公開

研 究 終 了 報 告 書

「 圏論と自動検証による機械学習の仕様保証 」

研究期間: 2021 年 10 月～2024 年 3 月
研 究 者: 内藏 理史

1. 研究のねらい

機械学習は様々な目的のために多くの手法が提案され研究されている. それらを実際に社会へ応用する際には**機械学習アルゴリズムが望ましい性質を満たしていることを保証したい**. 望ましい性質の例としては例えば**差分プライバシー**, **公平性**, **安全性**, **学習の収束性**などを挙げることができる.

プログラミング言語の研究分野では機械学習における典型的なパターンをうまく表現するために**確率的プログラミング言語**や**微分可能プログラミング言語**などを対象としてその圏論的意味論の研究を含めて様々な研究が行われている. 機械学習のアルゴリズムの性質を保証するためにはこうした言語で書かれたプログラムを対象としてそれが望ましい性質を満たしているかを保証するアプローチが考えられる.

プログラム検証の分野では, 与えられたプログラムが与えられた仕様を満たしているかどうかを数学的に保証する手法について様々な研究が行われている. しかし**確率的/微分可能プログラミング言語を対象とした検証やその自動化についてはまだまだ課題が多い**. プログラムの検証は一般には多大な労力を要する作業であり, 可能な限り検証を自動化することはプログラム検証の実用化を目指すうえでは重要な要求である.

本研究の目的は「確率的/微分可能プログラミング言語を対象とした自動検証」である. アプローチとしては**圏論的意味論**を用いて(1)既存の自動検証の手法に対して意味論の枠組みの中で数学的抽象化を行い, (2)確率的/微分可能プログラミング言語の意味論を用いて具体化を行うことで新しい自動検証手法の獲得を目指す. 言い換えると数学的抽象化により問題の本質的な部分を取り出すことで筋良く新しい自動検証手法を導こうというアプローチである. また本研究は「自動」検証を目的としているため, 自動化に有望そうであることが既存研究によりわかっている検証手法, 具体的には**プログラム論理**と**篩型システム**に焦点を当てる. これらの手法に関しては既存研究により圏論的意味論を用いた取り扱いについてもある程度わかっているため, それらを確率的/微分可能プログラミング言語を対象としたものへと寄せていく本研究は現実性もあると考えられる. 自動検証の研究分野において特定の対象に対する深い洞察から得られた検証手法と, それを抽象化により一般の対象へと拡張する圏論的意味論を組み合わせることで今までにない新しい自動検証手法の獲得を目指す.

2. 研究成果

(1)概要

圏論的意味論を用いて自動検証の一般的枠組みを新しく作ることができた. この枠組みでは一般の計算効果を含んだ高階関数型プログラムを対象として, あるクラスの一般化された最弱事前条件変換子を用いて表現された性質の検証を行うことができる. これは具体例としては(高階関数型)確率的プログラムが与えられたときに, ある条件がプログラム実行後に成り立

つ確率（あるいは事後確率）を検証したり，確率的プログラム内で発生する「コスト」に関してその期待値や高次のモーメント（=コストのべき乗の期待値）を検証したりできる．
この枠組みは二つの構成要素の組み合わせとなっている．一つは CPS 変換と呼ばれるプログラム変換により，与えられたプログラムに対する最弱事前条件変換子をある種の高階不動点論理の項として得る部分であり，もう一つは CPS 変換後に得られた高階不動点論理の項がより具体的にどのようなものになっているかを依存篩型システムを用いて検証する部分である．この二つの構成要素はどちらも自動検証器として実装することが可能であり，実際に実装したところいくつかの問題例を解くことができた．一方で解くことができなかった問題例もあり，それは内部で用いる制約ソルバーの性能不足や現時点での理論面での制約などが原因であるが，その解決は今後の課題となっている．

（2）詳細
本研究によって得られた枠組みでは右図のような流れで自動検証を行う．図の上部の「①CPS 変換」と書いてある部分が下記の研究項目 1 に関連する部分であり，「②篩型システム」とあるのが研究項目 2 に関連する．この枠組みそのものは一般的なものとなっており，「一般の計算効果と再帰ありの（単純型付き）関数型言語」が検証対象となっているが，適切な設定に具体化することで関数型確率的プログラミング言語がこの枠組みで扱える．

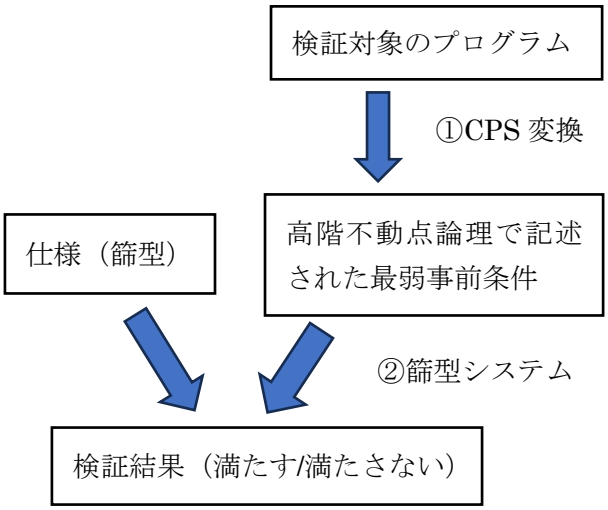

図 1 CPS 変換と篩型システムによる自動検証の流れ

以下で前半と後半に分けてもう少し具体的に検証の枠組みを説明する．

研究項目 1: プログラム論理
当初の研究計画においてはプログラム論理の研究をすることを予定していたが，実際には Hoare スタイルのプログラム論理とある意味で等価な概念である最弱事前条件変換子についての研究を行った．
一般にプログラム論理ではプログラム実行前に成り立つと仮定する事前条件 P，検証対象のプログラム c，実行後に成り立っていてほしい性質である事後条件 Q の 3 つ組（= Hoare Triple）を用いて{P}c{Q}という形で仕様を表現し，推論規則を用いて仕様が満たされているかを検証する．**最弱事前条件**とはプログラム c と事後条件 Q がに対して{wp[c](Q)}c{Q}が成り立つような最弱の事前条件 wp[c](Q)のことである．そして事後条件に対して最弱事前条件を対応させる写像のことを最弱事前条件変換子と呼ぶ．これは「{P}c{Q}が成り立つ ⇔ P が wp[c](Q)を含意する」という意味で Hoare 論理と等価な概念となっている．
上記の説明は計算効果を考えない場合の話であるが，一般の計算効果を考えると様々な種類の“最弱事前条件”が考えられることが知られており[Aguirre & Katsumata, MFPS’20]，その

公開

中には weakest pre-expectation transformer のような確率的プログラムの検証にかかわるようなものも含まれている（weakest pre-expectation を使うと例えば事後条件が成り立つ確率がわかる）. 構文を考えずに一般的な“最弱事前条件”を定義する既存研究はあったが, 今回行った研究では自動検証という目標のために最弱事前条件の**構文的な計算**のための理論を圏論的意味論を用いて考えた.

成果 1-1: CPS 変換による最弱事前条件の構文的な計算

一般の計算効果を考えている状況であっても最弱事前条件は構文的に計算することができ, しかもそれは CPS 変換と呼ばれるよく知られたプログラム変換によってできることがわかった. 技術的には検証の対象となるプログラム M に対してそれをある種の高階不動点論理の項 M’ へと変換する手続きであって, M’ が M の最弱事前条件変換子を表すようなものが定義できる（図 2）. この変換の健全性, つまり「M’ が M の最弱事前条件変換子を表す」ことがここでの主定理であるが, 圏論的意味論を用いることで一般の計算効果に対して成り立つ一般的な定理を得ることができた.

CPS 変換前: コストの確率分布を表す

$$\text{let rec coin } x = (\text{coin } ())^{\checkmark} \oplus_{1/2} () \text{ in } \quad \text{coin } ()$$

CPS 変換後: 平均コストを表す

$$\begin{aligned}&\text{let fix coin } x\ k = \\ &\qquad 1/2 \cdot (1 + \text{coin } ()\ k) + 1/2 \cdot (k\ ()) \text{ in} \\ &\text{coin } ()\ (\lambda y.0)\end{aligned}$$

図 2 CPS 変換による最弱事前条件（この場合は平均コスト）の計算

成果 1-2: プログラム検証に関するふたつの既存研究との関連やそのほかの具体例の発見

(1)プログラムのトレースの検証に関する既存研究[Kobayashi et al., ESOP’18]と(2)確率的プログラムの平均コストの検証に関する既存研究[Avanzini et al. ICFP’21]は成果 1-1 で示された CPS 変換と最弱事前条件の関係を特定の状況に具体化したものとして理解できることが今回の研究で分かった. これらの既存研究と最弱事前条件とのつながりはこれまで明示的には調べられていなかったものである. 言い方を変えると成果 1-1 はこれらの既存研究の一般化を与えるものであり, これらの既存研究をさらに拡張するうえで有用な知見となる. 実際, 確率的プログラムの平均コストの検証[Avanzini et al. ICFP’21]はもともと離散的な確率分布のみを対象としていたがこれを連続確率分布へと拡張することや, あるいはコストの平均をコストの高次のモーメントへと拡張するといったことが容易に実現できることが今回の研究により分かった. 他にも成果 1-1 の具体例として conditional weakest pre-expectation が扱えることもわかったが, これは確率的プログラムに conditioning が含まれている場合（事後確率を考える必要がある場合）の検証に用いることができる.

研究項目 2: 篩型システム

研究項目 1 で得られた成果を用いると与えられたプログラム M に対してその最弱事前条件を表す高階不動点論理の項 M’ が得られる. この M’ の解釈が最弱事前条件と一致することは健全性により保証されているが, 実際に M’ の解釈がどのような値になるのかというのは項 M’ を見て直ちにわかるものではない. CPS 変換後の M’ は「純粋な（=計算効果を含まない）」項にな

公開

っているため，計算効果を含んだプログラム M を直接調べるよりも M’を調べるほうが比較的簡単ではあるが，**CPS 変換して M’を得るだけでは検証が完了するわけではない**．そのため，今回は**依存篩型システムを用いて CPS 変換後の M’を調べる**ことで元のプログラム M が仕様を満たしているか否かを自動検証するという方針で実装に向けた研究を行った．例えば図 2 のプログラムに対して図 3 のような仕様が与えられたときにこの仕様が満たされているかを依存篩型システムを用いてチェックすることが目標である．

$$\begin{aligned}&\mathrm{coin} : \mathrm{unit}\\ &\quad \to (\mathrm{unit} \to \{x : \mathrm{real}_{\geq 0} \mid x = 0\})\\ &\quad \to \{r : \mathrm{real}_{\geq 0} \mid r \leq 1\}\end{aligned}$$

図 3 篩型による仕様の記述．図 2 での平均コストは高々1 であることを篩型で表現できる．この型は CPS 変換後の項に対するものである．

成果 2-1: 高階不動点論理に対する依存篩型システムの健全性の証明

高階不動点論理の構文の大部分は純粋な単純型付きラムダ計算と共通している．そのため CPS 変換後の M’を調べるために純粋な単純型付きラムダ計算に対する依存篩型システムがほぼそのままの形で利用できると期待されるが，実際には以下の 2 点に関して修正が必要となる．

1 点目は今回用いた高階不動点論理の意味論は圏論的意味論を用いて一般化されているためそれに対応して**依存篩型システムの健全性の証明の一般化**が必要になることである．これについては自身の過去の研究である[Kura, FoSSaCS’21]の結果を用いることで解決した．

2 点目は高階不動点論理の不動点に関する問題である．不動点の意味論は最小不動点を用いて定義されるという点で単純型付きラムダ計算における再帰と似たようなものである．しかし定義に用いる順序関係が異なるため，**再帰に対する型付け規則は不動点に対して用いることができない**（健全ではなくなる）．この問題については再帰に対する型付け規則の中で用いる述語の“admissibility”を追加で仮定することで不動点に対して健全な型付け規則を与えられることを証明した．

成果 2-2: 篩型システムの実装と確率的プログラムの検証への応用

成果 2-1 により得られた依存篩型システムに対して型検査器の実装も行った．型検査器は与えられた高階不動点論理の「項 M’」と M’に対する「型 T」が与えられたときに依存篩型システムの型付け規則を用いて「項 M’は型 T を持つ」ことが導けるかどうかを適切なアルゴリズムを用いて自動で判定する．これと研究項目 1 における CPS 変換を組み合わせることで全体として計算効果を持つプログラムに対する自動検証器が得られることになる（図 1）．つまりもし M を CPS 変換して得られた項 M’が仕様を表す型 T を持つならばプログラム M は仕様を満たしていると結論付けることができる．今回の行った実装では特に確率的プログラムのいくつかの性質の検証にターゲットを絞った．

今回のアプローチの長所として「様々な検証の問題がほぼ共通の型検査器によって自動で検証できる」という点が挙げられる．今回の枠組みは圏論的意味論によりかなり一般的に健全性が証明されており，上にも例を挙げた通り多くの問題がこの枠組みに入る．その一方で CPS 変換を経由して純粋な単純型付きラムダ計算の検証へと帰着することで，実装面では既存の型

公開

検査器を少し拡張するだけで対応できるようになっている. これは実装が楽というだけではなく, 依存篩型システムのより良い型検査アルゴリズムが提案されたときに, たとえそれが純粋なプログラムの検証を想定して提案されたものであっても, それをほぼそのまま確率的プログラムの検証へと利用できるようになるということでもある.

3. 今後の展開

今後の展開としてまず第一にここまでの CPS 変換と篩型システムについての研究を通して見えてきたいくつかの課題の解決が挙げられる. その中には型検査器やその内部の制約ソルバーの強化といった実装面での課題と, 検証対象のプログラムや検証可能な性質をより広げていくという理論面での課題を含む. 特に関係的な性質の検証は機械学習の検証においても重要であることがわかってきたためその方向についても研究を展開していきたい.

4. 自己評価

「圏論的意味論による理論と自動検証の組み合わせ」という部分に関しては自分なりに面白い結果が得られたと思っている. 詳細は「研究成果」に書いたとおりであるが, 圏論的意味論を用いて最弱事前条件と CPS 変換との間の一般的な関係を示しつつそれを篩型システムを使って自動検証まで落とし込むことができたのは, 圏論と自動検証というこれまで距離があった分野をつなぐ成果だと言える.

その一方で機械学習の検証という点に関してはまだまだやり残したことは多い.
「今後の展開」にも書いたように関係的な性質の検証は差分プライバシーや fairness などの検証で必要になってくるがそのような性質の自動検証はまだ実現できていない. また今回は主に確率的プログラムに関する結果であり, 微分可能プログラムについてはまだやれていない. 一般的な理論は今回得られているのでそれをそのまま微分可能プログラムに適用できる可能性もあるが, まだ具体的なところは分からない.

5. 主な研究成果リスト

（1）代表的な論文（原著論文）発表

研究期間累積件数: 1 件

1. Alejandro Aguirre, Shin-ya Katsumata, Satoshi Kura, “Weakest preconditions in fibrations”, Mathematical Structures in Computer Science, vol 32, issue 4, pp. 472-510, 2022
2.
3.

（2）特許出願

公開

研究期間全出願件数: 0 件（特許公開前のものは件数にのみ含む）

1	発　明　者	
	発明の名称	
	出　願　人	
	出　願　日	
	出願番号	
	概　　要	

（3）その他の成果（主要な学会発表、受賞、著作物、プレスリリース等）

学会発表: HOPE 2023, SYCO 10, CSCAT 2023, PPL 2023 など