

Лабораторная работа № 1.4 «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:

- пробелы — непустые последовательности пробельных символов (пробел, горизонтальная табуляция, маркеры конца строки);
- идентификаторы — непустые последовательности латинских букв и десятичных цифр, начинающиеся с буквы;
- целочисленные литералы — непустые последовательности десятичных цифр;
- ключевые слова (см. индивидуальный вариант);
- знаки операций (см. индивидуальный вариант);
- комментарии или строковые литералы (см. индивидуальный вариант).

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

3 Задание

Выполнение лабораторной работы состоит из пяти этапов:

- описание лексических доменов модельного языка в виде регулярных выражений;
- построение недетерминированного лексического распознавателя для модельного языка;
- детерминизация построенного лексического распознавателя и факторизация его алфавита;
- построение массива обобщённых символов, матрицы переходов и массива заключительных состояний для полученного детерминированного лексического распознавателя с факторизованным алфавитом;
- 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 лексемы

При этом лексемы, 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.

В реализации запрещается использование регулярных выражений.

4 Индивидуальный вариант

`do`, `while`, `<`, `>`, `<>`, строковые литералы ограничены апострофами, не могут пересекать границы строк текста.

5 Отчёт по лабораторной работе № 1.4

Отчёт выполняется в разметке Markdown по следующему шаблону:

```
% Лабораторная работа № 1.4 «Лексический распознаватель»
% 29 марта 2023 г.
% Вася Пупкин, ИУ9-63Б
# Цель работы
<переписываете цель работы из задания>
# Индивидуальный вариант
<переписываете индивидуальный вариант>
# Реализация
Лексическая структура языка — регулярные выражения для доменов:
* ...
* ...
Граф недетерминированного распознавателя:
```dot
...
...
Граф детерминированного распознавателя:
```dot
...
...
Реализация распознавателя:
```java
...
...
Тестирование
Входные данные
...
...
Вывод на `stdout`
...
...
Вывод
<пишете, чему научились>
```

В отчёте приведён лишь необходимый минимум.

[Шаблон отчёта](#)

Ваш отчёт будет конвертирован в PDF при помощи `pandoc` следующей командой:

```
pandoc \
 --pdf-engine=xelatex \
 -V 'mainfont:Liberation Serif' \
 -V 'monofont:Liberation Mono' \
 "$SOURCE" -o "$PDF"
```

**Язык реализации:** Markdown

Код решения

Из файла

Отправить