

РЕФЕРАТ

Актуальність теми. Задача порівняння абстрактних семантичних дерев програм є однією з тих задач, що дозволяє аналізувати зміни на структурному рівні в різних версіях кодової бази програми, виконувати пошук плагіату тощо. Наявні способи використовують різноманітні модифікації алгоритму пошуку TED (tree edit distance), де також додають можливість виявлення переміщених фрагментів програм. Так як пошук переміщених фрагментів у загальному випадку є NP-повною задачею, то наявні способи використовують різноманітні спрощення чи емпіричні підходи, які враховують специфіку саме тих мов програмування, для яких були розроблені. Оскільки популярність використання функціональних мов програмування зростає, то розробка особливого способу порівняння абстрактних семантичних дерев, який враховує особливості Lisp-синтаксису та вміє знаходити переміщені фрагменти, є задачею актуальною.

Об'єктом дослідження є процес порівняння абстрактних семантичних дерев програм написаних Lisp-подібними мовами.

Предметом дослідження є способи порівняння абстрактних семантичних дерев програм написаних Lisp-подібними мовами зі знаходженням переміщених фрагментів.

Мета роботи: Метою дослідження є покращення способу порівняння абстрактних семантичних дерев, з урахуванням особливостей Lisp-синтаксису та його програмна реалізація.

Наукова новизна:

Вперше запропоновано спосіб порівняння абстрактних семантичних дерев, який полягає в порівнянні абстрактних семантичних дерев (АСД) програм, написаних на Lisp-подібних мовах, відрізняється від існуючих врахуванням особливостей синтаксису Lisp-подібних мов, виконанням операції хешування АСД, маркуванням вузлів АСД всього лише на три види, обмеженням на вибір та форму можливих переміщених фрагментів, встановленням оригінальної пріоритетності фрагментів для пошуку, і дозволяє досягнути порівняння таких дерев з одночасним

знаходженням переміщених фрагментів без повного перебору усіх можливих варіацій піддерев.

Практична цінність Запропонований спосіб може бути використаний як основа для створення ефективних за часом роботи засобів, які будуть порівнювати на структурному рівні тексти програм написаних на Lisp-подібних мовах на співпадіння, а також знаходити переміщені фрагменти коду.

Апробація роботи. Основні положення і результати роботи були представлені та обговорювались на XV науковій конференції магістрантів та аспірантів «Прикладна математика та комп'ютинг» ПМК-2022 (Київ, 16-18 листопада 2022 р.). Алгоритми способу порівняння програм, написаних Lisp-подібними мовами, на основі абстрактних семантичних дерев опубліковані у науковому фаховому виданні “Комп'ютерно-інтегровані технології: освіта, наука, виробництво” №49, що індексується в міжнародних базах даних Index Copernicus Journal Master List, Open Academic Journals Index, Academic Resource Index ResearchBib, Rootindexing, Information Matrix for the Analysis of Journals.

Структура та обсяг роботи. Магістерська дисертація складається з вступу, чотирьох розділів та висновків.

У вступі подано загальну характеристику роботи, зроблено оцінку сучасного стану проблеми, обґрунтовано актуальність напрямку досліджень, сформульовано мету і задачі досліджень, показано наукову новизну отриманих результатів і практичну цінність роботи.

У першому розділі розглянуто наявні способи порівняння абстрактних семантичних дерев, розглянуто проблему виявлення переміщених фрагментів.

У другому розділі виділена особливість синтаксису Lisp-подібних мов й на основі цього було запропоновано спосіб порівняння програм, написаних Lisp-подібними мовами.

У третьому розділі наведено особливості реалізації засобу, який реалізовує алгоритми, які створені на основі запропонованого способу.

У четвертому розділ представлено підхід до генерації тестових вхідних файлів та описано процес та представлено результати порівняння алгоритмів створених на основі представленого способу.

У висновках представлені результати проведеної роботи.

Робота представлена на 80 аркушах, містить 4 додатки та містить посилання на список використаних літературних джерел.

Ключові слова: Lisp, абстрактне семантичне дерево, структурне порівняння, хеш-таблиця, хешування дерева.

Abstract

Actuality of theme. The task of comparing abstract semantic trees of programs is one of those tasks that allows analyzing changes at the structural level in different versions of the program's code base, performing a plagiarism search, etc. Existing methods use various modifications of the TED (tree edit distance) search algorithm, which also add the possibility of detecting moved program fragments. Since the search for moved fragments is, in general, an NP-complete problem, existing methods use various simplifications or empirical approaches that take into account the specifics of the programming languages for which they were developed. Since the popularity of using functional programming languages is growing, the development of a special technique of comparing abstract semantic trees, which takes into account the features of Lisp syntax and knows how to find moved fragments, is an urgent task.

Object of research is the process of comparing abstract semantic trees of programs written in Lisp-like languages.

Subject of research is technique for comparing abstract semantic trees of programs written in Lisp-like languages with the finding of moved fragments.

Purpose: The purpose of the research is to improve a technique for comparing abstract semantic trees, taking into account the features of Lisp syntax and its software implementation.

Scientific novelty

For the first time proposed a technique of comparing abstract semantic trees , which consists in comparing abstract semantic trees (AST) of programs written in Lisp-like languages, differs from existing ones by taking into account the features of the syntax of Lisp-like languages, performing an AST hashing operation, marking AST nodes by only three categories, limiting the choice and form of possible moved fragments, establishing the original priority of fragments for search, and allows to achieve the comparison of such trees with the simultaneous finding of moved fragments without a complete search of all possible variations of subtrees.

Practical value. The proposed technique can be used as a basis for creating time-efficient tools that will compare at the structural level the texts of programs written in Lisp-like languages for coincidence, as well as find moved code fragments.

Testing of work.

Main state and results of work were presented and discussed on XV scientific conference of masters and postgraduates “Applied math and computation” PMK-2022 (Kyiv, 16-18 November 2022). Algorithms for comparing programs written in Lisp-like languages based on abstract semantic trees are published in the scientific publication “Computer-Integrated Technologies: Education, Science, Production” No. 49, which is indexed in the international databases of the Index Copernicus Journal Master List, Open Academic Journals Index, Academic Resource Index, ResearchBib, Rootindexing, Information Matrix for the Analysis of Journals.

Structure and scope of work. Master’s dissertation consists with introduction, four chapters and conclusions.

In *introduction* filed general characteristic of work, estimated modern state of the problem, justified actuality and vector of the research, formulated purpose and tasks of research and showed practical value of work.

In *first chapter* the available techniques of comparing abstract semantic trees are considered, the problem of detecting moved fragments is considered.

In *second chapter* a special feature of the syntax of Lisp-like languages was highlighted, and based on this, a method of comparing programs written in Lisp-like languages was proposed..

In *third chapter* the features of the implementation of the tool, which implements the algorithms created on the basis of the proposed method, are given.

In *fourth chapter* the approach to the generation of test input files is presented, the process is described, and the results of the comparison of algorithms created on the basis of the presented technique are presented.

In *conclusions* presented results of the work.

The work is presented on 80 pages, contains 4 additions and used literature list.

Keywords: Lisp, abstract semantic tree, structural comparison, hash-table, tree structure hashing.