МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КПІ»

Кафедра
Автоматизованих систем обробки інформації та управління

Лабораторна робота №3

з дисципліни

«Теоретичні основи лексичного та синтаксичного аналізу»

на тему:

Побудова скінченного автомата за регулярною граматикою

Варіант 16

Перевірив:

Селін Ю. М.

Виконав:

Педоренко О. Р.

Студент груп ІП-31,
факультету ІОТ,
залікова книжка ІП-3114

Київ 2016

1. Мета роботи

Закрепить понятия «регулярная грамматика», «недетерминированный и детерминированный конечный автомат», сформировать умения и навыки построения конечного автомата по регулярной грамматике и преобразования недетерминированного конечного автомата к детерминированному конечному автомату.

2. Завдання на роботу

Разработать программное средство, реализующее следующие функции:

1) ввод произвольной формальной грамматики с клавиатуры и проверка ее на принадлежность к классу регулярных грамматик;

2) построение по заданной регулярной грамматике конечного автомата;

3) преобразование недетерминированного конечного автомата к детерминированному конечному автомату;

4) вывод графа результирующего конечного автомата на экран.

3. Варіант індивідуального завдання

16 mod 12 + 1 = 5

5	$G=(\{K, L, M, N, Q, P, R, S\}, \{0, 1, *, \$, /\}, V, K)$, где V: 1) $K \to 1L \mid 0N$; 2) $L \to 0M \mid 0P \mid /Q$; 3) $N \to 1R \mid 1M \mid *S$; 4) $Q \to 1P$; 5) $P \to *L \mid \$$; 6) $M \to \$$; 7) $S \to 0R$; 8) $R \to /N \mid \$$.

4. Виконання роботи

У результаті виконання роботи була розроблена комп'ютерна програма на мові Python з інтерфейсом командного рядка, що реалізує наступні функції:

1) Введення довільної граматики, перевірка її на належність до класу регулярних граматик.
2) Побудова НКА за заданою граматикою.
3) Побудова ДКА за заданою граматикою.
4) Виведення графа скінченого автомата на екран.

```
usage: lab3.py [-h] [--verbose] -T T [T ...] -N N [N ...] -P P [P ...] -S S
               [--non-deterministic-fsa | --deterministic-fsa]
               [--image-name IMAGE_NAME]

Build an automaton for regular grammar. \e and ` denote an empty string

optional arguments:
  -h, --help            show this help message and exit
  --verbose, -v         Enable verbose output
  -T T [T ...]          List of terminals of a grammar
  -N N [N ...]          List of non-terminals of a grammar
  -P P [P ...]          Rules of production for grammar
  -S S                  Starting symbol of a grammar

  --non-deterministic-fsa, -nfa
                        Build a non-deterministic finite state automaton
  --deterministic-fsa, -dfa
                        Build a deterministic finite state automaton
  --image-name IMAGE_NAME, -i IMAGE_NAME
                        Specify the name of the graph (*.svg) output file.
```

Рис. 1 Вбудована довідка для розробленої програми

5. Результати роботи програми

Для побудови НКА для індивідуального завдання, програма була запущена з наступними параметрами:

```
python ..\src\lab3.py -T "0" "1" "*" "$" "/" -N K L M N Q P R S -P "K->1L|0N"
"L->0M|0P|/Q" "N->1R|1M|*S" "Q->1P" "P->*L|$" "M->$" "S->0R" "R->/N|$" -S K -v
-nfa -i "graph-nfa"

.\graph-nfa.svg
```

Результат роботи:

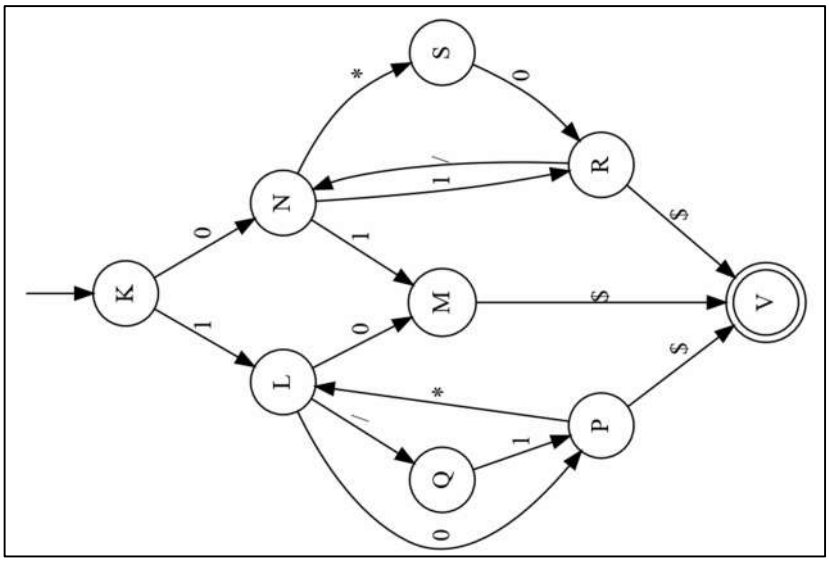
1) Визначення типу граматики

```
G = ({1, 0, *, $, /}, {K, M, L, N, Q, P, S, R}, P0..P7, K)
        P0 = K -> 1L | 0N;
        P1 = M -> $;
        P2 = L -> 0M | 0P | /Q;
        P3 = N -> 1R | 1M | *S;
        P4 = Q -> 1P;
        P5 = P -> *L | $;
        P6 = S -> 0R;
        P7 = R -> /N | $;
The grammar is a right regular grammar
```

2) Побудова НКА

```
Automaton:
        H: K
        Q: set(['K', 'M', 'L', 'N', 'Q', 'P', 'S', 'R', 'V'])
        T: set(['1', '0', '$', '*', '/'])
        Z: set(['V'])
        F:
                {('K', '0'): ['N'],
                 ('K', '1'): ['L'],
                 ('L', '/'): ['Q'],
                 ('L', '0'): ['M', 'P'],
                 ('M', '$'): ['V'],
                 ('N', '*'): ['S'],
                 ('N', '1'): ['R', 'M'],
                 ('P', '$'): ['V'],
                 ('P', '*'): ['L'],
                 ('Q', '1'): ['P'],
                 ('R', '$'): ['V'],
                 ('R', '/'): ['N'],
                 ('S', '0'): ['R']}
```

3) Виведення графу НКА на екран



Для побудови ДКА за індивідуального завдання, програма була запущена з наступними параметрами:

```
python ..\src\lab3.py -T "0" "1" "*" "$" "/" -N K L M N Q P R S -P "K->1L|0N"
"L->0M|0P|/Q" "N->1R|1M|*S" "Q->1P" "P->*L|$" "M->$" "S->0R" "R->/N|$" -S K -v
-nfa -i "graph-dfa"

.\graph-dfa.svg
```

Результат роботи:

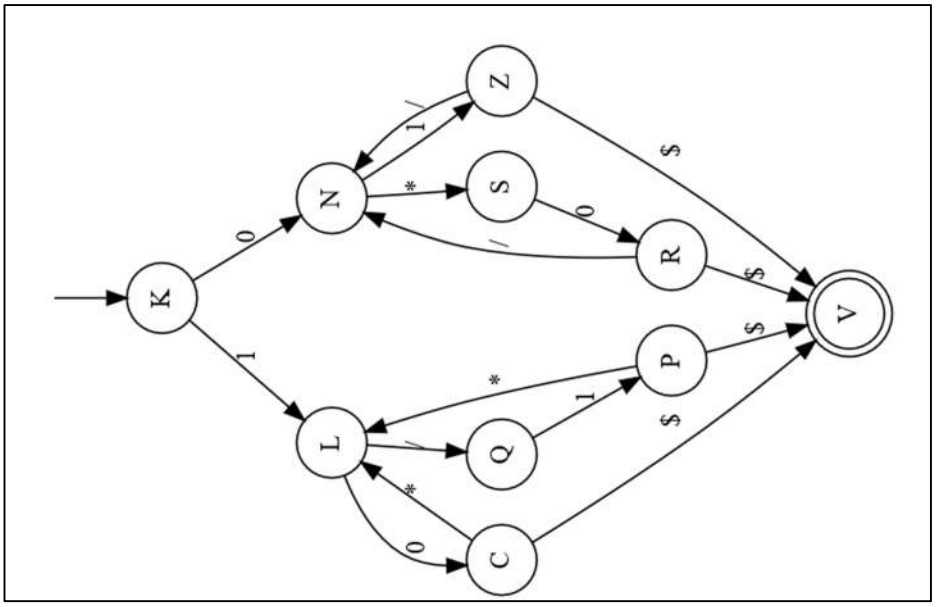
1) Визначення типу граматики

```
G = ({1, 0, *, $, /}, {K, M, L, N, Q, P, S, R}, P0..P7, K)
        P0 = K -> 1L | 0N;
        P1 = M -> $;
        P2 = L -> 0M | 0P | /Q;
        P3 = N -> 1R | 1M | *S;
        P4 = Q -> 1P;
        P5 = P -> *L | $;
        P6 = S -> 0R;
        P7 = R -> /N | $;
```

2) Побудова ДКА

```
Automaton:
        H: K
        Q: set(['C', 'K', 'L', 'N', 'Q', 'P', 'S', 'R', 'V', 'Z'])
        T: set(['1', '0', '*', '$', '/'])
        Z: set(['V'])
        F:
                {('C', '$'): ['V'],
                 ('C', '*'): ['L'],
                 ('K', '0'): ['N'],
                 ('K', '1'): ['L'],
                 ('L', '/'): ['Q'],
                 ('L', '0'): ['C'],
                 ('N', '*'): ['S'],
                 ('N', '1'): ['Z'],
                 ('P', '$'): ['V'],
                 ('P', '*'): ['L'],
                 ('Q', '1'): ['P'],
                 ('R', '$'): ['V'],
                 ('R', '/'): ['N'],
                 ('S', '0'): ['R'],
                 ('Z', '$'): ['V'],
                 ('Z', '/'): ['N']}
```

3) Виведення графу НКА на екран



6. Висновок

У ході виконання лабораторної роботи я детально ознайомився з наступними поняттями: «регулярна граматика», «граматика, вирівняна вліво», «граматика, вирівняна вправо», «детермінований скінченний автомат», «недетермінований скінченний автомат». Зрозумів різницю між НКА і ДКА, навчився виводити ДКА з НКА.

Перевага ДКА перед НКА полягає у тому, що за допомогою ДКА можна визначити належність певного рядка до мови, яку описує ДКА, за один прохід по рядку. При використанні НКА для задачі розпізнавання потрібно декілька проходів. Скінченні автомати – це ефективний спосіб опису та аналізу регулярних граматик.

Окрім того, навчився працювати з бібліотекою graphviz, argparse для python.