要 旨

意味的誤り検出ツール ASTgrep の AST 生成部の設計と実装

粕谷 彪人

ソフトウェアには，コンパイラで検出することができない誤りが存在する．この誤りを意味的誤りと言う．意味的誤りは，多くの場合ソフトウェアをリリースした後に見つかることが多い．さらに，意味的誤りはセキュリティホールの原因にもなる．そのため，ソフトウェア開発時に意味的誤りを発見するための研究が行われている．その研究の一つに，ASTgrep が存在する．

ASTgrep は，プログラムの構文的な構造に対して，意味的誤りの原因になりそうな箇所を探すツールである．そのために，検査対象プログラムの抽象構文木 (以下，AST と呼ぶ) を生成し，AST と意味的誤りの原因を表すパタンでパタンマッチする．ユーザは，パタンを作成することで，検出する誤りの種類を増やすことができる．ユーザがパタンを作成するとき，検査対象プログラムから生成した AST を見ることができるのが ASTgrep の特徴である．これによって，パタンの作成が容易になる．しかし，現在の ASTgrep には問題がある．まず，生成された AST の節点の種類名や属性名の命名規則が統一されていない．次に，生成された AST が膨大になる．さらに，AST の構造が固定されており，誤りの種類によっては，パタンが作成しづらい．そこで本研究では，AST 生成部を再設計して，これらの問題を解決した．

生成された AST の節点の種類名や属性名の命名規則が統一されていない問題については，統一した命名規則を導入することで解決した．生成された AST が膨大になる問題については，インクルードされたファイル内の定義に対応する AST の部分木を削除することで解

決した．AST の構造が固定されており，誤りの種類によってパタンが作成しづらい問題は，AST 生成後に，AST を変形する仕組みを実装することで解決した．

本研究で開発した AST 生成部の評価として，生成した AST を変形する仕組みの実行時間の比較と，出力した AST のファイルサイズの比較を行った．評価に使ったソースコードは，Apache Web サーバのソースコード中最大のソースファイルである mod_rewrite.c をである．評価の結果，実行時間は全体的にあまり変わりはなかった．AST のサイズは，生成した AST を変形する仕組みを実行したことにより，8,809,455 バイトから 6,167,078 バイトに減らすことができた．さらに，これまでパタンの作成が難しかった意味的誤りのパタンが作成できるようになったことを確認するために，AST 生成部で生成した木構造を変形して，その AST に対して，C 言語の `case` 句やラベルに関する違反のパタンを作成した．

キーワード　AST，パタンマッチ，意味的誤り検出

Abstract

Design and Implementation of AST Generator of Semantic Error Detector ASTgrep

Ayato KASUTANI

Software sometimes contains errors that can not be detected by the compiler. Such errors are called semantic errors. Semantic errors are often found after releasing software. These semantic errors may cause security problems. Therefore, many researchers are studying to find semantic errors before releasing software. ASTgrep is developed in such a research.

ASTgrep is a tool to detect semantic errors by looking for plausible places in the syntactic structure of programs. To do that, ASTgrep generates an abstract syntax tree (AST) of the target program and then matches a pattern representing the cause of a semantic error against the AST. If the users create patterns, ASTgrep can detect various kinds of errors. When the user create a pattern, the user can refer ASTs generated form the program to be checked. This makes pattern development easier. However, the current ASTgrep has problems. First, the naming rules of type names and attribute names of nodes in the AST are not uniform. Second, the generated AST is very large. Third, it is difficult to create patterns of some kinds of semantic errors due to the structure of the AST. In this study, we developed an alternative AST generator of ASTgrep to solve these problems.

We solved the first problem by employing a naming rule uniformly. We solved the second problem by deleting AST nodes representing definitions in included files. We

solved the third problem by implementing a mechanism to transform the structure of AST.

We evaluated our AST generator. We generated an AST from mod_rewrite.c, which is the largest source file among Apache web server's source files, using our AST generator and compared with the original AST generator with respect to the execution time and the AST size. As a result, the execution times were similar. The size of the AST was reduced from 8,809,455 bytest to 6,167,078 bytes by transforming the generated AST. Furthermore, in order to confirm that patterns of semantic errors which have been difficult to create can be created, we createde a patterns of a semantic error relevant to the switch-case statement and labels.

key words AST, pattern match, semantic error detection