CoBOT : A Smart Software Source Code Bug Detection Tool

A static analysis tool detects bugs without having to track the entire compilation process. The heuristic-based system renders debugging of complex legacy systems much easier and practical

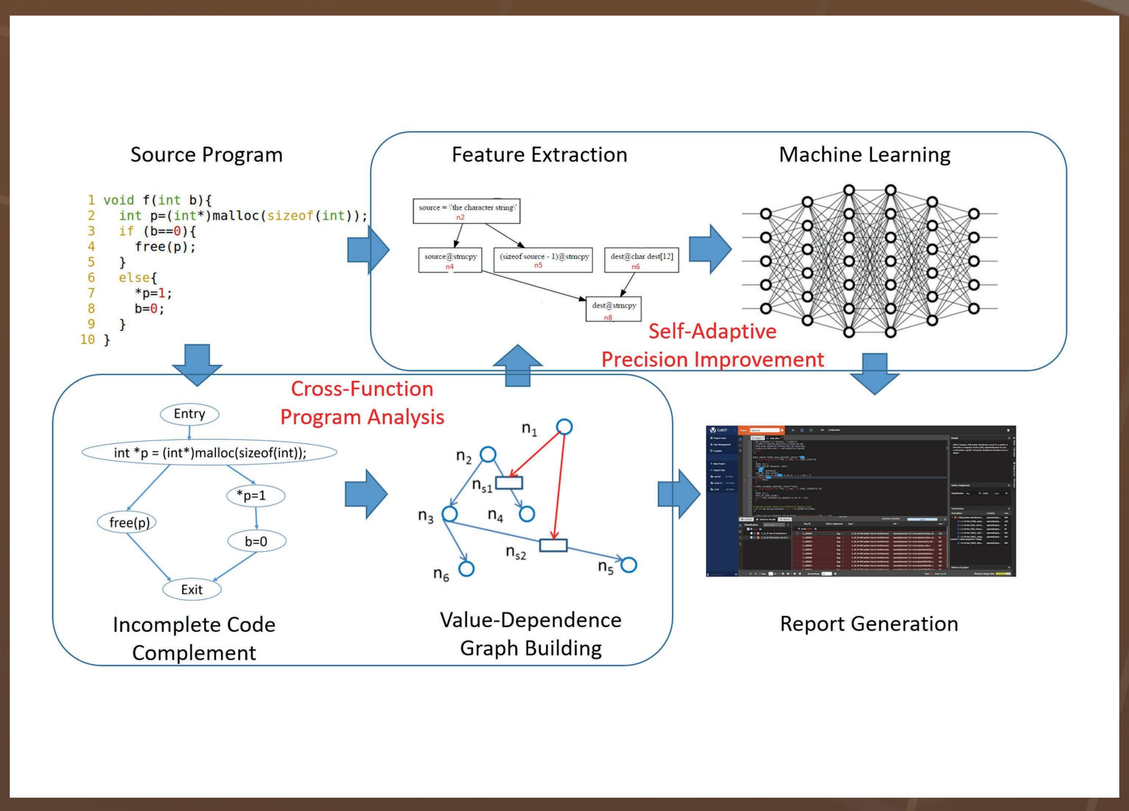
Le CoBOT: Un outil de Détection de Bogue de Code Source Logiciel Intelligent

Un outil d'analyse statique détecte les bogues sans avoir à suivre l'ensemble du processus de compilation. Le système basé sur l'heuristique rend le débogage des systèmes existants complexes beaucoup plus facile et pratique

Introduction

Compared with existing products, CoBOT has more powerful detection capabilities in the following aspects:

1. **High accuracy for detecting bugs in incomplete programs**. CoBOT adopts a series of heuristic strategies to maintain high accuracy in analyzing incomplete code, and greatly saves human efforts in using detection tools by eliminating the compiling process.
2. **High effectiveness and efficiency**. CoBOT uses our research achievement, Value-Dependence Graph, as the core model, and leverages iterative analysis techniques, so that it can scale to billion lines of code. The false positive/negative rates are both less than 20%, with an analysis speed of 2 million lines of code in an hour, leading other bug detection tools.
3. **Self-adaptive precision improvement**. Using deep learning techniques, it automatically filters possible false positives from the detection results, and keeps improving itself from previously detected programs. The more programs CoBOT analyzes, the higher accuracy of CoBOT achieves. The samples were from tool users' previous false positives and annotations.



Special Features and Advantages

- Adopt heuristic methods to complement incomplete code, and combine self-adapting engines with deep learning to improve detection accuracy and efficiency
- Detect software bugs faster and with more accuracy, which saves the overall software development cost up to 30% - 50%
- Especially effective in detecting cross-function vulnerabilities, which is difficult for existing tools based on traditional symbolic execution
- Support detection of over 1600 code rules, 200 defects and 100 security vulnerabilities

Applications

- Used for software development, testing and validation. It can also be developed for static testing of vulnerabilities, dynamic testing, unit testing aids, performance testing, etc.

Awards

Second Prize, The 2nd China Innovation Challenge, China (2017)

CWE-Compatible Certificate (2015)

Outstanding Achievement Award, National Science and Technology Innovation, China (2015)

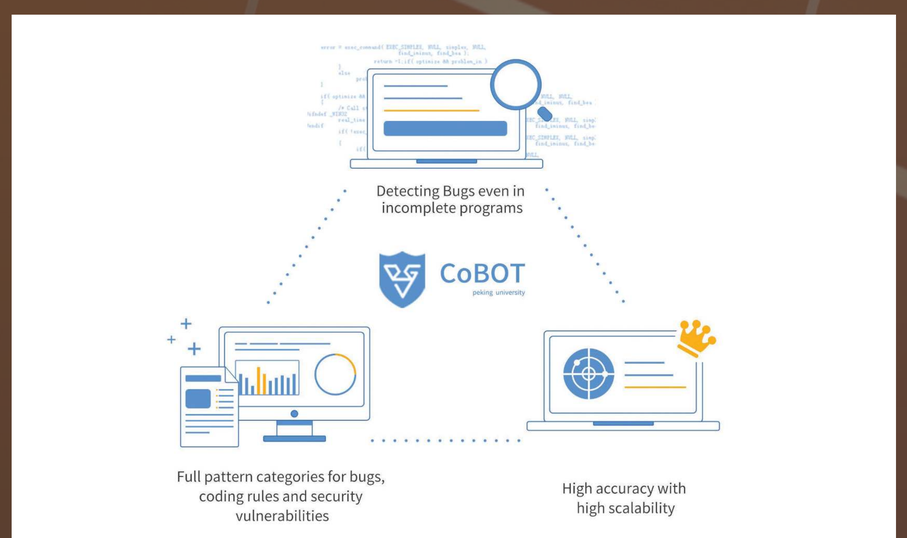
Intellectual Property

PRC Patent: 201310728361.3, 201510708750.9, 201710189131.2

Introduction

Par rapport aux produits existants, le CoBOT dispose des capacités de détection plus puissantes dans les aspects suivants:

1. **Haute précision pour détecter les bogues dans les programmes incomplets**. Adopte une série de stratégies heuristiques pour maintenir une haute précision dans l'analyse du code incomplet, et économise grandement les efforts humains dans l'utilisation des outils de détection en éliminant le processus de compilation.
2. **Haute efficacité et efficience**. Le CoBOT utilise notre performance de recherche, le graphique de la dépendance à la valeur, comme modèle de base, et tire parti des techniques d'analyse interactives, de sorte qu'il peut atteindre des milliards de lignes de code. Le taux de faux positifs/négatifs est inférieur à 20%, avec une vitesse d'analyse de 2 millions de lignes de code en une heure, conduisant à d'autres outils de détection de bogues.
3. **Amélioration de précision auto-adaptative**. En utilisant des techniques d'apprentissage en profondeur, il filtre automatiquement les faux positifs possibles à partir des résultats de la détection, et continue de s'améliorer à partir des programmes précédemment détectés. Plus les programmes CoBOT analysent, plus la précision du CoBOT est grande. Les échantillons provenaient des précédents faux positifs et annotations des utilisateurs d'outils.



Caractéristiques Particulières et Avantages

- Adopte des méthodes heuristiques pour compléter le code incomplet, et combine des moteurs auto-adaptatifs avec un apprentissage en profondeur pour améliorer la précision et l'efficacité de la détection
- Détecte les bogues logiciels plus rapidement et avec plus de précision, ce qui permet d'économiser jusqu'à 30 à 50% du coût global de développement logiciel
- Particulièrement efficace pour détecter les vulnérabilités de fonction croisées, ce qui est difficile pour les outils existants basés sur l'exécution symbolique traditionnelle
- Prise en charge de la détection de plus de 1600 règles de code, de 200 défauts et de 100 failles de sécurité

Applications

- Utilisé pour le développement de logiciels, les tests et la validation. Il peut également être développé pour les tests statiques des vulnérabilités, les tests dynamiques, les aides au test unitaire, les tests de performance, etc.

Principal Investigators

Dr. Sen MA, Dr. Qing GAO, Prof. Shikun ZHANG

National Engineering Research Center for Software Engineering

Peking University

E-mail: masen@pku.edu.cn