

4L-13

演繹的手法による故障診断システム

中田 恒夫、田中 英彦、元岡 達  
(東京大学工学部)

1. はじめに

論理システムの規模増大、複雑化に伴い故障診断の作業は困難なものになりつつある。殊に、従来の故障辞書による診断方式の場合、故障シミュレーションに要する時間や故障辞書の大きさなどの点でVLSIへの適用は非常に難しい。一方で故障辞書を用いずに出カデータから演繹により診断を行なう方法が提唱されている。<sup>[1][2]</sup>

現在我々は、演繹的故障診断手法をプロダクションシステム上で実現する方式について検討を行なっている。今回はその概要について述べる。

2. システムの特徴

従来の故障診断システムでは、回路情報と入出力データ列のみが用いられ効率向上のためには、回路の特性を熟知する人間の経験的知識が不可欠であった。

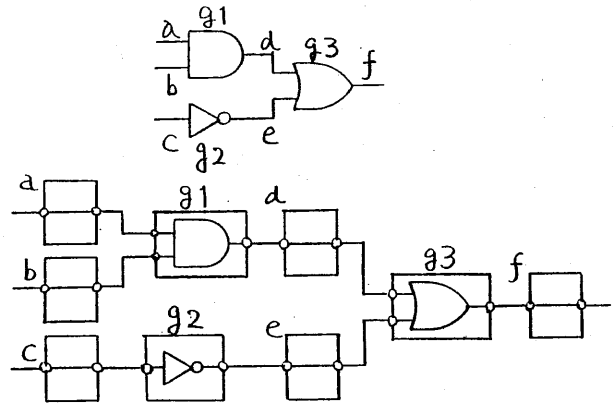
我々はこの経験的知識に頼る部分を可能な限り自動化することを目標としている。そのために従来では取り扱わなかった種々の情報をも診断を進める上で積極的に利用する。扱う情報としては以下のようなものを考えている。

- ▷ 回路構成
  - { 接続情報
  - { 機能情報
- ▷ 入出力データ列
- ▷ FF, メモリの初期値
- ▷ 正常な場合の応答
  - { 出力値
  - { 回路各部の論理値
- ▷ 正常な線・素子
- ▷ プロロービング可能な端子

これらの情報を統一的に扱うためにプロダクションシステムを導入する。種々の情報は事実と規則の形で表わされ、演繹に用いられる。

3. 回路の表現法

回路はいくつかのboxの接続で表現する。各boxは論理素子あるいは素子間を結ぶ線に対応し、それぞれfunction box, connection boxと呼ぶ(ただし線の分岐は前者に対応させる)。また、boxの内部に他のboxを持つことも許し、階層設計との整合性を良くしている。回路とbox表現の対応例を示す。



回路とboxの対応

各boxは、予め与えられた情報、演繹によって得られた知見に従って種々の状態をとる。状態はプロダクションシステムにおける事実の形で表現される。以下に主な状態の例を挙げる。

- (f: function box, c: connection box)
- ▷ 正常/故障
  - { normal (f)
  - { faulty (f)
  - { normal (c)
  - { faulty (c)
    - { sa0 (c) (縮退故障の場合)
    - { sa1 (c)

▷ アローピング可 / 不可

- { accessible (c)
- { inaccessible (c)

▷ 階層レベル：基本素子 / 複合素子

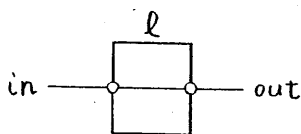
- { primitive (f)
- { compound (f)

この他にも、線が開放状態か否か、正常回路の論理値など、様々な状態が考えられる。

#### 4. 規則

診断プログラムは、診断を行なう人間が与えた情報をもとに事実のAND/ORグラフ表現を生成し、含意式で表わされた規則を次々と適用して演繹を進める。その際に、考え得るすべての状態を考慮した、必要以上に複雑な規則を適用することは効率の点で避けるべきである。すなわち、状況に応じて適用させる規則を変化させるような規則——メタ規則を導入すべきである。以下に簡単な規則の例を示す。

(1) 線 (縮退故障モデル)



$$l.out = 0 \Rightarrow sa0(l) \vee (normal(l) \wedge l.in = 0)$$

$$l.out = 1 \Rightarrow sa1(l) \vee (normal(l) \wedge l.in = 1)$$

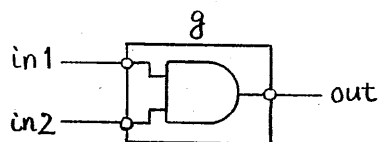
$$normal(l) \Rightarrow l.out = l.in$$

$$normal(l) \Rightarrow l.in = l.out$$

$$sa0(l) \Rightarrow l.out = 0$$

$$sa1(l) \Rightarrow l.out = 1$$

(2) 2入力ANDゲート

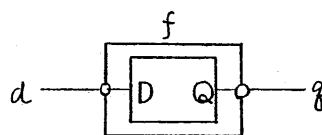


$$g.out = 0 \Rightarrow faulty(g) \vee (normal(g) \wedge g.in1 = 0) \vee (normal(g) \wedge g.in2 = 0)$$

$$g.out = 1 \Rightarrow faulty(g) \vee (normal(g) \wedge g.in1 = 1 \wedge g.in2 = 1)$$

$$normal(g) \Rightarrow g.out = g.in1 \cdot g.in2$$

(3) Dフリップフロップ



$$f.q(t) = a \Rightarrow faulty(f) \vee (normal(f) \wedge f.d(t-1) = a)$$

( $a \in \{0, 1\}$ )

$$normal(f) \Rightarrow f.q(t) = f.d(t-1)$$

(4) 単一縮退

$$sa0(l) \Rightarrow \bigcap_{li \neq l} normal(li)$$

$$sa1(l) \Rightarrow \bigcap_{li \neq l} normal(li)$$

$$(\bigwedge x_i = x_1 \wedge x_2 \wedge \dots \wedge x_n)$$

#### 5. おわりに

種々の回路情報を、事実・規則の形で表現し、アロダクションシステムに適用することによって回路の故障診断を行なうことができる。

今後は、並列処理を念頭に置いた制御戦略の確立を含め、より一層の具体化、精密化を進めて行き Prolog を用いて実装することを考えている。

#### << REFERENCES >>

[1] Abramovici, M. and Breuer, M.A.: "Fault Diagnosis in Synchronous Sequential Circuits Based on Effect-Cause Analysis", IEEE Trans. on Computers, July, 1982

[2] Genesereth, M.: "Diagnosis Using Hierarchical Models", Proc AAAI-82, Pittsburgh, August, 1982