

JaSST' 11 CEDEC

ソフトウェアテスト技法入門

～明日からのテストのために～

マルチパラダイムシステム
北都システム

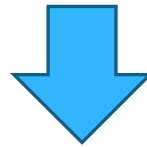
中野 直樹
長谷川 聡

目次

- * テスト技法とは
- * 同値分割、境界値分析
- * デイシジョンテーブル
- * 直交表を用いた組合せテストの技法

テスト技法とは

テスト技法ってなんでしょう？



効率よくテストを設計(デザイン)する
ための手法やテクニック

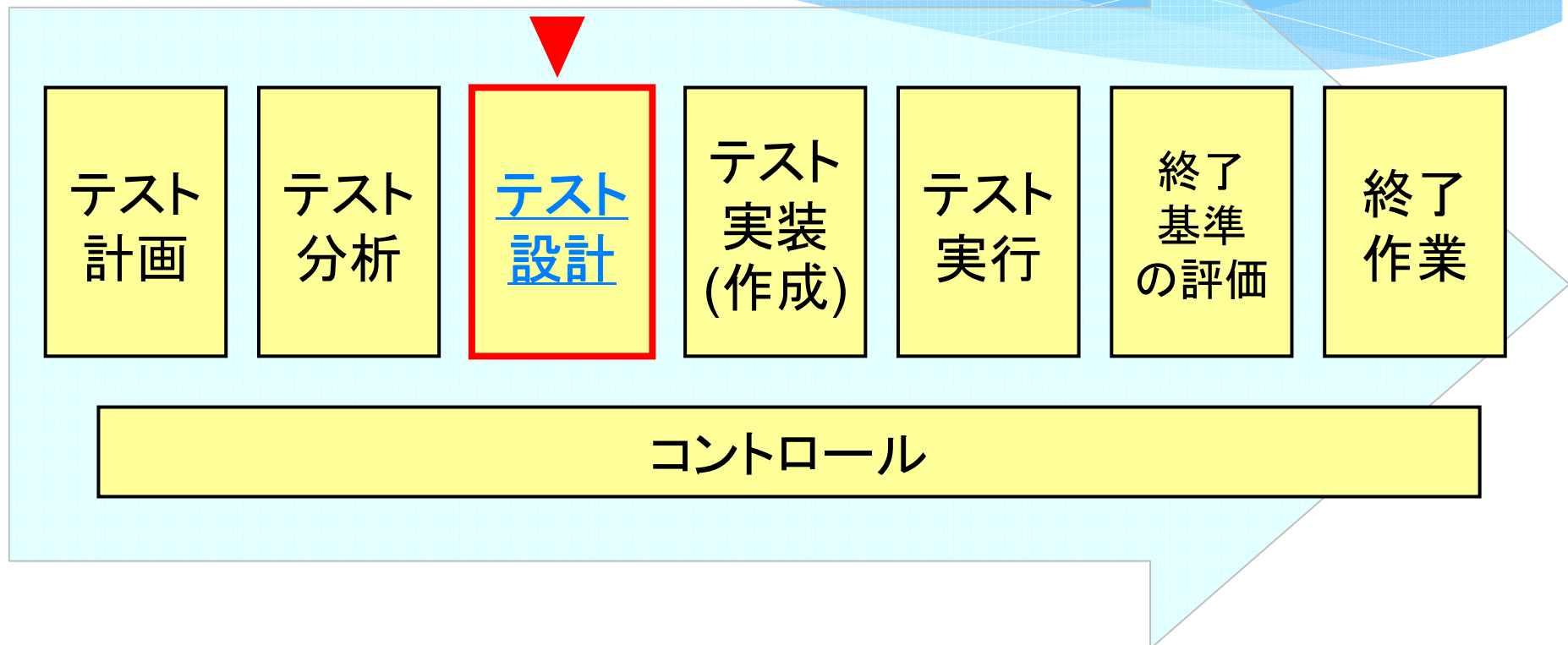
テスト技法とは

なぜ、テスト技法を使用した
テスト設計が必要なんでしょう？



全てをテストすることは不可能なので、
沢山のバグを見つけるためには「漏れ」
や「かぶり」のない、効率の良いテスト
を行う必要があるため

テストのライフサイクル



同值分割、境界値分析

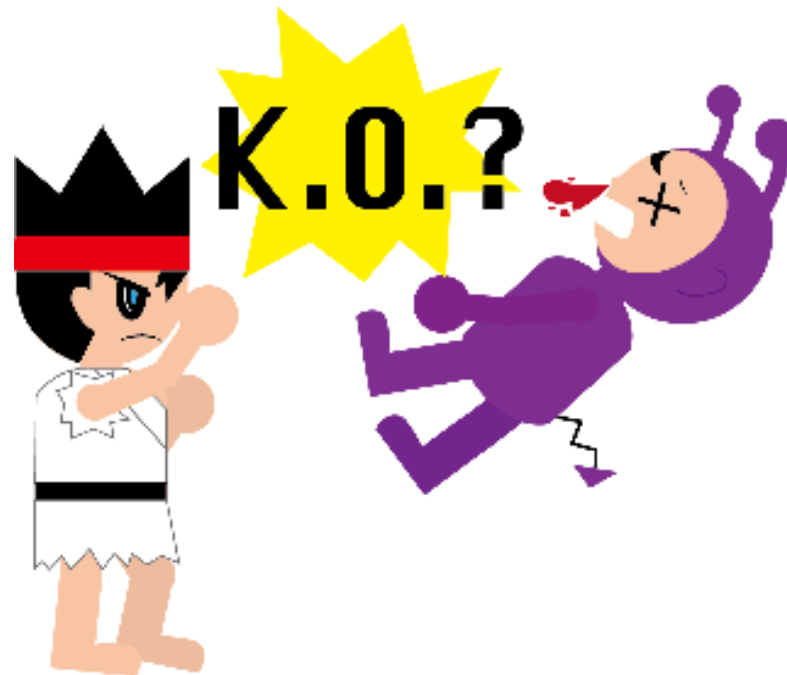
ある格闘ゲームの場合



ある格闘ゲームの場合

例題

この格闘ゲームの**KO判定**をテストしたい！



KO判定をテストするために

* 仕様を確認してみよう！

【仕様】

- ①. プレイヤーが対戦相手にダメージが発生した場合に
KO判定を行う。
- ②. 体力からダメージをマイナスした結果、残り体力が0以下に
なった場合には0として扱う。
- ③. ダメージ反映後に体力が0の場合はKOとし試合を終了する。

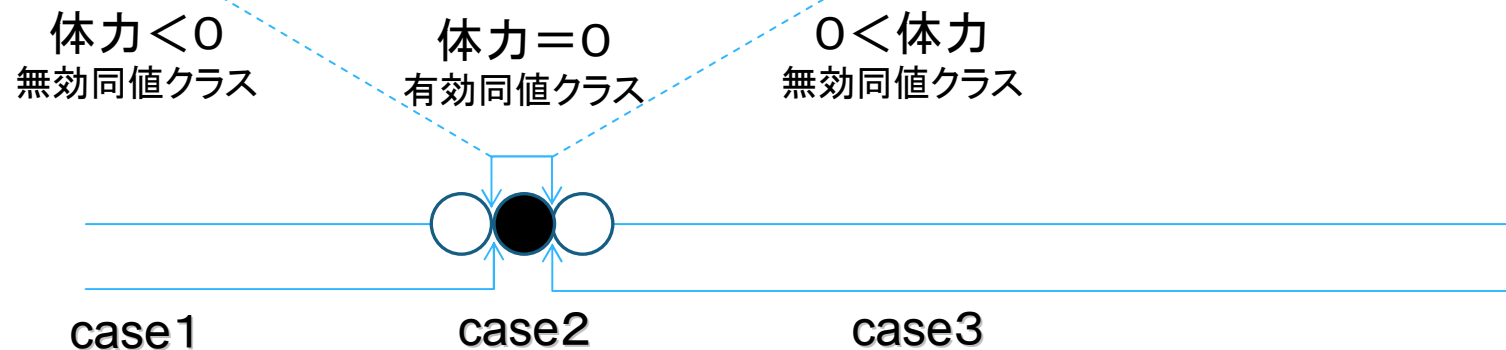
同値分割してみよう

同値分割によるテスト設計：

プログラムのパラメータとして結果が同じになる
クラス(グループ)を作成し、その中で正常な値の範囲を
有効同値、それ以外を無効同値と定義する。

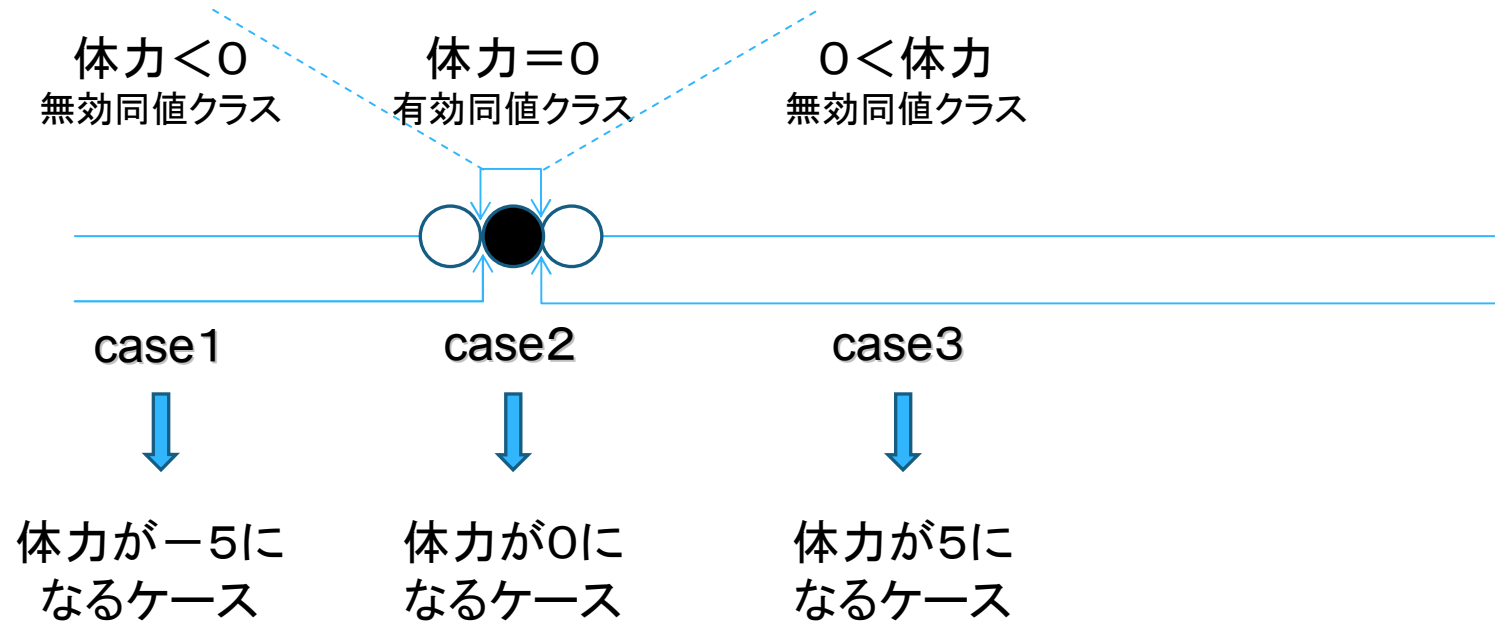
有効な値⇒有効同値(クラス)

無効な値⇒無効同値(クラス)



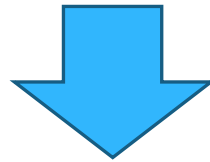
同値分割によるテスト

同値分割により発見した同値クラスに
該当するパラメータを算出する。



境界値分析を行う

境界値分析によるテスト設計：
同値分割により発見したクラスの境界となる値を探す。



境界値にはバグが潜んでいることが多い！

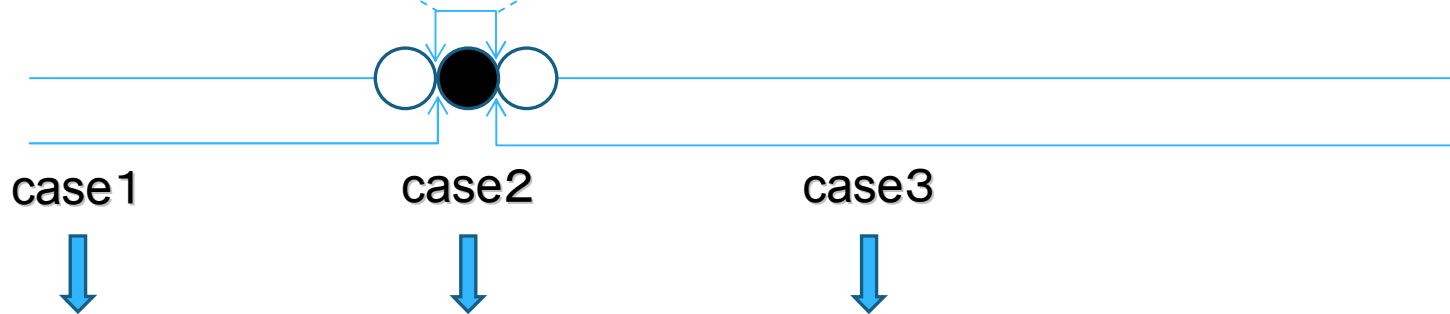
境界値分析を行う

同値分割により発見した同値クラス毎の
境界値となる値を探す。

体力: -1
無効同値クラス

体力: 0
有効同値クラス

体力: 1
無効同値クラス



体力が-1に
なるケース

体力が0に
なるケース

体力が1に
なるケース

境界値分析を行う

ポイント:

境界値で重要なのは、単位！

例えば・・・

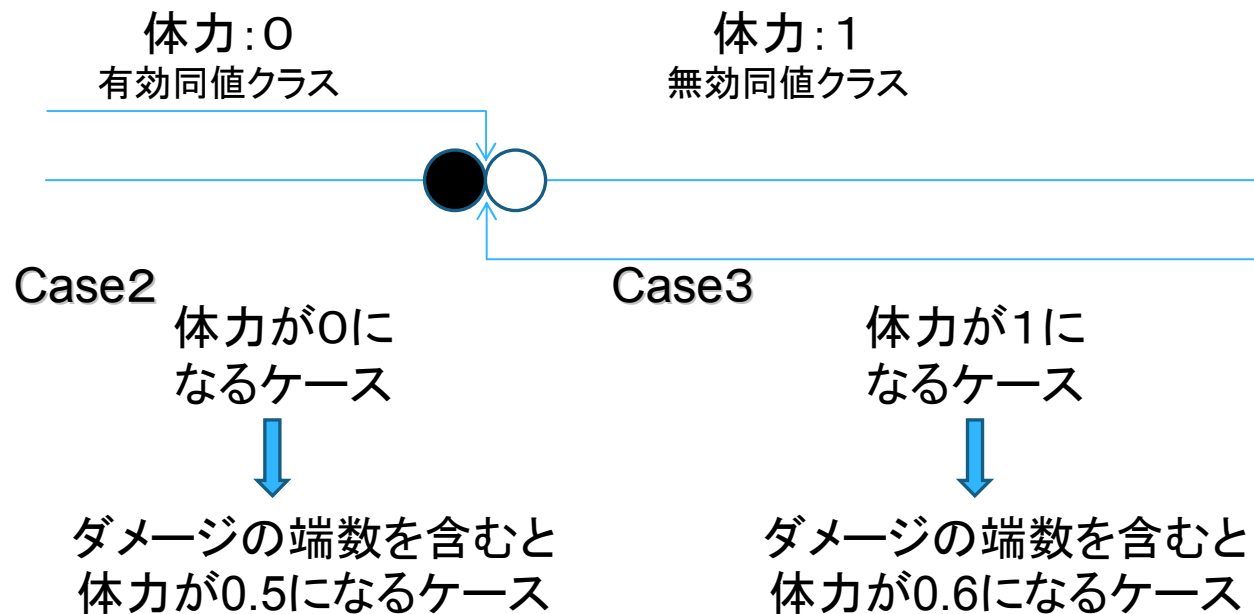
③. ダメージ反映後に体力が0の場合はKOとし試合を終了する。

③の仕様に関連する情報を調査したところ次の事がわかった。

- 体力の単位は整数(最少単位が1)で処理される。
- ダメージ計算の小数点以下の端数は四捨五入を行う。

境界値分析を行う

新たに発見した情報を元に境界値分析を行うと
次のようになる。



同値分割と境界値分析のまとめ

* 同値分割

プログラムが同じ処理単位とする値(クラス)を定義し、それぞれのクラス毎の代表する値を与えた時に期待する結果になることを確認する。

* 境界値分析

同値クラスの境界値を探し、各クラスの境界値を与えた時に期待する結果になることを確認する。

デインジョンテーブル

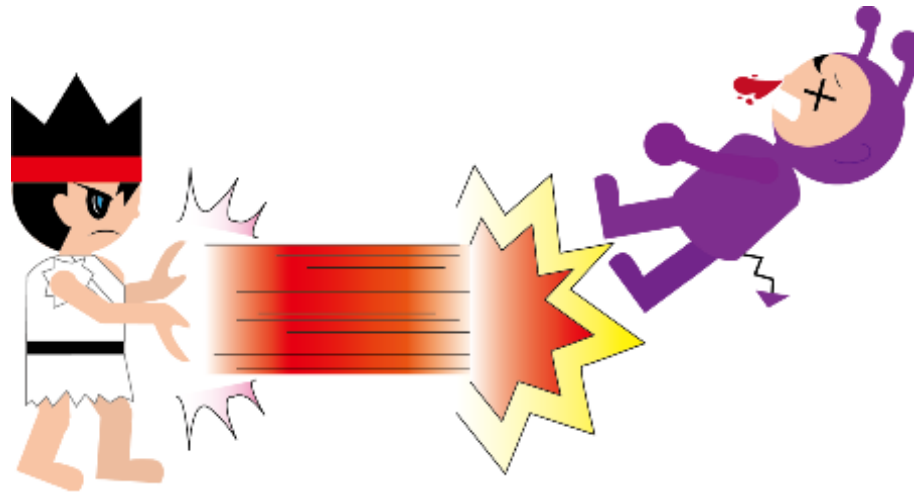
再び、ある格闘ゲームの場合

このゲームは次のよう攻撃パターンがある。

通常攻撃



必殺技



再び、ある格闘ゲームの場合

そして、受け手のパターンはこんな感じ。

通常状態



ガード状態

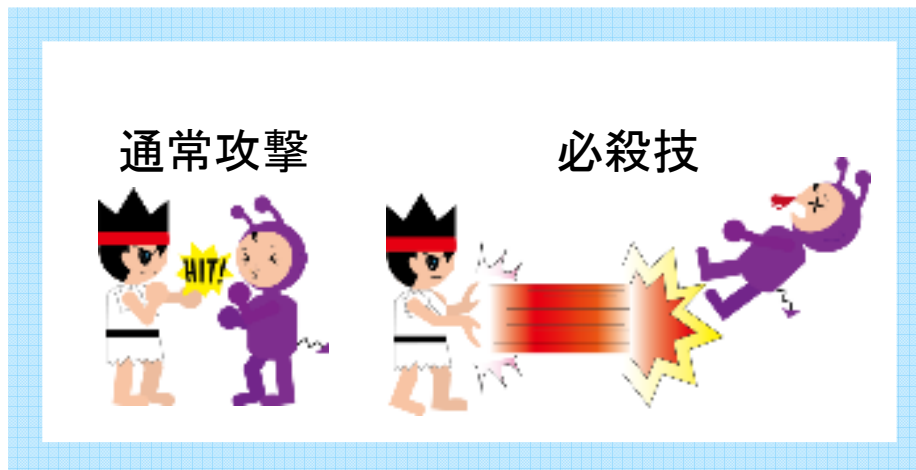


再び、ある格闘ゲームの場合

例題：

攻撃パターンと受け手のパターン組み合わせ条件から
ダメージ発生のテストを行いたい！

攻撃パターン



受け手のパターン



ダメージ発生を テストするために

* 仕様を分析してみよう！

【仕様】

- ①. 通常状態にダメージが発生した場合は、ダメージ量を1倍で計算する。
- ②. ガード状態は必殺技以外のダメージを0倍にする。
- ③. ガード状態は必殺技のダメージを0.5倍にする。
- ④. 体力残り10%以下のガード状態は全てのダメージを0倍にする。

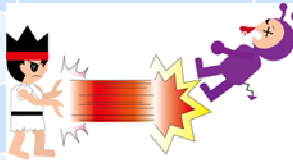
仕様が確認できたところで

* デイシジョンテーブルで書いてみよう！

ディンジョンテーブルを作成してみよう

①. 条件(前提、入力など)を書き出します

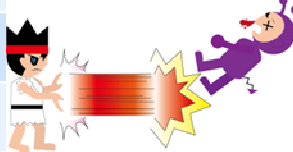
条件

通常状態			
ガード状態			
体力10%以下			
通常攻撃			
必殺技			

ディンジョンテーブルを作成してみよう

②.次に期待する(動作の)結果を書く

結果／動作

通常状態			
ガード状態			
体力10%以下			
通常攻撃			
必殺技			
ダメージ×0			
ダメージ×0.5			
ダメージ×1			

ディンジョンテーブルを作成してみよう

③.次に条件のY(Yes)/N(No)を書き出す。

[illegible]

1列が 1テストケース

ディンジョンテーブルを作成してみよう

④. ありえない動作に“-”と該当する結果に“X”を書き出す。

[illegible]

※この例題ではテストの対象とはしません。

ディンジョンテーブルを作成してみよう

⑤. 結果が確認できない(-のみの)ケースを削除する。

[illegible]

※便宜上、削除しています。

ディンジョンテーブルを作成してみよう

⑤. 結果が確認できない(-のみの)ケースを削除する。

|--|

ディンジョンテーブルを作成してみよう

⑤. 結果が確認できない(-のみの)ケースを削除する。

通常状態	Y	N	Y	N	Y	N	Y	N
ガード状態	N	Y	N	Y	N	Y	N	Y
体力10%以下	Y	Y	Y	Y	N	N	N	N
通常攻撃	Y	Y	N	N	Y	Y	N	N
必殺技	N	N	Y	Y	N	N	Y	Y
ダメージ×0		X		X		X		
ダメージ×0.5								X
ダメージ×1	X		X		X		X	

ディンジョンテーブルを作成してみよう

ガード状態で体力10%の組み合わせの条件はそれ以下の条件に何を当てはめても同じ結果になるので2, 4のケースは攻撃パターンを“-”にする。

⑥.テストケースをまとめる。

	1	2	3	4	5	6	7	8
通常状態	Y	N	Y	N	Y	N	Y	N
ガード状態	N	Y	N	Y	N	Y	N	Y
体力10%以下	Y	Y	Y	Y	N	N	N	N
通常攻撃	Y	Y	N	N	Y	Y	N	N
必殺技	N	N	Y	Y	N	N	Y	Y
ダメージ×0		X		X		X		
ダメージ×0.5								X
ダメージ×1	X		X		X		X	

ディンジョンテーブルを作成してみよう

ガード状態で体力10%の組み合わせの条件はそれ以下の条件に何を当てはめても同じ結果になるので2, 4のケースは攻撃パターンを“-”にする。

⑥.テストケースをまとめる。

	1	2	3		5	6	7	8
通常状態	Y	N	Y		Y	N	Y	N
ガード状態	N	Y	N		N	Y	N	Y
体力10%以下	Y	Y	Y		N	N	N	N
通常攻撃	Y	-	N		Y	Y	N	N
必殺技	N	-	Y		N	N	Y	Y
ダメージ×0		X				X		
ダメージ×0.5								X
ダメージ×1	X		X		X		X	

ディンジョンテーブルを作成してみよう

この列が最終的なテストケース

⑦.ディンジョンテーブル完成。

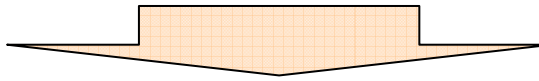
	1	2	3	5	6	7	8
通常状態	Y	N	Y	Y	N	Y	N
ガード状態	N	Y	N	N	Y	N	Y
体力10%以下	Y	Y	Y	N	N	N	N
通常攻撃	Y	Y	N	Y	Y	N	N
必殺技	N	N	Y	N	N	Y	Y
ダメージ×0		X			X		
ダメージ×0.5							X
ダメージ×1	X		X	X		X	

デシジョンテーブルのまとめ

デシジョンテーブルとは

入力データや入力条件の組合せに対する処理や出力結果をテーブルにまとめたもの。

自然言語で書かれた仕様を論理的に整理する場合に利用できる。



- 自然言語で書かれた仕様が論理的になる。
- 複雑な論理を把握することができる。
- 抜け・漏れを防ぐようにテスト設計できる。

条件の一覧	条件の組合せ			
結果の一覧	結果の組合せ			



組合せテストの技法

～直交表を用いたテスト技法～

例題

* 例題

ロボット対戦のゲームにおいて、選択したロボットや装備で正しくゲームが開始できるかをテストしたい。

例題

* 仕様

- コンピュータとプレイヤーの対戦型である。
- ロボットを操縦するパイロットキャラクターとして「男性」と「女性」のどちらかを選択できる。
- ロボットは4種類のロボットから選択できる。

ロボット：デバッグスター、ゴットハンドII、
ユーザビリティI、スタブエース

- ロボットには、4種類ある武器から1つと3種類の防具から1つを選択できる。

武器：バズーカ、マシンガン、ライフル、サーベル

防具：金のシールド、鉄のシールド、木のシールド

全組合せ

* 組合せを考えてみる。

パイロット(2) × ロボット(4) × 武器(4) × 防具(3)
= 96通り

パイロット ロボット 防具			パイロット								
			男性				女性				
			ロボット								
			ロボA	ロボB	ロボC	ロボD	ロボA	ロボB	ロボC	ロボD	
武器	武器A	防具	防具A	○	○	○	○	○	○	○	○
			防具B	○	○	○	○	○	○	○	○
			防具C	○	○	○	○	○	○	○	○
	武器B		防具A	○	○	○	○	○	○	○	○
			防具B	○	○	○	○	○	○	○	○
			防具C	○	○	○	○	○	○	○	○
	武器C		防具A	○	○	○	○	○	○	○	○
			防具B	○	○	○	○	○	○	○	○
			防具C	○	○	○	○	○	○	○	○
	武器D		防具A	○	○	○	○	○	○	○	○
			防具B	○	○	○	○	○	○	○	○
			防具C	○	○	○	○	○	○	○	○

全部やりますか？

全組合せ

* 減らしてみる。

パイロット ロボット 防具 防具				パイロット							
				男性				女性			
				ロボット							
				ロボA	ロボB	ロボC	ロボD	ロボA	ロボB	ロボC	ロボD
武器	武器A	防具	防具A	○	○	○	○	○	○	○	○
			防具B	○	○	○	○	○	○	○	○
			防具C	○	○	○	○	○	○	○	○
	武器B		防具A	○	○	○	○	○	○	○	○
			防具B	○	○	○	○	○	○	○	○
			防具C	○	○	○	○	○	○	○	○
	武器C		防具A	○	○	○	○	○	○	○	○
			防具B	○	○	○	○	○	○	○	○
			防具C	○	○	○	○	○	○	○	○
	武器D		防具A	○	○	○	○	○	○	○	○
			防具B	○	○	○	○	○	○	○	○
			防具C	○	○	○	○	○	○	○	○

 : 実施するテストケース

なんとなく網羅(カバレッジ)できた！！ 気がする。
こんなことになっていませんか？

直交表を使うと

* 直交表を使った場合のテストケース数は16通り。

Case1:	「男性」	「デバッグスター」	「バズーカ」	「金のシールド」
Case2:	「男性」	「ゴットハンドII」	「マシンガン」	「鉄のシールド」
Case3:	「男性」	「ユーザビリティI」	「ライフル」	「木のシールド」
Case4:	「男性」	「スタブエース」	「サーベル」	「金のシールド」
Case5:	「男性」	「デバッグスター」	「マシンガン」	「金のシールド」
Case6:	「男性」	「ゴットハンドII」	「バズーカ」	「木のシールド」
Case7:	「男性」	「ユーザビリティI」	「サーベル」	「鉄のシールド」
Case8:	「男性」	「スタブエース」	「ライフル」	「金のシールド」
Case9:	「女性」	「デバッグスター」	「ライフル」	「鉄のシールド」
Case10:	「女性」	「ゴットハンドII」	「サーベル」	「金のシールド」
Case11:	「女性」	「ユーザビリティI」	「バズーカ」	「金のシールド」
Case12:	「女性」	「スタブエース」	「マシンガン」	「木のシールド」
Case13:	「女性」	「デバッグスター」	「サーベル」	「木のシールド」
Case14:	「女性」	「ゴットハンドII」	「ライフル」	「金のシールド」
Case15:	「女性」	「ユーザビリティI」	「マシンガン」	「金のシールド」
Case16:	「女性」	「スタブエース」	「バズーカ」	「鉄のシールド」

因子と水準

* なぜこうなるのか？

→2因子の組合せに注目しているから。

用語の整理

因子： テスト対象機能(パラメータ)

※今回のケースでは「パイロット」や「ロボット」

水準： パラメータが取り得る値

※今回のケースでは「デバッグスター」や「ユーザビリティI」

直交表： 各因子の水準が同じ回数だけ出現する特徴を持った表

直交表

L4直交表

	因子		
	因子1	因子2	因子3
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	0

因子

L8直交表

	因子						
	因子1	因子2	因子3	因子4	因子5	因子6	因子7
1	0	0	0	0	0	0	0
2	0	0	0	1	1	1	1
3	0	1	1	0	0	1	1
4	0	1	1	1	1	0	0
5	1	0	1	0	1	0	1
6	1	0	1	1	0	1	0
7	1	1	0	0	1	1	0
8	1	1	0	1	0	0	1

- ・各因子の水準が同じ回数だけ出現
- ・2因子間のパラメータ組み合わせ100%

→L4直交表で因子1と因子2の水準の組み合わせパターンは「0,0」、「0,1」、「1,0」、「1,1」が出現していることが確認できる。因子1と因子3、因子2と因子、L8でも同様。

2因子の注目

* なぜ2因子なのか？

■ FTFI (failure-triggering fault interaction)

□ フォールトに関係しているパラメータ数

FTFI	エキスパートシステム				OS	組込み	Netscape N.	Apache	DB
1	61	72	48	39	82	66	29	42	68
2	36	10	6	8	*	31	47	28	25
3	*na	*	*	*	*	2	19	19	5
4	*	*	*	*	*	1	2	7	2
5	*	*	*	*	*		2	0	
6	*	*	*	*	*		1	4	

※出展: D. Richard Kuhn *Software Fault Interactions and Implications for Software Testing*
, *IEEE Transactions on software engineering*, Vol.30, 2004

直交表の作り方 ～1～

* 直交表の作り方

- L2直交表を作成する
- L4直交表へ拡張する
- L2からL4への拡張と同じようにして、目的のサイズになるまで直交表の拡張を繰り返す。

→目的のサイズとは：

直交表の大きさ = $\Sigma(\text{各因子の水準数}-1) + 1$

例題では、 $(2-1)+(4-1)+(4-1)+(3-1)+1=10 \div L16$

※ただし、多水準結合可能な制限にかかる場合もあり。

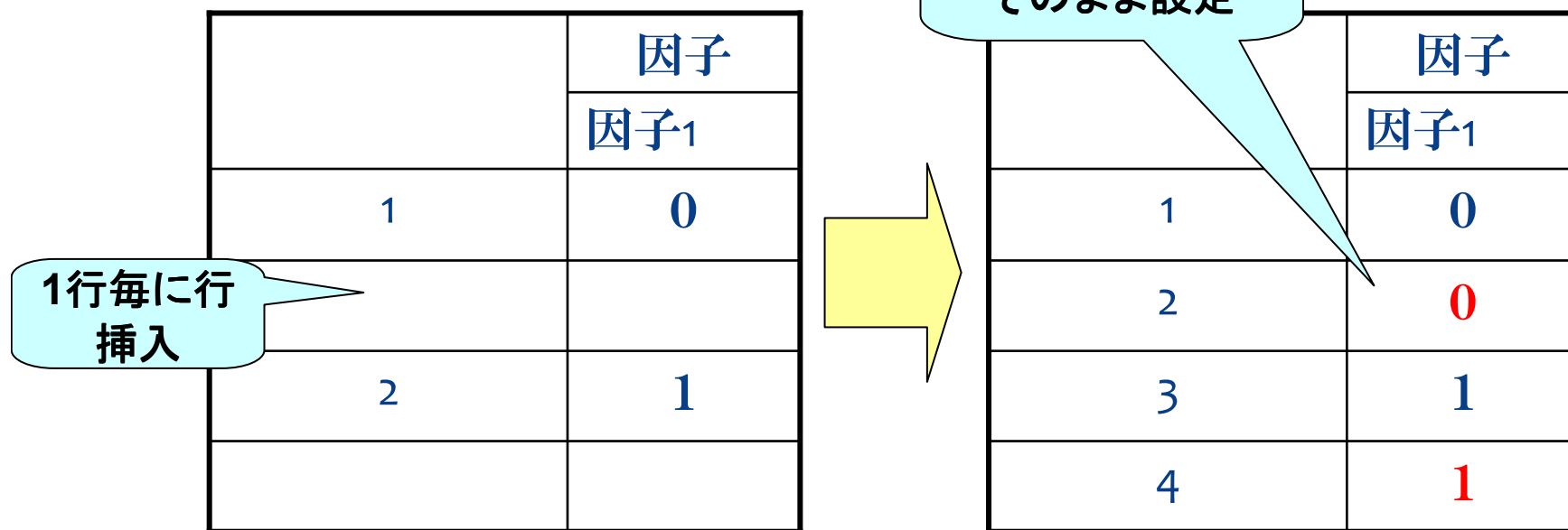
直交表の作り方 ～2～

- * L2直交表を作り、L4直交表に拡張する。
- * まずはL2直交表を作る

	因子
	因子1
1	0
2	1

直交表の作り方 ～3～

* L4直交表に拡張する。



①行増やし(1行毎に挿入)て、一つ上の水準をそのまま設定

②列(因子)を増やして、水準を交互(0/1)に設定

③既存の因子と、②で追加した因子の水準のXORを取って新しい列(因子)に設定

直交表の作り方 ～4～

* L4直交表に拡張する。

	因子	
	因子1	因子2
1	0	0
2	0	1
3	1	0
4	1	1

列(因子)を増やす

水準を交互に設定

①行増やし(1行毎に挿入)て、一つ上の水準をそのまま設定

②列(因子)を増やして、水準を交互(0/1)に設定

③既存の因子と、②で追加した因子の水準のXORを取って新しい列(因子)に設定

直交表の作り方 ～5～

* L4直交表に拡張する。

	因子		
	因子1	因子2	因子3
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	0

XORを取って
新しい因子に設
定

- ①行増やし(1行毎に挿入)て、一つ上の水準をそのまま設定
- ②列(因子)を増やして、水準を交互(0/1)に設定
- ③既存の因子と、②で追加した因子の水準のXORを取って新しい列(因子)に設定

直交表の作り方 ～6～

* L4直交表に拡張する。

	因子		
	因子1	因子2	因子3
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	0

L4直交表の完成

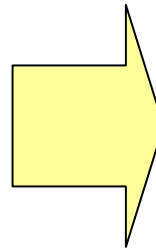
完成

直交表の作り方 ～7～

* L4→L8直交表に拡張する。

1行毎に行
挿入

	因子		
	因子1	因子2	因子3
1	0	0	0
3	0	1	1
5	1	0	1
7	1	1	0



一つ上の水準を
そのまま設定

	因子		
	因子1	因子2	因子3
1	0	0	0
2	0	0	0
3	0	1	1
4	0	1	1
5	1	0	1
6	1	0	1
7	1	1	0
8	1	1	0

直交表の作り方 ～8～

* L4→L8直交表に拡張する。

	因子			
	因子1	因子2	因子3	因子4
1	0	0	0	0
2	0	0	0	1
3	0	1	1	0
4	0	1	1	1
5	1	0	1	0
6	1	0	1	1
7	1	1	0	0
8	1	1	0	1

列(因子)を増やす

水準を交互に設定

直交表の作り方 ～9～

* L4→L8直交表に拡張する。

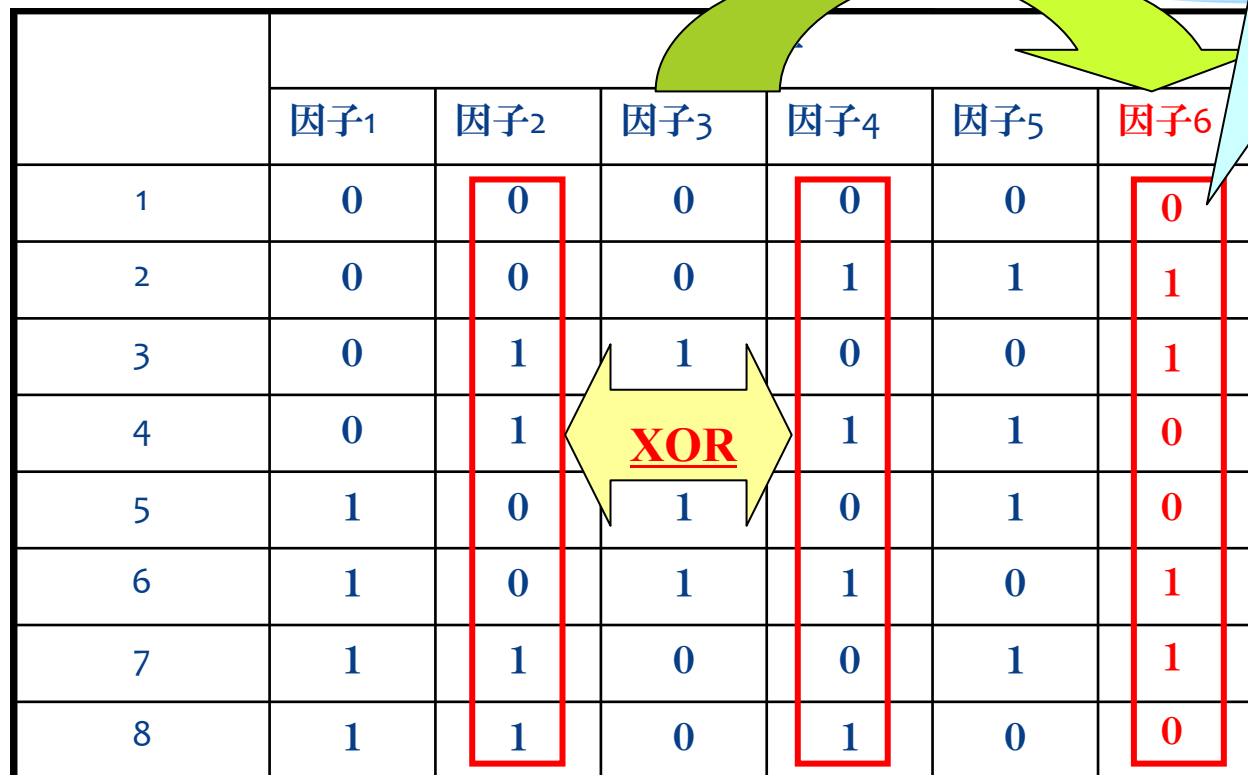
	因子				
	因子1	因子2	因子3	因子4	因子5
1	0	0	0	0	0
2	0	0	0	1	1
3	0	1	1	0	0
4	0			1	1
5	1	0	1	0	1
6	1	0	1	1	0
7	1	1	0	0	1
8	1	1	0	1	0

因子1と因子4の
XORを取って新しい因子に設定

直交表の作り方 ～10～

* L4→L8直交表に拡張する。

因子2と因子4の
XORを取って新し
い因子に設定

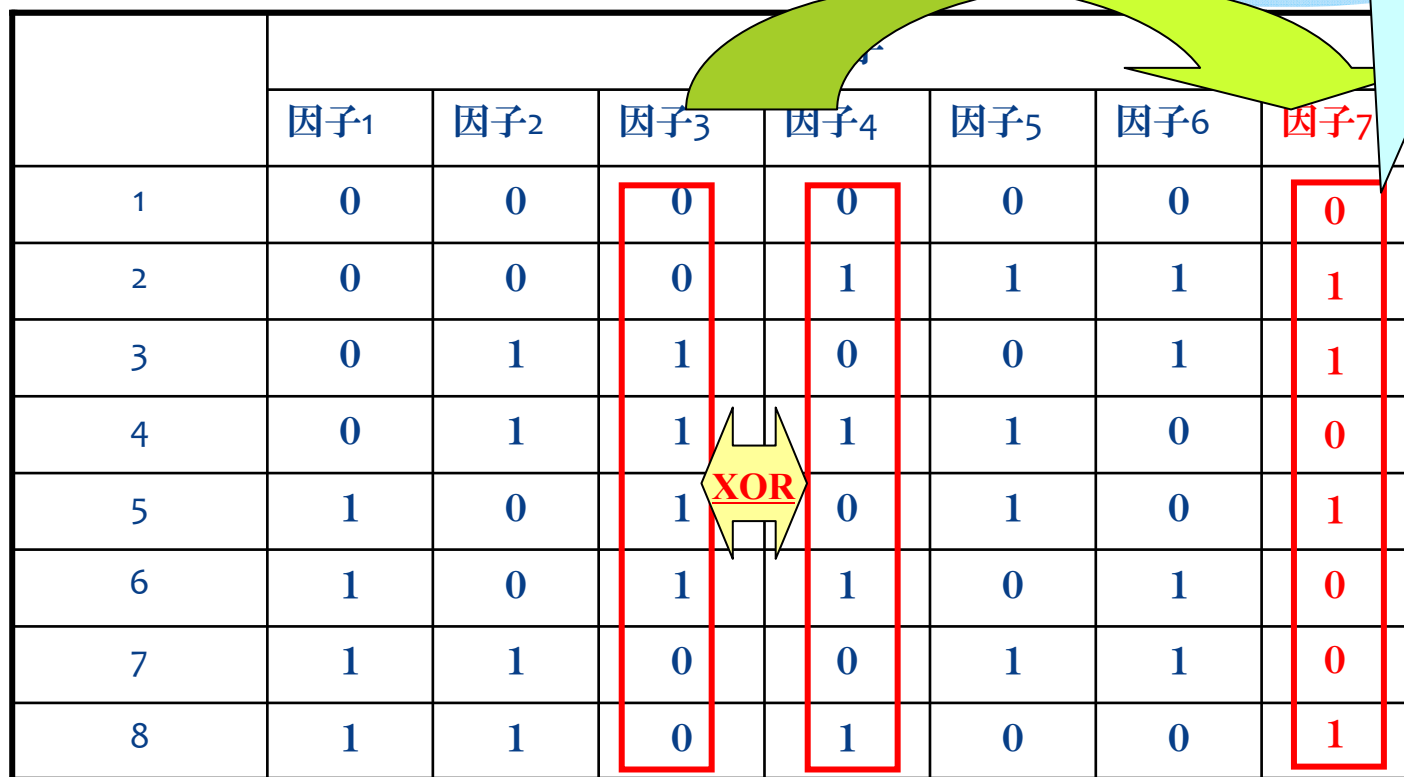


	因子1	因子2	因子3	因子4	因子5	因子6
1	0	0	0	0	0	0
2	0	0	0	1	1	1
3	0	1	1	0	0	1
4	0	1		1	1	0
5	1	0	1	0	1	0
6	1	0	1	1	0	1
7	1	1	0	0	1	1
8	1	1	0	1	0	0

直交表の作り方 ～11～

* L4→L8直交表に拡張する。

因子3と因子4の
XORを取って新し
い因子に設定



	因子1	因子2	因子3	因子4	因子5	因子6	因子7
1	0	0	0	0	0	0	0
2	0	0	0	1	1	1	1
3	0	1	1	0	0	1	1
4	0	1	1	1	1	0	0
5	1	0	1	0	1	0	1
6	1	0	1	1	0	1	0
7	1	1	0	0	1	1	0
8	1	1	0	1	0	0	1

直交表の作り方 ～12～

* L4→L8直交表に拡張する。

完成

	因子						
	因子1	因子2	因子3	因子4	因子5	因子6	因子7
1	0	0	0	0	0	0	0
2	0	0	0	1	1	1	1
3	0	1	1	0	0	1	1
4	0	1	1	1	1	0	0
5	1	0	1	0	1	0	1
6	1	0	1	1	0	1	0
7	1	1	0	0	1	1	0
8	1	1	0	1	0	0	1

同じ方法で、どんどん拡大可能！！ →L16, L32, L64, L128....

直交表への割付け ～1～

- * 因子と水準を整理する。

パイロット
男性
女性

ロボット
デバッグスター
ゴットハンドII
ユーザビリティI
スタブエース

武器
バズーカ
マシンガン
ライフル
サーベル

防具
金のシールド
鉄のシールド
木のシールド

- * 直交表の大きさを決める。

直交表の大きさ = $\Sigma(\text{各因子の水準数}-1) + 1$

例題では、 $(2-1)+(4-1)+(4-1)+(3-1)+1=10 \div L16$

※ただし、多水準結合可能な制限にかかる場合もあり。

直交表への割付け ～2～

- * 直交表を用意する。今回のケースではL16直交表。

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
0	1	1	0	0	1	1	1	1	0	0	1	1	0	0
0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
0	1	1	1	1	0	0	1	1	0	0	0	0	1	1
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
1	0	1	0	1	0	1	1	0	1	0	1	0	1	0
1	0	1	1	0	1	0	0	1	0	1	1	0	1	0
1	0	1	1	0	1	0	1	0	1	0	0	1	0	1
1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
1	1	0	0	1	1	0	1	0	0	1	1	0	0	1
1	1	0	1	0	0	1	0	1	1	0	1	0	0	1
1	1	0	1	0	0	1	1	0	0	1	0	1	1	0

直交表への割付け ～3～

* 「パイロット」を割り付ける。

→水準は「男性」、「女性」の2種類なので「0」、「1」に割り付ける

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
男性	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
0	1	1	0	0	1	1	1	1	0	0	1	1	0	0
0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
0	1	1	1	1	0	0	1	1	0	0	0	0	1	1
女性	0	1	0	1	0	1	0	1	0	1	0	1	0	1
1	0	1	0	1	0	1	1	0	1	0	1	0	1	0
1	0	1	1	0	1	0	0	1	0	1	1	0	1	0
1	0	1	1	0	1	0	1	0	1	0	0	1	0	1
1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
1	1	0	0	1	1	0	1	0	0	1	1	0	0	1
1	1	0	1	0	0	1	0	1	1	0	1	0	0	1
1	1	0	1	0	0	1	1	0	0	1	0	1	1	0

直交表への割付け ～4～

* 「パイロット」を割り付ける。

→水準は「男性」、「女性」の2種類なので「0」、「1」に割り付ける

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
男性	0	0	0	0	0	0	0	0	0	0	0	0	0	0
男性	0	0	0	0	0	0	1	1	1	1	1	1	1	1
男性	0	0	0	0	0	0	0	0	0	0	1	1	1	1
男性	0	0	0	0	0	0	1	1	1	1	0	0	0	0
男性	1	1	0	0	1	1	0	0	1	1	0	0	1	1
男性	1	1	0	0	1	1	1	1	0	0	1	1	0	0
男性	1	1	1	1	0	0	0	0	1	1	1	1	0	0
男性	1	1	1	1	0	0	1	1	0	0	0	0	1	1
女性	0	1	0	1	0	1	0	1	0	1	0	1	0	1
女性	0	1	0	1	0	1	1	0	1	0	1	0	1	0
女性	0	1	1	0	1	0	0	1	0	1	1	0	1	0
女性	0	1	1	0	1	0	1	0	1	0	0	1	0	1
女性	1	0	0	1	1	0	0	1	1	0	0	1	1	0
女性	1	0	0	1	1	0	1	0	0	1	1	1	0	0
女性	1	0	1	0	0	1	0	1	1	0	1	0	1	0
女性	1	0	1	0	0	1	1	0	0	1	0	1	1	0

0→男性
1→女性

割付完了

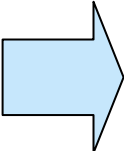
直交表への割付け ～5～

* 「ロボット」を割り付ける。

→水準は「デバッグスター」、「ゴットハンドⅡ」、「ユーザビリティⅠ」、
「スタブエース」の4種類なので「0」、「1」では割り付けられない！！

* 他水準直交表に変形させる。

→3以上の水準が割り付けられるようにする。

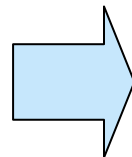


1	2	3
0	0	0
0	0	0
0	0	0
0	0	0
0	1	1
0	1	1
0	1	1
0	1	1
1	0	1

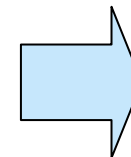
1	2	3
000		
000		
000		
000		
011		
011		
011		
011		
101		

直交表への割付け ～6～

1	2	3
0	0	0
0	0	0
0	0	0
0	0	0
0	1	1
0	1	1
0	1	1
0	1	1
1	0	1
1	0	1
1	0	1
1	0	1
1	1	0
1	1	0
1	1	0
1	1	0



1	2	3
	000	
	000	
	000	
	000	
	011	
	011	
	011	
	011	
	101	
	101	
	101	
	101	
	110	
	110	
	110	
	110	



000→デバッグスター
 011→ゴットハンドⅡ
 101→ユーザビリティⅠ
 110→スタブエース

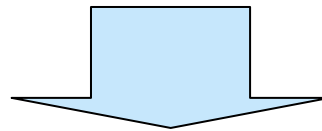
1	2	3
デバッグスター		
デバッグスター		
デバッグスター		
デバッグスター		
ゴットハンドⅡ		
ゴットハンドⅡ		
ゴットハンドⅡ		
ゴットハンドⅡ		
ユーザビリティⅠ		
ユーザビリティⅠ		
ユーザビリティⅠ		
ユーザビリティⅠ		
スタブエース		
スタブエース		
スタブエース		
スタブエース		

直交表への割付け ～7～

前の例ではさらっと結合したが、どの列でも適当にくっつけばよいというものではない。

直交表の特性上、結合できる列の組み合わせは決まっている。

→ex) 4水準列の場合、任意の2列とその交互作用に当たる列を結合。

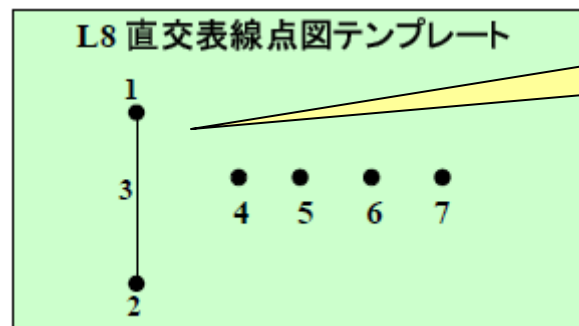


線点図を利用する。

直交表への割付け ～8～

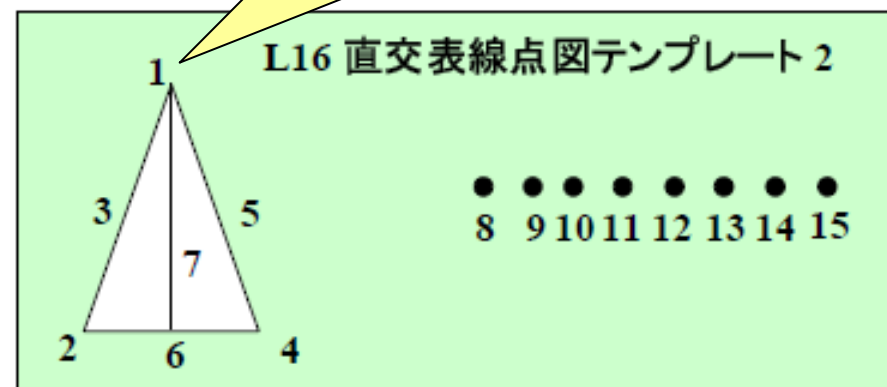
* 線点図

L8直交表用



L8の場合には、1,2,3列をくっつけて4水準にしてよいと見る

L16直交表用



L16の場合には、1～7列をくっつけて8水準にしてよいと見る

直交表への割付け ～9～

* ロボットの割付

→線点図より「4」、「8」、「12」列を利用して割り付ける。

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
男性	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
男性	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
男性	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0
男性	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
男性	1	1	0	0	1	1	0	0	1	1	1	1	1	1	1
男性	1	1	0	0	1	1	1	1	1	0	0	1	1	0	0
男性	1	1	1	1	0	0	0	0	0	1	1	1	1	0	0
男性	1	1	1	1	0	0	1	1	1	0	0	0	0	1	1
女性	0	1	0	1	0	1	0	0	1	0	1	0	1	0	1
女性	0	1	0	1	0	1	1	1	0	1	0	1	0	1	0
女性	0	1	1	0	1	0	0	0	1	0	1	1	0	1	0
女性	0	1	1	0	1	0	1	0	1	1	0	0	1	0	1
女性	1	0	0	1	1	0	0	0	1	1	0	0	1	1	0
女性	1	0	0	1	1	0	1	0	0	0	1	1	0	0	1
女性	1	0	1	0	0	1	0	1	1	1	0	1	0	0	1
女性	1	0	1	0	0	1	1	1	0	0	1	0	1	1	0

4,8,12列を抽出

直交表への割付け ～10～

* ロボットの割付

→線点図より「4」、「8」、「12」列を利用して割り付ける。

	1	4	8	12	2	3	5	6	7	9	10	11	13	14	15
男性	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
男性	0	1	1	0	0	0	0	0	0	1	1	1	1	1	1
男性	1	0	1	0	0	0	0	0	0	0	0	1	1	1	1
男性	1	1	0	0	0	0	0	0	0	1	1	0	0	0	0
男性	0	0	0	1	1	1	0	1	1	1	0	0	1	0	0
男性	0	1	1	1	1	1	1	0	0	0	1	1	1	0	0
男性	1	0	1	1	1	1	1	0	0	1	0	0	0	1	1
男性	1	1	0	1	1	1	1	0	0	1	0	0	0	1	1
女性	0	0	0	0	0	1	1	0	1	1	0	1	1	0	1
女性	0	1	1	0	0	1	1	0	1	0	1	0	0	1	0
女性	1	0	1	0	0	1	0	1	0	1	0	1	0	1	0
女性	1	1	0	0	0	1	0	1	0	0	1	0	1	0	1
女性	0	0	0	1	0	1	1	1	0	1	1	0	1	1	0
女性	0	1	1	1	0	1	1	1	0	0	0	1	0	0	1
女性	1	0	1	1	0	0	0	0	1	1	1	0	0	0	1
女性	1	1	0	1	0	0	0	0	1	0	0	1	1	1	0

4,8,12列を寄せる

直交表への割付け ～11～

* ロボットの割付

→線点図より「4」、「8」、「12」列を利用して割り付ける。

	1	4	8	12	2	3	5	6	7	9	10	11	13	14	15
男性		000			0	0	0	0	0	0	0	0	0	0	0
男性		011			0	0	0	0	0	0	1	1	1	1	1
男性		101			0	0	0	0	0	0	0	0	1	1	1
男性		110			0	0	0	0	0	0	1	1	0	0	0
男性		000			1	1	1	1	1	1	1	1	0	1	1
男性		011			1	1	0	1	1	1	0	0	1	0	0
男性		101			1	1	1	0	0	0	1	1	1	0	0
男性		110			1	1	1	0	0	1	0	0	0	1	1
女性		000			0	1	1	0	1	1	0	1	1	0	1
女性		011			0	1	1	0	1	0	1	0	0	1	0
女性		101			0	1	0	1	0	1	0	1	0	1	0
女性		110			0	1	0	1	0	0	1	0	1	0	1
女性		000			1	0	1	1	0	1	1	0	1	1	0
女性		011			1	0	1	1	0	0	0	1	0	0	1
女性		101			1	0	0	0	1	1	1	0	0	0	1
女性		110			1	0	0	0	1	0	0	1	1	1	0

4,8,12列を結合
する

直交表への割付け ～12～

* ロボットの割付

→線点図より「4」、「8」、「12」列を利用して割り付ける。

	1	4	8	12	2	3	5	6	7	9	10	11	13	14	15
男性	デバッグスター	0	0	0	0	0	0	0	0	0	0	0	0	0	0
男性	ゴットハンドⅡ	0	0	0	0	0	0	0	0	0	0	0	0	0	0
男性	ユーザビリティⅠ	0	0	0	0	0	0	0	0	0	0	0	0	0	0
男性	スタブエース	0	0	0	0	0	0	0	0	0	0	0	0	0	0
男性	デバッグスター	1	1	1	1	1	1	1	1	1	1	1	1	1	1
男性	ゴットハンドⅡ	1	1	1	1	1	1	1	1	1	1	1	1	1	1
男性	ユーザビリティⅠ	1	1	1	1	1	1	1	1	1	1	1	1	1	1
男性	スタブエース	1	1	1	1	1	1	1	1	1	1	1	1	1	1
女性	デバッグスター	0	1	1	0	1	1	0	1	1	0	0	0	1	1
女性	ゴットハンドⅡ	0	1	1	0	1	1	0	1	0	1	0	0	1	0
女性	ユーザビリティⅠ	0	1	0	1	0	1	0	1	0	1	0	1	0	0
女性	スタブエース	0	1	0	1	0	1	0	0	1	0	1	0	0	1
女性	デバッグスター	1	0	1	1	0	1	1	0	1	1	0	1	1	0
女性	ゴットハンドⅡ	1	0	1	1	0	1	0	0	0	1	0	0	0	1
女性	ユーザビリティⅠ	1	0	0	0	1	1	1	1	1	0	0	0	0	1
女性	スタブエース	1	0	0	0	1	0	1	0	0	1	1	1	1	0

000→デバッグスター
011→ゴットハンドⅡ
101→ユーザビリティⅠ
110→スタブエース

割付完了

直交表への割付け ～13～

* 武器の割付

→線点図より「5」、「10」、「15」列を利用して割り付け

- ①5,10,15列を抽出
- ②5,10,15列を寄せる
- ③5,10,15列を結合する
- ④水準を割り付ける

	1	4	8	12	5	10	15	2	3	6	7	9	11	13	14	16	17	18	19	20
男性	デバッグスター				バズーカ			0	0	0	0	0	0	0	0	0	0	0	0	0
男性	ゴットハンドⅡ				マシンガン			0	0											1
男性	ユーザビリティⅠ				ライフル			0												1
男性	スタブエース				サーベル			0	0											0
男性	デバッグスター				マシンガン			1	1											1
男性	ゴットハンドⅡ				バズーカ			1	1											0
男性	ユーザビリティⅠ				サーベル			1	1	1										0
男性	スタブエース				ライフル			1	1	0										1
女性	デバッグスター				ライフル			0	1	0										0
女性	ゴットハンドⅡ				サーベル			0	1	0										1
女性	ユーザビリティⅠ				バズーカ			0	1	1	0									1
女性	スタブエース				マシンガン			0	1	1	0	0	0	0	1	0				0
女性	デバッグスター				サーベル			1	0	1	0	1	0	1	0	1	1			1
女性	ゴットハンドⅡ				ライフル			1	0	1	0	0	1	0	0	0	0			0
女性	ユーザビリティⅠ				マシンガン			1	0	0	1	1	0	0	0	0	0			0
女性	スタブエース				バズーカ			1	0	0	1	0	1	1	1	1	1			1

000→バズーカ
011→マシンガン
101→ライフル
110→サーベル

割付完了

直交表への割付け ～14～

* 防具の割付

→線点図より「6」、「11」、「13」列を利用して割り付け

- ①6,11,13列を抽出
- ②6,11,13列を寄せる
- ③6,11,13列を結合する
- ④水準を割り付ける

	1	4	8	12	5	10	15	6	11	13	2	
男性	デバッグスター				バズーカ			金のシールド	0	0	0	0
男性	ゴットハンドⅡ				マシンガン			鉄のシールド	0	0	0	1
男性	ユーザビリティⅠ				ライフル			木のシールド	0	0	0	1
男性	スタブエース				サーベル			110	0	0	0	0
男性	デバッグスター				マシンガン			110	1	0	0	0
男性	ゴットハンドⅡ				バズーカ			木のシールド	1	0	0	0
男性	ユーザビリティⅠ				サーベル			鉄のシールド	1	1	0	0
男性	スタブエース				ライフル			金のシールド	1	0	1	1
女性	デバッグスター				ライフル			鉄のシールド	0	1	1	0
女性	ゴットハンドⅡ				サーベル			金のシールド	0	0	0	0
女性	ユーザビリティⅠ				バズーカ			110	0	0	0	0
女性	スタブエース				マシンガン			木のシールド	0	0	0	0
女性	デバッグスター				サーベル			木のシールド	1	0	0	0
女性	ゴットハンドⅡ				ライフル			110	1	0	0	0
女性	ユーザビリティⅠ				マシンガン			金のシールド	1	0	0	0
女性	スタブエース				バズーカ			鉄のシールド	1	0	0	0

④水準を割り付ける

000→金のシ
011→鉄のシ
101→木のシ

「110」があまって
→どの水準を割り付
が、よりテストをしたい
付けるのがよい。

ex) よく使う防具とか。

000→金のシールド
011→鉄のシールド
101→木のシールド

「110」があまっている。
→どの水準を割り付けてもよいが、よりテストをしたい水準を割り付けるのがよい。

ex) よく使う防具とか。。。

直交表への割付け ～15～

* 防具の割付

→線点図より「6」、「11」、「13」列を利用して割り付ける。

	1	4	8	12	5	10	15	6	11	13	2	3	7	9	14
男性	デバッグスター				バズーカ			金のシールド			0	0	0	0	0
男性	ゴットハンドⅡ				マシンガン			鉄のシールド			0	0	0	1	1
男性	ユーザビリティⅠ				ライフル			木のシールド			0	0	1	0	1
男性	スタブエース				サーベル			金のシールド			0				
男性	デバッグスター				マシンガン			金のシールド							
男性	ゴットハンドⅡ				バズーカ			木のシールド							
男性	ユーザビリティⅠ				サーベル			鉄のシールド			1				
男性	スタブエース				ライフル			金のシールド			1				
女性	デバッグスター				ライフル			鉄のシールド			0				
女性	ゴットハンドⅡ				サーベル			金のシールド			0				
女性	ユーザビリティⅠ				バズーカ			金のシールド			0				
女性	スタブエース				マシンガン			木のシールド			0	1	0	0	0
女性	デバッグスター				サーベル			木のシールド			1	0	0	1	1
女性	ゴットハンドⅡ				ライフル			金のシールド			1	0			
女性	ユーザビリティⅠ				マシンガン			金のシールド			1	0			
女性	スタブエース				バズーカ			鉄のシールド			1	0			

今回は、
デフォルト設定でよく使わ
れる「金のシールド」
を「110」に割り付けた。

割付完了

直交表への割付け ～16～

* 割付完了

→2因子間の網羅が100%になっている。

例えば、「ゴットハンドⅡ」に注目すると

因子	因子2	因子3	
男性	デバッグスター	バズーカ	金のシールド
男性	ゴットハンドⅡ	マシンガン	鉄のシールド
男性	ユーザビリティⅠ	ライフル	木のシールド
男性	スタブエース	サーベル	金のシールド
男性	デバッグスター	マシンガン	金のシールド
男性	ゴットハンドⅡ	バズーカ	木のシールド
男性	ユーザビリティⅠ	サーベル	鉄のシールド
男性	スタブエース	ライフル	金のシールド
女性	デバッグスター	ライフル	鉄のシールド
女性	ゴットハンドⅡ	サーベル	金のシールド
女性	ユーザビリティⅠ	バズーカ	金のシールド
女性	スタブエース	マシンガン	木のシールド
女性	デバッグスター	サーベル	木のシールド
女性	ゴットハンドⅡ	ライフル	金のシールド
女性	ユーザビリティⅠ	マシンガン	金のシールド
女性	スタブエース	バズーカ	鉄のシールド

パイロット:

男性、女性

武器:

バズーカ、マシンガン、
ライフル、サーベル

防具:

金、鉄、木

どれも必ず組合せになっている。

どの因子のどの水準
をとっても同様！！

テストケース～1～

* テストケースへ

因子	因子2	因子3	因子4	2	3	7	9	14
男性	デバッグスター	バズーカ	金のシールド	0	0	0	0	0
男性	ゴットハンドⅡ	マシンガン	鉄のシールド	0	0	0	1	1
男性	ユーザビリティⅠ	ライフル	木のシールド	0	0	1	0	1
男性	スタブエース	サーベル	金のシールド	0	1	1	1	0
男性	デバッグスター	マシンガン	金のシールド	1	1	0	0	1
男性	ゴットハンドⅡ	バズーカ	木のシールド	1	1	1	0	1
男性	ユーザビリティⅠ	サーベル	鉄のシールド	1	1	0	1	1
男性	スタブエース	ライフル	金のシールド	1	1	0	1	1
女性	デバッグスター	ライフル	鉄のシールド	0	1	1	1	0
女性	ゴットハンドⅡ	サーベル	金のシールド	0	1	1	0	1
女性	ユーザビリティⅠ	バズーカ	金のシールド	0	1	0	1	1
女性	スタブエース	マシンガン	木のシールド	0	1	0	0	0
女性	デバッグスター	サーベル	木のシールド	1	0	0	1	1
女性	ゴットハンドⅡ	ライフル	金のシールド	1	0	0	0	0
女性	ユーザビリティⅠ	マシンガン	金のシールド	1	0	1	1	0
女性	スタブエース	バズーカ	鉄のシールド	1	0	1	0	1

1行がそれぞれ
テストケース

テストケース ～2～

* テストケース

Case1:	「男性」	- 「デバッグスター」	- 「バズーカ」	- 「金のシールド」
Case2:	「男性」	- 「ゴットハンドⅡ」	- 「マシンガン」	- 「鉄のシールド」
Case3:	「男性」	- 「ユーザビリティⅠ」	- 「ライフル」	- 「木のシールド」
Case4:	「男性」	- 「スタブエース」	- 「サーベル」	- 「金のシールド」
Case5:	「男性」	- 「デバッグスター」	- 「マシンガン」	- 「金のシールド」
Case6:	「男性」	- 「ゴットハンドⅡ」	- 「バズーカ」	- 「木のシールド」
Case7:	「男性」	- 「ユーザビリティⅠ」	- 「サーベル」	- 「鉄のシールド」
Case8:	「男性」	- 「スタブエース」	- 「ライフル」	- 「金のシールド」
Case9:	「女性」	- 「デバッグスター」	- 「ライフル」	- 「鉄のシールド」
Case10:	「女性」	- 「ゴットハンドⅡ」	- 「サーベル」	- 「金のシールド」
Case11:	「女性」	- 「ユーザビリティⅠ」	- 「バズーカ」	- 「金のシールド」
Case12:	「女性」	- 「スタブエース」	- 「マシンガン」	- 「木のシールド」
Case13:	「女性」	- 「デバッグスター」	- 「サーベル」	- 「木のシールド」
Case14:	「女性」	- 「ゴットハンドⅡ」	- 「ライフル」	- 「金のシールド」
Case15:	「女性」	- 「ユーザビリティⅠ」	- 「マシンガン」	- 「金のシールド」
Case16:	「女性」	- 「スタブエース」	- 「バズーカ」	- 「鉄のシールド」

注意点

* 直交表の注意点。

- 直交表を使う場合は、あくまでも2因子網羅100%
→ 3因子以上のテストは別途検討
 - 影響範囲の大きい因子は固定でやる とか
 - 過去プロジェクトで問題傾向を分析して狙う とか
- 因子と水準の抽出がポイント
→ 実際にはもっと水準が多い場合もある
 - 他のテスト設計技法を組み合わせる効果的に。
例えば同値分割など。

JaSST'12 Tokyo

ソフトウェアテストシンポジウム2012 東京
<http://jasst.jp/>

ソフトウェアテストに関する国内最大級のシンポジウム

ソフトウェアテストシンポジウムは、ソフトウェアのテスト、レビューなど品質、信頼性に特化した国内最大級のイベントであり、昨年はのべ1,600人の方が来場されています。最先端の話題から実践事例まで、活発な議論や情報交換を行う場です。

2012年1月25,26日

東京 目黒雅叙園にて開催決定！！



JaSST'12 Tokyo

●基調講演 (これまでの実績)

2003年 山浦 恒央 氏
2004年 Tom Demarco 氏
2005年 Rex Black 氏
2006年 Rick Craig 氏
2007年 Ed Yourdon 氏
2008年 Capers Jones 氏
2009年 Roger S. Pressman 氏
2010年 Johanna Rothman 氏
2011年 Lee Copeland氏
2012年 ??? (今回もビッグネームを調整中)

●プログラム (予定)

- ・基調講演
- ・招待講演
- ・チュートリアル
- ・パネルディスカッション
- ・論文・事例発表
- ・テクノロジーセッション
- ・その他企画セッション多数
(テスト設計コンテスト、ライブセッションなど)



今後のJaSST開催のご案内

JaSSTは全国各地で行われています。

JaSST 北海道
2011.10.21(札幌) 開催決定

JaSST 新潟
2012.2(新潟) 開催予定

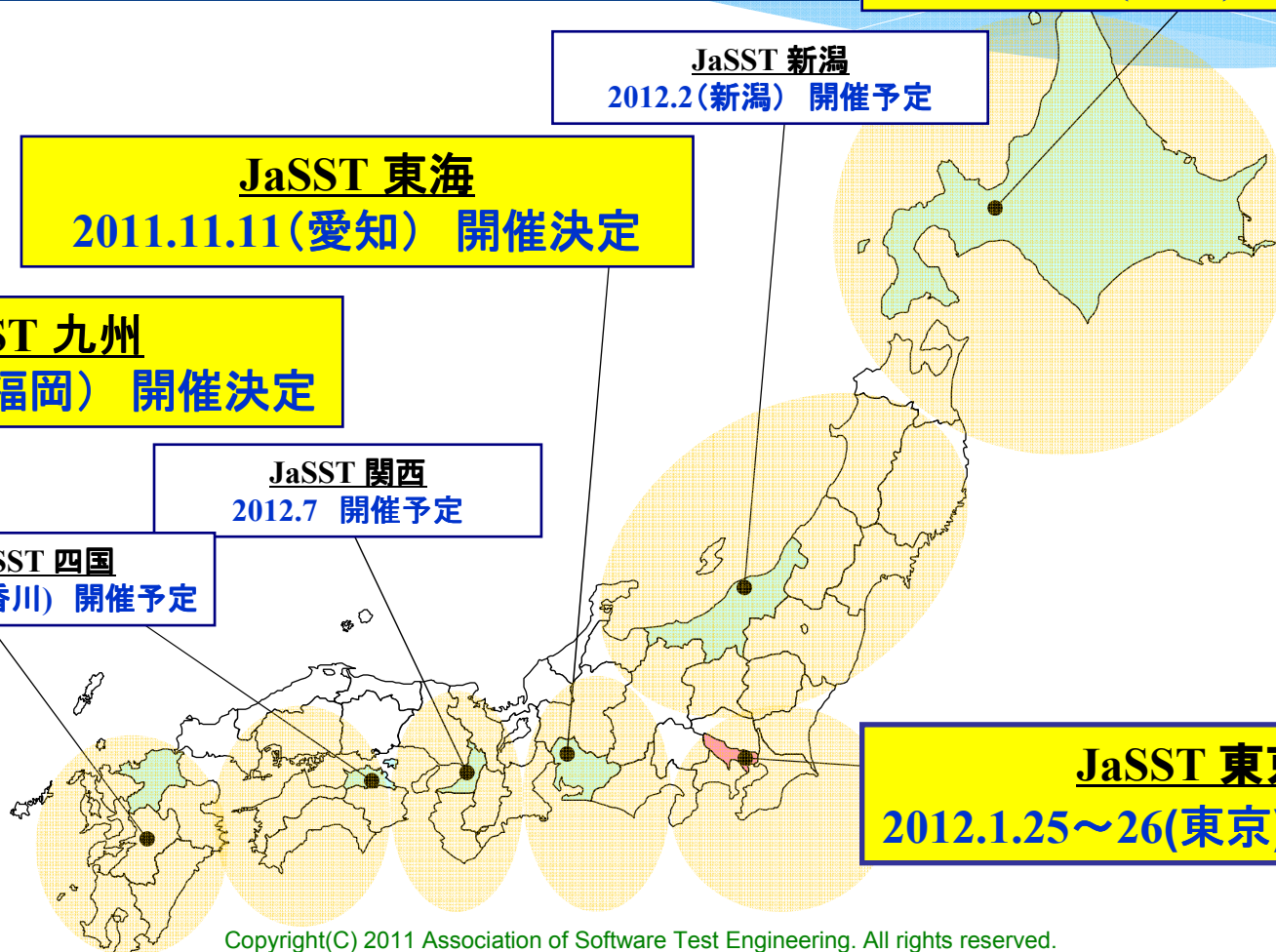
JaSST 東海
2011.11.11(愛知) 開催決定

JaSST 九州
2011.11.25(福岡) 開催決定

JaSST 関西
2012.7 開催予定

JaSST 四国
2012.7(香川) 開催予定

JaSST 東京
2012.1.25~26(東京) 開催決定



ありがとうございました。

*ご清聴ありがとうございました。

*次は1月25日に目黒雅叙園で
お会いしましょう。