

状態遷移表の MC/DC100%テストをやってみよう！

松本 充広[†]

Modified condition/decision coverage (MC/DC)は、ソフトウェアのテスト網羅度として非常に有効な指標である。しかし、一般のソースコードに対し、MC/DC100%テストを生成しようとすると、入力値から構成される膨大な空間内の探索が必要になり、大規模な一般のソースコードへの対応は困難、という問題がある。本稿では、状態遷移表から生成されたソースコードを対象に、状態遷移表の特徴を活かし、ユーザに初期状態からテスト状態へ遷移させる遷移シーケンスを与えてもらうことで、この問題を解決する。本稿では、状態を持つシステムへの網羅的なテストを検討している方を対象に、エアコンモデルを例題として、状態遷移表の MC/DC100% テストをどのように構成するのか、についてと、MC/DC100%テストをどのように実行するのか、について説明する。

Let's try MC/DC 100% test of a STM

MICHIHIRO MATSUMOTO[†]

Modified condition/decision coverage (MC/DC) is one of the most useful test coverage. For general large source codes, it is difficult to generate MC/DC100% tests. In this article, we propose a generation method of MC/DC 100% test for source codes generated from state transition matrices (STMs). The method makes use of the features of STMs. In this article, we explain how to construct a MC/DC 100% test of a STM and how to execute the test by using an air-conditioner model.

1. はじめに

Modified condition/decision coverage (MC/DC)は、条件／判定の確認はもちろん、ソースコードを記述する際、しばしば間違いが起きる A or B と A and B の記述ミスも検出でき、かつ、条件の数を n 個とすると、テストケースの数が 2 の n 乗個にならない、現実的なテスト網羅度である。

MC/DC100%テストでは、確認すべき条件／判定の値、及び、“確認すべき条件／判定を含む条件判定文 (if 文) に到達すること”という制約からテストケースを生成する。

一般のソースコードに対し、“確認すべき条件／判定を含む条件判定文 (if 文) に到達すること”という制約を満たすテストケースを生成するには、入力値から構成される膨大な空間内の探索が必要になる。このため、大規模な一般のソースコードに対しては、MC/DC100%テストの生成は非常に困難である、という問題がある。

状態遷移表、及び、状態遷移表から生成されたソースコードには、条件／判定を含む条件判定文 (if 文) がテストする状態 (テスト状態) の処理先頭にあるという特徴があり、“確認すべき条件／判定を含む条件判定文 (if 文) に到達すること”という制約は、“初期状態からテスト状態に遷移すること”という制約になる。状態遷移表の中身について熟知しているユーザであれば、“初期状態からテスト状態に遷移すること”を満たすシーケンス (遷移シーケンス) を容易に見つけ出せると考えられ、ユーザに遷移シーケンスを与えてもらうことで、一般のソースコードで必要であった膨大な空間内の探索が不要になる。このため、大規模な状態遷移表から生成されたソースコードに対しても、MC/DC100%テストの生成が比較的容易になる。

本稿は、状態を持つシステムへの網羅的なテストを検討している方を対象に、状態遷移表の MC/DC100% テストをどのように構成するのか、についてと、MC/DC100%テストをどのように実行するのか、について説明することを目的とする。

[†]キャッツ組込みソフトウェア研究所, CATS Embedded Software Laboratory

2. 背景

テスト網羅度は、ソフトウェアテストにおいて、十分にテストした、という品質基準を定義する指標の一つである。テスト網羅度を用いることで、ソフトウェア品質を保持しながら、コストや時間が無制限に増大するのを避けることができる。

MC/DCは、故障が人命に係わる航空機ソフトウェアのテスト時に使用するテスト網羅度として、米国の航空機産業の業界団体であるRTCAが策定したDO-178Bの中で開発された。MC/DC開発以前に存在した全てのテスト網羅度は、以下のどれかの問題を持っていたため、RTCAは、これらのテスト網羅度は、故障が人命に係わる航空機ソフトウェアのテスト時に使用するテスト網羅度として不十分、と判断したからである[1]。

1. 全ての分岐を確認していない可能性がある。
2. 全ての条件を確認していない可能性がある。
3. ソースコードを記述する際、しばしば間違いが起ころA or BとA and Bの記述ミスを検出できない可能性がある。
4. 条件の数をn個とすると、テストケースの数が2のn乗個になる。

これらの問題を全て解決したテスト網羅度がMC/DCである。

近年、信頼性向上を求める動きが高まり、車載ソフトウェアなどの航空機ソフトウェア以外の組み込みソフトウェアでも、MC/DCが注目されている。

MC/DCでは、確認すべき条件/判定の値、及び、“確認すべき条件/判定を含む条件判定文(if文)に到達すること”という制約からテストケースを生成する。

一般のソースコードに対し、“確認すべき条件/判定を含む条件判定文(if文)に到達すること”という制約を満たすテストケースを生成するには、入力値から構成される膨大な空間内の探索が必要になる。このため、大規模な一般のソースコードに対しては、MC/DC100%テストの生成は非常に困難である、という問題がある。

3. MC/DCの概要

MC/DCは、Modified condition/decision coverageの略で、ソフトウェアのテスト網羅度の一つである。

MC/DCは、条件/判定の確認はもちろん、ソースコードを記述する際、しばしば間違いが起きるA or BとA and Bの記述ミスも検出でき、かつ、条件の数をn個とすると、テストケースの数が2のn乗個にならない、

現実的なテスト網羅度である。

与えられたテストが、MC/DC100%である、の定義は、以下を満たすことである。

- ① プログラムの全入口/出口を少なくとも一回はテストすること
- ② プログラムの判定に含まれる全条件は可能な値を少なくとも一回はテストすること
- ③ プログラムの全判定は可能な値を少なくとも一回はテストすること
- ④ プログラムの判定の全条件は判定の出力に独立に影響することを示すこと

④の「判定の出力に独立に影響すること」の解釈として様々な解釈が存在するが、本稿では、最もきつい解釈である、「他の条件の値を固定し、条件の値を変更することで、判定の出力値を変更すること」を採用する。

MC/DCの詳細については、参考文献[2]を参照して下さい。

4. 状態遷移表のMC/DC100%テストの具体例

MC/DCでは、確認すべき条件/判定の値、及び、“確認すべき条件/判定を含む条件判定文(if文)に到達すること”という制約からテストケースを生成する。

状態遷移表、及び、状態遷移表から生成されたソースコードには、条件/判定を含む条件判定文(if文)がテストする状態(テスト状態)の処理先頭にあるという特徴があり、“確認すべき条件/判定を含む条件判定文(if文)に到達すること”という制約は、“初期状態からテスト状態に遷移すること”という制約になる。状態遷移表の中身について熟知しているユーザであれば、“初期状態からテスト状態に遷移すること”を満たすシーケンス(遷移シーケンス)を容易に見つけ出せると考えられ、ユーザに遷移シーケンスを与えてもらうことで、一般のソースコードで必要であった膨大な空間内の探索が不要になる。このため、大規模な状態遷移表から生成されたソースコードに対しても、MC/DC100%テストの生成が比較的容易になる。

4節では、エアコンモデル(図1)を例題に使い、状態遷移表のMC/DC100%テストをどのように構成するのか、について、MC/DC100%テストをどのように実

行するのか、について説明する。

最初に、準備として、4.1節にて、エアコンモデルの概要を説明し、4.2節にて、MC/DC100%テストを作成する際に着目する、エアコンモデルの分岐構造を説明する。次に、4.3節にて、状態遷移表のMC/DC100%テストをどのように構成すればよいのか、について説明する。最後に、4.4節にて、弊社製 ZIPC を用いたシミュレーションにより、MC/DC100%テストをどのように実行するのか、について説明する。

図 1 エアコンモデル

4.1. エアコンモデルの概要

本稿で使用するエアコンモデルは、図 1 のモデル(状態遷移表)である。

状態遷移表は、横軸に、モデルの状態を示す“状態”をとり、縦軸に、モデルの状態を変化させる“事象”をとる。また、状態と事象が交差する“セル”に、この状態のときに、この事象が発生したときの振舞いを記述する。

エアコンモデルには、4つの状態“電源オフ”、“冷房”、“除湿”、“暖房”がある。“電源オフ”は、電源オフ時の状態である。“冷房”と“除湿”は、エアコンモデルがクーラーとして動作するときの状態である。クーラーは省エネが無効/有効かにより動作が変わり、省エネが無効のときは、実際に冷却させ、有効のときは、除湿により冷却させる。省エネが無効なときの状態が“冷房”で、省エネが有効なときの状態が“除湿”である。“暖房”は、エアコンモデルがヒーターとして動作するときの状態である。

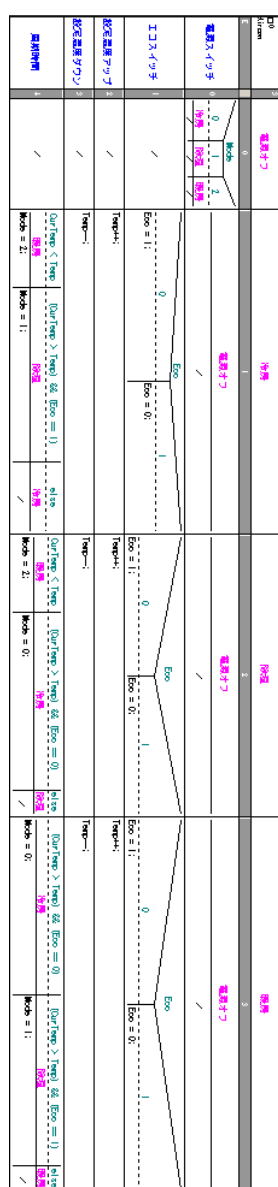
エアコンモデルには、5つの事象“電源スイッチ”、“エコスイッチ”、“設定温度アップ”、“設定温度ダウン”、“周期時間”がある。“電源スイッチ”は、電源をオン/オフさせる事象である。“エコスイッチ”は、省エネを無効/有効にさせる事象である。“設定温度アップ”は、設定温度を1度上げる事象で、“設定温度ダウン”は、設定温度を1度下げる事象である。“周期時間”は、一定時間毎に発生し、“冷房”、“除湿”、“暖房”の状態を切り替える事象である。現在気温が設定温度より低い場合は、“暖房”に切り替え、現在気温が設定温度より高く、かつ、省エネが無効のときは、“冷房”に切り替え、現在気温が設定温度より高く、かつ、省エネが有効のときは、“除湿”に切り替える。

エアコンモデルは、4つの変数“Mode”、“Eco”、“Temp”、“CurTemp”を持つ。“Mode”は、前回の電源オフ時に、“冷房”、“除湿”、“暖房”のどの状態だったかを保持する変数で、それぞれ、0, 1, 2 の値をとる。“Eco”は、省エネが無効/有効かを保持する変数で、それぞれ 0, 1 の値をとる。“Temp”は、設定温度を保持する変数で、“CurTemp”は現在温度を保持する変数である。

4.2. エアコンモデルの分岐構造

エアコンモデルは、状態遷移表を用いてモデル化されている。

状態遷移表の分岐構造は、状態遷移表の解釈の



仕方により異なるが、本稿では、ZIPC を用いた MC/DC100%テストを実施するので、ZIPC が採用する状態遷移表の解釈の仕方[3]に従う。

以下では、ZIPC が採用する状態遷移表の解釈の仕方に基づく、エアコンモデルの分岐構造について説明する。

エアコンモデルを、最も抽象度の高いレベルで見ると、どの事象が発生したのかを確認する条件判定文により、各事象の処理をするルーチンに分岐している。次に、抽象度を下げて、各事象の処理をするルーチンを見ると、現在どの状態なのかを確認する条件判定文により、各状態の処理をするルーチンに分岐している。更に、抽象度を下げて、各状態の処理をするルーチンを見ると、セル内部の条件判定文により分岐し、各条件分岐の処理をするルーチンに分岐している。

以降、まず、4.2.1節にて、事象による分岐について説明し、次に、4.2.2節にて、状態による分岐について説明し、4.2.3節にて、セル内部の if 文による分岐について説明する。最後に、4.2.4節にて、エアコンモデルの分岐構造についてまとめる。

4.2.1. 事象による分岐

エアコンモデルには、5つの事象“電源スイッチ”、“エコスイッチ”、“設定温度アップ”、“設定温度ダウン”、“周期時間”がある。

エアコンモデルを、最も抽象度の高いレベルで見ると、どの事象が発生したのかを確認する条件判定文が5つあり、それぞれ各事象の処理をするルーチンに分岐している(図 2)。

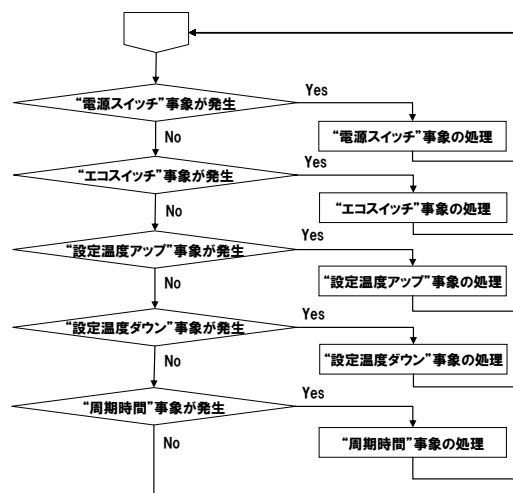


図 2 エアコンモデルの事象による分岐

4.2.2. 状態による分岐

エアコンモデルの5つの事象“電源スイッチ”、“エコスイッチ”、“設定温度アップ”、“設定温度ダウン”、“周期時間”の内、“周期時間”事象を代表として、事象の処理をするルーチンの分岐構造(状態による分岐)を説明する。

“周期時間”事象処理ルーチンは、4つの状態“電源オフ”、“冷房”、“除湿”、“暖房”を持つ。

“周期時間”事象処理ルーチンを、事象処理ルーチンレベルの抽象度で見ると、現在どの状態なのかを確認する条件判定文が4つあり、それぞれ各状態の処理をするルーチンに分岐している(図 3)。

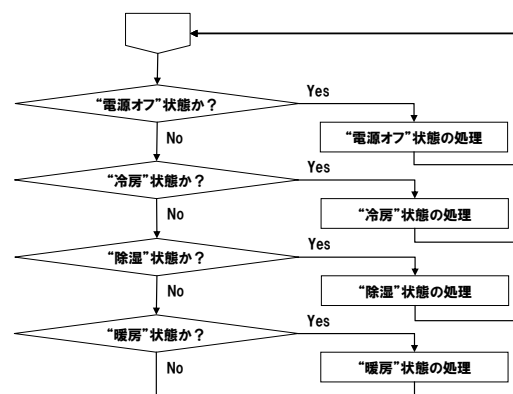


図 3 “周期時間”事象処理ルーチンの状態による分岐

4.2.3. セル内部の if 文による分岐

“周期時間”事象処理ルーチンが持つ4つの状態“電源オフ”, “冷房”, “除湿”, “暖房”の内, “暖房”状態を代表として, 状態の処理をするルーチンの分岐構造(セル内部の if 文による分岐)を説明する.

“周期時間”事象処理ルーチン内の“暖房”状態処理ルーチンは, “暖房”状態と“周期時間”事象が交差するセルに対応している. このセル内部には, “(CurTemp > Temp) && (Eco == 0)”と“(CurTemp > Temp) && (Eco == 1)”の2つの if 文による分岐がある.

“暖房”状態処理ルーチンを, 状態処理ルーチンレベルの抽象度で見ると, これらセル内部の if 文に対応する2つの条件判定文があり, それぞれ各条件分岐の処理をするルーチンに分岐している(図 4).

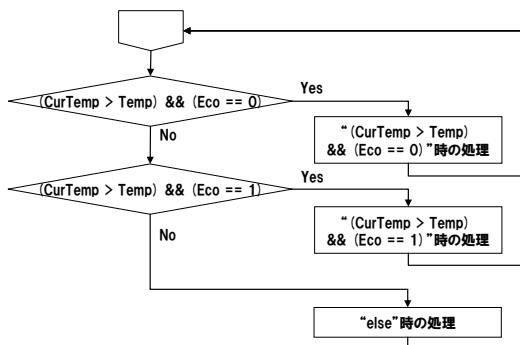


図 4 “周期時間”事象処理ルーチン内の“暖房”状態処理ルーチンの if 文による分岐

4.2.4. エアコンモデルの分岐構造のまとめ

エアコンモデルに現れる条件判定文の一覧が, 表 1 である. 事象による分岐が, 番号 1~5 の 5 個, 状態による分岐が, 番号 6~25 の 20 個, セル内部の if 文による分岐が, 番号 26~40 の 15 個で, 分岐は合計 40 個ある.

表 1 エアコンモデルの条件判定文一覧

番号	条件判定文	事象処理ルーチン	状態処理ルーチン
1	“電源スイッチ”事象が発生		
2	“エコスイッチ”事象が発生		
3	“設定温度アップ”事象が発生		
4	“設定温度ダウン”事象が発生		
5	“周期時間”事象が発生		
6	“電源オフ”状態か?	“電源スイッチ”事象	
7	“冷房”状態か?	“電源スイッチ”事象	
8	“除湿”状態か?	“電源スイッチ”事象	
9	“暖房”状態か?	“電源スイッチ”事象	
10	“電源オフ”状態か?	“エコスイッチ”事象	
11	“冷房”状態か?	“エコスイッチ”事象	
12	“除湿”状態か?	“エコスイッチ”事象	
13	“暖房”状態か?	“エコスイッチ”事象	
14	“電源オフ”状態か?	“設定温度アップ”事象	
15	“冷房”状態か?	“設定温度アップ”事象	
16	“除湿”状態か?	“設定温度アップ”事象	
17	“暖房”状態か?	“設定温度アップ”事象	
18	“電源オフ”状態か?	“設定温度ダウン”事象	
19	“冷房”状態か?	“設定温度ダウン”事象	
20	“除湿”状態か?	“設定温度ダウン”事象	
21	“暖房”状態か?	“設定温度ダウン”事象	
22	“電源オフ”状態か?	“周期時間”事象	
23	“冷房”状態か?	“周期時間”事象	
24	“除湿”状態か?	“周期時間”事象	
25	“暖房”状態か?	“周期時間”事象	
26	Mode == 0	“電源スイッチ”事象	“電源オフ”状態
27	Mode == 1	“電源スイッチ”事象	“電源オフ”状態
28	Mode == 2	“電源スイッチ”事象	“電源オフ”状態
29	Eco == 0	“エコスイッチ”事象	“冷房”状態
30	Eco == 1	“エコスイッチ”事象	“冷房”状態
31	Eco == 0	“エコスイッチ”事象	“除湿”状態
32	Eco == 1	“エコスイッチ”事象	“除湿”状態
33	Eco == 0	“エコスイッチ”事象	“暖房”状態
34	Eco == 1	“エコスイッチ”事象	“暖房”状態
35	CurTemp < Temp	“周期時間”事象	“冷房”状態
36	(CurTemp > Temp) && (Eco == 1)	“周期時間”事象	“冷房”状態
37	CurTemp < Temp	“周期時間”事象	“除湿”状態
38	(CurTemp > Temp) && (Eco == 0)	“周期時間”事象	“除湿”状態
39	(CurTemp > Temp) && (Eco == 0)	“周期時間”事象	“暖房”状態
40	(CurTemp > Temp) && (Eco == 1)	“周期時間”事象	“暖房”状態

4.3. エアコンモデルの MC/DC100% テスト

4.3 節では, エアコンモデルを例題として, 状態遷移表の MC/DC100% テストをどのように構成するのか, について説明する. まず, 準備として, 4.3.1 節にて, MC/DC100% テストに用いるテストケースの構造について説明し, その後, 4.3.2 節にて, 状態遷移表の MC/DC100% テストをどのように構成するのか, について, 具体的に説明する.

4.3.1. MC/DC100% テストに用いるテストケースの構造

エアコンモデルを, 最も抽象度の高いレベルで見ると, 図 2 になる. 図 2 には, 入口が一つ, 出口はないので, 一つ以上テストケースがあれば, MC/DC の定義①を満たす. 従って, 4.3.1 節では, MC/DC の定義②~④を満たすテストケースの構造について説明する.

MC/DC の定義②~④では, 条件判定文に着目し, 確認する条件値/判定値の必要条件について定義している. このため, テストケースは, まず, 初期状態から, 条件判定文を実行可能なテスト状態へ遷移させるシーケンスが必要である. そして, テスト状態で, 条件値

／判定値を確認する作業が必要である。

初期状態から、条件判定文を実行可能なテスト状態へ遷移させるシーケンスのことを、遷移シーケンス、テスト状態で、条件値／判定値を確認する作業のことを、テスト作業と呼ぶことにすると、テストケースは、遷移シーケンスとテスト作業から構成されている。

4.3.2. エアコンモデルの MC/DC100%テスト

表 2 がエアコンモデル MC/DC100%テストのテストケース一覧、表 3 がエアコンモデル MC/DC100%テストに現れる遷移シーケンス一覧である。表 2, 表 3 において、条番は、条件判定文番号の略で、遷番は、遷移シーケンス番号の略である。

表 2, 表 3 では、【と】に囲まれた部分に状態と変数値を記述している。但し、説明に関係しない変数値に関しては、省略している。

また、(と)に囲まれた部分に、【と】に囲まれた部分が示す状態、変数値のときに実施する作業を記述している。

そして、→は状態遷移を表す。

例えば、【電源オフ: Mode=0】(CurTemp=0)(Temp=10)(電源スイッチ)→【冷房: CurTemp=0: Temp=10】は、“電源オフ”状態で、“Mode”の値が0のときに、“CurTemp”を0に設定し、“Temp”を10に設定して、“電源スイッチ”事象を発生させると、冷房状態で、“CurTemp”が0、“Temp”が10になることを意味する。

4.3.1節で説明したように、エアコンモデル(図 2)には、入口が一つ、出口はないので、一つ以上テストケースがあれば、MC/DCの定義①を満たす。従って、以下では、このテストケースが、MC/DCの定義②～④を満たすことについて説明する。

エアコンモデルの条件判定文は、次のどちらかの構造を持っている。

クラス①: (条件1)

クラス②: (条件1) && (条件2)

番号1～35と37の条件判定文がクラス①、番号36, 38, 39, 40の条件判定文がクラス②である。

クラス①の条件判定文に対しては、

1. (判定)=(条件1)=T

2. (判定)=(条件1)=F

の2つのテストケースを用意することで、MC/DCの定

義②～④を満たすことができる。

クラス②の条件判定文に対しては、

1. (条件1)=T, (条件2)=T (このとき, (判定)=T)

2. (条件1)=F, (条件2)=T (このとき, (判定)=F)

3. (条件1)=T, (条件2)=F (このとき, (判定)=F)

の3つのテストケースを用意することで、MC/DCの定義②～④を満たすことができる。

但し、分岐の構造上、上記テストケースが実現不可能なことがある。例えば、図 3 のフローチャートでは、“暖房”状態か?から No の遷移が出ているが、エアコンモデルには、“電源オフ”、“冷房”、“除湿”、“暖房”の4つの状態しかないので、No の遷移が実行されることはありえない。エアコンモデルでは、番号 10, 18, 26, 34, 42, 50, 56, 60, 64, 68 の条件判定文が、ありえない組合せである。

確認すべき条件／判定の組合せから、このようなありえない組合せを除いたものが 100%になるテストケース群を、MC/DC100%テストと呼ぶ。

また、複数の確認すべき条件／判定の組合せを確認できるテストケースもある。例えば、図 2より、“エコスイッチ”事象が発生が Yes(表 2の番号3)になるには、“電源スイッチ”事象が発生が No(表 2の番号2)でなければならない。従って、“エコスイッチ”事象が発生が Yes(表 2の番号3)を確認するテストケースで、“電源スイッチ”事象が発生が No(表 2の番号2)になることを確認できる。つまり、番号3のテストケースは、番号2のテストケースとして使用可能である。表 2では、この使用関係を、テスト作業の⇒で記述している。

表 2のように、ありえない組合せ／重複したテストケース(表 2の網掛けした番号)を取り除くと、34個のテストケースが、エアコンモデルの MC/DC100%テストになる。

表 2 エアコンモデル MC/DC100%テストのテストケース一覧

番号	条番	条件判定文	判定	条件1	条件2	選番	テスト作業
1	1	"電源スイッチ"事象が発生	T	T			⇒11⇒51
2	1	"電源スイッチ"事象が発生	F	F			⇒3⇒19
3	2	"エコスイッチ"事象が発生	T	T			⇒19
4	2	"エコスイッチ"事象が発生	F	F			⇒5⇒27
5	3	"設定温度アップ"事象が発生	T	T			⇒27
6	3	"設定温度アップ"事象が発生	F	F			⇒7⇒35
7	4	"設定温度ダウン"事象が発生	T	T			⇒35
8	4	"設定温度ダウン"事象が発生	F	F			⇒9⇒43
9	5	"周期時間"事象が発生	T	T			⇒43
10	5	"周期時間"事象が発生	F	F			事象が発生しない状況
11	6	"電源オフ"状態か?	T	T			⇒51
12	6	"電源オフ"状態か?	F	F			⇒13
13	7	"冷房"状態か?	T	T		(4)	【冷房】(電源スイッチ)
14	7	"冷房"状態か?	F	F			⇒15
15	8	"除湿"状態か?	T	T		(8)	【除湿】(電源スイッチ)
16	8	"除湿"状態か?	F	F			⇒17
17	9	"暖房"状態か?	T	T		(12)	【暖房】(電源スイッチ)
18	9	"暖房"状態か?	F	F			状態=電源オフ冷房除湿暖房なので存在せず
19	10	"電源オフ"状態か?	T	T		(1)	【電源オフ】(エコスイッチ)
20	10	"電源オフ"状態か?	F	F			⇒21⇒57
21	11	"冷房"状態か?	T	T			⇒57
22	11	"冷房"状態か?	F	F			⇒23⇒61
23	12	"除湿"状態か?	T	T			⇒61
24	12	"除湿"状態か?	F	F			⇒25⇒65
25	13	"暖房"状態か?	T	T			⇒65
26	13	"暖房"状態か?	F	F			状態=電源オフ冷房除湿暖房なので存在せず
27	14	"電源オフ"状態か?	T	T		(1)	【電源オフ】(設定温度アップ)
28	14	"電源オフ"状態か?	F	F			⇒29
29	15	"冷房"状態か?	T	T		(4)	【冷房】(設定温度アップ)
30	15	"冷房"状態か?	F	F			⇒31
31	16	"除湿"状態か?	T	T			【除湿】(設定温度アップ)
32	16	"除湿"状態か?	F	F			⇒33
33	17	"暖房"状態か?	T	T			【暖房】(設定温度アップ)
34	17	"暖房"状態か?	F	F			状態=電源オフ冷房除湿暖房なので存在せず
35	18	"電源オフ"状態か?	T	T		(1)	【電源オフ】(設定温度ダウン)
36	18	"電源オフ"状態か?	F	F			⇒37
37	19	"冷房"状態か?	T	T		(4)	【冷房】(設定温度ダウン)
38	19	"冷房"状態か?	F	F			⇒39
39	20	"除湿"状態か?	T	T		(8)	【除湿】(設定温度ダウン)
40	20	"除湿"状態か?	F	F			⇒41
41	21	"暖房"状態か?	T	T		(12)	【暖房】(設定温度ダウン)
42	21	"暖房"状態か?	F	F			状態=電源オフ冷房除湿暖房なので存在せず
43	22	"電源オフ"状態か?	T	T		(1)	【電源オフ】(周期時間)
44	22	"電源オフ"状態か?	F	F			⇒45⇒69
45	23	"冷房"状態か?	T	T			⇒69
46	23	"冷房"状態か?	F	F			⇒47⇒74
47	24	"除湿"状態か?	T	T			⇒74
48	24	"除湿"状態か?	F	F			⇒49⇒79
49	25	"暖房"状態か?	T	T			⇒79
50	25	"暖房"状態か?	F	F			状態=電源オフ冷房除湿暖房なので存在せず
51	26	Mode == 0	T	T		(1)	【電源オフ:Mode=0】(電源スイッチ)
52	26	Mode == 0	F	F			⇒53
53	27	Mode == 1	T	T		(2)	【電源オフ:Mode=1】(電源スイッチ)
54	27	Mode == 1	F	F			⇒55
55	28	Mode == 2	T	T		(3)	【電源オフ:Mode=2】(電源スイッチ)
56	28	Mode == 2	F	F			Mode=0,1,2なので存在せず
57	29	Eco == 0	T	T		(7)	【冷房:Eco=0】(エコスイッチ)
58	29	Eco == 0	F	F			⇒59
59	30	Eco == 1	T	T		(5)	【冷房:Eco=1】(エコスイッチ)
60	30	Eco == 1	F	F			Eco=0,1なので存在せず
61	31	Eco == 0	T	T		(9)	【除湿:Eco=0】(エコスイッチ)
62	31	Eco == 0	F	F			⇒63
63	32	Eco == 1	T	T		(11)	【除湿:Eco=1】(エコスイッチ)
64	32	Eco == 1	F	F			Eco=0,1なので存在せず
65	33	Eco == 0	T	T		(12)	【暖房:Eco=0】(エコスイッチ)
66	33	Eco == 0	F	F			⇒67
67	34	Eco == 1	T	T		(14)	【暖房:Eco=1】(エコスイッチ)
68	34	Eco == 1	F	F			Eco=0,1なので存在せず
69	35	CurTemp < Temp	T	T		(4)	【冷房:CurTemp=0:Temp=10】(周期時間)
70	35	CurTemp < Temp	F	F			⇒71
71	36	(CurTemp > Temp) && (Eco == 1)	T	T		(5)	【冷房:CurTemp=20:Temp=10:Eco=1】(周期時間)
72	36	(CurTemp > Temp) && (Eco == 1)	F	F		(6)	【冷房:CurTemp=10:Temp=10:Eco=1】(周期時間)
73	36	(CurTemp > Temp) && (Eco == 1)	F	F		(7)	【冷房:CurTemp=20:Temp=10:Eco=0】(周期時間)
74	37	CurTemp < Temp	T	T		(8)	【除湿:CurTemp=0:Temp=10】(周期時間)
75	37	CurTemp < Temp	F	F			⇒76
76	38	(CurTemp > Temp) && (Eco == 0)	T	T		(9)	【除湿:CurTemp=20:Temp=10:Eco=0】(周期時間)
77	38	(CurTemp > Temp) && (Eco == 0)	F	F		(10)	【除湿:CurTemp=10:Temp=10:Eco=0】(周期時間)
78	38	(CurTemp > Temp) && (Eco == 0)	F	F		(11)	【除湿:CurTemp=20:Temp=10:Eco=1】(周期時間)
79	39	(CurTemp > Temp) && (Eco == 0)	T	T		(12)	【暖房:CurTemp=20:Temp=10:Eco=0】(周期時間)
80	39	(CurTemp > Temp) && (Eco == 0)	F	F		(13)	【暖房:CurTemp=10:Temp=10:Eco=0】(周期時間)
81	39	(CurTemp > Temp) && (Eco == 0)	F	F			⇒82
82	40	(CurTemp > Temp) && (Eco == 1)	T	T		(14)	【暖房:CurTemp=20:Temp=10:Eco=1】(周期時間)
83	40	(CurTemp > Temp) && (Eco == 1)	F	F		(15)	【暖房:CurTemp=10:Temp=10:Eco=1】(周期時間)
84	40	(CurTemp > Temp) && (Eco == 1)	F	F			76と同じ条件/判定値なので、40の条件判定文は実行されない

表 3 エアコンモデル MC/DC100%テストの遷移シーケンス一覧

順番	テスト状態	遷移シーケンス
(1)	【電源オフ:Mode=0】	【電源オフ:Mode=0】
(2)	【電源オフ:Mode=1】	【電源オフ:Mode=0:Eco=0】(CurTemp=20)(Temp=10)(電源スイッチ)→ 【冷房:Mode=0:CurTemp=20:Temp=10:Eco=0】(エコスイッチ)→ 【冷房:Mode=0:CurTemp=20:Temp=10:Eco=1】(周期時間)→ 【除湿:Mode=1:CurTemp=20:Temp=10:Eco=1】(電源スイッチ)→ 【電源オフ:Mode=1】
(3)	【電源オフ:Mode=2】	【電源オフ:Mode=0:CurTemp=0:Eco=0】(Temp=10)(電源スイッチ)→ 【冷房:Mode=0:CurTemp=0:Temp=10:Eco=0】(周期時間)→ 【暖房:Mode=2】(電源スイッチ)→ 【電源オフ:Mode=2】
(4)	【冷房:CurTemp=0:Temp=10】	【電源オフ:Mode=0:CurTemp=0】(Temp=10)(電源スイッチ)→ 【冷房:CurTemp=0:Temp=10】
(5)	【冷房:CurTemp=20:Temp=10:Eco=1】	【電源オフ:Mode=0:Eco=0】(CurTemp=20)(Temp=10)(電源スイッチ)→ 【冷房:CurTemp=20:Temp=10:Eco=0】(エコスイッチ)→ 【冷房:CurTemp=20:Temp=10:Eco=1】
(6)	【冷房:CurTemp=10:Temp=10:Eco=1】	【電源オフ:Mode=0:Eco=0】(CurTemp=10)(Temp=10)(電源スイッチ)→ 【冷房:CurTemp=10:Temp=10:Eco=0】(エコスイッチ)→ 【冷房:CurTemp=10:Temp=10:Eco=1】
(7)	【冷房:CurTemp=20:Temp=10:Eco=0】	【電源オフ:Mode=0:Eco=0】(CurTemp=20)(Temp=10)(電源スイッチ)→ 【冷房:CurTemp=20:Temp=10:Eco=0】
(8)	【除湿:CurTemp=0:Temp=10】	【電源オフ:Mode=0:Eco=0】(CurTemp=20)(Temp=9)(電源スイッチ)→ 【冷房:CurTemp=20:Temp=9:Eco=0】(エコスイッチ)→ 【冷房:CurTemp=20:Temp=9:Eco=1】(周期時間)→ 【除湿:CurTemp=20:Temp=9:Eco=1】(CurTemp=0)(設定温度アップ)→ 【除湿:CurTemp=0:Temp=10】
(9)	【除湿:CurTemp=20:Temp=10:Eco=0】	【電源オフ:Mode=0:Eco=0】(CurTemp=20)(Temp=10)(電源スイッチ)→ 【冷房:CurTemp=20:Temp=10:Eco=0】(エコスイッチ)→ 【冷房:CurTemp=20:Temp=10:Eco=1】(周期時間)→ 【除湿:CurTemp=20:Temp=10:Eco=1】(エコスイッチ)→ 【除湿:CurTemp=20:Temp=10:Eco=0】
(10)	【除湿:CurTemp=10:Temp=10:Eco=0】	【電源オフ:Mode=0:Eco=0】(CurTemp=20)(Temp=10)(電源スイッチ)→ 【冷房:CurTemp=20:Temp=10:Eco=0】(エコスイッチ)→ 【冷房:CurTemp=20:Temp=10:Eco=1】(周期時間)→ 【除湿:CurTemp=20:Temp=10:Eco=1】(エコスイッチ)→ 【除湿:CurTemp=20:Temp=10:Eco=0】(CurTemp=10)→ 【除湿:CurTemp=10:Temp=10:Eco=0】
(11)	【除湿:CurTemp=20:Temp=10:Eco=1】	【電源オフ:Mode=0:Eco=0】(CurTemp=20)(Temp=10)(電源スイッチ)→ 【冷房:CurTemp=20:Temp=10:Eco=0】(エコスイッチ)→ 【冷房:CurTemp=20:Temp=10:Eco=1】(周期時間)→ 【除湿:CurTemp=20:Temp=10:Eco=1】
(12)	【暖房:CurTemp=20:Temp=10:Eco=0】	【電源オフ:Mode=0:Eco=0:CurTemp=0】(Temp=9)(電源スイッチ)→ 【冷房:CurTemp=0:Temp=9:Eco=0】(周期時間)→ 【暖房:CurTemp=0:Temp=9:Eco=0】(CurTemp=20)(設定温度アップ)→ 【暖房:CurTemp=20:Temp=10:Eco=0】
(13)	【暖房:CurTemp=10:Temp=10:Eco=0】	【電源オフ:Mode=0:Eco=0:CurTemp=0】(Temp=9)(電源スイッチ)→ 【冷房:CurTemp=0:Temp=9:Eco=0】(周期時間)→ 【暖房:CurTemp=0:Temp=9:Eco=0】(CurTemp=10)(設定温度アップ)→ 【暖房:CurTemp=10:Temp=10:Eco=0】
(14)	【暖房:CurTemp=20:Temp=10:Eco=1】	【電源オフ:Mode=0:Eco=0:CurTemp=0】(Temp=9)(電源スイッチ)→ 【冷房:CurTemp=0:Temp=9:Eco=0】(周期時間)→ 【暖房:CurTemp=0:Temp=9:Eco=0】(CurTemp=20)(設定温度アップ)→ 【暖房:CurTemp=20:Temp=10:Eco=0】(エコスイッチ)→ 【暖房:CurTemp=20:Temp=10:Eco=1】
(15)	【暖房:CurTemp=10:Temp=10:Eco=1】	【電源オフ:Mode=0:Eco=0:CurTemp=0】(Temp=9)(電源スイッチ)→ 【冷房:CurTemp=0:Temp=9:Eco=0】(周期時間)→ 【暖房:CurTemp=0:Temp=9:Eco=0】(CurTemp=10)(設定温度アップ)→ 【暖房:CurTemp=10:Temp=10:Eco=0】(エコスイッチ)→ 【暖房:CurTemp=10:Temp=10:Eco=1】

4.4. シミュレーションを用いた MC/DC100%テスト実行

4.4節では、弊社製 ZIPC を用いたシミュレーションにより、MC/DC100%テストをどのように実行するのか、について説明する。

一般にテストでは、要求仕様を満たしているかどうかを確認するが、本稿では、要求仕様として、以下の項目を確認する。

- 要求①：“冷房”状態では、Mode = 0 であること
- 要求②：“除湿”状態では、Mode = 1 であること
- 要求③：“暖房”状態では、Mode = 2 であること

MC/DC100%テストでは、表 2のありえない組合せ／重複したテストケース(表 2の網掛けした番号)を取り除いた、34 個のテストケースに対し、シミュレーションを行い、要求①、要求②、要求③が成り立つことを確認する。ZIPC には、シミュレーションのログをとる機能があり、このログを確認することで、要求①、要求②、要求③が成り立つかどうかを確認できる。

以下、番号 79 のテストケース(表 2)を例にとり、具体的なテスト実行方法について説明する。

番号 79 のテストケースでは、番号 12 の遷移シーケンス(表 3)を用いてテスト状態へ遷移させるので、まず、シミュレーションにより、番号 12 の遷移シーケンスを実行する。

シミュレーションの開始時点は、図 5になる。ZIPC のシミュレーションでは、変数値の変更は、変数ウィンドウを使用して行き、事象の発行は、事象の上にカーソルを置き、右クリックしたときに現れるメニューで、イベント発行を選択することで行う。

番号 12 の遷移シーケンスの実行は、下記になる。

1. 初期状態(“電源オフ”状態で、全ての変数値は 0)において、変数ウィンドウで“Temp”の値を 9 に設定し、イベント発行により“電源スイッチ”事象を発生させる。
2. イベント発行により“周期時間”事象を発生させる。
3. 変数ウィンドウで“CurTemp”の値を 20 に設定し、イベント発行により“設定温度アップ”事象を発生させる。

1～3により、番号 79 のテストケースのテスト状態(“暖房”状態、CurTemp=20、Temp=10、Eco=0)に遷移する。

ここで、番号 79 のテストケースのテスト作業を行う。

- ・ イベント発行により“周期時間”事象を発生させ

る(図 6)。

番号 79 のテストケースのシミュレーションにより、図 7のログが生成される。

要求①、要求②、要求③が成り立つかどうかを確認するには、図 7のログにおける、状態の変更箇所と“Mode”値の変更箇所に着目すればよい。

状態は初期状態が“電源オフ”(2 行目で設定)で、変更箇所は、33 行目(“電源オフ”→“冷房”)、59 行目(“冷房”→“暖房”)、113 行目(“暖房”→“冷房”)である。“Mode”値は初期状態が 0 で、変更箇所は、57 行目(0→2)、111 行目(2→0)である。

33 行目(“電源オフ”→“冷房”)で“冷房”状態になったとき、“Mode”値は初期状態の 0 で、要求①を満たす。59 行目(“冷房”→“暖房”)で“暖房”状態になったとき、“Mode”値は 57 行目(0→2)で 2 になっているので、要求③を満たす。113 行目(“暖房”→“冷房”)で“冷房”状態になったとき、“Mode”値は 111 行目(2→0)で 0 になっているので要求①を満たす。このように、番号 79 のテストケースは、要求①、要求③を満たす。なお、番号 79 のテストケースでは、“除湿”状態にはならないので、要求②は関係しない。

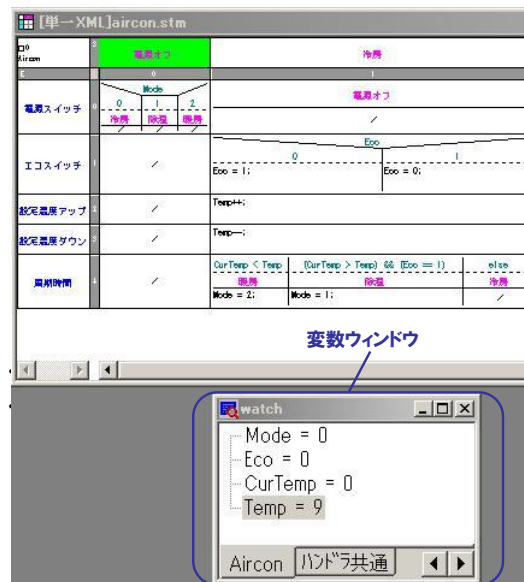


図 5 シミュレーションの開始時点

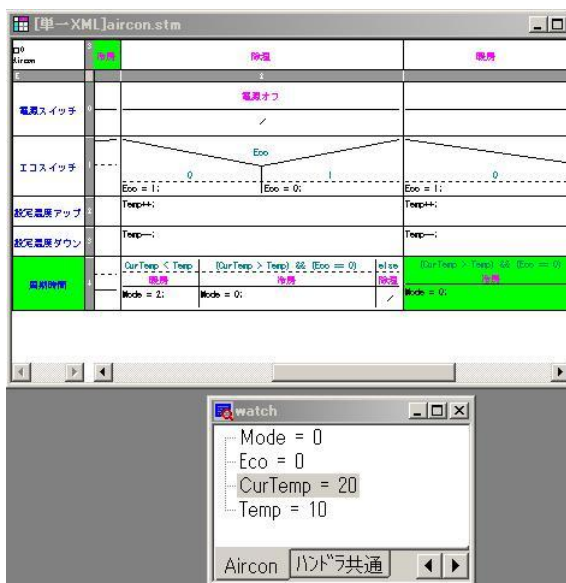


図 6 シミュレーションの終了時点

ZIPC Simulator Log File Format Ver.1.00

```

1 RUN 0 STOP ""
2 SME 4000000 RUN "Aircon" "Aircon" "ZAircon_m0State [0]" "0" "0"
3 ACS 8000000 RUN "Aircon" "Aircon" "Aircon" "-1" "3" "1:0"
4 ACS 9000000 RUN "Aircon" "Aircon" "Aircon" "-1" "10" "1:0"
5 DSP 13000000 RUN "Aircon" "RUN" "WAIT"
  【中略】
25 REV 2956553013000000 RUN "Aircon" "電源スイッチ" "@Simulator" "0"
26 ACS 2956553018000000 RUN "Aircon" "Aircon" "Aircon" "-1" "1" "0:1"
27 SEN 2956553019000000 RUN "Aircon" "Aircon" "Aircon" "0"
28 ACS 2956553019000000 RUN "Aircon" "Aircon" "Aircon" "-1" "9" "0:1"
29 ACS 2956553025000000 RUN "Aircon" "Aircon" "Aircon" "-1" "5" "1:1:0"
30 ACS 2956553027000000 RUN "Aircon" "Aircon" "Aircon" "-1" "5" "1:1:0"
31 ACS 2956553028000000 RUN "Aircon" "Aircon" "Aircon" "-1" "5" "1:1:1:6"
32 ACS 2956553029000000 RUN "Aircon" "Aircon" "Aircon" "-1" "5" "1:1:1:3"
33 SME 2956553030000000 RUN "Aircon" "Aircon" "ZAircon_m0State [0]" "0" "1"
  【中略】
57 SME 200791940710000000 RUN "Aircon" "Aircon" "Mode" "0" "2"
58 ACS 200791940720000000 RUN "Aircon" "Aircon" "Aircon" "-1" "5" "2:5:3"
59 SME 200791940730000000 RUN "Aircon" "Aircon" "ZAircon_m0State [0]" "1" "3"
  【中略】
111 SME 414511591370000000 RUN "Aircon" "Aircon" "Mode" "2" "0"
112 ACS 414511591380000000 RUN "Aircon" "Aircon" "Aircon" "-1" "5" "4:5:3"
113 SME 414511591390000000 RUN "Aircon" "Aircon" "ZAircon_m0State [0]" "3" "1"
  【後略】

```

図 7 シミュレーションのログ

5. 結論

4節でエアコンモデルを用いて説明したように、状態遷移表の MC/DC100%テストは、初期状態からテスト状態へ遷移させる遷移シーケンスをユーザが与えると、定型的で容易に作成できた。遷移シーケンスも、エアコンモデルの振舞いについて熟知したユーザであれば、容易に作成可能であると考えられる。

また、条件判定文 40 個(条件が1個の条件判定文

36 個と条件が 2 個の条件判定文 4 個)に対し、テストケース 34 個と、テストケースが条件判定文に対し膨大な数になっていない。

更に、シミュレーションを行い、ログを確認することで、要求仕様を満たすかどうかを容易に確認できた。

このため、状態遷移表の MC/DC100%テストをお勧めする。

6. 今後の課題

本稿では、状態遷移表の MC/DC100%テストの生成とシミュレーションを手動で実行した。しかし、MC/DC100%テストを生成する部分のうち、遷移シーケンスを求める部分以外は、定型的で自動化が可能である。

このことから、現在、弊社では、ユーザが入力した遷移シーケンスと要求仕様を利用して、ZIPC で状態遷移表の MC/DC100%テストを自動実行する方法を研究している。

7. むすび

本稿では、エアコンモデルを例題に、状態遷移表の MC/DC100%テストをどのように構成するのか、についてと、MC/DC100%テストをどのように実行するのか、について説明した。

状態遷移表の MC/DC100%テストは、ユーザが遷移シーケンスを与えることで、比較的容易に作成でき、また、シミュレーションを用いて、要求仕様を満たしていることを簡単に確認できることがお分かりいただけたいと思う。

状態を持つシステムへの網羅的なテストを検討している方は、是非、状態遷移表の MC/DC100%テストを試してほしい。

参考文献

- [1] Kelly J. Hayhurst, Dan S. Veerhusen, John J. Chilenski, and Leanna K. Rierson, “A Practical Tutorial on Modified Condition / Decision Coverage”, NASA/TM-2001-210876, 2001.
- [2] 松本充広, “MC/DC による現実的網羅のススメ”, <http://www.zipc.com/cesl/info/03.pdf>
- [3] 渡辺政彦, “拡張階層化状態遷移表設計手法 Ver.2.0 –Embedded SE のための設計手法–”, 1998.



松本充広

1990年 京都大学・理学部卒. 2002年 北陸先端科学技術大学院大学情報科学研究科博士後期課程修了. 博士(情報科学). 現在, キャッツ組込みソフトウェア研究所研究員.

みソフトウェア研究所研究員.