fuzzylite
A Fuzzy Logic Control Library in C++

Juan Rada-Vilela

www.fuzzylite.com

Introduction

Introduction

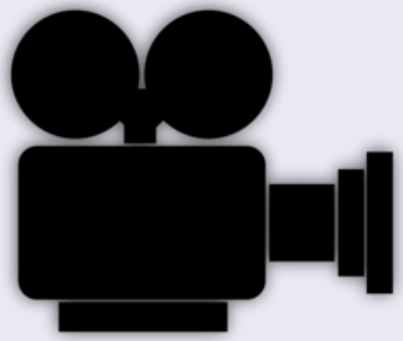
Are you familiar with. . . ?

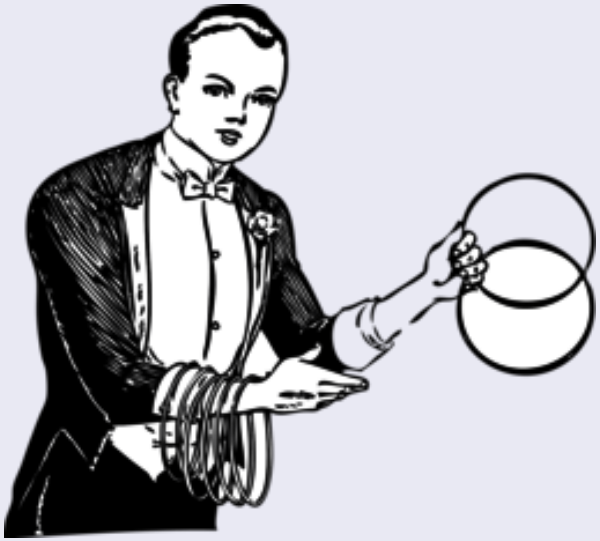
Introduction

Are you familiar with. . . ?

Introduction

Are you familiar with. . . ?





(Lord of the Rings)

Introduction

You should know that...

- Many of them use **Fuzzy Logic Control**
- Over **50 000** patents involve **Fuzzy Logic Control***
- Over **$10 000M** in product sales using **Fuzzy Logic Control**
- ...a *single* product: a blood pressure monitor*
- Profits estimated in **billions ($$$)** using **Fuzzy Logic Control**†

Can you believe that...?

State-of-the-art **FLC** libraries have **strong** limitations

*http://goo.gl/oYWGkM
†http://goo.gl/VDgOk9

State of the Art

Limitations

- Matlab and Fuzzy Logic Toolbox: **Costly** license
- Octave and Fuzzy Logic Toolkit: **Restrictive** license
- jFuzzyLogic: **Unfortunate** design choices
- Others: do **not** even bother...

`fuzzylite`: A Fuzzy Logic Control Library in C++

- **Free** and **open source**
- **Commercial friendly** license
- Mostly **fortunate** design choices
- **More** features
- Very **easy** to use
- Linux, Mac OSX, Windows, and others...

Objectives

Overall Goal

Introduction to **Fuzzy Logic Controllers**

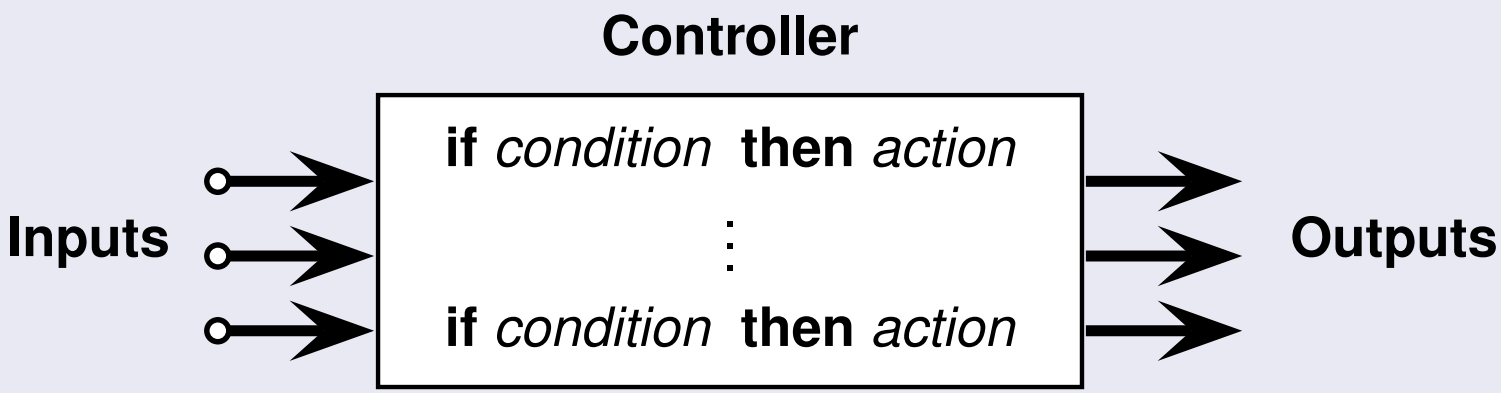
Specific Objectives

- **Design** of Fuzzy Logic Controllers
- **Operation** of Fuzzy Logic Controllers
- **Examples** of Fuzzy Logic Controllers
- **Description** of `fuzzylite`

Design

Fuzzy Logic Controller

Definition



Fuzzy Logic Controller

Example

Problem

How much to **tip** at a restaurant?

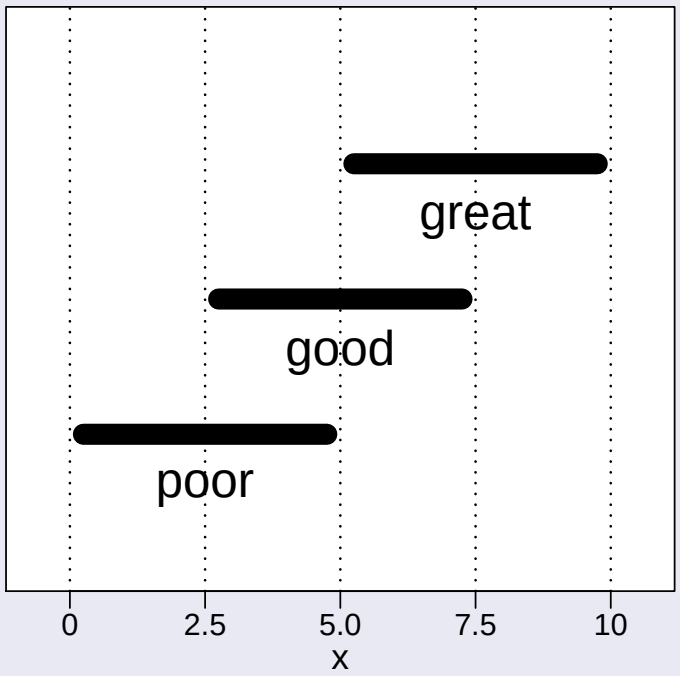
Solution

Controller

Service → if **Service** is **poor** then **Tip** is **cheap**
if **Service** is *good* then **Tip** is *average*
if **Service** is **great** then **Tip** is **generous** → Tip

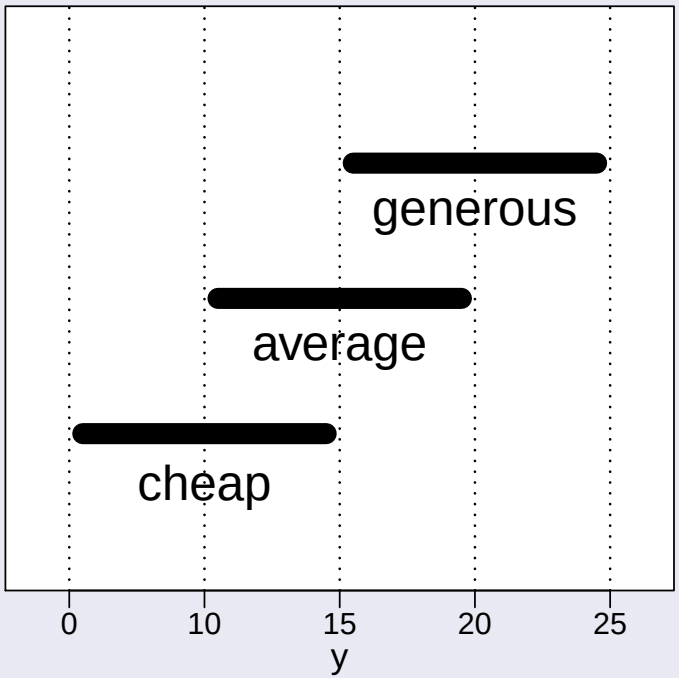
Linguistic Variables

Quality of Service



$S(2.5) = \text{poor}$

Tip

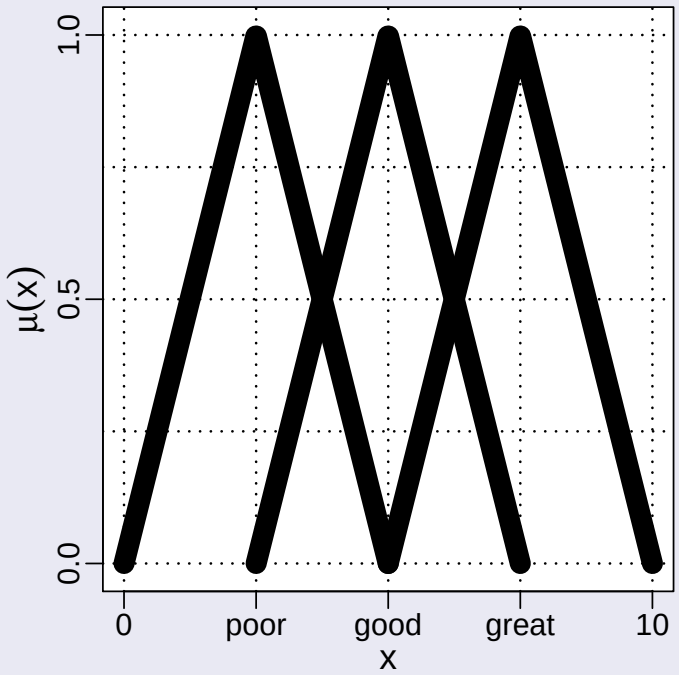


$T(17.5) = \text{average} + \text{generous}$

Linguistic variables with **crisp sets** as certainty $\in \{0, 1\}$

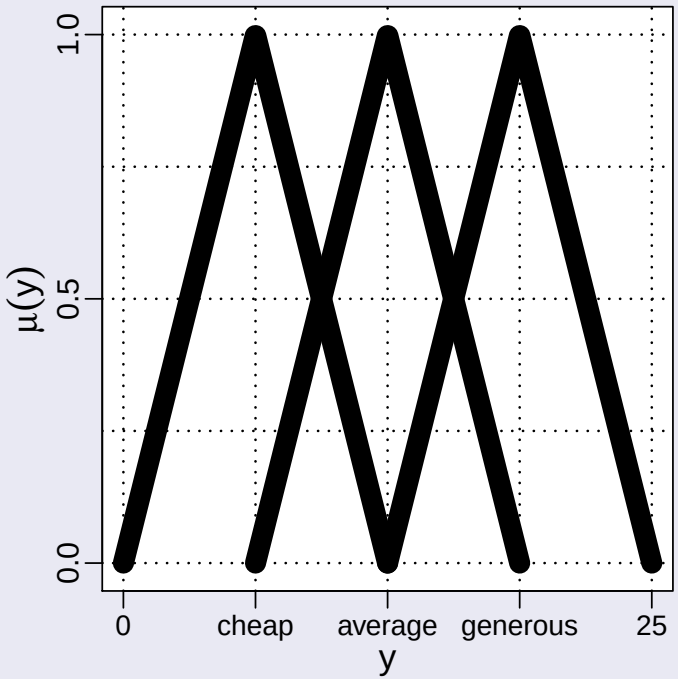
Linguistic Variables

Service



$$\tilde{S}(2.5) = 1.0/\text{poor}$$

Tip

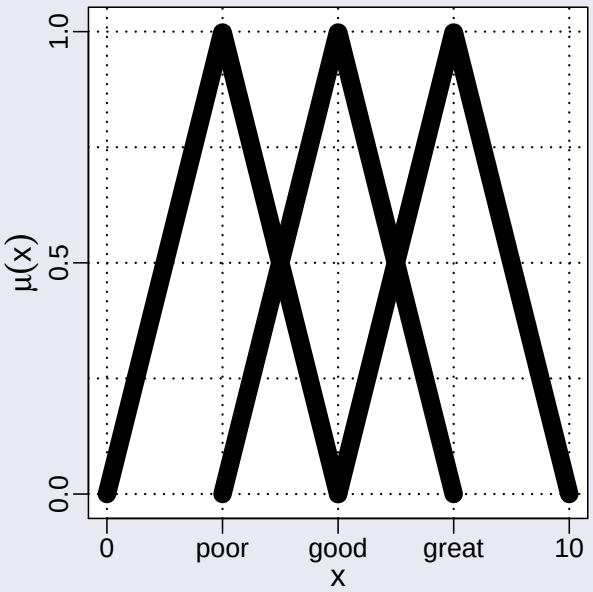


$$\tilde{T}(17.5) = 0.2/\text{average} + 0.8/\text{generous}$$

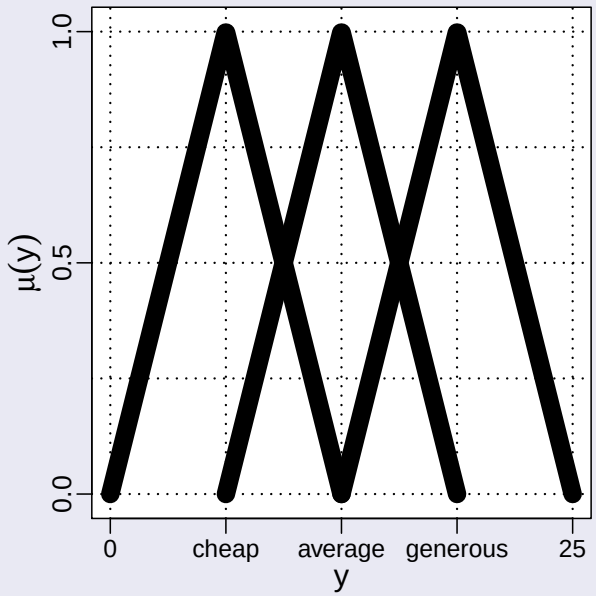
Linguistic variables with **fuzzy sets** as certainty $\mu \in [0.0, 1.0]$

Design of a Fuzzy Logic Controller

Input: Service



Output: Tip



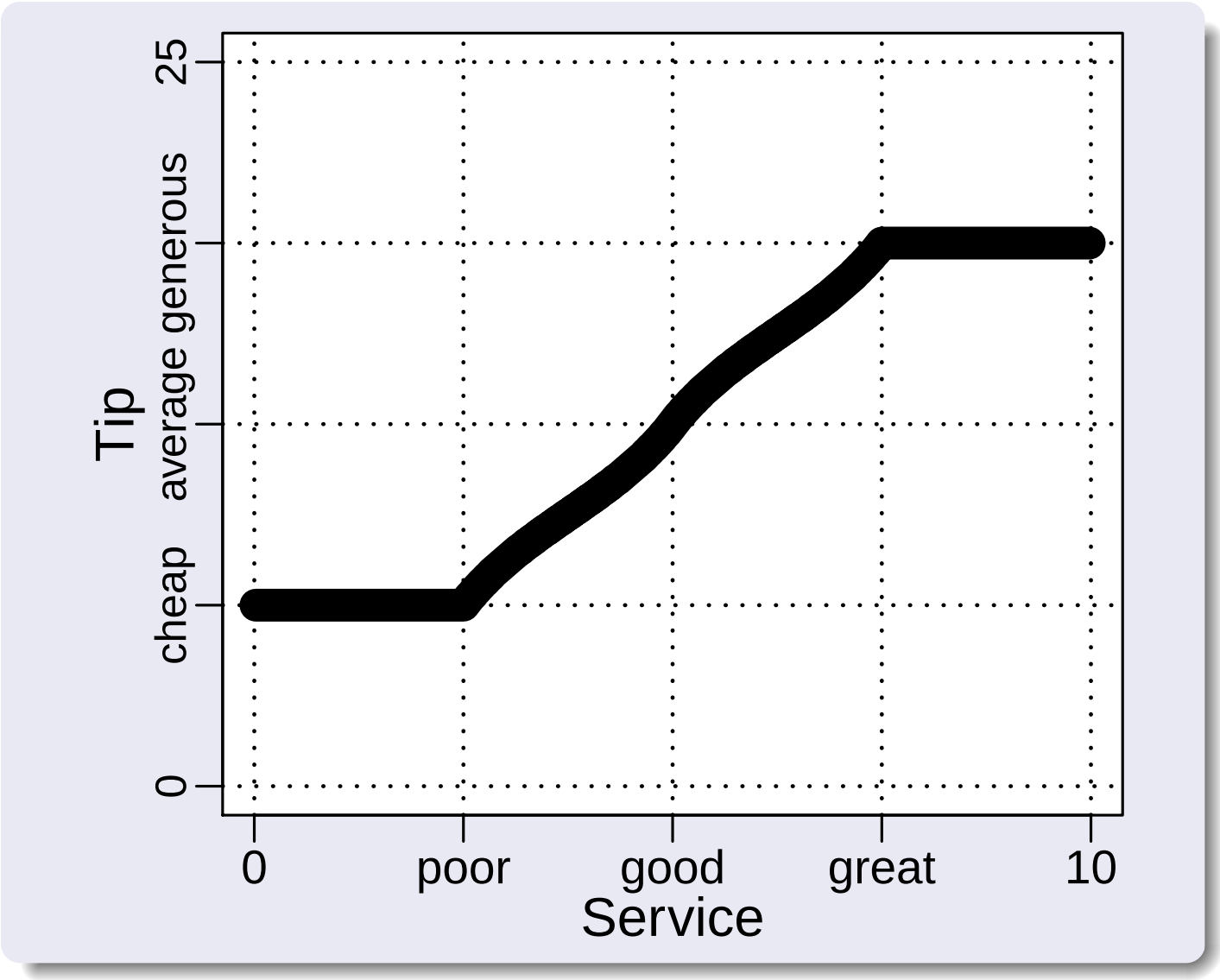
Rules

if **Service** is **poor** then **Tip** is **cheap**

if **Service** is *good* then **Tip** is *average*

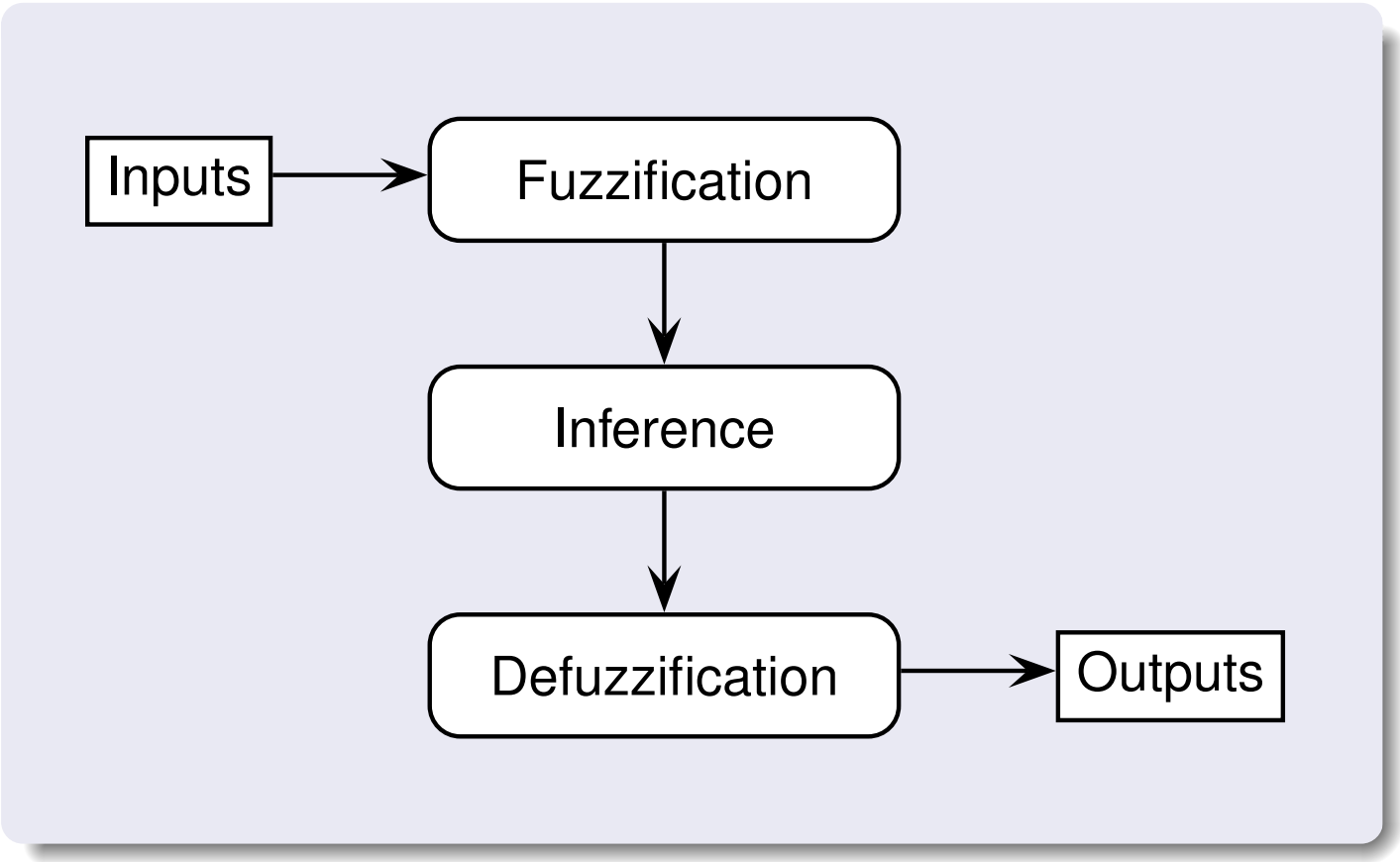
if **Service** is **great** then **Tip** is **generous**

Operation of a Fuzzy Logic Controller

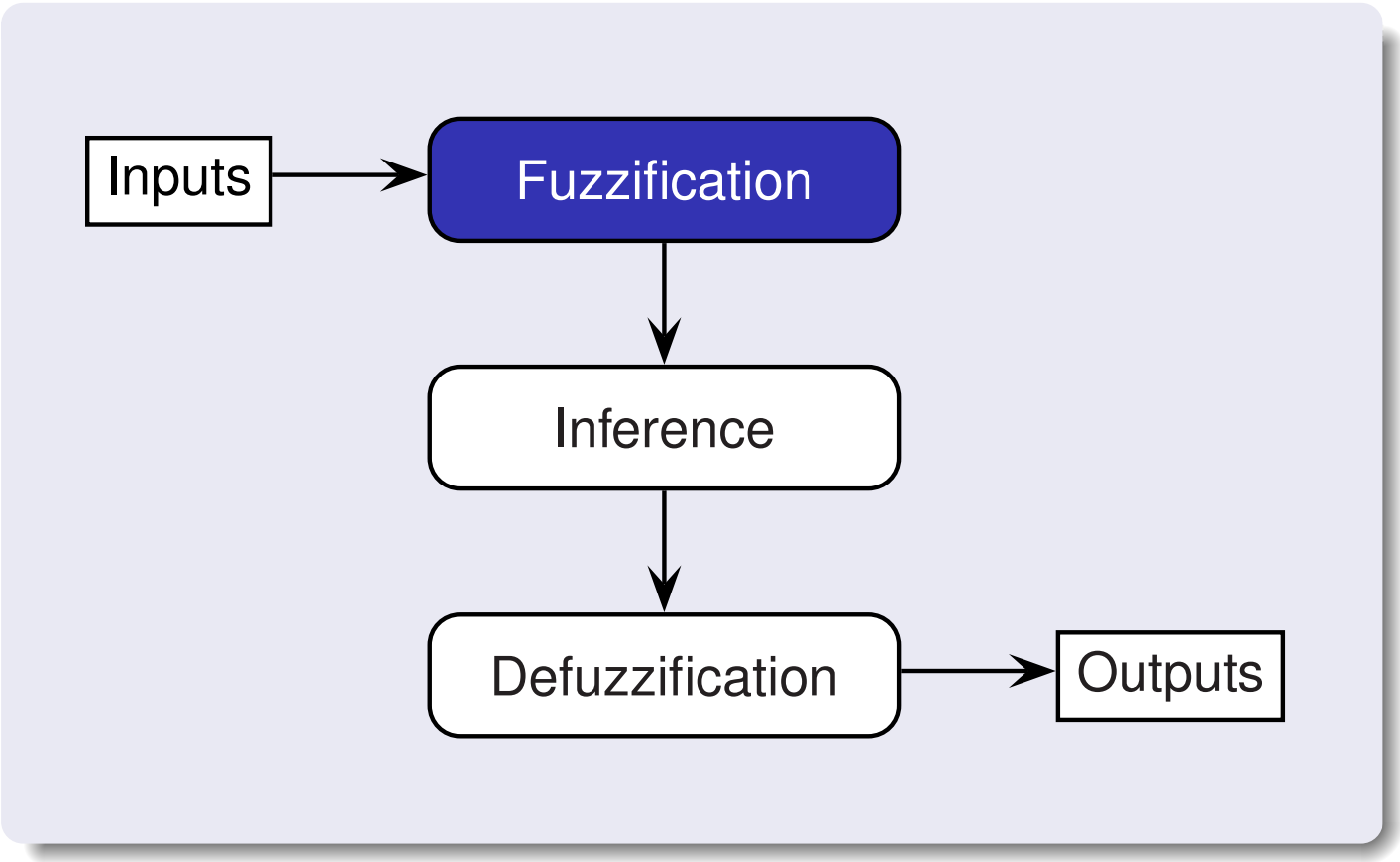


Operation

Stages of a Fuzzy Logic Controller



Stages of a Fuzzy Logic Controller



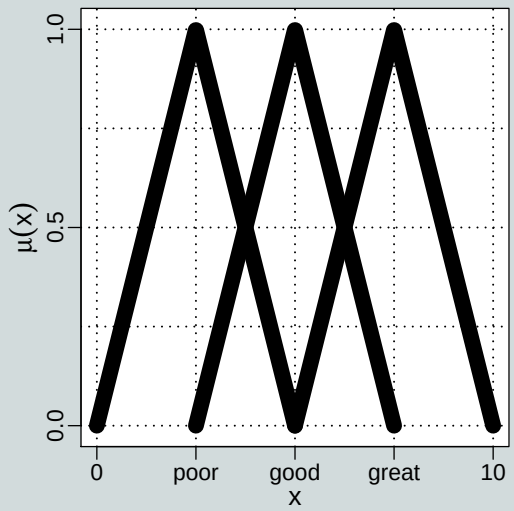
Fuzzification

Definition

Fuzzification: Converts **crisp** input values into a **fuzzy** set

Example

Service



Fuzzification

$$\tilde{S}(1.0) = 0.4/\text{poor} + 0.0/\text{good} + 0.0/\text{great}$$

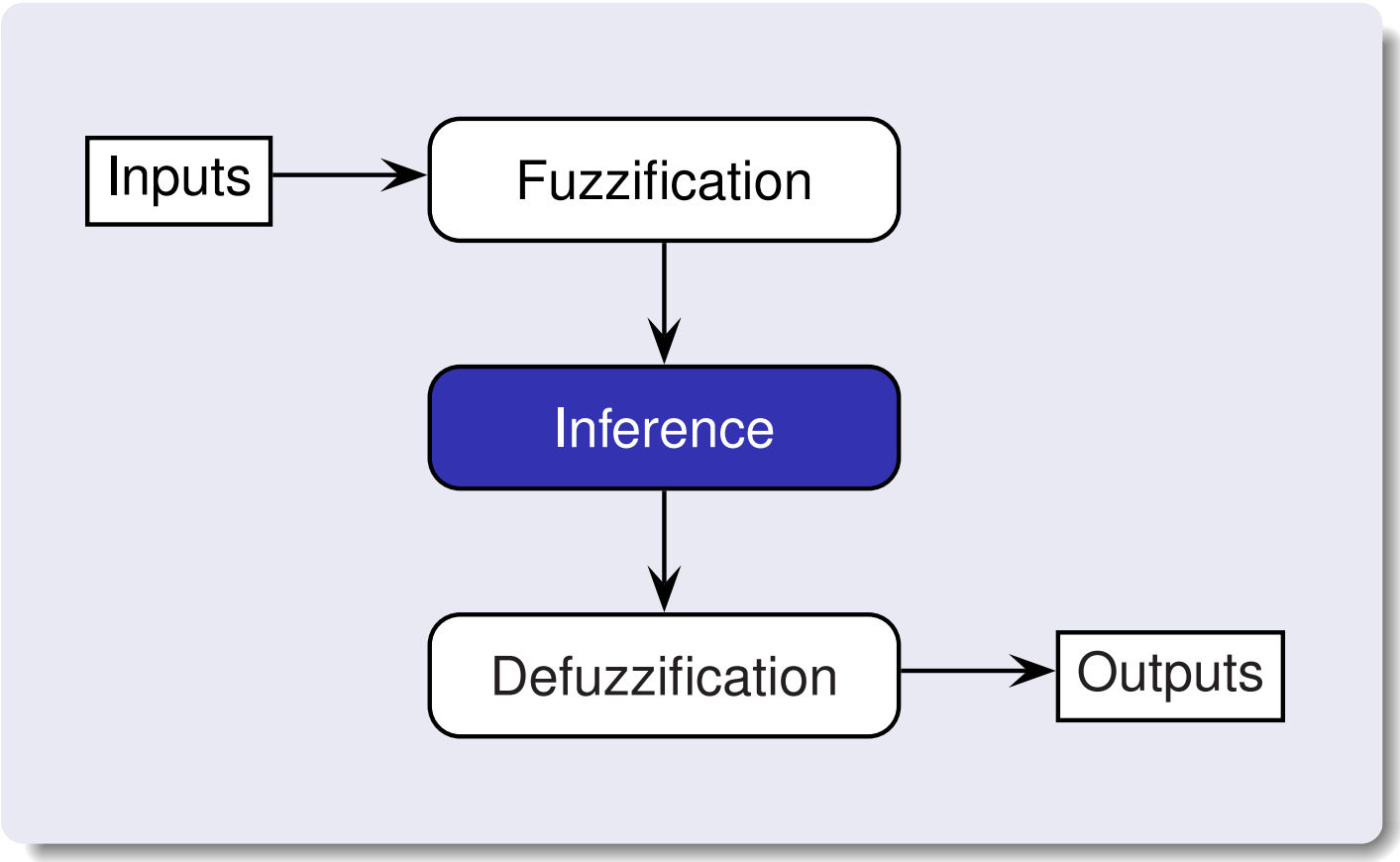
$$\tilde{S}(2.5) = 1.0/\text{poor} + 0.0/\text{good} + 0.0/\text{great}$$

$$\tilde{S}(7.0) = 0.0/\text{poor} + 0.2/\text{good} + 0.8/\text{great}$$

$$\tilde{S}(x) = \sum_{i \in S} \mu_i(x)/i$$

μ_i : **membership function** of term i

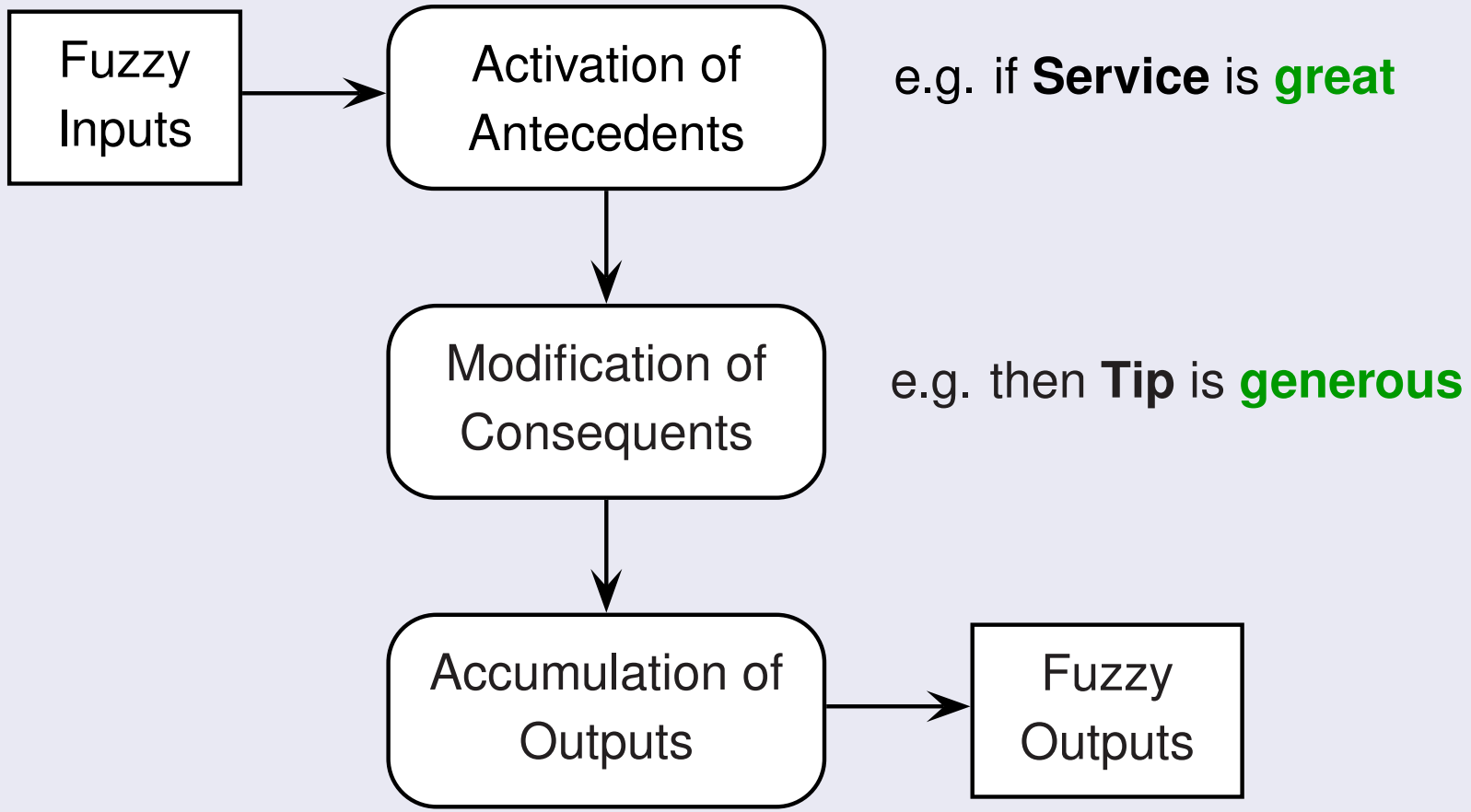
Stages of a Fuzzy Logic Controller



Inference

Definition

Inference: activates the rules to generate **fuzzy** outputs



Inference

Activation of the Antecedents

Example

if **Service** is **poor**	then **Tip** is **cheap**
if **Service** is *good*	then **Tip** is *average*
if **Service** is **great**	then **Tip** is **generous**

$$\tilde{S}(2.5) = 1.0/\text{poor} + 0.0/\text{good} + 0.0/\text{great}$$

Activation

if (**Service** is **poor**) $= 1.0$
if (**Service** is *good*) $= 0.0$
if (**Service** is **great**) $= 0.0$

Activation degree

Inference

Activation of the Antecedents

Example

if **Service** is **poor**	then **Tip** is **cheap**
if **Service** is *good*	then **Tip** is *average*
if **Service** is **great**	then **Tip** is **generous**

$$\tilde{S}(7.0) = 0.0/\text{poor} + 0.2/\text{good} + 0.8/\text{great}$$

Activation

if (**Service** is **poor**) $= 0.0$
if (**Service** is *good*) $= 0.2$
if (**Service** is **great**) $= 0.8$

Inference

Modification of Consequents

Example

if **Service** is **poor**	then **Tip** is **cheap**
if **Service** is *good*	then **Tip** is *average*
if **Service** is **great**	then **Tip** is **generous**

$$\tilde{S}(2.5) = 1.0/\text{poor} + 0.0/\text{good} + 0.0/\text{great}$$

Modification

then **Tip** is $(1.0 \otimes$ **cheap**$)$
then **Tip** is $(0.0 \otimes$ *average*$)$
then **Tip** is $(0.0 \otimes$ **generous**$)$

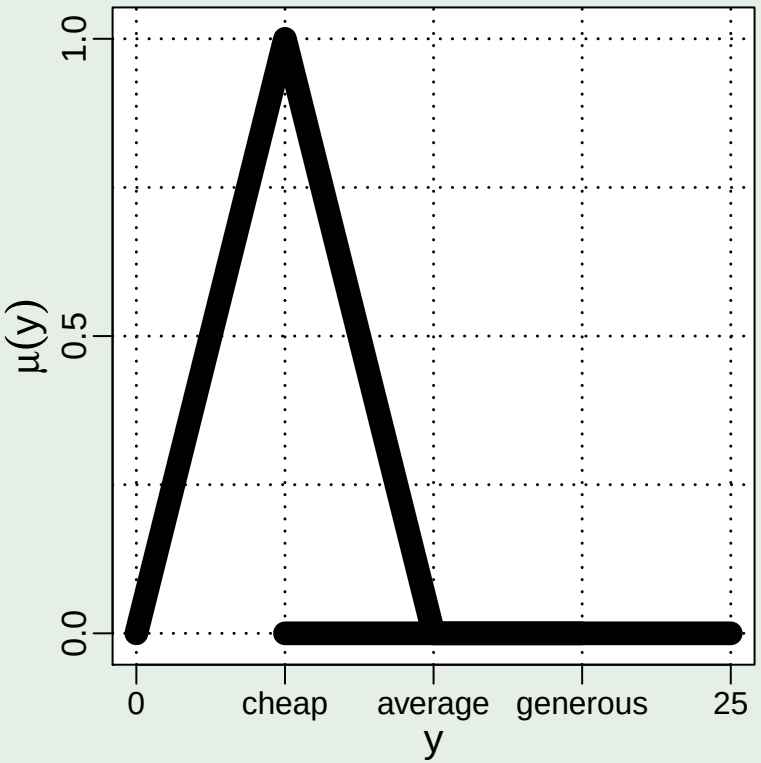
$\otimes$: **Activation Operator**

Inference

Modification of Consequents

Example

$$\{(1.0 \otimes \textbf{cheap}), (0.0 \otimes \textit{average}), (0.0 \otimes \textbf{generous})\}$$

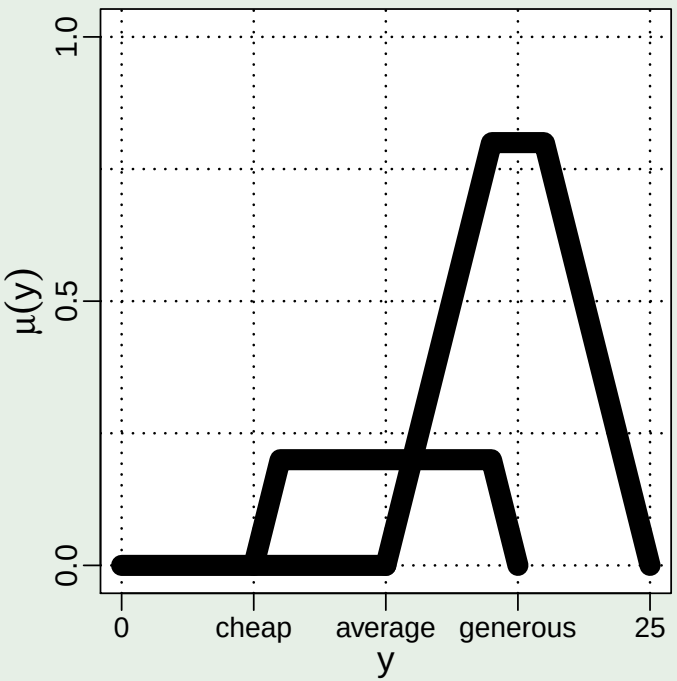


Inference

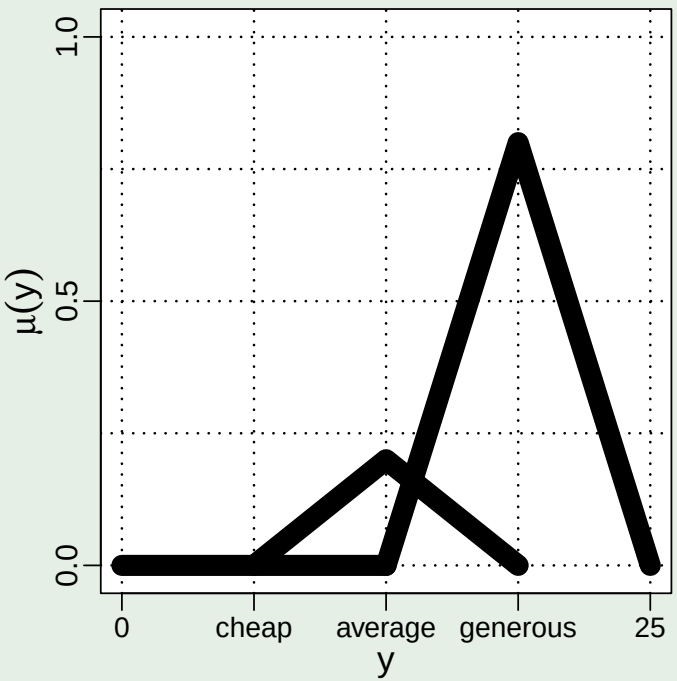
Modification of Consequents

Example

$$\{(0.0 \otimes \textbf{cheap}), (0.2 \otimes \textit{average}), (0.8 \otimes \textbf{generous})\}$$



$$\otimes : \min(\mu_i, \mu_j(x))$$



$$\otimes : \prod(\mu_i, \mu_j(x))$$

Inference

Accumulation of Consequents

Example

$$\tilde{S}(7.0) = 0.0/\text{poor} + 0.2/\text{good} + 0.8/\text{great}$$

Activation

if (**Service** is **poor**) $= 0.0$
if (**Service** is *good*) $= 0.2$
if (**Service** is **great**) $= 0.8$

Modification

then **Tip** is $(0.0 \otimes$ **cheap**$)$
then **Tip** is $(0.2 \otimes$ *average*$)$
then **Tip** is $(0.8 \otimes$ **generous**$)$

Accumulation

$$\tilde{T}_{7.0} = (0.0 \otimes \textbf{cheap}) \oplus (0.2 \otimes \textit{average}) \oplus (0.8 \otimes \textbf{generous})$$

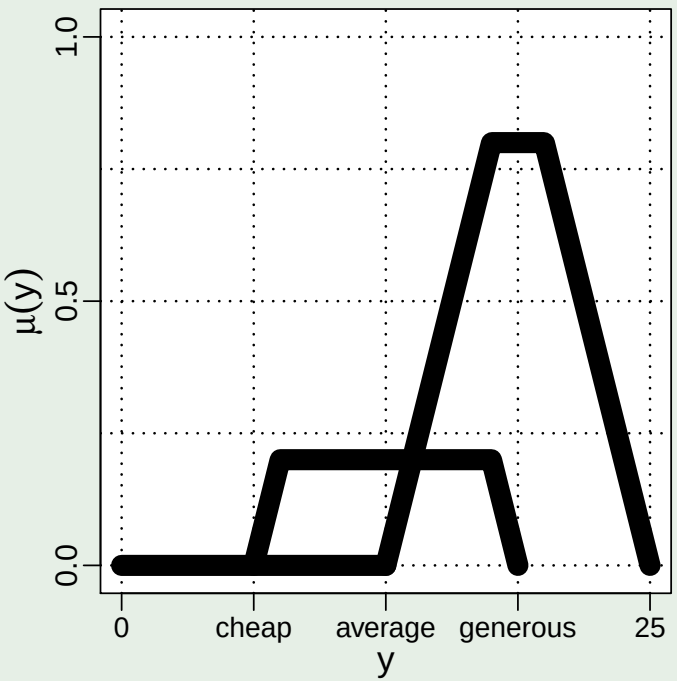
$\oplus$: **Accumulation Operator**

Inference

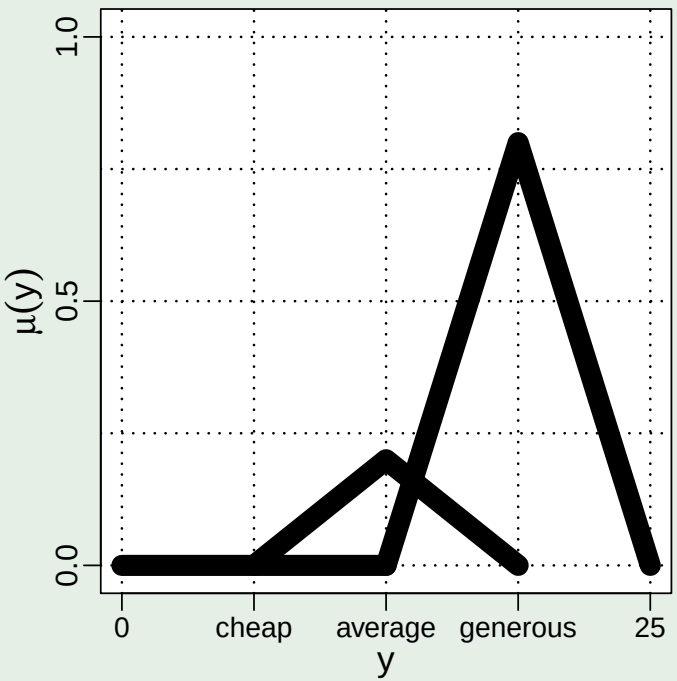
Modification of Consequents

Example

$$\{(0.0 \otimes \textbf{cheap}), (0.2 \otimes \textit{average}), (0.8 \otimes \textbf{generous})\}$$



$$\otimes : \min(\mu_i, \mu_j(x))$$



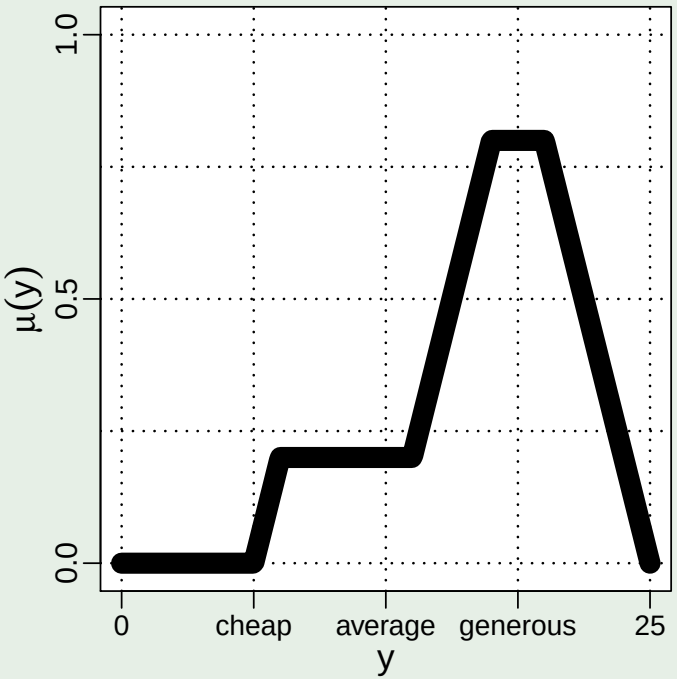
$$\otimes : \prod(\mu_i, \mu_j(x))$$

Inference

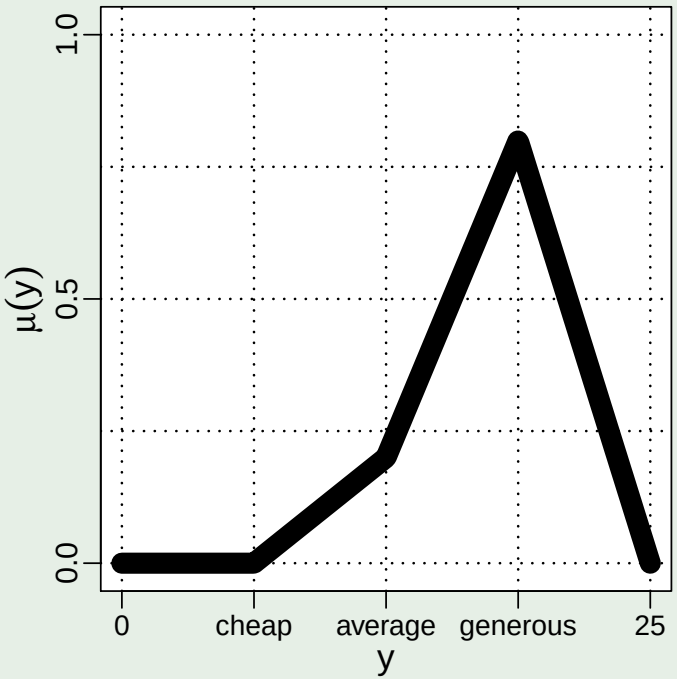
Accumulation of Consequents

Example

$$\tilde{T}_{7.0} = (0.0 \otimes \textbf{cheap}) \oplus (0.2 \otimes \textit{average}) \oplus (0.8 \otimes \textbf{generous})$$

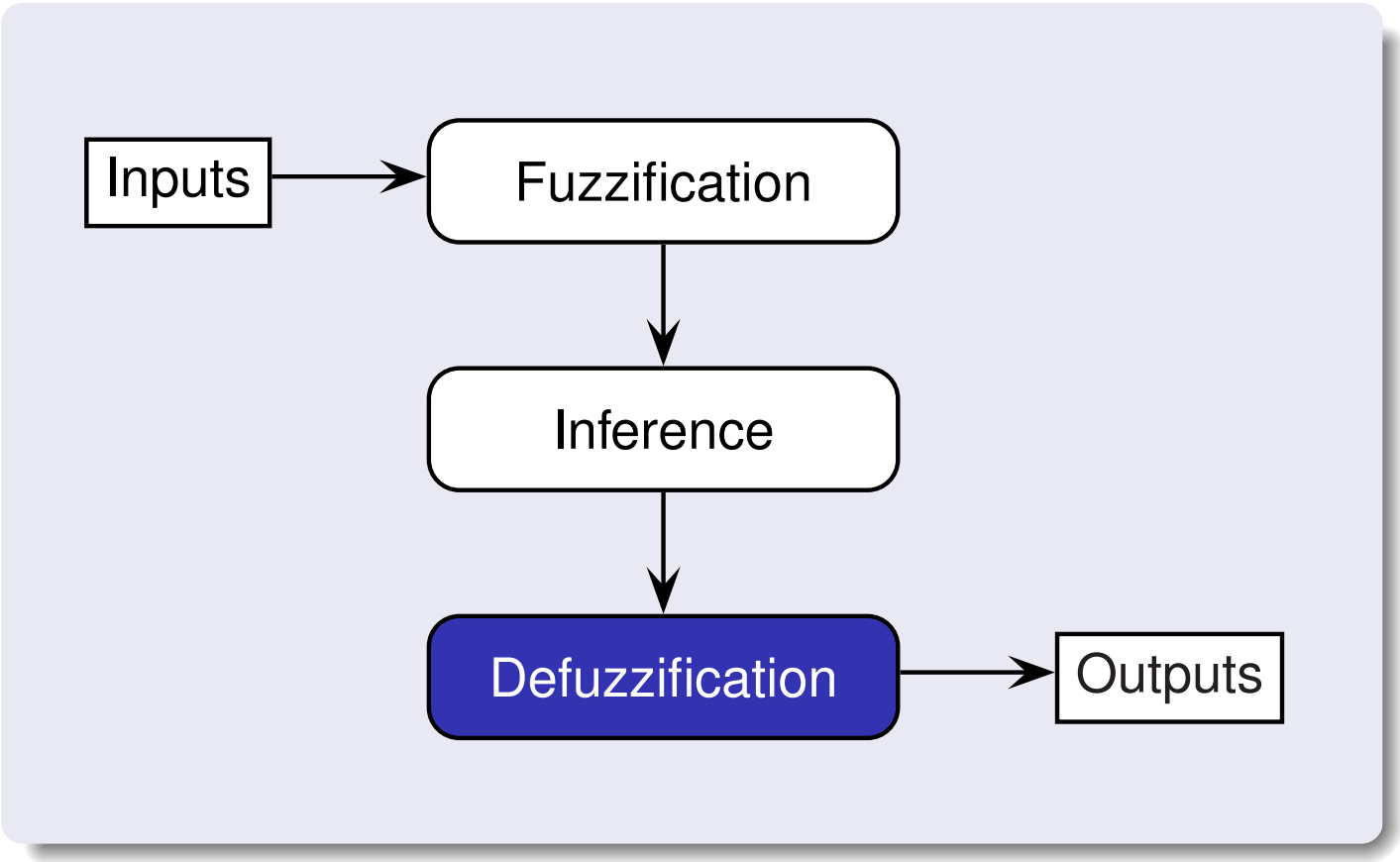


$$\otimes : \min(\mu_i, \mu_j(x))$$
$$\oplus : \max(\mu_i, \mu_j(x))$$



$$\otimes : \prod(\mu_i, \mu_j(x))$$
$$\oplus : \sum(\mu_i, \mu_j(x))$$

Stages of a Fuzzy Logic Controller



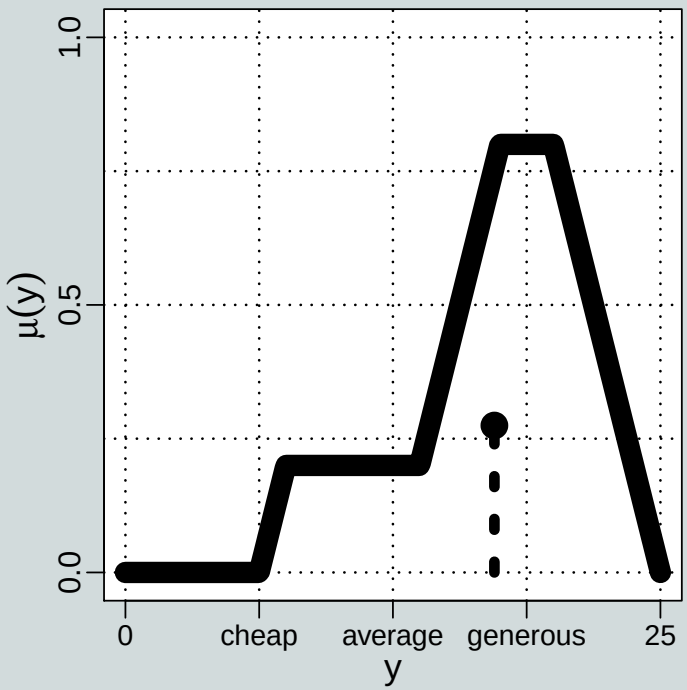
Defuzzification

Definition

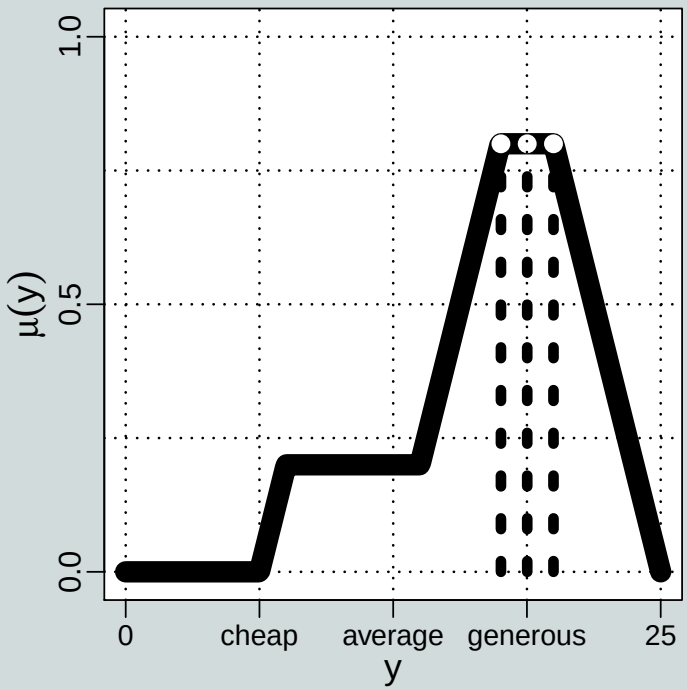
Defuzzification: converts the **fuzzy** outputs into **crisp** values

Example

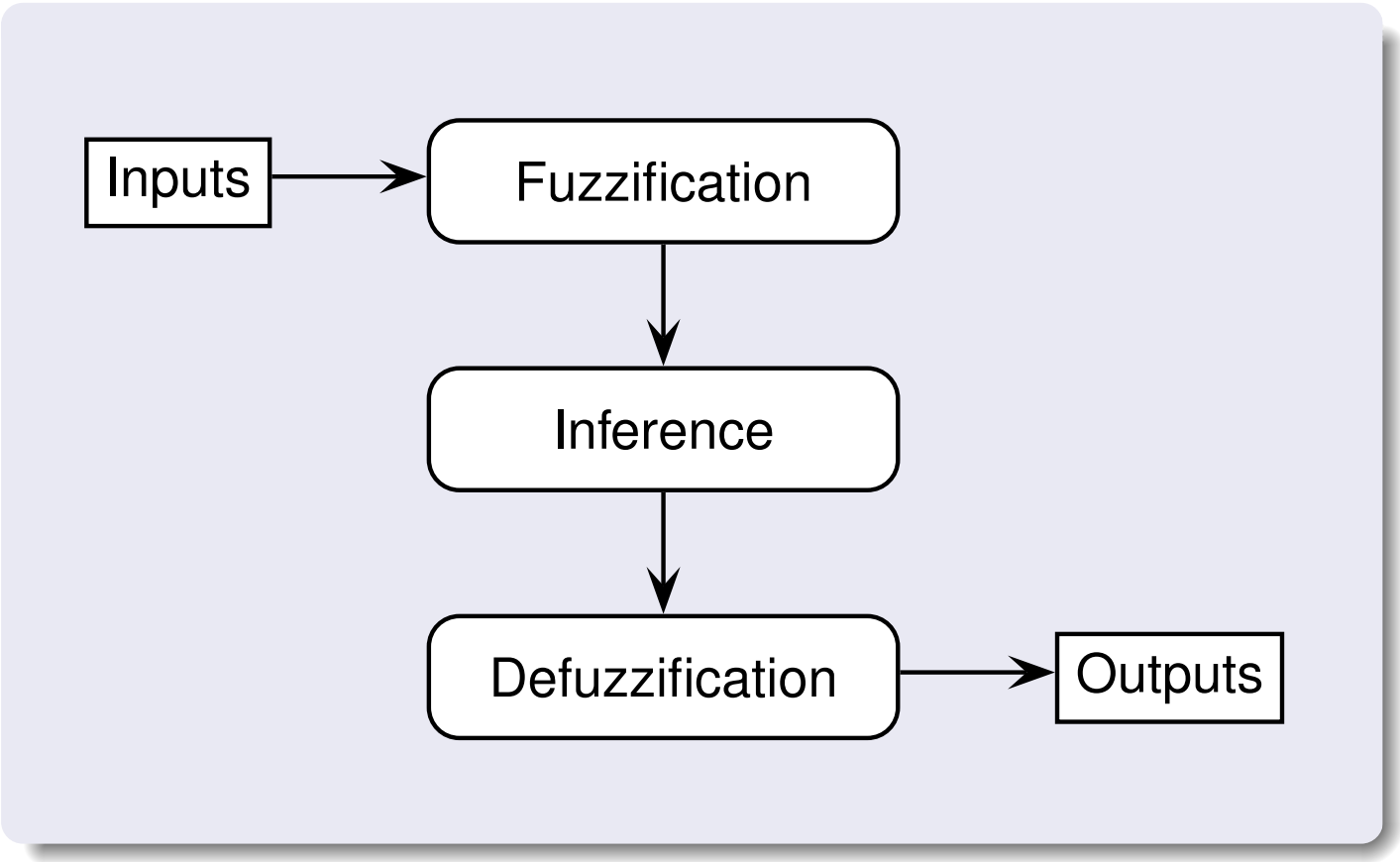
Centroid



Maxima



Stages of a Fuzzy Logic Controller



fuzzylite

Features

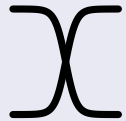
`fuzzylite` is a library (`.so`, `.dylib`, `.dll`)

`qtfuzzylite` links to `fuzzylite`

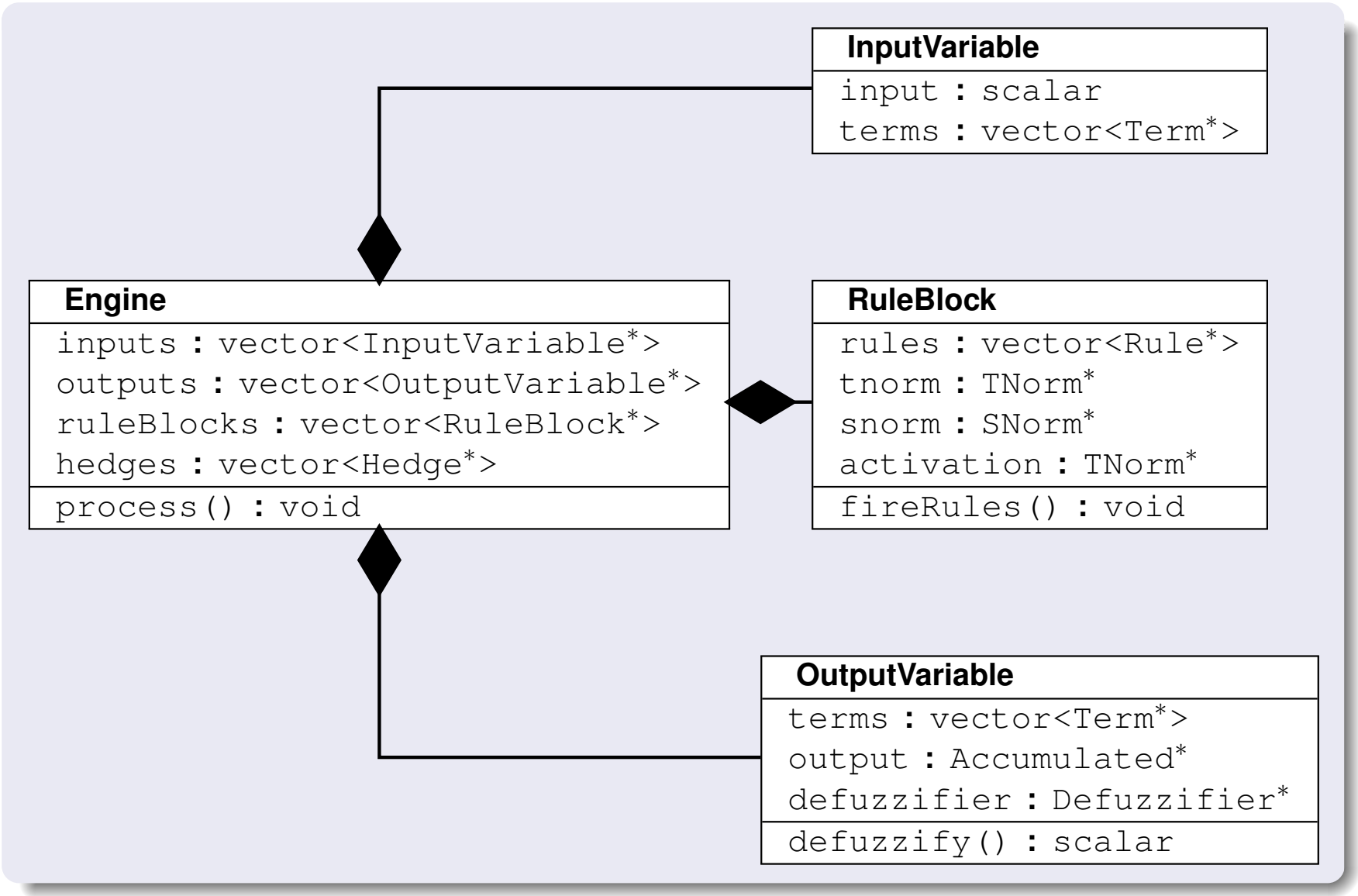
Main Features

- **Mamdani**, Takagi-Sugeno and Tsukamoto FLCs
- 17+ linguistic terms
- 13 fuzzy logic operators
- Seven defuzzifiers
- Six types of hedges (e.g. *very*, *somewhat*, *not*)
- Import and export using FCL, FIS, C++
- Extend and incorporate new components

Linguistic Terms

Basic		Extended		Edges
				
Triangle	Gaussian	Gaussian P.		Ramp
				
Trapezoid	Bell	Pi-Shape		Sigmoid
				
Rectangle	Sigmoid D.	Sigmoid P.		S-Shape
	$f(x) = c$ $f(x,y) = ax + by + c$			
Discrete	Constant, Linear, Custom			Z-Shape

Abstract Model



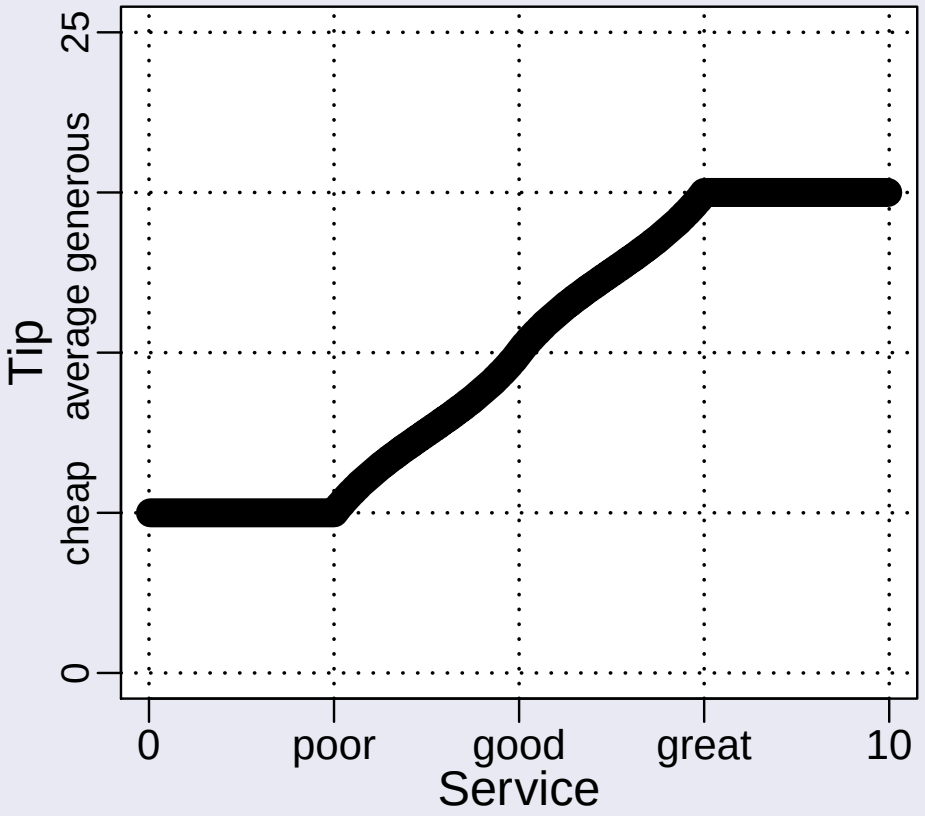
Conclusions and Future Work

Conclusions

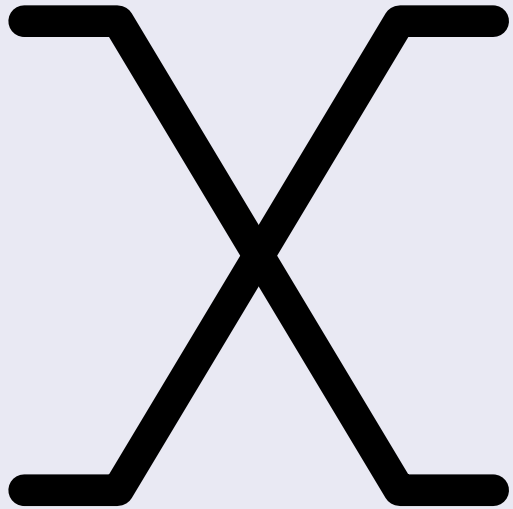
- FLCs are a **powerful** alternative to traditional control algorithms
  - **Easy** to design
  - **Easy** to operate
  - **Easy** to maintain over time
  - **Many** algorithms to tune FLCs

Conclusions

- Important to **recognize** when to utilize FLCs



vs.



Ramp term

Future Work

- **Type-2** Fuzzy Logic Controllers
- Adaptive **Neuro-Fuzzy** Inference System (ANFIS)
- Fuzzy C-Means **clustering** algorithm
- . . . and there are still ***many*** more things to do!

Donations

support **fuzzylite** with a donation

fuzzylite
A Fuzzy Logic Control Library in C++

Juan Rada-Vilela

www.fuzzylite.com