発表概要

カバレッジフィードバックを用いた自動テスト生成

實成 優馬[1,a)]　荒堀 喜貴[1]　権藤 克彦[1]

2019年3月18日発表

Grammar-based fuzzing は，コンパイラやインタプリタなどの複雑な文法構造を持った入力を受理するソフトェアの検査において有用である．しかし，Grammar-based fuzzing は実行パスによる制約と文法構造による制約の 2 つを同時に解く必要があるが，文法構造による制約は手書きで作成しなければならず，コスト面の課題が存在している．コスト面の課題に対し，深層学習による Neural Network based Fuzzing を用いて解決する研究（Learn&Fuzz）が存在する．しかし，Learn&Fuzz では適用範囲が狭いことに加えて，十分なカバレッジを達成できていない．そこで我々は，生成したテストの良し悪しをフィードバックすることで，より高いカバレッジを達成する Neural Network based Fuzzing を提案する．評価実験の結果より，従来手法が解決したコスト面のメリットを維持しつつ，従来手法より高いカバレッジを達成するテスト生成を可能にした．

Presentation Abstract

PathCondition-guided Learning-assisted Grammar-based Fuzzing

Yuma Jitsunari[1,a)]　Yoshitaka Arahori[1]　Katsuhiko Gondow[1]

Presented: March 18, 2019

Grammar-based fuzzing is an effective technique for checking security vulnerabilities in programs, such as parsers. However, most of technics of grammar-based fuzzing need a lot of domain knowledge and labor of writing input grammars. For solving this problem, Learn&Fuzz proposed the method uses Deep Learning technique to learn the structure of input grammars, and generates new inputs from the learnt model. But, Learn&Fuzz solved only cost problem, so Learn&Fuzz cannot consider about which generated inputs can inspect unexamined code blocks. Through our previous experiments using a reimplementation of Learn&Fuzz and real instruction-interpreting code, we measure the line coverage of the target program when tested by Learn&Fuzz, and the results show the coverage is surprisingly low. If we can know which inputs can examine the code blocks which have never examined, we can improve coverage and efficiency. So, we collect path conditions of target program when tested by the inputs generated from learnt model and specific which input has the possibility to inspect unexamined code blocks. We use the fine tuning algorithm to bias the inference probability of learnt model and increase the possibilities to generate specific input can examine the code blocks which have never examined.

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

[1] 東京工業大学
Tokyo Institute of Technology, Meguro, Tokyo 152–8550, Japan
[a)] jitsunari.y@sde.cs.titech.ac.jp