МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

# «БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА» (БГТУ им. В.Г. Шухова)

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

# Лабораторная работа №4

по дисциплине: Теория автоматов и формальных языков тема: «Регулярные языки и конечные распознаватели»

Выполнил: ст. группы ПВ-223 Дмитриев Андрей Александрович

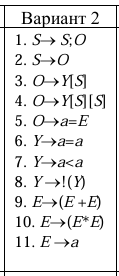
Проверил:

Рязанов Юрий Дмитриевич

Белгород 2024 г.

**Цель работы:** изучить основные способы задания регулярных языков, способы построения, алгоритмы преобразования, анализа и реализации конечных распознавателей.

Вариант 2:

****

**Задание 1.** Преобразовать исходную КС-грамматику в LL(1)-грамматику (см. варианты заданий).

1. S → S;O

2. S → O

3. O → Y[S]

4. O → Y[S][S]

5. O → a=E

6. Y → a=a

7. Y → a<a

8. Y → !(Y)

9. E → (E +E)

10. E → (E\*E)

11. E → a

Правило 1 – леворекурсивное:

1\_1. S → OS′

1\_2. S′ → ;OS′

1\_3. S′ → ϵ

Правила 3,4 и 9,10 – следует факторизовать:

3\_1. O → Y[S]O`

3\_2. O` → [S]

3\_3. O′ → ϵ

9\_0. E → (EE`

9. E` → +E)

10. E` → \*E)

Получим грамматику в LL(1):

1\_1. S → OS′

1\_2. S′ → ;OS′

1\_3. S′ → ϵ

2. S → O

3\_1. O → Y[S]O`

3\_2. O` → [S]

3\_3. O′ → ϵ

5. O → a=E

6. Y → a=a

7. Y → a<a

8. Y → !(Y)

9\_0. E → (EE`

9. E` → +E)

10. E` → \*E)

11. E → a

**Задание 2.** Определить множества ПЕРВЫХ для каждого символа LL(1) грамматики.

|  |  |
| --- | --- |
| Символ | ПЕРВ |
| S |  |
| S` |  |
| O |  |
| O` |  |
| Y |  |
| E |  |
| E` |  |

**Задание 3.** Определить множества СЛЕДУЮЩИХ для каждого символа LL(1)-грамматики.

|  |  |
| --- | --- |
| Символ | СЛЕД |
| S |  |
| S` |  |
| O |  |
| O` |  |
| Y |  |
| E |  |
| E` |  |

**Задание 4.** Определить множество ВЫБОРА для каждого правила LL(1) грамматики.

|  |  |
| --- | --- |
| Правило | ВЫБОР |
| 1\_1. S → OS′ |  |
| 1\_2. S′ → ;OS′ |  |
| 1\_3. S′ → ϵ |  |
| 2. S → O |  |
| 3\_1. O → Y[S]O` |  |
| 3\_2. O` → [S] |  |
| 3\_3. O′ → ϵ |  |
| 5. O → a=E |  |
| 6. Y → a=a |  |
| 7. Y → a<a |  |
| 8. Y → !(Y) |  |
| 9\_0. E → (EE` |  |
| 9. E` → +E) |  |
| 10. E` → \*E) |  |
| 11. E → a |  |

**Задание 5.** Написать программу-распознаватель методом рекурсивного спуска. Программа должна выводить последовательность номеров правил, применяемых при левом выводе обрабатываемой цепочки.

**Задание 6.** Сформировать наборы тестовых данных. Тестовые данные должны содержать цепочки, принадлежащие языку, заданному грамматикой, (допустимые цепочки) и цепочки, не принадлежащие языку. Для каждой допустимой цепочки построить дерево вывода и левый вывод. Каждое правило грамматики должно использоваться в выводах допустимых цепочек хотя бы один раз.

**Задание 7.** Обработать цепочки из набора тестовых данных (см. п.6) программой-распознавателем.

**Задание 8.** Построить нисходящий МП-распознаватель по LL(1)-грамматике.

**Задание 9.** Написать программу-распознаватель, реализующую построенный нисходящий МП-распознаватель. Программа должна выводить на каждом шаге номер применяемого правила и промежуточную цепочку левого вывода.

**Задание 10.** Обработать цепочки из набора тестовых данных (см. п.6) программой-распознавателем.

**Вывод:** в ходе работы изучены основные способы задания регулярных языков, способы построения, алгоритмы преобразования, анализа и реализации конечных распознавателей.