7R-1

テストケース抽出の一方式「順序回路を含む場合」

松尾谷　徹　（日本電気㈱　システム開発本部）

1．はじめに

ソフトウェアの信頼性は，誤り(error) だけの問題では無く、品質特性として「誤り許容性」や「安全性」を含んだ問題と認識されつつある。しかし、ソフトウェア開発過程における「誤り除去」の問題の占める重要性は変わらない。

ソフトウェア開発の現場から眺めると、まだまだ誤り除去過程の質的な増加が必要である。誤り除去の過程をデバグ過程(debuging process)と呼び、その工学的なアプローチをデバグ工学と呼ぶことにする。デバグ工学には色々な問題があり、またソフトウェアの生産性と信頼性に大きな貢献が期待出来る。【文献１】

本稿では、デバグ工学のなかのﾌﾞﾗｯｸﾎﾞｯｸｽ･ﾃｽﾄにおけるテスト項目抽出に関する問題を取り上げる。必要なテスト項目数について整理を行い、従来のテスト項目抽出技法では対象としなかった順序回路の問題を考え一解決方法を提案する。この方法はプログラムの設計過程おいて制御構造に制約を設けることによって実現している。【文献２】

2．ブラックボックステストのテスト項目数

ﾌﾞﾗｯｸﾎﾞｯｸｽ･ﾃｽﾄとは、一般にプログラムの機能仕様書や外部仕様書からテスト項目を抽出するテスト技法である。図1 に示すように、どの様な内部設計が行われたとしても、入出力の関係から正しさを検証しようとするものである。入力をI1,I2,‥‥,Ij とし、入力の取り得る値の範囲を2^nとする。そうする何の制約も存在しないﾌﾞﾗｯｸﾎﾞｯｸｽ･ﾃｽﾄのテスト項目数Ta は、次式(1) で与えられる。

$$Ta = 2^n \times 2^n \times \cdots \times 2^n = 2^{jn} \qquad (1)$$

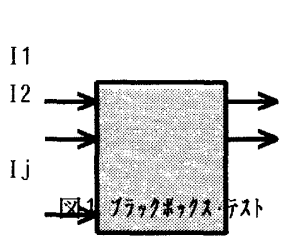

図1 ﾌﾞﾗｯｸﾎﾞｯｸｽ･ﾃｽﾄ

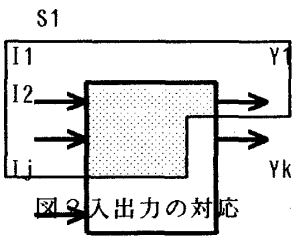

図２入出力の対応

A Derivation Method for Test Case
Toru Matsuodani　NEC,Corporation

式(1) のTa はとんでもない数で、これではテストにならない、そこで2^nの部分に着目し、同値分析と呼ばれる方法でテスト項目数の減少を行う。同値分析とはAda の型のような概念で、たとえば入力I1の取り得る範囲を{1,2,3} とすると、{1,2,3, それ以外} を考える。このようにIjの入力に対する同値の数をmj個とする。　同値分析を導入したときのテスト項目数をTb とすると、式(2) で与えられる。

$$Tb = m1 \times m2 \times \cdots \times mj \qquad (2)$$

式(2) の場合でも、組み合わせについてテストを行うには多すぎるテスト項目である。そこで、出力と入力の関係に着目し、図２に示すような出力Y1と関係のある入力の部分集合S1{I1,I2, ‥‥Is1}を考える。この場合の部分集合S1におけるテスト項目Ts1は式(2) と同様な式(3) となる。

$$Ts1 = m1 \times m2 \times \cdots \times mS1 \qquad (3)$$

ブラックボックス全体のテスト項目をTc とすると

$$Tc = Ts1 + TS2 + \cdots + TSk \qquad (4)$$

次の段階は、入力の部分集合Skと出力Ykとの対応関係についてまで入り込む事になる。つまり出力Ykと入力の部分集合Skの要素との関数関係を定義する。

$$Yk = B\ (Sk) \qquad (5)$$

関数B()はブール代数が用いられ、ブール代数としての関係を求めるため、「原因－結果グラフ技法」等により、入力と出力の関係を分析する。ブール関数B()からテスト項目を求める方法として、２つの方法がある。一つはorの場合、それぞれの積を用いる方法であり、もう一つは前回提案した「原因流れ図」を用い和とする方法である。【文献２】　それぞれをテスト項目数をTd,Te とし抽出関数をρとσとすると。

$$Td = \rho\ (\Sigma B\ (Sk)) \qquad (6)$$

$$Te = \sigma\ (\Sigma B\ (Sk)) \qquad (7)$$

$$Te \leqq Td \leqq Tc \leqq Tb \leqq Ta \qquad (8)$$

これまでの説明から不等式(8) が成立する。この不等式の差は、誤配線に対する検証の除外と、設計の自由度を制約によって成り立っている。

3．順序回路の問題

以上は、入力の集合S が与えられれば、出力Y1‥‥Ykが決まると言う暗黙の前提のもとで進めてきた。しかし現実のプログラムを考えると、何等かの内部変数が存在し、その値（状態）が出力に影響を与える。

つまり図３に示すように、入力の集合S のある時点t における値をS(t)とすると、出力Y1‥‥Ykは一意に

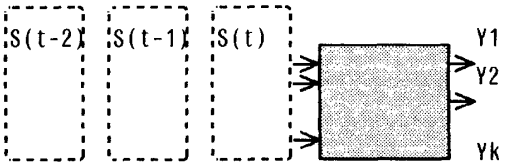

図３　順序回路の問題

決まらず、それ以前の入力の集合、S(t-1),S(t-2),‥‥の影響を受ける事になる。

式(3) のＴS1を例にすると、そのテスト項目は、

$$TS1 = TS1(0) \times TS1(1) \times \cdots \times TS1(t) \quad (9)$$

式(9) は大きな数になってしまう、一つの解決方法としては、順序的な変化に対しても、同値分析を用い順序回路に相当する内部状態変数の変化を、あたかも入力の状態として取り扱う方法である。しかし、この方法では限界があり、適用出来る範囲は限られる。

ソフトウェアにおける順序回路を単純化すると、図４に示すような構成が考えられる。ﾌﾞﾗｯｸﾎﾞｯｸｽ･ﾃｽﾄにおける内部状態変数は、外部仕様書によって初期化と値の設定の二つの条件について定義される必要がある。

内部状態はそれを初期化するﾀｲﾐﾝｸﾞとその値、および設定する入力SI1 ‥‥SIj とその組合せ条件について検証を行う事になる。内部状態は別の組合せ回路の入力となり間接的な出力Y を通して観測することになる。これでは、L2の条件と積になるので式(9) のような積の組合せになる。一つの解決案は、図５に示すような、内部状態を取り出すZ1と、強制的に内部状態を設定する入力D1を付加する方法である。ハードウェアのパッケージテスターは、同様な考え方に基ずき回路に探針を挿入することによって実現している。

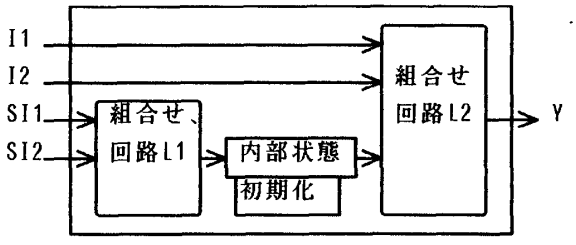

図４　順序回路の等価図

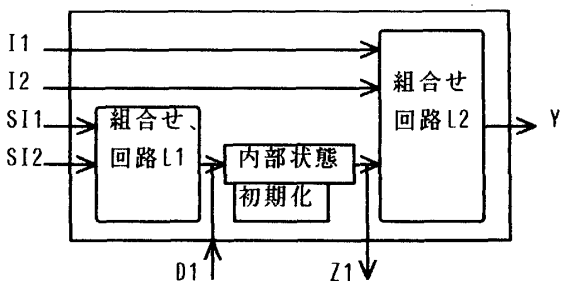

図５　内部状態の分離

ソフトウェアでこの機能に対応するのはデバガーのスナップショットダンプ（Z1を観測）や変数の強制変更（D1を設定）である。ソフトウェアにおいて順序回路のテストを行う為には次の準備が必要。

①機能・外部仕様で内部状態に対する明確な定義。

何をするか、を記述する部分とは別に、入出力と内部状態を明確に定義する工程を付加する。この段階でテスト項目の設計が外部仕様を包含させる。

②設計段階における制約の設定

制御構造、データ構造上、内部状態を分離する。

③ソフトウェア探針の実現

設計上分離された、内部状態を設定／確認するためのテスト支援機能の付加。

④探針に対するテスト項目設計

探針によって検証出来る範囲を分離する。

ﾌﾞﾗｯｸﾎﾞｯｸｽ･ﾃｽﾄは入力に乱数を与え、出力が正しい事を検証するには、入力数が多すぎる。テスト項目を減らすにはこの様な制約が必要となる。

4　まとめ

ソフトウェア開発過程に制約を課して、完全な検証を行う方法を検討している。今回は順序回路を対象とし、制約として必要な探針等の付加機能について提案した。今後、その具体的な実現方法について検討をおこなう課題が有る。

参　考　文　献

1)藤野・松尾谷”デバグ工学における精度配分問題”
平成元年電気・情報関連学会連合大会　1989年9 月
2)松尾谷”テストケース抽出の一方式　原因流れ図”
情処学会37回全国大会　1988年後期
3)松尾谷・赤沢”原因流れ図によるﾃｽﾄｹｰｽ抽出技法”
情処学会38回全国大会　1989年前期
4)平井・真野”実験計画法を用いたﾃｽﾄ 項目設計技法問題点とその改善”日科技連・88　1988年9 月
5)織田・他”原因－結果ｸﾞﾗﾌ 技法を用いたﾚﾋﾞｭｰ支援ﾂｰﾙSVA”　情処学会37回全国大会　1988年9 月
6)松尾谷・真野他”ソフトウェア検査モデル1 ～3”
情処学会29回全国大会　1984年後期