

# ソフトウェア工学III

## 第11回

---

高木 智彦

教科書: 中所武司, “ソフトウェア工学 第3版”, 朝倉書店, 2014.

## 7.3. 構造テスト

- ◆ **構造テスト(ホワイトボックステスト)**: プログラムの内部構造に基づいてテスト項目を作成する. 基本的には, プログラムの制御構造を表す制御フローグラフに基づいて, 入り口から出口に至るパス(制御フローグラフ上の経路)を作成する.
- ◆ どのようなパスを選択するか? → **テスト網羅性の基準**
- ◆ 基本的なテストデータ作成手順:
  - (1) テスト網羅基準を満たすパスの最小セットを選ぶ
  - (2) 各パスを通過するためのパス条件を選ぶ
  - (3) 各パス条件を満たすデータ値を求める

## 7.3. 構造テスト

- ◆ 例題: 0~100の整数の列を入力し, 最大値を求めるプログラム
- ◆ 制御フローグラフでは, 必ず先頭から実行され, かつ途中で分岐しない命令の集まりを1つのノード (基本ブロック) とする.

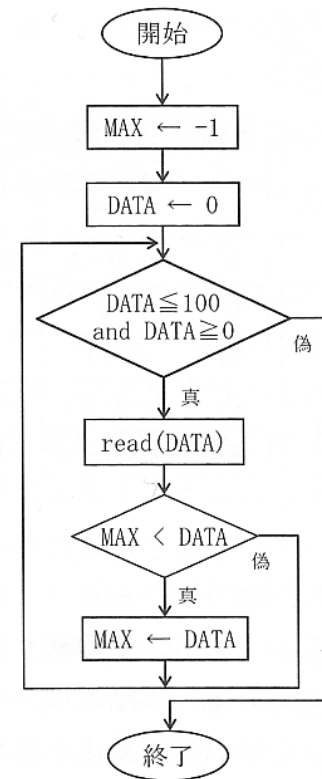


図 7.5 例題のフローチャート

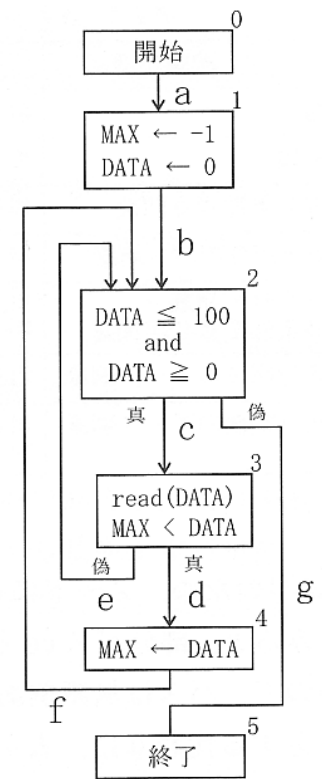


図 7.6 例題の制御フローグラフ

## 7.3. 構造テスト

- ◆ **全命令網羅(全文網羅)**: 全ノードを1回以上実行するパスのセットを選ぶ.
- ◆ **全分岐網羅**: 全アークを1回以上実行するパスのセットを選ぶ.
- ◆ **ループ繰り返し数を含む基準**: 繰り返し回数に依存した誤り検出のために, 0回や複数回(最大数または2回)のパスを選ぶ.
- ◆ **繰り返し処理を除く全パス網羅基準**: 繰り返し処理を0または1回に限定した全パス網羅. 繰り返し処理を対象外とすればパス数は有限となる.

表 7.2 各テスト網羅基準を満たすパスセットの例

	テスト網羅基準	パスセット	(注) パスの詳細		
			パス		入力データ例
簡 易 化	文網羅	#1	#1	abcdfg	(101)
	分岐網羅	#2	#2	abcdfceg	(100, -1)
	繰り返し条件成立回数	#3(0回), #1(1回), #2(2回)	#3	abg	なし
	繰り返し以外のパス網羅	#1, #3, #4	#4	abceg	(-1)
詳 細 化	データフロー網羅 (分岐条件変数参照網羅)	#1, #2, #3, #4 #5(ノード4→e)	#5	abcdfcdg	(0, 101)
	論理式分割後のアーク網羅	#1(g1), #2(h, g2)	#6	abcdfceceg	(0, 0, -1)
	比較式分割後のアーク網羅	#1(h1, c2), #2(h2, c1, e2), #6(e1)			



## 7.3. 構造テスト

- ◆ データフローに基づく基準: 変数の値の定義と参照の関係を示すデータフローに着目した網羅基準. 全分岐網羅と全パス網羅の間に位置する.
- (1) **分岐条件変数参照網羅**: ノード中の分岐条件に含まれる変数の参照を, そのノードの出力アークに対応付ける. そして, その変数を定義したノードと参照アークの組合せを網羅する.
- (2) 全変数参照網羅: 分岐条件で用いられる変数に限らず, すべての変数参照を対象とする.
- (3) 全定義参照パス網羅: ループを含まないすべての定義参照パスを網羅する.

	(c) $0 \leq \text{DATA} \leq 100$	(d) $\text{MAX} < \text{DATA}$	(e) $\text{MAX} \geq \text{DATA}$	(g) $\text{DATA} < 0,$ $100 < \text{DATA}$
(1) $\text{DATA} \leftarrow 0$	# 1	不可	不可	不可
(3) $\text{read}(\text{DATA})$	# 2	# 1	# 2	# 2, # 1

(a) 変数 DATA に関する組み合わせ

	(d) $\text{MAX} < \text{DATA}$	(e) $\text{MAX} \geq \text{DATA}$
(1) $\text{MAX} \leftarrow -1$	# 1	# 4
(4) $\text{MAX} \leftarrow \text{DATA}$	追加要 (# 5)	# 2

(b) 変数 MAX に関する組み合わせ

図 7.7 データフローに基づく基準 (分岐条件変数参照網羅) の例

# 7.3. 構造テスト

- ◆ **分岐アークの組み合わせによる基準**: n個以下の分岐アークの組み合わせからなるすべての部分パスを網羅する.
- ◆ 分岐条件の詳細化: 誤りの発生しやすい分岐条件自身のテストに注力.
  - (1) **論理式の分割**: 分岐条件が論理和や論理積を用いた複数の条件からなる場合は, それぞれが真と偽の場合をテストする.
  - (2) **比較式の分割**: さらに, 各条件が値の大小を比較する式の場合は,  $<, =, >$ の3種類のテストを行う.
- ◆ モジュール呼び出しに関連する基準: 全モジュールを1回以上実行, など.

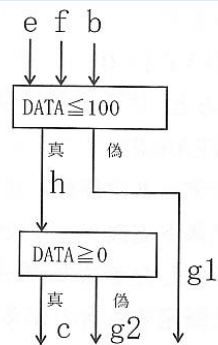


図 7.8 論理式の分割の例

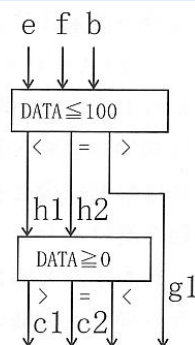


図 7.9 比較式の分割の例

	$DATA \leq 100$	$0 \leq DATA$	$MAX < DATA$
<	h1 : # 1	c1 : # 2	d : # 1
=	h2 : # 2	c2 : # 1	e1 : # 6
>	g1 : # 1	g2 : # 2	e2 : # 2

図 7.10 比較式の分割に基づく分岐網羅の見直しの例

## 7.3.2. 構造テストの問題点

- a. **実行不可能パスの選択**: 機械的にパスを選ぶと, 実行不可能なものが混じる.  
→テストデータは人手で作成・実行し, ツールを使って網羅できた部分を表示する, という方法が一般的.
- b. **大規模ソフトウェアのテスト網羅の困難性**: →サブシステムのテストで100%, システムテストで90%前後の全分岐網羅基準を用いることが实际的.
- c. **全パス網羅基準との格差**: 全分岐網羅基準の採用が一般的だが, 全パス網羅基準との格差が大きい →中間基準の採用
- d. **分岐条件のテスト不十分性**: →論理式や比較式を分割する方式の採用
- e. **機能欠如の検出不可**: →機能テストの適用
- f. **品質過大評価傾向**: 1つのテストデータで複数の被網羅要素が実行される →本質的でない被網羅要素を対象外とする

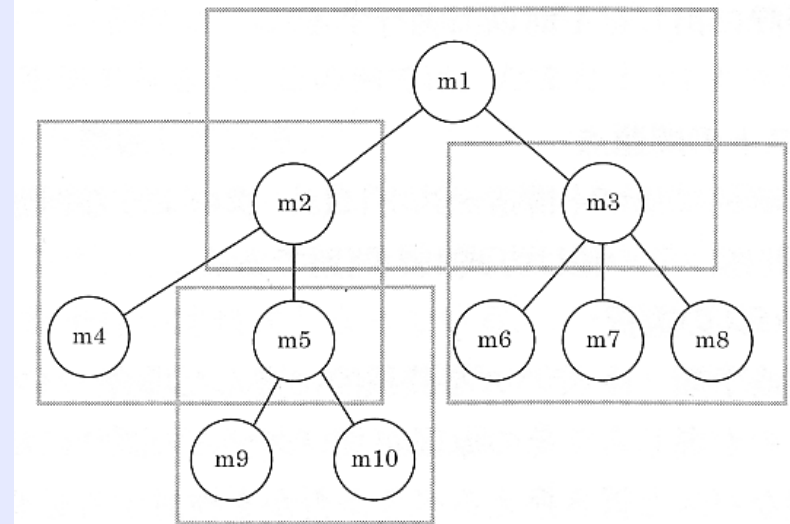


図 7.11 大規模ソフトウェアの構造テスト法

## 7.4. テスト手順

- (1) 機能テスト法によるテスト項目の選択
- (2) テストデータの作成とテスト実施(実行された命令や分岐の計測)
- (3) 未実行の命令や分岐を実行するテストデータの追加
- (4) 構造テスト法の観点でテスト網羅基準に基づく終了判断

