Unit 6

Fuzzy Inference

Motivation

Our ultimate goal is to be able to proceed IF-THEN rules involving vague linguistic expressions which are modeled by fuzzy sets.

Question: *What is still missing before we reach that goal?*

Nonchalantly speaking, fuzzy inference is about processing fuzzy rules.

The Basic Setup

Let us in the following consider a system with n inputs and one output. Assume that we have n linguistic variables

$$\begin{aligned} v_1 &= (N_1, G_1, T_1, X_1, M_1), \\ \vdots &= \vdots \\ v_n &= (N_n, G_n, T_n, X_n, M_n), \end{aligned}$$

associated to the n inputs of the system and one linguistic variable associated to the output:

$$v_y = (N_y, G_y, T_y, X_y, M_y)$$

Fuzzy Rule Base with m Rules

$$\begin{array}{llll} \text{IF} & \textit{cond}_1 & \text{THEN} & \textit{action}_1 \\ \vdots & \vdots & \vdots & \vdots \\ \text{IF} & \textit{cond}_m & \text{THEN} & \textit{action}_m \end{array}$$

The conditions $\textit{cond}_i$ and the actions $\textit{action}_i$ are expressions built up according to an appropriate syntax.

An Example of a General Syntax for Conditions

$$
\begin{array}{lcl}
\perp & := & \langle\text{exp}\rangle \text{ ;} \\
\langle\text{exp}\rangle & := & \langle\text{iscondition}\rangle \mid \text{“(”} \ \langle\text{exp}\rangle \ \langle\text{binary}\rangle \ \langle\text{exp}\rangle \ \text{“)”} \mid \\
 & & \text{“(not”} \ \langle\text{exp}\rangle \ \text{“)”;} \\
\langle\text{binary}\rangle & := & \text{“and”} \mid \text{“or” ;} \\
\langle\text{iscondition}\rangle & := & \langle N_i\rangle \ \text{“is”} \ \langle l^i_j\rangle \ \text{;}
\end{array}
$$

For some $i = 1, \ldots, n$, $\langle N_i\rangle$ may be expanded with the corresponding name of the i-th linguistic variable and $\langle l^i_j\rangle$ may be expanded with a corresponding term from T_i.

A Simple Syntax for Actions

$$\perp \quad := \quad \langle N_y \rangle \text{ “is” } \langle l_{y_j} \rangle \text{ ;}$$

$\langle l_{y_j} \rangle$ may be expanded with a corresponding term from T_y.

Example

Consider a system with two inputs and one output:

$$v_1 = (N_1 = \text{“}\varphi\text{”}, G_1, T_1 = \{\text{“nb”}, \text{“ns”}, \text{“z”}, \text{“ps”}, \text{“pb”}\},$$
$$X_1 = [-30, 30], M_1),$$

$$v_2 = (N_2 = \text{“}\dot{\varphi}\text{”}, G_2, T_2 = \{\text{“nb”}, \text{“ns”}, \text{“z”}, \text{“ps”}, \text{“pb”}\},$$
$$X_2 = [-30, 30], M_2),$$

$$v_y = (N_y = \text{“}f\text{”}, G_y, T_y = \{\text{“nb”}, \text{“ns”}, \text{“z”}, \text{“ps”}, \text{“pb”}\},$$
$$X_y = [-100, 100], M_y)$$

Example (cont'd)

$$
\begin{array}{ll}
\text{IF } (\varphi \text{ is z and } \dot{\varphi} \text{ is z}) & \text{THEN } f \text{ is z} \\
\text{IF } (\varphi \text{ is ns and } \dot{\varphi} \text{ is z}) & \text{THEN } f \text{ is ns} \\
\text{IF } (\varphi \text{ is ns and } \dot{\varphi} \text{ is ns}) & \text{THEN } f \text{ is nb} \\
\text{IF } (\varphi \text{ is ns and } \dot{\varphi} \text{ is ps}) & \text{THEN } f \text{ is z} \\
\vdots \quad \vdots & \vdots \qquad \vdots
\end{array}
$$

How can we define a control function from these rules?

[go to fuzzy sets]

What Do We Need?

1. We have to feed our input values into the system
2. We have to evaluate the truth values of the conditions
3. We have to come to some conclusions/actions for each rule
4. We have to come to an overall conclusion/action for the whole set of rules
5. We have to get an output value

Steps 3 and 4 are usually considered the steps of actual *inference*

Steps 1 and 2

Assume that we are given n crisp input values $x_i \in X_i$ $(i = 1, \ldots, n)$ and assume we have fixed a De Morgan triple (T, S, N).

Then we can compute the truth value $t(\textit{cond}_i)$ of each condition *cond*$_i$ recursively in the following way (assuming the syntax from the above example):

$$t(N_i \text{ is } l_j^i) = \mu_{M_i(l_j^i)}(x_i)$$

$$t(a \text{ and } b) = T(t(a), t(b))$$

$$t(a \text{ or } b) = S(t(a), t(b))$$

$$t(\text{not } a) = N(t(a))$$

Steps 3 and 4: Basic Remarks

1. It may happen that the conditions of two or more rules are fulfilled with a non-zero truth value
2. It may even happen that this is true for two or more rules with different (conflicting?) actions
3. This is not at all a problem, but a great advantage!
4. In any case, the following basic requirement is obvious: The higher the truth value of a rule's condition, the higher its influence on the output should be

Steps 3 and 4: Two Fundamental Approaches

Deductive interpretation: Rules are considered as logical deduction rules (implications)

Assignment interpretation: Rules are considered as conditional assignments (like in a procedural programming language)

Both approaches have in common that separate output/action fuzzy sets are computed for each rule. Finally, the output fuzzy sets of all rules are aggregated into one global output fuzzy set.

Step 3 in the Deductive Interpretation

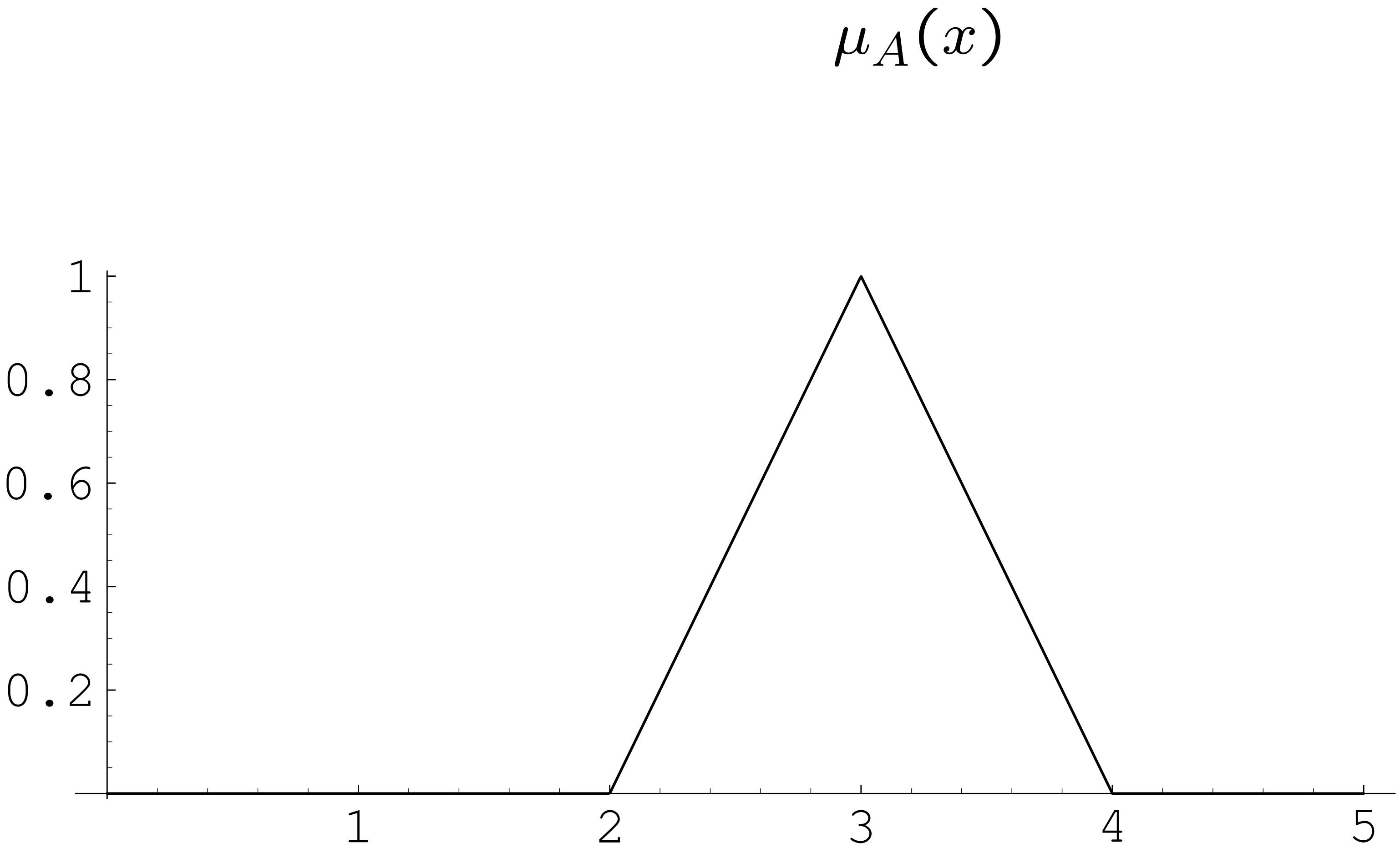
We fix a fuzzy implication $\tilde{I}$ in advance. Assume that we consider the i-th rule which looks as follows:

$$\text{IF } \textit{cond}_i \text{ THEN } N_y \text{ is } l^y_j$$

Assume that the condition *cond*$_i$ is fulfilled with a degree of t_i. Then the output fuzzy set O_i is defined in the following way:

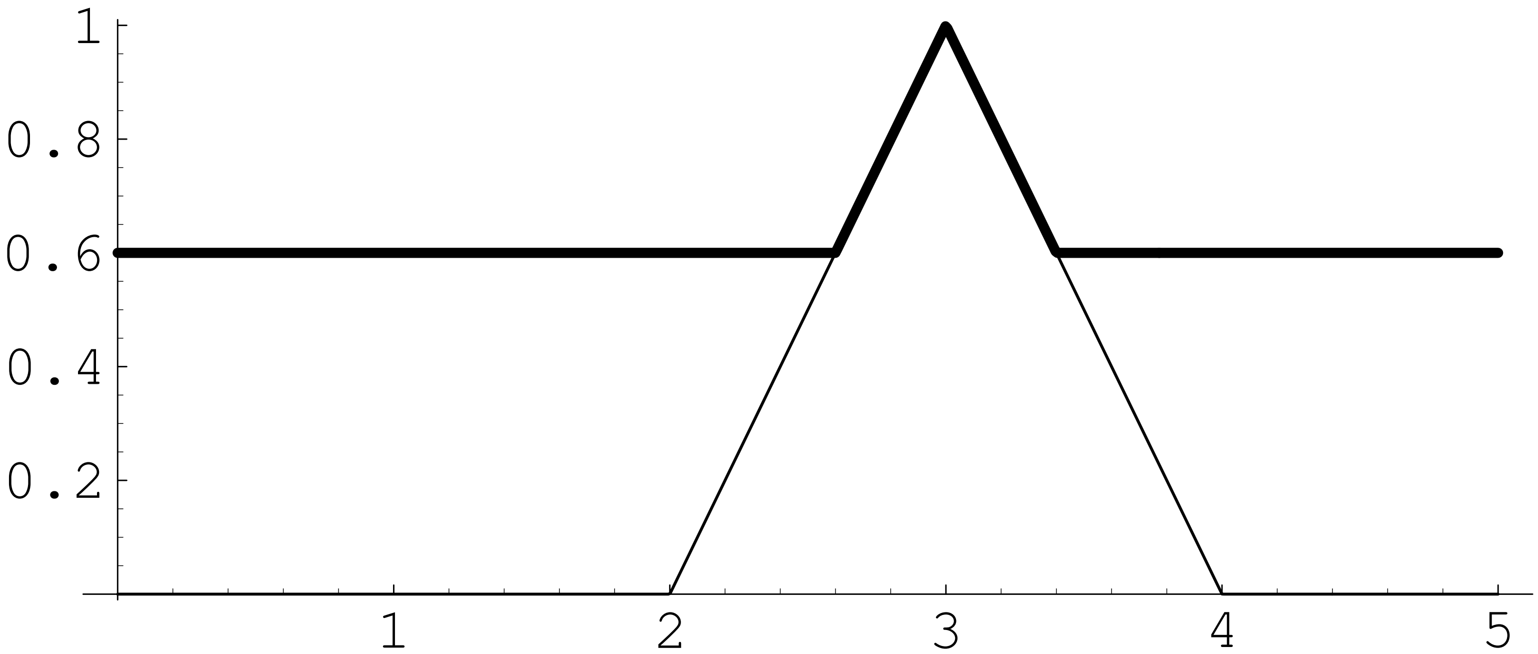
$$\mu_{O_i}(y) = \tilde{I}\big(t_i, \mu_{M(l^y_j)}(y)\big)$$

An Example



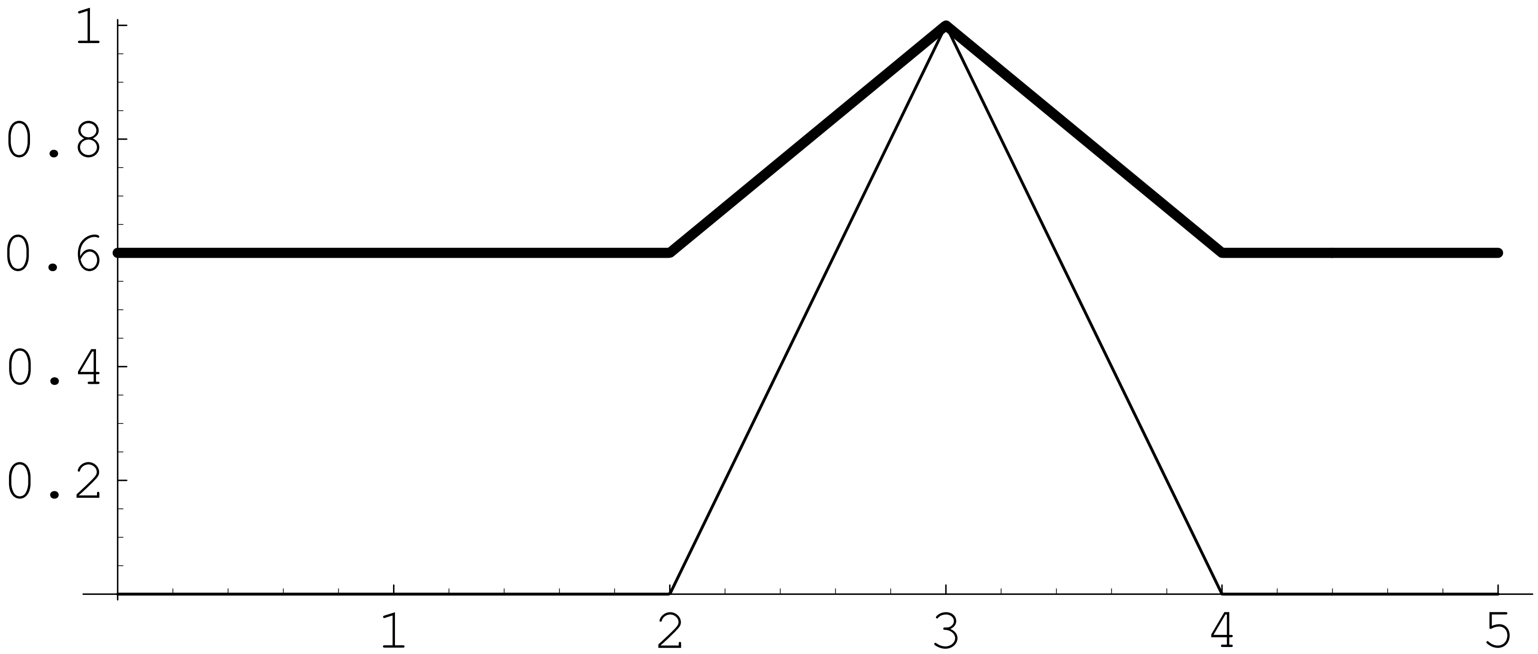
An Example

$$I_{S_{\mathbf{M}},N_{\mathbf{S}}}(0.4,\mu_A(x))$$



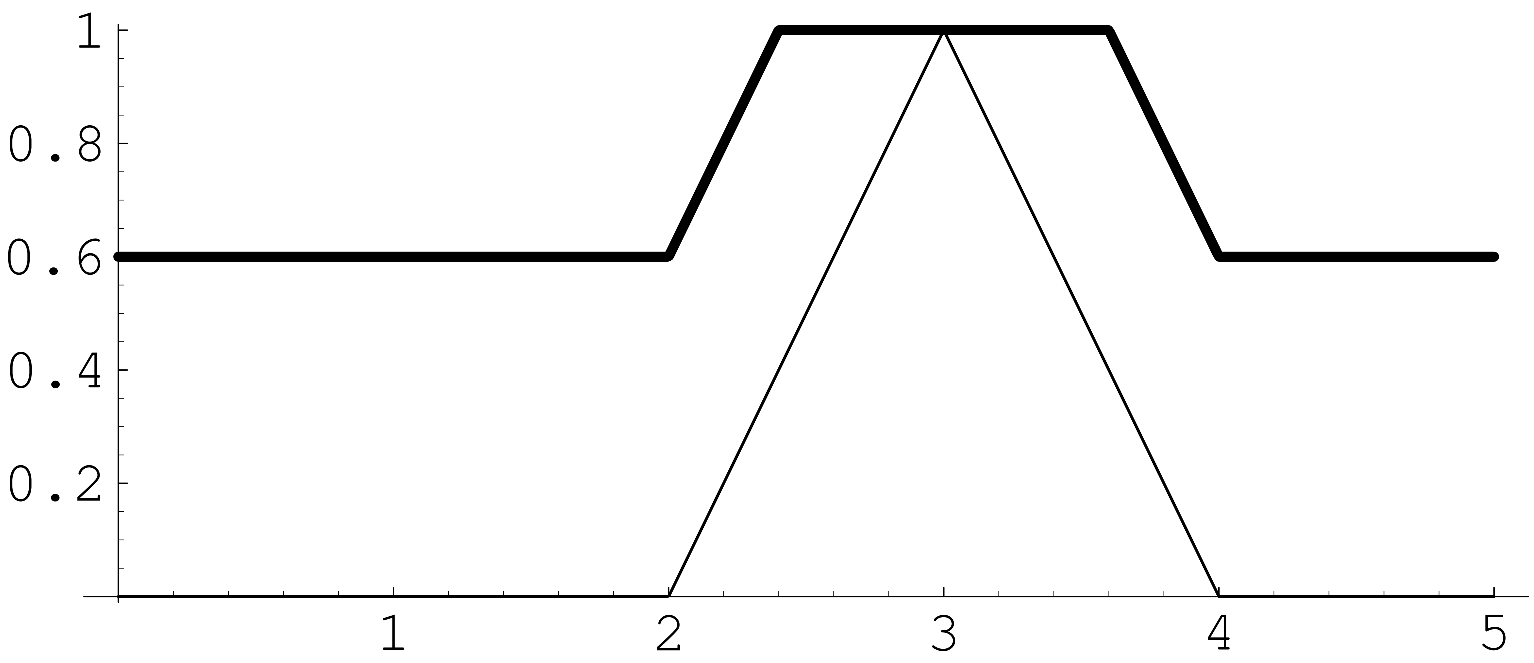
An Example

$$I_{S_{\mathbf{P}},N_{\mathbf{S}}}(0.4, \mu_A(x))$$



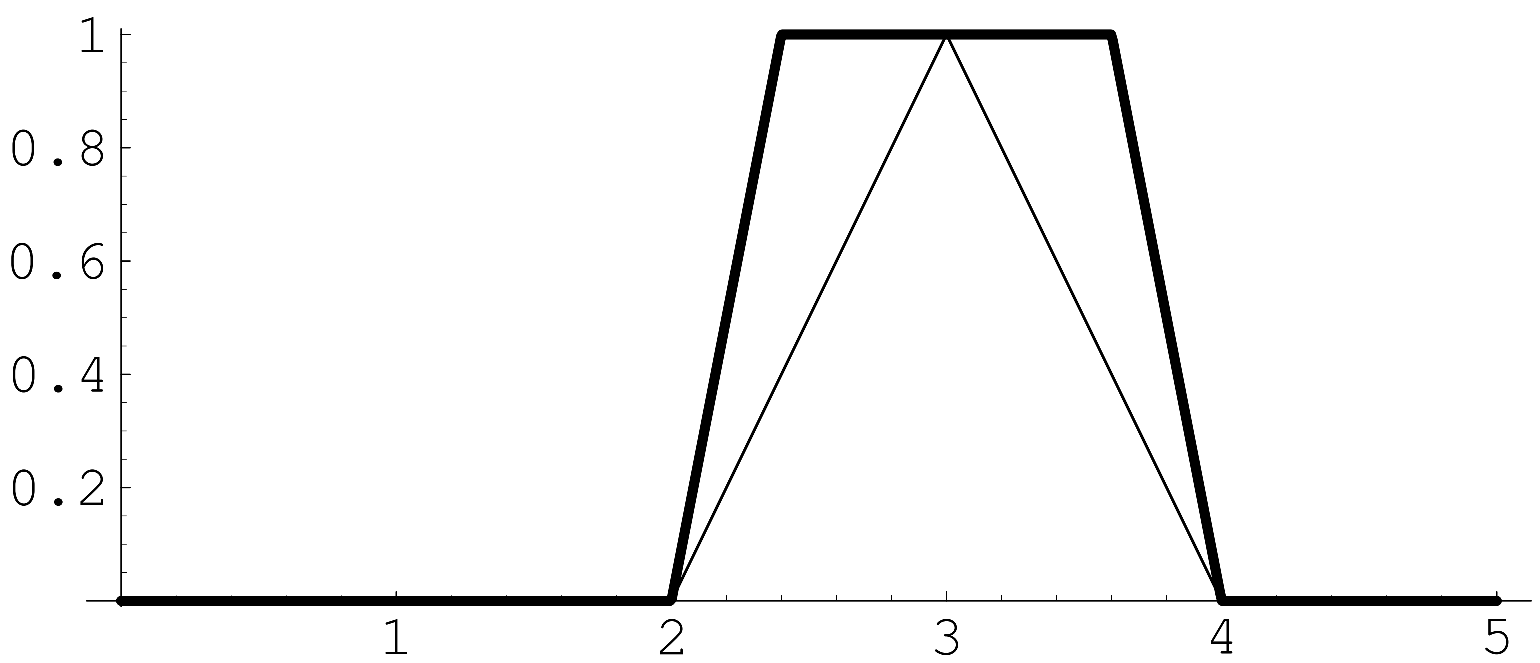
An Example

$$\overrightarrow{T}_{\mathbf{L}}(0.4, \mu_A(x))$$



An Example

$$\overrightarrow{T}_{\mathbf{P}}(0.4, \mu_A(x))$$



Step 4 in the Deductive Interpretation

We fix a t-norm $\tilde{T}$ in advance. Assume that the output fuzzy sets O_i of all rules ($i = 1, \ldots, m$) have been computed. Then the output fuzzy set $\tilde{O}$ is computed in the following way:

$$\mu_{\tilde{O}}(y) = \tilde{T}\big(\mu_{O_1}(y), \ldots, \mu_{O_m}(y)\big)$$

Step 3 in the Assignment Interpretation

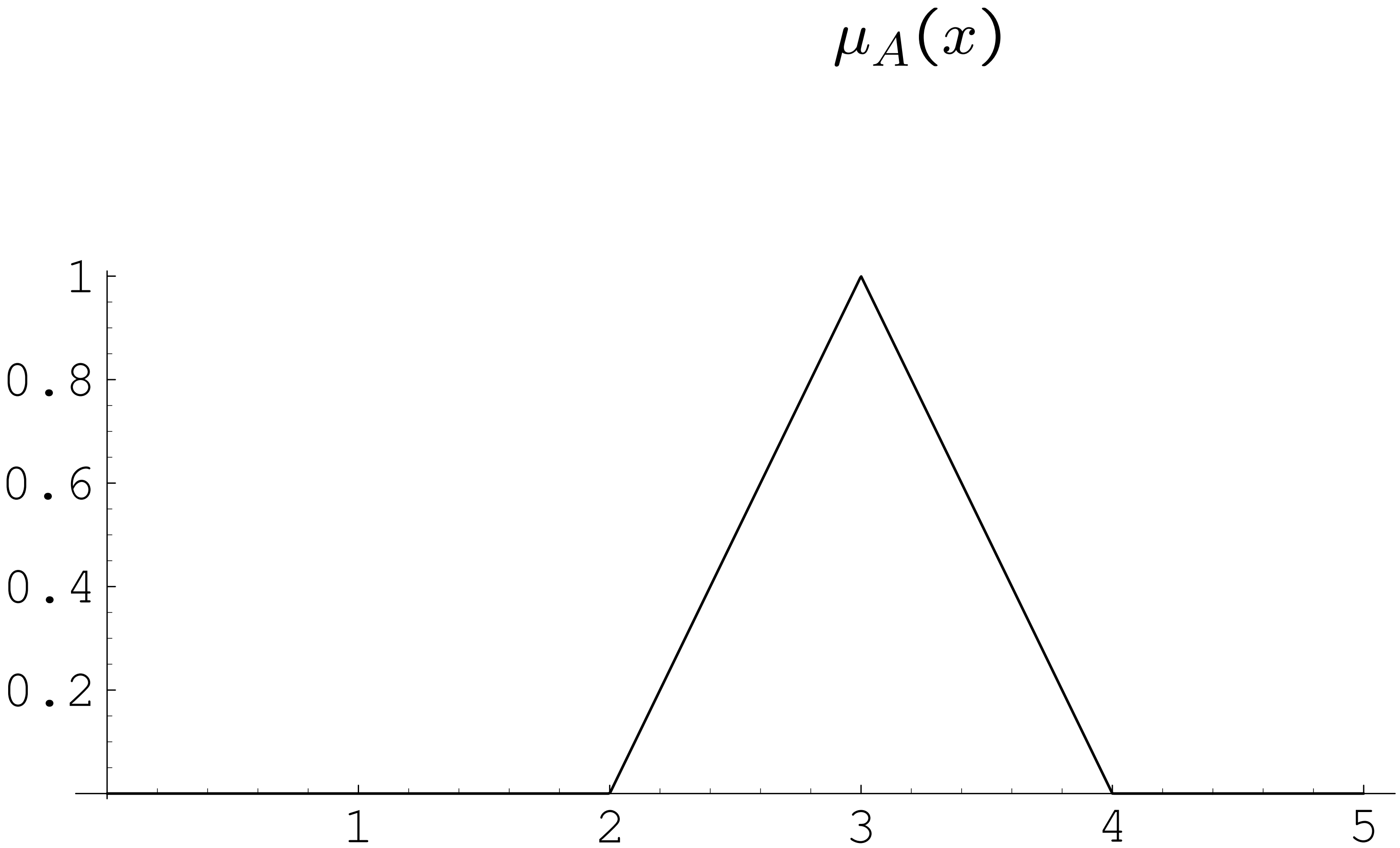
We fix a t-norm $\tilde{T}$ in advance. Assume that we consider the i-th rule which looks as follows:

$$\text{IF } \textit{cond}_i \text{ THEN } N_y \text{ is } l_j^y$$

Assume that the condition *cond*$_i$ is fulfilled with a degree of t_i. Then the output fuzzy set O_i is defined in the following way:

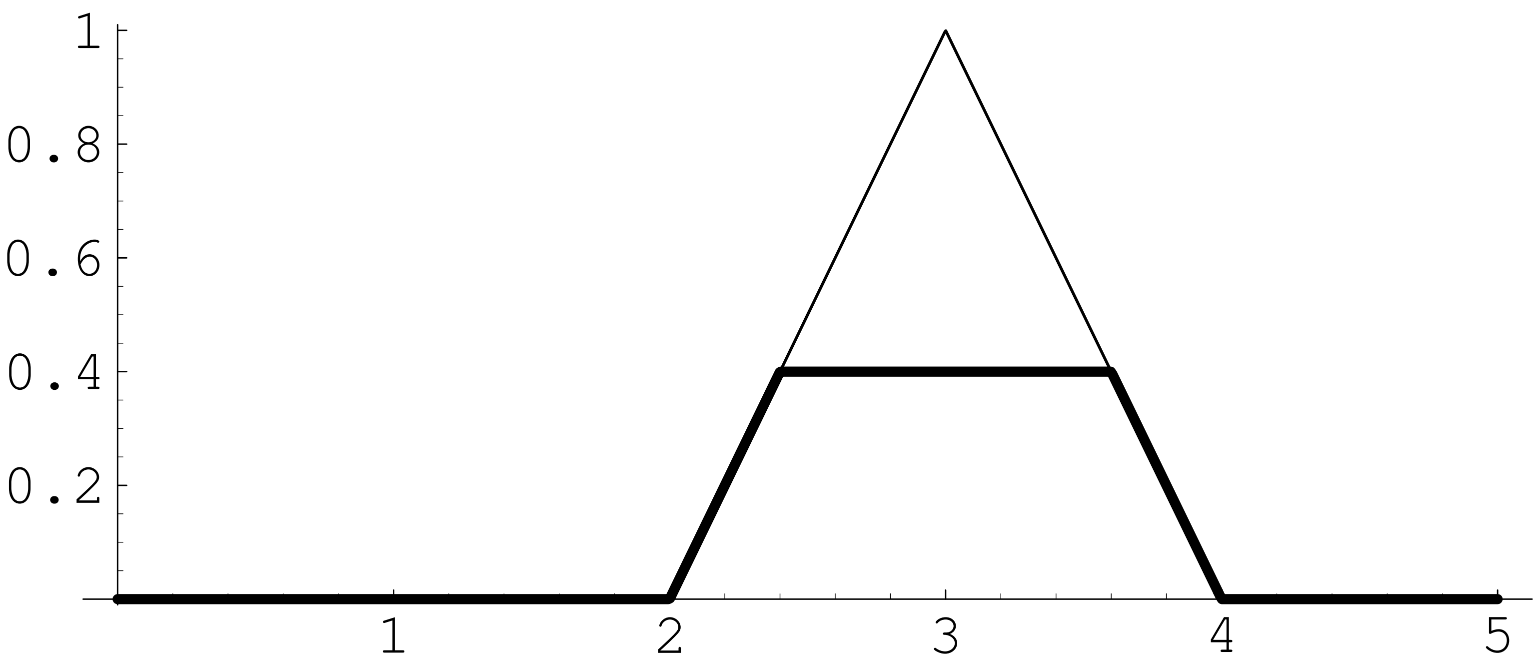
$$\mu_{O_i}(y) = \tilde{T}\big(t_i, \mu_{M(l_j^y)}(y)\big)$$

An Example



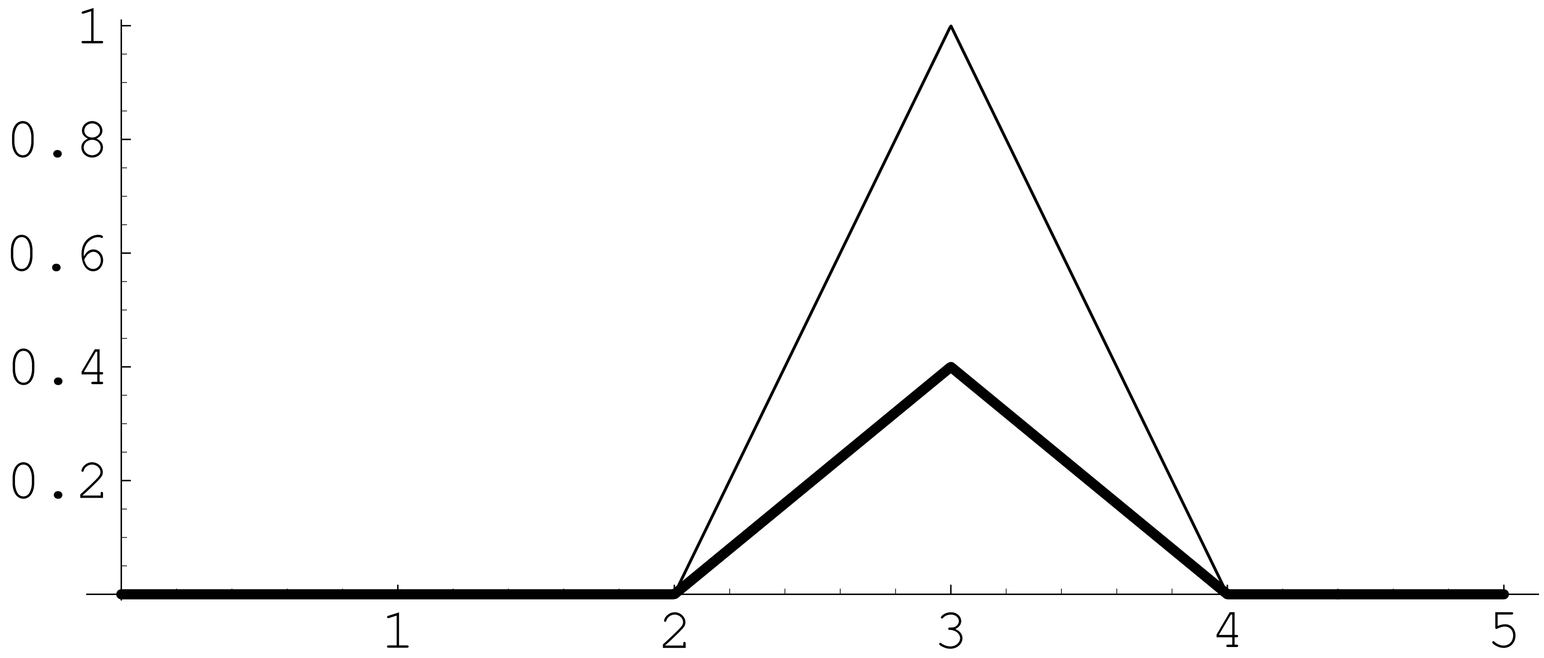
An Example

$$T_{\mathbf{M}}(0.4, \mu_A(x))$$



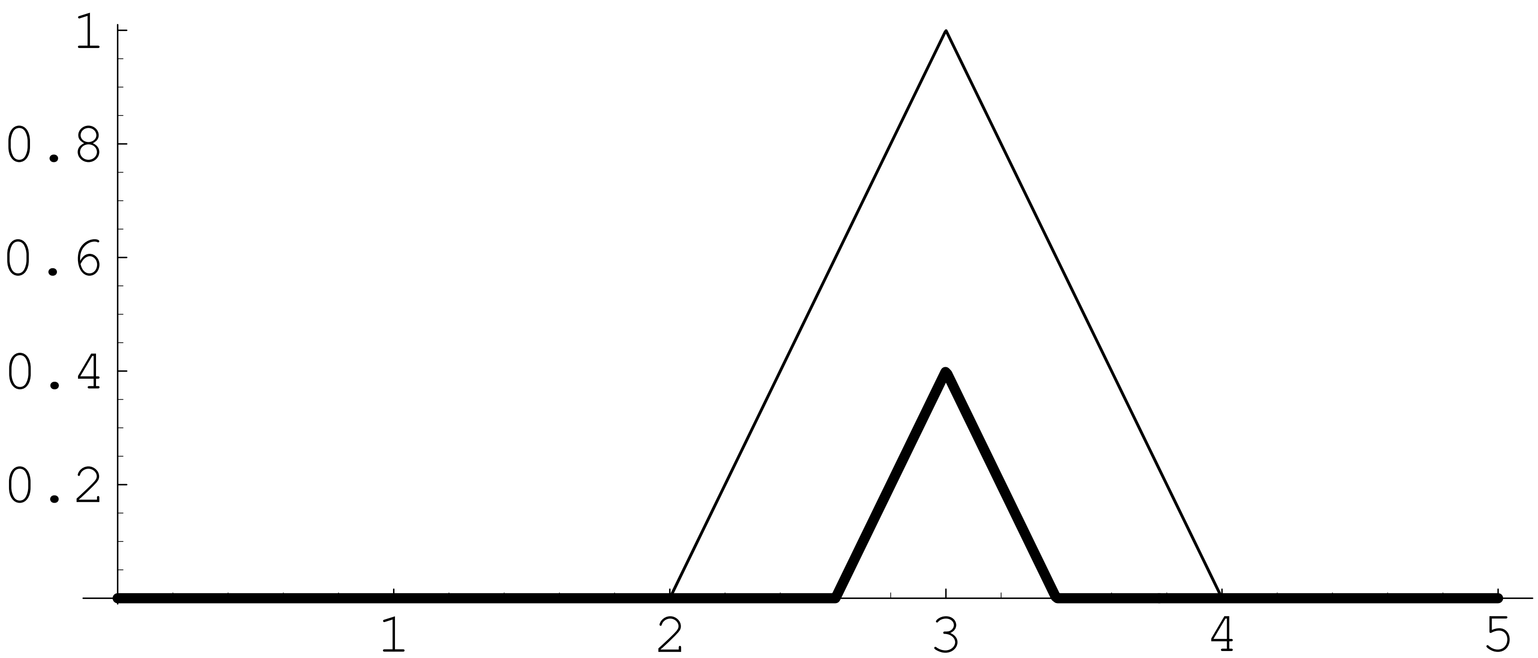
An Example

$$T_{\mathbf{P}}(0.4, \mu_A(x))$$



An Example

$$T_{\mathbf{L}}(0.4, \mu_A(x))$$



Step 4 in the Assignment Interpretation

We fix an aggregation operator $\tilde{A}$ in advance. Assume that the output fuzzy sets O_i of all rules ($i = 1, \ldots, m$) have been computed. Then the output fuzzy set $\tilde{O}$ is computed in the following way:

$$\mu_{\tilde{O}}(y) = \tilde{A}\big(\mu_{O_1}(y), \ldots, \mu_{O_m}(y)\big)$$

Some Remarks

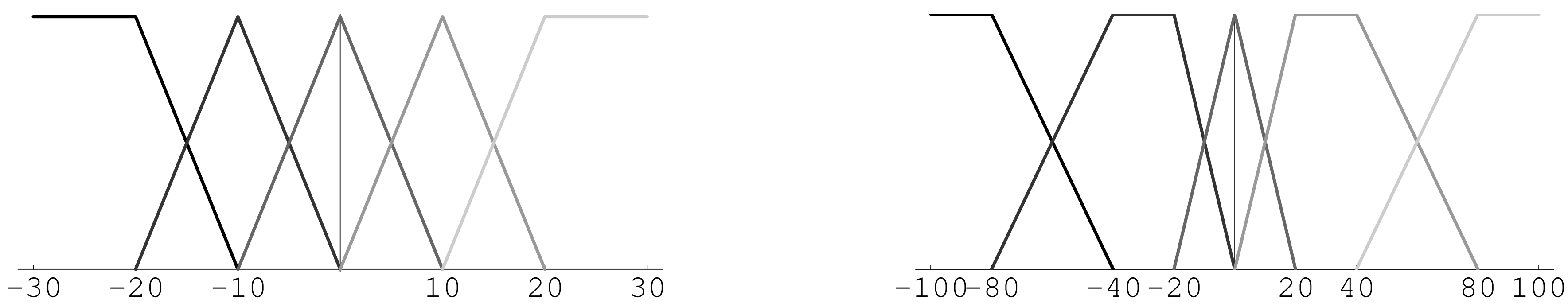
- The assignment interpretation is by far the more common one in practice. There is only one package that seriously offers the deductive interpretation (LFLC). It uses $\tilde{I} = \vec{T}_{\mathbf{L}}$ and $\tilde{T} = T_{\mathbf{M}}$.
- The most common variant of the assignment-based approach is $\tilde{T} = T_{\mathbf{M}}$ and $\tilde{A} = S_{\mathbf{M}}$. This classical variant is better known as *Mamdani/Assilian inference* or *max-min inference*. Another common variant uses $\tilde{T} = T_{\mathbf{P}}$ and the sum/arithmetic mean as aggregation $\tilde{A}$. This variant is often called *sum-prod inference.*

Example

We consider the rule base from the previous example.

[go back]

We define the following fuzzy sets for variables with names φ and $\dot{\varphi}$ (left) and f (right):



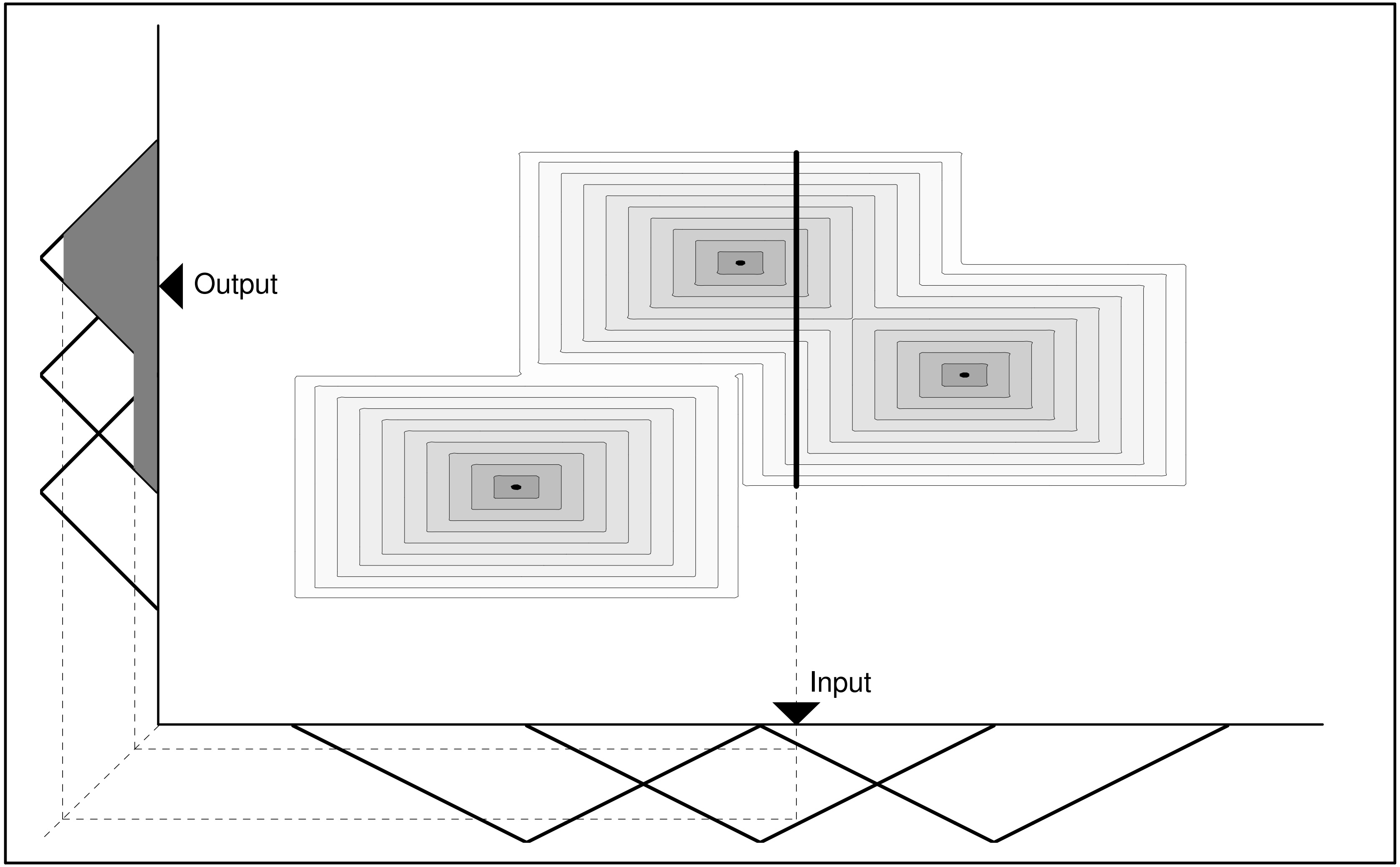
A Deeper Look Inside

- Each truth value t_i is from the unit interval and depending on the input vector $(x_1, \ldots, x_n)$. Therefore, we can consider t_i as a fuzzy set on $X_1 \times \cdots \times X_n$.
- For a given input vector $(x_1, \ldots, x_n)$ and an output value $y \in X_y$, the degree of relationship via the rule base is given as

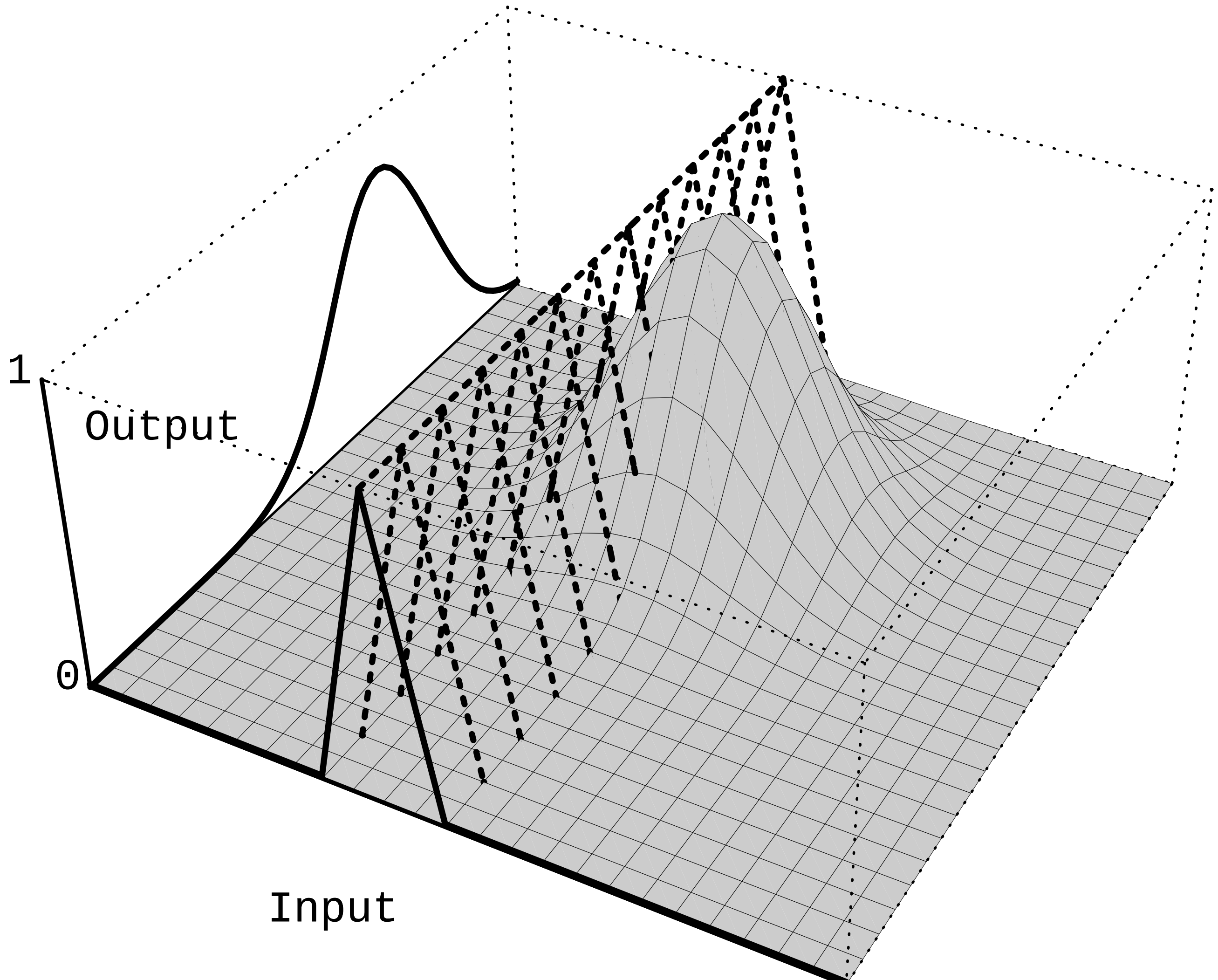
$$\tilde{I}\big(t_i(x_1, \ldots, x_n), \mu_{M(l_j^y)}(y)\big) \text{ or } \tilde{T}\big(t_i(x_1, \ldots, x_n), \mu_{M(l_j^y)}(y)\big).$$

  That means that each rule defines a fuzzy relation from $X_1 \times \cdots \times X_n$ to X_y.
- Correspondingly, the whole rule base defines a fuzzy relation from $X_1 \times \cdots \times X_n$ to X_y.

A Graphical Representation



What to Do With Fuzzy Inputs: CRI



Step 5: Defuzzification

In many applications, we need a crisp value as output. The following variants are common:

Mean of maximum (MOM): The output is computed as the center of gravity of the area where $\mu_{\tilde{O}}$ takes the maximum, i.e.

$$\xi_{\mathrm{MOM}}(\tilde{O}) := \frac{\int\limits_{\mathrm{Ceil}(\tilde{O})} y\, dy}{\int\limits_{\mathrm{Ceil}(\tilde{O})} 1\, dy},$$

where

$$\mathrm{Ceil}(\tilde{O}) := \{y \in X_y \mid \mu_{\tilde{O}}(y) = \{\mu_{\tilde{O}}(z) \mid z \in X_y\}\}$$

Step 5: Defuzzification (cont'd)

Center of gravity (COG): The output is computed as the center of gravity of the area under $\mu_{\tilde{O}}$:

$$\xi_{\mathrm{COG}}(\tilde{O}) := \frac{\int\limits_{X_y} y \cdot \mu_{\tilde{O}}(y)\, dy}{\int\limits_{X_y} \mu_{\tilde{O}}(y)\, dy}$$

Center of area (COA): The output is computed as the point which splits the area under $\mu_{\tilde{O}}$ into two equally-sized parts.

Summary: Deductive Interpretation

1. Feed our input values into the system: evaluate the truth degrees to which the inputs belong to the fuzzy sets associated to the linguistic terms
2. Evaluate the truth values of the conditions using fuzzy logical operations (a De Morgan triple (T, S, N))
3. Compute the conclusions/actions for each rule by connecting the truth value of the condition with the output fuzzy set using a fuzzy implication $\tilde{I}$
4. Compute the overall conclusion/action for the whole set of rules by aggregating the output fuzzy sets with a t-norm $\tilde{T}$
5. Use defuzzification to get a crisp output value (optional)

Summary: Assignment Interpretation

1. Analogous
2. Analogous
3. Compute the conclusions/actions for each rule by connecting the truth value of the condition with the output fuzzy set using a t-norm $\tilde{T}$
4. Compute the overall conclusion/action for the whole set of rules by aggregating the output fuzzy sets with an aggregation operator $\tilde{A}$ (most often a t-conorm)
5. Use defuzzification to get a crisp output value (optional)