

静的解析による UML ステートマシン図答案の 誤り特定自動化手法の提案

五島 光祥^{1,a)} 小形 真平^{1,b)} 槇原 絵里奈^{2,c)} 岡野 浩三^{1,d)}

概要：UML (Unified Modeling Language) モデリングの多人数教育では、得られる答案モデルが多数かつ多様になりやすいため、教育者が答案を手動で評価することに時間がかかってしまう。教育者が答案評価を通じてモデルの誤りについて学習者へフィードバックしようとするとき、答案の誤りを正確に把握することが必須である。その支援のために我々はこれまで、正答例に基づき答案の正誤を自動判定する手法を提案してきたが、誤りの箇所や種類の自動特定には至っていない。そこで本稿では、正答例と答案の記述を比較するよう静的解析を行うことで、ステートマシン図の誤り箇所と種類を自動で特定する手法を提案する。提案手法を 57 個の学習者の答案 (ステートマシン図) に適用した結果、状態、遷移、実行活動、開始疑似状態、終了状態で構成される答案において正しい特定結果が得られたため、提案手法が有用である見込みを得た。

An Automated Method of Identifying Errors in Learner-Created UML State Machine Diagrams with Static Analysis

Abstract: In teaching UML (Unified Modeling Language) state machine diagrams for many learners, the educators spend much time to evaluate learner-created models since those models are often many and varied. When educators want to make educational feedback on model errors, they firstly should grasp errors in learner-created models accurately. For this purpose, we have proposed a method to support the correctness of answer behavior using example answers. To this end, we have proposed a method to automatically judge the correctness of learner-created models based on the sample answer model. However, this method still cannot identify locations and types of the errors in the incorrect models. In this paper, we propose a method to automatically identify the location and type of errors in state machine diagrams by statically analyzing the differences between learner-created models and the sample answer model. The proposed method was applied to 57 learner-created models (state machine diagrams), and the results showed that the proposed method is likely to be useful, because the answers, which did not contain expressions with variables, gave correct identification results.

1. はじめに

UML のステートマシン図 [1] は、離散的なイベント駆動の振る舞いを表す図であり、組込みシステムや、多種多様なセンサ等デバイスを備える CPS (Cyber-Physical

System), IoT (Internet of Things) といった、様々なシステム設計に使用されている [2][3]。また、【技術・家庭編】中学校学習指導要領 (平成 29 年) では、UML を用いた教育が行われる可能性が示唆されており [4]、実際に中学校でステートマシン図を教育に用いる試みがなされている [5]。そのため、モデリング教育の重要性はますます増しており、ステートマシン図は積極的に学ぶべきモデル図の一つになっている。

モデリング教育では、“課題文などの要求から必要な情報を読み取る要求分析能力” [6] の向上が求められるため、学習者に要求文を与えてモデルを記述させる課題学習を行うことがある。そこに我々はこれまで、記法学習に集中しや

¹ 信州大学工学部 〒380-8553 長野県長野市若里 4-17-1
Faculty of Engineering, Shinshu University 4-17-1
Wakasato, Nagano-shi, Nagano, 380-8553 Japan

² 同志社大学 〒610-0394 京都府京田辺市多々羅都谷 1-3
Doshisha University 1-3 Tataramiyakodani, Kyotanabe-shi, Kyoto, 610-0394 Japan

a) 21w2031a@shinshu-u.ac.jp

b) ogata@cs.shinshu-u.ac.jp

c) emakihar@mail.doshisha.ac.jp

d) okano@cs.shinshu-u.ac.jp

すいように用語集を設けた課題学習とその支援を実践してきた [7]。しかし、それでもなお、多人数教育で得られる多数の答案モデルでは形や図要素の配置が多様であるため、教育者の手動による答案評価は依然として時間がかかる作業であり、結果として学習者へのフィードバックの遅延が生じてしまう。教育者がどのようなフィードバックを学習者へ与えるべきかを判断するためには、各答案について誤りがあれば、その箇所や種類を把握する必要がある、その作業の効率化が望まれる。

先行研究では、ステートマシン図に対し静的解析と動的解析を行うことで誤り箇所を特定する手法が提案されている [8]。このような解析はステートマシン図の変数やガード条件、全体の振る舞い等の正誤を判定することができる。しかし、教育に用いることを想定した場合、さらに詳細な誤りの原因や解決策をフィードバックすることが望まれる。

また、我々の先行研究では、モデリング学習支援のために、ステートマシン図の UML 記法に従っていない誤り箇所を自動で特定する記法チェッカ [9] というツールや、状態遷移テスト [10] の理論を用いることで、入力として与えた複数のステートマシン図の記述の多様性に依存せずに、振る舞いの等価性の観点で図を分類できる SML4C [7] というツールを提案してきた。このような従来手法は、用語集を用いたステートマシン図教育への適用が想定されているが、具体的な誤りの箇所や種類を特定する手法の研究が十分でない。

そこで、本研究では教育者によるステートマシン図評価の効率化を目的とし、正答例と学習者答案の記述を比較するために静的解析を行い、答案のさらに詳細な誤り箇所と種類を自動で特定する手法を提案する。提案手法の実現アイデアとして、まず正答例と答案のステートマシン図間で用語集を手がかりに図要素の対応付けを行う。そして、対応付けの結果に基づき正答例と答案間の差分を解析することで誤り箇所の特定を行う。

提案手法の有効性を評価するために、学習者 19 名が 3 つの記述課題に対して作成した答案計 57 個に提案手法を適用し、誤り箇所と種類を正しく特定できるか検証する実験を行った。答案の内訳は、正しい答案 42 個と誤りを含む答案 15 個であった。提案手法によって出力した誤り箇所が、実験前に課題作成者が手動で評価した結果と一致するか比較を行った。その結果、状態、遷移、実行活動、開始疑似状態、終了状態で構成されるステートマシン図において、全ての答案で正しい特定結果が得られたため、提案手法が有用である見込みを得た。

なお、これまでからの継続的な研究であるために、提案手法を適用した答案の一部は先行研究 [7] で用いたものを再利用しているが、適用対象を拡大してもいる。また、本稿では、我々の先行研究 [11] の手法に対し、図要素間の対応付け方法を改良して発展させたものであり、新規の部分

を含む。

本稿は以下のように構成されている。2 章ではステートマシン図の詳細について示す。3 章では提案手法の詳細について示す。4 章では本研究の実験結果及び考察を示す。5 章では関連研究について示す。最後に、6 章ではまとめと今後の課題について示す。

2. ステートマシン図

ステートマシン図とは、OMG (Object Management Group) [12] が標準化している UML 図の一つで、システムの離散的なイベント駆動の振る舞いを表す有限オートマトン等の状態機械について、図で表現したものである [1]。ステートマシン図は、図 1 に示すように状態を四角形、遷移を矢印で表記し、各状態を頂点として状態間を遷移で繋げることで状態の変化を表現する。UML の仕様書 [1] をもとに、ステートマシン図を構成する図要素の説明と、それらの図 1 との対応関係を表 1 に示す。ガードやエフェクトといった要素中では変数を用いた式を記述でき、変数の値に基づいて状態遷移を制御することもできる [1]。本研究で入力として用いる正答例と答案のステートマシン図は、UML の仕様書に適合するものとする。

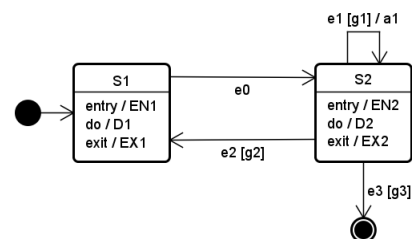


図 1 ステートマシン図例

Fig. 1 State Machine Diagram Example.

3. 提案手法

本章では、静的解析により誤りの箇所と種類を自動特定する手法を提案する。特に、「特定する誤りの種類」、「誤り箇所と誤りの種類を自動特定する手順」を中心に説明する。

3.1 概要

提案手法は、特にステートマシン図教育における学習者の記法学習に用いることを想定している。そして、記法を学ぶ早期段階を考慮して「状態、遷移、実行活動、開始疑似状態、終了状態」で構成された単純なステートマシン図を入力の対象とする。

本研究では先行研究 [7] と同様に、用語集を設けた記述課題の正答例と答案を提案手法への入力にとることを前提とする。言い換えれば、記述課題を行う学習者には、状態の動作を表す「実行活動」と、遷移の契機を表す「トリガー

表 1 ステートマシン図を構成する要素
Table 1 Elements of State Machine Diagrams.

要素名	図 1 との対応関係	説明
開始疑似状態	●	振る舞いの開始を表す疑似状態.
終了状態	●	振る舞いの終了を表す状態.
状態		ステートマシン図の実行において, ある不変的な条件が成立する状況をモデル化したもの. 四角形で表記され, 状態名, 入場動作, 実行活動, 退場動作で構成される.
状態名	S1,S2	状態の名前.
入場動作	EN1,EN2	前の状態から現在の状態へ入場した直後に実行される動作. entry と呼ばれる.
退場動作	EX1,EX2	現在の状態から次の状態へ退場する直前に実行される動作. exit と呼ばれる.
実行活動	D1,D2	入場動作が完了してから次の状態への遷移によって中断されるまでの間, 継続的に実行される動作. do と呼ばれる.
遷移	————→	状態から次の状態への変化を表す. 矢印で表記され, トリガー, ガード, エフェクトで構成される.
トリガー	e0,e1,e2,e3	遷移の契機となるイベント.
ガード	g1,g2,g3	遷移することが可能となる条件.
エフェクト	a1	状態から次の状態への遷移中に行われる処理.

名」に使用すべき用語を最低限として定めた用語集が提供されているものとする. これは開発現場において, 共同開発者間で用語集を設けることが一般的であり, 用語集の制約を学習に設けることで, 自由記述に比べて仕様書の要求通りに振る舞うステートマシン図を作成する練習となるためである. また, 過去の調査 [13] では, 用語集を設けていないステートマシン図学習において, 学習者が要求文の文脈情報を記憶し各要素に名称をつける作業は, 時間がかかりミスが生じやすいことが分かっている. このような調査の結果から, 正しい記法を学ぶことに集中するために用語集を用いている.

これらの入力をもとに誤り箇所と種類を特定する手順について以下に概説し, 図 2 に提案手法の全体像を示す. なお, 特定の対象とする誤りの詳細は 3.2 節で述べる.

- (1) 正答例と答案の記述を比較するために, それらの状態間や遷移間に対応する箇所を用語集に基づき対応付ける. 詳細は 3.3.1 項で述べる.
- (2) 対応付けの結果から差分を特定し, 誤り箇所と種類を特定する. 詳細は 3.3.2 項で述べる.

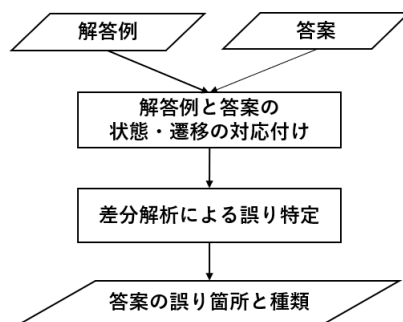


図 2 提案手法のフローチャート
Fig. 2 Flowchart of Proposed Method.

3.2 ステートマシン図の誤りの種類

提案手法において, 状態, 遷移, 実行活動, トリガー名, 開始疑似状態, 終了状態の観点で比較した際に特定すべき誤りの種類を表 2 に示す. 過去の調査 [14] によると, ステートマシン図の記述では, 状態遷移の過不足, 振る舞いの誤り, 開始疑似状態と終了状態の記述誤りに関する誤りが多いとの報告がある. そのため本研究では, 調査結果に関する誤りを詳細に分類して誤りの種類を定義し, 自動特定の対象とする.

まず状態遷移の過不足の詳細な誤り箇所と種類を特定するために, 「状態の不足」, 「状態の過剰」, 「遷移の不足」, 「遷移の過剰」として, より詳細な誤りの種類を定義した. これら 4 つの誤りは, 答案に不足, または過剰な状態遷移があることを指す. 次に, 状態遷移の過不足以外に振る舞いが誤りとなる要因として, 遷移を繋げる箇所の誤りが考えられる. そこで, 振る舞いに関する詳細な誤り箇所と種類を特定するために, 遷移を繋げる箇所の誤りを遷移の観点では「接続遷移の過剰」, 「接続遷移の不足」, 状態の観点では「遷移先誤り」, 「遷移元誤り」としてそれぞれ 2 つずつ定義した. 最後に, 開始疑似状態と終了状態の記述忘れは「状態の不足」, 「状態の過剰」とし, 開始疑似状態と終了状態を繋げる状態の誤りは「遷移先誤り」と「遷移元誤り」として上述の誤りの種類に含めて扱う.

表 2 の中には, 一つの誤りから複数項目出力される誤りの種類があり, 「遷移の不足と接続遷移の不足」, 「遷移の過剰と接続遷移の過剰」, 「接続遷移の不足と遷移先誤りまたは遷移元誤り」, 「接続遷移の過剰と遷移先誤りまたは遷移元誤り」の組み合わせが挙げられる. 一例として「遷移の不足と接続遷移の不足」を挙げると, ある状態 A から状態 B への遷移 e1 が答案に不足していた場合, 誤りの種類は「遷移の不足」, 「接続遷移の不足」と判定される. この場

合、「遷移の不足」は答案に遷移 e1 が不足していることを表し、「接続遷移の不足」は状態 A、状態 B に遷移が不足していることを表している。同一の誤り箇所から出力された二つの出力結果を最終的に組み合わせることで、「状態 A から状態 B への遷移 e1 が答案に不足している」といった、一つにまとめた形で出力を行う。

ステートマシン図には、状態、遷移、実行活動、開始疑似状態、終了状態以外に、入場動作や退場動作、ガード、エフェクト等の要素が存在する。今回特定の対象としていない要素に関する誤りの種類を整理することは今後の課題となる。表 2 の各種類について誤り箇所を特定する方針は 3.3 節で説明を行う。

表 2 ステートマシン図における誤りの種類

Table 2 Types of Errors in State Machine Diagrams

誤りの種類	説明
状態の不足	正答例にある状態が答案にない。
状態の過剰	答案にのみ含まれる状態がある。
遷移の不足	正答例にある遷移が答案にない。
遷移の過剰	答案にのみ含まれる遷移がある。
接続遷移の不足	状態の入退場遷移数が不足している。
接続遷移の過剰	状態の入退場遷移数が過剰である。
遷移先誤り	遷移を接続する遷移先の状態が誤り。
遷移元誤り	遷移を接続する遷移元の状態が誤り。

3.3 誤り特定方法

本節では、誤り箇所の特定、及び誤りの分類を行う手法の手順について説明を行う。以降では、誤り箇所の特定と誤りの種類の特定を総称して、誤り特定と呼ぶ。

3.3.1 手順 1. 正答例と答案の状態・遷移の対応付け

本項では、誤り箇所を特定するための前処理として、正答例と答案の図要素間を対応付ける手法について図 3 を正答例、図 4 を答案例として説明する。

答案の誤り箇所を見つけるためには、正答例と異なる部分を特定する必要がある。そのため、正答例と答案間の差分を解析できるよう、それらの状態間や遷移間（以下、簡単に状態遷移間と呼ぶ）を対応付ける必要がある。しかし、例えば状態名「電源 ON」と「電源入」が同じ意味を表すように、ステートマシン図の各要素の名称の付け方はモデル図の作成者ごとに異なることが多いため、各要素の名称の比較から正確に対応付けることはできない。そこで、提案手法では、用語集の存在を前提として、用語の一致により状態遷移間を正確に対応づけることとした。用語集で「実行活動」、「トリガー名」が定められた問題への適用を想定する。このような対応付けにおいて、「状態の対応付け」では「実行活動」、「遷移の対応付け」では「トリガー名」が一致する状態遷移を正答例と答案間で検出し、1 対 1 で対応付ける。図 3、図 4 では、実行活動「do1」、「do2」、トリ

ガー名「e1」、「e2」、「e3」、「e4」が用語集に定められている場合を例として示す。

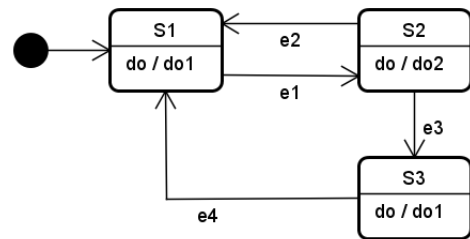


図 3 正答例

Fig. 3 Sample Answer.

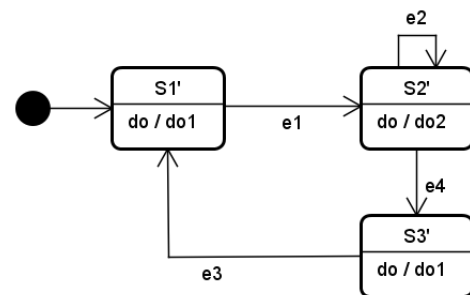


図 4 誤りを含む答案

Fig. 4 Answer Containing Errors.

提案手法では、まず正答例と答案の各図に 1 つしか登場しない「実行活動」または「トリガー名」を持つ状態遷移を対応付ける。例えば、図 3、図 4 の例では、各図に実行活動「do2」、トリガー名「e1」、「e2」、「e3」、「e4」が 1 つずつしか登場しないため、これらの状態遷移の対応付けを行う。

次に、図ごとに同名の「実行活動」または「トリガー名」を持つ状態遷移が複数個登場する場合、次に説明する状態や遷移の特徴について一致度が最も高い状態遷移間を対応付ける。例えば、図 3 の状態 S1 と状態 S3、図 4 の状態 S1' と状態 S3' はそれぞれ共通の実行活動「do1」が含まれている。このような重複する実行活動を持つ状態間では「隣接する状態」、「入場遷移のトリガー名」、「退場遷移のトリガー名」、「状態名」、「開始疑似状態との接続」といった、状態やその周辺の要素を状態の特徴と捉えて、より特徴の近い状態間で対応付ける。

図 3 の状態 S1 と図 4 の状態 S1' を例に、状態の特徴を以下に説明する。なお、開始疑似状態との接続遷移はトリガー名を与えられないため、以降では開始疑似状態の遷移を「開始」と表記する。また、図中にある終了状態は「隣接する状態」として扱う。

- 「隣接する状態」：比較したい 2 つの状態（例：S1 と S1'）それぞれについて、一つの遷移で直接繋いだ隣接する状態（例：S1 に隣接する状態は、S2 または S3）

との実行活動ペア（例：{S1 の do1, S2 の do2}, {S1 の do1, S3 の do1}）を求め、比較したい2つの状態間（例：S1 と S1'）で、隣接する状態との関係と実行活動ペアを比較する。ここで「隣接する状態との関係」を、遷移元の状態（例：S1 から見た S3, または S1' から見た S3'）、遷移先の状態（例：S1' から見た S2'）、遷移元かつ遷移先である状態（例：S1 から見た S2）の3つに分類する。そして、この「関係」と「実行活動ペア」の双方が一致するものをカウントする。なお、S1' に隣接する状態については、遷移先の状態 S2' との実行活動ペア {S1' の do1, S2' の do2}, 遷移元の状態 S3' との実行活動ペア {S1' の do1, S3' の do1} がある。例えば S1 と S1' の比較では、これらを踏まえると、(S1 と S3, S1' と S3') のペアのみが、隣接する状態との関係と実行活動ペアの双方が一致する。このことから S1 と S1' の間では、「do1 を持つ遷移元の状態」の一つだけが一致するため、その個数の1がカウントされる。

- 「入場遷移のトリガー名」：状態へ入力される遷移のトリガー名を重複を排除して抜き出し、比較したい状態間で一致する個数をカウントする。例えば、S1, S1' を比較すると、S1 では「開始」と e2 と e4, S1' では「開始」と e3 が入場遷移となり、「開始」のみが一致するために1がカウントされる。
- 「退場遷移のトリガー名」：状態から出力される遷移のトリガー名を重複を排除して抜き出し、比較したい状態間で一致する個数をカウントする。例えば、S1, S1' を比較すると、S1 では e1, S1' では e1 が退場遷移となり、e1 のみが一致するために1がカウントされる。
- 「状態名」：ペアとなった状態間で状態名が完全一致するか比較する。例えば、S1 と S1' を比較すると、これらの文字列は完全一致ではないため、カウントされない。
- 「開始疑似状態との接続」：状態の遷移元に開始疑似状態があるか判定する。存在すれば「1」、存在しなければ「-」と判定される。例えば、S1, S1' を比較すると、各状態の遷移元に開始疑似状態があるため、1がカウントされる。なお、開始疑似状態には実行活動がないため、「隣接する状態」の例外として「開始疑似状態との接続」を特徴として設けている。

また、遷移間は、「遷移元の実行活動」、「遷移先の実行活動」、「ガード」、「エフェクト」といった、遷移とその周辺の要素を遷移の特徴と捉えて対応付ける。図3と図4には重複する遷移がないため、遷移の特徴の説明のみ以下に示す。

- 「遷移元の実行活動」：遷移元の状態の実行活動が一致するか判定する。
- 「遷移先の実行活動」：遷移先の状態の実行活動が一

表3 (S1,S1'), (S3,S3') の組み合わせでの比較結果

Table 3 Comparison Results for Combination of (S1,S1'), (S3,S3') .

比較項目	S1 と S1'	S3 と S3'
隣接する状態	do1 (遷移元)	do1 (遷移先) do2 (遷移元)
入場遷移のトリガー名	「開始」	-
退場遷移のトリガー名	e1	-
状態名	-	-
開始疑似状態との接続	1	-
特徴の一致度	4	2

表4 (S1,S3'), (S3,S1') の組み合わせでの比較結果

Table 4 Comparison Results for Combination of (S1,S3'), (S3,S1') .

比較項目	S1 と S3'	S3 と S1'
隣接する状態	-	-
入場遷移のトリガー名	e4	e3
退場遷移のトリガー名	-	-
状態名	-	-
開始疑似状態との接続	-	-
特徴の一致度	1	1

致するか判定する。

- 「ガード」：ガード条件が一致するか判定する。
- 「エフェクト」：エフェクトが一致するか判定する。

そして、上記の特徴が最も一致し、カウントが最大になるように状態遷移を1対1で対応付ける。重複する実行活動 do1 を持つ図3の状態 S1 と S3, 図4の状態 S1' と S3' の組み合わせとしては、「S1 と S1', S3 と S3', 「S1 と S3', S3 と S1'」の2通りが考えられる。上記に箇条書きで示した状態の特徴に基づいて比較を行い、「S1 と S1', S3 と S3'」の組み合わせの特徴を表3, 「S1 と S3', S3 と S1'」の組み合わせの結果を表4に示す。表3, 表4では、特徴（比較項目）に基づくカウントの和を「特徴の一致度」と呼ぶ。「特徴の一致度」の和は、表3の結果から「S1 と S1', S3 と S3'」が6個、表4の結果から「S1 と S3', S3 と S1'」が2個であるため、最も和が多い「S1 と S1', S3 と S3'」の組み合わせで対応付けられる。なお、遷移の場合も同様に「特徴の一致度」の和が最も多いものが対応付けられる。

「特徴の一致度」の和が同数となる複数の組み合わせがあった場合は、それらの組み合わせそれぞれに対して誤り特定を行う。また、正答例にのみ存在する状態遷移や、答案にのみ存在する状態遷移は「対応する状態遷移が存在しない」と判定される。これらの対応付けの結果をもとに3.3.2項で誤り箇所を特定を行う。

3.3.2 手順2. 差分解析による誤り特定

本項では3.3.1項の手順で1対1で対応付けられた状態遷移と対応付けられなかった状態遷移に対してそれぞれ誤

表 5 誤りの種類の特定方針
Table 5 Policy for Identifying Types of Errors.

誤りの種類	特定方法
状態の不足	状態遷移間に対応付けた結果として、正答例にのみ含まれる状態を調べる。
状態の過剰	状態遷移間に対応付けた結果として、答案にのみ含まれる状態を調べる。
遷移の不足	状態遷移間に対応付けた結果として、正答例にのみ含まれる遷移を調べる。
遷移の過剰	状態遷移間に対応付けた結果として、答案にのみ含まれる遷移を調べる。
接続遷移の不足	対応付けた状態間で、正答例よりも少ない入場遷移または退場遷移を持つ状態を答案から調べる。
接続遷移の過剰	対応付けた状態間で、正答例よりも多い入場遷移または退場遷移を持つ状態を答案から調べる。
遷移先誤り	対応付けた遷移間で遷移先が互いに異なっている。
遷移元誤り	対応付けた遷移間で遷移元が互いに異なっている。

り特定を行う。

まず、表 2 の誤りの種類を特定するための方針を表 5 に示す。3.3.1 項で「対応する状態遷移が存在しない」と判定された要素は表 5 をもとに、正答例にのみ含まれている状態を「状態の不足」、答案にのみ含まれている状態を「状態の過剰」、正答例にのみ含まれている遷移を「遷移の不足」、答案にのみ含まれている遷移を「遷移の過剰」として判定する。

次に 1 対 1 で対応する状態遷移が存在する場合の判定について説明する。表 2 に示す誤りの種類では遷移先と遷移元の状態の両方を調べる必要があるため、対応付けられた状態遷移の隣接する状態まで比較を行い、差分のある箇所を誤りとする。図 3 の状態 S2 と隣接する状態の例を図 5 に示す。図 4 の状態 S2' と隣接する状態の例を図 6 に示す。対応する要素間の比較による誤り特定の説明を図 5、図 6 を用いて行う。

まず、「接続遷移の過剰」、「接続遷移の不足」かどうか判定するために、対応付けられた状態の入場遷移と退場遷移が一致しているかどうか調べる。例えば、状態 S2 の入場遷移は e1, 退場遷移は e2, e3 であり、状態 S2' の入場遷移は e1, e2, 退場遷移は e2, e4 である。このことから状態 S2 に対して状態 S2' は、入場遷移 e2 と退場遷移 e4 が「接続遷移の過剰」、退場遷移 e3 が「接続遷移の不足」と判定される。次に、図 5、図 6 に含まれる対応付けられた遷移に着目し、「遷移先誤り」、「遷移元誤り」かどうか判定するために、対応付けられた各遷移先と遷移元が対応付けられた状態と一致するかどうか調べる。ここで図 5 の e2 の遷移先が S1 なのに対し、図 6 の e2 は遷移先が S2' であるため、e2 は「遷移先誤り」と判定される。

上記の「接続遷移の過剰」、「接続遷移の不足」、「遷移先誤り」、「遷移元誤り」について、図 3 と図 4 に対して特定した結果を表 6 に示す。なお、正答例の S1 と対応する答案の状態は S1' であることから、これらの情報を組み合わせることで「e2 の遷移先は S2' でなく S1'」といった出力にまとめられる。

表 6 誤り特定の出力結果

Table 6 Output Results of Error Identification

比較箇所	誤り箇所	誤りの種類	出力
S1'	e3	接続遷移の過剰	S1' の入場遷移に e3 過剰
S1'	e2 e4	接続遷移の不足	S1' の入場遷移に e2 不足 S1' の入場遷移に e4 不足
S2'	e2 e4	接続遷移の過剰	S2' の入場遷移に e2 過剰 S2' の退場遷移に e4 過剰
S2'	e3	接続遷移の不足	S2' の退場遷移に e3 不足
S3'	e3 e4	接続遷移の過剰	S3' の退場遷移に e3 過剰 S3' の入場遷移に e4 過剰
S3'	e3 e4	接続遷移の不足	S3' の入場遷移に e3 不足 S3' の退場遷移に e4 不足
e2	e2	遷移先誤り	e2 の遷移先は S2' でなく S1'
e3	e3	遷移先誤り 遷移元誤り	e3 の遷移先は S1' でなく S3' e3 の遷移元は S3' でなく S2'
e4	e4	遷移先誤り 遷移元誤り	e4 の遷移先は S3' でなく S1' e4 の遷移元は S2' でなく S3'

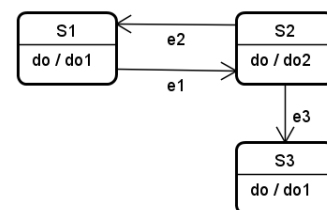


図 5 状態 S2 と隣接する状態

Fig. 5 State S2 and Adjacent States.

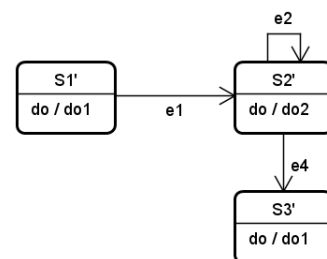


図 6 状態 S2' と隣接する状態

Fig. 6 State S2' and Adjacent States.

4. 評価実験

本章では提案手法の評価実験の内容について説明を行う。

4.1 実験目的

本実験では、提案手法の有用性を評価することを目的に、手動評価結果と提案手法による自動評価結果の比較を行う。手動評価結果の内、提案手法による自動化で代替できる内容を示すことによって提案手法が有用である見込みを示す。また、手動評価結果との比較では、どのような指摘項目が出力されるか、出力内容に项目的な差や特定できていない誤りが存在するかについて示し、提案手法で特定できていない誤りや、誤判定されてしまう答案のパターンを考慮した今後実現すべき改善点についての考察を行う。さらに、ツールの使用の観点から、提案手法によって誤り箇所特定を行う際に要する実行時間についても調べる。

4.2 実験の対象

提案手法では状態遷移を1対1で対応付ける特徴があることを踏まえ、重複する状態遷移が含まれる場合や状態遷移数が多い場合に正しく誤りを特定できるか検証することを考えた。そのため、重複する実行活動を含まない図、重複する実行活動を含む図、状態遷移の数が2倍程度に増え、かつ重複する実行活動を含む図の3種の答案を得られるようモデリング課題を構成した。今回の実験では、入力として与える正答例、及び答案の作成には、UMLに対応しているモデリングツールのastah*[15]を用いた。これら3つのモデリング課題を、初めてステートマシン図を学習する情報工学系の学科2年生の計19人の学習者に課し、答案となるステートマシン図を計57個(=19人*3課題)を得た。その57個の答案のうち、15個の答案が誤っていることが実験前の手動評価でわかっており、一つの答案に誤り箇所が複数箇所存在している答案は15個中5個であった。課題間で学習者に理解度の差が生じにくいように、3つの課題はいずれもファンヒーターを題材としたものであった。その中で、重複する実行活動を含む図を得るためのファンヒーターの要求文を以下に示す。

- チャイルドロック付きのファンヒーターがある。
- 電源を入れると温風が送風され、電源を切ると送風を停止する。
- チャイルドロック中は電源を切ることしかできない。
- 電源の状態によらず、チャイルドロックをかけたり、解除できたりする。
- ファンヒーターは最初、電源が切れていて、チャイルドロックもかかっている。

また、この要求文の用語集において、実行活動は「温風を送風する」、「送風を停止する」、トリガーは「電源を入れる」、「電源を切る」、「チャイルドロックをかける」、「チャ

イルドロックを解除する」、「」（空値）であった。なお、空値のトリガーは、開始疑似状態からの遷移など、トリガーを明に与えられない遷移のためのものであった。

4.3 実験結果

実験で使用した正答例、答案のステートマシン図の一例を図7、図8に示す。また、図7、図8間の手動評価結果と自動評価結果を表7に示す。図8の答案は手動評価結果から、状態「電源OFF__チャイルドロックOFF」から状態「電源OFF__チャイルドロックON」への遷移「チャイルドロックをかける」が不足している、という誤りが含まれるという結果であった。提案手法に図7、図8を入力として与えると、遷移「チャイルドロックをかける」不足、「電源OFF__チャイルドロックON」の入場遷移に「チャイルドロックをかける」不足、「電源OFF__チャイルドロックOFF」の退場遷移に「チャイルドロックをかける」不足、という表7に示した3つの自動評価結果が得られた。これら3つの自動評価結果は、それぞれ「チャイルドロックをかける」という遷移に関する誤りである。このような同一の誤り箇所から出力された自動評価結果を組み合わせることで、「電源OFF__チャイルドロックOFF」から「電源OFF__チャイルドロックON」への遷移「チャイルドロックをかける」が不足している、という指摘項目を最終的に出力する。上記の処理を行うことで、自動評価結果によって手動評価結果を代替する内容をフィードバックすることができる。

提案手法を計57個の答案に対して適用したところ、誤りを含む答案15個中15個の自動評価結果が手動評価結果を代替する内容であったため、各答案に含まれる全ての誤り箇所が正しく特定されたといえる。提案手法を適用した学習者の答案には、「状態の不足」、「接続遷移の過剰」、「接続遷移の不足」、「遷移先誤り」、「遷移元誤り」に関する誤りが含まれており、これらの誤りを指摘項目として出力することができた。また、各課題ごとに19人分の答案をまとめて入力として与えて実行すると、誤り箇所の特定結果を出力するまでにおよそ5秒の処理時間がかかった。

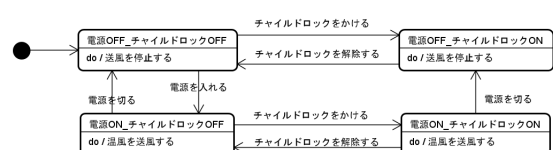


図7 重複する実行活動を含む正答例

Fig. 7 Sample Answer Including Duplicate Do Activities.

表 7 誤りの種類の特定結果
 Table 7 Results of Identifying Type of Errors.

手動評価結果	自動評価結果
状態「電源 OFF __チャイルドロック OFF」から状態「電源 OFF __チャイルドロック ON」への遷移「チャイルドロックをかける」が不足している	遷移「チャイルドロックをかける」不足 「電源 OFF __チャイルドロック ON」の入場遷移に「チャイルドロックをかける」不足 「電源 OFF __チャイルドロック OFF」の退場遷移に「チャイルドロックをかける」不足

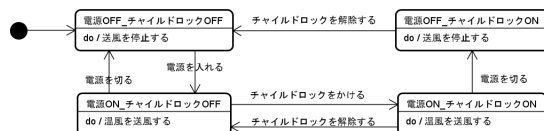


図 8 重複する実行活動を含む誤りを含む答案

Fig. 8 Answer that Contains Error Including Duplicate Do Activities.

しかし、正解していた 42 個の答案のうち 2 個の答案が、正答例と同じ振る舞いで誤りがないにも関わらず、対応付けがうまくいかず誤りであると誤判定されてしまった。この答案を以降では別解と呼び、誤判定された答案の一例を図 9 に示す。図 9 は、正答例では使用されていない変数が用いられており、変数を用いることで正答例とは大きく異なる記述だが、正答例と同じ振る舞いとなっている。

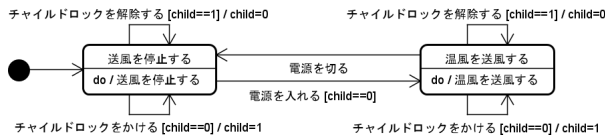


図 9 誤判定された答案

Fig. 9 Misjudged Answer.

4.4 考察

実験結果から、提案手法を用いることで 2 つの答案を除き、図要素の配置が異なっている図であっても、表 2 で示した誤り箇所を特定できることが確認できた。一方で、図 9 に示した別解のような例では誤判定されてしまうことがわかった。誤判定された別解を調査したところ、ステートマシン図が変数の利用や同じ状態を複数記述するといった冗長な表現で記述することで、正答例と同じ振る舞いで状態数の異なるステートマシン図が記述できることが原因であった。図 9 の別解では、ガードと変数を用いることで、正答例よりも少ない状態数で正しい図を実現していた。しかし、提案手法が変数やガードの利用、冗長な表現を前提とした遷移の制御に対応できておらず、正答例と形の異なる別解の比較で誤判定を生じてしまった。

この誤判定を改善するために、SML4C の出力を用いる必要がある。SML4C はガードや変数を用いた図にも対応しているため、正答例と図 9 の比較では振る舞いが正しい

と判定される。そのため、SML4C で正しいと判定された場合は、その答案を別解として新たな正答例とすることで対応ができると考えられる。今後は正答例と答案の間で図要素が 1 対 1 ではなく、多対多に結びつく状況に対処できるよう提案手法を拡張する必要がある。

5. 関連研究

提案手法ではステートマシン図の誤り箇所の特定を行うことが目的であるため、本章では先行研究でどのような誤りを自動特定する手法を提案しているのか、またどのような誤りを特定すべきかについての調査の研究を挙げ比較する。

ステートマシン図教育の記述や読解の課題において、「状態遷移の過不足」、「振る舞い」、「開始疑似状態と終了状態の見極め」といった誤りが統計的に多く、これらの誤りに留意した教育をする必要があるということが先行研究の調査結果からわかっている [14]。これらの調査結果の統計的に多い誤りに類する誤り箇所に基づき、本研究では自動で特定可能なレベルで誤りを詳細に分類化していることが新規な部分となる。

一方で、提案手法が対象とするステートマシン図の答案では、「ある状態から同じ状態に対して複数の遷移で繋ぐ」といった記法に関する誤りが考えられるが、このような誤りを判定するために記法チェッカというツールが提案されている [9]。本稿の提案手法と連携させることでより広範囲な誤りチェックを行えることが期待される。

また、振る舞いに関する研究として、SML4C (SMart-Learning for Classification) [7] というツールが提案されている。SML4C は、学習者が作成したモデル図から振る舞いのテストケースを自動的に生成する機能を持ち、テストの結果にもとづいて、ステートマシン図の振る舞いの等価性の観点で図を分類できる。この手法は、変数等によるステートマシン図の記述の多様性に依存することなく、振る舞いの等価性を検証できるため、正答例とは異なる振る舞いを持つ誤った答案を判定支援する。本研究では、誤りの種類や箇所を特定する方法を提供するが、SML4C ではその特定はできないため、その点が提案手法が優位な部分となる。

また、先行研究では、誤り箇所を特定するためにステー

トマシン図に対し静的解析と動的解析を行う手法が提案されている [8]. 静的解析では, 変数やガード条件等の正しさ等が検証され, 動的解析では, 実行不可能な通ることのできない遷移や, 特定の状態から動かなくなるデッドロックの存在等について検証される. この手法は, 主にステートマシン図の振る舞いに関する誤り箇所を出力することができる. しかし, この研究は開発者向けの手法であるため, 教育に使用した場合, より詳細な誤りに関するフィードバックをすることに適していない. 本研究は, 教育で使用することを想定しており, ステートマシン図上のどこにどのような誤りが含まれるかの詳細を教育者や学習者にフィードバックできる点が優位な部分となる.

6. まとめ

本稿では, ステートマシン図の答案から状態遷移に関する誤り箇所の特定, 及び誤り分類を自動で行う手法を提案した. 評価の結果, 「状態, 遷移, 実行活動, 開始疑似状態, 終了状態」で構成された単純なステートマシン図の答案であれば即時に誤り箇所を特定できることがわかった. そのため, 用語集を用いたステートマシン図の記法学習において使用できることが確かめられた.

今後の課題として, 評価実験の別解の例のように, 変数の使用や冗長な表現を含む正答例よりも状態数が削減, または追加されたステートマシン図に対処することが挙げられる. 解決アイデアとして, SML4C の利用や, 同一の実行活動を持つ等価な状態をまとめることで実現する方針である. さらに幅広い要求文やステートマシン図の仕様に対応するために, 表 1 に示した入場動作や退場動作, ガード, エフェクト, 変数の他, 直交状態, メッセージ通信等を含む答案, 及び用語集の存在しない問題文に対処できるよう提案手法を拡張する必要がある. また, 用語集の存在しない問題文へ対処するためのアイデアとして, 自然言語処理や深層学習等の手法を用い, 使用している文字列の類似度の比較を行うことで, 用語集の制約排除の実現を目指す.

参考文献

- [1] Object Management Group, “OMG Unified Modeling Language version 2.5.1,” (閲覧日:2021.01.29). ”入手先 <<https://www.omg.org/spec/UML/2.5.1/PDF>>”.
- [2] I. Graja, S. Kallel, N. Guermouche, S. Cheikhrouhou, and A. Hadj Kacem, “A comprehensive survey on modeling of cyber-physical systems,” *Concurrency and Computation: Practice and Experience*, vol.32, p.e4850, 2018.
- [3] L.T.W.Agner, I.W.Soaes, P.C.Stadzisz, and J.M.Simao, “A brazilian survey on uml and model-driven practices for embedded software development,” *Journal of Systems and Software*, vol.86, pp.997–1005, 2013.
- [4] 文部科学省, “【技術・家庭編】中学校学習指導要領 (平成 29 年告示) 解説,” (閲覧日:2021.05.09). ”

- 入手先 <https://www.mext.go.jp/a_menu/shotou/new-cs/1387016.htm>”.
- [5] S.Hara, M.Kayama, N.Nakano, T.Nagai, and N.Taguchi, “A uml programming environment for ict related subject at junior high school,” *Proceedings of the 2019 The 3rd International Conference on Digital Technology in Education*, pp.141–146, 2019.
 - [6] S. Sendall and W. Kozaczynski, “Model transformation: the heart and soul of model-driven software development,” *IEEE Software*, vol.20, no.5, pp.42–45, 2003.
 - [7] S. Ogata and M. Kayama, “Sml4c: Fully automatic classification of state machine models for model inspection in education,” *2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, pp.720–729, 2019.
 - [8] C.Schwarzl and B.Peischl, “Static and dynamic consistency analysis of uml state chart models,” *International Conference on Model Driven Engineering Languages and Systems*, pp.151–165, 2010.
 - [9] 但馬将貴, 香山瑞恵, 小形真平, 橋本昌己, “Uml に基づく概念モデリングにおける状態遷移図に対するモデル記法チェック機能の効果,” *電子情報通信学会技術研究報告*, 第 116 巻, pp.7–12, 2017.
 - [10] 古川善吾, 野木兼六, 徳永健司, “Agent: 機能テストのためのテスト項目作成の一手法,” *情報処理学会論文誌*, 第 25 巻, pp.736–744, 1984.
 - [11] 五島光祥, 小形真平, 槇原絵里奈, 岡野浩三, “Uml ステートマシン図の教育的フィードバック自動生成に向けた答案の誤り特定自動化手法の提案 ~ 特定すべき誤りの種類とその特定結果への着目 ~,” *信学技報*, 第 121 巻, pp.56–61, 2021.
 - [12] Object Management Group, “ABOUT OMG,” (閲覧日:2020.12.16). ”入手先 <<https://www.omg.org/>>”.
 - [13] P. Pourali and J.M. Atlee, “An empirical investigation to understand the difficulties and challenges of software modellers when using modelling tools,” *Proceedings of the 21th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems*, pp.224–234, 2018.
 - [14] M. Kayama, S. Ogata, K. Masumoto, K. Itou, M. Hashimoto, and M. Otani, “状態遷移図作成に際する初学者の誤り分析とそれに基づく教育方法の検討,” *情報処理学会研究報告コンピュータと教育 (CE)*, 第 2012-CE-117 巻, pp.1–9, 2012.
 - [15] Change Vision, “Change Vision :astah,” (閲覧日:2021.7.10). ”入手先 <<https://astah.change-vision.com/ja/index.html>>”.