

アブダクションに基づく会話体文の理解のための 一般化されたチャート法

伝 康 晴†

近年、自然言語処理において、話し言葉の解析/理解の研究が盛んになっている。話し言葉は、省略、提喻などのさまざまな不完全さを含んでおり、こういった不完全さを持つ入力に対して、その背後にある情報を補完しながら処理を行う能力が話し言葉を扱う自然言語処理システムには必要になる。このような処理を可能にするモデルとしてアブダクションに基づくモデルが有力である。しかし、その計算効率の悪さのため、アブダクションに基づく実用的なシステムはまだ実現されていない。そこで、本稿では、コスト付アブダクションのモデルを話し言葉の解析/理解に応用することを念頭におき、その計算効率の問題に対する根本的で実際的な解決策を提案する。その基本的なアイデアは、チャートパーサをアブダクションの手続きとみなすことである。チャートパーサには計算効率の観点からさまざまな利点があるので、このアイデアに基づきアブダクションの計算効率の問題をうまく解決できる。本手法は、(1)目標駆動上昇型導出、(2)部分解の表への登録、(3)アジェンダ制御機構という3つの特徴を持つ。これによってコスト付アブダクションの計算量が大きく改善されることを実験によって示す。

Generalized Chart Algorithm for Abduction-based Spoken-style Sentence Understanding

YASUHARU DEN†

Spoken language understanding is one of the challenging research areas in natural language processing. Since spoken language is in various ways incomplete, containing ellipsis, metonymy, etc., spoken language understanding systems should have the ability to process incomplete inputs by hypothesizing underlying information. The cost-based abduction model has provided a simple and elegant way of achieving such a task. However, due to the high computational cost, abduction-based systems are still far from practical use. In this paper, aiming at the application to spoken language understanding systems, we propose an efficient algorithm for cost-based abduction, which is inspired by the idea of using chart parsers as proof procedures. Because chart parsers have many advantages from the viewpoint of computational efficiency, chart-based abduction algorithm are expected to nicely solve the computation problem. We discuss in detail three features of our algorithm—*goal-driven bottom-up derivation*, *tabulation of partial results*, and *agenda control mechanism*—, and report the results of the preliminary experiments, which clearly show these features considerably improve the computational efficiency of cost-based abduction.

1. はじめに

近年、自然言語処理において、話し言葉の解析/理解の研究が盛んになっている。話し言葉は、省略^{*}、提喻^{**}などのさまざまな不完全さを含んでおり、こ

いった不完全さを持つ入力に対して、その背後にある情報を補完しながら処理を行う能力が話し言葉を扱う自然言語処理システムには必要になる。このような処理を可能にするモデルとしてアブダクション (abduction) に基づくモデル⁵⁾が有力である。

話し言葉の解析の例として、次の文を考えよう。

(1) 漱石買った。

ここには、話し言葉に特有の2つの問題がある。提喻と助詞の省略である (ここでは格要素の省略については考えない)。いま、話し手が自分の経験について話している状況でこの文を発話したとすると、その意味は「話し手は漱石の小説を買った」と考えるのが自然

† ATR 音声翻訳通信研究所

ATR Interpreting Telecommunications Research Laboratories

* 「本を買った」のように必要な格要素 (ここでは格) が欠落する現象。あるいは、「太郎がその本買った」のように助詞 (ここでは『その本』の後の『を』) が欠落する現象。

** 「漱石を読んだ」のように、作家の名前によってその作品を指示するなど、名詞によって字義どおりのものとは異なる対象が指示される現象。

である。この解釈を得るためには以下のことを行う必要がある。

提喩の解釈 名詞『漱石』による提喩は『漱石の小説』に解釈できる。この解釈は、「作家の名前はしばしばその作家の書いた小説を指すのに用いられる」という語用論的知識による。

助詞の省略された係り受けの解釈 名詞『漱石』（提喩により『漱石の小説』を指示する）と動詞『買った』の間の係り受けは対象格 (object) に解釈できる。この解釈は、「売買に関する行為が品物を対象格として持つとき、その関係を係り受けとして言語化できる」という意味的知識による。

これらの解釈はアブダクションによって行われる。アブダクションとは、規則 $B_1 \wedge \dots \wedge B_n \supset A$ と観測 A から前件 B_1, \dots, B_n を導く形式の推論である。例えば、上記の意味的知識が(2)のような規則によって表されているとする。

(2) $trade(e) \wedge commodity(x) \wedge obj(e, x) \supset sem(e, x)$
 ここで、名詞概念 x と動詞概念 e に対して、 $sem(e, x)$ と $obj(e, x)$ はそれぞれ「 e と x に係り受けがある」「 x が e の対象格である」を表し、 $trade(e)$ と $commodity(x)$ はそれぞれ「 e が売買に関する行為である」「 x が品物である」を表す。このとき、「買う行為と漱石の小説の間に係り受けがある」という観測 $sem(B, N)$ から、「買う行為は売買に関する行為である」「 $trade(B)$ 」「漱石の小説は品物である」「 $commodity(N)$ 」「漱石の小説は買う対象である」「 $obj(B, N)$ 」などが推論される。ただし、動詞概念 B と名詞概念 N はそれぞれ『買った』『漱石の小説』の指示対象である。このうち、入力に対する観測から直接確認できない $obj(B, N)$ は背後にある情報として仮定される。これは『漱石の小説』と『買った』の係り受けが対象格として解釈されたことを意味する。

もちろん、係り受けを導く規則はほかにもあり得る。例えば、『漱石』を字義どおりにとれば(1)の係り受けを行為者格 (agent) に解釈することができ、この場合(1)の意味は「漱石が何かを買った」ということになる。この解釈は、いま考えている状況（話し手が自分の経験について話している状況）では可能性が低い、他の状況では最初の解釈よりも可能性が高いかも知れない。このような可能性の高低をとらえるために、アブダクションに基づくモデルは、解釈の良否を評価する機構を備えている。これはコスト付アブダクション (cost-based abduction)^{31,32} と呼ばれる。

コスト付アブダクションでは、おのおのの仮説はある量のコストを持っている。例えば、規則(2)から得られる仮説 $obj(e, x)$ は、\$2 のコストを持つといったようなことが規則中に指定されている。解釈の良否は、その解釈を得るために必要なすべての仮説が持つコストを総計することによって評価する。すなわち、入力文の解釈は以下のようにして行う。

1. すべての可能な解釈を見つけ、
2. 最小のコストを持つものを最適解として選ぶ。

上記の例では、『漱石』を字義どおりにとり係り受けを行為者格とみなす解釈よりも、『漱石』を『漱石の小説』を指示する提喩にとり係り受けを対象とみなす解釈のほうがコストが小さい（そのように規則が書かれる）ので、後者が選ばれる。

ここで明らかに問題になるのは、アブダクションの計算効率である。一般にアブダクションは非常に多くの可能性を許してしまうので、上の解釈スキーマは非常に多くの計算を伴うことになる。特に、話し言葉の解析/理解に応用する場合には、入力文の背後にある情報を補完しながら、非常に多くの可能性を考慮する必要がある。これはアブダクションの計算を重くし、アブダクションに基づく実用的なシステムの実現を困難にする。現存するモデルはいずれもこの問題に対する根本的な解決策を提供していない。例えば、Charniak ら^{31,32}はこの問題を扱ってはいるが、彼らの解決策は、探索空間が基底式 (ground formulas) を節点とするグラフによって表現できる場合に限られており、変数を含む述語論理式によって規則が記述されている場合の扱いは明確でない。特に、再帰規則を含む場合には彼らの方法は適用できない。

そこで、本稿では、コスト付アブダクションのモデルを話し言葉の解析/理解に応用することを念頭におき、その計算効率の問題に対する根本的で実的な解決策を提案する。その基本的なアイデアは、自然言語の構文解析の研究から来ている。Pereira と Warren⁷⁾ が指摘したように、構文解析と推論の間には密接な関係がある。構文解析法の一つであるチャートパーサを演繹手続きとみなす彼らのアイデアの自然な拡張として、チャートパーサをアブダクションの手続きとみなすことができる。チャートパーサには計算効率の観点からさまざまな利点があるので、このアイデアに基づきアブダクションの計算効率の問題をうまく解決できる。本稿で提案する手法は以下の特徴を持ち、これによってコスト付アブダクションの計算効率が大きく改

善される。

目標駆動上昇型導出 探索空間を削減する。

部分解の表への登録 同一目標の再計算を避ける。

アジェンダ制御機構 最適解を効率的に発見するためのさまざまな探索方略を実現する。

本稿の構成は以下のとおりである。まず、2章で本手法の基本的なアイデアを説明する。次に、3章で、本手法の詳細を説明し例をあげる。さらに、4章では、簡単な実験の結果を報告し、上であげた本手法の特徴によって計算効率がいかに改善されるかを示す。最後に、5章でまとめを述べる。なお、本稿では、コスト付アブダクションの計算効率の問題に焦点をしばり、話し言葉の解析/理解への応用については簡単な例をあげるにとどめる。

2. 主辞駆動型導出

Pereira と Warren⁷⁾は、自然言語の構文解析と述語論理の推論の間に密接な関係があることを指摘し、構文解析を演繹推論として定式化した。このような関係に基づいて、構文解析におけるさまざまな知見を推論の問題に援用することができる。Pereira と Warren が論じたのは下降型導出 (top-down derivation) の場合であったが、ここでは、より効率的な導出法として**目標駆動上昇型導出** (goal-driven bottom-up derivation)*を考へる。

ホーン節の演繹を単純な下降型/上昇型導出で行う際に生じる以下の問題について考えてみよう。

下降型導出 規則選択の誤りに起因する頻繁な後戻り

上昇型導出 初期目標の証明に貢献しない無駄な導出

これらは、自然言語の構文解析を単純な下降型/上昇型の導出で行ったときに生じる問題と本質的に同じである。構文解析の場合には、こういった問題は、**左隅構文解析** (left-corner parsing) のようなより洗練された導出法を用いることによって解決されている。ここであげるアイデアは、これに類似した導出法をホーン節の演繹に用いるということである。

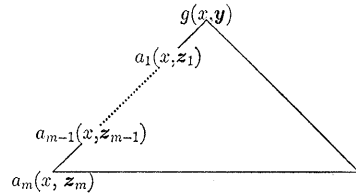


図1 主辞駆動型導出の概念
Fig. 1 Concept of head-driven derivation.

いま、目標 $g(x, y)$ の証明過程が図1に示す形になるとする。ただし、 g は述語、 x はこの要素式の先頭引数、 y は第2引数以下の引数列である (以下、要素式を同様の表現で表す)。図1では、目標 $g(x, y)$ の先頭引数 x が、左隅 $a_m(x, z_m)$ から目標 $g(x, y)$ に至る経路上で共有されている。このような場合に、以下に示すような左隅構文解析に類似した導出を考へることができる。これを**主辞駆動型導出** (head-driven derivation) と呼ぶ。ただし、以下で A^σ は要素式 A に単一化子 σ を適用した結果を表す*。

ステップ1 目標 $g(x, y)$ の先頭引数 x と単一化するような先頭引数 w を持つ事実 $a(w, z)$ をみつけ、要素式 $a(x^\sigma, z^\sigma)$ を証明木の左隅におく。ただし、 σ は x と w の最汎単一化子。

ステップ2 左隅キー $a(x, z)$ と単一化するような最左前件 $a(w, z_m)$ を持つ規則 $a(w, z_m) \wedge B_1 \wedge \dots \wedge B_n \supset a_{m-1}(w, z_{m-1})$ をみつけ、新たな目標 $B_1^\sigma, \dots, B_n^\sigma$ を導入する。ただし、 σ は $a(x, z)$ と $a(w, z_m)$ の最汎単一化子。これらの目標がすべて再帰的に導出されたら、 $a(x^\sigma, z_m^\sigma), B_1^\sigma, \dots, B_n^\sigma$ の親として、後件 $a_{m-1}(x^\sigma, z_{m-1}^\sigma)$ を作り、 $a(x, z)$ の代わりに左隅におく。

ステップ3 もし後件 $a_{m-1}(x, z_{m-1})$ と目標 $g(x, y)$ が単一化すれば終了。そうでなければ、 $a_{m-1}(x, z_{m-1})$ を新たに左隅キーとしてステップ2に戻る。

主辞駆動型導出によって証明できる目標の例を図2に、この目標を主辞駆動型導出によって証明する過程の1つ (解は2つある) を図3に示す。ただし、大文

* 初期目標 G の証明過程は、 G を根節点、規則中の事実 F_1, \dots, F_m を葉節点とし、各枝分かれの親節点 A と娘節点 B_1, \dots, B_n が規則 $B_1 \wedge \dots \wedge B_n \supset A$ に対応するような木によって表現できる。このとき、証明木を根節点から下向きに作っていく導出法を下降型導出、葉節点から上向きに作っていく導出法を上昇型導出という。また、上昇型導出において、根節点の持つ情報の一部を予測情報として用いることによって、将来根節点につながらないような部分木の作成を禁止するようにしたものを目標駆動上昇型導出という。

* 2つの要素式 (あるいは項) を同一にするような最も一般的な変数束縛の組を最汎単一化子という。例えば、 $p(S, y)$ と $p(x, x)$ の最汎単一化子 σ は $\{x/S, y/x\}$ である。ただし、 x, y は変数、 S は定数とする。また、要素式 A に単一化子 σ を適用した結果とは、 A 中の変数に σ で規定される変数束縛を施したもののことであり、上記の例で $p(S, y)$ に σ を適用した結果は、 $p(S, S)$ となる。

Goal $m(\langle[A],[B]\rangle, z)$

Rules

$h(p, a, q) \wedge m(q, z) \supset m(p, [a:z])$

$m(\langle[],\rangle, [])$

$h(\langle[a:x],y\rangle, a, \langle x,y\rangle)$

$h(\langle x,[a:y]\rangle, a, \langle x,y\rangle)$

図 2 主辞駆動型導出で証明できる目標の例
Fig. 2 Example of goals which can be proved by head-driven derivation.

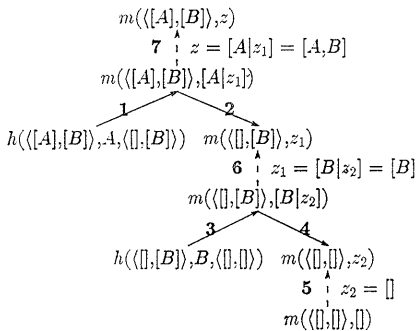


図 3 主辞駆動型導出による証明過程の例
Fig. 3 Example of proof process by head-driven derivation.

字で始まる引数は定数，小文字で始まる引数は変数を表す。また， $[a|x]$ は a を頭部， x を尾部に持つリストを表し， $[], []$ は空リストを表す。証明木中で，実線は左隅要素式をキーとしてステップ 2 で規則を適用したことを表し，破線はステップ 3 で部分目標に部分証明木を接合することによって未完成部分を補完したことを表す。また，数字は導出の順序を表す。

DCG 規則を用いた左隅構文解析は主辞駆動型導出の特殊例とみなすことができる。また，意味主辞駆動構文生成 (semantic-head-driven generation)^{8),11)} や主辞駆動構文解析 (head-corner parsing)^{9),12)} も主辞駆動型導出の一種である (文法規則を表す各節において意味主辞/統語主辞を右辺の先頭に移動し，文法範疇を表す各要素式において意味素性/主辞素性を引数列の先頭に移動して考えるとよい)。このように，主辞駆動型導出は自然言語の構文処理におけるさまざまな導出戦略を包含するものであり，この分野における知見をホーン節の演繹に援用する際の基礎となる。

さて，主辞駆動型導出が適用できるためには，すべての規則は連鎖形 (chain form) でなければならない。つまり，すべての規則は， $a_n(w, z_n) \wedge B_1 \wedge \dots \wedge B_n \supset a_{n-1}(w, z_{n-1})$ のように，最左前件の先頭引数と後件

の先頭引数が等しくなるように記述されていなければならない。これは制限としてきつすぎる。そこで，連鎖形でない規則 (非連鎖規則) を許すように，主辞駆動型導出を拡張する。すなわち，主辞駆動型導出のステップ 1 を以下のもの置き換える。この結果，主辞駆動型導出は意味主辞駆動構文生成と類似な下降型/上昇型混合の導出法になる。

ステップ 1 目標 $g(x, y)$ の先頭引数 x と単一化するような先頭引数 w を持つ後件 $a(w, z)$ からなる非連鎖規則 $B_1 \wedge \dots \wedge B_n \supset a(w, z)$ をみつけ，新たな目標 B_1^*, \dots, B_n^* を導入する (事実は前件が空の非連鎖規則とみなす)。ただし， σ は x と w の最汎単一化子。これらの目標がすべて再帰的に導出されたら， B_1^*, \dots, B_n^* の親として，後件 $a(x^\sigma, z^\sigma)$ を作り，証明木の左隅におく。

3. 一般化されたチャート法

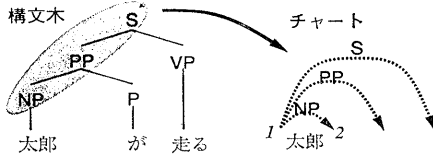
2章で述べたアイデアは，本稿で提案する手法の第 1 の特徴である目標駆動上昇型導出を実現する。本章では，このアイデアに基づき，残りの 2 つの特徴 (部分解の表への登録とアジェンダ制御機構) も実現する，より洗練されたアルゴリズムを提案する。

3.1 チャートパーサとその一般化

左隅構文解析が同一の部分構造を再計算してしまうという欠点を持つと同様に，主辞駆動型導出も深さ優先の後戻り型で実行すると同じ問題に直面する。左隅構文解析では，部分構造を表に登録することによってこの問題を解決しており，この手法はチャートパーサ (chart parser)⁶⁾ として知られている。最近，春野ら⁴⁾ により同様の手法が意味主辞駆動構文生成にも適用できることが示された。主辞駆動型導出は，意味主辞駆動構文生成を一般化したものなので，同様の手法は主辞駆動型導出にも適用できる。

チャートパーサでは，構文木を求める過程において得られる部分構造を弧と呼ばれるデータによって表し，これを随意チャートと呼ばれる表に登録していく。弧には活性弧と不活性弧の 2 種類があり，活性弧は予測列を含む未完成の部分構造に対応し，不活性弧は完成した部分構造に対応する。おのおのの弧は始点と終点を持ち，各頂点は入力文中の単語に対応している。例えば，図 4 (a) において，構文木中の楕円で囲った部分に含まれる非終端記号 S, PP, NP を根節点とする部分構造は，チャート中でそれぞれ，対応する

(a) チャートパーサ



(b) 一般化されたチャート法

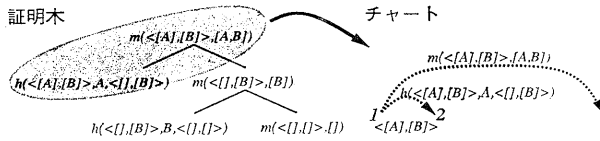


図4 弧の索引付け
Fig. 4 Indexing of edges.

ラベルを持つ不活性弧によって表される。ここで、これら3つの弧はいずれも頂点1を始点としている。頂点1は入力文中の単語「太郎」(1番めの単語)に対応しており、したがって、これらの弧は「太郎」によって索引付け (indexing) されているとみなせる。

同様の索引付けは一般化されたチャート法でも用いることができる。左隅構文解析と主辞駆動型導出の類似性から、図4(b)の楕円で囲った部分に含まれる要素式 $m\langle[A],[B]\rangle,[A,B]$, $h\langle[A],[B]\rangle,A,\langle[],[B]\rangle$ を根節点とする部分構造は、チャート中でそれぞれ、対応するラベルを持つ不活性弧によって表され、それらの始点は等しい。すなわち、これらの弧は同じ索引を持つ。ここで、主辞駆動型導出の性質より、これらの要素式の先頭引数 $\langle[A],[B]\rangle$ が等しいから、項 $\langle[A],[B]\rangle$ をこれらの弧に対する索引とみなせる。

さて、春野ら⁴⁾が示したように、チャートパーサと一般化されたチャート法で用いるチャートには、次の2つの大きな違いがある。

1. チャートパーサでは索引(入力文中の単語)は入力と与えられるとすぐに決まるが、一般化されたチャート法では索引(要素式の先頭引数)は漸時的に導入される。
2. チャートパーサでは索引(入力文中の単語)は線形の列として1対1に接続されるが、一般化されたチャート法では索引(要素式の先頭引数)の接続は多対多である。

一般化されたチャート法では、上記のように、要素式の先頭引数とその要素式に対応する弧に対する索引とみなすが、このような索引は要素式が部分目標として導入されるたびに動的に作られる。また、規則が非決定的な場合、複数の異なる部分目標が作られること

がある。例えば、前述の図2の目標には、図5の楕円で囲った部分に示す2つの異なる証明がある(参照の簡便のため要素式に番号をふってある)。ここで、上の証明木では、要素式1(先頭引数は $\langle[A],[B]\rangle$) が導出された後、次の部分目標として要素式10(先頭引数は $\langle[],[B]\rangle$) がたてられる。一方、下の証明木では、要素式2(先頭引数は $\langle[A],[B]\rangle$) が導出された後、次の部分目標として要素式11(先頭引数は $\langle[A],[B]\rangle$) がたてられる。したがって、この2つの証明過程で得られる部

分構造をチャートに登録する場合、索引 $\langle[A],[B]\rangle$ は2つの異なる索引 $\langle[],[B]\rangle$ と $\langle[A],[B]\rangle$ に隣接することになる。さらに、その先では、両者の証明木に同じ部分目標9(先頭引数は $\langle[],[B]\rangle$) が現れる。したがって、今度は、2つの異なる索引 $\langle[],[B]\rangle$ と $\langle[A],[B]\rangle$ が同じ索引 $\langle[],[B]\rangle$ に隣接する。このように、一般化されたチャート法では、索引の接続は多対多になる。図5の中央は、図2の目標に対する2つの

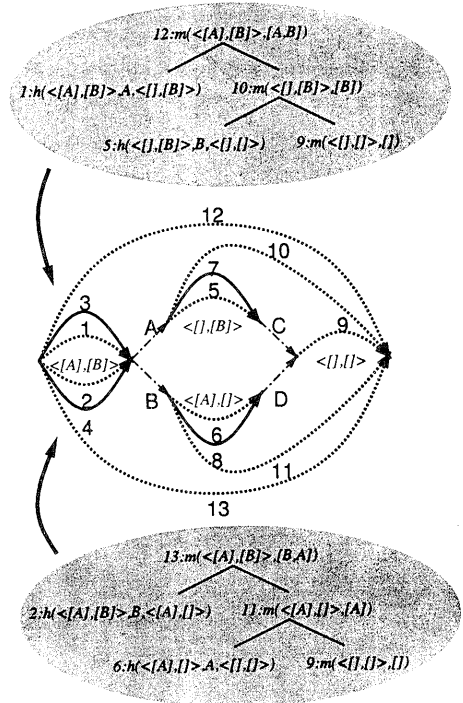


図5 弧の接続
Fig. 5 Connection among edges.

証明過程において得られる部分構造を登録したチャートであり、実線は活性弧、破線は不活性弧を表す（弧に付された数字は要素式の番号に対応、活性弧のラベルは省略してある）。また、多対多になる索引間の接続は、**A**、**B**などの鎖線のポイントによって示されている。

3.2 アルゴリズム

本節では、これまで演繹に関して説明してきたアイデアをアブダクションに適用できるように拡張し、本手法の形式的な定義を与える。

アブダクションへの拡張は簡単である。まず、目標 G に対して仮説 G を導入する手続きを新しく追加する。仮説は事実と同じように扱う。すなわち、仮説はチャート中では不活性弧として表す。次に、チャート中のおのおのの弧 e に対して**仮説集合** (assumption sets) S を定義する。仮説集合 S は弧 e に対応する部分証明木に含まれるすべての仮説を要素とする集合であり、形式的には以下のように定義される。

1. e が仮説 A を表す不活性弧のとき、 $S = \{A\}$ 。
2. e が非連鎖規則（事実を含む）によって導入された弧のとき、 S は空である。
3. e が連鎖規則によって不活性弧 e' から予測された弧のとき、 S は e' の仮説集合 S' に等しい。
4. e が活性弧 e_1 と不活性弧 e_2 の結合によって作られた弧のとき、 $S = S_1 \cup S_2$ である。ただし、 S_1 と S_2 はそれぞれ e_1 と e_2 の仮説集合である。

これらの変更を考慮して、一般化されたチャート法の定義は以下ようになる。ただし、 f はチャート中の各索引に対して一意な頂点を与える関数であり、活性弧、不活性弧のラベルをそれぞれ $[B_1 \dots B_{j-1} [?] B_j \dots [?] B_n] A : S$ および $A : S$ で表す。ここで、 B_1, \dots, B_n 、 A は要素式であり、活性弧のラベル表現において $[?]$ が前置しているものは未完成の部分目標に対応する（上では、 B_1 から B_{j-1} が完成部分であり、 B_j から B_n が未完成部分である）。未完成の部分目標のうち最も左にあるもの（上では B_j ）を最左空所と呼ぶ。また、 S は仮説集合である。

初期化 活性弧 $[?]G]T : \phi$ を頂点 0 から 0 に張る。ただし、 G は初期目標。

以下の手続きを可能な限り繰り返し適用する。

導入手続き 頂点 s から t に活性弧 $[\dots [?] B_j \dots] A : S$ があるとする。ただし、 $B_j \equiv b_j(x_j, y_j)$ は最左空所。このとき以下を行う。

1. 頂点 $f(x_j)$ がまだチャートに導入されていないな

ら、頂点 t から $f(x_j)$ にポイントをつけ以下を行う。

- (a) 後件の先頭引数 w が x_j と単一化するようなすべての非連鎖規則 $B_1 \wedge \dots \wedge B_n \supset a(w, z)$ に対して、頂点 $f(x_j)$ から $f(x_j)+1$ に活性弧 $[[?] B_1^? \dots [?] B_n^?] a(x_j^?, z^\sigma) : \phi$ を作る。ただし、規則が事実のとき ($n=0$ のとき) は、不活性弧 $a(x_j^?, z^\sigma) : \phi$ を作る。ここで、 σ は x_j と w の最汎単一化子である。

- (b) 頂点 $f(x_j)$ から $f(x_j)+1$ に不活性弧 $B_j : \{B_j\}$ を作る。

2. 頂点 $f(x_j)$ がすでにチャートに導入されているなら、頂点 t から $f(x_j)$ にポイントをつける。さらに、不活性弧 $B_j : \{B_j\}$ がチャート中になければ、頂点 $f(x_j)$ から $f(x_j)+1$ に作る。

予測手続き 頂点 s から t に不活性弧 $C : S$ があるとする。最左前件 A が C と単一化するようなすべての連鎖規則 $A \wedge B_1 \wedge \dots \wedge B_n \supset A'$ に対して、頂点 s から t に活性弧 $[A^? [?] B_1^? \dots [?] B_n^?] A'^\sigma : S^\sigma$ を作る。ただし、前件が A だけのとき ($n=0$ のとき) は、不活性弧 $A'^\sigma : S^\sigma$ を作る。ここで、 σ は C と A の最汎単一化子である。

結合手続き 頂点 s から t に活性弧 $[\dots [?] B_j [?] B_{j+1} \dots [?] B_n] A : S_1$ があり、頂点 u から v に不活性弧 $C : S_2$ があり、さらに、頂点 t から u にポイントがあるとする。ただし、 B_j は最左空所。このとき、 B_j と C が単一化するなら、頂点 s から v に活性弧 $[\dots B_j^? [?] B_{j+1}^? \dots [?] B_n^?] A^\sigma : S_1^? \cup S_2$ を作る。ただし、 B_j が最後の空所のとき ($j=n$ のとき) は、不活性弧 $A^\sigma : S_1^? \cup S_2$ を作る。ここで、 σ は C と B_j の最汎単一化子である。

不活性弧 $T : S$ のおのおのは解を表す。

3.3 例題

ここでは、本手法を話し言葉の解析に応用する簡単な例をあげる。図6は(1)のような日本語の話し言葉を解析するための規則である。これには、統語、意味、語用論に関する規則、辞書、一般知識が含まれており、それらはすべてホーン節によって同一の形式で記述されている。

統語規則は、動詞と名詞句を助詞のあるなしに関わらず結合するよう書かれている。これによって、 $[vp [pp [np 漱石]] [vp 買った]]$ のような構造が許される（話し言葉ではこのような緩い統語規則が適当と思われる）。このような係り受けは意味的・語用論的な基準

Syntactic Rules

$vp(i,k,e) \supset s(i,k,e)$
 $pp(i,j,c,x) \wedge vp(j,k,e) \wedge dep((c,e,x)d) \supset vp(i,k,e)$
 $np(i,j,x) \wedge p(j,k,c) \supset pp(i,k,c,x)$
 $np(i,k,x) \supset pp(i,k,c,x)$
 $prag((x)p,y) \wedge sem((c,e,y)s) \supset dep((c,e,x)d)$

Lexical Rules

$soseki(x)^{S1} \supset np([Soseki]k,k;x)$
 $buy(e)^{S1} \supset vp([Katta]k,k,e)$
 $ga(c)^{S1} \supset p([Ga]k,k,c)$
 $wo(c)^{S1} \supset p([Wo]k,k,c)$

Pragmatic Rules

$prag((x)p,x)$
 $writer(x) \wedge write((x;y)r)^{S10} \wedge novel(y)^{S1} \supset prag((x)p,y)$

Semantic Rules

$ga(s,c) \wedge ga(c)^{S3} \supset sem(s)$
 $wo(s,c) \wedge wo(c)^{S3} \supset sem(s)$
 $intend(e) \wedge person(x) \wedge agt((e,x)r)^{S20} \supset ga((c,e,x)s,c)$
 $trade(e) \wedge commodity(x) \wedge obj((e,x)r)^{S2} \supset wo((c,e,x)s,c)$

Knowledge Rules

$soseki(x) \supset person(x)$
 $soseki(x) \supset writer(x)$
 $novel(x) \supset book(x)$
 $book(x) \supset commodity(x)$
 $buy(e) \supset trade(e)$
 $trade(e) \supset intend(e)$

図 6 日本語の話し言葉解析のための規則
 Fig. 6 Rules for spoken Japanese understanding.

で評価される。すなわち、動詞概念 e と名詞概念 x が格マーカ c をともなって係り受けを形成するのは、 x との間に何らかの語用論的關係を持つ対象 y が存在し、 e と y との間に (c でマークされるべき) 何らかの意味的關係があるときである。語用論に関する規則は、例えば、「作家の名前はしばしばその作家の書いた小説を指すのに用いられる」といった提喩の關係を定義している。特殊な場合として、 $x=y$ 、つまり、 x が字義どおりに用いられる場合がある。意味に関する規則は、例えば、「売買に関する行為が品物を対象格として持つとき、その關係を『を』でマークされるべき) 係り受けとして言語化できる」といった深層格關係を定義している。さらに、一般知識を記述する規則があり、意味規則から参照される意味属性の階層關係が定義されている。

要素式 $A^{\$p}$ の上付添字 $\$p$ は、 A を仮定する際に払うべきコストを表す (添字を持たない要素式は十分に大きなコストを持ち、そのような仮説は処理時には枝刈りされるものとする)。 p が小さいとき A は仮定しやすく、 p が大きいとき A は仮定しにくい。例えば、売買に関する行為と品物との間に対象格の關係を仮定する際のコストは $\$2$ と小さい。一方、意図的な行為と話者・聴者以外の第三者との間に行為者格の關係を

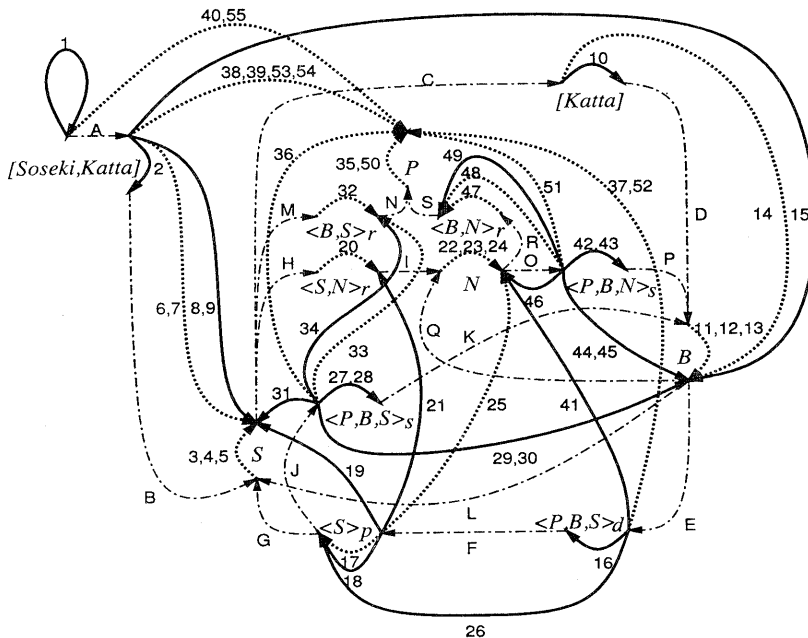


図 7 “漱石買った” の解析でできるチャート
 Fig. 7 Chart diagram for Soseki katta.

表 1 “漱石買った”の解析でできるチャートの表による表現
Table 1 Table representation of chart for *Sōseki katta*.

#	Label	A-Set	Proc	#	Label	A-Set	Proc
1	[?]s(Φ, \perp, e) \uparrow	ϕ	—	29	[<i>intend</i> (<i>B</i>) [?]person(<i>S</i>) [?]agt(<i>(B, S)</i> _r) ^{s20} ga(<i>(P, B, S)</i> _s , <i>P</i>)	{ β }	C:27+K+13
2	[?]soseki(<i>S</i>) ^{s1} np(Φ, Ψ, S)	ϕ	I:1	30	[<i>trade</i> (<i>B</i>) [?]commodity(<i>S</i>) [?]obj(<i>(B, S)</i> _r) ^{s2} wo(<i>(P, B, S)</i> _s , <i>P</i>)	{ β }	C:28+K+12
3	soseki(<i>S</i>) ^{s1}	{ α }	I:2, I:18, I:29	31	[<i>intend</i> (<i>B</i>) person(<i>S</i>) [?]agt(<i>(B, S)</i> _r) ^{s20} ga(<i>(P, B, S)</i> _s , <i>P</i>)	{ α, β }	C:29+L+4
4	person(<i>S</i>)	{ α }	P:3	32	agt(<i>(B, S)</i> _r) ^{s20}	{ ϵ }	I:31
5	writer(<i>S</i>)	{ α }	P:3	33	ga(<i>(P, B, S)</i> _s , <i>P</i>)	{ α, β, ϵ }	C:31+M+32
6	np(Φ, Ψ, S)	{ α }	C:2+B+3	34	ga(<i>(P, B, S)</i> _s , <i>P</i>) [?]ga(<i>P</i>) ^{s3} sem(<i>(P, B, S)</i> _s)	{ α, β, ϵ }	P:33
7	pp(Φ, Ψ, P, S)	{ α }	P:6	35	ga(<i>P</i>) ^{s3}	{ ζ }	I:34
8	[np(Φ, Ψ, S) [?]p(Ψ, k, c) pp(Φ, k, c, S)	{ α }	P:6	36	sem(<i>(P, B, S)</i> _s)	{ $\alpha, \beta, \epsilon, \zeta$ }	C:34+N+35
9	[pp(Φ, Ψ, P, S) [?]vp(Ψ, k, e) [?]dep(<i>(P, e, S)</i> _d) vp(Φ, k, e)	{ α }	P:7	37	dep(<i>(P, B, S)</i> _d)	{ $\alpha, \beta, \epsilon, \zeta$ }	C:26+J+36
10	[?]buy(<i>B</i>) ^{s1} vp(Ψ, \perp, B)	ϕ	I:8, I:9	38	vp(Φ, \perp, B)	{ $\alpha, \beta, \epsilon, \zeta$ }	C:15+E+37
11	buy(<i>B</i>) ^{s1}	{ β }	I:10, I:27, I:28, I:42, I:43	39	s(Φ, \perp, B)	{ $\alpha, \beta, \epsilon, \zeta$ }	P:38
12	trade(<i>B</i>)	{ β }	P:11	40	T	{ $\alpha, \beta, \epsilon, \zeta$ }	C:1+A+39
13	intend(<i>B</i>)	{ β }	P:12	41	[<i>prag</i> (<i>(S)</i> _p , <i>N</i>) [?]sem(<i>(P, B, N)</i> _s) dep(<i>(P, B, S)</i> _d)	{ α, γ, δ }	C:16+F+25
14	vp(Ψ, \perp, B)	{ β }	C:10+D+11	42	[?]intend(<i>B</i>) [?]person(<i>N</i>) [?]agt(<i>(B, N)</i> _r) ^{s20} ga(<i>(P, B, N)</i> _s , <i>P</i>)	ϕ	I:41
15	[pp(Φ, Ψ, P, S) vp(Ψ, \perp, B) [?]dep(<i>(P, B, S)</i> _d) vp(Φ, \perp, B)	{ α, β }	C:9+C+14	43	[?]trade(<i>B</i>) [?]commodity(<i>N</i>) [?]obj(<i>(B, N)</i> _r) ^{s2} wo(<i>(P, B, N)</i> _s , <i>P</i>)	ϕ	I:41
16	[?]prag(<i>(S)</i> _p , x) [?]sem(<i>(P, B, x)</i> _s) dep(<i>(P, B, S)</i> _d)	ϕ	I:15	44	[<i>intend</i> (<i>B</i>) [?]person(<i>N</i>) [?]agt(<i>(B, N)</i> _r) ^{s20} ga(<i>(P, B, N)</i> _s , <i>P</i>)	{ β }	C:42+P+13
17	prag(<i>(S)</i> _p , <i>S</i>)	ϕ	I:16	45	[<i>trade</i> (<i>B</i>) [?]commodity(<i>N</i>) [?]obj(<i>(B, N)</i> _r) ^{s2} wo(<i>(P, B, N)</i> _s , <i>P</i>)	{ β }	C:43+P+12
18	[?]writer(<i>S</i>) [?]write(<i>(S, N)</i> _r) ^{s10} [?]novel(<i>N</i>) ^{s1} prag(<i>(S)</i> _p , <i>N</i>)	ϕ	I:16	46	[<i>trade</i> (<i>B</i>) commodity(<i>N</i>) [?]obj(<i>(B, N)</i> _r) ^{s2} wo(<i>(P, B, N)</i> _s , <i>P</i>)	{ β, δ }	C:45+Q+24
19	[<i>writer</i> (<i>S</i>) [?]write(<i>(S, N)</i> _r) ^{s10} [?]novel(<i>N</i>) ^{s1} prag(<i>(S)</i> _p , <i>N</i>)	{ α }	C:18+G+5	47	obj(<i>(B, N)</i> _r) ^{s2}	{ η }	I:46
20	write(<i>(S, N)</i> _r) ^{s10}	{ γ }	I:19	48	wo(<i>(P, B, N)</i> _s , <i>P</i>)	{ β, δ, η }	C:46+R+47
21	[<i>writer</i> (<i>S</i>) write(<i>(S, N)</i> _r) ^{s10} [?]novel(<i>N</i>) ^{s1} prag(<i>(S)</i> _p , <i>N</i>)	{ α, γ }	C:19+H+20	49	wo(<i>(P, B, N)</i> _s , <i>P</i>) [?]wo(<i>P</i>) ^{s3} sem(<i>(P, B, N)</i> _s)	{ β, δ, η }	P:48
22	novel(<i>N</i>) ^{s1}	{ δ }	I:21	50	wo(<i>P</i>) ^{s3}	{ θ }	I:49
23	book(<i>N</i>)	{ δ }	P:22	51	sem(<i>(P, B, N)</i> _s)	{ $\beta, \delta, \eta, \theta$ }	C:49+S+50
24	commodity(<i>N</i>)	{ δ }	P:23	52	dep(<i>(P, B, S)</i> _d)	{ $\alpha, \beta, \gamma, \delta, \eta, \theta$ }	C:41+O+51
25	prag(<i>(S)</i> _p , <i>N</i>)	{ α, γ, δ }	C:21+I+22	53	vp(Φ, \perp, B)	{ $\alpha, \beta, \gamma, \delta, \eta, \theta$ }	C:15+E+52
26	[<i>prag</i> (<i>(S)</i> _p , <i>S</i>) [?]sem(<i>(P, B, S)</i> _s) dep(<i>(P, B, S)</i> _d)	ϕ	C:16+F+17	54	s(Φ, \perp, B)	{ $\alpha, \beta, \gamma, \delta, \eta, \theta$ }	P:53
27	[?]intend(<i>B</i>) [?]person(<i>S</i>) [?]agt(<i>(B, S)</i> _r) ^{s20} ga(<i>(P, B, S)</i> _s , <i>P</i>)	ϕ	I:26	55	T	{ $\alpha, \beta, \gamma, \delta, \eta, \theta$ }	C:1+A+54
28	[?]trade(<i>B</i>) [?]commodity(<i>S</i>) [?]obj(<i>(B, S)</i> _r) ^{s2} wo(<i>(P, B, S)</i> _s , <i>P</i>)	ϕ	I:26				

$\Phi = [Sōseki, Katta], \Psi = [Katta],$

$\alpha = soseki(S)^{s1}, \beta = buy(B)^{s1}, \gamma = write((S, N)_r)^{s10}, \delta = novel(N)^{s1},$
 $\epsilon = agt((B, S)_r)^{s20}, \zeta = ga(P)^{s3}, \eta = obj((B, N)_r)^{s2}, \theta = wo(P)^{s3}$

仮定する際のコストは \$20 と高い。ここで考えている状況（話し手が自分の経験について話している状況）では、第三者の行為について話すことはあまりないと考えて、コストをこのように設定している*。一般に、コストを適切に設定するのは困難であるが、上述したように現実世界での事象の起こりにくさを反映したのがコストであると考えれば、言語コーパスの統計情報から適切なコストを自動的に設定できる可能性もあろう。

図7と表1は(1)を本手法によって解析した際に作られるチャートである。図中の実線は活性弧、破線は不活性弧、鎖線はポイントを表す。図中の弧にふられた数字は表の左端の列に現れる数字と対応している。なお、複雑さを抑えるために、同一の箇所活性弧あるいは不活性弧が2つ以上ある場合には、弧の番号のみ書き、弧そのものは重複して書かない。表中のLabelは弧のラベル、A-Setは仮説集合、Procはその弧がどの弧（とポイント）からどの手続きによって作られたかを表す。例えば、表1において、弧2のProc欄のI:1は、それが弧1から導入手続きによって作られたことを表している。同様に、弧4のProc欄のP:3は、それが弧3から予測手続きによって作られたことを表し、弧6のProc欄のC:2+B+3は、それが弧2と弧3とポイントBから結合手続きによって作られたことを表す。

本手法では、構文解析と意味・語用論解釈は同時に行われる。つまり、本手法によって構文処理と意味・語用論処理は1つの過程に統合される。図7はこの統合された過程を表している。この図はいくぶん込み入っているように見えるが、いくつかの断片に分けて考えると理解しやすい。索引[Sōseki, Katta]の位置から張られている弧2, 6, 7, 9, 15, 53, 54は、構文解析が行われる様子を表す。表からわかるように、この過程は上昇型チャートパーサによる解析過程と全く同じである。この過程の途中（弧15と弧53の間）に意味・語用論解釈の過程がはさまっている。索引<S>の位置から張られている弧18, 19, 21, 25は、『漱石』による提喻を解釈する過程を表す。この語用論処理の過程において、『漱石の本』に対応する新た

な対象Nが導入され仮説 *novel(N)* (弧22に相当) が作られる。索引<P, B, N>の位置から張られている弧43, 45, 46, 48, 49, 51は、動詞概念B(買う)と名詞概念N(小説)の間の(Pでマークされるべき)深層格関係を解釈する過程を表す。この意味処理の過程において、仮説 *obj(B, N)* (弧47に相当) と仮説 *wo(P)* (弧50に相当) がそれぞれコスト\$2と\$3で仮定される。これは、『漱石』と『買った』の間の助詞『を』の省略された係り受けがコスト\$5で対象格に解釈されたことを意味する。最後に、ラベルTを持つ2つの弧40, 55はこの解析の2つの解を表す。コスト\$25を持つ弧40は「漱石が何かを買った」という解釈に対応し、コスト\$18を持つ弧55は「話し手が漱石の本を買った」という解釈に対応する。したがって、最適解はコストの低い後者である。

3.4 アジェンダ制御

コスト付アブダクションにおいては、すべての解を求めることが目的ではなく最適解を求めることが目的なので、最適解を効率的に見つけるヒューリスティック探索を本手法と組み合わせることは有効である。本手法では、チャートパーサでも広く用いられているアジェンダ制御機構 (agenda control mechanism) を利用することによって、このようなヒューリスティック探索を容易に実現できる。

チャート法の導入/予測/結合手続きによって新たに作られた弧は、一旦アジェンダと呼ばれる領域に格納される。アジェンダ中の弧はなんらかの基準にしたがって1つずつ取り出され、チャートに追加される。もっとも単純な基準は、「アジェンダ中の弧で仮説集合に含まれる仮説のコストの総和が最小になるものを選ぶ」というものである。これはいわゆる順序探索 (ordered search) である。

順序探索は、最初に見つかる解が最適解であることを保証するが、必ずしも効率的であるとはいえない。そこで、最良優先探索やビーム探索のような最適性を保証しないがより実用的な探索法をとり入れることも考えられる。現段階では、これらの探索法については試していないが、アジェンダ制御機構を備えた本手法において、このような実用的な探索を容易に実現できることは明らかである。

4. 実験および考察

本手法の有効性を示すために簡単な実験を行った。この実験では次の4つのアブダクション計算法を比較

* ここでは、コストは文脈によらず一定であるかのように扱っているが、この扱いは十分ではない（文脈によっては、『漱石』を行為者とみなす解釈のほうが優先しうる）。文脈的な要因をいかにしてコストに反映させるかは、興味深い話題であるが、本稿の範囲を越えるので、今後の課題としたい。

表 2 TD, HD, GCF, GCO の比較 (ステップ数)
Table 2 Comparison among TD, HD, GCF, and GCO (steps).

#	TD	HD	GCF	GCO
1	285	162(0.57)	141(0.49)	111(0.39)
2	832	467(0.56)	273(0.33)	206(0.25)
3	1916	1070(0.56)	417(0.22)	274(0.14)
4	4082	2269(0.56)	585(0.14)	399(0.10)

した。

TD 下降型+表登録なし+全探索

HD 主辞駆動型+表登録なし+全探索

GCF 主辞駆動型+表登録あり+全探索

GCO 主辞駆動型+表登録あり+順序探索

TD は下降型導出を深き優先後戻り型で行うもの¹⁰⁾, **HD** は主辞駆動型導出を深き優先後戻り型で行うものである。**GCF** は全探索 (アジェンダ制御なし) の一般化されたチャート法であり, **GCO** は一般化されたチャート法と順序探索の組合せである。

実験に用いた規則は, 話し言葉解析のドメインで, 67 の連鎖規則と 48 の非連鎖規則からなるかなり小さなものである。試験文は 1 つの動詞と 1 ~ 4 個の名詞からなり (例えば (1) のような文), 全探索した場合の可能な解は名詞の数に対して指数的に増大する。実験は SS10 上で SICStus Prolog 2.1 を用いて行った。結果を表 2 に示す。評価はステップ数で行った (**TD** と **HD** では導出のステップ数, **GCF** と **GCO** では作られた弧の数)。括弧内の小数は **TD** と比較した場合の性能比である。

表から明らかのように, 本手法の 3 つの特徴によって計算効率が大きく改善されている。まず, **TD** から **HD** への改善は, 第 1 の特徴である目標駆動上昇型導出によるものであり, これによってステップ数が 40 数 % 削減された。次に, **HD** から **GCF** への改善は, 第 2 の特徴である部分解の表への登録によるものであり, これによってステップ数がさらに 10%-40% 削減された。最後に, **GCF** から **GCO** への改善は, 第 3 の特徴であるアジェンダ制御機構によるものであり, これによってステップ数がさらに 4%-10% 削減された。計算効率は最大の場合で約 10 倍に改善されている。

次に, 本手法の理論的な計算量について考察してみる。チャート法に基づく本手法の計算量は, 多項式オーダーである。ただし, これは入力文の長さに対してではなく, チャートに導入された索引の数に対してで

ある*。入力文の長さ l と索引の数 n との関係は, 規則の内容やヒューリスティック探索の方法によってさまざまである。すなわち, 曖昧性の少ない規則と効率の良いヒューリスティック探索を用いることができれば, 索引の数は入力文の長さに対して直線的であるが, 曖昧性の多い規則と効率の悪いヒューリスティック探索しか用いることができなければ, 索引の数は入力文の長さに対して指数的に増大する。したがって, 本手法の計算量は多分に実際の側面に依存する。すなわち, 本手法の計算量 C は, チャート中の索引の数を n , 入力文の長さを l とすると, 次式で表される。

$$C = K \times f(n) \quad n = g(l)$$

このような計算量の観点から見て, 本手法の 3 つの特徴による効率の改善は, 以下のようにとらえられる。

1. 目標駆動上昇型導出は計算オーダーに対するスケール (係数 K) を改善する。
2. 部分解の表への登録は計算オーダー (関数 f) を指数オーダーから多項式オーダーに改善する (ただし, チャート中の索引の数 n に対して)。
3. アジェンダ制御機構は入力文の長さ l に対する索引の数 n の増大の仕方 (関数 g) を改善する。

5. おわりに

本稿では, コスト付アブダクションにおける計算効率の問題について, 根本的で実際の解決策を提案した。本手法は以下の 3 つの特徴を持つ。

- 目標駆動上昇型導出
- 部分解の表への登録
- アジェンダ制御機構

これらの特徴によって計算効率が大きく改善されることを簡単な実験によって確認した。

最初に述べたように, 本手法は話し言葉の解析/理解といった問題に適用される。本稿の方法によって, 計算効率が大きく改善されたとはいえ, 実用的なシステムを作るにはさらなる効率化が望まれる。本手法が備えるアジェンダ制御機構は, そのような方向へ向かうためのよい足場を提供している。構文解析における同種の問題に関する成果を利用できるのも大きな利点である。今後, この方向へ研究を進展させたい。

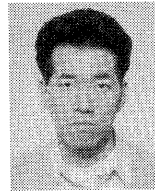
* 索引の数を n とすると, 結合手続きにおいて結合される活性弧の始点 s , 終点 t , 不活性弧の始点 u , 終点 v はそれぞれ n 通りずつあるから, この手続きが呼ばれる可能性は n^4 回ある。実際には, 弧の結合 (可能な t, u の組合せ) はポイントによって制限されているので, 結合手続きが呼ばれる回数は n^3 である。

謝辞 本研究の機会を与えて下さった ATR 音声翻訳通信研究所の山崎泰弘社長、御指導下さった飯田仁室長はじめ第3研究室の皆様へ感謝します。また、有益なコメントをいただいた奈良先端科学技術大学院大学の松本裕治教授および NTT コミュニケーション科学研究所の春野雅彦氏に感謝します。

参考文献

- 1) Charniak, E. and Husain, S.: A New Admissible Heuristic for Minimal-Cost Proofs, *Proceedings of the 12th IJCAI*, pp. 446-451 (1991).
- 2) Charniak, E. and Santos, E., Jr.: Dynamic MAP Calculations for Abduction, *Proceedings of the 10th AAAI*, pp. 552-557 (1992).
- 3) Charniak, E. and Shimony, S. E.: Probabilistic Semantics for Cost Based Abduction, *Proceedings of the 8th AAAI*, pp. 106-111 (1990).
- 4) Haruno, M., Den, Y., Matsumoto, Y. and Nagao, M.: Bidirectional Chart Generation of Natural Language Texts, *Proceedings of the 11th AAAI*, pp. 350-356 (1993).
- 5) Hobbs, J. R., Stickel, M. E., Appelt, D. E. and Martin, P.: Interpretation as Abduction, *Artif. Intell.*, Vol. 63, No. 1-2, pp. 69-142 (1993).
- 6) Kay, M.: Algorithm Schemata and Data Structures in Syntactic Processing, Technical Report CSL-80-12, XEROX Palo Alto Research Center (1980).
- 7) Pereira, F. C. N. and Warren, D. H. D.: Parsing as Deduction, *Proceedings of the 21st Annual Meeting of ACL*, pp. 137-144 (1983).
- 8) Shieber, S. M., van Noord, G., Moore, R. C. and Pereira, F. C. N.: A Semantic-Head-Driven Generation Algorithm for Unification-Based Formalisms, *Proceedings of the 27th Annual Meeting of ACL*, pp. 7-17 (1989).
- 9) Sikkel, K. and op den Akker, R.: Predictive Head-Corner Chart Parsing, *The 3rd International Workshop on Parsing Technologies*, pp. 267-276 (1993).
- 10) Stickel, M. E.: Rationale and Methods for Abductive Reasoning in Natural-Language Interpretation, Natural Language and Logic, *Proceedings of International Scientific Symposium*, Studer, R. (ed.) (1989), (*Lecture Notes in Artificial Intelligence*, Vol. 457, pp. 233-252, Springer-Verlag (1990)).
- 11) van Noord, G.: An Overview of Head-driven Bottom-up Generation, *Current Research in Natural Language Generation*, Dale, R., Mellish, C. and Zock, M. (eds.), Chap. 6, pp. 141-165, Academic Press (1990).
- 12) van Noord, G.: Head Corner Parsing for Discontinuous Constituency, *Proceedings of the 29th Annual Meeting of ACL*, pp. 114-121 (1991).

(平成6年2月14日受付)
(平成6年9月6日採録)



伝 康晴 (正会員)

1964年大阪府に生まれる。1988年京都大学工学部電気工学第二学科卒業。1993年京都大学大学院工学研究科博士後期課程(電気工学第二専攻)単位取得退学。在学中の1991年から1993年までATR自動翻訳電話研究所滞在研究員を勤める。1993年国際電気通信基礎技術研究所(ATR)入社。現在ATR音声翻訳通信研究所に勤務。自然言語処理、人工知能などの研究に従事。人工知能学会、日本ソフトウェア科学会、日本認知科学会、言語処理学会、ACL各会員。