知識工学と人工知能 51－6
（1987. 3. 12）

動作系列別解析法による定性的回路解析とその実現

外山滋　米澤明憲
東京工業大学理学部

時間経過に伴う対象系の動作を定性的に解析する枠組みとして、従来から定性的シミュレーションという手法がある。この手法によると解析は初期状態から開始し、次々と新状態を生成することにより実行される。状態は全ての変数の値を求めることにより決まるが、定性的な値は数値に比べて抽象的で情報量が少ないので値の曖昧な変数がでてくる。この場合にはあらゆる可能な新状態が作られ、シミュレーションは分岐して続けられる。このため回路などの複雑な解析対象の場合、状態数が爆発的に増大するという問題があった。

本稿ではこの問題を解決するために動作系列別解析法を提案する。動作系列別解析法は（1）状態は確定可能な変数により決定される、（2）独立した動作ごとにシミュレーションを行うという特徴を持つ。

また本稿では動作系列別解析法を使って回路の動的な動作を説明するシステムＱＣＳについて述べる。

The Event-Sequence-Wise Method for Qualitative Circuit Analysis

Shigeru Toyama and Akinori Yonezawa

Department of Information Science, Tokyo Institute of Technology
Oookayama 2-12-1, Meguro-ku, Tokyo 152, Japan

ABSTRACT

To make qualitative reasoning about the dynamic behavior of a target system, simulation-based qualitative analysis has been developed. In this approach, the analysis starts with the initial state and generates new states one after another. A state is determined when every variable is assigned its value. Since a qualitative value is abstract and has less information compared with a numerical value, many variable values cannot often be determined. In such a case, simulation proceeds by hypothesizing every possible new states. This causes the combinatorial explosion of the system states when a complicated system such as circuits is analyzed.

To alleviate this deficiency, we propose "Event-Sequence-wise Method" which has the following characteristics:

(1) A state is determined only through known variable values.

(2) Simulation is performed for each sequence of causally connected events.

We also present a qualitative circuit simulator QCS which is based on Event-Sequence-wise Method and explains the dynamic behavior of electronic circuits.

1.はじめに

我々は数値情報の欠落した回路図を見ても、その回路の動作を定性的に理解することができる。この能力があるために機能の推測や、故障診断が容易になるものと思われる。このような能力を計算機上で実現する枠組みとして定性的推論がある。特に回路に対する定性的推論に関する研究としては EQUAL[De Kleer84]、TQ解析[Williams84]、動的因果関係解析法[西田86]などがあげられる。 EQUALでは、平衡状態にある回路に対して入力に変化を与えたときの伝播状況の解析を行っている。またその解析結果をもとに回路機能の推測を行っている。TQ解析では時間的経過を伴う動作の解析、およびフィードバックの解析を行っている。動的因果関係解析法では不連続な変化の解析を行っている。これらのうちTQ解析と動的因果関係解析法は定性的なシミュレーションを基礎としている。

従来の定性的推論では対象系の状態は全ての変数の値を決定することによって定められた。このため扱う変数の数が高々数個の場合には問題はなかったが、回路の様に複雑な対象は変数の数が多く取り扱いが困難であった。一般に回路の動作モデルは非常に複雑で、素子数が高々数個の回路でも変数の数や回路方程式の数は共に数十のオーダーにのぼる。従来は状態を決定するために様々なヒューリスティックスを使って対処していたが、基本的な解決には至っていない。

筆者はこの問題を回避するために時間と状態の定義をより抽象化し、独立した動作系列を独立にシミュレートする動作系列別解析法を考えた。またこの方法に基づき定性的回路解析システムＱＣＳ(Qualitative Circuit Simulator)を作成した。

2.従来の定性的推論とその問題点

2.1.定性的概念とその表現法

2.1.1.定性値

一般に定性的推論では値は実数軸上の境界点と区間に対応させた定性値を使う。定性値の代表例としては、

$\{x|x\leqq 0\}$, $\{x|x=0\}$, $\{x|x\geqq 0\}$

に対応させた<->,<+>,<0>があげられる。 実数値の間に様々な演算が定義されているのと同様に、定性値の間にも演算が定義される。

例: x + y

		x		
		+	0	-
	+	+	+	?
y	0	+	0	-
	-	?	-	-

x * y

		x		
		+	0	-
	+	+	0	-
y	0	0	0	0
	-	-	0	+

2.1.2.変数

対象系の状態を表す最小単位が変数である。回路の場合は電圧や電流などを変数とする。（ただし本稿の例の中では電流に関する変数は省略した。）また通常の変数の外に、それらの変化傾向（微分値）を表す変数がある。変化傾向変数はもとの変数が、増加、一定、減少のいずれの傾向にあるかを示している。もとの変数をXとするとき、その変化傾向を ∂Xと表記する。変化傾向の値をもとに、変数の値がどのように変化するかを調べることができる。例えば、

X=<+>, ∂X=<->

であるとき、Xの値は<0>に変化する可能性がある。

2.1.3.対象系の動作モデル

対象系の動作は方程式を使ってモデル化される。ただし定性的推論で扱う方程式は通常の方程式と異なり、正負の値のわかっている係数は除外して表すことができる。例えば V=R*I という式においてRが正の定数ならば、V=I という形に簡略化される。ただしこの場合の等号は定性的に等しいことを意味する。

非線形な式は複数の線形な式で近似される。このときそれぞれの線形な領域を動作領域という。

2.1.4.従来の定性的時間

定性的推論では時間の表現には時点(instant) と時区間(interval)を使う。時点と時区間は時間の最小単位である。従来の定性的推論では時間は時点と時区間が交互に現れる系列として表される。

2.1.5.従来の定性的状態

対象系の状態は時間ごとに決められる。一つの時点あるいは時区間内では、対象系は同じ状態をとるものと考える。状態決定の際には全変数の値を求める必要がある。状態は変数と値の組を要素とする集合で表される。

例: {(A,<+>),(B,<->),・・・}

2.2.定性的シミュレーション

シミュレーションベースの定性的推論では、対象系の動作解析は初期状態から始めて逐次的に新状態を生成することによって行われる。一般に定性的情報は数値に比べて抽象的で情報が不足しているので、シミュレーションの際に一意に値を決定できない変数がある。例えば<+>+<->の値を一意に決定することはできない。このため状態を決定することが困難であることが多い。その場合には複数の遷移可能な状態について解析は分岐して続けられる。また解析の途中で以前と同じ状態が生成された場合はシミュレーションは統合される。解析結果は対象系がとりうるあらゆる可能な状態をもとにした状態遷移図として得られる。

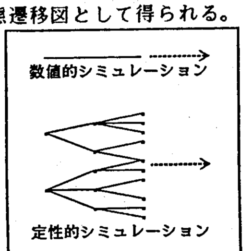


図1

図１に数値的シミュレーションによる解析の進行のし

かたと、定性的シミュレーションによる解析の進行のしかたを示す。この図では点は状態を表す。

2.3.問題点

従来の方式では次の二つの問題点が考えられる。

1）初期状態を一意に決定できない。

シミュレータへの入力情報が不十分であるために、初期状態における全ての変数の値を求めることができない。例えば図２の回路においてVcc>VIN>GNDであるとき、Vn1とVINの大小関係が与えられていないのでVR3の値を求めることができない。この様なときには、VR3=<->,<0>,<+>に対応する三通りの初期状態が作られる。

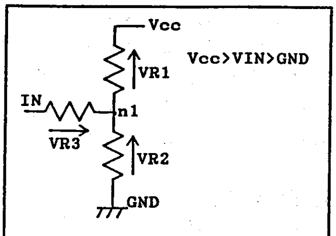
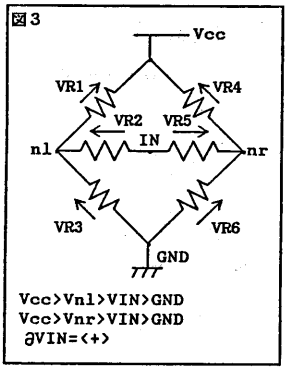


図２

2）状態遷移が一意に決まらない。

複数の独立な変数変化が予想されるとき、これらの時間的順序関係を決めなければ状態が決定できない。例えば図３の回路において、VR1からVR6の値が<+>で、VINが増加傾向にあるときは、 VR2=<0>とVR5=<0>のどちらが先に起きるか決定することができない。従来の方法に従えばVR2=<0>が先に起きる場合、VR5=<0>が先に起きる場合、同時に起きる場合の三通りの状態遷移が考えられる。

図３


上の１、２の問題のために状態の数は変数の数に対して指数関数的に増大する傾向がある。一般に回路が複雑になるに従って変数の数が増加するので、少し複雑な回路では実用的な解析ができない。

このことは単に計算量の問題のみならず、解析結果も意味のないものにする。解析結果の意味については、「全ての可能な状態を予測することが定性的推論の長所の一つである」と従来から言われている。確かに予測数が高々数通りに抑えられれば良いが、現実には回路が複雑になるにつれて予測数が指数関数的に増大する。

3.動作系列別解析法

以上の問題を回避するために次の二つの方針を考えた。

1）与えられた情報だけから導くことのできる変数の値のみを求める。

2）独立な動作系列は独立にシミュレートする。

これらの方針を実現するために動作系列別解析法では従来とは異なる時間と状態の定義をする。

3.1.定性的時間

従来の定性的時間は時点と時区間を要素とする<u>全順序集合</u>で表された。しかし動作系列別解析法では、時点と時区間を要素とする<u>半順序集合</u>で表される。ただしこの順序関係には次の二つの制約がある。

制約１　$p_1<p_2$なる任意の時点p_1, p_2に対し、$p_1<i<p_2$なる時区間iが存在する。

制約２　$i_1<i_2$なる任意の時区間i_1, i_2に対し、$i_1<p<i_2$なる時点pが存在する。

3.2.定性的状態

定性的推論では状態は時点あるいは時区間ごとに決定される。状態は変数と値の組の集合として表される。ただし従来の定性的推論ではその際に全ての変数の値が決定されている必要があった。しかし動作系列別解析法では確定可能な変数だけで状態は決定される。

3.3.動作系列別解析法の実行手順

動作系列別解析法ではキューが一つ使われる。このキューには、状態を決定するための必要最低限の情報（状態決定最小情報）が要素としていれられる。対象系は方程式を使ってモデル化されているので、一部の変数の値から他の変数の値を求めることができる。状態決定最小情報はこのような一部の変数の値を指定する情報のことで、変数と値の組の集合として与えられる。状態決定最小情報は状態を規定するが、状態そのものではない。一番最初の状態決定最小情報はシミュレータへの入力として与えられなければならない。

ステップ1．［初期設定］

最初の状態決定最小情報をキューに積む。

ステップ2．［終了判定］

キューが空なら終了

ステップ3．［状態解析］

キューから一つ状態決定最小情報を取り出し、各変数の値を求め状態を決定する。ただし一意に値が導ける変数のみを求める。各変数の値を求める具体的方法は動作系列別解析法を使う個々の解析システムに依存

する。
ステップ4. ［状態の重複のチェック］
　新状態と同一の状態がすでに記録の中に存在するか調べる。もし存在していれば状態の統合をする。

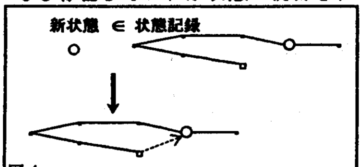


図4

状態が統合された場合はステップ2にいく。
ステップ5. ［解析の分岐］
　ここではまずどの変数の値が変化する可能性があるか調べる。次に関連をもって値が変化する変数どうしをグループ化する。例えば変数A,B間にA=Bという関係があるとき、これらは同じ変数グループに属する。一つの変数グループが一つの独立した動作系列の始まりを意味している。最後に変数グループごとに状態決定最小情報を作る。
ステップ6. ［状態の記録］
　現在の状態の記録をとりステップ2に進む。

3.4.動作系列別解析法と従来の定性的シミュレーションとの比較

・図１の例の場合

　従来の方法では全ての変数を決定しなくてはならないので、初期状態として三通りの解釈があった。これに対して新方法では確定可能な変数だけを使って状態を表せばよいので、初期状態は常に一通りに決まる。この例の場合は

{(VR1,<+>),(VR2,<+>)}

が初期状態となる。

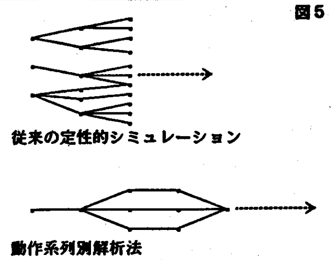
・図２の例の場合

　この例の場合、VR2=<0>とVR5=<0>のどちらが先に起きるか予測できない。従来の方法ではこの順番を仮定しなければ状態が決まらなかったため三通りの可能性があった。これに対して新方法では左右状態遷移を次の様に別々に表すことができる。

左：｛(VR1,<+>),(VR2,<0>),(VR3,<+>),
　　 (VR4,<+>),　　　　　(VR6,<+>)｝
右：｛(VR1,<+>),　　　　　(VR3,<+>),
　　 (VR4,<+>),(VR5,<0>),(VR6,<+>)｝

・シミュレーションの進行過程の比較

　図５に従来の定性的シミュレーション法と動作系列別解析法との違いを示す。図において点は状態を表す。従来の方法では初期状態として複数の可能性が存在するが、新方法では常に一つしか存在しない。また従来の方法ではあるゆる可能性に対して分岐が行われるが、新方法では独立した動作ごとに分岐する。

図5


・状態遷移における分岐の数

　独立な動作系列がｎ個あるとき、従来の方法では2^n-1個の新状態が生成された。これはｎ個の部分系の状態について、現在の状態に留まるか、新しい状態に遷移するかという組合せを表している。（ただし現在と同一の状態は新状態には含まれないので１つ減る。）
　これに対し新方法では独立した動作ごとに分岐するので、新状態はｎ個しか生成されない。

4.定性的回路解析システムＱＣＳ

　筆者は動作系列別解析法をもとにした回路解析システムＱＣＳを作成した。ＱＣＳには入力として回路の構造記述と回路への入力条件が与えられ、出力として回路の動的な動作を説明する。ＱＣＳでは動作系列別解析法のステップ３の状態解析部で回路の性質を利用した解析手段がとられている。現在のところＱＣＳは抵抗、ダイオード、トランジスタよりなる回路を取り扱うことができる。回路への入力は一節点に限られる。システムはモデル変換部、定性的シミュレータ、動作説明生成部で構成される。プログラムはCommon Lispで記述され約5200行になる。

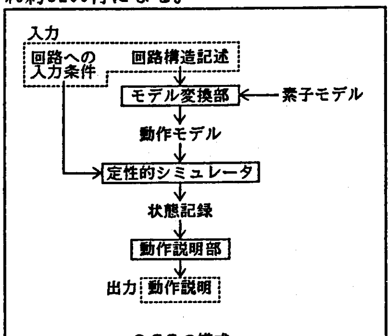


図6　ＱＣＳの構成

4.1.モデル変換部

　モデル変換部では回路構造記述を動作モデルに変換する。回路の構造記述は、回路に現れる節点と素子の一覧、及び各素子の端子がどの節点に接続されているかという記述よりなる。しかし構造記述はそのままでは回路の動作を直接に表現していないので、動作モデルに変換する必要がある。ＱＣＳでは動作モデルとして回路方程式と電流方向モデルの二種類の表現法を使った。このうち電流方向モデルは回路内の各素子に流

れうる電流の方向を示したものである。例えば抵抗の場合は両方向に流れ、ダイオードの場合はカソードからアノードに流れることが示される。方程式は多様な対象系の動作を記述するのに適しているが、回路の特殊な性質を反映した表現法ではない。これに対しＱＣＳで使った電流方向モデルは一つ一つの素子の回路全体に対する性質を表したものなので推論が軽減される。この様なモデル化はDe Kleerの示したNo-function-in-structure[De Kleer84]に反するが、実用的なシステムを目指す上では今後必要となってくるであろう。

モデル変換に際しては素子の特性を示した素子モデルが使われる。素子モデルは素子の種類ごとに必要とされる。素子モデルの記述は、素子の端子のリスト、使用される変数の宣言部、動作領域、素子の特性式、変数の制約、電流方向モデルより構成される。ＱＣＳで実際に使われる動作モデルの例を付録1で示す。

4.2.定性的シミュレータ

回路の動作モデルと入力条件をもとに、時間経過に伴う回路の動的な動作を解析する。入力条件は入力節点変数名とその初期値及び最終値で与えられる。ただし解析中は入力変数は単調かつ連続に変化するものとする。解析手順は三章で述べた動作系列別解析法に準ずる。動作系列別解析法ではステップ３の状態解析の具体的方法は解析システムに依存するので、ここではその具体的実現法について述べる。

4.2.1.状態解析

状態解析の目的は状態決定最小情報をもとに対象系の動作モデルを使って各変数の値を求めることである。変数には電圧や電流を表す通常の変数と、それらの変化傾向を表す変化傾向変数とがある。ＱＣＳの状態解析は、通常変数解析と変化傾向解析の二つに分けられる。

4.2.1.1.通常変数解析

通常変数解析では回路に流れうる全ての電流の経路が探索され、それをもとに各変数の値が求められる。経路の探索は電圧の高い節点から低い節点に向かって行われる。探索は電流方向モデルで指定された方向にのみ行われる。抵抗やトランジスタには極性があるので、実際には探索は非常に簡単に行うことができる。

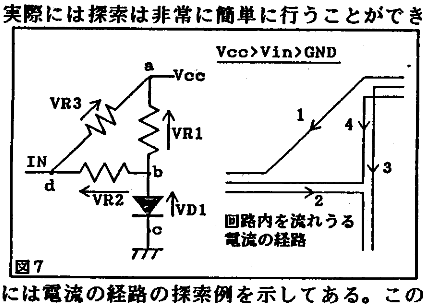


図7

図７には電流の経路の探索例を示してある。この例では節点ｄと節点ｂの電圧の大小関係が与えられていないので、ｄｂ間に流れうる電流として二通りの可能性がある。このためｄｂ間に関係する変数の値は求めることができない。この例では状態は次のようになる。

{(VR1,<+>),(VR3,<+>),(VD1,<+>)}

4.2.1.2.変化傾向解析

平衡状態にある回路に対して入力に変化を与えると、この変化は物理的に隣接する節点や素子を通して次々に伝播していく。変化傾向解析ではこの現象に対応させて、入力変数の変化が方程式を介して次々に伝播していくものと考える。（同一の方程式に属する変数どうしは隣接しているものと考える。）この考え方はCausal Analysis[De Kleer84]と同じである。

しかし三個以上の変数を含む方程式では、入力変化の波及状況を推論することが困難になる。例えばA=B+Cという式からは、Aの変化が他の二つの変数にそれぞれどの程度の影響を及ぼすのか推論することはできない。ＱＣＳではこれを解決するために、方程式の種類（KVL、KCL、素子特性式）や、変数の種類（電圧、電流）を手がかりにしたヒューリスティックスを用いている。また変数に特殊な性質がある場合、変数の制約として素子モデルに記述されているので解析の際に利用できる。この様な制約の例としてはAbsorbantがある。Absorbanat制約を受けた変数は、この変数を含む式において他の変数の変化を全て吸収する。

図8

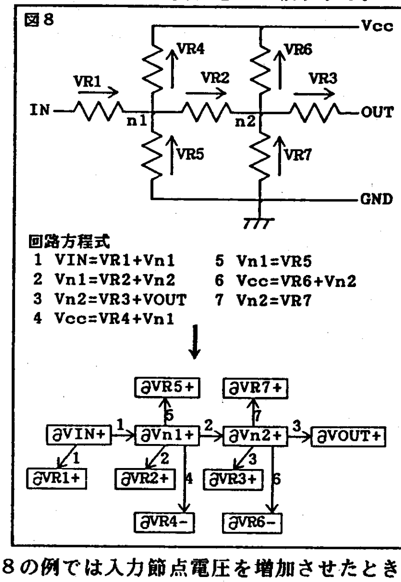


図８の例では入力節点電圧を増加させたときの波及状況が示されている。図中の矢線の番号は方程式を表している。この図では電圧の変化傾向のみが示されているが実際の解析では電流の変化傾向も調べられる。

4.3.動作説明部

定性的シミュレータの結果は状態記録として得られ

るが、状態記録は変数の値の変化の理由を表してはいない。このため動作説明部では状態記録をエピソード[Williams86]に変換し、動作説明をエピソード間の因果関係の集合として出力している。ここでエピソードとは事象の最小単位のことで、時間的に連続して同じ値をとる変数や動作領域に対応している。

図９

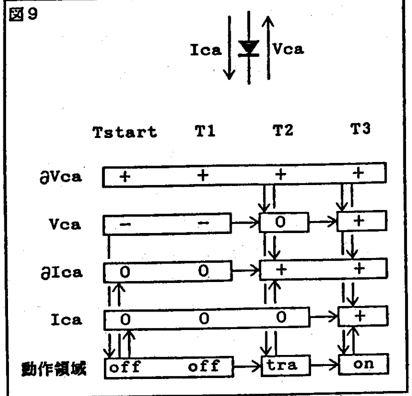


図９に一つのダイオードからなる回路の例を示す。この例では時間Tstartから解析が始まり、四つの時間における状態が示されている。図中の四角い枠はエピソードを表し、エピソード間の矢線は因果関係を表している。ＱＣＳでは矢線の集合を動作説明としている。

筆者は動作説明に際して、曖昧性のある変数は説明する必要がないと考えている。この考え方のもとでは目的を動作説明に限れば、値の曖昧な変数の存在は問題ではなくなる。ＱＣＳでは動作系列別解析法を使って曖昧性を回避した解析を行っているので、動作説明は一意に得ることができる。

4.4.ＱＣＳの今後の課題

ここでは現在のＱＣＳでいまだ実現されていないが、より実用的な回路解析システムを作るために必要であると思われる点について考察を行う。ただしこれらは動作系列別解析法とは関係ない。

4.4.1.複数の入力を扱う場合

ＱＣＳでは入力が一箇所のみである回路しか扱えない。ただし次の様な場合は容易に解決できる。

1）複数の入力節点があっても、変化する入力がそのうち一つしかないとき。

2）独立した動作を示す部分回路に分割した場合、各部分回路に高々一つの入力しか無いとき。

しかし回路内の同じ場所に対して二つの入力が同時に影響を及ぼす場合、現在の枠組みでは解析は困難である。例えば図１０において、IN1の節点電圧を上昇させ、IN2の節点電圧を下降させた場合、OUTの節点電圧の変化は推論できない。これを解決するためにはどちらの入力がより大きい影響力を持つかという情報が必要となる。

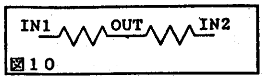


図１０

4.4.2.回路の構造情報を利用したモデル化

ＱＣＳではコンデンサを含む回路は解析できない。理由はいくつかあるが、一つの理由として回路によってコンデンサの使われ方が場合によって異なることがあげられる。我々がコンデンサを含む回路の推論をする場合には、回路の種類によって適切なコンデンサのモデルを選択する。例えば回路のパターンから入力信号のおおよその周波数やコンデンサの容量などを暗黙の内に仮定できる。

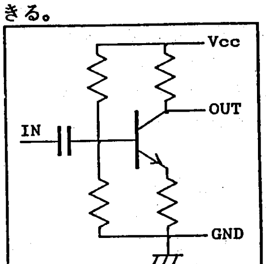


図１１

例えば図１１の例の場合はコンデンサの容量は比較的小さいものと判断できる。このため非常にゆっくりと変化する入力が回路の動作に影響を与えるとは考えない。これに対して定性的推論ではコンデンサの特性を$I=\partial V$という式で与えるので、いかなる電圧の変化も影響を与えるということが推論される。第二の例として図１２の回路の場合人間なら、コンデンサの容量が比較的大きいことが仮定できる。従って入力電圧の変化は無視できない。さらに第三の例として図１３の回路の場合、コンデンサが電荷を蓄えるということが意識される。この場合はコンデンサがあらかじめ充電されているかどうかが問題になる。

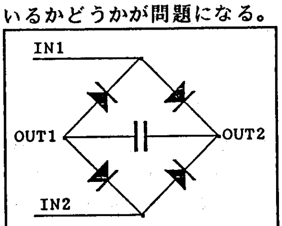


図１２

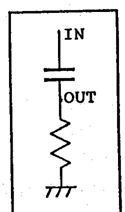


図１３

実際的な解析システムを作る上で、方程式による素子特性の表現は必ずしも適切ではない。様々な回路に対応するためには回路の構造的パターンから最も適切な素子のモデルをあらかじめ選択する必要がある。また場合によってはモデルだけでなく、素子の初期状態も知る必要がある。

4.4.3.ユーザ・インタフェースの拡充

現在のシステムでは回路の動作説明を細部まで全て出力している。しかし回路が複雑になれば、全ての動作を細かく説明することは、おそらくユーザにとって

冗長なものとなろう。これに対処するには説明内容を階層化し、ユーザの質問に応じて様々なレベルの説明ができるようにする必要がある。

5.まとめ

従来の定性的推論では全ての変数の値を求めなければ状態が決定できなかった。また解析の目的は全ての可能な状態を求めることであった。このために少し複雑なシステムになると一意に値を決定できない変数があるので状態数が膨大な数にのぼり解析が困難であった。筆者は従来の状態の考え方と解析目的が回路のように複雑なシステムでは不適当であると考え、これらを共に変更して新たなシミュレーションの枠組みとして動作系列別解析法を考えた。動作系列別解析法では与えられた情報から導くことのできる変数の値だけを求め、独立な動作系列は独立にシミュレートする。このため生成される状態数は実用的な数に抑えることができる。

従来から主張されている「あらゆる可能な状態の予測」は予測数が高々数通りの場合は非常に意味のあるものと思われる。数通りの予測を実現するためには、今後従来のシミュレーション法と動作系列別解析法を融合させる必要があろう。

[参考文献]

[deKleer84] de Kleer,J., How circuits work, Artificial Intelligence 24 (1984) 205-280.

[外山87] 外山滋：回路に対する定性的推論に関する研究、東京工業大学修士論文

[西田86] 西田豊明，川村正，堂下修司：動的因果関係解析法による電子回路の定性的解析，情報処理学会知識工学と人工知能研究会資料48-7 (1986).

[Williams.B.C.84] Williams,B.C., Qualitative analysis of MOS circuits, in: Artificial Intelligence 24 (1984) 281-346.

[Williams.B.C.86] Williams,B.C., Doing Time: Putting Qualitative Reasoning on Firmer Ground, Proceedings of National Conference on Artificial Intelligence, (1986) 105-112.

付録:ＱＣＳによる回路解析例

付録ではインプリメントされたＱＣＳによる解析例を示す。最初に示すのは素子モデルの記述である。その次に Schmidt-trigger回路[De Kleer84]について構造記述と動作説明を示す。

付録1. 素子モデル

ここでは素子モデルの例として特にダイオードを示す。素子モデルの記述はterminal, variable, oregion, formula, vrestriction, flowの六つの部分よりなる。これらの意味を以下に示す。

terminal: 素子の端子。

variable: 素子に関する変数。それぞれの変数には変数の種類と関連する端子を指定する必要がある。

oregion: 動作領域。それぞれの動作領域にはその活性化条件が必要。

formula: 素子特性式。それぞれの素子特性式には、その式が有効となる動作領域を指定する必要がある。（あらゆる動作領域において有効である場合には""としておく。）

vrestriction: 変数の制約。それぞれの制約には、それが有効となる動作領域を指定する必要がある。（あらゆる動作領域において有効である場合には""としておく。）

flow: 電流方向モデル。その素子に流れうる電流の経路を、始点と終点を指定することにより記述する。

```
(def-component "diode"
   (terminal   "anode" "cathode")
   (variable
    ("Ia"    "current"        "anode")
    ("Ic"    "current"        "cathode")
    ("Vac"   "edge-voltage"
                       "anode" "cathode"))
   (oregion
    ("on"               (< "Vac" "0"))
    ("transient"        (= "Vac" "0"))
    ("off"              (> "Vac" "0")))
   (formula
    ((=  "Ia" "Vac")      "on")
    ((= "dIa" "dVac")     "on")
    ((=  "Ia" "0")        "transient")
    ((= "dIa" "dVac")     "transient")
    ((=  "Ia" "0")        "off")
    ((= "dIa" "0")        "off"))
   (vrestriction
    ((absorbant "dVac") "off"))
   (flow
    ("cathode"  "anode")))
```

付録2. Schmidt-triggerの解析例

回路構造記述はnode, componentの二つの部分よりなる。これらの意味を次に示す。

node: 回路で使われる全ての節点が記述される。ただし特別な意味を持つ節点は、そのことを明確に示す必要がある。このような節点としては Vcc, GND, Input, Outputの四つがある。

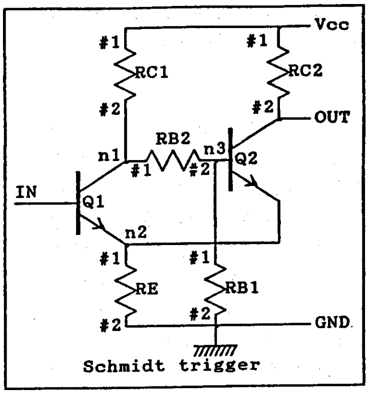
component: 回路で使われる全ての素子が記述される。それぞれの素子に対してその種類を指定し、素子の各端子がどの節点に接続されているかを記述する必要がある。

ＱＣＳに回路の動作説明をさせるためには、

```
(explain
  回路構造記述
  入力変数名  初期電圧  最終電圧)
```

と入力する。ＱＣＳは最初に入力変数の変化について示し、次に初期状態における各変数の値を示し、最後に時間経過に伴う回路の動的な動作の説明を行う。各行の(byの)左側は変数あるいは動作領域に変化が起きたことを示し、右側にはその理由が示してある。

```
(setq schmidt-trigger
  '((node ("nc" "Vcc") ("ng" "GND")
          ("in" "Input") ("out" "Output")
          "n1" "n2" "n3")
    (component
     ("Q1"  "npn-transistor"
      ("base" "in") ("emitter" "n2") ("collector" "n1"))
     ("Q2"  "npn-transistor"
      ("base" "n3") ("emitter" "n2") ("collector" "out"))
     ("RC1" "resistor" ("#1" "nc") ("#2" "n1"))
     ("RC2" "resistor" ("#1" "nc") ("#2" "out"))
     ("RB1" "resistor" ("#1" "n3") ("#2" "ng"))
     ("RB2" "resistor" ("#1" "n1") ("#2" "n3"))
     ("RE"  "resistor" ("#1" "n2") ("#2" "ng")))))
```




```
>(explain schmidt-trigger "Vin" "GND" "Vcc")

Explanation

Input variable is Vin changing from GND to Vcc.

Initial variable-value pairs are
   IcQ2=+        I2RC2=-       IcQ1=0        I2RC1=-       I1RB2=+
   IeQ1=0        IeQ2=-        I1RE=+        IbQ2=+        I1RB1=+
  I2RB2=-        VbeQ1=-       VceQ1=+       IbQ1=0        VbeQ2=+
  VceQ2=+        I1RC1=+       V12RC1=+      I1RC2=+       V12RC2=+
  I2RB1=-        V12RB1=+      V12RB2=+      I2RE=-        V12RE=+

The circuit's dynamic behavior is as follows.
   1.        onQ2 by VbeQ2=+
   2.       offQ1 by VbeQ1=-
   3.    dVbeQ1=+ by {dVin=+ kvl}
   4.     VbeQ1=0 by dVbeQ1=+ last_VbeQ1=-
   5. transientQ1 by VbeQ1=0
   6.   dV12RB2=- by {dVn1=- kvl} {dI1RB2=- resistor-model}
   7.   dV12RB1=- by {dVn3=- kvl} {dI1RB1=- resistor-model}
   8.    dI2RB1=+ by {dI1RB1=- kcl}
   9.   dV12RC2=- by {dI1RC2=- resistor-model}
  10.    dI1RC2=- by {dI2RC2=+ kcl}
  11.   dV12RC1=+ by {dI1RC1=+ resistor-model} {dVn1=- kvl}
  12.    dI1RC1=+ by {dI2RC1=- kcl} {dV12RC1=+ resistor-model}
  13.    dVbeQ2=- by {dVn3=- kvl}
  14.     dIbQ1=+ by transientQ1 {dVbeQ1=+ npn-transistor-model}
  15.    dI2RB2=+ by {dI1RB2=- kcl}
  16.    dI1RB1=- by {dV12RB1=- resistor-model} {dI2RB2=+ kcl}
  17.     dIbQ2=- by onQ2 {dVbeQ2=- npn-transistor-model}
  18.      dVn3=- by {dVn1=- kvl} {dV12RB1=- kvl}
  19.     dIeQ2=+ by {dIeQ1=- kcl} {dIcQ2=- kcl}
  20.     dIeQ1=- by {dIcQ1=+ kcl} {dIeQ2=+ kcl}
  21.    dI1RB2=- by {dV12RB2=- resistor-model} {dIcQ1=+ kcl}
  22.    dI2RC1=- by {dIcQ1=+ kcl} {dI1RC1=+ kcl}
  23.     dIcQ1=+ by transientQ1 {dIeQ1=- kcl} {dIbQ1=+ npn-transistor-model}
  24.      dVn1=- by {dV12RC1=+ kvl} {dVn3=- kvl}
  25.    dI2RC2=+ by {dIcQ2=- kcl}
  26.     dIcQ2=- by onQ2 {dIeQ2=+ kcl} {dIbQ2=- npn-transistor-model}
  27.     dVout=+ by {dV12RC2=- kvl}
  28.      IbQ1=+ by dIbQ1=+ last_IbQ1=0
  29.     VbeQ1=+ by dVbeQ1=+ last_VbeQ1=0
  30.      IeQ1=- by dIeQ1=- last_IeQ1=0
  31.      IcQ1=+ by dIcQ1=+ last_IcQ1=0
  32.        onQ1 by VbeQ1=+
nil
```