Московский авиационный институт

(национальный исследовательский институт)

Институт «Компьютерные науки и прикладная математика»

**Лабораторные работы**

**по курсу**

**«Системы программирования»**

**IV семестр**

1. Спроектировать грамматику по паттерн-модели регулярного языка.

2. Преобразовать спроектированную грамматику в конечный автомат, составить диаграмму переходов КА и реализовать.

3. Определить свойства КА. Изучить алгоритм преобразования НДКА в ДКА.

4. Устранить из КС-грамматики бесполезные символы и ε–правила.

5. Устранить из KС-грамматики цепные правила и устранить левую рекурсию.

6. Определить форму КС-грамматики и сделать ее приведение.

7. Спроектировать МП-автомат для приведенной КС-грамматики.

8. Реализовать МП-автомат для приведенной КС-грамматики.

9. Для LL(k) анализатора построить управляющую таблицу M.

10. Аналитически написать правила вывода для цепочки LL(k) анализатора.

11. Реализовать управляющую таблицу M Для LL(k) анализатора.

12. Построить множество LR(0)-таблиц не содержащих ε-правила.

13. Для LR(k) -грамматики cпроектировать матрицу oblow.

14. Определить функции перехода g(X).

15. Определить функцию переноса-свертки f(u).

16. Для функции перехода g(X) и функции переноса-свертки f(u) спроектировать управляющую таблицу.

*Студент:* Хомяков И.А.*Группа: М8О-207Б-21*

*Руководитель:* Семёнов А. С.

*Оценка:*

*Дата:*

**Москва. 2023**

**Практическая работа №1 (1-3 лаб.)**

*Лабораторные работы №1-2*

**Формулировка задания**:

Спроектировать грамматику для трёх заданных паттернов. Составить на основе разработанных регулярных грамматик конечные автоматы, распознающие эквивалентные им языки.

Спроектируем грамматику для заданного языка:

**6-2. pattern = "\s+"**

**Автоматная грамматика:**

L(pattern) = L("\\s+") = {‘ ‘}\*

G(T, V, P, S0) = G({‘ ‘}, {**S0**, **A**}, {p1, p2, p3}, **S0**)

*Правила регулярной грамматики:*

**p1:** **S0** → \_**A**

**p2:** **A** → ε

**p3:** **A** → \_**A**

*Пример цепочек: (\_ == ‘ ‘)*

**S0** =>1 \_**A** =>3 \_ \_**A** =>2 \_ \_

**S0** =>1 \_**A** =>2 \_

**Конечный автомат:**

L(КА) = L(G)

КА = (**Q**, **Σ**, **δ**, **S0**, **F**), где

**Q** = { **S0**, **A**, **qf** }, **Σ** = { ‘ ‘ }, **S0** = **S0**, **F** = **qf**,

**δ** = { 1. δ(**S0**, ‘ ‘) = {**A**},

2. δ(**A**, ‘ ‘) = {**A**},

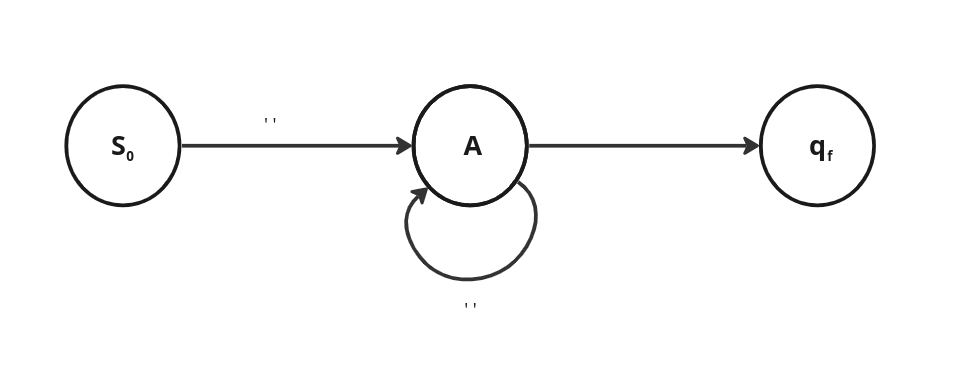
3. δ(**A**, ε) = {**qf**}

}

*Примеры конфигурации КА: (\_ == ‘ ‘)*

**1.** (**S0**, \_ \_ \_) ⸠1 (**A**, \_ \_) ⸠2 (**A**, \_) ⸠2 (**A**, ε) ⸠3 (**qf**, ε)

**2.** (**S0**, \_) ⸠1 (**A**, ε) ⸠3 (**qf**, ε)



*Лемма о накачке:*

...

**2. pattern = ^[A-Z][a-zAZ]\***

**Автоматная грамматика:**

L(pattern) = L("^[A-Z][a-zAZ]\*") = {A … Z}{{a … z} | A | Z}

G(T, V, P, S0) = G({a … z, A … Z}, {**S0**, **$**}, {p1, p2, p3}, **S0**)

*Правила регулярной грамматики:*

**p1:** **S0** → A**$** | … | Z**$**

**p2:** **$** → ε

**p3:** **$** → a**$** | … | z**$** | A**$** | Z**$**

*Пример цепочек:*

**S0** =>1 F**$** =>3 Ff**$** =>3 FfA**$** =>3 FfAf**$** =>2 FfAf

**S0** =>1 F**$** =>2 F

**S0** =>1 A**$** =>3 AA**$** =>2 AA

**Конечный автомат:**

L(КА) = L(G)

КА = (**Q**, **Σ**, **δ**, **S0**, **F**), где

**Q** = { **S0**, **$**, **qf** }, **Σ** = { A … Z, a … z }, **S0** = **S0**, **F** = **qf**,

**δ** = {

1. δ(**S0**, A) = {**$**},

...

26. δ(**A**, Z) = {**$**},

27. δ(**A**, a) = {**$**},

...

52. δ(**A**, z) = {**$**},

53. δ(**A**, A) = {**$**},

54. δ(**A**, Z) = {**$**},

55. δ(**A**, ε) = {**qf**}

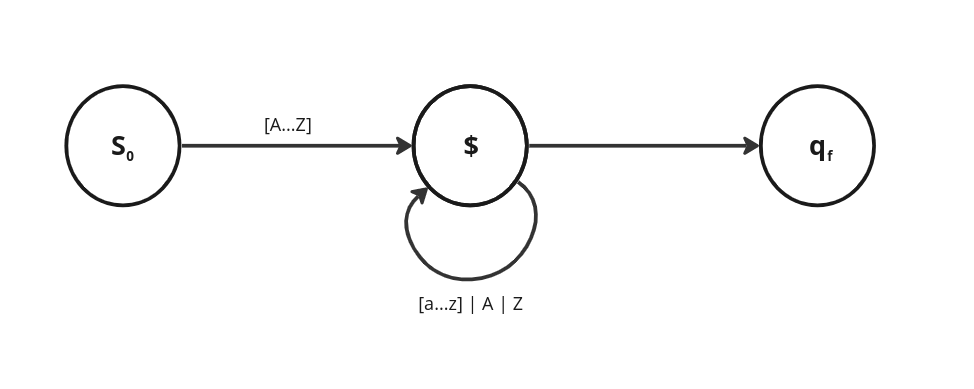
}

*Примеры конфигурации КА:*

**1.** (**S0**, FfAf) ⸠6 (**$**, fAf) ⸠32 (**$**, Af) ⸠53 (**$**, f) ⸠32 (**$**, ε) ⸠55 (**qf**, ε)

**2.** (**S0**, F) ⸠6 (**$**, ε) ⸠55 (**qf**, ε)

**3.** (**S0**, AA) ⸠1 (**$**, A) ⸠53 (**$**, ε) ⸠55 (**qf**, ε)



*Лемма о накачке:*

...

*Лабораторная работа №3*

**Формулировка задания:**

Реализовать конечные автоматы, составленные в ЛР №2

**Код программы:**

**https://github.com/EbumbaE/SP\_LAB**

**Пример работы программы:**

123