Lecture
Computational Intelligence
Winter Term 2007/2008

Stefan Droste

05.12.2007 (Wednesday)
## Plans for Today

1. **Introduction**
   - Plans for Today

2. **Fuzzy Subsets**
   - Definitions

3. **Fuzzy Relations**
   - Introduction

4. **Linguistic Variables**
   - Introduction
   - Term Sets and Hedges

5. **Fuzzy IF-THEN-Rules**
   - Introduction
   - Implementation of Fuzzy If-Then-Rules

6. **Fuzzy Controllers**
   - Introduction and Mamdani Fuzzy Controller
   - Takagi-Sugeno-Kang Fuzzy Controller
Subsets

We already defined

\[ A \subseteq B \iff \forall x: A(x) \leq B(x). \]
Subsets

We already defined

\[ A \subseteq B \iff \forall x : A(x) \leq B(x). \]

Note: This is crisp!
Subsets

We already defined
\[ A \subseteq B \iff \forall x : A(x) \leq B(x). \]

**Note:** This is crisp!

Is it plausible?
Subsets

We already defined
\[ A \subseteq B \iff \forall x : A(x) \leq B(x). \]

Note: This is crisp!

Is it plausible?

Example:
\[ U = \mathbb{N}_0 \]
\[ A(x) := \begin{cases} \frac{1}{x^2} & \text{if } x > 0 \\ a & \text{if } x = 0 \end{cases} \]
\[ B(x) := \begin{cases} \frac{1}{x} & \text{if } x > 0 \\ b & \text{if } x = 0 \end{cases} \]
\[ a > b \]
Subsets

We already defined
\[ A \subseteq B \iff \forall x: A(x) \leq B(x). \]

Note: This is crisp!

Is it plausible?

Example:
\[ \mathcal{U} = \mathbb{N}_0 \]
\[ A(x) := \begin{cases} \frac{1}{x^2} & \text{if } x > 0 \\ a & \text{if } x = 0 \end{cases} \quad B(x) := \begin{cases} \frac{1}{x} & \text{if } x > 0 \\ b & \text{if } x = 0 \end{cases} \quad a > b \]

Clearly, \( \forall x \in \mathcal{U} \setminus \{0\} : A(x) \leq B(x) \).
Subsets

We already defined
\[ A \subseteq B \iff \forall x: A(x) \leq B(x). \]

**Note:** This is crisp!

Is it plausible?

**Example:**
\[ \mathcal{U} = \mathbb{N}_0 \]
\[ A(x) := \begin{cases} \frac{1}{x^2} & \text{if } x > 0 \\ a & \text{if } x = 0 \end{cases} \quad B(x) := \begin{cases} \frac{1}{x} & \text{if } x > 0 \\ b & \text{if } x = 0 \end{cases} \]

Clearly, \( \forall x \in \mathcal{U} \setminus \{0\}: A(x) \leq B(x). \)
But because \( a > b \), \( \neg (A \subseteq B) \).
Fuzzy Subsets

A different perspective:
Crisp sets: $A \subseteq B \iff \forall x: x \in A \Rightarrow x \in B$
Fuzzy Subsets

A different perspective:

Crisp sets: \( A \subseteq B \iff \forall x: x \in A \Rightarrow x \in B \)

Definition:

For any lower semicontinuous \( t \)-norm \( t \):

\( A \subseteq_t B :\iff \forall x \in U : \phi_t(A(x), B(x)) \)
Fuzzy Subsets

A different perspective:
Crisp sets: \( A \subseteq B \iff \forall x: x \in A \Rightarrow x \in B \)

Definition:
For any lower semicontinuous \( t \)-norm \( t \):
\( A \subseteq_t B \iff \forall x \in U: \phi_t(A(x), B(x)) \)

\( A \equiv_t B \iff t(A \subseteq_t B, B \subseteq_t A) \)
Fuzzy Subsets

A different perspective:

Crisp sets: \( A \subseteq B \iff \forall x: x \in A \Rightarrow x \in B \)

Definition:
For any lower semicontinuous \( t \)-norm \( t \):
\( A \subseteq_t B :\iff \forall x \in \mathcal{U}: \phi_t(A(x), B(x)) \)

\( A \equiv_t B :\iff t(A \subseteq_t B, B \subseteq_t A) \)

What does \( \forall \) mean for fuzzy sets?

Definition:
\( \forall x: F(x) := \inf \{ F(u) \mid u \in \mathcal{U} \} \)
\( \exists x: F(x) := \sup \{ F(u) \mid u \in \mathcal{U} \} \)
Fuzzy Subsets

A different perspective:
Crisp sets: $A \subseteq B \iff \forall x: x \in A \Rightarrow x \in B$

Definition:
For any lower semicontinuous $t$-norm $t$:

$A \subseteq_t B \iff \forall x \in U: \phi_t(A(x), B(x))$

$= \inf \{\phi_t(A(x), B(x)) \mid x \in U\}$

$A \equiv_t B \iff t(A \subseteq_t B, B \subseteq_t A)$

What does $\forall$ mean for fuzzy sets?

Definition:
$\forall x: F(x) := \inf \{F(u) \mid u \in U\}$

$\exists x: F(x) := \sup \{F(u) \mid u \in U\}$
Subsets — Examples

\[ A(x) := \begin{cases} \frac{1}{x^2} & \text{if } x > 0 \\ a & \text{otherwise} \end{cases} \quad B(x) := \begin{cases} \frac{1}{x} & \text{if } x > 0 \\ b & \text{otherwise} \end{cases} \quad a > b \]
Subsets — Examples

\[ A(x) := \begin{cases} \frac{1}{x^2} & \text{if } x > 0 \\ a & \text{otherwise} \end{cases} \quad B(x) := \begin{cases} \frac{1}{x} & \text{if } x > 0 \\ b & \text{otherwise} \end{cases} \quad a > b \]

\[ A \subseteq_t B = \inf \{ \phi_t(A(x), B(x)) \mid x \in \mathcal{U} \} = \inf \{ \sup \{ w \mid t(A(x), w) \leq B(x) \} \mid x \in \mathcal{U} \} \]
Subsets — Examples

\[ A(x) := \begin{cases} \frac{1}{x^2} & \text{if } x > 0 \\ a & \text{otherwise} \end{cases}\]

\[ B(x) := \begin{cases} \frac{1}{x} & \text{if } x > 0 \\ b & \text{otherwise} \end{cases} \]

\[ A \subseteq_t B = \inf \{ \phi_t(A(x), B(x)) \mid x \in U \} = \inf \{ \sup \{ w \mid t(A(x), w) \leq B(x) \} \mid x \in U \} \]

\[ A \subseteq_{t_m} B = \inf \{ \sup \{ w \mid \min\{A(x), w\} \leq B(x) \} \mid x \in U \} = \]

\[ a > b \]
Subsets — Examples

\[ A(x) := \begin{cases} \frac{1}{x^2} & \text{if } x > 0 \\ a & \text{otherwise} \end{cases} \quad B(x) := \begin{cases} \frac{1}{x} & \text{if } x > 0 \\ b & \text{otherwise} \end{cases} \]

\( A \subseteq_t B = \inf \{ \phi_t(A(x), B(x)) \mid x \in \mathcal{U} \} = \inf \{ \sup \{ w \mid t(A(x), w) \leq B(x) \} \mid x \in \mathcal{U} \} \)

\( A \subseteq_{t_m} B = \inf \{ \sup \{ w \mid \min\{A(x), w\} \leq B(x) \} \mid x \in \mathcal{U} \} = b \)
Subsets — Examples

\[ A(x) := \begin{cases} \frac{1}{x^2} & \text{if } x > 0 \\ a & \text{otherwise} \end{cases} \quad B(x) := \begin{cases} \frac{1}{x} & \text{if } x > 0 \\ b & \text{otherwise} \end{cases} \]

\[ A \subseteq_t B = \inf \left\{ \phi_t(A(x), B(x)) \mid x \in \mathcal{U} \right\} = \inf \left\{ \sup \left\{ w \mid t(A(x), w) \leq B(x) \right\} \mid x \in \mathcal{U} \right\} \]

\[ A \subseteq_{tm} B = \inf \left\{ \sup \left\{ w \mid \min\{A(x), w\} \leq B(x) \right\} \mid x \in \mathcal{U} \right\} = b \]

\[ A \subseteq_{tl} B = \inf \left\{ \sup \left\{ w \mid \max\{0, A(x) + w - 1\} \leq B(x) \right\} \mid x \in \mathcal{U} \right\} \]
Subsets — Examples

\[ A(x) := \begin{cases} \frac{1}{x^2} & \text{if } x > 0 \\ a & \text{otherwise} \end{cases} \quad B(x) := \begin{cases} \frac{1}{x} & \text{if } x > 0 \\ b & \text{otherwise} \end{cases} \quad a > b \]

\[ A \subseteq_t B = \inf \{ \phi_t(A(x), B(x)) \mid x \in \mathcal{U} \} = \inf \{ \sup \{ w \mid t(A(x), w) \leq B(x) \} \mid x \in \mathcal{U} \} \]

\[ A \subseteq_{tm} B = \inf \{ \sup \{ w \mid \min\{A(x), w\} \leq B(x) \} \mid x \in \mathcal{U} \} = b \]

\[ A \subseteq_{tl} B = \inf \{ \sup \{ w \mid \max\{0, A(x) + w - 1\} \leq B(x) \} \mid x \in \mathcal{U} \} = b + (1 - a) \]
Subsets — Examples

\[ A(x) := \begin{cases} \frac{1}{x^2} & \text{if } x > 0 \\ a & \text{otherwise} \end{cases} \quad B(x) := \begin{cases} \frac{1}{x} & \text{if } x > 0 \\ b & \text{otherwise} \end{cases} \]

\[ A \subseteq_t B = \inf \{ \phi_t(A(x), B(x)) \mid x \in U \} = \inf \{ \sup \{ w \mid t(A(x), w) \leq B(x) \} \mid x \in U \} \]

\[ A \subseteq_{t_m} B = \inf \{ \sup \{ w \mid \min\{A(x), w\} \leq B(x) \} \mid x \in U \} = b \]

\[ A \subseteq_{t_l} B = \inf \{ \sup \{ w \mid \max\{0, A(x) + w - 1\} \leq B(x) \} \mid x \in U \} = b + (1 - a) \]

\[ A \subseteq_{t_p} B = \inf \{ \sup \{ w \mid A(x)w \leq B(x) \} \mid x \in U \} = \]
Subsets — Examples

\[ A(x) := \begin{cases} \frac{1}{x^2} & \text{if } x > 0 \\ a & \text{otherwise} \end{cases} \quad B(x) := \begin{cases} \frac{1}{x} & \text{if } x > 0 \\ b & \text{otherwise} \end{cases} \]

\[ A \subseteq_t B = \inf \{ \phi_t(A(x), B(x)) \mid x \in \mathcal{U} \} = \inf \{ \sup \{ w \mid t(A(x), w) \leq B(x) \} \mid x \in \mathcal{U} \} \]

\[ A \subseteq_{tm} B = \inf \{ \sup \{ w \mid \min\{A(x), w\} \leq B(x) \} \mid x \in \mathcal{U} \} = b \]

\[ A \subseteq_{tl} B = \inf \{ \sup \{ w \mid \max\{0, A(x) + w - 1\} \leq B(x) \} \mid x \in \mathcal{U} \} = b + (1 - a) \]

\[ A \subseteq_{tp} B = \inf \{ \sup \{ w \mid A(x)w \leq B(x) \} \mid x \in \mathcal{U} \} = \frac{b}{a} \]
Subsets — Examples

\[ A(x) := \begin{cases} \frac{1}{x^2} & \text{if } x > 0 \\ a & \text{otherwise} \end{cases} \quad B(x) := \begin{cases} \frac{1}{x} & \text{if } x > 0 \\ b & \text{otherwise} \end{cases} \quad a > b \]

\[ A \subseteq_t B = \inf \{ \phi_t(A(x), B(x)) \mid x \in \mathcal{U} \} = \inf \{ \sup \{ w \mid t(A(x), w) \leq B(x) \} \mid x \in \mathcal{U} \} \]

\[ A \subseteq_{tm} B = \inf \{ \sup \{ w \mid \min\{A(x), w\} \leq B(x) \} \mid x \in \mathcal{U} \} = b \]
\[ A \subseteq_{tl} B = \inf \{ \sup \{ w \mid \max\{0, A(x) + w - 1\} \leq B(x) \} \mid x \in \mathcal{U} \} = b + (1 - a) \]
\[ A \subseteq_{tp} B = \inf \{ \sup \{ w \mid A(x)w \leq B(x) \} \mid x \in \mathcal{U} \} = \frac{b}{a} \]

\[ A \subseteq_{td} B = \inf \{ \sup \{ w \mid [\max\{A(x), w\} = 1] \cdot \min\{A(x), w\} \leq B(x) \} \mid x \in \mathcal{U} \} \]
Subsets — Examples

\[ A(x) := \begin{cases} \frac{1}{x^2} & \text{if } x > 0 \\ a & \text{otherwise} \end{cases} \quad B(x) := \begin{cases} \frac{1}{x} & \text{if } x > 0 \\ b & \text{otherwise} \end{cases} \quad a > b \]

\[ A \subseteq_t B = \inf \{ \phi_t(A(x), B(x)) \mid x \in \mathcal{U} \} = \inf \{ \sup \{ w \mid t(A(x), w) \leq B(x) \} \mid x \in \mathcal{U} \} \]

\[ A \subseteq_{tm} B = \inf \{ \sup \{ w \mid \min\{A(x), w\} \leq B(x) \} \mid x \in \mathcal{U} \} = b \]

\[ A \subseteq_{tl} B = \inf \{ \sup \{ w \mid \max\{0, A(x) + w - 1\} \leq B(x) \} \mid x \in \mathcal{U} \} = b + (1 - a) \]

\[ A \subseteq_{tp} B = \inf \{ \sup \{ w \mid A(x)w \leq B(x) \} \mid x \in \mathcal{U} \} = \frac{b}{a} \]

\[ A \subseteq_{td} B \]

\[ = \inf \{ \sup \{ w \mid [\max\{A(x), w\} = 1] \cdot \min\{A(x), w\} \leq B(x) \} \mid x \in \mathcal{U} \} \]

\[ = \begin{cases} b & \text{if } a = 1 \\ 1 & \text{otherwise} \end{cases} \]
Relations

Crisp relation $R$ over $\mathcal{U}$ is a function $R : \mathcal{U} \times \mathcal{U} \rightarrow \{0, 1\}$. 
Relations

Crisp relation \( R \) over \( \mathcal{U} \) is a function \( R : \mathcal{U} \times \mathcal{U} \rightarrow \{0, 1\} \).

Fuzzy relation \( R \) over \( \mathcal{U} \) is a function \( R : \mathcal{U} \times \mathcal{U} \rightarrow [0, 1] \), i.e. fuzzy set over \( \mathcal{U} \times \mathcal{U} \).
Relations

Crisp relation $R$ over $\mathcal{U}$ is a function $R: \mathcal{U} \times \mathcal{U} \rightarrow \{0, 1\}$.

Fuzzy relation $R$ over $\mathcal{U}$ is a function $R: \mathcal{U} \times \mathcal{U} \rightarrow [0, 1]$, i.e. fuzzy set over $\mathcal{U} \times \mathcal{U}$.

**Definition:** Let $R$ and $S$ be fuzzy relations and $t$ a $t$-norm:
Relations

Crisp relation $R$ over $\mathcal{U}$ is a function $R : \mathcal{U} \times \mathcal{U} \rightarrow \{0, 1\}$.

Fuzzy relation $R$ over $\mathcal{U}$ is a function $R : \mathcal{U} \times \mathcal{U} \rightarrow [0, 1]$, i.e. fuzzy set over $\mathcal{U} \times \mathcal{U}$.

**Definition:** Let $R$ and $S$ be fuzzy relations and $t$ a $t$-norm:

domain $\text{dom} (R) := \{ x \mid \exists y : R(x, y) > 0 \}$
Relations

Crisp relation \( R \) over \( \mathcal{U} \) is a function \( R: \mathcal{U} \times \mathcal{U} \rightarrow \{0, 1\} \).

Fuzzy relation \( R \) over \( \mathcal{U} \) is a function \( R: \mathcal{U} \times \mathcal{U} \rightarrow [0, 1] \), i.e. fuzzy set over \( \mathcal{U} \times \mathcal{U} \).

**Definition:** Let \( R \) and \( S \) be fuzzy relations and \( t \) a \( t \)-norm:

- **domain** \( \text{dom}(R) := \{x \mid \exists y : R(x, y) > 0\} \)
- **range** \( \text{rg}(R) := \{y \mid \exists x : R(x, y) > 0\} \)
Relations

Crisp relation $R$ over $\mathcal{U}$ is a function $R: \mathcal{U} \times \mathcal{U} \rightarrow \{0, 1\}$.

Fuzzy relation $R$ over $\mathcal{U}$ is a function $R: \mathcal{U} \times \mathcal{U} \rightarrow [0, 1]$, i.e. fuzzy set over $\mathcal{U} \times \mathcal{U}$.

**Definition:** Let $R$ and $S$ be fuzzy relations and $t$ a $t$-norm:

- **domain** $\text{dom}(R) := \{x \mid \exists y: R(x, y) > 0\}$
- **range** $\text{rg}(R) := \{y \mid \exists x: R(x, y) > 0\}$
- **inverse** $R^{-1}: \mathcal{U} \times \mathcal{U} \rightarrow [0, 1]$ defined by $R^{-1}(y, x) := R(x, y)$
Relations

Crisp relation $R$ over $\mathcal{U}$ is a function $R: \mathcal{U} \times \mathcal{U} \rightarrow \{0, 1\}$.

Fuzzy relation $R$ over $\mathcal{U}$ is a function $R: \mathcal{U} \times \mathcal{U} \rightarrow [0, 1]$, i.e. fuzzy set over $\mathcal{U} \times \mathcal{U}$.

**Definition:** Let $R$ and $S$ be fuzzy relations and $t$ a $t$-norm:

- **domain** $\text{dom}(R) := \{x \mid \exists y: R(x, y) > 0\}$
- **range** $\text{rg}(R) := \{y \mid \exists x: R(x, y) > 0\}$
- **inverse** $R^{-1}: \mathcal{U} \times \mathcal{U} \rightarrow [0, 1]$ defined by $R^{-1}(y, x) := R(x, y)$
- **product** $R \circ_t S := \{(x, y) \mid \exists z: t(R(x, z), S(z, y))\}$
Crisp relation $R$ over $\mathcal{U}$ is a function $R : \mathcal{U} \times \mathcal{U} \rightarrow \{0, 1\}$.

Fuzzy relation $R$ over $\mathcal{U}$ is a function $R : \mathcal{U} \times \mathcal{U} \rightarrow [0, 1]$, i.e. fuzzy set over $\mathcal{U} \times \mathcal{U}$.

**Definition:** Let $R$ and $S$ be fuzzy relations and $t$ a $t$-norm:

- **domain** $\text{dom} (R) := \{x \mid \exists y : R(x, y) > 0\}$
- **range** $\text{rg} (R) := \{y \mid \exists x : R(x, y) > 0\}$
- **inverse** $R^{-1} : \mathcal{U} \times \mathcal{U} \rightarrow [0, 1]$ defined by $R^{-1}(y, x) := R(x, y)$
- **product** $R \circ_t S := \{(x, y) \mid \exists z : t(R(x, z), S(z, y))\}$

**Note:** All these sets are fuzzy!
Properties of Fuzzy Relations

**Definition:** Let $t$ be some lower semicontinuous $t$-norm, $n$ some negation. Fuzzy relation $R$ is called (with respect to $t$ and $n$)
Properties of Fuzzy Relations

**Definition:** Let $t$ be some lower semicontinuous $t$-norm, $n$ some negation.
Fuzzy relation $R$ is called (with respect to $t$ and $n$)

1. reflexive iff $\forall x : R(x, x) = 1$
Properties of Fuzzy Relations

**Definition:** Let $t$ be some lower semicontinuous $t$-norm, $n$ some negation. Fuzzy relation $R$ is called (with respect to $t$ and $n$)

1. reflexive iff $[\forall x : R(x, x)] = 1$
2. n-irreflexive iff $[\forall x : n(R(x, x))] = 1$
Properties of Fuzzy Relations

**Definition:** Let $t$ be some lower semicontinuous $t$-norm, $n$ some negation.
Fuzzy relation $R$ is called (with respect to $t$ and $n$)

1. **reflexive** iff $\forall x : R(x, x) = 1$
2. **n-irreflexive** iff $\forall x : n(R(x, x)) = 1$
3. **t-transitive** iff $\forall x, y, z : \phi_t(t(R(x, y), R(y, z)), R(x, z)) = 1$
Properties of Fuzzy Relations

**Definition:** Let $t$ be some lower semicontinuous $t$-norm, $n$ some negation.
Fuzzy relation $R$ is called (with respect to $t$ and $n$)

1. **reflexive** iff $[\forall x : R(x, x)] = 1$
2. **n-irreflexive** iff $[\forall x : n(R(x, x))] = 1$
3. **t-transitive** iff $[\forall x, y, z : \phi_t(t(R(x, y), R(y, z)), R(x, z))] = 1$
4. **symmetric** iff $[\forall x, y : \phi_t(R(x, y), R(y, x))] = 1$
Properties of Fuzzy Relations

**Definition:** Let $t$ be some lower semicontinuous $t$-norm, $n$ some negation.

Fuzzy relation $R$ is called (with respect to $t$ and $n$)

1. reflexive iff $\forall x: R(x, x) = 1$
2. $n$-irreflexive iff $\forall x: n(R(x, x)) = 1$
3. $t$-transitive iff $\forall x, y, z: \phi_t(t(R(x, y), R(y, z)), R(x, z)) = 1$
4. symmetric iff $\forall x, y: \phi_t(R(x, y), R(y, x)) = 1$
5. $t$-antisymmetric iff $\forall x, y: \phi_t(t(R(x, y), R(y, x)), [x = y]) = 1$
**Properties of Fuzzy Relations**

**Definition:** Let $t$ be some lower semicontinuous $t$-norm, $n$ some negation.

Fuzzy relation $R$ is called (with respect to $t$ and $n$)

1. **reflexive** iff $\forall x : R(x, x) = 1$
2. **n-irreflexive** iff $\forall x : n(R(x, x)) = 1$
3. **$t$-transitive** iff $\forall x, y, z : \phi_t(t(R(x, y), R(y, z)), R(x, z)) = 1$
4. **symmetric** iff $\forall x, y : \phi_t(R(x, y), R(y, x)) = 1$
5. **$t$-antisymmetric** iff $\forall x, y : \phi_t(t(R(x, y), R(y, x)), [x = y]) = 1$
6. **$t$-asymmetric** iff $\forall x, y : n(t(R(x, y), R(y, x))) = 1$
Properties of Fuzzy Relations

**Definition:** Let $t$ be some lower semicontinuous $t$-norm, $n$ some negation.

Fuzzy relation $R$ is called (with respect to $t$ and $n$)

1. **reflexive** iff $[\forall x : R(x, x)] = 1$
2. **$n$-irreflexive** iff $[\forall x : n(R(x, x))] = 1$
3. **$t$-transitive** iff $[\forall x, y, z : \phi_t(t(R(x, y), R(y, z)), R(x, z))] = 1$
4. **symmetric** iff $[\forall x, y : \phi_t(R(x, y), R(y, x))] = 1$
5. **$t$-antisymmetric** iff
   $[\forall x, y : \phi_t(t(R(x, y), R(y, x)), [x = y])] = 1$
6. **$t$-asymmetric** iff $[\forall x, y : n(t(R(x, y), R(y, x)))] = 1$

**Note:** All these properties are crisp!
Properties of Fuzzy Relations

**Definition:** Let $t$ be some lower semicontinuous $t$-norm, $n$ some negation.

Fuzzy relation $R$ is called (with respect to $t$ and $n$)

1. **reflexive** iff $[\forall x: R(x, x)] = 1$
2. **$n$-irreflexive** iff $[\forall x: n(R(x, x))] = 1$
3. **$t$-transitive** iff $[\forall x, y, z: \phi_t(t(R(x, y), R(y, z)), R(x, z))] = 1$
4. **symmetric** iff $[\forall x, y: \phi_t(R(x, y), R(y, x))] = 1$
5. **$t$-antisymmetric** iff $[\forall x, y: \phi_t(t(R(x, y), R(y, x)), [x = y])] = 1$
6. **$t$-asymmetric** iff $[\forall x, y: n(t(R(x, y), R(y, x)))] = 1$

**Note:** All these properties are crisp!

**Note:**

Removing ‘$[\cdot] = 1$’ introduces fuzzy versions of the properties.
On Properties of Fuzzy Relations

Why does it say “symmetric” and not “t-symmetric?”
On Properties of Fuzzy Relations

Why does it say “symmetric” and not “$t$-symmetric?”

$[\forall x, y : \phi_t(R(x, y), R(y, x))] = 1$
On Properties of Fuzzy Relations

Why does it say “symmetric” and not “\( t \)-symmetric?”

\[
[\forall x, y : \phi_t(R(x, y), R(y, x))] = 1 \\
\iff \inf\{\phi_t(R(x, y), R(y, x)) \mid x, y \in U\} = 1.
\]
On Properties of Fuzzy Relations

Why does it say “symmetric” and not “t-symmetric?”

\[ \forall x, y : \phi_t(R(x, y), R(y, x)) = 1 \]
\[ \iff \inf\{\phi_t(R(x, y), R(y, x)) \mid x, y \in U \} = 1. \]

\( (\phi_t(R(x, y), R(y, x)) = 1) \iff (R(x, y) \leq R(y, x)) \)
for any t-norm t


On Properties of Fuzzy Relations

Why does it say “symmetric” and not “$t$-symmetric”?

$$[\forall x, y : \phi_t(R(x, y), R(y, x))] = 1 \iff \inf\{\phi_t(R(x, y), R(y, x)) \mid x, y \in U\} = 1.$$  

$$(\phi_t(R(x, y), R(y, x)) = 1) \iff (R(x, y) \leq R(y, x))$$

for any $t$-norm $t$

Thus, “symmetric” is independent of the choice of the $t$-norm.
On Properties of Fuzzy Relations

Why does it say “symmetric” and not “$t$-symmetric?”

\[
[\forall x, y : \phi_t(R(x, y), R(y, x))] = 1 \\
\iff \inf \{\phi_t(R(x, y), R(y, x)) \mid x, y \in U\} = 1.
\]

\[
(\phi_t(R(x, y), R(y, x)) = 1) \iff (R(x, y) \leq R(y, x))
\]

for any $t$-norm $t$

Thus, “symmetric” is independent of the choice of the $t$-norm.

Note: This is not true for the fuzzy version of “symmetric.”
Properties of $t$-norms Reconsidered

$$\Delta_U := \{(x, x) \in U \times U\}$$
Properties of $t$-norms Reconsidered

$\Delta_u := \{(x, x) \in U \times U\}$

**Theorem:** ($R$ fuzzy relation, $t$ lower semicontinuous $t$-norm, $n$ negation)

1. $R$ reflexive iff $\Delta_u \subseteq R$
Properties of $t$-norms Reconsidered

$$\Delta_u := \{(x, x) \in U \times U\}$$

**Theorem:** ($R$ fuzzy relation, $t$ lower semicontinuous $t$-norm, $n$ negation)

1. $R$ reflexive iff $\Delta_u \subseteq R$
2. $R$ $n$-irreflexive iff $(R \cap_t \Delta_u)^c = \emptyset$
Properties of $t$-norms Reconsidered

$$\Delta_U := \{(x, x) \in U \times U\}$$

**Theorem:** (\(R\) fuzzy relation, \(t\) lower semicontinuous \(t\)-norm, \(n\) negation)

1. \(R\) reflexive iff \(\Delta_U \subseteq R\)
2. \(R\) \(n\)-irreflexive iff \((R \cap t \Delta_U)^C_n \equiv \emptyset\)
3. \(R\) \(t\)-transitive iff \(R \circ_t R \subseteq R\)

Properties of $t$-norms Reconsidered

$$\Delta_u := \{(x, x) \in U \times U\}$$

**Theorem:** ($R$ fuzzy relation, $t$ lower semicontinuous $t$-norm, $n$ negation)

1. $R$ reflexive iff $\Delta_u \subseteq R$
2. $R$ $n$-irreflexive iff $(R \cap_t \Delta_u)^c = \emptyset$
3. $R$ $t$-transitive iff $R \circ_t R \subseteq R$
4. $R$ symmetric iff $R \subseteq R^{-1}$
Properties of \( t \)-norms Reconsidered

\( \Delta_U := \{(x, x) \in U \times U\} \)

**Theorem:** (\( R \) fuzzy relation, \( t \) lower semicontinuous \( t \)-norm, \( n \) negation)

1. \( R \) reflexive iff \( \Delta_U \subseteq R \)
2. \( R \) n-irreflexive iff \( (R \cap_t \Delta_U)^C_n \equiv \emptyset \)
3. \( R \) \( t \)-transitive iff \( R \circ_t R \subseteq R \)
4. \( R \) symmetric iff \( R \subseteq R^{-1} \)
5. \( R \) \( t \)-antisymmetric iff \( R \cap_t R^{-1} \subseteq \Delta_U \)
Properties of $t$-norms Reconsidered

$\Delta_U := \{(x, x) \in U \times U\}$

**Theorem:** ($R$ fuzzy relation, $t$ lower semicontinuous $t$-norm, $n$ negation)

1. $R$ reflexive iff $\Delta_U \subseteq R$
2. $R$ $n$-irreflexive iff $(R \cap_t \Delta_U)^c_n \equiv \emptyset$
3. $R$ $t$-transitive iff $R \circ_t R \subseteq R$
4. $R$ symmetric iff $R \subseteq R^{-1}$
5. $R$ $t$-antisymmetric iff $R \cap_t R^{-1} \subseteq \Delta_U$
6. $R$ $t$-asymmetric iff $(R \cap_t R^{-1})^c_n \equiv 1$
Motivation

We have knowledge about

- fuzzy logic,
- fuzzy sets,
- fuzzy numbers, and
- fuzzy relations.
Motivation

We have knowledge about

- fuzzy logic,
- fuzzy sets,
- fuzzy numbers, and
- fuzzy relations.

We want to model and control technical systems.

We begin with a methodology for modeling.
Linguistic Variables

Definition: A linguistic variable $V$ is characterized by

- its name $x$, 

**Linguistic Variables**

**Definition:** A *linguistic variable* $V$ is characterized by

- its name $x$,
- a universe $U$,
Linguistic Variables

**Definition:** A **linguistic variable** $V$ is characterized by

- its name $x$,
- a universe $\mathcal{U}$,
- a term set $T(x)$,
Linguistic Variables

**Definition:** A **linguistic variable** $V$ is characterized by

- its name $x$,
- a universe $\mathcal{U}$,
- a term set $T(x)$,
- a syntactic rule $G$ for generating names of values of $x$, and
Linguistic Variables

**Definition:** A linguistic variable $V$ is characterized by

- its name $x$,
- a universe $\mathcal{U}$,
- a term set $T(x)$,
- a syntactic rule $G$ for generating names of values of $x$, and
- a semantic rule $M$ for associating meanings with values.
Linguistic Variables

**Definition:** A linguistic variable $V$ is characterized by

- its name $x$,
- a universe $U$,
- a term set $T(x)$,
- a syntactic rule $G$ for generating names of values of $x$, and
- a semantic rule $M$ for associating meanings with values.

**Example:** Describe the speed of a car.
Linguistic Variables

**Definition:** A **linguistic variable** $V$ is characterized by

- its name $x$,
- a universe $\mathcal{U}$,
- a term set $T(x)$,
- a syntactic rule $G$ for generating names of values of $x$, and
- a semantic rule $M$ for associating meanings with values.

**Example:** Describe the **speed** of a car.

- name $x = \text{speed}$
Linguistic Variables

**Definition:** A **linguistic variable** $V$ is characterized by

- its name $x$,  
- a universe $\mathcal{U}$,  
- a term set $T(x)$,  
- a syntactic rule $G$ for generating names of values of $x$, and  
- a semantic rule $M$ for associating meanings with values.

**Example:** Describe the **speed** of a car.

- name $x = \text{speed}$  
- universe $\mathcal{U} = [0; 250]$ (possible crisp values)
Linguistic Variables

**Definition:** A linguistic variable $V$ is characterized by

- its name $x$,
- a universe $U$,
- a term set $T(x)$,
- a syntactic rule $G$ for generating names of values of $x$, and
- a semantic rule $M$ for associating meanings with values.

**Example:** Describe the speed of a car.

- name $x = \text{speed}$
- universe $U = [0; 250]$ (possible crisp values)
- term set $T(x) = \{\text{slow, moderate, fast, too fast}\}$
**Linguistic Variables**

**Definition:** A linguistic variable $V$ is characterized by

- its name $x$,
- a universe $\mathcal{U}$,
- a term set $T(x)$,
- a syntactic rule $G$ for generating names of values of $x$, and
- a semantic rule $M$ for associating meanings with values.

**Example:** Describe the speed of a car.

- name $x = \text{speed}$
- universe $\mathcal{U} = [0; 250]$ (possible crisp values)
- term set $T(x) = \{\text{slow, moderate, fast, too fast}\}$
- syntactic rule $G$: a fuzzy set for each term from $T(x)$
Linguistic Variables

**Definition:** A linguistic variable $V$ is characterized by

- its name $x$,
- a universe $\mathcal{U}$,
- a term set $T(x)$,
- a syntactic rule $G$ for generating names of values of $x$, and
- a semantic rule $M$ for associating meanings with values.

**Example:** Describe the speed of a car.

- name $x = \text{speed}$
- universe $\mathcal{U} = [0; 250]$ (possible crisp values)
- term set $T(x) = \{\text{slow, moderate, fast, too fast}\}$
- syntactic rule $G$: a fuzzy set for each term from $T(x)$
- semantic rule $M$: an interpretation for each term from $T(x)$
Linguistic Variable Speed
Linguistic Variable Speed

slow
Linguistic Variable Speed

- slow
- moderate
Linguistic Variable Speed

- slow
- moderate
- fast
Linguistic Variable Speed

slow
moderate
fast
too fast

0 25 50 75 100 125 150 175 200 225 250
Standard Term Sets

Linguistic variables are a very general tool.
Standard Term Sets

Linguistic variables are a very general tool.

It helps to have standard solutions for standard situations.
Standard Term Sets

Linguistic variables are a very general tool.

It helps to have standard solutions for standard situations.

\[ T(x) = \{ \text{negative big (NB), negative medium (NM),} \]
\[ \text{negative small (NS), zero (ZE),} \]
\[ \text{positive small (PS), positive medium (PM), positive big (PB)} \} \]
Standard Term Sets

Linguistic variables are a very general tool.

It helps to have standard solutions for standard situations.

\[ T(x) = \{ \text{negative big (NB), negative medium (NM), negative small (NS), zero (ZE), positive small (PS), positive medium (PM), positive big (PB)} \} \]
Standard Term Sets

Linguistic variables are a very general tool.

It helps to have standard solutions for standard situations.

\[ T(x) = \{ \text{negative big (NB), negative medium (NM), negative small (NS), zero (ZE), positive small (PS), positive medium (PM), positive big (PB)} \} \]
Linguistic variables are a very general tool.

It helps to have standard solutions for standard situations.

\[ T(x) = \{ \text{negative big (NB), negative medium (NM),} \]
\[ \text{negative small (NS), zero (ZE),} \]
\[ \text{positive small (PS), positive medium (PM), positive big (PB)} \} \]
Standard Term Sets

Linguistic variables are a very general tool.

It helps to have standard solutions for standard situations.

\[ T(x) = \{ \text{negative big (NB), negative medium (NM),} \]
\[ \text{negative small (NS), zero (ZE),} \]
\[ \text{positive small (PS), positive medium (PM), positive big (PB)} \} \]
Standard Term Sets

Linguistic variables are a very general tool.

It helps to have standard solutions for standard situations.

\[ T(x) = \{ \text{negative big (NB), negative medium (NM),} \]
\[ \text{negative small (NS), zero (ZE),} \]
\[ \text{positive small (PS), positive medium (PM), positive big (PB)} \} \]
Standard Term Sets

Linguistic variables are a very general tool.

It helps to have standard solutions for standard situations.

\[ T(x) = \{ \text{negative big (NB), negative medium (NM),} \]
\[ \text{negative small (NS), zero (ZE),} \]
\[ \text{positive small (PS), positive medium (PM), positive big (PB)} \} \]
Standard Term Sets

Linguistic variables are a very general tool.

It helps to have standard solutions for standard situations.

\[ T(x) = \{ \text{negative big (NB), negative medium (NM),} \]
\[ \text{negative small (NS), zero (ZE),} \]
\[ \text{positive small (PS), positive medium (PM), positive big (PB)} \} \]
Linguistic Hedges

In natural language, properties can be modified using words like slightly, fairly, very . . .
Linguistic Hedges

In natural language, properties can be modified using words like slightly, fairly, very . . .

We can adopt this in fuzzy modeling using modifiers, also called linguistic hedges.

A linguistic hedge MOD is a function mapping fuzzy sets to fuzzy sets, i.e. \( \text{MOD} : \mathcal{F} \rightarrow \mathcal{F} \) (\( \mathcal{F} \) set of all fuzzs sets over universe \( \mathcal{U} \)).
Linguistic Hedges

In natural language, properties can be modified using words like slightly, fairly, very . . .

We can adopt this in fuzzy modeling using modifiers, also called linguistic hedges.

A linguistic hedge MOD is a function mapping fuzzy sets to fuzzy sets, i.e. MOD : \( \mathcal{F} \rightarrow \mathcal{F} \) (\( \mathcal{F} \) set of all fuzzs sets over universe \( \mathcal{U} \)).

Possible properties of linguistic hedges MOD:
Linguistic Hedges

In natural language, properties can be modified using words like slightly, fairly, very . . .

We can adopt this in fuzzy modeling using modifiers, also called linguistic hedges.

A linguistic hedge MOD is a function mapping fuzzy sets to fuzzy sets, i.e. \( \text{MOD} : \mathcal{F} \rightarrow \mathcal{F} \) (\( \mathcal{F} \) set of all fuzzs sets over universe \( \mathcal{U} \)).

Possible properties of linguistic hedges MOD:

- expanding: \( \forall F : F \subseteq \text{MOD}(F) \)
Linguistic Hedges

In natural language, properties can be modified using words like slightly, fairly, very . . .

We can adopt this in fuzzy modeling using modifiers, also called linguistic hedges.

A linguistic hedge MOD is a function mapping fuzzy sets to fuzzy sets, i.e. MOD : \(\mathcal{F} \rightarrow \mathcal{F}\) (\(\mathcal{F}\) set of all fuzz sets over universe \(\mathcal{U}\)).

Possible properties of linguistic hedges MOD:
- expanding: \(\forall F : F \subseteq \text{MOD}(F)\)
- compressing: \(\forall F : \text{MOD}(F) \subseteq F\)
Linguistic Hedges

In natural language, properties can be modified using words like slightly, fairly, very . . .

We can adopt this in fuzzy modeling using modifiers, also called linguistic hedges.

A linguistic hedge MOD is a function mapping fuzzy sets to fuzzy sets, i.e. MOD : \( \mathcal{F} \rightarrow \mathcal{F} \) (\( \mathcal{F} \) set of all fuzzs sets over universe \( \mathcal{U} \)).

Possible properties of linguistic hedges MOD:

- expanding: \( \forall F : F \subseteq \text{MOD}(F) \)
- compressing: \( \forall F : \text{MOD}(F) \subseteq F \)
- closed: \( \forall F : \text{MOD}(\text{MOD}(F)) = \text{MOD}(F) \)
Linguistic Hedges

In natural language, properties can be modified using words like slightly, fairly, very . . .

We can adopt this in fuzzy modeling using modifiers, also called linguistic hedges.

A linguistic hedge \( \text{MOD} \) is a function mapping fuzzy sets to fuzzy sets, i.e. \( \text{MOD} : \mathcal{F} \rightarrow \mathcal{F} \) (\( \mathcal{F} \) set of all fuzzs sets over universe \( \mathcal{U} \)).

Possible properties of linguistic hedges \( \text{MOD} \):

- expanding: \( \forall F : F \subseteq \text{MOD}(F) \)
- compressing: \( \forall F : \text{MOD}(F) \subseteq F \)
- closed: \( \forall F : \text{MOD}(\text{MOD}(F)) = \text{MOD}(F) \)

Often: \( \text{MOD}(F)(x) := (\text{MOD}(F(x)), \text{where MOD}: [0, 1] \rightarrow [0, 1] \).
Linguistic Hedge “VERY”

Typical definition: \( \text{VERY}(u) = u^2 \)
Linguistic Hedge “VERY”

Typical definition: \( \text{VERY}(u) = u^2 \)
Linguistic Hedge  “VERY”

Typical definition: VERY($u$) = $u^2$
Linguistic Hedge “VERY”

Typical definition: $\text{VERY}(u) = u^2$
Linguistic Hedge “MORE OR LESS”

Typical definition: \( \text{MORE OR LESS}(u) = \sqrt{u} \)
Linguistic Hedge "MORE OR LESS"

Typical definition: $\text{MORE OR LESS}(u) = \sqrt{u}$
Linguistic Hedge “MORE OR LESS”

Typical definition: \( \text{MORE OR LESS}(u) = \sqrt{u} \)
Linguistic Hedge “MORE OR LESS”

Typical definition: \[ \text{MORE OR LESS}(u) = \sqrt{u} \]
Fuzzy IF-THEN-Rules

Assume technical system described by means of linguistic variables.

How can we define control of this system?
Assume technical system described by means of linguistic variables.

How can we define control of this system?

**Example:** crash prevention
Fuzzy IF-THEN-Rules

Assume technical system described by means of linguistic variables.

How can we define control of this system?

**Example:** crash prevention

**input:** current *speed*, current *distance* to obstacle
Fuzzy IF-THEN-Rules

Assume technical system described by means of linguistic variables.

How can we define control of this system?

**Example:** crash prevention

input: current speed, current distance to obstacle

output: force used for braking
Fuzzy IF-THEN-Rules

Assume technical system described by means of linguistic variables.

How can we define control of this system?

Example: crash prevention

input: current speed, current distance to obstacle
output: force used for braking

Describe control as fuzzy IF-THEN-rule:
IF speed=high and distance=small THEN brake sharply.
Assume technical system described by means of linguistic variables. How can we define control of this system?

**Example:** crash prevention

- **input:** current speed, current distance to obstacle
- **output:** force used for braking

Describe control as fuzzy IF-THEN-rule: IF speed=high and distance=small THEN brake sharply.

**Clearly:** We can implement “and” using some $t$-norm.
A Fuzzy IF-THEN-Rule

IF $X = A$ THEN $Y = B$

$X, Y$: linguistic variables

$A, B$: linguistic terms
A Fuzzy IF-THEN-Rule

IF $X = A$ THEN $Y = B$

$X, Y$: linguistic variables
$A, B$: linguistic terms

Fuzzy IF-THEN-Rule constitutes a relation.
A Fuzzy IF-THEN-Rule

IF $X = A$ THEN $Y = B$

$X$, $Y$: linguistic variables
$A$, $B$: linguistic terms

Fuzzy IF-THEN-Rule constitutes a relation.

Possible interpretation:
Specific input defines the activation degree of the rule.
A Fuzzy IF-THEN-Rule

IF $X = A$ THEN $Y = B$

$X, Y$: linguistic variables
$A, B$: linguistic terms

Fuzzy IF-THEN-Rule constitutes a relation.

Possible interpretation:
Specific input defines the activation degree of the rule.

In general, input may be any fuzzy set $M$.

$$\text{act}(M, A) = \sup_{u \in \mathcal{U}} \{ t(M(u), A(u)) \}$$
A Fuzzy IF-THEN-Rule

IF \( X = A \) THEN \( Y = B \)

\( X, Y \): linguistic variables
\( A, B \): linguistic terms

Fuzzy IF-THEN-Rule constitutes a relation.

Possible interpretation:
Specific input defines the activation degree of the rule.

In general, input may be any fuzzy set \( M \).

\[
\text{act}(M, A) = \sup_{u \in \mathcal{U}} \{ t(M(u), A(u)) \}
\]

If \( M \) is singleton (only 1 for exactly one \( x \)), \( (\text{act})(M, A) = A(x) \).
A Fuzzy IF-THEN-Rule

IF $X = A$ THEN $Y = B$

$X, Y$: linguistic variables (over universes $U, U'$)

$A, B$: linguistic terms

We have fuzzy sets for ‘$A$’ and ‘$B$’ (over $U$ resp. $U'$).
A Fuzzy IF-THEN-Rule

If $X = A$ then $Y = B$

$X, Y$: linguistic variables (over universes $\mathcal{U}, \mathcal{U}'$)

$A, B$: linguistic terms

We have fuzzy sets for ‘$A$’ and ‘$B$’ (over $\mathcal{U}$ resp. $\mathcal{U}'$).

We need a fuzzy set $R$ over $\mathcal{U} \times \mathcal{U}'$ for the relation

If $X = A$ then $Y = B$. 
A Fuzzy IF-THEN-Rule

IF \( X = A \) THEN \( Y = B \)

\( X, Y \): linguistic variables (over universes \( U, U' \))
\( A, B \): linguistic terms

We have fuzzy sets for ‘\( A \)’ and ‘\( B \)’ (over \( U \) resp. \( U' \)).

We need a fuzzy set \( R \) over \( U \times U' \) for the relation
\( \text{IF } X = A \text{ THEN } Y = B \).

Possible definitions:
A Fuzzy IF-THEN-Rule

IF \( X = A \) THEN \( Y = B \)

\( X, Y \): linguistic variables (over universes \( U, U' \))
\( A, B \): linguistic terms

We have fuzzy sets for ‘A’ and ‘B’ (over \( U \) resp. \( U' \)).

We need a fuzzy set \( R \) over \( U \times U' \) for the relation
IF \( X = A \) THEN \( Y = B \).

Possible definitions:
- Read IF-THEN as implication → \( R(x, x') := \phi_t(A(x), B(x')) \)
A Fuzzy IF-THEN-Rule

IF $X = A$ THEN $Y = B$

$X, Y$: linguistic variables (over universes $U, U'$)

$A, B$: linguistic terms

We have fuzzy sets for ‘$A$’ and ‘$B$’(over $U$ resp. $U'$).

We need a fuzzy set $R$ over $U \times U'$ for the relation

IF $X = A$ THEN $Y = B$.

Possible definitions:

- Read IF-THEN as implication $\rightarrow R(x, x') := \phi_t(A(x), B(x'))$
- Read IF-THEN as $t$-norm $\rightarrow R(x, x') := t(A(x), B(x'))$
A Fuzzy IF-THEN-Rule

IF $X = A$ THEN $Y = B$

$X, Y$: linguistic variables (over universes $\mathcal{U}, \mathcal{U}'$)
$A, B$: linguistic terms

We have fuzzy sets for ‘$A$’ and ‘$B$’(over $\mathcal{U}$ resp. $\mathcal{U}'$).

We need a fuzzy set $R$ over $\mathcal{U} \times \mathcal{U}'$ for the relation
IF $X = A$ THEN $Y = B$.

Possible definitions:

- Read IF-THEN as implication $\rightarrow R(x, x') := \phi_t(A(x), B(x'))$
- Read IF-THEN as $t$-norm $\rightarrow R(x, x') := t(A(x), B(x'))$

Justification:

Read ‘IF $x = a$ THEN $y = b$’ as
It is true that $x = a$ holds and $y = b$ holds.
A Fuzzy IF-THEN-Rule

IF $X = A$ THEN $Y = B$

$X, Y$: linguistic variables (over universes $U, U'$)

$A, B$: linguistic terms

We have fuzzy sets for ‘$A$’ and ‘$B$’ (over $U$ resp. $U'$).

We need a fuzzy set $R$ over $U \times U'$ for the relation

IF $X = A$ THEN $Y = B$.

Possible definitions:

- Read IF-THEN as implication $\rightarrow$ $R(x, x') := \phi_t(A(x), B(x'))$
- Read IF-THEN as $t$-norm $\rightarrow$ $R(x, x') := t(A(x), B(x'))$

Justification:

Read ‘IF $x = a$ THEN $y = b$’ as

It is true that $x = a$ holds and $y = b$ holds.

Result: a fuzzy set $R(x') = R(\text{act}(M, A), x')$ for IF-THEN-rule (depending on the input $M$) that serves as a result.
Reconsidering the Approach

Obvious Problems:
Reconsidering the Approach

Obvious Problems:

1. The input parameters will be crisp.
Reconsidering the Approach

Obvious Problems:

1. The input parameters will be crisp.
2. One fuzzy IF-THEN-rule will not be sufficient.
   ~ How do we integrate many IF-THEN-rules?
Reconsidering the Approach

Obvious Problems:

1. The input parameters will be crisp.
2. One fuzzy IF-THEN-rule will not be sufficient. 
   \[ \Rightarrow \] How do we integrate many IF-THEN-rules?
3. Finally, a crisp value for the braking force is needed.
Reconsidering the Approach

Obvious Problems:

1. The input parameters will be crisp.
2. One fuzzy IF-THEN-rule will not be sufficient.
   \[ \sim \] How do we integrate many IF-THEN-rules?
3. Finally, a crisp value for the braking force is needed.

Answers:
Reconsidering the Approach

Obvious Problems:

1. The input parameters will be crisp.

2. One fuzzy IF-THEN-rule will not be sufficient.
   ~ How do we integrate many IF-THEN-rules?

3. Finally, a crisp value for the braking force is needed.

Answers:

1. fuzzification
Reconsidering the Approach

Obvious Problems:

1. The input parameters will be crisp.
2. One fuzzy IF-THEN-rule will not be sufficient.
   \(\leadsto\) How do we integrate many IF-THEN-rules?
3. Finally, a crisp value for the braking force is needed.

Answers:

1. fuzzification
2. either (1) do inference for each rule and (2) aggregate later or (1) aggregate all rules and (2) do inference
Reconsidering the Approach

Obvious Problems:

1. The input parameters will be crisp.
2. One fuzzy IF-THEN-rule will not be sufficient.
   ~ How do we integrate many IF-THEN-rules?
3. Finally, a crisp value for the braking force is needed.

Answers:

1. fuzzification
2. either (1) do inference for each rule and (2) aggregate later or (1) aggregate all rules and (2) do inference
3. defuzzification
Fuzzification

In general, we may convert crisp inputs into fuzzy sets arbitrarily.
Fuzzification

In general, we may convert crisp inputs into fuzzy sets arbitrarily.

Often, inputs are numbers $\mapsto$ fuzzy numbers
Fuzzification

In general, we may convert crisp inputs into fuzzy sets arbitrarily.

Often, inputs are numbers \( \rightsquigarrow \) fuzzy numbers

Most often, crisp inputs are taken as singletons.
**Fuzzification**

**In general**, we may convert crisp inputs into fuzzy sets arbitrarily.

Often, inputs are numbers $\mapsto$ fuzzy numbers

**Most often**, crisp inputs are taken as **singletons**.

Given such a crisp value, for each linguistic term $\in T(x)$ its membership function yields a degree of membership.
Fuzzification

In general, we may convert crisp inputs into fuzzy sets arbitrarily.

Often, inputs are numbers \( \rightsquigarrow \) fuzzy numbers

Most often, crisp inputs are taken as singletons.

Given such a crisp value, for each linguistic term \( \in T(x) \) its membership function yields a degree of membership.

Thus, we get a vector of length \( |T(x)| \) for one crisp value.
Fuzzification

Example: linguistic variable: speed
crisp input: 160 (in km/h)

Using 160 as a singleton yields:
Fuzzification

Example: linguistic variable: speed

crisp input: 160 (in km/h)

Using 160 as a singleton yields:
Fuzzification

Example: linguistic variable: speed
crisp input: 160 (in km/h)

Using 160 as a singleton yields:
Example: linguistic variable: speed
crisp input: 160 (in km/h)

Using 160 as a singleton yields:
Example: linguistic variable: speed
crisp input: 160 (in km/h)

Using 160 as a singleton yields:
Fuzzification

Example: linguistic variable: speed
crisp input: 160 (in km/h)

Using 160 as a singleton yields:
**Example:** linguistic variable: speed
crisp input: 160 (in km/h)

Using 160 as a singleton yields: 

\[
\begin{array}{c}
\text{speed} \\
\text{crisp input: } 160 \text{ (in km/h)} \\
\text{Using } 160 \text{ as a singleton yields: }
\end{array}
\]

\[
\begin{pmatrix}
0.54 \\
0.40 \\
\end{pmatrix}
\]
Fuzzy Inference for a Single Rule

Example: IF speed=fast THEN brake=sharply.
   crisp input: 160 (in km/h)
Fuzzy Inference for a Single Rule

Example: IF speed = fast THEN brake = sharply.
crisp input: 160 (in km/h)

We already know: [fast(160)] = .54
Fuzzy Inference for a Single Rule

**Example:** IF speed=fast THEN brake=sharply. crisp input: 160 (in km/h)

**We already know:** \([\text{fast}(160)] = .54\)

**Definition:** \(\text{sharply}(x) = \begin{cases} 0 & \text{if } x < 50 \\ 1 & \text{if } x > 90 \\ \frac{x-50}{40} & \text{otherwise} \end{cases}\)
Fuzzy Inference for a Single Rule

**Example:** IF speed=fast THEN brake=sharply.

crisp input: 160 (in km/h)

We already know: \([\text{fast}(160)] = .54\)

**Definition:**
\[
\text{sharply}(x) = \begin{cases} 
0 & \text{if } x < 50 \\
1 & \text{if } x > 90 \\
\frac{x-50}{40} & \text{otherwise}
\end{cases}
\]

What do we get if we use
- \(\phi_{tm}\) for IF-THEN?
Fuzzy Inference for a Single Rule

Example: IF speed=fast THEN brake=sharply.
crisp input: 160 (in km/h)

We already know: [fast(160)] = .54

Definition: sharply(x) = \[
\begin{cases} 
0 & \text{if } x < 50 \\
1 & \text{if } x > 90 \\
\frac{x-50}{40} & \text{otherwise}
\end{cases}
\]

What do we get if we use

- $\phi_{tm}$ for IF-THEN?
- $\phi_{tp}$ for IF-THEN?
Fuzzy Inference for a Single Rule

Example: IF speed=fast THEN brake=sharply.
crisp input: 160 (in km/h)

We already know: \([\text{fast}(160)] = .54\)

Definition: \(\text{sharply}(x) = \begin{cases} 
0 & \text{if } x < 50 \\
1 & \text{if } x > 90 \\
\frac{x-50}{40} & \text{otherwise}
\end{cases}\)

What do we get if we use
- \(\phi_{tm}\) for IF-THEN?
- \(\phi_{tp}\) for IF-THEN?
- \(t_m\) for IF-THEN?
Example continued

Example: IF speed=fast THEN brake=sharply.

crisp input: 160 (in km/h), [fast(160)] = .54
Example continued

Example: IF speed=fast THEN brake=sharply.

crisp input: 160 (in km/h), [fast(160)] = .54
Example continued

**Example:** IF speed = fast THEN brake = sharply.

crisp input: 160 (in km/h), \([\text{fast}(160)] = 0.54\)
Example continued

**Example:** IF speed = fast THEN brake = sharply.

crisp input: 160 (in km/h), [fast(160)] = .54
Example continued

Example: IF speed=fast THEN brake=sharply.

crisp input: 160 (in km/h), \([\text{fast}(160)] = 0.54\)
Fuzzy Inference for Many Rules

FITA: First Inference, Then Aggregate.
Fuzzy Inference for Many Rules

FITA: First Inference, Then Aggregate.

Do inference for each rule → one fuzzy set for each rule
Fuzzy Inference for Many Rules

**FITA:** First Inference, Then Aggregate.

Do inference for each rule $\rightarrow$ one fuzzy set for each rule

Aggregate results. — often: $t$-conorm
Fuzzy Inference for Many Rules

**FIT A**: *First Inference, Then Aggregate.*

Do inference for each rule → one fuzzy set for each rule

Aggregate results. — *often*: $t$-conorm

---

**FATI**: *First Aggregate, Then Inference.*
Fuzzy Inference for Many Rules

FIT A: First Inference, Then Aggregate.

Do inference for each rule → one fuzzy set for each rule

Aggregate results. — often: $t$-conorm

FATI: First Aggregate, Then Inference.

Aggregate all IF-THEN-Rules.

Do inference for the aggregated rule.
Fuzzy Inference for Many Rules

FITA: First Inference, Then Aggregate.
Do inference for each rule → one fuzzy set for each rule
Aggregate results. — *often*: $t$-conorm

FATI: First Aggregate, Then Inference.
Aggregate all IF-THEN-Rules.
Do inference for the aggregated rule.

Depending on operators, FATI and FITA may be equivalent.
Defuzzification

Our result is a fuzzy set.

In most applications, we need a crisp value.
Defuzzification

Our result is a fuzzy set.

In most applications, we need a crisp value.

For fuzzy set $M$ define crisp value $m$ by
Defuzzification

Our result is a fuzzy set.

In most applications, we need a crisp value.

For fuzzy set $M$ define crisp value $m$ by

- **center of gravity:** $m := \frac{\int_{\mathcal{U}} uM(u) \, du}{\int_{\mathcal{U}} M(u) \, du}$ for continuous $\mathcal{U}$

$$m := \frac{\sum_{\mathcal{U}} uM(u)}{\sum_{\mathcal{U}} M(u)}$$ for discrete $\mathcal{U}$
Defuzzification

Our result is a fuzzy set.

In most applications, we need a crisp value.

For fuzzy set $M$ define crisp value $m$ by

- **center of gravity:** $m := \frac{\int uM(u) \, du}{\int M(u) \, du}$ for continuous $\mathcal{U}$

  $$m := \frac{\sum_{\mathcal{U}} uM(u)}{\sum_{\mathcal{U}} M(u)}$$ for discrete $\mathcal{U}$

- **first of maxima:** $m := \min\{u \mid M(u) = \max_{\mathcal{V}} M(v)\}$
Defuzzification

Our result is a fuzzy set.

In most applications, we need a crisp value.

For fuzzy set $M$ define crisp value $m$ by

- **center of gravity:** $m := \frac{\int uM(u) \, du}{\int M(u) \, du}$ for continuous $\mathcal{U}$

  $$ m := \frac{\sum uM(u)}{\sum M(u)} $$ for discrete $\mathcal{U}$

- **first of maxima:** $m := \min \{u \mid M(u) = \max_v M(v)\}$

- **middle of maxima:** Let $\mathcal{N}$ be the set of maxima of $M$.

  $$ m := \frac{\int u \, du}{\int 1 \, du} $$ for continuous $\mathcal{N}$

  $$ m := \frac{1}{|\mathcal{N}|} \sum_{u \in \mathcal{N}} u $$ for discrete $\mathcal{N}$
Defuzzification of Example

Example: IF speed=fast THEN brake sharply.
crisp input: 160 (in km/h), [fast(160)] = .54
Defuzzification of Example

Example: IF speed=fast THEN brake sharply.

crisp input: 160 (in km/h), [fast(160)] = .54

\[ \phi_{tm} \]
Defuzzification of Example

Example: IF speed = fast THEN brake sharply.

crisp input: 160 (in km/h), [fast(160)] = .54

\[ \phi_{tm} \]

Center of Gravity: 82.40186
Defuzzification of Example

Example: IF speed = fast THEN brake sharply.
crisp input: 160 (in km/h), [fast(160)] = .54

\( \phi_{tp} \)
Defuzzification of Example

Example: IF speed=fast THEN brake sharply.

crisp input: 160 (in km/h), [fast(160)] = .54

\[ \phi_{tp} \]

Center of Gravity: 80.16259
Defuzzification of Example

Example: IF speed=fast THEN brake sharply.
crisp input: 160 (in km/h), \([\text{fast}(160)] = 0.54\)
Defuzzification of Example

Example: IF speed = fast THEN brake sharply.

crisp input: 160 (in km/h), [fast(160)] = .54

t_m

Center of Gravity: 82.16256
General Fuzzy Controller

crisp input \( x \in \mathcal{U} \)

Fuzzifier

fuzzy set in \( \mathcal{U} \)

Fuzzy Rule Base

Fuzzy Inference Engine

fuzzy set in \( \mathcal{V} \)

defuzzifier

crisp \( y \in \mathcal{V} \)
Mamdani Fuzzy Controller

Different types of fuzzy controllers are known.

Probably the best known: Mamdani (1974)
Mamdani Fuzzy Controller

Different types of fuzzy controllers are known.

Probably the best known: Mamdani (1974)

- use $t_m$ for ‘and’ in IF-part
Mamdani Fuzzy Controller

Different types of fuzzy controllers are known.

Probably the best known: Mamdani (1974)

- use $t_m$ for ‘and’ in IF-part
- use FITA
Mamdani Fuzzy Controller

Different types of fuzzy controllers are known.

Probably the best known: Mamdani (1974)

- use $t_m$ for ‘and’ in IF-part
- use FITA
- use $t_m$ for IF-THEN inference
Mamdani Fuzzy Controller

Different types of fuzzy controllers are known.

Probably the best known: Mamdani (1974)

- use $t_m$ for ‘and’ in IF-part
- use FITA
- use $t_m$ for IF-THEN inference
- use $s_{t_m}$ for aggregation
Mamdani Fuzzy Controller

Different types of fuzzy controllers are known.

Probably the best known: Mamdani (1974)

- use $t_m$ for ‘and’ in IF-part
- use FITA
- use $t_m$ for IF-THEN inference
- use $s_{tm}$ for aggregation
- use center of gravity defuzzification
Mamdani Fuzzy Controller

Different types of fuzzy controllers are known. Probably the best known: Mamdani (1974)

- use $t_m$ for ‘and’ in IF-part
- use FITA
- use $t_m$ for IF-THEN inference
- use $s_{t_m}$ for aggregation
- use center of gravity defuzzification

Properties:
Mamdani Fuzzy Controller

Different types of fuzzy controllers are known.

Probably the best known: Mamdani (1974)

- use $t_m$ for ‘and’ in IF-part
- use FITA
- use $t_m$ for IF-THEN inference
- use $s_{t_m}$ for aggregation
- use center of gravity defuzzification

Properties:

- intuitively understandable
Mamdani Fuzzy Controller

Different types of fuzzy controllers are known.

Probably the best known: Mamdani (1974)

- use $t_m$ for ‘and’ in IF-part
- use FITA
- use $t_m$ for IF-THEN inference
- use $s_{tm}$ for aggregation
- use center of gravity defuzzification

Properties:

- intuitively understandable
- easy to construct
Mamdani Fuzzy Controller

Different types of fuzzy controllers are known.

Probably the best known: Mamdani (1974)

- use $t_m$ for ‘and’ in IF-part
- use FITA
- use $t_m$ for IF-THEN inference
- use $s_{tm}$ for aggregation
- use center of gravity defuzzification

Properties:

- intuitively understandable
- easy to construct
- does not support data-driven automatic construction
Design of Mamdani Controllers

Heuristics for the construction Mamdani Controllers:
Design of Mamdani Controllers

Heuristics for the construction Mamdani Controllers:

- **Completeness**: For any possible input, have at least one rule.
Design of Mamdani Controllers

**Heuristics for the construction Mamdani Controllers:**

- **Completeness:** For any possible input, have at least one rule.
- **Consistency:** Don’t have two rules with equal IF-part and different THEN-parts.
Design of Mamdani Controllers

Heuristics for the construction Mamdani Controllers:

• **Completeness**: For any possible input, have at least one rule.

• **Consistency**: Don’t have two rules with equal IF-part and different THEN-parts.

• **Computational complexity**: Use membership functions that allow for easy calculations of membership values. Particularly good: triangular and trapezoidal membership functions
Design of Mamdani Controllers

Heuristics for the construction of Mamdani Controllers:

- **Completeness**: For any possible input, have at least one rule.
- **Consistency**: Don’t have two rules with equal IF-part and different THEN-parts.
- **Computational complexity**: Use membership functions that allow for easy calculations of membership values. Particularly good: triangular and trapezoidal membership functions.
- **Note**: Center of gravity defuzzification implies that extreme values cannot be obtained as output.
Implications for Inference

What changes if we use $\phi_t$ for IF-THEN?
Implications for Inference

What changes if we use $\phi_t$ for IF-THEN?

Smaller values in IF-part yield larger results.
Implications for Inference

What changes if we use $\phi_t$ for IF-THEN?

Smaller values in IF-part yield larger results.

New interpretation:
Implications for Inference

What changes if we use $\phi_t$ for IF-THEN?

Smaller values in IF-part yield larger results.

New interpretation:
Result is description of possibilities:
anything that cannot be ruled out.
Implications for Inference

What changes if we use $\phi_t$ for IF-THEN?

Smaller values in IF-part yield larger results.

New interpretation:
Result is description of possibilities:
anything that cannot be ruled out.

Less useful in technical applications.
Takagi-Sugeno-Kang Fuzzy Controller

Takagi/Sugeno/Kang (1975):

**Motivation:**

- allow for automatic data-driven tuning of parameters
- reduce number of fuzzy rules needed
Takagi-Sugeno-Kang Fuzzy Controller

Takagi/Sugeno/Kang (1975):

Motivation:

• allow for automatic data-driven tuning of parameters
• reduce number of fuzzy rules needed

Change format of IF-THEN-rules:

• leave IF-part unchanged
Takagi-Sugeno-Kang Fuzzy Controller

Takagi/Sugeno/Kang (1975):

Motivation:

- allow for automatic data-driven tuning of parameters
- reduce number of fuzzy rules needed

Change format of IF-THEN-rules:

- leave IF-part unchanged
- replace linguistic variable in THEN-part by linear function $f$
Takagi-Sugeno-Kang Fuzzy Controller

Takagi/Sugeno/Kang (1975):

Motivation:

- allow for automatic data-driven tuning of parameters
- reduce number of fuzzy rules needed

Change format of IF-THEN-rules:

- leave IF-part unchanged
- replace linguistic variable in THEN-part by linear function $f$

Consequence: Each rule describes conditions for application of specific linear model.
**Takagi-Sugeno-Kang Fuzzy Controller**

Takagi/Sugeno/Kang (1975):

**Motivation:**
- allow for automatic data-driven tuning of parameters
- reduce number of fuzzy rules needed

**Change format of IF-THEN-rules:**
- leave IF-part unchanged
- replace linguistic variable in THEN-part by linear function $f$

**Consequence:** Each rule describes conditions for application of specific linear model.

Aggregation yields aggregation of (independent) linear models.
Takagi-Sugeno-Kang Fuzzy Controller

Takagi/Sugeno/Kang (1975):

**Motivation:**
- allow for automatic data-driven tuning of parameters
- reduce number of fuzzy rules needed

**Change format of IF-THEN-rules:**
- leave IF-part unchanged
- replace linguistic variable in THEN-part by linear function $f$

**Consequence:** Each rule describes conditions for application of specific linear model.

Aggregation yields aggregation of (independent) linear models.

No defuzzification necessary.
Takagi-Sugeno-Kang Fuzzy Controllers

Takagi-Sugeno-Kang fuzzy controllers have many parameters.
Takagi-Sugeno-Kang Fuzzy Controllers

Takagi-Sugeno-Kang fuzzy controllers have many parameters.

How can we find appropriate linear functions?
Takagi-Sugeno-Kang Fuzzy Controllers

Takagi-Sugeno-Kang fuzzy controllers have many parameters.

How can we find appropriate linear functions?

- use analytical models if available
Takagi-Sugeno-Kang Fuzzy Controllers

Takagi-Sugeno-Kang fuzzy controllers have many parameters.

How can we find appropriate linear functions?

- use analytical models if available
- observe existing controller and derive linear model
Takagi-Sugeno-Kang Fuzzy Controllers

Takagi-Sugeno-Kang fuzzy controllers have many parameters. How can we find appropriate linear functions?

- use analytical models if available
- observe existing controller and derive linear model
- optimize numerically (for example using evolutionary algorithms)
Takagi-Sugeno-Kang Fuzzy Controllers

Takagi-Sugeno-Kang fuzzy controllers have many parameters.

How can we find appropriate linear functions?

- use analytical models if available
- observe existing controller and derive linear model
- optimize numerically (for example using evolutionary algorithms)
- use some ‘learning approach’ (for example artificial neural nets)
Conclusion

- Fuzzy sets allow to formalize linguistic concepts (small, big, cold, warm, slow, fast,...)
Conclusion

- Fuzzy sets allow to formalize linguistic concepts (small, big, cold, warm, slow, fast,...)
- Many possible operators on fuzzy sets: $t$-norm for conjunction, $t$-conorm for disjunction, $\Phi$-operator for implication
Conclusion

- Fuzzy sets allow to formalize linguistic concepts (small, big, cold, warm, slow, fast,...)
- Many possible operators on fuzzy sets: $t$-norm for conjunction, $t$-conorm for disjunction, $\Phi$-operator for implication
- Axiomatic approach allows to introduce new operators easily
Conclusion

- Fuzzy sets allow to formalize linguistic concepts (small, big, cold, warm, slow, fast, ...)
- Many possible operators on fuzzy sets: $t$-norm for conjunction, $t$-conorm for disjunction, $\Phi$-operator for implication
- Axiomatic approach allows to introduce new operators easily
- Fuzzy IF-THEN-rules formalize intuitive knowledge about control rules
Conclusion

- Fuzzy sets allow to formalize linguistic concepts (small, big, cold, warm, slow, fast,...)
- Many possible operators on fuzzy sets: $t$-norm for conjunction, $t$-conorm for disjunction, $\Phi$-operator for implication
- Axiomatic approach allows to introduce new operators easily
- Fuzzy IF-THEN-rules formalize intuitive knowledge about control rules
- Fuzzy control allows to build controllers based on intuitive knowledge only