

УДК 621.398

## СИНТАКСИЧЕСКИЙ АНАЛИЗАТОР КОДА ЯЗЫКА C#

Р. Е. ГРЕЧУХА

Научный руководитель Э. И. ЯСЮКОВИЧ, канд. техн. наук, доц.  
БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ

При обучении программированию довольно сложно проверить оригинальность работы, т.к. имеется большое количество работ студентов, часто довольно объемных, и проверка вручную займет много времени.

Целью работы является создание инструмента для анализа программного кода на языке C# и определения совпадающего кода (плагиата).

Программное обеспечение (ПО) разработано с использованием платформы Roslyn от Microsoft. Названная платформа предназначена для синтаксического анализа исходных кодов на языках C# и Visual Basic, и позволяет, кроме выполнения анализа, проверить код на соответствие определенным правилам; также возможно создать модуль позволяющий автоматически исправлять найденные проблемы.

Различают атрибутивные и структурные подходы для оценки близости программ. К атрибутивным относятся методы, использующие численное выражение каких-либо атрибутов программы, таких как количество операторов. Недостатком этих методов является то, что они плохо описывают логическую часть программы и разные программы могут получить одинаковые метрики. Поэтому, для анализа был выбран структурный метод поиска заимствований.

В структурном методе исходная программа сначала разбивается на классы и представляется в виде последовательности лексем языка программирования, при этом учитываются только операторы и ключевые слова языка; это необходимо для сохранения только существенных для анализа деталей кода программы и игнорирования несущественных, таких как имена переменных и т. п. Получившаяся последовательность кодов операторов (токенов) сравнивается с последовательностью из другой программы с помощью метода поиска наибольшей общей подпоследовательности LCS – longest common subsequence. Сравнение проводится следующим образом: первый оператор одной последовательности токенов сравнивается с последним оператором другой последовательности. Если операторы одинаковы, то счетчик совпадений увеличивается на единицу. Далее первая последовательность сдвигается и снова происходит поэлементное сравнение.

Для каждой итерации запоминается счетчик совпадений и длина непрерывного участка совпавших токенов.

В перспективе возможно добавление новых, более ресурсоемких методов анализа, таких как сравнение графов исполнения; возможности групповой проверки проектов и автоматического выявления плагиата на основе задаваемых метрик.