受験番号	※記入不要
氏　名	Andrew WongWeyChee
現在の専門	電気電子工学科
希望研究室	ソフトウェア工学研究室

小　論　文
（情報科学区分）

取り組みたい研究テーマ：トークン単位のバグ修復を対象としたPython向けエンドツーエンドAPRモデルの設計と評価

１．これまでの修学内容

1.1背景

卒業研究では分散型台帳に関する研究を行っています。近年ではビットコインをはじめとする分散型台帳技術が急速に発展し、経済や社会に多大な影響を与えています[2]。ビットコインは、純粋なピアツーピア型の電子通貨であり、金融機関を介さずに当事者間で直接オンライン決済を行うことを可能にする暗号通貨です[1]。現在では、ブロックチェーンや分散型台帳技術は、暗号通貨や金融分野を超えて、さまざまな分野で広く利用されるようになっています。しかし、悪意あるユーザーが特定のノードをネットワークから隔離し、台帳内の情報を隠蔽するなどの攻撃が可能であるという課題も指摘されています[2]。

1.2 既存研究

文献[2]では、悪意あるユーザーの攻撃により、分散型台帳の基盤となるアトミックブロードキャストが劣化することを想定し、劣化したアトミックブロードキャストのもとでも動作する分散型台帳を提案しています。また、アトミックブロードキャストの定義は、以下の４つの性質を満たすものとされています[2]：

1. 有効性(Validity):正しいサーバがメッセージを送信すれば、必ずそのメッセージを自分で受信します。
2. 均一合意性（Uniform Agreement）：ある正しいサーバがメッセージを受信したなら、他のすべての正しいサーバも必ずそのメッセージを受信します。
3. 均一整合性（Uniform Integrity）：メッセージは一度だけ受信され、かつ実際に送信されたメッセージのみが受信されます。
4. 均一全順序性（Uniform Total Order）：すべての正しいサーバは、メッセージを同じ順序で受信します。

[2]では、均一合意性が成り立たない場合があると指摘されており、従来の定義では、現実の攻撃に対する耐性を十分に考慮することができません。このため、[2]では最大 m 台の被害サーバの存在を許容する「劣化均一合意性（Degraded Uniform Agreement）」を導入し、正しいかつ非被害サーバ間では結果整合性を保証する新しいブロードキャストモデルを提案しました。

1.3 研究課題

本研究では、既存研究 [2] とは異なり、均一全順序性が成立しない状況を仮定します。これにより、正しいノード間でもメッセージの受信順序が異なることになり、新たな種類の劣化したアトミックブロードキャストを対象とします。このような状況では、従来のアルゴリズム[2]では台帳に一貫性のある記録順序を保つことが困難になります。そこで研究課題として、順序の不一致という困難な条件下でも結果整合性を維持できるようなアルゴリズムの設計をします。

1.4 研究方法・計画

本研究は、以下のステップで進める予定です：

1. アルゴリズムの設計
   複数ノードが異なる順序でメッセージを受信した場合でも、最終的に台帳の整合性が保たれるような新しいアルゴリズムを設計します。
2. 評価実験
   Python を対象として設計したアルゴリズムを模擬環境で実装し、故障ノード数、通信遅延、メッセージ損失率などのさまざまな条件下で、結果整合性の達成度や処理効率を評価します。

1.5 研究成果に期待されるもの

この研究によって、分散型台帳技術において悪意あるユーザーによる攻撃や通信障害が発生した場合でも、最終的にすべてのクライアントが同一の情報に到達できる、高い信頼性を持つシステムの設計に貢献できると期待されます。

２．取り組みたい内容

2.1 背景

私が取り組みたいのは、大規模言語モデル(LLM:Large Language Model)を活用したバグの予防および修正に関する研究です。2022 年 11 月 30 日、OpenAI は生成 AI チャット GPT を公開しました 。近年の大規模言語モデルの発展により、文章・画像・ソースコードの生成や自動化に加えて、ソフトウェアのバグの発見や修正も可能となってきました。しかし、LLM はバグ修復のために特化して設計また事

受験番号	※記入不要
氏　名	Andrew WongWeyChee
現在の専門	電気電子工学科
希望研究室	ソフトウェア工学研究室

小　論　文
（情報科学区分）

前学習されたものではありません［3］。そのため、LLM を応用した自動プログラム修正（APR: Automated Program Repair）の開発が重要視されています。APR とは、バグを含むソースコードを、人手を介さずに修正する技術です。しかし，既存の手順では、LLM はバグだけ修正し、置き換えることではなく、関数全体を再生成することは冗長であり、新たなバグを誘発するリスクもあります［3］。

2.2 既存研究

文献[3]では、「Toggle」と呼ばれる新しい手法が提案されています。本手法は、トークン単位でバグの位置を特定し修復するアプローチであり、従来の行単位の手法とは異なり、バグを含む関数の共通接頭辞や接尾辞を再生成せずに、関数全体を再構築することなくバグ部分のみを修正できることを目的としています。Toggle は以下の３つのコンポーネントから構成されています［3］：

1. ローカライゼーションモデル（Localization Model）:CodeT5 などのエンコーダ型 LLM を用いて、バグの開始・終了トークンを予測する。
2. バグ修正モデル（Bug Fix Model）:Code Parrot や Code Gen といったデコーダ型 LLM に対し、バグ部分のみを生成するためのプロンプトを設計する。
3. 調整モジュール（Adjustment Model）：ローカライゼーションモデルと修正モデルのトークナイザの不整合を緩和するために予測位置を微調整する。

2.3 研究課題

本研究では、既存研究［3］とは異なり、３つの個別コンポーネントを用いるのではなく、トークン単位でのバグの特定と修復を一度の前向き処理で同時に行うエンドツーエンド型の APR モデルを提案・実装します。提案手法では、関数全体を再生成することなく、バグ箇所のみを局所的に修復することを目指します。さらに、既存の Toggle 手法と比較して、修復精度を維持しながら、処理全体の推論効率が実用レベルに達するかどうかを検証します。なお、従来の Java を対象とした研究[3]とは異なり、取り組みたい研究では Python を対象とすることを前提としています。

2.4 研究方法・計画

以下のステップで研究を進める予定です：

1. データセットの準備と LLM の選定
   Python 向けのバグ付きコードと修正済みコードを含むデータセットを用意し、学習に適したモデルを選定します。
2. エンドツーエンドモデルの設計・実装
   バグの検出と修復を同時に実行可能な、トークンレベルの統合モデルを構築し、効率的なプロンプト設計および学習手法を導入します。
3. 評価と比較実験の実施
   修復精度、推論時間、処理トークン数、誤修正率など、複数の指標に基づいて、既存の Toggle 手法と比較評価を行います。

2.5 研究成果に期待されるもの

研究が遂行されれば、軽量かつ高精度な自動バグ修正支援システムの構築を目指します。この成果は、開発現場でのデバッグ効率の向上だけでなく、大規模ソフトウェアの開発コストを低減でき、プログラミング初学者への教育支援にも応用できる可能性があります。

参考文献

[1] Satoshi Nakamoto. Bitcoin: a peer-to-peer electronic cash system. Bitcoin.org. 2008. https://bitcoin.org/bitcoin.pdf.

[2] Grégory Bénassy, Fukuhito Ooshita, Michiko Inoue. Eventually consistent distributed ledger despite degraded atomic broadcast. Concurrency and Computation: Practice and Experience. Volume 35, Issue 11. 2021.

[3] Soneya Binta Hossain, Nan Jiang, Qiang Zhou, Xiaopeng Li, Wen-Hao Chiang, Yingjun Lyu, Hoan Nguyen, Omer Tripp. A Deep Dive into Large Language Models for Automated Bug Localization and Repair. Proceedings of the ACM on Software Engineering. Volume 1, Issue FSE. 2024.