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 6 группы

Жучкевич Екатерина Сергеевна

Преподаватель: Наркевич Аделина Сергеевна

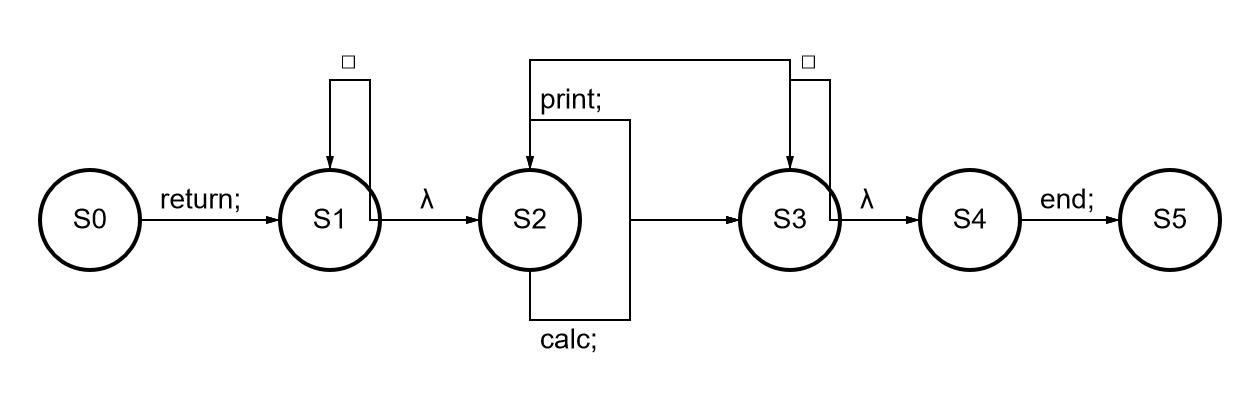
Минск, 2024



**Задание 3:**

1. return; □calc; □end;
2. return; **λ** calc; **λ** end;
3. return; □□□ print; □□ end;
4. return; **λ** calc; calc; □ end;
5. return; □ print; print; print; □□ end;
6. return; □□ calc; **λ** end;
7. return; □ print; □□□□ end;

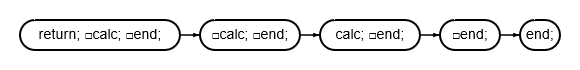
**Задание 5:**

****



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **return** | **□** | **λ** | **calc** | **print** | **end** |
| **S0** | **S1** |  |  |  |  |  |
| **S1** |  | **S1** | **S2** |  |  |  |
| **S2** |  |  |  | **S3** | **S3** |  |
| **S3** |  | **S3** | **S4** |  |  |  |
| **S4** |  |  |  |  |  | **S5** |
| **S5** |  |  |  |  |  |  |

**Задание 4:**



1. **Что такое алфавит I?**
   1. **Алфавит** — это конечное непустое множество допустимых символов языка (букв языка). Пример: 𝑉={𝑎,𝑏}V={a,b} — алфавит 𝑉V, состоящий из двух символов a и b​​.
2. **Поясните обозначение λ, I, I+:**\*
   1. **λ** (лямбда) обозначает пустую цепочку, которая не содержит ни одного символа.
   2. **I**\* — множество всех возможных цепочек, которые можно составить из символов алфавита 𝐼I, включая пустую цепочку.
   3. **I+** — множество всех возможных непустых цепочек, составленных из символов алфавита 𝐼I​​.
3. **Что такое язык L(I) над алфавитом I?**
   1. **Язык L(I)** над алфавитом 𝐼I называется произвольное подмножество цепочек из 𝐼∗I∗ (множества всех возможных цепочек из символов алфавита 𝐼I)​​.
4. **Дайте определение формальной грамматике G:**
   1. **Формальная грамматика** — это способ задания языка, который включает множество правил для порождения цепочек символов этого языка. Формальная грамматика обычно записывается как 𝐺=(𝑁,𝑇,𝑃,𝑆)G=(N,T,P,S), где:
      1. 𝑁N — конечное множество нетерминальных символов;
      2. 𝑇T — конечное множество терминальных символов;
      3. 𝑃P — конечное множество правил вывода;
      4. 𝑆S — стартовый символ, из которого начинаются выводы​​.
5. **Поясните обозначение α ⇒ β и α ⇒ β:**\*
   1. **α ⇒ β** — цепочка 𝛽β непосредственно выводима из цепочки 𝛼α по одному правилу грамматики.
   2. **α ⇒ β**\* — цепочка 𝛽β выводима из цепочки 𝛼α за несколько (возможно, ноль) шагов по правилам грамматики​​.
6. **Что такое язык L(G), порождаемый грамматикой G?**
   1. **Язык L(G)** — это множество всех терминальных цепочек, которые можно вывести из стартового символа 𝑆S грамматики 𝐺G с использованием правил этой грамматики. Формально это записывается как 𝐿(𝐺)={𝛼∈𝑇∗∣𝑆⇒∗𝛼}L(G)={α∈T∗∣S⇒∗α}, где 𝑇T — множество терминалов​​.
7. **Что такое форма Бэкуса-Наура?**
   1. **Форма Бэкуса-Наура (BNF)** — это нотация для формального описания синтаксиса языков программирования. Она используется для задания правил грамматики в виде производственных правил, которые описывают, как строки языка могут быть порождены из начального символа​​.
8. **Что такое регулярная грамматика?**
   1. **Регулярная грамматика** — это грамматика, в правилах вывода которой с правой стороны находится либо одиночный терминал, либо терминал, за которым следует нетерминал. Примеры правил регулярной грамматики: 𝐴→𝑎𝐵A→aB, 𝐵→𝑏B→b, 𝐶→𝜀C→ε​​.
9. **Поясните понятие "регулярное множество":**
   1. **Регулярное множество** — это множество, которое может быть описано регулярным выражением или распознано конечным автоматом. Это подмножество цепочек, которые удовлетворяют регулярному выражению​​.
10. **Поясните понятие "регулярный язык":**
    1. **Регулярный язык** — это язык, который может быть описан регулярной грамматикой, регулярным выражением или распознан конечным автоматом. Все регулярные языки являются регулярными множествами​​.
11. **Что такое лексический анализ?**
    1. **Лексический анализ** — это процесс разбора входного потока символов на токены (лексемы), которые являются смысловыми единицами языка программирования. Этот процесс выполняется лексическим анализатором​​.
12. **Что такое лексический анализатор?**
    1. **Лексический анализатор** (сканер) — это компонент компилятора, который выполняет лексический анализ, то есть преобразует входной поток символов в последовательность токенов​​.
13. **Какая информация является входной для лексического анализатора, какая — выходной?**
    1. **Входная информация:** исходный код программы в виде потока символов.
    2. **Выходная информация:** последовательность токенов, где каждый токен представляет собой отдельную смысловую единицу программы (ключевые слова, идентификаторы, операторы и т.д.)​​.
14. **Поясните различия между последовательным и параллельным лексическими анализаторами:**
    1. **Последовательный лексический анализатор** обрабатывает входной поток символов линейно, один за другим.
    2. **Параллельный лексический анализатор** может обрабатывать несколько потоков символов одновременно, что может увеличить производительность, особенно в многопроцессорных системах.
15. **Дайте определение регулярного выражения над алфавитом I:**
    1. **Регулярное выражение** — это формальное средство для описания множества строк в алфавите 𝐼I. Регулярные выражения состоят из символов алфавита и специальных символов (метасимволов), которые обозначают операции объединения, конкатенации и замыкания (звезда Клини)​​.
16. **Дайте определение конечному автомату M = (S, I, δ, s0, F):**
    1. **Конечный автомат** — это математическая модель, состоящая из:
       1. 𝑆S — конечного множества состояний,
       2. 𝐼I — конечного множества входных символов (алфавит),
       3. 𝛿δ — функции переходов, 𝛿:𝑆×𝐼→𝑆δ:S×I→S,
       4. 𝑠0s0 — начального состояния, 𝑠0∈𝑆s0∈S,
       5. 𝐹F — множества конечных состояний, 𝐹⊆𝑆F⊆S​​.
17. **В чем отличие между детерминированным и недетерминированным автоматом?**
    1. **Детерминированный конечный автомат (DFA):** в каждом состоянии для каждого входного символа определен только один переход.
    2. **Недетерминированный конечный автомат (NFA):** в каждом состоянии для каждого входного символа может быть несколько возможных переходов, включая переходы без потребления символа (ε-переходы)​​.
18. **Определите понятие: мгновенное состояние конечного автомата:**
    1. **Мгновенное состояние конечного автомата** — это пара (s, w), где 𝑠s — текущее состояние автомата, а 𝑤w — оставшаяся часть входной строки, которая еще не обработана.
19. **Поясните обозначения (s, aw) ⟹ (s', w) и (sj, wi) ⟹ (sk, wk):**\*
    1. **(s, aw) ⟹ (s', w)** — автомат переходит из состояния 𝑠s в состояние 𝑠′s′ при чтении символа 𝑎a из входной строки 𝑤w.
    2. **(sj, wi) ⟹ (sk, wk)**\* — автомат за несколько шагов переходит из состояния 𝑠𝑗sj в состояние 𝑠𝑘sk, обрабатывая строку 𝑤𝑖wi, и оставляя остаток строки 𝑤𝑘wk​​.
20. **В каком соотношении находятся регулярная грамматика, регулярный язык, регулярное выражение, конечный автомат, граф состояний конечного автомата?**
    1. **Регулярная грамматика, регулярный язык, регулярное выражение и конечный автомат** описывают один и тот же класс языков — регулярные языки. Регулярные выражения и регулярные грамматики могут быть преобразованы в конечные автоматы, которые могут быть представлены в виде графов состояний, описывающих их работу. Эти концепции взаимосвязаны и используются для описания и анализа регулярных языков​​.
21. Конечный автомат - модель, с помощью которой удобно представлять процесс, имеющий конечное число дискретных управляющих состояний.