



Title	コードクローン管理手法に基づく大規模ソフトウェア保守支援に関する研究
Author(s)	徳井, 翔悟
Citation	大阪大学, 2025, 博士論文
Version Type	VoR
URL	https://doi.org/10.18910/103164
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

論文内容の要旨

氏名（徳井翔梧）	
論文題名	コードクローン管理手法に基づく大規模ソフトウェア保守支援に関する研究

論文内容の要旨

近年、社会におけるソフトウェアの重要性が高まっており、社会基盤を支える大規模なソフトウェアを保守し、長期間にわたって品質を保つことが求められている。ソフトウェア保守において、潜在的な障害を未然に検出・対処することは重要な課題の1つである。しかし、限られた時間と工数の中でソフトウェアの潜在的な障害をすべて取り除くことは困難である。そのため、潜在的な障害を未然に防ぐための支援開発技術が必要とされている。

リファクタリングやバグ修正を行う際には、修正箇所に類似するコード片にも同様の問題が残っている可能性が高く、同様に修正を適用する必要がある。既存コードのコピー・アンド・ペーストによる再利用等で発生する、一致または類似した部分を持つコード片であるコードクローンはソフトウェア保守を困難にさせる要因の1つであると指摘されており、コードクローンを管理する手法が盛んに研究されている。

コードクローンを保守するために、ソースコード中からコードクローンを識別して管理する必要がある。しかし、ソースコードの規模が大きくなるとソースコード中に含まれるコードクローンも膨大な量となり、手作業でコードクローンを管理することが困難となる。そこで、ソフトウェアの保守作業を効率化させるために、ソースコードから自動的に検出し、コードクローンの存在を把握する手法が研究されている。さらに、ソフトウェアの保守性を向上させるために、コードクローンを1つの関数やクラスに集約して除去する手法が研究されている。しかし、一部の処理が変更されたコードクローンや継承関係にあるクラスなど、集約が困難なコードクローンは潜在的な障害としてソースコード中に残存するため、コードクローンを追跡して管理する手法が提案されている。

本論文では、ソフトウェア保守の品質を低下させる要因の1つであるコードクローンに着目し、その検出・追跡・集約という管理手法に関する3つの研究を実施した。

1. Cross-Polytope LSH を用いたコードクローン検出のためのパラメータ決定手法
2. コードクローン変更管理システムの開発と改善
3. コードクローン集約によるファジングの実行効率調査

1については、局所性鋭敏型ハッシュ（Locality-Sensitive Hashing、以降LSH）を用いたコードクローン検出器のパラメータ値を自動的に決定する手法を提案した。Cross-Polytope LSHは、高速かつ低メモリ消費を実現するLSHアルゴリズムの1つである。ただし、検出精度や実行時間に大きな影響を与えるパラメータの数が多く、適切なパラメータ選択にはアルゴリズムに対する深い知見が必要となる問題がある。本手法では、類似度を用いたコードクローン検出が、利用者が与えた再現率の目標値を満たしつつ可能な限り短時間で実行できることを目的として、プロジェクトの規模から適切なパラメータ値を求める線形回帰モデルを構築し、コードクローン検出対象に適したCross-Polytope LSHに与えるパラメータ値の組を決定する。これにより、CCVoltiの利用者は、高速なパラメータを選択することで、大規模なプロジェクトに対して頻繁にコードクローン検出し、修正の即時対応やコードクローンの早期発見、追跡手法への応用を可能とする。

2については、コードクローンの変更情報を開発者に通知するツールであるコードクローン変更管理システムに、一貫性のない変更を識別する機能を追加した。コードクローンに対する一貫性のない変更は数多く確認されており、その中にはバグ修正の変更が含まれることが指摘されている。既存システムでは、開発者が一貫性のない変更を手作業で識別する必要があり、特に大規模なソフトウェアの検出結果を人手で確認するのは困難である。本手法では、この課題を解決するため、意味的に一致するコードクローン検出器の導入、追跡方法の改善、一貫性のない変更の分類の追加など、4点の改善を施した。これにより、一貫性のない変更を含むクローンセットの検出と通知を実現し、大規模ソフトウェアの検出結果の確認コストを軽減した。

3については、テスト対象のソースコード内のコードクローンを集約することによる、ファジングのパス探索の実行効率について調査した。ファジングとは、実行パスに基づいて大量の入力を自動生成し、対象プログラムの潜在的な障害を検出する、ソフトウェア保守における手法の1つである。大規模なソフトウェアでは、長時間ファジングを実行してもより深い階層にある未知のパスや潜在的な障害箇所へ到達することが難しい。この課題に対処するため、基本ブロックを含むコードクローンを集約することで、そのコードクローンを含むパスが集約され、未発見のパスに到達しやすくなるという仮説のもと、コードクローン集約前後のプログラムに対するAFLの比較評価を実施した。結果として、コードクローン集約はファジングの挙動を変化させ、大規模ソフトウェアの潜在的な障害の早期発見に寄与する可能性があることを示した。

これらの研究により、大規模ソフトウェア保守における、潜在的な障害の検出を効率よく行うことができる。

論文審査の結果の要旨及び担当者

氏 名 (徳井翔梧)	
	(職) 氏 名
論文審査担当者	主査 教授 肥後 芳樹
	副査 教授 楠本 真二
	副査 教授 伊野 文彦

論文審査の結果の要旨

近年、社会基盤を支える大規模ソフトウェアの重要性が高まっており、長期的な品質確保のための保守が不可欠である。保守では潜在障害の早期検出・対処が重要だが、時間と工数が限られるため、潜在的な欠陥を未然に検出する技術が求められている。リファクタリングやバグ修正では、類似コードにも同様の欠陥が残存しやすい。コピーアンドペーストで生じるコードクローンは保守を複雑化させる主要因とされ、管理手法が盛んに研究されてきた。大規模プロジェクトではクローンの数が膨大となり手動管理は現実的でないため、自動検出や関数・クラス集約による除去が提案されている。しかし、一部改変されたクローンや継承を跨ぐクローンは集約が難しく、潜在障害になり得るため、追跡・管理手法の高度化が進められている。

本論文では、ソフトウェア保守の品質を低下させる要因の一つであるクローンに着目し、その検出・追跡・集約という管理手法に関する3つの研究を実施した。

1. Cross-Polytope LSH を用いたコードクローン検出のためのパラメータ決定手法
2. コードクローン変更管理システムの開発と改善
3. コードクローン集約によるファジングの実行効率調査

1. では、Cross-Polytope LSH を用いるコードクローン検出器のパラメータを自動決定する手法を提案した。Cross-Polytope LSH は高速・低メモリである一方、精度と実行時間に影響するパラメータが多く、適切な設定には専門知識が必要である。本手法は、プロジェクト規模を入力とする線形回帰モデルで、再現率目標を満たしつつ最短時間となるパラメータ集合を推定する。これにより、Cross-Polytope HSH を用いるコードクローン検出器の利用者は目標再現率を調整して大規模プロジェクトでも高速にクローン検出を繰り返せるため、修正の即応やクローン追跡が容易になる。実験では従来のデフォルト設定より高い検出速度を確認した。

2. では、コードクローン変更管理システムを強化し、一貫性のない変更を自動識別・通知できるようにした。コードクローンへの一貫性のない変更は多く、バグ修正も含まれるが、従来は開発者が手動で判別する必要があり、大規模プロジェクトでは困難であった。本手法は、意的・意的でない変更を自動識別し、追跡手法の改良、一貫性のない変更分類の追加など四点を実施し、該当クローンセットを自動検出・提示して確認コストを軽減した。

3. では、テスト対象ソースコード内のコードクローンを集約することでファジングのパス探索効率が向上するかを調査した。ファジングは実行パスに基づき大量の入力を自動生成し、潜在障害を検出する保守手法の一つである。しかし大規模ソフトウェアでは探索深度を十分に拡大することが難しく、長時間実行しても深層の未知パスや潜在障害に到達できない場合がある。これは複数のコードクローンが存在すると、挙動がほぼ同一であるにもかかわらず、各コードクローンが独立したパスとして冗長に探索されてしまうのが原因との仮説を立てた。つまり、基本ブロックを含むコードクローンを集約し、AFL が観測するパス総数を削減すれば未発見パスに到達しやすくなるとの仮説のもと、集約前後のプログラムを比較評価した。その結果、コードクローン集約はファジングの探索行動を変化させ、大規模ソフトウェアの潜在障害をより早期に発見できる可能性を示した。以上の成果は、大規模ソフトウェア保守における効率的な潜在障害検出に寄与する。

以上のことから本学位論文で得られた研究成果は、大規模ソフトウェアにおける潜在的な障害の検出を効率的に行うことに寄与し、ひいてはソフトウェア保守作業の効率化に寄与する。よって、本論文は、博士（情報科学）の学位論文として価値のあるものと認める。