

# Лабораторная работа №5

## «Лексический распознаватель»

Скоробогатов С. Ю., Коновалов А. В.

29 марта 2016

### 1 Цель работы

Целью данной работы является изучение использования детерминированных конечных автоматов с размеченными заключительными состояниями (лексических распознавателей) для решения задачи лексического анализа.

### 2 Исходные данные

Пусть лексическая структура модельного языка состоит из шести лексических доменов:

1. пробелы — непустые последовательности пробельных символов (пробел, горизонтальная табуляция, маркеры конца строки);
2. идентификаторы — непустые последовательности латинских букв и десятичных цифр, начинающиеся с буквы;
3. целочисленные литералы — непустые последовательности десятичных цифр;
4. ключевые слова (варианты ключевых слов перечислены в таблицах 1, 2 и 3);
5. знаки операций (варианты знаков операций перечислены в таблицах 1, 2 и 3);
6. комментарии или строковые литералы (варианты перечислены в таблицах 1, 2 и 3).

Чтобы не усложнять лексический анализатор, разрешим идентификаторам примыкать справа к целочисленным литералам.

### 3 Задание

Выполнение лабораторной работы состоит из пяти этапов:

1. описание лексических доменов модельного языка в виде регулярных выражений;
2. построение недетерминированного лексического распознавателя для модельного языка;
3. детерминизация построенного лексического распознавателя и факторизация его алфавита;

4. построение массива обобщённых символов, матрицы переходов и массива заключительных состояний для полученного детерминированного лексического распознавателя с факторизованным алфавитом;
5. разработка лексического анализатора, работающего на основе интерпретации построенных структур данных.

Входной поток для лексического анализатора должен загружаться из файла (в ASCII). В результате работы программы в стандартный поток вывода должны выдаваться описания распознанных лексем в формате

Тег (координаты\_фрагмента): изображение\_лексем

При этом лексем, принадлежащие домену пробелов, должны отбрасываться.

Лексический анализатор должен иметь программный интерфейс для взаимодействия с парсером. Рекомендуется реализовывать его как метод `nextToken()` для императивных языков или функцию, возвращающую список лексем, для функциональных языков.

Входной файл может содержать ошибки, при обнаружении которых лексический анализатор должен выдавать сообщение с указанием координаты, восстанавливаться и продолжать работу.

В лабораторной работе разрешается использовать любой язык программирования, поддерживающий массивы с операцией доступа к элементу по индексу, работающей за константное время.

Таблица 1: Наборы ключевых слов и знаков операций по вариантам

1	<b>if, then, else, (, ),</b> комментарии ограничены знаками <b>{, }</b> могут пересекать границы строк текста.
2	<b>do, while, &lt;, &gt;, &lt;&gt;</b> , строковые литералы ограничены апострофами, не могут пересекать границы строк текста.
3	<b>begin, end, &gt;, &gt;=</b> , комментарии начинаются со знака <b>#</b> и продолжаются до конца строки.
4	<b>for, range, ==, !=</b> , строковые литералы ограничены двойными кавычками, не могут пересекать границы строк текста.
5	<b>while, wend, &lt;, &lt;=</b> , комментарии начинаются со знака <b>--</b> и продолжаются до конца строки.
6	<b>if, fi, do, od, {, }</b> , строковые литералы ограничены обратными кавычками, могут пересекать границы строк текста.
7	<b>case, break, +, ++</b> , комментарии ограничены знаками <b>(*, *)</b> могут пересекать границы строк текста.
8	<b>class, super, -, -&gt;</b> , строковые литералы ограничены двойными кавычками, могут содержать эскапе-последовательности вида <b>\x</b> , где <b>x</b> — любой символ, не могут пересекать границы строк текста.
9	<b>struct, type, =, ==</b> , комментарии начинаются с <b>//</b> и продолжаются до конца строки.
10	<b>def, val, var, [, ]</b> , строковые литералы ограничены одинарными кавычками, для включения одинарной кавычки в строку она удваивается, не могут пересекать границы строк текста.
11	<b>int, float, &lt;, &lt;&lt;</b> , комментарии ограничены знаками <b>@</b> , могут пересекать границы строк текста.
12	<b>fun, let, in, :, ::</b> , строковые литералы ограничены обратными кавычками, могут содержать эскапе-последовательности <b>\n, \\, \'</b> (остальные недопустимы), не могут пересекать границы строк текста.
13	<b>select, from, :), : (</b> , комментарии ограничены знаками <b>[ и ]</b> , не могут пересекать границы строк текста.
14	<b>ford, forward, &amp;&amp;,   </b> , строковые литералы ограничены двойными кавычками, для включения кавычки в строковой литерал она предваряется знаком <b>«\»</b> (но знак <b>«\»</b> без последующей кавычки ошибкой не является), могут пересекать границы строк текста.
15	<b>real, longreal, &gt;=, :=</b> , комментарии начинаются с <b>::</b> и продолжаются до конца строки.

Таблица 2: Наборы ключевых слов и знаков операций по вариантам

16	<code>uint8t, uint128t, (*, *)</code> , строковые литералы ограничены знаками <code>/</code> , допустимы <code>escape</code> -последовательности вида <code>\x</code> , где <code>x</code> — любой символ, могут пересекать границы строк текста.
17	<code>goto, gosub, \(\, \)</code> , комментарии начинаются с <code>\</code> и заканчиваются в конце строки.
18	<code>delete, decltype, -&gt;, &lt;-</code> , строковые литералы ограничены апострофами, могут пересекать границы строк текста, для включения апострофа внутрь строки он удваивается.
19	<code>if, elif, *, /</code> , комментарии ограничены знаками <code>/*, */</code> могут пересекать границы строк текста.
20	<code>lambda, quote, +, =, +=</code> , строковые литералы ограничены двойными кавычками, не могут пересекать границы строк текста.
21	<code>get, set,  , +</code> , комментарии начинаются с <code> +</code> и заканчиваются на <code>+ </code> , могут пересекать границы строк текста, последовательность знаков <code> +</code> внутри комментария является синтаксической ошибкой.
22	<code>signed, unsigned, ., ...</code> , строковые литералы ограничены запятыми, могут пересекать границы строк текста.
23	<code>plus, minus, &gt;=, &lt;=, ==</code> , комментарии начинаются с апострофа и продолжаются до конца строки.
24	<code>eq, neq, +-, -+</code> , строковые литералы ограничены знаками <code>:</code> , для включения двоеточия в строку оно удваивается, не могут пересекать границы строк текста.
25	<code>key, val, ~</code> , комментарии ограничены знаками <code>~</code> , могут пересекать границы строк текста.
26	<code>write, read, ^_^, ^__^</code> , строковые литералы ограничены двойными кавычками, кавычку в конце строки допустимо не ставить.
27	<code>exit, exist, !, !~</code> , комментарии начинаются со знака <code>~</code> и продолжаются до конца строки.
28	<code>set, unset, ()</code> , строковые литералы начинаются с <code>(</code> , заканчиваются на <code>)</code> , не могут содержать внутри круглые скобки и не могут пересекать границы строк текста.
29	<code>ax, eax, rax, [, ]</code> , комментарии начинаются со знака <code>;</code> или <code>#</code> и продолжаются до конца строки.
30	<code>begin, end, {, }</code> , строковые литералы ограничены знаками <code>\$</code> , допустимы <code>escape</code> -последовательности вида <code>\x</code> , где <code>x</code> — любой символ, не могут пересекать границы строк текста.

Таблица 3: Наборы ключевых слов и знаков операций по вариантам

31	<b>def, return, (, ), :,</b> комментарии ограничены знаками <code>"",</code> могут пересекать границы строк текста.
32	<b>mov, jmp, -&gt;, \$,</b> строковые литералы начинаются со знака <code>#</code> и оканчиваются знаком <code>h,</code> не могут пересекать границы строк текста.
33	<b>switch, case, :, {, },</b> комментарии начинаются со знака <code>!</code> и продолжаются до конца строки.
34	<b>open, close, &lt;&lt;, &gt;&gt;,</b> строковые литералы ограничены одинарными кавычками, для включения кавычки в строковой литерал она предваряется знаком <code>&lt;\,</code> могут пересекать границы строк текста.
35	<b>Integer, Float, ::, -&gt;, =,</b> комментарии ограничены знаками <code>{-, -},</code> могут пересекать границы строк текста.
36	<b>all, clean, :, !!,</b> строковые литералы начинаются со знака <code>#</code> и продолжаются до конца строки, и если последним символом в строке является знак <code>\,</code> то следующая строка также считается частью строкового литерала, иначе строковый литерал заканчивается.
37	<b>update, where, ==, !=,</b> комментарии начинаются со знака <code>/\</code> и продолжаются до конца строки, и если последним символом в строке является знак <code>\,</code> то следующая строка также считается частью комментария, иначе комментарий заканчивается.
38	<b>define, typedef, #, ;,</b> строковые литералы ограничены знаками <code>%,</code> допустимы escape-последовательности вида <code>\x,</code> где <code>x</code> — любой символ, могут пересекать границы строк текста.
39	<b>title, body, &lt;, &gt;, &lt;/, /&gt;,</b> комментарии ограничены знаками <code>&lt;!--, --&gt;,</code> могут пересекать границы строк текста.
40	<b>new, delete, =, (, ),</b> строковые литералы ограничены знаками <code>-,</code> для включения знака <code>-</code> в строку он удваивается, не могут пересекать границы строк текста.