**Fuzzy Rules, A Reminder**

Remember, when creating your rulebase, for a 3 input 1 output system for example, the following column indexes will be in use when creating your system in a script file (.m file):

***[****Input 1, Input 2, Input 3, Output 1, Weight, Operator****]***

No matter how many inputs (n) your system makes use of, the first most left column will always be indicative of **Input 1**, the second column will be indicative of **Input 2**, and so on and so forth.

(n + 1) is the column directly after the last specified input. In our case, this is column 4. n + 1 is therefore equal to 3 + 1 = 4, the 3 is the number of inputs, so therefore the column directly right of the last input will always be the **first** **Output**.

Column number 5 is the representative of the weight. Typically, this is set to 1, but it can be any value in the range of between 0 and 1. For example, the weight for a particular rule could be 0.8. The weight allows you to dilute the strength of the rule further still; even after the firing strength has been computed.

The final column, is always indicative of the operator that the rule is making use of. This column can take 1 of 3 values:

* 0 - Indicates that it is a THEN
* 1 - Indicates it is an AND
* 2 - Indicates it is an OR

For example, assume we had a Mamdani type fuzzy system with the following membership functions for the following inputs and output:

* **Input 1:**
* Small
* Medium
* Big
* **Input 2:**
* Light
* Moderate
* Heavy
* **Input 3:**
* Cold
* Moderate
* Warm
* **Output 1:**
* A
* B
* C

It doesn’t matter what the system is representing, but assume we has the following rule:

***[****1, 1, 1, 1, 1, 1****]***

This would equate to the following understanding:

*If* ***Input 1*** *is Small AND* ***Input 2*** *is Light AND* ***Input 3*** *is Cold THEN* ***Output 1*** *is A with a weighting of (1)*

***[****0, 0, 3, 3, 1, 0****]***

*If* ***Input 3*** *is Warm THEN* ***Output 1*** *is C with a weighting of (1)*

Please Remember that a single rule cannot make use of different operators, it has to be one type, for example the following rule will not be acceptable:

*If Input 1 is Big* ***OR*** *Input 2 is Heavy* ***AND*** *Input 3 is Warm THEN Output 1 is C with a weighting of (0.8)*

Instead, what you should do is break the rule down into sub-rules, for example the above rule can be written as 2 new rule sub-rules:

*If Input 1 is Big* ***OR*** *Input 2 is Heavy THEN Output 1 is C with a weighting of (0.8)*

*If Input 2 is Heavy* ***AND*** *Input 3 is Warm THEN Output 1 is C with a weighting of (0.8)*

Each rule of your system does not have to make use of all 3 inputs, you could use any combination of inputs to generate each rule. For example, you could have a rule with simply one input, or a rule that used just 2 inputs. If you want to have rules which make use of all 3 inputs, that is completely acceptable. As there will be many different variations of fuzzy systems, it will understandably be dependent on the application of the domain, that provides the basis for the structuring of the rules.