



■キーワード

ソフトウェア ソフトウェア保守 コードクローン

■研究の概要

コードクローン(ソースコード中の類似あるいは一致しているコード片)は、主にコピーアンドペーストにより作り込まれ、ソフトウェアの保守性を低下させる原因と言われています。大規模なソフトウェアには多数のコードクローンが含まれるため、どのコードクローンにどのような対処を行うかといった戦略を考案することにより効果的な保守プロセスが可能になります。本研究は、コードクローンの管理を組み合わせ最適化問題として捉え、Search-based Software Engineering で利用される多目的最適化(複数の目的関数に対して最適な組み合わせを求める手法)を用いて解決する手法を提案します。具体的には、集約効果とコストの2つの目的変数を最適化する組み合わせを多目的最適化手法を用いて求めることで、効率的なコードクローン集約を支援します。これにより、ソフトウェアの保守作業を大域的に効率化することが期待されます。

■研究・技術のプロセス／研究事例

(1) プロセス

コードクローン集約作業支援環境を以下の3つのステップに従って実装する。

ステップ1: コードクローンの集約コストを測る指標の実装

コードクローンの集約作業にかかるコストを「コードクローン間の差異情報」および「集約前のコードクローンから集約後のコードクローンへの修正コスト」などの指標を用いて計測する。しかし、コードクローンの集約作業にかかるコストは、コードクローン間の差異によって異なる。例えば、コードが完全に一致するコードクローンに比べて、変数の型が異なるコードクローンを集約するコストは高い。従って、本研究では抽象構文木(AST: Abstract Syntax Tree)を作成し、AST情報を用いてコードクローン間の差異を検出する。さらに、コードクローン間の差異情報に基づいてコードクローンをパターンに分類し、それぞれのパターンに応じた集約コストを計測する。

ステップ2: 集約効果の大きいコードクローンの特定

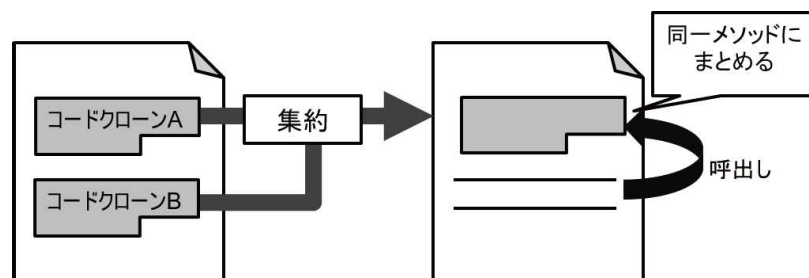
集約効果の大きいコードクローンを特定するために、コードクローンの修正履歴を用いてコードクローンの同時修正および欠陥混入を特定して集約効果を計測する。頻繁に同時修正されるコードクローンや欠陥が多数混入されたコードクローンを集約することで、ソフトウェアの保守コストを大幅に削減できるからである。

ステップ3: 多目的最適化手法による集約対象のコードクローンの識別

大規模なソフトウェアからは多数のコードクローンが検出され、また、集約の対象となるコードクローンの組み合わせによりコストや集約効果は大きく異なる。そのため、多目的最適化を用いて集約対象のコードクローンを識別する。

(2) 事例

コードクローンが多数存在するオープンソースプロジェクトに対して、ASTを用いてコードクローンの差異を分類し、それぞれの集約コストを計測した。次に、バージョン管理システムから取得した修正履歴を解析し、頻繁に同時修正が行われるコードクローンや欠陥が混入したコードクローンを特定した。最後に、集約コストと集約効果のバランスを考慮した多目的最適化手法を用いて、最適な集約対象を決定した。実際に集約対象のコードクローンを集約した結果、ソフトウェアの保守コストが削減されることが確認できた。



コードクローン集約の例

■セールスポイント

本研究で、コードクローンの管理を組み合わせ最適化問題として捉え、Search-based Software Engineeringで利用される多目的最適化を用いて解決する手法を提案します。本研究により、正確かつ、効率良い集約支援が実現できれば、ソフトウェアの保守プロセス削減ができ、インパクトが大きい研究になりえます。

ソフトウェアの保守性向上を支援するシステム
多目的最適化手法によるコードクローン集約支援環境の構築