AI Guided Corpus Generation For Fuzzing

Xing Yu, Charles Gretton, Adrian Herrera, Alwen Tiu
u6034476@anu.edu.au, charles.gretton@anu.edu.au, adrian.herrera@anu.edu.au,
alwen.tiu@anu.edu.au

Australian National University

Problem

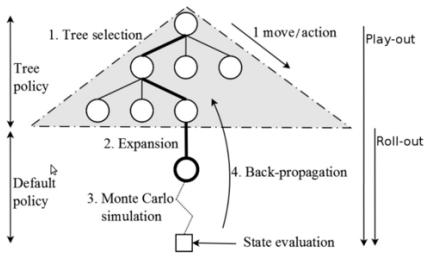
This project was inspired by generation-based fuzzing. Generation based fuzzing could make effective test cases which could trigger bugs in the target program. But using this technique, it requires a large amount of pre-work to learn the specifications and manually generate the test cases. It is even harder for highly-structured file (e.g. XML, Javascript, DtD) to reach the final stage of the target program. So, grammar-based fuzzing could be used effectively to pass the syntax parsing stage.

Monte Carlo Tree Search

UCT(*Monte Carlo Tree Search*)
UCT is a Monte-Carlo planning algorithm (Kocis and Szepesvari 2006)[2], which extends the multi-armed bandit algorithm to make decisions based on the probability.
We would like to propose a new method in seeds generation using this algorithm in a nongame domain and to do the enhancements of the Skyfire team's approach.

This algorithm including four steps.

1. *selection*. It will apply the tree policy to choose the best node recursively from the root. The score of each node based on the UCB score.
2. *expansion*. The chosen child will added to the parent node.
3. *simulation*. The simulation will run from the chosen node using the default policy to do the random rollout and collect the estimated reward.
4. *backpropogation*: The simulation result will backtracked to the root node and with the reward function updated.



References

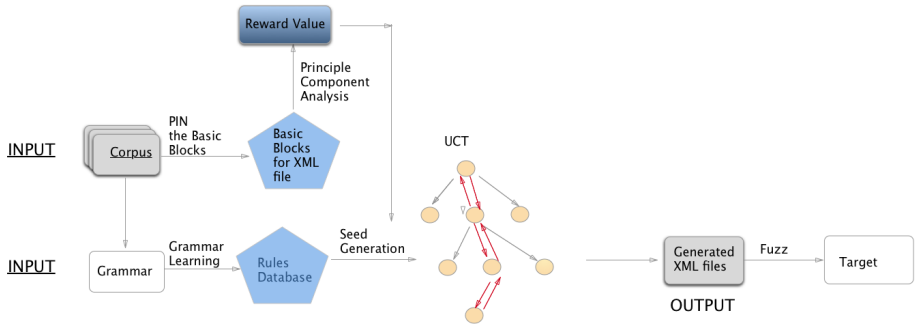
[1] Wang, Junjie, et al: *Skyfire: Data-Driven Seed Generation for Fuzzing.*, IEEE, 2017
[2] A. Zakery, A. Afrazeh and J. Dumay *Analysing and improving the strategic alignment of firms' resource dynamics*, Journal of Intellectual Capital, vol. 18, (1), pp. 217-240, 2017.
[3] M. Świechowski and J. Mańdziuk, *A hybrid approach to parallelization of monte carlo tree search in general game playing.* 2016

Acknowledgements

We thank the Skyfire team for sharing the Skyfire source code and helpul correspondence.

The Overview of Our Approach

Here is the graph demonstrated the approach we proposed:

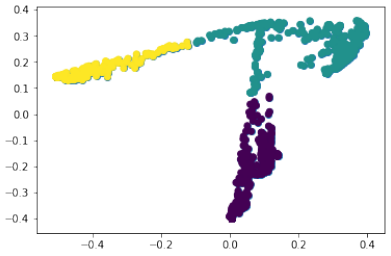


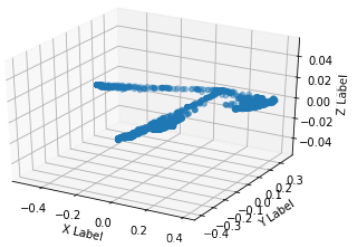
This figure presents a overview of our approach.

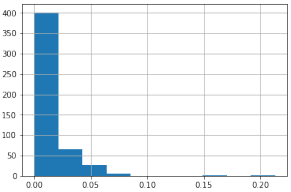
1. We adopt the learning approch used in Skyfire to parses all the file in the corpus into the given grammar and get the syntax tree with a probabilistic context-sensitive grammar (PCSG) and production rules.
2. Based on the rules stored in the database, we would like to use the Monte Carlo Tree algorithm to do the production rule expansion and generate the seed iteratively.
3. We would like to store each node with a rule and a UCT score. Before implement the UCT algorithm , we run the PIN tool to analyse the basic block executed in the libxml internal dynamically. We treat basic blocks executed as a feature vector. In order to find the novelty, Principle Component Analysis is used to reduce the dimension for each file and use that data to find the maximum nearest 5 points' distance using KNN algorithm. The distance value here is the reward value for a node.
4. In this process, we prefer to expand the node with highest UCT score to produce uncommon inputs with diverse grammar.

Generation: Reinforcement Learning Algorithm

The generation of the seeds using the reinforcement learning algorithm (UCT) to select the learned rule in PCSG.







- The 1_{st} , 2_{nd} figures presents the data provided by the experiments on Principle Component Analysis algorithm and K-means clustering algorithm.
- The 3_{rd} graph summarized the data collected during the process in finding the maximum distance of nearest 5 points in PCA analysis using KNN algorithm.