



博士學位請求論文

299

多項式時間で仮説推論の準最適解を求める  
ネットワーク化バブル伝播法  
に関する研究

指導教官 石塚満 教授

東京大学大学院  
工学系研究科 電子工学専攻

大澤幸生

(27111)

# 目次

第1章 序論	1
1.1. 仮説推論の由来	1
1.2. 仮説推論の実用性	4
1.3. 非単調推論の一般的定式化としての仮説推論	7
1.4. 仮説推論の最大の課題 = 低い推論速度	10
第2章 従来の研究から数理計画法のネットワーク化へ	13
2.1. バックトラックによる計算量の爆発と、 望ましいヒューリスティックス	13
2.2. 仮説推論高速化の為の従来研究	20
2.2.1. 記号処理的なロジックのアプローチ	20
2.2.2. 数理計画法を用いた方法論	22
2.2.3. AIとORの融合	23
2.3. 次章以下の概要	25
第3章 ネットワーク化バブル伝播法（命題論理版）	28
3.1. 仮説推論から数理計画法への帰着	28
3.2. PC法による多項式時間仮説推論	30
3.2.1. 掃き出し補数（PC）法の動作原理	30
3.2.2. 掃き出し補数法とその他の探索手法の比較	34
3.3. ネットワーク化バブル伝播法の構成	37
3.3.1. バブル伝播ネットワークの構成	37
3.3.2. 4つのノード状態と「バブル」	39
3.3.3. 支配ベクトルと伝播パス	41
3.4. NBP法の動作	45

3.4.1. ターゲットの交換相手の選択	45
3.4.2. NBP法におけるPC法からの他の改良点	52
3.4.3. NBP法の動作例	57
3.5. NBP法の知識处理的な意味	60
3.5.1. 初期値計算の意義	60
3.5.2. 指数深さと、0-1整数値に落ち着く順序	62
3.5.3. 問題のネットワーク構造と取扱い易さの関係	66
3.6. 実験と評価	69
3.6.1. 実験とその結果	69
3.6.2. 推論時間と解仮説に関する考察	70
本章の付録	71
付録1：同じ枝の逆向きの影響度の積=-1の説明	71
付録2：図2-9の手続き終了時での $\text{rank}=\text{rank}'$	71

第4章 NBP法の計算量とネットワーク構造との関係	72
4.1. 数学的解析から基本的改良へ	72
4.2. 支配ベクトルとネットワーク構造の関係	74
4.3. 新しい制約不等式を用いたネットワーク化バブル伝播法 (I-NBP法)	
4.3.1. 新しい制約不等式	74
4.3.2. I-NBP法のアルゴリズム	75
4.4. I-NBP法における計算量	79
4.4.1. I-NBP法に基づく計算量クラス	79
4.4.2. 多項式時間での近似推論の十分条件	81
4.5. 従来理論との比較	82
4.6. 実験による評価	94
本章の付録 (定理2以外の定理, 補題の略証)	100

第5章 知識の実行時リフォーメーションによる高速化	107
第6章 述語論理への拡張	115
6.1. ネットワーク化の本来の意義	115
6.2. 述語論理表現の仮説推論へのアプローチ	116
6.3. 述語論理に拡張したNBP法	117
6.3.1. 述語論理に対する拡張NBP法の概要	117
6.3.2. 述語版NBP法で用いる不等式制約	117
6.3.3. 述語版NBP法のBPN	119
6.3.4. 短期記憶と長期記憶のやり取りによる述語版NBP法	123
6.3.5. $\phi$ 束縛の問題とその対策	127
6.3.6. 矛盾制約の表現方法	132
6.3.7. 述語版NBP法の表現能力と推論の健全性	141
6.3.8. 述語版NBP法の動作例	145
6.4. 例題による実験とその評価	148
6.5. 計算時間の理論的裏付けと、例による理論の説明	151
本章の付録	169
第7章. 結論	171
謝辞	174
参考文献	176
業績リスト	182
索引	188
A.用語索引	188
B.人物索引	193



# 第1章

## 序論

複雑な問題を解こうとする時、我々は試行錯誤だけで答えを得ることはしていない。目星を立てて、その後意識しない内に次第に解に近づいて行くことが多い。この論文は、計算機にその様なプロセスを実行させる試みである。知識を表現する新しいタイプのネットワークの上でそれを実現することにより、自動的に知識の構造も考慮した高速推論を行う。

本章では、この論文が高速化しようとしている仮説推論がいかなる問題であるのかを明らかにする。

### 1.1.仮説推論の由来

仮説推論とは、知っている知識からだけでは説明しきれない様な事象を説明するために必要な仮説を求める推論である。この概念は、Aristotleが示した3段論法を基にプラグマティズムの創始者Pierceの示したアブダクション（発想的推論）に由来する。Pierceは、次の豆の例を好んで用い、アブダクションとは大前提と結論から小前提を導く推論であるとした。

(大前提  $A \supset C$ ) この袋の中の豆はすべて白い。

(結論  $C$ ) これらの豆は白い。

$C \quad A \supset C$

(小前提  $A$ ) これらの豆はこの袋から取り出された

$A$

形式的には、上式の様になるので後件肯定の虚偽とよばれる妥当でない

推論となっている。これに対しPierceは、大前提と小前提から結論を導く演繹推論と大前提を小前提と結論から導く帰納推論を定義した。その上で、科学的探求の方法論としてはアブダクション→演繹→検証というサイクルが本質的なプロセスであると考えた（この枠組みは、最近の複合戦略による機械学習システムEITHER[Moo 93]にも具現化されている）。

彼の考えたアブダクションのプロセスは、仮説の提起と仮説の選択からなっている。仮説の提起は天啓によって与えられるとしたが、これはこの意味では人工知能では殆ど扱われていないプロセスであり、本論文でも扱わない。次の仮説の選択では、様々な要素が考慮される必要があるが最も単純な仮説を選択するのが自然であり、後述のコストに基づくアブダクションはその一つの具体化である。アブダクションはPierceの後、Popper, Hempelの哲学を経てPoole, Goebelらの論理的な仮説推論の枠組みに定式化されて行く。これに前後してMcDermottは演繹法則的説明が機能しない日常の人間による説明が存在することを主張した[McD 87]が、これについても本論文では捨象する。

即ち、実用的に重要な推論を機械によって実行する上で欠かせない、論理的な基盤性を有する枠組みのみ扱うこととする。Popleは、公理集合 $\Sigma$ を事実と呼び、これを含む言語のある部分集合からなる仮説 $h$ から説明するという枠組みを提案し、初めてアブダクションの機械化を行った。この様に、論理的な枠組みにおいてはPierceの仮説提起を完全に捨象することが前提となった訳であるが、その後[Poo87,88a]によって仮説集合として有限集合を与えておくという簡略化がなされて一挙に工学的な現実性を帯びたものとなった。

以下では、ここでPooleらの定式化した論理に基づく仮説推論を扱う。アブダクションの一種である論理に基づく仮説推論は、理論的な基盤性と診断や設計に応用できる実用性の両面で、知識処理の重要な枠組みで

ある[石塚91, 92]. 以下, 単に仮説推論と呼ぶのはこの枠組みとする.

はじめに仮説推論の基本動作について簡単に述べる. まず知識を, 常に成立し他と矛盾の可能性のない背景知識( $\Sigma$ )と, 常に真とは限らず他と矛盾の可能性を有する仮説知識(H)に分ける. このような知識ベースからある事象 (ゴールG) を説明, あるいは証明したいが,  $\Sigma$ からだけではゴールGを証明できないとする. このとき,  $\Sigma$ と合わせてGの証明を可能とするようなHの無矛盾な部分集合(h)を見出すのが仮説推論の基本動作である. 即ち,

$$\begin{aligned} h &\subseteq H && (h \text{ は } H \text{ の部分集合}) \\ \Sigma \cup h &\vdash G && (\Sigma \cup h \text{ からゴール } G \text{ が証明できる}) \\ \Sigma \cup h &\not\vdash \square && (\Sigma \cup h \text{ は無矛盾; } \square \text{ は空節すなわち矛盾を表す}) \end{aligned} \quad (1-1)$$

なる解仮説hを求める. 簡単に言えば, これまでの知識では説明の付かない様なこと (ゴールG) に遭遇したとき, 仮説を立ててそれを説明するのである. この仮説は真偽が未知であって, 人間であれば説明に成功した仮説を知識に付加することで科学を進歩させたりもする.

多くの場合, 他の解仮説に包摂されるような冗長な解仮説でないこと, 即ち解の極小性が要求される. これが先に述べた, 最も単純な仮説を選択することに対応する. 更に, 本論文で対象とする仮説推論の様に各要素仮説に数値的重みが付されているとき, 含まれる要素仮説の重みの和 (コスト) が最小となる解仮説を求めることが望ましい. Charniakは [Cha 90]において, この様に与えた重みの逆指数を各仮説の信頼度 (真である確率) とすれば, このコストに基づくアブダクションによって最も確率の高い解仮説が得られることを示した. 最近では, Pooleも確率的な知識を取り扱うことのできる枠組みを提案している [Poo93]が, 基本的にはこれまでの定式化の範囲で (Charniakの延長を含む) 表せる内容で, 却

って折角のシンプルな枠組みを乱す結果となったというのが筆者の私見である。

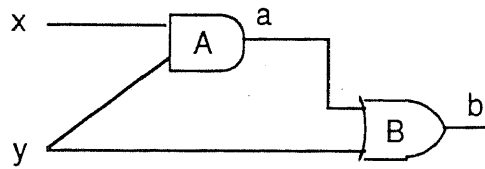
## 1.2. 仮説推論の実用性

仮説推論の具体的な適用例として、あるシステムの各部品の故障状態など可能な状態を各要素仮説とし、故障の観測事象をゴールとすれば、仮説推論により故障診断が可能となる[石塚91]。また設計に使用可能な部品や部品間の可能な接続法を仮説とし、満たすべき設計の仕様をゴールとすることによって、設計システムが可能となる[牧野90]。後者の場合、ある部品を用いるという仮説のコストにその部品の単価を与えてやれば、コストに基づくアブダクションによって最も低い予算でシステムを構築可能となる。Charniakによるコストの意味づけ[Cha90]に従えば、これは前者の場合には最も可能性の高い診断ということになる。例えば、図1.1の例は非常に単純な回路の故障診断である（読者が全ての意味をフォローできる様に簡単な例としたので、その確認により理解を深められたい）。

Charniak流のコストの意味づけとは、次の様なものである。この例で、要素仮説Xが真となる確率を $p(X)$ と表記すると、仮にAonとBoffの独立性を仮定すると、両者が同時に生起する（真となる）確率  $P = p(\text{Aon}) * p(\text{Boff})$  の対数をとれば、

$$\log P = \log p(\text{Aon}) + \log p(\text{Boff}).$$

これを最大とするには、要素仮説Xの重みを正の実数 $\text{weight}(X) = -\log p(X)$ で与え、含まれる要素仮説の重みの和が最小の解仮説を求めれば良い。即ち、各仮説の重みとして真となる確率の対数を与えたコストに基づく仮説推論により、信頼度の高い仮説を立てることが可能となる。



$\Sigma$	H
bt :- at, Bok.	Aon, weight = 3.
bt :- yt, Bok.	Aoff, weight = 5.
bf :- af, yf, Bok.	Aok, weight = 0.
bt :- Bon.	Bon, weight = 3.
bf :- Boff.	Boff, weight = 5.
af :- xf, Aok.	Bok, weight = 0.
af :- yt, Aok.	
at :- xt, yt, Aok.	
af :- Aoff.	
at :- Aon.	
inc :- Aon, Aoff.	
inc :- Bon, Boff.	
inc :- Aon, Aok.	
inc :- Bon, Bok.	
inc :- Aoff, Aok.	
inc :- Boff, Bok.	
	G
	bt :- xt, yf.
	↓
	G: g.
	( $\Sigma+$ ) g: bt.
	yf.
	xt.

$h : \{Aon\}.$

図1.1 回路の故障診断の例： x, yの入力がそれぞれ1, 0であるのにbへの出力が1であるという症状は，Aonという簡単で可能性の高い仮説で説明できる（AonはAがStack-on，即ちoffにできない故障状態にあることを意味する．Aoffはその反対で，Aokは正常）．

ここで，Aon, xtなど，知識の最小単位をアトムと呼ぶ．節  $a :- b, c.$  はホーン節と呼ばれ，"If b & c, Then a." を意味する．"a" を後件（又は結論又はヘッド）部，"b, c" を前件（又は条件部又はボディ）部と呼ぶ．ホーン

節のヘッド部は単一のアトムからなる.

上の例ではゴールが,  $a :- b, c.$  というホーン節の形で与えられている.  
ホーン節知識の範囲内でこれを最も一般化すると, 下の様に節の連言で与えられるゴールが考えられる.

$$X_1 :- Y_{11}, Y_{12}, \dots Y_{1n_1}.$$

$$X_2 :- Y_{21}, Y_{22}, \dots Y_{2n_2}.$$

.

.

$$X_m :- Y_{m1}, Y_{m2}, \dots Y_{mn_m}.$$

このような場合でも, 次の様に書き換えることによってゴールをアトムとして与えることができるようになる.

$$Y_{11}, Y_{12}, \dots Y_{1n_1},$$

$$Y_{21}, Y_{22}, \dots Y_{2n_2},$$

.

.

$$Y_{m1}, Y_{m2}, \dots Y_{mn_m},$$

$$G :- X_1, X_2, \dots X_m.$$

又, 仮説が節形式で与えられた場合も,

$$(\text{仮説 } a = "X_1 :- Y_1, Y_2, \dots Y_n.") \leftrightarrow (X_1 :- a, Y_1, Y_2, \dots Y_n.)$$

となり仮説はアトム $a$ として与えられる書き換えが可能となる. 従って,  
以下では仮説とゴールは共にアトム (ファクト型, 即ち節形式でない)  
で与えることを前提とする. 又, この例で見られるようにホーン節 " $a :- b,$

c." は "If b & c, Then a." という意味を定式化したものであり、実用面でも十分豊かな表現方法であるので、この論文では扱う知識をホーン節に限る。

モデルに基づく診断としては、[deK 91] に示された様に、確率の高い部分に焦点を当てる focusing を行いながら検査点での検査を含めて推論を進めるのも一つの考え方であろう。このようなアプローチは、解を出力するまでに複数回観測を行う点で本論文で述べる仮説推論と異なる枠組みであるが、高速に可能性の高い解仮説を求めることが可能となる。即ち、推論パスネットワークやSLD導出木と同様な知識ネットワークにおいて、ゴールを起点として全体を検査回数に当たる部分木あるいは部分問題に分割して解くことになるので、非常に高速な推論方式となる。しかし反面、この考え方が通用するのは自由に指定した部品の検査が可能である様な場合に限られる。例えば、医療診断を行う場合などを考えると、病気とは無関係かもしれない臓器や神経系に検査器を差し込むことが許されないわけであるから、[deK 91] のアイデアは限られた応用例についてのみ著しい高速化を達成するものであると言える。

これと本論文で取り扱う論理に基づく仮説推論を比較すると、仮説推論の立場がより明瞭となる。即ち、診断の場合には、症状が観測された後は推論システムは一切外界との接触をせずに推論だけを行う。当然、途中で検査が可能な場合に比べて推論速度は低くなるが、この方が一般的に用いることのできるシステムであると言える。

### 1.3.非単調推論の一般的定式化としての仮説推論

非単調推論とは、新しい知識が加えられるとその前までの知識が否定されてしまう可能性を有する推論のことである。例えば、次の様に演繹推論によって導かれる結論が事実（新しい知識）と異なるという場合がそれにあたる。

(前提1) 鳥は飛ぶ

(前提2) 鶏は鳥である

(結論) 鶏は飛ぶ

この例で、鳥は飛ぶというのは実はそれまでの乏しい経験から暫定的に得た仮説である。この様に仮説推論は、後件肯定の虚偽から非単調性を備えている。非単調な推論は、単調な推論に比べると我々が日常で経験する推論に近いと言える。例えば、深酒が続いたある日急に右横腹から中央にかけて激痛が生じ、これは肝硬変だと早とちりしていると、暫くして用を足すと治まったので単なる腹痛であったことに気づく、という様な過ちの多くが非単調推論によるものである。非単調推論系には仮説推論の他にも、Levesqueの知識レベルのアブダクション[Lev 89]や様相論理の様に信念を扱う推論、Reiterのデフォルト理論や極小限定(Circumscription)などがある。しかしそれらの殆どは仮説推論の枠組みで記述できる。例としてここではこのうち、知識レベルのアブダクションとデフォルト理論が仮説推論で記述できることを示す。

### 知識レベルのアブダクション

問題固有の公理はそのまま、信念に対する解釈の仕方を変えることによって異なるタイプの信念を扱う。今、ある認識状態(背景)  $E$  においてタイプ  $t$  の信念  $b$  がある時  $E \rightarrow B_t(b)$  と書く。そして、 $b$  が  $f$  を説明することを

$$E \rightarrow B_t(b \supset f), \neg B_t(\neg b). \quad (1-2)$$

で定義する。分かりやすく言えば、 $b$  の否定が信じられていないときはその帰結を信じることを説明するのである。この時信念の世界での背景知



識 $\Sigma$ に対して

$$E \rightarrow B_t(b) \leftrightarrow \Sigma \vdash b. \quad (1-3)$$

と書き換えると, (1-2)は

$$\Sigma \vdash b \supset f, \Sigma \not\vdash \neg b. \quad (1-4)$$

となる. 仮説推論では否定を明に表せないので, 後半を $\Sigma \cup b \not\vdash \square$ とすることによって式(1-1)と一致する.

### デフォルト論理

デフォルト論理[Rei 80] も意味は知識レベルのアブダクションと良く似ており, "特に論理的矛盾がないならば, 暗黙に... を認める" ということを記述する論理体系である. これにはいくつかのパターンがあるが, 一般には次の式で表される.

$$\frac{\alpha : \beta_1, \beta_2, \dots, \beta_n}{\gamma} \quad (1-5)$$

この記述で,  $\alpha$  (前提) が成立し $\beta_1, \beta_2, \dots, \beta_n$  が互いに矛盾しなければ,  $\gamma$  が成立することを表す. この枠組みで新しく導ける知識を得て行くことを拡張と呼んでいるが, 一般には拡張を求めるのは非単調である.

(1-5)

式(1-5)で $\alpha = \text{true}$ のとき, これを無前提と呼び,  $m=1$ で $\beta_1 = \gamma$ のとき正規デフォルトと呼ぶ. 正規デフォルト理論は, 正規デフォルトのみからなる理論をいう. デフォルト理論にも仮説推論を用いることが可能であるということは, 次の定理による.

[無前提正規デフォルト理論に関する定理[Poo 88]]

Gが仮説知識H,背景知識Σから導出可能ならば, 次の $D_H$ と公理集合Σからなるデフォルト理論( $D_H, \Sigma$ )の拡張にGを含むものが存在する.

$$D_H = \left\{ \frac{\beta}{\beta} \mid \beta \in H \right\}. \quad (1-6)$$

即ち, アブダクションによる説明は可能な信念に対応しており, トップダウンデフォルト証明[Rei 80] によってアブダクションが可能となる. 同時に, アブダクションも又デフォルト論理に用いることができる[Poo 88]. 無前提正規デフォルト理論の拡張を求める問題がNP完全であるという, 計算量の観点からも両者の等価性が示されている[Sti 90].

#### 1.4. 仮説推論の最大の課題 = 低い推論速度

ここまで, 仮説推論が実用性と論理的な基盤性を備えた枠組みであることを示した. しかし, 仮説推論の最大の課題は低い推論速度である[石塚94]. 即ち, 他と矛盾し否定される可能性を有する (defeasibleな) 知識を仮説として扱う仮説推論は非単調推論の一種であり, 最悪の場合, 知識の規模に対して指数的に推論時間が増大する. このことが致命的な欠点であることは, 少しの応用例を想定してみれば明らかである.

例えば, 計算機のメモリ装置が大容量化し, データの通信容量が拡大するにつれて扱える知識の量は爆発的に増大しつつある. 研究段階まで目を向けると, その傾向はさらにジャンプして行くことになりそうである (筆者自身, 一本の光ファイバーだけでTbit/sオーダの伝送容量を実現するのも夢ではないという結論を得ている[Ohs92]). 通信容量が上がるとなれば, 例えばリアルタイム地震学において地震そのもののデータ以

外にも地熱や海底の生態系など細かいデータが全国から一瞬で集まることになる。しかし、その時折角のデータをどのように用いるのかが問題である。この例で見れば現時点では観測データの一点一点から地震予知委員会が地下の現象に関する確率的な判断を行うが、一般に豊富なデータを利用してどの仮説が正しいかを調べるには機械推論が必要となる。

また、これらの複雑なシステム自身を診断又は設計することを考えても、その部品数は多く、即ち仮説推論で行うには非常に多くの仮説数に対しても実用的な時間で計算する技術が必要となる。LSIの中でどのゲートがStack-Off状態にあるか、ということ調べるには全てのゲートの取りうるあらゆる状態の組み合わせを仮定して、観測されている故障状態と比較すれば良いことはわかる。しかし、これはゲート数に対して指数オーダーの計算時間を有するので紙と鉛筆だけでは扱いきれない。

実際には、機械の計算パワーを用いてもこの様な問題はゲート数がそのパワーを上回って多くなるとこの様な枚挙法では歯が立たなくなる。並列計算にこの課題を委ねるのは、ある面では解決の先送りとする意見もある。即ち、人間→機械という手段の移動を逐次計算→並列計算と置き換えただけであるから、ゲート数がもっと多くなると同様の問題が残ったままとなる。計算時間が1000倍となっても、(ゲート数 $\times$ 1ゲートに可能な状態数)が10増えればそれらの真か偽かで場合分けして出力をチェックする計算量が1024倍となるので追い付かれてしまうからである。この場合分けが先の迷路でバックトラックしたことに当たる。

従って本論文では、計算機のハードウェアのパワーはこれ以上向上しないとしても速度が大きく短縮できる様な、ソフトウェア面からのアプローチを採る。即ち、厳密に解くと指数オーダーかかってしまう問題を近似解法によって知識の規模(要素仮説数)に対して多項式時間で解き、解が存在すれば最適解をかならず求めるという制約を緩めて、解が存在する殆ど全ての場合に準最適解(最適解に近い解)を求める様な手法を

提案する。尚、本論文の執筆にあたってはこの章を含め、冗長な表現や可読性を妨げる他研究の過剰な紹介記述は行わない様に心掛けた。従来研究の原論文を読めば理解可能な部分は大枠のみ簡潔に示し、その他の全ての記述に筆者独自の考えが反映されている。従って、読者にとって周知を理由に読み飛ばせる箇所は存在しないことをここに記しておくたい。

## 第2章

### 従来の研究から

### 数理計画法のネットワーク化へ

本章では、第1章末尾に記した仮説推論の課題の起源を示し、身近な例でこの同じ原因がいかなる問題をもたらしているかを述べる。より形式的な理論の構築は[Byl 91, Kau 90]に詳しく述べられているのでそれらを参照されたい。ただし、それらの従来理論は本論文の第4章で行う理論解析に比較すると一般性ではない。次に、この章の後半でこれまでこの問題にどの様に対処されてきたかを示し、本研究の立場を相対的に明らかにすることによりその意義を主張する。

#### 2.1. バックトラックによる計算量の爆発と、望ましいヒューリスティックス

図2.1の迷路を考えてみる。入り口から進んで地点Aに達すると、右に行くか左に進むか決めなければならない。もし左に進めばかなり遠回りをして地点Bに達する。右に進んでいれば地点Bへはずっと簡単に到達する。この迷路ではこの地点Bは必ず通らなければならない。地点Bからは方向1と2を採ることが可能である。もし地点Cがこの図の様に塞がっていないければ、方向2を選択すればいとも簡単にOUTから迷路を抜け出すことができる。しかしこの場合の解は地点Bで方向1を採り、点線の経路を進むこととなる。同様の進路選択は地点Dでも必要となる。

もし進路を誤れば、引き返さなければならない。例えば地点Bにおいて、全く闇雲に方向2を選択するならば地点Cから地点Bに立ち戻り、改めて方

向1を選択仕直さなければならない。これをバックトラックと呼ぶ。我々

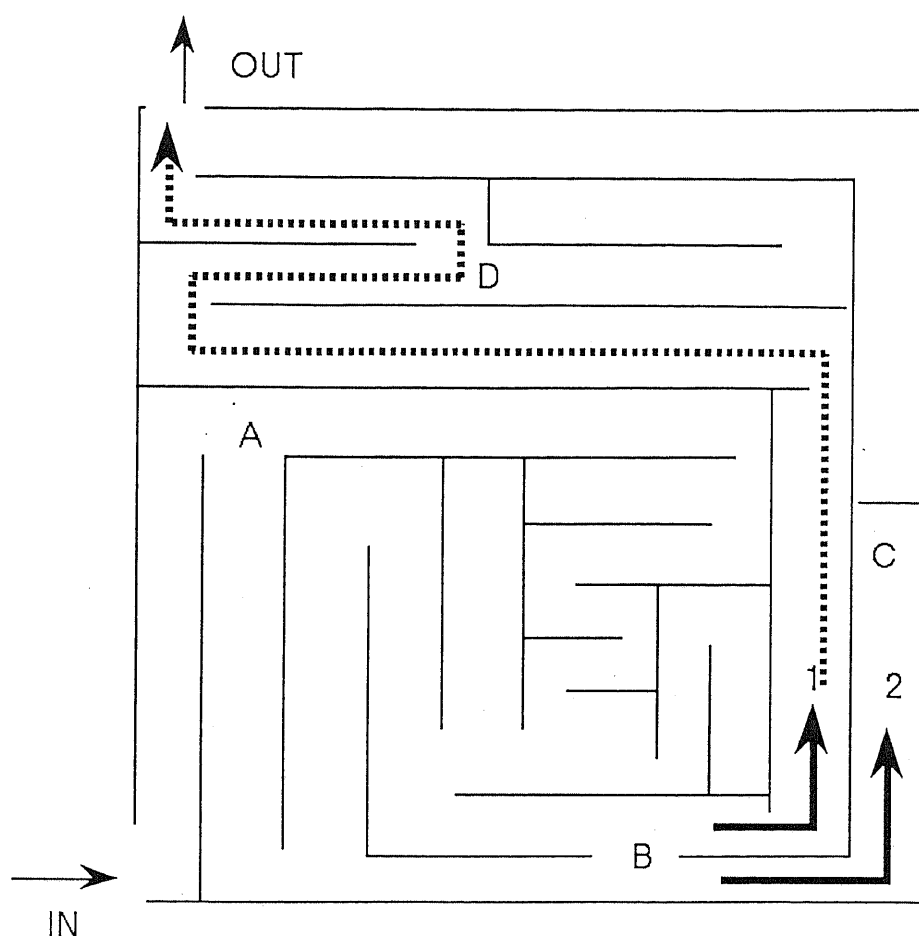


図2.1. 迷路におけるバックトラック.

人間の日常ではこのバックトラックが頻繁に繰り返されている。例えば、デパートで小さな子供が親を見失った場合、あちらこちらの売り場を見入ってどこにもいないと思い込んで泣き出すことがある。

しかし、役に立つ探索の指針が得られる場合にはこの様な闇雲な探索をしなくなる。迷路の例でもし、ゴール（OUT）に強い香水がおいてあ

るならば、壁がないところは臭いも通過して来る訳であるからその臭いの強い方向を選んで行けば頻繁なバックトラックの必要はなくなる。

この様な指針はしかし、絶対的な指標ではない場合が多い。この例でも、臭いが混ざり合うことが障害となり、多少のバックトラックで時間をロスすることも考えられる。この様に確実ではないが行動の指標として大抵の場合に役立つ指針をヒューリスティックスという。ヒューリスティックスは、必ずしも有限時間内に解が得られるとは限らないという特徴を有する。これに対し、道路標識の様に進む方向が明示してあれば、それが正しい限りこの指標が必ず役に立つ。この様に必ず有限時間内に解が得られる絶対的な手順をアルゴリズムと呼ぶ。

ヒューリスティックスとアルゴリズムはこの様に比較すると対照的であるが、より根源的な差異はそれぞれの獲得され方である。アルゴリズムは役立つことが保証されているわけであるから、問題が与えられてから対処を始めるまでの間には獲得されている必要がある。図2.1の点線が始めに示されていればアルゴリズムが示されたことになる。これに対してヒューリスティックスが獲得されるには様々なケースがある。一つは経験によるヒューリスティックスの獲得である。先のデパートの例で、大人が迷子になることは滅多にない。私の妻が私を見失ったならば、当然の様にトイレの前で私が出てくるまで待つに違いない。大抵の場合、あるいは必ずそこにいるからである。もう一つは目の前に置かれた状況をみて比較的解に至る可能性の高い選択を繰り返して行く方法である。これは経験に基づくとは限らない。迷路で臭いのする方向に進むというのは、どの様な人生を送ってきた人にとっても信頼に値する一般的な指針である。

機械に問題解決をさせる場合、その問題がルーチン化された工程を経て配線の固定された回路を作成せよ、というのであれば予めアルゴリズムを与えておけばよい。しかし、病気を診断せよという問題であれば症

状から推察しておそらくこれだ、という病名を出力しなければならない。その際は何らかのチャートに従うのではなく、症例などを参考にして推論を行う必要がある。一般に、多くの可能性から一つの解を選ぶ様な場合にはアルゴリズムだけではなくヒューリスティックスが必要となるのである。

ところが、機械による問題解決では全てのヒューリスティックスを経験から得ることは現実的ではない。人間の場合は非常に多くの感覚器官からの情報を合成し、非常に多くの経験例と高速に比較することが可能である。しかし、機械が同程度の素子数を持てば高い故障率とエラーを繰り返すであろう。素子数と故障率は比例するからである。もし素子数をクリアできたとしても、未だにその機能の殆ど不明となっている脳を含む人体を模倣することは非常に難しい。又、機械に人間並み知識を与える段階では知識獲得のボトルネックという問題が生じる。人間自身がどうやって知識を得たか自覚していない場合すらあるのであるから、機械に人間が知識を与えるのが困難であるのは当然といえよう。特に扱う知識が多くなればなるほどこの問題は深刻となる。

知識の完全な自動獲得というのも非常に困難である。筆者自身、未知語を含む自然言語文の解析[大沢90]を行ったが、未知語の属性を細かく推定して確信を持ってそれを辞書に付け加えるというシステムとはしなかった。文脈を部分的に知ってそれに基づいて文章の解析を進め、残った文字列が未知語と判定されるが、人間が監視していなければ知識は少しずつ間違いを重ね、辞書は使い物にならなくなったしまうからである。

ただし、バックトラックなしに自然言語文を解析するメカニズムとしては有効性が示せたので次に簡単に紹介しておこう。

この研究においては、既知の単語を予めよく使用する単語と余り頻繁に用いない単語に分け、この境目も順序も動的に変化させながら文章を解析するシステムを開発した。



まず、予め各単語の使用頻度（一般的な頻度と分野ごとの頻度に分けた）を木構造辞書の葉に与え、辞書を使用頻度毎にクラス分けしておく。始めは高い使用頻度のクラスだけ用いて島駆動方式で文章を読み、それではパーキングできないところは次のクラスを加えて少し拡大した辞書を使う。そして、辞書中の単語の分野毎の使用頻度を、既に読んだ単語がどの分野でどの程度頻繁に用いられるかを表す数値を掛け合わせたものに変更し、一文毎に新しく辞書のクラス分けを行う。こうしてできた新しいクラス分けを用いるようにすれば、次第に辞書そのものが読んでいる文章用にカスタマイズされて精度と速度が上がって行く。島駆動で、辞書にのっていない単語は結局読み飛ばすので、それを未知語として推定される品詞とともに登録する。この方式は、文章が未知語を含んでいても読める上に、速度の向上が可能となる。即ち、本来、左から順に文を読む場合は、ここで切れるのだ、と仮説を立てながら読んで、もし理解不能なときは途中まで戻る必要が生ずるが、この手法ではそういうバックトラックの必要がない。

この手法は、眼球のサッケード運動が情報処理機構として有効に働いていることをモデル化したものといえる。次の文を見てみる。

ほにゃらけんなんぶにはつよいあめがふっています

数十ミリ秒の提示から次第に短時間の提示をを追加して行くと、始めは「つよいあめ」位しか目に入らないが、次第に全体を理解して行く。これは、一つのバックトラックの解消法である。即ち、下の切れ目(A)~(F)は、23箇所の字間から選択されたものであるので $2^{23}$ 通りの中から選ばれた切り方である。これを全て我々が試しているとは考えられない。

ほにゃらら(A)      けん(B)      なんぶ(C)      には(D)      つよい(E)  
あめ(D)      が(E)      ふって(F)      います

かといって、左から最長一致法で単語を抽出しては、始めに未知語があるこの様な例は取り扱えない。又、未知語が含まれていなくても、

ねえ、ちゃんと、ふろは、いった？

ねえちゃんと、ふろ、はいった？

の様に、文脈によってある切りだし方が誤りとなることもある。形態素解析における先の筆者の戦略は、後述の仮説推論で言えばどの仮説を優先するか予め目星を付けておくことに当たる。又、辞書内での優先順位が動的に変化する点はヒューリスティックスの自動獲得を行っていることに当たる。

しかし、このアプローチは十分一般的とはいえない。というのは、ここでは優先度以外には統語的な規則だけで次に拾う形態素（自然言語の最小構成要素）を制限しているが、実際には自然言語では意味的な制約が存在するからである。上の「ねえ、ちゃんと..」の場合はその典型例であって、前後に対話者の体臭について記述があれば前の切りだし方を取らなければならない。これをこのシステムで実現するには、風呂をテーマとする世界や風俗をテーマとする世界を一つ一つ別々に計算機に記憶させる必要がある（風俗の世界なら、「ねえちゃん」の優先度は高いなどとして）。これが可能であるためには、知識獲得のボトルネックが解消可能であるという前提が必要となるのである。一般的なバックトラック解消の手段は、もっと論理的に明確で一般的な定式化がなされた well defined な問題に対してのみ得られるものであろう。

以上の様に、機械が探索や診断、設計と行った複雑な問題を扱う際の

ヒューリスティックスを与えるためには、機械に理解可能な一般的な言語で記述された現時点の状況から行動への写像を予め定義しておくことが必要となる。これがアルゴリズムと異なるのは、ある状況に対する行動は経験と無関係に一意に決定できるがその行動が解を求める上で真に有効かどうかは確信できない点である。この妥協は、厳密に解を求めると組み合わせ爆発を起こすような手に負えない(intractable)問題を高速で解くという目的にとっては不可欠なものとなる。

本論文では、診断や設計の問題を一般的な形で記述した仮説推論の問題を解く高速推論のヒューリスティックスを提案する。論理に基づく仮説推論の定式化から従来の研究と、それらに対する筆者の立場については第2章の全体と各章の必要な箇所の詳細を述べるが、このヒューリスティックスは人工知能と数理計画法の橋渡しとしての新しい意義を有する。

即ち、数学的な解析によって確立したゆるぎない基盤の上で高速探索手法を練り上げてきた数理計画法は、探索空間の中で解に至るまでの道のりを短縮する強力な枠組みである。これに対して、知識処理特に記号表現された論理的な枠組みから一步も出ずに物事を説明するしようという立場では、記号操作という全く離散的な世界に限定されてしまう。しかし、後者では推論の根拠や筋道を明示的に表せるので人間が後で見て理解できるという長所がある。Newellの知識レベルでは記号レベルと知識レベルが分離されたが、知識を記号化する方針が適切である限り記号論理の枠組みは知識を伝播するメディアとして非常に有効である。

離散的な情報のもう一つの意義は、直接に行動の指標となる点である。例えば、降水確率というデータをテレビが放送し始めたのは、行動の責任を視聴者に着せただけのことである。降水確率50%という場合に傘を持って出るといえるかどうかは、視聴者の方で決めなければならない。これまでは予報者の方で「雨でしょう」といえば迷うことなく傘を持って出ていた

のであるが、降水確率が示されるようになってようやく、連続的なデータを離散化（カテゴライズ）する必要と難しさを知らされたのである。

こういう訳であるから、理解可能な記号処理の世界と効率的な数値計算の橋渡しを達成し、両者の長所を取り入れることは意義の深いことである。本論文の全文が狙うのは、その具体的な実現方針である数理計画法の知識ネットワーク化という基本理念の開拓である。次の節では、従来の研究を振り返ってこの考えの相対的な位置づけを行う。その後の各章と節においても随時関連研究との比較を示すが、それによって、本論文の新手法が、計算時間の短縮と共に従来の手法と異なるアプローチから包括的な枠組みを構築することが理解されるであろう。

## 2.2. 仮説推論高速化の為の従来研究

### 2.2.1. 記号処理的なロジックのアプローチ

仮説推論の低い推論時間という課題を克服するための研究は、ロジック・プログラミングの初期まで遡る。当時における高速な推論手法としては、SL導出[Kow 71]が提案されていた。これは、Robinsonの融合原理[Rob 63]を適用する際にアトムに融合の優先順位を示す関数を与えて反駁を導くものである。ホーン節に限定すれば、このような順序融合は完全であることから、KowalskyらとColmerauerはほぼ同時期にホーン節に問題を限定する方針に固まった。RousselがSL導出システムを構築した経緯については[Kow 88]にも述べられており、この70年代の動きがその後Prologへと繋がったと言える。この時期は論理に基づく仮説推論の創成期であって、記号処理という直接的な知識表現に基づくアプローチが固まり、以後の研究の歴史を決定づけたのも当然と見られる。

記号処理という枠内での研究は、バックトラックなどによる再計算の回避と計算量の大きな仮説合成の演算の最小化にあてられている。代表的

な一つは、バックトラックなどによる再計算の回避と計算量の大きな仮説合成の演算を最小化することによる高速化を図った推論パスネットワーク法である[伊藤91]。これは命題論理表現の仮説推論を対象とするもので、ゴールの証明に関係する知識を抽出し、バックトラックのないボトムアップな解仮説の合成を並列的に行なう。これはネットワークの無駄を除去して推論しやすい形に変形するものであるから、処理前のネットワークを用いた場合に比べて1000倍程度の速度を達成している。

[伊藤91]の様なゴール指向型の考え方は、変数を含む述語論理表現の仮説推論の効率化にも有効である。具体的には演繹データベース分野の手法であるマジックセット法やQSQR法[Vie 87,88]に準拠する述語論理表現の高速推論法が開発されている[井上93,近藤93]。

[井上93]で重要なもう一つの点は、並列推論マシンを用いてモデルを並列に保持し、ボトムアップな処理を行っていることであり、ATMS(Assumption-based Truth Maintenance System [deK 86])が命題論理で用いた基本的な考え方を継承している。しかし、並列推論の効果ではCPUの台数効果をフルに発揮してもユニット数に比例してしか推論速度は向上しない。逐次型のソフトウェア的な工夫としてはQSQR法を取り入れた[近藤93]の仮説推論手法が第6章で述べる筆者の手法と比較しやすいのでその箇所で紹介する。この手法では、推論速度のオーダーそのものを向上させるわけではないが、指数の底を2から約1.3に低減する。

[阿部92]では、類推を用いた仮説推論の高速化が研究されている。この手法では、過去の仮説推論の解 $h$ を今回使用できる部分 $h_1$ と使用しない部分 $h_2$ とに分割する。この $h_1$ として、[伊藤91]の方式により過去と現在の例のゴールに共通のサブゴールからなるネットワークを取り出し、この中に含まれる要素仮説を用いる。そして、共通でない(未経験の)部分について通常どおり推論を行うのである。この推論法は、一度行った推論については繰り返し同じ動作をしない点で効率化が得られるが、 $h_1$ と今回

使用する仮説との間に矛盾が生じるところに非単調性が現われる。又、当然未経験の問題は指数オーダーの推論時間が必要となる。しかしこれは最悪のケースであって、経験が豊かになって使用可能な $h_1$ が拡大すると共に多項式時間の仮説推論も可能となる。

経験を利用するアプローチには[牧野93]もある。これは説明に基づく学習(Explanation-Based Learning: EBL)を援用する。通常のEBLでは過去に証明された最終ゴールに対して解仮説を保存するが、ここでは中間ノードについても学習の対象とした。しかし、全ての中間ノードで学習するのは効率的でないので矛盾に関わる仮説の組の祖先ノードである中間ノードのみを学習の対象とする工夫が成されている。この方法をよりシステムチックに一般化したのが知識コンパイル[鶴田91,92]であり(発表されたのはむしろこちらが古いが)、それを効率化したのが知識の部分コンパイル[堂前94]であるが、これらについては第5章で述べる。

### 2.2.2. 数理計画法を用いた方法論 ～ 連続値を経て離散解へ ～

以上はいずれも、離散的な記号処理によって推論を行う手法である。

しかし、他の有効なアプローチとして、最適解計算の仮説推論をこれと等価な0-1整数計画問題に帰着して解を求める方法が注目されつつある。両者は共にNP完全あるいはNP困難[Byl 88, Kau 90]であるので、厳密な解法で最適解計算を行なう場合には問題の規模に対して指数オーダーの推論時間を必要とする。この問題を克服するため、数理計画法の分野では0-1整数計画問題の高速解法が盛んに研究されてきた。中でも掃き出し補数法(Pivot and Complement法[Bal 80])、以下PC法と略記)は有効な近似解法であり、0-1整数計画問題の準最適解を $N^4$ のオーダーの推論時間で求められる。この近似解法を用いることにより、同様の多項式オーダーの推論時間で準最適解計算の仮説推論を実行する手法も開発された[岡本93]。こ

の近似解法によって得られる解は最適解に非常に近く，実用上極めて有用な手法である．

本論文の研究はこのPC法による仮説推論を知識ネットワークの上で動作させることにより高速化を図るものであるから，その詳細については第3章に譲ることとし，ここではさらに筆者のモチベーションに至る大筋を述べる．

### 2.2.3. AIとORの融合 ～ 数理計画法を知識ネットワーク上で～

[岡本93]の有効性はPC法を用いることを決めた時点で予想されていた．知識処理の問題を変換して数理計画法の領域での解法を得たものであり，元のPC法はBalasの論文によって十分効果を見い出されていたからである．しかし，動作中は数字の羅列ばかりで推論中のプロセスが元の知識をどう反映しているのか把握しにくかった．これは，与えられた背景知識を制約と見なす際にそれを不等式の集合，即ち制約充足する多面体として扱うからである．この多面体にはもはや，元の知識の構造の名残を見いだすことも難しい．もしさらに知識が付け加わった場合には，新しい多面体を切り出し直さなければならず，それまでの推論のプロセスも結果も参考することができなくなる．

では，知識の構造を見やすい形でPC法的高速メカニズムを実現できるとすれば，それはどのような形であろうか．構造とは部分と部分との関係の集合であるから，知識を構成する各部分は陽な形で異なる位置に配置され，しかも関係が知識に示された部分間にはリンクを与えなければならない．その様な性質を持つ知識表現を知識ネットワークと呼ぶ．例えば，[伊藤91]の推論パスネットワークやSLD導出木は知識ネットワークである．以下では特に，図2.2に示す様な知識ネットワークのことを命題知識ネットワークと呼ぶことにする．

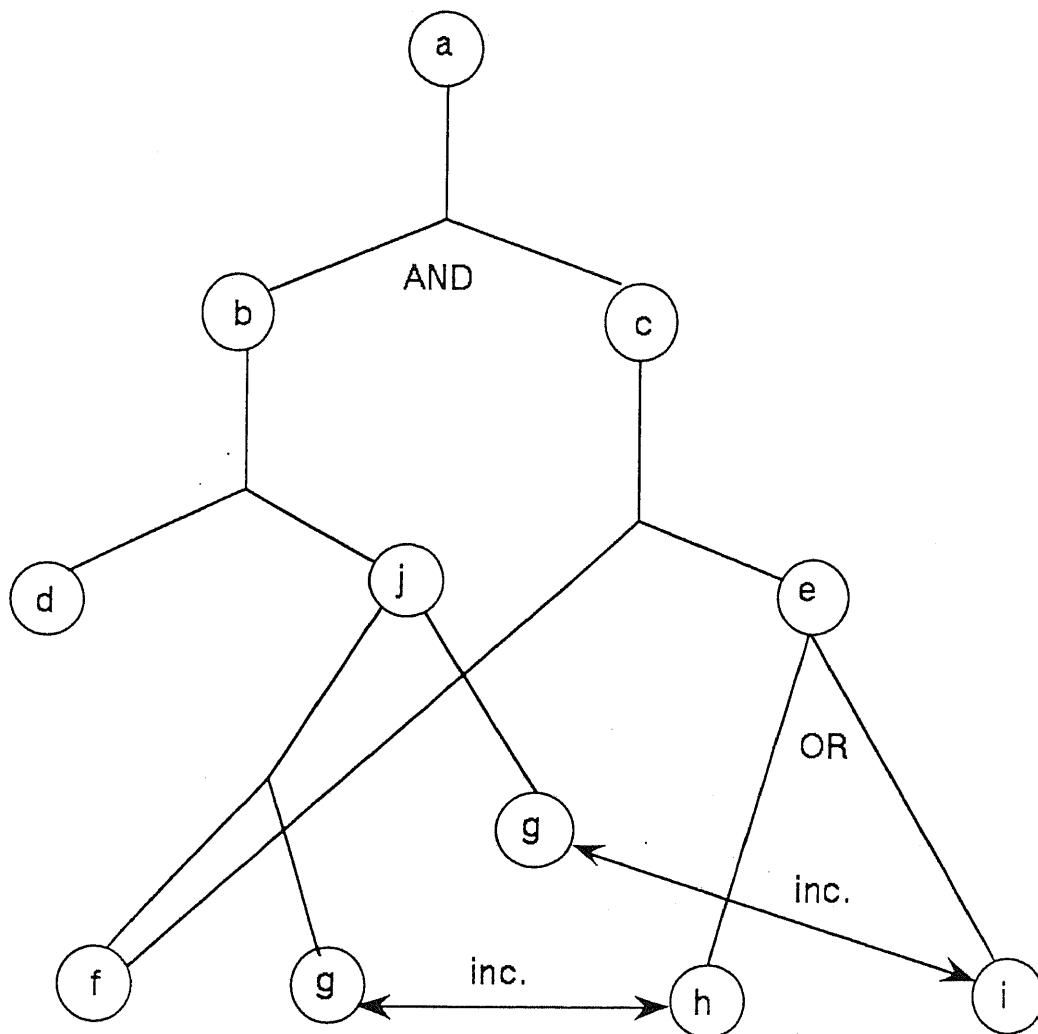


図2.2.命題知識ネットワークの例

即ち，プロセスを知識ネットワーク上で実現できれば，先ず第1に知識処理の観点から理解し易いものとなり，第2に知識構造をネットワークを媒介として利用できるのも更なる高速化も可能となる[Iri 81]．切除平面法を知識ネットワークの上で実行した[San 93]はその簡単な例である．彼は知識ネットワークのある分岐点以下の部分問題を0-1整数計画法と見做し，



整数制約を取り除いた線形計画問題に対して単体法によって得た最小コストの解をしきい値とする高速な分枝限定法を示した。

第1のメリットは本論文の第6章で表現の枠組みを拡げた場合への拡張に、第2のメリットは次章以降の随所で実現を示されるであろう。

### 2.3. 次章以下の概要

本論文の全体を見渡せるように、以下の章の概要を目次とは異なる副題と共に示す。3章ではネットワーク化バブル伝播法の原理を述べたので、その後のどの章を読む場合にも準備として理解する必要がある。4章で構築する新理論は、5章において一層の高速化が可能となる理由となっている。又、6章が本論文のクライマックスであると筆者は考えている。

#### 第3章：数理計画法の知識ネットワーク化の具体的手法とその原理

本研究では、[岡本93]と同等のプロセスを、知識ネットワーク上で更に効率化するネットワーク化バブル伝播(NBP)法と名付けた推論手法を提案、開発した。このネットワークでは、ノードの意味や配置はこれまで用いられてきた知識ネットワークとほぼ同じである。これらのノード間で $[0, 1]$ 間の実数真理値の変化を伝播させることによって、PC法と同等の操作を実現する。得られる解仮説（要素仮説の集合）は[岡本93]と同様に最適解に十分近い準最適解となる上、知識の構造的関係（幾何学的情報）の利用により[岡本93]を上回る約 $N^2$ オーダーの推論時間を達成した[大沢94a]。

#### 第4章：知識構造と計算量の関係をバブル伝播ネットワークを用いて整理

仮説推論の推論速度は、矛盾を含む場合などネットワークが複雑な場合に非常に遅くなる。では、どのような構造の背景知識までで厳密に多項式時間推論が可能であろうか？この問題に対する新しい定式化も本手法

の枠内で行うことができる。それによれば、単結合の知識ネットワークを含むあるクラスの問題では非常に高速に厳密解が得られる。又、推論とともに問題の複雑さが解消されて行く様子も定式化できる。これを利用してネットワーク化バブル伝播法の推論速度を更に高めることも可能である。ただし、この改良手法ではネットワークが元の知識ネットワークよりも複雑となり、メモリを3倍近く必要とする[大沢95]。

#### 第5章：新しい方法論”知識の実時間リフォーメーション”を提案

さて、ネットワーク化バブル伝播法は近似解法であって必ずしも解が得られるとは限らない。実際、問題が複雑（大規模とは違う）になると解を得ない場合が増える。そこで、論理回路設計に用いられる論理多段化の技術を取り入れた。この手法は、元の問題を論理式として因数分解し、複雑に絡まった知識ネットワークをある程度ほどいてからネットワーク化バブル伝播法を適用しようというものである。結果は期待通りであった。改良型のネットワーク化バブル伝播法では解が得られなかった問題を解くことが可能となり、速度も単独のネットワーク化バブル伝播法を1桁以上上回った[高間94]。

#### 第6章：AI-ORの橋渡しとしては初めて命題論理から述語論理へ拡張

数理計画法を知識ネットワーク上で走らせるメリットは、この様に狭い枠組みの推論を高速化するだけではない。知識ネットワークの良さは、焦点を知識の一部に絞ったり拡大したりできる点にある。

述語論理の仮説推論では特に、注目する知識の部位を動的に変化させるQSQアプローチなどの手法が有効である[Vie 87,88, Has93]が、PC法の制約充足領域の多面体としてこれを実現するのは事実上不可能に見える。そこで、述語ホーン節を全く新しい不等式制約に書き換えることによっ

て述語版ネットワーク化バブル伝播法ができないか考えてみた。エルブラン空間にまで落とせば可能だが、それではそもそも何のための述語論理か不明になってしまう。

しかし、述語論理から不等式制約に置き換えることは難しい。そこで助けになったのは[Has 93]の考え方であった。述語ホーン節を変数間の関係に対する制約と見なすのである。真理値伝播から制約活性値伝播への発想の転換によって、道は一気に開けた。エルブラン空間に展開せずに、表現の意味を不等式に書き換える方法を編み出したのである。

とはいえ、述語論理において本質的な再帰構造という問題をクリアしなければならない。考案した述語版のバブル伝播ネットワークは、あまりに複雑で再帰構造をある深さまで先に書いてしまうのではグラフを記述するだけでメモリがとんでしまった。[Vie 88]はこの問題への光を投げ込んでくれた。ゴールからサブゴールを必要なだけトップダウンに生成していき、仮説の情報からのボトムアップな処理も加えれば必ず再帰構造は停止する。この考え方を長期記憶と短期記憶という概念に置き換えることにより、NBP法と有機的に組み合わせることが可能となった[大沢 94b]。AIとORの融合法に対する一つの回答であると考えている。

# 第3章

## ネットワーク化バブル伝播法

### (命題論理版)

#### 3.1. 仮説推論から数理計画法への帰着

この章では先ず、命題論理ホーン節表現の仮説推論を扱う。これは6章に述べる述語表現よりは同じ記述長での表現力に乏しいが、述語論理をエルブラン空間（具体例の代入された無変数記述の空間）に展開すれば命題論理に帰着可能である点では一般的な論理体系であるといえる。

これを先に扱う理由は、仮説推論から数理計画法に帰着しやすいからである。本研究以前からも、論理式制約の充足可能性や命題論理の仮説推論は等価な0-1整数計画問題に帰着可能であることは知られていた[Hoo88, Dha90, 岡本93]。しかし、本研究の主眼は知識ネットワーク上での動作を解してAIとORの橋渡しを目指すことにあるので、ネットワーク化し易いような帰着方法から出発する必要がある。この節では、その様な帰着方法を述べる。以下では各要素仮説は重みを持つアトムとして与え、仮説に含まれる要素仮説の重みの和を評価関数（コスト）とし、この値が最小の最適な解仮説を求めるコストに基づく仮説推論を対象とする。

まず、命題ホーン節集合で与えられた背景知識を不等式集合に変換する。以下で、小文字( $x_1, y$  など) はそれぞれ大文字 ( $X_1, Y$  など) のアトムの真理値（真を1, 偽を0とする）を表すとする。式(3-1)のホーン節は不等式(3-2)と等価になる。即ち、真理値が0または1という二値条件の下では、式(3-1)と式(3-2)は全く同じ解を持つ。

$$Y \leftarrow X_1 \wedge X_2 \wedge \dots \wedge X_{n-1} \wedge X_n, \quad (3-1)$$

$$\frac{x_1 + x_2 + \dots + x_{n-1} + x_n - n + 1}{n} \leq y \leq \frac{x_1 + x_2 + \dots + x_{n-1} + x_n}{n} \quad (3-2)$$

同じ意味で、同一のヘッドを持つ下記(3-3)のようなルールの集合を論理和の形でまとめた式(3-4)は不等式(3-5)と等価になる。

$$Y \leftarrow X_1, Y \leftarrow X_2, \dots, Y \leftarrow X_n. \quad (3-3)$$

$$Y \leftarrow X_1 \vee X_2 \vee \dots \vee X_{n-1} \vee X_n. \quad (3-4)$$

$$\frac{x_1 + x_2 + \dots + x_{n-1} + x_n}{n} \leq y \leq \frac{x_1 + x_2 + \dots + x_{n-1} + x_n + n - 1}{n}. \quad (3-5)$$

従って、ホーン節集合で記述された背景知識と合わせてゴールを証明可能とする仮説集合を求めるという仮説推論の問題は、以上の不等式制約を充足する0-1整数解  $(x_1, x_2, \dots, x_n)$  で、要素仮説の真理値を表す  $x_i$  が1に等しいものの集合を解仮説とすることに対応する。即ち、証明するゴール  $G$  は " $g=1$ " と真理値の固定された変数と見なし、仮説間の矛盾は式(3-2)の  $y$  を0に固定することによって表現が可能である。

本論文では、解仮説の良さを表す評価関数またはコストを式(3-6)で表される重みの和として定義し、これを最小とする解仮説を求める仮説推論を扱う。

$$\text{cost} = \sum_{i=1}^n w_i x_i \quad (3-6)$$

ここで  $w_i$  は各要素仮説  $X_i$  に付された数値的重みで、例えば、全ての  $i$  について  $w_i = 1$  とすることによって最少の要素仮説からなる解仮説を得ることができる。この様に、コストに基づく命題論理仮説推論を0-1整数計画問題に帰着することができる。

## 3.2. PC法による多項式時間仮説推論

### 3.2.1. 掃き出し補数 (PC) 法の動作原理

掃き出し補数 (PC) 法[Bal80]は、0-1整数計画問題の高速近似解法として提案された手法である。得られる解は必ずしも最適解ではないが、多項式時間で準最適解を求めることができ、この準最適解は実用上十分最適解に近い。そのメカニズムについて簡単に説明する。

図3.1はPC法の手続きを示している。非基底変数、基底変数、構造変数、スラック変数などの用語は単体法におけるものと同定義とする[茨木93]。又、制約違反度（ある時点での探索点の座標が不等式制約を違反する度合い）、非整数度（実数真理値が0-1整数値と離れている大きさで、全ての真理値が0.5なら最大となる）、真理値固定（あるアトムの真理値についてはステップ11以降で補数を取る必要がないことを、ステップ10で判定する）に関する形式的な定義は、[Bal 80]を参照されたい。

初期フェーズで、与えられた問題から0-1整数制約を外した線形計画問題(LP)の実数最適解を求め、続く探索フェーズでその近傍の0-1整数解を得て、改良フェーズで評価関数をさらに最適値に近付ける。良く知られているようにLPは多項式時間で解けるので、その後の近傍探索に細かい工夫がされている。ここでStep1の0-1整数条件を外した制約とそれ以降の制約（充足条件）は同意味で、構造変数の値が範囲 $[0,1]$ に含まれるという制約を含む不等式全体の集合を指す。

図3.2の様な、与えられた制約を満たす変数値の範囲を表す多面体（以下、単に多面体と呼ぶ）を見ながらPC法の動作を概説する。まず、与えられた制約の中で、多項式時間で頂点1の実数最適解を得る（初期フェーズ）。

### <初期フェーズ>

1. 与えられた制約から0-1の整数条件をはずし単体法によって線形計画問題を解く.

### <探索フェーズ>

2. 制約充足条件を満たしたまま実行可能な [基底構造変数と非基底スラック変数]の交換をさがす. この交換が存在する場合, 評価関数の最良のものを実行して4へ.  
それ以外は3へ.
3. 制約充足条件を満たしたまま実行可能な [基底構造変数と非基底構造変数]または[基底スラック変数と非基底スラック変数]の交換で, 非整数度指数の減少するものをさがす. 最初に見つかった交換を実行して4へ. それ以外は5へ.
4. 0-1整数解を得た場合, 10へ. それ以外は2へ.
5. 丸め (四捨五入) により1整数解が得られる場合, これを得て10へ. それ以外は6へ.
6. [基底構造変数と非基底スラック変数]の交換のうち, 制約違反度の最小となるものを実行する.
7. 1つの変数の補数をとることによって制約違反度が低くなる場合, 最もこれを低くするものを実行して8へ. それ以外は9へ.
8. 制約非充足ならば7へ. 制約充足条件を満たし, 四捨五入して0-1整数解が得られればこれを得て10へ. 得られなければ2へ.
9. 2つの変数の補数を取ることによって制約違反度が低くなる場合, 最もこれを低くするものを実行して8へ. それ以外は探索失敗として終了.

### <改良フェーズ>

10. 現在の0-1整数値で固定すべき変数を固定する.
11. 1つの変数の補数をとることによって評価関数を改良できる場合, これを行ない10へ. それ以外は12.
12. 2つの変数の補数をとることによって評価関数を改良できる場合, これを行ない10へ. それ以外は13.
13. 3つの変数の補数をとることによって評価関数を改良できる場合, これを行ない10へ. それ以外は探索成功として終了.

図3.1. 掃き出し補数法の手続き

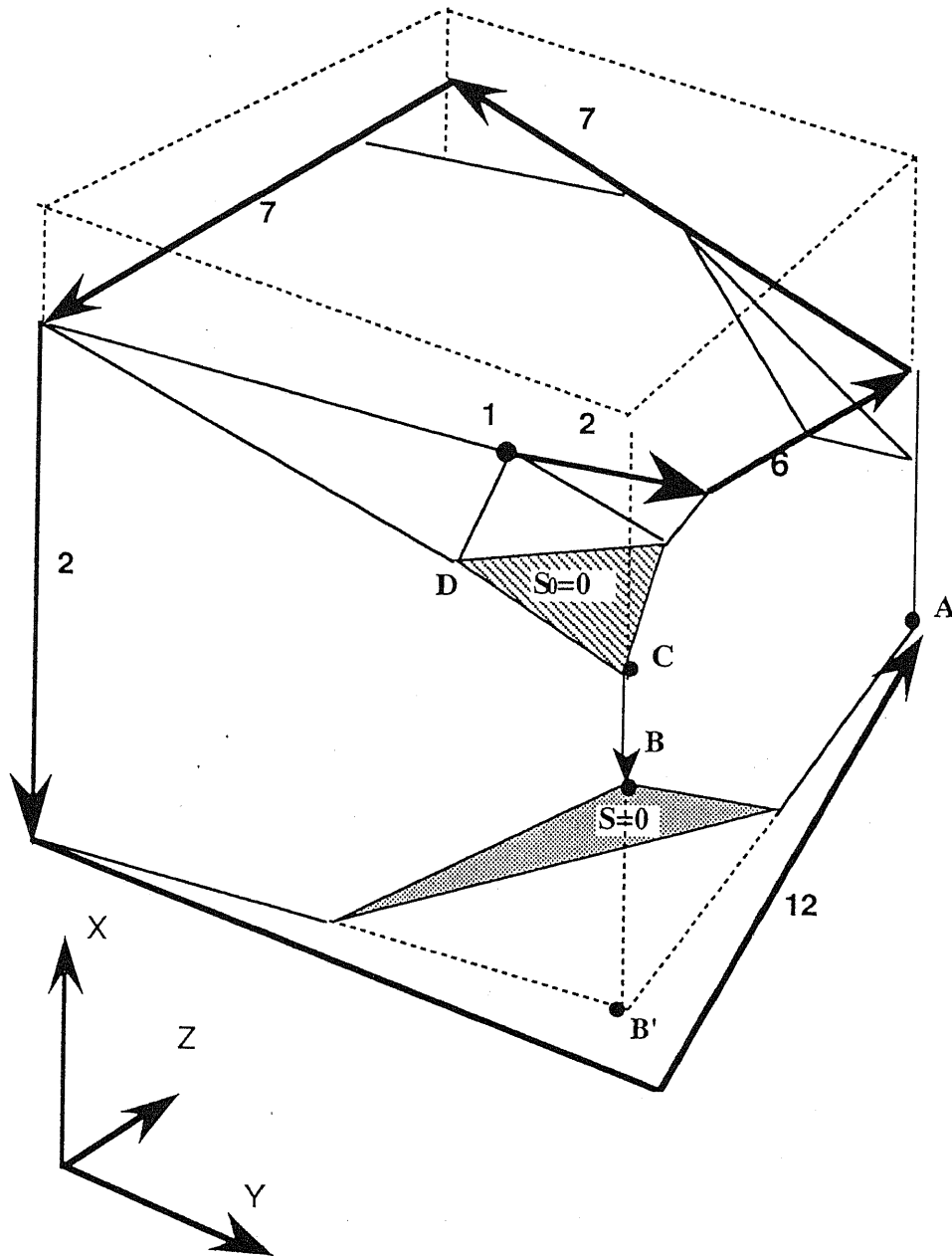


図3.2. 掃き出し補数法の探索手順（番号は図3.1におけるステップ, 太線は解Aに辿り着くまでの探索経路を表す）



ここでは単体法を用いると[Bal 80]に記されているのは、当時この計算を最悪でも多項式時間で行うことの可能なKarmarkerの手法[Kar 82]が編み出されていなかった事情に起因するとも考えられる。しかし、単体法は単純明解なアルゴリズムに加えて、極めて作為的な問題を除けば実際のステップ数が制約数の線形オーダーで収まるという実用向きの探索法であるので、現在でも適切な選択とすることができよう。

次に多面体の頂点から頂点へ縁沿いにトレースし、近傍の格子点（全ての構造変数が0-1整数値となる頂点）を探索する。この頂点から頂点への移動は掃き出し操作で実現されている。図3.1の手順全体で掃き出し操作が占める計算時間は特に大である。

仮に頂点C→頂点Bの移動を考えてこの掃き出し操作を説明する。今、多面体の頂点C(0.3, 1, 0)を探索点が-Xの方向に通過しようとしている。この時、Cを含む3平面（斜線の面とY=1, Z=0の3平面）にそれぞれ対応する3変数が移動前の非基底変数である。即ち、仮に斜線の面を次の式で表すと、非基底変数はS0とY'=1-YとZである。

$$S0 = X + 2Y - 2Z - 2.3 = 0 \quad (3-7)$$

探索点が矢印の向きに頂点Cを離れると、変数S0は非基底から基底変数となる。そして、Y'=1-YとZを非基底に保ちながら、最初に出会う頂点B(0.2, 1, 0)に移動する。さらに進むと多面体の外に出てしまうからである。即ち、移動後の非基底変数Sは網点面

$$S = 2X - Y + Z + 0.6 = 0 \quad (3-8)$$

で表される。B'(0, 1, 0)まで進んでいたらS=-1となり、この面による式(3-9)の制約を違反してしまう。

$$2X - Y + Z \geq -0.6 \quad \text{即ち, } S \geq 0. \quad (3-9)$$

この時、非基底変数 $S_0$ の値は探索点が多面体の内側の向きに変化し、 $S_0$ は $S_0$ の値の変化量を最小とする基底変数と交換される。我々は、この点に注目して知識ネットワーク上でこのプロセスを実現する。

### 3.2.2.掃き出し補数法とその他の探索手法の比較

擬似焼きなまし法[Kir 83]や遺伝的アルゴリズム[Gol 89]の様に、評価関数が非線形の場合にも用いることの出来る0-1整数計画問題の解法に確率的戦略を用いることは、最近のブームと言っても良からう。しかし、これらは非線形関数に特有の局所解の問題を回避するための手法であって、その分、線形評価関数を最適化しようとする場合の探索速度は掃き出し補数法程高速ではない。0-1整数計画問題と同じくNP完全な最大クリーク問題の解を確率論的 (Stochastic) な戦略によって近似的に多項式時間オーダーで求める手法も示されている[富田88] (なぜか読売新聞の一面トップで報道された) が、信頼に足る理論的根拠は示されていない。

最近注目されている方法論としては局所探索法(local search)を上げることができる。これはある限定された範囲で小刻みに探索点を移動するものの[Min 92]で、[Glo 86a,b]のタブー探索もその一つである。これは一度通過した探索点をタブーリストとして保持し、繰り返しトレースしない様にする事によって効率化するという考え方である。容易に理解されるように、この手法は離散的な解空間を辿って厳密解を求めようとするものであり、指数オーダーの壁を超えることはできない。離散的な局所探索の範囲を限定することによって効率的な探索を行っている点では、GSAT(Greedy Local Search Algorithm for SAT Problem)[Sel 92, 93a, 93b]も同様である。しかし、GSATは基本的に探索アルゴリズムのループ回数に上限を設けて高速化を狙っているので、完全性 (存在する解を得る能力)

を妥協したことになる。従って、この点ではGSATは実際にはPC法と同様に近似解法と解釈することもできる。本来厳密に解くと計算時間の組み合わせ爆発が起きるのであるから、この様に近似解を求める考え方は、探索の高速化にとって非常に本質的であるといえる。ただしGSATの場合、初期点をいくつかランダムに生成するのであり、PC法の様に的確な目星を単体法によって指定する探索法の方が速度と解の最適性の両面で勝ると考えられる。

その他の数理計画法の援用としては、[Gu94]において充足可能性 (Satisfiability, SA) 問題を制約なし非線形最適化問題に書き直した解法が提案されている。これは、その非常に平易な書き換え規則と、単なる山下り法からなる単純なアルゴリズムは明解さの点で非常に魅力的な定式化である（非線形計画法の他の技術も利用可能であることは言うまでもない）。しかし、非線形計画問題はNP困難な一般的問題であり仮説推論よりも難しい問題に定式化したことになる可能性を有する点で、この手法単独の特性（速度と精度）には期待できないのではないかというのが筆者の予想である。一つの有効な改良方法はPC法と同様に適切な初期点を指定することであろう[二田95]（[Kam 92]ではPC法と同様に初期点を指定した後、内点法によって格子点を探索する）が、PC法の様に複雑な手法を編み出した研究者達はその前にこれほど単純な試行を怠ったとすれば、それは非常に不思議である。初期点設定の知識处理的意味については、後ほど改めて考察を行う。

推論速度とトレードオフ関係にあるのは、近似解法は一般に完全ではないことである。即ち、解があっても必ずしもそれを見つけることができない。しかし、解が存在する問題に対してPC法が解を得る可能性は98%以上であることが[Bal 80], [岡本93] のデータから計算でき、その他にも[野中94]では、得られる解のコストは最適値から平均でわずかに0.4%劣るだけという報告もなされている。従って、完全ではないものの実用的

には十分完全に近い探索手法であると評価できる。

尚、PC法以外の手法の中にもBalasらのグループは数々のヒューリスティックを発表しており、中でもクリティカルパス法を有効に用いた Shifting Bottleneck [Ada 88] 法は行商人(Travelling Salesman, TS) 問題の構造を活かした高速解法として高い評価を受けている。しかし、既に特定の問題を志向して作られた手法を知識ネットワークという新たな形式に置き換えることは、極めて困難な仕事である。筆者はこれを非常に長期的な課題と睨み、本論文では扱わないことにする。

### 3.3. ネットワーク化バブル伝播法の構成

PC法と同等の機能をネットワーク上のプロセスとして実現するのがネットワーク化バブル伝播法 (Networked Bubble Propagation method, 以下NBP法) である。NBP法は、掃き出し操作のネットワーク化を軸とした推論法である。PC法の計算時間の殆どはこの掃き出し操作にかかっていた。

#### 3.3.1. バブル伝播ネットワークの構成

まず式(3-2)又は(3-5)の形の不等式が一つ与えられた場合、これをそれぞれ不等式(3-10)、(3-11)に変形する。ここでスラック変数 $s_{NBP}$ は式(3-12)で定義する。スラック変数は上限値、下限値によって制約される（下限値0だけで制約される通常の定義と異なる）。

$$-\frac{n-1}{n} \leq s_{NBP} \leq 0. \quad (3-10)$$

$$0 \leq s_{NBP} \leq \frac{n-1}{n}. \quad (3-11)$$

$$s_{NBP} = y - \frac{x_1 + x_2 + \dots + x_{n-1} + x_n}{n}. \quad (3-12)$$

式(3-12)を図3.3のようにネットワーク化する．スラックノード $S_{NBP}$ ，構造ノード $X_1, Y$ などはそれぞれ，スラック変数 $s_{NBP}$ ，構造変数 $x_1, y$ などの値をもつ．

グラフ中の枝はスラックノードと構造ノードの間を接続し，枝には伝播率を与える．隣り合うスラックノード $A$ と構造ノード $B$ を結ぶ枝の $B \rightarrow A$ の向きの伝播率  $p(A,B)$  とは，ノード $B$ の単位変化が枝 $BA$ を経て $A$ に伝播する量をいう．即ち，式(3-12)から

$$p(S_{NBP}, X_i) = -1/n \ (i=1,2,\dots,n), \quad p(S_{NBP}, Y) = 1$$

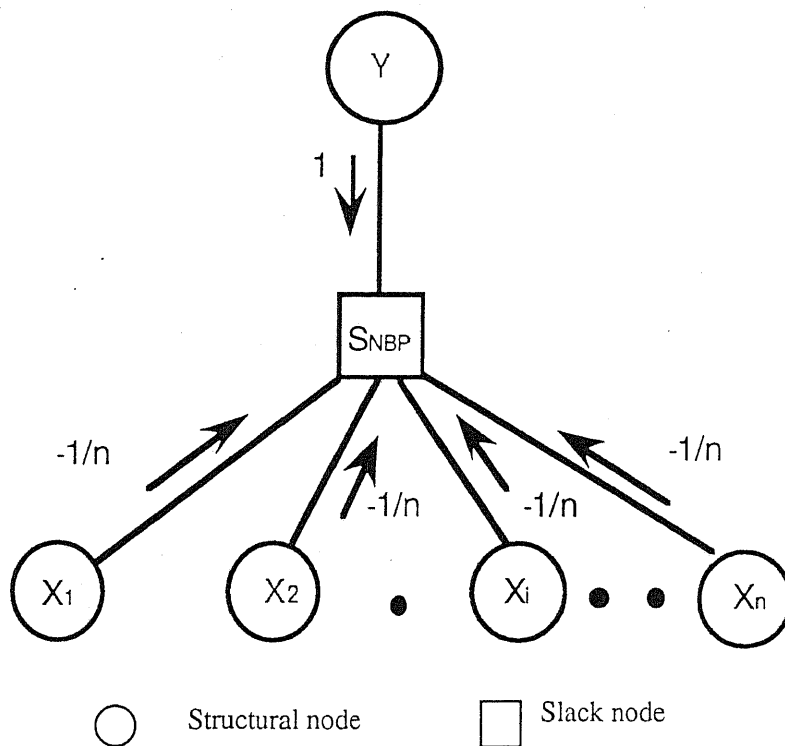


図3.3. ホーン節と等価な単一の制約不等式からバブル伝播ネットワークへの変換

となる．また，同じ枝の反対向きの伝播率 $p(B, A)$ を $-1/p(A, B)$ と定義するが，その理由は付録1に記す．伝播率を用いると，式(3-12)は式(3-13)のように表せる．

$$S_{NBP} = p(S_{NBP}, Y) y + \sum_{i=1}^n p(S_{NBP}, X_i) x_i, \quad (3-13)$$

where  $p(S_{NBP}, Y) = 1, p(S_{NBP}, X_i) = -1/n \ (i=1, 2, \dots, n)$ .

伝播率の利用法は後述する．不等式集合に変換された背景知識を，このようなサブグラフ（一つが式(3-1)，(3-3)の1ルールを表す）を接続して構成したものをバブル伝播ネットワーク（以下単にBPN）と呼ぶ．例えば，図3.4の仮説推論の推論木は図3.5のBPNとなる．

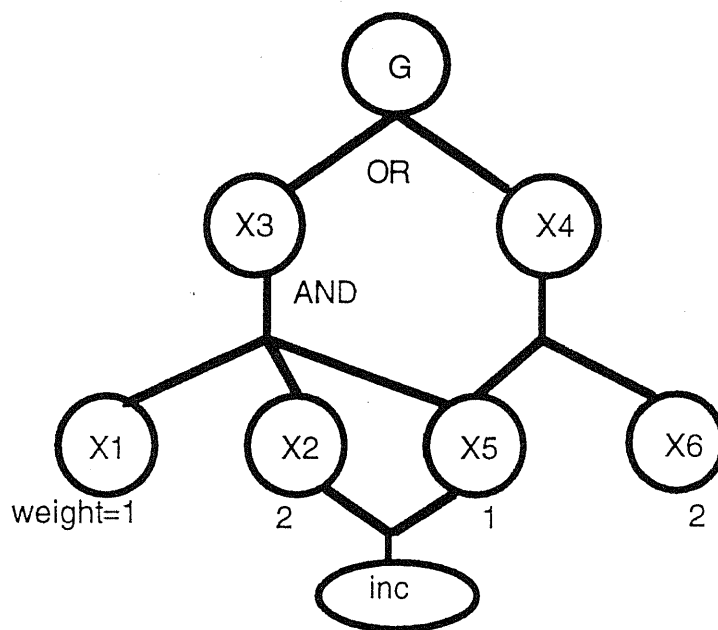


図3.4. 解仮説が複数存在する簡単な問題の例

### 3.3.2. 4つのノード状態と「バブル」

図3.5にも用いているように、ノードの状態として次の4通り（2クラス×2状態）があり、推論はノードの色の伝播によって進行する。

- 非基底構造ノード
- 基底構造ノード
- 非基底スラックノード
- 基底スラックノード

ここで、●、○、■及び□を以上の4種類の状態を区別するために用いる。非基底状態と基底状態は従来の非基底変数、基底変数に対応する。このようにノードの非基底状態を表す黒い色のことをバブルと呼ぶことにする。

単体法によって実数最適解と同時にこれを支配する非基底変数の集合が決まる[茨木93]。このことから、初期フェーズまではPC法と全く同様に単体法を実行すると、NBP法でもこの時点で各ノードの初期状態が決まる。PC法の探索フェーズがすべての構造変数が非基底、即ち0-1整数値に落ち着いた状態で終了するのと同様に、NBP法では●と□のみから構成されるネットワーク状態を探索フェーズの最終状態とする。初期状態から最終状態に至る遷移がバブルの伝播で実現されるのである。

バブルの個数はこの間一定に保たれる。図3.2に戻ってこのことを説明する。真理値空間内で移動している探索点はN次元の真理値空間では多面体のN面の交点となっている筈である。仮に探索点が頂点Dの様に3次元空間で4平面の交点となっても、探索点はそれらのうち任意の3平面即ち任意の3非基底変数で決定される。この点は単体法でも同様である。

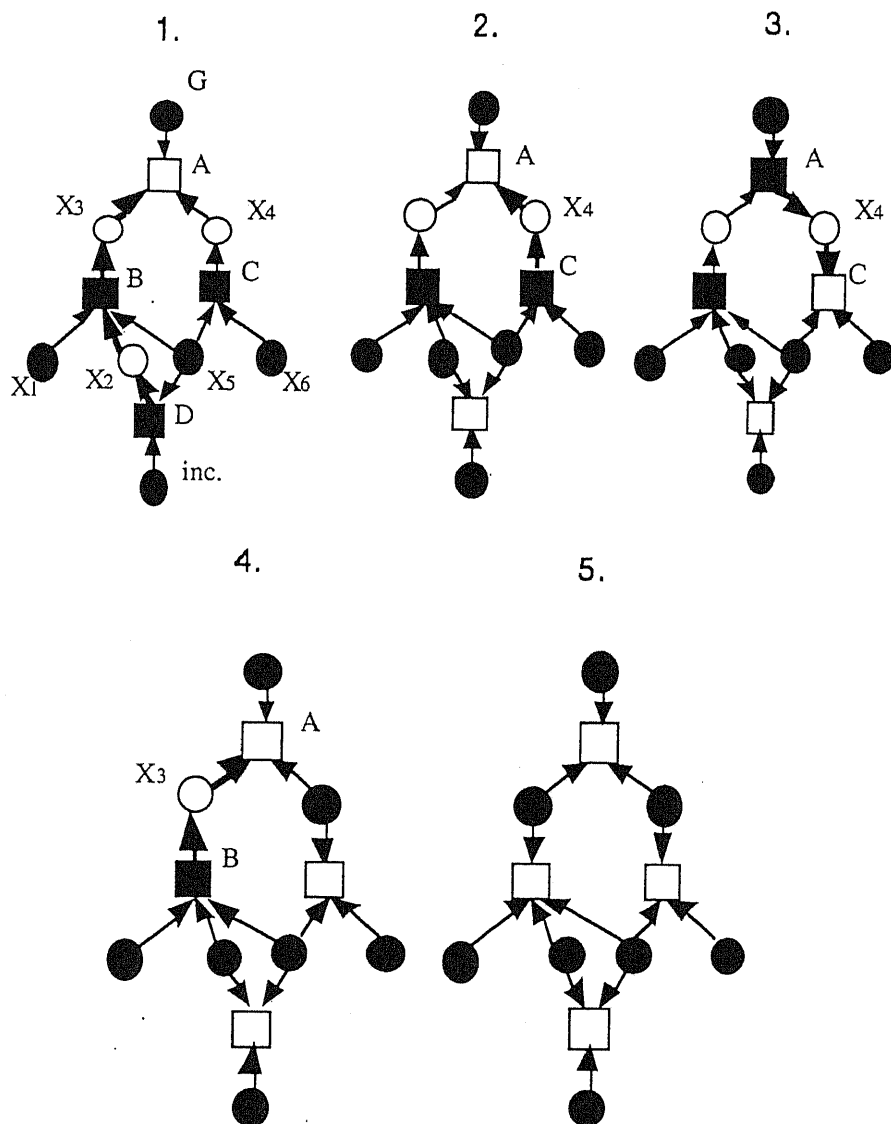


図3.5. NBPを用いた仮説推論の例（探索フェーズ）．このあと改良フェーズを実行する．太い矢印は伝播パスで表し，状態番号は表1と同じものとなっている．

従って，推論中バブルの個数を一定に保ち，バブルがどのノードにあるかで真理値空間での探索点の座標を表しながら推論することにより，PC法と同等のプロセスを実行できる．即ち，バブル一つの伝播が一回の掃



き出し操作を実現する.

### 3.3.3. 支配ベクトルと伝播パス

ここまでは真理値空間での状態をネットワークとして表現したに過ぎないが、NBP法では探索点の座標に当たる各ノードの値の決まり方が目に見える形で表せる。これが支配ベクトルであり、支配ベクトルから得られる伝播パスがバブル伝播（掃き出し）操作において（非基底，基底）の交換ペアの候補を絞り込むことがNBP法の高速推論を可能にする。支配ベクトルとは、各ノードの値（構造ノードの場合は真理値）がどの隣接ノードの値を支配するかを示す方向ベクトルである。真理値空間と関連づけてこの支配関係を見てみると、●ノードは自分自身の値が0か1に等しいという平面を成している。これに対し、N個の構造ノードで囲まれた■ノードは、式(3-7)の様にある線形関数 $f(X)=0$ （ただしXはN個の構造変数を表すN次元ベクトル）と表されるので、周囲の構造変数のN-1個が決まったとき残る1変数の値を決定する。○ノードはこの残り1変数としてのみ値が決まり、初めて周囲の値を支配するようになる。そして□ノードは、周囲の構造ノードの線形和として受動的に値が決められる。これらの支配関係は、図3.6の手順で矢印として付与された支配ベクトルで表現できる。手順5までは支配するノードから支配されるノードを支配ベクトルで指し示し、手順6は逆に支配されるノードから支配するノードを指定することで、ネットワークから読み取れる支配関係を網羅している。

1. ●ノードから、接する全ての枝に矢印を放出する。
2. 接する枝の1つを残す全てから矢印を受け取った各■ノードは、残りの1枝に矢印を放出する。その様な■ノードがなければ4へ。
3. 接する枝の1つから矢印を受けとった各○ノードは、接する他の全ての

- 枝に矢印を放出する．2へ．
4. □ノードは，接する全ての枝から矢印を受け取る．
5. 接する枝の1つを残す全てに矢印を放出した各○ノードは，残りの1枝から矢印を受け取る．その様な○ノードがなければ終了．
6. 接する枝の1つに矢印を放出した各■ノードは，接する他の全ての枝から矢印を受け取る．5へ．

図3.6. 支配ベクトル ("矢印") の付与手順

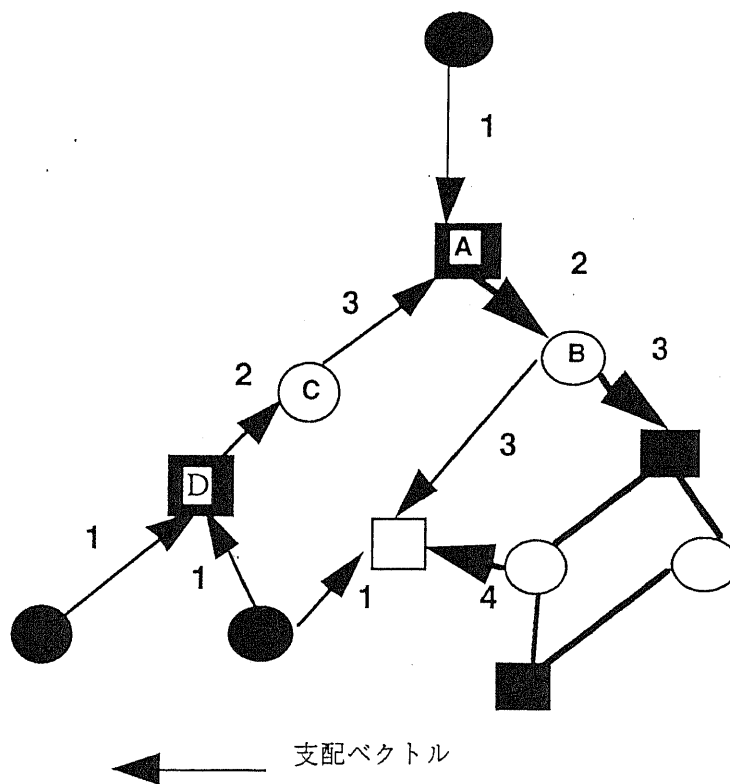


図3.7. 支配ベクトル付与の1例--番号は上述の付与手順．太線はAをターゲットとした伝播パス．

### 伝播パスの決定

支配ベクトルの付与の一例を図3.7に示す。枝の添え字は図3.6での手順の番号である。ノードの添え字は集合Sに入った手順の番号である。また、6-1'などで、手順6で書き換えられた手順1などを表す。

次に、支配ベクトルから伝播パスを与える。伝播パスとは、掃き出し操作で基底状態になろうとする非基底ノード（以下、ターゲットと呼ぶ）に対して図3.8の手順によって与えられるパスをいう。

1. 伝播パス=ターゲットとする。
2. 伝播パス中の探索トレース済みでないノードがあればその一つをAとする。なければ終了。
3. Aに接し、支配ベクトルがAに対し内向きに付与されていない枝とそのAと反対側の端点を伝播パスに加え、Aを探索トレース済みとし2へ。

図3.8.伝播パスを与える手順

伝播パスは、支配ベクトルを流れの向きとみたターゲットの下流といってもよい。一般には図3.7右端の区域の様に支配ベクトルを施せないことがあり、このようなターゲットの下流にある流れの「平坦地」も含めて伝播パスとする。掃き出し操作でターゲットの交換相手となる基底ノード（バブルの伝播先で、以下交換相手と呼ぶ）は常にこの伝播パスに含まれる。実際、支配ベクトルは真理値空間で平面が交差して探索点を決定する様子を反映するので、ターゲットより上流または伝播パス外のノードは交換前の支配ベクトルで既に値が決まっている。もし上流にバブ

ルを伝播すると、これにまた一つ拘束が加わることになり、矛盾または冗長となる。例えば図3.7の場合、ターゲットAに対する交換相手は太線を施した伝播パス、即ちAの下流の中から選択される。仮にBが交換相手となってバブルがAからBに伝播するならば、伝播後の新しい状態で支配ベクトルを付け直すときに■から□に変わる（図3.6の手順2）のはノードAだけでありBが●となるのでバブルの個数は保たれる。しかし、Cに伝播すればCが●となり、ターゲットAだけでなくノードDも■から□に変わりバブルが不足してしまう。

この、交換相手が伝播パス上に絞られる性質によって各時点でのバブル伝播（後述）が効率的に実行され、高速推論の達成が可能になる。

### 3.4.NBP法の動作

前節で構成法を示したBPNを用いた，NBP法の動作原理について述べる．初期フェーズがPC法と同じであることは既に述べたので，ここには探索フェーズで特に時間のかかる掃き出し操作を，バブル伝播で実現する方法を中心に述べる（改良フェーズについては本節末尾）．

#### 3.4.1. ターゲットの交換相手の選択

伝播パスからバブルの伝播先を選択する手法を述べる．2節の終りに記したように，PC法の掃き出し操作では元の非基底変数は探索点が多面体から抜け出さない向きに値が変化し，その変化量が最小となる基底変数と交換される．同じ理由から，NBP法では伝播パス上でターゲットの値を最も小さく変化させる基底ノードにバブルを伝播させる．ここでは，新しく定義する三つの量を用いてこれを実現する．処理の流れの概要は，後述する 1) 影響度 2) 変化率 3) 相対深度 をこの順序で算出し，3) の値が最小となる基底ノードを交換相手とする．

まず，ノードYからノードXへの影響度とは，図3.9の手順で得られる量  $w(X, Y)$  をいう．ただし， $p(X, Y)$  は3.3.1節で定義した伝播率とする．

図3.10はrank=2で図3.9の手順が終了する例である．図3.10 (1) のターゲットをT1とし，交換相手を選びたいとする．図3.10 (2) ではじめに太線を施したrank=1での集合Iを得る．これだけでは伝播パスに矢印が行き渡らないが，rank=2で全ての枝に矢印を施すことができる．このように，支配ベクトルがない枝にも矢印を施せるように一部の○ノードも●ノード同様に矢印を出すのである．

1.  $\text{rank}=1, \text{rank}'=1$ .

伝播パス上で最も上流の構造ノード  $S_1$  とターゲット  $T_1$  を集合  $I$  の要素とする。伝播パスから支配ベクトルを取り除き、伝播パス上の全てのノードを発火前とする。7の"終了"まで  $T_1$  を基底として (■は□, ●は○として) 扱う。

2. 伝播パス上の各ノード  $X$  について  $S_{\text{rank}}$  に対する変化率を,

$w(X, S_{\text{rank}})=0$  ( $X \neq S_{\text{rank}}$ ),  $w(S_{\text{rank}}, S_{\text{rank}})=1$  とする。

3. 集合  $I$  に発火前の○が含まれれば任意の一つを  $A$  とする。なければ集合  $I$  中で (隣接枝数-1) 本以上の内向きの矢印に接する発火前の■があれば任意の一つを  $B$  とし,  $B$  を発火後として5へ。それとなければ7へ。

4.  $A$  から矢印の未付与の枝全てに外向きの矢印を施し, その端点ノードそれぞれを  $B$  として集合  $I$  に加え,  $w(B, S_i)$  に  $p(B, A) * w(A, S_i)$  を加算する ( $i=1, 2, \dots, \text{rank}$ )。  $A$  を発火後としその後3へ。

5.  $B$  に伝播パス外の支配ベクトルと併せて (隣接枝数-1) 本の内向きの矢印が接していれば残りの隣接枝に外向きの矢印を施し, その端点の○ノード  $C$  を集合  $I$  に加え,  $w(C, S_i)=p(C, B) * w(B, S_i)$  ( $i=1, 2, \dots, \text{rank}$ ) とする。

6.  $B$  に伝播パス外の支配ベクトルと併せて (隣接枝数) 本の内向きの矢印が接していれば,  $\text{rank}'=\text{rank}'+1$  として  $T_{\text{rank}'}=B$  とする。この条件と無関係に必ず3に戻る。

7. 集合  $I$  に含まれず, 集合  $I$  中のノードに隣接する伝播パス上の○があれば,  $\text{rank}=\text{rank}+1$  としてその任意の一つを  $S_{\text{rank}}$  として集合  $I$  に加え2へ。なければ手続きを終了する。

### 図3.9: 影響度計算の手順

上記の計算手順を, 図3.10の例を用いて説明する。

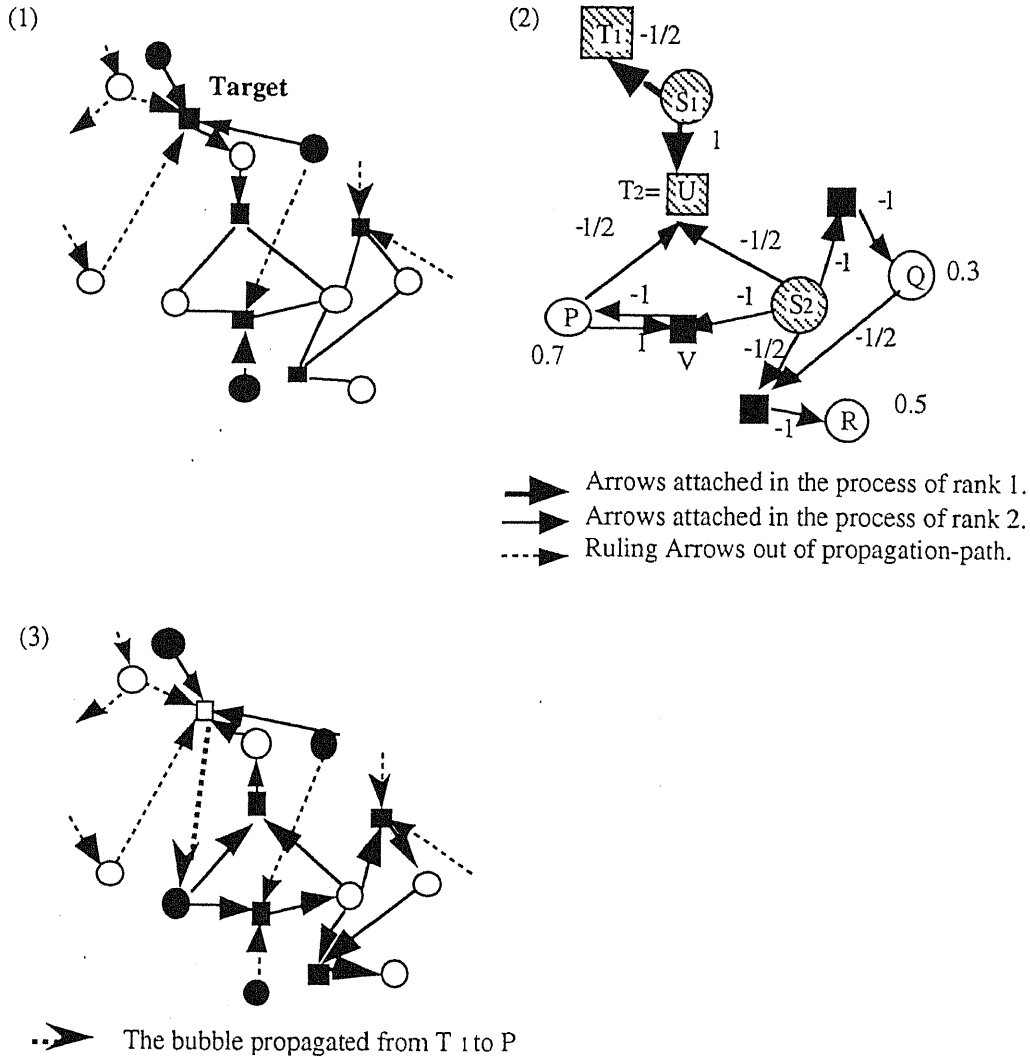


図3.10. ターゲットT1からノードPへのバブル伝播

はじめに仮説推論の問題がBPNの形に書き直された時点で、各枝の伝播率は与えられる。これを図の各枝に添えた値とする ( $p(U, S1) = 1$  など)。

rank=1の間に、 $w(S1, S1) = 1$  から始まって枝の伝播率  $p(T1, S1) = -1/2$ ,  $p(U, S1) = 1$  から影響度が  $w(T1, S1) = -1/2$ ,  $w(U, S1) = 1$  となる。即ち、ノードS1の値の単位量 (=1) 変化の影響でノードT1、ノードUはそれぞれ  $-1/2$ ,  $1$

だけ変化しようとする。以下ではノードXの変化量を $\text{var}(X)$ で表す。

次に $\text{rank}=2$ となって手順7で選ばれた $S_2$ から各ノードへの影響度を求める。

$p(V, S_2)=-1$ よりノードVへの影響度 $w(V, S_2)=-1$ が求まり、 $p(P, V)=-1$ より $w(P, S_2)=1$ となる（ $p(P, V)$ の値の由来は付録1参照）。同様の計算を続けて $w(U, S_2)=-1$ を得たところで、手順6で $T_2=U$ となる。他のノードの影響度の算出は省略するが、図3.9の手順の結果、式(3-14)の影響度を得る。

$$\begin{pmatrix} w(T_1, S_1) & w(T_1, S_2) \\ w(T_2, S_1) & w(T_2, S_2) \\ w(P, S_1) & w(P, S_2) \\ w(Q, S_1) & w(Q, S_2) \\ w(R, S_1) & w(R, S_2) \end{pmatrix} = \begin{pmatrix} -1/2 & 0 \\ 1 & -1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix} \quad (3-14)$$

この手順で得た影響度は、変化率を算出するための材料となる。変化率とは、伝播パス上にあるターゲット及び基底ノードの変化量の比率をいう。即ち、ターゲット以外の非基底ノードの値が変化しないという条件を表した式(3-15)を解けばよい。式(3-15)で右辺1行目がターゲットの変化量を表し、右辺2行目以下は $T_i(i>1)$ の値が変化しないことを表している（図3.9の手順終了時の $\text{rank}=\text{rank}'$ を前提としているが、理由は付録2参照）。

$$W \cdot \begin{pmatrix} \text{var}(S_1) \\ \text{var}(S_2) \\ \vdots \\ \text{var}(S_{\text{rank}}) \end{pmatrix} = \begin{pmatrix} \text{var}(T_1) \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

$$\text{where } W = \begin{pmatrix} w(T_1, S_1) & w(T_1, S_2) & \cdots & w(T_1, S_{\text{rank}}) \\ \vdots & \vdots & \ddots & \vdots \\ w(T_{\text{rank}}, S_1) & \cdots & w(T_{\text{rank}}, S_{\text{rank}}) \end{pmatrix}. \quad (3-15)$$

この式では $\text{rank}$ 個の変数に対し、2行目以下の $\text{rank}-1$ 個の制約があるので、得られるのは $\text{var}(S_i)(i=1, 2, \dots, \text{rank})$ の比率、従ってノード  $S_1, S_2, \dots$



Srank の変化率である.

図3.10の例では, 式(3-15)の右辺2行目から得られる" $\text{var}(S1)-\text{var}(S2)=0$ "より変化率 $\text{var}(S1):\text{var}(S2)$ は1:1となる. ノード  $S1, S2, \dots S_{\text{rank}}$  から伝播パス上の全てのノードへの影響度は既に得られているので, 式 (3-16) に示すようにターゲットまたは伝播パス上の基底ノード $X$ の変化量は, ノード  $S1, S2, \dots S_{\text{rank}}$  の値の変化による影響の総和となる.

$$\text{var}(X) = \sum_{i=1}^{\text{rank}} w(X, S_i) \text{var}(S_i). \quad (3-16)$$

図3.10では, 変化率 $\text{var}(T1):\text{var}(S1):\text{var}(S2):\text{var}(P):\text{var}(Q):\text{var}(R)$ は  $-1/2:1:1:1:1:1$ となる. これを基に, ノード $P$ が仮に交換相手となる場合のターゲット $T1$ の変化量を考えてみる. 今,  $p$ を0.7,  $t1$ を0 (上限) とする. ターゲット  $T1$  の値  $t1$  は上限から減少する方向に変化しなければ制約を違反するので, 変化率が  $T1$  と異符号の $P$ の値は増加して上限値1となる. 従って $\text{var}(P)=0.3$ となるので, 変化率から $\text{var}(T1)=-0.15$ となり  $t1$  は-0.15に変化する. これを一般的な形に定式化すると, ターゲットまたは伝播パス上の基底ノードを $X$ とし $\text{up}(X), \text{lw}(X)$ をそれぞれノード $X$ の上限値及び下限値とすると以下のようなになる.

$$\begin{aligned} &\text{if } \left( \frac{\text{up}(T_1) + \text{lw}(T_1)}{2} - t_1 \right) \frac{\text{var}(X)}{\text{var}(T_1)} > 0 \text{ then } \text{depth}(X) = \frac{\text{up}(X) - x}{\text{var}(X)}, \\ &\text{if } \left( \frac{\text{up}(T_1) + \text{lw}(T_1)}{2} - t_1 \right) \frac{\text{var}(X)}{\text{var}(T_1)} < 0 \text{ then } \text{depth}(X) = \frac{\text{lw}(X) - x}{\text{var}(X)}, \\ &\text{if } \left( \frac{\text{up}(T_1) + \text{lw}(T_1)}{2} - t_1 \right) \frac{\text{var}(X)}{\text{var}(T_1)} = 0 \text{ then} \\ &\quad \text{if } x = \text{up}(X) \text{ or } x = \text{lw}(X) \text{ then } \text{depth}(X) = 0, \\ &\quad \text{otherwise } \text{depth}(X) = +\infty. \end{aligned} \quad (3-17)$$

ここで、 $\text{depth}(X)$ の絶対値を $X$ の相対深度と呼ぶ。これは伝播パス上のノードが変化率に比例して一様にどれだけ変化すれば、基底ノード $X$ の値が上限値または下限値に達するかを表す。即ち、 $T1$ はこの値が最小の基底ノードと交換したとき最小の変化量となるので多面体を抜け出さない。相対深度は大小だけを比較するので、変化率だけ正しく考慮し $\text{var}(T1)=-1/2$ ,  $\text{var}(S1)=1$ などとして計算してよい。式(3-17)の1, 2行目でifに続く不等式はそれぞれ $x$ を増大, 減少させる条件を表し, 3行目以下はターゲット $T1$ の上限と下限が等しいため値 $t1$ が変化できないとき, 既に値が0または1の○ノードだけを(値を変えずに)●に換えてよいことを表している。尚,  $\text{var}(X) \neq 0$ ,  $\text{var}(T1) \neq 0$ を前提としている(このことは $\text{var}(X)=0$ ,  $\text{var}(T1)=0$ はいずれも稀で6章の実験で一度も起きなかったので今回は考慮しない)。

上の例では始め $T1$ の値が上限なので1,2行目でifに続く括弧内は負の値になり,  $P$ と $T1$ の変化率が異符号なので1行目の式が実行され $\text{depth}(P)$ が0.3となる。

今, バブル伝播前の各ノードの値が

$$(t_1, s_1, s_2, p, q, r) = (0, 0.1, 0.7, 0.7, 0.3, 0.5)$$

であったとすると, 式(3-15)から $\{T1, S1, S2, P, Q, R\}$ の相対深度はそれぞれ $\{1, 0.9, 0.3, 0.3, 0.7, 0.5\}$ となる( $T1$ の下限を $-0.5$ とした)。そして相対深度の最小なものの一つである $P$ を交換相手とする。この結果,  $P$ の変化が $+0.3$ となるように各ノードの値が変化率に比例して変化する量は,

$$d(t_1, s_1, s_2, p, q, r) = (-0.15, 0.3, 0.3, 0.3, 0.3, 0.3).$$

となり, 変化後の値はそれぞれ次の様になる。

$(t_1, s_1, s_2, p, q, r) = (-0.15, 0.4, 1.0, 1.0, 0.6, 0.8).$

実際、どれも上限と下限の制約を満たしている。この場合ターゲットT1からノードPにバブルが伝播し、図3.9の手順1で除去されていた伝播パス上の支配ベクトルを図3.6の手順で付け直し図3.10(3)となる。

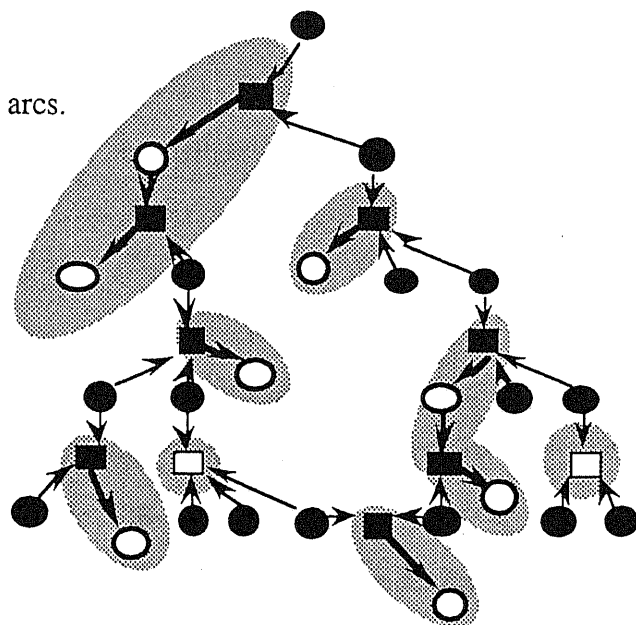
この様に、推論におけるバブル伝播（掃き出し）操作は各時点で、1) 交換相手の候補を伝播パス上に絞り、2) 枝に沿った伝播率の乗算で実行できる。5節に示す実験では、仮説数100（ノード数300前後）以上の大規模な知識ネットワークでも伝播パスは23ノード以下の比較的小規模なサイズに収まり、仮説数がさらに大になっても著しいサイズの増大は見られなかった。では、伝播パスのサイズは何を反映しているのだろうか。

図3.11に見られるようにネットワークが複雑になると伝播パスのサイズは大になる。この図で網点を施した各区域は、どの区域のノードからも●を越えずには他の区域に入れない支配ベクトルの「盆地」のような範囲である。伝播パスはターゲットの下流であるから、もしターゲットが■ノードならどの伝播パスもこれらの区域のどれか一つに含まれている。

5.2のStep3の改良理由"1)"に示すようにターゲットが●ノードとなるのはこれに比べると極めて稀なので、この区域がほぼ伝播パスのサイズの上限を与えているとみなすことができる。

このことから、伝播パスのサイズは知識ネットワークの規模よりも、構造の複雑さを反映していると考えられる。このような伝播パスのサイズの知識構造への依存に関する厳密な解析は、今後の課題である。しかし、この様に各時点での操作の対象を知識の一部分に絞る考え方は知識処理の観点から重要な視点であり、高速な推論につながる点で実用的な意義を持つと考えられる。実際、6.2に述べるように、本手法による計算時間の短縮はこの効果を反映したものと考えると説明しやすい傾向を示している。

Graph A, with 34 arcs.



Graph B, with 48 arcs and the same nodes as Graph A.

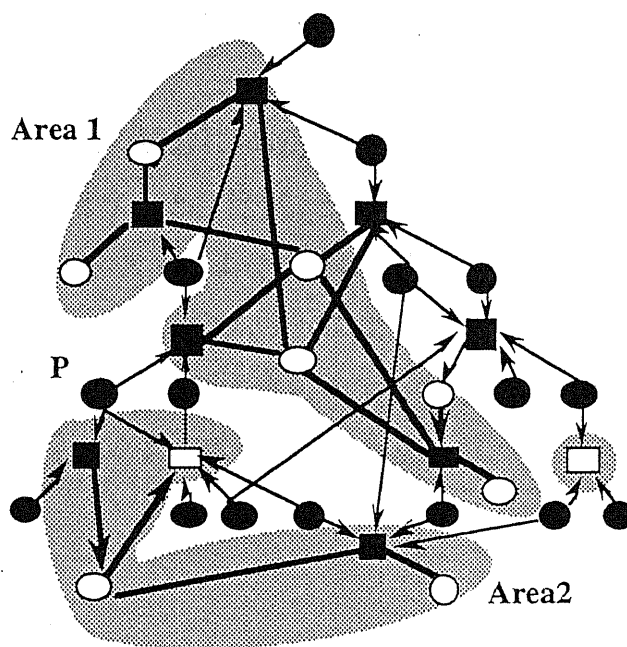


図3.11. 支配ベクトルによるBPNの分割

### 3.4.2.NBP法におけるPC法からの他の改良点

上に述べたバブル伝播は、PC法における制約を充足したままでの掃き出し操作（図3.1のStep2, 3）をNBP法のプロセスとして実現したものである。NBP法ではこれ以外に以下の点を改良している。

#### Step3

本論文の手法ではStep3で■ノードと□ノードの交換を優先する。即ち、本来Step3の交換はいずれも精度を高め制約充足（実行可能）解をより確実に求めるためのものだが、■ノードと□ノードの交換が見つからない時だけ●ノードと○ノードの交換を実行する（下記理由1により稀である）ことにする。この理由は、次の3点である。

- 1) Step3の条件さえ満足すればいずれの交換でも制約充足を維持して整数解に接近でき、殆どの場合、そのような■ノードと□ノードの交換を見つけることができる。
- 2) ●ノードをターゲットとした場合には伝播パスが広くなり（図3.11下図でノードPをターゲットとすると、区域1と区域2を併せたものが伝播パとなる）、Step3の条件のテストおよび交換の実行に時間がかかる。
- 3) 一旦0.1に落ち着いた●ノードを○ノードに変化させても、再度●にもどった時に値も元にもどる 場合は遠回りをしたことになる。あるいは、値が元の補数になる場合の探索点の移動はStep7で代行できることがある。

#### Step6

Step3でも交換が見つからない時は、多面体の外へ最小限度はみ出すことを許しており、Step6で非基底スラック変数を最小限変化させる基底構造変数と交換している。これと同等の機能をNBP法の操作として実現する

ため、■ノードをターゲットとして伝播パス上の○ノードのうち最小の相対深度のものを交換相手とする。

#### Step7-9

あるノードの変化の影響で●の値が変化したり、□の変化がさらに周囲のノードに伝播することはない。この性質は、伝播パスという形に集約され掃き出し操作を効率化する鍵となったが、Step7, 9で補数をとる際もこの性質を活かすことができる。即ち、●ノードの値の変化は支配ベクトルの向きに逆らってまで遠くに伝播することはない。そこで、Step7, 9では上限と下限の制約を違反する□ノードの上流の●ノードのみ補数をとる。こうすれば、そのスラック変数を含む区域（図3.11）に隣接する●ノードだけに補数をとるノードが絞られる（4.6節で更に改良する）。

#### Step10-13

図3.1の改良フェーズStep10-13を次の操作10で置き換える。これは、なくてもゴールの証明が可能な冗長仮説を消去する操作で、仮説からボトムアップにゴールの証明を試みる高速なプロセスとなる。

10. ここまでに得られた●ノードの値を対応するアトムの真理値とし、真（真理値が1）となっている仮説をそれぞれ一時的に偽としても式(3-1), (3-3)の推論ルールからゴールGの証明が可能か試みる。成功したらこの仮説とこの証明試行中に偽となるアトムを偽に変更する。

最後に、もし混合計画問題も取り扱えう必要があれば、次の改良を行えう。これは、一端●ノードとなったノードがその後○ノードとなるチャンスはNBP法では少なくなってしまうためである。

#### Step2（混合計画問題を扱う場合の変更）

- a) 0-1整数値となる必要のない構造変数は、■ノードと○ノードの交換の一方の○ノードと見做さない。
- b) 探索フェーズは、0-1整数値をとる様に定義された構造変数のみに全てのバブルが集結した時点で終了とする。

ただし本論文では、混合計画問題は述語論理版のタイプ2の矛盾のみで生じるが、これはa)の改良が不要となる特殊な場合である（6.3.6参照）。

### 見えない支配関係が存在する場合

最後に、比較的まれなケースだが起こると致命的なエラーとなる一つの場合とその対策について述べておく。

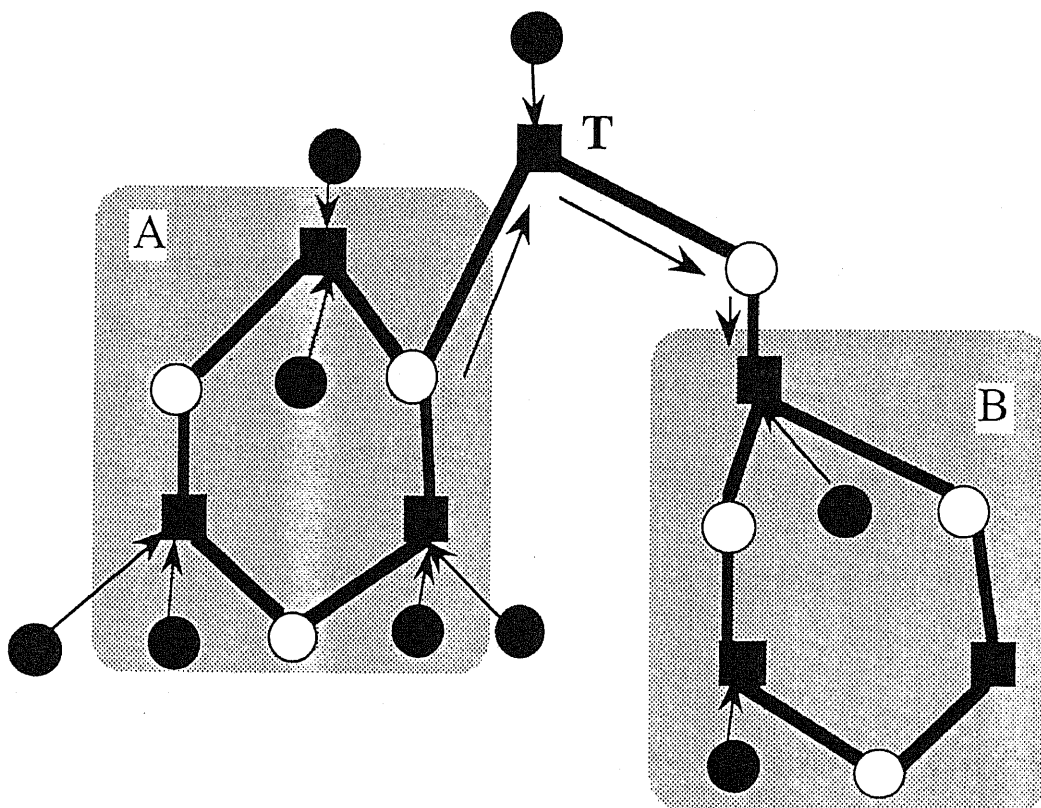


図3.12. 見えない支配関係

先に示した支配ベクトル付与は、ノードのレベルでバブルの位置から容易に視覚あるいは理解可能な支配関係を表すものであった。これに対し、図3.12に示す様に支配ベクトルは付与できなくても実際には支配関係が存在する場合がある。

この図では、サブグラフAには3つの■ノードが存在してそれらに隣接する○ノードは全てAの内部に含まれるので、Aに含まれる3つの○ノードの値はAの内部だけで決定されている。これに対し、サブグラフBに含まれる■ノードの一つは隣接する○ノードがBの外にあって、しかもAに繋がるパス上にある。従って、両サブグラフ間には実は→で示した支配関係が存在していることになる。これを知らずに影響度計算の $S_1$ をターゲットTより支配関係が上位の部分グラフAの中に置いた場合、真の伝播パス（ターゲットTに支配されるノードからなる部分グラフ）に対する計算は実行できなくなる。

しかし、この見えない支配関係を支配ベクトルで表現するには新たに計算量のオーバーヘッドがかかることになる。即ち、（次章の定理1に示すように）支配ベクトルが付与できないのは特定の閉部分グラフに対してのみ生じるケースであるので、支配する部分グラフを特定するためにはループ検出という計算コストの高い処理が必要となる。そこで、次の処理によってこの問題に対処する。

1. 伝播パス中で支配ベクトルの最も上流にある基底構造ノードを $S_1$ とする。
2. 図3.9の影響度計算の手順を実行する。
3. もし、ターゲット以外のいずれかの■ノードが周囲の全ての枝から矢印を受け取るならば、伝播パス内の矢印を消し、支配ベクトルの最も上流にある現時点の $S_1$ とは別の基底構造ノードを $S_1$ として2へ。それ以外ならば影響度計算を終了する（変化率計算へ）。



ここで、今はターゲットが■と○のみからなる閉グラフ（ループ）に含まれる場合を考えているので、「最も上流」の構造ノードが少なくとも2つ以上存在することが前提となっている。3の条件は、支配ベクトルだけでは情報が不足していた為に誤って実際にはターゲットを支配する部分グラフにS1をおいていた場合を指している。この様にすれば、ターゲットに隣接して支配する部分グラフが存在しても、必ずその外にS1をおくことが可能となる。その様な部分グラフが存在しない場合には、見えない支配関係が存在しない訳であるから、手順2を1度だけ実行して変化率計算へ移行する。

### 3.4.3.NBP法の動作例

ここに述べた手法の例として、図3.4の仮説推論の問題を考える。'G'で示したゴールには真理値1が与えられ、'inc'で表されている矛盾ノードは矛盾のホーン節のヘッドで、その値を0とする。問題は初めにBPNに変換され、初期フェーズで単体法により実数最適解が求められ、図3.5の状態1となる。この後の探索フェーズでバブル伝播により推論が行われる様子を図3.5を用いて以下に述べる。図3.5の状態1,2,3,4,5以外に立ち寄る状態を含め、各状態でのノードの値は表3.1に記した。

状態1→状態2 まず、状態1でDをターゲットT1とすれば伝播パスで最も上流の○ノードX2がS1となり

$$w(X2, X2) = 1, w(T1=D, X2) = p(D, X2) = -1/2,$$

$$w(X3, X2) = p(X3, B) p(B, X2) = 1/3,$$

$$w(A, X2) = p(A, X3) w(X3, X2) = -1/6.$$

と影響度がrank=1で計算できる。故に式(3-16)から影響度がそのまま、

$$\text{var}(D) : \text{var}(X2) : \text{var}(X3) : \text{var}(A) = -1/2 : 1 : 1/3 : -1/6.$$

と変化率になる．この変化率と，Dの値が下限であることから，式(3-17)で  $\{D, X2, X3, A\}$  の相対深度はそれぞれ  $\{1, 0, 2, 0.48\}$  となる．この値が最小であるX2は○ノードなので，図3.1のStep2でDとX2の交換が可能となり，D→X2のバブル伝播により状態2となる．以下，相対深度の計算は同様であるので省略する．

状態2→状態3 状態2では図3.1のStep2の交換にあたるバブル伝播は見つからず，ターゲットCから伝播パス上で相対深度が最小である基底ノードAにバブルが伝播するとX4が  $0.5 \rightarrow 0.34$  と変化し，非整数度を減らせる（図3.1のStep3）．結果は状態3となる．

状態3→状態4 状態3では可能なバブル伝播は見つからない．実際，Aをターゲットとし状態2に戻ると非整数度指数を増やしてしまう．X1をターゲットとすると伝播パス上の  $\{X1, X3, X4, C\}$  のうち相対深度はCが最小となるが●ノードから□ノードへのバブル伝播にあたる非基底構造変数と基底スラック変数の交換は図3.1の手順にはない．ターゲットがX2又はX5の場合も相対深度がそれぞれノードD, Cで最小となり同様である．そこでStep5で四捨五入を行ってみるが，状態3'のようにノードBが上限を上回ってしまう（状態3'の評価"×"）ので，状態3に戻る．ここでStep6が実行される（4.2参照）．即ち，Aをターゲットとした伝播パスで，相対深度が○ノードとしては最小のノードX4へバブルを伝播させて，ノードCだけが上限の制約を越える状態4'に至る．Step7でX6の補数をとることにより状態4で制約充足する．

状態4→状態6 状態4でBをターゲットとし，相対深度が最小のノードX3にバブルを伝播し探索フェーズは終了する（状態5）．この後，改良フェーズ（5.2末尾）でX1を偽として最適解を得る（状態6）．

	X <sub>1</sub>	X <sub>2</sub>	X <sub>3</sub>	X <sub>4</sub>	X <sub>5</sub>	X <sub>6</sub>	A	B	C	D	手順
1	1.0	0.0	0.67	0.5	1.0	0.0	0.42	0.0	0.0	-0.5	2
2	1.0	0.0	0.67	0.5	1.0	0.0	0.42	0.0	0.0	-0.5	3
3	1.0	0.0	0.67	0.34	1.0	0.0	0.5	0.0	-0.16	-0.5	5,6
3'	1.0	0.0	1.0	0.0	1.0	0.0	0.5	<u>0.33</u>	-0.5	-0.5	×
4'	1.0	0.0	0.67	1.0	1.0	0.0	0.17	0.0	<u>0.5</u>	-0.5	7
4	1.0	0.0	0.67	1.0	1.0	1.0	0.17	0.0	0.0	-0.5	2
5	1.0	0.0	0.0	1.0	1.0	1.0	0.5	-0.67	0.0	-0.5	10
6	偽	偽	偽	真	真	真					成功

表3.1. 本動作例におけるバブル伝播ネットワークの状態遷移

### 3.5. NBP法の知識処理的な意味

知識ベースをネットワーク化することで知識処理としての意義を持つようになることは先に述べた通りである．このことによる拡張性については，第6章において述語論理版NBP法が得られることなどに現われるであろう．

この節では，この様な拡張性について論じる以前に，命題論理の仮説推論の問題ではNBP法が知識処理の手法としてはどのような戦略をとっていることになるのかを知ることによって，理解を深める．

#### 3.5.1. 初期値設定の意義

NBP法の母体となった掃き出し補数法は，簡約すると，

1. 単体法によって瞬時に最適解に接近する．
2. 探索フェーズにおいて，0-1整数値の実行可能解に到達する．

という2段階でNP完全（NP困難な問題は解けない）な問題を解いていることになる．その後の改良フェーズは最適解に一層近づくための工夫であるが，探索フェーズまでに $NP \rightarrow P$ という短縮が図られていることから，この改良フェーズは速度向上に寄与していない（実際，探索フェーズ終了時点で最も近くにある極小解に辿り着く程度の役割しか果たさないの

で，これをボトムアップ推論形式の手続きに改良し，これにより本研究では一層の速度向上を果たしている）．

1と2のフェーズはそれぞれ，人の思考における直感と検討に対応する．1は厳密に多項式時間で計算が可能であり，2が $NP \rightarrow P$ という短縮の全てになっているからである．ここで直感とは，記号や文章として明示できる以前に瞬時に生成される意味体で，現時点よりも問題の解決に近いと予想される状態を把握したものと考える．

この意味では，白み潰しに探し回る枚挙法というのは直観を用いていな

いことになる。というのは、この中には現在より良いか悪いか検討もつかない状態が多く含まれるからである。一方、補助線を用いた定理証明というのは、対象(制約)である図形の状態を目標状態に近付けることに当たり、一種の直観と呼べる。

この意味では、初期値を0-1整数制約を外した線形計画問題の最適解とすることは次の意味で直感である。

- 1) 問題を与えられた時点よりも評価関数、実行可能性の両面で目標に近い。
- 2) 非常に高速に得られるが、離散的ではないので有限長の記号列で明に記述できない。

同時に、この最適の状態が探索空間の次元数と同数の超平面の集合と等価であることも一定の意義を有する。例えば、ここに非常にぎこちないロボットの制御を考える。各関節は、90度と180度の2通りの角度しかとれないとする。この関節が $N$ 箇所あったとし、狭い空間で目標物に手を伸ばすことを考えよう。空間の狭さが制約となり、始めにロボットが座っていたとすれば立つための足の関節の位置は限定される上、膝の位置などにも角度の2値制約によって壁にぶつかるよりもかなり手前で妥協する必要が生じる。従って、座った姿勢から体を起こして行き、目標物に触れるまでには最悪で $2^N$ 回の試行錯誤を行うこととなる。そこで、同じ位の大きさの人がその場でその物を取り、その時と良く似た姿勢からロボットを動作させることを試みる。人の場合には部屋の狭さの制約は残り、手足の数だけの壁の障害に妨げられながらも、ロボットよりは幾分伸びやかに目標に到達するであろう。即ち、どの状態でも用いる手段の数だけクリティカルな障害を考慮する必要がある。単体法の場合、空間次元数=制御対象の構造変数の個数=各時点でクリティカルな制約の数という点で丁度この例で述べた原則を具現化していることになる。狭い部屋で

も、柔らかな関節を持つ我々人は体を瞬時に伸ばすことができるが、この瞬時の動作が直感と対応するのである。

### 3.5.2. NBP法のトップダウン（ゴール指向）な性質

NBP法の特徴の一つは、ゴールからの距離に近いアトムから順番に真偽が確定することである。例えば、次の問題を解かせてみる。

背景知識:

1	:-	a1.
a1	:-	b4, b5, c4.
a1	:-	b1, b2, c1.
a1	:-	a2, a3, b1.
a2	:-	b3, c2, a4.
a2	:-	c2, a5, b3.
b1	:-	c1, c2, c3.
0	:-	b4, b5.
0	:-	b3, c2.

候補仮説: a1, a2, a3, a4, a5, b1, b2, b3, b4, b5, c2, c3, c4

知識ベースは、読み込み時に、次の手順によってゴール指向性を有する知識ネットワークにトップダウンに変換される。この手続きでは、ホーン節の条件部の各アトムが出現回数だけ読み込まれ、それら全てが読み込まれた時点で読み込み操作が終了するのであるから、その所要時間は知識の規模に線形に比例する。伊藤らの推論パスネットワークと同様のゴール指向性を、この部分で瞬時に得ることが可能となっている。

### ゴール指向性を実現する手順

1. ゴールを集合Queriesの最初の要素とする.
2. ヘッドが集合Queriesに含まれるホーン節を読み込み, その条件部に現われたアトムを, 集合Queriesに加える.
3. 知識ベースファイルの終点まで到達した時点で, 既に知識ベース内の全てのホーン節を読み込んでいたら終了. それ以外の場合には知識ベースの先頭の文字に戻り2へ.

### ◎トートロジーの省略によるメリット

以前のNBP法では, 次項の例の4行目までの様に選言ルールがあった場合, 子アトムである3つのa1の値が全て一旦0.5となるとその後どの枝を真理値1に近付けるかで結果が左右されていた(図3.13のa). 即ち, この選択を誤ると補数といえども多面体内に戻れないことがある. これは, (図3.13のb)の様に上のゴール指向な読み込みによりトートロジーを避けることで防ぐことができる様になった. これについて簡単に説明しておく.

$X:-Y$ の様に, ヘッドをXとするホーン節が一つだけで, しかもそのボディのアトムが一つしかない様な場合には, これは $X=Y$ なるトートロジーを意味する. 図3.13のaは, ゴールにこの形のトートロジー " $1:-a1$ " (このシステムではゴールを1と表している) が存在することを示しており, 図の網点のスラックノードまで制約違反状態が伝播すると, その直前に補数計算をしたa1は補数をとることが許されない為にこの状態で推論失敗と判定されてしまう(4.6節参照). 同様の問題は, ヘッドが0である矛盾制約も  $0:-X, Y. \leftrightarrow \neg X \vee \neg Y$  の様に対偶が選言ルールとなるので発生する. 一般に, 図3.13のaの様にグラフを構成する場合には, トートロジーの箇所では制約違反が固定されてしまうケースが推論の失敗の多くのケースを占める. 従って, トートロジーの両辺のアトムを同一視する機構が

必要となるが、 $X:-Y$ という知識が両辺を別の箇所で見つけた場合にはこの機構が働かないといえる。そこで、上に示したトップダウンな（ゴール指向の）読み込みの過程の中で両辺を同一視する方法をとる。図3.13のbではトートロジーの両辺を同一のアトムに縮退させるので、この問題が起きない。

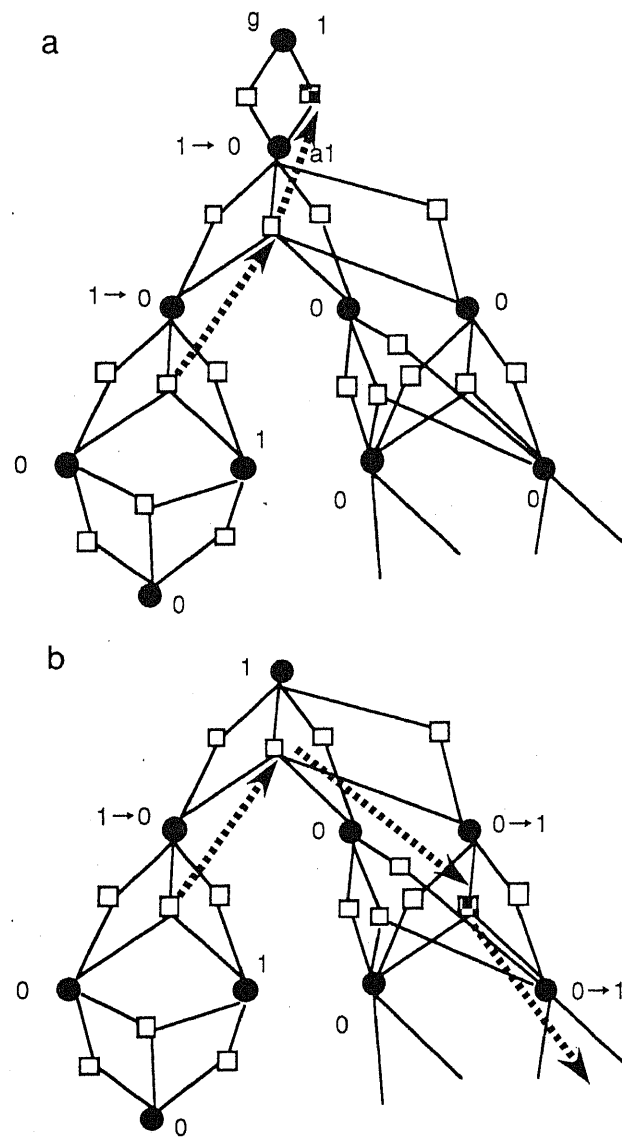


図3.13.宣言ルールでの推論失敗(a)とその解決(b).



ここで各要素仮説に重みを与えることも可能であるが、以下ではデフォルト値として各仮説の重みを1として計算した結果を基に考察する。この後初期フェーズに続く探索フェーズの計算中に、値が1となったアトムを順次書き出させる様になると、下のようなになった。

```
1, a1,  
1, a1, b4,  
1, a1, b4, b2,  
1, a1, b4, b2, c1,  
1, a1, b4, b2, c1, a3,  
1, a1, b4, b1, b2, c1, a3, c2, c3,  
SUCCESS
```

Improvement Done. SOLUTION SET: 1, a1, b1, b2, c1, c2, c3,

これを知識ネットワークの形で見てみると、図3.14の矢印の太い順にアトムが1となっていっていったことになる。最後に改良フェーズによってb4, a3が解仮説から除去され極小解（この場合は最小コスト）となる。これを見ると、ゴールを起点としてトップダウンに真理値1の範囲が広がっていく様子が分かる。同時に、矛盾制約の箇所は（水流が岩をよけて行くように）避けながら広がることも分かる。即ち、始めから真であることが分かっているゴールにより近いアトムが真であるをより強く確信するという戦略であると言える。

実数真理値は確率とは別の指標であるが、両者を一旦ここで同等と見做すとこの傾向は、ゴール（図3.15のノードG）に近いアトムは1に近い真理値に早く到達する、即ち、真であることを強く確信していることを意味している。これは、[Poo 93]においてルール  $b:-a$  を満たすアトムa, bの真となる確率  $p(a)$ ,  $p(b)$  の間に与えられた制約  $p(a) < p(b)$  と対応している。

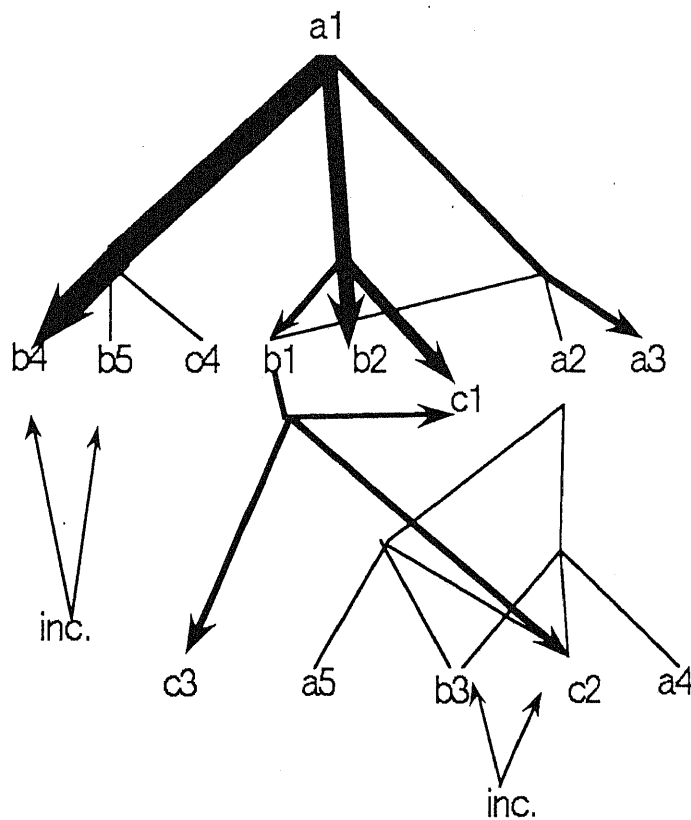


図3.14. NBP法の計算中に真理値が1となる順序（矢印の太い順）。

### 3.5.3. 問題のネットワーク構造と取扱い易さの関係

前項に見た様に、単結合のツリー状の知識ネットワークでは各ノード（アトム）の変化率、ひいては影響度をノード数に対して線形時間で求めることができる。しかし、一般には式(3-15)を解くことになり、計算時間はノード数 $N$ に対して $N\log(N)$ となる。勿論、これも多項式オーダーであるのでこのことはNP完全性とはなんら関係しない（実際、グラフのこのような複雑さ自体は厳密解を得るには指数オーダーの計算時間が必要である

ことと密接に関係するのであるが)。しかし、行列演算が必要となる理由は単結合以外の場合を考慮したためであることを、この項で示す。

即ち、バブル伝播ネットワークは次の様な性質を有する。

定義：内向きの支配ベクトルが2つ衝突する扇状の間隙を衝突点と呼ぶことにする。

命題：基本サーキットを $m$ 個含む様な知識ネットワークにおいて、■と○だけからなる状態で付与手順(図3.6)に従って付与した支配ベクトルの衝突点の個数は $m-1$ 個である。

証明 ■が $N_1$ 個、○が $N_2$ 個、含まれるとする。このとき、支配ベクトル付与の手順2から、■に吸収される矢印の先の数  $A$  は次の式で表すことが出来る。

$$A = \sum_{i=1}^{N_1} (C_i - 1) = \left( \sum_{i=1}^{N_1} C_i \right) - N_1. \quad (3-18)$$

手順2の■の矢印放出の条件をそのまま式としたのが左辺で、 $C_i$ は $i$ 番目の■ノードに隣接する枝数である。右辺はこれに移項だけのものであるが、第1項がグラフ中の枝の総数を表していることが分かる。■と○だけで連結されている区域においてバブルの個数は構造変数の個数に等しいので、次の式から $N_1=N_2$ が成り立つ（右辺はバブルの個数、左辺は構造ノードの個数）。

$$N_1 + \text{Number of } (\bullet) = N_1 + \text{Number of } (\bullet). \quad (3-19)$$

よって、衝突点の個数 $(A-N_1)$ は、”枝の総数－ノードの総数 $(N_1+N_2)$ ”で表すことができる。この式はEulerの定理における基本サーキットの数より1だけ少ない。(証明終)

この命題から、衝突点の個数はcycle cutset（グラフを単結合とするために必要な切れ目）数に満たないことが分かる。即ち、単結合のグラフでなく■と○だけからなる状態では、支配ベクトルを全ての枝には付与できないか、もし付与できたとしても矢印がループを描いてしまうので単結合の場合のように変化率等の計算を行うことはできない（従って、○と■だけからなるグラフは必ず閉グラフを含むことを同時に示したことになる）。従って、このような場合には別の方策として行列演算が必要となったのである。

このことだけからもグラフ表現の複雑となる問題の取扱いのしにくさが理解できるのであるが、実は行列演算によってもその壁は本質的に越えることはできない。この様な、手段に拠らない本質的な困難さを前提として、どの様な構造の問題が果たしてどの程度まで簡単に解けるかという解析もなされたが、バブル伝播ネットワークの解析によってこれと同様の知見を得る試みは第4章後半に譲ることとする。ただ読者が、ここに示した様に問題の取り扱いにくさには視覚的なモデル（目に見えるネットワークの構造からたちどころに予想できる性質）が存在し、日常「ややこしそうだ」と感じる問題に直面した時の印象と深く結びついていることを把握して頂ければこの節の目的は遂げたことになる。

### 3.6. 実験と評価

#### 3.6.1. 実験とその結果

NBP法をSun Sparc Station 10上に実装し，推論時間を計測した．

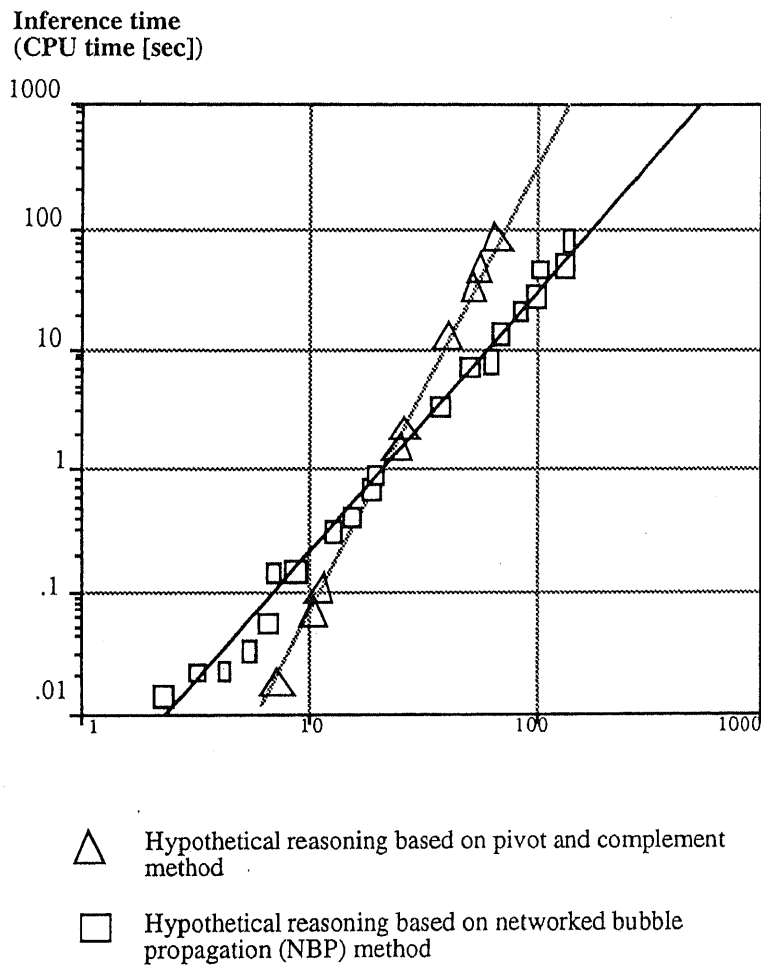


図3.16. NBP法による推論時間の実験結果

ここでは，あらかじめ真偽の分かっているアトムを含まない知識ベースを扱った．単一のホーン節の条件部のアトム数は $3 \pm 2$ 個，どのアトムも知識ベースでの出現数の上限は10としている．図3.16の各ドット（□）は，

要素仮説数の等しい10通りの問題の推論時間を平均した結果である。PC法を用いた推論時間は、要素仮説数 $N$ に対してほぼ $N^4$ のオーダーとなっているが、NBP法では推論時間がおおよそ $N^2$ のオーダーとなっている。また制約充足解のコストの順位を確認できる範囲では、得られた解仮説は制約充足解のうちコスト（式(3-6)）が最小から3つまでに入るものである（全解探索には指数オーダーの計算時間がかかるので、要素仮説数が60以下の問題に限り全ての充足解の評価関数を求めた）。また、実際の計算では図3.1のStep9で探索失敗となるのは、全てのケースの1%以下となった。

### 3.6.2. 推論時間と解仮説に関する考察

本論文ではホーン節集合から連立不等式への変換法が[岡本93]とわずかに異なるのでこれとの解との一致は保証されないが、推論の結果得られる0-1整数解はほぼ同一で実用上最適解に十分近い。この精度の近似解を、可能な要素仮説数 $N$ に対しPC法を用いて得られる $N^4$ オーダーに比較して、約 $N^2$ オーダーの計算時間で達成したことは実用上の意義を持つといえる。NBP法の高速化は、4.1節に仮定した様に伝播パスが知識の規模に大きく依存しないとすれば説明できる。

即ち、PC法の場合、一回の掃き出しにつき行列演算の計算時間は $N^2$ オーダーかかる。この計算は、各ターゲット候補（ $\propto N$ 個）からのバブル伝播を比べつつ多くが■と○の交換（ $\propto N$ 回）で進められるので、探索フェーズには $N^4$ オーダーの時間が必要となる。一方、NBP法では伝播パス内だけで一回の掃き出し操作に当たるバブル伝播が行われる。従って、3.5.1節の仮定が正しければ、その所要時間は $N^2$ オーダー（PC法）→定数オーダー（NBP法）と短縮でき、上の計算時間が説明できる。

## 本章の付録

### 付録1：同じ枝の逆向きの影響度の積=-1の説明

図3.10を用いて説明する。○→■の向きの伝播率の定義は4.1節で与えたので、ここでは $p(V, P)=1$ のとき逆方向の伝播率を $p(P, V)=-1$ とする根拠を示す。まず、 $s_2$ が $\text{var}(S_2)$ 変化すると $v$ は式(2-13)より

$p(V, S_2)\text{var}(S_2)$  変化しようとする。しかし、バブル伝播後も非基底に留まるノード $V$ の値 $v$ が（上限か下限のまま）変化しないように、ノード $P$ の値の変化で $v$ のこの変化を打ち消す必要がある。このことは、

$$p(V, P) \text{var}(P) + p(V, S_2) \text{var}(S_2) = 0, \quad \text{即ち}$$

$$\text{var}(P) = -\text{var}(S_2) p(V, S_2) / p(V, P)$$

と書ける。つまり、 $p(P, V) = -1 / p(V, P)$ とすれば

$$\text{var}(P) = \text{var}(S_2) p(V, S_2) p(P, V)$$

とノード $P$ の値の変化量が計算できることになる。

### 付録2：図3.9の手続き終了時での $\text{rank}=\text{rank}'$

$S_1, S_2, \dots$  (○) と  $T_1, T_2, \dots$  (■) をそれぞれ●と□で置き換えたグラフに図3.6の手順で支配ベクトルを施すと、■を□に又は○を●に変える（手順2, 6-2'）ことなく図3.9と同一の矢印を得る。即ち、置き換えたグラフでもバブルの個数は構造ノード数に等しい。従って、○から●, ■から□への置き換えの数、即ち $\text{rank}, \text{rank}'$  は等しいことになる。





NBP法では活かせなかったこの様な問題固有の構造的な簡単さをも活かすことができれば、更なる高速化が図れよう。本章の研究は、まず以下のことを証明する。即ち、結論から先に示せば、

- 1) 知識ネットワークが単結合となる場合を部分として含む問題(後述するクラス3)の厳密な最適解を多項式時間で求めることができる。
- 2) クラス3より複雑な問題の準最適解を求める場合も、推論とともにネットワークの状態がクラス3に収束するので効率的推論が達成できる。

クラス3は知識ネットワークの1)はじめに与えられた構造と、2)推論中変化する状態によって与えられ、推論とともに問題が簡単化されることも理解できる。これは、単体法 (Simplex Method) やKarmarker法[Kar 84]といった線形制約充足問題 (Linear Programming) の解が0-1整数解にもなるというケースを問題の構造から示したものであり、理論的な点でもAIとORの橋渡しを得たものである。

一方、Freuder, Dechterなどによるバックトラックなし探索の十分条件は、本章で述べる新手法のNBP法で多項式時間で解が得られ条件よりもさらに広いことも明かとなった。彼らの理論を厳密に表現するならば、計算時間には制約ネットワークの密度に近い、"グラフのregularな幅" に対しては指数オーダーの因子がかかるが、グラフの大きさに対しては多項式オーダーに過ぎない。即ち、最も複雑な完全グラフ (全ての2ノード間に枝が存在する) ではノード数即ち変数の個数に対して指数オーダーで計算時間が増加するが、グラフの幅を一定にして拡大してもその大きさには多項式オーダーでしか依存しないのである。本研究でもここまでの一般化を目指したが、クラス3以上には拡張できない原因が、根本的なアプローチの差であることが分かった。しかし、その同じ原因は、本手法の計算時間をグラフの幅に対しても多項式オーダーとする効果を發揮している。

## 4.2. 支配ベクトルとネットワーク構造の関係

単結合のBPNには、定理1（証明は本章の付録）に示すように全ての枝に支配ベクトルを付与できる。

定理1：BPNが有限な単結合グラフのとき、上の手続きで全ての枝に支配ベクトルの矢印を施すことができる。

知識ネットワークの構造が単結合のとき、これを書き換えたBPNも又単結合となるので、上の手続きで全ての枝に支配ベクトルの矢印を施すことができる。これは問題が特に簡単な場合であることを考えると、支配ベクトルの付与が問題の複雑さと密接に関係することが推察される。バブルが伝播すると上の手続きによって支配ベクトルも更新されるので、問題は推論が進むにつれて平易化することになる。

## 4.3. 新しい制約不等式を用いたネットワーク化バブル伝播法

### (I-NBP法)

#### 4.3.1. 新しい制約不等式

NBP法において厳密な最適解計算は単結合など特に単純な知識構造に限っても保証できない原因は、背景知識と等価な不等式制約の与え方にあった。即ち、真理値空間内で不等式制約を満たす多面体内の格子点（全ての構造変数値が0又は1となる点）を求めて多面体の縁沿いにトレースするとき、格子点以外の頂点が多いと効率が悪い。実際、不等式(3-3)、(3-4)を満たす多面体には、 $n=2$ の場合の頂点 $(X_1, X_2, Y) = (1, 1, 0.5)$ の様に格子点以外の頂点が多い。

そこでI-NBP法では、別の不等式で背景知識を表すことにより一層の高速化を狙う。即ち、ルール(3-1), (3-3)を、制約を充足する多面体の頂点の多くが格子点となる制約不等式(4-1), (4-2)に書き換える。

$$\left(\sum_{i=1}^n x_i\right) - n + 1 \leq y, \quad (4-1-1) \quad 0 \leq y \leq x_i. \quad (4-1-2)$$

$$y \leq \left(\sum_{i=1}^n x_i\right), \quad (4-2-1) \quad x_i \leq y \leq 1. \quad (4-2-2)$$

#### 4.3.2. I-NBP法のアゴリズム

図4.2(b)図はこれらの不等式をBPNの形に記述したものである。

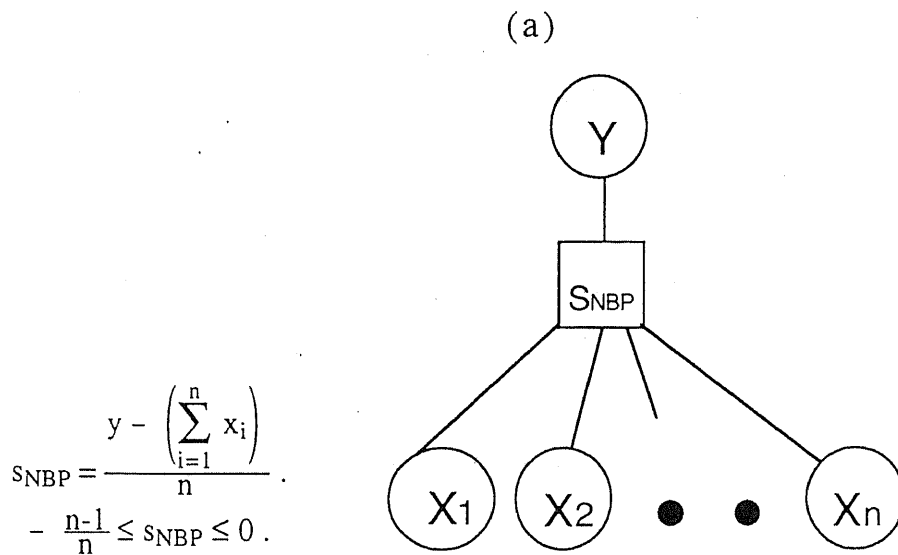


図4.2.Y ← X1, X2, ...Xn. に対する旧 (a) → 新 (b) BPN.

(b)

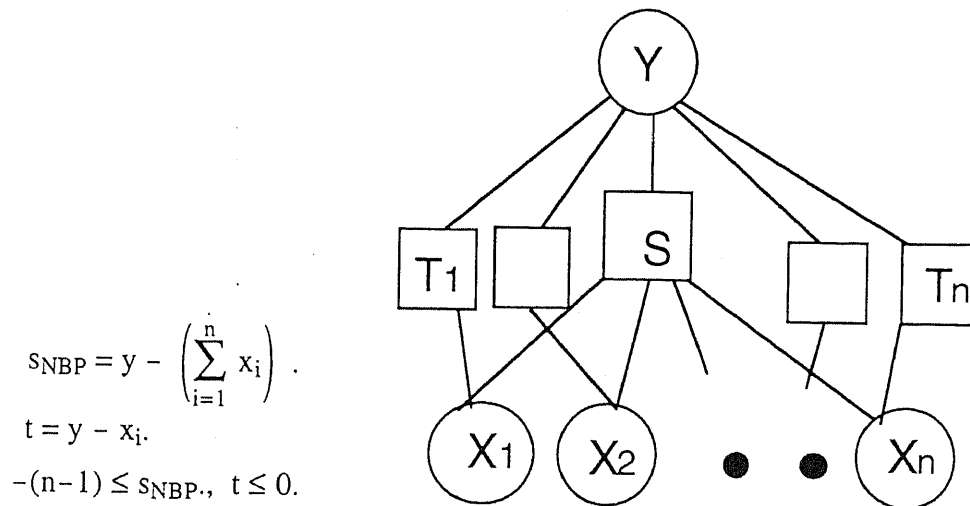


図4.2.Y ← X1, X2, ...Xn. に対する旧 (a) → 新 (b) BPN (続き) .

I-NBP法では、NBP法の操作をこの改良型のBPN上で行う。下にI-NBP法の手順を示すが、Infeasibility(IFと略記), Non-integer degree(NIと略記)はPC法と同義であるので、[Bal 80]を参照されたい。又、バブルを■ノードから○ノードへ伝播させることを(○, ■)の交換等と表記し、第1項を交換相手、第2項をターゲットと呼ぶ。交換相手はターゲットから支配ベクトルに逆らわない向きに枝をトレースして迎れる白いノードから、バブル伝播によるターゲットの値の変化が最小となる様を選ぶ(第3章)。

### I-NBP法のアルゴリズム

#### <初期フェーズ>

1. 与えられた制約(不等式(4-1), (4-2))から0-1整数条件をはずした線形計画問題を単体法によって解く。この結果を初期状態とする。

#### <探索フェーズ>

2. (○, ■) の交換が可能ならば, そのうちコストの最小となるものを実行して4へ, さもなくば3へ.
3. 3.4.2のStep3の改良を前提として, (□, ■) か (○, ●) の交換でNIが減少するものがあればこれを実行して4へ, なければ5へ.
4. 値が0か1となった○ノードを●ノードとする. このときバブルの数が増加するので, 接する枝の全てから支配される■ノードを□ノードとする. ○ノードがなくなれば10, さもなくば2へ.
5. 丸め (四捨五入) によって0-1整数解が得られる場合, これを得て10へ. それ以外は6へ.
6. IFを最小とする (○, ■) の交換を実行する.
- 3.4.2のStep7-9の改良を前提として,
7. 1つの●の値の補数をとることによってIFが低くなる場合, 最もこれを低くするものを実行して8へ. それ以外は9へ.
8.  $IF > 0$ ならば7へ.  $IF = 0$ で四捨五入により0-1整数解が得られればこれを得て10へ. それ以外は2へ.
9. 2つの●の値の補数をとることによってIFが低くなる場合, 最もこれを低くするものを実行して8へ. それ以外は失敗として終了.

#### <改良フェーズ>

10. 第3章のNBP法の改良フェーズと同一とする.

支配ベクトルがネットワークを網羅するにつれて手順4で●が増加する (後述の定理2参照) ので, 最終状態に高速に収束する. 図4.3の例では, 状態1から手順2の一回の交換で支配ベクトルが全ての枝を網羅する状態2へ移行し0-1整数解を得, 手順4で改良フェーズへ移行する (図3.5比較参照).

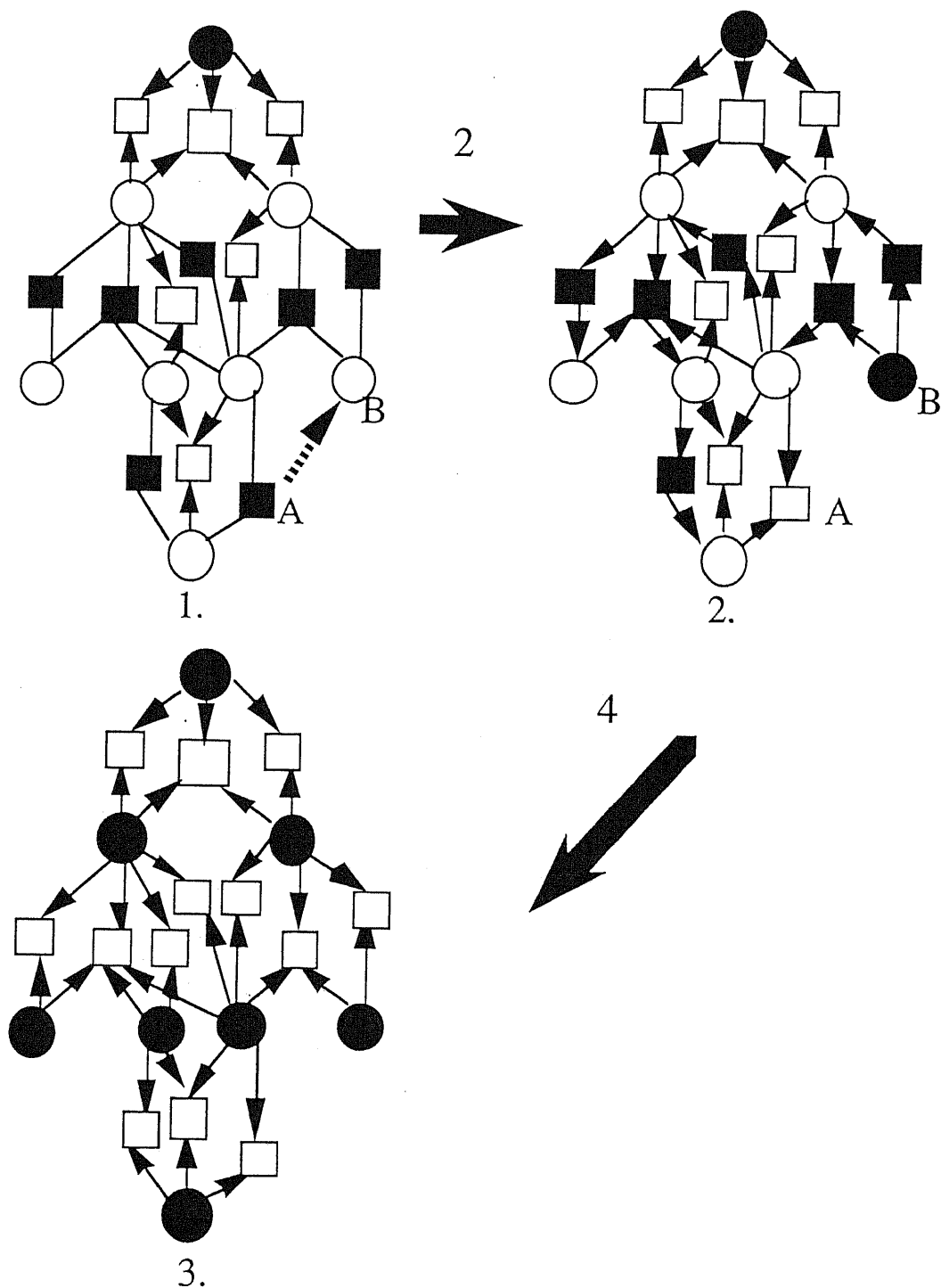


図4.3. 図3.4の例に対するI-NBPのプロセス.

上の番号はI-NBPの手順, 下の番号は状態遷移.

## 4.4.1-NBP法における計算量

### 4.4.1.1-NBP法に基づく計算量クラス

I-NBP法では、支配ベクトルの付与と計算量の間の密接な関係を理論的に示すことが可能となる。支配ベクトルは推論中、知識ネットワークの構造に影響されながらBPNの状態とともに変化するが、このような知識構造の反映と変化への順応性がI-NBP法の高速推論の基礎となる。

先ず、仮説推論の問題を次の3クラスに分類する。

クラス1: 知識ネットワークが単結合となるクラス。

クラス2: NBP法においてBPNの全ての枝に支配ベクトルを付与可能なクラス。

クラス3: I-NBP法においてBPNの全ての枝に支配ベクトルを付与可能なクラス。

定理1からクラス2 $\supset$ クラス1である。本節では、3クラス間の包含関係を更に知ることによってI-NBP法の高速推論を理論的に裏付ける。以下では次の仮定を真とする。

仮定1. ルールの前件部にはアトムが必ず2つ以上存在する。

即ち、 $A \leftarrow A$ のようなトートロジーは知識ベースから全て削除する。又、他のホーン節とヘッドBを共有せず、かつ条件部が単一のアトムである $B \leftarrow A$ のようなルールは、知識ベース中のAを全てBに書き換えてに削除する。こうして、仮定1が成り立つように知識ベースを自動的に変換することが可能である。付録に示す様に、この仮定1の下で次の補題が成り立つ。

補題1:全ての構造変数値が $[0, 1]$ の範囲にあって不等式制約(4-1), (4-2)を満たすとき, 値が0又は1の(隣接枝数-1)ノードから支配ベクトルを受取る■ノードの, 残る1つの隣接ノードの値も0又は1となる.

補題2:クラス3のBPNにおいて, 支配ベクトル付与の手順を用いなくても全ての枝に支配ベクトルを付与できる.

補題1と補題2から定理2が導かれる.

定理2:クラス3のBPNにおいて, 全ての構造ノードの値は0又は1となる.

証明要約: ●ノード(0又は1の値をとる)を始点として支配ベクトルの向きに辿ることができる全ての○ノードの値は, 補題1より0又は1である. 補題2より, この際BPN上の全てのノードを通ることができる(証明終).

系:クラス3に属さないネットワークでも, ●ノードから支配ベクトルを辿ることによって得られる構造ノードは0-1整数値をとる.

証明:定理2の証明前半で示された(証明終).

通常, I-NBP法の手順2と5で推論が進むにつれ●ノードが増加するため, 支配ベクトルは推論が進むと密になり, ついにはネットワークはクラス3の状態となる(探索フェーズの最終状態では全ての構造ノードが●ノードとなり, 明らかにクラス3に含まれる).

ネットワークの構造によっては, 初期状態で既にクラス3の状態となっている. 定理3に示す様に, 知識ネットワークが単結合となる場合はこれに含まれる.



定理3：単結合の知識ネットワークに相当するI-NBP法のBPNは全ての状態でクラス3に属する（証明は付録）。

定理2と定理3から，次の系が導かれる。

系：単結合の知識ネットワークに相当するI-NBP法のBPN上の全ての構造ノードの値は初期状態でも0又は1となる。

初期フェーズはNP完全な0-1整数計画問題ではなく線形計画問題を解くことになるので，知識ネットワークが単結合の場合は多項式時間で厳密に最適な0-1整数解（解仮説）が得られる。以上を以下にまとめる。

- 1) クラス3（ $\supset$ クラス1）の問題は，多項式時間で厳密解が得られる。
- 2) クラス3に属さない問題でも，定理2の系に示した一部のノードは0又は1の値をとることが保証されるので，準最適解を得る時間が短縮される。

#### 4.4.2.多項式時間での厳密解計算の十分条件 " $\supset$ " 単結合

前述のように，知識ネットワークが単結合グラフで表される場合の仮説推論は，多項式時間で厳密な最適解を得ることが可能である。しかし，定理2と定理3によれば，単結合とは多項式時間で解ける問題の中でも特殊なクラスに過ぎない。この点は[Fre 82]に示されたバックトラックせずに探索できる十分条件（次節参照）も同様であるが，我々の定理3には次の特徴がある。

- 1) [Fre 82]に基づく[Dec 87, 88, 91]などでは，厳密な最適解を求めるので，最悪の場合，指数オーダの計算時間を要する。これに対し，我々の理論を裏付けとするI-NBP法では，近似によって多項式時間で準最適解を

求めることができる。

- 2) 定理2の条件部は、推論中変化しない推論ネットワークの構造と、推論とともに変化する支配ベクトルによって与えられるので、推論とともに問題が簡単なクラスに近づいて行く変形を実現できる。

#### 4.5. 従来理論との比較

ここで知識ネットワークの構造と計算量との間の相関を解析したことは、AIとORの橋渡しの中から創った実数計算による推論の特徴を浮き彫りにしたと言える。人工知能の一分野としての論理の世界において、一般的な仮説推論の計算量はNP完全または困難であるが、特殊な場合にはこれより容易であることが確認されたのは[Sel 90, Kau 90]あるいは[Byl 91]などの研究による。

より具体的に述べると、[Sel 90]では様々な場合の仮説推論の計算量が解析され、例えば背景知識としてホーン節集合 $\Sigma$ 、仮説集合 $H$ を与えて一つの説明を求める問題はNP完全であることを示した。しかし同時に、 $H$ が限定されなければ多項式時間で解を得ることができるという事実も述べられている。これは、ゴールを起点としてトップダウンに知識ネットワークをトレースし、葉（末端）のアトムにたどり着く前までにトレースした部分的な知識だけを用いた仮説推論を行う場合に当たる。単結合の部分ネットワークをこのトレースによって切り出すことを想定れば、本論文の定理1の結果と完全に一致する。又、[Byl 91]においては、仮説集合 $H$ を限定しない場合でも背景知識 $\Sigma$ に演繹の連鎖を持たせず、互いに矛盾する要素仮説の組み合わせを指定してその組を解仮説が包含することを禁止した場合の仮説推論がNP完全であることが示されている。これに対して、矛盾する要素仮説の組が存在しないのであれば多項式時間で

解が得られる。極端な言い方をすれば、この場合には全てのアトムを真としても許容されるのであるからこの結論はごく当然である。

これらの解析は、DowlingとGalliarの結果[Dow 84]とも関連づけることができる。彼らは、制約がホーン節だけからなる制約充足問題が多項式時間で解けることを、その単純なアルゴリズムと共に示した。ここで、

”ホーン節だけからなる制約充足問題”と、

”背景知識がホーン節だけからなる仮説推論”

とは全く別の問題であることを注意しておきたい。例えば、ホーン節

$a:- b, c.$

は、 $a$ が真であることを証明するには  $(a, b, c) = (\text{True}, \text{True}, \text{True})$  が必要となるが、本来論理式として見ればこれは

”もし $b$ かつ $c$ ならば、 $a$ である” ことを意味しているのであって、

”もし $a$ ならば、 $b$ かつ $c$ ” ことは意味していないのである。

(この点は、本論文が命題論理版で用いている制約不等式に上限と下限の両方を指定した理由でもある。[Dha 90]において示された様に、一般に論理制約としての節を不等式制約として書き直すには、例えばホーン節  $a:- b, c.$  を ”  $a + (1-b) + (1-c) \geq 1$  ” とする様に不等式は上限か下限の片方のみを指定すればよい。しかし、証明に用いる文脈の一部としてホーン節を用いるには完備性、即ち上に述べた様にヘッドを他のホーン節と共有しないホーン節は逆も真であることが要請される。この逆の制約を含めた記述が[岡本93]および本論文の不等式制約の与え方となっている。)

従って、Selmanの考えた多項式時間の仮説推論の様に、知識ネットワークのうちで推論の対象となったあるアトムが真で、その条件アトムが全て偽であったとしても単なる制約充足問題では許容されるのである。即ち、この特殊に定義された問題の解は、Dowlingらの制約充足問題の解ともなっているのである。ということは、以上の従来研究によって多項式

時間で解けると分かったのは単結合の制約ネットワークで表わせる制約充足問題に限られることになる。

一方、以下に述べるFrueiderからDechterに到る一連の研究（後述）では、図4.4の様な場合にはSeries-Parallelという範疇に属するprimal graphで書き直すことができるので、この問題は多項式時間で解ける問題に分類される。[Fre 82, Dec 88] は現在までのところ、ネットワークの形状を基に計算量を評価する上では、多項式時間で解を得ることのできる問題の十分条件として最も緩いものである。言い換えればこれらの研究は、簡単に解くことのできる問題の具体的な構造を与え、広い範囲の問題をこれらに近付けて解くという方法論の基礎を作ったといえることができる。

先ず、Freuderによるバックトラックなし探索が可能な問題の必要条件を示す。

[Freuderの定理] strong-d-i-consistentな制約ネットワークのCSP（制約充足問題）は、グラフの幅（width）が $i$ よりも小さい時バックトラックなしに解くことが可能である。

ここで、**d-i-consistent**とは、順序 $d$ で $i-1$ 変数への任意の代入値に対して、 $i$ 番目の変数に対して制約を充足する値が少なくとも一つ存在することである。更に、**strong-d-i-consistent**とは、 $i$ 以下の全ての自然数 $j$ に対して**d-j-consistent**であることをいう。又、あるノードの親ノード（枝で隣接する順序が高いノード）の個数をそのノードの幅といい、あるグラフ中のノードのうちで最大の幅をそのグラフの幅と呼んでいる。

即ち、この定理は、簡単に言うと次の様なことを意味している。

「ある順序 $d$ において、グラフの幅（ $k$ 個）だけの個数のノードを順位の低い順にとる。これらのノードにどのような値を代入しようと、それに対して $k+1$ 番目のノードに制約を充足する値を代入できる

( $k \leq i$ で、かつstrong-d-i-consistentよりこのことが成り立つ)。この様に制約充足を保持してボトムアップに代入して行くことができるので、バックトラックの必要がない。」

但し、この定理を含めてこの節では、バブル伝播ネットワークと断らない場合は異種の制約ネットワークを用いる。即ち、各変数を各ノードで表し、同一の制約に出現する変数同士を枝で結合して表したprimal graph (図4.4参照)を用いて議論し、バブル伝播ネットワークを用いる場合にはそれを随時記す。

この定理をうまく利用することによって、高速解法を得たのがDechterとPearlによる、Adaptive Consistencyである[Dec 88]。これは、元の問題を変化させてFreuderの十分条件を満たすようにしてから解を得るものである。即ち、以下の操作を行う。

#### Adaptive Consistencyの操作 (より形式的には[Dec 88]参照)

1. 制約ネットワーク上である順序 $d$ を決める。
2. 順位 $d$ の上で、ネットワーク中の順位の低い順に、全ての親ノード（枝で隣接するを順位が高いノード）を互いに新たな枝で結合し、自分の子ノードに制約充足値を与えられないような親ノードの代入値の組み合わせを除去する（親ノードの代入値に関わらず、少なくとも一つ子ノードに制約充足な代入値が存在するようにする）。
3. この処理を順序の最高位まで到達したら終了。さもなければ2へ。

この操作によって、その後の探索を上記の様にバックトラックなしに行えるようになる。この時、グラフがregular width- $k$ であるとは、新たな結合によってグラフの幅が $k$ となる場合をいう。

グラフがregular width- $k$ の時，そのCSPを解くのに要する時間は $2^k$ オーダーとなる[Dec 88]．これは，各ノードに対してその親ノードの集合の間に $2^k$ 回の代入値の整合性の照合を行うことによる．制約ネットワークが単結合の場合には，木の端点から順に順位を付ければ枝は全く増えないので多項式オーダーで解を得ることができる．Arnborgは， $k=2$ の場合にこのような性質を持つネットワークを判別し，順序を与える手法として次の手順を考案している[Arn 87]．

#### Arnborgのグラフ変形操作

1. 次数が2以下で最小のノードを取り除き，その元の隣接ノード間を全て枝で結合する．
2. グラフが空となれば，それで操作を終了し，ここまでに削除した順序を順序 $d$ とする．即ち，始めに取り去ったノードが最子孫である（Arnborg の順序の定義とは反対である）．
3. それ以上ノードを削除できない時，グラフはregular width- $k$ よりも複雑であったことになる．
4. 2と3のいずれにも該当しない場合には1へ．

特に，regular width-2の場合をseries-parallelという．問題がこの特徴を有する場合は，Adaptive consistency の手順によって多項式時間で厳密解を得ることができる．但し，ここで厳密解というのは最適解のことではなく，制約を必ず充足する解である．その様な解を一つを見つけることは，最適解を見いだす探索問題と同様に指数オーダーの時間を要する問題である．制約充足解があれば準最適な制約充足解を必ず見いだす手法は，制約充足解を見いだすこと自体保証できない近似解より有意に厳密さにおいて勝っていることから厳密解と呼ぶことにした．

平易に言えば，Frueider-Dechterの路線はアブダクションや論理制約について個別になされた研究よりも一般的な十分条件を求めていたのである。

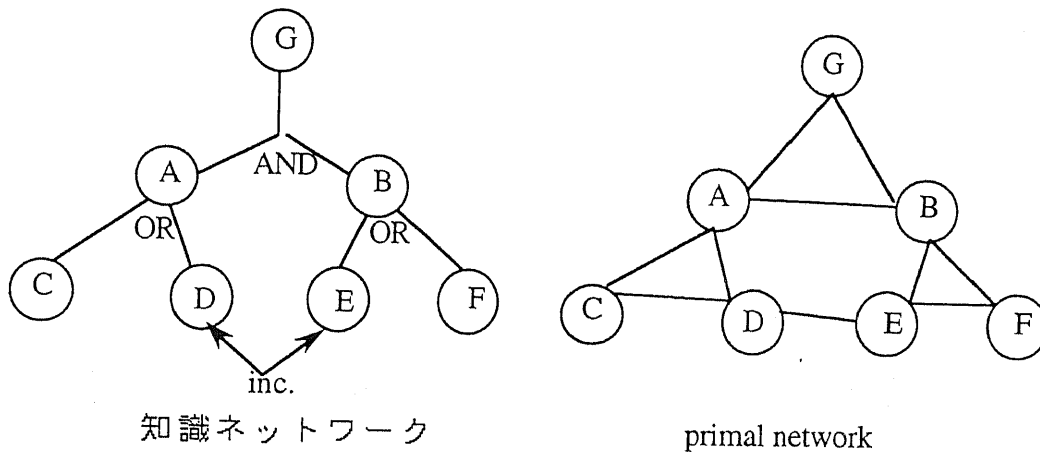


図4.4 要素仮説間の矛盾を含むが，多項式時間で解ける仮説推論の問題

この歴史の中でバブル伝播ネットワークの占める位置を考えると，人工知能サイド，即ち状態に依存せず多項式時間で厳密解が得られるのは単結合までである（Series Parallelの場合には，補数計算だけは省くことが可能であるが = 付録補題5参照）。では，NBP法とAdaptive Consistencyの本質的な差異はどこにあるのか。

筆者は当初，この差異を見落としてNBP法でもAdaptive Consistencyと同じ問題の厳密解を多項式時間で得る能力を有するという仮説を立て，これの証明に長い期間の努力を掛けた。例えば，はじめにグラフの一部を0-1整数値に固定し（図4.5のF），これを起点としてバブル伝播を繰り返すという手法を考えた。この手法では，ルール中央のSNBPにあたるスラックノードが，他の■を含むルール内に伝播しない（補題3～4 = NBP法に

にとって重要な特徴であるので、付録に証明とともに示す) ことから図のH  
 を介して0-1整数値がGにも及んで行き、全てのノードの値が0-1整数値に  
 単調に接近することを利用する。しかし、値ループがグラフに含まれる  
 などの場合に、その分岐点では0-1整数値だけのサブグラフへのバブル伝  
 播が継続できなくなる。

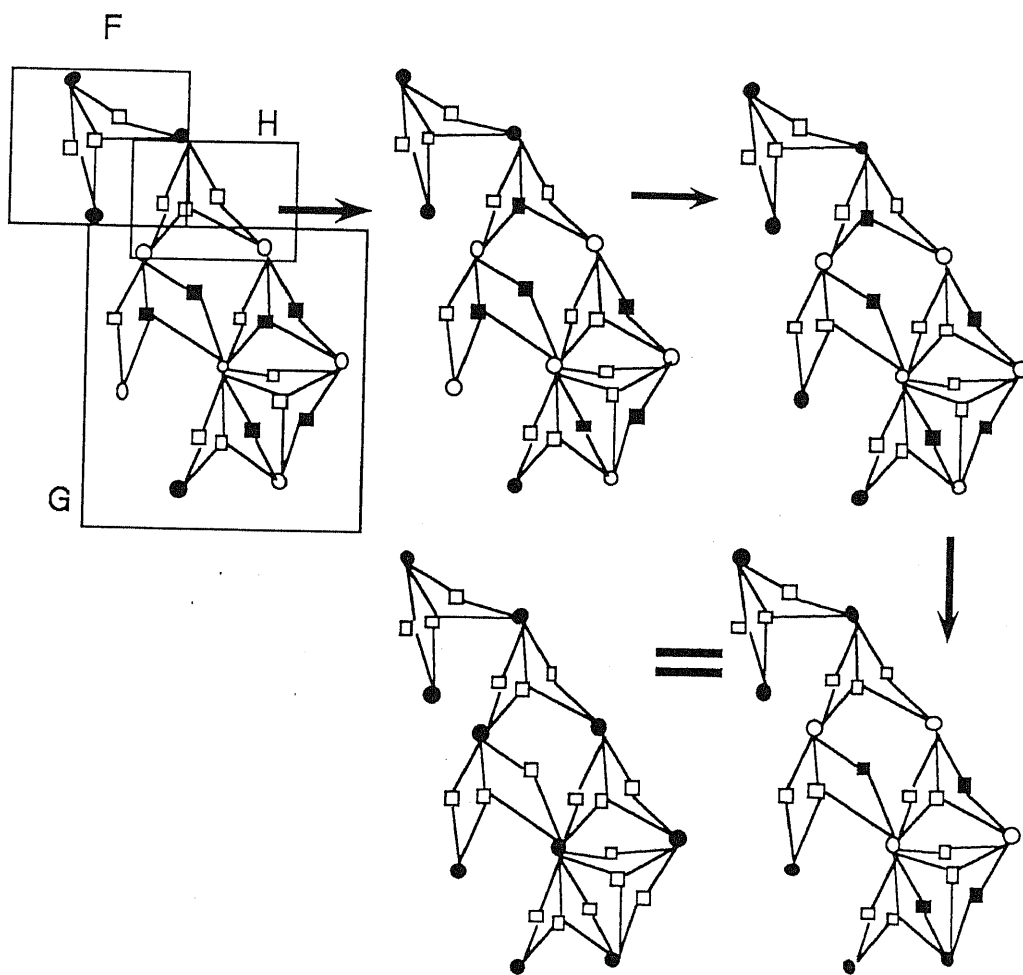


図4.5. NBP法によってFreuder, Dechterと同等の性質を実現する一つの  
 案.



従って、新たに別の箇所に0-1整数を固定することになるので結局全体ではネットワークのサイズに対して指数オーダーとなってしまふ。

他にも数十種類の手法を仮定して理論構築を試みたが、結果として、次のようにこの仮説の誤りが見いだされたのである。

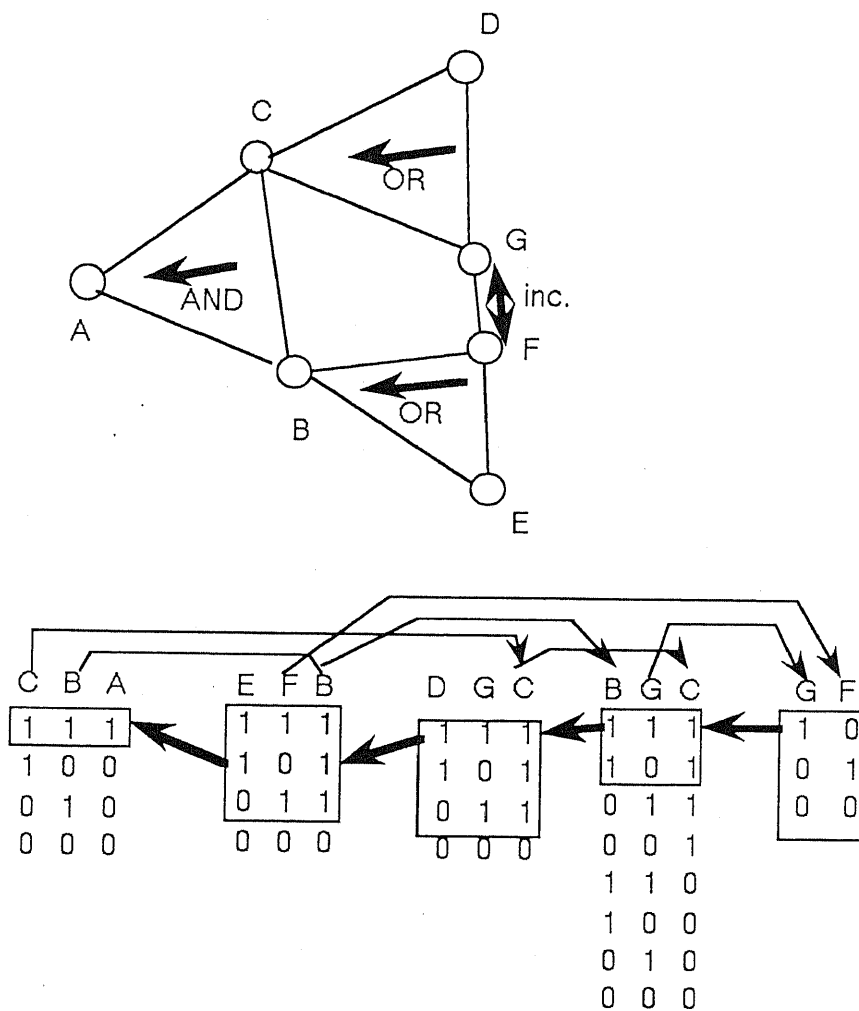


図4.6. Adaptive Consistencyにおける、可能な状態の並列保持。この後、  
下図の太い矢印の順で解を得る。

NBP法とAdaptive Consistencyの本質的な差異は、連続系と離散系の差であった。即ち、前者が連続値の実数真理値によって単一の状態の変更の繰り返しであるのに対し、後者は離散的な個々の可能な状態を並列に保持しているのがその差異である。図4.6にAdaptive Consistencyの実行例を示す。Aが真であることを証明する場合、 $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow F \rightarrow G$ の順にAmborgの操作を適用する。四角で囲った組み合わせが許される組み合わせとして矢印の様に（矢印の両端の変数を参照して）伝播される。これは、一操作を部分に絞り、その部分を取りうるあらゆる可能性を網羅しておくことにあたる。これらの可能性は、その後も否定されることがない単調な操作であるので、多項式時間のアルゴリズムとなる。

一方、NBP法においては、任意の時点でのネットワークの状態は連続実数で表される単一の状態である。これは、その後の値の二値化によっては実行可能解に到達不能な場合もある。従って、その場合にはバックトラックと同様の役割を担う補数計算を行う必要を取り去ることは一般にはできない。即ち、離散的で解の一部となることが約束されている状態を保持する操作と、連続値を準備する方法論とでは厳密解計算では問題の構造によっては前者が高速となるのである。

そこで、NBP法で行っているのは近似を取り入れることによってこの差を縮小あるいは反転することなのである。仮にI-NBP法の実行中に、実行可能な連続値の状態から0-1値にある構造ノードAが変化したところ制約違反となったとする。この時、Aの周囲にN個のスラックノードが存在していれば、最大でInfeasibilityがNとなる（伝播率が全て-1又は1であるので）。その後、このInfeasibilityが減少するようにここで違反したスラックノードの一つBを支配する●ノードCの補数をとると、一度について（やはりパスBC上の枝の伝播率が全て-1又は1であるので）BのInfeasibilityは1以上減少する（減少する場合のみ補数をとる = I-NBP法の手順を参照）。

即ち、一度制約違反が起きてからこれを実行可能状態に戻すまでの手順数の上限は、 $N$ の最大値、即ちグラフの最大出次数で与えられる。即ち、これがある構造ノードを非基底化する際の最大のバックトラックの回数に相当するのであるから、一箇所の矛盾回避を複数の状態保持で $2^N$ 回のステップをかけて行っているAdaptive Consistencyよりも高速な推論を近似的に達成できるのである。

例を図示してこれを説明すると、図3.13の例を実行中に出力した図4.7では、■の網点で示したスラックノードが制約を違反している。この後、ノード17(●で真理値が1のノードを示す)を反転(真理値を0と)し、改良フェーズにおいてノード23を反転(真理値を0と)した結果が図4.8である。この場合は、Infeasibilityが1から0へと減少している。

これらの議論から、この近似の度合いは、全体のサイズだけではなくグラフの複雑さにも依存することになる。一般に、近似の精度は速度とトレードオフの関係にあるので、速度はもまたグラフの複雑さに依存するが、今述べたことを考慮すれば、Freuderらのいうグラフの幅に対して指数オーダーではなく多項式時間に比例することが予想される。この実験としては、比較的小規模の問題に対して矛盾制約を増加させて計算時間を測定し、複雑さとの相関を見ればよい。図4.9は、図3.13の例に10個まで矛盾を増やした結果である。この結果は、ここでの予想を裏付ける一例となっている。いずれのドットも3通りの問題のうち成功例の平均時間であるが、スラックノードが55個以下までは2番目までの解を必ず得たが、60個以上の2ノードでは1ケースずつ探索に失敗している。密度(出次数)に対して多項式時間の速度と引替に、近似精度も次第に低下していることが分かる。

Bubble Propagation Network.

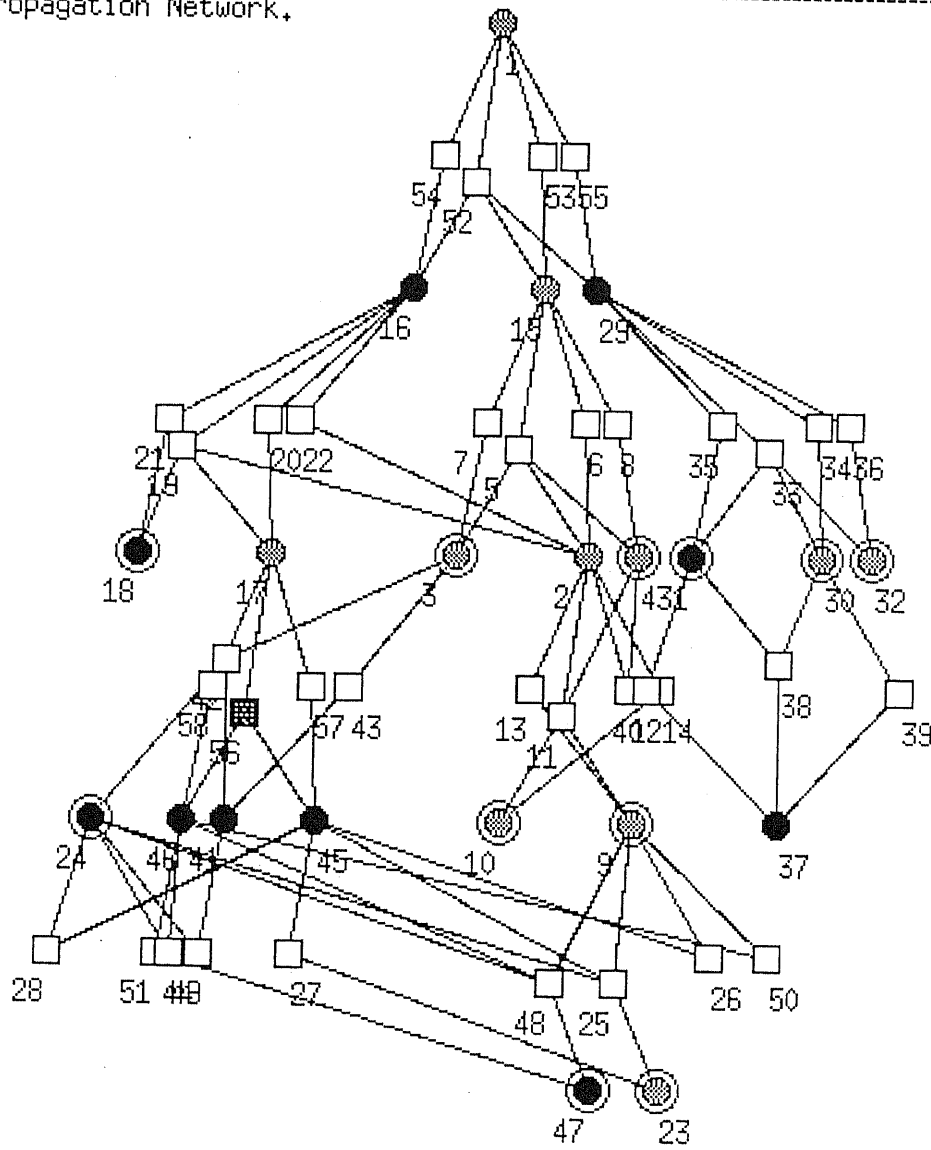


図4.7. 図3.12の例題のバブル伝播ネットワーク. ノード56が制約を違反している.

Bubble Propagation Network.

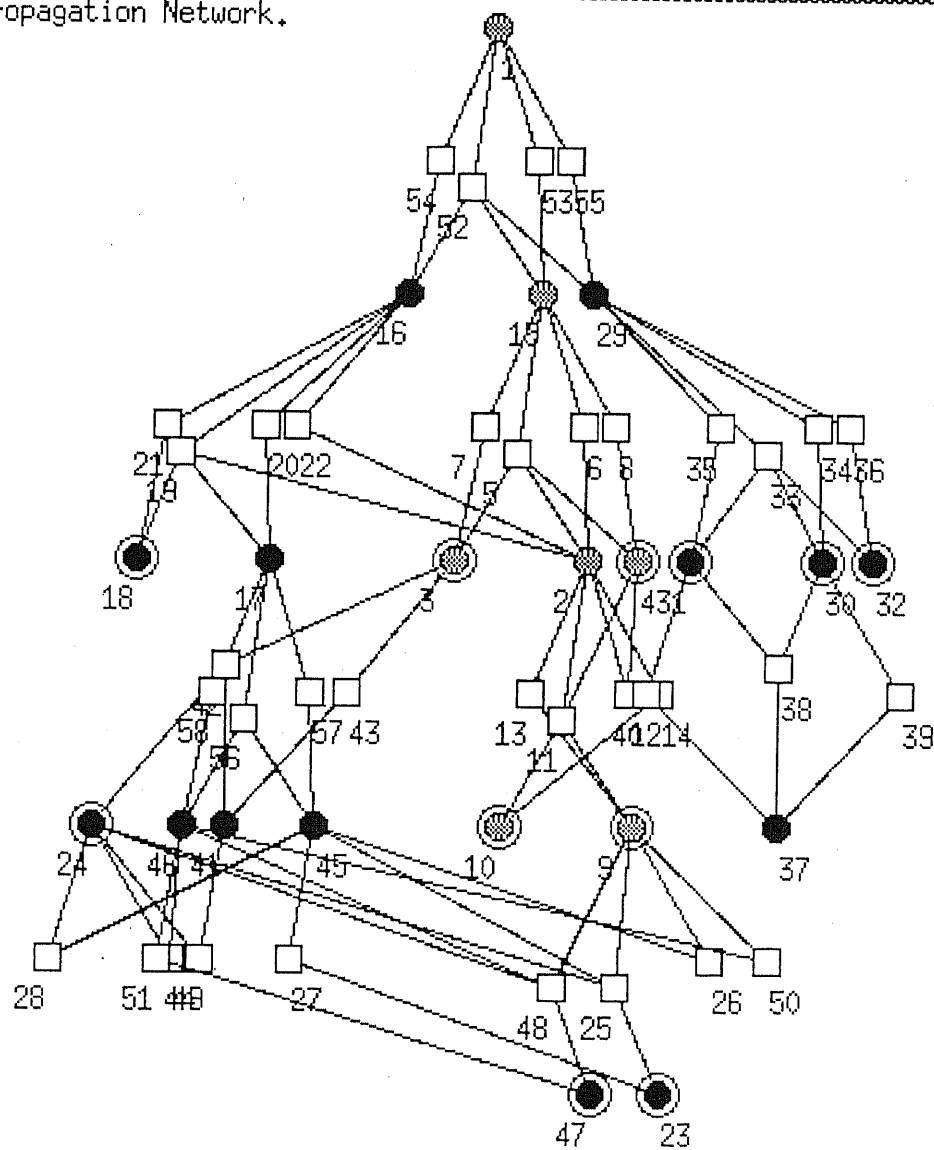


図4.8. 図3.12の例題のバブル伝播ネットワーク。ノード17を反転し、その後改良フェーズでノード23を反転している。

Inference time  
(CPU time [sec])

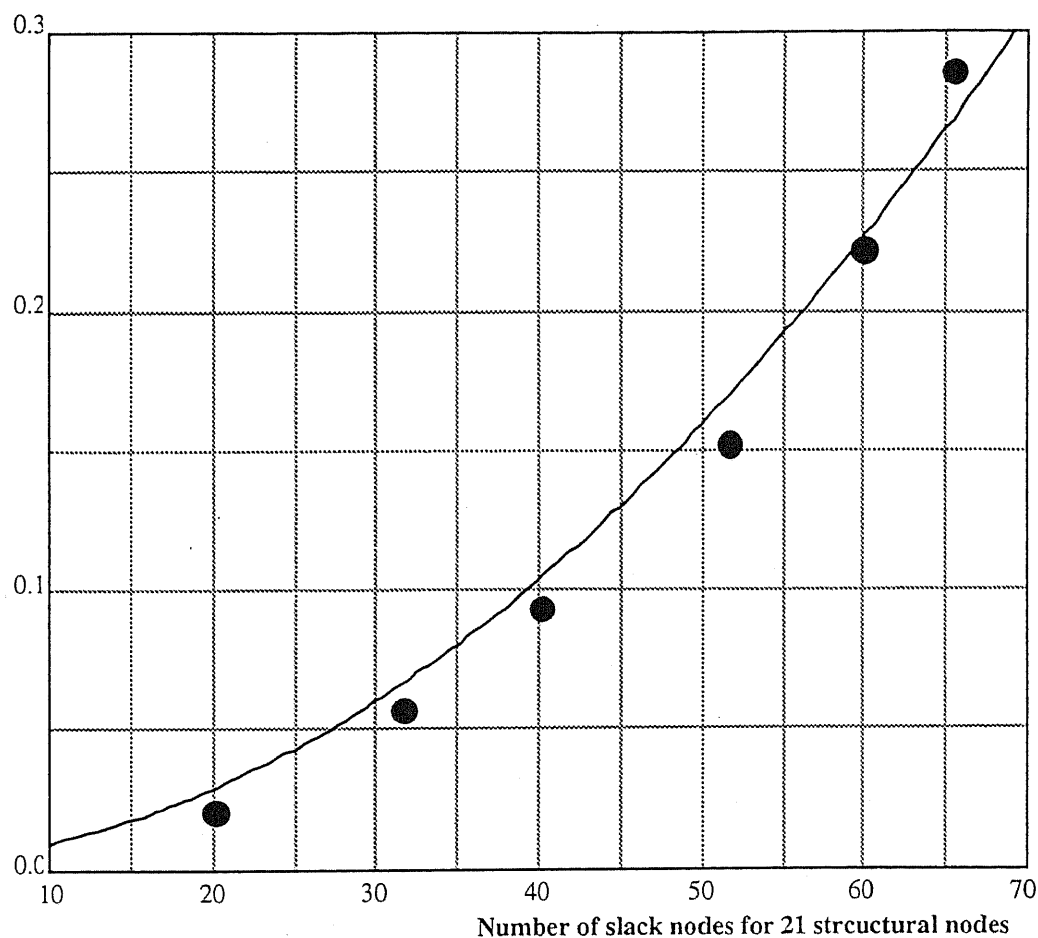


図4.9. 複雑さと推論時間の相関：縦軸の値が多項式オーダーである。

#### 4.6. 実験によるI-NBP法の評価

図4.10に新手法による実験結果を示す。システムはC言語で記述し、SPARC-Station10上で配列変数用に200Mbyteを確保して実行した。単一のホーン節の条件部のアトム数は $3 \pm 2$ 個、どのアトムも出現数の上限は10としている。多くの場合に単体法のみで計算を終了するので、単体法特有

のばらつきが見られるが、近似的に評価すると可能な要素仮説数( $N$ )の1.4乗程度の計算時間を達成している。これはもとのNBP法の推論速度( $N^2$ )を上回る結果であるほか、可能な要素仮説数が50程度までは初期フェーズの単体法のみで最適解探索が達成されるケースが頻繁に出現する。

Inference time  
(CPU time [sec])

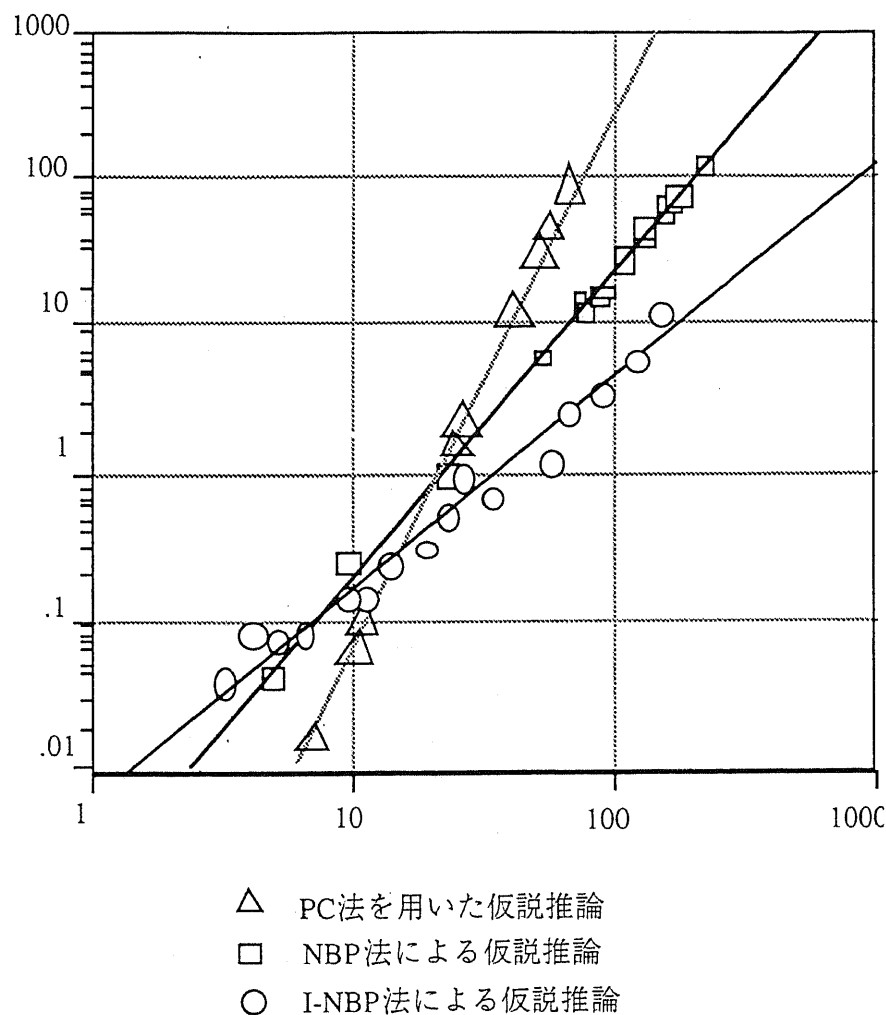


図4.10.従来手法と比較したI-NBP法による仮説推論の計算時間

## 付加的改良 ～ 違反制約の伝播 ～

この実験を行うにあたり，細部については上述に漏れた改良を加えた．  
その中では比較的重要度の高い，違反制約の移動方式についてここに述べておく．当初，掃き出し補数法をそのまま適用すると，掃き出し補数法のステップ7から9までにかけて最大2つまでの補数をとることによっても Infeasibility の改善が見られない場合が頻繁に発生することが観測された．

例えば，図4.11はその典型的な例である．やはり，■で違反されている制約のスラックノードを，●で真理値が1の非基底構造ノードを特に●特別して記すのに用いる．この場合，F,GをそれぞれAND,OR制約とすればをゴールとすれば，Fの値は1で上限が0であるので Infeasibility=1である．これを改良できる最小の補数の集合は{A, E, C}である．即ち，縦型探索でいうG以下のサブツリーを選択する場合に，A以下ではなくCを選択することに方針を変更する様な場合に，2変数の補数では不足するのである．

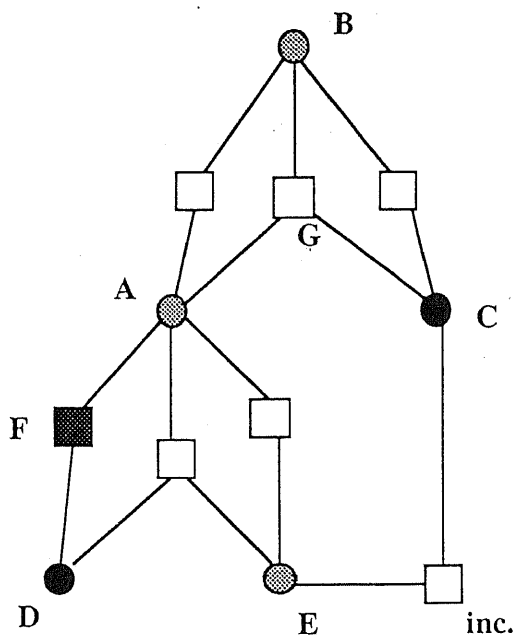


図4.11.

PC法では復帰できない制約違反



これを3変数に増やすのでは大きな改良とはいえない。条件部に3アトムあればそれだけで新たに同じ問題が発生するからである。従って、次の改良を行った。

#### 掃き出し補数法のステップ7～9の改良

1. 現在補数をとればInfeasibilityを小さくできる変数があれば、その補数を取り、 $\text{Infeasibility}=0$ となれば通常の掃き出し操作へ戻る。補数をとっても $\text{Infeasibility}>0$ であれば1へ。なければ2へ。
2. 現在補数をとった場合最もInfeasibilityを小さく出来るもののうち、最近10回補数計算をしていないノードの補数を取り1へ。なければ3へ。
3. 現在補数をとった場合最もInfeasibilityを小さく出来るもののうち、最近5回補数計算をしていないノードの補数を取り1へ。なければ探索失敗として終了。

図示すると、図4.12の様になる。このプロセスをもとのステップ7-9と入れ替えると、補数をとる後は必ずしもInfeasibilityは改善されない。しかし、補数計算は直前に補数をとったノードとも比較してそれよりもInfeasibilityの大なものは補数をとらないので、一回置きに少なくとも前と同等か又は改善されることになる。その結果、比較的広い範囲で補数をとることが必要である場合にも、違反制約の位置を長距離移動させることによって対処できるようになる。従来のProlog-likeなバックトラックでは制約違反と出会うとツリーの上位ノードに戻ってアトムの真偽をあてはめ直すのが、この手法では図4.12からも読み取れる様に、違反制約そのものを出口と思われるの向きに移動して行くのである。

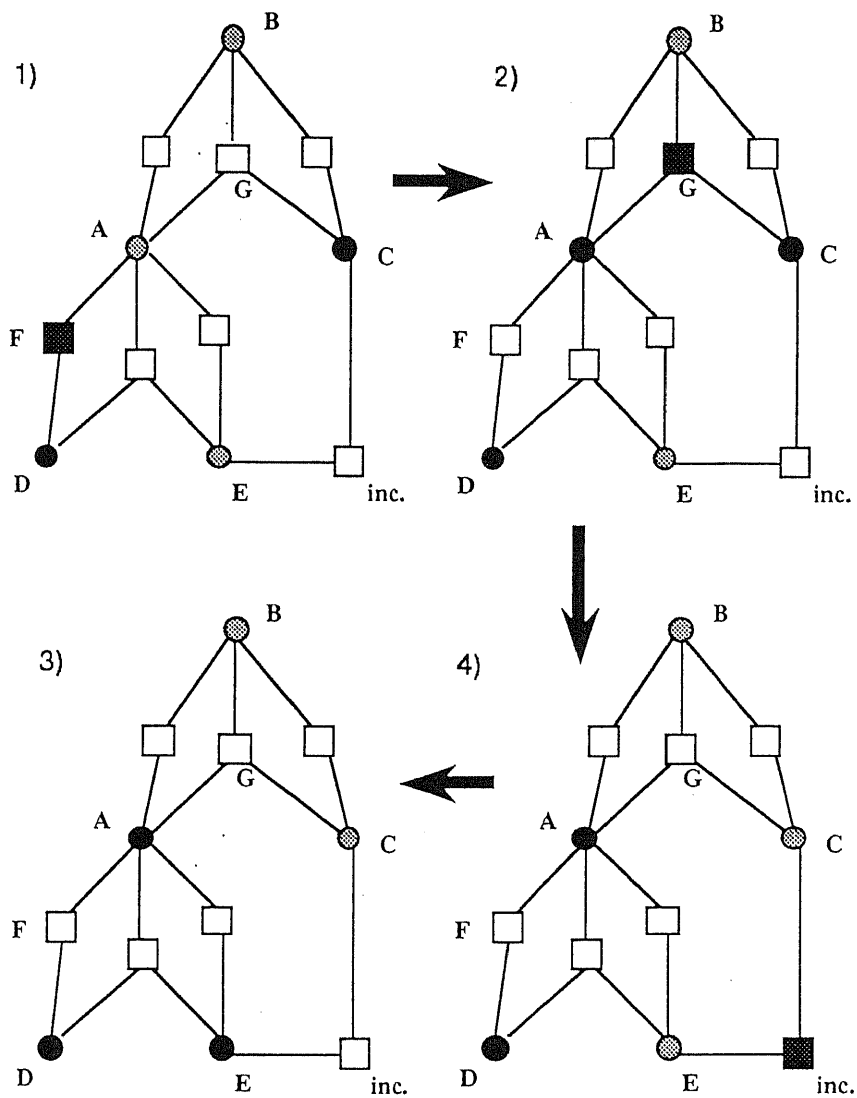


図4.12. 制約違反からの復帰方法の改良

違反制約の原因がそのアトムであるかは決定論的な探索では一般に計算コストのかかる問題であるので、この様に出口の「臭い」を頼りとするヒューリスティックは有効である。

又、図4.12の2)のGで、もし出次数（隣接ノード数）がKであれば最小では適切な選択をする確率が $1/K$ となるので、実際にInfeasibilityが1だけ減少するのは近似的にK回の補数をとる後ということになる。4.5節で述べた様に、最大でInfeasibilityはKとなるので、一度制約を違反してから実行可能な範囲に戻るまでの補数をとる回数の上限はほぼ $K^2$ 程度と見積ることができる。4.5節の副実験から推察されるように、これはもともと一回の掃き出しに要する時間と同程度であるから、明に実行結果には現れない範囲で補数をとる回数は増加していることになる。尚、もとの掃き出し補数法では2変数の補数計算を全て試みる一回の補数をとる作業数を考えれば、この改良手法自体の計算時間の方が短縮されている可能性が高い。

## 本章の付録（定理2以外の定理，補題の略証）

定理1 定理の否定が偽であることを証明する．即ち，図3.6の手続きだけでは支配ベクトルを付与できない枝があるとする．このとき，その様な枝全て（集合 $E$ ）を除去しても連結グラフとなる連結部分グラフで少なくとも一つ，ただ一つの枝 $e(e \in E)$ にのみ連結するもの（これを $G$ とする）が存在する．

というのは，もしそのような連結部分グラフがなければ，どの連結部分グラフにも二つ以上の $E$ の要素が連結されるので，無限に長い枝の鎖またはループが得られ，もとのグラフが有限かつ単連結であることに矛盾するからである．

$e$ と $G$ はただ一つのノードで接するが，このノードの $e$ 以外の隣接枝は全て $G$ に含まれるので支配ベクトルが施されている．よって，図3.6の手続きによりこのノードが●，■，○又は□のいずれでも $e$ に支配ベクトルが付与可能となり仮定に矛盾する． ◆

補題1 ■ノードは，不等式(4-1),(4-2)の不等号のいずれかの等号条件を表している．不等式は定数項が整数値で一次の係数は1であるから，出現する構造変数のうち1つを除く全てが整数値をとれば等号条件下ではもう1つの値も整数となる．このことと，全てのアトムの真理値が $[0, 1]$ の範囲に制約されることから本補題が成り立つ． ◆

補題2 クラス3のネットワークにおいて，どのノードも全ての隣接枝に支配ベクトルが付与されている．このうち支配ベクトルを全く吸収しないのは●ノードだけであるから，任意のノード $A$ から支配ベクトルの逆向

きに辿ると必ず終点は●ノードである。そこでそれらの終点ノードから今辿ったパスを支配ベクトルの向きに辿るとノードAに届く。その途中に□ノードはないので題意が成立する。◆

定理3 以下で、単結合の推論ネットワークと等価なBPNもまた単結合であることを断りなく用いる。まず、与えられたI-NBP法のBPNGにサブグラフとして含まれている同じ知識構造を表すNBP法のBPNを $G'$ とする(仮定より $G'$ は単結合となる)。このとき、次の2ケースに分けて考える。

1)  $G-G'$  上のスラックノードが全て基底状態のとき

定理1から、 $G'$ の全ての枝に支配ベクトルを付与できる。又、 $G-G'$ 上の枝は全て□ノードに接するのでこれに吸収される支配ベクトルを付与できる。

2)  $G-G'$  に■ノードが含まれるとき

$G-G'$  に含まれる■ノードの一つをAとする。ノードAに接する枝は2本だけ存在する。それぞれのA以外の端を構造ノードB, Cとする(図4.13左上図)。ノードAはノードBから矢印を受け取ると矢印をノードCに放出する(支配ベクトルの付与手順2)。同時にノードCは全方向に矢印を放出する(同手順3)のでノードB, Cは同時に支配ベクトルに包囲される。しかも、BからD, Cから図のノードDへの影響は互いに相殺し合うのでこの両影響を示す枝は無視してよい。即ち、元の図4.13左上図から枝BDとCDを刈り取りノードBとCを単一化した右下図の全ての枝に支配ベクトルが付与できるならば変形前のグラフでも全ての枝に支配ベクトルが付与できる。この変形の反復によりグラフは単結合となることから本定理が成り立つ。

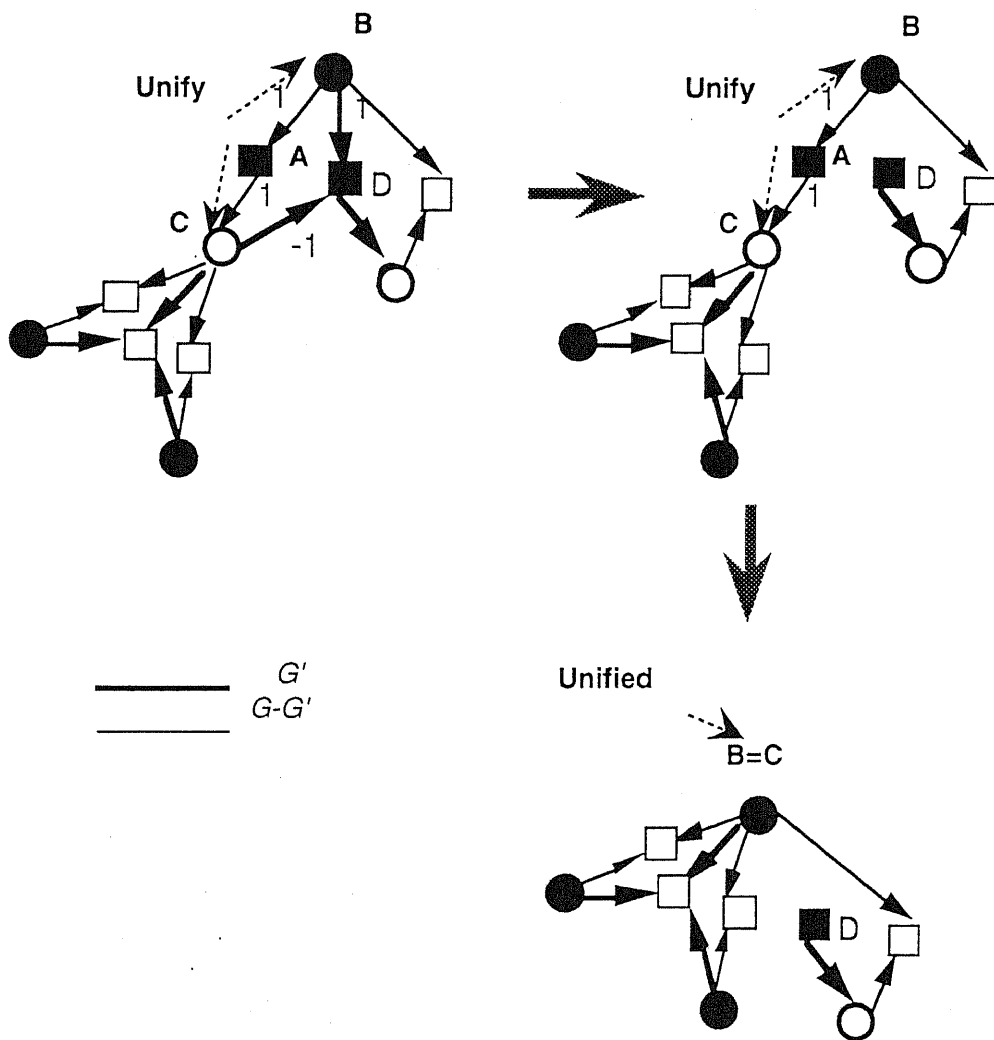


図4.13. 定理3の証明後半

□

補題3  $n$ アトムを条件部とするホーン節あるいは選言ルールにおいて,  
 $n+1$ 個のバブルが含まれるならば, このルール中の全ての構造ノードは  
 $0-1$ 整数値をとっている.

証明 ANDルールについて, 次の2つの場合に分けて証明する.

1) ルール中央のスラックノード (SNBP) に位置するバブルがない場合

全てのバブルが●か脇の■として含まれることになる。もし、後者のみであれば全ては等号条件なので、 $n+1$ アトムの真理値間で $n+1$ 個の等号条件という状態となり、その一つは冗長な（探索点の座標決定に寄与しない）制約となる。よって、全ての構造ノードは●か、又は等号で●に繋がる○ということになるので題意を満たす。

2) ルール中央のスラックノード (SNBP) に位置するバブルがある場合  
脇の■が含まれない場合には、単結合のグラフと同じことになり定理3より題意を満たすので、 $k$ 個 ( $k \geq 1$ ) の脇の■が含まれる場合を考える。これらの■を介して $Y$ （結論のノード）に結ばれる全ての構造ノード $k$ 個の真理値と $y$ の値は等しいのでその値を $x$ とおくと、式(4-1-1)の等号成立に当たる次の式が成り立つ。

$$(k-1)x + \left( \sum_{i=1}^{n-k} x_i \right) = n-1. \quad (4-1-1')$$

ここで、上の $k$ 個以外の構造ノードを一般性を失わずに $x_1, x_2, \dots, x_{n-k}$ とした。 $x, x_i$ は全て1以下の値であることを考慮すると、この式を満たすのは全てが1に等しい場合になる。

ORルールについても同様に証明できる。 $Y \leftarrow X_1 \vee X_2 \vee \dots \vee X_n$ が、完備性から $Y: \neg X_1, \neg X_2, \dots, \neg X_n$ と同値であることからこの同様性が保証できる。 ◆

補題4 バブルの伝播先が支配連結なバブル伝播ネットワーク中で伝播する先は、既に■ノードを含むどのルール内の□ノードにもならない。

補題4 1) 中央のスラックノード、および 2) 脇のスラックノードへのバ

ブル伝播に場合分けして証明する．尚，□内は，説明図にある伝播先のルールの条件部アトム数を2アトムからnアトムに拡張した場合である．

#### 1) 中央のスラックノードへのバブル伝播

伝播先のルール中央のスラックノードは□なのであるから，両脇のいずれかが■となっていることになる．即ち，下の図の状態であるから一般性を失わずに $y=z$ とおける．

よって，この時，中央の□ノードの値は  $z-y-x=-x[\rightarrow z-(x_1+x_2+\dots+x_n)=-x_2+\dots+x_n]$ となるが， $x$ に対する制約 $0\leq x\leq 1$ が満たされている限りこの値は  $-1[\rightarrow n-1]\leq \text{value}(\square)\leq 0$ となる．これは当該ルールがホーン節でも選言的ルールでも中央のスラックノードに対する制約を満たす範囲である．

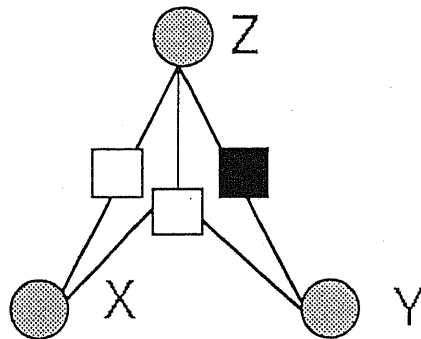


図4.14. 補題4の証明(1)

従って， $x$ さえ制約を見たしておればこのスラックノードは制約を満たしているのであるが，このことは， $x$ の相対深度は恒にこのスラックノードの相対深度よりも小さいか又は等しいということになる．又，バブル伝播前から $x$ にバブル存在する様な場合には必ず中央の□の制約は満たされるのでどの基底ノードよりもこのノードの相対深度は大きいことになる．伝播先には相対深度の最小のノードが選択されるので，いずれの場



合も，中央の□がバブルを受け取って■になることはない（脇の□ノードはバブルを受け取る可能性を有する）．

2) 上と同様の証明を行えばよい．即ち、中央のスラックノードにバブルが存在する場合にはすでに□の制約は満たされており、これを伝播先としてバブル伝播を行うことはないことを示す．

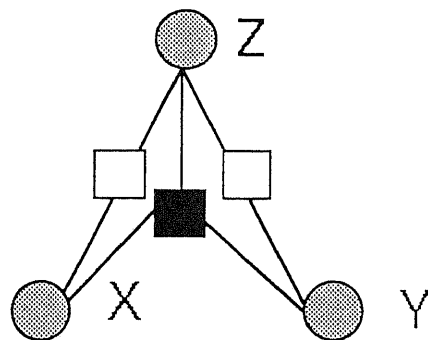


図4.15. 補題4の証明(2)

(b)図において、伝播先のルール中央のスラックノードは■なのであるから、これがOR(AND)制約であれば $z = x + y [\rightarrow x_1 + x_2 + \dots + x_n]$  ( $z = x + y - 1 [\rightarrow (x_1 + x_2 + \dots + x_n) - (n-1)]$ )とおける．よって $z - x = y [\rightarrow z - x_1 = x_2 + \dots + x_n]$ (AND制約では $z - x = y - 1 [\rightarrow z - x_1 = (x_2 + \dots + x_n) - (n-1)]$ )となるので、脇の□の値は0以上（AND制約では0以下）となる．これは脇のスラックノードの制約を満たしている． ◆

補題5 I-NBP法の場合，Series-Parallelなグラフでは補数計算が不必要である（補数がバックトラックに相当するが，バブルが同じノードを出たり入ったりすることもありうるので，この命題と直接多項式時間計算を結び付けることはできない）．

証明 知識ネットワークにおいて、拡張不能な0-1整数値の代入 $\theta$ 、即ち $\theta$ を含む0-1整数の制約充足解が存在しない場合というのは次の2通りである。

- a. 他の代入値に関わらず真理値1のアトムがボトムアップに否定される場合。
  - b. 他の代入値に関わらず真理値0のアトムがボトムアップに導ける場合。
- aの「否定」とはNegation as failureであるが、制約として知識を扱っている今の場合には、ある理由で真理値が0となることを差す。

又、あるアトムがボトムアップに導出できる場合には、次の場合がある。

すぐ下が選言 (OR) ルールであり、子の少なくとも一つが真。

すぐ下が連言 (AND) ルールであり、子の全てが真。

aでは、導出を妨げる原因となる0-1整数値は明らかに0である。そして、それがANDの子ノード $y$ であれば親ノード $x$ も $x \leq y$ から0となる。従って、この偽のノードと元の真のノードAの間にORノードが存在する。そこで、AB間には最も上位の値0のノードが存在し、このノードBはORノードの子であることがわかる。このORの結論は真理値1で0より大なのであるから、 $y \leq x_1 + x_2 + \dots + x_n$ よりBの兄弟ノードの少なくとも一つは0より大である。同様に、AからBまでのどのノードにも0より大の兄弟が存在するので、それを1とすれば結局ノードBはノードAの導出を阻止しないことになる。よって、それを1とできない場合にだけノードAの導出は阻止される。ボトムアップな導出は阻止できないことが分かったのであるから、これは矛盾によるトップダウンな（即ち、空節を導くという）制約に限られる。しかし、この場合には矛盾制約でこのノードCと結合されているノードの値が1の●である場合であって、ノードCの値は始めから0であったことになり仮定と矛盾する。 ◆

## 第5章

# 知識の実行時リフォーメーション による高速化

説明に基づく学習 (Explanation-based learning, EBL) は、以前に解いた問題の解や解に到る文脈を保持し、同様の問題を同様に解こうとする手法である。[牧野93]では、これを仮説推論に適用した。この研究は、過去の証明に用いたサブゴールの支持仮説を記録し、次回からこのサブゴールを証明する手間を省くことによって高速化を達成したものである。

しかし、経験から学習するだけでは当然、未経験の問題に対応することができない。そこで、知識コンパイル法では背景知識中の Prime Implicants (主項) を計算する (全てのゴールについて予め指示する前提を全て用意しておくことに当たる) [鶴田91,92]。定義を少し簡単化して全てのアトムを仮説アトムの積和形で表すと、

$a :- b, c.$

$b :- d.$

$b :- e.$

$c :- d.$

$c :- f.$

という知識は、

$$a = bc + d + de + df + ef (= d + ef).$$

$$b = d + e.$$

$$c = d + f.$$

となり，例えばeとfが矛盾するならば更に，  
 $a = d$  と変形する．この変形をしておけば後でどのアトムをゴールとした仮説推論の問題もメモリ探索のみに置き換えることができるので高速な探索が可能となった．しかし，この処理は仮説推論を解くことに当たり，所要時間と所要メモリは指数オーダーで爆発的に増大する．そこで，背景知識中の一部のゴールのみ展開することによってこれらを節約した部分コンパイル法では，例えば矛盾する知識に共通の親を展開するといった様にコンパイルの手間を簡潔化した[堂前94]．これによってメモリを節約することに成功し，コンパイル後の仮説推論の実行速度も向上した．しかしコンパイルの実行時間は，無駄が省けたとはいえやはり爆発的に増大する．

これらの知識リフォメーションは，長期に渡り使用するシステムの仕組みを背景知識として故障が起きたら診断するという場合の手法であるといえる．だからこそコンパイルに時間がかかってもする価値があるのである．逆に言えば，こういう場合以外，即ち変動する知識を前提とした場合には適用しても全解探索することになり高速化に結び付かない．

とはいえ，3章に示したように問題の難しさはその構造に依存するのであるから予め簡単化しておくことは推論の高速化にとって極めて有効な手段である．そこで，ゴールが与えられ，仮説推論の問題が与えられてから初めてリフォメーションすることを考えた．それには，リフォメーションそのものの実行時間を短縮する必要がある．

こうしたモチベーションから我々は，知識の実行時リフォメーションを提案，開発した．その原理は非常に簡単である．即ち，与えられた知

識をLSI設計の分野で示された手法を一部用いながら論理簡単化を行う。

ここでサブルーチンとして用いるMIS[Bra 87]は論理簡単化システムとして知られるシステムであり、これまでも多段論理回路の設計を用途として用いられてきた実績を有する[藤田91a,b]。ユーザは、簡単化したい論理式をシステムに入力する。これは仮説だけで全てのアトムを表すという上記の知識コンパイル手法とは逆に、代数的あるいは論理的に因数分解するというものである。この因数分解こそ冗長な表現を解消して問題の構造を簡単化する処理に当たる。

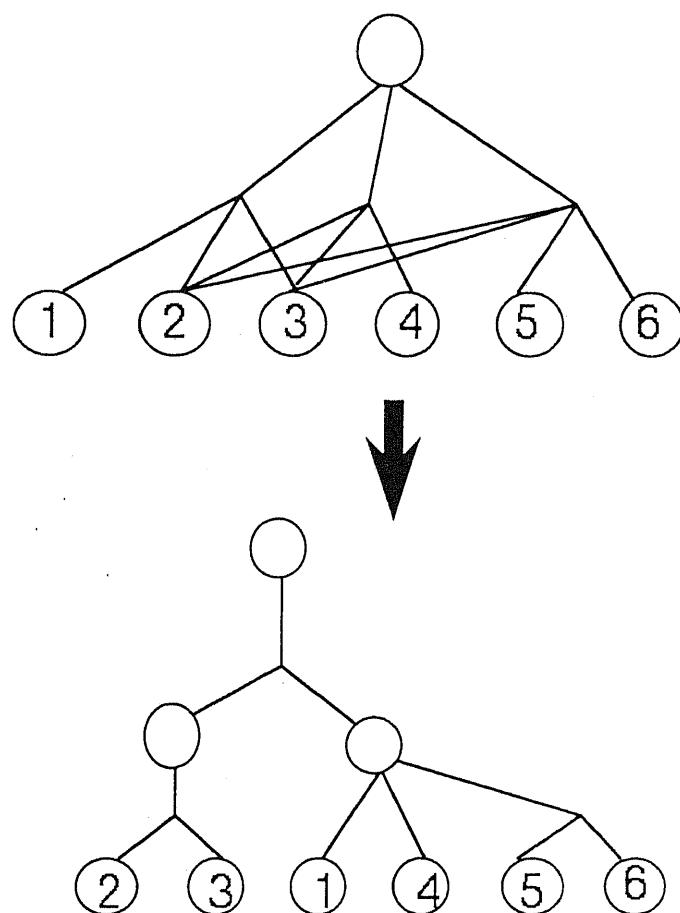


図5.1. 論理多段化による簡単化

例えば、図5.1の様に閉路の解消によって構造的に簡単になるのが分かる。3章に示したように、この様に構造を単結合に近付けるタイプの単純化は、NBP法にとって高速化に直接繋がる。ただし、実際には論理的な因数分解は代数的な因数分解よりも難しい。例えば、

$$\begin{aligned}b + (-b) a &= b(1 + a) + -b a \\&= b + (b + (-b)) a. \\&= b + a.\end{aligned}$$

を用いて

$$\begin{aligned}b + a + (-b) d + ae + de \\&= b + (-b) a + (-b) d + ae + de \\&= b + (-b + e)(a + d).\end{aligned}$$

とすることが可能となるが、ここで " $b + -b = 1$ " をどの様に用いるかということは代数的な因数分解では生じない、計算量のかかる問題である。

しかし、我々が扱うのは負のリテラルを明に含まない仮説推論の問題であり、 $-b$ を $b$ と矛盾するアトムとしても $b + -b = 0$ は許されるので、このケースは生じない。そこで、我々はMISを用いるに当たり代数的因数分解と論理的因数分解との相違は幾つか残るがこの点は妥協して簡単な前者を選択する。

更に、知識コンパイルの様に知識を完全に展開して（=全アトムを仮説のみで表して）から因数分解するのではなく、短時間に展開できる範囲で因数分解しやすい形に変形した知識をMISに与える手法を取り入れた。その部分のアルゴリズムは以下の通りである。

1. アトム2集合KB1およびKB2を、次のように与える.

KB1: 複数のホーン節に共通のヘッド.

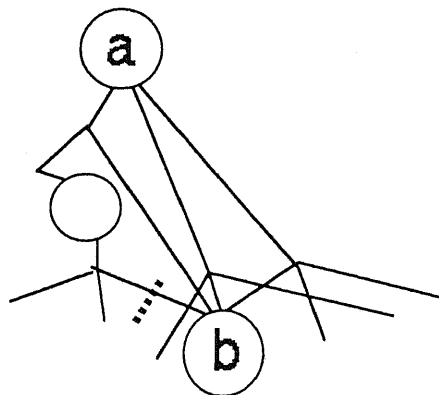
KB2: KB1以外のアトム.

2. KB2の前件をKB1に代入

3. 多段化

下位から上位へ、ヘッドを共通するホーン節が共有する前件部のアトムを移動

4. 矛盾を除去 (矛盾するアトムを共に前件に有するホーン節のヘッドを偽として削除可能なアトムを削除する. その中には, その矛盾するアトムの組も含まれる).
5. アトムaより下位のアトムをヘッドとするホーン節の前件からaをヘッドとするホーン節の全ての前件に共通のアトムを除去する.



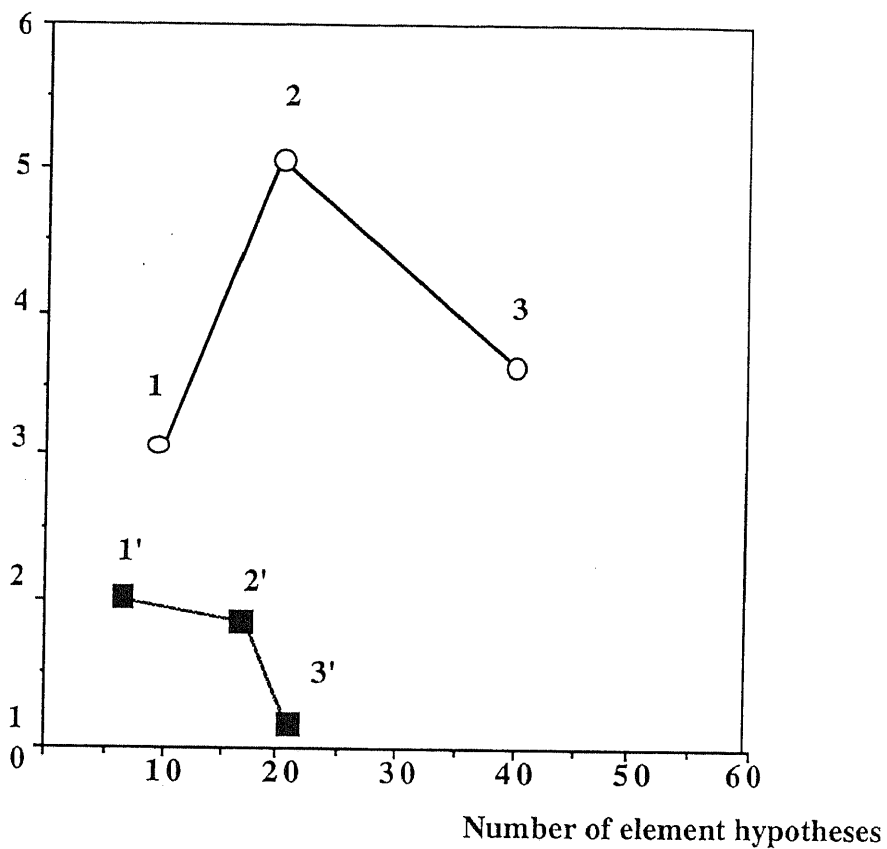
6. MISで多段化処理.

図5.2. 知識の実行時リフォーメーションのアルゴリズム

上の操作によって、ゴールの論理式表現に変化は生じない. この操作の結果、先ず知識の構造が次のように簡単化された. この図の各ドットに示した数字は、データ1'がリフォーメーション前のデータ1の知識ベース

をリフォーメーション後のデータなどの意味を持つ。

Average  
degree of  
nodes



—○— before reformation  
—■— After reformation

図5.3 知識の実行時リフォーメーションによる，構造の簡単化

ここで，縦軸は平均出次数，即ちグラフ中の全リンク（枝）数をノード数で割った実数値で，グラフの複雑さの指標である．この簡単化によって，仮説推論の推論時間（afterにはこの処理を含めた）は次のように短



縮された。矢印によって同一の問題のリフォーメーション前後の計算時間の変化を示した。他にもKICK-HOPE[近藤93]の開始時の処理として本手法を用いてこれと同様の計算時間短縮の効果が得られている。

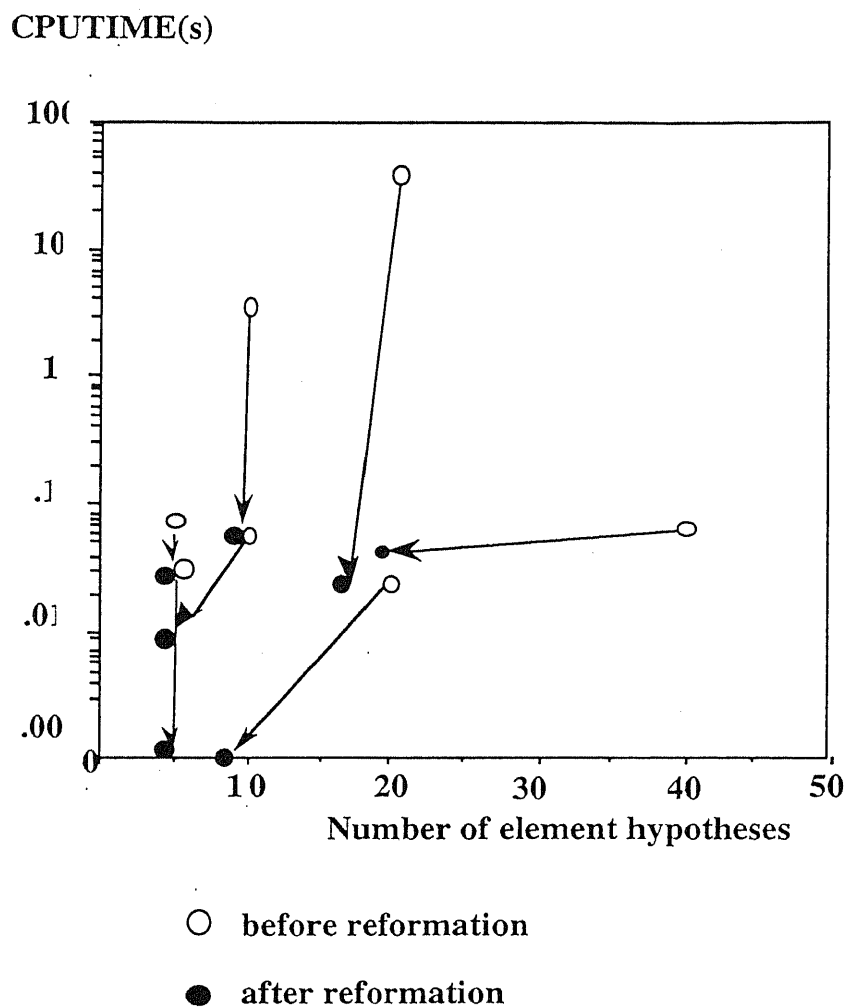


図5.4. 仮説推論の計算時間の短縮

又、近似解法であるNBP法では解の良さ、完全性に妥協が含まれるので非常に複雑な問題は解くことができない場合もあったが、本手法を導

入することにより最適解をさらに多くの問題について求めることができるようになった[高間94].

我々人間が推論する時も、余り時間を取らない範囲で与えられた問題を簡単な形に書き換えるように思われる。特に、あることを証明する場合に頭に出てくるいくつかの方針のなかで共通なものや、本質的だと思われる部分をまずしっかり証明するのは一般的な考え方である。この共通部分を分けておき、何度も証明しないで済むようにしたのがここに述べた実行時リフォーメーションであると考えれば、汎用的な前処理であると結論することができよう。

## 第6章

# 述語論理への拡張

### 6.1. ネットワーク化の本来の意義

数理計画法のネットワーク化による解法は、単に命題論理の仮説推論を高速化するばかりの目的ではなく、より広い知識処理の枠組みに拡張していける可能性を有する。数理計画法自体は効率的な最適解の探索法を開発し、数学的な基礎を与えてきたが、知識処理としての推論の枠組みは人間の知的な思考法をヒントにして、数理計画法と異なる視点から構築されてきた。そこで、数理計画法を知識ネットワーク上での動作に置き換えるということは、前者の数学を基礎とする計算と後者の知識構造の利用を組み合わせる新しい方法論であり、数理計画法の分野で開発されてきた高速探索法を知識処理技術としても浸透させていく上で重要な掛け橋となろう。

実際、[岡本93]の推論プロセスは数理計画法の枠内で自動的に解仮説を求めるため、動作が知識処理の側から理解できない。これに対し、NBP法は知識ネットワーク上で動作するので知識処理の観点から理解しやすく、推論を効率化するために有用な考え方と融合することが可能となる。

本章では関数なしの述語ホーン節論理表現のコストに基づく仮説推論を取り扱う。述語論理の場合、表現が変数を含むので数理計画法に帰着する方法がこれまでに提案されていない。ここでは、エルブラン空間への展開を行わずに述語論理仮説推論の問題を等価な0-1整数計画問題に帰着し、知識構造を反映したBPNを生成する。又、ゴールの証明には同一ルールを何度も再帰的に用いる必要がある知識構造に対処するため、推論ネットワークをトップダウンに必要なだけ拡大するメカニズムを取り

入れる。これにより、無駄に大きなネットワークを用いずにNBP法の高  
速推論を活かせる。

## 6.2.述語論理表現の仮説推論へのアプローチ

述語論理では変数を含む知識を扱う。各変数の具体例の全て、即ちエ  
ルブラン空間に展開すると命題論理知識ベースに置き換えられるが、サ  
イズが膨大になりメモリと推論速度の両面から現実的ではない。

モデル生成に基づく並列アブダクション（仮説推論）手法[井上93]では、  
効率的にボトムアップにモデルを生成することで述語論理の仮説推論を  
高速化しており、マジックセット法[Ban 86]を用いてゴール指向の性質を  
加える手法の有効性も示している。一方、トップダウン推論形式を採る  
と知識に再帰構造のある場合に一般に非効率となるが、SLD-AL反駁、  
QSQR法[Vieille87,88]やこれに準拠した[近藤93]は使用する仮説の情報をボ  
トムアップに伝播することでこの問題に対処している。

本論文では、述語論理仮説推論に拡張したNBP法により従来の手法を  
上回る推論速度を達成できることを示す。これは数理計画法の延長から  
述語論理仮説推論を実行する新しい試みといえる。即ち、アトム真理  
値以外に変数の束縛情報も含めた連続値の情報をBPN上で伝播させ、最  
後に推論結果として解仮説を出力する。さらに、動作中のBPNを短期記  
憶とみなし、これと長期記憶（知識ベース）とのやり取りから短期記憶  
を拡大することによって再帰構造に対処する。

実験により示される推論速度は候補要素仮説数に対して指数オーダで  
増加する傾向ではなく、実際にゴールの証明に用いる要素仮説数の多項  
式オーダとなっている。

### 6.3. 述語論理に拡張したNBP法

#### 6.3.1. 述語論理に対する拡張NBP法の概要

命題ホーン節の背景知識は容易に線形不等式制約に変換でき [Ohs93, 岡本93], 数理計画法の既存技術を適用できる. これに対し, 変数を含む述語ホーン節については同様のアプローチは困難とされてきた. しかし以下に示すように, 関数のない述語ホーン節で表現された推論ルールは変数と変数の関係と見なすことで不等式制約に変換できる. これにより, 元の知識ベースをその構造を反映したBPNに変換でき, 次の特徴が得られる.

- 1) 命題論理のNBP法と同様の高速化が実現できる.
- 2) ゴールを起点としてネットワークを段階的に拡大しながらNBP法を適用することにより, ゴール指向の推論が実行できる.

以下では, ここで述べる述語ホーン節表現の仮説推論に対するNBP法を述語版NBP法と呼ぶことにする.

#### 6.3.2. 述語版NBP法で用いる不等式制約

述語ホーン節の仮説推論を0-1整数計画問題に変換する方法を示す. まず, 関数なしの述語ホーン節を, 変数間の関係に課された制約と見なす. 例えば例1の2)では, 前件の $h_2$ と $g$ のそれぞれ第2, 第1引数が等しい値 $Z$ をもつとき,  $X, Y$ の値が $g$ に代入される.

例1: 背景知識: 1)  $g(X, X) :- h_1(X).$

2)  $g(X, Y) :- h_2(X, Z), g(Z, Y).$

3)  $h_1(X) :- h_2(X, X).$

4)  $inc :- h_1(X), h_1(Y), X \neq Y.$  (incは矛盾)

ゴール:  $g(a, X).$

要素仮説:  $h1(c), h2(a,b), h2(b,b).$

一般に, Range restricted な (結論部の変数は全て前件部に出現する) 関数なしホーン節は, 前件部の同一名の変数の代入値間の等号成立を引き金とした結論部の変数への値の代入を表している. この等号成立から代入に至る動作を代入スイッチと呼ぶ. 代入スイッチの動作を制約不等式で表すために, 以下の方法をとる.

代入スイッチ内では, 変数への代入値を  $a=01, b=10, c=11$  などのビットベクトルで表す. その上で, 等号条件をテストされる  $n$  個の同一名の変数 (例えば "X") を出現位置により区別して  $X_1, X_2, \dots, X_n$  と表すことにし, それらへの代入値のビットベクトルの各  $p$  桁目の値を  $TP_1, TP_2, \dots, TP_n$  で表す. このとき代入スイッチの ON/OFF を変数  $S$  の値の 1/0 で表すと, 等号条件テストは,  $p=1, 2, \dots, m$  ( $m$ : ビットベクトルの桁数) の全てに対し  $TP_1 = TP_2 = \dots = TP_n$  のとき  $S \leq 1$  即ち  $S=0$  又は  $1$  とし ( $S=1$  に限らない理由については後述), それ以外では  $S=0$  とする制約を意味する. この様な制約は,  $p=1, 2, \dots, m$  に対する  $m$  個の制約不等式 (6-1) で与えられる (変数名の異なる  $q(\geq 2)$  個の等号成立が要請される,

$odd(X) :- plus\_1(X, Y), plus\_1(Y, Z), odd(Z).$

の様な場合は  $mq$  個の不等式を重ねる) .

$$1 \leq S + \left(2 - \frac{1}{n+1}\right) R^p - \frac{\sum_{i=1}^n TP_i}{n+1} \leq 2. \quad (6-1)$$

ここで  $R^p$  は補助変数で, 制約式 (6-1) に  $R^p=0$  又は  $1$  という制約を加えると, 0-1 整数値の  $TP_1, TP_2, \dots, TP_n (n \geq 2)$  と  $S$  に対し上記の等号条件テストの制

約になる（証明は本章の付録）．

桁毎に等号の成立不成立が食い違っても，ホーン節1つに対し単一の変数 $S$ で矛盾なく代入スイッチのON/OFFが表せる．これは，式(6-1)で $TP_1 = TP_2 = \dots = TP_n$ のときの変数 $S$ は0又は1のどちらの値を取ることもでき，ある桁で等号が成立しないときは自動的に $S=0$ となるためである（全ての桁で等号が成立しても代入が行われないことがあるが，ゴール証明にそのルールが必要ならば自動的に $S$ の値は1となるので支障とならない）．

次に代入実行部の機能は，変数 $S$ の値が1の時には前件部の変数値を結論部の変数に代入し，0の時には代入しないというものである．何も代入されていない変数の値は各ビットが0のビットベクトル $\phi$ で定義しておく．即ち，前件部と結論部で同一名の変数（例えば，例1の2)の $X$ ）の代入値の $p$ 桁（ビット）目をそれぞれ $X_{in}^p, X_{out}^p$ とすると， $S=1$ のとき $X_{in}^p = X_{out}^p$ ， $S=0$ のとき $X_{out}^p = 0$ とする．この関係は，変数 $X_{in}^p, X_{out}^p, S$ が0-1整数値をとる制約下で不等式(6-2)と等価である．

$$\frac{X_{in}^p + S}{2} - \frac{1}{2} \leq X_{out}^p \leq \frac{X_{in}^p + S}{2}. \quad (6-2)$$

以上の様に，述語ホーン節と等価な制約不等式が得られた．以下3.3節でこれらの不等式から基本的なBPNを構成し，6.3.4節ではこれらを結合しながら推論を行う手法を述べる．

### 6.3.3. 述語版NBP法のBPN

等号条件テスト部と代入実行部を組み合わせると図6.1の形の代入スイッチとなる．図6.1下部の長方形内の $T^1_i, T^2_i, \dots, T^m_i$ は変数 $X_i$ のビットベクトル（ $m$ 桁）の各成分を表す．式(6-1)の変数 $S$ の値を持ち，代入の引き金の役割を果たすノード $S$ を以下スイッチノードと呼ぶことにする．即ち，

スイッチノードの1/0がそのルールの使用/不使用を表す．この様に桁数だけ等号条件テスト部を重ねるが，変数Sと同様にスイッチノードはホーン節1つに対して1つで良い．

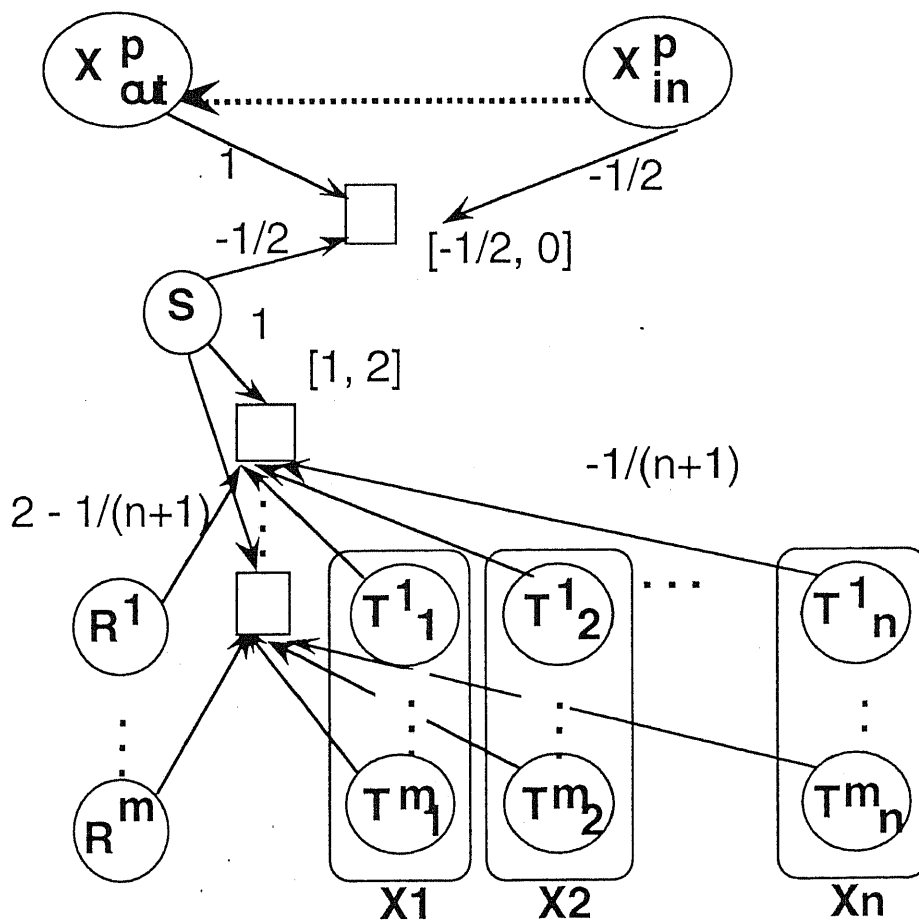


図6.1.代入スイッチのBPNによる表現.ノード S は，変数 Sなどに当たる．

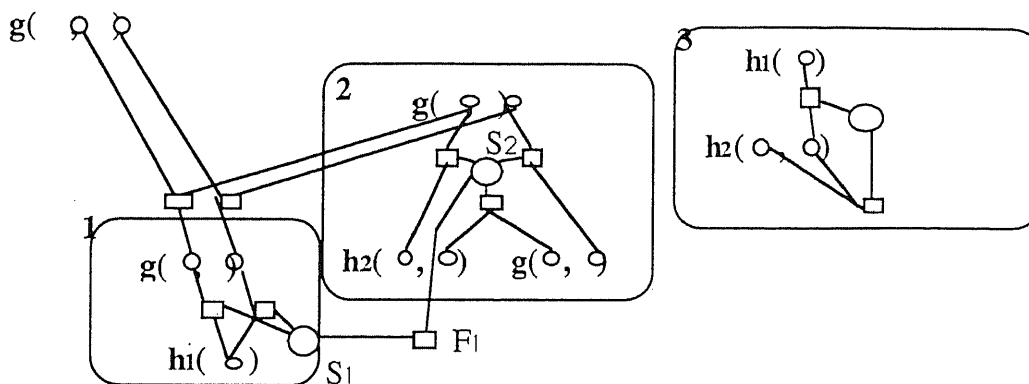
矢印，□ノードに添えた数字はそれぞれその向きの伝播率，上限  
下限の制約を表す．

知識ベース全体を図6.1の様なBPNの基本単位の集まりとして構成し直して長期記憶(Long Term Memory, 以下単にLTM)と呼ぶ．この中で，結論



が同一の述語記号を持つホーン節はまとめて一つの**BK**（背景知識）単位とする．例えば，先の例1のルール1)と2)を図6.2左上の様に組み合わせ，述語記号 $g$ を定義するBK単位を生成する（各枠に付した数字は例1のルール番号に対応する）．尚，図6.2とこの後の図6.4以降は概念図で，実際にはビットベクトルの桁数だけ重なった図6.3（長円，長方形，ノードF1，S1，S2が図6.2左上図のノードに当る）の様なBPNを用いる．

### Background:



### Elementary hypotheses:

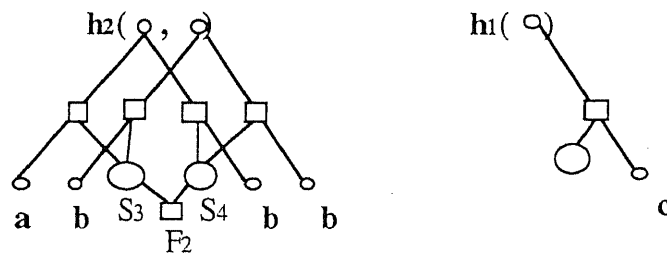


図6.2. 長期記憶に保存された各ルールと等価なサブネットワーク（略図）．大きな円はスイッチノードである．

この様な複数のホーン節からなるBK単位の結論部の変数値は，これらホーン節のうち一つだけを用いるという制約の下でそれら全てのホーン節

の結論部の変数値のビットベクトルの和とする。例えば図6.3では、  
 $F1 = S1 + S2$  なるノードF1の値に  $0 \leq F1 \leq 1$  の制約を与え、ホーン節1,2の各結論部のビット $\xi, \psi$ の値の和をBK単位の結論のビット $\omega$ の値とする ( $\xi = \omega - \psi$ なるノード $\zeta$ に、制約 $\zeta = 0$ を課す)。

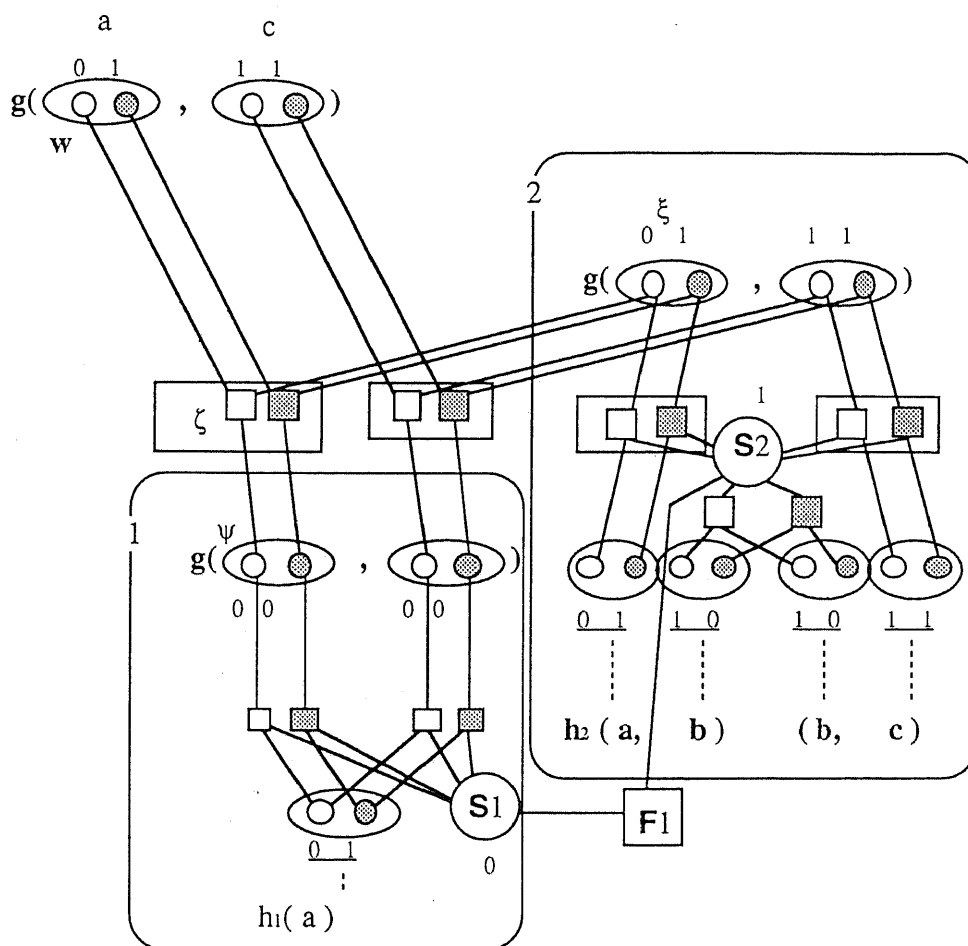


図6.3. 例1のルール1), 2)の一方から $g$ の変数への代入例

(ビットベクトルの1桁目を白色, 2桁目を網点で区別した)。

即ち、スイッチノード $S1$ ,  $S2$ を同時にONとしないノード $F1$  (フィルターノードと呼ぶ) の制約により証明に用いるホーン節が片方に限定され、

単純な算術和で結論への代入が可能となる。

又、背景知識とは別に、仮説は同一の述語記号のもの毎にまとめて仮説単位としてLTMに保持する。例えば、図6.2左下図ではノードS3が  $h2(X,Y)$  に  $(X,Y)=(a,b)$  を代入するスイッチで、その値の1/0が仮説  $h2(a,b)$  の使用／不使用を表し、同様にノードS4は仮説  $h2(b,b)$  を表している。推論に仮説として用いられる  $h2(X,Y)$  の変数値は  $h2(a,b)$  か  $h2(b,b)$  のいずれか一方に確定するので、フィルターノードF<sub>2</sub>を設け  $0 \leq F_2 (= S_3+S_4) \leq 1$  とする。  $S_3 = S_4 = 0$  の場合は  $h2(\phi, \phi)$  ( $\phi \equiv 00$ ) となり、  $h2(a,b)$ ,  $h2(b,b)$  のいずれの仮説も証明に用いないことになる。

以上により、ホーン節のヘッドとして現われる述語記号と同数のBK単位、および仮説として現われる述語記号と同数の仮説単位からLTMが構成される。これらの単位を推論過程で段階的に結合しながらNBP法を適用して推論を行うのが述語版NBP法である。

#### 6.3.4.短期記憶と長期記憶のやり取りによる述語版NBP法

人間がトップダウンに推論を行う場合、証明したいゴールを直接導くことのできるルールと仮説を先ず探すであろう。それらが見つからない場合には問題がそれほど簡単ではないと考え、少し広い範囲でルールと仮説を探す。もし解仮説が存在するならばこの方法でいつか求めることが出来るし、余り深い推論が必要な場合は証明を諦める。このゴールを起点として推論の範囲を段階的に拡大する方法は、Iterative Deepening [Korf85]と共通する考え方である。

又、述語表現の知識には再帰的なルールが存在しうる。これは、あるアトムからこれと同じ述語記号のアトムを導く様なルールであり、推論の効率化を妨げる要因となる[近藤93]。例えば、

$\text{man}(X) \text{ :- father}(X, Y), \text{man}(Y).$

の様に、前件と後件に同一の述語記号を含む場合がそれである。より一般的には、

$\text{man}(X) :- \text{father}(X, Y), \text{human}(Y).$

$\text{human}(X) :- \text{man}(Y).$

の様に、演繹によってある述語からこれと同一の述語が導かれる場合も再帰構造を有することになる。

上述の様なトップダウンなアプローチから再帰構造を取り扱う手法については、演繹データベースの分野においてこの15年程度で何度も改良が加えられてきた。

[Aho 79]が発案し[Ioannidis 85]によってStatic Filteringに拡張されたFixpoint Operationは、束縛情報のトップダウンな伝播によって解空間を限定することでこの問題を処理することを提案した。しかし、ゴールが変数しか含まない様な場合を考えれば容易に理解される様に、これでは一般的な問題で束縛情報を使いこなしたことはない。[近藤93]にも示された様に、この方法では再帰構造の無限ループに陥って推論がストップしない場合さえある。これに対して、[Ioannidis 85]や[Min 83]の様に一定の深さまで推論パスを生成しておく手法は、平易にこのループを食い止めることができる。しかし、その深さの上限を決める時点で問題が発生する。即ち、これが浅すぎると推論の完全性を著しく損ない、深すぎると無駄にサイズの大な問題を解くことが必要となる。これら二つのアプローチが抱える問題は、Iterative Deepeningの考え方では、解仮説に至る所定の深さまでネットワークを拡大した時点で推論を停止させることによって対処できることとなった。しかし、Fixpoint Operationにおける束縛情報の伝播は、利用可能な情報はできるだけ利用して解空間

を絞ることができる点で有効であり、新しいアイデアを加えることによって発展させる価値があった。そこで、ボトムアップあるいはサイドウェイな伝播までを実現したのが[Vieille87,88]のQSQアプローチである。その一つのバージョンであるQSQR[Vieille87,88]法は、ゴールからトップダウンにサブゴールを生成して行くプロセスに、仮説の変数束縛情報のボトムアップな伝播を組み合わせることによって再起構造を有する問題を効率的に処理する手法である。この束縛情報伝播は[近藤93]にも取り入れられた。

推論ネットワークの拡大によって再帰構造を扱う方針は、自然言語研究における包括的な研究[Hasida93]にも取り入れられている。

我々は、Iterative DeepeningとQSQRの考え方を、次の手順に示す様にBPNの段階的な拡大という形で取り入れる。即ち、ゴールを起点として推論過程で段階的に拡大するBPNを短期記憶(Short Term Memory,以下単にSTM)と呼び、初期状態ではゴールのみからなるSTM(手順1)に、解が見つかるまで必要なルール(手順2-1)及び仮説(手順2-2)をLTMからコピーして行く。

#### STMの段階的な拡大により仮説推論を行う手順

1. ゴールの述語だけを含み枝のないBPNを初期のSTMとし、ゴールの述語を初期の結合部とする。
2. 2-1と2-2を現在のSTMの結合部全て(ここで新しくできる結合部を除く)に対して実行する。STMが拡張されなければ解なしとして終了する。
  - 2-1. 結合部(Cとする)の述語をヘッドとし、まだCに結合されていないBK単位を(後述の基準1により)Cの下にコピーする。ここでできたSTMの葉ノードの述語を新しい結合部とする。
  - 2-2. 結合部Cの具体化でまだCに結合されていない仮説を(後述の基準2

により) 結合部にコピーし, 仮説間の矛盾制約ノード(後述)を追加する.

3. 現時点のSTM上でNBP法を実行し, 解が得られればスイッチノードがONの使用する仮説を解仮説として出力して終了する. 得られなければ2へ.

[Korf85]のIterative Deepeningと同様に最も浅い解仮説を深いノードまでトレースせずに求めることに加え, BPNのサイズに対して多項式時間の計算を行う(手順3)ことにより高速化を達成する. このBPNは, 述語表現の知識をエルブラン空間に展開せずにそのままネットワーク化したコンパクトなものとなる.

小さいネットワークから得られる準最適解よりも大きいネットワークから得られる準最適解の方がコストが低いことも稀にあり得るが, この方法ではこのような場合を無視している. これは通常, ゴール証明に関係する推論ネットワークが大きくなると用いる要素仮説数も増大し, その重みの和であるコストも増大することから妥当な方針となる. そして, 解が得られた段階でそれを準最適解とし推論を終了する高速化のメリットが得られる.

#### LTMからのSTMに結合するサブグラフの選択

上の手順2で, 詳しくは推論に必要として選択されたネットワークのみLTMからSTMに結合する. 即ち, ゴール証明に無関係な知識を考慮しない様に, 結合する部位を次の2基準で限定する.

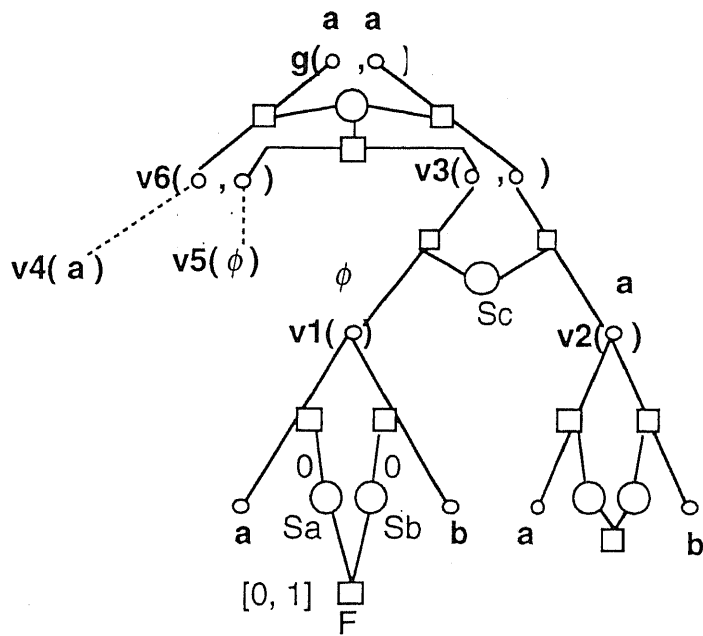
- 基準1) BK単位を結合する際は, 結合部をヘッドとするBK単位のうち, STMのゴールから伝播した変数束縛情報に矛盾しないものを結合する.

基準2) 仮説を結合する際は、1仮説単位の中でもゴール及び他の結合部の仮説から伝播した束縛情報と一致する部分を選択して結合する。

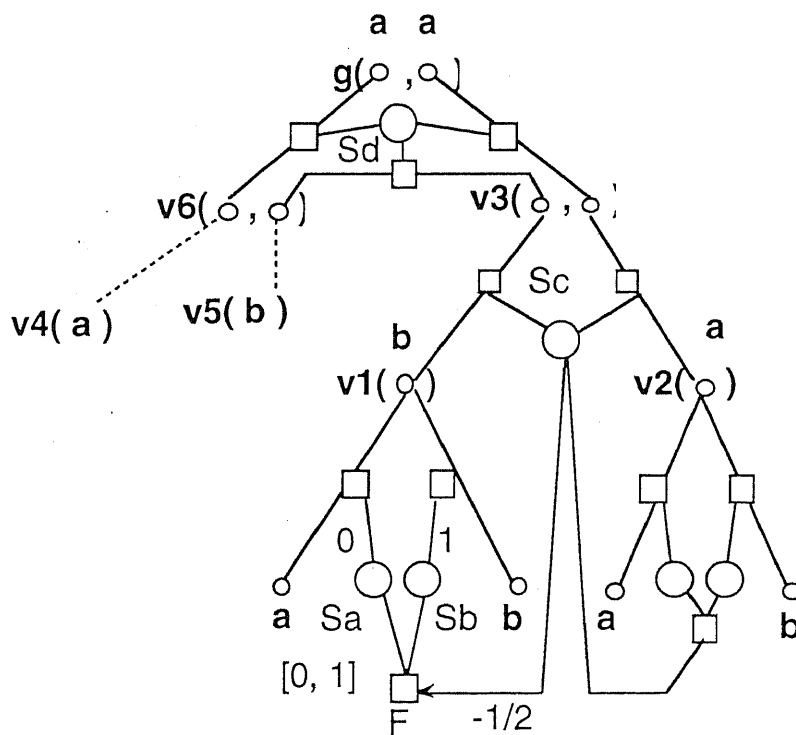
図6.9の状態1を例にとる。基準1では、矢印の向きにゴール $g(a, b)$ の変数の束縛情報をトップダウンに伝播すると、例1の1)では直接ゴールを導けない ( $a \neq b$ より $g(X, Y) = g(a, b)$  はありえない) ことが分かる。従って、仮説 $h_1(c)$ も $h_1$ をヘッドとするBK単位もSTMへ結合されない。又、基準2では、ゴールから伝播した束縛情報によって $h_2$ の第1変数は $a$ に限定され、図6.9の点線の部位は結合されない。もしこの深さで $h_2(b, b)$ を結合しても、それは $g(a, b)$ の導出とは無関係だからである。この時結合される仮説 $h_2(a, b)$ から図中  $\rightarrow b$  の束縛情報の伝播を行い、後にこの方向に結合する仮説の基準2による限定に用いる (後述の動作例参照)。このように、STMにコピーするLTM中の部位を限定することにより、NBP法が実行されるネットワークが不必要に大きくなるのを防ぐことが可能となる。

#### 6.3.5. $\phi$ 束縛の問題とその対策

別の問題が、" $v_3(X, Y) :- v_1(X), v_2(Y).$ "の様に、結論部の変数値に前件部の複数の述語中の変数から代入されるが、等号条件テストが述語間にまたがらない (特徴A) ルールで生じる。即ち、図6.4上図では下位のスイッチノード  $S_a, S_b$  の値が0となり束縛 $v_1(\phi)$ から $v_3(\phi, a), v_5(\phi)$ から $v_6(a, \phi)$ と



Error solution :  $\{v2(a), v4(a)\}$ .



Correct solution :  $\{v1(b), v2(a), v4(a), v5(b)\}$ .

図6.4.  $\phi$  による束縛が誤りの解を導く場合（上）と、その予防策（下）.



代入が行われた結果、不十分な解仮説{v2(a),v4(a)}で  $g(a, a) :- v5(a, \phi), v3(\phi, a).$ と  $g(a, a)$  が誤って導かれてしまう.

これを防ぐための手法について説明する. これを防ぐには,  $S=1$ の時は必ず  $S_1, S_2$ の少なくとも一つの値が1になる様な機構を与えておけばよい.

この様な機構は, 上述の特徴Aを有する各スイッチノードSと, そのすぐ下位でフィルターの役割を担うノードFをリンクで結び,

$0 \leq F(=S_1+S_2-S/2) \leq 1$  (Fに接するスイッチノードがk個の時

$$F = \sum_{i=1}^k S_i - S/2 \quad (6-3)$$

とすることで実現できる.

こうして  $S_c=1$ の時  $S_a, S_b$ の少なくとも一つの値が1とすればゴール証明に使用するルール内の変数への  $\phi$ の代入を禁止することができる.

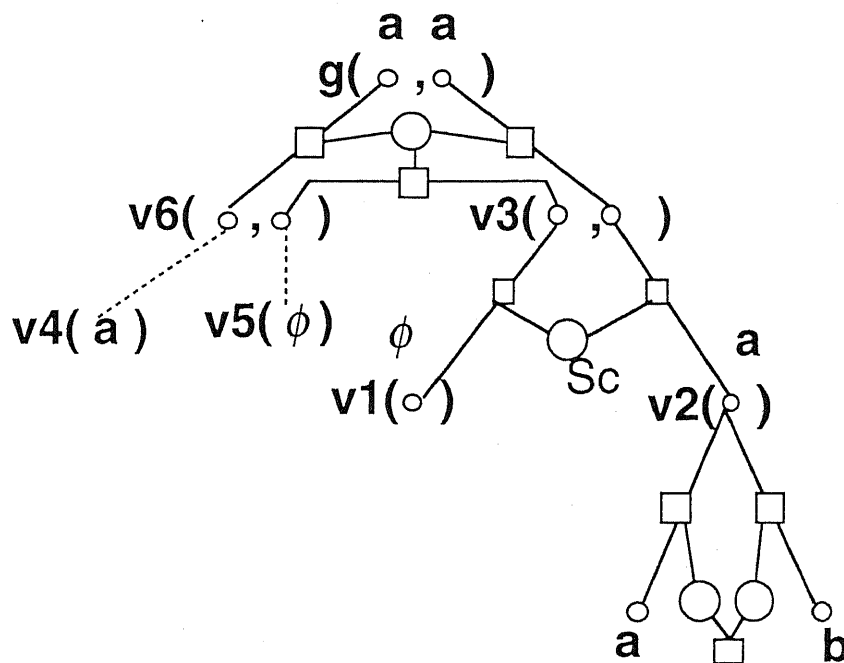
この新しく付加された枝 (枝  $S_cF$  など) を命綱と呼ぶことにする. これらが必要なだけ無ければ, 推論は誤りをおかしてしまうからである.

STM中で特徴Aを有する最上位のルールを起点として, 命綱をネットワークの下方に向けて延ばして行き (実際には2つの命綱が連結されることはないが, ある命綱Aのフィルタノードと別の命綱Bの一端のスイッチノードがSTM内で同一のヘッド述語の直接下になるときBはAの延長であるという), 次の停止条件で延長をストップする.

### 命綱延長の停止条件

命綱の一端にあるフィルタノードが仮説のものである (仮説単位に含まれる) とき, それより下にはスイッチノードがないので命綱は延長しない.

ところが、ネットワークの葉ノードまで達しても尚、仮説単位がそこに無い場合がある。即ち、図6.5の様に、葉の述語の具体例である様な仮説が元の候補仮説集合に含まれない場合である。先のSTMの段階的拡大の手順では深さを一層毎に増して行くだけであるので、この様な状況は頻繁に生じる可能性がある。



**Error solution : {v2(a), v4(a)}.**

図6.5. v1の具体例である仮説が存在しない場合。このときv1(φ)も許されるので、やはりg(a,a)が先と同じ様に導かれてしまう。

このように具体例である仮説のない様な述語が前件部にあるホーン節は、その述語から後件への代入が指定されているならば発火（スイッチノードがONとなった状態）してはならないという制約を与えることでこの問

題を解決した。定式的には次の様になる。

### 命網延長の停止条件（追加）

命網の下端となるべきフィルタノードが仮説単位内に見つからない場合は、現時点で命網の最下端のスイッチノードの値を0に固定してそれより下には命網の延長を行わない。ただし、この固定はSTMの次の拡張の際に解く。

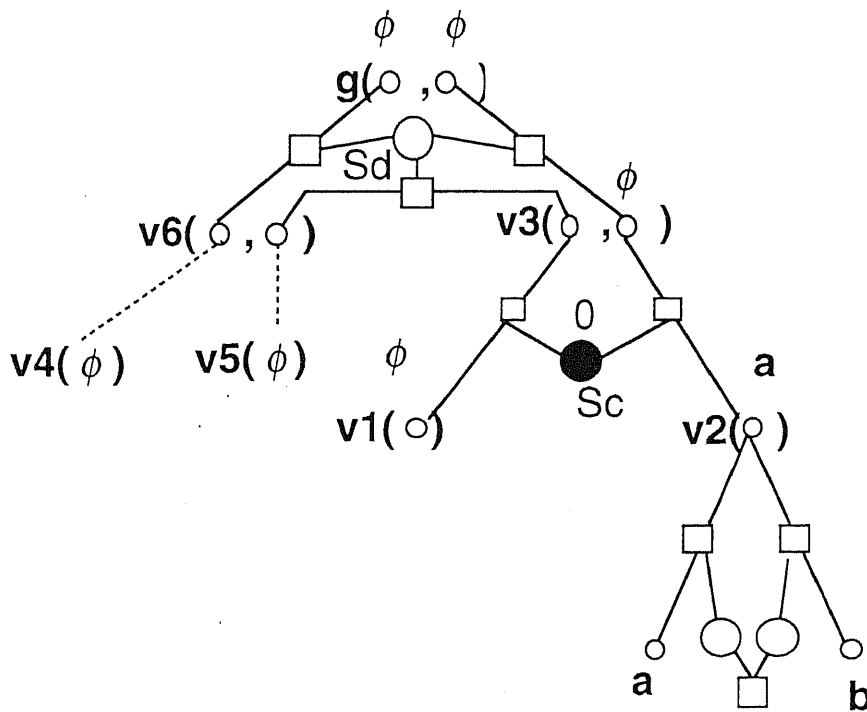


図6.6. v1の具体例である仮説が存在しない場合でも、Sc=0とすることによって命網延長の停止が可能となる。

即ち、もし発火すると再び上記の問題を起こしてしまうが、発火しないのであれば問題が解消される。例えば、今の例ではSc=0のとき、図6.6右

下部に当たるホーン節の後件部は $v3(\phi, a)$ ではなく必ず $v3(\phi, \phi)$ となる。もし左側のホーン節前件に同様の問題が無いならば、" $\phi = \phi$ " の成立で $Sd=1$ となることは起こらない。又、左側に同様の問題があっても、左側と同様の対策によって $v5(\phi, \phi)$ となるので、" $\phi = \phi$ " の成立で $Sd=1$ となっても $g(\phi, \phi)$ が導かれるだけであり、ゴールに $\phi$  束縛を許さない制約を設ければこれによって誤った証明が出力されることは起こらない。即ち、ある述語の一部だけが $\phi$  束縛をおこすことは決して起こらなくなる。ここで扱う例はゴールが具体例なのでもともとこの制約は与えられていることになるが、一般の場合にもゴールの変数のうちどれか任意の一つを選び、そのビットベクトル  $(x1, x2, x3, \dots, xm)$  に対して制約 " $1 \leq x1+x2+ x3+ \dots +xm$ " を与えておけばゴールに $\phi$  束縛を禁止したことになる。これに要する追加のノードはフィルタノード1つだけであるので、大きなオーバーヘッドとはならない。

### 6.3.6. 矛盾制約の表現方法

矛盾制約としては、2リテラル間の矛盾のみ扱う。3リテラル以上はその合成として扱えるからである。

#### (タイプ1) 同一の述語記号間の矛盾制約

制約不等式の拡張により、同一の述語記号での異なる変数値束縛を禁ずる矛盾制約も簡潔に表現できる。即ち、

$$\text{inc} :- p(X), p(Y), X \neq Y. \quad (6-4)$$

は $X=\phi, Y=b$ の場合を許すのにも関わらず $X=Y$ と簡潔に表せる。これは、 $X=b, Y=b$ であっても、これらの値を実際にゴール証明に用いるかどうかは $p$ を前件に含むホーン節のスイッチで決まるからである。等号成立時に代

入スイッチを0又は1としたことの副次的なメリットと言える。取扱いの困難な矛盾制約の一つである同一の述語記号への異なる変数束縛の禁止[牧野93]がこの様に単純に扱えるのは、式(6-2)で $T^P_1=T^P_2=\dots=T^P_n$ が満足されても $S=1$ ではなく $S \leq 1$ としたことによる。

#### (タイプ2) 異なる述語記号間の矛盾制約

$\text{inc} :- p(X), q(Y), X \neq Y.$ の形の制約は上同様である。さらに、

$$\text{inc} :- p(X_1), q(X_2), r(X_3), s(X_4), \dots, X_1 \neq X_2 \text{ or } X_2 \neq X_3 \dots X_{n-1} \neq X_n.$$

の様に条件部が増えても、条件リテラルと同じ個数の同様の等号制約で繋がればよいので、ノード数は比較的少数の増加で抑えることが可能である。しかし、

$$\text{inc} :- p(X_1), q(X_2), r(X_3), s(X_4), \dots, X_1 = X_2 = \dots = X_n. \quad (6-5)$$

を扱うにはタイプ1に比べるとネットワークが複雑になる。

この制約の記述には、以下の様に $X_i$ 間に等号成立を禁止することになる。まず、この制約を表すネットワークの階層構造において、下層では $X_1$ と $X_2$ の様に隣合う変数間の等号成立、不成立をチェックする。そして、 $n$ 変数を繋ぐ $n-1$ 個の中で一つでも等号不成立のものがあれば、この制約は満たされる。下層は図6.7に示した2通りの構築法が考えられる。(a)は各変数の桁(ビット)数分の値を合成( $X_1 = 0011$ を $0*1/2 + 0*1/2^2 + 1*1/2^3 + 1*1/2^4$ などと)してから合成してできた値の間の等号関係を調べるものである。又、(b)は、各ビット成分の等号関係を調べてからその結果を桁数分の合成(全てのビットで等号が成立すれば1, 1

ビットでも不成立ならば0などと) するものである。図6.7はバブル伝播ネットワークのノード数が比較できるように記したものであるが、(a)の方が(b)よりもコンパクトなネットワークが得られることが視察からも解かる(この図では(a)が33ノード、(b)が46ノードであるが、各述語がSTM中に複数回出現したりするとさらに差は広がる)。

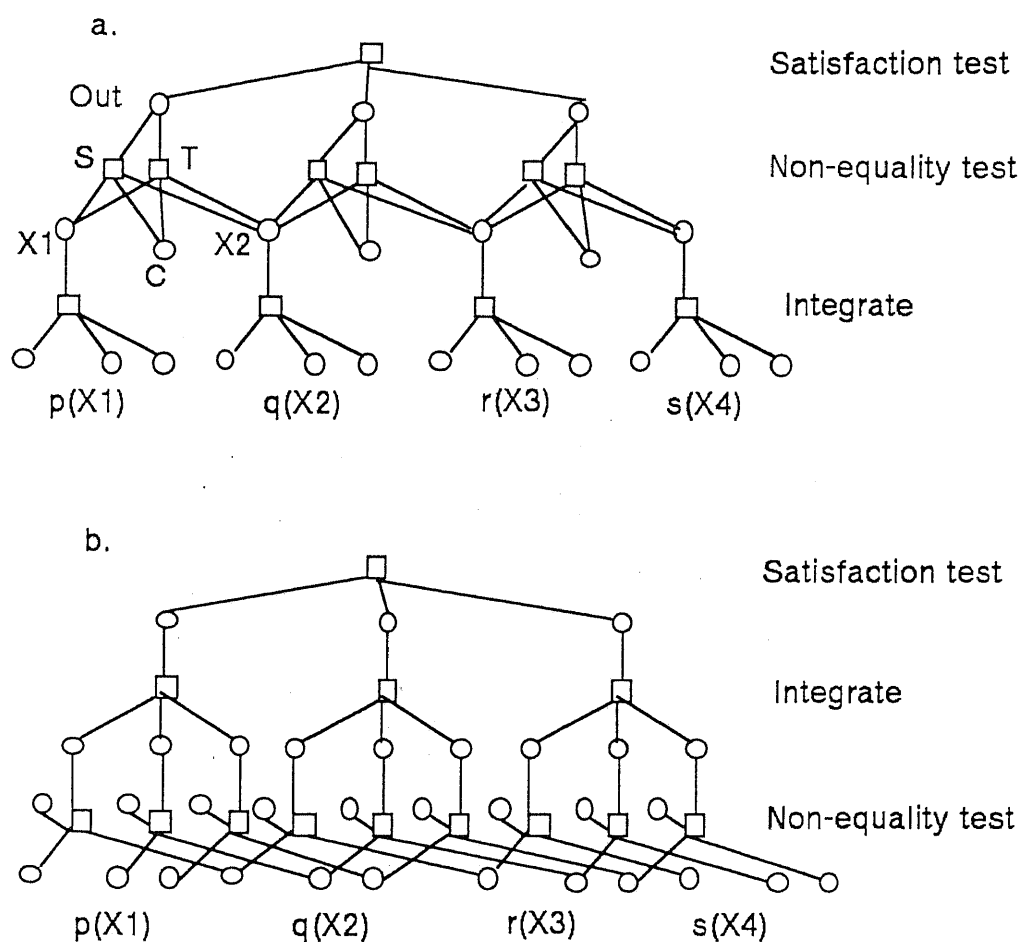


図6.7. タイプ2の矛盾制約を実現する2種類のバブル伝播ネットワーク。

そこで、(a)のみの数式を示す。Integrate部は上述の通り、Satisfaction test部については子の構造ノード(等号不成立で1)の和を1以上とすること

によって実現できる．残りのNon-equality testの部分の数式を示す．

値の合成 (Integrate) 後の変数  $X_1, X_2$  を比較するとき,  $In, Out$  をそれぞれ  $X_1 - X_2$ , および等号成立なら0, 不成立なら1の値をとる変数とする． $In$  は連続値変数で  $Out$  は最終的には離散値となる0-1変数である．桁数を  $bits$  と書くことにすると,  $Out=1$  となるのは  $In \leq -1/(2^{bits+1})$  と  $In \geq 1/(2^{bits+1})$  の2つの領域に分かれるので, 先にホーン節の条件部の真理値を代入スイッチの値とした時と同様に, 一つの補助変数を設ける必要がある．これを変数  $C$  とする． $In, Out, C$  の3変数の値が図6.8の2平面に含まれる領域に含まれるならば求める制約を満たす．

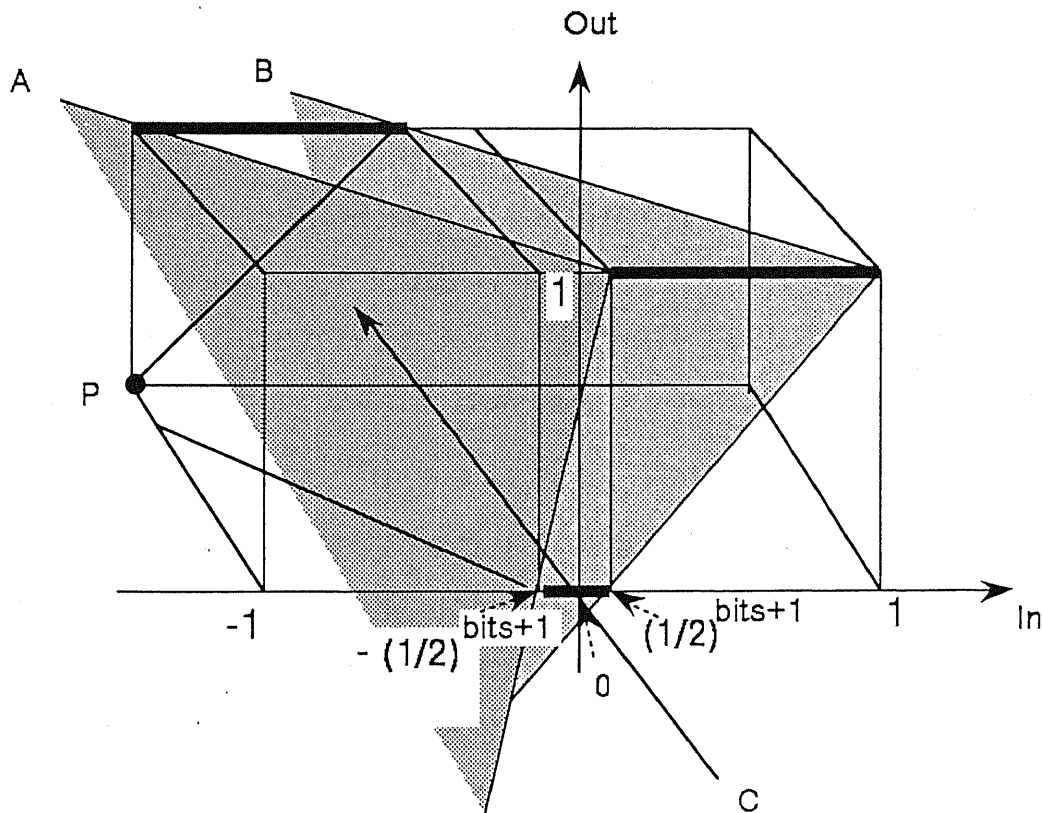


図6.8. 図6.7(a)の不等号テスト部の制約空間

平面A, Bはそれぞれ次の制約式の上下限となっており, これらをネット

ワーク化することによってタイプ3の矛盾制約を表すことができる。

$$In \geq \frac{Out}{2^{bits}} - \left(1 + \frac{1}{2^{bit+1}}\right) C - \frac{1}{2^{bit+1}}, \quad (6-6)$$

$$In \leq \left(1 - \frac{1}{2^{bit+1}}\right) Out - \left(1 + \frac{1}{2^{bit+1}}\right) C + \frac{1}{2^{bit+1}}. \quad (6-7)$$

図6.8で太線および点Pが以上の制約を満たす座標であり，これらのうちX1, X2の合成前の各ビット値, Out, Cの全てが0-1整数値となる点ではこの矛盾制約が満たされる．従って，Inは必ずしも整数値とするには必要はないことから混合整数計画問題を解くこととなる．しかし，その為にNBP法の探索フェーズに3.4.2に示した改良を施す必要はない（簡単な改良であるが，これを用いずに統一的に整数計画法の枠組みが適用できる方がよりシンプルである）．その理由は，Inはノードとしては存在せず，実際に非整数値をとることが許されるノードは合成後のX1およびX2であるが，両者の差が1又は-1となることは有りえないからである．即ち，両者の値の範囲は本来 $[0, 1 - 1/2^{bits}]$ にあって，仮に一方でもバブルを受け取って値が1となるならば合成前の0-1変数のビット値が $[0, 1]$ の範囲から外に出ていることになる．そのような場合には，バブルのこの受取りは起きない様に相対深度最小のノードを伝播パスから選んでバブル伝播先としていたのであった．従って，最後にこれら合成後の変数値ノード以外の構造ノードが全てバブルを受け取った時点で探索フェーズを終了するという変更だけで十分である．

以上から，図6.7においてノードS, Tを平面A, Bに対応させて伝播率を

$$p(S, X1) = 1.0, \quad p(S, X2) = -1.0, \quad p(S, Out) = (1/2)^{bits},$$



$$p(S, C) = -\{1 + (1/2)^{\text{bits}+1}\},$$

$$p(T, X1) = 1.0, \quad p(T, X2) = -1.0, \quad p(T, \text{Out}) = 1 - (1/2)^{\text{bits}+1},$$

$$p(T, C) = -\{1 + (1/2)^{\text{bits}+1}\},$$

と与えてこれをネットワーク化する。

本研究において試みた範囲では、タイプ2の意味をそのままネットワーク化する手段としてはここに示したものが最もコンパクトなバブル伝播ネットワークを生成できる。しかし尚かつ、一つの矛盾制約を扱うためにこの方法を用いるのはノード数を余計に増やしてしまう結果となる。これは、タイプ1の等号制約と異なり、本質的には全てのビットを統合する機能が不等号制約では必要となるためである。

多くの場合、少しでも複雑なタイプ2の制約はエルブラン展開を行って以下に述べるタイプ3およびタイプ4の制約として取り扱う方が容易である。ただし、一述語の具体例である仮説が特に多い場合には、それらを一般的な形でまとめたタイプ2として取り扱う方法が効果を発揮する。

### (タイプ3) 基底アトム又はその組み合わせの禁止

具体的な値の代入されたアトム間での矛盾については命題論理同様の不等式制約を用いて表すことができる。

ここで、もう一つ別の制約表現がある。それは、ある述語について特定の仮説（基底アトム）のみを禁止するというものである。例えば、ボートレースに出場する50クルーを3艇レースに組分けする時、A,B2チームは友好関係が深いので応援を考慮して別の組としたい、という意味の

`inc :- same_race(A, B).`

を考える。この場合、後述のX表現による述語版NBP法を用いると、

```
same(X, Y) :- x_is(X), y_is(Y).
```

なるホーン節によって次の様書き表すことが可能となる。

```
inc :- x_is(A), x_is(B).
```

ただし、 $x\_is(X)$ ,  $y\_is(Y)$ は特定のヘッドについてそれぞれ変数名X, Yの変数への代入値がそれぞれX, Yであることを表しており、一般性がない点で通常の述語とは異なる。このX表現による制約表現も、エルブラン空間への展開に比べて効率的な場合がある（6.5節参照）。

（タイプ4）矛盾制約を表すフィルターノードを人為的に与える場合

例えば、複数の等号成立を禁止する

```
inc :- high_tool(Z, Y), high_tool(Y, X), T0(Y), T1(X).
```

の様な場合、バブル伝播ネットワークを構成した場合にどの様な位置の仮説間で矛盾を禁止されているかが容易に理解できる場合にはその様にする。後述の例2) に出現するこの制約は、そのままタイプ2) として表すと最低100ノード以上の増加を必要とする。これに対して、この制約が実は深さ一つおきに $high\_tool(\\_, \\_)$ を説明に用いることを禁ずるものであることを用いれば深さ $2n$ に対して $n$ 個のフィルターノードで表すことが可能となる。この様な記述方法は、問題の記述から解を得るまでに人手を伴う点で半自動的といえるが、計算機が指数オーダーかかる仮説推論の問題を解いていることに変わりはない。又、一見非常に複雑な制約記述が出現するのは、意味的には単純なことを記そうとしていることが多く、

その場合は意味を知る側にとっては知識の意味を直接反映するバブル伝播ネットワークでこれを指定するのは容易なことである。では、どのような場合がこのタイプと判断すべきであろうか。これは、このタイプを半自動化の対象とする限りシステムの外の考察であるが、半自動＝対話型とみることが可能であるので、対話型人工知能に対する筆者の立場を明らかにする意味で述べておきたい。

人工知能を強い人工知能と弱い人工知能に分類する観点がある。前者は、処理の一切を計算機に任せるという信頼感に基づく考え方であるのに対し、後者はそれはそもそも不可能であるとの前提を受け入れて部分的に計算機に任せる考え方である。例えば、人の情報処理を人工知能で実現するというのは、計算機のパワーによって究極的には機械で人のあらゆる知識処理が実現可能であると見做す点で強い人工知能に属する。又、人の行為の一部をシミュレーションするのに用いるというのは、後者ということになる。仮説推論の様に計算量を費やす種類の問題の場合、人よりも計算能力において優れる計算機に一任するのは自然な考え方であろう。この点が、この種の問題解決に対話形式のシステムが用いられなかった理由であると考えられる。

しかし、これは問題を計算機に与えた後の話であって、問題の入力段階では人と計算機で扱いやすい言語が異なることは考慮の必要がある。例えば、ここで用いている例がそうである。ここでは高性能のツールを連続して用いることを禁止したのであり、設計する目的の工程自体の時間の流れに対して制約を設けるわけである。ところが、そもそもルールを制約とみなす本論文の立場は、制約ロジックプログラミングや制約の力学と同様に情報の時間的な流れを捨象しようとするものである。即ち、記述したい内容と記述する側の立場に大きな開きが存在する。

では、この種の問題は制約のアプローチでは解けないのであろうか？  
筆者のこれに対する答えは、解ける、である。その理由は、時間の流れ

を含むルールも、一つの一般的な流れのモデルを設定することにより時間に直交する制約とすることが可能となるからである。実際、実世界には多くの因果関係が存在し、それらを原因から結果へと連鎖的にたどることによって歴史が構築されている。しかし、ある時点で歴史を振り返るならば、時間と反対の方向に推論することも可能である。例えば、第二次世界は人為的に予め作られたプログラムの通り進んだというのは本当であろうか。だとすれば、このプログラムの上に、戦争初期のや戦争末期の日本の状態がそれぞれの時点に対応する位置に置かれていることになる。プログラム作成者は予め（時点×状態）の要素間に矛盾を指摘することも可能であろう。これが、一般的な時間の流れのモデルである。全ての可能な時間の流れはこのモデルのある具体化によって再生が可能であるが、歴史以前にモデルはそこに存在するのである。

この一般的な時間の流れのモデルは、このプログラム制作者が行えたところで想像したと同様の作業のためには不可欠であった。タイプ4の矛盾制約の与え方は、これと似た発想である。即ち、先に述語論理という一般的な形で記述された一般的な工程のモデルを与えておき、この上で制約を与えるのである。その結果、この種の問題に適したコンパクトなネットワークの上での推論が可能とすることができた。

計算機には一般的な時間の流れのモデル作成を任せ、その上に付加的に矛盾制約を与え、その後の計算は又計算機に任せるという考え方は、対話型の人工知能としては人の負担が小さな処理であると考えている。

（タイプ5）変数間の制約と呼べない  $\text{inc} :- p(X1), q(X2), r(X3), \dots$  の型

この場合は、変数間の等号や不等号の関係をネットワーク化せずに、単に条件部の各述語をヘッドとするBK単位のスイッチノード間にフィルターノードを設ければ扱うことができる。

### 6.3.7. 述語版NBP法の表現能力と推論の健全性

以上が述語版NBP法による知識のネットワーク表現の仕方であるが、この表現方法によってRange restrictedで関数なし (Function-free) のホーン節からなる知識ベース、及び変数間の不等号にたいする制約 (タイプ1の矛盾制約で、一種の関数と見做せる) の表現とこれを用いた健全な推論が可能である。

これを示すには、以下の2点を示せばよい。

- 1) 述語版NBP法によって、これらの表現が完全に可能であること。
- 2) ゴールを導くことができない解仮説もしくは矛盾する解仮説を出力することがない (健全性) 。

1)の証明 Range restrictedなホーン節を、2通りに分類する。範疇の第一は、ヘッド部に変数を含む場合である。range restrictedなホーン節の場合この変数はボディ部にも含まれるので、最も一般的には次の様に書ける。

$$\text{head}(X1, X2, \dots, Xn) \quad (6-8)$$

$$\text{:- body1}(X1, Y1, \dots), \text{body2}(X2, X1, \dots, Y2, \dots), \dots, \text{bodyk}(Xn, X2, \dots, Yn).$$

これは、式(6-9)の様に変形できる。

$$\text{head}(X1, X2, \dots, Xn) \quad (6-9)$$

$$\text{:- shared}(X1, X2, \dots, Xn), \text{condition}(X1, X2, \dots, Xn, Y1, Y2, \dots, Ym).$$

ただし、

shared(X1, X2, ..., Xn)はボディとヘッドにX1, X2, ..., Xnが共有されることを表し、condition(X1, X2, ..., Xn, Y1, Y2, ..., Ym)は次の様に、各変数がボディ部に出現する位置を表す (variable(x, y)で述語xのy番目を意味する) 。

$$\text{condition}(X1, X2, \dots, Xn, Y1, Y2, \dots, Ym) =$$

$$X1=\text{variable}(\text{body1}, 1) \ \&\& \ X1=\text{variable}(\text{body2}, 2).$$

$$\begin{aligned}
&\& X2=\text{variable}(\text{body}2,1) \ \&\& X2=\text{variable}(\text{body}k, 2). \\
&\quad \dots \\
&\& Xn=\text{variable}(\text{body}k, 1). \\
&\& Y1=\text{variable}(\text{body}1,2) \ \&\& \dots \\
&\& Y2=\text{variable}(\text{body}2,\dots \\
&\quad \dots \\
&\& Yn=\text{variable}(\text{body}k, n).
\end{aligned} \tag{6-10}$$

式(6-9)を全称限定すると，下のように変形できる(s.t. = such that ).

$$\begin{aligned}
&\forall (X1,X2,\dots Xn,Y1,Y2,\dots Yn), \exists S, \text{ s.t. } \{ \\
&\text{head}(X1, X2, \dots Xn):-\text{shared}(X1, X2, \dots Xn), S., \\
&S :- \text{condition}(X1, X2, \dots Xn, Y1, Y2, \dots Ym)\}.
\end{aligned}$$

Sは(X1,X2,...Xn,Y1,Y2,...Yn)と独立であるので，

$$\begin{aligned}
&\exists S, \text{ s.t.} \{ \\
&\forall (X1,X2,\dots Xn,Y1,Y2,\dots Yn) \\
&\quad \text{head}(X1, X2, \dots Xn):-\text{shared}(X1, X2, \dots Xn), S., \\
&\forall (X1,X2,\dots Xn,Y1,Y2,\dots Yn) \\
&\quad S :- \text{condition}(X1, X2, \dots Xn, Y1, Y2, \dots Ym)\}.
\end{aligned} \tag{6-11}$$

を経て，式(6-12)に変形できる．

$$\begin{aligned}
&\exists S, \text{ s.t.} \{ \\
&\forall (X1,X2,\dots Xn,Y1,Y2,\dots Yn) \\
&\quad \text{head}(X1, X2, \dots Xn):-\text{shared}(X1, X2, \dots Xn), S.,
\end{aligned} \tag{6-12}$$

$S:-\exists(X1,X2,...Xn,Y1,Y2,...Yn), \text{ s.t., condition}(X1, ...Xn, Y1, ...Ym)\}.$

第2の節の条件部は、例えば

$\exists X1, \text{ s.t.}\{X1=\text{variable}(\text{body1},1) \ \&\& X1=\text{variable}(\text{body2}, 2)\}.$

$\leftrightarrow \text{variable}(\text{body1},1)=\text{variable}(\text{body2}, 2)\}.$  の様に変換えられるので、式(6.12)は式(6.13)に変換えられる。

$$\begin{aligned} &\exists S, \text{ s.t.}\{ \\ &\quad \forall(X1,X2,...Xn,Y1,Y2,...Yn) \\ &\quad \quad \text{head}(X1, X2, ...Xn):-\text{shared}(X1, X2, ...Xn), S., \\ &\quad S :- \{ \quad \text{variable}(\text{body1},1) =\text{variable}(\text{body2}, 2). \\ &\quad \quad \&\& \quad \text{variable}(\text{body2},1) =\text{variable}(\text{bodyk}, 2). \\ &\quad \quad \dots \\ &\quad \quad \&\& \quad \text{variable}(\text{body1},2) =\dots \} \\ &\quad \} \end{aligned} \tag{6-13}$$

これは、式(6.1)の代入スイッチ（アトムSがスイッチノードに相当する）そのものである。

第二の範疇は、ヘッドが変数を持たない場合である。ヘッドが真に固定されている場合は、式(6-13)が次の様に変化するだけであるから、対応する全ての変数を統合制約の■ノードで結合すればよい。

$$\begin{aligned} &1 :- \{ \quad \text{variable}(\text{body1},1) =\text{variable}(\text{body2}, 2). \\ &\quad \&\& \quad \text{variable}(\text{body2},1) =\text{variable}(\text{bodyk}, 2). \\ &\quad \quad \dots \\ &\quad \&\& \quad \text{variable}(\text{body1},2) =\dots \} \end{aligned} \tag{6-14}$$

ヘッドが偽に固定されている場合は、先述の、ボディ部に変数を含むタイプ2及び5と、ボディ部に変数を含まないタイプ3の矛盾がそれに相当する。以上から、関数を含まないRange restrictedなホーン節は述語版のバブル伝播ネットワークによって完全に記述される。(証明終り)

次に、2)を示す。

2) (健全性) の証明 変数に相当する全てのビットベクトルは、 $\phi$ 又は与えられた可能な例の値をとる。なぜならば、range restrictedなホーン節の場合に限定するので、あるビットベクトルの値はこれをヘッドとするホーン節があればそのボディ部か、又は仮説に含まれている値がそれらのスイッチノードの値が1であれば代入されており、0であれば $\phi$ が代入されるからである。このうち、仮説の値が代入される場合に仮説の変数値が $\phi$ であることは有りえないので、ホーン節のボディ部から代入される場合を考える。

あるホーン節のボディ部からスイッチノードを経てそのヘッドの値に代入される場合、ボディ部の全ての変数値が $\phi$ 以外の場合には、先述のrange restrictedなホーン節の制約が実現されていることになる。一方、ボディ部に $\phi$ が含まれている場合は、先の $\phi$ 束縛の回避(予防)の方法(6.3.5節)で既に解決済みである。よって、述語版NBP法はゴールを導くことができない解仮説は出力しない。

最後に、互いに矛盾する仮説間には6.3.6節に述べた各タイプの矛盾制約が課されており、これを充足する限り矛盾する解仮説を出力することがない。この内、ボディ部に変数を含むタイプ2及び5と、ボディ部に変数を含まないタイプ3によって、range restrictedなホーン節表現による制約は網羅している。(証明終り)



### 6.3.8. 述語版NBP法の動作例

STMを得ると、命題論理の場合と同様にNBP法の動作を開始する（先の手順3）。手順3で実行可能解を得られない場合は、現在の短期記憶内の知識ではゴール証明に不十分であることが分かる。そこで、ゴール証明に関与する可能性のある単位を更にLTMからSTMにコピーする（手順2）。こうして、制約充足の準最適解を得るまでネットワークの拡張を繰り返す。例1の場合のSTMの変化を図6.9に示す。ただし、図6.9の番号0～3は下の各状態とする。

状態0→1. 手順1, 2でSTMを生成する。仮説 $h2(a, b)$ がSTMに結合される様子は既に述べた。この時、ノード $S'3$ は $S'3 = 1$ の時仮説 $h2(a, b)$ の使用を意味するので、仮説 $h2(a, b)$ の重みをノード $S'3$ の重みとする。この段階では、手順3で解仮説を求めることができない。実際、現時点のSTMの仮説とルールだけではゴールは証明できない。

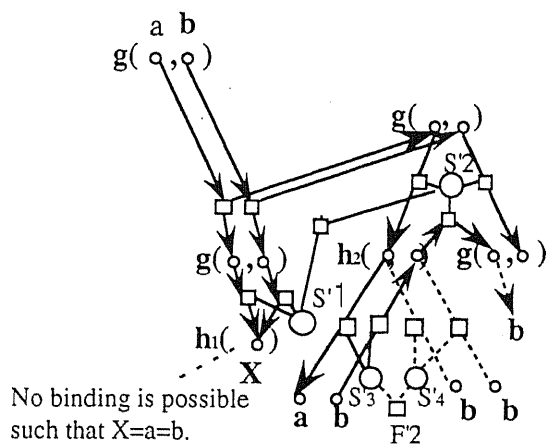
状態2. 次に、先の仮説 $h2(a, b)$ からの束縛情報（状態1の $\rightarrow b$ による限定の結果、 $g$ を定義するBK単位と、仮説 $h2(b, b)$ を結合する。まだ解仮説は得られない。ノード $inc1$ は矛盾制約4)を制約 $X1=X2$ で実現したものである。矛盾制約4)は $X1=\phi, X2=b$ の場合を許すので $X1=X2$ では誤りに見えるが、 $X1=b, X2=b$ となっても $S'1=0$ とすれば $X1$ の値はそのすぐ上の $g$ に代入されず $X1=\phi$ と同等の結果を得ることができる(6.3.6"タイプ1"参照)。

状態3. 次の拡大では $h2$ の仮説が両方結合される結果、解 $\{h2(a, b), h2(b, b)\}$ が得られる。同時に、斜線を施した構造ノードが値1をとるのでゴール証明に用いる仮説及びルールが決まる。即ち、次の順序で仮説（下線部）からゴールが証明される。

0. The initial STM.

a b  
g( o, o )

1. Extend h2, but result in failure.



2. Extend g, but result in failure.

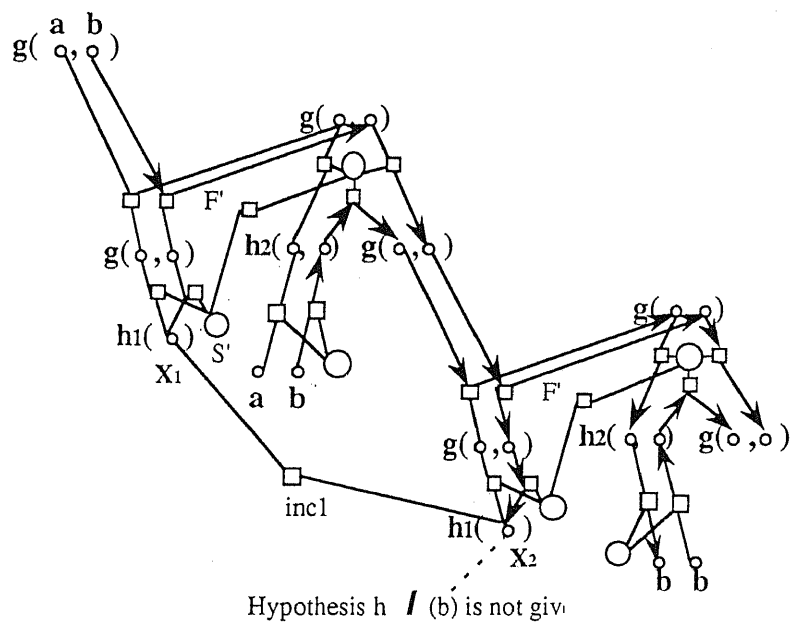


図6.9. 短期記憶の拡大を伴う述語版NBP法による仮説推論の進行（次頁へ続く）。

3. Entend g and h1, and result in success.

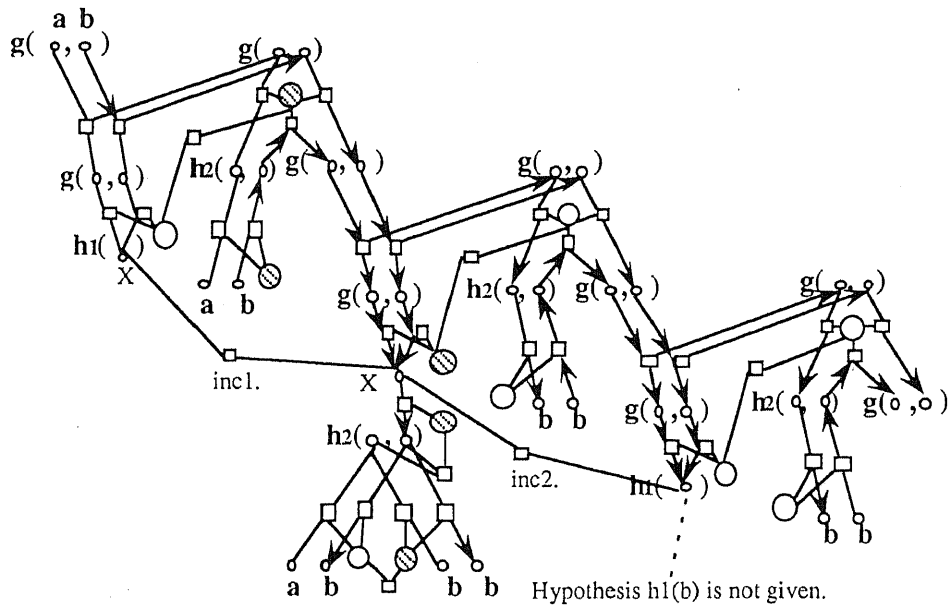


図6.9. 短期記憶の拡大を伴う述語版NBP法による仮説推論の進行（前頁から続き）。

$h1(b):-h2(b, b).$

$g(b, b):-h1(b).$

$g(a, b):-h2(a, b), g(b, b).$

この様に、代入スイッチの導入とSTMの段階的な拡張がネットワークの不必要な大規模化を避ける効果を生む。しかし、状態1と2の様な解が得られない冗長なプロセスが含まれることになる。この非効率性を軽減するため、予め設定した定数 $c$ から $[ce]+1$ 個以上のルールが新しくSTMに結合するまで手順2を繰り返してから手順3を実行するようにする。ただし、 $[x]$ は $x$ 以下の最大の整数値、 $e$ は手順3失敗時に全構造ノードの値を丸めた（四捨五入した）場合の制約違反度（定義は[Balas80]参照）とする。定数 $c$ は今のところ経験的に設定しており、次章実験では0.3とした。

## 6.4. 例題による実験とその評価

以上に述べた述語版NBP法をSun SPARCstation 10 上に実装した。手続きはC言語を用いて記述し、変数領域として200MBを確保した。ここでは、ベルトコンベアー等を用いた一連の流れ作業生産において、製品の不良品率が最小となる様にある制約の下で各生産段階への作業機械の割当を決定する例題を扱う。

⊙ : High-performance tool.

● : Low-performance tool.

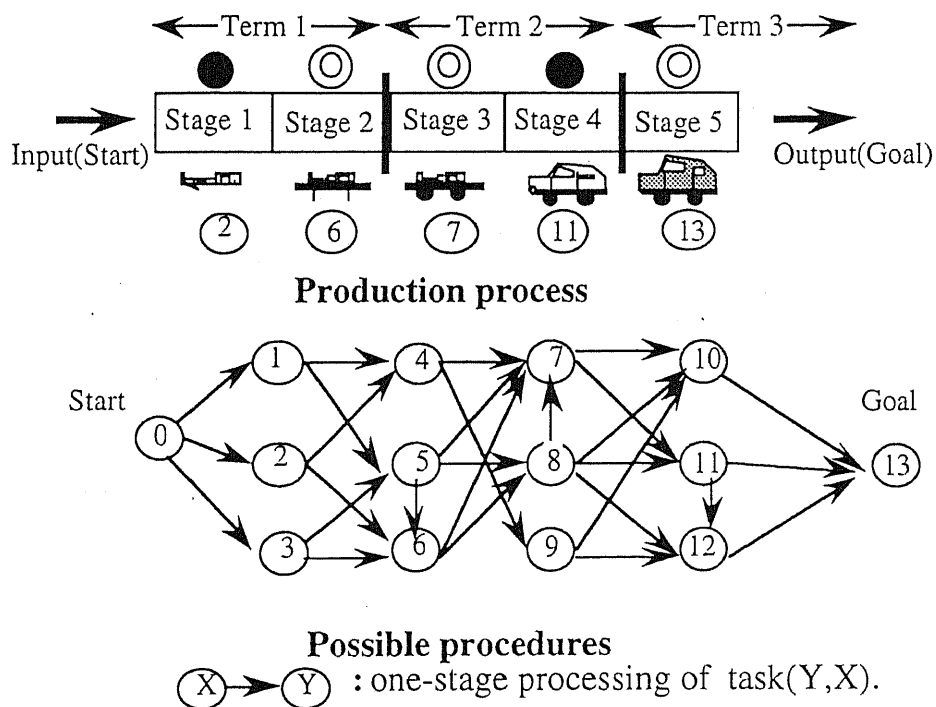


図6.10. 高性能ツールの使用に制約の課された流れ作業の最適化問題.

下の図の各番号は中間生産物.

即ち、図6.10の①、②...は中間生産物で、それらの間を人手や機械を用いて図6.10下部の様に遷移する。同じ製品を製造するにも組立てる工程によって必要なツールは異なるので、一つの生産工程と各段階でのツールの選択が必要となる。この選択によって最終生産物の性能も左右されるが、高価格の高性能ツールを全ての作業段階に設けるのは経済的ではない。そこで、工程を作業開始から2作業段階毎に1期間（例えば”塗装期間”や”検査期間”）とし、高性能ツールの使用は1期間あたり高々一つとする制約下で最終製品の誤動作率 $R$ を最小化したい。作業段階 $i$ のエラー率を $r_i$  ( $\ll 1$ )、工程中の作業段階数を $n$ とすると

$$R = 1 - \prod_{i=1}^n (1 - r_i) \\ \equiv \sum_{i=1}^n r_i.$$

となる。このことから、各単位作業で使用可能なツール（高性能と低性能）を各要素仮説とし、エラー率をその重みとして仮説の重みの和を最小化する下の例2の様なコストに基づく仮説推論に本手法を適用した。

例2) 背景知識:  $T0(n)$ .

$make(0)$ .

$make(X) :- task(X, Y), make(Y)$ .

$task(X, Y) :- high\_tool(X, Y)$ .

$task(X, Y) :- low\_tool(X, Y)$ .

$T0(X) :- task(X, Y), T1(Y)$ .

$T1(X) :- task(X, Y), T0(Y)$ .

$inc :- high\_tool(Z, Y), high\_tool(Y, X), T0(Y), T1(X)$ .

候補仮説:  $high\_tool(1, 0)$ .  $weight = 1$

(高性能ツールで①から②を製造するとエラー率が1/1000),

low\_tool(1, 0). weight = 1

(低性能ツールで①から②を製造するとエラー率が5/1000),

ゴール: make(n) (nは最終生産物の名前として与えられた番号)

ここで, make(0)は作業開始時の状態を表す. 又,  $T_0(X)$ ,  $T_1(X)$ でそれぞれ作業開始から偶数番目, 奇数番目の作業段階での中間生産物Xの生産を表すことにより, 背景知識の矛盾制約で高性能ツール割り当ての制約を実現する. この矛盾制約は, STM中でゴールから $(2n-1)$ 番目と $2n$ 番目 ( $n=1, 2, \dots$ )のtask(X, Y) :- high\_tool(X, Y).のスイッチノード間のフィルターノードで実現される.

図6.11にnを変化させた本手法の計算時間を示した. この計算時間は, 次の2つの特徴を示している.

- 1) 候補仮説数の多項式時間で準最適解を計算する.
  - 2) 同じ候補仮説数でもゴール証明に必要な要素仮説数に対し計算時間が単調増加する. 即ち, ゴール指向の推論が可能となっている.
- 尚, 最右下の解が3番目の準最適解であるのを除けば全て結果は最適解又はその次の解であった.

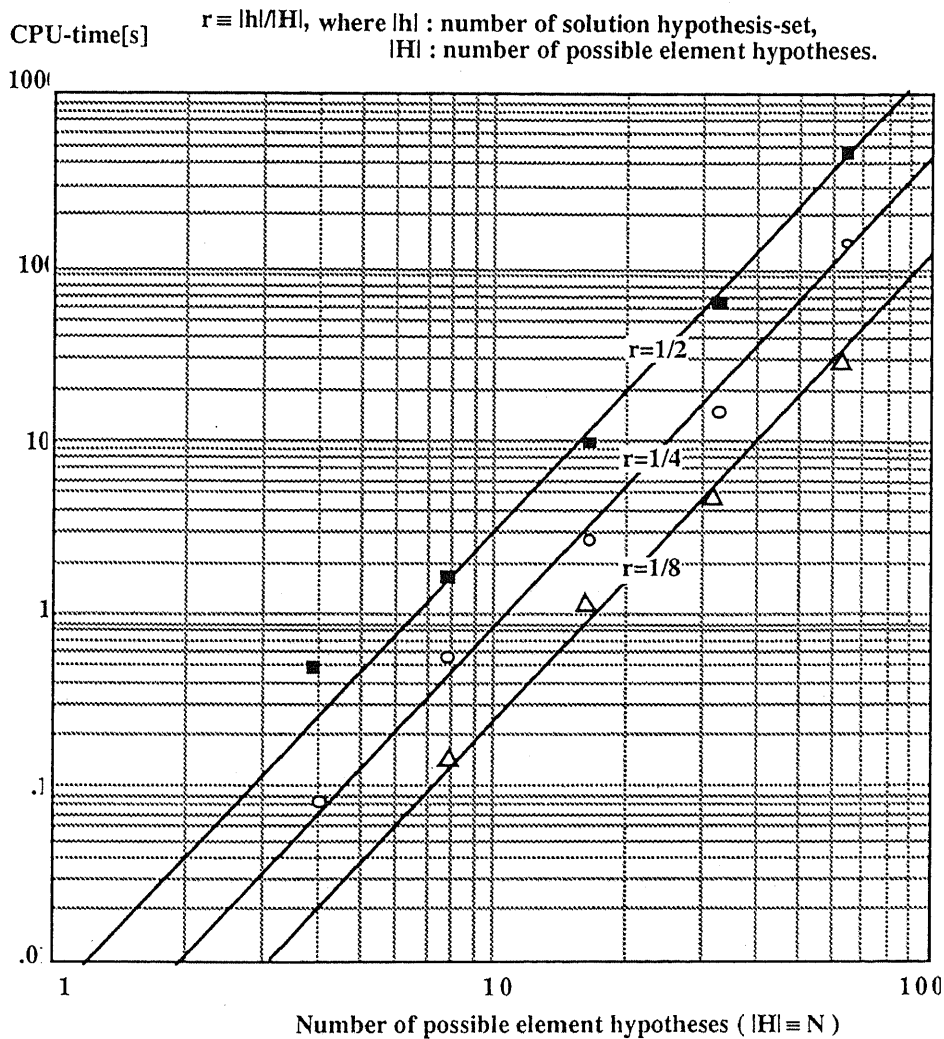


図6.11. 述語版NBP法による仮説推論の推論時間.

## 6.5. 計算時間の理論的裏付けと、例による理論の説明

理論的に上のデータを解釈してみると次のようになる。この例では、ゴールの証明に用いる仮説数におよそ比例する形でBPNを拡張して行く。となれば、命題論理版での推論時間 $N^2$ オーダーはここでも成立することになる。ただ違うのは、述語版ではさらに変数の具体例として有りうる候

補数の対数に比例したネットワークの重なりがある点である。即ち、 $a1=00..01$ ,  $a2=00..10$ ,  $a3=00..11$  と二進表現するには  $a_n$  まで表すのに  $\log_2 n$  桁のビットベクトルを用いなければならない。この例題の場合には中間生産物の数がこの  $n$  になり、それに大体比例する数の仮説を用いるので  $n$  と  $N$  が比例関係となる。この結果  $(N \log_2 N)^2$  オーダの計算時間となることが予想される。実際には  $N^3$  程度の時間となっており、ほぼこの予想と一致している。

ここに記した解析の結果、推論時間が一般的な述語論理表現と等価な BPN の大きさの多項式オーダーとなった点は、変数および各変数の具体例の数が多い場合、命題論理による記述に比べて述語論理の記述が効率的であることを示す。即ち、この様な問題を解くには命題論理版のネットワーク化バブル伝播法よりも述語版の方が具体例が多くなると効率的である。この点を理解する為に、例題を取り上げる。これは、震度6の地震の被災地および震源地となる確率の高い地点を知る問題である。

例3)

背景知識：

quake6(Y4):-distance(Y1,depth(Y2),magnitude(Y3), bad(Y3,Y1,Y2), soft(Y4).

quake6(Y4):-distance(Y1), depth(Y2),magnitude(Y3), terrible(Y3,Y1,Y2),

usual(Y4).

bad (X, Y, Z)           :- strong(X), nearer(Y, Z).

bad (X, Y, Z)           :- strong(X), near(Y), shallow(Z).

bad (X, Y, Z)           :- great(X), ordinary(Y, Z).

terrible (X, Y, Z)       :- great(X), nearer(Y).

terrible (X, Y, Z)       :- great(X), near(Y), shallow(Z).

nearer (Y, Z)           :- near(Y), middle\_depth(Z).

nearer (Y, Z)           :- middle\_distance(Y), shallow(Z).



ordinary(Y, Z)           :- middle\_distance(Y), middle\_depth(Z).  
 ordinary(Y, Z)           :- far(Y), shallow(Z).  
 ordinary(Y, Z)           :- near(Y), deep(Z).  
 inc. :- shallow(X), great(Y).  
 inc :- shallow(X), far(Y).  
 shallow(10)., shallow(20)., shallow(30).,  
 middle\_depth (50)., middle\_depth (100)., middle\_depth (200).,  
 deep(300)., deep(400)., deep(500), deep(1000).  
 near(10), near(20), near(50), near(70),  
 middleid\_distance(100), middle\_distance(200), middle\_distance(300),  
 far(500), far(700), far(1000),  
 weak(3), weak(4), middle\_energy(5), middle\_energy(5.5),  
 strong(6), strong(6.5), strong(7), great(7.5), great(8), great(8.5),

仮説集合：

depth(10)	(weight = 5)	distance(10)	(weight = 5)
depth(20)	(weight = 5)	distance(20)	(weight = 5)
depth(30)	(weight = 4)	distance(50)	(weight = 4)
depth(50)	(weight = 3)	distance(70)	(weight = 3)
depth(100)	(weight = 2)	distance(100)	(weight = 2)
depth(200)	(weight = 1)	distance(200)	(weight = 1)
depth(300)	(weight = 3)	distance(300)	(weight = 1)
depth(400)	(weight = 4)	distance(500)	(weight = 2)
depth(500)	(weight = 5)	distance(700)	(weight = 2)
depth(1000)	(weight = 7)	distance(1000)	(weight = 2)
magnitude(3)	(weight = 1)	magnitude(6.5)	(weight = 6)
magnitude(4)	(weight = 2)	magnitude(7)	(weight = 7)
magnitude(5)	(weight = 3)	magnitude(7.5)	(weight = 8)
magnitude(5.5)	(weight = 4)	magnitude(8)	(weight = 9)
magnitude(6)	(weight = 5)	magnitude(8.5)	(weight = 10)

hard(Hongo), hard(Koukyo), hard(Dencho), hard(Maebashi),       ...(10 cities)

usual(Hakusan),usual(Yotsuya),usual(Shinjuku),usual(Ohyanomizu),...(10 cities)  
soft(Funabashi), soft(Ikebukuro), soft(Yokohama), soft(Nezu), ... (10 cities)

ゴール：quake6(Y4).

この場合，属性値は必要上細かく（10区分に）カテゴライズされており，エルブラン展開すれば(Y1,Y2,Y3,Y4)の具体例だけで $10^4$ アトムからなる．又，指数オーダーの計算量の原因となる矛盾制約も，命題論理では great(X)にあたる100アトムと shallow(Y)の100アトムで $10^4$ 個，および far(X)にあたる100アトムと shallow(Y)の100アトムで $10^4$ 個の制約が加わる．このノード数ではネットワーク化バブル伝播法が良い近似解を得ることすら難しくなる．

これに対して，述語版バブル伝播ネットワーク（図6.12）で表現する場合には仮説として最下位の変数一つに対し10個の例が割り当てられる．これらは●に10個ずつなので280アトムとなり，それが $\log_2 10=4$ 桁のビットベクトル分重なるので，1120ノードとなる．又，●で表すスイッチノード（この例でのみ非基底構造ノードと区別が必要）は1つずつなのでこれらは15個である．最後に○で表す中間ノードおよび補助ノード（この例でのみ基底構造ノードと区別が必要）が4個ずつなので76個となるので，3種類の構造ノードの合計個数1211個となる．矛盾制約も，述語論理ではタイプ5の制約を3箇所にもうければよく，それに必要なノードの個数は僅かに3個で済む．

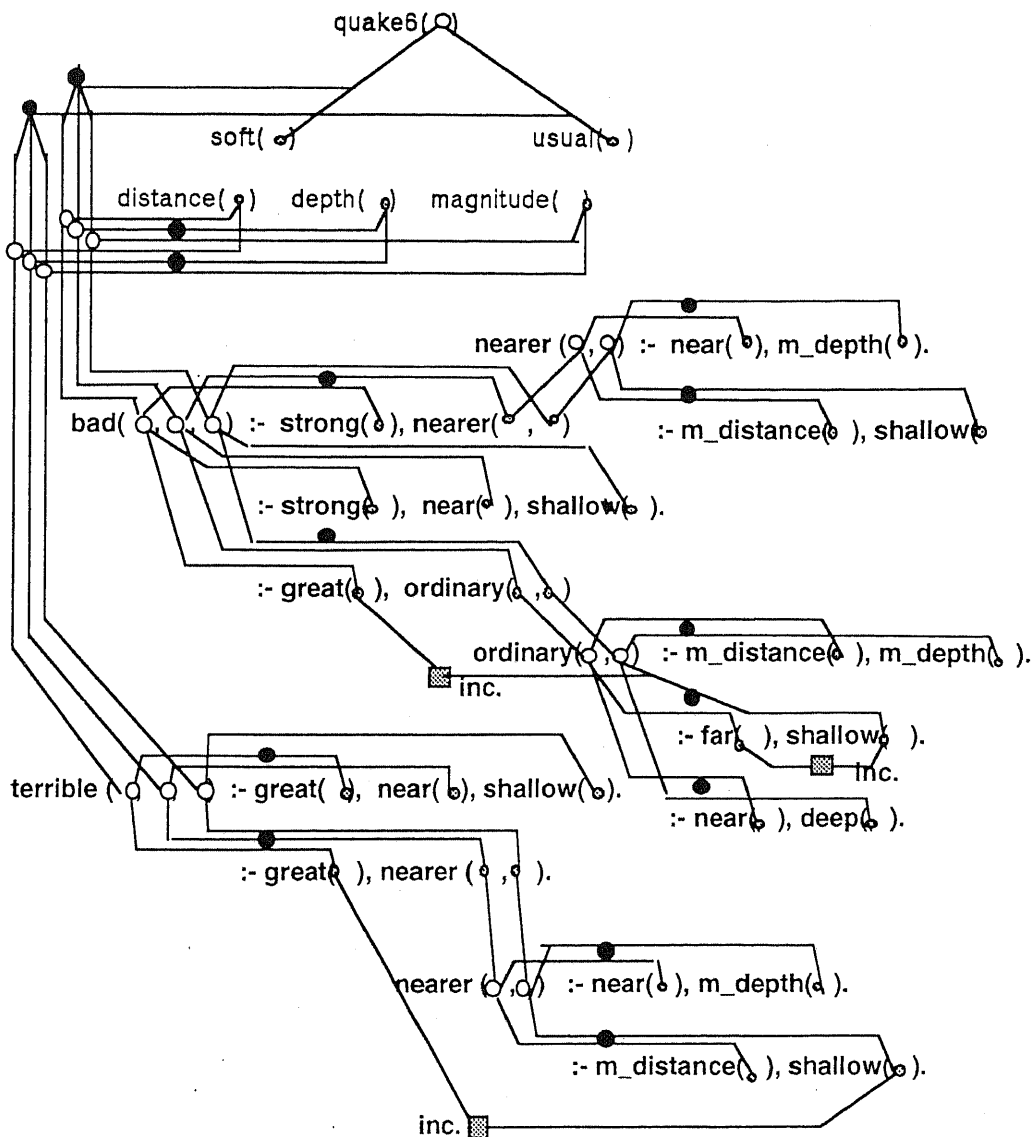


図6.12 ノード数比較の為の例3) の述語版バブル伝播ネットワーク

命題論理と述語論理とのこの差は、カテゴライズの幅を小刻みにするとさらに開く。

別の例を今一つ想定し、やはりノード数を比較してみる。直線と点からなる2つの異なるグラフの中で、一致する部分グラフを求める問題は、

NP完全な問題である。神経の様な複雑なネットワークの機能を解析する場合では閉曲線を含む部位同士を対応付けるとというのが一つの有効な方針と考えられるが、差し当たり5角形の頂点同士を比較する候補に加えない。これは次の述語形式で表現できる。ここで、 $\text{trace}(B, A)$ でAからBへ両ノードを結ぶ辺を一度だけたどることを表すことにする。即ち、ここに記した背景知識はX,Yを対応点とするための基準を定義したものである。

例4)

背景知識：  $p(X) :- \text{trace1}(X, Z_1), \text{trace2}(Z_1, Z_2), \text{trace3}(Z_2, Z_3), \text{trace4}(Z_3, Z_4),$   
 $\text{trace5}(Z_4, X).$   
 $\text{trace1}(X, Y) :- \text{trace}(X, Y).$   
 $\text{trace2}(X, Y) :- \text{trace}(X, Y).$   
 $\text{trace3}(X, Y) :- \text{trace}(X, Y).$   
 $\text{trace4}(X, Y) :- \text{trace}(X, Y).$   
 $\text{trace5}(X, Y) :- \text{trace}(X, Y).$   
 $\text{correspond}(X, Y) :- p(X), p(Y).$   
 $\text{inc} :- \text{trace1}(X, Y), \text{trace2}(Y, X).$   
 $\text{inc} :- \text{trace2}(X, Y), \text{trace3}(Y, X).$   
 $\text{inc} :- \text{trace3}(X, Y), \text{trace4}(Y, X).$   
 $\text{inc} :- \text{trace4}(X, Y), \text{trace5}(Y, X).$   
 $\text{inc} :- \text{trace4}(X, Y), \text{trace1}(Y, X).$   
 ゴール：  $\text{correspond}(X, Y).$

要素仮説集合：  $\text{trace}(1, 3), \text{trace}(1, 4), \text{trace}(2, 4), \text{trace}(2, 7)...$

もしこれを命題論理で表現するならば、まず背景知識をエルブラン展開すると次の様になる。

```

p(1) :- trace1(1, 3), trace2(3, 4), trace3(4, 2), trace4(2, 7), trace5(7, 1).,
p(1) :- trace1(1, 3), trace2(3, 4), trace3(4, 2), trace4(2, 7), trace5(7, 3).,
...
p(k) :- trace1(k, l), trace2(l, m), trace3(m, n), trace4(n, s), trace5(s, k).,
...
p(100) :- trace1(100, l), trace2(l, m), trace3(m, n), trace4(n, s), trace5(s, 100).

```

ここで、各ノードからの出次数を均一に15、対象とするグラフは100ノードからなると仮定すると5062500個のホーン節の個数と同数の中間ノードがゴールのX、Yのそれぞれについて存在することになり、これだけで $10^7$ ノードを越える。（Yについては別のグラフの頂点なので別の名前を付け、101から200までとする。例えば、 $p(101) :- \text{trace1}(101, 103), \text{trace2}(103, \dots)$ などとなる。）さらに、 $\text{trace1}(1, 3)$ と $\text{trace2}(3, 1)$ などのペア全てに対し矛盾制約が存在するので、それに対応するノードが加えられる。

一方、これを述語版ネットワーク化バブル伝播法であらわすと図6.13の様な形となる。問題の構造を直接表した構造となっていることがわかる。その構造ノード数は、ビットベクトル1桁について $(15 \cdot 100 \cdot 5 + 6) \cdot 2$ 個となり、7桁（変数が100値であるので）では $10^5$ ノードと、命題論理より桁違いにコンパクトな表現となる。しかも、制約を表すスラックノードの個数は、命題論理の場合5062500個の上記ホーン節分に加え矛盾制約が7500ノード必要であるのに対して述語の場合は全部合計しても $10^5$ ノードに収まる（矛盾ノードinc.は、両隣の変数に対するタイプ2の矛盾を表す）。ただし、この例の規模はネットワーク化バブル伝播法の近似精度に期待できる範囲を越えている。あくまでも、エルブラン空間への展開と述語表現をそのまま用いたネットワーク規模の比較の一例である。

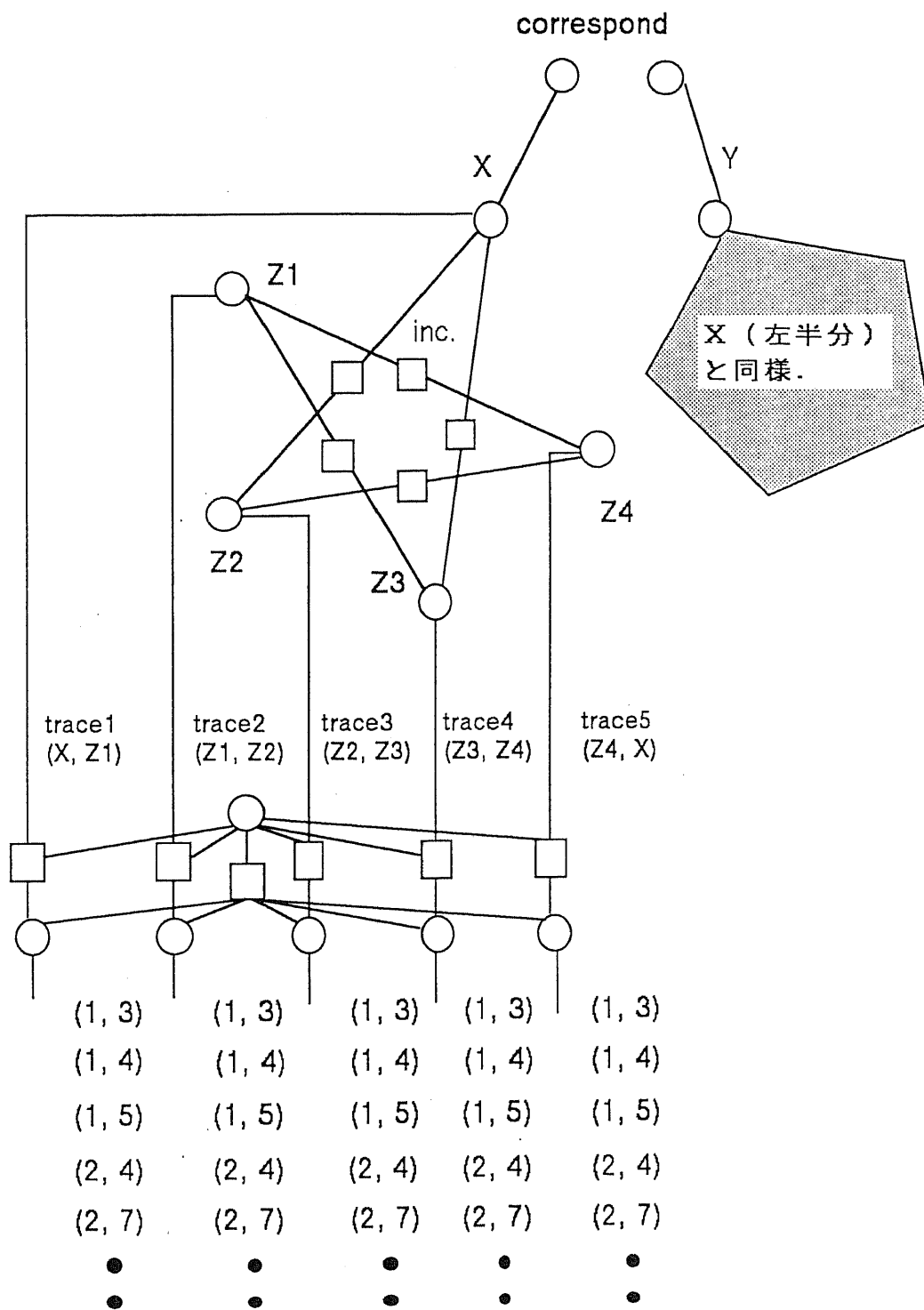


図6.13 例4) を述語論理で表現した場合のネットワーク化バブル伝播法

## エルブラン空間に展開する以外の命題論理による記述との比較

筆者は、エルブラン空間への展開以外にも命題論理に置き換える方法があることに最近になって気づいた。もし、上に示した述語版ネットワーク化バブル伝播法でのネットワーク以上に効率的な表現の手法があるとするればこれであろうと思われた。その理由は、これは述語単位でアトムとみなすのではなく、各変数への束縛状態をアトムとみなす考え方をとるからである。ここに述べる表現方法（仮にX表現と呼ぶ）とは、次の様なものである。

1. 与えられた問題をそのまま短期記憶と長期記憶を用いながら知識ネットワーク化する。
2. 変数を位置と等号条件から区別し、それぞれへの束縛状態（代入値）を命題として具体化し、これに対してネットワーク化バブル伝播法を動作させる。

例として、下の問題を解くことを考える。

例5)

背景知識	$\text{car}(X) :- \text{bridge}(X, Y), \text{car}(Y).$
候補仮説集合	$\text{bridge}(1, 2)., \text{bridge}(1, 5)., \text{bridge}(2, 3)., \text{bridge}(3, 4),$ $\text{bridge}(2, 5), \text{bridge}(5, 7), \dots$ $\text{car}(1)., \text{car}(4)., \text{car}(5).$
ゴール	$\text{car}(X) ?$

$\text{bridge}(1,2)$  で島1から島2への橋が渡されていることを表している。 $\text{car}(1)$ ,  $\text{car}(4)$ は島1と4とにレンタカー営業所があってこれらの地点に車が待機中であることを意味している。さて、どの島に（船などで）旅行すればレ

レンタカーを用いた観光ができるだろうか？ 言うまでもなく、安くて、しかも現地で借りた方がいいと決まれば早く来てもらえる方がいい。仮説bridge(1, 2)などに重みとして橋を渡る所要時間などを与え、仮説car(1)に島1の営業所で借りた場合の値段を与えればコストに基づく仮説推論でこの判断が可能となる。

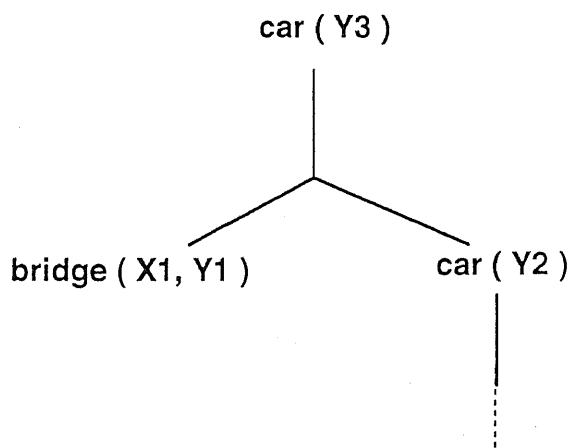


図6.14.知識ネットワークによる例5) の表現

通常の知識ネットワークでは図6.14のようになる。ここでは、手順1で図6.14の様にYを出現位置によつY1, Y2,...などと区別した変数名を与え、再帰構造の1段目だけを短期記憶とする段階について考える。即ち、図6.14の実線部である。

X表現では、この図の問題に対して図6.15の様な命題知識ネットワークを生成する。ネットワーク化バブル伝播法との共通点は元の問題を変数間の制約と見做した点である。この為、一見すると構造ノードはエルブラン空間に展開した場合より少なく見える。即ち、構造ノード数がほぼ（述語の変数の個数A）＊（変数の代入値の個数B）のオーダーで十分であ





同じ問題を述語版バブル伝播ネットワークで表すと、図6.16のようになる。やはりノード数計上のため、●で1個ずつのノードを、○と●で4ビット分重複するノードを色分けして表す。それぞれ13個と84個で合計87個となっていることが分かる。この場合も、バブル伝播ネットワークの方がX表現よりもコンパクトなネットワークを得ていることが分かる。

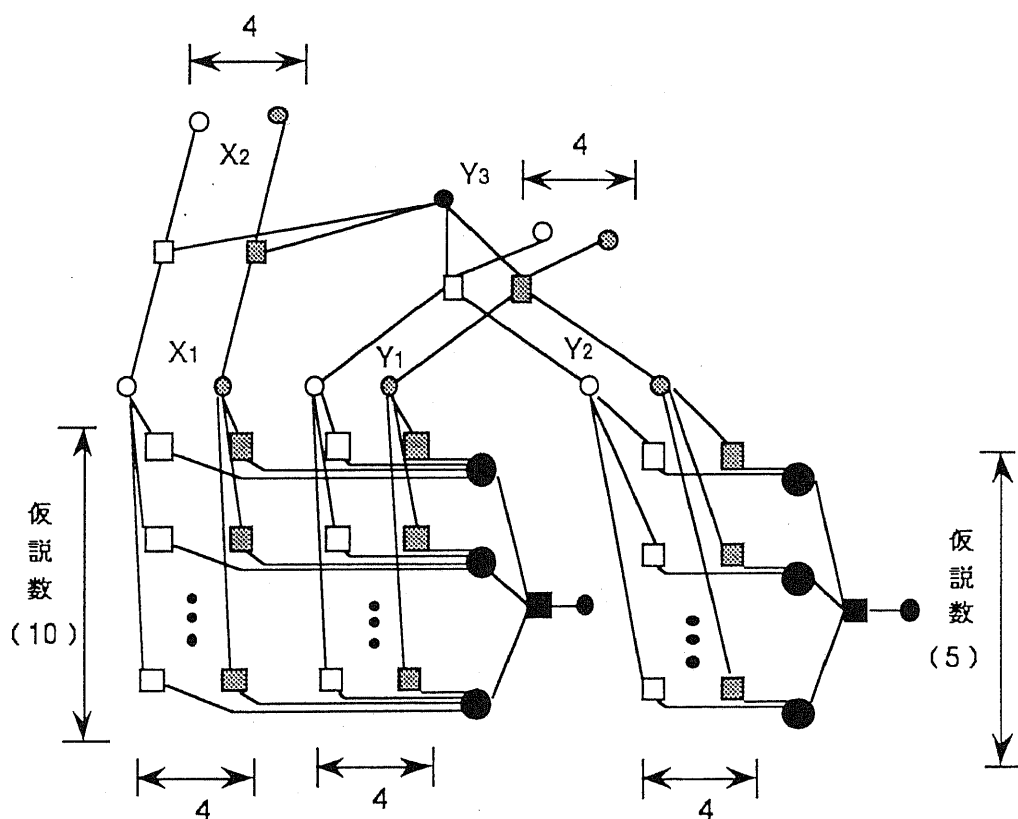


図6.16. 述語版バブル伝播ネットワークによる例5)の問題の表現

一般には、先の例で見た様に仮説からの代入値の組み合わせ数の上位アトムが必要となったり、特定の具体例だけ禁止されるなどの影響でエルブラン空間への展開もX表現も述語版バブル伝播ネットワークより低

効率となる場合が多い。後者の例として、特定の橋の通行を許可する仮説として  $p(\_, \_)$  の具体例を与えるのではなく、地点1から地点2への直通路のみ通行禁止などを意味する  $\text{bridge}(3,7)$  の禁止の制約 (6.3.6"タイプ3") を含む、次の問題を考える。

例6)

背景知識	$\text{stay}(X) \text{ :- } \text{bridge}(X, Y), \text{admit}(X, Y), \text{car}(Y).$ $\text{inc} \text{ :- } \text{bridge}(3, 7).$
候補仮説集合	$\text{bridge}(1, 2), \text{bridge}(2, 5), \text{bridge}(3, 4), \text{bridge}(4, 6),$ $\text{bridge}(5, 7), \text{bridge}(10, 3), \text{bridge}(7, 9), \text{bridge}(8, 1),$ $\text{bridge}(9, 10), \text{bridge}(6, 9).$ $\text{car}(1), \text{car}(4), \text{car}(5), \text{car}(7), \text{car}(10).$ $\text{admit}(X, Y). (X = 1, 2, 3, \dots, 10), (Y = 1, 2, 3, \dots, 10),$
ゴール	$\text{car}(X) ?$

例5のときと同様に述語版バブル伝播ネットワークを用いてこれを表すと、図6.17を得る。ここでは、構造ノード517個とスラックノード442個となっている。エルブラン展開を行った場合には、400個の  $\text{bridge}(\_, \_)$ 、400個の  $\text{admit}(\_, \_)$ 、について少なくともスラックノード400ノードを含む。結論の  $\text{stay}(\_)$  と結ぶスラックノードも含むので少なくとも1200ノードのグラフを用意しなければならない。一方、X表現を用いた場合には図6.18のようになる。

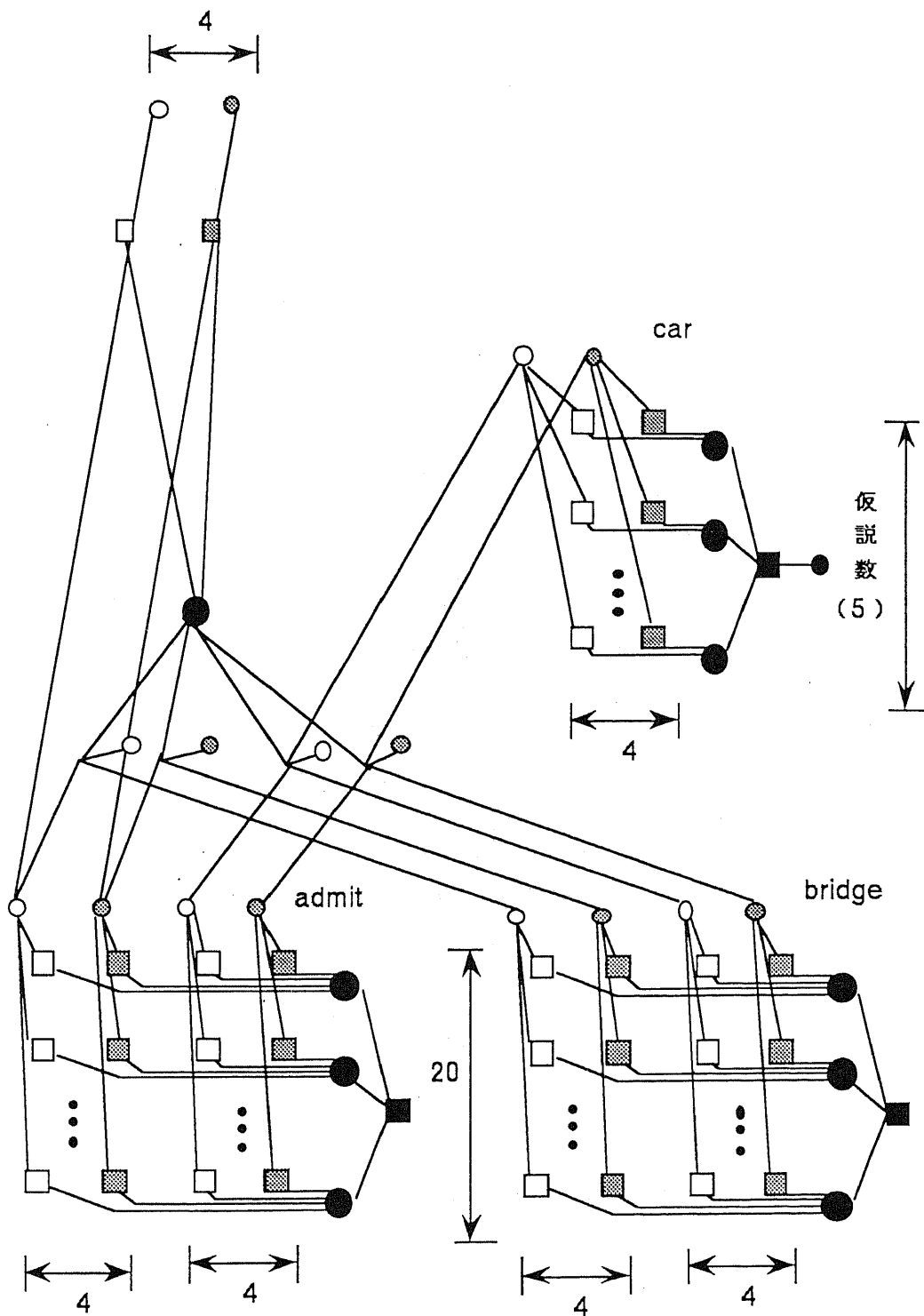


図6.17. 述語版バブル伝播ネットワークによる例6)の問題の表現

ただし図6.18では、ホーン節の位置によって変数を区別し、  
`stay(X1) :- bridge(X2, Y1), admit(X3, Y2), car(Y3).`の様に扱っている。

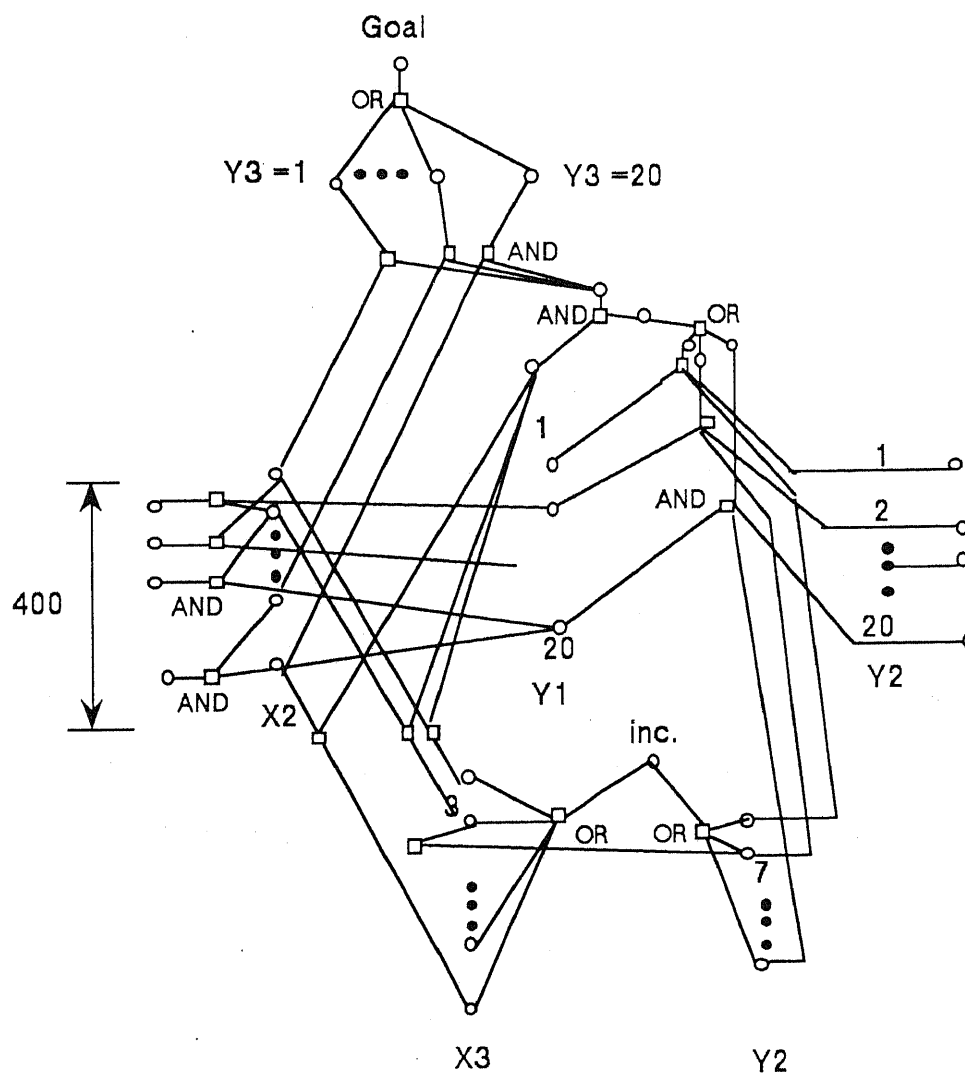


図6.18. X表現による例6)の問題の表現

この図6.18のノード数は構造ノードが944個、スラックノードが864個と述  
 語版NBP法よりも大きくなっている。即ち、一般の問題で変数の個数と  
 各変数への代入値の場合の数が多いような場合には、

述語版バブル伝播ネットワークがX表現よりもコンパクトであるが、  
エルブラン空間への展開の方が更にコンパクトとなる構造

や、

述語版バブル伝播ネットワークよりもX表現がコンパクトであるが、  
エルブラン空間への展開よりは述語版バブル伝播ネットワークが  
コンパクトとなる構造

も存在し得るが、述語版バブル伝播ネットワークがコンパクトさにおいて劣る場合もノード数のオーダーが低いので大きな差とはならない。一方、述語版バブル伝播ネットワークが勝る場合にはオーダーが比較対象のネットワークより低いノード数となるために、種々の構造の混ざった一般の問題においては結局述語版バブル伝播ネットワークが最もコンパクトになる場合が多いのである。

一方、次の様な場合には命題版バブル伝播ネットワークの方が述語版よりもコンパクトなバブル伝播ネットワークとなる場合が多くなる。

- 1) ルールあたりの変数の個数又は各変数の具体例の場合の数が少ない場合。
- 2) 知識の関連性にばらつきが大きく、述語論理で表現することができない、又は述語論理表現のメリットがない場合。

1) の例を挙げると、各変数の取り得る値の場合の数を10から3に減らせば、例3の問題は述語版バブル伝播ネットワークでは265ノード、エルブラン展開では243ノードとエルブラン展開の方がコンパクトなる。同じ条件下で例4の問題では、それぞれ6024ノード、40500ノードとなり、両者の差が1桁に縮まる。又、例5と6では、(述語版バブル伝播ネットワーク、X表現、エルブラン展開)でそれぞれ(55, 40, 38)及び(49, 36, 56)と殆ど

差がなくなることが分かる。図で示すと、述語論理と命題論理のNBP法の使い分けは図6.19の境界線で行えばよい。この図で、各ドットは、ある問題が上記の3通りの方法のうちどれを用いると最もコンパクトに表現されるかを示している。×を示したデータは、解が得られなかった場合である。

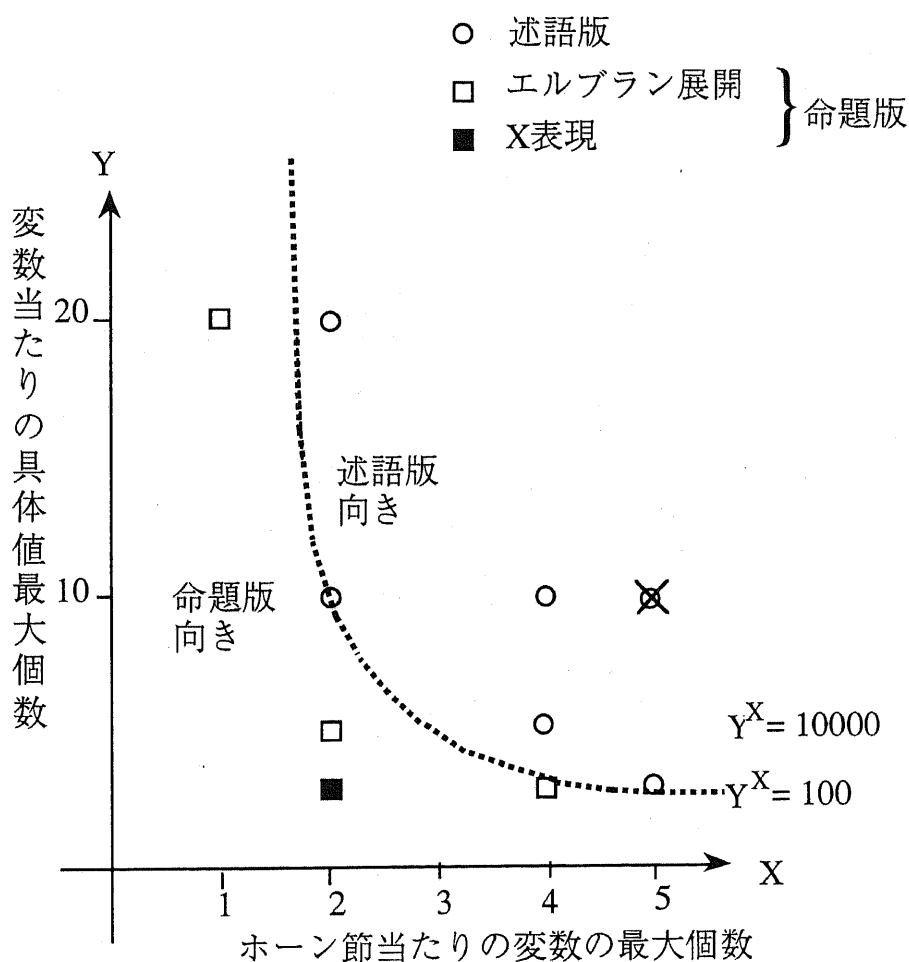


図6.19. 命題論理向きと述語論理向きの境界

又、2) は図6.20の様な問題を指している。

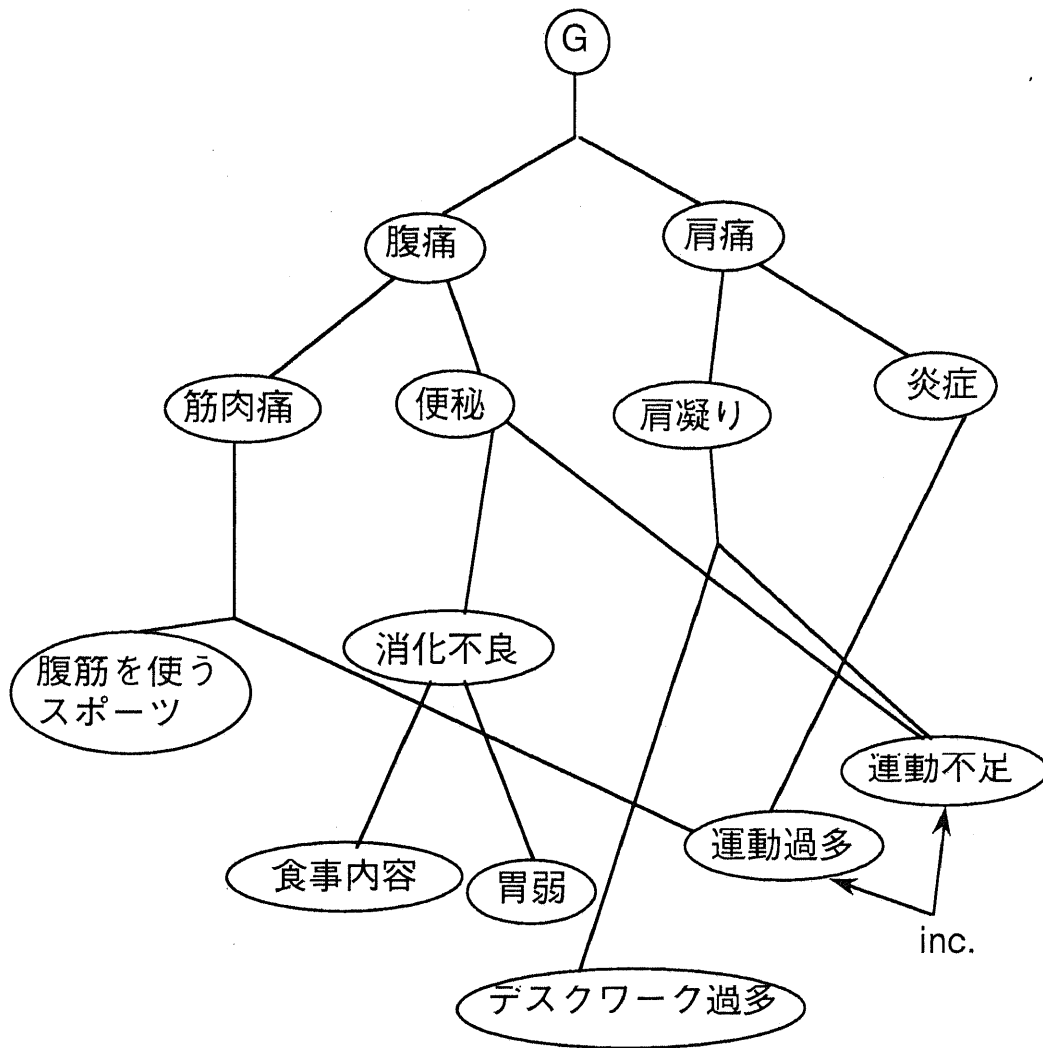


図6.20.述語論理を用いる意味のない知識ベース

この例は，アトムやルールの上に規則性がなく，述語によって一般化できない．従って，この場合は命題論理を用いる．

以上の様に，述語版バブル伝播ネットワークと命題論理を用いた表現の効率の間の一般的な比較によって，それぞれを臨機応変に用いる為の使い分けの基準を知ることができた．このことから，本論文で命題論



理と述語論理の双方の高速推論手法を編み出したことの意義がある。即ち、命題版バブル伝播ネットワークによって記述すべき、あるいはその方が効率的な場合と、述語版バブル伝播ネットワークの方が効率的な場合があり、ここに示した基準に応じてユーザは表現と用いるべき推論機構を選択することができる。この判断は、ここに示した例からも極めて日常的な感覚と一致しているのであるから、予めノード数を計算するのではなく、上に示した基準から定性的にカテゴライズすることで選択するのが容易である。

## 本章の付録

### 式(2)で代入スイッチを表せる理由

式(2)を満たす0-1整数解において,  $TP_1, TP_2, \dots, TP_n$ が全て等しい場合  $S$ の値は0又は1となり, それ以外の場合  $S$ は0となることを示す. 以下,  $n$ は2以上の整数とする.

$TP_1 = TP_2 = \dots = TP_n = 0$ の場合  $R^P=0$ の時, 式(2)から  $1 \leq S \leq 2$ となるので0-1整数解では  $S=1$ となる. 又,  $R^P=1$ の時,  $1/(n+1)-1 \leq S \leq 1/(n+1)$ となり, 0-1整数解では  $S=0$ となる.

$TP_1 = TP_2 = \dots = TP_n = 1$ の場合  $R^P=0$ とすれば  $S>1$ となってしまうので, 式(2)を満たすいかなる0-1整数解においても  $R^P=0$ となることはない.

$R^P=1$ の時, 式(2)は  $0 \leq S \leq 1$ となる.

$TP_1 = TP_2 = \dots = TP_n$ でない場合  $R^P=0$ の時, 式(2)の  $1 \leq S + \dots \leq 2$ の...の部分に負の値となり,  $S>1$ で解なし.

$R^P=1$ の時,  $U = \left( \sum_{i=1}^n TP_i \right) / (n+1)$ とおくと,  $1/(n+1)-1+U \leq S \leq 1/(n+1)+U$ となるが, 今の場合  $1/(n+1) \leq U \leq 1-2/(n+1)$ より0-1整数解では  $S=0$ .

以上より, 式(2)は先の条件を満たす.

(証明終了)

## 第7章.

### 結論

本論文では、0-1整数計画法の効率的な近似解法である掃き出し補数(PC)法の考え方を知識ネットワーク上での探索操作で実現することにより、多項式時間仮説推論を達成するネットワーク化バブル伝播(NBP)法を考案、開発した。この新手法はPC法の特徴に加えて、ネットワーク化された知識の構造的情報も有効に利用することにより、コストに基づく仮説推論の準最適解を $N^2$ オーダーの計算時間で求めることができる。

本論文に述べた様な数理計画法の知識ネットワーク上での実現によって、知識処理の視点から高速探索技術を理解することも可能となり、その理解の中から将来に向けて、知識処理の分野で発展するアイデアも生まれよう。本論文に示した中では、次の3つのアイデアがこれを物語っている。

その一つは、3.4.2に示した仮説推論の知識ネットワーク構造を利用して掃き出し補数法自体の使用手順を改良した点である。これらは、計算量のオーダーを下げた主たる要因ではないが、解の近似精度、推論速度の向上に寄与している。

今一つは、第4章の理論解析であった。これによって新たな計算複雑性のクラス分けを提案し、それをヒントに更なる高速解法を開発できた。

最後の一つは、第5章の述語論理への拡張である。従来の推論機構の研究者の間には、命題論理でなければ数理計画法を使えない、というのが常識として定着していた。エルブラン空間に落としさえすれば勿論可能だが、それではそもそも述語論理表現の意味が半減する。これに対し、第6

章の出発点はホーン節を変数間の制約と見なすところにある。この発想は、論理の意味を情報の流れに直交する制約と見なす橋田浩一の制約的意味論[Hasida93,橋田94]にヒントを得たものである。

これにより、ロジック・プログラミングを一般化した制約ロジック・プログラミングとの関連も考えることができる。ColmerauerらがPrologからProlog2, Prolog3[Col 84, 87]へと進化させて行った背景には、数値的な制約を含めて表わすことができその用途が広い（[Hug 88]では、金融の世界でも有効性が示された）という他に、記述の構文的でなく意味的な取り扱いを可能とする目的があった。例えば、[横井89]の例を挙げれば、 $3+4$ は構文的には $3+X$ ,  $X+4$ ,  $X+Y$ などの表現とのみユニフィケーションが可能であるのに対し、意味的には $7$ ,  $4+Y$ ともユニフィケーション可能でなければならない。即ち、記述の順序や形式を捨象した制約というものによって、初めて取り扱うことのできる意味というものが存在するのである。これは、橋田が強調している、外界との情報の流れと直交という意味での制約に包含されている[Has 93]。

筆者の扱った “制約として見た述語論理表現” は、表現の直接的な意味であり、エルブラン空間に展開した個々の例について一つ一つ関係を記述したものではない。このことは、制約＝意味という見方が背景にあることを示しており、上記の様な経緯が反映されたものとなっている。

ただし、橋田らが確率論的な定式化の可能な力学モデルを用いたのに対して、本研究では実数真理値の伝播で推論を実現した点が大きな差異である。確率的な推論から得られる結果は確率的である場合が多く、この研究の様に最終的に明確な離散的判断の要請される問題を扱うのには適していない。この点で、確率と実数真理値という二つの指標が全く別の意味を有することには十分注意が必要である。

表現の意味をそのまま制約不等式に置き換えたことで、当然その後の研究にもこれまでには全くなかった指針が可能になった。

その一つは、NBP法による高速解法をエルブラン空間よりもオーダー的に狭い探索空間で行うことができるようになったことである。Vieilleらの研究では多くのサブゴールを生成するので、この壁は越えていない。そして同時に、彼らのアプローチを長期記憶、短期記憶というこの分野においては独自の概念を通して取り入れることに成功した。

それに加えて推論機構（エンジン部）そのものが2章で述べた多項式時間の推論手法であったために、三重のメリットが活かされる。即ち、

1. 非エルブラン展開の方針による、扱うネットワークのサイズの節約
2. 長期記憶、短期記憶の併用によるゴール指向性  
ゴールを起点としたトップダウンな段階的拡大であると同時に、ゴールの変数値と可能な仮説の変数値の整合性を手掛かりにネットワーク拡大の規模を制御するボトムアップな要素も備えている。
3. ネットワークの大きさに対する多項式時間の推論時間である。

言い換えると、数理計画問題の近似解法をネットワーク化することから出発して、従来の枠組みを越えたところでの高速な準最適計算の推論手法を見い出せたことになる。これは、近似解法の有効なヒューリスティックスの発見において、数学的枠組み上での操作を考えるだけでなく、知識の空間構造的側面をも考慮した知識処理的あるいは情報処理的な操作を導入した成果であると言えよう。

## 謝辞

本研究について、多くの方々にお礼を申し述べる必要があります。

先ず本研究を進めるスタートにおいて、修士課程とは異質の研究にとりかかろうとした筆者を快く迎え入れてくださいました石塚満教授への超指数オーダーの感謝は表現する言葉も見当たりません。そしてその後も、研究室に入って一月目に早くも「数理計画法のネットワーク化」という、大きなヒントを与えてくださいました。この10字足らずのコメントが本研究の心臓となり、以来、先生のご指導を道しるべとしたことへの感謝は、第1章から7章までと同量の紙面でも言い尽くせません。

又、修士課程で研究生生活の細部までご指導下さった上に、分野の異なる人工知能の研究への旅立ちの切符を与えてくださいました藤井陽一教授には、その後の様々な形でのご指導を併せて感謝致します。

又、本研究の根本にあるモチベーションである推論の高速化は、5年前に藤崎博也教授（当時）の下での卒業研究から継続したものです。学部時代に学問そのものに嫌気がさし退屈な日々を送っていた私が研究への憧憬と情熱に駆り立てられたのは、「知識とは何か」と藤崎先生が講義中に問いを発せられたまさにその瞬間からでした。私に人生の突破口を下された藤崎先生に心から感謝致します。

又、研究を進める上での議論および有効なアドバイスを頂きました、東京大学工学部の田中英彦教授、淵一博教授、相田仁先生、横山先生そして学術情報センターの安達先生のコメントと本学医学部の高橋國太郎教授のご意見は、その後の研究をまとめる上で非常に参考になりましたことを記し、感謝致します。さらに投稿論文や学会での発表を基に数々のコメントを日々頂きました大阪大学の谷内田正彦教授、八木康史先生、東京理科大学の溝口文雄先生、筑波大学の寺野先生、電子技術総合研究

所の橋田浩一氏と富士通研究所の山本栄一郎氏に感謝申し上げ、今後とも変わらぬご指導をお願い申し上げます。

先生方以外にも、頻繁な議論に参加して下さいました先輩及び友人の皆様に感謝致します。特に、NTTの阿部明典氏、牧野俊朗氏、電子技術総合研究所の長谷川修氏、富士通研究所の近藤朗子氏、および本学大学院のBuss Martin氏、杉本正則氏、藤井俊彰氏、柏野邦夫氏、片山紀生氏、酒井美和女史、日本工業新聞社の原田成樹氏、住友信託銀行の西村太植氏そしてマッキンゼー・アンド・カンパニーの堂前宣夫君に感謝致します。

又、本研究の一部の共同研究者である高間康史君とのディスカッションが実り多いものであったことを記し、同君に感謝します。研究室では又、土肥浩先生の計算機に関するご知見からの適切なアドバイスや、徐行俊氏をはじめとする皆様の常日頃の励ましに精神的な面を支えられ、呉楽水君から示された文献がその後の研究に生かされたこと、五味・エジソン・サトシ氏、二田丈之君とのディスカッションでは自分の考えを一部整理できました。石塚研究室では、日々の生活の細部に至る快適な環境によって研究生活全般が支えられました。私の場合、議論で熱くなると周りの方々を困らせてしまうことがあり、あるいは不愉快な思いをさせてしまうことが多かったのではないかと、反省致しております。このような自分を排することもなく、お付き合い下さいましたことを研究室の皆様に感謝致しております。

経済面では、日本学術振興会から頂きました生活および研究の助成金は大きなよりどころとなりました。又、本研究が文部省科研費一般研究B(No.04452190)、同重点研究領域「知識科学」及び特別奨励金(No.06004269)の援助によったことも合わせて記し感謝します。

そして、この歳まで働きもせず学問一筋に生きる機会を与えてくれた両親に感謝し、その仕上げとしてのこの論文を捧げます。

最後に妻 朝子に一言。愛を、有難う。

## ◇ 参考文献 ◇

- [阿部92] 阿部明典, 石塚満: 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, 人工知能学会誌, 7, pp.77-86 (1992)
- [Ada 88] Adams, J., Balas, E., et.al: The Shifting Bottleneck Procedure for Job  
Shop Scheduling, Management Science, 34, pp.391-401 (1988)
- [Aho 79] Aho, A., Ullman, J.D.: Universality of Data Retrieval Languages,  
Proc. 6th ACM Symp. on Principles of Programming Languages, pp.110-  
120 (1979)
- [Arn 87] Arnborg, S., Corneil, D.G., Proskurowski, A., Complexity of finding  
embeddings in a k-tree, SIAM J. Algebraic Discrete Methods Vol.8.,  
pp.177-184 (1987)
- [Bal 80] Balas, E. & Martin, C. : Pivot and Compliment -- A Heuristic for 0-1  
Programming, Management Science, 26, pp.86-96 (1980)
- [Ban 86] Bancilhon, F., Maier, D., Sagiv, Y. and Ullman, J.D.: Magic Sets and  
Other Strange Ways to Implement Logic Programs, Proc. ACM PODS,  
pp.1-15 (1986)
- [Bra 87] Brayton, R.K., Rudell, R., Sangiovanni-Vincentelli, A.L. and Wang,  
A.R. : MIS: A Multiple-Level Logic Optimization System, IEEE Trans.  
CAD, CAD-6, pp.1062-1081 (1987)
- [Byl 91] Bylander, T. , Allemang, D. , et al. : The Computational Complexity of  
Abduction, Artif. Intell., 49, pp.25-60(1991)
- [Cha 90] Charniak, E. Shimony, S.E. Probabilistic Semantics for Cost Based  
Abduction, Proc. AAAI-90, pp106-111(1990)
- [Col 84] Colmerauer, A.: Equations and Inequations on Finite and Infinite Trees,



- Proc. of FGCS '84 (1984)
- [Col 87] Colmerauer, A.: Opening the Prolog 3 Universe, BYTE, August 1987, pp.177-182 (1987)
- [Dan 63] Dantzig, G.B., Linear Programming and Extension, Princeton University Press (1963)
- [Dec 87] Dechter, R. & Pearl, J. :Network-based Heuristics for Constraint Satisfaction Problem, Artif. Intell., Vol.34, pp.1-38 (1987)
- [Dec 88] Dechter, R. & Pearl, J. : Tree-clustering schemes for constraint-processing, Proc. of AAAI'88, pp.150-154 (1988)
- [Dec 91] Dechter, R: Temporal constraint Network: Journal of Artif. Intell.,49, pp.61-95 (1991)
- [deK 86] De Kleer, J: An Assumption-based TMS, Artificial intelligence, 28, pp.127-162 (1986)
- [deK 91] De Kleer, J: Focusing on Probable Diagnosis, AAAI-91, pp.842-848 (1991)
- [Dha 90] Dhar, V And Ranganathan, N.: Integer Programming vs. Expert Systems: An Experimental Comparison, Com. ACM, 33 (1990)
- [堂前94] 堂前宣夫, 石塚満: 仮説推論高速化のための知識ベースリフォメーション人工知能学会誌, 9, pp.595-603
- [Dow 84] Dowling, W.F. and Galliar, J.H.: Linear Time Algorithm for Testing the Satisfiability of Propositional Horn Formulae, J. Logic Programming, 3, pp.267-284 (1984)
- [Fre 82]Freuder,E.C.: "A sufficient condition of backtrack-free search," Journal of the ACM, 29, pp.24-32 (1982)
- [藤田91a] 藤田昌宏: 多段論理回路合成技術の動向, 信学会論文誌A, J74-A, pp.152-161(1991)
- [藤田91b] 松永裕介, 藤田昌宏: 順序付き 2 分決定グラフと許容関数を用い

- た多段論理回路簡単化手法, 信学会論文誌A, J74-A, pp.196-205  
(1991)
- [二田95] 二田丈之, 大澤幸生, 石塚満: 線形・非線形計画法の併用による高速仮説推論, 情処春全大'95 (予定)
- [Gef 87] Geffner, H. & Pearl, J.: An Improved Constraint-Propagation Algorithm for Diagnosis, 10th-IJCAI, Milan, pp.1105-1111 (1987)
- [Glo 86a] Glover, F: Tabu Search, Part1, ORSA J. Computing,1, pp.190-206
- [Glo 86b] Glover, F: Tabu Search, Part2, ORSA J. Computing,2, pp.4-32
- [Gol 89] Goldberg, D.E., Genetic Algorithms in search, Optimization & Machine Learning, Addison-Wesley (1989)
- [Gu94] Gu, J.: Local Search for Satisfiability (SAT) Problem, IEEE Trans. on Systems, Man, and Cybernetics, Vol.SMC-23, pp.1108-1129 (1993)
- [Has 93] Hasida, K.: Dynamics of Symbol Systems - An Integral Architecture of Cognition -, Proc. Int'l Conf. on FGCS, 2, pp.1141-1148 (1992)
- [橋田94] 橋田浩一, 宮田高志, 長尾確: 確率的制約プログラミング, 情処人工知能研究会95-6, 予稿集pp.49-58 (1994)
- [Hoo 88] Hooker, J.N.: A Quantative Approach to Logical Inference, Decision Support Systems, 4, No.1, pp.45-69, 1988.
- [Hug 88] Hugh, T., Lasses, C.: A CLP(R) Options Trading Analysis System, Proc., of 5th ICLP, pp.59-69, the MIT Press (1988)
- [茨木93] 茨木俊秀, 福島雅夫: 最適化の手法, 共立出版社 情報数学講座 14, pp.5-16(1993)
- [井上93] 井上克巳, 大田好彦, 長谷川隆三, 中島誠: モデル生成に基づく並列アプリケーション人工知能学会誌, 8, pp.786-796(1991)
- [Ioannidis 85] Ioannidis, Y.E.: A Time Bound on the Materialization of Some Recursively Defined Views, Proc. VLDB'85, pp.219-225 (1985)
- [Iri 81] Iri, M.: Use of Matroid Theory in Operations Research, Circuits and

- Systems Theory, International Journal of Systems Science 12, pp.27-54  
(1981)
- [石塚91] 石塚満: 次世代エキスパートシステム, エクスパート システム  
(石塚, 小林 (編) ) 第7章, 丸善 (1991)
- [石塚92] 石塚満: 仮説推論, 日本ファジィ学会誌, 4, pp.620-630(1992)
- [石塚94] 石塚満: 仮説推論の計算量と高速メカニズム, 人工知能学会誌,  
Vol.9, pp.342-349 (1994)
- [伊藤91] 伊藤史朗, 石塚満: 推論パスネットワークによる高速 仮説推論  
システム, 人工知能学会誌, 6, pp.501-509(1991)
- [Kam 92] Kamath, A.P., Karmarkar, N.K., et.al: A continuous approach to  
inductive inference, Mathematical Programming, 57, pp.215-238(1992)
- [Kar 82] Karmarkar, N.: A Polynomial-time algorithm for linear programming,  
Combinatorica, 4, (1984) pp.373-pp.395
- [Kau 90] Kautz, H.A and Selman, B.: Hard Problems for Simple Default Logics,  
Proc. 1st Int'l Conf. on Principles of Knowledge Representation and  
Reasoning (KR'89), pp.189-197, 1989.
- [Kif 86] Kifer, M. and Lozinskii, E. Filtering Data Flow in Deductive Databases,  
Proc. International Conference on Database Theory, Lecture Notes in  
Computer Science, No.243, Springer-Verlag, pp.186-202.
- [Kir 83] Kirkpatrick, S, et.al.: Optimization by Simulated Annealing, Science,  
220, pp.671-680 (1983)
- [近藤93] 近藤朗子, 牧野俊郎, 石塚満: 述語論理知識を扱う高速 仮説推論  
システム, 人工知能学会誌, 8, pp.819-827(1993)
- [Kor 85] Korf, R.E. : Depth-first iterative deepening: An optimal admissible tree  
search, Artificial Intelligence, Vol.27, pp.97-109 (1985)
- [Kow 71] Kowalsky, R: Linear Resolution with Selection Function, Artificial  
Intelligence, Vol.2, pp.227-260 (1971)

- [Kow 88] Kowalsky, R: The Early Years of Logic Programming, CACM, Vol.31, pp.38-43 (1988)
- [Lev 89] Levesque, H.J.: A Knowledge-level Account of Abduction (preliminary version), Proc. IJCAI-89, pp.1061-1067 (1989)
- [牧野90] 牧野俊朗, 石塚満: 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, 5, pp.640-648(1990)
- [牧野93] 牧野俊朗, 石塚満: 経験に基づく学習による仮説推論の高速化, 人工知能学会誌, 8, pp.320-327(1993)
- [McD 87] McDermott, D.: A Critique of Pure Reason, Computational Intelligence, 3, pp.151-160 (1987)
- [Min 83] Minker, J., Nicholas, J.M.: On Recursive Axioms in Deductive Database, Information Systems, 1, (1983)
- [Min 92] Minton, S.: Minimizing Conflicts: a Heuristics Repair Method for Constraint Satisfaction and Scheduling Problems, Artif. Intell. 58, pp.161-205, (1992)
- [Moo 93] Mooney, R.J., Ourstin, D.: A Multistrategy Approach to Theory Refinement, pp.141-164, Machine Learning, 4, (Michalski, R.S. and Tecuci, G., eds.), Morgan Kaufmann (1994)
- [野中94] 野中久典, 高元政典, 小林康弘: 0-1計画問題の近似解法プログラムの開発と評価, 情報処理学会論文集 41-94, pp.93, 94
- [Ohs 92] Yukio Ohsawa and Yoichi Fujii: Stationary solution of nonlinear propagation of femtosecond optical pulse by circuit model, J. of Physical Society of Japan, 61, p.3977-3983 (1992)
- [Ohs 93] Ohsawa, Y., Ishizuka, M. : Networked Bubble Propagation Method as a Polynomial-time Hypothetical Reasoning for Computing Quasi-optimal solution, Proc. of IEEE-Int'l Conf. of TAI'93, pp.184-187 (1993)
- [大沢94a] 大沢幸生, 石塚 満: 仮説推論における準最適解を多項式時間で計

- 算するネットワーク化バブル伝播法, 信学論文誌V77-D2, pp.1817-1829 (1994)
- [大沢94b] 大沢幸生, 石塚 満: 多項式時間仮説推論を達成するネットワーク化バブル伝播法の述語論理への拡張, 人工知能学会誌, (1995.9)
- [大沢94c] 大沢幸生, 石塚 満: 動的に拡大するバブル伝播ネットワークによる多項式時間の述語論理仮説推論, 人工知能学会全大'94, 3-2 (1994)
- [大沢94e] 大沢幸生, 石塚 満: 制約不等式の変形に基づくネットワーク化バブル伝播アルゴリズムによる仮説推論の高速化, 情処人工知能研究会資料 93AI-90-9 (1993)
- [大沢95] 大沢幸生, 石塚 満: 改良型ネットワーク化バブル伝播法による低次多項式時間仮説推論, 人工知能学会誌, (1995.1)
- [岡本93] 岡本知樹, 石塚満: 整数計画法の近似解法を適用した準最適解計算の高速仮説推論法, 人工知能学会誌8, pp.222-229, (1993)
- [Pie 32] Elements of Logic, Hartshorne, C. and Weiss, P.eds. Collected Papers of Charles Sanders Peirce, 2, Harverd Univ. Press, MA (1932)
- [Poo 87] Poole, D. ,Aleliunas, R. & Goebel, R. : Theorist : A Logiocal Reasoning System for Defaults and Diagnosis, in the Knowledge Frontier : Essays in the Knowledge Representation (Cercone, N. J. and Macalla, G., eds.) Springer-Verlag, N.Y. (1987).
- [Pol 88] A Logical Framework for Default Reasoning, Artif. Intell., 36, pp.27-47 (1988)
- [Pol 92] Pool, D.: Logic Programming, Aduction and Probability, Proc.Int'l Conf. on Fifth Generation Computer Systems (FGCS'92), pp.530-538 (1992)
- [Pol93] Poole, D : Probabilistic Horn abduction and Bayesian networks, Artif. Intell., 64, pp.81-129 (1993)
- [Rei 80] Reiter, R: A Logic for Default Reasoning, Artif. Intell., 13, pp.81-132 (1980)

- [Rob 63] Robinson, J.A.: Theorem - Proving on the computer, J. of ACM, 10, pp.163-174 (1963)
- [San 93] Santos, E, Jr. : A Fast Hill-Climbing Approach Without an Energy Function for Probabilistic Reasoning, Proc. of IEEE-Int'l Conf.of TAI'93, pp.170-179 (1993)
- [Sel 90] Selman, B., Leveque, H.J., Abductive and Default Reasoning: A Computational Core, Proc. AAAI-90, pp.343-348 (1990)
- [Sel 92] Selman, B., Levesque, H.J., Mitchell, D.: A New Method for Solving Hard Satisfiability Problems, Proc. AAAI'92, pp.440-446
- [Sel 93a] Selman, B. and Kautz, H., " An Empirical Study of Greedy Search for Satisfiability Testing," Proc. AAAI-93, pp.46-51, 1993.
- [Sel 93b] Selman, B. and Kautz, H., " Domain-Independent Extensions to GSAT : Solving Large Structured Satisfiability Problems," Proc.IJCAI-93, pp.290-295, 1993.
- [Sti 90] Stillman, J: It's not my Default: The Complexity of Membership Problems in Restricted Propositional Default Logics, Proc. AAAI '90, pp.571-578 (1990)
- [高間94] 高間康史, 大沢幸生, 石塚満: 論理の多段化による知識リフォーメーションに基づく仮説推論の高速化, 人工知能学会全大'94, 3-2 (1994)
- [鶴田91] 鶴田三郎, 石塚満: 発想的知識生成のための命題論理知識ベースのコンパイル法, 人工知能学会誌, 6, pp.117-123 (1991)
- [鶴田92] 鶴田三郎, 石塚満: 発想的知識生成のため述語論理知識ベースのコンパイル法, 人工知能学会誌, 7, pp.130-137 (1992)
- [富田88] 新道美喜男, 富田悦次: 最大クリークを抽出する単純なアルゴリズムとその最大時間計算量, 信学会論文誌, J71-D, pp.472-481 (1988)
- [Vielle 87a] Vielle, L: Recursive Axioms in Deductive Databases: The Query/Subquery approach, Proc. Expert Database Systems, pp.253-267(1987)

- [Vielle 87b] Vielle, L: A Database-complete Proof Procedure Based on SLD Resolution, Proc. 4th ICLP, pp.74-103(1987)
- [Vielle 88] Vielle, L: From QSQ towards Q0SnQ: Global Optimization of Recursive Queries, Proc. Expert Database Systems, pp.421-435(1988)
- [横井89] 横井俊夫, 相場亮: 制約ロジック・プログラミング知識処理への新しいパラダイム, 情報処理, Vol.30, pp.29-38 (1989)

## 業績リスト

### 論文誌

- [1-1] Stationary solution of nonlinear propagation of femtosecond optical pulse by circuit model, Yukio Ohsawa and Yoichi Fujii, J. of Physical Society of Japan, **61**, p.3977-3983 (1992)
  
- [1-2] 仮説推論における準最適解を多項式時間で計算するネットワーク化バブル伝播法, 大澤幸生, 石塚満; 信学会D2論文誌, J76-D-2, pp.1817-1829 (1994)
  
- [1-3] 改良型ネットワーク化バブル伝播法による低次多項式時間仮説推論, 大澤幸生, 石塚満; 人工知能学会誌, 10, No.1, pp.123-130 (1995)
  
- [1-4] 多項式時間仮説推論を達成するネットワーク化バブル伝播法の述語論理への拡張, 大澤幸生, 石塚満; 人工知能学会誌 Sept. (1995) 掲載決定
  
- [1-5] Networked Bubble Propagation method as a Polynomial-time Hypothetical Reasoning for Computing Near-Optimal Solution, 採録決定  
Yukio Ohsawa, Mitsuru Ishizuka; Systems and Computers in Japan.
  
- [1-6] 知識の実行時リフォーメーションに基づく仮説推論の高速化手法 (投稿中), 高間康史, 大澤幸生, 石塚満; 人工知能学会誌
  
- [1-7] Networked Bubble Propagation method as a Polynomial-time Hypothetical Reasoning for Computing Near-Optimal Solution (投稿中), Yukio Ohsawa, Mitsuru Ishizuka; IEEE, Trans. on Knowledge and Data Engineering.

### 海外発表 (査読付き)

- [2-1] Networked Bubble Propagation method as a Polynomial-time Hypothetical Reasoning for Computing Quasi-optimal Solution,



Yukio Ohsawa, Mitsuru Ishizuka; 採択率=65 / 165

Proc. of IEEE, Int'l conf. on Tools with Artificial Intelligence(TAI '93) p.184-187

- [2-2] A Polynomial-time Predicate-logic Hypothetical Reasoning by Networked Bubble Propagation Method (投稿中),  
Yukio Ohsawa, Mitsuru Ishizuka; IJCAI '95

## 研究会

- [3-1] フェムト秒光パルス伝搬の一般的解析,  
大澤幸生, 藤井陽一; '91重点領域超高速超並列光エレ研究会1, pp.63-68 (1991)
- [3-2] 非線形伝搬の基本方程式に基づくフェムト秒光パルス伝搬の伝送限界,  
大澤幸生, 藤井陽一; 信学会 OQR研究会技報 OQE91-116, pp.89-94 (1991)
- [3-3] 光パルス非線形伝搬方程式のフェムト秒領域における固有解,  
大澤幸生, 藤井陽一; 91重点領域超高速超並列光エレ研究会3, pp.37-40 (1991)
- [3-4] フェムト秒光パルス伝搬,  
大澤幸生, 藤井陽一; '92重点領域超高速超並列光エレ研究会1, pp.57-60 (1992)
- [3-5] 多項式時間仮説推論を達成するネットワーク化バブル伝播アルゴリズム,  
大澤幸生, 石塚満; 信学会人工知能研究会, 信学技法 AI92-76 pp.17-24 (1992)
- [3-6] 新しい制約不等式に基づくネットワーク化バブル伝播アルゴリズムによる  
高速仮説推論,  
大澤幸生, 石塚満; 情処人工知能研究会資料 93AI-90-9 pp.71-80 (1993)

## 全国大会 (題名, 著者, 学会, 論文番号, 頁数, 年の順)

- [4-1] 未知語を含む文の形態素解析システム,  
大澤幸生, 藤崎博也; 情処秋全大'90, 1S-5, 3-pp.107-108 (1990)
- [4-2] フェムト秒光パルス伝搬の一般的解析,  
大澤幸生, 藤井陽一; 信学秋全大'91, C-255, 4-pp.285 (1991)
- [4-3] フェムト秒領域の非線形光パルス伝搬の基本方程式の解析的固有解と  
これに対する自己誘導ラマン散乱の影響,  
大澤幸生, 藤井陽一; 信学春全大'92, C-329, 4-pp.371 (1992)
- [4-4] 多項式時間仮説推論を達成するネットワーク化バブル伝播アルゴリズム,  
大澤幸生, 石塚満; 情処秋全大'92, 5G-4, 3-pp.43-44 (1992)
- [4-5] ネットワーク化バブル伝播アルゴリズムによる多項式時間仮説推論,  
大澤幸生, 石塚満; 情処春全大'93, 2D-1, 2-pp.33-34 (1993)
- [4-6] 述語論理に適用できるネットワーク化バブル伝播アルゴリズムによる  
多項式時間仮説推論の達成,  
大澤幸生, 石塚満; 人工知能学会全大'93, 3-5, pp.83-86 (1993)
- [4-7] 制約不等式の変形に基づくネットワーク化バブル伝播アルゴリズムによる  
仮説推論の高速化,  
大澤幸生, 石塚満; 信学秋全大'93, D-128, 6-pp.130 (1993)
- [4-8] 記憶の階層性に基づくネットワーク化バブル伝播法による  
述語論理仮説推論の高速化,  
大澤幸生, 石塚満; 情処春全大'94, 5R-7, 3-pp.209-210 (1994)

[4-9] 論理の多段化による知識リフォーメーションに基づく仮説推論の高速化手法,

高間康史, 大澤幸生, 石塚満; 情処春全大'94, 5R-8, 3-pp.211-212 (1994)

[4-10] 優秀論文賞受賞

動的に拡大するバブル伝播ネットワークによる多項式時間の述語論理仮説推論,

大澤幸生, 石塚満; 人工知能学会全大'94, 3-2, pp.51-54 (1994)

[4-11] 論理の多段化による知識リフォーメーションに基づく仮説推論の高速化,

高間康史, 大澤幸生, 石塚満; 人工知能学会全大'94, 3-4, pp.59-62 (1994)

[4-12] 一般的制約を含む述語ホーン節の不等式制約による表現,

大澤幸生, 石塚満; 情処秋全大'94, 1J-2, 3-pp.3-4 (1994)

## その他

### 報道メディア

日本工業新聞 ('93.8/6 15面)

bit bitリーダー ('93.10, Vol.25 No.10, pp.82)

# 索引

## A. 用語索引 (五十音順)

I-NBP法	76	演繹法則的説明	2
EITHER	2	Eulerの定理	68
Adaptive Consistency	85	解の極小性	3
アトム	5	(デフォルト推論における) 拡張	10
アブダクション	1	改良フェーズ	30, 53
アルゴリズム	15	確率	65
Iterative Deepening	123	確率的(Stochastic)戦略	34
遺伝的アルゴリズム	34	確率的な知識	3
命網	128	仮説	2
命網延長の停止条件	129, 130	仮説集合	2
違反制約の伝播	95	仮説推論	2
違反制約	95	仮説単位	123
一般的な流れのモデル	140	仮説知識	3
EBL	21	仮説合成	21
意味体	60	仮説の選択	2
因数分解	109	仮説の提起	2
Integrate (値の合成) 部	133	可能なバブル伝播	58
Infeasibility(IF=制約違反度)	76, 145	Karmarker法	33, 74
AIとORの融合	23	完全性	34, 113
影響度	45	関数なし (Function-free)	140
X表現	158	完備性	83
エンジン部	167	観測事象	4
相対深度	45, 50, 53	機械推論	10
ATMS	21	KICK-HOPE	112
SLD導出木	7, 23	擬似焼きなまし法	34
SLD-AL反駁	116	基底アトム	137
SL導出	20	基底構造 (○) ノード	39
SL導出システム	20	基底スラック (□) ノード	39
NBP法	26	帰納推論	2
NP完全	22, 34	帰着可能	28
NP困難	22	基底変数	30
エルブラン空間	126	基本サーキット	68
演繹推論	2	QSQアプローチ	125
演繹の連鎖	81	QSQR	21, 116, 125

行商人(Travelling Salesman, TS) 問題	36	ゴール指向性	62
極小解	60	混合計画問題	54, 136
極小限定 (Circumscription)	8	再帰構造	27, 123
局所解の問題	34	再帰的な知識 (ルール)	123
局所探索法(local search)	34	cycle cutset	68
近似解法	22	再計算の回避	20
近傍探索	30	最大クリーク問題	34
区域	51	最適化	34
空節	3, 106	最適解	22, 30
クラス1	79	サイドウェイ	124
クラス2	79	Satisfaction test部	133
クラス3	72, 79	事実	2
グラフの (regularな) 幅	73, 84	地震予知委員会	10
クリティカルな障害	61	試行錯誤	61
クリティカルな制約	61	GSAT	35
クリティカルパス法	36	指数オーダ	11
計算量クラス	79	指数深さ	62, 64
形態素	18	実行可能解	53, 60
結論	5	実数最適解	39
検査点	7	支配ベクトル	41
健全性	141, 144	Shifting Bottleneck	36
検討	60	充足可能性	28, 35
厳密解	71	述語 (論理) 版NBP法	60, 117
交換相手	42, 76	述語版NBP法の表現能力	141
交換相手の選択	45	述語ホーン節	117
(○, ■) の交換	76	述語論理	27
後件	5	準最適解	11, 22, 30
後件肯定の虚偽	1	条件部	5
格子点	32	小前提	1
構造ノード	37	状態遷移	78
構造の複雑さ	51	衝突点	67
構造変数	30	使用頻度	16
公理集合	2	情報の時間的な流れ	139
コスト	3, 28	初期状態	39
コストに基づくアブダクション (仮説推論)	4, 28, 149	初期フェーズ	30
ゴール	3	Series-Parallel	84, 86, 105
ゴール指向	21	信念	8
		真理値空間	39

真理値固定	30	ターゲット	42, 45, 76
スイッチノード	120	対話型人工知能	139
推論時間	10	多項式(オーダ)時間	11, 22, 30
推論速度	7)	縦型探索	95
推論パスネットワーク	7, 21, 23	タブー探索	34
数値計画法	28	タブーリスト	34
数値計画法の(知識)ネットワーク化	20, 115	多面体	30
Stack-on	5	短期記憶(STM)	27, 116, 125
Stack-off	5	短期記憶の拡大	145
strong-d-i-consistent <sup>84</sup>	84	単結合	71
スラック変数	30	探索失敗	96
スラックノード	37	探索速度	34
正規デフォルト	9	探索点	34
制約	117, 139	探索トレース済み	42
制約違反	91	探索フェーズ	30
制約違反度(Infeasibility, IF)	30	単体法	33, 60
制約ロジックプログラミング	117, 139, 169	単調推論	54
積和形	108	知識獲得のボトルネック	16
切除平面法	24	知識コンパイル	21, 107
説明に基づく学習	21	知識処理	60
0-1整数計画問題	22	知識ネットワーク	23, 36, 105
全解探索	108	知識の実行時リフォーメーション	108
線形(オーダ)時間	30	知識の部分コンパイル	21
線形計画問題(LP)	30	知識リフォーメーション	108
線形評価関数	34	知識レベル	19
選言(OR)ルール	106	知識レベルのアブダクション	8
前件	5	長期記憶(LTM)	27, 116, 121
前提	9	直感	60
束縛情報伝播	125	強い人工知能	139
大前提	1	手に負えない(intractable)問題	19
代入実行部	119	デフォルト理論	8
代入スイッチ	118	天啓	2
タイプ1(の矛盾)	132	伝播パス	41
タイプ2(の矛盾)	54, 133	伝播パスのサイズ	51
タイプ3(の矛盾)	137	伝播率	37, 45
タイプ4(の矛盾)	138	等号条件テスト部	118
タイプ4(の矛盾)	140	特徴A	127
		トートロジー	79

トップダウン	64, 115, 124	BPN	38, 115
二値条件	28	BPNの動作原理	45
ネットワーク化バブル伝播法	24, 35	$\phi$ 束縛の問題	127
ネットワーク状態	39	$\phi$ による束縛 ( $\phi$ 束縛)	128, 132
ノードの状態	39	ファクト型	6
ノードの幅	84	フィルターノード	123
Non-equality test	133	Fixpoint Operation	124, 125
背景知識	3	不等式集合	28
背景知識と等価な不等式制約	74	primal network	84, 85
発火	130	Prime Implicants	107
発火前	46	Prolog	20, 168
発火後	46	Prolog2	168
掃き出し補数法(Pivot and Complement法)		Prolog3	168
	22, 30	focusing	7
掃き出し操作	33	分枝限定法	24
バックトラック	14	平均出次数	111
バックトラックなし探索	73	論理簡単化	108
発想的推論	1	平坦地	42
バブル	39	閉部分グラフ	56
バブル伝播	39, 41	並列計算	11
バブル伝播(掃き出し)操作	51	並列推論マシン(PIM)	21
バブル伝播ネットワーク	35, 38	閉路解消	110
バブルの個数	39	ヘッド部	5
バブルの伝播先	45	変化率	45, 46, 48
半自動化	139	補助線を用いた定理証明	61
非基底変数	30	補数	31
否定	9	ボディ部	5
非基底構造 (●) ノード	39	ボトムアップ	21, 124
非基底スラック (■) ノード	39	ホーン節	5
BK (背景知識) 単位	121	枚挙法	61
非整数度	30, 58	マジックセット法	116
非線形関数	34	見えない支配関係	55
非線形計画問題	35	MIS	109
PC法	22	未知語を含む自然言語文の解析	16
ビットベクトル	118	矛盾	10
ビットベクトル $\phi$	119	無前提正規デフォルト理論	10
ヒューリスティックス	15	命題知識ネットワーク	23
評価関数	29	迷路	13

目星	34
モデル生成に基づく並列アブダクション	116
問題の解決	60
問題の取扱い易さ	66
問題の複雑さ	74
山下り法	35
融合原理	20
様相論理	8
要素仮説	3, 29
弱い人工知能	139
rank	45
リアルタイム地震学	10
離散的な情報	19
ループ検出	55
ルール	29, 38
ルール中央のスラックノード	102
regular width-k	85
Range restricted	118, 141
論理式制約	28
論理に基づく仮説推論	2
論理和	29
離散的な解空間	34
連言 (AND) ルール	106
連言	6



## B. 人物索引 (アルファベット順)

主な参考文献の第一著者, および文中で紹介した実名

阿部明典	21	伊藤史朗	21
Adams, J.	35	井上克巳	21
Aho, A.	116	伊理正男 (Iri, M.)	24
Aristotle	1	Kamath, A.P.	35
Arnborg, S.	87	Karmarker, N.	30, 73
Balas, E.	26, 30	Kautz, H.A.	13, 22
Bancilhon, F.	116	Kilfer, M.	124
Brayton, R.K.	109	Kirkpatrick, S.	33
Bylander, T.	13, 22, 82	近藤朗子	21
Charniak, E.	3	Korf, R.E.	123
Colmerauer, A.	20, 168	Kowalski, R.	20
de Kleer, J.	7	Levesque, H.J.	8
Dantzig, G.B.	35	牧野俊朗	4
Dechter, R.	72, 73, 81, 85	McDermott, D.	9
Dhar, V.	28, 83	Minker, J.	116
堂前宣夫	108	Minton, S.	34
Dowling, W.F.	83	Mooney, R.J.	2
Freuder, E.C.	73, 81, 84	Newell, A.	19
藤田 昌宏	109	岡本知樹	28, 69, 115
二田丈之	35	Pierce, C.S.	2
Gallier, J.H.	83	Popper, K.R.	2
Geffner, H.	72	Poole, D.	2, 65
Glovner, F.	34	Pople, Jr, H.E.	9
Goebel, R.	2, 9	Reiter, R.	8
Goldberg, D.E.	33	Robinson, J.A.	20
Gu, J.	35	Roussel	20
橋田浩一 (Hasida, K)	125, 167, 125	Santos, E.Jr	24
Hempel, C.G.	2	Selman, B.	34, 82
Hooke, J.N.	28	Stillman, J.	10
Hugh, T.	168	高間康史	26, 114
茨木俊秀	30, 35	鶴田三郎	107
Ioannidis, Y.E.	116	富田悦次	34
Iri, M	24	Vieille, L.	21, 116, 125
石塚満	4	横井俊夫	168