KAZAKH BRITISH TECHNICAL UNIVERSITY

REPORT

Fuzzy logic

Basics of Computer Simulation

Prepared by: Balganym Tulebayeva
Checked by: Assel Akzhalova

Almaty 2018

JS-Fuzzy

Fuzzy logic, invented by a man named Lotfi Zadeh in the mid-sixties, enables a computer to reason about linguistic terms and rules in a way similar to humans. Concepts like "far" or "slightly" are not represented by discrete intervals, but by fuzzy sets, enabling values to be assigned to sets to a matter of a degree - a process called fuzzification. Using fuzzified values computers are able to interpret linguistic rules and produce an output that may remain fuzzy or - more commonly, especially in video games - can be defuzzified to provide a crisp value. This is known as fuzzy rule-based inference, and is one of the most popular uses of fuzzy logic. In this library is used the Combs Method. William Combs in 1997 Combs proposed a system that enables the number of rules to grow linearly with the number of member sets instead of exponentially.

Object initialization:

```
var ai = new FuzzyLogic();
```

The method to use to process the output is:

```
ai.getResult(object);
```

Object passed to the script

We need an array of input variables called variables_input and an array of input value called crisp_input: a crisp value for each variable. Every variable is composed by a set of function. In this library I used only trapezoidal function to achieve a goal: remain in a linear environment and get good performance. A trapezoidal function is composed by four numbers corresponding the four x coordinates. The inferences, permit for every set of every input variable to define the corresponding set of the output variable called variable_output.

```
{
crisp_input: [NUMBER, ...],
variables_input: [
        {
                name: STRING,
                setsName: [STRING, STRING, STRING, ...],
```

```
                sets: [
                        [NUMBER, NUMBER, NUMBER, NUMBER],
                        [NUMBER, NUMBER, NUMBER, NUMBER],
                        [NUMBER, NUMBER, NUMBER, NUMBER],
                        ...
                ]
        },
        ...
    ],
    variable_output: {
        name: STRING,
        setsName: [STRING, STRING, ...],
        sets: [
            [NUMBER, NUMBER, NUMBER, NUMBER],
            [NUMBER, NUMBER, NUMBER, NUMBER],
            ...
        ]
    },
    inferences: [
        [id_Ref_Output_Set, id_Ref_Output_Set, id_Ref_Output_Set, ...],
        ...
    ]
}
```