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»

Выполнил:

Студент 1 курса 7 группы

Ленкевич Павел Андреевич

Преподаватель: Наркевич А. С.

2024, Минск

**Отчёт**

**Вариант №9**

**Регулярное выражение:**

start(□)+ ((send+wait+show)□+ ) \* □ + stop

Примеры цепочек символов:

1. Start □ send □ wait □ show □ □ stop
2. Start □ □ stop □ □ stop
3. Start □ send □ □ □ stop
4. Start □ send □ □ wait □ □ show □ □ □ stop
5. Start □ send □ send □ □ stop
6. Start □ □ □ show □ □ show □ □ □ stop
7. Start □ □ stop

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| start | □ | send | wait | show | stop |
| a | b | c | d | t | f |

1. abcbdbtbbf
2. abbfbbf
3. abcbbbf
4. abcbbdbbtbbbf
5. abcbcbbf
6. abbbtbbtbbbf
7. abbf

Регулярное выражение после переопределения:

a(b)+ ((c+d+t)b+ )\* b+f

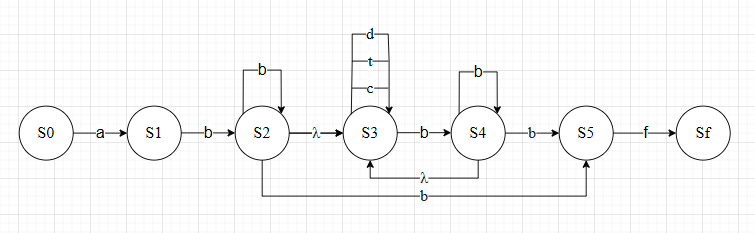
**Таблица состояний конечного автомата для выражения:**

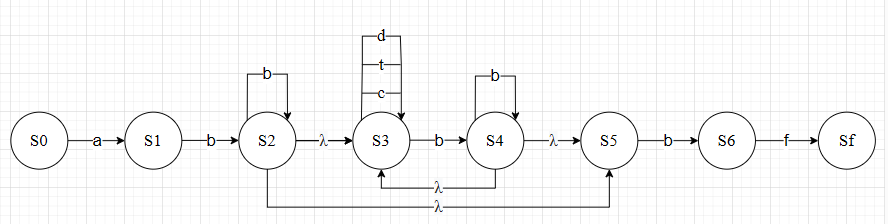
**a(b)+ ((c+d+t) b+ )\* b+f**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | a | b | c | d | t | λ | f |
| S0 | S1 |  |  |  |  |  |  |
| S1 |  | S2 |  |  |  |  |  |
| S2 |  | S5 |  |  |  | S3 |  |
| S3 |  | S4 | S3 | S3 | S3 |  |  |
| S4 |  | S4, S5 |  |  |  | S3 |  |
| S5 |  |  |  |  |  |  | Sf |
| Sf |  |  |  |  |  |  |  |

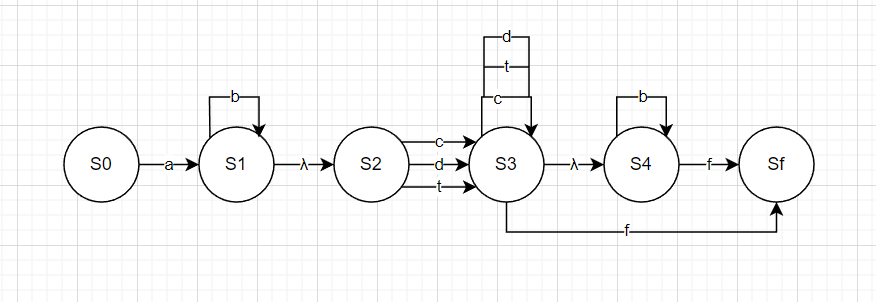
**Граф состояний конечного автомата для выражения a(b)+ ((c+d+t) b+ )\* b+f**

**(новый):**



**(новый\_v2):  
**

**(старый):**



**Ответы на вопросы:**

1. Алфавит I: Это набор символов, используемых для построения строк в формальных языках.
2. Формальная грамматика G: Система продукционных правил для генерации строк в формальном языке.
3. Язык L(G): Множество всех строк, которые могут быть порождены грамматикой G.
4. Форма Бэкуса-Наура: Нотация для выражения продукционных правил в контекстно-свободных грамматиках.
5. Регулярная грамматика: Грамматика, порождающая регулярный язык, с ограниченными правилами продукции.
6. Регулярное множество: Множество, которое может быть описано регулярным выражением.
7. Регулярный язык: Язык, который может быть порожден регулярной грамматикой.
8. Лексический анализ: Процесс преобразования последовательности символов в последовательность лексем.
9. Лексический анализатор: Программа или часть компилятора, выполняющая лексический анализ.
10. Информация для лексического анализатора: Входная - исходный код программы; выходная - последовательность лексем.
11. Лексические анализаторы: Последовательный обрабатывает входные данные по порядку; параллельный может обрабатывать входные данные одновременно в разных потоках.
12. Регулярное выражение: Формальное выражение для описания регулярного множества.
13. Детерминированный/недетерминированный автомат: Детерминированный имеет однозначно определенные переходы; недетерминированный может иметь несколько возможных переходов.
14. Мгновенное состояние автомата: Текущее состояние автомата в определенный момент времени.
15. Соотношение элементов: Регулярная грамматика порождает регулярный язык, который может быть описан регулярным выражением и представлен конечным автоматом или графом состояний.
16. \*λ, , →, ∪: Обозначения для пустой строки, звездочка Клини (повторение), стрелка (продукция), объединение.
17. S → α | β: Правило продукции, где S может быть заменено на α или β.
18. Конечный автомат M: Определяется как M = (S,I,δ,s0,F), где S - множество состояний, I - алфавит, δ - функция переходов, s0 - начальное состояние, F - множество конечных состояний.
19. (S,w) ⇒ (Sk,wk)\*: Обозначение для последовательности переходов от начального состояния S с входной строкой w к состоянию Sk с оставшейся строкой wk.