

TM-0173

故障診断における仮定に基づいた
推論方式

和田慎一，古関義幸
(日本電気)

June, 1986

©1986, ICOT

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

故障診断における仮定に基づいた推論方式

和田 慎一 古閑 義幸

日本電気㈱ C & C システム研究所

〔概要〕

電子装置の故障診断における不確かな推論を実現するための推論方式として、仮定に基づいた推論方式について述べる。判断上不確かな内容を仮定して推論を進めた結果、故障原因として導かれた部品を交換しても故障が直らない場合などを矛盾として検出し、仮定をくつがえすことにより正しい推論結果を導く方式である。この方式では、テストの予想結果を仮定して推論を進め、矛盾の検出によってそれが疑わしくなったときにテストを実行させることも実現でき、実行コストの大きなテストの実行が省略できる。本稿ではこのような推論方式の故障診断における必要性、適用の方法、および矛盾が検出された際の仮定の制御の方法について述べる。

1. はじめに

電子装置等の故障診断では、そのための判断に不確かさが含まれており、このため故障診断エキスパートシステムにおける推論では不確かさを取り扱う必要がある。判断の不確かさを取り扱う方法としては確信度など事実の確からしさを数値として評価する方法もある[1]。しかし、このような方法は不確かさを与える個々の理由を区別し、個々の理由の成立、不成立に応じて変化する事実関係を取り扱うものとはなっていない。

このための推論の形式として、dependency-directed backtracking あるいは truth maintenance[2], [3]と呼ばれる、仮定に基づいた推論方式がある。この方法は、不確かな事実や判断を正しいものと仮定して推論を進め、矛盾の検出された場合に誤った仮定を修正するというものであり、この方法を故障診断へ適用することが考えられる。すなわち、不確かさを含む判断を正しいと仮定した上で推論を進めて交換部品を導いていき、部品を交換しても故障が直らなかった場合等を矛盾とし、その場合には関連している仮定を疑って別の推論を行なうという形で適用することができる。

この推論の方法を利用する場合、矛盾が検出されたときにどの仮定を疑い、くつがえすかを決定する上で複数の選択の可能性がある。複数の可能性の中から選択を行なう際の方針として、各仮定に確度を表わす数値を与え、矛盾を含まない範囲で信じる仮定の確度の和が最大となるように仮定を信じる、という方針を考える。この方針に従った仮定の制御の

ためには、ある仮定をくつがえすことによって、すでにくつがえされていた別の仮定が再び信じられるようになる場合も扱える必要がある。

また、この推論方式により、実行コストが大きく、かつ、結果も予想できるようなテストについて、通常は実行を省略して代わりに予想結果に基づいて不確かな推論を進め、テスト結果についての予想が疑わしくなって初めてそのテストの実行を行なう、という形の診断の実現も可能になる。

本稿ではこのような推論方式の故障診断における必要性、適用の方法、および矛盾が検出された際の仮定の制御の方法について述べる。

2. 故障診断における不確かな判断

2.1 故障診断の流れ

人間の専門家は故障診断において、テスト（故障の原因について知るためのデータ・情報の収集）を行ない、その結果として得られた情報に基づいて被疑範囲（装置の中で故障箇所の候補として考えられる範囲）がどの範囲に限定されるか判断する、ということを繰り返す。テスト結果として得られた情報を基にして導きうる限りの結論を得ることによって被疑範囲を狭く絞り込むことができ、その範囲に該当するパッケージやケーブルなど交換可能な部品一個または数個を故障の原因として挙げることができる。その部品を交換することによって故障が直れば、診断を終了する。

2.2 不確かな判断の存在と必要性

このような診断で行なわれる判断のうち、テスト結果を基にした被疑範囲についての判断には不確かさが含まれる。電子装置の故障診断の場合、判断の基にする情報・データは装置から出力される異常メッセージや種々のコマンドに対する装置の応答など外部的に観察されるものが多い。このような情報は内部の故障を起因として外部に現われるまでに、装置の実行の状況やタイミングなど種々の条件の影響を受けているものである。このような情報を基にして被疑範囲に関して行なう判断は、「大体・・・だろう。」という不確かな判断となってしまう。

不確かな判断の具体的な例を挙げる。図1のような構成の装置においてC1, C2, C3が同一の構成の部品であり、A-Ci (i=1, 2, 3)間では定常的に信号の伝達が行われているものとする。A-Ci間の信号の伝達に

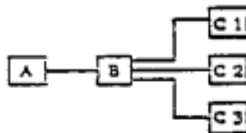


図1 装置の構成

ついで異常検出の機能として、異常が存在し、かつ、さらにある確率的条件が満たされた場合に限って異常が検出・出力されるという機能があるものとする。このような装置において、A-C3間のみで異常が検出され、A-C1, A-C2間では異常が検出されていない状況を考える。この場合、異常に関係している部分Aを故障原因として疑うとすると「Aが故障の原因であるならばおそらくA-C1, A-C2間でも同様に異常が検出されることになるはず」と考えられるが、A-C3間のみで異常が検出されているという状況と合わない。このことから不確かな判断として「Aは故障の原因ではない」という判断が行なえる。

今まで述べてきたように、故障診断には不確かな判断が存在する。有効なテストの数が限られている場合、得られたテスト結果のデータをもとに被疑範囲をより狭く絞り込むためには不確かな判断も含めて用いる必要がある。判断に不確かさが含まれているとしても、それに基づいて絞り込まれた被疑範囲に該当する部品を交換し、それによって故障が直れば診断を終了することができる。この意味で不確かな判断は有効なものとなっている。

不確かな判断を含めて推論を進めていった場合、もしも判断に誤りがあるときには、推論の結果導かれる故障部品が誤ったものとなる。部品を交換してみても故障が直らない場合には、それまでの推論の過程で用いられた不確かな判断に誤りがあったものと考え、その判断を取り消して別の結論を導くなどの対処を行なうことになる。

2.3 テスト実行の省略

故障診断では、テストに関しても不確かな推論が行なわれる。

テストの中には実行コスト（時間、手間など）が大きい実行が好ましくなく、かつ、結果も予想できるようなものがある。そのようなテストについては実行を省略し、代わりに予想される結果に基づいて被疑範囲について不確かながら判断をするということが行なわれる。推論上何らかの誤りが見つかり、テスト結果についての予想が疑わしくなってからそのテストの実行が行なわれる。

3. 仮定に基づいた推論の表現形式と推論機能

2. で述べた故障診断における不確かな推論を計算機上のエキスパートシステムとして実現するための知識表現形式・推論機能について説明する。故障診断における判断はルールとして表現し、これに基

づいて推論することができる。これを3. 1で説明する。次に不確かさを含む場合のための表現・機能の拡張として、3. 2で仮定・矛盾を用いた表現形式について、3. 3では推論機能として仮定の制御の機能について述べる。

3.1 ルールによる判断の知識の表現と前向き推論

故障診断では、故障の症状や種々のテスト結果の事実を基にして被疑範囲を限定していくための判断の表現として、次のようなルールを用いることができる。

```

if    premise1.
      premise2.
then conclusion
  〈意味〉premise1とpremise2が成立するならば
          conclusionが成立する。
  
```

ここで、if部のpremiseでは事実の成立のみを記述し、then部のconclusionではルールにより導かれる事実を宣言するのみで事実の消去やその他のアクションを含まない形に記述を制限している。これにより、ルールの実行においては事実の導出が単調に行なわれる。

簡単なルールの例として次の2つを示す。

(ルール1)

```

if    電源アラームが点灯していない
then 電源は正常（被疑範囲は電源以外の部分である）
  
```

(ルール2)

```

if    電源は正常.
      パイロットエラーXが検出されていない
then 部分Aは正常
  
```

テスト結果の事実が与えられたときに、このようなルールに対して前向き推論が行なわれる。発火可能なルールをすべて発火することにより、テスト結果を基に導出するすべての事実が導き出される。前述のようにルール記述を制限していることから、ルールの発火順序には関係なく、最終的には定まった事実が導き出されることになる。

3.2 仮定・矛盾の導入

判断に不確かさが含まれている場合の表現形式について説明する。仮定や矛盾を用い、不確かさや誤りの存在を表わす。この表現形式に対する推論機能については3. 3で説明する。

3.2.1 仮定の導入

判断に含まれる不確かさを表わすために仮定を用いる。不確かな判断は例えば次のような形で表現する。

```

if    premise1,
      premise2,
      believed asmxxx
then conclusion

```

〈意味〉premise1とpremise2が成立し、仮定(asmxxxとして表わす)が信じられているならばconclusionが成立する。

仮定は信じられている／信じられていない、のいずれかの状態をとる。ルールの場合部のbelieved asmxxxは仮定が信じられているときに成立する。このため、ルールが成立するのは仮定が信じられているときに限られる。仮定の状態(信じられている／信じられていない)は3.3で述べる仮定の制御の機能によって決定・制御されるが、通常はこのような仮定を信じ、不確かな判断がより多く用いられるように制御される。

故障診断におけるルールの例を示す。前述の(ルール1)を、その不確かさについて仮定を用いることにより正確に表わすと次のようになる。

(ルール1a)

```

if 電源アラームが点灯していない,
   believed "電源アラームは正常"という
   仮定
then 電源は正常(被疑範囲は電源以外の部分である)

```

また、テスト実行を省略させたい場合、省略させたいテストの予想結果も仮定としてその仮定に基づく判断をルールで表現する。その仮定が疑わしくなったときにはテスト実行が起動される。これについては3.3.2で説明する。

3.2.2 矛盾の導入

仮定に基づいて推論を行なう場合、仮定が誤っていると事実間に起こり得ない関係が生じる。これを矛盾とする。矛盾があるということは例えば次のような形で表現する。

```

if    premise1,
      premise2,
then contradiction

```

〈意味〉premise1とpremise2が同時に成立しているならば矛盾である。

故障診断においては故障の原因として導かれた部品を交換しても故障が直らなかった場合や、被疑範囲が空集合となった場合、その他、事実間に起こり得ない関係が生じた場合を矛盾として表わす。

3.3 仮定の制御の機能

以上に述べた表現形式に対する推論の機能としては、与えられたルールを前向きに発火していく前向

き推論機能の他に、導入した仮定・矛盾に関する機能として、矛盾が検出された場合にその矛盾にdependency(ルール実行履歴)の上で関係している仮定の状態を操作し、矛盾を解消する仮定の制御の機能がある。

3.3.1 仮定の制御における複数の可能性

仮定の制御の機能は[2],[3]における機能と同様なものである。しかし、検出された矛盾を解消する際にとりうる仮定の状態として複数の可能性があり、そのどれを選択するかという点で問題が残されている。ここでは、矛盾を解消するためにとりうる仮定の状態として複数の可能性があることを例によって説明する。特に複数回矛盾が検出された場合には、すでにくつがえされていた仮定を再び信じ得ることを示す。次の3.3.2では複数の可能性のある中からの選択するために採用した仮定の制御の方針について述べる。

図2のようなルールがあるとする。そのようなルールについて図3の順序で推論が進められているとする。その場合、次のような仮定の状態をとる可能性がある。

(1) a)の状態fact1a, fact2aが成立した場合、矛盾を解消するためには

- ①asm2をくつがえし、b)の状態をとる
- ②asm1をくつがえし、c)の状態をとる

の2つの可能性がある。

(2) さらに、b)の状態においてasm1, asm3の間で矛盾が検出された場合、矛盾を解消するために単に仮定をくつがえすのみを行なうものとする

- ①asm3をくつがえし、d)の状態をとる
- ②asm1をくつがえし、e)の状態をとる

の2つが可能であるが、別に一旦くつがえされていた仮定asm2を再び信じることにより

- ③asm1をくつがえしてasm2を再び信じ、c)の状態をとる

も可能である。

3.3.2 仮定の制御の方針

3.3.1で述べたように矛盾を解消するためにとりうる仮定の状態として複数の可能性があるが、その選択のための方針について述べる。

故障診断ではより多くの判断を行なうことによって被疑範囲をより狭く絞りこむことができる。そのためには、矛盾を含まない範囲でより多くの独立した仮定を信じ、より多くの有効な判断が用いられるように保つことが望ましい。

また、仮定によっても信じることの有効性に差がある。仮定によって成立の確からしさの度合いが異なる。あるいは、仮定によってそれを信じた場合の問題解決の複雑さが異なっている。それを信じる場合は問題が単純で、簡単に解決できるという意味か

ら、信じることが望ましい仮定もある。

このため、各仮定毎に優先度を表わす正の値を確度として与えることにし、仮定の制御の際には矛盾を含まない範囲でその値の和が最大になるように信じる仮定を選ぶものとする。すなわち

〈方針1〉信じる仮定の確度の和が最大となるように仮定を信じる

という方針をとる。

また、2. で述べたように結果の予想でき、実行コストの大きなテストについては通常は実行を省略し、代わりに予想される結果に基づいて推論を進めておき、それが疑わしくなったときに初めて実行させるという処理が望まれる。この処理のためにテストの予想結果を仮定としてルール記述をすることは3. 2. 1の仮定の導入の項で述べた。矛盾が検出された際には、そのような仮定が矛盾の原因として疑わしいならば、省略していたテストを実行し、その結果により仮定の真偽を確認する必要がある。こ

のため、

〈方針2〉省略していたテストの予想結果として立てられた仮定が疑わしい場合にはテスト実行により実際の真偽を確認する

という方針をとる。

今述べた2つの方針に基づく仮定の制御としては、まず、方針2により、矛盾の原因として疑わしい仮定のうち、実際の真偽を調べられるものについては、省略していたテストを実行することにより実際の真偽を確認する。これによって仮定の状態の選択の可能性が狭められる。次に、方針1に基づき、仮定の実際の真偽ではなく、各仮定に数値として与えられた確度に基づいて、狭められた可能性の中から選択を行なって仮定の状態を決定するものとする。

4. 仮定に基づいた推論の実現方法

3. で述べた推論の機能は前向き推論や仮定・矛盾の基本的操作を行なう“前向き推論システム”とこ

```

<rule1>
if fact1a,
   believed asm1
then fact1b

<rule2>
if fact2a,
   believed asm2
then fact2b

<rule3>
if fact3a,
   believed asm3
then fact3b

<rule4>
if fact1b,
   fact2b
then contradiction

<rule5>
if fact1b,
   fact3b
then contradiction
  
```

図2 ルールの例

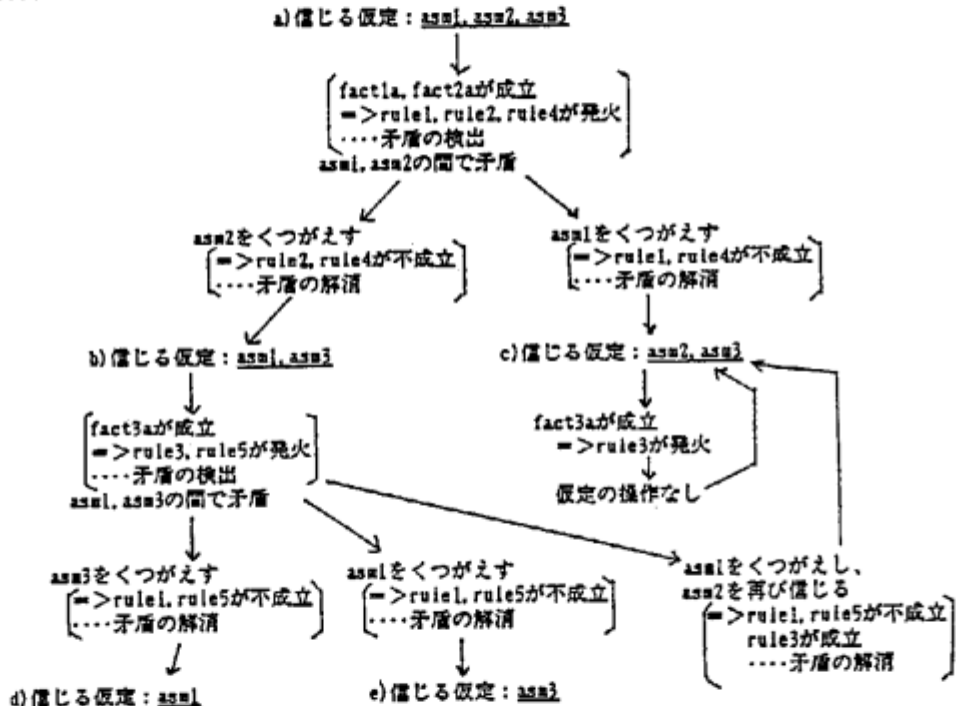


図3 仮定の操作の例

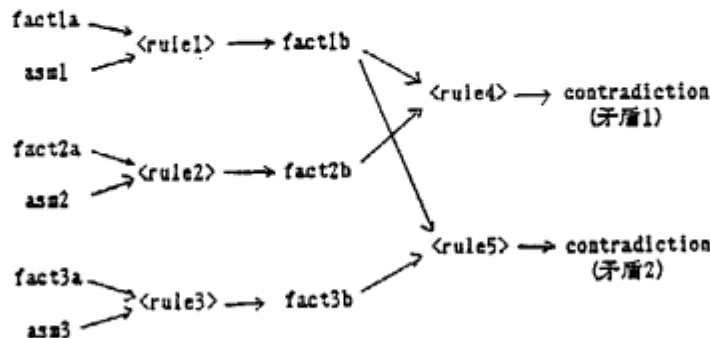


図4 dependencyの例

れに対する制御を行ない、矛盾が検出された際にはその解消を図る“仮定制御システム”の2つの部分によって実現される。

4. 1 前向き推論システム

前向き推論システムは3. 1で説明したような前向き推論の機能の他に仮定や矛盾の取り扱いのため、矛盾の存在を検出する機能、矛盾と関係する仮定を取り出す機能、及び仮定の状態を操作する機能などを持ち、仮定制御システムの制御により動作する。

前向き推論システムは仮定・矛盾の取り扱いのためdependency（ルール実行履歴）を記録する。dependencyとはルールの発火に応じて記録する。a)前提の事実や仮定、b)発火したルール、c)結論として導かれた事実や矛盾、の間の関係である。例えば図2のルールがすべて発火した場合のdependencyは図4のように表わせる。前向き推論システムはdependencyを利用して次の処理を行なう。

①前向き推論システムとしてルール発火を進める。ルール発火の過程で矛盾が検出された場合には、仮定制御システムの処理を起動する。発火しうるすべてのルールを発火した場合には終了する。この場合、事実間に矛盾の存在しないことがわかる。

②各事実や矛盾がどの仮定に基づいて導かれたものか調べることで、矛盾を含む仮定の組（＝含まれているすべての仮定を信じると矛盾を生じさせるような仮定の組）を得ることができる。これをconflict setと呼ぶ。

図4の例では矛盾1のconflict setは{asm1, asm2}となる。

③仮定や事実に基づいて導かれている事実の関係を参照することによって、仮定をくつがえす処理（仮定の状態を切り替えるとともに仮定に基づいて導かれた事実を消去すること）を行なう。

図4の例ではasm1をくつがえすとfact1b, contradiction（矛盾1）, contradiction（矛盾2）が消去される。

4. 2 仮定制御システム

仮定制御システムは前向き推論実行中に矛盾が検出されたときに、3. 3. 2で述べた仮定の制御の

方針に従って仮定の状態を決定し、前向き推論システムの機能を利用して仮定を操作することにより矛盾を解消する部分である。仮定の制御の方針1の実現のために、同時に信じ得る仮定の組み合わせの集合としてCBS（Consistent Belief Set）を保持する。まず4. 2. 1でCBSについて説明し、その後仮定の制御の処理の全体の手順について説明する。

4. 2. 1 CBSの導入による仮定の管理

仮定の制御の方針1のためには、信じる仮定の確度の和を最大とするように保つ必要がある。矛盾が検出された場合には、仮定の制御の処理としてconflict setに含まれる複数の仮定のうち、どれかをくつがえす必要がある。仮定に関して行なう処理が仮定をくつがえすのみであるとするならば、「矛盾が検出された際には、conflict setに含まれる複数の仮定のうち確度の最も小さな仮定をくつがえす」という方法により仮定の制御の方針1が満足される。

しかし、3. 3. 1で例示したように一旦くつがえされていた仮定を再び信じることの可能な場合もあり、それも考慮する必要がある。このためには同時に信じうる仮定の組、すなわち、矛盾を含まない仮定の組をすべて保持する方法が考えられる（ただし、他の組に包含されるものは除く）。実際に信じる仮定の組としてはその中のどれかを選べばよい。このため、CBS（Consistent Belief Set）として次の2つの条件を満たす仮定の組をすべて含む集合を保持するものとする。

- ・以前に検出された矛盾のconflict setを包含したり、同一ではない。
- ・CBSに含まれる他の要素（仮定の組）に包含されない

矛盾が検出された際にはそのconflict setに応じてCBSを更新する。

仮定の制御の方針1は、CBSに含まれる仮定の組のうち、仮定の確度の和が最大になるような仮定の組を求め、その組に含まれる仮定を信じることによって実現できる。

CBSの更新の方法を例によって説明する。図5は3. 3. 1で示した例と同じ仮定・同じ矛盾が生

状態	矛盾の conflict set	CBS	信じる仮定の組	信じる/ くつがえす仮定
a)		{asm1, asm2, asm3}	{asm1, asm2, asm3}	
	{asm1, asm2}			—/asm2
b)		{asm1, asm3}, {asm2, asm3}	{asm1, asm3}	
	{asm1, asm3}			asm2/asm1
c)		{asm1}, {asm2, asm3}	{asm2, asm3}	

図5 CBSの更新の例

じた場合のCBSの更新の過程を示したものである。(実際に信じる仮定の組も示されている)。特に状態b)からのCBSの更新について以下に示す。

①CBSの要素となっている仮定の組のうち、矛盾のconflict setを包含しているか、同一となっている仮定の組を取り出す：

状態b)のCBS $\{(asm1, asm3), (asm2, asm3)\}$ の要素のうち、 $(asm1, asm3)$ が矛盾のconflict setと同一のものとして取り出される。

②conflict setに含まれる各々の仮定を①で取り出された仮定の組から除くことによって新たな仮定の組(複数)を得る：

conflict setも、①で取り出された仮定の組も同じ $(asm1, asm3)$ である。新たな仮定の組として $(asm1)$ と $(asm3)$ が得られる。

③CBSの要素のうち、①で取り出された仮定の組を②で得られた仮定の組で置き換えることにより、新たなCBSを構成する。その際、CBSの要素(仮定の組)の中で別の要素に包含される仮定の組はCBSから除く：

CBSの要素のうち、①で取り出された仮定の組 $(asm1, asm3)$ を②で得られた仮定の組 $(asm1)$ 及び $(asm3)$ で置き換える。この際、 $(asm3)$ はCBSのもう一方の仮定の組 $(asm2, asm3)$ に包含されるため、それを除くことにより、新たなCBSとして $\{(asm1), (asm2, asm3)\}$ が作られる。

4.2.2 仮定の制御の手順

仮定の制御の処理は前向き推論システムで矛盾が検出された場合に起動され、前述のCBSを保持・更新しながら矛盾の解消を図る。処理の全体の手順を以下に示す。[*], [**], [***]のついた項については、後で詳しく説明する。

- (1) 検出された矛盾に対して次の処理を行なう：
 - (1-1) 前向き推論システムの機能を利用して、conflict set(矛盾を含む仮定の組)を取り出す。
 - (1-2) conflict setに応じてCBSを更新する。
- (2) 省略したテストの予想結果として立てた仮定が疑わしい場合のテストの実行(方針2に基づく)
 - (2-1) 真偽を確認すべき仮定の取り出し：
次のような仮定を探す。もしも該当する仮定が存在しなければ省略されたテストとして実行すべきものはないことになる。
 - a) 省略したテストの予想結果として立てられた仮定である。
 - b) 矛盾の原因として疑わしい仮定であり、仮定の状態を決定する上で真偽を調べる意味がある。[*]
 - (2-2) 該当する仮定が存在する場合には省略

していたテストを実行して仮定の真偽を調べる。

(2-3) テスト実行結果として得られた仮定の真偽によりCBSを更新する。[**]

(2-4) 再び(2)の始めから実行する。

(3) 仮定の制御の方針1に従い、CBSを参照して信じる仮定、くつがえす仮定を決定する。その決定に基づいて仮定の状態を実際に操作し矛盾を解消する。[***]

[*] (2-1) のb)の条件の意味の方法：

仮定が“矛盾の原因として疑わしい”、あるいは、“真偽を調べる意味がある”ことの確認の方法について述べる。仮定の“真偽を調べる意味がある”のは、“矛盾を解消するような仮定の状態として、a)その仮定を信じるような状態と、b)その仮定を信じないような状態の両方が存在する”場合であると考えることができる。仮定の実際の真偽がわかれば、仮定の状態の候補をそのどちらかに絞ることができる。

矛盾を解消するような仮定の状態は、検出された矛盾に応じて更新されたCBSの要素(仮定の組)により表わすことができる。これにより、真偽を調べる意味のある仮定とは次の2つを満たすものであるとすることができる。

(1) その仮定を含み、実際にルール発火をしてみても矛盾を含まないことの確認された仮定の組がCBSの要素として含まれる。

(2) その仮定を含まず、実際にルール発火をしてみても矛盾を含まないことの確認された仮定の組がCBSの要素として含まれる。

ここで、仮定の組が実際にルール発火をしてみても矛盾を含まないことを確認する方法と理由について述べる。CBSは前向き推論中に検出された矛盾に応じて更新されているが、前向き推論において発火されていないルールの中に矛盾が含まれているとしても、それに応じた更新は行なわれていない。したがってCBSの要素の仮定の組が矛盾を含まないことの確認をとるため、実際に仮定の状態を操作して前向き推論システムにより未発火のルールを発火してみるものとする。もしも矛盾が検出された場合にはそのconflict setに応じてCBSを更新し、全体の手順の(2)の始めから実行すればよい。

[**] (2-3) における仮定の真偽についてのテスト実行結果に基づくCBSの更新：

もしも真であると確認された場合にはCBSから、その仮定を含まない仮定の組を除く。

もしも偽であると確認された場合にはconflict setがその仮定のみを要素として含む集合であるものとして、そのconflict setに応じたCBSの更新を行なう。

[***] (3)の方法:

CBSに含まれている仮定の組のうちで、仮定の確度の和が最大となるような仮定の組を求め、その組を信じる仮定の組とする。それまでに信じていた仮定の組と比較して、くつがえす仮定、信じる仮定を決定し、状態を操作する。

5. おわりに

本稿では仮定に基づいた推論の故障診断における必要性、適用の方法、そのための仮定の制御の方法について述べた。この方法を用いることにより、判断に含まれる不確かさが自然に記述でき、実行においては矛盾の生じない範囲でより多くの有効な判断が用いられる。また、実行コストの大きなテストについて、通常は実行を省略して予想される結果に基づいて推論を進め、予想結果が疑わしくなったときになって初めてそのテストの実行を行なうことが実現できる。

Prologを用いて仮定に基づいた推論を行なうシステムおよびその上の知識ベースを試作しているが、処理の効率などに課題がある。

また、このような推論の応用として、ユーザの推測する内容を一つの仮定として事実間の関係の一貫性を保った形で取り入れ、利用するという処理の実現の可能性も考えられる。すなわち、利用者の推測する内容を、仮定に基づいて成立する事実として(その仮定の確度を十分大きな値として)加えるものとする。もしもその推測内容が他の事実と矛盾する場合、矛盾した事実が何らかの仮定に基づいているならばその仮定をくつがえすことにより矛盾が解消され、事実間の一貫性が保たれることになる。また、矛盾した事実が何の仮定にも基づいていない場合は推測内容のほうが誤ったものとして棄却されることになる。今後はこれらの課題について検討していきたい。

なお本研究は第5世代コンピュータプロジェクトの一環として進めているものである。

【謝辞】

本研究を進めるにあたり、貴重な意見を頂いた(財)新世代コンピュータ技術開発機構研究所第五研究室岩下室長に深謝します。日頃御指導頂く当研究所真名垣部長、後藤課長、森主任に感謝します。

【参考文献】

- [1]Shortliffe, E.H. : Computer-based medical consultation:MYCIN. New York:American Elsevier (1976)
- [2]Stallman, R.M. and Sussman, G.J., "Forward reasoning and dependency-directed backtracking in a system for computer-aided circuit analysis", Artificial Intelligence 9(1977)135-196

- [3]Doyle, J., "A truth maintenance system", Artificial Intelligence 12(1979)231-272
- [4]Charniak, E. et al.: Artificial Intelligence Programming, Lawrence Erlbaum Ass. (1980)