

発表概要

双方向変換を用いたコードクローン間の選択的な更新伝播手法

相原 崇弘^{1,a)} 日高 宗一郎^{2,b)}

2019年3月18日発表

ソフトウェア開発において、コードクローンはソフトウェアの保守を困難にする原因であることが知られている。その一因として、バグを含むコード片をコピーすることにより生じたコードクローンにまたがるバグ修正等の、一貫した編集が困難であることがあげられる。コードクローンの編集の支援を行う研究としては、プログラムの編集に行われるカット、コピー、ペーストといった動作を追跡することによって、コードクローンを特定し、プログラムのトークン単位の差分に基づくコードクローン同期を実現する研究やコードクローン間で抽象構文木の比較を行い、ルールに基づく同期を実現する研究などがある。しかし、コードクローンの編集における性質を明らかにしている研究は見当たらない。本発表では、カット、コピー、ペーストといった動作を追跡することによってコードクローンを特定し、コードクローンの抽象構文木の間に保持されるべき一貫性を双方向変換言語によって記述することにより、一貫性を保持したコードクローンの系統的な編集を柔軟に行う機構の提案を行う。編集の伝播は選択的に行うことができる。本発表では、選択的に伝播するうえでの満たすべき性質を明確にしたうえで実際に提案機構がその性質を満たすことを示し、プロトタイプの実装について紹介する。本手法を用いることで、一貫性を保持したコードクローン間の更新伝播が可能であり、実装として実現することが可能である。

Presentation Abstract

Selective Update Propagation Method between Code Clones through Bidirectional Transformation

TAKAHIRO AIHARA^{1,a)} SOICHIRO HIDAKA^{2,b)}

Presented: March 18, 2019

In software development, code clones are known to cause difficulty in maintaining software. One of the reasons is that it is difficult to systematically fix bugs that straddle code clones caused by copying code fragments containing bugs. Existing works supporting edits of code clones include synchronization based on difference between tokens after identifying code clones by tracing actions such as cutting, copying, and pasting that are performed while editing a program, and synchronization based on rules by comparing abstract syntax trees between code clones. However, no research clarifying the nature of code clone editing has been found. In this presentation, we propose a mechanism that flexibly and systematically edits code clones that maintain consistency by describing the consistency to be retained between abstract syntax trees of code clones by a bidirectional transformation language after identifying code clones by tracing actions such as cutting, copying, and pasting. Propagation of editing can be done selectively. In this presentation, after clarifying desirable formal properties in selective propagation, we show that the proposed mechanism actually satisfies its properties and demonstrate prototype implementation. Thus, our method enables an implementation of update propagation while maintaining consistency of code clones.

This is the abstract of an unrefereed presentation, and it should not preclude subsequent publication.

¹ 法政大学大学院情報科学研究科
Graduate School of Computer and Information Sciences,
Hosei University, Koganei, Tokyo 184-8585, Japan

² 法政大学情報科学部
Faculty of Computer and Information Sciences, Hosei University, Koganei, Tokyo 184-8585, Japan
^{a)} takahiro.aihara.3i@stu.hosei.ac.jp
^{b)} hidaka@hosei.ac.jp