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 9 группы

Павлович Ян Андреевич

Преподаватель: Панченко О.Л.

2024, Минск

Павлович Ян Андреевич

Вариант №11

Регулярное выражение:

while(□)\* (read+seek)\*□+do{write□\* };

1. while (□)\* do { write }
2. while □, □ read+seek do { write □ }
3. while read+seek □ do { write }
4. while □ read+seek read+seek □ do { write □ }
5. while □, □, □ read+seek read+seek □ do { write □, □ }
6. while read+seek □, □, □ do { write }
7. while □, □, □ do { write □, □, □ }

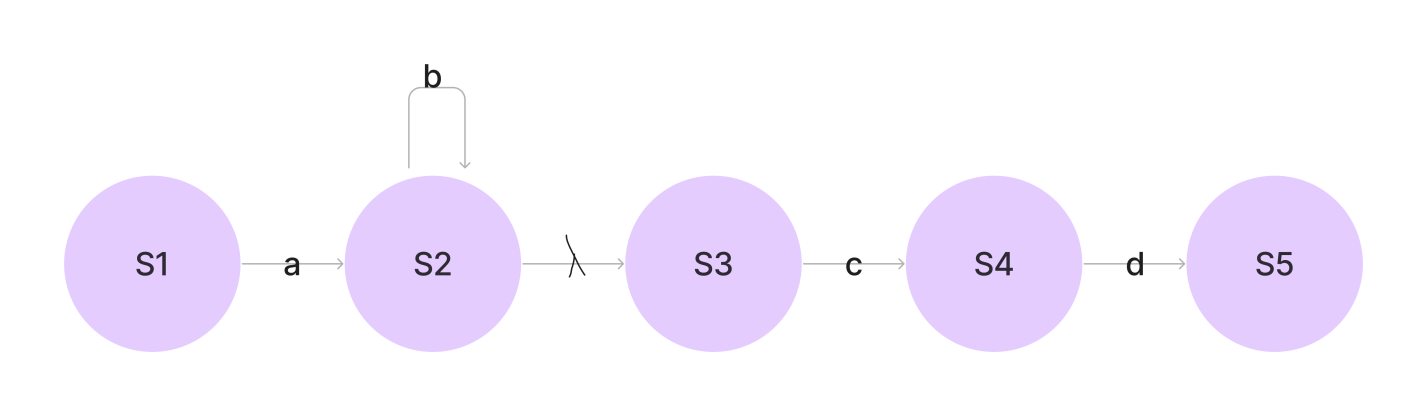
while (□)\* do { write }

Диаграмма мгновенных состояний конечного автомата для данной цепочки:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| while | (□)\* | do | { | write | } |
| a | b | c | d | e | g |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | a | b | c | d | e | g |
| S0 | S1 | ∅ | ∅ | ∅ | ∅ | ∅ |
| S1 | ∅ | S2 | ∅ | ∅ | ∅ | ∅ |
| S2 | ∅ | S2 | S3 | ∅ | ∅ | ∅ |
| S3 | ∅ | ∅ | ∅ | S4 | ∅ | ∅ |
| S4 | ∅ | ∅ | ∅ | ∅ | S5 | ∅ |
| S5 | ∅ | ∅ | ∅ | ∅ | ∅ | S6 |
| S6 | ∅ | ∅ | ∅ | ∅ | ∅ | Sf |
|  |  |  |  |  |  |  |

Граф состояний конечного автомата:



Ответы на вопросы:

1. Алфавитом I называется конечное множество символов, допустимых в языке.
2. λ – пустая цепочка в алфавите I; I – множество всех цепочек, состоящих из символов алфавита I+, исключая пустую цепочку (λ); I – множество всех цепочек, состоящих из символов алфавита I\*, включая пустую цепочку.
3. . Языком L(I) над алфавитом I называется произвольное подмножество цепочек из \* I.
4. G = <T,N,P, S> – грамматика языка (порождающая грамматика) – это четверка, где: T – множество терминальных символов, N – множество нетерминальных символов, P – множество правил (или продукций) грамматики, S – начальный символ грамматики.
5. α ⇒ β - это один шаг вывода; α ⇒ β \* - это ноль и более шагов вывода.
6. L(G) – язык, порождаемый грамматикой G, это множество всех выводимых из аксиомы грамматики терминальных цепочек.
7. Форма Бэкуса–Наура – формальная система описания синтаксиса, в которой одни синтаксические категории последовательно определяются через другие категории.
8. GIII = <T,N,P, S> – регулярные грамматики, которые имеют вид: A→α или A→αB , где A,B∈ N , \* α ∈T .
9. Множество цепочек описанных регулярным выражением называется регулярным множеством.
10. Язык программирования на уровне лексического анализа представляет собой регулярный язык (язык типа 3 иерархии Хомского).
11. Лексический анализ – первая (наиболее простая) фаза трансляции.
12. Лексический анализатор – это программа для выполнения лексического анализа(входящая в состав транслятора).
13. Входная информация: исходный код, алфавит символов; выходная: таблица лексем, таблица идентификаторов.
14. Последовательный лексический анализатор: однопоточный, линейная обработка, простота реализации. Параллельный лексический анализатор: многопоточный, параллельная обработка, сложность синхронизации.
15. Регулярные выражения над алфавитом I и языки, представляемые ими, рекурсивно определяются следующим образом: 1) ∅ – регулярное выражение, представляет пустое множество; 2) λ – регулярное выражение, представляет множество {λ} ; 3) для каждого a∈I символ a является регулярным выражением и представляет множество {a} ; 4) если p – регулярное выражение, представляющее множество P и q – регулярное выражение, представляющее множество Q , то p + q, pq, q \* являются регулярными выражениями и представляют множества: a) P ∪ Q (объединение), b) PQ (конкатенация множеств), c) P\* (итерация) соответственно. 5) + pp\* = p.
16. КА это пятерка ( , , , , ) M = S I δ s0 F , 22. где S – конечное множество состояний устройства управления; I – алфавит входных символов; δ – функция переходов, отображающая S × (I ∪{λ}) во множество подмножеств S : δ(s,i) ⊂ S,s∈S,i∈ I ; s0 ∈ S - начальное состояние устройства управления; F ⊆ S - множество заключительных (допускающих) состояний устройства управления.
17. Если δ(s,λ) = ∅ и |δ (s,a)|≤1, то конечный автомат – детерминированный (ДКА). Т.е. отсутствуют состояния, имеющие λ-переходы и для каждого состояния s и входного символа а существует не более одной дуги, выходящей из s и помеченной как а. ДКА – это автомат, который переходит из любого состояния по любому символу точно в одно состояние. Иначе - конечный автомат является недетерминированным (НКА).
18. Мгновенное описание КА – пара (s,w), где s∈S – состояние КА, \* w∈ I – неиспользованная часть входной цепочки.
19. (s,aw) > (s ′ ,w) – читается: непосредственно следует. (s(i), w(i)) >\*(s(k), w(k)) – следует.
20. любой регулярный язык может быть задан регулярной грамматикой, регулярным выражением или конечным автоматом; любой конечный автомат задает регулярный язык, а значит грамматику или регулярное выражение.