Лабораторная работа 13

По дисциплине «Конструирование программного обеспечения»

На тему «Подготовка к разработке лексического распознавателя»

Выполнил:

Студент 1 курса 9 группы

Хуторцов Кирилл Владимирович

Преподаватель: Панченко О.Л.

2024, Минск

Вариант 15

return(□)\* (begin□+(calc|print);)+ □\*end; **- регулярное выражение**

**Задание 3:**

return, □,□,□, begin, □,□, calc, λ, end;

return, □, □, begin, □,□,□,□, print, λ, end;

return, □, □, begin, □, calc, □,□, end;

return, □, begin, □,□, print, calc, □, begin, □,□, end;

return, □, begin, □,□,□, print, print, begin, □,□,□, □, end;

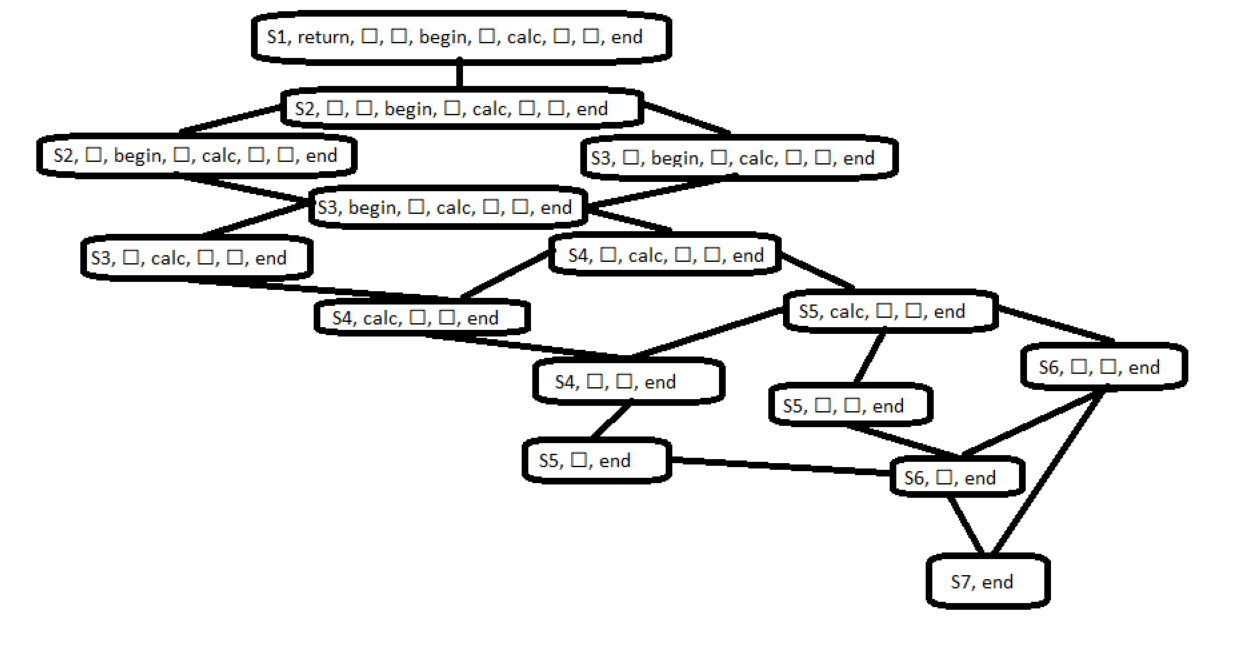
return, □, □, □, □, □, □, begin, □,□, calc, □,□, □,□,□, end;

return, □, begin, □,□,□,□, print, calc, begin, λ, end;

**Задание 4:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | return | □ | λ | begin | calc | print | end |
| S1 | S2 |  |  |  |  |  |  |
| S2 |  | S2, S3 | S3 |  |  |  |  |
| S3 |  |  |  | S4 |  |  |  |
| S4 |  | S4, S5 |  |  |  |  |  |
| S5 |  |  |  |  | S3, S6 | S3, S6 |  |
| S6 |  | S6, S7 | S7 |  |  |  |  |
| S7 |  |  |  |  |  |  |  |

**Задание 5:**





Ответы на вопросы:

Давайте разберем ваши вопросы по очереди:

1. \*\*Что такое алфавит \( I \)?\*\*

Алфавит \( I \) — это конечное множество символов.Каждый символ в алфавите используется для построения строк и языков.

2. \*\*Поясните обозначение \(\lambda\), \( I^\* \), \( I^+ \)\*\*

- \(\lambda\) — пустая строка, строка нулевой длины.

- \( I^\* \) — множество всех возможных строк (включая пустую строку) над алфавитом \( I \).

- \( I^+ \) — множество всех возможных непустых строк над алфавитом \( I \).

3. \*\*Что такое язык \( L(I) \) над алфавитом \( I \)?\*\*

Язык \( L(I) \) — это любое подмножество множества всех строк \( I^\* \).

4. \*\*Дайте определение формальной грамматике \( G \)\*\*

условная Грамматика при которой ты сам можешь создавать правила вывода свои териминлы и нетерминалы для того чтобы обЪяснить другим программистам определенный код



5. \*\*Поясните обозначение \(\alpha \Rightarrow \beta\) и \(\alpha \Rightarrow^\* \beta\) для цепочек символов \(\alpha\) и \(\beta\)\*\*

- \(\alpha \Rightarrow \beta\) означает, что \(\beta\) может быть получена из \(\alpha\) применением одного правила из \( P \).

- \(\alpha \Rightarrow^\* \beta\) означает, что \(\beta\) может быть получена из \(\alpha\) за конечное число применений правил из \( P \).

6. \*\*Что такое язык \( L(G) \), порождаемый грамматикой \( G \)?\*\*

Язык \( L(G) \), порождаемый грамматикой \( G \), — это множество всех строк из терминальных символов, которые могут быть выведены из начального символа \( S \) с помощью правил из \( P \).

7. \*\*Что такое форма Бэкуса-Наура?\*\*

Форма Бэкуса-Наура (BNF) — это нотация для задания формальных грамматик, использующая комбинацию терминальных и нетерминальных символов и правил производства.

8. \*\*Поясните понятие «регулярная грамматика»\*\*

Регулярная грамматика — это грамматика, в которой все правила производства имеют одну из двух форм:

- \( A \rightarrow aB \) или \( A \rightarrow a \), где \( A \) и \( B \) — нетерминальные символы, а \( a \) — терминальный символ (праволинейная грамматика).

- \( A \rightarrow Ba \) или \( A \rightarrow a \), где \( A \) и \( B \) — нетерминальные символы, а \( a \) — терминальный символ (леволинейная грамматика).

9. \*\*Поясните понятие «регулярное множество»\*\*

Регулярное множество — это множество строк, которое может быть распознано регулярной грамматикой или регулярным выражением.

10. \*\*Поясните понятие «регулярный язык»\*\*

Регулярный язык — это язык, который может быть описан регулярным выражением или порожден регулярной грамматикой.

11. \*\*Что такое лексический анализ?\*\*

Лексический анализ — это процесс разбора входной последовательности символов на лексемы (токены), которые являются логическими единицами языка программирования.

12. \*\*Что такое лексический анализатор?\*\*

Лексический анализатор (сканер) — это программа или компонент компилятора, который выполняет лексический анализ.

13. \*\*Какая информация является входной для лексического анализатора, какая — выходной?\*\*

- Входной информацией для лексического анализатора является исходный текст программы.

- Выходной информацией являются лексемы (токены), которые используются для дальнейшей обработки компилятором.

14. \*\*Поясните различия между последовательным и параллельным лексическими анализаторами\*\*

- Последовательный лексический анализатор обрабатывает входной текст последовательно, символ за символом.

- Параллельный лексический анализатор может обрабатывать различные части текста одновременно, используя многопоточность или распределенные вычисления.

15. \*\*Дайте определение регулярному выражению над алфавитом \( I \)\*\*

Регулярное выражение над алфавитом \( I \) — это строка, которая описывает множество строк, составленных из символов алфавита \( I \), с использованием операций конкатенации, альтернативы (или) и замыкания Клини (\(\*\)).

16. \*\*Дайте определение конечному автомату \( M = (S, I, \delta, s\_0, F) \)\*\*

Конечный автомат \( M \) определяется как пятерка \( (S, I, \delta, s\_0, F) \), где:

- \( S \) — конечное множество состояний.

- \( I \) — алфавит (конечное множество символов).

- \( \delta \) — функция переходов \( \delta: S \times I \rightarrow S \).

- \( s\_0 \) — начальное состояние (\( s\_0 \in S \)).

- \( F \) — множество конечных состояний (\( F \subseteq S \)).

17. \*\*В чем отличие между детерминированным и недетерминированным автоматом?\*\*

- Детерминированный конечный автомат (DFA) имеет одну возможную функцию перехода для каждого состояния и символа.

- Недетерминированный конечный автомат (NFA) может иметь несколько возможных переходов для каждого состояния и символа или включать \(\epsilon\)-переходы (переходы без ввода символа).

18. \*\*Определите понятие: мгновенное состояние конечного автомата\*\*

Мгновенное состояние конечного автомата — это текущее состояние автомата и оставшаяся часть входного слова.

19. \*\*Поясните обозначения \( (s, aw) \overset{\phi}{\rightarrow} (s', w) \) и \( (s, w\_i) \overset{\phi^\*}{\rightarrow} (s\_k, w\_k) \)\*\*

- \( (s, aw) \overset{\phi}{\rightarrow} (s', w) \) означает, что автомат из состояния \( s \) по символу \( a \) переходит в состояние \( s' \), оставляя слово \( w \).

- \( (s, w\_i) \overset{\phi^\*}{\rightarrow} (s\_k, w\_k) \) означает, что автомат из состояния \( s \) по слову \( w\_i \) за конечное количество шагов переходит в состояние \( s\_k \), оставляя слово \( w\_k \).

20. \*\*В каком соотношении находятся регулярная грамматика, регулярный язык, регулярное выражение, конечный автомат, граф состояний конечного автомата?\*\*

- Регулярная грамматика, регулярное выражение, конечный автомат и граф состояний конечного автомата описывают одни и те же множества строк, то есть регулярные языки. Они являются разными способами представления и анализа регулярных языков.