

# セッション: A5

## 三賢者、テストを語る

---

悩める若者とスーパー上司:

若者: 池田 暁 (日立情報通信エンジニアリング)

上司: 前畑 佳史 (島津エス・ディー)

テストの賢者:

CFDの賢者: 松尾谷 徹 (デバッグ工学研究所)

CEGの賢者: 秋山 浩一 (富士ゼロックス)

DTの賢者: 鈴木 三紀夫 (TIS)

# テーマ1 クイズ判定BOX問題

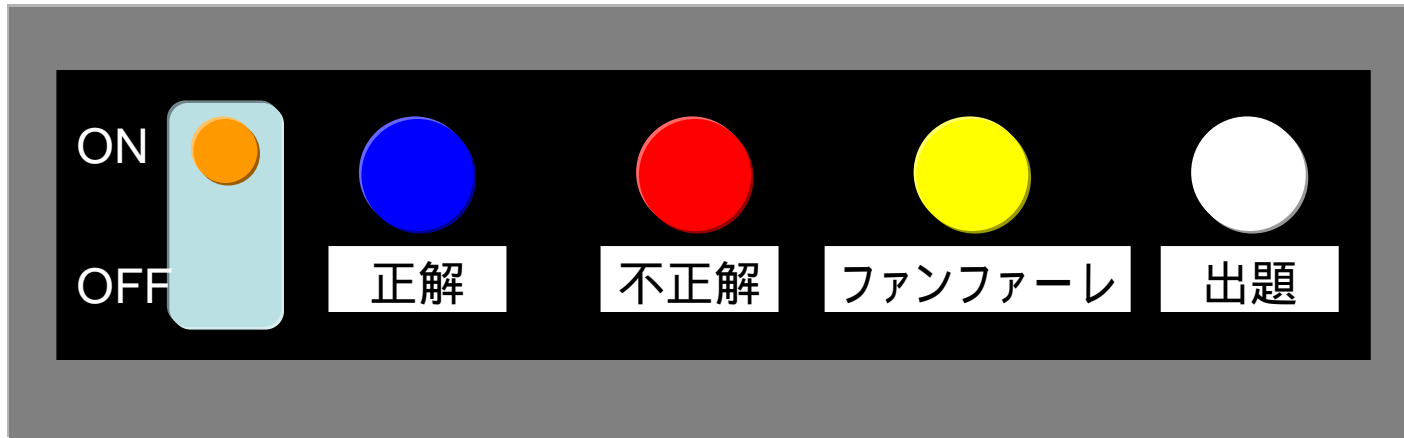
---

組込み系

# テーマ1 クイズ判定BOX問題

- 問題

- これはクイズで用いる道具「クイズ判定BOX」です。



- 箱にはボタンが4つ、スイッチが1つ付いています。
  - 電源スイッチ (ON / OFFを切り替えられます)
  - 正解ボタン (ピンポンという音になります)
  - 不正解ボタン (ブーという音になります)
  - ファンファーレボタン (ファンファーレになります)
  - 出題ボタン (クイズの出題のときに使用します。ジャンという音になります)

# テーマ1 クイズ判定BOX問題

- 問題


- 電源がONの場合

- 「正解」「不正解」「ファンファーレ」「出題」のボタンを押すと、それぞれの音がなります。
    - 音が鳴っている最中に、他のボタンを押しても無効になります。  
(最初の音がそのまま鳴り続けます)

- 電源がOFFの場合

- 「正解」「不正解」「ファンファーレ」「出題」のボタンを押しても、音は鳴りません。

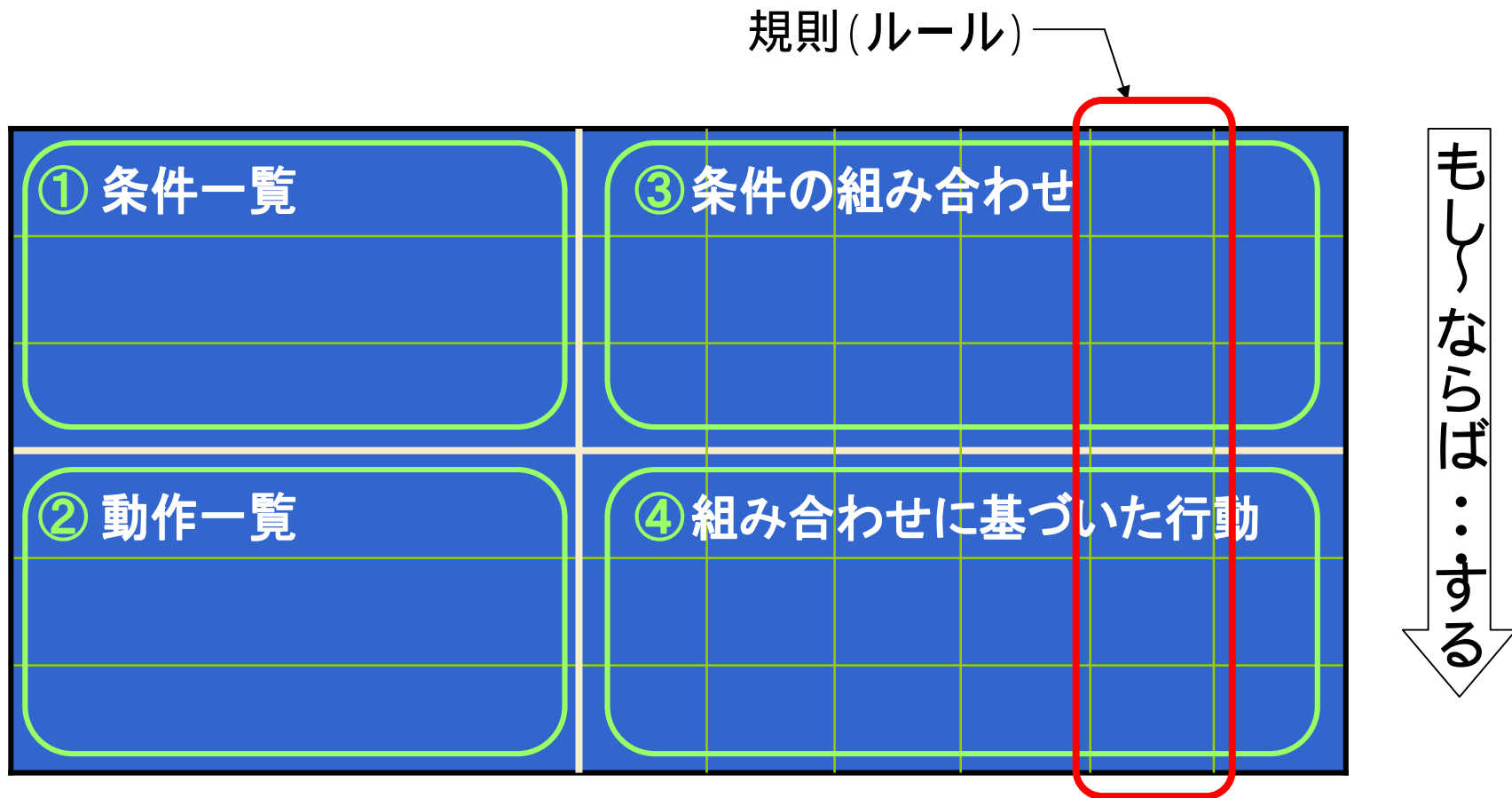
# テーマ1 クイズ判定BOX問題



DT (デシジョンテーブル) の場合

# デシジョンテーブル(DT)の説明

- デシジョンテーブル(DT)の説明



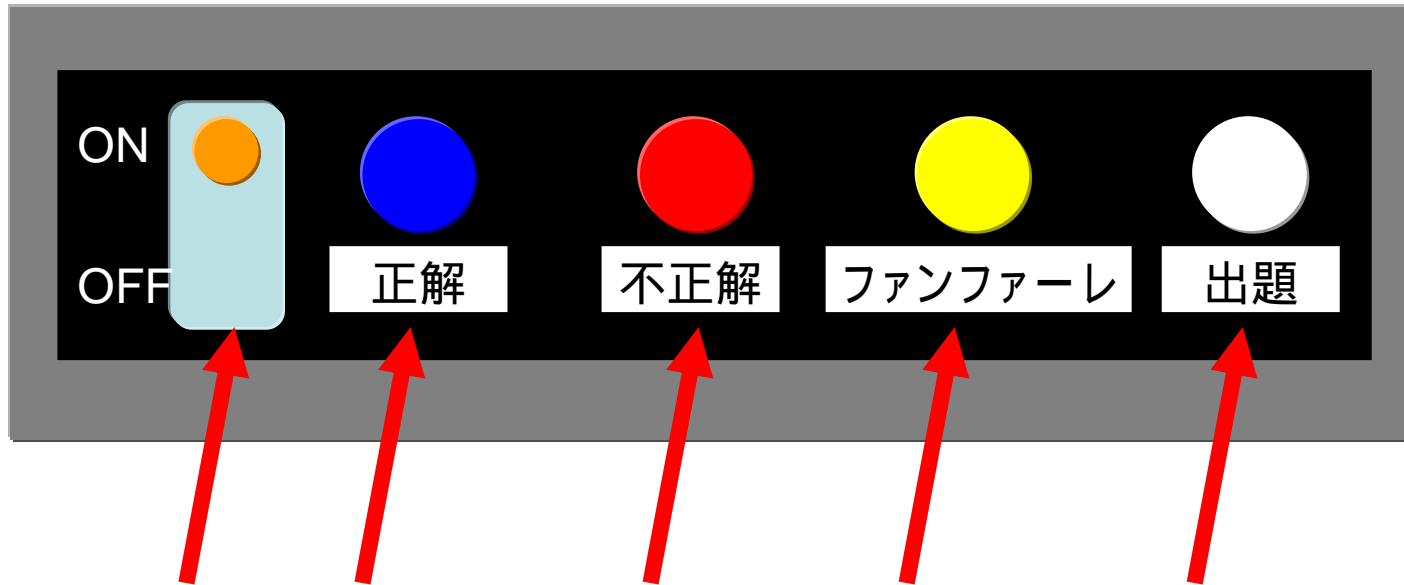
# DTで解いてみる



- 前提
  - 単機能のテストは実施済みであること。
    - 電源ON / OFFで、電源が付くこと
    - 電源が付いただけでは音が鳴らないこと
    - 各ボタンを押して、それぞれの音が鳴ること
    - 「音が鳴りっぱなしで止まらない」ことは無いこと

# DTで解いてみる

- 最初の考え



五つあるが、OFFのときはどのボタンを押しても音がならないことを確認すればよいだろう。

だから正解ボタン～出題ボタンを条件にして、その組合せでできるのではないか？







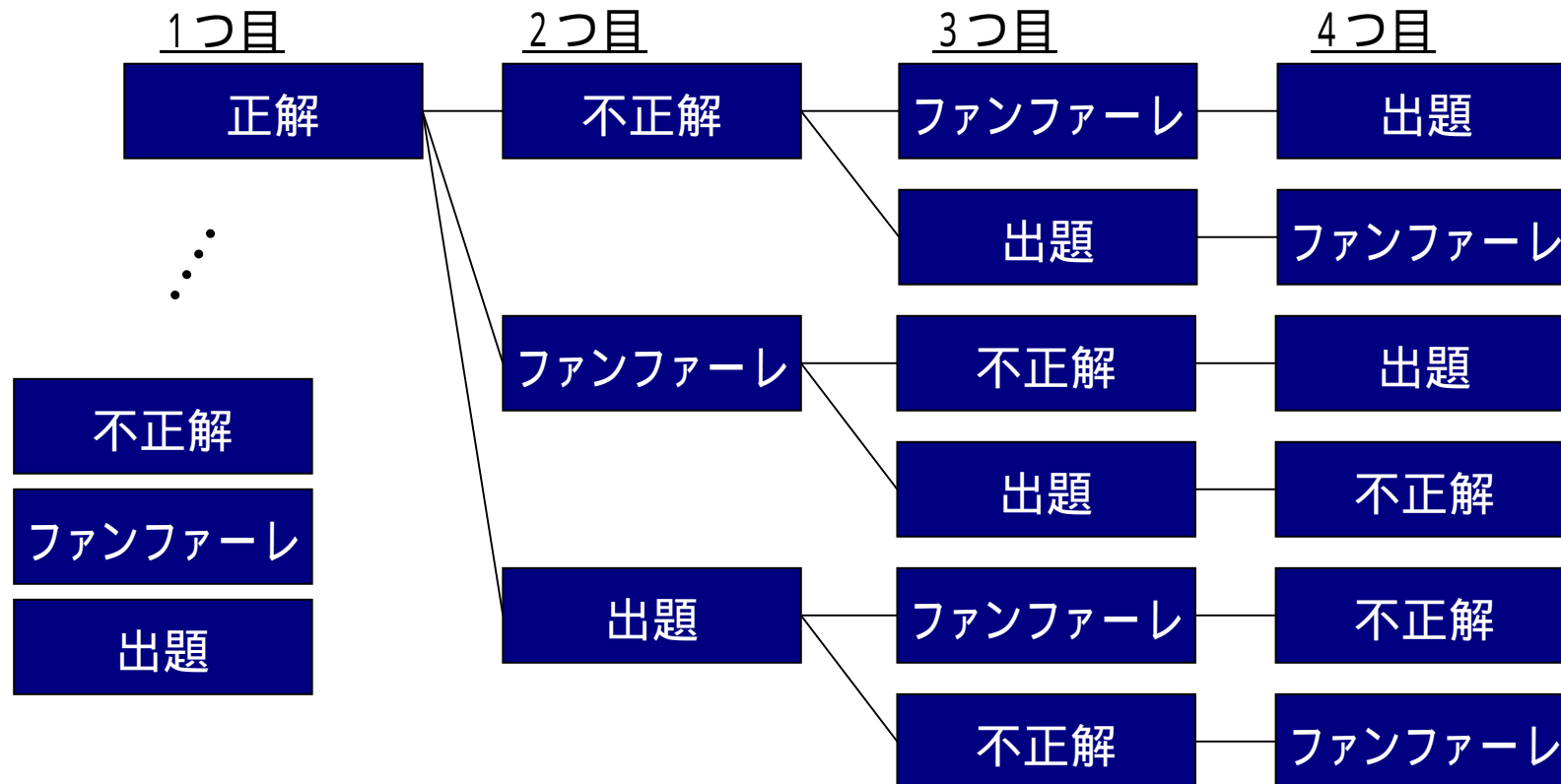
# 単機能テスト分のルールを取り除く

	1	2	3	4	5	6	7	8	9	10	11	12
電源スイッチ	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
正解ボタン	Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N
不正解ボタン	N	Y	Y	Y	Y	N	N	N	Y	Y	Y	N
ファンファーレ	N	Y	Y	N	N	Y	Y	N	Y	Y	N	Y
出題ボタン	N	Y	N	Y	N	Y	N	Y	Y	N	Y	Y
音が鳴る		X	X	X	X	X	X	X	X	X	X	X
音が鳴らない	X											

何か違うような？？

# 考えを変えてみる

- 正解ボタン 不正解ボタンと 不正解ボタン 正解ボタン  
は同じ？ 順番を意識する必要は無いか？
- 4つのボタンを一度に押すことを考えてみる



## DTで解いてみる

- 4つの組み合わせは、 $4 \times 3 \times 2 \times 1 = 24$
- 電源OFFのときのテストを入れると、 $24 + 1 = 25$  ケース
- 仕様を確認してみる
  - 4つのボタンを押して意味があるか？
  - 「ピンポン」と一回だけしかならなかったら、4つ押す時間は無いだろう。せいぜい正解と不正解のように2つのボタンしか押せないのでは無いか
  - 仕様を見直してみる

これでテストしてみる

### –電源がONの場合

- 「正解」「不正解」「ファンファーレ」「出題」のボタンを押すと、それぞれの音がなります。
- 音が鳴っている最中に、他のボタンを押しても無効になります。
- (最初の音がそのまま鳴り続けます)

鳴り続けるのであれば4つのボタンは押せるのではないか。

# テーマ1 クイズ判定BOX問題



CEG (原因結果グラフ) の場合

# 原因結果グラフとは

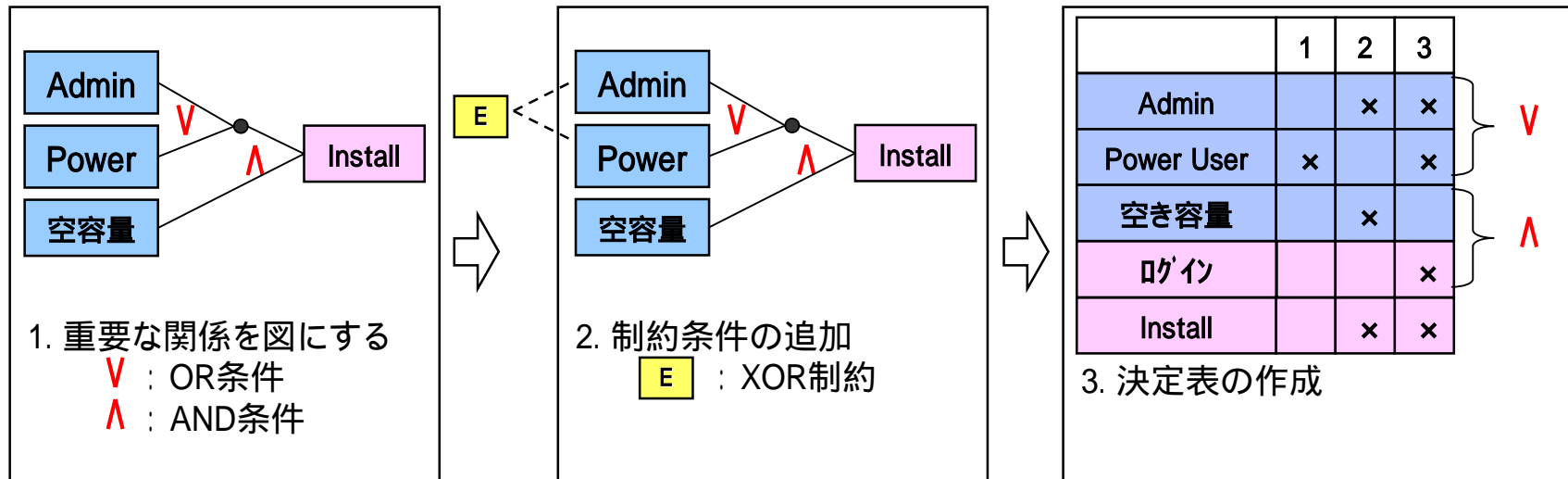
## ◆ 原因結果グラフ(Cause-Effect Graph = CEG)

--- 複雑なケースを持つ仕様からテストスクリプトを作成する手法 ---

1. 仕様を原因と結果に分けて原因結果グラフにまとめる
2. 制約条件を書き加えグラフを完成させる
3. 出来上がったグラフを決定表(デシジョンテーブル)に変換する

< 例 >

「Administratorまたは、Power Userがログインしている場合に、ディスク容量があれば、インストールを開始することができる」といった機能(断片化された仕様)を確認する場合、

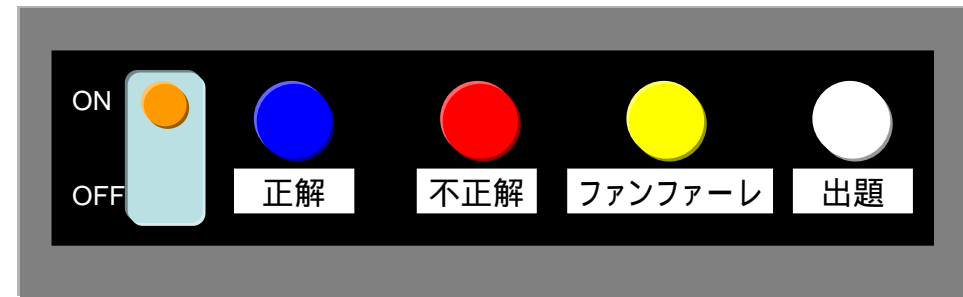


# 原因は色々ありそうだが...

- ボタンの押し方(原因)には

- 軽く押す
- 長押しする
- 押し続ける
- 普通に押す
- 同時に複数押す
- 押す順番を変える
- 必要以上の力で押す
- 同じボタンを連打する
- すばやく違うボタンを押す
- ボタンを押した状態で電源を入れる
- ボタンを押した状態で電源を切り入りする
- 電池が切れそうな状況を作ってボタンを押す
- 過酷な条件(寒い、湿度が高い、水の中、真空)で押してみる

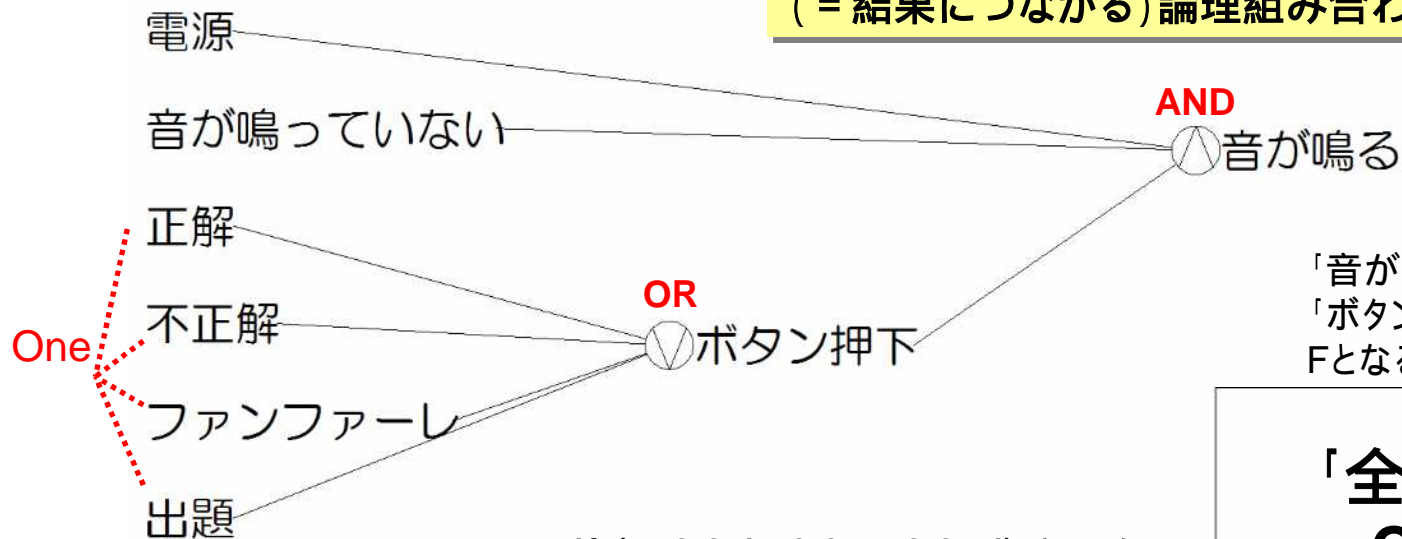
左記のとおり、考え出すときりがない。  
「クイズ判定BOX」の利用者の立場から  
◆ 重要な原因( = 普通にやること)と、  
◆ 重要な結果( = 音が鳴らない/音が鳴り続ける)  
について論理判定可能な原因結果グラフを作成する





# クイズ判定BOXの原因結果グラフ

全ノードについてTrueとFalseの両方が必ずテストされる  
(=結果につながる)論理組み合わせを選択する



「音が鳴る」の#4はテストされないが  
「ボタン押下」が制約Oneにより  
Fとなる条件がありえ無いので良い

if (a || b || c || d) {xxx}

「ボタン押下」の CEG (OR)		TEST # 1	TEST # 2	TEST # 3	TEST # 4
Causes:					
正解		T	F	F	F
不正解		F	T	F	F
ファンファーレ		F	F	T	F
出題		F	F	F	T
Effects:					
音が鳴る	{obs}	T	T	T	T

if (a && b && c && d) {xxx}

「音が鳴る」の CEG (AND)		TEST # 1	TEST # 2	TEST # 3	TEST # 4
Causes:					
電源		T	F	T	T
音が鳴っていない		T	T	F	T
ボタン押下		T	T	T	F
Effects:					
音が鳴る	{obs}	T	F	F	F

「全体」の CEG		TEST # 1	TEST # 2	TEST # 3	TEST # 4
Causes:					
電源		T	F	T	T
音が鳴っていない		T	T	F	T
正解		T	F	F	F
不正解		F	T	F	F
ファンファーレ		F	F	T	F
出題		F	F	F	T
Effects:					
ボタン押下		T	T	T	T
音が鳴る	{obs}	T	F	F	T

# テーマ1 クイズ判定BOX問題



CFDの場合

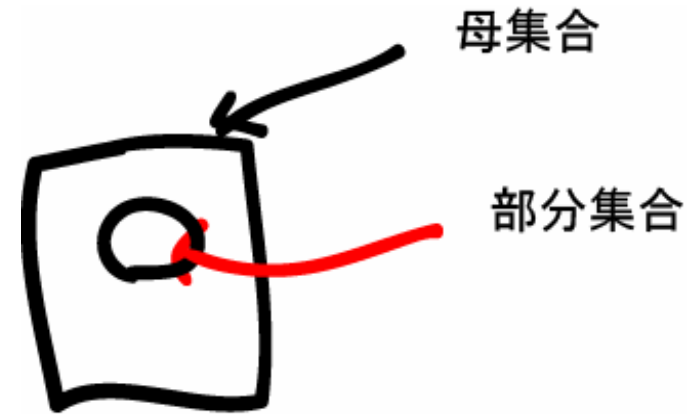
# CFD (Cause Flow Diagram) で考える

- 悩める若者へ
  - 簡単な方法がある

まず

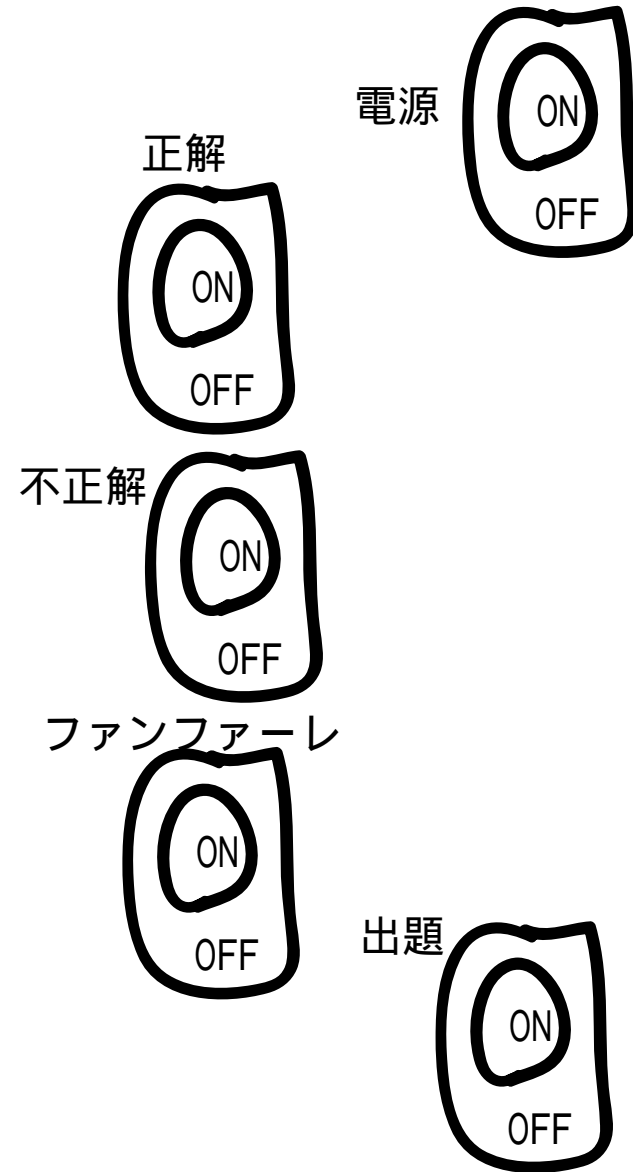
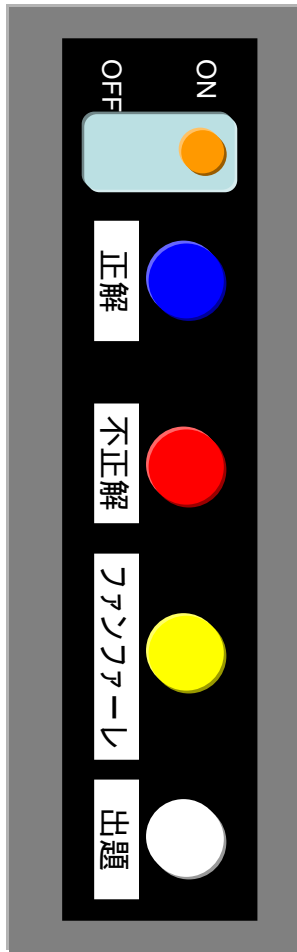
入力をベン図で表す

同値分割 (入力原因)

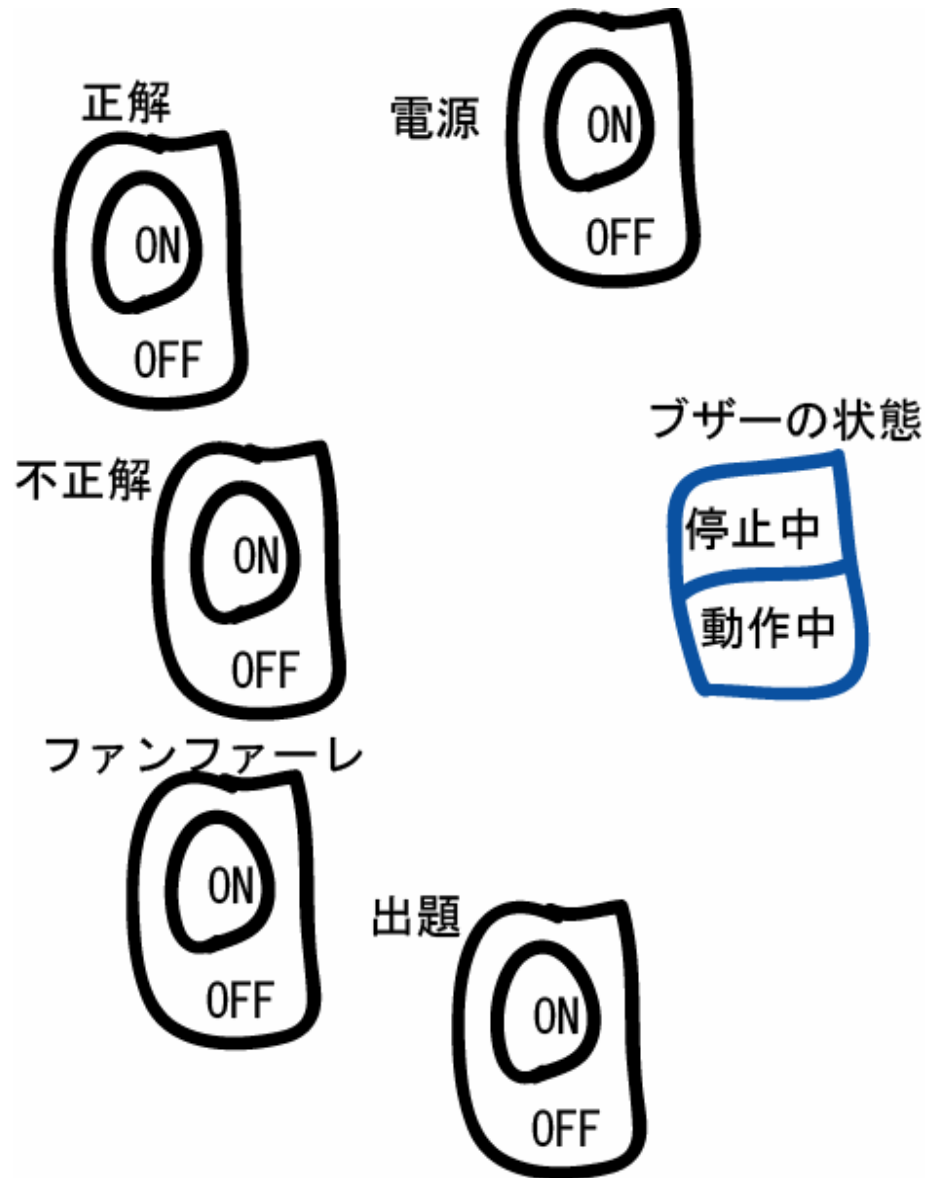


書き方

# 横のものは、立てて考える！？



# 状態を原因に加える(状態原因)



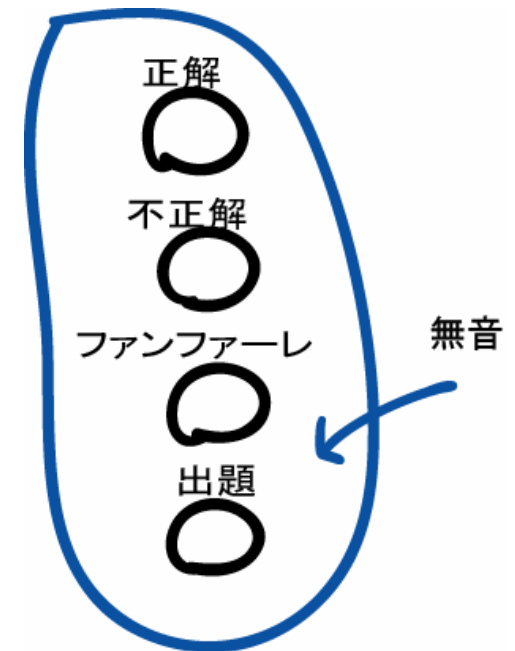
入力原因と  
状態原因を  
洗い出す

# 出力(結果)を洗い出す

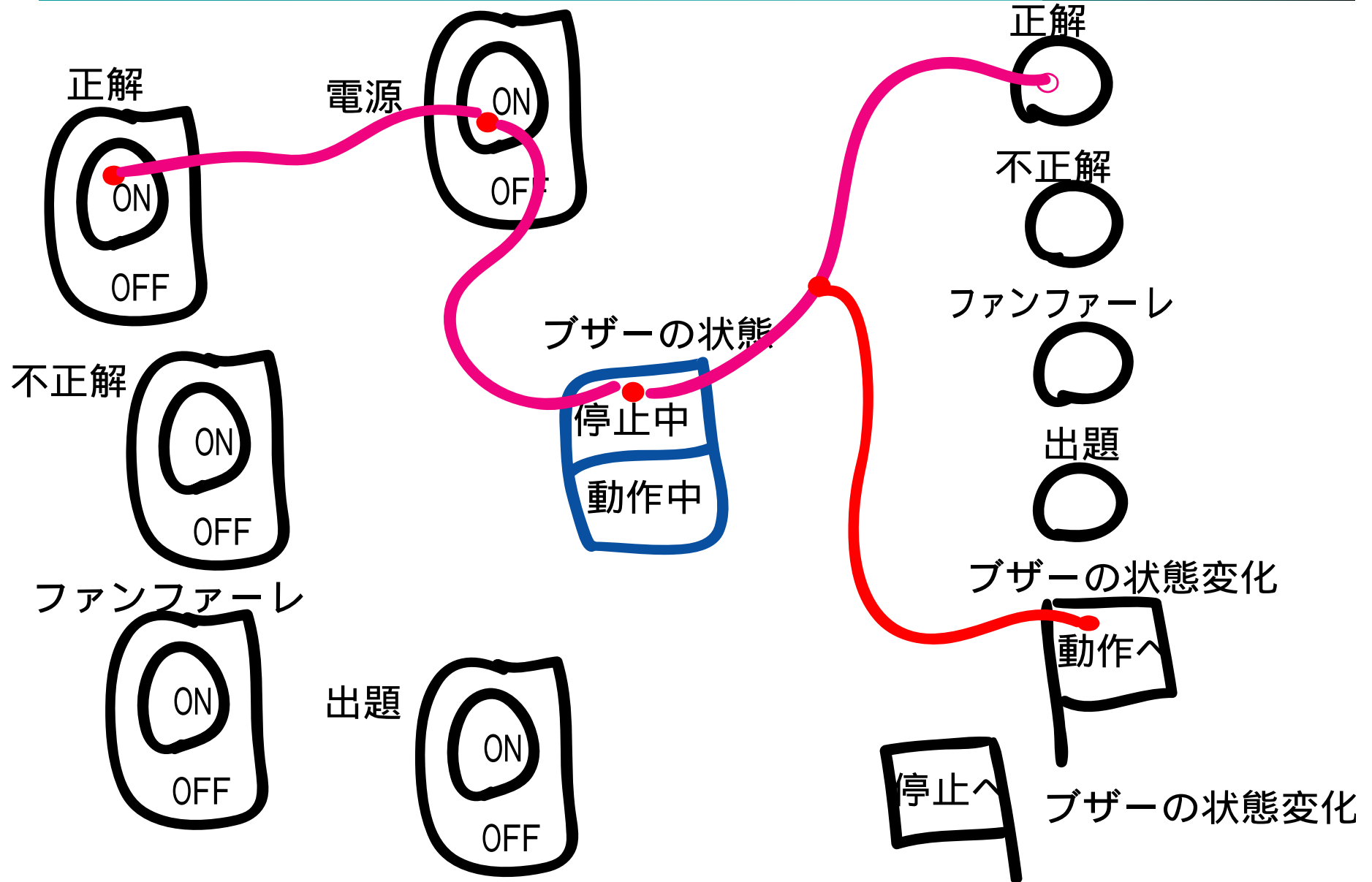


出力を洗い出す  
状態の変化も結果(状態結果)

結果間の排他関係を表す  
こともできる



# 機能を流れ線で表す







## 禁則、無則のテスト



- 午後の講演を聞いてください
- 禁則
  - ブザー動作中は他のSWが禁止されていること
  - 電源OFFでは…
  - ある種の機能
- 無則
  - 何もしないのにブザーが鳴ることは無い
  - マトリックステストの大部分

# 仕様問題



- ブザーの状態を戻す機能、タイミング
  - ブザーが鳴り終わったら
  - ブザーが起動してXX秒後
  - 「正解、不正解、ファンファーレ、出題」のすべてがOFFの場合

## テーマ2 入場料問題

---

エンブラ系


## 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

## テーマ2 入場料問題



DT (デシジョンテーブル) の場合

# 条件を探す

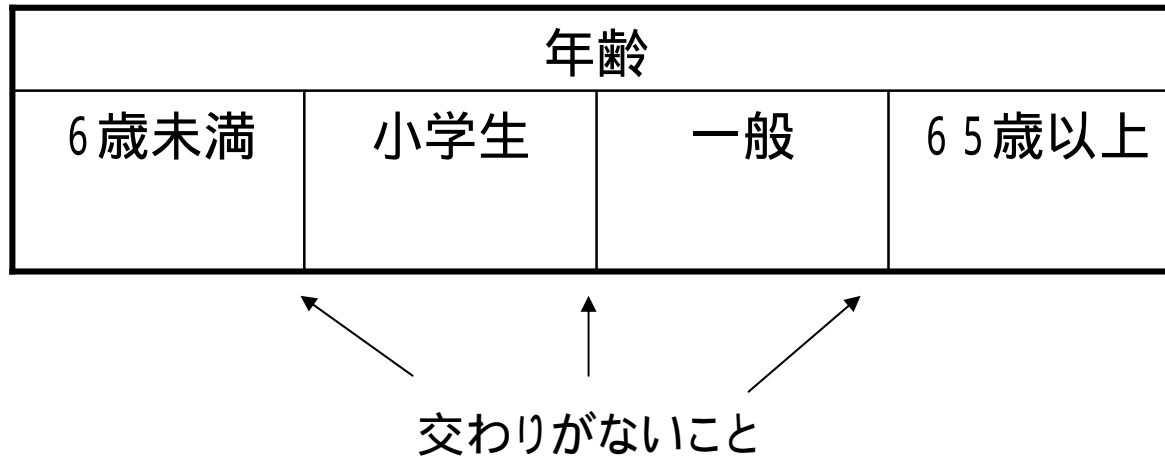
	個人	団体
一般	1200円	1000円
小学生	600円	500円

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

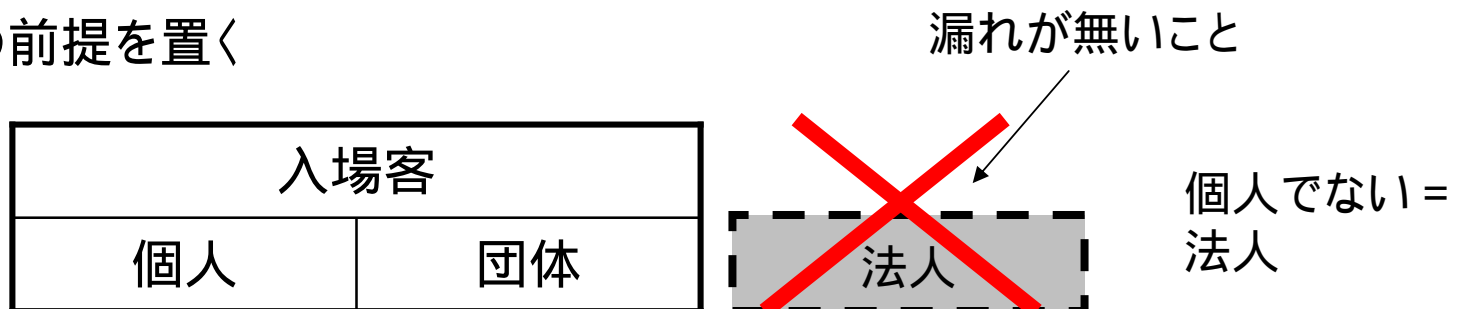
# 条件を探す

- 条件を探す

- 6歳未満、小学生、一般、65歳以上 という条件を考える
- 次の前提を置く



- 個人、団体 という条件を考える
- 次の前提を置く



# DTを作成する

		ル ル1	ル ル2	ル ル3	ル ル4	ル ル5	ル ル6	ル ル7	ル ル8	ル ル9	ル ル10	ル ル11	ル ル12	ル ル13	ル ル14	ル ル15	ル ル16
条件	6歳未満	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N	N	N
	小学生	N	N	N	N	Y	Y	Y	Y	N	N	N	N	N	N	N	N
	一般	N	N	N	N	N	N	N	N	Y	Y	Y	Y	N	N	N	N
	65歳以上	N	N	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y
	県内在住	Y	Y	N	N	Y	Y	N	N	Y	Y	N	N	Y	Y	N	N
	個人	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N
動作	無料	X	X	X	X	X	X							X	X	X	X
	500								X								
	600							X									
	1000										X		X				
	1200									X		X					



# DTを圧縮する

圧縮する

		ル ル1	ル ル2	ル ル3	ル ル4	ル ル5	ル ル6	ル ル7	ル ル8	ル ル9	ル ル10	ル ル11	ル ル12	ル ル13	ル ル14	ル ル15	ル ル16
条件	6歳未満	Y	Y	Y	Y	N	N	N	N	N	N	N	N	N	N	N	N
	小学生	N	N	N	N	Y	Y	Y	Y	N	N	N	N	N	N	N	N
	一般	N	N	N	N	N	N	N	N	Y	Y	Y	Y	N	N	N	N
	65歳以上	N	N	N	N	N	N	N	N	N	N	N	N	Y	Y	Y	Y
	県内在住	Y	Y	N	N	Y	Y	N	N	Y	N	Y	N	Y	Y	N	N
	個人	Y	N	Y	N	Y	N	Y	N	Y	Y	N	N	Y	N	Y	N
動作	無料	X	X	X	X	X	X							X	X	X	X
	500								X								
	600							X									
	1000										X	X					
	1200									X	X						

入れ替える

# DTを圧縮する

		ル ル 1	ル ル 2	ル ル 3	ル ル 4	ル ル 5	ル ル 6	ル ル 7
条件	6歳未満	Y	N	N	N	N	N	N
	小学生	N	N	Y	Y	Y	N	N
	一般	N	N	N	N	N	Y	Y
	65歳以上	N	Y	N	N	N	N	N
	県内在住			Y	N	N		
	個人				Y	N	Y	N
動作	無料	X	X	X				
	500					X		
	600				X			
	1000							X
	1200						X	

## テーマ2 入場料問題

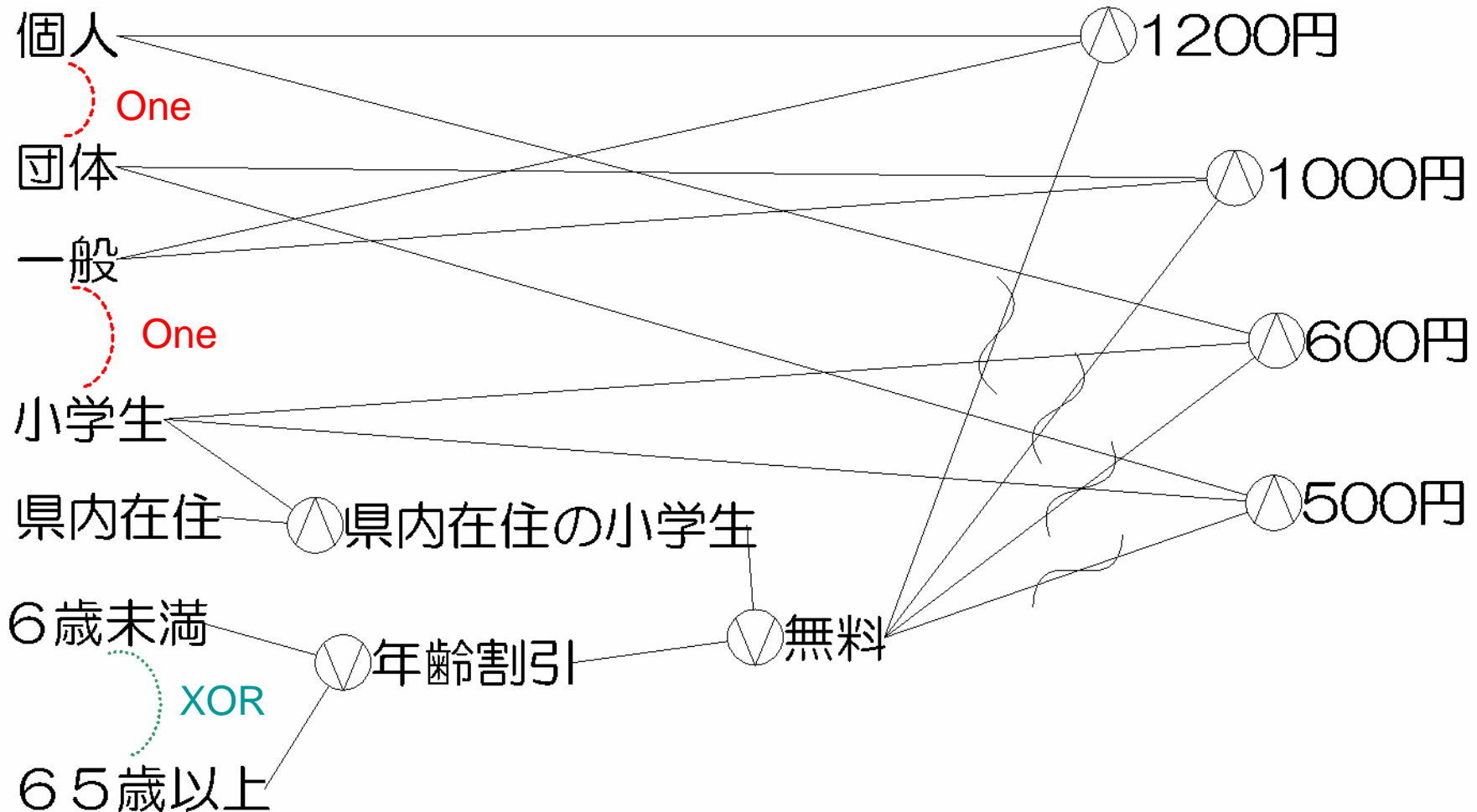


CEG (原因結果グラフ) の場合



# 入場料問題の原因結果グラフ

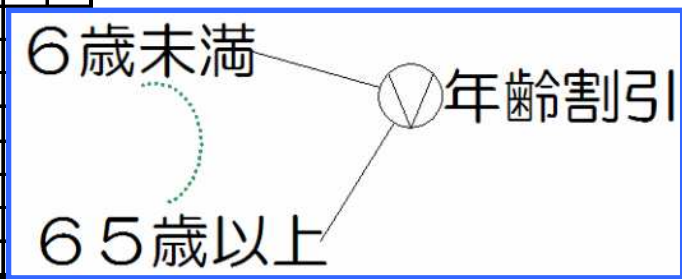
- ◆ 論理接点で必要なロジックが検証されていることを確認する
- ◆ 確認すべき論理展開式は25個



# カバレッジ表 (論理展開式の出現チェック)

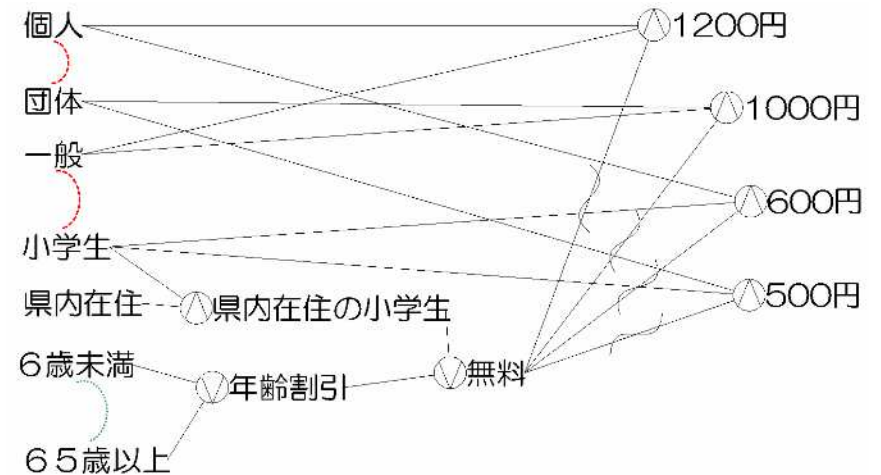
	デシジョンテーブルのテストNo.							AND	OR	OR	AND	AND	AND	AND										
	TEST # 1	TEST # 2	TEST # 3	TEST # 4	TEST # 5	TEST # 6	TEST # 7	小学生	県内在住	県内小学生	6歳未満	65歳以上	年齢割引	not 無料	個人	一般	1200円	小学生	600円	団体	一般	1000円	小学生	500円
1							#	T	T	T														
2	x			x		x	x	F	T	F														
3		x	x		x			T	F	F														
4			x				x				T	F	T											
5	#										F	T	T											
6		x		x	x	x	x				F	F	F											
7	x		x				x			F			T	T										
8							#			T			F	T										
9		x		x	x	x				F			F	F										
10						#							F	T	T	T								
11				#									F	F	T	F								
12		#											F	T	F	F								
13	#												T	T	T	F								
14				#									F	T			T	T						
15					#								F	F			T	F						
16					#								F	T			F	F						
17						#							T	T			T	F						
18		#											F						T	T	T			
19					#								F						F	T	F			
20					#								F						T	F	F			
21						#							T						T	T	F			
22					#								F						T				T	T
23		#											F						F				T	F
24					#								F						T				F	F
25					#								T						T				T	F
xの数	2	3	3	3	3	3	3	1																
#の数	2	3	1	3	3	3	1	3																

- ◆ 各々の論理接点に必要なロジックを展開してカバレッジ表を作成する
- ◆ デシジョンテーブルを作りながらカバレッジ表を埋めていく



# 完成したデシジョンテーブルのチェック

		TEST # 1	TEST # 2	TEST # 3	TEST # 4	TEST # 5	TEST # 6	TEST # 7	TEST # 8
Causes:									
個人		T	T	F	F	F	T	F	T
団体		F	F	T	T	T	F	T	F
一般		T	F	F	T	F	T	T	F
小学生		F	T	T	F	T	F	F	T
6歳未満		F	F	T	F	F	F	T	F
65歳以上		T	F	F	F	F	F	F	F
県内在住		T	F	F	T	F	T	T	T
Effects:									
県内在住の小学生		F	F	F	F	F	F	F	T
年齢割引		T	F	T	F	F	F	T	F
無料		T	F	T	F	F	F	T	T
1200円	{obs}	F	F	f	F	f	T	f	f
1000円	{obs}	f	f	f	T	F	F	F	f
600円	{obs}	f	T	f	f	F	F	f	F
500円	{obs}	f	F	F	F	T	f	f	f

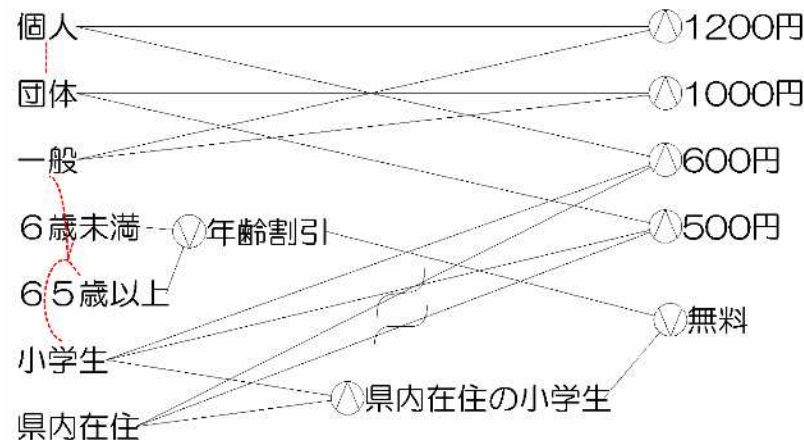


- ◆ 原因結果グラフはグラフさえ正しく書ければあとは単純作業で、テストケースを作成することができる
- ◆ 完成したデシジョンテーブルのチェックで仕様のあいまいさに気づくことができる
  - 65歳以上の一般？
  - 6歳未満の小学生？
  - 一般・小学生・6歳未満・65歳以上は年齢の違いを示す仕様？

「t」 と 「f」 は、論理的にそうならざるを得ないことを表している

# デシジョンテーブルに境界値を入れる

		TEST # 1	TEST # 2	TEST # 3	TEST # 4	TEST # 5	TEST # 6	TEST # 7	TEST # 8
Causes:									
個人		T	T	F	F	F	T	T	F
団体		F	F	T	T	T	F	F	T
一般		F	F	F	F	F	F	T	T
6歳未満		F	F	F	F	F	T	F	F
65歳以上		F	F	F	T	F	F	F	F
小学生		T	T	T	F	T	F	F	F
県内在住		T	F	T	F	F	F	T	T
Effects:									
県内在住の小学生		T	F	T	F	F	F	F	F
年齢割引		F	F	F	T	F	T	F	F
1200円	{obs}	F	F	f	f	f	F	T	F
1000円	{obs}	f	f	F	F	F	f	F	T
600円	{obs}	F	T	f	f	F	F	f	f
500円	{obs}	f	F	F	F	T	f	f	f
無料	{obs}	T	F	T	T	F	T	F	F



## ◆ 境界値問題

**6歳で小学生でない人がいる！！**

**2007年8月1日に6歳になった人は、  
2008年4月1日に小学生になる！**

**(6歳未満の扱いでよいか仕様確認)**

	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

## テーマ2 入場料問題



CFDの場合



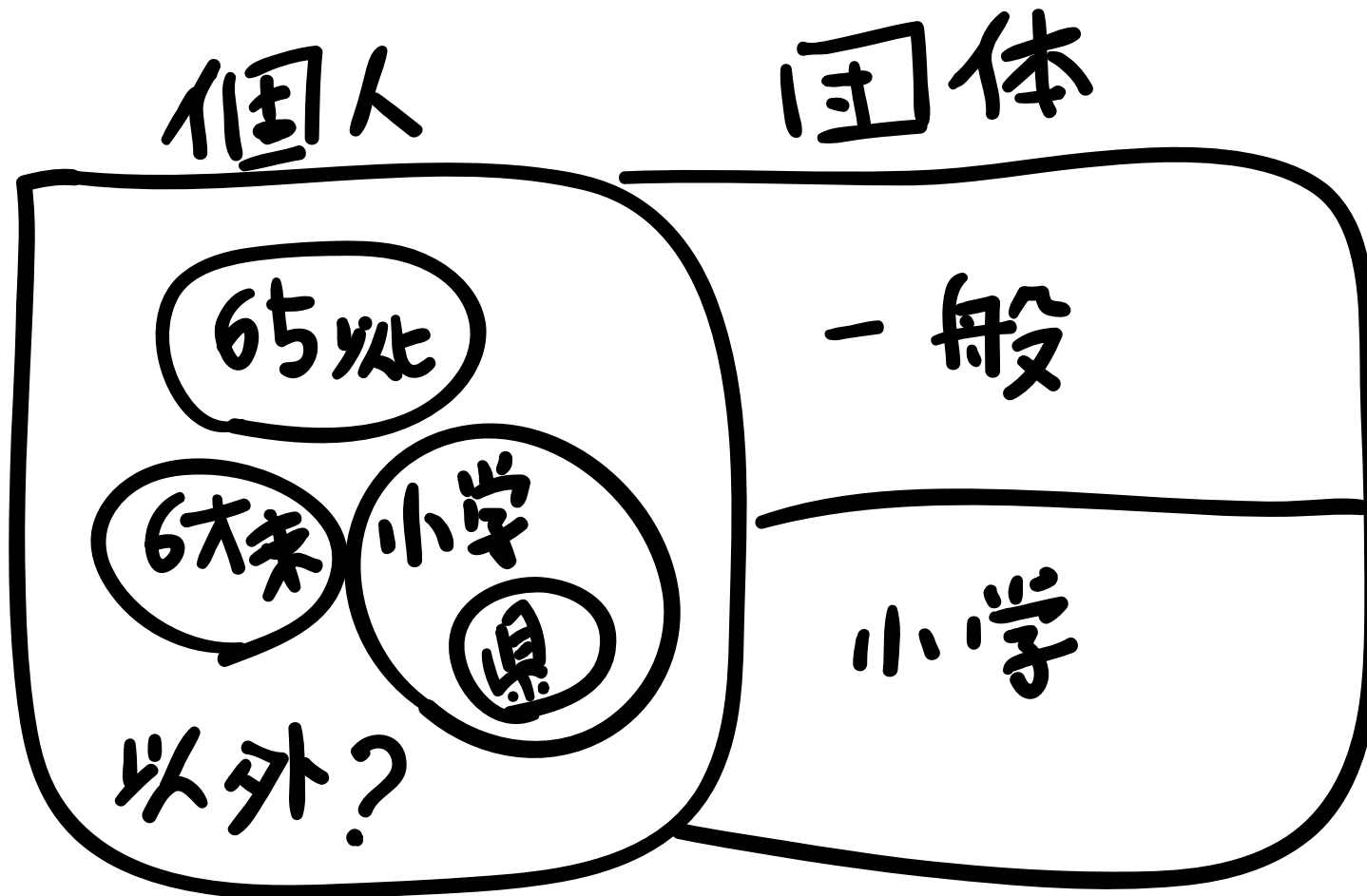
## CFDは、DTの前処理



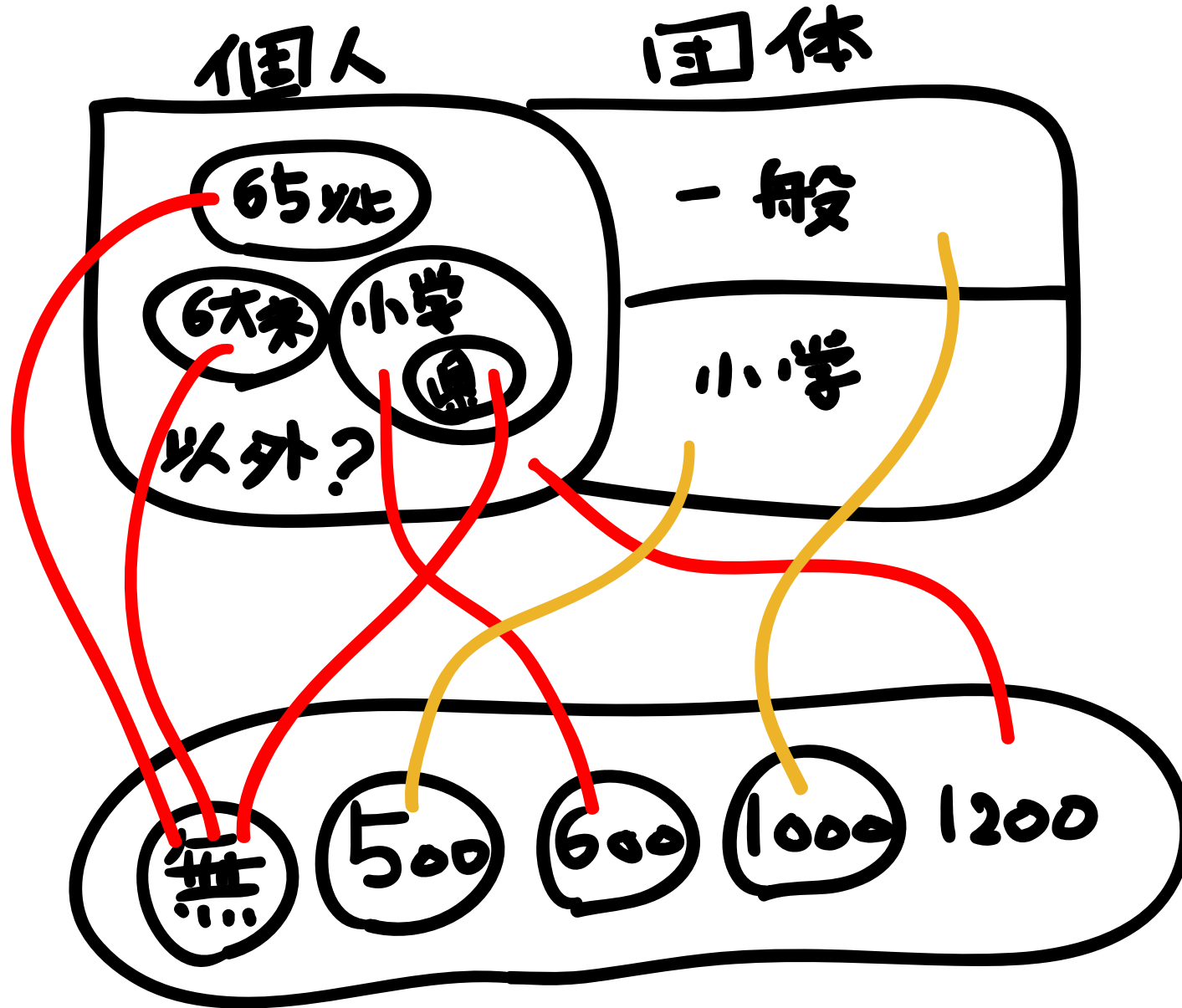
1. 入力<sup>1</sup>の集合演算 同値化 (排他的部分集合)
2. 出力<sup>2</sup>の同値化
3. 原因間<sup>3</sup>の関係 流れ線
4. SWの構造と対比しレビュー
5. DTに表す

# 同値化 同値間を定義する

- 論理的に独立 同値の掛け算になる
- 現実的な入力の関係に注目
  - 65歳と6歳未満が含まれる団体……団体とはしない



# 組合せにならないとして



# DTに直す

			ル ル 1	ル ル 2	ル ル 3	ル ル 4	ル ル 5	ル ル 6	ル ル 7	
条件	個人	6歳未満	Y	N	N	N	N			
		小学生	県内	N	Y	N	N	Y		
			県外	N	N	Y	N	N		
		65歳以上	N	N	N	Y	N			
		それ以外	N	N	N	N	Y			
	団体	一般						Y	N	
		小学生						N	Y	
動作		無料	X	X		X				
		500							Y	
		600			X					
		1000						X		
		1200					X			

## 境界値テストはデフォルト



- 同値間の境界はテストする(仕様の検算を含め)
- 6歳未満と小学生
  - 6歳で未就学者は一般？

# まとめ

---

## 3つの技法のまとめ

- DT

- CEG、CFDの基礎
- 比較的、気軽に使える
- 条件の組合せを導き出すのに、多少のコツが必要

- CEG

- 論理式と制約を用いて、テストケースを作成
- 単純なマトリクステストに比べ、テストケースが圧倒的に削減される
- 論理式と制約の理解が難しいと感じる人が多い

- CFD

- ベン図と流れ図を用いて、テストケースを作成
- 単純なマトリクステストに比べ、テストケースが圧倒的に削減される
- ベン図で制約条件を表現するので、誰でも簡単に書くことができる
- 論理の流れを意識するとよい

# 言いたいこと

若者よ、テスト(デバッグ)は

- 仕様の漏れ、曖昧、誤解を含めて検算すること
- 理屈に溺れるとマトリックステストの無限地獄に堕ちる





また、来年お会いしましょう！