7J−10　画面インタフェースをもつ会話型ソフトウェアの試験方式の検討

高木浩則　橋本辰範
NTTソフトウェア研究所

1．はじめに

画面インタフェースをもつ会話型ソフトウェアは、プロトタイピング手法を用いて開発されることが多く、動作確認のための試験が繰り返し必要となる。この試験では入力・出力確認という手順を試験オペレータが繰り返し行なう。このため、試験実施に費やす時間・費用と共に試験オペレータの労力も増大する。さらに、試験オペレータに対する負荷が増すと、入力・出力確認を誤る可能性が高まるといった、試験品質面での問題も生じる。したがって、画面インタフェースをもつ会話型ソフトウェアの試験では、試験オペレータが試験実施に費やす労力を極力削減することが重要である。

本稿では、画面インタフェースをもつ会話型ソフトウェア（以下、試験システムと呼ぶ）のブラックボックス試験を対象に、試験実施工数の少ない試験系列を生成する手法を提案する。

2．試験システムのモデル化と前提条件

2．1　試験システムのモデル化

（1）試験システムの基本動作を、(a)オペレータからの入力が発生すると、(b)システムの内部状態によって画面に対して何か出力を行ない、(c)内部状態を変化させ、オペレータからの入力待ちになると考え、以下の6項組で表される有限状態機械（FSM）でモデル化する。

FSM=＜Q，I，O，f，g，q＞
ただし　Q：状態集合　q：初期状態
O：出力集合　I：入力集合
f：遷移関数（=Q×I→Q）
g：出力関数（=Q×I→O）

（2）入力と出力各々に試験オペレータが入力に費やす手間、出力確認に費やす手間を表すコストを対応づける。このコストを試験実施工数の最適性評価の材料とする。

2．2　試験系列生成のための前提条件

上記のモデルは、通信プロトコルの動作モデルと類似のモデルとなる。本稿で提案する試験系列生成手法では、状態確認系列として、通信システムの動作試験において提案されているU法[1]により生成されるUIO系列（Unique Input / Output Sequence）を用いる。よって、対象とするFSMは、U法が適用可能な条件である、最小（冗長な状態を含まない）、強連結（任意の2つの状態間を遷移するための系列が存在する）を満たすものとする。

Test method for systems with interactive and graphical user interface

Hironori TAKAKI, Tatsunori HASHIMOTO
NTT Software Laboratories

3．検討対象

上記のモデル上で、全ての遷移の遷移先状態を確認し、かつコストの総和が極力少ない試験系列の生成手法を検討する。

4．提案する手法

4．1　提案する手法の特徴

U法で生成される1つの試験系列は、状態設定系列、試験対象遷移、及び遷移先状態確認系列から構成される。試験対象遷移と遷移先状態確認系列をひとまとめにしたものを仮に部分試験系列と呼ぶことにすると、文献[2]で提案されているU法の改良版は、各遷移に対する部分試験系列をつなぐことにより状態設定系列を極力省くものである。本稿で採用した手法は、各遷移に対する部分試験系列を重ねてつなぐことにより、状態設定系列、遷移先状態確認系列を極力省くものであり、各部分試験系列間を遷移する最短コストの系列を導出し、全ての部分試験系列を通過する最小コストの経路を求めるという手順を踏んで試験系列を導出することに特徴がある。

4．2　試験系列を求める手順

FSMをグラフ表現した状態遷移図において、状態集合 $S=\{S_0, S_1,\ldots\}$、遷移集合 $T=\{T_0, T_1,\ldots\}$ とする。ここでS_0は初期状態である。また$T_r=(S_i, S_j, W_r)$であり、状態S_iから状態S_jへのコストW_rを持つ遷移である。W_rはその遷移における入力・出力確認に対応づけられたコストの和である。また、状態と遷移が交互に表れる1本の系列を遷移系列と呼ぶ。以下で定義する関数を用いて（1）～（5）の手順により状態遷移図からノード（遷移系列を付加情報として持つ）とアーク（出発ノード、到着ノード、コスト、遷移系列を付加情報として持つ）からなる新たなグラフを作成しその上で試験系列を求める。以下で定義する関数において、X_i、X_jは遷移系列、S_m、S_nは状態、N_p、N_qはノード、A_kはアークを表すものとする。

START(X_i)：X_iが始まる先頭の状態
FINISH(X_i)：X_iが到達する最後の状態
COST(X_i)：X_iの各遷移のコストの合計
OVERLAP(X_i,X_j)：X_iとX_jが重なる部分の遷移系列
TAIL(X_i,X_j)：X_jからX_iと重なる部分を除いた遷移系列
CONS(X_i,X_j)：X_iとX_jを連結した遷移系列
MIN(S_m,S_n)：S_mからS_nへの最短コストの遷移系列
ARC(N_p,N_q)：N_pからN_qへのアーク
N_p.seq：ノードN_pに割り当てられた遷移系列
A_k.seq：アークA_kに付加された遷移系列

（関数定義終了）

以下では、N：ノード集合、A：アーク集合、r：リセット系列、S0：初期状態とする。

手順

（1）ノード割り当て

各状態に対するUIO系列を求め、各遷移に遷移先状態確認系列を付加した遷移系列を1つのノードに割り当てる。

（2）初期ノード選択

START(n.seq)が初期状態S0であるノードの1つを初期ノードとする。該当するノードがなければ初期状態のみを含むノードを作成し、それを初期ノードとする。

（3）アーク付加

```
let A := φ
for n1∈N
  for n2∈N and n1≠n2 {
    if OVERLAP(n1.seq, n2.seq) = φ
      then a := CONS(MIN(FINISH(n1.seq), START(n2.seq)), n2.seq)
      else  a := TAIL(n1.seq, n2.seq)
      A := A∪{(n1, n2, COST(a), a)}
  }
```

（4）アーク置換

付加したアークより、リセット系列を考慮した系列の方がコストが小さければ置換する。

```
for n1∈N
  for n2∈N and n1≠n2 and START(n2.seq) = S0 {
    a := CONS(r, n2.seq)
    if COST(a) < COST(ARC(n1, n2).seq)
      then A := (A - {ARC(n1, n2)}) ∪{(n1, n2, COST(a), a)}
  }
```

さらに、Floyd法を利用して各ノード間の最短コストの経路を求めると同時にアークを置換する。

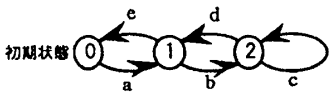
```
for n1∈N
  A := A∪{ARC(n1, n1, 0, )}
for n3∈N
  for n1∈N
    for n2∈N {
      a := CONS(ARC(n1, n3).seq, ARC(n3, n2).seq)
      if COST(a) < COST(ARC(n1, n2).seq)
        then A := (A - {ARC(n1, n2)}) ∪{(n1, n2, COST(a), a)}
    }
```

（5）以上の手続きを取ると、初期ノードから出発し、全てのノードを最短のコストで通過する経路を求める問題になる。この問題の最適解、或は近似解を何等かの方法で求め、初めに初期ノードに付加されている遷移系列を書き出し、初期ノードから全てのノードを通過する際に、アークに付加されている遷移系列を順次書き出すという手順を踏み、試験系列を求める。

4．3　代替系列

提案した手法により求めた試験系列は、重ねてつなぐために、試験系列1本あたりに複数の試験対象遷移が含まれる場合が多い。そのため提案した手法は、基本的には試験系列1本あたりに1つの試験対象遷移が含まれるU法と比較して、工数が少なくて済むという利点を持つ反面、途中でエラーが発生すると、それ以降の試験系列の実施継続が不可能であるという欠点を持つ。そこで、この欠点を補うために、初期状態から各状態への最短経路をDijkstra法により求めて、エラーが検出された際の代替系列とする。これによって、エラーが検出された際の試験系列実施継続の可能性はU法と同等である。

5．試験系列のコスト比較



コスト　a：b：c：d：e＝3：1：2：1：3
（リセット系列rのコストR＞aのコストの2倍）とする

図1　状態遷移図

図1で表現されるFSMに対してU法、U法の改良版、及び提案した手法（ただし代替系列は省略）を適用した結果を表1に示す。

表1　適用結果

試験系列生成手法	試験系列	コスト合計
U法（文献[1]）	rabcd, rabdb, raea	3R + 22
U法の改良版（文献[2]）	rabcdeabcde	R + 20
提案した手法	rabcdea	R + 13

この例では提案した手法は、全ての遷移の遷移先状態を確認するという基準を満たす、さらにコストの小さい試験系列を生成することが分かる。

6．おわりに

画面インタフェースを持つ会話型ソフトウェアの動作試験用の、全ての遷移の遷移先状態を確認するという基準を満たす試験系列生成において、文献[2]で述べられているように試験系列をつなぐだけでなく、重ねてつなぐ手法を採用し、試験実施コストの小さい試験系列の生成手法を提案した。また、エラー検出の際の代替系列を用意した。

本手法は、試験システムの機能を変更・拡張した部分を狙った試験、或は機能ごとの試験をする際に、その部分以外を出来るだけ通過することの少ない試験系列の生成にも適用可能である。

なお、手順（1）でのUIO系列の選択方法、手順（2）での初期ノードの選択方法、及び手順（5）での最適解・近似解の求め方は今後の検討課題である。

参考文献

[1] K.Sabnani and A.Dahbura, "A Protocol Test Generation Procedure," Computer Networks and ISDN Systems, vol.15, pp.285-297, 1988.

[2] A.V.Aho et al., "An Optimization Technique for Protocol Conformance Test Generation Based on UIO Sequences and Rural Chinese Postman Tours," Protocol Specification, Testing, and Verification VIII, pp.75-86, 1988.