Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний інститут ім. Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

Лабораторна робота №5

з дисципліни:

«Мультипарадигменне програмування»

Виконав:

студент групи ІК-21

Бераудо Лоренцо Раффаелович

Київ 2025

ЛАБОРАТОРНА РОБОТА №5

Завдання: за допомогою продукційного програмування реалізувати перетворення чисельного ряду до лінгвістичного ланцюжка за певним розподілом ймовірностей потрапляння значень до інтервалів.

Вхідні данні: чисельний ряд, вид розподілу ймовірностей, потужність алфавіту.

Вихідні дані: лінгвістичний ряд.

Мова програмування: CLIPS.

Варіант 1: Дискретний рівномірний розподіл (рівноймовірний)

ХІД ВИКОНАННЯ ЗАВДАННЯ

1. **Генерація числового ряду.**

Вхідні числові дані задаються у вигляді списку (факт numeric_series). Можуть бути отримані з попередніх лабораторних або створені вручну.

2. **Сортування числового ряду.**

Для подальшого розбиття на інтервали ряд сортується у порядку зростання. Це дозволяє точно визначити мінімальне та максимальне значення, які задають межі діапазону значень.

3. **Розбиття діапазону значень на інтервали.**

Діапазон розбивається на N рівних інтервалів, де N — потужність алфавіту. Для дискретного рівномірного розподілу всі інтервали мають однакову довжину. Розрахунок проводиться за формулою:

`Step = (max - min + 1) / N`, де Step — довжина одного інтервалу.

Межі інтервалів формуються як список пар [a,b].

4. **Перетворення чисел у символи алфавіту.**

Для кожного числового значення визначається інтервал, у який воно

потрапляє, після чого числу присвоюється відповідний символ алфавіту за індексом цього інтервалу.

5. **Формування лінгвістичного ряду.**

Усі числа замінюються на відповідні символи, і формується лінгвістичний ряд.

6. **Побудова матриці передування.**

Із сформованого лінгвістичного ряду будується матриця переходів між символами.

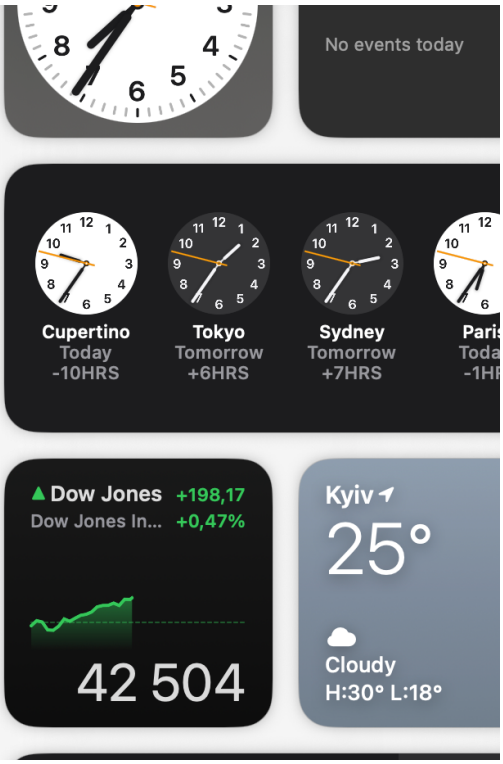
Вона має вигляд квадратної матриці розміром N x N, де N — потужність алфавіту.

Кожен елемент a(i,j) вказує кількість випадків появи символа j після символа i.

ПРИКЛАД ВИВОДУ

Лінгвістичний ряд та матриця передування (фрагмент):

```
CLIPS> (reset)
CLIPS> (run)
Лінгвістичний ряд: ['B', 'J', 'G', 'C', 'I', 'J',
Матриця передування:
A -> A: 0
A -> B: 0
A -> C: 0
A -> D: 1
A -> E: 0
A -> F: 0
A -> G: 0
A -> H: 0
A -> I: 0
A -> J: 0
B -> A: 0
B -> B: 0
B -> C: 0
B -> D: 0
B -> E: 0
B -> F: 0
B -> G: 0
B -> H: 0
```



ВИСНОВОК

У ході виконання лабораторної роботи було реалізовано програму на мові логічного програмування CLIPS для перетворення чисельного ряду у лінгвістичний ланцюжок на основі дискретного рівномірного розподілу.

Програма дозволяє:

- автоматично будувати інтервали на основі вхідного числового ряду та заданої потужності алфавіту;
- відображати кожне числове значення на відповідний символ алфавіту згідно з розподілом;
- формувати лінгвістичний ряд;
- будувати матрицю передування символів, яка показує частоту переходів між літерами.

Особливістю реалізації є гнучкість у виборі алфавіту, що дозволяє змінювати як його кількість, так і зміст без потреби модифікації логіки програми.

Усі компоненти були реалізовані у вигляді фактів і правил, що повністю відповідає парадигмі логічного програмування. Отримані результати збігаються з результатами, отриманими у попередніх лабораторних роботах, виконаних на інших мовах (R, Clojure тощо), що підтверджує правильність обраного підходу.

Таким чином, у процесі виконання лабораторної роботи було:

- поглиблено знання з логічного програмування;
- здобуто практичні навички роботи з базами фактів, рекурсією та обробкою списків у CLIPS;
- продемонстровано, що логічне програмування є ефективним інструментом для розв’язання задач класифікації та аналізу даних.

Код програми

```
;; Лабораторна №6 — CLIPS
(deffacts initial-data
  (alphabet A B C D E F G H I J)
  (alphabet-power 10)
  (numeric-series 134 987 543 234 765 876 453 100 321 678)
)

(defglobal ?*intervals* = nil)
(defglobal ?*letters* = nil)

(deffunction min-max (?list)
  (bind ?min (nth$ 1 ?list))
  (bind ?max ?min)
  (foreach ?x ?list
    (if (< ?x ?min) then (bind ?min ?x))
    (if (> ?x ?max) then (bind ?max ?x))
  )
  (return (create$ ?min ?max))
)

(deffunction build-intervals (?min ?max ?power)
  (bind ?step (/ (+ (- ?max ?min) 1) ?power))
  (bind ?intervals (create$))
  (loop-for-count (?i 0 (- ?power 1))
    (bind ?start (+ ?min (* ?i ?step)))
    (bind ?end (+ ?start ?step))
    (bind ?intervals (insert$ ?intervals (+ (* ?i 2) 1) (create$ ?start ?end)))
  )
  (return ?intervals)
)

(deffunction map-symbol (?value ?intervals ?alphabet)
  (bind ?index 1)
  (loop-for-count (?i 1 (length$ ?intervals) 2)
    (bind ?low (nth$ ?i ?intervals))
    (bind ?high (nth$ (+ ?i 1) ?intervals))
    (if (and (>= ?value ?low) (< ?value ?high)) then
      (return (nth$ ?index ?alphabet))
    )
    (bind ?index (+ ?index 1))
  )
  (return ?UNKNOWN)
)

(deffunction build-letter-series (?nums ?intervals ?alphabet)
  (bind ?letters (create$))
  (foreach ?n ?nums
    (bind ?symbol (map-symbol ?n ?intervals ?alphabet))
    (bind ?letters (create$ ?letters ?symbol))
  )
  (return ?letters)
)

(deffunction build-transitions (?letters ?alphabet)
  (printout t "Матриця передування:" crlf)
  (loop-for-count (?i 1 (length$ ?alphabet))
    (bind ?from (nth$ ?i ?alphabet))
    (loop-for-count (?j 1 (length$ ?alphabet))
      (bind ?to (nth$ ?j ?alphabet))
      (bind ?count 0)
      (loop-for-count (?k 1 (- (length$ ?letters) 1))
        (if (and (eq (nth$ ?k ?letters) ?from)
```

```
                (eq (nth$ (+ ?k 1) ?letters) ?to)) then
          (bind ?count (+ ?count 1))
        )
      )
      (printout t ?from " -> " ?to ": " ?count crlf)
    )
  )
)

(defrule start
  (declare (salience 100))
  (alphabet $?alpha)
  (alphabet-power ?p)
  (numeric-series $?nums)
=>
  (bind ?sorted (sort ?nums <))
  (bind ?mm (min-max ?sorted))
  (bind ?min (nth$ 1 ?mm))
  (bind ?max (nth$ 2 ?mm))
  (bind ?*intervals* (build-intervals ?min ?max ?p))
  (bind ?*letters* (build-letter-series ?nums ?*intervals* ?alpha))
  (printout t "Лінгвістичний ряд: " ?*letters* crlf crlf)
  (build-transitions ?*letters* ?alpha)
)
```