

令和 6 年 6 月 4 日現在

機関番号：62615

研究種目：若手研究

研究期間：2019～2023

課題番号：19K20248

研究課題名（和文）データに基づき自動修正も含む型エラーデバ깅の理論と実際

研究課題名（英文）Theory and practice of type error debugging, including automatic correction based on data

研究代表者

対馬 かなえ (Tsushima, Kanae)

国立情報学研究所・情報学プリンシプル研究系・特任研究員

研究者番号：80754663

交付決定額（研究期間全体）：（直接経費） 2,700,000円

研究成果の概要（和文）：本研究では主に二つの成果が得られた。一つ目は GPT-3 を使用した型エラーの自動修正である。具体的には、型エラーのあるプログラムに対して、ソースコード解析のタスクを実行して有用なプロンプトを生成する。それを GPT-3 に提供して、型エラーを修正するパッチを生成する。三種類のモードを作成し、OCaml の既存公開データセットで検証を行った。比較においても、他の OCaml を対象とした自動修正の研究二つを上回った。二つ目は部分評価を使用して実行することにより、型エラーの箇所をわかりやすく提示する手法である。プロトタイプを実装し、前述の既存公開データセットのプログラムで効果を確認した。

研究成果の学術的意義や社会的意義
型はプログラムの信頼性を高めるために重要である一方で、プログラミング時の型エラーはプログラマにとって手間である。そのデバッグを容易にすることは、プログラマの実際の生産性、およびプログラムの信頼度を高めるために重要である。また、本研究では大規模言語モデルのプログラミング言語に対する深い理解を活用することで、プログラマのデバッグを手助けすることが可能であることを示した。これはこれからのプログラミングをサポートしていく上で重要な成果である。

研究成果の概要（英文）：Two main results were obtained in this study. The first is type error debugging using GPT-3. Specifically, it performs a source code analysis task on a program with type errors to generate useful prompts. Provide it to GPT-3 to generate a patch to correct the type error. We created three different modes and validated on the existing public dataset of OCaml. In comparison, it outperformed two other OCaml-based studies of automatic correction. The second is a method that uses partial evaluation to perform and present the location of type errors in an easy-to-understand manner. We implemented a prototype and verified its effectiveness on the aforementioned programs on the existing public datasets.

研究分野：プログラミング言語

キーワード：関数型言語 プログラミング言語 デバ깅 型エラー 大規模言語モデル

1. 研究開始当初の背景

プログラミング言語において、型はプログラムの安全性を保証する道具として広く使用されている。型の整合性を検査することで、数字と文字列の足し算(1 + "2") のように実行結果が決まっていないプログラムを排除して、正しく安全に実行が出来るプログラムだけを残すことができる。

しかし、正しく型がつくプログラムを書くことは容易ではない。プログラムに型がつかない場合、プログラマはコンパイラのエラーメッセージを参考に型エラーの原因を探ることになるが、型エラーメッセージが役に立たないことも多い。プログラマが型エラーデバッグに多くの時間を費やすことを避けるために、型エラーのデバッグ手段の提供は重要である。さもなくば、型の付いた信頼性の高いプログラムを書くことを妨げ、さらには型の付いた言語の使用を諦め、その結果、プログラムの信頼性を落とす結果にもなりかねない。

「より速く簡単に型エラーの原因を特定し、修正できる手法」を求めて、これまで多くの研究が行われてきた。型エラーの原因特定・修正に関する研究の難しさは、`正しい修正` が一意には定まらず、プログラムを書いたユーザの意図によって異なることが一因となっている。例えば、(1 + "2") という型エラーのプログラムを考えた時、正しいプログラムは (1 + 2) や ("1" (文字列を結合する演算子) "2") など多くの可能性が考えることができ、そのうちのどれが正しいかはプログラムを書いたユーザの意図によって異なる。そのため、これまでの研究では、ユーザの入力を用いて原因を特定するか、ユーザの入力なしにデバッグを実装した者が考案した経験則等に基づいて可能性の高い順に示す (ランキング) 方法を取っている。

2. 研究の目的

本研究では、以下の二点に注目することにより、これまでの既存研究に比べて「より速く簡単に型エラーの原因を特定し、修正できる手法」を実現することを目的とした。

- 経験則ではなく、実際のデータに基づいたランキングへ。

これまで型エラーのデバッグでは、デバッグを作るような熟練者の経験則に基づいた手法が用いられており、実際のデータに基づいていない。これでは多くの場合で速く特定できるような順のランキングを提供しているとは限らず、ユーザの負担を減らしきれない点が問題である。

- ユーザによる修正から、ログデータ・プログラム合成を用いた自動での修正へ。

ほとんどの既存研究の対象範囲はエラーの特定補助までであり、システムを用いて特定したのち、ユーザが自分で修正を行う必要がある。そのため、現状では、修正がユーザの負担になっている点が問題である。本研究では、これまでの研究で行われていない、ログデータやプログラム合成を用いてユーザの意図した実行結果になるような修正案を自動的に生成する点が独自である。

3. 研究の方法

当初、我々の実装が使用される際にエラープログラムのログを記録しており、約5年分 (2.5GB) のデータを用いることで、ランキングや自動修正を行おうと考えていた。一方で、2020年ごろから、大規模言語モデル(LLM) が活発となり、容易に使用が可能となった。LLM のデータにはプログラムも含まれており、それらの多くの正しいプログラムは有用である。

そのため、当初の方針を変更して、(1) 大規模言語モデルを使用した型エラー自動修正について研究を行った。また、LLMを使用することで、当初の予定よりも効率的に実装が可能となった。

(1) の結果も常に有用であるとは限らないため、(2) 部分評価を使用した型エラーデバッグについても研究を行った。

4. 研究成果

(1) 大規模言語モデルを使用した型エラー自動修正

本研究では、研究時に最新であった GPT-3 を用いて、型エラーの自動修正を行った。まず、型エラーのプログラムに対して、ソースコード解析のタスクを実行して有用なプロンプトを生成する。ソースコード解析においては、まず対象の型エラーのプログラムに型キャストのプログラムを挿入することによって、どの部分が型エラーの原因となっているかを見つける。次に、コン

パイラから型の情報をより得るために、インライン展開を行う。これらでどの部分が原因か、その部分にどのような型が期待されているかの情報を得ることができる。加えて、型の単一化を行うことで、期待されている部分に当てはまる関数などの型がどれなのかを知ることができる。これらの情報を GPT-3 に提供することで、型エラーを修正するパッチを生成することができる。GPT-3 では、3種類のモードが存在するため、それに合わせて3種類の方法を用い、比較を行った。OCamlプログラムを含む既存の公開データセットで検証を行った。Quickcheck を用いて、生成されたパッチがユーザの修正バージョンと同じ出力を生成するかを検証し、39%の修復率を達成した。比較研究において、型エラーのあるOCamlプログラムの自動修正において他の2つの既存研究を上回った。

(2) 部分評価を用いた型エラーデバッグ手法の開発

本研究では、型エラーがあるプログラムを部分的に実行する方法を開発した。そのアイデアは、型エラーが発生する可能性が最も高いプログラムの部分を「フリーズ」させ、それ以外の部分をすべて実行することである。型エラーのあるプログラムを部分的に実行することには2つの利点がある：1つ目の利点は、型エラーの原因と、そのエラーと衝突するコードの部分との間のテキスト的な距離が小さくなることが多いため、部分実行後に型エラーの原因を見つけやすくなることである。第二の利点は、言語が静的に型付けされていても、型付けされていないプログラムを部分的に実行できることである。これに対する我々のアプローチには3つの段階がある。第一段階では、コンパイラの型推論機能からのエラーメッセージを使用して、プログラムの一部を「フリーズ」(部分評価的にはdynamic)にする。第二段階では、型推論器を使って「フリーズ」した部分の数を減らす。第三段階では、部分評価器が部分的に「フリーズ」されたプログラムを評価する。部分評価の間、「フリーズ」にされた部分は動的なものとして扱われるため、評価されずにコードとして残り、他の部分は静的なものとして扱われて評価される。本デバッグ手法のための汎用アルゴリズム、そのアルゴリズムに使用する二つの戦略、およびOCamlのサブセットのプロトタイプを用いた有望な結果を示した。

5. 主な発表論文等

〔雑誌論文〕 計4件（うち査読付論文 4件 / うち国際共著 3件 / うちオープンアクセス 0件）

1. 著者名 Kanae Tsushima, Robert Glueck	4. 巻 1
2. 論文標題 型エラープログラムの部分評価	5. 発行年 2022年
3. 雑誌名 第24回プログラミングおよびプログラミング言語ワークショップ	6. 最初と最後の頁 -
掲載論文のDOI (デジタルオブジェクト識別子) なし	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

1. 著者名 Kanae Tsushima, Olaf Chitil, Joanna Sharrad	4. 巻 -
2. 論文標題 Type Debugging with Counter-Factual Type Error Messages Using an Existing Type Checker	5. 発行年 2020年
3. 雑誌名 The 31st symposium on Implementation and Application of Functional Languages	6. 最初と最後の頁 12pages
掲載論文のDOI (デジタルオブジェクト識別子) なし	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 該当する

1. 著者名 Ribeiro Francisco, Macedo Jos Nuno Castro, Tsushima Kanae, Abreu Rui, Saraiva Joao	4. 巻 -
2. 論文標題 GPT-3-Powered Type Error Debugging: Investigating the Use of Large Language Models for Code Repair	5. 発行年 2023年
3. 雑誌名 ACM SIGPLAN International Conference on Software Language Engineering	6. 最初と最後の頁 -
掲載論文のDOI (デジタルオブジェクト識別子) 10.1145/3623476.3623522	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 該当する

1. 著者名 Kanae Tsushima, Robert Glueck	4. 巻 -
2. 論文標題 Towards Type Debugging using Partial Evaluation	5. 発行年 2023年
3. 雑誌名 ACM SIGPLAN Workshop on Partial Evaluation and Program Manipulation	6. 最初と最後の頁 -
掲載論文のDOI (デジタルオブジェクト識別子) なし	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 該当する

〔学会発表〕 計4件（うち招待講演 0件 / うち国際学会 3件）

1. 発表者名 Kanae Tsushima
2. 発表標題 型エラープログラムの部分評価
3. 学会等名 第24回プログラミングおよびプログラミング言語ワークショップ
4. 発表年 2022年

1. 発表者名 Kanae Tsushima, Olaf Chitil, Joanna Sharrad
2. 発表標題 Type Debugging with Counter-Factual Type Error Messages Using an Existing Type Checker
3. 学会等名 The 31st symposium on Implementation and Application of Functional Languages (国際学会)
4. 発表年 2019年

1. 発表者名 Ribeiro Francisco
2. 発表標題 GPT-3-Powered Type Error Debugging: Investigating the Use of Large Language Models for Code Repair
3. 学会等名 ACM SIGPLAN International Conference on Software Language Engineering (国際学会)
4. 発表年 2023年

1. 発表者名 Kanae Tsushima
2. 発表標題 Towards Type Debugging using Partial Evaluation
3. 学会等名 ACM SIGPLAN Workshop on Partial Evaluation and Program Manipulation (国際学会)
4. 発表年 2023年

〔図書〕 計0件

〔産業財産権〕

〔その他〕

-

6. 研究組織

	氏名 (ローマ字氏名) (研究者番号)	所属研究機関・部局・職 (機関番号)	備考
--	---------------------------	-----------------------	----

7. 科研費を使用して開催した国際研究集会

〔国際研究集会〕 計0件

8. 本研究に関連して実施した国際共同研究の実施状況

共同研究相手国	相手方研究機関			
デンマーク	University of Copenhagen			
ポルトガル	University of Minho	University of Porto		