

テスト記述に基づくテスト生成手法

7J-10

中田 恒夫, 田中 英彦, 元岡 達
(東京大学 工学部)

1. はじめに

論理装置の規模の増大・複雑化に伴い、従来から用いられてきたテスト自動生成アルゴリズムは計算量の点で非実用的なものとなりつつある。これに対し知識工学を応用した故障診断/テスト生成システムが、近年いくつか提案されている [1, 2] が、これらのシステムでは知識として回路の構造記述および機能記述のみを用いており、従来のアルゴリズムによる方式に比べて、大きな性能の向上は期待できない。効率化を図るには従来の方式では取り扱わなかった情報を知識として取り入れる必要がある。

ここでは、回路の構造記述・機能記述の他に各素子に対してテスト用に設計者が与えた記述を利用して効率良く故障診断/テスト生成を行う手法について述べる。

2. 故障診断手法

2.1 診断手順

診断を行うには、まず観測値列と期待値列が一致しない外部出力端子から信号線を外部入力端子に向かってさかのぼり故障素子の候補を十分絞り込んだ後、信号線の実際の値を推論によって求め候補を狭めていく。このうち、後半の部分は各素子の動作記述を用い、出力値から入力値を求める操作を繰り返すことになるが、一般には入力値は一意に決まらず自由度があるためバックトラックを必要とし計算量が大きくなる可能性が高い。そこで、ここでは絞り込みにある程度大きい手間をかけて故障素子の候補をなるべく減らし、後半の処理を軽くすることを考える。

2.2 候補の絞り込み

観測値列と期待値列が一致するか否かのみによって各素子の状態を次の3種(色)に分類する。

①白: 素子は正常で、観測値列と期待値列が一致する。

②灰: 素子は正常であるが、必ずしも観測値列と期待値列が一致するとは限らない。

③黒: 故障素子の候補。

故障診断は各素子を黒→灰→白と「白く」していく操作であると考えられる。また、テスト生成は候補を分割し狭めていくことから黒の素子の一部を「白く」し、残りを黒いままにしておくことに対応する。

外部出力における不一致が1箇所の場合の絞り込み手順をまとめると次のようになる。

①不一致の検出された外部出力端子Tを黒くする。

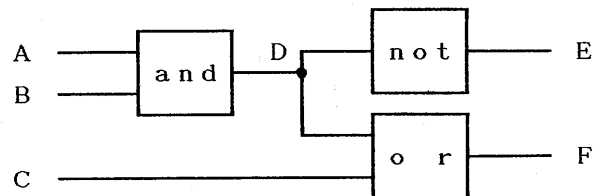
②Tから回路を外部入力端子方向に向かって進み、途中にある素子をすべて黒くする。

③T以外の外部出力端子をすべて白くし、②と同様に外部入力端子に向かって各素子を灰色にする。既に他の色に塗られた素子に出会ったときはそのままにしておく。

④灰色の外部入力端子を白に変え、出力に向かって白くすべき素子を白く塗りかえる。

図1に絞り込みの例を示す。図では故障箇所を信号線に限定し、各ゲートは予め白く塗られているとする。

不一致が複数存在する場合は、それぞれについて絞り込みを行い、その結果を重ね合わせる。各色の強さは白>灰>黒であり、重ね合わせると強い方の色となる。



信号線Eで不一致を検出

①E→黒、②D→黒、A→黒、B→黒

③F→白、C→灰、④C→白

図1 候補の絞り込み例

3. テスト生成手法

3.1 生成手順

故障診断の段階で「黒」とされた素子が複数あるとき、その中から1つを分離するようなテスト入力を求める。対象となる素子の機能が複雑でいくつかの素子の集まりで表される場合は階層レベルを下げて、基本素子のレベルで考える。基本素子としては、各種ゲート、フリップフロップ、レジスタ、RAM、ROMなどがあるが、これらは機能が単純かつ明確であるため機能をテストする手法も簡単に求められる。そこで、これら基本素子のテスト手法は予め人手でテスト記述として与えておき、システム側ではテスト記述を組み合わせ外部入力端子に与えるべき値を決定する。

3.2 テスト記述

各基本素子はデータ入力と制御入力の2種の入力を持ち、制御入力によって素子の動作モードが定められるものとする。テストを行う際に各素子は、①テストされる、②他の素子の出力値を出力側へ伝搬させる、③他の素子の入力値の設定をする、といった3種類の振る舞いのいずれかをする。各素子はこれら3種類の振る舞いそれぞれに適した/適さない動作モードを持つと考えられるため、適した動作モードを設定する制御入力と望ましいデータ入力をテスト用に記述しておくことでテストの効率が良くなると期待される。

3.3 テスト生成例

図2の回路においてM1、M2、M3は4ビットの乗算器、A1、A2は4ビットの加算器である。いま、M1をテストする入力を求めるとする。乗算器、加算器は組合せ回路で動作モードは1つであるためテスト記述は以下のように簡単になる。

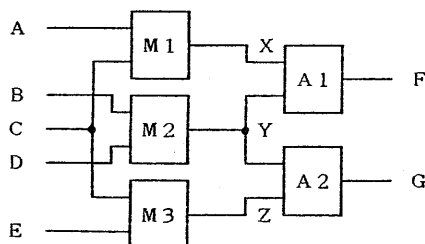


図2 回路例

・乗算器 (入力A, B; 出力S)

①テスト $A = 1, B = p; A = p, B = 1$

②伝搬 $A = 1; B = 1$

③設定 $A = 1, B = v; A = v, B = 1$

・加算器 (入力A, B; 出力S)

①テスト $A = 0, B = p; A = p, B = 0$

②伝搬 $A = 0; B = 0$

③設定 $A = 0, B = v; A = v, B = 0$

p: いくつかのテストバタンの代表

v: 設定すべき値

まず、M1をテストするために乗算器のテスト記述のうちテストの項を見て、 $A = 1, C = p$ とした後、 $X = p$ を伝搬させるためにA1に着目し加算器のテスト記述から $Y = 0, F = p$ を得る。次に $Y = 0$ とするためにM2に着目し同様に $B = 0$ または $D = 0$ が得られる。また、故障候補としてM2が挙がっている場合はM2の出力値を確認するために $Z = 0$ とすべきであり、そこから $E = 0$ を得る。従ってテスト入力としては、 $A = 1, B = 0, E = 0$ として、CにM1をテストするために十分と思われる入力列を加えれば良いことになる。

4. おわりに

故障診断を行う際に、素子を3種類に色分けすることにより早い時期に故障候補を絞り込む手法、およびテスト用の記述を用いてテスト生成を行う手法の概要について述べた。例でも示したように本手法によりテスト生成を行う場合は、論理値として記号値が現れるため記号値を扱える論理シミュレータを開発し実働している。今後はテスト記述の記述法、利用する際の戦略などを定めていく予定である。

参考文献

1. Genesereth: The Use of Design Descriptions in Automated Diagnosis, Artificial Intelligence, Vol. 24, No.1-3, pp.411-436, 1984.
2. Davis: Diagnostic Reasoning Based on Structure and Behavior, Artificial Intelligence, Vol.24, No.1-3, pp.347-410, 1984.