知識工学と人工知能 40 － 7
（1985. 5. 7）

多重世界機構を用いた非線形系の質的推論

中川裕志
.（横浜国立大学・工学部）

1．はじめに

抽象的な toy world から離脱し、現実の世界を直視し、常識的な知識を扱うことに真正面から取り組もう、というスローガンは70年代後半の衰退した人工知能の再生を目指して叫ばれたものであった。Pat.J.Hayes は彼の高名な論文（The Naive Physics Manifesto ）[6] はこのスローガンを雄弁に語っている。しかしながら現在の計算機パワーでは直接的なアタックによっては現実の世界のちゅう密さの壁を突き崩すことはできない。そこで考えられたのが、人間が物理学等の問題を解く際に行なう直感的な推論を範とする質的推論（ qualitative reasoning ）と呼ばれる分野である。

従来の物理学においては世界をアナログ的にとらえ、微分方程式によって世界の状態変化を表わす法則を記述した。世界の状態自体は微分方程式を解くことによって得ている。しかし人間の直感的な推論は例えば“体積一定なら気体の温度が上がると圧力が増す。”というように、世界の状態あるいはその変化を、抽象的な形で推論している。抽象的とは即ち変数が極く僅かな値しかとり得ないことを意味する。上記の例では温度変化は“上がる、下がる、変化せず”の３値しかとらない。質的推論とは一口で言えば上記の例のような推論法を定式化したものである。このような質的推論の目的としては次のようなものが考えられている。

(1) 従来の物理学で欠落していた物理現象の推論方式の確立
(2) 物理現象の理解しやすい説明の構成
(3) (1),(2) の物理学の教育への応用
(4) 人工知能システムへ常識を組み込むにあたっての有力な定式化の候補

質的推論に関してはアメリカの東海岸を中心に80年代初頭から研究が進められた。最近 Artificial Intelligence Journal に質的推論が特集されており（ Vol. 24）人工知能の有力な分野を形成している。De Kleer等の constraint oriented な方法[1],[2],[3] 、Ken Forbus等のprocess oriented な方法[3],[4] 、Kuipers の質的シミュレーション[7] などが知られている。De Kleer 等の方法は、物理系の各要素間に定義される質的微分方程式群を制約条件として捉え、質的変数の値は+ 0 - の３値しかとらない。各質的変数の値は質的微分方程式群を満たすように制約条件を伝播させるpropagation 法で推論される。この推論方法については次節で詳述する。Ken Forbus 等の方法は、より抽象的な process についての質的方程式を用いる。質的変数に Magunitude と呼ばれる比較尺度を導入しており、各質的変数間に半順序関係がある。したがって、システムの状態を推論するにあたっては、半順序集合上での扱いに近い方法が可能である。Kuipers 等の質的シミュレーションは、質的微分方程式を制約条件の伝播によって解く点では、De Kleer の方法とにている。しかし時間推移に伴う状態変化のシミュレ

ーションは、landmarkと呼ばれる一種の平衡点への移行を目指して行なわれ、微分方程式を直接解く方法ではない。この点で、彼らの方法は推論以前に各状態に関して相当な意味を持ち込んでいる。それ故、De Kleer によって、qualitative physics でなく qualitative mathematics であると位置付けられている。

直感的にもわかるように、質的推論が線形なシステムのみならず、時変系や非線形系の解析あるいは動作の解釈、説明に有効であろう、と指摘されてはいる。しかし、以上の研究においては、現在のところ対象としている系は主として線形系であり、時変系、非線形系の扱いに関して確立された方法は定まっていない。そこで本論文では質的推論にUranusの多重世界機構[10]を組み合わせることにより非線形な回路の動作を推論する方法について述べる。

2．Propagation による状態の質的推論

本節では、propagation による質的推論について De Kleer の論文[1],[3] をベースにして説明し、併せて我々がUranusを用いて開発したシステムの推論法について触れる。（正確には中島氏によって開発されたUranusのサブセットをPC-9801 上へ移植したVersion である。）

質的変数の演算

質的変数は正 ＋ 、零 0 、負 - のいずれかの値しかとらない。ふたつの質的変数 X , Y の間での加算は表１に示すように定義されている。

表１

Y ＼ X	-	0	+
-	-	-	?
0	-	0	+
+	?	+	+

なお ? は + 0 - のいずれもとりうる可能性がある、と解釈する。

質的微分方程式

質的微分方程式（De Kleer はこれを confluence と呼んでいる。）は連続系に対する解析的な微分方程式から導く。これを次のインダクタンスL に流れる電流 iと電圧 vの間に成り立つ関係の例で示す。

$$v = L \cdot di/dt \qquad (1)$$

ここで L は常に正だから、v の正、負、0はdi/dt にだけ依存して決まる。従って、v の質的変数をV とし、（以後小文字で表わされた連続系の変数に対応する質的変数は同じ文字の大文字で表わす。また今後は特に断わらない限り、変数あるいは方程式と書いた時は質的のものを意味する。） 微分di/dt の質的表現を D・I で表わすと、(1) に対応する質的微分方程式は次のようになる。

V = D・I (2)

拘束条件の伝播 (propagation)による推論

質的微分方程式系が決まると、ある時点における各質的変数の値は propagationと呼ばれる方法で推論される。De Kleerの論文で引用されている pressure regulator の例でこの方法を示そう。ここでは物理的背景については触れずに、次の質的微分方程式系が与えられたとして話を始める。

D・PIO + D・XF = D・Q1V ---(3.1)
D・XF + D・POS = 0 ---(3.2)
D・PIO + D・POS = D・PIS ---(3.3)
D・QT2 + D・Q2V = 0 ---(3.4)
D・QT1 + D・Q1V = 0 ---(3.5)
D・Q1V + D・Q2V = 0 ---(3.6)
D・QT2 = D・POS ---(3.7)

まず D・Q1V = + という条件が与えられたとする。すると(3.6) と表1により D・Q2V = - であることがわかる。次に値の決まった D・Q2V と(3.4) より D・QT2 = + が求まり、同様にして(3.7) により D・POS = + 、(3.2) より D・XF = - 、(3.1) より D・PIO = + 、(3.3) より D・PIS = + 、(3.5) より D・QT1 = - 、が求まる。このように、既に値の求まった質的変数とまだ使われていない質的微分方程式を用いて値の定まっていない質的変数の値を次々に求めていく方法をpropagation という。De Kleer のENVISIONというシステムでは自然演えき法によってこの推論を行なっている。

Prologによるインプリメンテーション

我々はProlog(Uranus)によって上記のpropagation 法をインプリメントした。各質的方程式には番号がついており、これを cs という述語で呼び出す。例えば上記の例の D・Q1V = + の条件で、(3.6) からpropagation を開始するには次のようにする。

```
(cs 6 (( D・Q1V  D・Q2V)) NIL (( D・Q1V +)) *X F)                (4)
```

第一引数は方程式番号、第二引数はこの方程式で使っている変数名、第三引数は既に適用済みの方程式番号のリスト、第四引数は与えられた条件、第五引数は推論結果で各質的変数とその値のペアのリストである。上記の例だと、次の結果になる。

```
(( D・QT1 -)(D・PIS +)(D・PIO +)(D・XF -)( D・POS +)(D・QT2 +)
 (D ・Q2V -)(D・Q1V +))                                          (5)
```

第六引数は動作モードを示すフラグである。

おおよその動作は次のようである。

(1) その番号の方程式と、第四引数で与えられた条件、及び既に値の定まった変数の値を考慮して未定の変数を求める。この時、演算は表1に従うが、値が一意的に定まらな

い場合もある。例えば、A = + で、A + B = + を解くと、B は + 0 - のいずれも可能性がある。従って一応、その変数の値を決めておき、もしバックトラックしてきたら別の解を返すようにする。また方程式を満たす値が存在しない場合は、failしてバックトラックする。
(2) (1) で新たに決まった変数を含む方程式を探す。もし見つかれば(1) へ。見つからなければまだ使われていない方程式を選び(1) へ。但し全変数の値が既に決定しているなら(3) へ。
(3) もし全方程式が既に使われているなら、結果を出力して終了。もしまだ使われていない方程式があれば、それらの方程式が各変数への値の割り当て方と矛盾しないかどうかチェックする。チェックモードでは CS の第六引数の動作モードフラグは R とする。もし矛盾がなければ結果を出力して終了。もし矛盾が検出されたら(2) へバックトラックする。

要約すればこの方法は、使用した方程式と、各変数とそれにバインドされた値の組を保持しておいて、バックトラックによって無矛盾な割り当てを探索するものである。

<u>推論結果の一意性</u>

propagation によって(D・Q1V = +)→(D・XF = -) を推論することはできたが、 D・XF = - を導出するのが D・Q1V = + に限られること、すなわち一意性はまだ示せていない。De Kleerはこれを D・Q1V ≠ + （すなわち 0 あるいは -)の場合には矛盾が導出されることによって証明している。De Kleerは各変数の値が一意的に定まることを基本的性質とみなしている。従って矛盾とは、この場合、propagation によって推論すると、D ・XF がふたつの値を持ちうることである。但し、propagation が局所的な制約条件のみによって推論したのに対し、一意性は大域的な性質に頼った推論になっている点を不満としている。なお我々のシステムでは述語 CS の第四引数の条件が ((D・Q1V 0) (D・XF -))及び ((D・Q1V -)(D・XF -))の場合に CS の呼び出しがfailすることを用いれば同様の推論ができる。

３．時間推移による状態変化の推論

前節ではある時刻における物理系の質的状態を推論する方法について述べた。しかし物理現象の把握のためには状態の時間的変化を推論しなければならない。この節では時間推移に伴う状態変化の推論法について検討する。

<u>De Kleerの状態推移の推論法</u>

De Kleerはの方法は、まず与えられた質的微分方程式系において可能な解を全て求める。次に各種の物理的条件を考慮して、時間的に引き続いて起こる可能性のある組を全て調べ上げ、これによって時間推移に伴う状態の推移を推論する。この際に考慮すべき物理的条件とは次のようなものがある。
(1) しきい値規則：変数の値が予め与えられたしきい値を越えない。

(2) 順序性の規則：変数Ｘの値が増加しているとしよう。ある要素Ａはｘに対してしきい値ｎで影響され、別の要素ＢはＸに対してしきい値ｍで影響される。いま、ｎ＜ｍなら要素Ａへの影響が先に現われる。これが順序性である。
(3) 特定の値からの変化：ある変数の特定の値からの変化は速やかに起こる。
(4) 微小順序規則：ある値＝０の変数の＋かーへの変化は、０以外の値を持つ別の変数の０への変化より早く起こる。
(5) 矛盾回避の規則：方程式系を満足しない解への推移や、物理的に直接インタラクションのない要素間にまたがる変化は起こさない。
(6) 連続性：全ての変数は連続的に変化する。
(7) 中間値の規則：変数Ｘに微分Ｄ・Ｘが定義されている時は次の時刻でのＸの値Ｘ'はＸ'＝Ｘ＋Ｄ・Ｘである。但しこの加算に表１を直接適用すると次の場合に不都合である。例えばＸ＝＋でＤ・Ｘ＝－の場合、表１からはＸ'＝－も可能である。しかしＸ'は＋→０→－の順で変化するものであり、０をとばすと不都合が生ずる可能性がある。従って、中間値の規則においては、積分された変数の連続性をも満たさなければならない。

時間推移のシミュレーション

ここでは、微分の定義されている変数の次の時刻での値を積分して求めることによって時間推移に伴う状態の変化をシミュレートする。この方法は後に述べる非線形系の動作解析に役立つ。さて、積分は上記の連続性を考慮した中間値の規則によって定義する。例題によってシミュレーションの様子を示そう。
（ＬＣ回路の振動のシミュレーション）
図１に示すＬＣ回路の微分方程式は次のようになる。

L di/dt = v　　　(6.1)
dv/dt + C i = 0　　　(6.2)

L と Cは常に正なので、質的微分方程式は、次のようになる。

D・I = V　　　(7.1)
D・V + I = 0　　　(7.2)

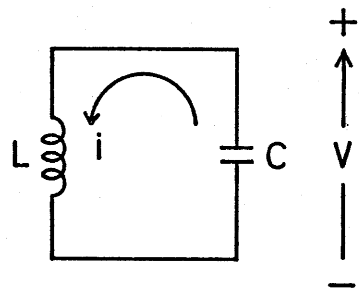

図１．ＬＣ回路

質的シミュレーションを行なうために、state という述語を用意した。

(state *oldstate *ic *newstate *integral *newintegral *intv)　　(8)

state は直前の時刻の状態を踏まえて、propagation による状態の推論を行ない、さら

に推論された状態において微分の定義されている変数の積分を行なって、次の時刻の状態を作り出す述語である。 *oldstate は直前の時刻の状態、*ic はpropagation を行なうための初期条件、*newstate はpropagation によって推論された状態である。但し、シミュレーションの進行を保証するため、*oldstate と*newstate は一致しないものでかつ連続性を満足する解を選ぶ。*integral はある変数とその微分の対からなるリストであり、この情報によって積分を行なう。積分の結果は*intv に得られ、また積分された以外の変数も含む新しい状態が *newintegral に得られる。

この述語state によって(7.1),(7.2) をシミュレートすると次の結果が得られ、振動していることがV とI の変化の様子からわかる。なお左からみて奇数番目の状態（上の行に * が付いている）は積分した結果で一時的なものであり必ずしも質的方程式を満足しない。この状態をDeKleer はmythical time と呼んでいる。また *ic の条件は直前に積分された変数とする。すなわち積分された変数の値はpropagation にあたっての前提条件となる。

		*		*		*		*		*		*		*		*		*
I	-	0	0	+	+	+	+	+	+	0	0	-	-	-	-	-	-	0
D・I	+	+	+	+	+	+	0	0	-	-	-	-	-	-	0	0	+	+
V	+	+	+	+	+	0	0	-	-	-	-	-	-	0	0	+	+	+
D・V	+	+	0	0	-	-	-	-	-	-	0	0	+	+	+	+	+	+

4．多重世界を利用した非線形系の質的推論

上記の質的シミュレーションは、propagation のベースとなる質的微分方程式系を変化させることによって、非線形系あるいは時変系をシミュレートすることができる。問題は(1) いつ方程式系を変化させるか、(2) 複数の質的方程式系を並列に保持する方法、である。我々は(1) の問題にはPrologのバックトラックを用い、(2) の問題はUranusの多重世界機構で対処している。簡単に言えば、propagation によって推論された状態が非線形領域に入って、それまでの方程式系では扱えなくなったら、バックトラックを起こして別の方程式系で扱うという方法である。この方法は我々が教室で物理を習った時よく出て来た“．．．になろうとすると、何々力が働き、逆にこうなる。”という仮定法の表現に対応している、と考えられる。以上の考えに従って、以下の述語を作った。

```
(assert (fork *oldwl *newwl *from *newstate)
        (mcscall *oldwl *from *newstate))
(assert (fork *oldwl *newwl *from *newstate)
        (append (wl) *oldwl *newwl)
        (mcscall *newwl *from *newstate))                    (9)
(assert (mcscall *wl *from *newstate)
        (within *wl (cscall *from *newstate))
        (within *wl (statechek consist))!(condc *wl *newstate))      (10)
```

各世界には非線形系の各動作領域における動作を定義する質的方程式がassertされている。具体的なイメージは後の例で示す。mcscall は、リスト *wlで与えられた見ることができる世界のネストにおいて、*from を条件とするpropagation による推論を述語cscallによって行なう。cscallはその世界のネストで見得る質的方程式 cs を順々に試して無矛盾な状態を推論する。結果は*newstate に得られ、さらにこの解に対し、上記と同じ世界のネストにおいて無矛盾性のチェックを行なう（述語statechek ）。その上でその動作領域で妥当な結果かどうか、を述語condc によってテストする。condc が成功すれば*newstate は妥当な結果と見なされる。ここで失敗すると！の効果でmcscall 自体が失敗する。condc は問題毎に独自のものが作成される。さて以上の説明を基に、非線形系の状態を推論する述語forkについて説明する。forkの第一引数*oldwlは，ともあれ推論を行なってみる世界のネストを表わすリストである。第三引数*from がpropagation の条件であり、第四引数*newstate が推論結果である。第一のclauseにおいて呼び出したmcscall が成功すると、この時点で正しい推論結果が得られる。前記のmcscall の説明からわかるように、mcscall が失敗し、第二のclauseを実行する時には、別の動作領域での推論を行なう必要がある。このため、appendによって、見得る世界のネストを表わすリスト*oldwlにw1をつなげて*newwlを作る。この*newwlを見得る世界のネストのリストとしてmcscall を呼び新たな動作領域での推論を行なう。

以上の説明により非線形系の動作領域の選択がバックトラックによって実現されていることが理解されるであろう。次に以上の説明ではあまり明らかにできなかった各動作領域と多重世界との関係をつぎの例題で説明する。

（ダイオード）

図２の抵抗とダイオードからなる回路を定電圧源で駆動する回路について考えてみる。世界W0にダイオードに逆方向の電圧がかかる場合に対応する次の方程式を定義しておく。

VH = I , I2 = 0 , VR = I1 ,
VH + VR = VS , I1 + I2 = I

一方、世界W1にはダイオードに順方向に電圧がかかった場合に対応する方程式のうち、W0と相違するもののみを定義しておけばよい。すなわち次の定義である。

I2 = I , VR = 0

またこれによりW0のI2 = 0 , VR = I1は見えなくなる。これは世界のネストを(W1 W0) としてwithinを使えば、W0の定義のうちW1と重ならない部分は継承される、というUranusの機能により許される。

さてcondc は次のようになる。

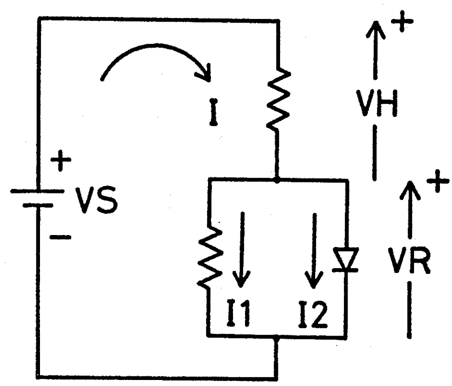

図２．ダイオードと抵抗

```
(assert (condc (w0) *state)
        (if (and (member (VR *X) *state)(= *X +))(fail)(true)))     (11)
```

ダイオードの順方向に電圧をかけた場合の状態は次の質問で推論する。
(fork (w0) *newwl ((VS +)(I +)) *newstate) (12)
質問(12)に対する解は、バックトラックによって、
(within (w1 w0) (cscall ((VS +)(I +)) *newstate) の解であり、次のようになる。
((VS +)(I +)(VH +)(VR 0)(I1 0)(I2 +))

ダイオードに逆方向の電圧をかけた場合の状態は次の質問で推論する。
(fork (w0) *newwl ((VS -)(I -)) *newstate) (13)
この質問の場合は、(within (w0) (cscall ((VS -)(I -)) *newstate) の解が正しい解であり次のようになる。
((VS -)(I -)(VH -)(VR -)(I1 -)(I2 0))

以上の例からUranusの多重世界機構が非線形系の質的推論に役だっていることが理解いただけるであろう。

（参考文献）

[1] J.de Kleer, J.S.Brown, "The origin, form and logic of qualitative physical laws", Proc. of 8th IJCAI pp.1158-1168, 1983
[2] J. De Kleer , D.G.Bobrow, "Qualitaive reasoning with higher-order derivatives" , Proc. of AAAI-84 pp. 86-91, 1984
[3] J. De Kleer , J.S.Brown, "A Qualitative Physics Based on Confluence" , Artficial Intelligence Vol.24 pp.7-83, 1984
[4] Ken Forbus, "Qualitative process theory", MIT AI Lab Memo No. 664, Cambrige, MA, 1982
[5] Ken Forbus, "Qualitative Process theory", Artficial Intelligence Vol.24 pp.85-168, 1984
[6] P.J.Hayes, "The Naive Physics Manifesto" in Expert Systems in the Micro-Electronic Age", edited by D.Michie, Edinburgh University press, May 1979
[7] B. Kuipers, "Commonsense Reasoning about Causality: Deriving Behavior from Structure", Artficial Intelligence Vol.24 , pp.169-203, 1984
[8] D.Mcdermott,"A temporal logic for reasoning about process and plans",Cog.Sci.6,1982
[9] G.L.Steel Jr, "The definition and implementation of a computer programming language based on constraints",MIT AI Lab TR-595, Cambrige , MA, 1980
[10]中島秀之、他、"Prolog/KR からUranusへ－多重世界機構の拡張－"，情報処理学会 知識工学と人工知能研究会36-2, 1984