

## ОПТИМИЗАЦИЯ АЛГОРИТМА ЛЕКСИЧЕСКОГО АНАЛИЗА ДИНАМИЧЕСКИ ФОРМИРУЕМОГО КОДА

Динамически формируемый код – это код, который может быть получен и использован внутри другого кода при помощи строковых операций, таких как конкатенация, циклы, замена подстроки. Пример такого кода – запросы SQL, которые составляются динамически в языках более общего назначения (C#, PHP и др.). Для поддержки такого кода в IDE необходима подсветка синтаксиса и диагностика ошибок [1]. Целью этой работы является увеличение производительности этих функций за счет оптимизации алгоритма лексического анализа.

Поток лексем при анализе динамически формируемого кода, в отличие от классического лексического анализа, нелинеен, что делает невозможным непосредственное применение существующих подходов. Одно из возможных решений [2] заключается в применении операции композиции [3] к двум конечным преобразователям: один из них построен по регулярной аппроксимации множества значений динамически генерируемого кода, второй является лексическим анализатором. Данная работа выполнена в рамках проекта YaccConstructor [4, 5] лаборатории языковых инструментов JetBrains, в котором реализовано такое решение [2], однако, производительность оказалась недостаточной. В работе [3] представлен алгоритм композиции, который обладает лучшей производительностью. Таким образом, необходимо реализовать алгоритм и сравнить её с существующей.

Алгоритм был реализован и в результате измерений, проведенных на заранее подготовленных регулярных аппроксимациях и лексических анализаторах, построенных по реальным описаниям соответствующих языков, были получены результаты, представленные в таблице 1.

**Таблица 1.** Сравнение производительности алгоритмов композиции конечных преобразователей

Тест	Время работы исходного алгоритма (мс)	Время работы нового алгоритма (мс)	Ускорение (раз)
Calc	0.23	0.07	3
Calc	0.33	0.08	4
TSQL	13.57	1.31	10
TSQL	17.65	1.56	11

Результаты проведённых измерений показывают существенный прирост производительности. В дальнейшем необходима проверка производительности алгоритма на реальных данных и его интеграция в проект YaccConstructor.

### ЛИТЕРАТУРА:

[1] Semen Grigorev, Ekaterina Verbitskaia, Andrei Ivanov, Marina Polubelova, and Ekaterina Mavchun. 2014. String-embedded language support in integrated development

- environment. In Proceedings of the 10th Central and Eastern European Software Engineering Conference in Russia (CEE-SECR '14). ACM, New York, NY, USA, , Article 21 , 11 pages.
- [2] Полубелова М.И. Лексический анализ динамически формируемых строковых выражений // Дипломная работа кафедры системного программирования СПбГУ. — 2015. — URL: <http://se.math.spbu.ru/SE/diploma/2015/bmo/444-Polubelova-report.pdf>.
- [3] Mehryar Mohri. Weighted Automata Algorithms. In Manfred Droste, Werner Kuich, and Heiko Vogler, editors, Handbook of Weighted Automata. Monographs in Theoretical Computer Science, pages 213-254. Springer, 2009.
- [4] Сайт проекта YaccConstructor.— URL: <http://yaccconstructor.github.io>.
- [5] Кириленко Я.А. Григорьев С. В. Авдюхин Д. А. Разработка синтаксических анализаторов в проектах по автоматизированному реинжинирингу информационных систем. — 2013. — С. 94–98. — URL: [http://ntv.spbstu.ru/fulltext/T3.174.2013\\_11.PDF](http://ntv.spbstu.ru/fulltext/T3.174.2013_11.PDF)