УДК 004.91

Магістрант Балаценко О.В., д.т.н., доц. Сулема Є.С.

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

ВИКОРИСТАННЯ НЕЧІТКОЇ ЛОГІКИ ДЛЯ АНАЛІЗУ ДАНИХ СЕНСОРІВ ЗА ДОПОМОГОЮ МОВИ ПРОГРАМУВАННЯ ASAMPL

Abstract

Oleksandr Balatsenko, student; Yevgeniya Sulema, DSc, assoc. prof.
Using fuzzy logic to analyze sensor data by means the ASAMPL programming language

This paper presents the result of using fuzzy logic to analyze sensor data by means the ASAMPL programming language. ASAMPL, based on aggregates and tuples, enables structured processing of uncertain and modal information. A fuzzy model for light intensity evaluation demonstrates fuzzification in practice. The approach improves intelligent sensor data interpretation and decision-making under uncertainty.

Вступ

У сучасному світі існують системи, що працюють із великою кількістю сенсорів та обробляють значні обсяги даних. Такі дані часто є неповними, зашумленими або містять неточності, що ускладнює їх класичну аналітичну обробку. Для підвищення надійності та адаптивності подібних систем доцільним є використання нечіткої логіки, яка дозволяє моделювати невизначеність і приймати рішення в умовах неповних даних. Одним із ефективних інструментів реалізації таких підходів є мова програмування ASAMPL, побудована на основі алгебраїчної системи агрегатів і призначена для обробки мультимедійних та мульсемедійних даних [1].

Термінологія

Сенсор – це пристрій, який вимірює фізичні параметри навколишнього середовища (температуру, освітленість, вологість, тиск тощо) та перетворює їх у цифровий або аналоговий сигнал, придатний для подальшої обробки. Сенсор є основним джерелом даних у системах моніторингу та керування.

Актуатор – це пристрій, який виконує певну фізичну дію у відповідь на сигнал керування: вмикає або вимикає освітлення, відкриває клапан, переміщує механізм тощо.

Нечітка логіка [2] – це розширення класичної булевої логіки, у якій істинність висловлювання може приймати будь-яке значення в діапазоні [0, 1], а не лише 0 або 1.

ASAMPL [3] – це спеціалізована мова програмування, призначена для обробки мультимедійних та мульсемедійних даних. Ця мова програмування отримала свою назву від словосполучень «Algebraic System of Aggregates» (алгебраїчна система агрегатів) та «Mulsemedia data Processing Language» (мова обробки мульсемедійних даних).

Постановка задачі

Показати ефективність застосування мови ASAMPL для обробки нечітких даних в автоматизованих системах.

Розв'язання задачі за допомогою нечіткої логіки

Розглянемо задачу автоматичної оцінки освітленості у кімнаті на основі даних сенсора світла. Сенсор вимірює рівень освітленості у люксах (Lux), і ці значення можуть змінюватися в діапазоні від 0 до 1000. Необхідно визначити, наскільки яскраво в кімнаті, і класифікувати стан освітлення як одне з трьох нечітких значень: темно (*dark*), помірно (*moderate*), світло (*bright*). Для математичного опису цих значень пропонується використовувати трикутні функції приналежності. Такий вибір пояснюється простою структурою цих функцій, зрозумілою інтерпретацією параметрів (a, b, c) як меж та піку функції, а також невисокою обчислювальною складністю. Трикутні функції забезпечують плавні переходи між станами і добре апроксимують реальні зміни яскравості, які не мають різких границь.

Введемо нечітку змінну $X = LightLevel$ і опишемо її через три нечіткі множини (A_1, A_2, A_3); кожна нечітка множина $A_i \in X$ визначається функцією приналежності:

$$\left[\mu_{A_i}(x): X \to [0,1]\right]$$

Формула трикутної функції приналежності має наступний вигляд:

$$\mu(x,a,b,c) = \begin{cases} 0, & x \le a \\ \frac{x-a}{b-a}, & a < x \le b \\ \frac{c-x}{c-b}, & b < x < c \\ 0, & x \ge c \end{cases},$$

де x – це рівень освітлення,
(a, b, c) – параметри, що визначають форму трикутника.
Визначимо нечіткі множини як показано у табл. 1 та на рис. 1.

Таблиця 1

Визначення нечітких значень

Термін	**Позначення**	**Діапазон**	**Параметри (a,b,c)**
Темно	A_1	[0, 400]	(0, 0, 400)
Помірно	A_2	[200, 800]	(200, 500, 800)
Світло	A_3	[600, 1000]	(600, 1000, 1000)

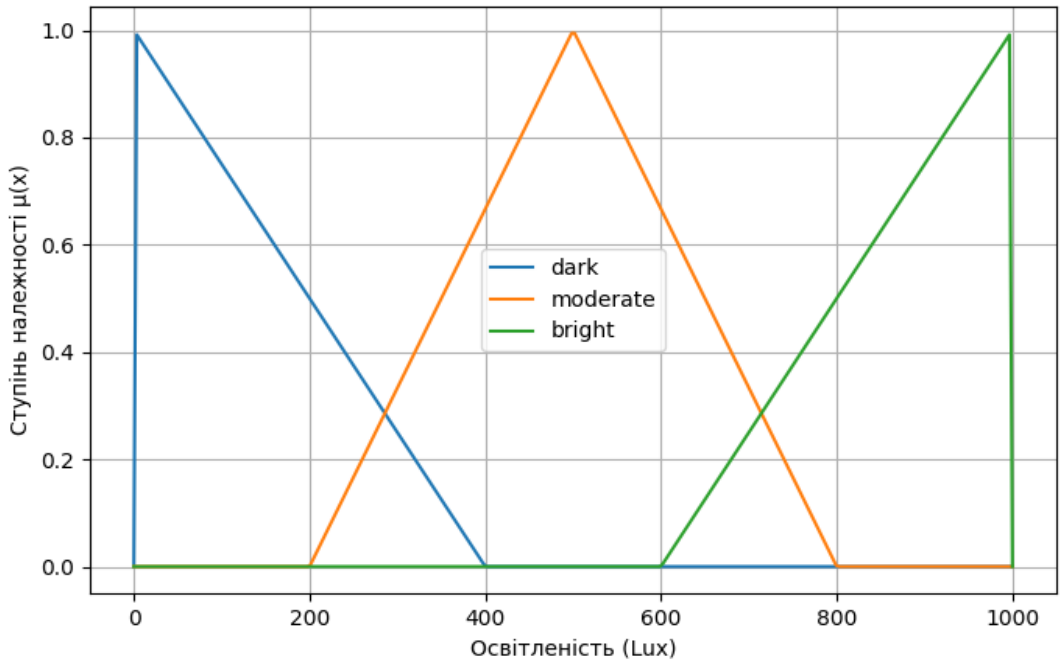

Рис. 1. Функції приналежності для змінної «Освітленість»

Зробимо фузифікацію значень; для цього припустимо що сенсор повернув значення 650, отже результат буде мати наступний результат:

$$\mu_{A_1}(650) = 0, \quad \mu_{A_2}(650) = \frac{800-650}{800-500} = 0.5, \mu_{A_3}(650) = \frac{650-600}{1000-600} = 0.125$$

Розв'яжемо цю задачу за допомогою алгебраїчної системи агрегатів (АСА) [1]. Згідно визначення агрегату [4]: $A = \left[M_j \middle| \langle a_i^j \rangle_{i=1}^{n_j}\right]_{j=1}^{N}$ нечіткі значення можна представити у наступному вигляді:

$$P = [P_1, P_2, P_3 \mid \langle x_1, a_1, b_1, c_1\rangle, \ \langle x_2, a_2, b_2, c_2\rangle, \ \langle x_3, a_3, b_3, c_3\rangle,]$$

$$P = [(650,0,0,400), (650, 200, 500, 800), (650,600, 1000, 1000)]$$

$$f_{P_i}: P \rightarrow [0,1]$$

$$f(p) = \mu(p_x, p_a, p_b, p_c)$$

$$f(P) = [P_1', P_2', P_3' \mid f(\langle x_1, a_1, b_1, c_1\rangle), \ f(\langle x_2, a_2, b_2, c_2\rangle), f(\langle x_3, a_3, b_3, c_3\rangle)$$

Таким чином застосувавши до агрегату P функцію f отримуємо агрегат ступеней залежності $P' = [0, 0.5, 0.125]$. Для інтерпретації результату введемо набір правил для ввімкнення лампи:

1. *Якщо Темно, то 1.*

2. *Якщо Помірно, то 0.5.*
3. *Якщо Світло, то 0.*

Тепер можна інтерпретувати результат, обчисливши наступну рівність: $\frac{0\cdot 1 + 0.5\cdot 0.5\cdot + 0.125 * 0}{0+0.5+0.125} = \frac{0.25}{0.625} = 0.4$, тобто можна стверджувати, що лампа має світити з яскравістю 40%, або використовувати це значення як поріг, при якому лампу необхідно вмикати.

Для демонстрації ефективності вирішення цієї задачі за допомогою мови програмування ASAMPL, напишемо сценарій визначення значення яскравості. Для цього імпортуємо бібліотеку для роботи с нечіткою логікою за допомогою виразу *import system.fuzzy as fuzzy*, далі визначимо агрегат з параметрами належності *Aggregte p = { (650, 0, 0, 400), (650,200, 500, 800), (650, 600, 1000, 1000) },* за допомогою функції *fuzzy.trimpfA* розрахуємо параметри належності, та за допомогою вбудованих методів агрегату отримаємо результат. Для визначення ефективності застосування мови ASAMPL ця задача програмно розв'язана також за допомогою мов C# та Python. Відповідно до метрик, наведених у табл. 2, мова програмування ASAMPL уможливлює підвищити ефективність обчислення нечітких даних.

Таблиця 2

Метрики коду

Мова програмування	**Кількість рядків коду**	**Кількість логічних рядків**	**Обчислювальна складність**
ASAMPL	4	6	O(1)
C#	25	14	O(n)
Python	15	9	O(n)

Висновок

Застосування алгебраїчної системи агрегатів дає змогу подати нечітку змінну у вигляді впорядкованої структури, у якій кожен терм має свій ступінь приналежності. Це уможливлює формалізацію процесів обробки нечітких даних для спрощення моделювання процесів, які визначаються нечіткими значеннями, за допомогою мови програмування та мови програмування ASAMPL.

Література

1. Sulema Ye., Kerre E. Multimodal Data Representation and Processing Based on Algebraic System of Aggregates, Wiley, USA, 2020.
2. Sulema Ye., Kerre E. On Fuzziness in Algebraic System of Aggregates. New Mathematics and Natural Computation, Vol. 17, No. 1, 2021.