

直交表 超入門 (2)

状態遷移パスのテスト

キャッツ組込みソフトウェア研究所 川崎 秀二*

組込みソフトウェアにおいては、一連の機能動作の検証 状態遷移テスト は基本的に重要である。この状態遷移テストについても、前稿で紹介したソフトウェアテストの効率的な網羅法である直交表を適用することができる。本稿では、この適用の仕方について、CD プレーヤの機能仕様のテストを例題として述べる。

In the test of embedded softwares, those that considers a series of functional behaviors, called state transition (ST) test, are of fundamental importance. One can apply the method of orthogonal array (OA), which is a useful method of software test providing a high and effective coverage of test cases, to the ST tests as well. In this paper, we indicate how to apply the OA to ST tests.

1 はじめに 状態遷移パスのテスト

2 状態遷移表とは CD プレーヤ設計の例題

前稿「直交表 超入門 (1)」では、直交表のごく専門的な事項のうち、全般的基礎部分を紹介した。ソフトウェアテストの重要な対象の一つに、システムの静的構造の検証として、操作や処理のフローに沿った機能テストがある。これは、操作等の順序付きの組み合わせ 状態遷移パス へのテストと考えられる。

本稿では、状態遷移パスのテストが、直交表を用いてどのように行われるかを説明する。

内容としては、先ず状態遷移の情報システムにおける記述法として有効性が認識されている状態遷移表について説明する。次いで、状態遷移表からテストすべき各状態遷移パスをどのようにして機械的に求めるかを示し、その上で抽出されたパスのテストケースをどのようにして直交表へ割り付けるかについて例に沿って述べる。

いま、例題として、CD プレーヤの開発における設計仕様のテストを行う事を考える。次の表 1 のような機能仕様が要求されるとする ([1] より転載)。機能仕様として、例えば電源 OFF かつ CD 未挿入状態からの「再生」は、

- | |
|--|
| 1. (E1) 電源 ON ボタンによる電源 ON 処理 ()
(S1) 電源 ON 状態 |
| 2. (E2) トレーボタンによる CD 挿入処理 ()
(S2) CD 挿入状態 |
| 3. (E3) 再生ボタンによる再生処理 ()
(S3) 再生状態 |

という流れに従って実現される。表 1 から抽出される事象 (event) および状態 (state) として、次のよう

*キャッツ組込みソフトウェア研究所, CATS Embedded Software Laboratory

1	電源ボタンがあり、初期状態は電源 OFF になっている .
2	電源を入れると、最初は電源 ON 状態で停止している .
3	トレーボタンがあり、トレーボタンを押すと即演奏を中止して、トレーを排出する . また電源を OFF にするときにも、即演奏を中止して電源 OFF 状態にする .
4	再生ボタンを押すと、演奏を開始する .
5	停止中には、電源ボタン、トレーボタン、および再生ボタンのみに反応する .
6	一時停止ボタンがあり、演奏を再生している場合に一時停止ボタンを押すと演奏が一時停止され、再度一時停止ボタンを押すと演奏を開始し、再生中状態に戻る .
7	早送りボタンがあり、演奏を再生中に早送りボタンを押すと早送りを始め、再度早送りボタンを押すと再生中状態に戻る .

CD プレーヤの操作は、本体パネルあるいはリモコンにてなされるものとする . CD プレーヤの対応可能なメディアは CD あるいは SACD であるとする .

表 1: CD プレーヤ仕様

な項目がリストされる :

事象	
• E ₀ : 電源ボタン	• E ₁ : トレーボタン
• E ₂ : 再生ボタン	• E ₃ : 停止ボタン
• E ₄ : 一時停止ボタン	• E ₅ : 早送りボタン押下
• E ₆ : 早送りボタン押上	• E ₇ : 巻戻しボタン押下
• E ₈ : 巻戻しボタン押上	

状態	
• S ₀ : 電源 ON	• S ₁ : 電源 OFF
• S ₂ : CD 挿入	• S ₃ : CD 未挿入
• S ₄ : 停止	• S ₅ : 再生
• S ₆ : 早送り	• S ₇ : 巻戻し
• S ₈ : 一時停止	

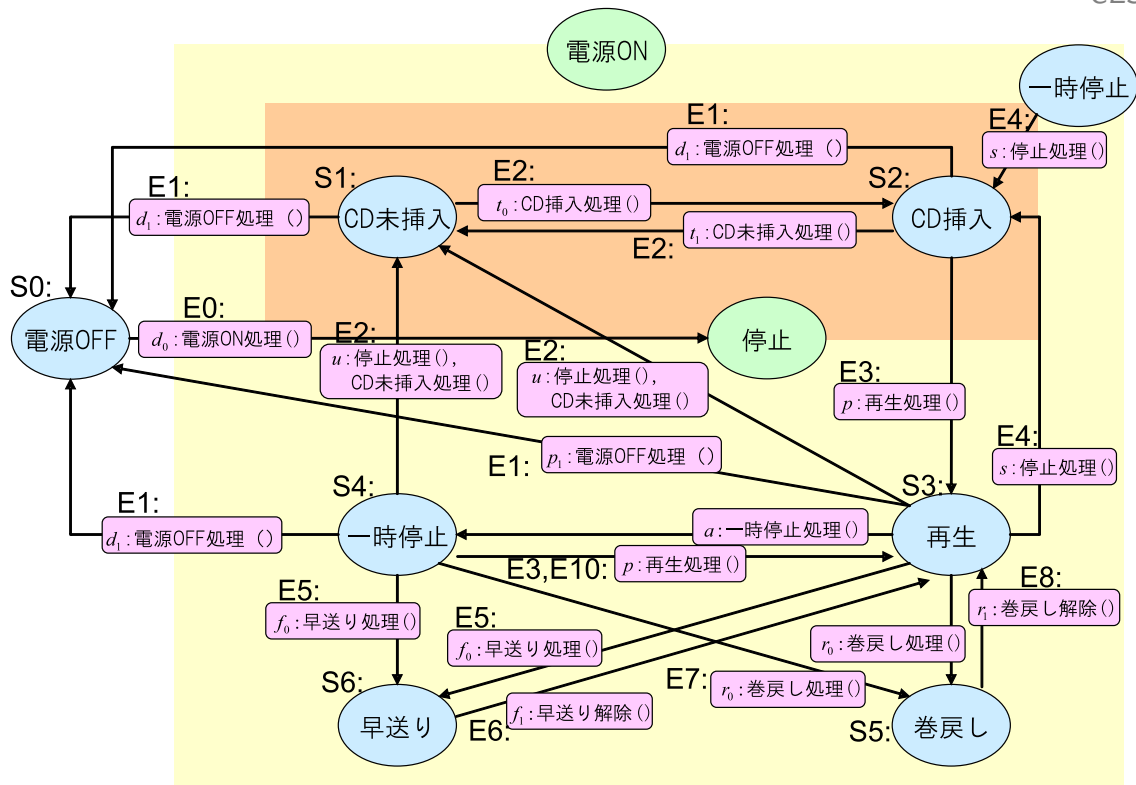
このように事象と状態がリストされたら、それらを上記「再生」の例のような流れを状態遷移図に描く事ができる (図 1(a)) .

状態遷移のテストにおいては、状態遷移図だけを眺めてのテスト構成は甚だ不十分である . 図 1(b) は、図

1(a) を状態遷移表で描いたものである . 表 1 の状態遷移に対し、図 1(b) では、各列の状態へ左カラムのイベントが生じたら、どんな処理を行い次にどの状態に遷移するかが、各セルに規定されている . その処理のリストは次の通り :

処理	
• d ₀ : 電源 ON 処理 ()	• d ₁ : 電源 OFF 処理 ()
• t ₀ : CD 挿入処理 ()	• t ₁ : CD 排出処理 ()
• u: 停止処理 () +CD 排出処理 ()	• p: 再生処理 ()
• s: 停止処理 ()	• a: 一時停止処理 ()
• f ₀ : 早送り処理 ()	• f ₁ : 早送り解除
• r ₀ : 巻戻し処理 ()	• r ₁ : 巻戻し解除 ()

状態遷移は図で描くと状態遷移が分かり易く表示できるのに対し、表で描くとリストされたイベントと状態から全ての状態遷移組合せを洗い出す事ができ、想定していなかったパターンの把握等を含む漏れ抜けのない設計が可能になるという大きなメリットがある .



(a)

状態 イベント	S0: 電源OFF	電源ON					
		停止		S3: 再生	S4: 一時停止	S5: 早送り	S6: 巻戻し
		S1: CD 未挿入	S2: CD 挿入				
E0: 電源ボタン	S1またはS2 ※ d ₀ : 電源ON処理()	S0 d ₁ : 電源OFF処理()	S0 d ₁ : 電源OFF処理()	S0 d ₁ : 電源OFF処理()	S0 d ₁ : 電源OFF処理()	—	—
E1: トレーボタン	—	S2 t ₀ : CD挿入処理()	S1 t ₁ : CD排出処理()	S1 u: 停止処理(), CD排出処理()	S1 u: 停止処理(), CD排出処理()	—	—
E2: 再生ボタン	—	—	S3 p: 再生処理()	—	S3 p: 再生処理()	—	—
E3: 停止ボタン	—	—	—	S2 s: 停止処理()	S2 s: 停止処理()	—	—
E4: 早送りボタン押下	—	—	—	S5 f ₀ : 早送り処理()	S5 f ₀ : 早送り処理()	—	—
E5: 早送りボタン押上	—	—	—	—	—	S3 f ₁ : 早送り解除()	—
E6: 巻戻しボタン押下	—	—	—	S6 r ₀ : 巻戻し処理()	S6 r ₀ : 巻戻し処理()	—	—
E7: 巻戻しボタン押上	—	—	—	—	—	S3 r ₁ : 巻戻し解除()	—
E8: 一時停止ボタン	—	—	—	S4 a: 一時停止処理()	S3 p: 再生処理()	—	—

※ 過去にアクティブだった子孫状態を全て記憶しており、その該当状態をアクティブにする

(b)

図 1: CD プレーヤ機能仕様: (a) 状態遷移図 (b) 状態遷移表

3 状態遷移表からの遷移パスの抽出

上記の CD プレーヤ機能仕様における状態遷移パスを行列の形に書いたのが、図 2(a) である。左カラムが前状態、上ロウが後状態であり、各セルにある処理により前状態から後状態に遷移する事を表している。例えば、 (E_3, S_4) -要素の「 S_2 / p : 再生処理 ()」は、「一時停止状態 S_4 のときにイベント E_3 の停止ボタンが押されたら p : 停止処理 () を実行して、CD 挿入での停止状態 S_2 に移る」という事になる。

この遷移行列を T とおく。数学で言う行列のように書き出すと

$$(1) \quad T = \begin{bmatrix} 0 & d_0 & d_0 & 0 & 0 & 0 & 0 \\ d_1 & 0 & t_0 & 0 & 0 & 0 & 0 \\ d_1 & t_1 & 0 & p & 0 & 0 & 0 \\ d_1 & u & s & 0 & a & f_0 & r_0 \\ d_1 & u & s & p & 0 & f_0 & r_0 \\ 0 & 0 & 0 & f_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & r_1 & 0 & 0 & 0 \end{bmatrix} .$$

T は、グラフ理論における隣接行列を拡張した概念として、状態遷移のグラフを 1 対 1 に表現している。特に、 n 回のイベント列による状態遷移パスは、 T^n により与えられる。この事は、状態のベクトル (縦ベクトル)

$$(2) \quad S^{(0)} = {}^t[S_0^{(0)} \ S_1^{(0)} \ \dots \ S_8^{(0)}]$$

を取ったとき状態 $S^{(0)}$ への T の作用として書ける事からも分かる。即ち、1 回の状態遷移で新しい状態ベクトル $S^{(1)} = {}^t[S_0^{(1)} \ S_1^{(1)} \ \dots \ S_8^{(1)}]$ が得られたとすると、

$$(3) \quad S^{(1)} = TS^{(0)}$$

である。同様に 2 回の遷移で得られる状態ベクトル $S^{(2)}$ は

$$(4) \quad S^{(2)} = TS^{(1)} = T^2S^{(0)} .$$

図 2(b) の各セルには、各パス (前状態 1 回目操作後の状態 2 回目操作後の状態) のパターンが記入されている。例えば、「 S_3 : 再生」状態から「 S_2 : 停止 / CD 挿入」状態に 2 回の操作で遷移するには、 (S_3, S_2) -

要素を見れば $d_1d_0 + ut_0 + as$ となっており、つまり d_1d_0 、 ut_0 、 as の三通りのパターンがあり得る。

このようにして 39 個の有効なセル / 61 本の有効な状態遷移パスが求められた事になるが、状態遷移のテストとしては、このそれぞれのパスに対し直交表を組んでテストを構成すれば良い。状態遷移表と遷移行列 T からテストケースを生成するまでの流れを以下に記しておく。

テストケース生成までの流れ

1. テストしたい機能仕様を状態遷移の順に沿って書き出す。
2. 「状態」、「イベント」、「処理」をリストする。
3. それらを状態遷移表へ書き出す。
4. 「前状態 後状態」の表の各セルへ、対応する処理を書き出す。これを遷移行列 T とする。
5. T^2 を求める。
6. T^2 の各セル中のパスにつきテストシナリオを考え、因子と水準を設定する。ここでは、テストしたい因子として、各状態に対応した因子がとられる事になる。
7. あとは直交表の割付けの問題として、各テストケースへ因子の水準組み合わせを行っていく。

6. の「 T^2 の各セル中のパスにつき」という点について注意しておく。さきの例で、 T^2 の (S_3, S_2) 要素が $d_1d_0 + ut_0 + as$ と、2 回の操作 (処理) で同じ状態遷移をするパターンが三通りあるという事であった。この三通りは異なるパスであるから、 (S_3, S_2) -要素 1 つについて d_1d_0 、 ut_0 、 as の三通りのパターンをテストする事になる。

なお、ここでは 2 回の状態遷移パスのみを議論しているが、本来は一般に n 回の遷移 ($n \geq 2$) で実現される機能に対し状態遷移テストをしたいところであろう。逆に、1 回の遷移を繰り返す事で n 回の遷移テストに代えられないかという、1 回毎のテストの簡単化への期待を持つ事もあるかも知れない。しかし、[2] に

より、結論は「1回では不十分、3回以上は難しい」という事が指摘されている。

1回では不十分なのは、 $S^{(k)} \rightarrow S^{(k+1)}$, $k = 0, 1, \dots$ の1回毎の遷移をテストするだけでは、表面上は状態遷移に問題無いように見えても、そこで状態遷移のために行った処理により、例えば内部の変数領域を壊してしまうといった事が起こり得るからである。

また、3回以上が難しいのは、 T^n から求められるテストすべき遷移パスの数が急激に大きくなる（元の T が十分疎な行列でない限り）という事である。実際、 T が $K \times K$ 行列でもし全ての要素が有効（非零）とすると、 T^2 の全要素中のパス数の和の上限は K^3 本、 T^3 だと K^4 本、 \dots という具合である（一般に T^n ならパス数の和の上限は K^{n+1} 本）。

このため、2回毎の遷移パステストを逐次的に繰り返してパス中の遷移の過程を網羅する事を行う。例えば、上記の「 S_3 : 再生」状態から「 S_2 : 停止 / CD 挿入」状態への遷移を3回の操作で実現するパターンのテストを考えると、 T^3 の (S_3, S_2) -要素が $d_1(d_0t_1) + t_1(d_1d_0 + t_0t_1) + p(d_1d_0 + st_1 + au)$ となっており、 $d_1d_0t_1S^{(0)}$ のテスト

$$(5) \quad t_1S^{(0)} \rightarrow d_0t_1S^{(0)} \rightarrow d_1d_0t_1S^{(0)}$$

を

$$(6) \quad \begin{aligned} &t_1S^{(0)} \rightarrow d_0t_1S^{(0)} \quad \text{および} \\ &d_0t_1S^{(0)} \rightarrow d_1d_0t_1S^{(0)} \end{aligned}$$

の2つのテストにて代える。

4 抽出された遷移パスのテストケースを直交表で組む

前節で抽出された遷移パスに対して、2回の状態遷移のテストケース作成を直交表を用いて行う。ここでは例として、図2(b)の(3,2)-要素中に現れるパス“pa”、

$$(7) \quad \begin{aligned} pa S_3 &= (S_3 : \text{再生}) \rightarrow [a : \text{一時停止処理} ()] \\ &\rightarrow (S_4 : \text{一時停止}) \\ &\rightarrow [p : \text{再生処理} ()] \rightarrow (S_3 : \text{再生}) \end{aligned}$$

について考える事にする。

いま、テストしたい事として、「一時停止」、「再生」を本体パネルにより操作するかあるいはリモコンにより操作するか、という事とメディアは CD, SACD のいずれでも OK かという事を考える。

簡単な例題ではあるが整理すると、

- やる事：式(7)の動作が問題なくできているかをテストする事
- 考える事：最終状態以外の各状態それぞれでのテストしたい条件を挙げる（状態の時系列に対応した因子）:

{	* 一時停止 : 本体パネル / リモコンのいずれか
	* 再生 : 本体パネル / リモコンのいずれか
	* メディア : CD / SACD のいずれか

と切替えているいろいろな組み合わせ方をしても OK か? をテストしよう。デフォルトのテストケースを『本体パネルボタンにて一時停止』『本体パネルボタンにて再生』、メディアは CD」として。

- そのいろいろな組み合わせ方を網羅するような直交表を作成しよう。

といった状況を想定する。

これらを因子 / 水準としてリストしたものを表2(a)に示す。また、表2(b)はテストケースを L_4 へ割り付けたものである。この場合、3因子の水準組み合わせは $2 \times 2 \times 2 = 8$ 通りであるが、前稿で述べたように表2(b)に掲示された4通りのテストケースを実施すれば、2機能間組み合わせとしては4通りの全てが網羅され、また3機能間組み合わせについても8通り中7通りが網羅された事になる（複数機能間組み合わせテストの前段階で単機能テストがなされている事が前提であり、それは本デフォルトケースの一因子だけ変えたテストケースに相当する）。

こうして直交表によりテストケースがリストされたら、No. 1 2 3 4 と順次因子の水準を変化させながらテストを実施していけば良い。

もし本当にバグがあった場合に、表2によるテストで次のような事が分かる。その前に、単機能テスト

前状態 \ 後状態		電源ON						
		S0: 電源OFF	停止		S3: 再生	S4: 一時 停止	S5: 早送り	S6: 巻戻し
			S1: CD未挿入	S2: CD挿入				
S0: 電源OFF		—	d_0	d_0	—	—	—	—
電源ON	停止	S1: CD未挿入	d_1	—	t_0	—	—	—
		S2: CD挿入	d_1	t_1	—	p	—	—
	S3: 再生	d_1	u	s	—	a	f_0	r_0
	S4: 一時停止	d_1	u	s	p	—	f_0	r_0
	S5: 早送り	—	—	—	f_1	—	—	—
	S6: 巻戻し	—	—	—	r_1	—	—	—

(a)

前状態 \ 後状態		電源ON						
		S0: 電源OFF	停止		S3: 再生	S4: 一時 停止	S5: 早送り	S6: 巻戻し
			S1: CD未挿入	S2: CD挿入				
S0: 電源OFF		d_0d_1	d_0t_1	d_0t_0	d_0p	—	—	—
電源ON	停止	S1: CD未挿入	t_0d_1	$d_0d_1 + t_0t_1$	d_1d_0	t_0p	—	—
		S2: CD挿入	$t_1d_1 + pd_1d_0$	$d_1d_0 + pu$	$d_1d_0 + t_1t_0 + ps$	—	pa	pf_0
	S3: 再生	$ud_1 + sd_1 + ad_1$	$d_1d_0 + st_1 + au$	$d_1d_0 + ut_0 + as$	$sp + ap + f_0f_1 + r_0r_1$	—	af_0	ar_0
	S4: 一時停止	$ud_1 + sd_1 + pd_1$	$d_1d_0 + st_1 + pu$	$d_1d_0 + ut_0 + ps$	$sp + f f + r r$	pa	pf_0	pr_0
	S5: 早送り	f_1d_1	f_1u	f_1s	—	f_1a	f_1f_0	f_1r_0
	S6: 巻戻し	r_1d_1	r_1u	r_1s	—	r_1a	r_1f_0	r_1r_0

(b)

図 2: 状態遷移を表す行列 : (a) T , 1 回の操作による遷移 (b) T^2 , 2 回の操作による遷移

因子	水準 1	水準 2
一時停止	本体パネル	リモコン
再生	本体パネル	リモコン
メディア	CD	SACD

(a)

Test No.	因子 A	因子 B	因子 C
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	0

(b)

Test No.	一時停止	再生	メディア
1	本体パネル	本体パネル	CD
2	本体パネル	リモコン	SACD
3	リモコン	本体パネル	SACD
4	リモコン	リモコン	CD

(c)

表 2: “ap”(「再生 一時停止」)の遷移テスト:(a) 組み合わせテストしたい因子とその水準 (b) 一般の直交表 L_4 (c) L_4 へのテストケースの割付け

で実施されたテストケースを表 3 にリストしておく。Test No. は便宜上 No. 5~8 としてある。既に単機能テストの段階で、No. 1, 5, 6, 7 のテストケースは動作確認がなされている。

そこで、例えば No. 2 のテストケースで動作に問題があった場合、組み込みソフトウェアの、リモコンからの SACD 再生指示の部分にバグがあるものと考えられる。それは No. 5 の正常動作確認により SACD の再生機構自体に問題は無く、また、No. 6 の正常動作確認によりリモコンからの CD 再生指示には問題無いからである。このように、バグ要因の絞り込みができる。

以上のように、各パスでの状態遷移テストを行っていく事ができる。

Test No.	一時停止	再生	メディア
5	本体パネル	本体パネル	SACD
6	本体パネル	リモコン	CD
7	リモコン	本体パネル	CD
8	リモコン	リモコン	SACD

表 3: 表 2 に現れていない組み合わせ

5 結論

本稿では、前稿で紹介したソフトウェアテストのテストケースの効率的な網羅法の 1 つである、直交表を適用して、組み込みソフトウェアにおいて重要となる状態遷移のテストを行う方法について CD プレーヤの機能仕様の設計を例として説明した。考え方はいたってシンプルであり、状態遷移を表す作用行列 T に対し、 T^2 から求められる各遷移パスにつき、直交表によりテストを組めば良い。 T のサイズが大きいつきは遷移パスの数が大きくなるという事はあるものの、そのシンプルさ故に適用し易さは高く、また直交表のテストケース網羅の効率性の高さが活かせる。現状で組み込みソフトウェアの状態遷移テストに大変有効な手法の 1 つとして推奨されよう。

6 今後の課題

上述のように、本手法は、作用行列 K のサイズが大きくなるとテストすべき遷移パス数が大きくなるという問題がある。今後は、この問題への対策として、テストすべき遷移パスの自動抽出や、各遷移パスの直交表への割付けの自動化 / 支援システム構築といった事が考えられるであろう。これからの研究の発展に期待したい。

7 むすび

読者諸兄には、是非直交表によるソフトウェアテスト法、とりわけ状態遷移テスト法をお試しいただければ幸甚である。

参考文献

- [1] 弊社製品 ZIPC Ver. 9.2 インストールフォルダ内ドキュメント: ZIPC チュートリアル /ZIPC V9/Guide/ZIPCTutorial.pdf
- [2] 吉澤、秋山、仙石 . ソフトウェアテスト Hayst 法入門、日科技連出版、2007.
- [3] リー・コーブランド、宗 訳 . はじめて学ぶソフトウェアのテスト技法、日経 BP 社、2005.
- [4] ボーリス・バイザー、小野間・山浦 訳 . ソフトウェアテスト技法 自動化・品質保証、そしてバグの未然防止のために、日経 BP 出版センター、1994.
- [5] 秋山 . 直交表を活用したソフトウェアテストの効率化 HAYST 法の活用 、
<http://www.swtest.jp/jasst05w/S4-1.pdf>,
 JaSST'05, 2005.
- [6] 山本、秋山 . 直交表を利用したソフトウェアテスト HAYST 法 、 JaSST'04, 2004.
- [7] 高橋 他 . 組み合わせテストを用いたソフトウェアテストとその限界、
http://www.juse.or.jp/software/pdf/22_spc/5/5_2_1_report.pdf
- [8] 田口、小西 . 直交表による実験のわりつけ方 例題と演習、日科技連出版、1959.
- [9] 渡辺 . 組み込みソフトウェア開発課題への挑戦～網羅度～、キャッツ株式会社 CESL 資料、2009.
- [10] A. S. Hedayat, N. J. A. Sloane and J. Stufien, *Orthogonal Arrays: Theory and Applications*, Springer, 1999.
- [11] D. R. Kuhn and D. R. Wallace, A. M. Gallo. "Software Fault Interactions and Implications for Software Testing", IEEE Trans. Software Eng., VOL. 30, pp.418-421, 2004.



川崎 秀二

1991 年 電通大・電気通信・通信工 卒 . 1993 年慶大・理工・電気 修士 . 1996 年慶大・理工・数理 単位取得退学 . 博士 (工学) . 日本数学会 会員 .

現在、キャッツ株式会社 CESL (組み込みソフトウェア研究所) 研究員 .