Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университетимени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

Классификация известных методов определения заимствований в исходных кодах программ

Студент: Чепиго Д.С., ИУ7-54Б

Руководитель: Майков К.А.

Москва, 2022 г.

Цель и задачи

Цель работы — провести обзор существующих алгоритмических реализаций известных методов определения заимствований в исходных кодах программ.

Задачи:

— провести анализ предметной области и обзор существующих решений;

— установить критерии принадлежности программного кода к категории плагиата;

— сформулировать критерии сравнения методов;

— классифицировать существующие методы определения заимствований в исходных кодах программ.

Анализ предметной области

Области применения определения заимствований в исходных кодах программ:

— сфера образования;

— соревнования по программированию;

— разработка программного обеспечения.

Рассмотрены следующие методы:

— метод сравнения текста;
— метод сравнения токенов;
— метод сравнения метрик;
— метод сравнения деревьев;
— метод низкоуровневого сравнения кода.

Метод сравнения текста

Заключается в сравнении текстовых представлений программ на основе таких метрик, как:

— расстояние Жаккара;

— расстояние Джаро-Виклера;

— расстояние Левенштейна;

— Колмогоровская сложность.

Метод сравнения токенов

Метод основан на преобразовании программы в токены.

Процесс токенизации:

— каждому оператору присваивает уникальный идентификатор;

— по полученным идентификаторам строится строка;

— полученные строки сравниваются любым доступным способом.

Метод сравнения метрик

Метод дополняет алгоритмы токенизации. На их основе он определяет метрики, например:

— количество условных конструкций;

— количество используемых циклов;

— количество глобальных переменных.

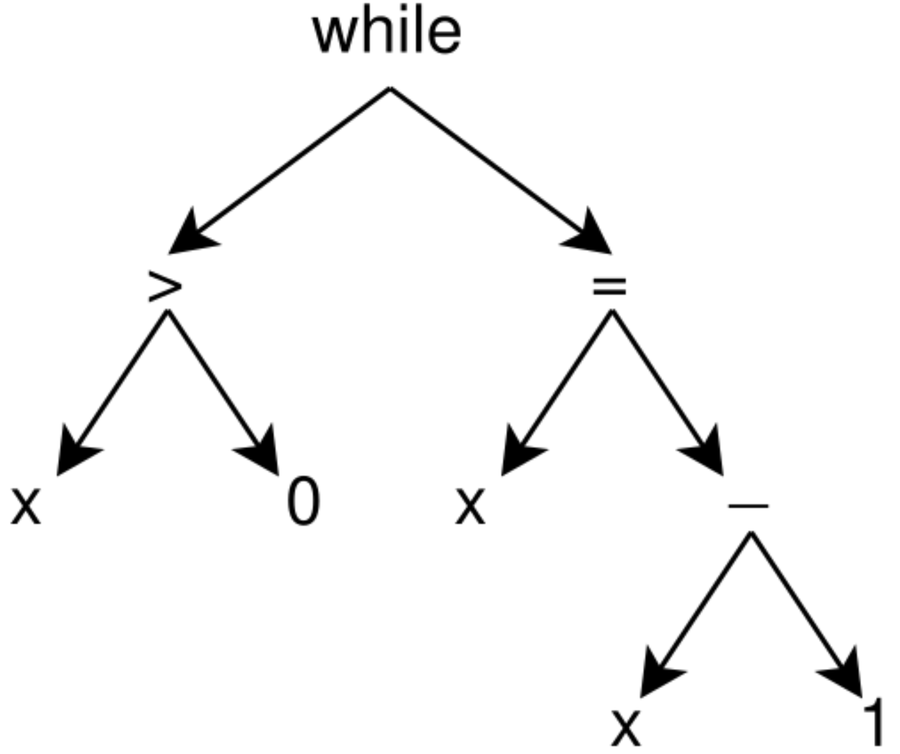
Метод низкоуровневого сравнения кода

Данный метод сравнивает код после этапа компиляции или интерпретации.

В зависимости от специфики языка программирования, такой подход может показывать разную точность.

Метод сравнения деревьев

Подход основан на представлении кода в виде абстрактного синтаксического дерева. Таким образом, метод состоит из двух этапов: построение деревьев и их анализ, любым доступным способом.



Пример абстрактного синтаксического дерева для выражения while (x > 0) x = x - 1.

Критерии принадлежности кода к плагиату

1. Программный код скопирован без каких-либо изменений.
2. Код скопирован с «косметическими» заменами идентификаторов.
3. Код может включать заимствования второго типа и модифицирован путем добавления, редактирование или удаления его фрагментов или бесполезных участков кода. Также возможны изменения порядка, не влияющие на логику самой программы.
4. Программа некоторым образом переписана с общим сохранением логики работы и функциональности, однако синтаксически она может абсолютно отличаться от оригинала.

Классификация существующих методов определения заимствований в исходных кодах программ

	Поддержка разных языков	Замена комментариев и пустых строк	Изменение имен, типов данных	Изменение порядка выполнения кода	Выделение частей кода в функции	Средняя точность на небольшом объеме	Низкоуровневое сравнение
Текст	Нет	100 %	80 %	76 %	65 %	80 %	Нет
Токены	Да	100 %	92 %	87 %	74 %	88 %	Нет
Метрики	Да	100 %	87 %	95 %	72 %	89 %	Нет
Деревья	Да	100 %	93 $	91 %	60 %	84 %	Нет
Низкоуровневый код	Да	99 %	100 %	85 %	80 %	91 %	Да