

(2) 同値分割と限界値分析

テスト対象のプログラムの入力データが取りうる値を、いくつかのクラスに分割して、各クラスを代表するような値をテストデータとする。この方法に同値分割と限界値分析がある。

表 8-2 にクラス分けの例を示す。なお、正しくない値のテストデータのクラスを無効同値クラス、正しい値のテストデータのクラスを有効同値クラスと呼ぶ。同値分割では、各クラスを代表するような値をテストデータとして選び、限界値分析では、同値クラスの端値（境界値）をテストデータとして選択する。図 8-2 の例では、たとえば次のような値をテストデータとして選択する。

- 同値分割 : たとえば-2, 3, 9をテストデータとして選択する。
- 限界値分析 : たとえば0, 1, 6, 7をテストデータとして選択する。

表 8-2 入力データのクラス分けの例

エラーデータ (無効同値クラス)	正常データ (有効同値クラス)	エラーデータ (無効同値クラス)
-4, -3, -2, -1, 0	1, 2, 3, 4, 5, 6	7, 8, 9, 10, 11

(3) 因果グラフ

(a) 因果グラフによるテストケース作成手順

因果グラフ（原因-結果グラフともいう）は、設計書に記述された内容をテストするためのテストケースを導くための人手による技法である。すなわち、原因と結果の論理的な関係を論理を人手で解析し、グラフを用いて形式的に記述し、テストケースをデシジョンテーブルとして表現し、入力値を選択する。手順は以下のとおりである。

- ① 設計書から入出力を識別する。
- ② 入出力の定義域（ドメインと呼ぶ）を分析する。
- ③ 入出力ドメインに識別番号を付ける。
- ④ 入出力の論理的な関係や制約条件を、表 8-3 に示すような記法によりグラフで表現する。
- ⑤ グラフをデシジョンテーブルで表現する。
- ⑥ デシジョンテーブルによりテストの入力値を決定する。

表 8-3 因果グラフの記号

種類	記号	意味
含意(→)		AならばBが発生する。
否定(Not)		Not AならばBが発生する。
和(Or)		A Or BならばCが発生する。
積(And)		A And BならばCが発生する。
Neither Or		Neither A Nor BならばCが発生する。
Not And		Not (A And B)ならばCが発生する。
排他的制約 (Exclusive)		A または B のいずれかが発生する。すなわち A ならば B が発生し、A ならば B は発生しない。
包含的制約 (Inclusive)		A, B のうち、少なくとも1つは発生する。
唯一制約 (Only One)		A, B のうち、1つだけが発生する。
要求制約 (Require)		A のとき、必ず B も発生する。
マスク制約 (Mask)		A は B にマスクをかける。

(b) 因果グラフによるテストケース作成例

[機能仕様] 文字列の置換

① 入力仕様

- ・ 1文字以上の長さの検索文字列を指定する。
- ・ 0文字以上の置換文字列を指定する。0文字のときは、検索文字列を削除することを示す。
- ・ 以上の文字列を指定した後、「置換」ボタンをクリックする。

② 処理内容

- ・ 現在カーソル位置がある位置から後の文字列を検索する。
- ・ 現在行に検索文字列がない場合、後続する行の検索を行い、最後の行まで検索する。
- ・ エラーの場合、何もしない。

③ 出力仕様

- ・ 最後まで検索文字列が見つからなかったとき「文字列が見つかりませんでした」と表示する。
- ・ 検索文字列の長さが0文字のとき「検索文字列を指定して下さい」と表示する。
- ・ 正常に置換が終わると「置換が終わりました」と表示する。

[ドメイン分析]

機能仕様のドメイン分析を行い、識別番号を付けると次のようになる。

① 入力(原因)

- I1 : 「置換」ボタンのクリック
- I2 : 1文字以上の検索文字列
- I3 : 0文字の置換文字列 (空文字列)
- I4 : 1文字以上の置換文字列
- I5 : カーソル位置より後に検索文字列を含む文字列がある。

② 出力(結果)

- O1 : メッセージ「検索文字列を指定して下さい」の表示。
- O2 : メッセージ「文字列が見つかりませんでした」の表示
- O3 : 検索文字列を置換文字列で置換したあと、メッセージ「置換されました」の表示。

[因果グラフの作成]

因果グラフで表現すると図8-3のようになる。

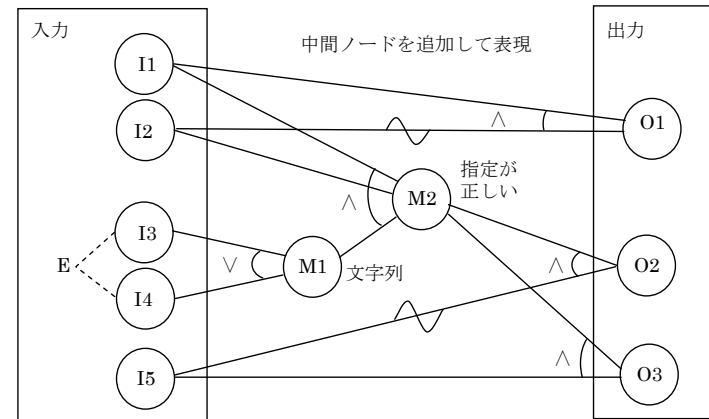


図 8-3 因果グラフ

[デジジョンテーブルの作成]

出力を中心として入力側の論理式を作成し、積和形式に変換し、これをデジジョンテーブルに直すことでテストケースを生成することができる。

$$O1 = I1 \wedge \text{not } I2$$

$$O2 = I1 \wedge I2 \wedge (I3 \vee I4) \wedge \text{not } I5$$

$$= (I1 \wedge I2 \wedge I3 \wedge \text{not } I5) \vee (I1 \wedge I2 \wedge I4 \wedge \text{not } I5)$$

$$O3 = I1 \wedge I2 \wedge (I3 \vee I4) \wedge I5$$

$$= (I1 \wedge I2 \wedge I3 \wedge I5) \vee (I1 \wedge I2 \wedge I4 \wedge I5)$$

表 8-4 テストケースのデジジョンテーブル

出力 ノード番号	テストケース番号					
	1	2	3	4	5	
入力 (原因)	I1	Y	Y	Y	Y	Y
	I2	N	Y	Y	Y	Y
	I3		Y		Y	
	I4			Y		Y
	I5		N	N	Y	Y
出力 (結果)	O1	X				
	O2		X	X		
	O3				X	X