人 工 知 能 72－2
（1990．9．12）

# 推論パス・ネットワーク上での類推による高速仮説推論システム

阿部 明典　石塚 満

東京大学

仮説推論は実用性も備えた有用な知識処理の枠組みであるが、推論速度が十分でないことが大きな問題である。これに対し、過去に推論された類似事例の結果を利用して高速化を図る方法を記す。類似事例の解仮説のどの部分が利用できるかの選択、および必要な新たな仮説の生成を推論パス・ネットワークを利用することで極めて効率的に行えることを示す。非単調推論系である仮説推論の計算複雑度はＮＰ－ｃｏｍｐｌｅｔｅであり、通常の推論の範囲に留まっていたのでは指数オーダーの推論時間の壁は越えられないが、類推の利用によりこの壁を平均値的に越える方法を与えている。

# Fast Hypothetical Reasoning System using Analogy on Inference-path Network

Akinori ABE and Mitsuru ISHIZUKA

Univ. of Tokyo

7-22-1 Roppongi,Minato-ku,Tokyo 106,JAPAN

The crucial problem of hypothetical reasoning system is its slow inference speed, while it is a very useful framework in knowledge processing. We present a fast inference mechanism for the hypothetical reasoning by using analogy of the results which were previously proved to be true. An inference-path network can be effectively used for selecting useful hypotheses from analogical case, and for generating new hypotheses which are necessary for inference. The inference speed of the hypothetical reasoning, whose computational complexity has been proved to be NP-complete, can not be improved from the exponential-order limit, if we still use ordinary search methods. However, this paper shows that this limit can be overcome in average inference time by using analogy.

## 1. まえがき

仮説推論は不完全な知識も扱えるという点、及び、設計や診断などの領域の実用的問題に適用できるという点で今後の知識処理に於ける有力な枠組みである[1-6]。しかし、推論速度の遅さが大きな問題であった。この問題に対処すべく我々のグループにより推論パス・ネットワークを利用する仮説推論システム[7]が開発された。これは仮説間の無矛盾性管理を行う際に生ずるバックトラックを推論パス・ネットワークに沿う前向き並列推論によって回避することによってＰｒｏｌｏｇの推論機構を利用するインプリメントと比較して１０００倍以上の推論速度の向上が達成されている。しかしながら、推論時間は仮説数の増大につれて指数オーダーに増大している。最近、仮説推論を始めとする非単調推論の計算複雑度はＮＰ－ｃｏｍｐｌｅｔｅであると証明されており[8]、[9]、この指数オーダーの推論速度の壁は通常の探索による求解の方法に留まっていたのでは越えることができない。

本稿では過去に成功した解を事例として利用する類推機構により、この指数オーダーの推論速度の壁を越える高速仮説推論を実現する手法を示す[10-12]。設計問題などでは過去の成功例を部分的に修正することによって新たな課題に対する解を求めることが多いが、ここに示す手法はこのような思考法をメカニズム化したものであるといえる。

類推は最近ではＣＢＲ（Case-based Reasoning）[15]としての実用的応用が図られている。通常の類推では過去に経験し、求解を行なった事例を活用して知識の拡大を行い、未経験の状況に対して知識ベースの知識の不足を補う意味が強いが、この場合には類推によって得られた解の妥当性が大きな問題となる。これに対し、本稿での類推の利用は零から出発して仮説の生成（即ち求解）を行なうのではなく、事例仮説を出発点にして仮説の生成を行なうことにより高速化を図るもので、解の妥当性は類似事例を見いだした後の論理を基盤とした推論の正当性によって保証される。

推論パス・ネットワークを基にすることによりこのような高速仮説推論用の類推機能が極めて効果的に実現されることを示す。ここでは知識表現は変数を含まないホーン節命題論理を対象にする。

## 2. 仮説推論

### 2．1 仮説推論の原理

仮説推論とは一般に真か偽か不明な事柄をとりあえず真として（仮説をたてて）推論を進め、うまくゴールに到達したら、たてた仮説は正しかったとみなす推論法である。ここでの仮説推論はＰｏｏｌｅ等がＴｈｅｏｒｉｓｔ[2]によって提示した論理に基づく仮説推論であり、矛盾を含む知識を仮説として扱うことにより不完全な知識を扱うことが出来る。

基本的には仮説推論は、ある事実（Ｆ）では観測（Ｏ）の説明がつかない時、仮説集合（Ｈ）から事実及び、他の仮説と無矛盾な仮説（ｈ）を取り出し、それを事実に加えることで観測を説明しようとする推論法であり、概念的には図１に示すような推論形態をとる。論理的に表すと以下のようになる。

Ｆ　・・・　問題を解くのに利用可能な既知の事実（矛盾性のない知識）
Ｏ　・・・　与えられた証明されるべき観測の集合
Ｈ　・・・　前以て与えられた、推論に使用可能な仮説集合（矛盾の可能性がある）
ｈ　・・・　Ｏを説明するのに適切な仮説の部分集合（ｈ⊆Ｈ）

とすれば、Ｏを証明するのに必要な仮説の部分集合ｈとは次のものである。

```
Ｆ⊬Ｏ              ／＊ ⊬とは証明不可能を表す ＊／
但し、Ｆ⊬□         ／＊ □は空節であり、Ｆに矛盾が含まれないことを表す ＊／

ＦＵｈ⊢Ｏ                    ／＊ Ｕは和集合を表す ＊／
ＦＵｈ⊬□
```

即ち、Ｆからだけではを証明できないとき、Ｆと無矛盾な仮説の部分集合ｈを抜きだしてＦに追加することにより、Ｏを証明できるようなｈを見いだす。

### 2．2 仮説推論の問題点

仮説推論システムでは仮説という常に真とは限らない（矛盾の可能性がある）知識の導入により知識の幅が広がった。それにより、非単調推論を行えるようになった。更に、この推論法は設計や診断等の領域に直接適用でき、考え方の枠組みは良い推論法である。しかし、推論速度の遅さが大きな問題である。

仮説推論システムはＰｒｏｌｏｇの推論機構を利用することにより比較的容易にインプリメントできるが、Ｐｒｏｌｏｇベースの仮説推論では探索に於けるバックトラックを単純なバックトラックに頼っているので、場合によっては観測の証明に寄与しないものまで探索してしまうなど十分な推論速度が得られない。更に、仮説推論固有の制御機構として、仮説間の無矛盾性維持によるバックトラックも生じることになるが、それも同じ無矛盾チェックを何度も行う等、非効率的である。

例えば、図2はＰｒｏｌｏｇの推論機構に基づき観測“ａ”を証明するための仮説生成のプロセスを示しているが、2回｛ｃ，ｇ，ｈ｝を生成してしまっている。

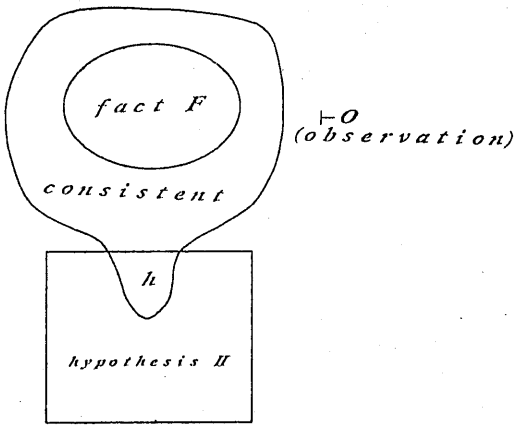


図1　仮説推論の概念図

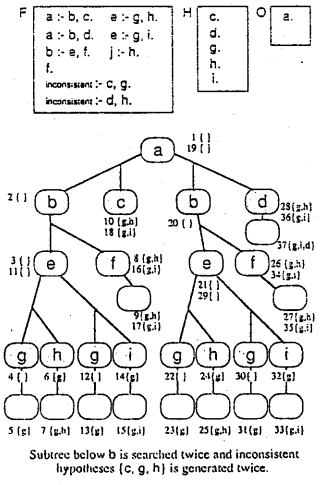


図2　Ｐｒｏｌｏｇ上での推論

## ３．　推論パス・ネットワークを利用する高速仮説推論法[7]

Ｐｒｏｌｏｇベースの仮説推論システムでは仮説間の矛盾によるバックトラック、無矛盾性のチェックを行う際に同じ事を再度行ってしまう点、仮説の変更に関係無いところまでチェックしてしまう点等という問題があり、それが推論速度を大きく落としていた。推論パス・ネットワークを利用する高速仮説推論法では先ずルール型の仮説、例えば

**仮説　a :- b.**

があるとき、ボディ部に仮説アトム“c ”を加えて以下に示すように変換する。

**事実　a :- b, c.**

**仮説　c.**

これにより、仮説は全て探索木上でリーフ位置にくるようになる。そして、仮説を除く完全な知識に対して後向き推論により推論パスの設定を行い、更にはこの推論パスのネットワークへのコンパイルを行う。その後に前向き推論によりこの推論パス・ネットワークに沿って並列に仮説の合成を行なうことにより仮説間の矛盾によるバックトラックなどを回避し、上記のような問題を解決している。推論パス・ネットワークの形式はホーン節命題論理に対する充足判定の線型時間アルゴリズム[13]に基づいている。又、前向きの並列仮

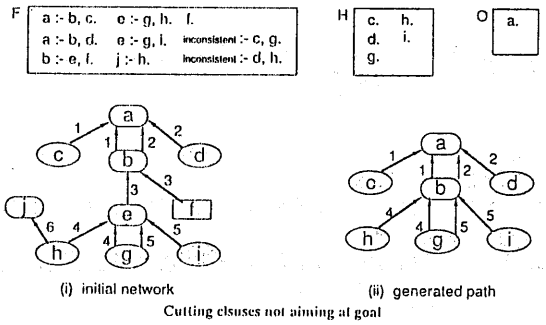


図3　推論パス・ネットワークでのパス生成例

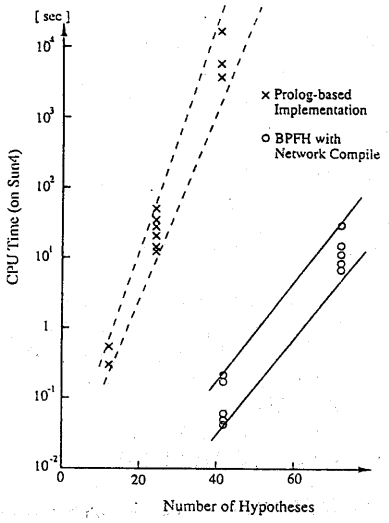


図4　Ｐｒｏｌｏｇに基づくシステムとの速度比

説合成はＡＴＭＳ[14]で採られているのと同様の方法で行われている。後向き推論による推論パス・ネットワーク形成と前向き並列推論を組み合わすことにより、同じサブゴールを複数回探索してしまう後向き推論の非効
率性、ゴール導出と関係の無い解まで探索してしまう前向き推論の非効率性を取り除いている。

図3にこのシステムによる推論パスの生成の様子を示し、図4にＰｒｏｌｏｇによる仮説推論システムとの速度比を示すグラフを示す。尚、この速度はＳＵＮ４／２６０上で測られており、Ｐｒｏｌｏｇベースの仮説推
論システムと比較して１０００倍以上の高速化が達成されていることを示している。しかしながら、推論時間は仮説数に対して指数オーダーに留まっている。

## ４．　類推を用いる仮説推論の高速化

### ４．１　仮説推論のプロセスと結果の保持[10]、[11]

類推は人間が日常的に行う推論の形式であり、過去の事例を参考にして欠落した知識を補ったり、推論の効率化に利用したりしている。これまで類推は基本メカニズムの研究が中心であったが、最近ではＣＢＲ（Case-b
ased Reasoning）[15]としての実用的利用の研究が進んでいる。ここでは仮説推論の効率化に類推を利用するメカニズムについて考える。

人間が問題を解決にあたるときに、全くの無から考えることはほとんどないといえる。過去の経験に類似性を見いだし、過去の経験に基づく台本に従って問題にあたり、多くの場合時間のかかる深い推論をする事なく妥
当な解を求めている。従って、過去の思考結果を台本のような形で保存しておくのは探索の効率化に有効である。この考えを仮説推論の高速化に利用するのであるが、過去の推論プロセスと結果を示す台本に相当するデータの記録法としてｊｕｓｔｉｆｉｃａｔｉｏｎの考え方を用いる。

ｊｕｓｔｉｆｉｃａｔｉｏｎとは本来はＴＭＳ（Truth Maintenance System）[16]においてそれぞれのデータの由来を示しておき、それによりバックトラックの制御を行う為に導入された手法であるが、本システムではバックトラック制御の為ではなく、仮説候補選択の指針としてｊｕｓｔｉｆｉｃａｔｉｏｎの考え方を利用する。即ち、ｊｕｓｔｉｆｉｃａｔｉｏｎの表記を踏襲しているが、以下に示すように、“事実集合Ｆだけでは証明できなかった観測Ｏ（①式）が、無矛盾な仮説集合ｈを追加することにより観測Ｏが証明できるようになった（②式）”との旨を、③のように記録しておくというものである。

$$F \not\vdash O \quad \cdots ①$$
$$F \cup h \vdash O \quad \cdots ②$$
ただし、$F \cup h \not\vdash \square$
$$\langle O, \{\{F\}, \{h\}\}\rangle \quad \cdots ③$$

実際には本システムでは、③は

```
justi([Obs], [Hyp])        ・・・④
```

の形で保存している。ただし、Ｏｂｓは観測（①式のＯ）、Ｈｙｐは仮説集合（②式のｈ）を表す。

本稿で示すシステムでは、推論で使用した事実知識（Ｆ）も保持しておくことは、推論パス・ネットワークの構成を保持しておくことに相当する。新たな観測に対して推論パス・ネットワークを形成する時間は大きくないし、一般的にも事実知識は無矛盾チェックが不要な知識であるので、使用した事実知識の利用は高速化にはあまり寄与しない。従って、このｊｕｓｔｉｆｉｃａｔｉｏｎにおける事実知識の保持は明示的には行っていない。

### ４．２　類似事例の解仮説を利用する仮説推論の考え方

仮説推論システムで過去の推論経験で得られた仮説集合を利用して新たな観測を証明し得る仮説を探索する場合、以下のような場合に類推が有効となる。ここで、過去の仮説推論を行ったとき証明した観測を$O'$、新たな観測をＯとする。

1）$O = O'$であり、同じ仮説集合でＯが証明できる。
2）$O \approx O'$であり、同じ仮説集合、あるいはその部分集合でＯが証明できる。
3）$O \approx O'$ではあるが、仮説集合を一部を入れ換えることでＯが証明できる。
4）$O \neq O'$であるが、$O'$からＯへの写像が判っているとき、仮説集合をその写像に従って変換すればその変換した仮説集合がそのままＯの証明に使える。
5）$O \neq O'$であるが、$O'$からＯへの写像が判っているとき、仮説集合をその写像に従って変換すればその変換した仮説集合の一部を入れ換えることでＯの証明が出来る。

（1）は自明な場合であり、（2）の特殊例として扱える。

（2）と（3）は本稿で対象とする類似事例の解仮説を類推に利用することにより高速化が図れる場合である。

（4）と（5）は本稿では扱っていないが、類推により知識ベースに無い仮説を知識ベースにある仮説から発想しようというも

のであり、本システムの次期システムで目標としている。

過去において現在の観測Oと似た観測O´に対して下記の⑤のような仮説推論結果が得られたら、観測O´の解仮説の一部（$h_1$）を使い、更に⑥の形で新たな観測Oに対して不足している仮説$h_3$を探索することで推論を行う。これにより$h_1$を探索し、それらの間の無矛盾性チェックをするのに使われていた時間が削減され、仮説推論の時間が短縮される。

$$F \cup (h_1 + h_2) \vdash O' \quad \cdots ⑤$$

$$F \cup (h_1 + h_3) \vdash O \quad \cdots ⑥$$

ここで、Fは矛盾の可能性の無い知識、OとO´はそれぞれ証明すべきゴールとなる観測である。

ただし、一般には⑤で新たなゴールである観測Oにも使用できる仮説$h_1$と不要な仮説$h_2$を選択するには多大な時間を要することになる。この課題に対しては５節に述べるように推論パス・ネットワークを利用することが極めて有効な手段となる。

図５は以上の類推を利用する仮説推論の考え方を図によって表したものである。

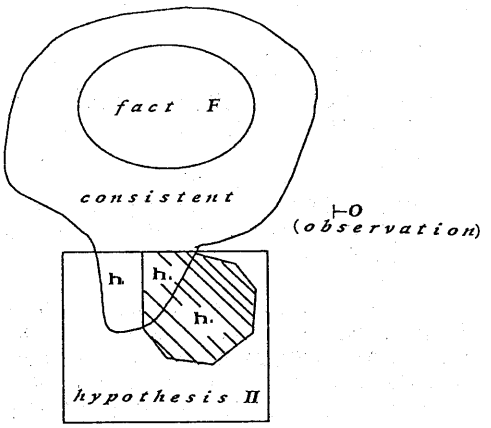


図５　類似仮説集合を用いることによる仮説推論の概念図

## ４．３　一般の類推推論との差異

類推の考え方は、例えば、『高い水圧から低い水圧の方へ水が流れる』という現象が判っているとき『熱が高い方から低い方へ流れる』という構造の似たものを、様子の判っている事例を写像によって変換して説明しようというGentner のStructure Mapping の手法[17]等、ソースとターゲットが与えられており、写像によって知識の欠けているターゲットの事象を予測しようというものが現在の主流である。しかし、この類推においては解の正当性は余程完璧な経験則がないと得られない。それに対して、Russell は類似性を推論するときに用いる経験則（上記の写像函数）を演繹的な知識として整理し、論理的演繹推論をそのまま類推に適用しようとするＤＢＡＲ（Determination-based Analogical Reasoning）[18]の考え方を提示している。この場合には得られた解の正当性の根拠はあることになるが、経験則を強いて含意知識にしてしまわなければならない点が問題となる。

以上に示したように通常の類推では過去に経験し、求解を行なった事例を活用して写像するなりして知識の拡大を行い、未経験の状況に対して知識ベースの知識の不足を補う意味が強いが、この場合には類推によって得られた解の妥当性が大きな問題となる。これに対し、本システムでの仮説推論における類推の利用は零から出発して仮説の生成（求解）を行なうのではなく、事例を出発点にして仮説生成を行なうことにより高速化を図るもので、解の妥当性は類似事例を見いだした後の論理を基盤とした推論の正当性によって保証される。更に、ＤＢＡＲで必要とされる類推を行うための強い含意知識も不要である。

## ５．　事例の解仮説を用いる推論パス・ネットワーク上の高速仮説推論システム[12]

４．２節に記した新たなゴールにも使用できる事例の解仮説$h_1$と不要な仮説$h_2$の分離、および新たな仮説$h_3$の生成が、推論パス・ネットワークを利用することにより非常に効果的に実現されることを示す。なお、推論パス・ネットワークに沿う前向き並列推論による元の仮説推論が全解探索であったのに対し、類推による仮説推論は事例に近い１個以上の解を出すものであり、単解探索に近い型に性格が変わっている。

## ５．１　推論パス・ネットワーク上のデータ構造

類推を効率的に実現するのに有効な推論パス・ネットワークのデータ構造を以下に記す。

与えられたゴール（観測）に対して推論パス・ネットワークを生成すると、事実知識や仮説知識をノードとする、下図に一部示すようなネットワークが形成される。ただし、ルール型仮説知識の変換により、仮説はリーフ・ノードにのみ位置する。このネットワークを参照するとゴールの証明に無関係な仮説は推論パス・ネットワークに含まれないことから容易に削除できる。更に、あるノードから見たネットワーク上でのノードの結合関係が判り、不足した仮説の生成や新仮説の導入の障害となる仮説削除を効率的に実現できる。

例えば図６を例にとると、以下に記す結合したノードの双方向の関係が判るデータ構造を得る。

●あるノードが前提に現れる節のうち、ゴールへと導くもの（clause_gb）

例えばＢに対しては図の
　Ａ　：－　Ｂ，Ｃ，Ｄ
の関係にある節がＢが前提に現れ、かつゴールへつながる節（clause_gb）　であると判る。

●あるノードが結論となる節（clause_c）、及びそれを支持する事実や仮説（support_h）

例えばBに対しては図の
　B　：－　E，F．．．
の関係にある節がBを結論とする節（clause_c）で、その節の前提には事実EとF等がある。
Aに対しては
　A　：－　B，C，D
の関係にある節がAを結論とする節（clause_gb）で、Aを真にするためには事実BとC、仮説D（support_h）のサポートが必要と判る。

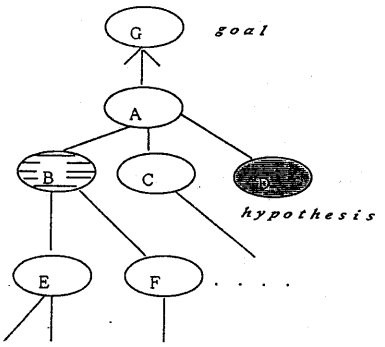


図6　あるノードに対するネットワーク
－Bを中心として－
A　：－　B，C，D　（Dが仮説）
B　：－　E，F．．．
の場合

５．２にもアルゴリズムを示すが、本システムでは先ず類似事例の仮説集合を全てこの推論パス・ネットワークのリーフに位置する仮説に当てはめる。この時点で事例仮説中で新たなゴールへと導かない節を支持するものは捨てられる。従って、４．３で述べたh₁がシステマティックに切り出されることになる。又、リーフの仮説が事例の解仮説により充当されなかったことにより必要な支持が得られず、ゴールに向かう推論が止まっているリンクが存在する場合には、そのノードに関する（clause_gb）の節を参照して、ゴール方向にに推論を進めるのに必要な推論パスから深さ１の位置にある仮説を順次選択でき、ゴール指向的に仮説をネットワークに当てはめることが出来る。

更に、事例解仮説を当てはめたことにより、それと矛盾するため新たな仮説を当てはめられなくなる場合や、ネットワーク上でゴールから見て、推論に必要な仮説より先端の位置（ネットワークの木構造ではより下の枝）に類似仮説が無いため上記の手法で仮説が求められない場合も生じる。この場合はネットワークを前後にたどることであるノードを支持するのに必要な全ての仮説候補を求め、始めの事例仮説の代わりとなり、新たな仮説と矛盾しない仮説を探索したり、新たな仮説を探索することが出来る。

以上に述べたように、このネットワークのデータ構造を利用することで事例仮説に近い解を高速に得ることが出来る。

## ５．２　アルゴリズム

以下に事例から不要な仮説を除くアルゴリズム、及びゴール（観測）証明のために必要な新仮説を探索するアルゴリズムを示す。

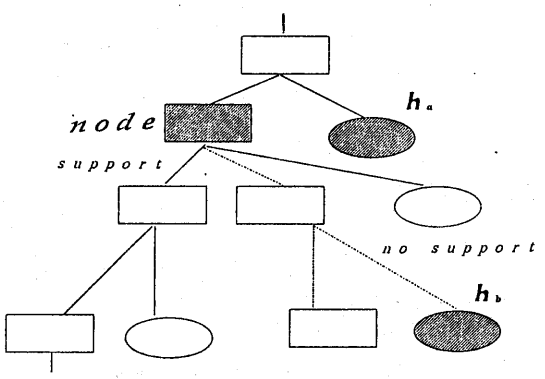


図7　あるノードに対するh。とh₁

ｐ０　【事例仮説の前処理】
事例の解仮説から現在の知識ベースに無いものを除く。残りの仮説を現在の知識ベースの順序にソートする（ネットワークへの解仮説当てはめの効率化の為）

ｐ１　事実知識（F）だけで与えられたゴールの証明が可能か調べる。この過程でリーフ位置に仮説も関係づけて推論パスを張り、コンパイルすることによりゴール指向性を有するネットワークを設定する。この過程でゴールを証明するのに不必要な仮説は削除される。しかし、ゴールを証明する経路にはあるが、今後導入することが必要な新仮説と矛盾してしまうことにより推論の障害となる可能性のある仮説の削除は出来ない。

ｐ２　ｐ１でゴールの証明ができない場合は、類似事例の解仮説をネットワークのリーフの位置に当てはめる。ゴールまで支持が到達すれば類似事例の解仮説、もしくは部分集合が解となる。以後のステップのために、それぞれのノードでゴール方向の次のノードを支持し、かつゴールの証明に必要な仮説（h。）と、あればゴールとは反対側ノードを支持する仮説（h₁）を調べて記録しておく（h。とh₁は図７参照）。類似事例仮説は重複するので記録しない。なお、この過程では、類似事例解仮説同士は無矛盾性は保証されているので、無矛盾性チェックは行わない。これだけでゴールの証明が可能な場合はほとんど時間をかけずにゴールの証明が可能である。これは４．２で示した（２）の場合である。

以下は４．２で示した（３）にあたる。

ｐ３　ｐ２でゴールの証明ができなかった場合は、ｐ２で求まった仮説h。をネットワークのリーフの位置に当てはめる。更に、この過程でも仮説h。を求める。
この過程を仮説h。が見つからなくなるか、仮説を伴ってゴールに到達する推論が得られるまで行う。

ｐ４　ｐ３までで求まらないときは事例仮説中にゴールから見て最先端のリーフ（ネットワークの木構造では一番先端の枝）にあるべき仮説が欠けているということなので、必要な支持が得られていないノードについて、支持する可能性のある仮説（これはh₁に相当する）をデータ構造を利用してネットワークの最先端のリーフまで全て求める。そして基本的には前向き並列推論による仮説の合成を行うのであるが、単解を得れば十分なので、仮説候補を順次ネットワークに加えていき、ゴールが証明さ

れるまで仮説合成を行う。

p５　ｐ４までの過程で、もし類似事例仮説と矛盾しているため、ゴールの証明に必要な新仮説が導入できない場合は、問題の類似事例仮説をキャンセルし、更に無矛盾チェックにより支持のなくなったノードからｐ４と同様にデータ構造を利用して前向き並列推論をすることで、矛盾しない新仮説を生成してゴールを証明する。

## ５．３　推論プロセス例

上記のアルゴリズムで得られる推論の様子を簡単な例で示す。

知識ベースとして以下に示すものを考える。

| Complete Knowledge (F) | Hypotheses(H) | Inconsistency Constraints |
|---|---|---|
| a ← h, f. | f | inc ← i, p. |
| b ← d, w. | i | inc ← k, y. |
| c ← s, v. | j | inc ← m, x. |
| d ← m,n. | k | inc ← i, t. |
| d ← p, n. | m | |
| e ← u, v. | p | |
| h ← i, j. | r | |
| h ← k, j. | t | |
| l ← v, t. | v | |
| n ← q, k. | w | |
| n ← q, r. | x | |
| q. | y | |
| s ← x. | z | |
| s ← y, z. | | |
| u. | | |

そして、ゴールが“$g_{c1}$ ＝ （a，d，e）”である時は解として｛f，i，j，m，r，v｝が得られ、その推論パス・ネットワークは下図に示すようになったとする。以下ではこの推論で得られた仮説を類似事例仮説として用いて仮説推論を行う様子を示す。

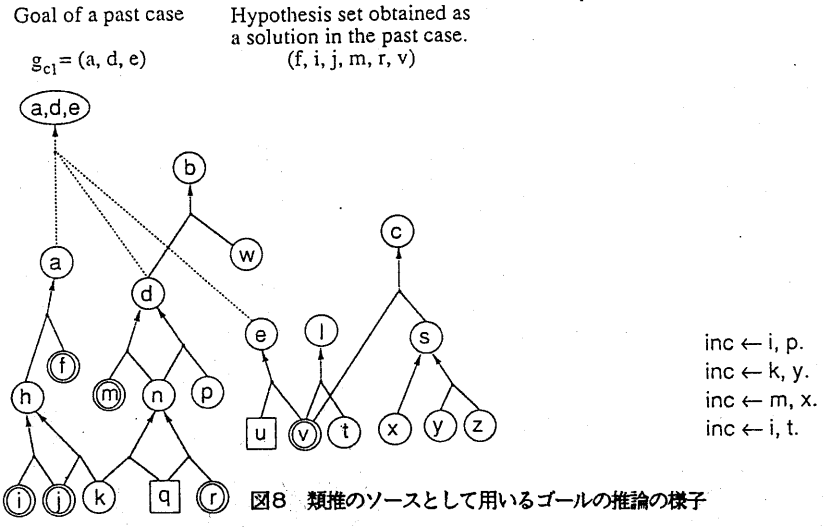


図８　類推のソースとして用いるゴールの推論の様子

Ｍｏｄｅ　Ｉ　（類似事例仮説そのもの、もしくは部分集合）

ｐ２までででゴールを証明できる例である。

新たなゴールが、“ｇ ＝ （ａ，ｅ）”、もしくは“ｇ ＝ （ａ，ｄ）”となったときに、それぞれの解仮説｛ｆ，ｉ，ｊ，ｖ｝

と｛f，i，j，m，r｝が得られる様子は図９に示す通りである。

ただし、以下に示す図９から図１２において、図中、方形ノードの知識は事実の知識、円ノードの知識は仮説の知識あるいは仮説の支持により真となる知識を表す。特に二重円ノードの知識は類似事例仮説である。方形ノードは推論パス・ネットワークのコンパイル時に消去されるのであるが、ここでは説明のために残してある。

このＭｏｄｅＩでは単に類似事例仮説をリーフの位置に置くだけで新たなゴールを証明できる最も簡単な例である。

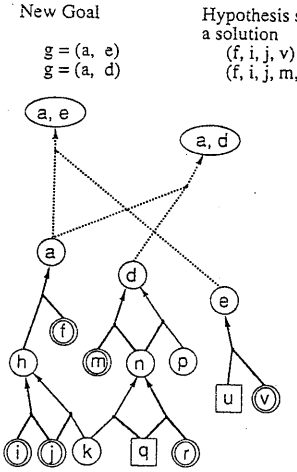


図9　Ｍｏｄｅ　Ｉ　の様子

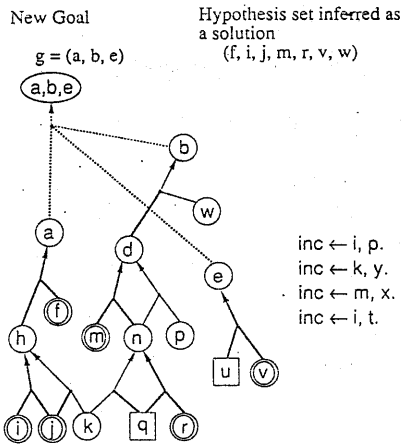


図１０　Ｍｏｄｅ　Ⅱ－１　の様子

Ｍｏｄｅ　Ⅱ－１　（類似事例仮説＋ゴール指向の推論パスから深さ１の探索のみで見つかる新仮説）

ｐ３まででゴールを証明できる例である。

新たなゴールが、“ｇ　＝　（ａ，ｂ，ｅ）”となったときに、解仮説｛ｆ，ｉ，ｊ，ｍ，ｒ，ｖ，ｗ｝が得られる様子は図１０に示す通りである。これは“ｗ”が新仮説として追加される例であるが、この“ｗ”は“ｄ”からゴール方向に辿ると５．１で言及した（clause_gb）に示すノード構造により、“ｂ”が見つかり、すぐ“ｗ”が必要な仮説であると判る。従って、このモードも深い探索を必要とせず新たなゴールに対する解が得られる。

Ｍｏｄｅ　Ⅱ－２　（類似事例仮説＋支持を必要とするノードの前向き並列推論により生成される新仮説）

ｐ４までかかってゴールを証明できる例である。

新たな観測が、“ｇ　＝　（ａ，ｃ，ｅ）”となったときに、解仮説｛ｆ，ｉ，ｊ，ｖ、ｘ｝と｛ｆ，ｉ，ｊ，ｖ，ｙ，ｚ｝が得られる様子は図１１に示す通りである。これは“ｘ”もしくは“ｙ”と“ｚ”が追加される例であるが、例えば“ｘ”の場合、“ｘ”は“ｖ”から５．１に示したデータ構造の（support_h）を探すことで見つかる。

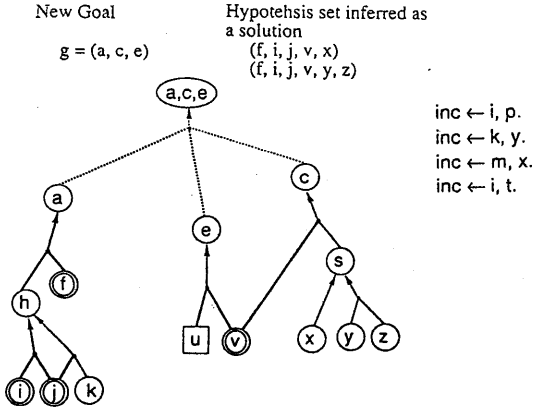


図１１　Ｍｏｄｅ　Ⅱ－２　の様子

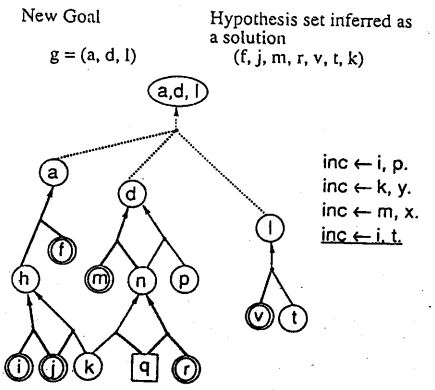


図１２　Ｍｏｄｅ　Ⅲ　の様子

Ｍｏｄｅ　Ⅲ　（類似事例仮説－新仮説と矛盾する類似事例仮説＋新仮説）

ｐ５までかかってゴールを証明できる例である。『Ｍｏｄｅ　Ⅱ－２』までては新仮説はそのまま追加可能であったが、ここでは、追加する仮説が類似事例仮説と矛盾してそのままでは解として採用できない。しかし、解として使わなくてはゴールを証明できないので、新仮説と矛盾する類似事例仮説をはずし、矛盾しない仮説を新たに生成するいった操作を行っている。

新たなゴールが、“ｇ　＝　（ａ，ｄ，ｌ）”となったときに、解仮説｛ｆ，ｊ，ｍ，ｒ，ｖ，ｔ，ｋ｝が得られる様子は図１１に示す通りである。これは“ｔ”が追加され、それに伴い、“ｔ”と矛盾する“ｉ”が除かれて代わりに“ｔ”とは矛盾しない“ｋ”が解として加わる例である。『Ｍｏｄｅ　Ⅱ－２』と同様、ゴールに対して前向き後向き双方のノードのデータ構造を参照することによって必要な解を探索している。例えば、“ｔ”は『Ｍｏｄｅ　Ⅱ－２』と同様にして求める。又、矛盾により除かれた仮説“ｉ”の代わりの仮説“ｋ”を探索に行くときは５．１で示したデータ構造の（support_h）を利用する。

以上に示した推論パス・ネットワークのデータ構造を利用するアルゴリズムで、類似仮説に近い解が、不要な事例仮説の削除と最小限の必要な新仮説の生成で、少なくとも一つ得られる。

５．４　実行例

図１３に仮説数４５個、６０個、７５個の場合について類似仮説集合を使わない場合と使った場合の推論速度の違いを示す。例としては、解の一部を除く、もしくは交換した（矛盾するものと）例と、観測の部分解を一部変えた例にある仮説推論で得られた事例仮説を適用してみたものである。なお、時間はＳＵＮ４／３３０を用いて計測されている。

図の実線で囲まれた領域は類似事例仮説集合を用いない場合の推論速度であるが、この速度は仮説数の増加とともに指数的に増加していく。

一方、類似事例仮説集合を用いた場合は、『Ｍｏｄｅ　Ｉ』の時（全く同じ推論を行わせる）は推論パス・ネットワーク形成と類似事例仮説のあてはめの時間だけであるので、ほとんど瞬時に解が得られる。『Ｍｏｄｅ　Ⅱ－１』、『Ｍｏｄｅ　Ⅱ－２』では新たに生成される仮説は一部に限られることから横軸の推論に使う可能性のある仮説数に対して指数オーダー以下の時間が達成される。

『Ｍｏｄｅ　Ⅲ』に入ると速度が低下するのは探索量が増大するのに加え、現在のインプリメントでは必要な新仮説と矛盾していて解が得られないサブゴールから探索することを行っており、まだ不十分な点があり、不必要な解まで探索していることも影響している。実際、０．３４秒かかっている例では、１３５個の仮説セットを生成している。

類推では適当な類似事例がなかった場合、あるいは一見類似しているように見えるが、新ゴールの推論パス・ネットワークにうまく適合しない事例であったりする場合には、零から仮説を生成することになり、最悪ケースで評価すれば指数オーダーの時間となる。しかしながらここでの例に見られるように利用できる事例があるときには多大な高速化が図れ、確率的あるいは平均値として指数オーダーの壁を越える高速化が達成される。

類似事例の検索時間にについてはここでは考慮しなかったが、記録する事例が増えると検索時間が増大する。しかしながら、ここでＭＯＤＥⅡ－２、あるいはＭＯＤＥⅢ、あるいは類似事例がなかった場合のみ新たな事例として追加していく方法により、無駄な事例を増やすことなく問題空間を均等にカバーする有効な事例ベースを構成できると考えている。

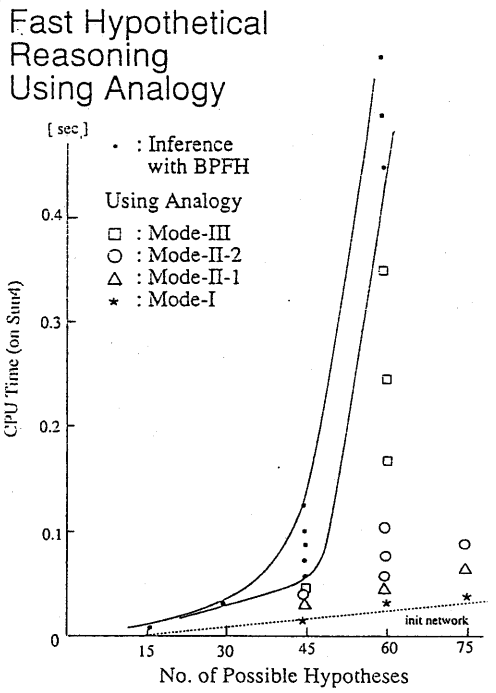


図１３　類似事例仮説を用いた場合と用いない場合の速度比

## 6. むすび

仮説推論は実用性を備えた有用な知識処理の枠組みであるが、推論速度が遅いという欠点を有しているという点に対して、過去に推論された類似事例の結果を利用して仮説推論の速度を高めることが出来ることを示した。本システムでは類似事例の解仮説のどの部分が利用できるかの選択、および必要な新たな仮説の生成を推論パス・ネットワークを利用することにより非常に効率的に行える。非単調推論系である仮説推論の計算複雑度はＮＰ－ｃｏｍｐｌｅｔｅであり、通常の推論の範囲に留まっていたのでは指数オーダーの推論時間の壁を越えられない。確率的ではあるが、この推論時間の壁を越える一つの方法として本稿に示した方法は重要である。他の高速化の方法に知識を変形してしまうコンパイル技術[19]があり、並行して研究を進めている。

ここでは変数を含まないホーン節命題論理による仮説推論システムを対象としたが、変数を含む述語論理による仮説推論システムに対しても高速推論のメカニズムを見いだしており[20]、類推の利用も可能と考えている。

また、ここで記した類推の利用の発展として４．２節の（４）、（５）に記したように類推により知識ベースにない知識の発想機能の実現を図りたいと考えている。

## ＊＊＊　参考文献　＊＊＊


[1] 石塚：“不完全な知識の操作による次世代知識ベース・システムへのアプローチ”、人工知能学会誌、ｖｏｌ．３、Ｎｏ．５、ｐｐ．５５２－５６２、１９８８

[2] D. Poole、R. Aleliunas、R. Gobel ：“Theorist:A Logical Reasoning System for Defaults and Diagnosis”、in The Knowledge Frontier:Essays in the Representation of Knowledge (N. J. Cercone, G. MaCalla Eds.)、Springer-Verlag 、１９８７

[3] 國藤：“仮説推論”、人工知能学会誌、Ｖｏｌ．２－１、ｐｐ．２２－２９

[4] 松田、石塚：“仮説推論システムの拡張知識表現と概念学習機構”、人工知能学会誌、ｖｏｌ．３、Ｎｏ．１、ｐｐ．９４－１０２、１９８８

[5] T. Matsuda, M. Ishizuka：“Knowledge Acquisition Mechanisms for a Logical Knowledge Base Including Hypotheses”、Knowledge-Based Systems 、ｖｏｌ．３、Ｎｏ．２、ｐｐ．７７－８６、１９９０

[6] 牧野、石塚：“制約評価機構付き仮説推論システムとその回路ブロック設計への応用”、人工知能学会誌、ｖｏｌ．５、Ｎｏ．５、１９９０

[7] 伊藤、石塚：“論理的制約利用による仮説推論システム”、情処学会人工知能研、７０－５、１９９０．５

[8] H. A. Kautz, B. Selman：“Hard Problems for Simple Default Logic”、Proc. of KR'89、ｐｐ．１８７－１９７、１９８９

[9] T. Bylander, D. Allenang, et al.：“Some Results Concerning the Computional Complexity of Abduction ”、Proc. of KR'89、１９８９

[10] 阿部、石塚：“仮説推論システムにおけるｊｕｓｔｉｆｉｃａｔｉｏｎによる発想の手法の提案”、電子情報通信学会全国大会、Ｄ－３７８、１９８９．３

[11] 阿部、石塚：“仮説推論システムにおける事例を利用した推論の高速化”、情報処理学会全国大会、５Ｃ－７、１９８９．１０

[12] 阿部、石塚：“推論パス・ネットワーク上での類推による高速仮説推論システム”、情報処理学会全国大会、２Ｋ－６、１９９０．９

[13] W. F. Dowling, J. H. Gallier ：“Linear-time Algolithm for Testing the Satisfability of Propositional Horn Formulae”、J. of Logic Program 、ｖｏｌ．３、ｐｐ．２６７－２８４、１９８４

[14] J. de Kleer、：“An assumption-based TMS ”、Artif. Intell.、Ｖｏｌ．２８、ｐｐ．１２７－１６２、１９８６

[15] K. J. Hammond ：“CASE-BASED PLANNING ”、Proc. of a Workshop on CBR、１９８７

[16] J. Doyle：“A Truth Maintenance System”、Artif. Intell.、ｐｐ．２３１－２７２、１９７９

[17] D. Gentner ：“Srtucture-Mapping:A Theoretical Framework for Analogy ”、COGNITIVE SCIENCE 、vol.7 、ｐｐ．１５５－１７０、１９８３

[18] T. R. Davies, S. J. Russell：“A Logical Approach to Reasoning by Analogy ”、Proc. of IJCAI、ｐｐ．２６４－２７０、１９８７

[19] 鶴田、石塚：“命題論理知識ベースのコンパイル法”、情処学会人工知能研、７０－６、１９９０．５

[20] 近藤、石塚：“述語論理知識ベースに適用できる高速仮説推論システム”、情報処理学会全国大会、２Ｋ－８、１９９０．９