

# PictMaster ユーザーズマニュアル

2012年3月27日      第2.0版      v5.4対応

## 本マニュアルの読み方

### 1. とにかく早く使ってみたい方へ

詳しいことは後でよいからとにかく早く使ってみたいという方は、少なくとも第0章と第3章を読んでから使ってください。デフォルトの **PictMaster** の設定は必要最小限の機能しか動作しない設定になっています。**PictMaster** を有効に使うためには、後でもよいですから本マニュアル全体を読まれることをお勧めします。

### 2. 生成結果の例について

説明の例として示されている組み合わせ生成結果の表および組み合わせ数は、**PictMaster** の最小テストケース生成の機能を用いて生成したものです。最小テストケース生成の機能は、生成結果の組み合わせ再現性を保証していません。したがってユーザの方が同じモデルで最小テストケース生成を行った場合、本マニュアルで示されている生成結果とは異なる生成結果となる場合があります。

### 3. **PictMaster** を業務で使う前に

**PictMaster** は組み合わせテストをはじめ、複数の種類のテストに対応したテストケースを生成するツールです。本ツールを使いこなすには基礎的なテスト技術を一通り習得していることが必須条件です。

現在は、テスト技術を解説した書籍が何種類か出版されています。基礎的なテスト技術が不足していると感じられた方は、まずこうした書籍などを通じて幅広いテスト技術の知識を学んでから **PictMaster** を業務で使用されることをお勧めします。

組み合わせテスト技法は数あるテスト技法のひとつにすぎません。組み合わせテストよりも他のテスト技法がふさわしい場合がよくあります。テスト対象の性質に合ったテスト技法を使うようにしましょう。

### ユーザーズマニュアル 更新履歴

版数	更新日	対応 Ver.	更新内容
1.0	2011.6.18	4.3	新規作成
1.1	2011.7.19	4.3.2	0. PictMaster のインストールの章に、Excel のバージョンの違いに起因する問題と対処法についての記述を追記した。
1.2	2011.11.28	5.0	3. 3 環境設定ボタンの章の内容を PictMaster 5.0 の変更に合わせて改定した。 6. 5 ショートカットメニューを使って編集する の章を追加した。
1.3	2011.12.19	5.1	3. 5. 1 拡張サブモデルの章を追記した。
1.4	2012.1.10	5.2	4. 7 ワイルドカードの使用の章で全角文字を “?” で指定する方法を追記した。
1.5	2012.1.11	5.3	「全数組み合わせと比較」の記述を削除した。 6. 6 希望する 3-way カバレッジを確保したテストケースを生成する の章を追加した。
2.0	2012.3.27	5.4	ユーザーズガイドからユーザーズマニュアルへの変更により内容を全面改定した。

更新履歴の「対応 Ver.」はこのマニュアルが対応している PictMaster の最初のバージョンを表しています。

## PictMaster 使用規定

以下の使用規定にすべて同意される場合のみ PictMaster を使用することを許可します。

1. PictMaster（以後 本ソフトと表記）はフリーソフトで自由に使用することができますが、著作権は岩通ソフトシステム株式会社にあります。
2. 本ソフトは[オープン・ソフトウェア・ライセンス v2.0](#)に基づき、本ソフト（オリジナル成果物）をもとに派生成果物を作成し、配布することができます。ただし、PICTを使用する限り、Microsoftのライセンス条項により、派生成果物を販売することは禁じられています。同様にPICTを同梱しての配布も禁じられています。
3. 本ソフトを販売して収益を得る行為を禁じます。
4. 本ソフトの著作権表示（© IWATSU System & Software Co., Ltd.）を読めないようにすることを禁じます。ただし、本ソフトをもとに派生成果物を作成し、配布する場合はこの限りではありません。
5. 本ソフトを使用したことによるいかなる損害に対しても著作権所有者は一切の責任を負いません。
6. この「PictMaster 使用規定」は予告なく変更を行なうことがあります。

## 目 次

0. PictMasterのインストール .....	- 7 -
1. はじめに .....	- 8 -
2. PictMasterの仕組み .....	- 8 -
3. PictMasterの使い方 .....	- 9 -
3. 1 「実行」 ボタン .....	- 12 -
3. 2 「整形」 ボタン .....	- 12 -
3. 3 「環境設定」 ボタン .....	- 13 -
3. 3. 1 原型シートの使い方 .....	- 18 -
3. 3. 2 制約式の最適化 .....	- 21 -
3. 4 値の並び欄への記入のしかた .....	- 23 -
3. 4. 1 1つの値に複数の名称を与えるエイリアス .....	- 24 -
3. 4. 2 機能が動作しない無効値テストの方法 .....	- 25 -
3. 4. 3 値の重み付け .....	- 26 -
3. 5 サブモデル .....	- 28 -
3. 6 拡張サブモデル（パラメータの重み付け） .....	- 29 -
3. 7 希望するカバレッジを確保したテストケースを生成する .....	- 31 -
3. 8 網羅すべき組み合わせが漏れていないかを手軽に確認する .....	- 32 -
4. 制約表への記入のしかた .....	- 34 -
4. 1 制約に関する用語の定義 .....	- 34 -
4. 2 制約表の構成 .....	- 34 -
4. 3 制約条件と制約対象の指定方法 .....	- 35 -
4. 3. 1 条件付き制約 .....	- 35 -
4. 3. 1. 1 パラメータと値との制約 .....	- 35 -
4. 3. 1. 2 パラメータとパラメータとの制約 .....	- 39 -
4. 3. 2 無条件制約 .....	- 41 -
4. 4 使用できる演算子の一覧 .....	- 44 -
4. 5 制約表の編集方法 .....	- 44 -
4. 6 ワイルドカードの使用 .....	- 45 -
5. 結果表への記入のしかた .....	- 46 -
5. 1 結果表の構成 .....	- 46 -
5. 2 一致条件の指定方法 .....	- 46 -
5. 3 使用できる演算子の一覧 .....	- 48 -
5. 4 記入上の注意事項 .....	- 48 -
5. 5 結果表のもうひとつの使い方 結果内容の連結 .....	- 49 -
6. より有効な使い方 .....	- 51 -
6. 1 PictMasterのカスタマイズ .....	- 51 -
6. 2 エラー／警告メッセージが表示された場合 .....	- 51 -
6. 2. 1 矛盾した制約をすばやく見つけるには .....	- 52 -
6. 3 画面を分割し制約表を記入しやすくする .....	- 52 -
6. 4 要因列举テストでテストケースを削減する .....	- 54 -
6. 5 パラメータのマイナスの重み付け .....	- 56 -
6. 6 ショートカットメニューを使って編集する .....	- 58 -
6. 7 テストケース数を調整するひとつの方法 .....	- 59 -
6. 8 効率のよい無効値テストを行なうには .....	- 61 -

6. 9	間違いやすい制約指定の例	- 63 -
6. 10	制約指定が無視される例	- 65 -
6. 11	時間のかかるテストケースを少なくする	- 67 -
6. 12	組み合わせにエイリアスのすべての値が出現するテストケースを生成する方法	- 69 -
6. 13	テスト対象にふさわしいテスト技法を選択する	- 72 -
7.	デシジョンテーブルテストへの適用	- 73 -
7. 1	デシジョンテーブルのモデルの作成	- 73 -
7. 2	テーブルを圧縮する	- 75 -
8.	状態遷移テストへの適用	- 77 -
8. 1	簡単な状態遷移のテストケースの生成	- 77 -
8. 2	状態遷移テストのモデルの作成	- 77 -
8. 3	状態遷移テストの組み合わせ結果	- 78 -
8. 4	状態遷移のくり返しを含むテストケースの生成	- 79 -
8. 6	繰返しを含む状態遷移テストの組み合わせ結果	- 81 -
9.	制御パステストへの適用	- 83 -
9. 1	従来の制御パステストとの違い	- 83 -
9. 2	パラメータと値を決めるには	- 84 -
9. 3	前の条件文によって次の条件文のパスが影響を受ける場合の値の決め方	- 85 -
9. 4	1つの条件文が複数の条件式を含む場合の対処法	- 88 -
9. 5	プログラムがループを含む場合の記入方法	- 90 -
9. 6	制御パステストのまとめ	- 92 -
7.	困ったときは	- 94 -
付録A	仕様	- 96 -
付録B	制限事項	- 96 -
付録C	拡張サブモデル（パラメータの重み付け）のアルゴリズム	- 97 -
付録D	希望するT-wayカバレッジを確保したテストケースを生成するアルゴリズム	- 100 -
付録E	「既存テストケースのカバレッジを表示」の使い方	- 102 -

## 0. PictMasterのインストール

### 【PictMaster を使う上で用意するもの】

- (1) PICTそのものは <http://msdn.microsoft.com/en-us/testing/bb980925.aspx> のサイトで入手できます。  
あらかじめダウンロードし、インストールしておいてください。インストール先は必ず以下のデフォルトのフォルダ内にインストールする必要があります。  
C:\Program Files      64bit OS の場合は C:\Program Files (x86)

- (2) Excel2000 以降の Excel。

### 【インストール方法】

- (1) PictMaster.zip の圧縮ファイルを開き PictMaster.xls を PC 内の任意の場所に置きます。サーバー上に置くこともできます。ただしネットワークドライブの割り当てを行っていないサーバー上に置いた場合、生成されたテストケースファイル“a.xls”、モデルファイル (PICT への入力となるファイル) “a.txt” などは PICT があるフォルダ内に作成されます。
- (2) Excel のセットアップを行いません。  
**Excel2007 より前のバージョン**では、ツール → オプション → セキュリティ → マクロのセキュリティ で「中」を選択してください。  
**Excel2007** では以下の手順を行ってください。  
Office ボタン → Excel のオプション → セキュリティセンター → セキュリティセンターの設定 → 信頼できる場所 → 新しい場所の追加 → 参照 → 任意の PictMaster の保存場所を指定してOKをクリックします。  
このとき、サブフォルダも含めて指定できます。
- (3) PictMaster.zip の圧縮ファイルに同梱されている **nkf.exe** を PICT がインストールされたフォルダ内にコピーします。

以上でインストール作業は終了です。

### 【重要な注意点】

PictMaster は日本語が使用できますが、スペースについては、必ず半角スペースを使用してください。

### 【Excel のバージョンについて】

PictMaster の Excel ファイルは、Excel2000 から Excel2010 まで使用できます。ファイル形式は「Excel 97-2003 ブック」です。Excel2007 以降の Excel で使用する場合は、Excel の互換モードでの使用となります。もしも、今後 **Excel2003 以前の Excel** を使用することはないと分かっている場合は、Excel2007 以降の Excel でファイル形式を「Excel マクロ有効 Book」に変換すればファイルサイズを約半分に削減することができます。Microsoft Office ボタン（左上の丸いボタン）から「変換」を選び、ファイルの種類に「Excel マクロ有効 Book」を指定して保存してください。ファイル形式が変換されます。

## 1. はじめに

**PictMaster** は Pairwise 法(All-Pairs 法ともいう)を採用した組み合わせテストケース生成を行なう Microsoft のフリーソフトである **PICT** (Pairwise Independent Combinatorial Testing Tool) をより使いやすく、より高機能にした Excel ベースのフリーソフトです。

**PictMaster** を公開する目的は **PICT** という非常に優れた組み合わせテストケース生成ツールを Excel 上で簡単に使用することができるようにすることによって、完全に無償のツールとして多くの人に使ってもらいたいからです。直交表をベースにした非常に優れたツールは存在しますが、公開されておらず、自作することはコスト的にも技術的にもかなり困難です。

このような状況を少しでも改善することを目的として **PictMaster** を公開するものです。

## 2. PictMasterの仕組み

**PICT** そのものはコマンドプロンプト上で動作する CUI (キャラクタユーザインターフェース) ベースのアプリケーションです。今となってはコマンドプロンプトになじみのない人が大半です。コマンドプロンプト上で動作する **PICT** に抵抗感を感じる方も少なくないと思います。コマンドプロンプト上で動作させて、テスト仕様書にテストケースとして組み込むまでに文字コードを 2 回変更し、Excel でファイルを読み込むなどいろいろな作業をしなければなりません。このように、**PICT** そのもののだけでテストケースを作成しようとすると、かかる手間が無視できません。

Excel でテスト仕様書を作成しているのなら、Excel 上で組み合わせテストケースも生成できたらとても便利になります。これを実現したのが Excel の Book である **PictMaster** です。**PictMaster** は、CUI ベースの **PICT** に Excel の GUI (グラフィカルユーザインターフェース) ベースの皮をかぶせます。イメージ的には図 2-1 のようになります。

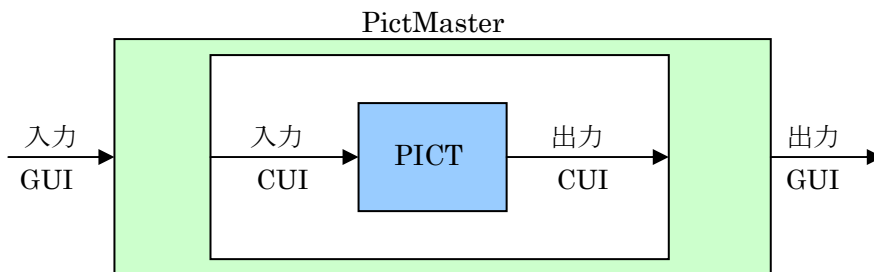


図 2-1 PictMaster のイメージ

図 2-1 に示すように、ユーザからは **PICT** の存在はまったく見えません。GUI ベースですべての作業を行なうことができます。

**PictMaster** は次に示す 5 つのソフトの連携で動作します。

- (1) Excel の VBA
- (2) コマンドプロンプト
- (3) バッチファイル
- (4) nkf
- (5) **PICT**

**VBA** (Visual Basic for Application) は、Excel 用のプログラミング言語 (Visual Basic) です。**PictMaster** では **VBA** を使用することで Excel のさまざまな GUI をコントロールします。またモデルファイルの作成、バッチファイルの作成、コマンドプロンプトの起動およびバッチファイルの実行も行ないます。さらにユーザの指定に応じて生成結果の並び替え、罫線を描くなどの処理も行ないます。



コマンドプロンプトのバッチファイルは、nkfの実行と PICTの実行を行ないます。

**nkf**は、モデルファイルと PICT が出力したファイルの文字コードの変換を実行します。**nkf**はオープンソースのソフトウェアを扱うサイトである [sourceforge.jp](http://sourceforge.jp) で公開されているフリーソフトです。**nkf** の URL は以下のとおりです。

<https://sourceforge.jp/projects/nkf/>

**PICT** はモデルファイルの構文解析と組み合わせ生成エンジンの役割を果たします。

### 3. PictMasterの使い方

PictMaster は、Excel 2000 以降の Excel で動作します。Windows 7、Windows Vista、Windows XP、Windows 2000 での動作を確認しています。PictMaster を使用するためには以下のものを用意します。

- (1) PICT そのもの
- (2) Excel 2000 以降の Excel

**PICT** は以下のサイトからダウンロードすることができます。

<http://msdn.microsoft.com/en-us/testing/bb980925.aspx>

PictMaster のデフォルト画面イメージの例を図 3－1 に示します。

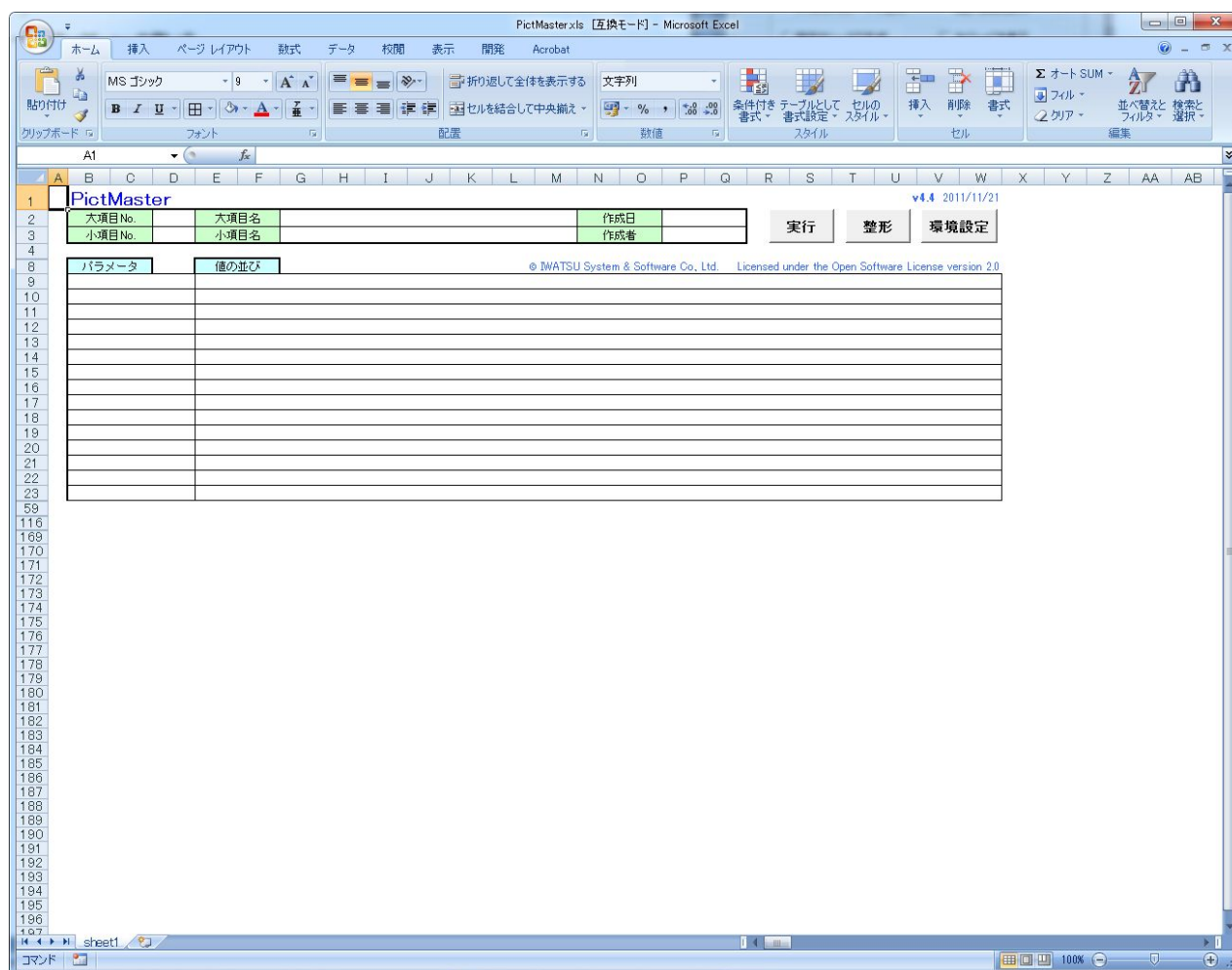


図 3－1 PictMaster のデフォルトの画面イメージ

PictMaster は以下の各部分からなっています。

## 1 ～ 7 行目 フリーエリア

ユーザが任意にレイアウト可能なエリアです。テスト大項目番号、小項目番号、作成日、作成者など、実際にユーザが使いやすいようにレイアウトを決めてください。なお 5 ～ 7 行目は非表示になっているため、そのエリアを使いたい場合は書式メニューから行の再表示を行なってください。

デフォルトのフリーエリアのレイアウトを図 3-2 に示します。

1	PictMaster				
2	大項目 No.		大項目名		作成日
3	小項目 No.		小項目名		作成者
4					

図 3-2 デフォルトのフリーエリアのレイアウト

## 9 ～ 58 行目 パラメータ欄と値の並び欄

パラメータと、値の並びをカンマ ( , ) で区切って記入します。パラメータと値の並び欄は 50 行で固定です。デフォルトでは 16 行目以降は非表示となっています。値の並び欄には 50 個までの値を記入することができます。いずれの欄も行を開けずに詰めて記入してください。値にはエイリアス記号 ( | )、無効値記号 ( ~ ) および重みづけ指定の ( n ) を付加することができます。

この欄を編集する際の注意点があります。行の削除、挿入は行を行わないで下さい。かわりに行のクリア、コピーと貼り付けで対応してください。コピー元の行と貼り付け先の行が重なると正しく貼り付けされません。コピー先が重ならないようあらかじめ間をあけておくようにしてください。A 列の部分を右クリックすると編集専用のショートカットメニューが表示され、行の移動、行の挿入、行の削除および元に戻す、を行なうことができます。パラメータ、値の並び欄の例を図 3-3 に示します。

パラメータ	値の並び
A	a1, a2, a3
B	b1, b2, b3, b4
C	c1, c2, c3
D	d1, d2, d3, d4
E	e1, e2, e3

図 3-3 パラメータ、値の並び欄の例

※パラメータ欄の最初の行（9 行目）には半角大文字の“ID”で始まる名称は使用しないでください。

## 61 ～ 62 行目 サブモデル欄

デフォルトの状態では非表示となっており、記入内容は無効です。サブモデル欄は後述する環境設定フォームで「サブモデルを使用」をチェックすることで表示され、記入内容が有効となります。任意の数のサブモデルを記入することができます。サブモデル欄の例を図 3-4 に示します。

サブモデルについては [3. 5 章](#) で説明します。

サブモデル
B, C, 2

図 3-4 サブモデル欄の例

## 6 6～9 5行目 制約表欄

デフォルトの状態では非表示となっており、記入内容は無効です。制約表欄は後述する環境設定フォームで「制約表を使用」をチェックすることで表示され、記入内容が有効となります。制約表は組み合わせできない組み合わせ（制約）を除外するために使用します。制約の内容を表形式で記入します。50行まで用意されています。デフォルトでは16行目以降は非表示となっています。この欄を編集する際は、パラメータ、値の並び欄と同じ注意事項があります。

制約表欄の例を図3-5に示します。

制約表			
パラメータ	制約1	制約2	制約3
A	a1	a2, a3	
B	b1, b2	b3, b4	
C			c1
D			d2
E			e2

図3-5 制約表の例

制約表への記入のしかたは次の[第4章](#)で説明します。

## 1 1 9～1 6 8行目 結果表欄

デフォルトの状態では非表示となっており、記入内容は無効です。結果表欄は後述する環境設定フォームで「結果表を使用」をチェックすることで表示され、記入内容が有効となります。結果表は組み合わせ内容に応じてあらかじめ期待する結果（結果内容）を記入する表です。50行まで用意されています。デフォルトでは16行目以降は非表示となっています。結果表を使用するとテストケース生成後、自動的にテストケースごとの結果内容欄に期待する結果が設定されます。

結果表の例を図3-6に示します。

結果表			
結果内容	A	B	C
aaaとなる	a1	b1	
aaaとなる		b3, b4	c1
bbbとなる	#a1		#c1
cccとなる			

図3-6 結果表の例

結果表への記入のしかたは[第5章](#)で説明します。

## 2～3行目 「実行」、「整形」、「環境設定」ボタン

デフォルトのレイアウトでは、2～3行目の右端に図3-7に示す3つのボタンがあります。



図3-7 3つのボタン

### 3.1 「実行」ボタン

パラメータ、値の並び欄などに必要な記入を行なった後に、このボタンを押すことでテストケースが“a.xls”という Book 名で作成されます。どのような条件でテストケースを生成するかを「環境設定」ボタンで指定します。また、生成済みテストケースの N-way カバレッジ (パラメータ間の組み合わせ網羅率) を表示する際にも使用します。

### 3.2 「整形」ボタン

テストケースが生成された後で、行の並び替え、罫線を描く、など指定した条件でテストケースの形を整えることができます。「整形」ボタンをクリックすると、図3-8の例のようなフォームが表示されます。

The image shows a Windows-style dialog box titled '整形' (Format). It has a close button (X) in the top right corner. The dialog is divided into two main sections. The top section is titled '値の並び替え' (Value Sorting) and contains a checked checkbox '値の並び替えを行う' (Perform value sorting), two radio buttons '昇順' (Ascending) and '降順' (Descending), and three dropdown menus labeled 'A', 'B', and 'C' with corresponding labels '最優先するキー' (Highest priority key), '2番目に優先するキー' (Second priority key), and '3番目に優先するキー' (Third priority key). The bottom section contains three checked checkboxes: '罫線を描く' (Draw grid lines), '行番号列を追加' (Add row number column), and '列幅を合わせる' (Adjust column width). At the bottom of the dialog are two buttons: 'OK' and 'キャンセル' (Cancel).

図3-8 「整形」ボタンのクリックで表示されるフォーム

並べ替えのキーには先頭から3つのパラメータが選択され、罫線を描く、行番号列を追加、列幅を合わせる、にすべてチェックが入れます。すでに行番号列が追加されている場合は「行番号列を追加」にはチェックが入りません。もちろん、ユーザが好きなように手直しすることもできます。「OK」ボタンのクリックで処理が行なわれます。

生成結果を整形ボタンで整形した結果の例を表3-1に示します。

表 3-1 整形されたテストケースの例

No.	A	B	C	D	E
1	a1	b1	c2	d1	e2
2	a1	b2	c1	d2	e2
3	a1	b2	c1	d1	e3
4	a1	b3	c2	d1	e1
5	a1	b4	c1	d1	e3
6	a2	b1	c1	d2	e3
7	a2	b2	c2	d1	e2
8	a2	b3	c1	d2	e1
9	a2	b3	c1	d2	e3
10	a2	b4	c2	d2	e2
11	a3	b1	c2	d1	e2
12	a3	b2	c2	d2	e2
13	a3	b3	c2	d2	e2
14	a3	b4	c1	d2	e1
15	a3	b4	c2	d1	e1

### 3. 3 「環境設定」 ボタン

このボタンをクリックすると図 3-9 のフォームが表示されます。

図 3-9 環境設定ボタンのクリックで表示されるフォーム

「自動整形を実行」にチェックを入れて、テストケースの生成を行なうと、テストケースが生成された後、自動的にテストケースの整形が行なわれます。この際の整形の条件は、図 3-8 と同様な条件で行なわれます。異なる条件でテストケースの整形を行ないたい場合は、チェックを外し、テストケースが生成

されてから「整形」ボタンをクリックして任意の条件を指定してください。

「**制約表を使用**」にチェックを入れると先頭から15行について、50制約の制約表欄が表示され、記入された制約指定が有効となります。制約表欄は50行設けていますので15行で不足する場合は残りの行を再表示させてください。詳細は[第4章](#)で説明します。

「**結果表を使用**」にチェックを入れると15行の結果表欄が表示され、記入された条件が有効となります。結果表欄は50行設けていますので15行で不足する場合は残りの行を再表示させてください。詳細は[第5章](#)で説明します。

「**原型シートを使用**」にチェックを入れてテストケースの生成を行なうと、すぐ右隣のシートを原型シートとして扱います。詳細は次の[3. 3. 1章](#)で説明します。

すぐ右隣にシートがない、あったとしても原型シートの形式と異なる形式の場合はエラーメッセージが表示されます。原型シートの1桁目が**ダラマーク (\$)** の場合その行は**コメント行**として扱われます。原型シートは空白行を含まず詰めて記入してください。コメント行を除いた最初の行にはすべてのパラメータを記入します。原型シート上の値のチェックは **PictMaster** では行なっていません。原型シートには65535件までのテストケースを記入できます。

「**モデルファイルを表示**」にチェックを入れて、テストケースの生成を行なうと、テストケースが生成された後、**PictMaster** がパラメータ欄、値の並び欄、制約表などをもとに生成し、**PICT** に渡したモデルファイル “a.txt” がメモ帳によって表示されます。

「**サブモデルを使用**」にチェックを入れるとサブモデル欄が表示され、記入されたサブモデルが有効となります。同時に次の「**拡張サブモデルを使用**」のチェックボックスが有効となります。詳細は[3. 5章](#)で説明します。

「**拡張サブモデルを使用**」にチェックを入れるとサブモデル欄が表示され、記入された拡張サブモデルが有効となります。このチェックボックスは「サブモデルを使用」にチェックを入れると使用可能となります。詳細は[3. 6章](#)で説明します。

「**統計情報を表示**」にチェックを入れると、最小テストケース生成が完了した時点で図3-10の例に示す生成回数、最小数、最大数、初期数、最小時シードおよびテストケース生成にかかった経過時間が表示されます。こうした表示はデフォルトのシードを指定して1回だけ生成する場合も行なうことが可能です。

初期数の値は、**PICT**のデフォルトの生成結果を表します。最小時シードは、最小テストケースを生成したシードを表します。

ランダムな条件で生成した場合、生成されるテストケース数には最大で10%程度のバラツキが発生し、多くの場合、最小テストケース生成を行なうことにより5%程度テストケース数を減らすことができます。生成回数を増やすほど、テストケース数を減らせる確率が高くなりますが、ほとんどの場合、30回程度行えば充分のようです。1件でもテストケース数を減らしたい場合は、生成回数に100～300程度の値を入力して最小テストケース生成を行なってみてください。

このチェックを入れると同時に次の「**カバレッジを表示**」のチェックボックスが有効となります。



図 3－1 0 表示される統計情報の例

「カバレッジを表示」にチェックを入れると、テストケース生成時に生成されたテストケースの N-way カバレッジ（N パラメータ間の組み合わせ網羅率）と T-way カバレッジが図 3－1 1 のように表示されます。N は「組み合わせるパラメータ数」で指定した値です。T は N に＋1 した値です。

「組み合わせるパラメータ数」に 6 以上の値が設定されている場合は、カバレッジの表示は行なわれません。

N-way カバレッジを算出するために、N パラメータ間の組み合わせで 1 回生成を行ないます。また T-way カバレッジを算出するために T パラメータ間の組み合わせで 1 回生成を行ないます。そのため、大きなモデルでは結果が表示されるまでに時間がかかる場合があることに注意してください。



図 3－1 1 表示されるカバレッジの例



このチェックボックスは「統計情報を表示」にチェックを入れると使用可能となります。

「組み合わせるパラメータ数」には1～50までの数字を指定します。「実行」ボタンのクリックで、指定されたパラメータ数での組み合わせが生成されます。大きな数を指定した場合、テストケース生成に非常に長い時間がかかる場合があることに注意してください。

「制約式を最適化」にチェックを入れてテストケースの生成を行なうと、パラメータの値の数が多い場合に制約指定が適切でないため、テストケース生成に時間がかかると判断されると、プログラム内部で制約指定の記入形式を最適化し、より短時間で生成が完了するようにします。詳細は[3. 3. 2章](#)で説明します。このチェックボックスは「制約表を使用」がチェックされている場合に使用可能となります。

「セル書式は文字列」にチェックを入れてテストケースの生成を行なうと、生成結果のセルの書式が文字列として生成されます。このことにより、先頭が0で始まる値（001、010 など）であってもゼロサプレスされることなくそのまま出力されます。ただし、値が文字列として扱われるため、生成結果の並び替えでは、値の大小だけでなく、文字の長さも含めて並び替えが行なわれるようになります。チェックを入れていないと全角の数字だけからなるパラメータや値は半角の数字として扱われます。

「設定を常時表示」にチェックを入れると環境設定フォームの主な設定内容が、画面右上に常時表示されるようになります。これで設定内容を確認するためにいちいち環境設定フォームを開かなくて済みます。この表示内容のフォントの色やサイズ、太字にするなどは自由にカスタマイズすることができます。

「K-way 組み合わせと比較」にチェックを入れると、テストケース生成と同時に「対象パラメータ数」で指定されたパラメータ間の組み合わせをファイル k.xls で生成します。生成された組み合わせは、生成されたテストケースで網羅されていない組み合わせが灰色で塗りつぶされます。同時に Excel のフィルタが設定されます。

「対象パラメータ数」での組み合わせのうち、生成したテストケースが、どの組み合わせを網羅していないかを簡単に把握することができれば、網羅すべき重要な組み合わせが網羅されていないことを見つけ出すことができます。

「K-way組み合わせと比較」では、色分けが行なわれるため、網羅すべき重要な組み合わせが漏れていないかを、Excelのフィルタ機能を使って容易に確認することができます。この機能の使い方を[3. 8章](#)で説明しています。

「対象パラメータ数」に大きな値を指定すると、モデルの大きさによっては組み合わせ数が発散してしまう場合があることに注意してください。いくら待っても処理が完了せず、中止したいときはタスクマネージャを起動し、PICTのプロセスを終了させてください。

「K-way 組み合わせと比較」とサブモデルを同時に指定することはできません。

「ウインドウ分割ショートカットキー」の入力欄に任意の半角英文字または半角数字1文字を入力しておくと、コントロールキーと入力したキーを押すことでPictMasterのウインドウが2つ開かれ、上下に整列され、下側のウインドウはパラメータ欄と制約欄との間で分割されます。この機能は多くの制約がある場合に制約表への記入をやりやすくするためのものです。結果表への記入時にも使えます。詳細は[6. 3章](#)で説明します。

「最小テストケースを生成」を選択すると、テストケース生成の際、ランダムな条件（シード）でPICTを実行し、最もテストケース数の少ないテストケースを生成結果として出力します。「生成回数」で何回テストケース生成を行なうかを指定します。2から999回まで指定できます。デフォルトは30回です。

PICTは内部で固有のシード（デフォルトは0）を使用してテストケースの生成を行なっています。このシードを変えることにより、生成される組み合わせ数が若干違ってきます。



最小テストケース生成実行中は、図 3-12 の例に示すプログレスバーが表示されます。



図 3-12 最小テストケース生成中のプログレスバーの例

「デフォルトのシードで生成」を選択すると、テストケース生成の際、デフォルトのシード（0）で PICT を実行します。

「特定のシードで生成」を選択すると、テストケース生成の際、「シード」欄に記入されたシードで PICT を実行します。0 から 6 5 5 3 5 までの値を設定することができます。最小テストケース生成を実行すると、最もテストケース数が少なかったときのシードが「シード」欄に設定されます。

「カバレッジを指定して生成」を選択すると、希望する T-way カバレッジを確保したテストケースを生成結果として出力します。T は「組み合わせるパラメータ数」で指定されている値 N に + 1 した値です。希望する T-way カバレッジを「(T)-way カバレッジ」の欄に記入します。30%～95% の範囲内で指定することができます。希望するカバレッジを確保したテストケースを出力するために、何回かテストケースの生成またはテストケースの変更を繰り返す必要があります。「生成試行回数」で繰り返しを何回行なうかを指定します。3 回～10 回の範囲内で指定することができます。この機能の有効な使い方を [3.7 章](#) で説明しています。

この機能は「組み合わせるパラメータ数」が 1～5 の範囲内である場合に使用することができます。この機能は PictMaster の原型シートの機能を利用しています。環境設定で「原型シートを使用」が指定されている場合、「カバレッジを指定して生成」は指定できません。

この機能は原型シートの機能を使用するため、Excel のワークシート名の番号が増加します。例えばこの機能を 3 度実行すると、Excel のワークシートは Sheet4 まで使用されます。この状態でユーザが Excel の新規ワークシートを作成するとそのワークシート名は Sheet5 となります。ワークシートの番号を元に戻したい場合は、PictMaster をセーブしたうえで PictMaster を再度開いてください。ワークシートの番号がリセットされます。

### 【重要な注意】

希望する T-way カバレッジに低い値を指定した場合、希望する値よりもかなり高いカバレッジとなる場合があります。この現象はデフォルトのシードで生成したテストケースのカバレッジが、希望するカバレッジよりも高い場合に起こります。この場合は希望する低い T-way カバレッジとした場合に N-way カバレッジ 100% を確保できなくなるために、指定した T-way カバレッジは無効となります。

「デフォルトのシードで生成」を指定して生成した場合のカバレッジを確認することで、この現象が起きているかどうか分かります。

値の重み付けを行なっている場合は、希望するカバレッジが得られない場合があります。これは「カバレッジを指定して生成」の機能が原型シートの機能を使用していることからくる制限事項です。

繰り返し処理中は、図 3-13 に示すプログレスバーが表示されます。



図 3-13 カバレッジを指定して生成中のプログレスバーの例

「既存テストケースのカバレッジを表示」を選択すると、生成結果のファイル a.xls のカバレッジ（組み合わせ網羅率）を表示します。この指定を選択すると、「統計情報を表示」と「カバレッジを表示」も自動的に選択された状態になります。

表示するカバレッジは「組み合わせるパラメータ数」が N の場合、N-way カバレッジと N+1-way カバレッジです。N が 1～5 の範囲内である場合に表示することができます。

この指定では、結果として残すテストケースを生成することはないため、「組合せるパラメータ数」とサブモデル指定を除いて、テストケース生成に関係するすべての指定は無効となります。ファイル a.xls が開かれていない場合はエラーメッセージが表示されます。カバレッジ算出のためにモデルファイル a.txt の内容を参照しています。ファイル a.xls は既存のモデルファイル a.txt と必ず対応している必要があります。対応していない場合の動作は保証されません。この機能の詳しい使い方を[付録E](#)で説明しています。

### 3. 3. 1 原型シートの使い方

原型シートには2つの使用方法があります。

- (1) 前に使用したモデルを変更する必要がある場合、以前のモデルで作成したテストケースを再利用して、できるだけ少ない変更で新しいテストケースを作成します。
- (2) 生成されるテストケースに必ず含まれるべきである**重要な組み合わせ**を指定します。指定された組み合わせで出力が初期化され、次に残りの組み合わせが生成されます。

環境設定で「原型シートを使用」にチェックを入れてテストケースの生成を行なうと、**すぐ右隣のワークシートを原型シートとして扱います**。すぐ右隣にワークシートがない、あったとしても原型シートの形式と異なる形式の場合はエラーメッセージが表示されます。

原型シートの1桁目が**ダラーマーク (\$)** の場合その行は**コメント行**として扱われます。原型シートは空白行を含まず詰めて記入してください。原型シート上の値のチェックは PictMaster では行なっていません。

原型シートには6 5 5 3 5件までのテストケースを記入できます。原型シートは PICT によって生成されたテストケースと同じフォーマット（行番号は除く）を使用します。コメント行を除いた最初の行にはすべてのパラメータ名を記入します。PictMaster のパラメータ欄と同じパラメータがすべて記述されている必要があります。次の行以降は組み合わせられた値の並びを記入します。値が空白のパラメータがあってもかまいません。

#### 【重要な注意】

- (a) 原型シートが現在のモデルにない値を含んでいる場合、そのパラメータの値の組み合わせは行われません。現在のモデルに含まれるパラメータの値の組み合わせだけが生成されます。無効とされた値の列は不完全であるか部分的な列になります。
- (b) 原型シートの値の列が現在の制約指定のどれかに違反する場合、その列は無視されます。
- (c) 原型シートを使用する場合、**値の重み付け (3. 4. 3章)** は正常に動作しません。値の重み付けを行なって生成した結果を原型シートとして使用する場合は、PictMasterの値の並び欄から値の重み付けを取り除いてテストケースの生成を行なうことを推奨します。

- (d) 原型シートの組み合わせが 1 行ずつ PICT に読み込まれます。読み込んでいる途中ですべての N パラメータ間の組み合わせが網羅された場合、PICT はそれ以上原型シートを読み込もうとはしなくなります。N は環境設定の「組み合わせるパラメータ数」です。ただし、現在のモデルが原型シートにない値を含んでいる場合は、原型シートの組み合わせ全体が読み込まれます。
- (e) この機能は「カバレッジを指定して生成」とは同時には使用できません。

原型シートの使用例を以下に示します。

#### 例 1：新しい値を追加する

	A	B	C
1	A	B	C
2	a1	b2	c2
3	a1	b1	c1
4	a3	b2	c1
5	a2	b1	c2
6	a2	b2	c1
7	a3	b1	c2

図 3－1 4 原型シートの例（その 1）

パラメータ	値の並び
A	a1, a2, a3
B	b1, b2, b3
C	c1, c2

図 3－1 5 新しいモデルの例（その 1）

新しいモデルでは、パラメータ B に値 b3 が追加されています。

表 3－2 新しい値が追加されたテストケース

No.	A	B	C
1	a1	b2	c2
2	a1	b1	c1
3	a3	b2	c1
4	a2	b1	c2
5	a2	b2	c1
6	a3	b1	c2
7	a3	b3	c1
8	a1	b3	c2
9	a2	b3	c1

新しく生成されたテストケースには、原型シートの内容がそのまま流用され、追加した値 b3 に関する組み合わせが追加されています。これにより既存のテストケースでテストした後で仕様変更などにより、新しい値を追加してテストを行う必要が生じた場合、追加された値に関する組み合わせだけテストすればよい場合もあります。（注：表 3－2 は値の並べ替えを行っていません）

## 例 2：新しいパラメータを追加する

	A	B	C	D
1	A	B	C	D
2	a1	b2	c2	
3	a1	b1	c1	
4	a3	b2	c1	
5	a2	b1	c2	
6	a2	b2	c1	
7	a3	b1	c2	

図 3－1 6 原型シートの例（その 2）

パラメータ	値の並び
A	a1, a2, a3
B	b1, b2
C	c1, c2
D	d1, d2, d3

図 3－1 7 新しいモデルの例（その 2）

新しいモデルではパラメータ D が追加されています。

表 3－3 新しいパラメータが追加されたテストケース

No.	A	B	C	D
1	a1	b2	c2	d3
2	a1	b1	c1	d2
3	a3	b2	c1	d1
4	a2	b1	c2	d1
5	a2	b2	c1	d2
6	a3	b1	c2	d2
7	a2	b1	c1	d3
8	a3	b1	c2	d3
9	a1	b2	c1	d1

新しく生成されたテストケースには、原型シートの内容がそのまま流用され、追加したパラメータ D に関する組み合わせが追加されています。（注：表 3－3 は値の並べ替えを行っていません）

原型シートに新しく追加したいパラメータの名称を記入しておくだけで新しいパラメータを追加したテストケースを簡単に生成することができます。

例 3：必ず含まれなければならない組み合わせを指定する

	A	B	C
1	A	B	C
2	a1	b1	c1
3	a1	b1	c2
4	a1	b2	c1
5	a1	b2	c2

図 3－1 6 原型シートの例（その 2）

パラメータ	値の並び
A	a1, a2, a3
B	b1, b2
C	c1, c2

図 3－1 6 モデルの例

原型シートでは値 a1 について、他のパラメータ B と C のすべての値と組み合わせられています。このような、あるパラメータの特定の値についてのみ、ほかのすべてのパラメータが持つ値と組み合わせることは、PICT の柔軟な制約条件指定機能をもってしても指定することができません。したがって、このような特殊または複雑な組み合わせは原型シートを使って指定します。

表 3－4 指定した組み合わせが含まれたテストケース

No.	A	B	C
1	a1	b1	c1
2	a1	b1	c2
3	a1	b2	c1
4	a1	b2	c2
5	a2	b1	c2
6	a2	b2	c1
7	a3	b1	c1
8	a3	b2	c2

新しく生成されたテストケースには、原型シートの内容がそのまま流用され、その他のパラメータの組み合わせが追加されています。この例のように、制約条件では指定できない複雑で特殊な組み合わせでも、原型シートで指定することにより、簡単に扱うことができます。

### 3. 3. 2 制約式の最適化

組み合わせ生成エンジンの PICT は、組み合わせできない組み合わせを指定するのに、IF 文に似たスクリプト言語（制約式）を用います。PictMaster では、制約表への記入を PICT が理解できる制約式に変換します。PICT の特性として、値の個数が多いパラメータに制約の指定を行なうと、制約の指定方法によって組み合わせの生成時間に大きな差が出ます。「制約式の最適化」を指定すると、PictMaster は制約指定が行なわれているパラメータの値の個数を調べ、8 個以上の場合で、制約式の最適化の処理が有効な条件である場合、制約指定をプログラム内部で変更し、最も生成時間が短くなるような制約式を生成します。

制約指定の違いによる生成時間の違いを示すために図 3－1 7 のモデルを使用することにします。使用した PC 環境 は、CPU が PentiumD 2.8GHz、RAM 1 GB、OS は Windows XP Professional SP3 です。

パラメータ	値の並び
A	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
B	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
C	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
D	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
E	1, 2, 3, 4, 5, 6, 7, 8, 9, 10

図 3-17 生成時間比較用のモデルの例

このモデルに次の制約指定を行なった場合の生成時間を示します。これらの制約指定は異なる記述ですがいずれも等価な制約指定です。生成時間は最小テストケース生成を 30 回行なった場合の経過時間です。

(1) 制約指定その 1

制約表		
パラメータ	制約1	制約2
A	1, 2, 3, 4, 5	
B	1, 2, 3, 4, 5	1, 2, 3, 4, 5
C		1, 2, 3, 4, 5
D		
E		

制約式最適化=False 生成時間 = 7 秒

制約式最適化=True 生成時間 = 4 秒

(2) 制約指定その 2

制約表		
パラメータ	制約1	制約2
A	1, 2, 3, 4, 5	
B	#6, 7, 8, 9, 10	1, 2, 3, 4, 5
C		#6, 7, 8, 9, 10
D		
E		

制約式最適化=False 生成時間 = 3 秒

制約式最適化=True とはならない。(すでに最適化の形式となっている)

(3) 制約指定その 3

制約表		
パラメータ	制約1	制約2
A	#6, 7, 8, 9, 10	
B	1, 2, 3, 4, 5	#6, 7, 8, 9, 10
C		1, 2, 3, 4, 5
D		
E		

制約式最適化=False 生成時間 = 7 分 20 秒

制約式最適化=True 生成時間 = 4 秒



#### (4) 制約指定その4

制約表		
パラメータ	制約1	制約2
A	#6, 7, 8, 9, 10	
B	#6, 7, 8, 9, 10	#6, 7, 8, 9, 10
C		#6, 7, 8, 9, 10
D		
E		

制約式最適化=False 生成時間 = 7 秒

制約式最適化=True 生成時間 = 3 秒

図 3-18 制約式最適化の有無と生成時間

以上の結果から、制約指定が(3)の場合に制約式最適化の効果が最も顕著であることが分かります。それ以外では(2)の場合、最適化が行なわれないことがわかります。この制約指定では最適化が不要なためです。制約式最適化の処理が行なわれるのは以下の条件をすべて満たす場合です。

- ① 環境設定で「制約式を最適化」が指定されている。
- ② 制約条件と制約対象のパラメータに属する値の個数が共に8個以上である。
- ③ 制約対象で指定された値の個数が2個以上である。
- ④ 制約条件が逆制約か、または制約対象が順制約である。

制約指定の意味については[第4章](#)を参照してください。

PictMaster は制約式の最適化が指定されると、①から④の条件をすべて満たす制約欄がある場合に、プログラム内部でその制約欄の記述形式を(2)の制約指定その2の形式に変換します。したがって、環境設定で「制約式を最適化」が指定されていても、実際には最適化が行なわれない場合があります。また制約式の最適化が行なわれたとしても、多くの制約指定の一部についてだけ行なわれた場合は、生成時間が短縮されない場合もあります。

パラメータとパラメータの比較を行なう制約指定は最適化の対象外です。

制約式の最適化が行なわれた場合は、制約表の制約指定とは異なる制約式が生成されることになるため、万が一、PICT 内部でエラーを検出した場合、エラーメッセージとして表示される制約式と、記述されている制約指定が対応しないことがあることに注意してください。

「制約式の最適化」を指定した場合、「統計情報を表示」が同時に指定されていると、表示される統計情報に制約表の最適化が実際に行なわれたかどうかが表示されます。

### 3. 4 値の並び欄への記入のしかた

値の並びを半角のカンマ(,)で区切って記入します。値の前に特定の記号を付加することで特殊な処理を行なうことができます。特定の記号にはパイプ(|)とチルダ(~)があります。パイプはひとつの値に複数の名称を与えるエイリアスで使います。チルダは相互に組み合わせ不可の値を指定する無効値テストで使います。値の後ろに半角の括弧()で囲まれた数字を付加することで、その値が他の値より多く組み合わせに出現するようになる値の重み付けを行なうことができます。

これらの記号はパラメータおよび値の名称には使用できません。値の名称に半角の括弧を使いたい場合は代わりに半角のブラケット([])または全角の括弧などを使ってください。

### 3. 4. 1 1つの値に複数の名称を与えるエイリアス

エイリアスは、1つの値に複数の異なる名称を与える機能です。まったく同一とは言えないがほぼ同一と考えることのできる複数の値に対して、同値分割の考え方を適用し、エイリアス機能を用いることができます。これにより組み合わせ生成の際に1つの値として扱われ、組み合わせ完成後、1つの値はエイリアスで与えた複数の名称に戻されます。

エイリアスを使用することによって生成される組み合わせの数を減らすことが可能です。Pairwise法で生成される組み合わせの数は最も多くの値を持つパラメータ P1 と、P1 と同じか次に多くの値を持つパラメータ P2 の、それぞれの値の個数を V1n、V2n とすると、これらを積算した値か、それよりもやや多い値になります。生成される組み合わせ（テストケース）のおおよその数 TCn は次式で表されます。

$$TCn = (V1n * V2n + \alpha) * m$$

ここで $\alpha$ の値は、パラメータ P2 の次に多くの値を持つパラメータ P3 以降が持つ値の個数 P3n、P4n・・・によって異なります。この数が P2n と等しいかほとんど同じ場合は、多くの場合 $\alpha$ は比較的大きな値になります。逆に P2n より小さい場合は、 $\alpha$ は小さな値になります。この場合多くは $\alpha$ の値は0になります。このことから、エイリアスを適用するパラメータは、**最も多くの個数の値を持つパラメータを対象**にしたほうが、組み合わせ数を削減する効果が最も大きくなります。逆に、比較的少ない個数の値しか持たないパラメータに適用しても効果がありません。mは制約の数が0の場合1です。制約の数が増えるに従って徐々に増加しますが、増加の程度は緩やかです。

エイリアスを使用したモデルの例を図3-19に示します。この例では OS 種別と HD 容量および HD インターフェースの組み合わせをテストします。このモデルでは OS 種別の5個の値から、Windows 系3個をエイリアスの記号“|”で1つにまとめ、全体で3個にしています。エイリアスでまとめた場合、先頭の値の名称を使用することで制約などを記述することができます。

パラメータ	値の並び
OS種別	Windows Vista   Windows XP   Windows 2000, Linux, Mac OS X
HD容量	250GB, 500GB, 750GB
HDインターフェース	USB2.0, IEEE1394, eSATA

図3-19 エイリアスを使用したモデルの例

エイリアスを使用した図3-19の組み合わせ生成結果を表3-5に、エイリアスを使用しなかった場合の組み合わせ生成結果を表3-6に示します。

表3-5 エイリアスを使用した場合

No.	OS 種別	HD 容量	HD インターフェース
1	Linux	250GB	IEEE1394
2	Linux	500GB	USB2.0
3	Linux	750GB	eSATA
4	Mac OS X	250GB	USB2.0
5	Mac OS X	500GB	eSATA
6	Mac OS X	750GB	IEEE1394
7	Windows 2000	250GB	eSATA
8	Windows Vista	750GB	USB2.0
9	Windows XP	500GB	IEEE1394



表 3－6 エイリアスを使用しなかった場合

No.	OS 種別	HD 容量	HD インターフェース
1	Linux	250GB	eSATA
2	Linux	500GB	USB2.0
3	Linux	750GB	IEEE1394
4	Mac OS X	250GB	eSATA
5	Mac OS X	500GB	IEEE1394
6	Mac OS X	750GB	USB2.0
7	Windows 2000	250GB	USB2.0
8	Windows 2000	500GB	IEEE1394
9	Windows 2000	750GB	eSATA
10	Windows Vista	250GB	USB2.0
11	Windows Vista	500GB	eSATA
12	Windows Vista	750GB	IEEE1394
13	Windows XP	250GB	IEEE1394
14	Windows XP	500GB	eSATA
15	Windows XP	750GB	USB2.0

この例ではエイリアスを使用した場合、使用しなかった場合に比べてテストケース数が 3 分の 2 未満に減少しています。値の数の多いパラメータで同値と考えることのできる値がある場合は、エイリアスを積極的に使ったほうがよいでしょう。

エイリアスで 1 つにまとめた複数の値のうち、後述する制約表や結果表で指定できるのは先頭の値の名称のみです。

#### ※ 注意事項

エイリアスとして 1 つにまとめた値の数が多い場合は、生成結果に一度も現れない値が存在する可能性があります。これはエイリアスを含まないパラメータに属する値が少ない場合に発生します。エイリアスの値すべてを組み合わせに使わなくてもすべての組み合わせが網羅されてしまうからです。本来、エイリアスは同値とみなせるものを 1 つにまとめたものですから出現しない値があっても直ちに問題とは言えませんが、こうした性質があることは知っておいたほうがよいでしょう。

### 3. 4. 2 機能が動作しない無効値テストの方法

組み合わせテストでは通常、機能が動作しない無効値を含まないようにする必要があります。テストケースに無効値を含むと機能が動作しなくなり、他のパラメータと組み合わせの意味がなくなります。

ここでは無効値を含むテストの実行を目的としている場合について説明します。1 つのテストケースに無効値を 2 つ以上含むと 1 つの無効値についてのテストとなり、残りの無効値についてのテストが行われなくなり、不完全なテストとなります。機能が動作しない場合の組み合わせでは、無効値どうしの組み合わせが行われなくする必要があります。

PICT には無効値テストという機能があり、無効値どうしの組み合わせが生成されないようにすることができます。値の並び欄で値の前に半角の記号 “ ~ ” を置くことで無効値を指定します。

図 3－20 に無効値を含むモデルの例を示します。このモデルでは FAX と通信回線の組み合わせをテストします。FAX とは通信できない無効値である電話機が含まれています。さらに FAX 使用に制限がある IP 外線も無効値としています（実際には多くの場合、支障なく使用できます）。

パラメータ	値の並び
発信端末	FAX, ~電話機
通信回線	アナログ, ISDN, ~IP外線
着信端末	FAX, ~電話機

図 3－20 無効値を含むモデルの例

表 3－7 無効値テストの生成結果

No.	発信端末	通信回線	着信端末
1	～電話機	ISDN	FAX
2	～電話機	アナログ	FAX
3	FAX	～IP外線	FAX
4	FAX	ISDN	～電話機
5	FAX	ISDN	FAX
6	FAX	アナログ	～電話機
7	FAX	アナログ	FAX

表 3－7 が生成結果です。この生成結果には無効値どうしの組み合わせがありません。この例では無効値の場合でも無効値以外のすべてのペアを組み合わせています。そこまでの徹底したテストが不要な場合は、制約定義で各無効値 1 つに 1 つのテストケースのみが生成されるように指定することもできます。

値が複数の名前を持つエイリアスの場合、最初の名前に記号“～”を付けます。制約の記述で無効値を指定する場合は、記号“～”は省略します。

### 3. 4. 3 値の重み付け

特定の値を重点的にテストしたい場合、重み付けの機能が役に立ちます。重み付けを使用すると指定された値がより多くテストケースに現れるようになり、その値についてのカバレッジが向上します。重み付けは重点的にテストしたい値の右側に半角の括弧（ ） で数値を付加します。

図 3－2 1 に重み付けの例を示します。

パラメータ	値の並び
A	a1(2), a2, a3
B	b1, b2, b3, b4
C	c1, c2, c3(3)

図 3－2 1 重み付けの例

括弧内に記入できる数値は 2～10 です。基本的には記入された数値をかけた分だけ他の値より多く組み合わせに出現するようになります。例えば c3(3) を記入した場合、値 c3 は他の値 c1、c2 より 3 倍多く出現するようになります。ただし、生成された組み合わせで重複する組み合わせが存在する場合、重複する組み合わせは 1 つを残し削除されて最終的な生成結果となります。

図 3－2 1 で重複する組み合わせを含んだ生成結果を表 3－8 に示します。

表 3－8 重複する組み合わせを含んだ生成結果

No.	A	B	C
1	a1	b1	c3
2	a1	b1	c3
3	a1	b1	c3
4	a1	b2	c1
5	a1	b2	c2
6	a1	b3	c3
7	a1	b3	c3
8	a1	b4	c1
9	a1	b4	c2
10	a1	b4	c3
11	a2	b1	c1
12	a2	b2	c3
13	a2	b3	c2
14	a2	b3	c3
15	a2	b4	c3
16	a3	b1	c2
17	a3	b2	c3
18	a3	b2	c3
19	a3	b3	c1
20	a3	b4	c3

表 3－8 でパラメータ C の各値の個数は、c1 が 4 個、c2 が 4 個、c3 が 12 個です。この時点で c3 は他の値より 3 倍多く出現しています。これは重み付けの指定どおりです。しかし、網掛けした組み合わせが重複しています。組み合わせの重複が発生するのは、1 つの値を重み付けにより 3 倍多く用いて組み合わせを生成しているため、他のパラメータの値の数が少ないと、同じ組み合わせが生成されるためです。

PictMaster は、重複した組み合わせのうち 1 つを残して他の組み合わせを削除します。そのため重複した組み合わせが存在する場合は、結果的に重み付けで指定した数値より少ない重み付けとなります。

最終的な生成結果を表 3－9 に示します。なお、表 3－8 とは異なる生成実行のため、組み合わせ内容は一部異なります。

表 3－9 重複分を取り除いた最終的な生成結果

No.	A	B	C
1	a1	b1	c1
2	a1	b1	c3
3	a1	b2	c3
4	a1	b3	c2
5	a1	b3	c3
6	a1	b4	c1
7	a1	b4	c2
8	a2	b1	c3
9	a2	b2	c2
10	a2	b3	c1
11	a2	b3	c3
12	a2	b4	c3
13	a3	b1	c2
14	a3	b2	c1
15	a3	b3	c3
16	a3	b4	c3

最終的な生成結果では、重複していた 4 個の組み合わせが削除され、16 個の組み合わせとなりました。この結果では、パラメータ C の各値の個数は、c1 が 4 個、c2 が 4 個、c3 が 8 個です。重み付けを行なって最小テストケース生成を行なうと、環境設定フォームで「統計情報を表示」がチェックされていれば、図 3-22 の統計情報が表示され、重複した組み合わせがいくつあったかが分ります。

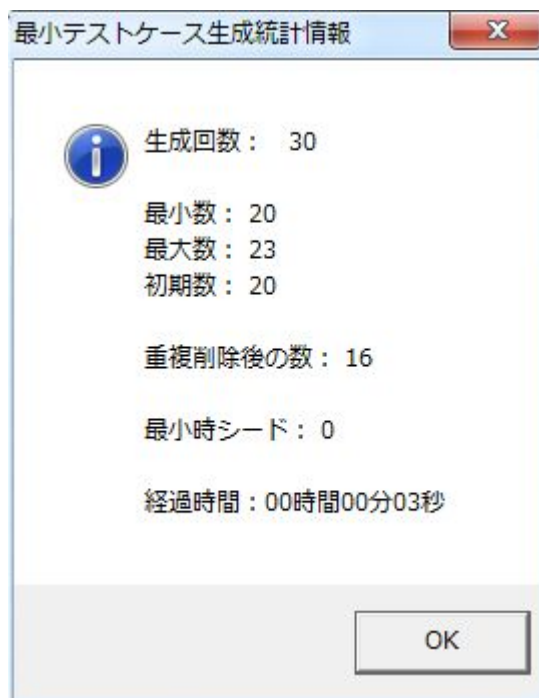


図 3-22 重み付けのあるモデルの生成統計情報の例

重み付けの数値が正確に反映された生成結果が得られるのは、他のパラメータの数が多い場合か、他のパラメータで値の数が多い場合です。この場合は重複が起こらず、生成されるテストケース数も重み付けを行なわない場合と比べてそれほど増加することはありません。そうでない場合、重み付けの数値は目安的な意味合いを持ちます。

重み付けを行なうと値の数を実質的に多くなります。入力できる値の個数は 50 個までです。重み付けを行なう場合は実質的な値の個数が 50 個を超えないようにしてください。

値の重み付けと原型シートを同時に使用した場合、値の重み付けが正常に行なわれません。値の重み付けと原型シートは同時に使用しないことを推奨します。

※ PICT 自体が備えている重み付けの機能は使用できません。

### 3.5 サブモデル

すべてのパラメータがすべて同じ重要度を持つというケースはそれほど多くはありません。PICT ではサブモデル定義を使用することで、特に重要と考えられる限定したパラメータについて異なるパラメータ組み合わせ数を指定することができます。サブモデル欄を表示させるには環境設定フォームの「サブモデルを使用する」にチェックを入れます。

サブモデルの記入形式を図 3-23 に示します。

<パラメータ 1>, <パラメータ 2>, …, <組み合わせるパラメータ数>

図 3-23 サブモデルの記入形式

サブモデルの対象としたい複数のパラメータ名を半角のカンマで区切って記入します。最後に組み合わせるパラメータ数を半角で記入します。この数は1から、指定したパラメータの数までの任意の値が記入できます。セミコロン (;) で区切ることで、1つの行に複数のサブモデルを記入することができます。

サブモデルで指定された複数のパラメータは、“組み合わせるパラメータ数”の数のパラメータ間の組み合わせが生成されます。例えば、この数が3の場合、指定された複数（3個以上）のパラメータのみ3パラメータ間の組み合わせが生成されます。その後で、サブモデルで指定されなかった残りのパラメータと、2パラメータ間の組み合わせとして統合されます。最終結果として、サブモデルで指定された複数のパラメータどうしでは3パラメータ間の組み合わせとなりますが、サブモデルで指定されなかった残りのパラメータとの間では、1つ多い4パラメータ間の組み合わせとなります。そしてサブモデルで指定されなかった残りのパラメータどうしの間ではデフォルトの2パラメータ間の組み合わせとなります。

サブモデルと「K-way 組み合わせと比較」は同時には指定できません。

### 3. 6 拡張サブモデル（パラメータの重み付け）

PICT がサポートしているサブモデルの機能では、テストケース数が増加してしまうため、使いやすくないといえます。

PictMaster では、環境設定フォームの「**拡張サブモデルを使用する**」にチェックを入れることで、特に重要と考えられるパラメータのみ、環境設定の「組み合わせるパラメータ数」とは異なる任意のパラメータ間の組み合わせとしたテストケースを生成することができます。

拡張サブモデルは、希望するパラメータのみを3パラメータ以上の任意の組み合わせとすることに特化したサブモデルであり、通常のサブモデルと比較して、テストケース数の増加を大幅に抑えることができます。例えば、5つの値を持つパラメータが7つあるモデルで、そのうちの3つのパラメータのみ3パラメータの組み合わせとしたサブモデルでは、生成されるテストケース数は次の通りとなります。

通常のサブモデル	6 2 5 件
拡張サブモデル	1 2 9 件

環境設定で「組み合わせるパラメータ数」に3を指定した場合に生成されるテストケース数が2 3 8 件ですから、拡張サブモデルを使用することでテストケース数を大幅に削減することが可能です。

拡張サブモデルの記入形式を図3-24に示します。

＜パラメータ 1＞, ＜パラメータ 2＞, … , ＜組み合わせるパラメータ数（3以上）＞

図3-24 拡張サブモデルの記入形式

拡張サブモデルの対象としたい複数のパラメータ名を半角のカンマで区切って記入します。最後に組み合わせるパラメータ数として3以上から拡張サブモデルで指定されたパラメータ数までの範囲の値を記入します。

記入例を図3-25に示します。

サブモデル
A, B, C, 3

図3-25 拡張サブモデルの記入例（その1）

拡張サブモデルでは、サブモデルはサブモデル欄の最初の行に1つだけ記入できます。セミコロンで区

切って複数のサブモデルを記入することはできません。

3つ以上のパラメータを指定した拡張サブモデルで、最小テストケース生成を実行した場合、統計情報で表示される「最小数」、「最大数」および「初期数」の値は、拡張サブモデルで指定されたパラメータについての組み合わせ数です。最終的な生成数は別の項目として表示されます。

拡張サブモデルで指定されたパラメータの個数と拡張サブモデルの組み合わせるパラメータ数が等しい場合は、拡張サブモデルで指定されたパラメータについての全数組み合わせとなるため、生成される組み合わせは1通りしかありません。そのためこの場合は最小テストケース生成を行なう必要はありません。

拡張サブモデルを使用する際には次に示す条件を満たす必要があります。

- (1) 2つ以上のパラメータが指定されていること。
- (2) 拡張サブモデルの組み合わせるパラメータ数が環境設定の「組み合わせるパラメータ数」よりも大きいこと。
- (3) 拡張サブモデルの組み合わせるパラメータ数で指定できる範囲は3～拡張サブモデルで指定されたパラメータの個数までの範囲の値であること。(パラメータの個数が2つの場合は3のみ)
- (4) 3つ以上のパラメータが指定されている場合、環境設定の「原型シートを使用」および「カバレッジを指定して生成」が指定されていないこと。

例えば、**図3-26**のように環境設定の「組み合わせるパラメータ数」が3で、拡張サブモデルで指定されたパラメータの個数が5個で、拡張サブモデルの組み合わせるパラメータ数が4であるといった指定が可能です。

サブモデル
A, B, C, D, E, 4

**図3-26** 拡張サブモデルの記入例（その2）

拡張サブモデルで2つのパラメータが指定された場合と3つ以上のパラメータが指定された場合とでは組み合わせの内容が異なります。2つのパラメータが指定された場合は、指定された2つのパラメータが他のパラメータとの間で3パラメータ間の組み合わせとなります。3つ以上のパラメータが指定された場合は、指定されたパラメータ間で指定されたパラメータ数の組み合わせとなります。他のパラメータとの間では個々のパラメータが2パラメータ間の組み合わせとなります。

拡張サブモデルの機能は、値の数が少ないパラメータに適用した場合にテストケース削減の効果が大きくなります。逆に値の数が多いパラメータに適用した場合は、テストケース削減の効果がでない場合があります。

3つ以上のパラメータを指定した場合は原型シートの機能を使用するため、Excel のワークシート名の番号が増加します。

拡張サブモデルのアルゴリズムを[付録C](#)で詳しく説明しています。

### 3. 7 希望するカバレッジを確保したテストケースを生成する

PictMaster では、希望するカバレッジを確保したテストケースを生成することができます。3 パラメータ間の組み合わせとするとテストケース数が多くなりすぎて問題となる場合、例えば 3-way カバレッジを 80%としたテストケースでは、生成されるテストケース数を大幅に削減することが可能となります。

例としてパラメータが 15 個あり、それぞれの値の数が 10, 8, 8, 6, 6, 6, 4, 4, 4, 4, 2, 2, 2, 2, 2 であるモデルの場合、3-way カバレッジを 5%刻みで変化させた場合のテストケース数を次に示します。

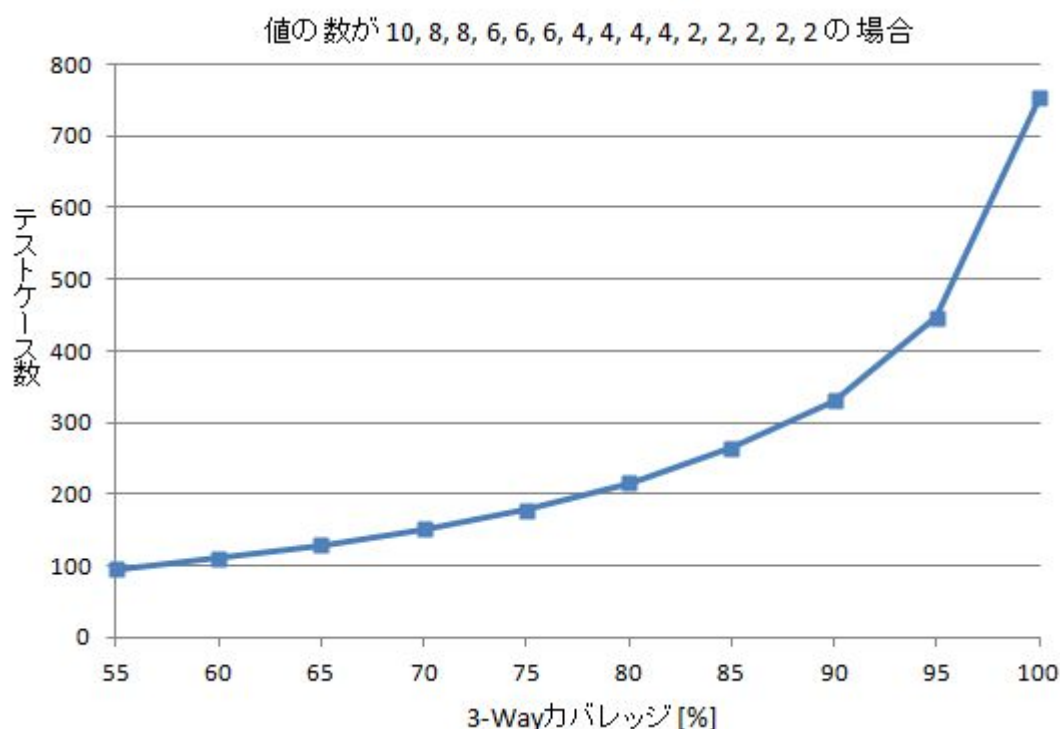


図 3. 27 3-way カバレッジを変化させた場合のテストケース数

3-way カバレッジを 80%とした場合のテストケース数は 216 件となりました。これは 2 パラメータ間の組み合わせを網羅したテストケース数の 2.4 倍であり、3 パラメータ間の組み合わせを網羅したテストケース数の 28.6%にあたります。この程度の増加で 80%という高い 3-way カバレッジを網羅することができるのであれば、特に重要と考えられるテスト対象についてはテストを実施するだけの価値があるといえるでしょう。

希望するカバレッジを確保したテストケースを生成する機能は、2-way から 6-way までのカバレッジを指定することができます。環境設定の「組み合わせるパラメータ数」で指定された値 N に +1 した T-way のカバレッジが確保されます。例えば「組み合わせるパラメータ数」に 5 を指定した場合は、5-way カバレッジが 100%確保され、6-way カバレッジが「希望するカバレッジ」で指定された値で確保されます。

希望する T-way カバレッジを確保したテストケースを生成するアルゴリズムを [付録 D](#) で詳しく説明しています。



### 3. 8 網羅すべき組み合わせが漏れていないかを手軽に確認する

「K-way 組み合わせと比較」の機能では、網羅すべき重要な組み合わせが漏れていないかを手軽に確認することができます。テストケース生成と同時に「対象パラメータ数」で指定されたパラメータ間の組み合わせがファイル k.xls で生成されます。生成された組み合わせは、生成されたテストケースで網羅されていない組み合わせが灰色で塗りつぶされます。同時に Excel のフィルタが設定されます。

例えば図3-28のモデルの場合、2パラメータ間の組み合わせ生成と同時に3パラメータ間の組み合わせを生成し、その中で2パラメータ間の組み合わせで網羅されていない組み合わせが灰色で塗りつぶされ、Excel のフィルタが設定されます。(表3-10)

パラメータ	値の並び
A	a1, a2, a3
B	b1, b2
C	c1, c2, c3
D	d1, d2, d3, d4
E	e1, e2
F	f1, f2
G	g1, g2, g3
H	h1, h2

図3-28 モデルの例

表3-10 3-way 組み合わせと比較したテストケースの例

	A	B	C	D	E	F	G	H
1	A	B	C	D	E	F	G	H
2	a3	b1	c2	d2	e1	f1	g3	h2
3	a3	b2	c1	d1	e2	f2	g2	h1
4	a2	b1	c1	d4	e2	f1	g1	h2
5	a1	b2	c2	d1	e1	f2	g1	h1
6	a1	b2	c1	d3	e2	f1	g3	h1
7	a2	b1	c3	d1	e1	f2	g2	h2
8	a3	b2	c3	d3	e1	f2	g1	h2
9	a2	b1	c2	d3	e2	f1	g2	h1
10	a1	b2	c3	d1	e1	f1	g3	h2
11	a2	b2	c3	d4	e1	f2	g3	h1
12	a2	b2	c3	d2	e2	f2	g1	h1
13	a3	b1	c2	d4	e1	f2	g2	h1
14	a1	b1	c1	d2	e1	f1	g2	h2
15	a1	b2	c2	d4	e1	f1	g2	h2
16	a1	b1	c2	d4	e2	f2	g3	h2
17	a3	b1	c1	d3	e1	f2	g2	h2
18	a3	b1	c1	d4	e1	f1	g3	h1
19	a3	b2	c2	d3	e2	f1	g1	h2
20	a3	b1	c3	d1	e2	f1	g3	h1
21	a1	b2	c3	d4	e2	f1	g2	h1
22	a2	b1	c1	d3	e1	f2	g1	h1
23	a3	b1	c3	d4	e1	f1	g1	h1
24	a2	b1	c2	d1	e2	f1	g2	h2

この例では、テストケースの自動整形を行なっていません。最初の塗りつぶしが行なわれていないテス



トケースが2パラメータ間の組み合わせであり、塗りつぶされているテストケースが2パラメータ間の組み合わせで網羅されていない3パラメータ間の組み合わせです。この例では全体の組み合わせは52件となりました。

Excel のフィルタが自動的に設定されますのでフィルタ機能を使って3パラメータ間の組み合わせで重要な組み合わせが漏れていないかを容易に確認することができます。この機能がない場合は、網羅されていない組み合わせは明示されないのので分かりにくいものです。この機能では、網羅されていない組み合わせが灰色に塗りつぶして明示されるため、確認漏れの心配がありません。

図3-29にフィルタを使用した例を示します。

図3-29 フィルタを使用した例

	A	B	C	D	E	F	G	H
1	A	B	C	D	E	F	G	H
13	a3	b1	c2	d4	e1	f2	g2	h1
15	a1	b2	c2	d4	e1	f1	g2	h2
16	a1	b1	c2	d4	e2	f2	g3	h2
37	a3	b2	c2	d4	e2	f2	g1	h2
46	a2	b1	c2	d4	e2	f2	g3	h2

この例では、値 c2 と d4 でフィルタを使用しています。この2パラメータの値の組み合わせで、網羅されたいない値が灰色で示されています。例えば値 c2 と d4 は値 a2 との組み合わせがありません。もしも値 a2 との組み合わせが重要であるならば、テストケースにその組み合わせが含まれるように原型シートで指定すればよいことになります。

#### 【重要な注意】

「K-way 組み合わせと比較」の機能で「対象パラメータ数」に大きな値を指定した場合、K-way 組み合わせを生成するまでに非常に長い時間がかかる場合があります。途中で処理を中断したい場合はタスクマネージャを起動し、PICT のプロセスを終了してください。

## 4. 制約表への記入のしかた

### 4. 1 制約に関する用語の定義

制約表の説明をする前に制約に関する用語の定義を明確にしておきます。

**制約**とは、組み合わせることのできないパラメータと値の組み合わせのことです。この制約を if 関係式 then 関係式 の形式で記述した式を**制約式**といいます。制約式によって、組み合わせできない組み合わせを除外し、組み合わせできる組み合わせだけを残すことができます

**関係式**は、パラメータと値、またはパラメータとパラメータを比較するために = 、< など**演算子**として用います。PICT はこの制約式を入力として制約に対応した組み合わせを生成します。

if と then との間**の関係式**を**制約条件**といい、then 以降**の関係式**を**制約対象**と言います。関係式の演算子が等号 (=) のとき**順制約**といい、関係式の演算子が不等号 (<) のとき**逆制約**といいます。制約条件が成立するとき、制約対象で指定された組み合わせとなります。

以上の用語を使って制約式を記述すると以下のとおりとなります。

if 制約条件 then 制約対象 … 制約条件が真のとき、制約対象の組み合わせとなる。

以上の制約式を**条件付き制約式**といいます。制約条件のある制約式はすべて条件付き制約式です。これに対して、制約条件がなく、制約対象のみからなる制約式を**無条件制約式**といいます。無条件制約式ではパラメータのみが対象となります。基本的な無条件制約式の記述を以下に示します。

パラメータ 演算子 パラメータ

**無条件制約**はパラメータの値の如何にかかわらず、**常に成立する制約**です。

### 4. 2 制約表の構成

未記入の制約表を図 4-1 に示します。

制約表			
パラメータ	制約1	制約2	制約3

図 4-1 未記入の制約表

図 4-1 でパラメータの列には 9 行目からのパラメータ欄の内容をコピーして貼り付けます。環境設定フォームを開いて OK ボタンをクリックすることで自動的に張り付けを行なうこともできます。ただし**制約がない場合はパラメータ欄のコピーは不要です**。制約 1、2、3、～ は、基本的には 1 つの制約式が 1 つの制約に対応します。パラメータ欄で指定されたパラメータの値のうち、任意の値を制約欄に記入します。また別のパラメータそのものを記入することもできます。

パラメータは 50 個まで、制約は 50 個まで記入できます。パラメータ欄は間に空白行を置かず詰めて記入してください。PictMaster はパラメータ欄、値の並び欄および制約表欄の記述内容を、a.txt というモデルファイルに出力し、PICT に入力データとして渡します。この際、PictMaster は制約表の記入内容を PICT が理解できる制約式に変換します。

#### 4. 3 制約条件と制約対象の指定方法

制約条件と制約対象で、記述できる形式に差はありません。以降の説明で制約条件と制約対象の例として挙げられている記述は、制約条件と制約対象のどちらでも記述可能です。

##### 4. 3. 1 条件付き制約

**制約条件**とする制約欄は白色以外の任意の色で塗りつぶしてください。塗りつぶされた制約欄に記入された値またはパラメータが制約条件となります。

**制約対象**とする制約欄は塗りつぶさないでください。白色で塗りつぶしても塗りつぶしなしとして扱われます。制約欄に記入された値またはパラメータが制約対象となります。

制約欄に複数の値を記入する場合はそれぞれの値を半角のカンマ(,)で区切ります。

先頭にシャープ(#)をつけると逆制約となり、記入した値以外の値を意味します。この場合もカンマで区切って複数の値を記入することができます。

先頭にグレートザン(>)、レスザン(<)をつけることで値の大小比較ができます。あるパラメータの1つの値に“>”と“<”をつけることで任意の範囲の値を指定することができます。この場合はカンマ(,)で区切られた二つの値のAND条件となります。値が数字だけの場合は問題ありませんが、数字と文字が混在する場合は全体が文字と見なされ、文字コードでの大小比較になります。

エイリアスの値を指定する場合は、先頭の値の名称を使用します。

**パラメータそのものを記入する場合は**、順制約の場合は先頭にイコール(=)をつけます。逆制約の場合はエクスクラメーション(!)を付けます。複数のパラメータを指定する場合はそれぞれのパラメータを半角のカンマ(,)で区切り、それぞれのパラメータにイコール(=)またはエクスクラメーション(!)を付けます。値を記入する場合と異なりますので注意してください。パラメータそのものを記入する場合はパラメータ欄と制約欄のパラメータ双方で少なくとも一部が同じ値を含んでいる必要があります。

制約条件でもなく制約対象でもない場合、制約欄は空白とします。

制約が1つもない場合は制約表のパラメータ欄は空白でもかまいません。

##### 4. 3. 1. 1 パラメータと値との制約

この章では、パラメータの値がある特定の値の場合に発生する制約の指定方法を説明します。モデルが図4-2で、制約表が図4-3の場合、生成されるモデルファイル“a.txt”はリスト4-1となります。モデルファイルは、モデルそのものの部分を省略してあります。

パラメータ	値の並び
A	a1, a2, a3
B	b1, b2, b3
C	c1, c2, c3

図4-2 モデルの例(その1)

制約表		
パラメータ	制約1	制約2
A	a1	#a1
B	b1	#b1
C		

図4-3 制約表の例(その1)

リスト4－1 制約式の例（その1）

```
if ([A] = "a1" )
    then ([B] = "b1" ) ;
if ([A] <> "a1" )
    then ([B] <> "b1" ) ;
```

生成結果の例を表4－1に示します。

表4－1 生成結果の例（その1）

No.	A	B	C
1	a1	b1	c1
2	a1	b1	c2
3	a1	b1	c3
4	a2	b2	c1
5	a2	b2	c3
6	a2	b3	c2
7	a3	b2	c2
8	a3	b3	c1
9	a3	b3	c3

図4－3の制約指定の意味は次の通りです。

制約1：パラメータAの値が a1 の場合、パラメータBの値で組み合わせ可能な値は b1 である。

制約2：パラメータAの値が a1 以外の場合、パラメータBの値で組み合わせ可能な値は b1 以外である。

表4－1の生成結果を見ると、パラメータAの値が a1 の場合、パラメータBの値は b1 のみであり、パラメータAの値が a1 以外の場合、パラメータBの値は b1 以外となっています。これは図4－3の制約指定の通りです。

図4－2のモデルで制約表が図4－4の場合、生成されるモデルファイル“a.txt”はリスト4－2となります。

制約表		
パラメータ	制約1	制約2
A	a1,a2	a3
B	b1	#b1
C		

図4－4 制約表の例（その2）

リスト4－2 制約式の例（その2）

```
if ([A] = "a1" or [A] = "a2" )
    then ([B] = "b1" ) ;
if ([A] = "a3" )
    then ([B] <> "b1" ) ;
```

生成結果の例を表4－2に示します。

表 4－2 生成結果の例（その 2）

No.	A	B	C
1	a1	b1	c1
2	a1	b1	c2
3	a1	b1	c3
4	a2	b1	c1
5	a2	b1	c2
6	a2	b1	c3
7	a3	b2	c1
8	a3	b2	c2
9	a3	b2	c3
10	a3	b3	c1
11	a3	b3	c2
12	a3	b3	c3

図 4－2 のモデルで制約表が図 4－5 の場合、生成されるモデルファイル “a. txt” はリスト 4－3 となります。

制約表		
パラメータ	制約1	制約2
A	a1, a2	a3
B	b1	#b1
C	#c2, c3	c1

図 4－5 制約表の例（その 3）

リスト 4－3 制約式の例（その 2）

```

if ([A] = "a1" or [A] = "a2" ) and ([B] = "b1" )
    then ([C] <> "c2" and [C] <> "c3" ) ;
if ([A] = "a3" )
    then ([B] <> "b1" ) and ([C] = "c1" ) ;

```

生成結果の例を表 4－3 に示します。

表 4－3 生成結果の例（その 3）

No.	A	B	C
1	a1	b1	c1
2	a1	b2	c3
3	a1	b3	c2
4	a2	b1	c1
5	a2	b2	c2
6	a2	b3	c3
7	a3	b2	c1
8	a3	b3	c1

これまでの例で分かるようにある制約の 1 つの欄に複数の値が記入されている場合は、それぞれの値は基本的には OR 条件になります。これに対してある制約の異なる行に値が記入されている場合は、それぞれの値は AND 条件になります。これはパラメータそのものが記入された場合も同様です。

**制約条件**として異なるパラメータの値を OR 条件で指定したい場合は、異なる制約に制約条件として異なるパラメータの値を記入し、制約対象は同じにします。

**制約対象**として異なるパラメータの値を OR 条件で指定したい場合は、すぐ右側の制約欄に制約条件として同じパラメータの値を記入し、制約対象に異なるパラメータの値を記入します。図 4－6 にこの場合の制約表を示します。このとき生成される制約式をリスト 4－4 に、生成結果の例を表 4－4 に示します。複数の制約条件がある場合はすべての制約条件が同一である必要があります。

制約表			
パラメータ	制約1	制約2	制約3
A			a1
B	b1	b3	b3
C	c1	c2	

図 4－6 制約表の例（その 4）

リスト 4－4 制約式の例（その 3）

```

if ([B] = "b1" )
  then ([C] = "c1" ) ;
if ([B] = "b3" )
  then ([C] = "c2" ) or ([A] = "a1" ) ;

```

表 4－4 生成結果の例（その 4）

No.	A	B	C
1	a1	b1	c1
2	a1	b2	c1
3	a1	b2	c2
4	a1	b3	c1
5	a1	b3	c3
6	a2	b1	c1
7	a2	b2	c3
8	a2	b3	c2
9	a3	b1	c1
10	a3	b2	c3
11	a3	b3	c2

この例では制約 2 と制約 3 の制約条件が 1 つの制約式として統合され、制約対象が OR 条件となっています。制約対象がいくつあっても同じ制約では AND 条件となり、異なる制約間では OR 条件となります。ただし隣り合う制約の制約条件の欄を異なる色にすると制約式の統合は行われず、2 つの異なる制約式となります。

#### 4. 3. 1. 2 パラメータとパラメータとの制約

次に制約欄にパラメータを指定する場合を示します。このときのモデルを図 4－7、制約表を図 4－8、生成される制約式をリスト 4－5、生成結果の例を表 4－5 に示します。

パラメータ	値の並び
A	a1, a2, a3
B	1, 2, 3
C	1, 2, 3
D	1, 2, 3

図 4－7 モデルの例（その 2）

制約表		
パラメータ	制約1	制約2
A	a1	a3
B		=C
C	!B	
D		

図 4－8 制約表の例（その 5）

リスト 4－5 制約式の例（その 4）

```
if ([A] = "a1" )
    then ([C] <> [B] ) ;
if ([B] = [C] )
    then ([A] = "a3" ) ;
```

表 4－5 生成結果の例（その 5）

No.	A	B	C	D
1	a1	1	3	3
2	a1	2	1	2
3	a1	3	2	1
4	a2	1	2	2
5	a2	2	3	1
6	a2	3	1	3
7	a3	1	1	1
8	a3	2	2	2
9	a3	2	2	3
10	a3	3	3	2

パラメータを指定する場合は、指定する側と指定される側のパラメータの値に一部でも**同じ値が含まれている必要があります**。また、指定する側と指定される側の値の種類（文字列か数値）が一致しなければなりません。1つの欄に複数のパラメータを記入することもできます。この場合はそれぞれのパラメータの前にイコール (=) またはエクスクラメーション (!) を付加し、カンマ (,) で区切ります。各パラメータはOR条件での指定となります。

1つの欄にAND条件で複数のパラメータを指定する場合は演算子の前に**アンパサンド (&)** を付加します。  
 このときのモデルを図 4－7、制約表を図 4－9、生成される制約式をリスト 4－6、生成結果の例を表 4－6 に示します。

制約表		
パラメータ	制約1	制約2
A	a1	#a1
B	=C, &=D	!C, !D
C		
D		

図 4－9 制約表の例（その 6）

リスト 4－6 制約式の例（その 6）

```
if ([B] = [C] and [B] = [D] )
    then ([A] = "a1" ) ;
if ([B] <> [C] or [B] <> [D] )
    then ([A] <> "a1" ) ;
```



表 4－6 生成結果の例（その 6）

No.	A	B	C	D
1	a1	1	1	1
2	a1	2	2	2
3	a1	3	3	3
4	a2	1	2	3
5	a2	2	3	1
6	a2	3	1	2
7	a3	1	3	2
8	a3	2	1	3
9	a3	3	2	1

#### 4. 3. 2 無条件制約

無条件制約には制約条件がありません。そのため、**無条件制約は常に成立する制約です**。制約表に記入できるのはパラメータのみです。制約欄に演算子付きで記入することで、2つのパラメータ間の制約を指定します。

使用できる演算子には、**イコール (=)**、イコールと逆の意味を表す**エクスクラメーション (!)** があります。1つの制約欄に**カンマ (,)** で区切って複数のパラメータを指定することができます。この場合、それぞれの **OR** 条件となりますが、2つめ以降のパラメータの先頭に**アンバサンド (&)** を置くことで **AND** 条件とすることができます。

モデルを図 4－1 0、AND条件で逆制約の制約表を図 4－1 1、生成される制約式をリスト 4－7、生成結果の例を表 4－7 に示します。

パラメータ	値の並び
A	a1, a2, a3
B	1, 2, 3
C	1, 2, 3
D	1, 2, 3

図 4－1 0 モデルの例（その 3）

制約表	
パラメータ	制約1
A	
B	!C
C	!D
D	!B

図 4－1 1 制約表の例（その 7）

リスト 4－7 制約式の例（その 7）

([B] <> [C] ) and ([C] <> [D] ) and ([D] <> [B] ) ;

表 4－7 生成結果の例（その 7）

No.	A	B	C	D
1	a1	1	3	2
2	a1	2	1	3
3	a1	3	2	1
4	a2	1	2	3
5	a2	2	3	1
6	a2	3	1	2
7	a3	1	3	2
8	a3	2	1	3
9	a3	3	2	1

この例では 1 つの式になっていますが、記入する制約を別にする、異なる独立した式になります。その場合でも意味的には同一の制約です。

OR 条件で順制約の制約表を図 4－1 2、生成される制約式をリスト 4－8、生成結果の例を表 4－8 に示します。OR 条件を指定するには無条件制約の右側の制約で、先頭にプラス（+）を置きます。

制約表			
パラメータ	制約1	制約2	制約3
A			
B	=C		
C		+D	
D			+B

図 4－1 2 制約表の例（その 8）

リスト 4－8 制約式の例（その 8）

([B] = [C] ) or ([C] = [D] ) or ([D] = [B] ) ；

表 4－8 生成結果の例（その 8）

No.	A	B	C	D
1	a1	1	1	2
2	a1	2	2	3
3	a1	3	3	1
4	a2	1	2	1
5	a2	2	3	2
6	a2	3	1	3
7	a3	1	3	3
8	a3	2	1	1
9	a3	3	2	2

1 つの制約欄に複数のパラメータを記入した例として、モデルを図 4－1 2、制約表を図 4－1 3、生成される制約式をリスト 4－9、生成結果の例を表 4－9 に示します。

パラメータ	値の並び
A	a1, a2, a3
B	1, 2, 3
C	1, 2, 3
D	1, 2, 3
E	1, 2, 3
F	1, 2, 3

図 4-12 モデルの例（その 4）

制約表		
パラメータ	制約1	
A		
B	=C, !D	
C		
D	!E, &=F	
E		
F		

図 4-13 制約表の例（その 8）

リスト 4-9 制約式の例（その 9）

(([B] = [C] or [B] <> [D] ) and ([D] <> [E] and [D] = [F] ) ) ;

表 4-9 生成結果の例（その 9）

No.	A	B	C	D	E	F
1	a1	1	1	3	1	3
2	a1	1	3	2	1	2
3	a1	2	2	2	3	2
4	a1	3	2	1	2	1
5	a2	1	3	3	2	3
6	a2	2	2	3	1	3
7	a2	2	3	1	3	1
8	a2	3	1	2	3	2
9	a3	1	1	1	3	1
10	a3	1	2	2	3	2
11	a3	2	1	1	2	1
12	a3	3	3	3	1	3

#### 4. 4 使用できる演算子の一覧

関係式で使用できる演算子はパラメータと値、パラメータとパラメータで異なります。**表 4－10**に使用できる演算子を示します。＃を除いていずれの演算子もカンマ（,）で区切って複数回記述することができます。条件の欄は1つの制約欄に値またはパラメータを複数記述した場合にOR条件となるのかAND条件となるのかを表します。

表 4－10 使用できる演算子

関係式の対象	演算子	条件	説明
パラメータと値	(指定しない)	OR	パラメータに属する値を表します。＝と同じ意味です。
	＃	AND	記述した値を除いた残りの値を表します。任意の1つの値について先頭に記述しますが値は複数記述することができます。
	>, <	AND	値の大小関係を表します。
パラメータとパラメータ	=	OR	パラメータ間の値が等しいことを表します。
	!	OR	パラメータ間の値が異なることを表します。
	&=	AND	パラメータ間の値が等しいことをAND条件で表します。パラメータを複数記述した場合に、2つめ以降のパラメータの前に記述します。
	&!	AND	パラメータ間の値が異なることをAND条件で表します。パラメータを複数記述した場合に、2つめ以降のパラメータの前に記述します。
	+	(OR)	無条件制約で異なる制約で OR 条件となることを表します。無条件制約の右側の制約で無条件制約の最初のパラメータの前に記述します。

表中の演算子（+）の条件欄が(OR)となっているのは、異なる制約間の条件を表しているためです。

これらの演算子と同じ記号をパラメータおよび値の名称の先頭に使用することはできません。

#### 4. 5 制約表の編集方法

制約表の編集方法について説明します。

Excel のメニューからの**制約表の行の削除や挿入は行なわないでください**。制約表の行数が変化すると次に続く結果表欄の行位置がずれてしまいます。

行の削除や挿入を行ないたい場合はその1つの行の**1列目を右クリック**してください。追加、削除、元へ戻すなどの**編集専用のショートカットメニュー**が表示され、メニューを選択することで行の削除や挿入を行なうことができます。

制約の削除や挿入を行ないたい場合は**制約のタイトル欄を右クリック**してください。追加、削除、元へ戻すなどの**編集専用のショートカットメニュー**が表示され、メニューを選択することで制約の削除や挿入を行なうことができます。

詳細は[6. 5章](#)を参照してください。

## 4. 6 ワイルドカードの使用

制約表に記入する値には**ワイルドカード**を使用することができます。ワイルドカードとは、「任意の文字」を意味する特殊な文字です。「\*」は0個以上の任意の文字を、「?」は1個の任意の半角文字を意味します。

例えば「A\*」は名称が「A」で始まるすべての値を意味します。「A」だけでも含まれます。「???A」は半角4文字で最後が「A」で終わるすべての値を意味します。全角文字を「?」で指定する場合は、全角1文字につき2文字の「??」で指定します。

ワイルドカードは、制約表では条件付制約の値の名称のみに使用することができます。値のタイプが数値の場合はワイルドカードを使用することはできません。あるパラメータの値が数値の値と文字の値が混在している場合、値のタイプは文字となります。

ワイルドカードの使用例と PictMaster が生成する制約式を以下に示します。

制約表	
パラメータ	制約1
A	
B	*B*
C	#C??

図 4-14 ワイルドカードの使用例

### リスト 4-10 ワイルドカードを含む制約式

```
if ([B] LIKE "*B*")  
    then ([C] NOT LIKE "C??") ;
```

値の名称に適切な文字列を付与することで、ワイルドカードを使用して制約指定を簡潔に記述することができます。

## 5. 結果表への記入のしかた

### 5. 1 結果表の構成

テストケースの組み合わせによってテスト結果が異なる場合があります。テストケースの数が多い場合は、期待する結果（結果内容）を記入するのに手間がかかりますが、結果表を使用することでこの手間を省くことができます。テストケースの生成を繰り返す場合は大幅な省力化になります。

未記入の結果表を図5-1に示します。

結果表		
結果内容	パラメータ1	パラメータ2

図5-1 未記入の結果表

結果表の**結果内容欄**には、右側の各パラメータの、値の組み合わせの場合にどのような結果となるべきかを記入します。記入内容が長くなる場合は、結果内容欄に（\*1）（\*2）…などの番号を記入し、生成されたテストケースが書かれたワークシートの欄外で番号ごとに具体的な結果内容を記述すればよいでしょう。結果内容欄は50行設けられており、デフォルトでは15行分表示されています。不足する場合は残りの行を再表示してください。環境設定フォームを表示させて OK ボタンをクリックすると、結果表のパラメータ欄に実際のパラメータ名が設定されます。

結果内容欄は1行目から間を空けずに詰めて記入してください。

結果表の**パラメータ欄**の列には、左側の結果内容欄の結果となる値の組み合わせ（一致条件）を制約表の記述と同じ方法で記入します。ただしパラメータそのものを記入することはできません。

パラメータ欄の列は50列設けられています。

### 5. 2 一致条件の指定方法

一致条件は制約表で値を指定する場合と同じ方法で指定します。左側の結果内容欄に書かれている結果となる値の組み合わせを必要なパラメータごとに記入します。1つの欄には値をカンマ（,）で区切って複数記入することができます。先頭にシャープ（#）をつけると逆条件となり、指定した値以外の値を意味します。色の塗りつぶしは不要です。

環境設定フォームで「結果表を使用」にチェックを入れて OK をクリックすると、モデルのパラメータ欄に記述されているパラメータ名が結果表のパラメータ欄に自動的に記入されます。

一致条件欄の記入は制約表の記述に違反しないようにします。結果内容欄に記入があり、すべてのパラメータの一致条件欄になにも記入がない場合は、他の結果内容欄の一致条件にいずれにも一致しなかったテストケースの結果内容欄にその記述内容が記入されます。

図5-2にモデルの例、および図5-3に結果表の例を示します。

パラメータ	値の並び
A	a1, a2, a3, a4
B	b1, b2, b3, b4
C	c1, c2, c3

表5-2 モデルの例

結果表			
結果内容	A	B	C
1111となる	a1, a2	b1, b2	
1111となる		b3	c3
2222となる	#a1, a2		c2
3333となる			

図 5－3 結果表の例

結果表の記入を行なう際は、CTL-e（デフォルト）を押してウインドウの分割を行ない、モデルと結果表がともに見えるようにすると記入しやすくなります。

図 5－3 では結果内容欄には 4 つの確認内容が記入されており、最後の確認内容には一致条件が何も記入されていません。それまでの 3 つの条件に一致しなかったテストケースに最後の結果内容が記入されます。最初の 2 つの結果内容欄は同じ内容です。このような指定も可能です。

一致条件はパラメータ間の AND 条件となります。記入されたすべての条件が一致するテストケースのみに左側の結果内容が記入されます。記入された値がエイリアスの場合はそのすべての値を意味します。

図 5－2 から図 5－3 の条件で生成されたテストケースの例を表 5－1 に示します。

表 5－1 生成されたテストケース

No.	A	B	C	結果内容
1	a1	b1	c3	1111 となる
2	a1	b2	c1	1111 となる
3	a1	b3	c1	3333 となる
4	a1	b4	c2	3333 となる
5	a2	b1	c1	1111 となる
6	a2	b2	c3	1111 となる
7	a2	b3	c2	3333 となる
8	a2	b4	c3	3333 となる
9	a3	b1	c3	3333 となる
10	a3	b2	c3	3333 となる
11	a3	b3	c2	2222 となる
12	a3	b4	c1	3333 となる
13	a4	b1	c2	2222 となる
14	a4	b2	c2	2222 となる
15	a4	b3	c3	1111 となる
16	a4	b4	c1	3333 となる

生成されたテストケースには右端に結果内容の欄が追加され、一致条件に一致したテストケースについて結果表の結果内容欄の記述が記入されています。テストケース数が数十個以上と多い場合は結果表機能があると時間の節約になります。生成を繰り返した場合は時間短縮効果が大変大きくなります。

エイリアスの値を指定する場合は、先頭の値の名称を使用します。

### 5. 3 使用できる演算子の一覧

一致条件で使用できる演算子を表5-2に示します。値はカンマ(,)で区切って複数記述することができます。条件の欄は1つの一致条件欄に複数の値を記述した場合にOR条件となるのかAND条件となるのかを表します。

表5-2 使用できる演算子

関係式の対象	演算子	条件	説明
パラメータと値	(指定しない)	OR	パラメータに属する値を表します。=と同じ意味です。カンマ(,)で区切って複数記述することができます。
	#	AND	記述した値を除いた残りの値を表します。値の先頭に1つだけ記述しますが値は複数記述することができます。

### 5. 4 記入上の注意事項

結果表の記入に関する注意事項を説明します。

#### (1) 一致条件の重複

複数の結果内容を定義した場合、同じ組み合わせに2つの異なる結果内容が一致することがあります。これは一致条件が重複する条件を含んでいるためです。この状態となった場合、PictMasterは図5-4の例のようなエラーメッセージを表示し、そこで処理を中止します。このエラーとなった場合は、条件が重複しないように「#」を使った逆条件を追加するとうまいきます。

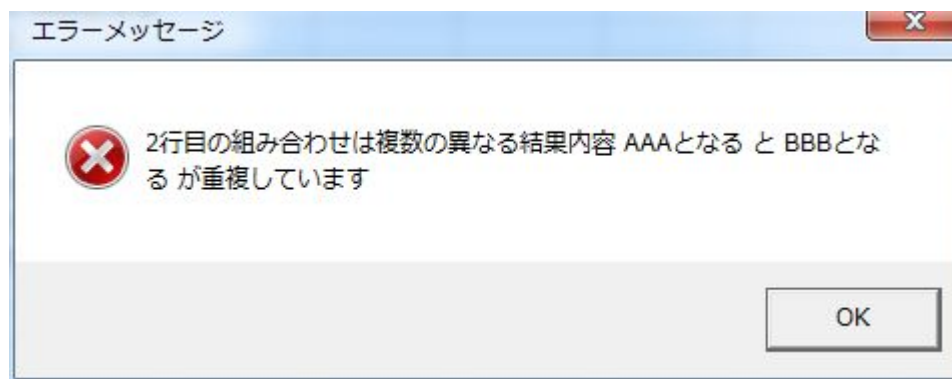


図5-4 一致条件重複のエラーメッセージの例

#### (2) 一致条件の不一致

一致条件に一致する組み合わせが1つもない場合があります。たとえば組み合わせるパラメータ数を2としている場合に一致条件として3つ以上のパラメータの値について指定した場合、起きる可能性が高くなります。また制約表の記述に違反した組み合わせを指定する一致条件の場合もこのエラーとなります。この状態となった場合、PictMasterは図5-5の例のようなエラーメッセージを表示し、そこで処理を中止します。



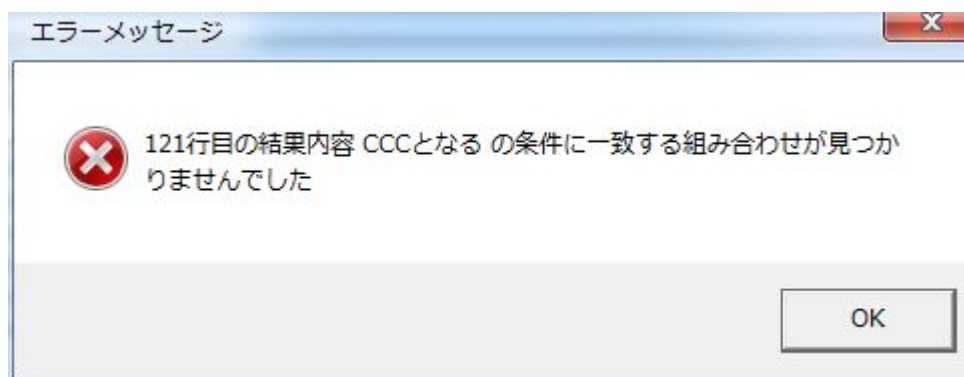


図 5－5 一致条件不一致のエラーメッセージの例

## 5. 5 結果表のもうひとつの使い方 結果内容の連結

これまで説明した結果表の使い方は、1つのテストケースは1つの結果内容にのみ一致することを条件としていました。しかし、テスト対象によっては、この仕様では結果表への記入が煩雑になることがあります。それは、1つのテストケースを構成する多数のパラメータの値によって結果内容が様々に変化する場合があります。

例えば図 5－6 に示すモデルで、結果内容がパラメータ A の値と B の値によって変化する場合、これまでの方法では図 5－6 の結果表のように  $3 \times 3 = 9$  個の結果内容を記入しなければなりません。生成された結果は、表 5－3 のようになります。

パラメータ	値の並び
A	a1, a2, a3
B	b1, b2, b3
C	c1, c2, c3
D	d1, d2, d3

結果表		
結果内容	A	B
A1B1	a1	b1
A1B2	a1	b2
A1B3	a1	b3
A2B1	a2	b1
A2B2	a2	b2
A2B3	a2	b3
A3B1	a3	b1
A3B2	a3	b2
A3B3	a3	b3

図 5－6 モデルと結果表の例

表 5－3 生成結果の例

No.	A	B	C	D	結果内容
1	a1	b1	c3	d3	A1 B1
2	a1	b2	c1	d2	A1 B2
3	a1	b2	c2	d2	A1 B2
4	a1	b3	c2	d1	A1 B3
5	a2	b1	c2	d2	A2 B1
6	a2	b2	c3	d1	A2 B2
7	a2	b3	c1	d3	A2 B3
8	a3	b1	c1	d1	A3 B1
9	a3	b2	c2	d3	A3 B2
10	a3	b3	c3	d2	A3 B3

これまで説明した方法では、この例のようにすべての組み合わせ結果ごとに結果表の結果内容を記述しなければなりません。まだ生成結果に影響を及ぼす値の数が少ないからいいのですが、増えてくると必要となる結果内容の欄が急激に増加してしまいます。

この例のように1つのテストケースを構成する多数のパラメータの値によって結果内容が様々に変化する場合、パラメータの値の組み合わせに対して結果内容を記入するのではなく、個々のパラメータの値ごとに独立して結果内容を記述することができれば、記入しなければならない結果内容欄の数を少なくすることができます。生成されたテストケースの結果内容欄には、一致した各内容欄の記入内容が「連結」されて設定されることになります。

図 5－7 にそうした結果表の例を示します。

結果表		
結果内容	A	B
A1	a1	
A2	a2	
A3	a3	
&B1		b1
&B2		b2
&B3		b3

図 5－7 結果内容を連結する結果表の例

この結果表による生成結果の結果内容欄は、表 5－3 とまったく同一となります。結果内容欄の先頭の文字がアンパサンド (&) で始まっている場合、テストケースの値が結果表の一致条件に一致したときの結果内容が異なると、それぞれの異なる結果内容の文字列が「連結」され、新しい結果内容となります。言い換えると、生成されたテストケースの「結果内容」欄にすでに記入されている内容と、今現在一致条件に一致したときの「結果内容」が異なる場合、それぞれの異なる結果内容の文字列が「連結」され、新しい結果内容となります。

この例では結果内容欄は、これまでの方法の 9 個に対して  $3 + 3 = 6$  個で済んでいます。結果内容に影響を与える値が増えるほど、この差は大きなものとなります。

結果表の一致条件に一致したとき、現在一致した結果内容とすでにテストケースの結果内容欄に記入されている内容が異なる場合、順番が後の結果内容欄の先頭に & がないときは、これまでどおり「一致条件の重複」のエラーメッセージが表示されます。

## 6. より有効な使い方

### 6. 1 PictMasterのカスタマイズ

PictMaster は、Excel の Book であることから使いやすいようにカスタマイズすることが可能です。1 ～ 7 行目は自由にレイアウトしてかまいません。PictMaster のファイル名、シート名は任意の名前に変更してかまいません。

PictMaster で編集メニューから「シートの移動またはコピー」を選択し、表示されたフォームの「コピーを作成する」にチェックを入れ「OK」ボタンをクリックすることで異なるテストケースを生成するシートを任意の枚数設けることができます。これに対して、挿入メニューからワークシートを選択して新しいシートを作成し、PictMaster のシートをコピーして貼り付けてもそのシートでは正常に動作しませんので注意してください。

テスト対象の大きな機能ごとに PictMaster の Book を設け、そのいくつかの組み合わせテストケースのモデルを複数のシートに分けて管理するという方法がよいかもしれません。またテスト仕様書を Excel で作成している場合は、PictMaster を使用してテストケースを作成したテスト仕様書を PictMaster の別シート上に記述し、テスト仕様書と PictMaster を 1 つの Book に統合することも可能です。その場合、「PictMaster」という Book 名はテスト対象を表す機能名などの名称に変更することになるでしょう。このような例でのシート名の並びを図 6-1 に示します。

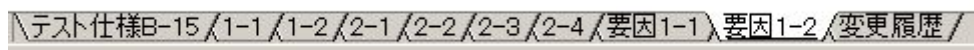


図 6-1 シート名の並びの例

この例ではシート名「テスト仕様 B-15」がテストの記号名であり Book 名でもあります。1-1～2-4 のシートに記入されているテストケースについてのテスト方法、確認内容、データ設定内容などを記述したテスト設計仕様書(\*1)のシートです。1-1～2-4 のシートは個々の確認内容に応じたテストケース仕様書(\*1)です。これらのテストケースのうち、“要因”がシート名についているシートが PictMaster を使用して組み合わせテストケースの作成に使用した PictMaster のシートです。

カスタマイズする際、パラメータ、値の並び、サブモデル、制約表および結果表の文字が記入されたセルの行番号と列番号は変えないで下さい。VBA がモデルなどの位置を認識できず実行できなくなります。

\*1: IEEE 829 標準規格に準拠したドキュメント。テスト手順書に該当する内容は、テスト設計仕様書またはテストケース仕様書で包含しているので設けていない。

### 6. 2 エラー／警告メッセージが表示された場合

PictMaster では制約表から制約式に変換する過程で多くのチェックを行っており、PICT 自体からエラー/警告メッセージが表示されることはまれだと思います。

エラー/警告メッセージが表示されている間はそのメッセージ内容が記述されているファイル “e.txt” が存在するので、メモ帳などで e.txt のファイルを開くことができます。ファイルを開いた後で、エラー（警告）メッセージの OK ボタンをクリックすれば、メモ帳などでエラー/警告メッセージを見ながら PictMaster のパラメータ定義や制約表の間違いを調べることができます。また警告メッセージの場合は、警告メッセージの OK ボタンをクリックしても、テストケースは生成されているので “a.xls” のファイルを開いて生成結果を確認することができます。

### 6. 2. 1 矛盾した制約をすばやく見つけるには

PICT から表示されるメッセージで多いものとして図 6－2 に示すようなメッセージがあります。

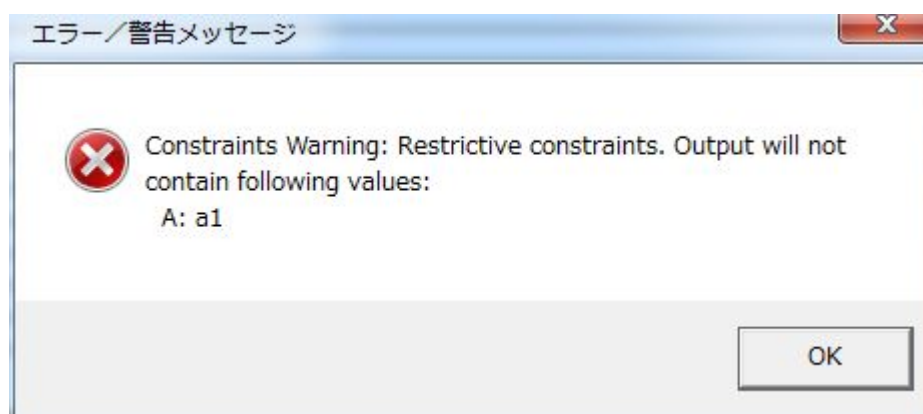


図 6－2 PICT が表示するメッセージの例

この例はパラメータ“A”の値“a1”が組み合わせに1つも含まれていないことを警告するメッセージです。この例では1つだけですが場合によっては5～6個の指摘がなされることもあります。

こうしたメッセージが出る原因は制約の指定に誤って相互に矛盾する複数の制約を指定したためです。間違った制約を突き止めるには指摘されたパラメータの値が組み合わせに表れないような矛盾した制約の指定を行っていないか制約表で各制約の関係を見直すことです。いくつも指摘された場合はどれか1つに的を絞って調べます。

このような警告が現れる最も単純な制約指定を以下に示します。

制約表		
パラメータ	制約1	制約2
A	a1	a2
B	b1	b1

図 6－3 矛盾した制約指定の例（その1）

この制約指定は制約条件の値とそれに対応する制約対象の値が不一致であるため値 a1 が組み合わせに現れないことになっています。この制約指定の誤りは一目見れば分かりますが、実際には多くの制約が関係した結果として、こうした制約条件と制約対象との矛盾が起こります。

基本として矛盾した制約かどうかは、同じ行に位置する2つのセルの値が異なる場合、値が同じでなくとも矛盾しない関係にあるかどうかを調べることにあります。

### 6. 3 画面を分割し制約表を記入しやすくする

制約の数が多くなると右に横スクロールしなければならないため、各パラメータの値の名称が見えなくなります。そのため左に横スクロールして名称を確認しなければなりません。また結果表の記入を行なう際もパラメータの値が見えず面倒です。これを頻繁に行なうことは煩わしいので簡単に記入できる方法を紹介します。

環境設定ボタンを押すと表示される「ウインドウ分割ショートカットキー」に任意の1文字（デフォルトは“e”）を入力しておくと、コントロールキーを押しながらショートカットキーを押すことで PictMaster のウインドウが2つ開かれ、上下に整列され、下側のウインドウはパラメータ欄と制約欄との間で分割されます。PictMaster が複数のワークシートで構成されている場合は、各ワークシートの環境設定フォーム

で指定されたショートカットキーが有効となります。

PictMaster v4.3 以降の新しいバージョンと v4.22 以前の古いバージョンを同時に開いている場合は、古いバージョンでのウインドウ分割ショートカットキーは無効となります。古いバージョンでウインドウ分割ショートカットキーを使用したい場合は、いったん Excel そのものを終了する必要があります。

図 6-5 にウインドウ分割ショートカットキーを押した後の画面例を示します。

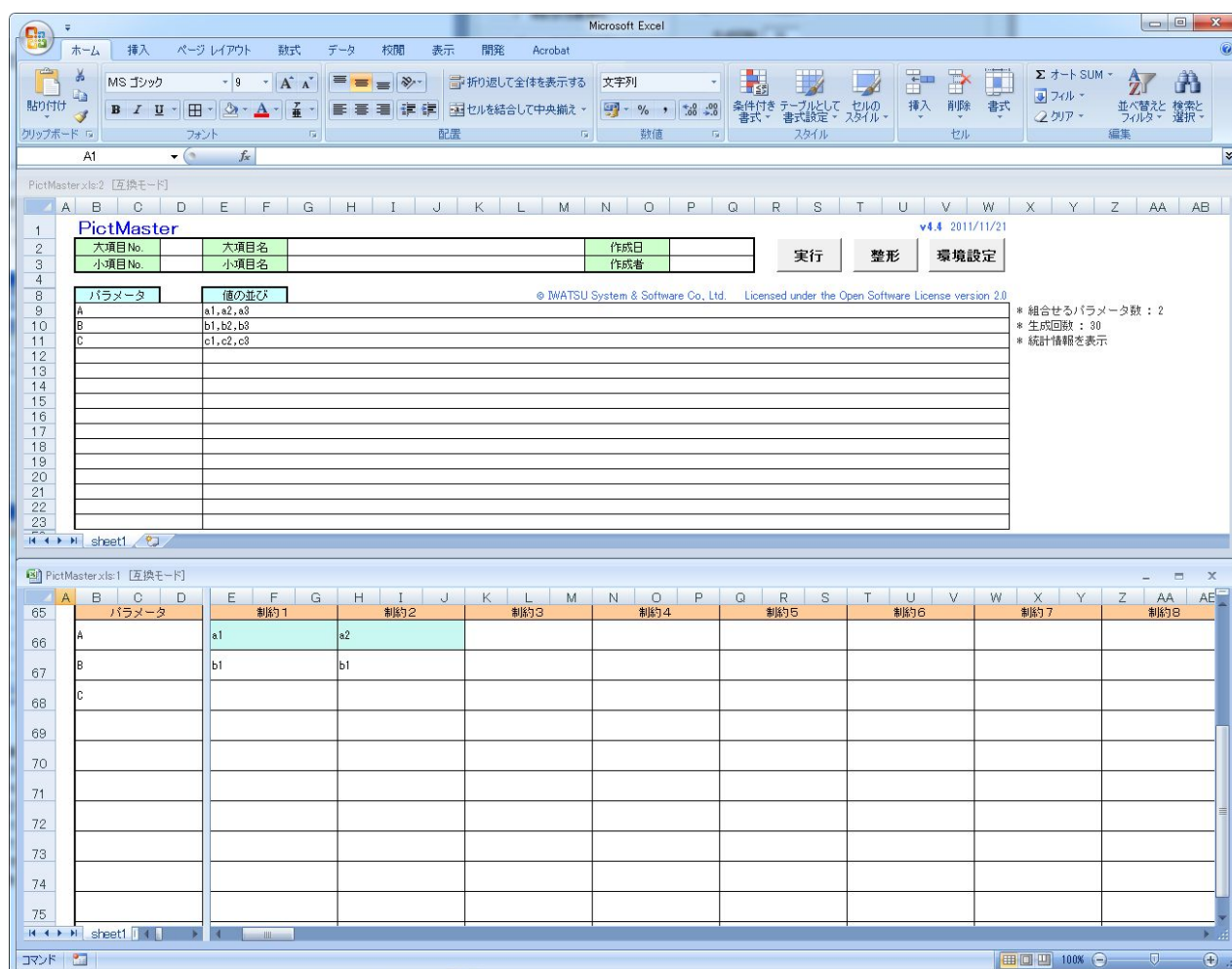


図 6-5 ウインドウ分割の画面例

この状態の画面では制約表を横スクロールしてもパラメータと値が常に見えているので多くの制約を必要とする場合に制約表への記入がしやすくなります。

この状態で再度コントロールキーを押しながらショートカットキーを押すと分割前の画面に戻ります。

なお他のシートや Book に一旦切り替えた場合、元の画面に戻るとウインドウ分割の状態が変化しています。ここで再度ウインドウ分割のショートカットキーを 2～3 回押すことで正しい分割画面とすることができます。他のシートを表示したい場合は画面の下の方のウインドウからシートを選択することで、選択したシートの本来のズーム（倍率）で表示させることができます。画面の上の方のウインドウから他のシートを選択すると常に 100% のズームで表示されます。



## 6. 4 要因列挙テストでテストケースを削減する

ソフトウェア開発ではまったくの新規開発は少なく、大部分は既存製品をベースにしたソフトウェアの再利用をとまなうソフトウェア開発です。

既存製品をベースにした新製品の開発では製品の各機能は、新製品で新たに追加された「**新規機能**」と、既存製品の機能をそのまま流用した「**既存機能**」に分けることができます。こうした場合、ソフトウェアテストを各機能に対して均等に工数を割り当てるのではなく、ソフトウェアの変更のない既存機能についてのテストは思い切って工数を減らし、新規機能のテストに工数を割り当てることになります。これを組み合わせテストに当てはめれば、新規機能については**要因組み合わせテスト**を多めに行うが、既存機能についてはテストケース数が多くなりがちな**要因組み合わせテスト**ではなく、組み合わせを行わない**要因列挙テスト**で済みますということになります。

要因列挙テストのテストケースを作成するには環境設定で「**組み合わせるパラメータ数**」に**1**を設定して生成を行なうだけです。

パラメータが5つ、各パラメータあたり5つの値を持つモデルでの要因列挙のテストケースの例を以下に示します。

表 6－1 要因列挙のテストケースの例

No.	A	B	C	D	E
1	a1	b2	c1	d3	e1
2	a2	b1	c5	d2	e4
3	a3	b3	c4	d4	e2
4	a4	b5	c3	d5	e3
5	a5	b4	c2	d1	e5

全てを要因列挙にする必要がない場合もあります。それは最も多くの数の値を持つパラメータとそれ以外のパラメータの持つ値の数に大きな開きがある場合です。

パラメータが5つ、8個の値を持つパラメータが2つ、残りは3個のパラメータを持つモデルを例として説明します。（図 6－6）

パラメータ	値の並び
A	a1, a2, a3, a4, a5, a6, a7, a8
B	b1, b2, b3, b4, b5, b6, b7, b8
C	c1, c2, c3
D	d1, d2, d3
E	e1, e2, e3

図 6－6 要因組み合わせと要因列挙を合成するのに適したモデル

このモデルのテストケース数は要因列挙テストでは8件となります。この場合、3個の値をもつパラメータが3つありますが、このパラメータについては、要因組み合わせテストとしてもテストケース数は $3 \times 3 = 9$ 件となり、8件とほとんど変わりありません。

ここで1つのテストケースで要因組み合わせと要因列挙を合成する方法を説明します。

環境設定で「**組み合わせるパラメータ数**」に**1**を設定します。「**サブモデルを使用する**」にチェックを入れます。サブモデルの記入欄に、要因組み合わせを行なうパラメータを記入し、**組み合わせ数に2**を記入します。（図 6－7）

サブモデル
C, D, E, 2

図 6-7 サブモデルで要因組み合わせを行なうパラメータを指定する。

このモデルでの生成結果を以下に示します。このテストケースではパラメータ A と B が要因列挙であり、パラメータ C、D および E が 2 パラメータ間の要因組み合わせとなっています。

表 6-2 要因組み合わせと要因列挙を合成したテストケース

No.	A	B	C	D	E
1	a1	b3	c1	d2	e1
2	a2	b7	c3	d2	e3
3	a3	b5	c2	d3	e1
4	a3	b7	c1	d1	e2
5	a4	b1	c2	d2	e2
6	a5	b8	c3	d1	e1
7	a6	b4	c2	d1	e3
8	a7	b6	c1	d3	e3
9	a8	b2	c3	d3	e2

このように多くの値を持つパラメータだけ選択して要因列挙とし、それ以外のパラメータを要因組合せとすることで、テストケース数をそれほど増加させることなく、要因組み合わせを混在させた要因列挙のテストケースを生成することができます。

この例では要因列挙テストとするパラメータが 2 つでしたが、3 つ以上とした場合でも各パラメータの組み合わせはランダムとなり、組み合わせが 1 対 1 となるようなことはありません。

要因組み合わせと要因列挙を混在したテストケースはそれほど多くはないと思いますが、要因列挙のみのテストケースはよく使われることになると思います。要因列挙であればツールを使わなくてもいいのではないかと考える方がいるかもしれませんが、制約がある場合はツールを使ったほうが間違いのないものになりますし、組み合わせがランダムとなるのでよいでしょう。

要因列挙テストは、要因の組み合わせによるソフトウェアの欠陥はないだろうということを前提としています。既存機能の流用であれば、要因を組み合わせてまでの徹底したテストは不要だとすることでテスト工数を大きく削減してよいと考えます。

## 6. 5 パラメータのマイナスの重み付け

サブモデルや拡張サブモデルは、特に重要と考えられる限定したパラメータのみ通常のパラメータ組み合わせ数とは異なる多くの組み合わせ数を適用することで、テストケースの増加を抑えつつ高い組み合わせ網羅度を確保することができます。これはパラメータのプラスの重み付けです。

これに対してパラメータのマイナスの重み付けも行なうことができます。特定のパラメータ間の組み合わせの重要度が低いと考えられる場合、それらのパラメータについて、サブモデルで組み合わせるパラメータ数に1を指定することでテストケースを削減することができます。

図6-8のモデルは、A社ルータとB社ルータの接続テストを、速度とOSを変えながら実施するものです。このモデルの通常のテストケースは表6-3の16件となります。

パラメータ	値の並び
A社ルータ	ルータA1, ルータA2, ルータA3
B社ルータ	ルータB1, ルータB2, ルータB3
速度	100M, 1000M
A側OS	Windows 7, Windows XP, Mac OS X, Linux
B側OS	Windows 7, Windows XP, Mac OS X, Linux

図6-8 A社ルータとB社ルータの接続テストのモデル

表6-3 通常のテストケース

No.	A社ルータ	B社ルータ	速度	A側OS	B側OS
1	ルータA1	ルータB1	100M	Windows XP	Linux
2	ルータA1	ルータB2	1000M	Windows 7	Mac OS X
3	ルータA1	ルータB2	100M	Mac OS X	Windows XP
4	ルータA1	ルータB2	100M	Linux	Windows 7
5	ルータA1	ルータB3	1000M	Linux	Windows XP
6	ルータA2	ルータB1	100M	Windows 7	Windows 7
7	ルータA2	ルータB2	1000M	Linux	Linux
8	ルータA2	ルータB3	1000M	Windows XP	Windows 7
9	ルータA2	ルータB3	100M	Mac OS X	Mac OS X
10	ルータA2	ルータB3	100M	Mac OS X	Linux
11	ルータA2	ルータB3	100M	Windows 7	Windows XP
12	ルータA3	ルータB1	1000M	Linux	Mac OS X
13	ルータA3	ルータB1	1000M	Mac OS X	Windows 7
14	ルータA3	ルータB1	1000M	Windows XP	Windows XP
15	ルータA3	ルータB2	100M	Windows XP	Mac OS X
16	ルータA3	ルータB3	100M	Windows 7	Linux

ここで、A側OSとB側OSの組み合わせの重要度が低いと考えられる場合、図6-9のサブモデルを用いると、A側OSとB側OS間の組み合わせは行われず、値が列挙されるだけになります。

この結果、サブモデルを用いない場合のテストケースが16件に対して、組み合わせるパラメータ数に1を指定した場合のテストケースは12件となります。このときの生成結果を表6-4に示します。



サブモデル
A側OS, B側OS, 1

図 6－9 マイナスの重み付けを行なうサブモデルの例

表 6－4 サブモデルでマイナスの重み付けをしたテストケース

No.	A 社ルータ	B 社ルータ	速度	A 側 OS	B 側 OS
1	ルータ A1	ルータ B1	1000M	Linux	Mac OS X
2	ルータ A1	ルータ B2	100M	Windows 7	Windows XP
3	ルータ A1	ルータ B2	100M	Windows XP	Linux
4	ルータ A1	ルータ B3	100M	Mac OS X	Windows 7
5	ルータ A2	ルータ B1	100M	Windows XP	Linux
6	ルータ A2	ルータ B2	1000M	Mac OS X	Windows 7
7	ルータ A2	ルータ B3	100M	Linux	Mac OS X
8	ルータ A2	ルータ B3	100M	Windows 7	Windows XP
9	ルータ A3	ルータ B1	1000M	Windows 7	Windows XP
10	ルータ A3	ルータ B1	100M	Mac OS X	Windows 7
11	ルータ A3	ルータ B2	100M	Linux	Mac OS X
12	ルータ A3	ルータ B3	1000M	Windows XP	Linux

この表では、A 側 OS と B 側 OS との間では組み合わせは行なわれず、単に値が列挙されているだけであることが分かります。そのことで組み合わせ数が減少しています。

ここで注意しておきたい点があります。値が列挙されている A 側 OS と B 側 OS は、同じ組み合わせとなっています。例えば A 側 OS の Linux は B 側 OS の Mac OS X とだけの組み合わせとなっています。重要でないとはいえ、組み合わせのパターンを増やした方が望ましいですから、ことです。制約の関係がないのであれば B 側 OS の列のみを Excel の並び替えの機能で並び替えることで組み合わせのパターンを増やすことができます（表 6－5）。

表 6－5 B 側 OS の列を並び変えて組み合わせのパターンを増やしたテストケース

No.	A 社ルータ	B 社ルータ	速度	A 側 OS	B 側 OS
1	ルータ A1	ルータ B1	1000M	Linux	Linux
2	ルータ A1	ルータ B2	100M	Windows 7	Linux
3	ルータ A1	ルータ B2	100M	Windows XP	Linux
4	ルータ A1	ルータ B3	100M	Mac OS X	Mac OS X
5	ルータ A2	ルータ B1	100M	Windows XP	Mac OS X
6	ルータ A2	ルータ B2	1000M	Mac OS X	Mac OS X
7	ルータ A2	ルータ B3	100M	Linux	Windows 7
8	ルータ A2	ルータ B3	100M	Windows 7	Windows 7
9	ルータ A3	ルータ B1	1000M	Windows 7	Windows 7
10	ルータ A3	ルータ B1	100M	Mac OS X	Windows XP
11	ルータ A3	ルータ B2	100M	Linux	Windows XP
12	ルータ A3	ルータ B3	1000M	Windows XP	Windows XP

この例のように特定のパラメータ間でのみ組み合わせが不要と考えられるパラメータについては、サブモデル定義でマイナスの重み付けを行なうことで組み合わせ数を削減することができます。

マイナスの重み付けは、値の数が多いパラメータに適用した場合に、組み合わせ数削減の効果を発揮します。

## 6. 6 ショートカットメニューを使って編集する

パラメータ欄と値の並び欄、制約表および結果表は専用のショートカットメニューを使って編集することができます。

第1列目を右クリックすると図6-10に示す編集専用のショートカットメニューが表示されます。

	パラメータ	値の並び
8		
9	A	a1, a2, a3
10	B	b1, b2, b3, b4
11	C	c1, c2, c3
12		d1, d2, d3
13		e1, e2, e3
14		
15		
16		
17		
18		

図6-10 行編集用のショートカットメニュー

制約表の制約タイトル欄を右クリックすると図6-11に示す編集専用のショートカットメニューが表示されます。

制約表			
パラメータ	制約1	制約2	制約3
A			
B	b1, b2		
C	c1	c2	
D		d1, d2	

図6-11 列編集用のショートカットメニュー

ショートカットメニューを使って行や列の移動、挿入および削除を行なうことができます。削除を行なった場合は次のショートカットメニューで「元に戻す」が表示され、削除前の状態に戻すことができます。一度に編集できる行または列は1つです。挿入と削除は最後の行や列では無効です。

## 6. 7 テストケース数を調整するひとつの方法

テスト計画を立てる上で実施するテストケース数をいくつと見積もるかは重要な事項です。多すぎれば実施できないテストが残ってしまうことになり、少なすぎれば市場障害のリスクが高くなります。そうした問題をクリアして決定したテストケース数であっても、テスト設計を進めていくと、見積もったテストケース数からの乖離が大きくなる場合があります。こうした場合、テスト対象のリスクに応じてテストケース数を簡単に調節することができれば大変助かります。

テストケースが、組み合わせテスト技法で作成したものであるならば、組み合わせるパラメータ数を変更することでテストケース数を調節することができます。例えば2パラメータ間の組み合わせとして生成したテストケースでも、比較的风险が低いと考えられれば組み合わせをせずに（1パラメータ間の組み合わせとし）要因を列挙したテストケースとすることで、テストケース数を大幅に削減することができます。

ただし、この方法は変更前と変更後のテストケース数の差が大きく、リスクに見合うだけのテストとなっているかについては十分とはいえない場合があります。こうした場合に、テストケース数をより細かく調節できる方法があります。ここでは図6－12のモデルを例にこの方法を説明します。

パラメータ	値の並び
A	a1, a2, a3, a4
B	b1, b2, b3, b4
C	c1, c2, c3
D	d1, d2, d3, d4, d5, d6
E	e1, e2
F	f1, f2
G	g1, g2, g3
H	h1, h2

図6－12 モデルの例

このモデルの2パラメータ間の組み合わせで生成されるテストケース数は25件です。要因列挙（1パラメータ間の組み合わせ）でのテストケース数は6件です。6件では少なすぎる場合は、環境設定の「組み合わせるパラメータ数」に1を設定し、サブモデルを使用します。サブモデルで2パラメータ間の組み合わせとするパラメータを指定することで生成されるテストケース数を細かく調節することができます。

例えば、図6－13のサブモデルを使用すると生成されるテストケース数は12件となります。（表6－6）

サブモデル
B, C, E, 2

図6－13 サブモデルの例

サブモデルで組み合わせるパラメータは比較的、値の数が少ないパラメータが含まれている必要があります。値の数が多パラメータだけではテストケース数の削減になりません。

表 6－6 サブモデルを使用して生成したテストケースの例（その 1）

No.	A	B	C	D	E	F	G	H
1	a1	b2	c3	d5	e1	f2	g3	h1
2	a1	b3	c3	d5	e2	f2	g2	h1
3	a1	b4	c1	d3	e2	f1	g3	h1
4	a1	b4	c3	d2	e2	f1	g3	h2
5	a2	b2	c1	d2	e2	f2	g1	h1
6	a2	b2	c2	d1	e2	f1	g1	h2
7	a3	b1	c2	d4	e2	f2	g2	h1
8	a3	b3	c1	d6	e1	f2	g1	h1
9	a3	b3	c2	d6	e2	f1	g2	h2
10	a4	b1	c1	d6	e1	f1	g2	h2
11	a4	b1	c3	d1	e1	f1	g3	h2
12	a4	b4	c2	d2	e1	f1	g2	h2

このモデルではパラメータ D と、パラメータ E またはパラメータ F またはパラメータ H をサブモデルで指定した場合もテストケース数は 12 件となります。

パラメータ C、E、F および G をサブモデルで指定した場合のテストケース数は 9 件となります。（表 6－7）

表 6－7 サブモデルを使用して生成したテストケースの例（その 2）

No.	A	B	C	D	E	F	G	H
1	a1	b4	c2	d1	e2	f1	g3	h1
2	a2	b1	c3	d2	e2	f2	g3	h1
3	a2	b3	c1	d4	e1	f2	g3	h2
4	a2	b3	c3	d3	e2	f1	g1	h2
5	a2	b4	c3	d2	e1	f1	g2	h2
6	a3	b2	c1	d1	e2	f1	g1	h1
7	a3	b2	c2	d5	e1	f2	g1	h1
8	a4	b3	c1	d4	e2	f1	g2	h1
9	a4	b3	c2	d6	e1	f2	g2	h2

サブモデルで指定したパラメータの間では 2 パラメータ間の組み合わせが網羅されます。生成されるテストケース数との兼ね合いがありますが、重要と考えられるパラメータをサブモデルで指定できれば理想的です。そうでない場合でも単純な要因列挙テストよりも組み合わせのパターンが多くなります。

## 6. 8 効率のよい無効値テストを行なうには

PICT の無効値テストの機能を使用せずに、制約表で制約指定を行なうことで、1つのテストケースには1つの無効値だけが含まれるようにすることができます。

例えば図6－14のモデルがあるとき、値 a1、b2、c3 が無効値である場合、PICT がそなえている無効値機能を使用すると表6－8のテストケースが得られます。

パラメータ	値の並び
A	~a1, a2, a3
B	b1, ~b2, b3
C	c1, c2, ~c3
D	d1, d2, d3
E	e1, e2, e3

図6－14 モデルの例

表6－8 PICT の無効値機能を使用したテストケース

No.	A	B	C	D	E
1	~a1	b1	c1	d3	e1
2	~a1	b3	c1	d1	e3
3	~a1	b3	c2	d2	e2
4	a2	~b2	c1	d3	e3
5	a2	b1	c2	d2	e2
6	a2	b3	~c3	d3	e2
7	a2	b3	c1	d1	e2
8	a2	b3	c2	d1	e1
9	a2	b3	c2	d3	e3
10	a3	~b2	c1	d1	e2
11	a3	~b2	c2	d2	e1
12	a3	b1	~c3	d2	e3
13	a3	b1	c1	d3	e1
14	a3	b1	c2	d1	e3
15	a3	b1	c2	d2	e1
16	a3	b3	~c3	d1	e1
17	a3	b3	c1	d2	e3
18	a3	b3	c2	d3	e2

このテストケースで、色がついているテストケースが無効値テストに相当します。一方、色のついていないテストケースは無効値テストではなく通常のテストケースです。本来、無効値テストを行ないたいのですが、PICT の無効値テストの機能では、無効値テスト以外のテストケースも生成されてしまいます。それでは生成されたテストケースのうち、無効値テストのテストケースだけ実行すればよいかというと、そうではありません。色のついた無効値テストのテストケースだけテストを実施した場合、組み合わせに網羅されない値が出てしまいます。例えばc2 とd1 の組み合わせが含まれていません。

PICT の無効値テストでは無駄なテストケースが多く生成されてしまいます。PICT の無効値テストの機能は使用せずに、制約表を用いて無駄のないテストケースを生成することができます。この例での制約表を用いた方法での制約表の記述内容を図6－15に示します。そしてその生成結果を表6－9に示します。

制約表					
パラメータ	制約1	制約2	制約3	制約4	制約5
A	a1	#a1	#a1	#a1	a1
B	#b2	b2		b2	#b2
C	#c3		c3	#c3	
D					
E					

制約表					
パラメータ	制約6	制約7	制約8	制約9	制約10
A		#a1	a1		
B	#b2	#b2		b2	
C	c3	c3	#c3	#c3	
D					
E					

図 6 - 1 5 制約表による無効値テストの制約指定の例

この制約表では、制約 1、制約 4、制約 7 で無効値同士の組み合わせが生成されないよう指定しています。残りの 6 個の制約指定で、無効値でない場合に組み合わせ可能な値を OR 条件で指定しています。

この制約表で生成したテストケースを次に示します。

表 6 - 9 制約表による無効値テストケース

No.	A	B	C	D	E
1	a1	b1	c1	d3	e1
2	a1	b1	c2	d2	e3
3	a1	b3	c1	d2	e2
4	a1	b3	c2	d1	e1
5	a2	b1	c3	d1	e2
6	a2	b2	c1	d3	e1
7	a2	b2	c2	d2	e1
8	a2	b3	c3	d3	e3
9	a3	b1	c3	d3	e1
10	a3	b2	c1	d1	e3
11	a3	b2	c2	d3	e2
12	a3	b3	c3	d2	e2

このテストケースには、無駄なテストケースは含まれていません。すべてが無効値テストのテストケースとなっています。その結果、テストケース数は PICT の無効値機能を使用した場合の 18 個に対して 12 個で済んでいます。この方法は無効値を含むパラメータの値の個数が多い場合に有効な方法です。

## 6. 9 間違いやすい制約指定の例

制約表で制約指定を行なう際、間違いやすい制約指定というものがあります。 今、図 6－2 1 に示すモデルにおいて、制約 1 で指定されなかった組み合わせのすべてについて、制約 2 で値 c1 以外とのみ組み合わせ可能と指定したとします。

パラメータ	値の並び
A	a1, a2, a3
B	b1, b2, b3
C	c1, c2, c3

制約表		
パラメータ	制約 1	制約 2
A	a1	#a1
B	b1	#b1
C	c1	#c1

図 6－1 6 間違いやすい制約指定の例

この制約指定で生成された組み合わせを表 6－1 0 に示します。

表 6－1 0 正しくない組み合わせの例

No.	A	B	C
1	a1	b1	c1
2	a1	b2	c1
3	a1	b2	c2
4	a1	b3	c1
5	a1	b3	c3
6	a2	b1	c1
7	a2	b1	c3
8	a2	b2	c3
9	a2	b3	c2
10	a3	b1	c1
11	a3	b1	c2
12	a3	b2	c3
13	a3	b3	c3

この組み合わせ結果をみると、制約 1 で指定されなかった組み合わせであるにもかかわらず、値 c1 との組み合わせが含まれています。 理由は、制約 2 の指定の意味が、値 a1 でなく、かつ、値 b1 でもない場合のみ、値 c1 以外との組み合わせとなる、というものだからです。 もともとの意図は、制約 1 で指定されなかった組み合わせのすべてを、値 c1 以外とするというものでした。

正しい制約指定を図 6－1 7 に示します。

パラメータ	値の並び
A	a1, a2, a3
B	b1, b2, b3
C	c1, c2, c3

© IWATSU

制約表			
パラメータ	制約1	制約2	制約3
A	a1	#a1	
B	b1		#b1
C	c1	#c1	#c1

図 6 - 1 7 正しい制約指定

この制約指定で生成された組み合わせを表 6 - 1 1 に示します。

表 6 - 1 1 正しい組み合わせの例

No.	A	B	C
1	a1	b1	c1
2	a1	b2	c2
3	a1	b3	c3
4	a2	b1	c3
5	a2	b2	c2
6	a2	b3	c3
7	a3	b1	c2
8	a3	b2	c3
9	a3	b3	c2

一見すると、図 6 - 1 6 の制約指定でよいように思われますが、AND 条件に一致しなかったすべての組み合わせは OR 条件で指定する必要があるということです。制約表は、他の制約指定の方法（例えば PICT の制約式など）に比べて制約指定が直感的に分かりやすいというメリットがありますが、今回の例のように例外的にうっかり間違った制約指定を行ってしまう可能性がありますので注意が必要です。



## 6. 1 0 制約指定が無視される例

PICT は制約指定の内容によっては、その制約指定を無視する場合があります。例えばパラメータがA～Fまでの6個あり、いずれも3個の値を持っているモデルがあるものとします。このモデルで図6－18の制約指定を行なった場合、その制約指定はPICTによって無視されます。この場合の生成結果の一例を表6－12に示します。

パラメータ	値の並び
A	a1, a2, a3
B	b1, b2, b3
C	c1, c2, c3
D	d1, d2, d3
E	e1, e2, e3
F	f1, f2, f3

制約表	
パラメータ	制約1
A	a2
B	b2
C	c2
D	
E	
F	f2

図6－18 モデルの例

表6－12 生成結果の例

No.	A	B	C	D	E	F
1	a1	b1	c1	d3	e3	f2
2	a1	b2	c1	d2	e2	f2
3	a1	b2	c2	d1	e3	f3
4	a1	b3	c1	d2	e1	f2
5	a1	b3	c3	d3	e3	f1
6	a2	b1	c2	d1	e1	f3
7	a2	b1	c3	d2	e3	f2
8	a2	b2	c1	d1	e1	f1
9	a2	b3	c1	d3	e2	f3
10	a3	b1	c2	d2	e2	f1
11	a3	b2	c3	d3	e1	f3
12	a3	b3	c1	d2	e1	f3
13	a3	b3	c2	d3	e3	f2
14	a3	b3	c3	d1	e2	f2

生成結果を見ると、制約指定で制約条件が a2, b2, c2 の場合、制約対象が f2 となるという組み合わせがありません。この理由は、制約指定で制約条件として、a2, b2, c2 という組み合わせを指定してい

ますが、Paiewise (All-Pairs) 法で生成する組み合わせは2つのパラメータの組み合わせを最少の数で実現するようになっています。そのため、a2, b2, c2 という3パラメータにわたる組み合わせが生成される可能性は少なくなります。したがって、PictMaster での制約指定では、基本的に制約条件で指定するパラメータは2つまでに限定すれば、その制約指定はかならず有効となります。例外として、他の制約指定によって制約対象として3つ以上のパラメータの組み合わせを指定した場合、それらのパラメータについては3つ以上のパラメータについても制約条件で指定することができます。

PICT の特性として、指定された制約条件が必ずしも生成されるわけではありません。今回の例のような制約指定を有効とするには、制約指定そのものを原型シートに記入(図6-19)して生成を行なうことで可能となります(表6-13)。このときの生成では制約指定は削除し、環境設定で「原型シートを使用」にチェックを入れます

	A	B	C	D	E	F
1	A	B	C	D	E	F
2	a2	b2	c2			f2

図6-19 原型シートで組み合わせを指定する

表6-13 指定した組み合わせを含んだ生成結果

No.	A	B	C	D	E	F
1	a1	b1	c2	d1	e1	f3
2	a1	b2	c3	d3	e3	f1
3	a1	b3	c1	d2	e2	f2
4	a2	b1	c1	d3	e3	f2
5	a2	b1	c3	d2	e2	f1
6	a2	b2	c2	d2	e1	f2
7	a2	b3	c2	d1	e2	f1
8	a2	b3	c3	d1	e3	f3
9	a3	b1	c1	d1	e1	f1
10	a3	b2	c1	d2	e3	f3
11	a3	b2	c2	d3	e2	f3
12	a3	b2	c2	d1	e3	f2
13	a3	b3	c3	d3	e1	f2

制約条件に3パラメータ以上を指定する場合は、その組み合わせが他の制約指定で制約対象に指定されているのでない限り、原型シートであらかじめ組み合わせを指定したほうがよいでしょう。

## 6. 1 1 時間のかかるテストケースを少なくする

パラメータの値によってはテストに時間のかかる場合があります。例えばパラメータがタイマ値で値として1秒と255秒の2つがある場合、255秒の場合はタイムアウトするまで、4分以上待たなければなりません。こうしたモデルの例を図6-20に示します。

パラメータ	値の並び
A	a1, a2, a3, a4
B	b1, b2, b3, b4
C	c1, c2, c3, c4
タイマ値	1, 255

図6-20 タイマ値を含む通常のモデル

このモデルの生成結果は表6-14のようになります。

表6-14 通常のモデルの生成結果

No.	A	B	C	タイマ値
1	a1	b1	c1	255
2	a1	b2	c3	255
3	a1	b3	c2	1
4	a1	b4	c4	255
5	a2	b1	c2	1
6	a2	b2	c4	1
7	a2	b3	c1	255
8	a2	b4	c3	1
9	a3	b1	c3	255
10	a3	b2	c2	1
11	a3	b3	c4	255
12	a3	b4	c1	1
13	a4	b1	c4	255
14	a4	b2	c1	1
15	a4	b3	c3	1
16	a4	b4	c2	255

生成結果では、タイマ値が255秒のテストケースが8件あります。このテストケースだけでテスト実施に30分以上かかることになります。

値の重み付けを使って、タイマ値が1秒のテストケースを255秒のテストケースより、多く生成することで、テストケース全体の作業時間を短縮することができます。具体的にはタイマ値が1秒の値に重み付けを行い、通常より約3倍多く生成されるようにします。その結果、タイマ値が255秒のテストケースは約3分の1に減少することになります。

このモデルを図6-21に示します。

パラメータ	値の並び
A	a1, a2, a3, a4
B	b1, b2, b3, b4
C	c1, c2, c3, c4
タイマ値	1(3), 255

図 6－2 1 重み付けを行なったモデル

このモデルの生成結果を表 6－1 5 に示します。

表 6－1 5 重み付けを行なったモデルの生成結果

No.	A	B	C	タイマ値
1	a1	b1	c1	1
2	a1	b2	c2	255
3	a1	b3	c3	1
4	a1	b4	c4	1
5	a2	b1	c2	1
6	a2	b2	c1	1
7	a2	b3	c4	255
8	a2	b4	c3	1
9	a3	b1	c3	255
10	a3	b2	c4	1
11	a3	b3	c1	1
12	a3	b3	c2	1
13	a3	b4	c2	1
14	a4	b1	c4	1
15	a4	b2	c3	1
16	a4	b3	c2	1
17	a4	b4	c1	255

重み付けを行なった生成結果では、タイマ値が 2 5 5 秒のテストケースが 8 件から 4 件に減少しています。4 件に減少してもすべてのペアは網羅されています。値の重み付けによる選択的なテストケースの削減は、対象となるパラメータの値の個数が、最も多くの値を持つ 2 つ以上のパラメータの値の個数より少ない場合です。そうでない場合、生成されるテストケース数が通常の生成の場合よりも増加します。そのことに注意して適用すれば、効率的なテストを行なうことができます。

## 6. 1 2 組み合わせにエイリアスのすべての値が出現するテストケースを生成する方法

エイリアスを使用することで、テストケースの数を合理的に削減することができますが、場合によっては組み合わせ結果に含まれないエイリアスの値が出てしまうことがあります。これは、エイリアスで指定した値の数が多く、他のパラメータの値の数が少ない場合に発生します。エイリアスで指定した値のすべてを用いなくてもすべてのペアの組み合わせが網羅されてしまうからです。

エイリアスのすべての値が組み合わせに現れるようにするには、組み合わせる相手側のパラメータの値の数を増やせばよいわけですが、そうすると組み合わせ数が大きく増加してしまい、せっかくエイリアスにした意味がなくなります。

いま図6-22のモデルで生成を行なうと、表6-16の組み合わせが得られます。この組み合わせには、エイリアスであるパラメータAの値a21～a25の5つのうち、a21とa22の2つしか出現していません。

パラメータ	値の並び
A	a1, a21   a22   a23   a24   a25, a3, a4
B	b1, b2
C	c1, c2
D	d1, d2
E	e1, e2

図6-22 モデルの例

表6-16 生成結果

No.	A	B	C	D	E
1	a1	b1	c2	d1	e2
2	a1	b2	c1	d2	e1
3	a21	b2	c2	d2	e2
4	a22	b1	c1	d1	e1
5	a3	b1	c1	d2	e2
6	a3	b2	c2	d1	e1
7	a4	b1	c1	d2	e2
8	a4	b2	c2	d1	e1

それではエイリアスのすべての値が組み合わせに出現するように、パラメータBにダミーの値を1つ追加します(図6-23)。ダミーの値を1つ追加することによって、パラメータBの値の数が3個となり、エイリアスを含むパラメータAを除く残りのパラメータの値の数2個と、パラメータBの値の数3個をかけた数が6となり、エイリアスの数5個以上の値となるため、すべてのエイリアスの値が組み合わせに出現するようになります。

そして原型シートを設け、それに図6-24に示す値を書き込みます。この原型シートでパラメータCまで指定しているのは、パラメータAとBだけでは、エイリアスのすべての値を用いてユニークなペア構成することができないためです。パラメータAの値はすべてエイリアスなので実質的な値の数は1つと見なされます。そのため、新たにパラメータCについても値を指定し、パラメータBとCとの組み合わせがすべてユニークなペアとなるようにします。こうしないと、実際に原型シートを用いて組み合わせを生成した場合に、ユニークでないペアの組み合わせは1つを残し、あとはPICTによって削除されてしまいます。

パラメータ	値の並び
A	a1, a21   a22   a23   a24   a25, a3, a4
B	b1, b2, dummy
C	c1, c2
D	d1, d2
E	e1, e2

図 6-23 パラメータ B にダミーの値を追加したモデル

	A	B	C	D	E
1	A	B	C	D	E
2	a21	b1	c1		
3	a22	b2	c2		
4	a23	dummy	c2		
5	a24	b1	c2		
6	a25	b2	c1		

図 6-24 ユニークなペアだけで構成した原型シート

以上のモデルと原型シートを用いて生成した組み合わせを表 6-17 に示します。

表 6-17 ダミーの値と原型シートを使用した生成結果

No.	A	B	C	D	E
1	a1	b1	c1	d1	e1
2	a1	b2	c2	d2	e2
3	a1	dummy	c1	d1	e1
4	a21	b1	c1	d1	e2
5	a22	b2	c2	d2	e1
6	a23	dummy	c2	d2	e2
7	a24	b1	c2	d1	e1
8	a25	b2	c1	d1	e1
9	a3	b1	c1	d2	e2
10	a3	b2	c2	d2	e2
11	a3	dummy	c2	d1	e1
12	a4	b1	c1	d1	e1
13	a4	b2	c2	d2	e1
14	a4	dummy	c2	d2	e2

この生成結果では、**テストケース数が 14 件**となっています。その内容をみると、ダミーの値がパラメータ A のエイリアス以外の値とも組み合わせられています。この組み合わせは本来不要なものです。そこで図 6-25 の制約指定（他の制約指定がある場合は、それに違反しないようにします）を追加し、ダミーの値がエイリアスの値とのみ組み合わせられるようにします。この場合の組み合わせ結果を表 6-18 に示します。

パラメータ	値の並び
A	a1, a21   a22   a23   a24   a25, a3, a4
B	b1, b2, dummy
C	c1, c2
D	d1, d2
E	e1, e2

パラメータ	制約1	制約2
A	a21	
B	dummy	
C	c2	
D	d1	
E	e1	

図 6-25 制約指定を追加したモデル

表 6-18 ダミーの値と原型シートそして制約指定を追加した生成結果

No.	A	B	C	D	E
1	a1	b1	c1	d2	e1
2	a1	b2	c2	d1	e2
3	a21	b1	c1	d1	e2
4	a22	b2	c2	d2	e1
5	a23	dummy	c2	d1	e1
6	a24	b1	c2	d2	e2
7	a25	b2	c1	d1	e1
8	a3	b1	c2	d1	e1
9	a3	b2	c1	d2	e2
10	a4	b1	c2	d2	e1
11	a4	b2	c1	d1	e2

この生成結果では、テストケース数が11件となりました。組み合わせ内容を見ると、エイリアスのすべての値が出現しており、かつ、ダミーの値が1個のみとなっています。この件数が、組み合わせにエイリアスのすべての値が出現する最少テストケース数となります。最後に dummy の値を適当な値に置き換えて完成です。

### 6. 1 3 テスト対象にふさわしいテスト技法を選択する

組み合わせテスト技法を使い慣れてくるにしたがい、ややもするとテスト対象の性質を見極めずにテスト技法として組み合わせテスト技法を適用してしまう間違いをおかす場合がでてくるようです。

テスト対象の中には複雑な論理を含むものや優先順位が関係するものがあります。例えば5つの要因があり、それらの値の組み合わせに応じて異なる結果となるような場合です。この場合は、5つの要因の全数組み合わせテストが必要となります。この場合の全数組み合わせテストとは言い換えれば**デシジョンテーブルテスト**ということです。

5つの要因の全数組み合わせとなると、要因の値の数にもよりますが、テストケース数がかかなり多くなります。通常は、**同じ結果となる組み合わせは1つを残して他は削除することでデシジョンテーブルを「圧縮」**することになります。ただし、デシジョンテーブルの圧縮は大きなリスクを伴います。圧縮を行なうということは、圧縮を行なった部分のプログラミングが正しく行なわれていることを前提にしていることとなります。しかし、**プログラミングが正しく行なわれている保証は何もありません**。テスト漏れを防ぐために圧縮せずにテストを行なうことを推奨します。

テスト対象がどのような性質を持っているかの見極めが不完全だと、デシジョンテーブルテストが必要なのに組み合わせテストを適用してしまう間違いをおかしてしまいます。この場合、組み合わせテストでは通常の場合、2つの要因の組み合わせしか網羅することが保障されないため、本来5つの要因の全数組み合わせが必要なのに部分的な組み合わせしか網羅されず、テスト漏れが起きてしまうこととなります。

テスト設計を行なう場合は、まず**テスト対象の性質を分析することからはじめ、テスト対象の性質を見極めたうえでそれに応じた適用すべきテスト技法を選択する**必要があります。

テスト対象によっては組み合わせテストとデシジョンテーブルテストの両方が必要となるケースもあります。こうした場合は、デシジョンテーブル全体を1つのパラメータと見なし、デシジョンテーブルの1つずつのテストケースをその要因の値として組み合わせテストに統合することがよいでしょう。



## 7. デシジョンテーブルテストへの適用

デシジョンテーブルは、テスト対象が複雑な論理を含む場合、優先度が関係する場合などに最適なテスト技法です。PictMaster はデシジョンテーブルのテストケースを生成することができます。

それではデシジョンテーブルの問題として一般に公開されているJaSST'07 Tokyo での「[三賢者、テストを語る \(DTvsCEGvsCFD\)](#)」の「入場料問題」(28 ページ目)を使うことにします。この問題を以下に示します。

### テーマ2 入場料問題

#### ・ 問題

- － ある遊園地の入場料に関する仕様です。

	個人	団体
一般	1200円	1000円
小学生	600円	500円
ただし六歳未満と六十五歳以上、及び県内在住の小学生は無料		

図 7－1 デシジョンテーブルを作成する入場料問題

デシジョンテーブルに PictMaster を適用する際は、デシジョンテーブルの各条件がパラメータとなり、条件のとりうる値 (Yes か No かなど) がそのままパラメータの値となります。そして**組み合わせるパラメータ数を実際のパラメータ数と同じに設定**します。これはデシジョンテーブルがパラメータ数のレベルを持つ木構造となっており、そのすべての組み合わせを生成する必要があるためです。その意味では全数組み合わせテストと同じです。

#### 7. 1 デシジョンテーブルのモデルの作成

図 7－1 の問題では、個人と団体は排他的関係にあるため、パラメータは個人のみとし、その値の Yes、No で個人と団体を表します。その他に必要なパラメータは、一般、小学生、6 歳未満、6 5 歳以上、県内在住、で全体では 6 個の パラメータとなります。いずれも値は Yes と No です。この問題で 6 歳未満、小学生、一般、6 5 歳以上は、それぞれ排他的関係にあるパラメータであり、同時には Yes とはなりません。

この問題を解くモデル (パラメータと値の並び、制約表および結果表) を図 7－2 に示します。

パラメータ	値の並び
6歳未満	Yes, No
小学生	Yes, No
一般	Yes, No
65歳以上	Yes, No
県内在住	Yes, No
個人	Yes, No

制約表						
パラメータ	制約1	制約2	制約3	制約4	制約5	制約6
6歳未満	Yes	No	No	No	Yes	No
小学生	No	Yes	No	No	No	No
一般	No	No	Yes	No	No	No
65歳以上	No	No	No	Yes	No	Yes
県内在住						
個人						

結果表						
結果内容	6歳未満	小学生	一般	65歳以上	県内在住	個人
1200円			Yes			Yes
1000円			Yes			No
600円		Yes			No	Yes
500円		Yes			No	No
無料				Yes		
無料	Yes					
無料		Yes			Yes	

図7-2 入場料問題のモデル

パラメータの県内在住と個人（団体）は、組み合わせに制約を持ちません。それ以外の6歳未満、小学生、一般、65歳以上は、それぞれ排他的関係となるよう、**制約1～4**で指定しています。**制約5と6**で、小学生でもなく一般でもない場合は、6歳未満か65歳以上のいずれかであることを指定しています（制約対象がOR指定となります）。

結果表を用いてパラメータの値の組み合わせに応じた入場料を自動的に設定するようにしています。デシジョンテーブルでは必ず結果表を使用することになります。

こうして生成した組み合わせを表7-1に示します。

表 7-1 生成された組み合わせ

No.	6歳未満	小学生	一般	65歳以上	県内在住	個人	結果内容
1	No	No	No	Yes	Yes	Yes	無料
2	No	No	No	Yes	Yes	No	無料
3	No	No	No	Yes	No	Yes	無料
4	No	No	No	Yes	No	No	無料
5	No	No	Yes	No	Yes	No	1000円
6	No	No	Yes	No	Yes	Yes	1200円
7	No	No	Yes	No	No	No	1000円
8	No	No	Yes	No	No	Yes	1200円
9	No	Yes	No	No	No	Yes	600円
10	No	Yes	No	No	Yes	No	無料
11	No	Yes	No	No	Yes	Yes	無料
12	No	Yes	No	No	No	No	500円
13	Yes	No	No	No	No	No	無料
14	Yes	No	No	No	No	Yes	無料
15	Yes	No	No	No	Yes	Yes	無料
16	Yes	No	No	No	Yes	No	無料

生成された組み合わせは16通りとなりました。アクション（入場料）を結果表で決定しています。徹底したテストが必要な場合は、この組み合わせでテストを行なうことになります。それほど徹底したテストが必要でない場合は、16通りの組み合わせの中からアクション（入場料）に影響を与えない組み合わせを削除し、テストケース数を削減することになります。

## 7. 2 テーブルを圧縮する

「整形」ボタンで生成結果の「結果内容」を第一優先のパラメータとして並べ替えを行います。こうすることで組み合わせが入場料ごとに並び、入場料に影響を与えない組み合わせを見つけやすくなります。

入場料に影響を与えない組み合わせ（入場料が同じで組み合わせに共通点がある組み合わせ）をそれぞれ異なる色で塗りつぶした組み合わせを表7-2に示します。

表 7-2 アクション（入場料）に影響を与えない組み合わせ

No.	6歳未満	小学生	一般	65歳以上	県内在住	個人	結果内容
5	No	No	Yes	No	Yes	No	1000円
7	No	No	Yes	No	No	No	1000円
6	No	No	Yes	No	Yes	Yes	1200円
8	No	No	Yes	No	No	Yes	1200円
12	No	Yes	No	No	No	No	500円
9	No	Yes	No	No	No	Yes	600円
1	No	No	No	Yes	Yes	Yes	無料
2	No	No	No	Yes	Yes	No	無料
3	No	No	No	Yes	No	Yes	無料
4	No	No	No	Yes	No	No	無料
10	No	Yes	No	No	Yes	No	無料
11	No	Yes	No	No	Yes	Yes	無料
13	Yes	No	No	No	No	No	無料
14	Yes	No	No	No	No	Yes	無料
15	Yes	No	No	No	Yes	Yes	無料
16	Yes	No	No	No	Yes	No	無料

異なる色で塗りつぶした組み合わせについて、各色に1つを残してあとは削除します。この際、入場料に影響を与えないパラメータについてはそうであることを示す意味で「－」を記入します。このパラメータの値は Yes でも No でも結果に影響を与えません。

最終的に決定したデシジョンテーブルを以下に示します。

表 7－3 テストケースを削減したデシジョンテーブル

No.	6歳未満	小学生	一般	65歳以上	県内在住	個人	結果内容
1	No	No	Yes	No	－	No	1000円
2	No	No	Yes	No	－	Yes	1200円
3	No	Yes	No	No	No	No	500円
4	No	Yes	No	No	No	Yes	600円
5	No	No	No	Yes	－	－	無料
6	No	Yes	No	No	Yes	－	無料
7	Yes	No	No	No	－	－	無料

この組み合わせは JaSST'07 Tokyo での「三賢者、テストを語る (DTvsCEGvsCFD)」の 34 ページ目の組み合わせと実質的に同じ組み合わせです。

従来の方法では、テスト対象からいきなりデシジョンテーブル作成となり、デシジョンテーブル作成の過程が明らかではありませんが、PictMaster を使用することで「モデル」の形でデシジョンテーブル作成の根拠を明らかな形で残すことができます。

## 8. 状態遷移テストへの適用

テスト対象が状態を持ち、イベントにより状態の遷移が起こる場合、状態遷移テストが有効なテスト技法となります。PictMaster は状態遷移テストのテストケースを生成することができます。

### 8. 1 簡単な状態遷移のテストケースの生成

パラメータには各状態の名称を採用し、パラメータの値には、その状態で有効なイベントの名称を採用します。なお、状態遷移の過程で通らない状態が発生するので、そのことを表すダミーの値「-」も定義することになります。

例として図8-1の状態遷移図を用いることにします。この図で円形の s0 から s2 は状態を表し、状態から他の状態に接続する矢印は状態遷移の方向を表し、矢印に併記されている e1～e3 はその状態で有効なイベントを表します。

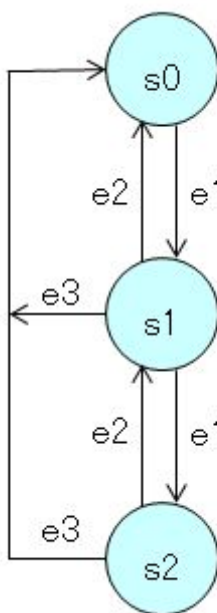


図8-1 状態遷移図の例（その1）

状態遷移は s0 から始まるものとします。この状態遷移図ですべての状態の遷移を網羅するために、パラメータとして状態 s0～s2 と s0' ～s1' を定義します。こうすることですべての状態を通る状態遷移のルートが得られます。

### 8. 2 状態遷移テストのモデルの作成

図8-1の状態遷移図をもとにした状態遷移テストのモデルを図8-2に示します。

パラメータ	値の並び
s0	e1, -
s1	e1, e2, e3, -
s2	e2, e3, -
s1'	e2, e3, -
s0'	end, -

図8-2 状態遷移テストのモデル（1／2）

制約表				
パラメータ	制約1	制約2	制約3	制約4
s0	-	e1		
s1	-	e1, e2, e3	e1	e2
s2	-		e2, e3	-
s1'	-			-
s0'	-			end

制約表				
パラメータ	制約5	制約6	制約7	制約8
s0				
s1	e3			
s2	-	e2	e3	
s1'	-	e2, e3	-	
s0'	end	end	end	

図8-2 状態遷移テストのモデル (2/2)

制約表への記入は、ある状態で有効なイベントを制約条件として記入し、イベントごとに遷移先の状態の欄に、その状態で有効なイベントを制約対象として記入します。遷移先の状態が他の状態の行をまたぐ場合は、その状態との組み合わせがないことを表すダミー値の「-」をその状態の欄に記入します。

制約対象として記入した1つ以上のイベントは、新たな制約欄に制約条件として1つずつ記入します。後はこれのくり返しです。

制約1は、制約表で値を指定するパラメータは2つ以上の値である必要があるために、ダミーとして設けた値について処理したものです。

### 8.3 状態遷移テストの組み合わせ結果

図8-2のモデルで生成した組み合わせ結果を表8-1に示します。

表8-1 状態遷移テストの組み合わせ結果

No.	s0	s1	s2	s1'	s0'
1	-	-	-	-	-
2	e1	e1	e2	e3	end
3	e1	e1	e2	e2	end
4	e1	e1	e3	-	end
5	e1	e2	-	-	end
6	e1	e3	-	-	end

この表でNo.1はダミーなので削除します。実際のテストケースは状態間の遷移の間にその状態で確認すべき確認内容を追加したのになります。(表8-2)

表8-2 実際のテストケース

No.	s0	確認内容	s1	確認内容	s2	確認内容	s1'	確認内容
1	e1	1	e1	2	e2	5	e3	7
2	e1	1	e1	2	e2	5	e2	8
3	e1	1	e1	2	e3	6	-	-
4	e1	1	e2	3	-	-	-	-
5	e1	1	e3	4	-	-	-	-

このテストケースでは、確認内容が8種類あります。Pairwise 法(All-Pair 法)によらない通常の方法で作成した状態遷移のテストケースは表8-3の通りです。

表8-3 通常の状態遷移のテストケース

No.	現状態	イベント	確認内容	次状態
1	s0	e1	1	s1
2	s1	e1	2	s2
3	s1	e2	3	s0
4	s1	e3	4	s0
5	s2	e2	5	s1
6	s2	e3	6	s0

このテストケースでは確認内容が6種類となっています。Pairwise 法によるテストケースに比較して2種類少なくなっています。この違いは、Pairwise 法によるテストケースでは、状態 s1' のイベント e2 と e3 による遷移が含まれていますが、通常のテストケースでは含まれていないためです。Pairwise 法による状態遷移のテストケースでは、状態の往復を網羅することができることから生じる違いです。その意味で Pairwise 法による状態遷移のテストケースは遷移ルートの網羅度の高いテストを行なうことができます。

#### 8. 4 状態遷移のくり返しを含むテストケースの生成

状態遷移テストでは1回の状態遷移では正常に動作するが、状態遷移を繰り返し繰り返すと発生するバグが少なからずあります。状態遷移のくり返しを含むテストケースを生成する方法を説明します。例として図8-3の状態遷移図を用いることにします。

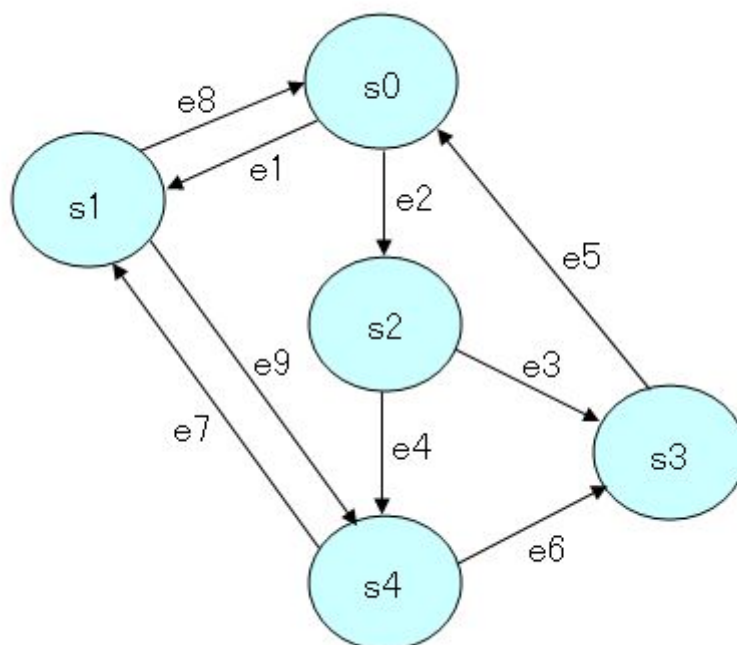


図8-3 状態遷移図の例（その2）

状態遷移は s0 から始まるものとしします。この状態遷移図ですべての状態の遷移を網羅し、かつ状態遷移のくり返しも網羅するために、パラメータとして状態 s0～s4 と s0' ～s4' そして s0'' を定義します。こうすることですべての状態を通る状態遷移のルートのセットと同じ状態を複数回通るルートのセットが得られます。



## 8. 5 繰り返しを含む状態遷移テストのモデルの作成

図8-3の状態遷移図をもとにした状態遷移テストのモデルを図8-4に示します。

パラメータ	値の並び
s0	e1, e2
s1	e8, e9, -
s2	e3, e4, -
s3	e5, -
s4	e6, e7, -
s0'	e1, e2, -
s1'	e8, e9, -
s2'	e3, e4, -
s3'	e5, -
s4'	end, -
s0''	end, -

パラメータ	制約1	制約2	制約3	制約4	制約5	制約6
s0	e1	e2				
s1	e8, e9	-	e8	e9		
s2		e3, e4	-	-	e3	e4
s3			-	-	e5	-
s4			-	e6, e7		e6, e7
s0'			e1, e2			
s1'						
s2'						
s3'						
s4'						
s0''						

パラメータ	制約7	制約8	制約9	制約10	制約11	制約12
s0						
s1						
s2						
s3	e5					
s4	-	e6	e7			
s0'	e1, e2	-	-	e1	e2	
s1'		-	e8, e9	e8, e9	-	e8
s2'		-			e3, e4	-
s3'		e5				-
s4'						-
s0''						end

図8-4 繰り返しを含む状態遷移テストのモデル (1/2)



制約表						
パラメータ	制約13	制約14	制約15	制約16	制約17	制約18
s0						
s1						
s2						
s3						
s4						
s0'						
s1'	e9					
s2'	-	e3	e4			
s3'	-	e5	-	e5		
s4'	end		end	-		
s0''	-		-	end		

図 8-4 繰り返しを含む状態遷移テストのモデル (2 / 2)

このモデルでは制約が 16 個となりました。制約の数は状態の数とイベントの数で大きく異なってきます。状態遷移図の制約表では、ある状態であるイベントがどの状態へ遷移できるのかをすべて記述します。このことは Pairwise 法 (All-Pairs 法) で最少テストケース生成を行なっても、**かならず一定のテストケース数となることを意味しています。**

## 8. 6 繰り返しを含む状態遷移テストの組み合わせ結果

図 8-4 のモデルで生成した組み合わせ結果を表 8-4 に示します。

表 8-4 繰り返しを含む状態遷移テストの組み合わせ結果

No.	s0	s1	s2	s3	s4	s0'	s1'	s2'	s3'	s4'	s0''
1	e1	e8	-	-	-	e1	e9	-	-	end	-
2	e1	e8	-	-	-	e1	e8	-	-	-	end
3	e1	e8	-	-	-	e2	-	e3	e5	-	end
4	e1	e8	-	-	-	e2	-	e4	-	end	-
5	e1	e9	-	-	e7	-	e8	-	-	-	end
6	e1	e9	-	-	e6	-	-	-	e5	-	end
7	e1	e9	-	-	e7	-	e9	-	-	end	-
8	e2	-	e3	e5	-	e1	e8	-	-	-	end
9	e2	-	e3	e5	-	e1	e9	-	-	end	-
10	e2	-	e3	e5	-	e2	-	e4	-	end	-
11	e2	-	e3	e5	-	e2	-	e3	e5	-	end
12	e2	-	e4	-	e7	-	e8	-	-	-	end
13	e2	-	e4	-	e7	-	e9	-	-	end	-
14	e2	-	e4	-	e6	-	-	-	e5	-	end

この例では組み合わせ数は 14 個となりました。PictMaster の最少テストケース生成を行なうと、最大数と最大数はともに 14 個となります。このことは、組み合わせ条件が制約表の記述によって完全に決定されていることを意味しています。逆にいえば、最少テストケース生成で最大数と最小数が異なる場合は、制約表の記述に不足した点があることになります。特に状態遷移の最後の状態で、後続く状態欄がある場合は、その状態の欄にダミー値の「-」を記入することを忘れないようにしましょう。図 8-2 でいえば制約 13 と制約 15 がそれにあたります。

表 8 の組み合わせ結果では、最少で 3 つの状態、最大で 5 つの状態を経由する最小限の組み合わせとなっています。ダッシュ (') のついた状態は、ついていない状態と同じ状態とみなします。そうすると同じ

状態を 2 回通る組み合わせが数多く網羅されていることが分かります。

PictMaster による状態遷移テストでは数多くの制約を指定することになります。組み合わせ生成エンジン PICT の処理能力により、あまり多くの状態を含む組み合わせの生成には非常に長い時間がかかります。**実用的にはパラメータ数が 20 未満に収まるようにしたほうが良いでしょう。**時間がかかる場合は、最少テストケース生成ではなく、デフォルトの条件による 1 回だけの生成にしましょう。

対象とする状態遷移図がこの制限を超えて大きい場合は、可能であれば大きな状態遷移図をいくつかのサブの状態遷移図に分けて扱うほうが良いかもしれません。

生成した組み合わせ結果には状態遷移のルートしか記述されていません。実際のテストケース仕様書では、**表 8-4** の各列の間に、その状態遷移の結果として起きることの確認内容を記述した列を挿入して完成となります。

## 9. 制御パステストへの適用

制御パステストはホワイトボックステストのテスト技法で、単体テストレベルで使います。プログラムから一連のテストパスを選んでそのパスを実行し、処理結果を確認するテスト技法です。

### 9. 1 従来の制御パステストとの違い

従来の制御パステストの方法は、まずプログラムの条件文以外を省略して制御構造のみを表した制御フローグラフを作成します。次にプログラムのソースリストを何部かコピーしたものを用意し、ソースリスト上に1つずつ制御パスを記入していきます。その記入内容を制御フローグラフにプロットします。この作業を繰り返し、制御フローグラフ上のすべての条件分岐を網羅するまで続けます。すべてを網羅した時点で、何部かのソースリスト上に書き込まれた個々の制御パスの集まりがテストケースとなります。こうした作業はかなり手間のかかる作業です。**PictMaster** は制御フローグラフを作成することなく、直接ソースリストからモデルを作成し、制御パステストのテストケースを生成することができます。

パスの網羅基準には大きく分けて、命令網羅、条件網羅、経路網羅の3つがあります。**PictMaster** は条件網羅と経路網羅に対応しています。通常の場合では、すべての条件文の真と偽の両方を通るパスを網羅します。これに対して Pairwise 法 (All-Pairs 法) では、すべての2つの条件文との組み合わせにおいて真と偽の両方を通るパスを網羅します。**PictMaster** の組み合わせ生成エンジン **PICT** は K-way テスト (組み合わせるパラメータ数が3個以上) に対応しています。それゆえ組み合わせるパラメータ数をすべての条件文の合計数とすることで経路網羅のテストケースも生成することができます。

**PictMaster** を使用することで制御パステストの網羅基準は以下ようになります。ここで括弧内の  $P_n$  などは組み合わせる条件文の数を表しています。 $P_{max}$  はすべての条件文の合計数を意味します。

1. 命令網羅 ( $P_0$ )
2. 通常の場合網羅 ( $P_1$ )
3. Pairwise 法による条件網羅 ( $P_2$ )
4.  $n$  個の条件文の組み合わせによる条件網羅 ( $P_n$ )
5. 経路網羅 ( $P_{max}$ )

以上の網羅基準によれば、**PictMaster** を使用することで  $P_1 \sim P_{max}$  までの制御パステストを実施することができます。

複数の条件文の組み合わせを網羅することで通常の場合網羅では発見できないバグを発見することができます。例えばリスト 9-1 のプログラムがあるとします。左端の番号は行番号です。

リスト 9-1 誤りのあるプログラム

```
1   if x1 = 3 then
2       y = 1
3   else
4       y = 0  ... 間違った代入文
5   end if
6   if x2 = 1 then
7       z = y
8   else
9       z = x3/y  ... 4行目を通てくると0除算となる代入文
10  end if
```

パスが4行目と9行目を通る場合のみ、0除算のバグが発見できます。

このプログラムの通常の場合網羅のテストケースは表 9-1 となる可能性があります。

表 9－1 通常の条件網羅のテストケースの例

No.	1if	2s1	4s2	6if	7s3	9s4
1	False	－	y=0	True	z=y	－
2	True	y=1	－	False	－	z=x/y

通常の条件網羅ではテストケース数が 2 となります。パラメータの名称には分かりやすいように行番号が追加されています。この例では 0 除算のバグが発生する 4 行目と 9 行目を通るパス（2 つの条件文がともに偽の場合）が網羅されていません。

次に Pairwise 法による条件網羅のテストケースの例を表 9－2 に示します。

表 9－2 Pairwise 法による条件網羅のテストケースの例

No.	1if	2s1	4s2	6if	7s3	9s4
1	False	－	y=0	True	z=y	－
2	False	－	y=0	False	－	z=x/y
3	True	y=1	－	False	－	z=x/y
4	True	y=1	－	True	z=y	－

Pairwise 法による条件網羅ではテストケース数が 4 となりました。この例では No. 1 で 0 除算のバグが発生する 4 行目と 9 行目を通るパスが網羅されています。このように **2 つの条件文の組み合わせで発生するバグを Pairwise 法による条件網羅で確実に発見することができます。**

この例では条件文だけでなく代入文もパラメータに採用していますが、基本的には条件文だけをパラメータにすれば十分です。テストケース数が通常の条件網羅の 2 に対して 4 と倍増していますが、実際のプログラム全体を対象としたテストケース数では、Pairwise 法による条件網羅は通常の条件網羅に対して約 50%前後の増加にとどまります。

## 9. 2 パラメータと値を決めるには

パラメータはパスの流れを変えるすべての文が対象となります。具体的には C 言語でいえば、**if else switch case for do while break continue** プログラム途中での **return** などです。言語によっては、**end if** などのようにパスの合流を意味する文がありますが、こうした文は基本的にはパラメータにする必要がありません。なぜなら次に続く条件文でパスの合流が表現されるからです。

パラメータの値は、条件文を含む文の場合は基本的には **TRUE と FALSE** と、その条件文を通らないパスを表現するためのダミーの値「－」を定義すれば十分です。条件文を含まない場合は、任意の名称の値とダミーの値を定義します。

例としてリスト 9－2 のプログラムで説明します。このプログラムのモデルを図 9－1 に示します。このプログラムでは制御パスを変える文が 2 つあります。

リスト 9－2 条件文が 2 つ続くプログラム

```

1  if (a == 1) { ... パスを変える文
2      x = 1;
3  }
4  if (b == 2) { ... パスを変える文
5      y = 2;
6  } else {
7      y = 3;
8  }
```

パラメータには制御パスを変える文である 1 行目と 4 行目を採用します。このモデルでは分かりやすさのために 6 行目の else もパラメータに含めています。パラメータが条件文の場合は、真と偽が値になります。条件文以外のパラメータの場合、値の名称は分かりやすい任意の名称とします。

パラメータ	値の並び
1if	TRUE, FALSE
4if	TRUE, FALSE
5else	else, -

制約表				
パラメータ	制約1	制約2	制約3	制約4
1if	TRUE	FALSE		
4if	TRUE, FALSE	TRUE, FALSE	TRUE	FALSE
5else			-	else

図 9-1 プログラムのモデルの例

このモデルでは、分かりやすいように else 節をパラメータに含めています。リスト 9-2 のように else 節の中にパスを変える文がなければ、else 節は省略してもかまいません。

制約表への記入は、制約対象 (TRUE と FALSE) を次の制約欄でそれぞれ独立した制約条件として記入します。

このモデルで生成された Pairwise 法 (All-Pairs 法) によるテストケースを表 9-3 に示します。

表 9-3 2つの if 文によるテストケース

No.	1if	4if	5else
1	False	False	else
2	False	True	-
3	True	False	else
4	True	True	-

実際のテストでは、この 4 つのパスを通る 4 つのテストデータのセットを用意します。この場合のテストデータのセットは以下の通りとなります。これは 1 つの例です。

- (1) a = 2, b = 1
- (2) a = 2, b = 2
- (3) a = 1, b = 1
- (4) a = 1, b = 2

### 9. 3 前の条件文によって次の条件文のパスが影響を受ける場合の値の決め方

条件文が 2 つ以上連結しているプログラムでは、前の条件文によって次の条件文のパスが影響を受ける場合があります。これは前の条件文で分かれたパスで同じ変数に異なる値を代入しており、次の条件文でその変数を判定対象としている場合などが該当します。こうした条件文を相互作用のある条件文と言います。

相互作用のある条件文の例をリスト 9-3 に示します。

リスト 9－3 相互作用のある条件文のプログラム

```

1  if (a == 1) {
2      a = a + 3;
3  } else {
4      b = b + 1;
5  }
6  if (b == 2) {
7      b = b + 2;
8  } else {
9      a = a + 1;
10 }
11 if (a > 3) {
12     x = 1;
13 } else {
14     x = 0;
15 }

```

このプログラムのモデル（図 9－2）と生成されたテストケース（表 9－4）を示します。ここでは分かりやすいように代入文もパラメータに含めています。

パラメータ	値の並び
1if	a == 1, a < 1
2s1	a = a + 3, -
3s2	b = b + 1, -
6if	b == 2, b < 2
7s3	b = b + 2, -
9s4	a = a + 1, -
11if	a > 3, a <= 3
12s5	x = 1, -
14s6	x = 0, -

制約表						
パラメータ	制約1	制約2	制約3	制約4	制約5	制約6
1if	a == 1	a < 1				
2s1	a = a + 3	-				
3s2	-	b = b + 1				
6if	b == 2, b < 2	b == 2, b < 2	b == 2	b < 2		
7s3			b = b + 2	-		
9s4			-	a = a + 1		
11if			a > 3, a <= 3	a > 3, a <= 3	a > 3	a <= 3
12s5					x = 1	-
14s6					-	x = 0

図 9－2 相互作用のある条件文のモデル

表 9-4 相互作用のある条件文のテストケース

No.	1if	2si	3s2	6if	7s3	9s4	11if	12s5	14s6	a, b
1	$a < 1$	-	$b = b + 1$	$b < 2$	-	$a = a + 1$	$a > 3$	$x = 1$	-	$a=3, b=0$
2	$a < 1$	-	$b = b + 1$	$b == 2$	$b = b + 2$	-	$a <= 3$	-	$x = 0$	$a=0, b=1$
3	$a == 1$	$a = a + 3$	-	$b < 2$	-	$a = a + 1$	$a <= 3$	-	$x = 0$	$a=?, b=1$
4	$a == 1$	$a = a + 3$	-	$b == 2$	$b = b + 2$	-	$a > 3$	$x = 1$	-	$a=1, b=2$

このテストケースの右端の列は、対応するテストケースの制御パスを通すための変数  $a$  と  $b$  の値のセットです。このテストケースでは No. 3 で変数  $a$  の値が 1 1 行目の if 文と矛盾しており決定できません。このような場合は新たに制約を追加して、すべてのテストケースの制御パスが通れる変数  $a$  と  $b$  の値が決定できるようにします。新しく追加した制約 (図 9-3) と生成されたテストケース (表 9-5) を示します。

制約表	
パラメータ	制約7
1if	$a == 1$
2s1	
3s2	
6if	
7s3	
9s4	
11if	$a > 3$
12s5	
14s6	

図 9-3 通れるパスを生成するために追加した制約

表 9-5 すべてのテストケースが通れるように修正されたテストケース

No.	1if	2si	3s2	6if	7s3	9s4	11if	12s5	14s6	a, b
1	$a < 1$	-	$b = b + 1$	$b < 2$	-	$a = a + 1$	$a > 3$	$x = 1$	-	$a=3, b=0$
2	$a < 1$	-	$b = b + 1$	$b < 2$	-	$a = a + 1$	$a <= 3$	-	$x = 0$	$a=0, b=0$
3	$a < 1$	-	$b = b + 1$	$b == 2$	$b = b + 2$	-	$a <= 3$	-	$x = 0$	$a=2, b=1$
4	$a == 1$	$a = a + 3$	-	$b == 2$	$b = b + 2$	-	$a > 3$	$x = 1$	-	$a=1, b=2$
5	$a == 1$	$a = a + 3$	-	$b < 2$	-	$a = a + 1$	$a > 3$	$x = 1$	-	$a=1, b=0$

追加した制約で、変数  $a$  が 1 のときは 1 1 行目の if 文で、 $a > 3$  とのみ組み合わせ可能としています。相互作用のある if 文で通れない矛盾したパスが生成された場合は、このような方法で新しい制約をすることによって矛盾のないテストケースを生成することができます。

#### 9. 4 1つの条件文が複数の条件式を含む場合の対処法

1つの条件文の中で2つ以上の条件式を含む場合があります。例えばリスト9-4の場合です。

リスト9-4 複数の比較判定を行なう条件文

```
1  if (a > 0 && c == 1) {  
2      x = x + 1;  
3  }  
4  if (b == 3 || d < 0) {  
5      y = 0;  
6  }
```

この例では、1行目と4行目の if 文でそれぞれ2つの条件式を含んでいます。1行目と4行目の個々の2つの条件式をまとめて1つずつの条件式として扱うこともできますが、こうした場合は個々の条件式がすべて評価される訳ではないため、どちらかの条件式にバグがあっても発見できない可能性があります。それはコンパイラが条件式をどのような順番で評価するかには依存する事項です。これに対して、個々の条件式をすべてパラメータの値として扱うことでバグの見逃しを防ぐことができます。こうした1つの条件文が複数の条件式から成り立っている場合に、個々の条件式まで網羅することを、**複合コンディションカバレッジ**を満たしていると言います。

リスト9-4のプログラムの、条件文の2つの条件式を独立の if 文としたフローチャートを図9-4に示します。

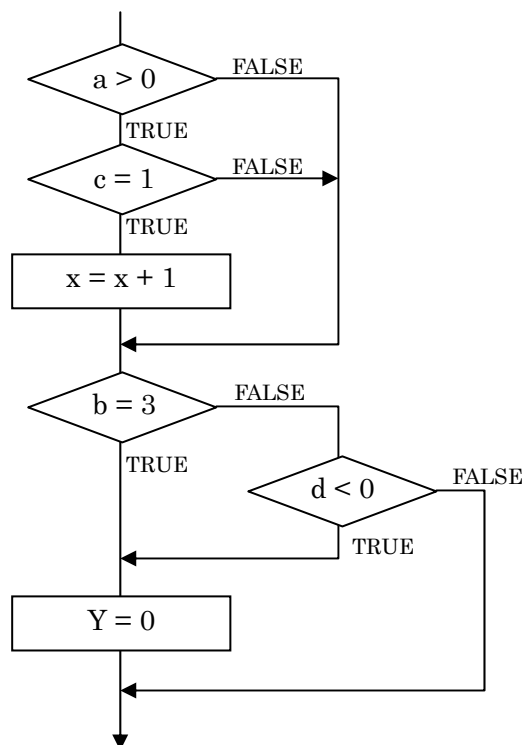


図9-4 複数の条件式を含む条件文を詳細化したフローチャート



リスト 9-4 の複合コンディションカバレッジを満たすモデルを図 9-5 に、そのテストケースを表 9-6 に示します。このモデルでは分かりやすいように代入文も含めています。1 つの if 文が 2 つのパラメータで構成されていることに注意してください。この例の場合は、コンパイラが if 文の左側から条件式を評価するものとしています。

パラメータ	値の並び
1if	$a > 0, a \leq 0$
1if'	$c == 1, c < 1, -$
2x	$x = x + 1, -$
4if	$b == 3, b < 3$
4if'	$d < 0, d \geq 0, -$
5y	$y = 0, -$

制約表				
パラメータ	制約1	制約2	制約3	制約4
1if	$a > 0$	$a \leq 0$		
1if'	$c == 1, c < 1$	-	$c == 1$	$c < 1$
2x		-	$x = x + 1$	-
4if		$b == 3, b < 3$	$b == 3, b < 3$	$b == 3, b < 3$
4if'				
5y				

制約表				
パラメータ	制約5	制約6	制約7	制約8
1if				
1if'				
2x				
4if	$b == 3$	$b < 3$		
4if'	-	$d < 0, d \geq 0$	$d < 0$	$d \geq 0$
5y	$y = 0$		$y = 0$	-

図 9-5 複合コンディションカバレッジを満たすモデルの例

表 9－6 複合コンディションカバレッジのテストケースの例 (Pairwise 法による条件網羅)

No.	1if	1if'	2x	4if	4if'	5y
1	a <= 0	-	-	b <> 3	d >= 0	-
2	a <= 0	-	-	b == 3	-	y = 0
3	a <= 0	-	-	b <> 3	d < 0	y = 0
4	a > 0	c <> 1	-	b == 3	-	y = 0
5	a > 0	c <> 1	-	b <> 3	d < 0	y = 0
6	a > 0	c <> 1	-	b <> 3	d >= 0	-
7	a > 0	c == 1	x = x + 1	b <> 3	d >= 0	-
8	a > 0	c == 1	x = x + 1	b == 3	-	y = 0
9	a > 0	c == 1	x = x + 1	b <> 3	d < 0	y = 0

テストケースは9個となりました。これは図 9－2 のプログラムを対象としていることによります。これに対して、Pairwise 法によらない通常の条件網羅のテストケースは表 9－7 となります。

表 9－7 複合コンディションカバレッジのテストケースの例 (通常の条件網羅)

No.	1if	1if'	2x	4if	4if'	5y
1	a <= 0	-	-	b == 3	-	y = 0
2	a > 0	c <> 1	-	b <> 3	d < 0	y = 0
3	a > 0	c == 1	x = x + 1	b <> 3	d >= 0	-

通常の条件網羅では複数の条件の組み合わせは網羅しないため、テストケース数は3個となります。

## 9. 5 プログラムがループを含む場合の記入方法

プログラムがループを含んでいる場合でもモデルを記述することができます。例として文字列 aStrings をもう 1 つの文字列 aText でサーチし、一致した最初の文字の桁位置を返す関数を取り上げてみます。一致しなかった場合は -1 が返されます。ソースリストは以下の通りです。VBA で書かれています。

この関数で Len(文字列) は文字列の長さを意味し、Mid(文字列, 桁数 1, 桁数 2) は文字列の先頭から桁数 1 の位置から桁数 2 の長さの文字を意味します。

リスト 9－5 ループを含むプログラム (文字列のサーチを行なう関数)

```

1 |Function StringSearch(aStrings As String, aText As String) As Integer
2 |
3 |    Dim i As Integer, j As Integer
4 |
5 |    If Len(aStrings) < Len(aText) Then
6 |        StringSearch = -1
7 |        Exit Function
8 |    End If
9 |    For i = 0 To Len(aStrings) - Len(aText)
10 |        j = 1
11 |        Do
12 |            If Mid(aStrings, i + j, 1) <> Mid(aText, j, 1) Then
13 |                Exit Do
14 |            End If
15 |            j = j + 1
16 |            If j = Len(aText) + 1 Then
17 |                StringSearch = i + 1
18 |                Exit Function
19 |            End If
20 |        Loop
21 |    Next i
22 |    StringSearch = -1
23 |
24 |End Function

```

リスト 9－5を見ると、この関数は2つのループが入れ子になった構造となっています。  
この関数の制御パステストのテストケースを生成するモデルを図 9－6 に示します。

パラメータ	値の並び
5if	TRUE, FALSE
7Exit	Exit F, -
9For	TRUE, FALSE, -
11Do	Do, -
12if	TRUE, FALSE, -
13Exit	Exit Do, -
16if	TRUE, FALSE, -
18Exit	Exit F, -
20Loop	Loop, -
21Next	Next, -
24End	End, -

制約表				
パラメータ	制約1	制約2	制約3	制約4
5if	TRUE	FALSE		
7Exit	Exit F	-		
9For	-	TRUE, FALSE	TRUE	FALSE
11Do	-		Do	-
12if	-		TRUE, FALSE	-
13Exit	-			-
16if	-			-
18Exit	-			-
20Loop	-			-
21Next	-			-
24End	-			End

制約表				
パラメータ	制約5	制約6	制約7	制約8
5if				
7Exit				
9For				
11Do				
12if	TRUE	FALSE		
13Exit	Exit Do	-		
16if	-	TRUE, FALSE	TRUE	FALSE
18Exit	-		Exit F	-
20Loop	-		-	Loop
21Next	Next		-	-
24End	-		-	-

図 9－6 ループを含むプログラムのモデル（その1）

このモデルでは代入文を含んでおらず、制御パスに影響を与える条件文などのみから成り立っています。  
生成したテストケースを表 9－8 に示します。

表 9－8 ループを含むプログラムのテストケース

No.	5if	7Exit	9For	11Do	12if	13Exit	16if	18Exit	20Loop	21Next	24End	Data #
1	FALSE	-	FALSE	-	-	-	-	-	-	-	End	*2
2	FALSE	-	TRUE	Do	TRUE	Exit Do	-	-	-	Next	-	*2, *3, *5
3	FALSE	-	TRUE	Do	FALSE	-	TRUE	Exit F	-	-	-	*3, *4, *5
4	FALSE	-	TRUE	Do	FALSE	-	FALSE	-	Loop	-	-	*2, *3
5	TRUE	Exit F	-	-	-	-	-	-	-	-	-	*1

右端の Data# は、関数の入力パラメータ（テストデータ）の識別番号であり、その内容は以下の通りです。

表 9－9 テストデータと関数の戻り値 (Value)

Data #	aStrings	aText	Value
*1	a	ab	-1
*2	abc	bcd	-1
*3	ababcab	abc	3
*4	a	a	1
*5	abc	c	3

表 9－8 から、すべての制御パスを網羅していることが分かります。ここでわかることは、**1つのテストデータで2つ以上の制御パスを通る場合がある**ということです。プログラムがループを含む場合、生成されたテストケースは、**ループの途中までの制御パスを含みます**。そのため、1つのテストデータで2つ以上の制御パスを通ることになります。もう1つ注意が必要なのは、Data # の **\*4** です。このテストデータは他のテストデータも通る制御パスを通っていますが、No. 3 の制御パスしか通りません。こうしたテストデータをもれなくテストする必要があります。

今回のプログラムでは、組み合わせるパラメータ数が1、2、すべて、のいずれの場合でも制御パスは5で同じ数です。条件網羅と経路網羅で制御パスの数が違ってくるのは、**条件文が直列に2つ以上接続されている場合**です。それも条件文の真と偽の双方で1つのルートに合流する場合です。どちらかがパスから外れる場合は除かれます。その意味で今回のプログラムには2つ以上の条件文が完全に直列になっている箇所がないため、条件網羅と経路網羅で同じ制御パスとなります。

このプログラムの性質上、ループについては、ループする、ループから抜ける、の2つしかありませんが、プログラムによっては境界値分析・同値分割の考え方をループに適用し、1回もループしない、1回だけループする、最大回数までループする、の3つのケースをカバーする方がよい場合もあります。

## 9. 6 制御パステストのまとめ

PictMaster を制御パステストのテストケース生成に適用する場合のポイントをまとめます。

- (1) プログラムの重要性などに応じて通常の条件網羅とするか、Pairwise 法による条件網羅とするかを決める。
- (2) 原則として制御パスを変える if 文などをパラメータに採用する。ただし、分かりやすさのために代入文も含めてよい。
- (3) パラメータの値は制御パスを変更する TRUE、FALSE などを採用する。ただし、分かりやすさのために実際の値を採用してもよい。この場合は境界値分析・同値分割の考え方を適用する。ダミーの値も定義する。
- (4) 生成したテストケースをすべて網羅するようにテストデータのセットを決める。

Pairwise 法 (All-Pair 法) による制御パステストのテストケースを生成するやり方には、こうしなければならないという重要な決まりごとはほとんどありません。それだけ記述の自由度が高い訳ですが、どの

ように記述するかによって制御パステストの質が決まると言えます。そしてこれは生成エンジンの処理能力の問題ですが、パラメータの数はむやみに多くすることはできないことに注意してください。

## 7. 困ったときは

### (1) 「'PictMaster.xls' が見つかりません。」というエラーメッセージが表示される

ローカルPC上で作成したBookを別の環境（他のPCなど）で開き、生成、整形、環境設定のボタンをクリックすると、「'PictMaster.xls' が見つかりません。ファイル名およびファイルの保存場所が正しいかどうか確認してください」といったメッセージが表示される場合があります。このメッセージが表示されたときは、当該ボタンを右クリックし、「マクロの登録…」を選び、表示されるマクロのリストから、ボタンの名称と同じマクロを選択し、OKをクリックすれば問題は解決します。

### (2) ウィンドウ分割のショートカットキーが動作しない

PictMaster v4.3以降の新しいバージョンとv4.22以前の古いバージョンを同時に開いている場合は、古いバージョンでのウィンドウ分割ショートカットキーは無効となります。古いバージョンでウィンドウ分割ショートカットキーを使用したい場合は、いったんExcelそのものを終了する必要があります。

### (3) 既存のBookをPictMasterのワークシートのコピー先に指定するとエラーとなる

既存のBook（PictMasterではない）をPictMasterのワークシートのコピー先に指定することはできません。PictMasterのBookに新規ワークシートを設けて、そこに既存のBookのワークシートの内容をコピー&ペーストしてください。

### (4) 「実行時エラー '62' : ファイルにこれ以上データがありません」というエラーメッセージが表示される

文字コード変換プログラム **nkf.exe** がPICTのあるフォルダ内になく、PictMasterのあるフォルダ内にもないとこのエラーメッセージが表示されます。これは最小テストケース生成のときで、1回だけの生成時は空白のシートが表示されます。PICTのあるフォルダ内に **nkf.exe** を置いてください。

### (5) Excel2007以降のExcelで編集して保存したファイルをExcel2003以前のExcelで開くと、制約表のセルの色が変わってしまう

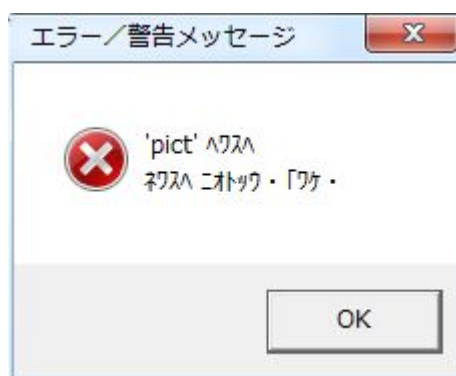
Excel2003以前とExcel2007以降とで、色の扱いが変わったことに起因する問題です。Excel2003以前のExcelで利用できる色数は64色まででしたが、Excel2007以降では制限がなくなりました。

### (6) 「カバレッジを表示」を指定して生成を行なうと時間がかかる

PictMasterは3-wayカバレッジを計算するために、内部で組合せるパラメータ数に3を指定して生成を行ないます。この生成に時間がかかるためです。

### (7) 「pict」の文字列を含む文字化けしたメッセージが表示される

PICTをインストールせずに実行ボタンをクリックすると次のメッセージが表示されます。



このマニュアルの[第0章](#)を参照し、PICTをインストールしてください。同時にnkf.exeをPICTがあるフォルダにコピーしてください。

**(8) 「フォルダ ○○○ 内にモデルファイル a.txt を新規に作成することができませんでした」というエラーメッセージが表示される**

モデルファイル a.txt が他のエディタで開かれているとこのエラーメッセージが表示されます。また、PictMasterがサーバー上にある場合に、管理者権限のないユーザでログインしているときにもこのエラーメッセージが表示されます。PictMasterがサーバー上にある場合、PictMasterは種々のファイルをProgram Filesフォルダ内に作成しようとするからです。

## 付録A 仕様

No.	項 目	値
1	パラメータの最大個数	50
2	パラメータあたり値の最大個数	50
3	制約表の最大制約数	50
4	結果表の最大条件数	50
5	結果表の1つの行に記入可能な値展開後の値の最大個数 (*1)	500
6	結果表で処理可能な生成結果の最大行数	65535
7	パラメータの組み合わせ数範囲	1～50
8	最小テストケース生成回数範囲	2～999
9	最小テストケース生成シードの範囲	0～65535
10	整形可能な最大行数	65535
11	原型シートの最大行数	65535
12	指定可能なサブモデルの最大数	PICT に依存
13	指定可能な拡張サブモデルの最大数	1
14	拡張サブモデルで指定可能な組み合わせるパラメータ数範囲	1～拡張サブモデルで指定されたパラメータの個数
15	値の重み付けの指定範囲	2 倍～10 倍
16	重複した組み合わせを削除可能な最大行数	65535
17	カバレッジを指定可能な組み合わせ数	2～6
18	指定可能な N-way カバレッジ範囲	30～95 [%]
19	指定カバレッジ生成試行回数範囲	3～10

\*1: 1つの行のすべてのパラメータの列に記入された値の個数の合計。値がエイリアスを含む場合は、エイリアスで指定した値の数をその値の個数に加算してカウントする。

## 付録B 制限事項

### 【重要な注意点】

多くのパラメータ、値、制約または組み合わせ数を指定した場合など、PICT の処理能力の限界により、テストケースを生成するまできわめて長時間かかる場合があります。こうした場合に処理を途中で止めた場合は、タスクマネージャを起動し、プロセスから pict.exe を選択し、プロセスの終了を行なってください。



## 付録C 拡張サブモデル（パラメータの重み付け）のアルゴリズム

任意の複数のパラメータのみに異なる T パラメータ間の組み合わせを指定できる拡張サブモデルのアルゴリズムを説明します。ここで T は 2 から拡張サブモデルで指定されたパラメータの個数までの範囲の値です。

徹底した組み合わせテストが必要な場合、組み合わせるパラメータ数に 3 を指定すると、生成されるテストケース数が膨大となり、テストを実施することが困難となる場合があります。

このような場合、すべてのパラメータではなく、特に重要と考えられる特定のパラメータについてのみ、3 パラメータの組み合わせとしたテストケースを生成することができれば、テストケース数が大幅に増加してしまうことを回避することができます。

PICT には、指定した特定のパラメータについて、他のパラメータとは異なる組み合わせ数を指定するサブモデルという機能があります。しかし、このサブモデルという機能は実際には使いにくい仕様となっています。

例えば 5 つの値を持つパラメータが 7 つ (A, B, C, D, E, F, G) あるモデルの場合、このうちの 3 つのパラメータ (A, B, C) を PictMaster のサブモデル欄で組み合わせ数に 3 を指定して生成すると、テストケース数は 6 2 5 件となります。サブモデルを使用せずに、環境設定で組み合わせるパラメータ数に 3 を指定して、生成した場合のテストケース数が 2 3 8 件ですから、それよりも 3 倍近いテストケース数となってしまいます。

どうしてこういう結果になるかというと、サブモデルで組み合わせ数に 3 を指定したパラメータ同士は、3 パラメータ間の組み合わせとなりますが、それらのパラメータのすべてが、サブモデルで指定されなかった残りのパラメータと 2 パラメータ間の組み合わせとなります。結果としてサブモデルで指定されたパラメータは 4 パラメータの組み合わせを網羅したものとなるからです。

行ないたいのは、サブモデルで指定されたパラメータ同士は 3 パラメータの組み合わせですが、サブモデルで指定されなかった残りのパラメータとは、個々のパラメータが独立して 2 パラメータの組み合わせとなることです。これが PICT の仕様では、サブモデルで指定された 3 つ以上のパラメータからなる 1 つ 1 つの組み合わせが、サブモデルで指定されなかった残りのパラメータと 2 パラメータの組み合わせとなるものですから、生成されたテストケース数は膨大な件数となってしまいます。

PICT のサブモデルのこうした問題を回避して、任意の複数のパラメータのみ T パラメータ間の組み合わせを生成するアルゴリズムは次の通りです。

1. T パラメータの組み合わせとしたい複数のパラメータについてサブモデルで T パラメータ間の組み合わせを指定します。同時に環境設定の組み合わせるパラメータ数に 1 を指定します。
2. 以上の条件でテストケースを生成します。このときのテストケース数を記憶します。
3. 生成されたテストケースを原型シートのワークシートに貼り付けます。
4. 環境設定の組み合わせるパラメータ数を最初にユーザが指定した値 N にします。この値は T よりも小さい値である必要があります。そして原型シートを使用する設定とします。同時にサブモデルを使用しない設定とします。
5. 以上の条件でテストケースを生成します。このときのテストケース数が 2 項でのテストケース数より多いことを確認します。
6. 前項 2 でのテストケース数を下回っている場合は、最も値の数が少ないパラメータの任意の値と同じ値を 1 つだけモデルに追加して、再度テストケースを生成します。

5 項で生成されたテストケース数が、2 項でのテストケース数以上であれば完了です。2 項でのテスト

ケース数を下回っている場合は、Tパラメータの組み合わせの一部が欠落していることになります。この現象は、原型シートに貼り付けたTパラメータの組み合わせすべてを取り込まなくても、Nパラメータ間のすべての組み合わせが網羅される場合に発生します。

こうなったときは、最も値の数が少ないパラメータの任意の値と同じ値を1つだけモデルに追加して、再度組み合わせを生成すると、原型シートに貼り付けたTパラメータの組み合わせが、すべて取り込まれてテストケースが生成されます。値を追加するパラメータとして最も値の数が少ないパラメータを選択するのは、テストケース数の増加を抑えるためです。この場合のテストケース数は、原型シートのテストケース数に最も値の数が多いパラメータの値の数を加えた件数となります。

1項から6項のアルゴリズムによるサブモデルの使い方を「**拡張サブモデル**」と呼ぶことにします。

原型シートに貼り付けるための組み合わせを生成した時のモデルと、原型シートを使用して組み合わせを生成した時のモデルに違いがあると、原型シートに貼り付けた組み合わせすべてが取り込まれてテストケースが生成されるようになります。これはPICTの仕様と考えられます。

図1に示すモデルとサブモデルを用いて通常の方法で生成したテストケースを図2に、拡張サブモデルで生成したテストケースを図3に示します。図2と図3では、Excelのフィルタを使って、サブモデルで指定したパラメータのうち、3つのパラメータの特定の値の組み合わせだけを抽出しています。同時に生成されたテストケース数も示します。

パラメータ	値の並び
A	1,2,3,4,5
B	1,2,3,4,5
C	1,2,3,4,5
D	1,2,3,4,5
E	1,2,3,4,5
F	1,2,3,4,5
G	1,2,3,4,5
サブモデル	
A, B, C, D, 3	

図1. モデルとサブモデル

	A	B	C	D	E	F	G
1	A	▼ B	▼ C	▼ D	▼ E	▼ F	▼ G
162		1	2	3	2	1	5
293		1	2	3	2	2	3
410		1	2	3	2	5	1
525		1	2	3	2	4	2
602		1	2	3	2	3	4

図2. 通常のサブモデルを用いて生成したテストケースのフィルタ適用結果

生成されたテストケース数：7 3 5

	A	B	C	D	E	F	G
1	A	▼ B	▼ C	▼ D	▼ E	▼ F	▼ G
56		1	2	3	5	4	5

図3. 拡張サブモデルを用いて生成したテストケースのフィルタ適用結果

生成されたテストケース数：1 5 2

環境設定で組み合わせるパラメータ数に 3 を指定して生成したテストケース数は 2 3 9 件でした。

通常サブモデルでは、サブモデルで指定した 3 つのパラメータの組み合わせが、サブモデルで指定されなかったパラメータのすべての値と組み合わせられています。一方、拡張サブモデルでは、サブモデルで指定した 3 つのパラメータの組み合わせは、サブモデルで指定されなかったパラメータの 1 つの値としか組み合わせられていません。このモデルの場合、通常サブモデルに対して拡張サブモデルでのテストケース数は約 1 / 5 となっています。

次に、2 つのパラメータ (A, B) のみ 3 パラメータの組み合わせとするには、サブモデルで次のように指定します。

A, B, 2 --- (1)

このときのテストケース数は 1 2 6 件となりました。

拡張サブモデルでは、(1) のサブモデルの記述は次のように直感的に分かりやすく記述することができます。

A, B, 3 --- (1)

## 付録D 希望するT-wayカバレッジを確保したテストケースを生成するアルゴリズム

希望する T-way カバレッジを確保したテストケースを生成するアルゴリズムは次の通りです。ここで T はユーザが環境設定の「組み合わせるパラメータ数」で指定したパラメータ数 N に + 1 した値です。

- ①. T パラメータ間の組み合わせでテストケースを生成する。
- ②. 生成したテストケースを PictMaster の原型シートに貼り付ける。
- ③. 組み合わせるパラメータ数にユーザが指定した組み合わせ数 N を指定して、原型シートを使用したテストケースを生成する。
- ④. 生成したテストケースの T パラメータ間網羅率の値に応じて次の手順を行なう。

### (1) 希望する T-way カバレッジよりも高い場合

- 1) 生成されたテストケースの件数を確認し、その件数よりも少ない件数に相当する原型シートの行に空きの行を挿入する。
- 2) 再度、テストケースの生成を行ない、3-way カバレッジを確認する。
- 3) 希望する T-way カバレッジよりも高い場合、原型シートに挿入する空きの行の位置をさらに小さい位置とする。希望する T-way カバレッジよりも低い場合、原型シートに挿入する空きの行の位置をより大きい位置とする。
- 4) 希望する T-way カバレッジに近いテストケースが生成されるまで、前項 2) ～ 3) の手順を繰り返す。

### (2) 希望する T-way カバレッジよりも低い場合

- 1) 原型シートのテストケースをすべてコピーし、生成されたテストケース全体に上書きし、生成されたテストケースの件数よりも大きい任意の位置の行に空きの行を挿入する。
- 2) テストケースの T-way カバレッジを確認する。
- 3) 希望する T-way カバレッジよりも高い場合、挿入する空きの行の位置をさらに小さい位置とする。希望する T-way カバレッジよりも低い場合、挿入する空きの行の位置をより大きい位置とする。
- 4) 希望する T-way カバレッジに近い値が得られるまで、前項 2) ～ 3) の手順を繰り返す。
- 5) 希望するテストケースが得られたら、上書きしたテストケースに挿入された空きの行から以降のテストケースをすべて削除する。

④の(1)の方法は、原型シートに貼り付けられた T-way カバレッジ 100% のテストケースのうち、N-way カバレッジが 100% となる部分までが、生成されるテストケースにそのまま流用されるという PICT の機能を利用しています。原型シートの途中に空きの行があると、それ以降のテストケースは PICT に取り込まれなくなります。

④の(2)の方法では、テストケースの途中に空きの行があると、それ以降は T-way カバレッジ測定の対象とはならないようにしています。

最終的に得られるテストケースでは、T-way カバレッジに少ない値が指定された場合でも、N-way カバレッジは 100% であることを保証する必要があります。

PictMaster では、30% ～ 95% の範囲内で**任意の T-way カバレッジを指定**することができます。T-way カバレッジ測定で必要となる比較対象データ取得のために 1 回だけ T パラメータ間の組み合わせ生成を行なう必要があるため、大きなモデルでは時間がかかる場合がありますが、それ以降は N パラメータ間の組み合わせ生成の繰り返しであり、それほど時間はかかりません。希望する T-way カバレッジよりも低い場合ならば、N パラメータ間の組み合わせ生成は必要なく T-way カバレッジの測定を繰り返すだけです。

希望する T-way カバレッジに近い値が得られるまでの繰り返し処理には、**2 分探索法を用いて最小の回数で希望する 3-way カバレッジに近い値を得ることができる**ようにしています。実測では、多くの場合 5 回の繰り返しで  $\pm 2\%$  以内に収まる結果となっています。

T-way カバレッジに低い値が指定された場合、モデルによっては最終的に得られるテストケースの T-way カバレッジが、指定された T-way カバレッジよりもかなり高い結果となる場合があります。これはそのモデルの N-way カバレッジを 100% とした場合の T-way カバレッジが、指定された値よりも高い場合です。T-way カバレッジを指定された低い値にしてしまうと、N-way カバレッジ 100% が確保できなくなるため、この場合の T-way カバレッジの指定は無効となり、N-way カバレッジ 100% が確保できる最小のテストケースが生成されます。

この処理中で生成されたテストケースは並び変えを行っていません。そのため、各テストケースの並びはほぼランダムと考えられ、先頭から任意の行までの部分的なテストケースの組み合わせのばらつきはほぼランダムと見なすことができます。

## 付録E 「既存テストケースのカバレッジを表示」の使い方

既存テストケースのカバレッジを表示する機能の最も一般的な使用法は、生成したテストケースの任意の行に空白行を挿入してカバレッジを表示させることです。テストケースの途中に空白行があると、それ以降のテストケースはカバレッジ表示の対象から除外されます。これは任意のテストケース数とした場合にカバレッジがいくつになるかが把握できることを意味しています。

この場合のテストケースは自動整形を行っていない必要があります。自動整形を行っていないと、生成されたテストケースの並びはほぼランダムとなるため、先頭から任意の行までの部分的なテストケースであってもテストケースに大きな偏りが生じることはありません。

ずっと以前に作成した既存テストケースのカバレッジを表示させるには次の手順を行なう必要があります。ここでは以前に作成した既存テストケースのファイル名は a.xls 以外である必要があります。ここでのパラメータの名称と値の名称は既存テストケースにおける名称と一致している必要はありません。パラメータの数とパラメータごとの値の数が一致していれば問題はありません。

1. テストケースに含まれるパラメータと値のすべてを PictMaster のパラメータ欄と値の並び欄に記入する。
2. テストケースにエイリアスの値が含まれている場合は、値の並び欄にそのエイリアスを記入する。
3. テストケースが生成された際の「組み合わせるパラメータ数」を環境設定で指定する。
4. この条件でテストケースを生成する。
5. 生成された a.xls のファイルを削除する。
6. カバレッジを測定したい既存テストケースのファイル名を a.xls にリネームし、ファイルを開く。
7. PictMaster で「既存テストケースのカバレッジを表示」を指定して実行する。

厳密に言うと、このほかに値の重み付けの有無、サブモデル指定の有無、原型シート使用の有無、制約指定使用の有無も既存テストケース生成時の指定に合わせてから一度テストケースを生成しておく必要があります。こうした条件があるため、ずっと以前に作成されたテストケースのカバレッジを表示させることはあまり現実的ではないかもしれません。

この他に直交表のカバレッジを表示するという使い方があります。直交表の場合は「組み合わせるパラメータ数」に2を指定します。これによって 2-way カバレッジと 3-way カバレッジを表示させることができます。

モデルファイル a.xls は既存テストケースと対応している必要があります。対応していない場合の動作は保証されません。