedman, "The Syntactic Process (Language, Speech, and Communication)", The MIT Press, 2001.

[13] F. Xia, "Extracting Tree Adjoining Grammars from Bracketed Corpora", the 5th Natural Language Processing Pacific Rim Symposium (NLPRS-99), Beijing, China, Nov 1999.

[14] 戸次大介、川添愛「等位接続構文の並行的解釈」, 日本言語学会 第 130 回大会, pp.236-241, 2005.