ソフトウェア工学III
第10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.
参考書：紫合治, “プログラム工学”, サイエンス社, 2002.（ソフトウェア工学Iの教科書）

6. プログラミング

- 記述性より理解性を重視 → 構造化プログラミング
- 構造化コーディング：goto文を使用せず，逐次，選択，繰り返しの基本制御構造のみでプログラムを記述する方法．図式にも適用．
- 段階的詳細化：おおまかなことを決め，次第に詳細な設計へ進む方法．
- データ抽象化：データとそのデータの操作手続きをまとめておき，データへのアクセスは操作手続きを通してのみ可能とする方法 → 抽象データ型，クラス

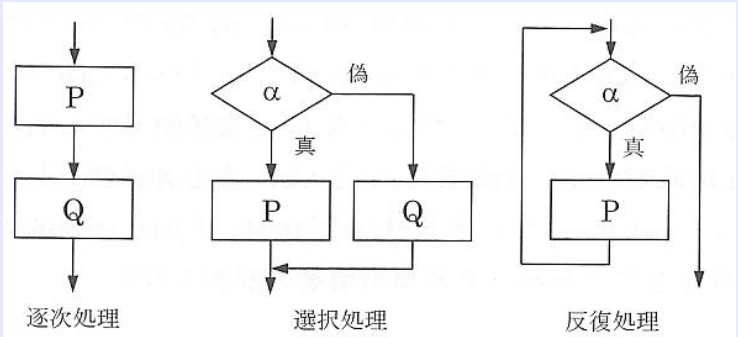


図 6.1　構造化コーディングのための制御構造の基本形

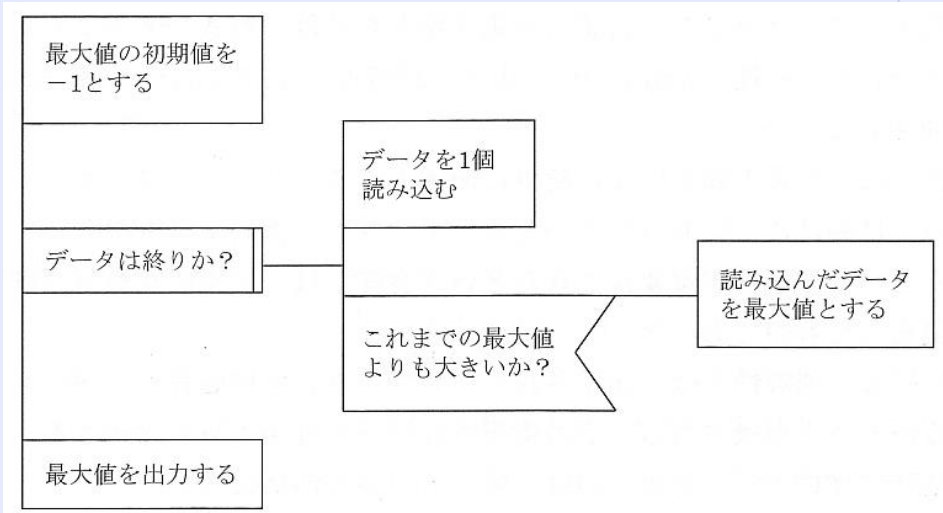


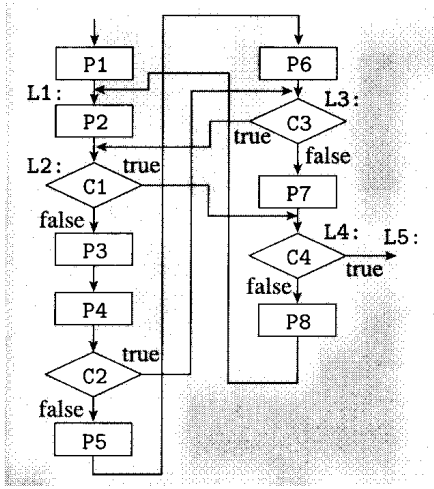
図 6.2　構造化図式の例（最大値を求めるプログラム）

6. プログラミング

・制御構造の標準化
　逐次，選択，繰返しの3基本制御構造の組合せで全てのプログラムを作成する．
・段階的詳細化
　いきなり細かい処理の記述をせずに，プログラムの処理概要から段階的に詳細化していくことによってプログラムを完成させる．

複雑な事柄を，いくつかのより単純な事柄の単純な関連に解きほぐすことにより，一度に考えるべき事柄の量を減らす．

図 2.1　構造化プログラミングの原理



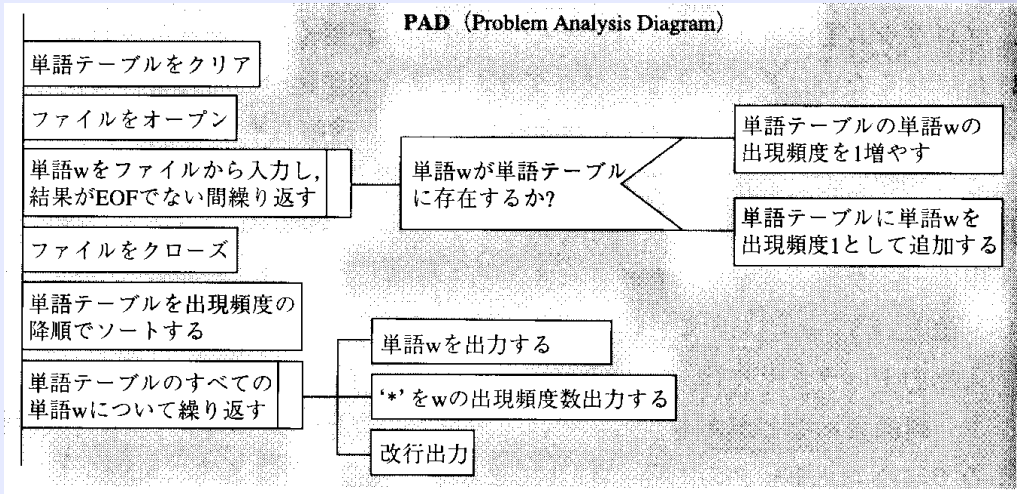

```
        P1;
L1:     P2;
L2:     if( C1 ) goto L4;
        P3;
        P4;
        if( C2 ) goto L3;
        P5;
        P6;
L3:     if( C3 ) goto L2;
        P7;
L4:     if( C4 ) goto L5;
        P8;
        goto L1;
L5:
```

図 2.4　Goto 文を使ったスパゲティプログラム

```
単語の出現頻度表の出力：  {
      単語テーブルをクリア；
      ファイルをオープン；   (†1)
      while( 単語wをファイルから入力し，結果がEOFでない ) {
            if( 単語wが単語テーブルに存在する )
                  単語テーブルのwの出現度を1増やす；
            else
                  単語テーブルにwを出現度1として追加する；(†2)
      }
      ファイルをクローズ；
      単語テーブルを出現頻度の降順でソートする；
      forall 単語w in 単語テーブル {
            単語wを出力する；
            '*' をwの出現頻度数出力する；
            改行出力；
      }
}
```

ただし，
†1：ファイルがオープンできない場合はメッセージ出力後終了する．
†2：単語の種類が上限を超えた場合はメッセージ出力後，ファイル読込み処理を中断し，先(ファイルクローズ以下)に進む．

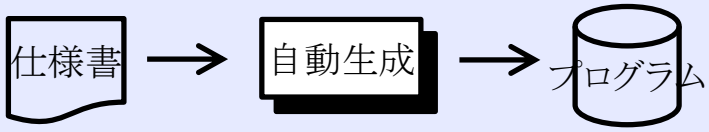
図 2.10　擬似言語による論理表現の例



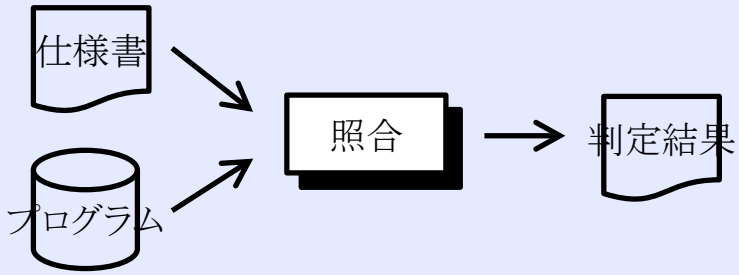
7.1.1 プログラム検証の方法

- **verification**(プログラムの検証)とは，プログラムが仕様通りに作られていることを確かめる作業である.
- 顧客の要求通りに作られていることの確認は**validation** という.
- verificationの方法として，以下の3つが考えられる.
  ① 仕様書からプログラムを自動生成
  ② 仕様書とプログラムの等価性証明
  ③ テストデータによる動的テスト
- 動的テストが最も現実的で一般的に用いられている.
- プログラムの作成を人手に頼る現在の開発方式では，動的テストによって品質保証を行わざるをえない.

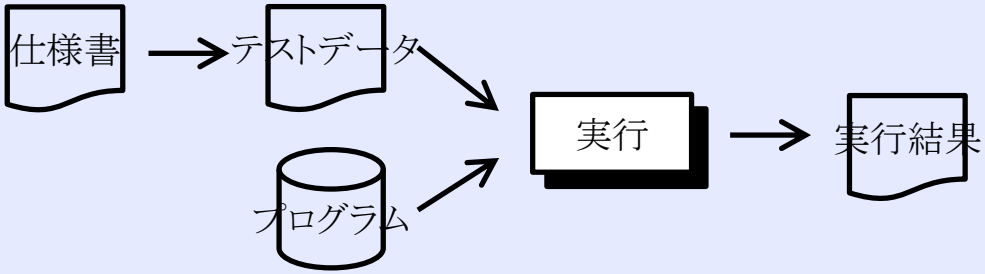
7.1.1 プログラム検証の方法



① 仕様書からプログラムを自動生成



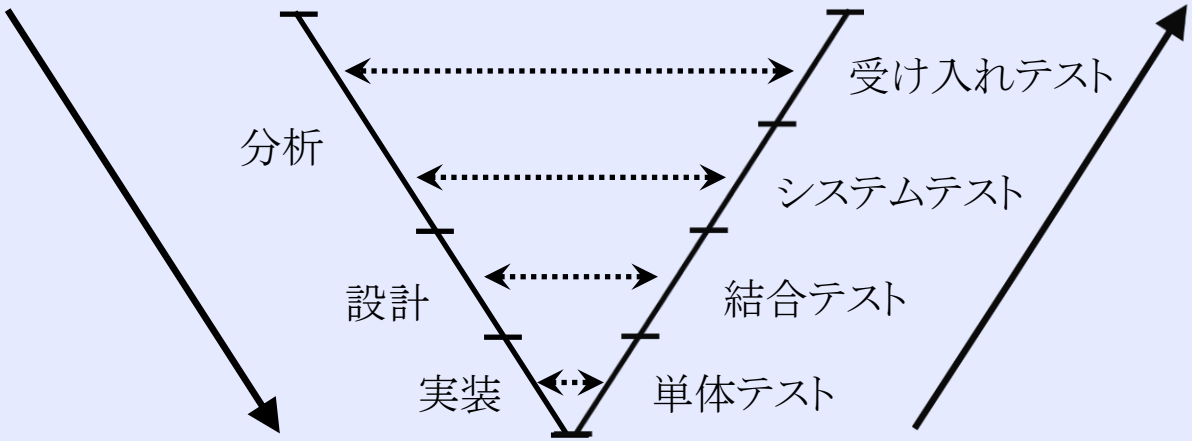
② 仕様書とプログラムの等価性証明



③ テストデータによる動的テスト

7.1.2 テスト工程

- テスト工程は4つの小工程から構成される.
- 各小工程は，上流の各工程と対応している.
- 受け入れテストは顧客またはメーカの検査部門で実施するテスト. その他は開発者によるテスト.



分析・設計・実装工程とテスト工程の対応（V字モデル）

7.1.2 テスト工程

- **単体テスト:** モジュールが仕様書通りに作られているか否かを検査. **ドライバ**や**スタブ**などが必要.
- **結合テスト:** 関連するモジュールを結合して実行し, 呼び出す側と呼び出される側のインタフェースが一致しているか否かを検査.
- **システムテスト:** プログラム全体が最終的にシステムの機能仕様や性能仕様を満たし, 所期の目的を達成しているか否かを検査.
- **受け入れテスト:** 顧客への納入時に顧客自身によって行われるテスト. Validationが目的. パッケージ製品の場合は, メーカの検査部門が顧客の立場で行う.

7.1.3 テスト技術

- 考えうる入力データ数は膨大．限られた費用と時間の中で高い品質保証の与えられる入力データ(テストデータ)を設計するためにテスト技術が必要．
- **機能テスト**：プログラムの機能仕様に基づいて**テストケース**(テストデータと期待結果)を設計する方法．同値分割法，限界値分析法，原因結果グラフ法など．
- **構造テスト**：プログラムの内部構造に基づいてテストケースを設計する方法．パステスト法など．

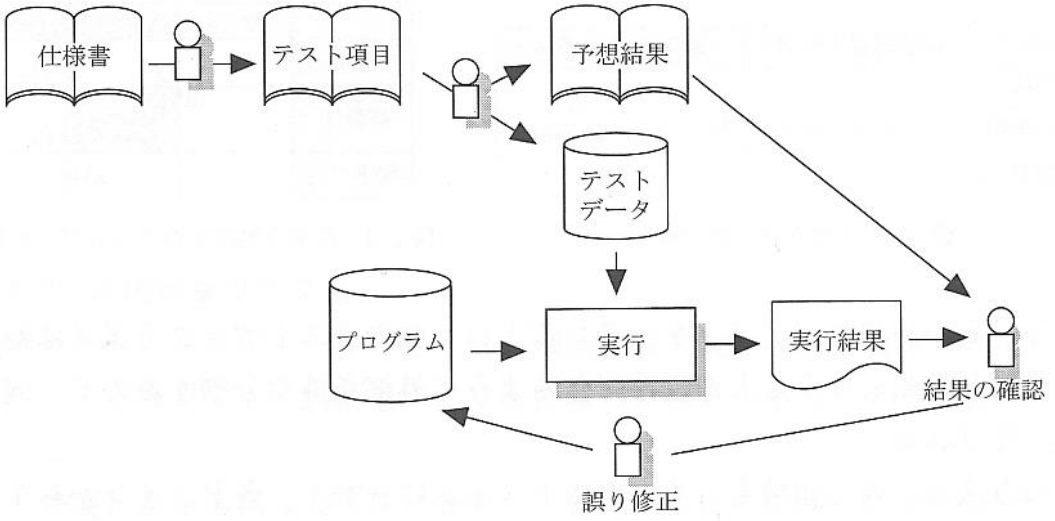
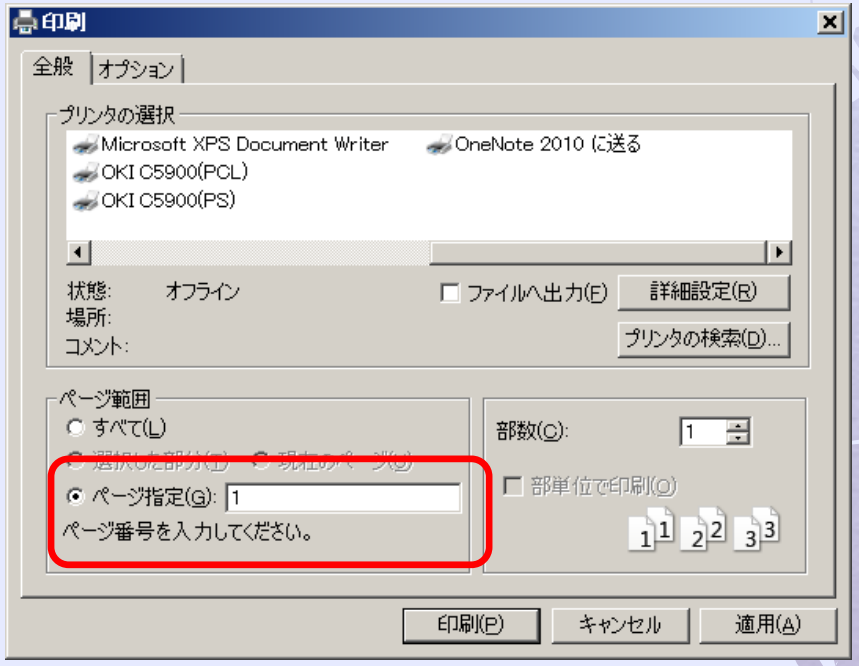


図 7.1　動的テストの実行手順

7.2.1 同値分割法

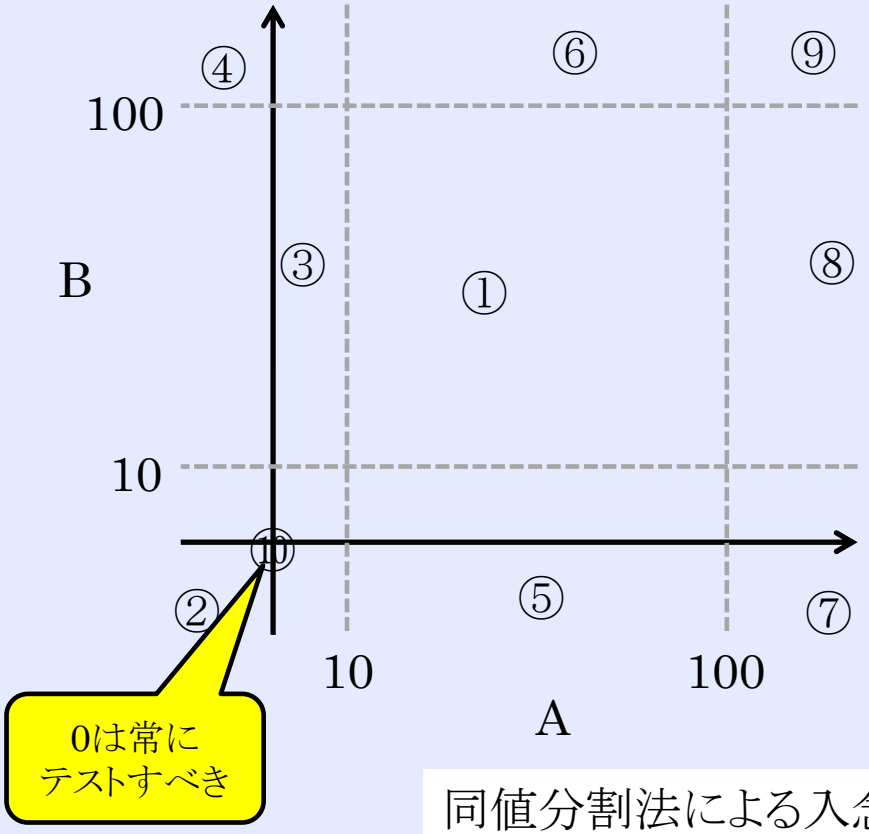
- プログラムの入力領域を**同値クラス**(プログラムが同じ振る舞いを行うとみなすことができる入力値の集合)に分割し，各同値クラスの代表値をテストデータとする手法.
- プログラムが期待する入力値の同値クラスを**有効同値クラス**，それ以外を**無効同値クラス**という.

例題)
特定のページ(1ページだけ)を指定して印刷する右図の機能の有効同値クラスと無効同値クラスはどのように定義できるか．また，テストデータとしてどのような値が考えられるか.



7.2.1 同値分割法

- 入力Aと入力B(それぞれ10以上100以下の整数の入力を期待)をとるプログラムを仮定すると・・・



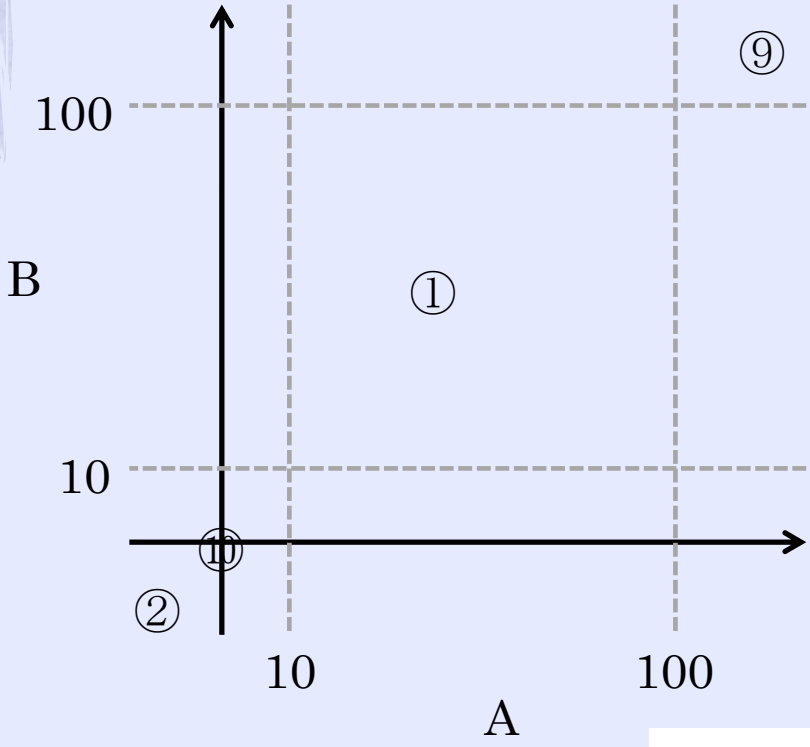
＜有効同値クラスのテストデータ＞
① A=30, B=40

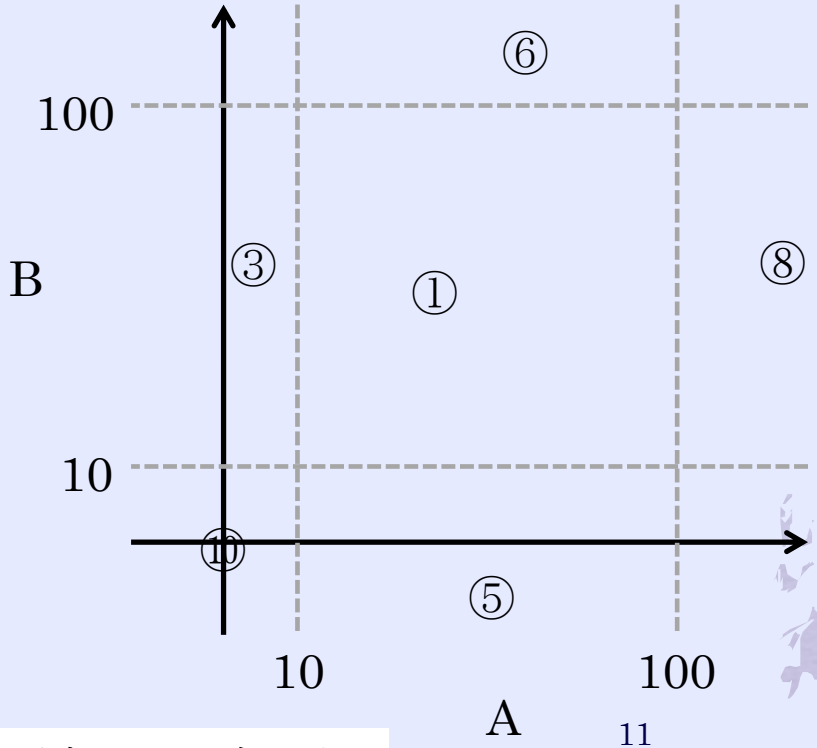
＜無効同値クラスのテストデータ＞
② A=-10, B=-5
③ A=4, B=60
④ A=-5, B=105
⑤ A=50, B=-3
⑥ A=60, B=120
⑦ A=110, B=-5
⑧ A=120, B=70
⑨ A=350, B=450
⑩ A=0, B=0

同値分割法による入念なテストデータの例

7.2.1 同値分割法

- テストデータ数を減らす場合，**各同値クラスを少なくとも1回はテストできるように**すること．
- 無効同値クラスについては個別に対応するテストデータを作成するとよい．

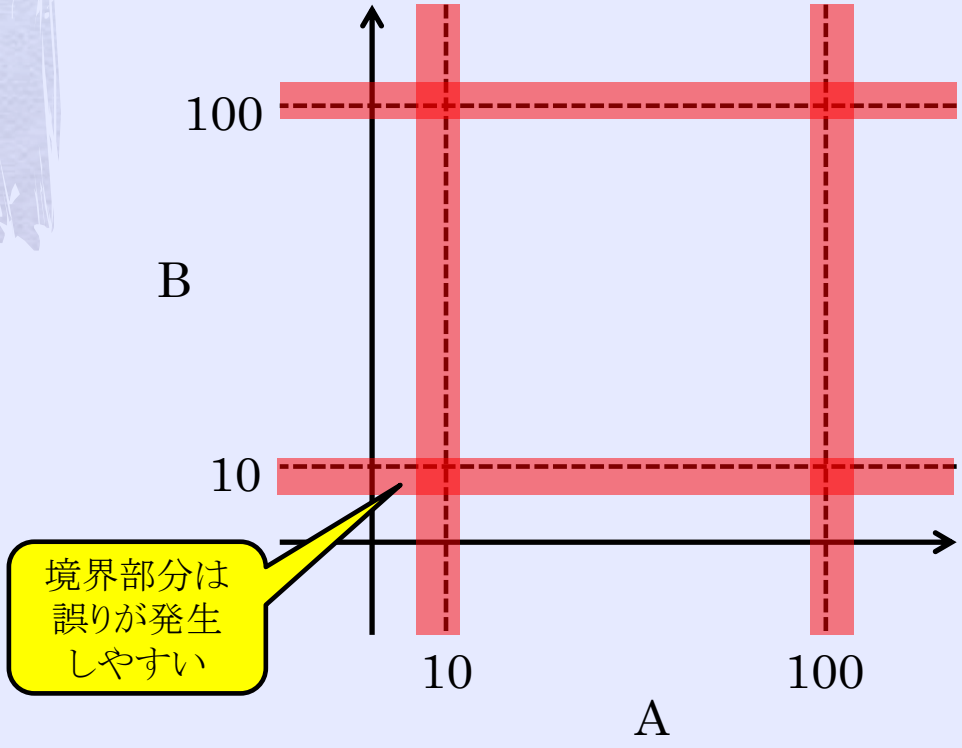




テストデータを減らす場合の例

7.2.1 限界値分析法

- テストデータを，誤りが発生しやすい同値クラスの境界部分（限界値，および限界値から最小単位分だけずらした値）とする方法．



本来

```
If (A>=10 && A<=100)
    // 処理1
else
    // 処理2
```

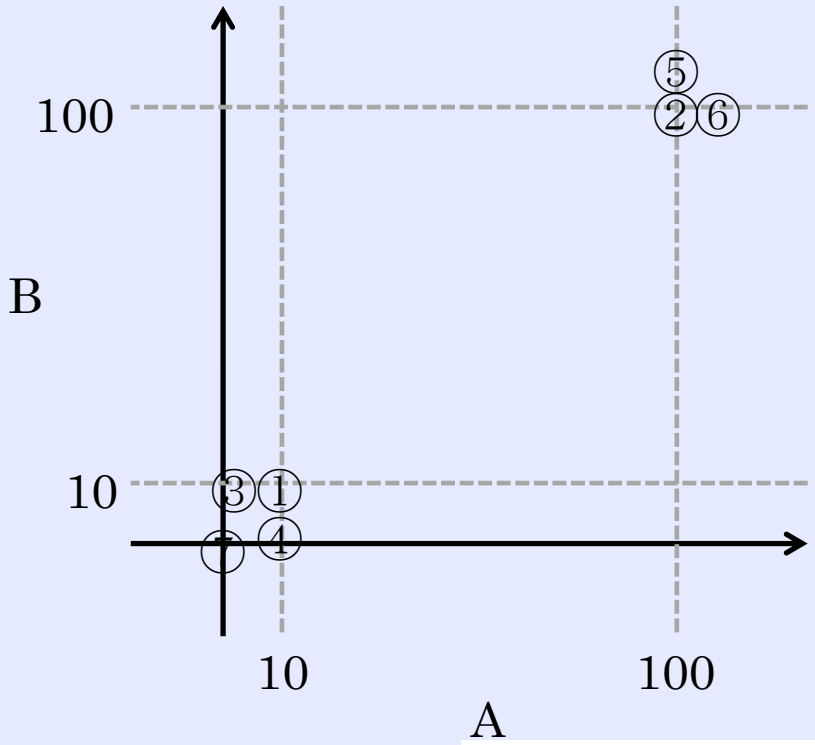
とすべきところを

```
If (A>=9 && A<100)
    // 処理1
else
    // 処理2
```

などとしてしまう誤りを効果的に発見できる．

7. 2. 1 限界値分析法

- 有効同値クラスについては1つのテストデータでできるだけ多くの同値クラスを網羅するように，無効同値クラスについては個別に対応するテストデータを作成する.



＜有効同値クラスのテストデータ＞
① A=10, B=10
② A=100, B=100

＜無効同値クラスのテストデータ＞
③ A=9, B=10
④ A=10, B=9
⑤ A=100, B=101
⑥ A=101, B=100
⑦ A=0, B=0

限界値分析法によるテストデータの例

7.2.2 原因結果グラフ法

- 機能仕様から組み合わせ論理でテスト項目を作成する

(1) 機能仕様を分割し，それぞれについて原因（入力条件）と結果（期待出力）を識別

(2) 原因を入力，結果を出力とする原因結果グラフを作成する：図7.4

(3) このグラフを規則の下で決定表に変換

(4) 決定表の各列をテスト項目に変換する：表7.1

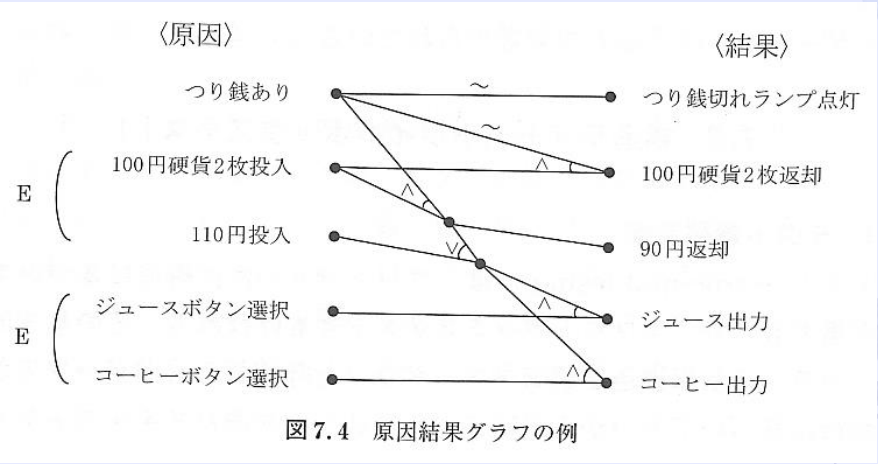


図7.4 原因結果グラフの例

表7.1 原因結果グラフに基づくテストケースの決定表

	原因／結果	テスト項目					
		1	2	3	4	5	6
入力	(n1) つり銭あり		×	×		×	×
	(n2) 100円硬貨2枚投入	×		×		×	
	(n3) 110円投入				×		×
	(n4) ジュースボタン選択		×	×	×		
	(n5) コーヒーボタン選択					×	×
出力	(x1) つり銭切れランプ点灯	×			×		
	(x2) 100円硬貨2枚返却	×					
	(x3) 90円返却			×		×	
	(x4) ジュース出力			×	×		
	(x5) コーヒー出力					×	×