Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Лабораторная работа № 13

По дисциплине «Конструирование программного обеспечения»

На тему «Подготовка к разработке лексического распознавателя»

Выполнила:

Студентка 1 курса 10 группы

Рублевская Маргарита Владимировна

Преподаватель: Север А.С.

2024, Минск

**Вариант № 13**

func(□)+(begin;(□)\*(abs()|□+(x+));)\* □+end; — где символ □ – это обозначение пробела в вариантах заданий

3. Приведите 7 примеров цепочек символов, принадлежащих регулярному множеству.

***Пусть***

func = a, □ = b, begin = c, abs = d, x = e, end = f.

***Цепочки:***

abc

abf

abbf

abbbbf

abcbdbe

abcbdbebf

abdbe

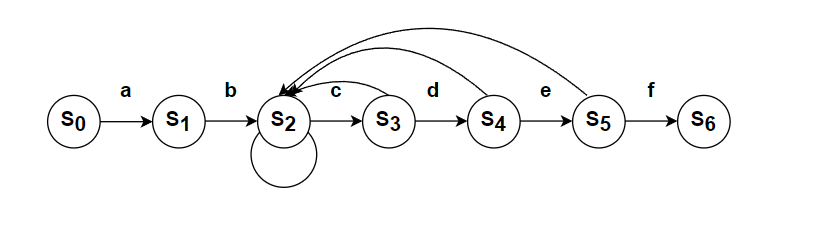
***Тогда:***

a(b) + (c; (b) \* (d () | b + (e+));)\* b + f.

***Диаграмма состояний:***

abcbdbebf→bcbdbebf→cbdbebf→bdbebf→dbebf→bebf→ebf→bf→f→

***Граф конечного автомата:***



Ответы на вопросы:

1. Алфавит I — это конечное множество символов, из которых формируются цепочки символов. Символы алфавита могут быть любыми, например, буквы, цифры или специальные символы.
2. λ– регулярное выражение, представляет множество {λ};

I\* все цепочки, которые можно образовать используя символы I

I+ все НЕПУСТЫЕ цепочки, которые можно образовать используя I

1. Язык L(I) над алфавитом I представляет собой множество всех возможных цепочек или слов, которые могут быть образованы из символов алфавита I.
2. Формальная грамматика G представляет собой набор правил, которые определяют структуру формального языка.
3. описывают процесс замены одной цепочки символов на другую в грамматике.
4. представляет собой множество всех цепочек или слов, которые можно породить или сгенерировать с помощью правил продукции данной грамматики G.
5. Форма Бэкуса-Наура — формальная система описания синтаксиса, в которой одни синтаксические категории последовательно определяются через другие категории.
6. Регулярная грамматика является формальной грамматикой, определяющей регулярный язык. Она состоит из четырех компонентов: алфавита (терминальные символы), нетерминальных символов, стартового символа и правил продукции.
7. Регулярное множество представляет собой множество строк или цепочек символов, которые могут быть порождены или распознаны с использованием регулярного выражения или конечного автомата.
8. Регулярный язык — это язык, который может быть описан с помощью регулярного выражения, регулярной грамматики или распознан с использованием конечного автомата.
9. Лексический анализ является первым этапом в компиляции или интерпретации программного кода. Он отвечает за разбиение исходного кода на лексемы или токены, которые являются минимальными значимыми элементами языка программирования.
10. Лексический анализатор является компонентом компилятора или интерпретатора, отвечающим за выполнение лексического анализа.
11. Входная информация: Поток символов исходной программы. Выходная информация: Поток токенов, где каждый токен представляет лексему вместе с ее атрибутами.
12. Последовательный лексический анализатор: распознаёт лексемы одну за другой, читая входной поток слева направо. Параллельный лексический анализатор: распознаёт лексемы одновременно, используя несколько потоков или процессов.
13. Регулярное выражение над алфавитом I — это формальное выражение, используемое для описания и поиска шаблонов в строках, составленных из символов алфавита I.
14. S – множество состояний, I – множество входных символов, Σ – множество выходных символов, s0 – начальное состояние, F – множество конечных состояний.
15. Детерминированный конечный автомат (ДКА): для каждого состояния и входного символа существует ровно один следующий переход. Недетерминированный конечный автомат (НКА): для каждого состояния и входного символа может быть несколько следующих переходов.
16. Мгновенное состояние конечного автомата — это комбинация текущего состояния автомата, входного символа и оставшейся части входной цепочки символов, на основе которой происходит переход в новое состояние.
17. Это выражение описывает, как одно состояние или конфигурация переходит в другое состояние или конфигурацию.
18. любой регулярный язык может быть задан регулярной грамматикой,  
    регулярным выражением или конечным автоматом. Регулярная грамматика описывает регулярный язык, регулярные выражения используются для работы с регулярными языками, а конечный автомат и его граф состояний могут быть использованы для распознавания и генерации регулярных языков.
19. Он определяет набор символов, из которых состоят допустимые строки или цепочки языка.
20. I\* все цепочки, которые можно образовать используя символы I

I+ все НЕПУСТЫЕ цепочки, которые можно образовать используя I

1. Язык L(I) над алфавитом I представляет собой множество всех возможных цепочек или слов, которые могут быть образованы из символов алфавита I.
2. Формальная грамматика G представляет собой набор правил, которые определяют структуру формального языка.
3. описывают процесс замены одной цепочки символов на другую в грамматике.
4. представляет собой множество всех цепочек или слов, которые можно породить или сгенерировать с помощью правил продукции данной грамматики G.
5. Система для описания синтаксических категорий через последовательное описание других синтаксических категорий
6. Регулярная грамматика является формальной грамматикой, определяющей регулярный язык. Она состоит из четырех компонентов: алфавита (терминальные символы), нетерминальных символов, стартового символа и правил продукции.
7. Регулярное множество представляет собой множество строк или цепочек символов, которые могут быть порождены или распознаны с использованием регулярного выражения или конечного автомата.
8. Регулярный язык — это язык, который может быть описан с помощью регулярного выражения, регулярной грамматики или распознан с использованием конечного автомата.
9. Лексический анализ является первым этапом в компиляции или интерпретации программного кода. Он отвечает за разбиение исходного кода на лексемы или токены, которые являются минимальными значимыми элементами языка программирования.
10. Лексический анализатор является компонентом компилятора или интерпретатора, отвечающим за выполнение лексического анализа.
11. Вход – программный код, выход – токены
12. Параллельный – многопоточный, последовательный – однопоточный
13. Регулярное выражение над алфавитом I - это формальное выражение, используемое для описания и поиска шаблонов в строках, составленных из символов алфавита I.
14. S – множество состояний, I – множество входных символов, Σ – множество выходных символов, s0 – начальное состояние, F – множество конечных состояний.
15. В детерминированном – один переход для каждого состояния
16. Мгновенное состояние конечного автомата - это комбинация текущего состояния автомата, входного символа и оставшейся части входной цепочки символов, на основе которой происходит переход в новое состояние.
17. Это выражение описывает, как одно состояние или конфигурация переходит в другое состояние или конфигурацию.
18. Читай остальные вопросы)