

形式分類木手法に向けて

北村 崇 師[†] Ling Fang[†]
矢田 部 俊 介[†] 大 崎 人 士[†]

本稿では、分類木手法の形式文法・意味論を与え、分類木手法の形式的な側面からの確立を試みる。
さらにその議論を基に SAT を用いたテストケースの自動生成について触れる。

Toward a Formal Classification Tree Method

TAKASHI KITAMURA^{,†} LING FANG^{,†} SHUNSUKE YATABE[†]
and HITOSHI OHSAKI[†]

This work presents a formal semantics for Classification Tree Method (CTM), and discusses
an automatic technique of test-case generation for CTM using SAT based on it.

1. 導 入

分類木手法 (Classification Tree Method)¹⁾ は、木構造を用いて系統的にテスト関心事を分析し、テストケースを設計するテスト技法である。図 1 は分類木による、実時間 OS である OSEK/VDX³⁾ の API 関数 `activateTask` (suspend 状態のタスクを起動する関数) の機能テスト分析の見本である。この分類木分析では、まずその関数のテスト関心事を「呼出しタスク」と「対象タスク」の状況に分けて分析している。「呼出しタスク」の状況としては「カテゴリ」と「優先度」の二つの観点に分け、さらに「カテゴリ」は「タスク / 割込み」に「優先度」は「高 / 低」に分けて考慮する。枝の種類は、枝先が丸い -and 分解と、枝の元が弧で括られた -xor 分解の二種類があり、それぞれ直交的観点か排他的観点かで使い分ける。また「割込み」から「優先度」に延びる `excludes` の矢印は、呼出しタスクが割込みの場合は優先度の観点は考慮するのに意味がないことを明示的に示す。対象タスクに関して同様に分析を行うと、図 1 中の上部の木構造が得られる。下部の表は、上の分類木分析によるテスト関心事を網羅するためのテストケースを列挙している。表中の横線が一つのテストケースに相当し、よって、図 1 では、この分類木から 12 個のテストケースが得られることを示す。分類木の葉から延びる縦線と各テストケース (横線) の交差上の黒丸は、各テストケース

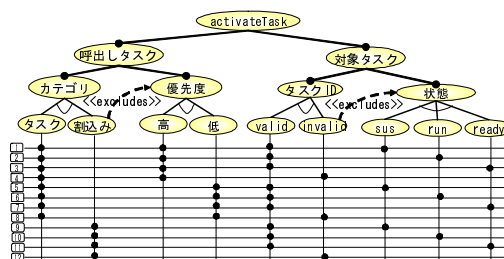


図 1 分類木による `activateTask` 機能テスト分析の例

が考慮するテスト関心事を示す。例えばテストケース 1 は「高優先度のタスクから、妥当なタスク ID で suspend 状態のタスクに対し `activateTask` 関数を呼出した場合」を確認する。各テストケースは、おおまかに (正確な説明は本稿の主要な議論であり、次節で述べる)、-and 分解要素は全て、-xor 分解要素はそのうちの一つを考慮し、テストケースの集合 (テストスイート) はそうして考慮される全ての組合せを確認するよう設計される。また、制約記述は、考慮すべきテストケースに制約を与える。例えば、テストケース 8-12 は、「優先度」の関心事を考慮しないが、これは上述の分類木中 `excludes` 制約によるものである。

分類木手法はテスト分析・設計に有効なアプローチであるが、その問題の一つは形式的な議論が欠けていることである。これにより、分類木の理解が読み手によって異なるといった問題が生じる。実際に²⁾では、`excludes` 記述の対象は葉に限るという不自然な制限を設けているが、この制限が一般的なものか不明である。また、形式議論は、分類木からテストケースの計

[†] 産業技術総合研究所・組込みシステム技術連携研究体

CT	$::=$	$Tree ; Consts$
$Tree$	$::=$	$n = \mathbf{and}(Tree, \dots, Tree)$
	$ $	$n = \mathbf{xor}(Tree, \dots, Tree)$
	$ $	$leaf$
$Consts$	$::=$	$Const \mid Const ; Consts$
$Const$	$::=$	$n \mathbf{requires} n \mid n \mathbf{excludes} n$

表 1 分類木の文法

算方法やその効率性, 妥当性の議論にも不可欠である。

本稿では, 分類手法の形式文法・意味, また分類木手法におけるテストケースやテストスイートについて形式的定義を与えることで, 分類木手法の形式的な側面からの確立を試みる。さらにその議論を基に SAT を用いたテストケースの自動生成について触れる。

2. 形式分類木

分類木の設計と形式文法 まず分類木の設計を定める。本稿で扱う分類木は, その木構造は and-xor 木とし, 制約記述は requires, excludes を持ち, それらの引数は, 節 (葉を含む) を対象とする。

定義 1 (分類木手法の形式文法) 分類木の節の集合を N とし, その要素を $node, nn, n$, と書く。また, 分類木の節集合の部分集合である葉の集合を $L(\subseteq N)$ とし, その要素を $leaf, ln, l$ で表す。分類木の文法は表 1 の BNF 表現によって定義される。□

分類木 (CT) は, その基本構造である木構造 ($Tree$) と木構造中の横断的な葉・節間の制約を表現する制約記述 ($Consts$) から成る。分類木の根 $r \in N$ は, その分類木分析のテスト対象である。各節は各々 $-and$, $-xor$ 関数記号で表現される $-and$ 分解か $-xor$ 分解, もしくは葉である。木構造中の横断的な制約記述 $Consts$ は, 個々の制約式 $Const$ (requires / excludes 関数) の列である。

分類木の形式意味 分類木の意味は「分類木から得られるテストケースの集合」により定義される。本稿ではこれを正確に命題論理を用いて述べる。つまり, 分類木を命題論理式で表現し, 次にそうした命題論理式に関し妥当なモデルを定義, さらにそれを基に分類木から得られるテストケースの集合を定義する。

定義 2 分類木の命題論理式の表現は, 表 2 に示す分類木から命題論理式への変換関数 ($\llbracket \cdot \rrbracket$) により得られる。但し, $root(Tree)$ は引数に渡された木の根を返す関数を, \oplus は排他的論理和を表す。□

定義 3 (モデルと妥当なモデル) (1) 分類木 $ct \in$

$\llbracket ct \rrbracket = \mathbf{case} \ ct \ \mathbf{of}$	
$n = \mathbf{and}(t_1, \dots, t_k)$	$\Rightarrow n \Leftrightarrow \bigwedge_k root(t_k) \bigwedge_k \llbracket t_k \rrbracket$
$n = \mathbf{xor}(t_1, \dots, t_k)$	$\Rightarrow n \Leftrightarrow \bigoplus_k root(t_k) \bigwedge_k \llbracket t_k \rrbracket$
$leaf$	$\Rightarrow \mathbf{T}$
$n_1 \mathbf{requires} n_2$	$\Rightarrow n_1 \Rightarrow n_2$
$n_1 \mathbf{excludes} n_2$	$\Rightarrow \neg(n_1 \wedge n_2)$
$ct_1 ; ct_2$	$\Rightarrow \llbracket ct_1 \rrbracket \wedge \llbracket ct_2 \rrbracket$

表 2 分類木から命題論理式への変換関数

CT) の命題論理式 $\llbracket ct \rrbracket$ のモデル $m(\in M)$ とは, その分類木の節集合の部分集合である: $M = \mathcal{P}N$. (2) 次の条件 (a)(b) を満たす時, モデル m は分類木 ct に関し妥当であると言い, $m \models ct$ と書く: (a) モデル m が $\llbracket ct \rrbracket$ を満たす: $m \models \llbracket ct \rrbracket$; (b) 根 (テスト対象) r がモデル m に含まれる: $r \in m$. □

定義 4 (テストケースとテストスイート) (1) テストケース (c) とは, 葉集合 L の部分集合である: $c \in \mathcal{P}L$. (2) モデル m のテストケースとは $m \cap L$ であり, m° と書く。(3) 分類木 ct から得られるテストスイート (s) とは, ct に関し妥当なモデルのテストケースの集合で, $\llbracket ct \rrbracket$ と書く: $\llbracket ct \rrbracket = \{m^\circ \mid m \models ct\}$. □

SAT によるテストケース生成へ向けて 以上では, 分類木手法におけるテストケースを, 命題論理とその意味論 (モデル) を通じて議論した。これは, 命題論理式のモデルの発見がテストケース作成に通ずることを意味する。こうした命題論理式のモデルの発見は SAT により自動化可能で, つまり, テストケースが自動生成できることになる。現在この考えに基づき, SAT によるテストケース自動生成に取り組んでいる。

3. 結論と今後の課題

本稿では分類木手法の確立するため, 分類木手法について形式文法・意味論を与えることでその形式的な基礎付けを行った。本手法は, 形式手法とテストティングの両方の特徴をもつ検証技法である。本研究は実時間 OS の検証技法の研究開発の一環として行われたもので, 現在は本手法の実証実験を行っている。

謝辞 本研究を遂行するにあたり, 多くの有益なコメントを頂いた 東芝セミコンダクター社の古森誠司氏・渡邊竜明氏と, Witz の片岡歩氏に感謝する。

参 考 文 献

- 1) Lehmann, E.; Wegener, J., Test Case Design by Means of the CTE XL. EuroSTAR 2000
- 2) A. Schürr, S. Oster and F. Markert, Model-Driven Software Product Line Testing: An Integrated Approach, SOFSEM 2010
- 3) OSEK/VDX Operating System Specification 2.2.3, 2005

木構造の厳密な条件 (例えば, 分岐した枝は合流しない, ループ構造を許さない, など) の記述はスペースの関係上省略する。