УДК 004.4'412
doi: 10.21685/2587-7704-2025-10-1-5

Open Access | RESEARCH ARTICLE

Лексический анализ в процессе компиляции: принципы и методы

Даниил Вениаминович Крутяков
Пензенский государственный университет, Россия, г. Пенза, ул. Красная, 40
Krutyakov231@yandex.ru

Георгий Гочаевич Кеделидзе
Пензенский государственный университет, Россия, г. Пенза, ул. Красная, 40
Gioman5817@gmail.com

Аннотация. Направлена на исследование и понимание принципа работы лексического анализа в процессе компиляции, а также его основных методов и принципов. Лексический анализ обеспечивает разбиение исходного кода на последовательность токенов, которые затем используются на следующих этапах компиляции. Рассматриваются принципы работы лексического анализатора и его роль в процессе компиляции программ.

Ключевые слова: лексический анализ, токены, компилятор, лексер, регулярные выражения, конечные автоматы

Для цитирования: Крутяков Д. В., Кеделидзе Г. Г. Лексический анализ в процессе компиляции: принципы и методы // Инжиниринг и технологии. 2025. Т. 10 (1). С. 1–3. doi: 10.21685/2587-7704-2025-10-1-5

Lexical analysis in the compilation process: principles and methods

Daniil V. Krutyakov
Penza State University, 40 Krasnaya Street, Penza, Russia
Krutyakov231@yandex.ru

Georgy G. Kedelidze
Penza State University, 40 Krasnaya Street, Penza, Russia
Gioman5817@gmail.com

Abstract. The article is aimed at researching and understanding the principle of lexical analysis in the compilation process, as well as its main methods and principles. Lexical analysis provides a breakdown of the source code into a sequence of tokens, which are then used in the next stages of compilation. The article discusses the principles of the lexical analyzer and its role in the compilation of programs.

Keywords: lexical analysis, tokens, compiler, lexer, regular expressions, finite automata

For citation: Krutyakov D.V., Kedelidze G.G. Lexical analysis in the compilation process: principles and methods. *Inzhiniring i tekhnologii = Engineering and Technology*. 2025;10(1):1–3. (In Russ.). doi: 10.21685/2587-7704-2025-10-1-5

Лексический анализ является важным этапом в компиляции программы, так как он позволяет преобразовать исходный код в последовательность лексем, которые будут использованы на последующих этапах компиляции. Это важный процесс, так как именно лексический анализ формирует основу для синтаксического анализа. Важно понять, каким образом лексический анализ и его базовые методы формируют основу для работы компилятора. Данный подход может быть ориентирован на тестирование новых алгоритмов с синтаксическими сложными конструкциями.

Основная задача лексического анализа – разбить входной текст, состоящий из последовательности одиночных символов, на последовательность слов, или лексем, т.е. выделить эти слова из непрерывной последовательности символов [1]. Лексический токен – это последовательность символов, которую можно рассматривать как единицу в грамматике языков программирования [2].

Лексический анализ основывается на ряде фундаментальных принципов, определяющих его точность и достоверность. Для того чтобы понять, как осуществляется лексический анализ, необходимо в первую очередь обратиться к грамматике языка программирования, который мы собираемся анализировать. Грамматикой называется формальное описание синтаксиса языка программирования [3]. Грамматика определяет синтаксические правила, по которым формируются допустимые конструкции языка, а также описывает, какие символы и последовательности могут встречаться в коде. Данный пример грамматики приведен на рис. 1.

Символ	Тип лексемы
VAR	Ключевое слово
LOGICAL	Ключевое слово
BEGIN	Ключевое слово
END	Ключевое слово
IF	Оператор
THEN	Оператор
ELSE	Оператор
END_IF	Оператор
READ	Оператор
WRITE	Оператор
:	Оператор
;	Оператор
=	Оператор
(	Оператор
)	Оператор
,	Оператор
AND	Бинарный оператор
OR	Бинарный оператор
EQU	Бинарный оператор
NOT	Унарный оператор
Буква	Идентификатор
Цифра	Значение

Рис. 1. Грамматика языка

Понимание грамматики позволяет лексическому анализатору точно идентифицировать, какие лексемы следует выделять из входной последовательности символов и как их классифицировать в соответствии с определенными типами токенов.

Другим принципом является разграничение лексического и синтаксического анализатора. Лексический анализатор отвечает только лишь за разбиение входного потока символов на токены, в то время как синтаксический анализатор обрабатывает те самые токены для построения синтаксического дерева. Это разделение позволяет оптимизировать каждый из этапов компиляции, делая весь процесс более понятным.

Рассмотрим пример, целью которого является выделение лексем из непрерывной последовательности символов исходного кода. В листинге 1 показан код, подлежащий анализу.

Листинг 1

```
VAR a, b, c, d : LOGICAL;
BEGIN
   READ(a, b);
   IF a THEN
      c = NOT b;
   ELSE
      c = a EQU (b AND 1) OR 0;
   END_IF;
   WRITE(c);
END.
```

Для решения такого кода анализатор применяет несколько ключевых методов. Одним из таких методов является использование регулярных выражений. Каждая разбираемая лексема может быть описана с помощью регулярных выражений. К примеру, переменные могут быть представленными определенной последовательностью, которые начинаются с букв или символов. Операторы или ключевые слова могут быть описаны аналогичным образом с использованием похожих шаблонов. Такие ключевые слова, как VAR, LOGICAL, BEGIN, END, можно описать как строки, точно совпадающие с их значением.

Одним из важных инструментов также является использование конечных автоматов. Конечный автомат представляет собой математическую модель, которая составит из конечного числа состояний и переходов между ними в ответ на входные символы. Существуют детерминированный конечный автомат (ДКА) и нетерминированный конечный автомат (НКА). ДКА – это такой тип автомата, который для каждого состояния и символа входа имеет только один возможный переход. НКА – это такой тип автомата, где у каждого состояния может быть несколько возможных переходов. В случае лексического анализа КА используется для того, чтобы корректно обрабатывать все символы исходного кода и делать переходы состояний в зависимости от того, какие лексемы встретились у него на пути. Например, для строки *c* = NOT *b* ДКА может последовательно обработать символы, распознав переменную *c*, оператор =, ключевое слово NOT и переменную *b*.

Для повышения надежды также важно обрабатывать все возможные ошибки. Ошибки могут появиться в случае некорректных символов или ошибок в грамматике. Лексический анализатор должен сразу обнаруживать такие ошибки и сообщать о них.

Таким образом, применяя все ключевые методы, удалось создать корректно работающий лексический анализатор. Результат продемонстрирован на рис. 2.

```
Ключевые слова: BEGIN, END, LOGICAL, VAR
Идентификаторы: a, b, c, d
Операторы: (, ), ,, :, ;, =, ELSE, END_IF, IF, READ, THEN, WRITE
Унарные операторы: NOT
Бинарные операторы: AND, EQU, OR
Константы: 0, 1
Неизвестные лексемы:
```

Рис. 2. Результат работы анализатора

Так, можно сделать вывод, что при применении регулярных выражений, конечных автоматов и обработки ошибок можно сделать качественный лексический анализатор. Корректно работающий лексический анализатор сильно упрощает синтаксический анализ, так как верно определяет все лексемы, составленные ранее из грамматики языка. В целом, лексический анализ по праву считается одним из ключевых этапов в процессе компиляции. Лексический анализ действительно является основой для дальнейшей работы компилятора, и правильная реализация этих принципов и методов является залогом успешного выполнения программы.

Список литературы

1. Лексический анализ // Citforum. URL: https://citforum.ru/programming/theory/serebryakov/3.shtml
2. Introduction to Lexical Analysis // GeeksforGeeks. URL: https://www.geeksforgeeks.org/introduction-of-lexical-analysis/
3. Лексический анализ и его значение // Интуит. URL: https://intuit.ru/studies/courses/27/27/lecture/827?page=2

References

1. Lexical analysis. *Citforum*. Available at: https://citforum.ru/programming/theory/serebryakov/3.shtml
2. Introduction to Lexical Analysis. *GeeksforGeeks*. (In Russ.). Available at: https://www.geeksforgeeks.org/introduction-of-lexical-analysis/
3. Lexical analysis and its meaning. *Intuit*. (In Russ.). Available at: https://intuit.ru/studies/courses/27/27/lecture/827?page=2

Поступила в редакцию / Received 10.03.2025

Принята к публикации / Accepted 28.03.2025