自然言語処理 55－5
（1986. 5. 23）

自然言語の構文・意味解析規則の帰納的学習システム

山本 幹雄　　中川 聖一
豊橋技術科学大学　情報工学系

自然言語の構文・意味解析規則を、文とその意味の例から帰納的に学習するシステムを開発した。本システムは、J.Andersonの言語獲得の理論を用いており、Andersonが理論を実証するために作ったＬＡＳ（Language Acquisition System）を改良したものとなっている。改良したのは以下の点である。

（1）単語の意味も知らない状態から学習できるようにした。

（2）意味表現にセマンティック・ネットを用いて、複数の意味を持つ単語も学習できるようにした。

（3）学習のためのヒューリスティックを改良した。第一に、一般化しすぎたり、足らない場合を、第二に、意味を持たない単語に関する文法を学習するときの問題を改良した。

An Inductive Learning System of Natural Language Grammar

Mikio Yamamoto　and　Sei-ichi Nakagawa
Department of Information and Computer Sciences, Toyohashi University of Technology

We have developed a system that inductively learns the grammar from the examples of pairs of a sentence and its semantic representation.
This system employs the Anderson's language acquisition theory in cognitive psychology. He proved the validness of his theory by developing the LAS(Language Acquisition System). However, when we apply LAS to the natural language processing system, some problems arise as follows:
(1)to learn nothing but syntactic grammar.
(2)can't treat the word with multiple meaning.
(3)Heuristics of learnig sometimes makes the over- or undergeneralizaton and its ability of heuristics to process the morphemes is poor.
Our system uses the semantic network as the semantic reprisentation and CFG(Context-Free Grammar) as syntactical reprisentation. Our system can learn a grammar,without the meaning of words and a priori syntactic grammar, and improved of the each problem in LAS. Our system may be applied to any languages.

自然言語の構文・意味解析規則の帰納的学習システム

山本　幹雄*　　中川　聖一

豊橋技術科学大学　　情報工学系

1．はじめに

自然言語処理は１９７０年代からさかんに研究されているが、完全に実用化されたシステムはまだほとんどない。問題点の１つに、自然言語処理に必要な知識の膨大さがある。そのため現在の自然言語処理システムでは、領域を限定して限られた知識で動くように作られており、その知識を入れ替える事によって多くの領域に適用できる。しかし、このように領域を限定しても、なお知識を作るのは膨大な作業となる。たとえば、すべての現在の機械翻訳システムは基本的な知識は持っているが、使用する領域特有の知識はユーザーが自分で入れてやらなければならない。ここで、もし必要な知識をある程度自動的に獲得できればユーザーの負担は軽減できる。今回、我々は自然言語処理システムに必要な、構文・意味解析規則を解析例から帰納的に推論し学習するシステムを試作したので報告する。以前にも、構文規則を帰納的に学習する研究は、行なわれている。[3)5)8)] 本研究では、学習用の例として、自然言語文とその意味表現を入力する。

本研究の理論的背景としては、認知心理学における赤ちゃんの言語獲得のJ.Andersonの理論がある。[1)] 彼の理論の中心となる主張は文法規則のほとんどが意味構造から引き出されるということである。すなわち入力として文と意味のペアをいれてやれば解析規則のほとんとが推論可能であると言っている。さらに彼は、ＬＡＳというシステムで自分の理論をある程度実証した。このシステムは文とその意味が入力されると、その文の解析を試みる。解析が失敗したら、ある精密化のヒューリスティックによって、文法は訓練例を解析でき正しい意味を付与できるように拡張される。ここでAndersonは意味の使用以外に次の３つの仮定をしている。

（１）子供は単語の意味を知っている。
（２）子供は意味構造のトップ・ノードが、どれか知っている。
（３）解析木の枝は交差してはならない。

しかし、このシステムをそのまま自然言語処理に応用するには、以下の問題点がある。[2)]

（ａ）文法を学習するシステムであり、単語の学習ができない。
（ｂ）意味表現として単語を直接ノードとして使用したＨＡＭと呼ばれる特殊な意味表現を用いている。そのため、複数の意味を持つ単語が扱えない。
（ｃ）学習のためのヒューリスティックにいくつかの問題点がある。第一に、規則を一般化するときに一般化しすぎたり、したらない場合がある。第二に、意味を持っていない単語に関する規則を学習するときに問題がある。

本研究では上記の（２）と（３）の仮定はしているが、（１）の仮定はしていない。

精密化のヒューリスティックはいくつかあるが、Andersonの中心的な主張である「文法規則のほとんどが意味構造から引き出される」を手続き化したものは Tree Fitting Heuristic とSemantics-Based Equivalence Heuristic である。１つめのヒューリスティックが文法を提案し、２つ目のヒューリスティックが一般化する。２つ目のヒューリスティックを、一言でいうと「意味構造の中で同じ役割を持っていて、文の同じ位置に来る単語は同じクラスとしてよい」ということになる。

本研究では意味表現に一般的なセマンティック・ネットワーク、構文解析に文脈自由文法（ＣＦＧ）を用い、単語の意味も知らない状態から構文・意味解析規則を解析例から帰納的に学習するヒューリスティックを開発した。このヒューリスティックはＬＡＳの問題点を改良してある。どんな言語でも学習できることを目標としている。

2．システムの構成

図１に本システムの構成を示した。学習部は文と意味のペアを受け取り、新しい構文・意味解析規則を推論して古い構文・意味解析規則を変更する。推論するときに構文・意味解析部を使用するヒューリスティックもあるので、図１のような表現になっている。学習が終わった状態では、構文

＊現在；（株）沖テクノシステムズ・ラボラトリ

・意味解析部に文を入力することによってその文の意味を得ることができる。

意味表現には現在の自然言語処理で一般的に使用されているセマンティック・ネットワークをもちいた。本研究ではNorman and Rumelhart[4]のセマンティック・ネットワークに基づいたものを使っている。さらに、学習部におくられる意味表現は、解析部が意味表現を構成するときに使われる命令に分解される。基本的には、ノードを発生する命令と、ノードを接続するリンクの発生の命令に分解される。図２に命令の種類をあげ、図３に分解の例を示す。

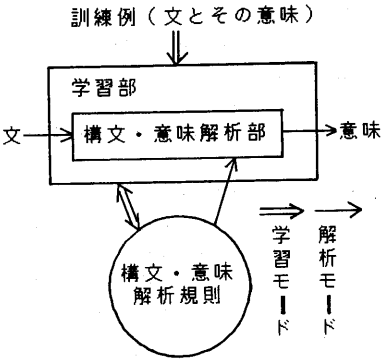

図１　本システムの構成

ノードを発生する命令

ＧＥＮＲ：述語を発生
（図４の"○"に対応）
ＧＥＮＯ：対照を発生
（図４の"< >"に対応）
ＧＥＮＥ：概念を発生
（図４の"【 】"に対応）

リンクする命令

ＰＵＴＮ：２つのノードをある
ラベルでつなげる

図２　意味表現を作る命令

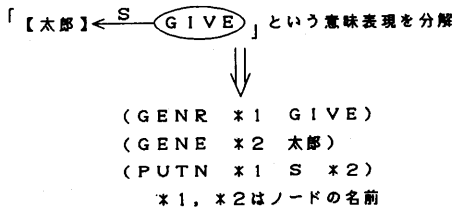

図３　意味表現の分解

解析部には拡張ＬＩＮＧＯＬ[07]を改造したものを使用している。主な改造は以下のとおりである。

（１）意味解析部が構文解析と同時に走るようにし、意味解析の失敗からもバック・トッラクがかかるようにした。
（２）解析結果としての意味表現に、セマンティック・ネットワークを用いるために、そのように意味解析部を改造した。

解析規則と解析例を図４に示す。

学習部は文とその意味としてのセマンティック・ネットワークを受け取り、まず文の解析を試みる。その結果が与えられたセマンティック・ネットワークと同じになれば、学習部はなにもしない。しかし解析が失敗するか、又は解析結果が違って

解析規則

表層部	意味部
S→NP　VP	― NPをVPにラベルSでリンク
VP→V　NP	― NPをVにラベルｏｂｊでリンク
V→ｇａｖｅ	― ○ｇｉｖｅ○ノードを発生
NP→ｔｈｅ N	― Nの意味
NP→ N	――― Nの意味
N→ Ｍａｒｙ	―― 【Ｍａｒｙ】ノードを発生
N→ ｄｏｇ	―― <ｄｏｇ>ノードを発生

解析例（上の規則で）

入力：Ｍａｒｙ　ｇａｖｅ　ｔｈｅ　ｄｏｇ．

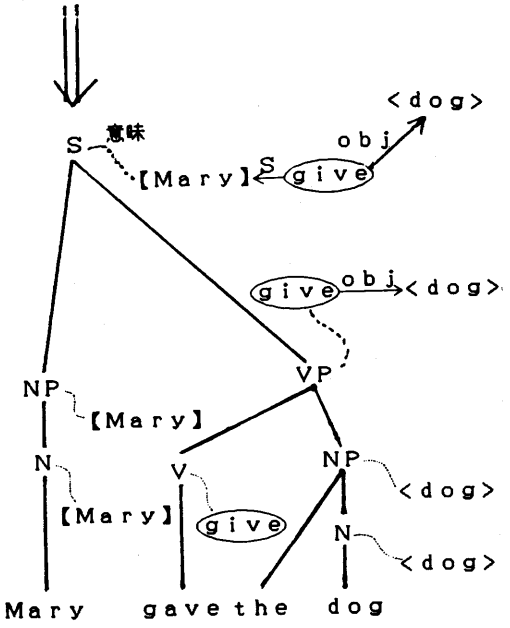

図４　解析部の解析規則と解析例

いればシステムはまずそれを暗記学習する。その後にその暗記学習したものと、もともと持っていた規則の間に一般化できる部分を探し一般化する。また、さらに暗記学習したものの解析規則を推論しその文を正しく解析できるように規則を一般化する。この一般化の手続きは「学習のヒューリスティック」の節で詳しく論じる。またシステムは、いつでも規則の一般化をできるように、逐次的な処理となっている。学習部の流れ図を図５に示す。

３．学習のヒューリスティク

本研究で使った学習部の一般化のヒューリスティックを具体的に述べる。以下に挙げるように３つのヒューリスティックがある。

（１）ＡＮＤヒューリスティック
（２）部分解析ヒューリスティック
（３）マージヒューリスティック

まず３つのヒューリスティックに共通する学習される言語と意味の制約と、学習に使用する情報を述べてから、それぞれのヒューリスティックを説明する。

本研究で仮定した、学習される言語と意味の制約は以下の２つである。

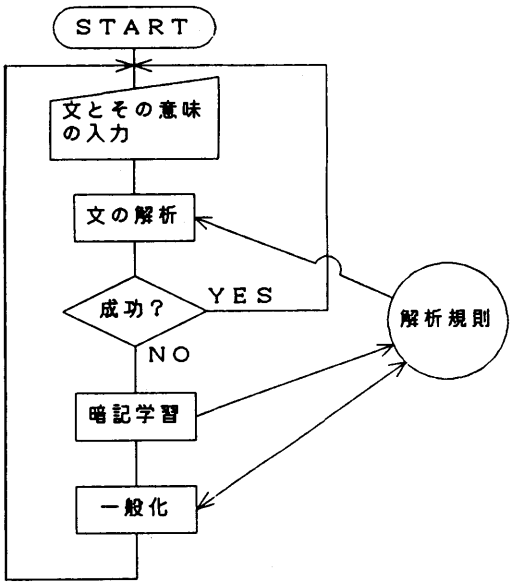

図５　学習部の流れ図

（１）ＣＦＧで解析できる言語であること。
（２）意味構造のノードをたどっていった時にループしてはならない。

帰納的推論では可能な推論結果の制約がきびしいほど早く、正確に推論できる。だから、できるだけ制約のある文法を使用したほうが、よい結果が得られると考えられる。Chomsky の文法の一般性の４階層で２番目に制約のきびしい文法がＣＦＧである。ＣＦＧは完全ではないが、ほとんどの自然言語の文を解析できるので、自然言語処理では、よく使われている。そこで、本研究でもＣＦＧを文法として使用した。またＣＦＧを文法とすることで、自動的にAndersonの仮定した「解析木の枝が交差してはならない」という条件を本研究でも仮定している事になる。もしこの仮定をしなければ、文法が学習不能になることをAndersonは示唆している。

意味構造から文法を引き出すのだから、意味構造がＣＦＧと似ても似つかぬものであったら、推論不能となる。そこで意味構造にもＣＦＧに基づいた制約が必要となる。意味構造がＣＦＧと似た形（木構造）になるためには、ノードをたどっていった時にループしない事が要求される。

前節で述べたように人力として表層文と意味構造を受け取るが、それ以外に意味構造の、どのノードがトップ・ノードであるかの情報も入れてやらなければならない。トップ・ノードは意味の視点を表わしている。たとえば、”猫がねずみを食べた。”という文のトップ・ノード、すなわち視点は、”食べた”である。”ねずみを食べた猫”という文の視点は”猫”である。この２つの文の意味構造は同じであるが、解析木は異なっている。視点によって推論されるべき解析木が変化するということは、視点の情報も文法推論に重要な役割を持っているという事である。

３．１　ＡＮＤヒューリスティック

このヒューリスティックによって、単語の意味を決める事ができる。解析できない文は、まず暗記学習される。すなわち、その文しか解析できない１個の文法が作られる。そのような２つの文が暗記学習されており、それらの表層文に共通な単語列がただ１つだけあり、かつそれらの意味構造にも共通な構造がただ１つだけあるとき、その単語列がその意味を持っていると考えてよい。ただし、その意味構造は残りの意味構造と１つだけのリンクによってつながっていなければならない。そのリンクでつながっているノードがその単語列のトップ・ノードとなる。２つのリンクでつなが

っていると、トップ・ノードが決定できない。また、もとの意味のトップ・ノードを含んでいる場合は、そのトップ・ノードがリンクしていなければならない。こうして、より小さな単語列へ分解してゆき、最後には１つの単語の意味がわかる。

次に、このヒューリスティックは共通部分以外の残りの部分に対する文法も作ろうとする。残りの単語列が共通部分によって２つに分割されずに連続して残っているならば、残りの単語列を残りの意味に対応付ける文法を作る。分割されていれば、そのような文法は作らない。

さらに、分解した単語列を１つにまとめる文法を作ってこのヒューリスティックは終わる。

図６に例を示す。

３．２　部分解析ヒューリスティック

文を全部は解析できないが部分的には解析できるとき、このヒューリスティックが使われる。１つの部分単語列が解析され、残りの部分が解析不能の場合には、ＡＮＤヒューリスティックと全く同じ働きをする。解析できた単語列と意味がＡＮＤヒューリスティックで処理対象となった共通部分の対象になる。又、２つの部分単語列が解析されそれぞれの意味でもとの文の意味を作る事ができる時は、それらをまとめて、その文の意味になるような文法を作る。その時、２つの意味のトップ・ノードを結ぶことによって意味が完成されなければならない。これらの過程を図７に具体例で説明する。

意味を持たない単語が発見された時は、その単語の表層上の両どなりのどちらの単語にその単語を付けるか問題となる。本システムでは、セマンティック・ネットワーク上で指されている単語の方に意味を持たない単語を結びつける。又、両どなりのどちらも指されていたら、解析木の枝がその意味を持たない単語のすぐ上をとおる単語に結

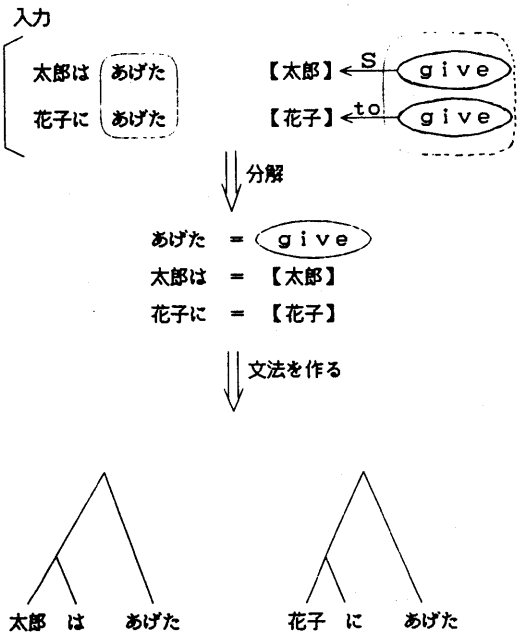

図６　ANDヒューリスティックの動作例

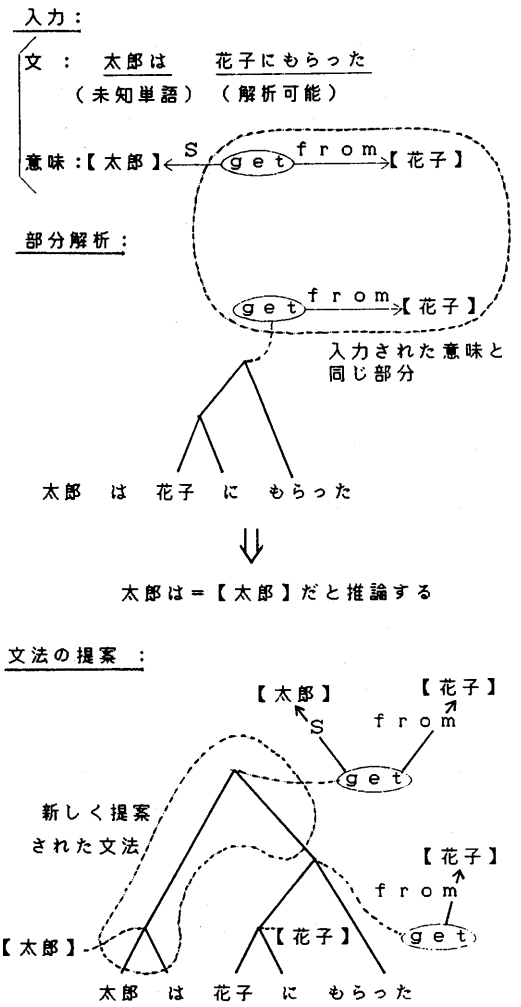

図７　部分解析ヒューリスティックの動作例

びつける。これはＬＡＳのように常に右の単語に結びつけるより、妥当性のある文法を生み出す可能性が、日本語と英語の場合は高い。

３．３　マージヒューリスティック

ＡＮＤヒューリスティックと部分解析ヒューリスティックは、暗記学習された文法をより構造的に解析できるように特殊な文法を付け加えていくだけである。ある程度の一般化はそのヒューリスティック内に内在しているものの、まだ一般化の余地はある。その一般化をするのがマージヒューリスティックである。これはＬＡＳのSemantics-Based Equivalence Heuristic と等価である。この Anderson のヒューリスティックを本システムの言い方に直すと「文法の右辺のある非終端記号の名前を変えると、文法もそれに付いている意味規則も同じになる２つの文法がある時、それらの文法は１つの文法にしてよい。」という事になる。具体的に、どうなるか以下に示す。つぎの４つの文法があるとする。

（１）Ｓ＝ＮＰ　Ｖ
　　（意味部：ＮＰをＶにラベルＳのリンクでつなげる）
（２）Ｓ＝ＮＰ　Ｖ１
　　（意味部：ＮＰをＶ１にラベルＳのリンクでつなげる）
（３）Ｖ＝give
　　（意味部：giveの概念）
（４）Ｖ１＝have
　　（意味部：haveの概念）

これは give と have が、まだ１つのカテゴリーになっていない状態である。マージ・ヒューリスティックによって、１つのカテゴリーにまとめられる。（２）のＶ１をＶに変えると、文法も意味も完全に（１）と同じになる。この事は２つの文法を１つにしてよいとマージ・ヒューリスティックは判断する。　Ｖ１をＶに規則全体にわたって書き換えて、（２）を消す。文法は次のように変更される。

（１）Ｓ＝ＮＰ　Ｖ
　　（意味部：ＮＰをＶにラベルＳのリンクでつなげる）
（３）Ｖ＝give
　　（意味部：giveの概念）
（４）Ｖ＝have
　　（意味部：haveの概念）

４．学習ヒューリスティックの評価

帰納的学習とは、特殊なデータから一般的なデータを推論する過程であるから、その評価は、正しい一般化ができているか否かと、一般化の程度によってなされる。すなわち、一般化のための特殊なデータがその一般化によって解析できなくなる、あるいは誤った解析をするようになるとしたら、その一般化は根本において誤っている。また一般化の程度とは、一般化をしてはいけない所まで一般化をした場合（　over　generalization　）と、一般化をしてもいいのに、しなかった場合（under generalization）を評価するものである。以下、それぞれのヒューリスティックについて順番に評価をする。

４．１　ＡＮＤヒューリスティックの評価

このヒューリスティックの誤った動作は以下の３つが考えられる。

（１）単語列に意味を付け過ぎる。
（２）単語列に付けた意味が足らない。
（３）単語列に間違った意味を付ける。

表層と意味が１対１で対応している単語と多義語だけの時はうまくいくが、同義語をいくつか持つ単語が単語列にはいっている場合に上であげた誤った動作をする可能性がある。図８、図９にこの様子を示す。

この問題を解決するためには、ＡＮＤヒューリスティックで使用するデータを多くすることが考えられる。すなわち、いまのヒューリスティックでは２つの文しか使っていないが３つ以上の文を

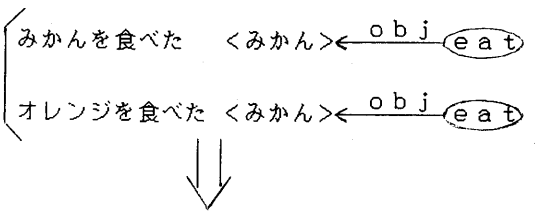

図８　意味の過不足

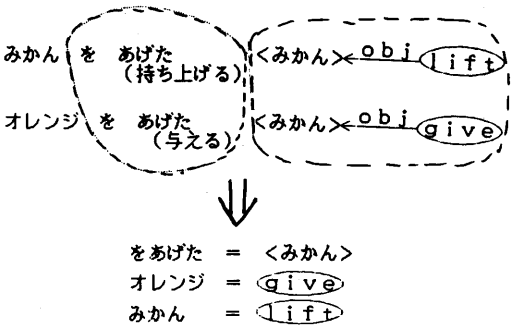

図９　誤った意味付け

使用すると安全性が高まる。しかし現実には、使用するデータを増やすと全ての文に対してこのヒューリスティックを使う時に、組み合わせ的爆発がおこるため、よい方法とはいいがたい。本システムでは、以前すでに入力されている単語の同義語となる単語を学習する時は、以前に入力された単語が十分に学習され意味が確立しているか、あるいはその単語が入っている単語列の他の単語はすでに解析可能となっている事を条件として、この問題を回避している。この条件は、ようするに同義語は十分時間間隔をおいて入力してやることを指示している。またシステムが持っている文法が増えれば増えるほどこの条件を犯す確率は減る。このような条件のもとで、ＡＮＤヒューリスティックのまえに必ず部分解析ヒューリスティックが起動されるようにすれば、ＡＮＤヒューリスティックが過ちを起こす前に部分解析ヒューリスティックがその文をばらばらにしてしまい、ＡＮＤヒューリスティックに文がかからなくしてしまう。このようにすれば、問題はおきなくなる。

４．２　部分解析ヒューリスティックの評価

このヒューリスティックは、Anderson のヒューリスティックの一般化が足らない場合を改善している。Andersonのヒューリスティックが一般化できない例は次の文のような場合である。

The cop frightens the thief.
The cop is frightened by the thief.
The cop tends to like thieves.

”The cop ”は文上の同じ構成要素でありながら意味表現での役割が違うため、ＬＡＳの「意味表現での役割が同じ文法は１つにまとめてよい」というヒューリスティックでは一般化できない。そのため、上の例文の”The cop ”は全部違う文法でそれぞれ解析される。しかし部分解析ヒューリスティックは、前に学習した文法が使えるときはそれを利用して新しい文法を作って行くのでこのような問題は起こらない。

部分解析ヒューリスティックは、基本的には文法を提案するヒューリスティックであり、大きな一般化は行なわないので一般化しすぎる問題は起きない。

４．３　マージヒューリスティックの評価

まず、一般化のしすぎから検討していく。このヒューリスティックが一般化しすぎる例をあげる。次のような文のための文法が学習されていたとする。

Ｓ＋gave＋人＋物　　（１）
Ｓ＋gave＋物＋to＋人　　（２）

この文法は同じ意味を生成する。ここで次のような文のための文法が学習されたとする。

Ｓ＋donated ＋物＋to＋人　（３）

（２）と（３）の文法が比較され gave と donated　が同じクラスにマージされる。すると、次のような文も許されることになる。

Ｓ＋donated ＋人＋物　　（４）

（４）のような文は英語では許されない。一般化をしすぎたことになる。この問題は微妙な意味の違い、あるいは感覚の問題であるため解決はむずかしく、ＬＡＳでも解決はされていない。ＬＡＳで問題となった、他の一般化のしすぎは次のような場合である。名詞句を学習している時、学習している言語ではsquareは名詞にも形容詞にもなるが redは形容詞にしかならないと仮定する。the square thing, the red thing, the square は名詞句になるがthe red は名詞句にならない言語である。Andersonのシステムでは the redが許される文法を作ってしまう。本システムでは、この一般化のしすぎは起こらない。しかし、この違いは２つのシステムの意味構造の違いからきているものである。本システムでは、red の役割（すなわちred の概念を指すリンクのラベル）と、square の役割が違う構造になっているが、Andresonの意味表現では同じ構造になっているからである。この場合は本システムのほうがよくなったが、このために悪くなる場合も生じる。一般化が足らない場合が出て来るのである。

これは意味ネットワークのリンクのラベル名の不統一から生じる。例えば、giveのリンクのラベル名と、haveのリンクのラベル名を全部違えたら、

giveとhaveは永遠に１つのクラスにはマージされない。

５．システムの学習例

日本語の実際の学習例を説明する。図１０にシステムへの入力、図１１にその入力により得られた文法を示す。また、図１１の文法によって解析された文の例を、図１２に示す。

図１０のＩＮＰＵＴ１、ＩＮＰＵＴ２で、”太郎は”と、”犬を”と、”あげた”が、ＡＮＤヒューリスティックにより学習されている。ＩＮＰＵＴ３で、”もらった”が部分解析ヒューリスティックにより学習されている。ＩＮＰＵＴ４で、”～は＋～を＋動詞”の文形が学習されている。ＩＮＰＵＴ５で、”花子”と”太郎”が同じカテゴリーに属している事と、助詞（”は”、”を”）を学習している。ＩＮＰＵＴ６で”猫”を、ＩＮＰＵＴ７で”持っている”を学習している。ＩＮＰＵＴ８，ＩＮＰＵＴ９で、”白い”と、”黒い”の形容詞を学習している。ＩＮＰＵＴ１０，ＩＮＰＵＴ１１で、助詞”に”と、その働きを学習している。ＩＮＰＵＴ１２で、動詞の連体形を学習し複文が解析できるようになった。

```
INPUT1: TAROU WA AGETA
  MEAN; ((GENE *1 TAROU)(GENR *2 GIVE)
         (TOP *2)(PUTN *2 S *1))
INPUT2: INU WO AGETA
  MEAN; ((GENO *1 DOG)(GENR *2 GIVE)
         (TOP *2)(PUTN *2 OBJ *1))
INPUT3: INU WO MORATTA
  MEAN; ((GENO *1 DOG)(GENR *2 GET)
         (TOP *2)(PUTN *2 OBJ *1))
INPUT4: HANAKO WA NEKO WO MORATTA
  MEAN; ((GENE *1 HANAKO)(GENO *2 CAT)
         (GENR *3 GET)(TOP *3)
       (PUTN *3 S *1)(PUTN *3 OBJ *2))
INPUT5: HANAKO WA MORATTA
  MEAN; ((GENE *1 HANAKO)(GENR *2 GET)
         (TOP *2)(PUTN *2 S *1))
INPUT6: NEKO WO MORATTA
  MEAN; ((GENO *1 CAT)(GENR *2 GET)
         (TOP *2)(PUTN *2 OBJ *1))
INPUT7: INU WO MOTTEIRU
  MEAN; ((GENO *1 DOG)(GENR *2 HAVE)
         (TOP *2)(PUTN *2 OBJ *1))
INPUT8: SIROI INU WO AGETA
  MEAN; ((GENE *1 WHITE)(GENO *2 DOG)
         (GENR *3 GIVE)(TOP *3)
         (PUTN *3 OBJ *2)
         (PUTN *2 COLOR *1))
INPUT9: KUROI NEKO WO MOTTEIRU
  MEAN; ((GENE *1 BLACK)(GENO *2 CAT)
         (GENR *3 HAVE)(TOP *3)
         (PUTN *3 OBJ *2)
         (PUTN *2 COLOR *1))
```

図１０　入力データ

（右上に続く）

```
INPUT10: NEKO NI AGETA
  MEAN; ((GENO *1 CAT)(GENR *2 GIVE)
         (TOP *2)(PUTN *2 TO *1))
INPUT11: TAROU WA NEKO WO HANAKO NI
         AGETA
  MEAN; ((GENE *1 TAROU)(GENO *2 CAT)
         (GENE *3 HANAKO)(GENR *4 GIVE)
         (TOP *4)(PUTN *4 S *1)
         (PUTN *4 OBJ *2)
         (PUTN *4 TO *3))
INPUT12: NEKO WO AGETA TAROU WA INU
         WO MORATTA
  MEAN; ((GENO *1 CAT)(GENR *2 GIVE)
         (GENE *3 TAROU)(GENO *4 DOG)
         (GENR *5 GET)(TOP *5)
         (PUTN *5 S *3)
         (PUTN *5 OBJ *4)
         (PUTN *2 OBJ *1)
         (PUTN *2 S *3))
```

図１０　入力データ

（左下からの続き）

```
#1 n1->TAROU ((TOP *1)(GENE *1 TAROU))
#2 n1->HANAKO((TOP *1)(GENE *1 HANAKO))
#3 n2->INU ((TOP *1)(GENO *1 DOG))
#4 n2->NEKO((TOP *1)(GENO *1 CAT))
#5 a->KUROI ((TOP *1)(GENE *1 BLACK))
#6 a->SIROI ((TOP *1)(GENE *1 WHITE))
#7 v->MOTTEIRU((TOP *1)(GENR *1 HAVE))
#8 v->AGETA ((TOP *1)(GENR *1 GIVE))
#9 v->MORATTA((TOP *1)(GENR *1 GET))
#10 par1->WA    ()
#11 par2->WO    ()
#12 par3->NI    ()
#13 np1->n1 par1 ((TOP n1))
#14 np1->rentai np1
      ((TOP np1)(PUTN rentai S np1))
#15 rentai->np2 v
      ((TOP v)(PUTN v OBJ np2))
#16 np2->n2 par2 ((TOP n2))
#17 np2->a np2
      ((TOP np2)(PUTN np2 COLOR a))
#18 np3->n1 par3 ((TOP n1))
#19 np4->n2 par3 ((TOP n2))
#20 s->np1 v((TOP v)(PUTN v S np1))
#21 s->np2 v((TOP v)(PUTN v OBJ np2))
#22 s->np4 v((TOP v)(PUTN v TO np4))
#23 s->np1 np2 v
      ((TOP v)(PUTN v S np1)
               (PUTN v OBJ np2))
#24 s->np1 np2 np3 v
      ((TOP v)(PUTN v S np1)
               (PUTN v OBJ np2)
               (PUTN v TO np3))
```

() の中が意味部

カテゴリー名は、わかりやすいように、変えてある。

図１１　学習された文法