**UNIVERSITATEA DE STAT DIN MOLDOVA**

**FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ**

**DEPARTAMENTUL DE INFORMATICĂ**

**MAMALIGA ARTUR**

**Реализация конечного автомата для распознавания слова имеющее определенный особый вид.**

**Информатика**

**Лабораторная №1.**

**CHIȘINĂU – 2025**

# **Задание**

Задание: написать программу реализующую ĸонечный автомат, ĸоторый распознает слова, имеющие следующий общий вид:

a(bc)nd(de)m , n ≥0, m ≥0 .

### **Код:**

from enum import Enum

# States

# a(bc)^nd(de)^m ,n>=0,m>=0.

# n or m can be 0

# So there can be no bc or de or both

class State(Enum):

    STATE\_START\_IS\_A = 0

    STATE\_BC\_ACCEPT = 1

    STATE\_B\_THEN\_C = 2

    STATE\_DE\_ACCEPT = 3 # Final accepted state

    STATE\_D\_THEN\_E = 4

    STATE\_INVALID = 5

# The state consideed as the final accepted state

accepted\_state = {State.STATE\_DE\_ACCEPT}

# The table of transitions

transition\_table = {

    State.STATE\_START\_IS\_A:   {'a':State.STATE\_BC\_ACCEPT},

    State.STATE\_BC\_ACCEPT:    {'b':State.STATE\_B\_THEN\_C,'d':State.STATE\_DE\_ACCEPT},

    State.STATE\_B\_THEN\_C:     {'c':State.STATE\_BC\_ACCEPT},

    State.STATE\_DE\_ACCEPT:    {'d':State.STATE\_D\_THEN\_E},

    State.STATE\_D\_THEN\_E:      {'e':State.STATE\_DE\_ACCEPT},

}

# The function that recognizes the words

def fsm\_recognize\_words(input\_string) -> bool:

    current\_state = State.STATE\_START\_IS\_A

    for ch in input\_string:

        current\_state = transition\_table.get(current\_state, {}).get(ch,State.STATE\_INVALID)

        if current\_state == State.STATE\_INVALID:

            return False

    return current\_state in accepted\_state

def main():

    print("Finite State Machine: Recognize Words")

    #Examples

    tobe\_accepted = ["ad", "abcd", "abcbcd", "adde", "addede"]

    tobe\_rejected = ["a", "abd", "ade", "abccd", "abcdd", "abcbcded"]

    print("[INFO] Testing the FSM with some examples")

    print("[INFO] Accepted words:")

    for str in tobe\_accepted:

        assert fsm\_recognize\_words(str) == True, f"[Error] {str} should be accepted"

        print(f"\"{str}\" is accepted")

    print("[INFO] Rejected words:")

    for str in tobe\_rejected:

        assert fsm\_recognize\_words(str) == False, f"[Error] {str} should be rejected"

        print(f"\"{str}\" is rejected")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

### **Пример вывода:**

python .\lab01\src\fsm\_recognize\_words.py

Finite State Machine: Recognize Words

[INFO] Testing the FSM with some examples

[INFO] Accepted words:

"ad" is accepted

"abcd" is accepted

"abcbcd" is accepted

"adde" is accepted

"addede" is accepted

[INFO] Rejected words:

"a" is rejected

"abd" is rejected

"ade" is rejected

"abccd" is rejected

"abcdd" is rejected

"abcbcded" is rejected