

# GSLetterNeo vol.152

2021 年 3 月

## PICT で始める組合せテストティング

熊澤 努 kumazawa @ sra.co.jp

### はじめに

---

PICT は Microsoft 社が開発しているテストケース生成ツールです。ソフトウェアシステムの入力、設定、構成の組合せが仕様として与えられたときに、それらの組合せをテストする組合せテストティングのために開発されました。組合せテストティングの考え方を使うことで、多数の組合せがある場合でも、系統的にテストケースを作ることができます。本稿では、簡単な例を使いながら PICT の使い方を紹介します。

### PICT を入手する

---

PICT を入手する方法は簡単です。Windows の場合には、PICT の公式サイト<sup>1</sup>から、公開されている実行ファイルをダウンロードするだけです。本稿を執筆している 2021 年 2 月現在、最新版は以下のサイトからダウンロードできます。

<http://www.pairwise.org/pict/win/pict.exe>.

---

<sup>1</sup> <https://github.com/microsoft/pict>

## 架空のショッピングサイトを例に

PCを販売するショッピングサイトの開発を行っているところを考えます。このサイトでPCを購入するユーザは、支払方法や、搭載するCPUなどのスペックを、下の表に示す選択肢から一つずつ選ぶことになっています。選択画面は一通り完成しているので、これらの組合せがきちんと選択できて、購入できるかテストをしようと思います。ここで購入できる商品には、例えば、「クレジットカード払いのときだけは赤色を選択できない」、など、ある項目の設定値によっては別の項目での選択肢に制限があるとします。このように、項目間の依存関係が正しく実装されているかテストする場合には、込み入ったテストケースを考える必要があるかもしれません。どのようなテストケースでテストをすればよいでしょうか。

項目	選択肢
支払方法	クレジットカード、コンビニ払い、口座振替
色	赤、白、銀、金
CPU	2GHz、2.3GHz、2.7GHz
ディスプレイ	13インチ、16インチ
キーボード	中国語、日本語、英語
搭載メモリ	16GB、24GB、32GB、64GB
SSD (ストレージ)	128GB、256GB、512GB、1TB、2TB

すぐに思いつくのが、全ての組合せをテストすることでしょう。しかし、この方法を採用すると、支払方法3通りについて、それぞれ色が4通り、CPUが3通り、・・・の組合せをテストしなければなりません。その組合せの総数は、 $3 \times 4 \times 3 \times 2 \times 3 \times 4 \times 5 = 4320$ 通りあります。これですべてのテストが終えられれば良いかもしれませんが、他にもテストすべき画面や開発作業が残っている場合には、あまり現実的ではなさそうです。テストケースを減らすことはできないでしょうか。

もう一つ、無作為（ランダム）に選択肢を決めてテストケースを作ることも考えられます。しかし、先に述べたような選択肢に依存関係がある場合には、この方法は、依存関係のテストを実行しない可能性が生じます。テストをする選択肢の網羅性を、完璧にするとはいわないまでも、ある程度は高めたいところです。

## ショッピングサイトのテストケースを作る

PICT を使って、上で見たショッピングサイトのテストケースを作ってみましょう。

PICT の入力は、設定項目と選択肢の一覧表を書き下したテキストファイルです。ユーザは、可能な選択項目と選択肢の値を列挙した以下のファイルを作ります。保存するファイル名は何でもよいので、ここでは ShoppingSite.txt とします。

```
#
# PC Shopping
#

Payment:      CreditCard, ConvenienceStore, Debit
Color:        Red, White, Silver, Gold
CPU:          2G, 2.3G, 2.7G
Display:      13, 16
Keyboard:     CN, JP, US
Memory:       16G, 24G, 32G, 64G
Storage:      128G, 256G, 512G, 1T, 2T
```

1行目から3行目の#で始まる行はコメントです。コメントはなくても構いません。5行目からがPICTが読み込む入力の情報です。各行に「**項目名: 設定値1, 設定値2, …**」と記述します。項目名は文字列で、設定値は数値または文字列で指定します。

Windows の場合、テストケースを生成するには、コマンドプロンプトを起動してダウンロードした pict.exe を実行します。下のコマンドを実行すると、結果は ShoppingSiteTestCases.txt に保存されます。出力ファイルを指定しないと、PICT は標準出力にテストケースを表示します。

```
> pict.exe ShoppingSite.txt > ShoppingSiteTestCases.txt
```

ShoppingSiteTestCases.txt は下のようになります。結果を見やすくするために、少し整形しました。

Payment	Color	CPU	Display	Keyboard	Memory	Storage
CreditCard	Gold	2.7G	13	CN	24G	2T
ConvenienceStore	Gold	2G	16	JP	64G	1T
CreditCard	Silver	2.3G	16	US	16G	256G
Debit	Red	2G	13	US	32G	2T
ConvenienceStore	White	2.3G	13	JP	64G	2T
Debit	White	2.7G	16	US	24G	1T
ConvenienceStore	Red	2.7G	16	CN	64G	128G
CreditCard	Silver	2G	13	JP	24G	128G
Debit	Gold	2.3G	13	CN	16G	512G
CreditCard	Red	2.7G	16	JP	24G	512G
CreditCard	Silver	2.3G	13	CN	32G	1T
ConvenienceStore	White	2.3G	16	US	32G	128G
Debit	Red	2.7G	13	JP	16G	256G
ConvenienceStore	White	2G	13	CN	24G	256G
Debit	Gold	2G	13	US	16G	128G
ConvenienceStore	Silver	2.3G	16	CN	16G	2T
Debit	Silver	2G	13	US	64G	512G
CreditCard	Red	2.3G	13	JP	64G	256G
ConvenienceStore	Red	2.7G	13	JP	16G	1T
Debit	Gold	2.7G	16	JP	32G	256G
ConvenienceStore	White	2.3G	13	CN	24G	512G
Debit	Silver	2.7G	13	CN	32G	512G
CreditCard	White	2.7G	13	CN	16G	1T

結果は表形式になっています。1 行目は項目名を並べたタイトル行です。2 行目以降、1 行につき 1 つのテストケースが挙げられています。例えば、2 行目は、「支払方法をクレジットカード、色を金、CPU を 2.7GHz、ディスプレイを 13 インチ、・・・、SDD が 2T」という条件のテストケースであることを意味します。テストケースの数が、4320 通りから 23 通りにまで少なくなっている点に注目です。

## 生成されたテストケースの特徴を知る

上のテストケースは、ペアワイズ法（全ペア法）といわれる方法に従って生成されています。ペアワイズ法は、「**どの二つの項目についても、選択肢の全ての組合せが一回以上列挙されている**」という性質を満たすようにテストケースを生成する方法です。例えば、支払方法と色の二つの項目について見ると、下に赤字で示すように、選択肢の可能な組合せが残らず列挙されていることが分かります。また、メモリーと SDD（ストレージ）についても選択肢

の全組合せが列挙されています。こちらは青字で示します。その他のどの二組の項目についても同様です。

Payment	Color	CPU	Display	Keyboard	Memory	Storage
<b>CreditCard</b>	<b>Gold</b>	2.7G	13	CN	<b>24G</b>	<b>2T</b>
<b>ConvenienceStore</b>	<b>Gold</b>	2G	16	JP	<b>64G</b>	<b>1T</b>
<b>CreditCard</b>	<b>Silver</b>	2.3G	16	US	<b>16G</b>	<b>256G</b>
<b>Debit</b>	<b>Red</b>	2G	13	US	<b>32G</b>	<b>2T</b>
<b>ConvenienceStore</b>	<b>White</b>	2.3G	13	JP	<b>64G</b>	<b>2T</b>
<b>Debit</b>	<b>White</b>	2.7G	16	US	<b>24G</b>	<b>1T</b>
<b>ConvenienceStore</b>	<b>Red</b>	2.7G	16	CN	<b>64G</b>	<b>128G</b>
CreditCard	Silver	2G	13	JP	<b>24G</b>	<b>128G</b>
<b>Debit</b>	<b>Gold</b>	2.3G	13	CN	<b>16G</b>	<b>512G</b>
<b>CreditCard</b>	<b>Red</b>	2.7G	16	JP	<b>24G</b>	<b>512G</b>
CreditCard	Silver	2.3G	13	CN	<b>32G</b>	<b>1T</b>
ConvenienceStore	White	2.3G	16	US	<b>32G</b>	<b>128G</b>
Debit	Red	2.7G	13	JP	16G	256G
ConvenienceStore	White	2G	13	CN	<b>24G</b>	<b>256G</b>
Debit	Gold	2G	13	US	<b>16G</b>	<b>128G</b>
<b>ConvenienceStore</b>	<b>Silver</b>	2.3G	16	CN	<b>16G</b>	<b>2T</b>
<b>Debit</b>	<b>Silver</b>	2G	13	US	<b>64G</b>	<b>512G</b>
CreditCard	Red	2.3G	13	JP	<b>64G</b>	<b>256G</b>
ConvenienceStore	Red	2.7G	13	JP	<b>16G</b>	<b>1T</b>
Debit	Gold	2.7G	16	JP	<b>32G</b>	<b>256G</b>
ConvenienceStore	White	2.3G	13	CN	24G	512G
Debit	Silver	2.7G	13	CN	<b>32G</b>	<b>512G</b>
<b>CreditCard</b>	<b>White</b>	2.7G	13	CN	16G	1T

ペアワイズ法に従ってテストケースを生成すると、先に述べた「クレジットカード払いのときだけは赤色を選択できない」という、二項目間の選択肢に制限がある場合をテストすることができます。上のテストケースを見ると、支払方法がクレジットカード払いのときに、赤色を選択できないこと、そして、赤色以外の色を正しく選択できることのチェックできることが分かります。また、クレジットカード払い以外の支払方法のとき赤色を選択できるかどうか、もテストが可能です。

組合せテストの用語では、項目のことを、不具合を起こす可能性のある要因とみて、因子と呼びます。選択肢のことは水準と呼びます。これらは実験計画法で使われている用語です。ペアワイズ法は、「複数の水準からなる二つの因子が関連（交互作用といいます）して、不具合を引き起こすかもしれない」と考えて、その関連性をテストする方法です。

PICT は、二つに限らず、三つ以上の項目（因子）についての交互作用をもとにテストするテストケース（カバリングアレイと呼ばれます）を生成することもできます。ただし、経験

的には、多くの因子の交互作用を考慮する必要はあまりないとされています<sup>2</sup>。あまりに多くの因子が関連する不具合を検出するテストを実施しなければならない場合には、実は小数の因子しか関連していないのでは、と仕様を見直してみることも有効そうです<sup>3</sup>。

## テストケースに制約を設定する

PICT には、特定の条件を満たすテストケースを生成しない、といった制約を付与することができます。例えば、「クレジットカード払いのときだけは赤色を選択できない」というケースはすでに十分テストがなされている場合を考えます。今生成しようとしているテストに、クレジットカード払いであり、しかも、色が赤というケースを含めないようにしてみます。そのために、PICT への入力である ShoppigSite.txt の最後の行に、以下のように制約を追加します。

```
#
# PC Shopping
#

Payment:      CreditCard, ConvenienceStore, Debit
Color:        Red, White, Silver, Gold
CPU:          2G, 2.3G, 2.7G
Display:      13, 16
Keyboard:     CN, JP, US
Memory:       16G, 24G, 32G, 64G
Storage:      128G, 256G, 512G, 1T, 2T

IF [Payment] = "CreditCard" THEN [Color] <> "Red";
```

この制約は、「クレジットカード払いならば、色は赤ではない」という制約です。因子である項目はかっこ[]で、水準である選択肢は二重引用符""で囲います(数値には二重引用符は不要です)。<>は等しくないこと(≠)を意味しています。

---

<sup>2</sup> D. R. Kuhn, R. N. Kacke and Y. Lei, "Practical Combinatorial Testing," *NIST Special Publication 800-142*, National Institute of Standards and Technology, U.S. Department of Commerce, 2010. <http://core.ecu.edu/STRG/materials/SP800-142-101006.pdf>

<sup>3</sup> 組合せテストイングにおけるテストケースの設計の指針については、以下の文献を見てください。

秋山浩一：「組合せテストの設計」, 情報処理, Vol. 49, No.2, 2008.

この入力でテストケースを生成すると、以下のようになります。赤字で示すように、「CreditCard Red ...」の組合せが生成されていない点に注意してください。

Payment	Color	CPU	Display	Keyboard	Memory	Storage
<b>CreditCard</b>	<b>Gold</b>	2.7G	13	CN	24G	2T
ConvenienceStore	Gold	2G	16	JP	64G	1T
<b>CreditCard</b>	<b>Silver</b>	2.3G	16	US	16G	256G
Debit	White	2G	13	US	32G	2T
Debit	Red	2.3G	13	CN	64G	256G
ConvenienceStore	Red	2.7G	13	JP	16G	128G
ConvenienceStore	White	2.7G	16	CN	32G	512G
Debit	Silver	2.3G	16	JP	32G	128G
<b>CreditCard</b>	<b>Silver</b>	2G	13	CN	16G	1T
ConvenienceStore	White	2.3G	16	JP	24G	256G
ConvenienceStore	Silver	2.7G	16	US	64G	2T
Debit	Red	2G	16	US	24G	512G
Debit	Gold	2.7G	16	US	32G	256G
<b>CreditCard</b>	<b>White</b>	2.3G	13	US	64G	1T
<b>CreditCard</b>	<b>Silver</b>	2G	13	US	24G	128G
ConvenienceStore	Red	2.3G	16	JP	32G	2T
<b>CreditCard</b>	<b>Gold</b>	2.3G	13	JP	16G	512G
Debit	Silver	2G	13	CN	64G	512G
Debit	Red	2.7G	13	JP	32G	1T
Debit	White	2.3G	13	JP	16G	2T
ConvenienceStore	Gold	2G	16	CN	64G	128G
ConvenienceStore	Red	2G	13	JP	24G	1T
<b>CreditCard</b>	<b>White</b>	2.7G	16	CN	32G	128G
<b>CreditCard</b>	<b>Gold</b>	2G	13	JP	16G	256G

## おわりに

今回は、テストケース生成ツールである PICT を使い、ペアワイズ法によりテストケースを生成してみました。組合せテストは、工夫次第で単体テストでもシステムテストでも利用でき、しかも気軽に使えるツールだと思います。PICT に関しては、Excel 上で実行する PictMaster<sup>4</sup>というツールも公開されています。PictMaster は、大阪大学で開発されているテストケース生成ツール CIT-BACH<sup>5</sup>を実行することもできます。ぜひ試してみてください。

<sup>4</sup> <https://ja.osdn.net/projects/pictmaster/>

<sup>5</sup> [https://ja.osdn.net/users/t-tutiya/pf/cit\\_bach/wiki/FrontPage](https://ja.osdn.net/users/t-tutiya/pf/cit_bach/wiki/FrontPage)

GSLetterNeo Vol.152  
2021年3月20日発行  
発行者 株式会社 SRA 先端技術研究所

編集者 土屋正人  
バックナンバー <http://www.sra.co.jp/gslletter>  
お問い合わせ [gsneo@sra.co.jp](mailto:gsneo@sra.co.jp)



株式会社SRA

〒171-8513 東京都豊島区南池袋 2-32-8

夢を。



夢を。Yawaraka Innovation  
やわらかいのバージョン