Sztuczna inteligencja

WYKŁAD 6

Logika

- Logika – narzędzie formalnego opisu dedukcji.
- Dedukcja – wyprowadzanie prawdziwych stwierdzeń (wniosków) ze stwierdzeń przyjmowanych za prawdziwe (przesłanek).

Wnioskowanie logiczne

- Sylogizm
  - przykład:

    Przesłanka: Wszyscy ludzie są śmiertelni

    Przesłanka: A jest człowiekiem

    Wniosek: A jest śmiertelny

Wnioskowanie logiczne

Reguła *modus ponens*:

A
JEŚLI A, TO B

B

Z prawdziwości ***A*** oraz ***JEŚLI A, TO B*** wynika prawdziwość ***B***.

Wnioskowanie logiczne

Reguła *modus tollens*:

NIE B
JEŚLI A, TO B

NIE A

Z fałszywości ***B*** oraz prawdziwości ***JEŚLI A, TO B*** wynika fałszywość ***A***.

Rachunek zdań i rachunek predykatów

- W rachunku zdań stwierdzenia są zbudowane ze zmiennych zdaniowych reprezentujących zdania niemające wewnętrznej struktury.
- Niektóre teorie matematyczne nie mogą być wyrażone za pomocą rachunku zdań.
- Rachunek predykatów (logika pierwszego rzędu) – system logiczny dopuszczający funkcje określone w pewnych dziedzinach (np. liczbowych) o wartościach logicznych.

Rachunek predykatów

- Przykłady:
  - Dla każdej chwili czasowej *t* wartość wyrażenia 10>5 jest zawsze prawdą
  - Istnieje chwila czasowa *t* taka, że stwierdzenie „temperatura powietrza w chwili *t* wynosi 25 st. C" jest prawdziwe

Programowanie w logice

- Programowanie w logice – zastosowanie systemów dowodzenia twierdzeń do wykonywania obliczeń.
- Programując w logice nie podaje się kolejnych kroków obliczeniowych koniecznych do znalezienia rozwiązania, lecz tworzy się formułę logiczną opisującą relacje między danymi wejściowymi a wyjściowymi, a następnie uruchamia się system dowodzenia twierdzeń w celu znalezienia odpowiedzi.
- Programowanie w logice jest programowaniem deklaratywnym.

Klauzula Horna

- Klauzulą Horna nazywamy klauzulę

$$A \leftarrow B_1, B_2, \dots, B_m$$

mającą co najwyżej jeden literał pozytywny.

- Literał pozytywny *A* nazywany jest nagłówkiem klauzuli.
- Literały negatywne

$$B_1, B_2, \dots, B_m$$

nazywane są treścią klauzuli.

Klauzula Horna

- Pozytywna klauzula Horna

$$A \leftarrow$$

nazywana jest faktem (klauzulą unarną).

- Klauzula Horna

$$\leftarrow B_1, B_2, \ldots, B_m$$

bez żadnego literału pozytywnego nazywana jest klauzulą negatywną.

- Klauzula Horna zawierającą jeden literał pozytywny i dowolną liczbę literałów negatywnych nazywana jest klauzulą programu.

Postać klauzulowa twierdzeń w rachunku predykatów

- Przekształcenie wyrażenia rachunku predykatów do postaci klauzulowej:
  - **ETAP 1**: Przemianowanie zmiennyc tak, aby żadna zmienna nie występowała w dwóch kwantyfikatorach jednocześnie.
  - **ETAP 2:** Usunięcie wszystkich operatorów logicznych z wyjątkiem negacji, alternatywy i koniunkcji przez zastosowanie równoważności logicznych.
  - **ETAP 3**: Przekształcenie wyrażenia do postaci, w której negacje występują tylko przy wyrażeniach atomowych.

Postać klauzulowa twierdzeń w rachunku predykatów

- Przekształcenie wyrażenia rachunku predykatów do postaci klauzulowej (cd.):
  - **ETAP 4**: Skolemizacja - wyeliminowanie kwantyfikatorów egzystencjalnych przez wprowadzenie stałych Skolema lub funkcji Skolema w miejsce zmiennych wprowadzanych przez te kwantyfikatory, np.:

Stała Skolema

$$\exists_{x} \alpha(x) \Rightarrow \alpha(K)$$

Funkcja Skolema

$$\forall_{y} \exists_{x} \alpha(x) \Rightarrow \alpha(f(y))$$

Postać klauzulowa twierdzeń w rachunku predykatów

- Przekształcenie wyrażenia rachunku predykatów do postaci klauzulowej (cd.):
  - **ETAP 5**: Przesunięcie kwantyfikatorów uniwersalnych na zewnątrz (poza wyrażenie).
  - **ETAP 6**: Przekształcenie wyrażenia do normalnej postaci koniunkcyjnej.
  - **ETAP 7**: Przekształcenie wyrażenia w klauzule.

Równoważności logiczne

$$\neg \underset{x}{\forall}\ A(x) \Leftrightarrow \underset{x}{\exists} \neg A(x)$$

$$\neg \underset{x}{\exists} A(x) \Leftrightarrow \underset{x}{\forall}\ \neg A(x)$$

$$A\ op \underset{x}{Q}\ B(x) \Leftrightarrow \underset{x}{Q}\ A\ op\ B(x)$$

$$\underset{x}{Q}\ A(x)\ op\ B \Leftrightarrow \underset{x}{Q}\left(A(x)\ op\ B\right)$$

- *Q* - kwantyfikator
- *op* - operator ($\wedge$, $\vee$)

Zapis klauzul

$$P_1; P_2; \ldots; P_m :- Q_1, Q_2, \ldots, Q_n$$

- Literały bez negacji:

  $P_1, P_2, \ldots, P_m$

- Literały z negacjami:

  $Q_1, Q_2, \ldots, Q_n$

Procedura, program w logice, baza danych

- Procedura – zbiór klauzul Horna, w których nagłówku występuje ten sam symbol predykatywny.
- Program w logice – zbiór procedur.
- Baza danych – procedura złożona wyłącznie z ustalonych faktów.

Język Prolog

- Język Prolog jest deklaratywnym językiem programowania w logice.
- Program w Prologu składa się ze zbioru klauzul.
- Programowanie w Prologu składa się z:
  - Deklarowania faktów.
  - Definiowania reguł.
  - Konstruowania zapytań.

Fakty

***relacja*(*obiekt*1, *obiekt*2, ..., *obiekt*N).**

- Przykłady:

rodzic(jan, piotr).

kobieta(anna).

przesyla(komputer1, dane, komputer2).

- Nazwa relacji jest predykatem
- Obiekty nazywamy argumentami

Zapytania

?- *relacja*(*obiekt*1, *obiekt*2, ..., *obiekt*N).

- Przykłady:

?- rodzic(jan, piotr).

?- kobieta(anna).

?- przesyla(komputer1, dane, komputer2).

Zmienne

- Nazwy zmiennych w Prologu muszą zaczynać się od dużych liter.
- Jeśli w zapytaniu występuje zmienna, Prolog przeszukuje wszystkie fakty, aby znaleźć obiekt, który można podstawić za zmienną.
- Prolog przeszukuje bazę danych w takiej kolejności w jakiej została zapisana.

Koniunkcja

?- *relacja*1(*obiekt*11, *obiekt*21, ..., *obiekt*N1),

...,

***relacja*K(obiekt1K, *obiekt*2K, ..., *obiekt*MK).**

- Przykład:

  ?- rodzic(jan, X), rodzic(piotr, X).

Reguły

- Reguły to ogólne stwierdzenia dotyczące obiektów i ich powiązań
- Reguła składa się z głowy i treści:

głowa* :- *treść

- Przykład:

siostra(X,Y) :- kobieta(X), rodzice(X,M,O), rodzice(Y,M,O).

  - M – matka
  - O - ojciec

Zbiór klasyczny („ostry”)

W zbiorze klasycznym elementy albo należą, albo nie należą do tego zbioru.

$a \in A$ - element a należy do zbioru A

$b \notin A$ - element b nie należy do zbioru A

Funkcja charakterystyczna:

$\phi_A(x) = 1$ - gdy element x należy do zbioru A

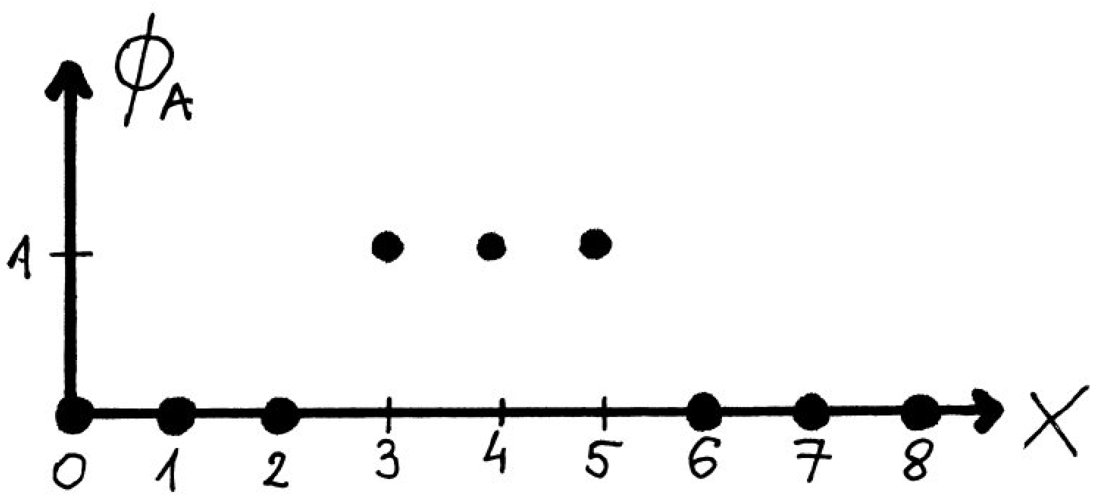
$\phi_A(x) = 0$ - gdy element x nie należy do zbioru A

Zbiór klasyczny („ostry”)

Przykład

U – uniwersum (przestrzeń rozważań): zbiór liczb $\{0,1,2,3,4,5,6,7,8\}$
A – zbiór liczb bliskich liczbie 4,
np. $A=\{3,4,5\}$

$$3,4,5 \in A$$
$$0,1,2,6,7,8 \notin A$$



Zbiór rozmyty

W zbiorze rozmytym elementy mogą należeć do tego zbioru w pewnym stopniu (częściowo).

Funkcja przynależności:

$\mu_A(x) = 1$ - gdy element *x* w pełni należy do zbioru *A*

$0 < \mu_A(x) < 1$ - gdy element *x* częściowo należy do zbioru *A*

$\mu_A(x) = 0$ - gdy element *x* nie należy do zbioru *A*

Zbiór rozmyty

$$A=\{(x,\mu_A(x)):x\in \boldsymbol{X}\},\quad gdzie\ \mu_A\in[0,1]$$

$\boldsymbol{X}$ – uniwersum (przestrzeń rozważań)

$$A=\frac{\mu_A(x_1)}{x_1}+\frac{\mu_A(x_2)}{x_2}+\ldots+\frac{\mu_A(x_k)}{x_k}$$

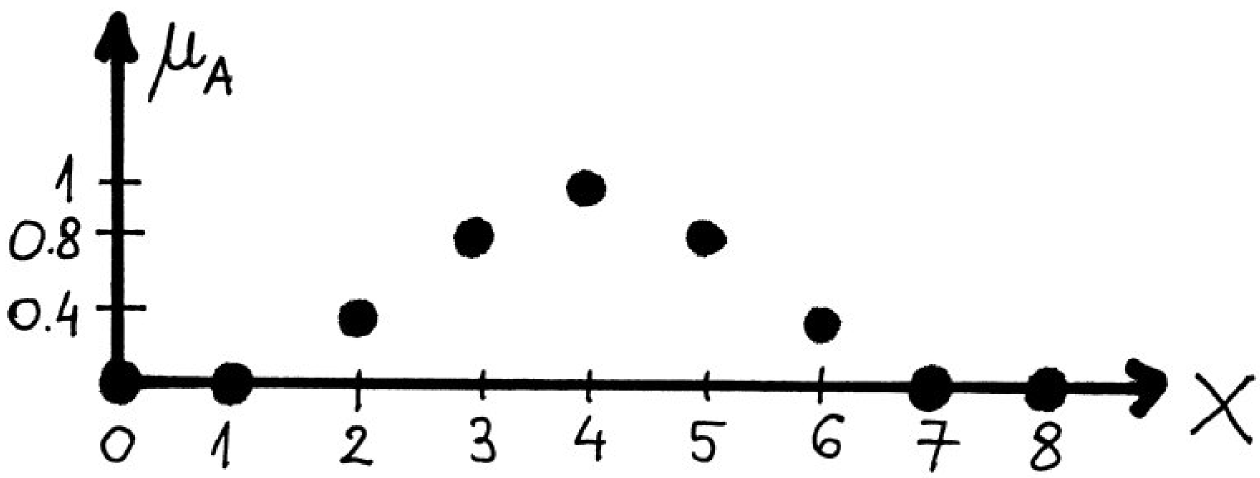
$$x_{1,}x_{2,}\ldots,x_k\in \boldsymbol{X}$$

Zbiór rozmyty

Przykład

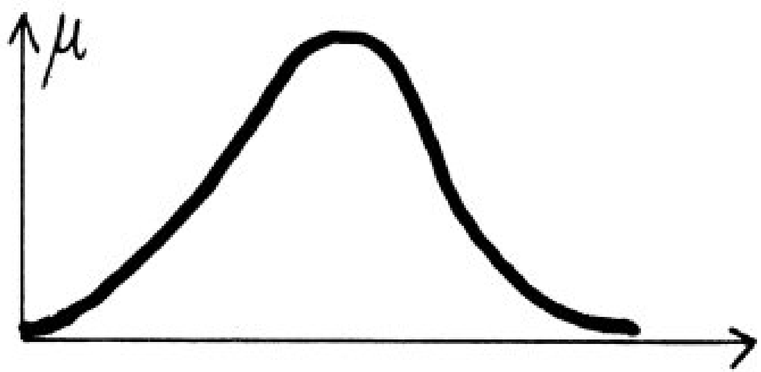
U – uniwersum (przestrzeń rozważań): zbiór liczb {0,1,2,3,4,5,6,7,8}
A – zbiór liczb bliskich liczbie 4,

$$A=\frac{0}{0}+\frac{0}{1}+\frac{0.4}{2}+\frac{0.8}{3}+\frac{1}{4}+\frac{0.8}{5}+\frac{0.4}{6}+\frac{0}{7}+\frac{0}{8}$$

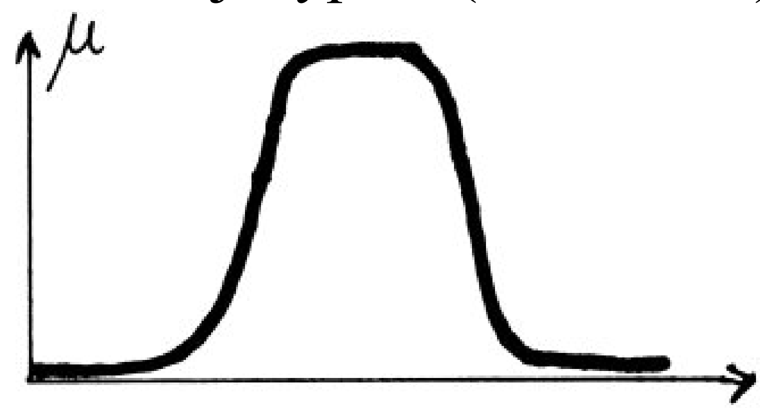


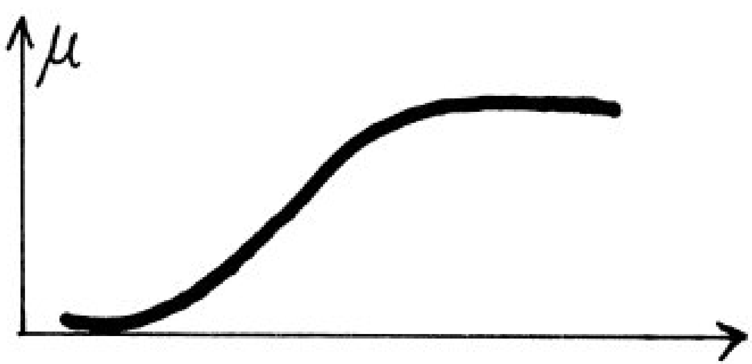
Typy funkcji przynależności

Funkcja typu Gaussa



Funkcja typu π (dzwonowa)

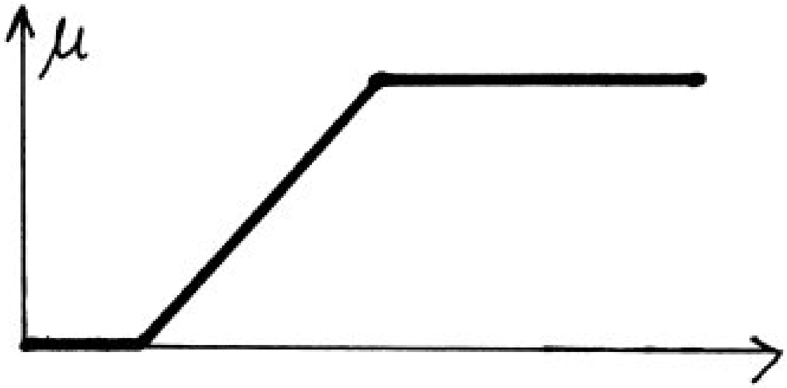




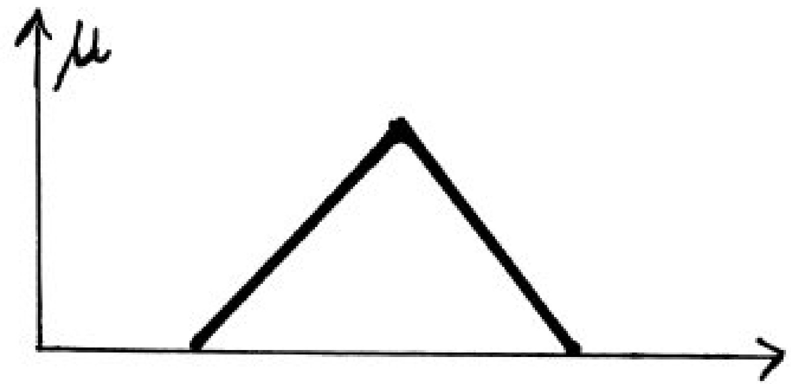
Funkcja typu s (sigmoidalna)

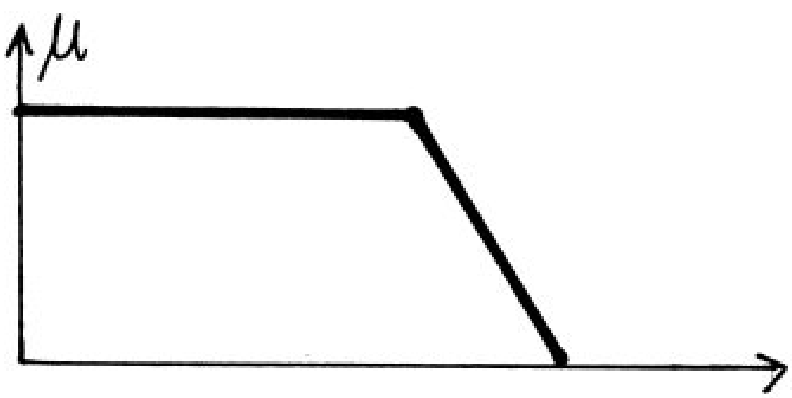
Typy funkcji przynależności

Funkcja typu Γ

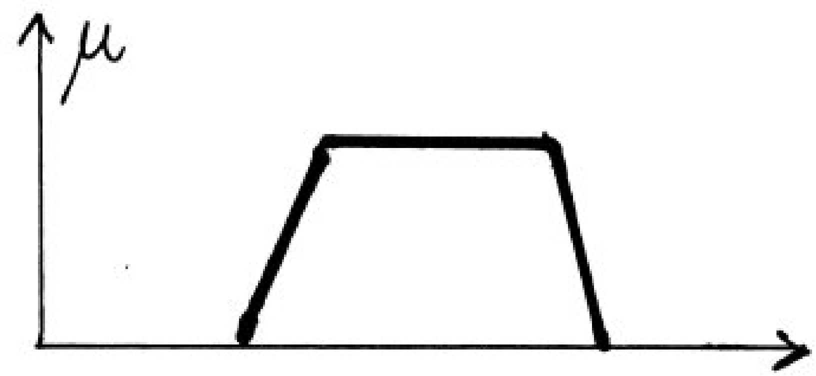


Funkcja typu Λ (trójkątna)





Funkcja typu L



Funkcja typu Π

Operacje na zbiorach rozmytych

A, B – zbiory rozmyte w przestrzeni rozważań ***X***

<u>Przecięcie zbiorów rozmytych:</u>

$$\mu_{A \cap B}(x) = min(\mu_A(x), \mu_B(x)) \quad dla\,każdego\; x \in \boldsymbol{X}$$

<u>Suma zbiorów rozmytych:</u>

$$\mu_{A \cup B}(x) = max(\mu_A(x), \mu_B(x)) \quad dla\,każdego\; x \in \boldsymbol{X}$$

<u>Dopełnienie zbioru rozmytego:</u>

$$\mu_{\neg A}(x) = 1 - \mu_A(x) \quad dla\,każdego\; x \in \boldsymbol{X}$$

Normy trójkątne

A, B – zbiory rozmyte w przestrzeni rozważań ***X***

Uogólnienie przecięcia zbiorów rozmytych:

$$\mu_{A\cap B}(x) = T(\mu_A(x), \mu_B(x)) \quad dla\ każdego\ x \in \boldsymbol{X}$$

T – *t*-norma

Uogólnienie sumy zbiorów rozmytych:

$$\mu_{A\cup B}(x) = S(\mu_A(x), \mu_B(x)) \quad dla\ każdego\ x \in \boldsymbol{X}$$

S – *s*-norma (*t*-konorma)

Normy trójkątne

***t*-norma**

$$T:[0,1]\times[0,1]\to[0,1]$$

$$T(a,c)\leqslant T(b,d) \quad dla \quad a\leqslant b, c\leqslant d$$
$$T(a,b)=T(b,a)$$
$$T(T(a,b),c)=T(a,T(b,c))$$
$$T(a,1)=a$$
$$T(a,0)=0$$

$$a,b,c,d\in[0,1]$$

Normy trójkątne

s-norma

$$S:[0,1]\times[0,1]\rightarrow[0,1]$$

$$S(a,c)\leqslant S(b,d) \quad dla \quad a\leqslant b, c\leqslant d$$
$$S(a,b)=S(b,a)$$
$$S(S(a,b),c)=S(a,S(b,c))$$
$$S(a,0)=a$$
$$S(a,1)=1$$

$$a,b,c,d\in[0,1]$$

Normy trójkątne

Przykładowe *t*-normy

$T(a,b)=min(a,b)$ - minimum

$T(a,b)=a\cdot b$ - iloczyn algebraiczny

$T(a,b)=max(0,a+b-1)$ - *t*-norma Łukasiewicza

Przykładowe *s*-normy

$S(a,b)=max(a,b)$ - maksimum

$S(a,b)=a+b-a\cdot b$ - iloczyn probabilistyczny

$S(a,b)=min(a+b,1)$ - *s*-norma Łukasiewicza

Relacje rozmyte

X, Y – zbiory nierozmyte, R – relacja rozmyta określona w X x Y

$$R = \{((x, y), \mu_R(x, y)) : x \in X, y \in Y\}, \quad gdzie\, \mu_R(x, y) \in [0,1]$$

$$\begin{array}{ccccc}
 & y_1 & y_2 & \dots & y_m \\
x_1 & \mu_R(x_{1,} y_1) & \mu_R(x_{1,} y_2) & \dots & \mu_R(x_{1,} y_m) \\
x_2 & \mu_R(x_{2,} y_1) & \mu_R(x_{2,} y_2) & \dots & \mu_R(x_{2,} y_m) \\
\dots & \dots & \dots & \dots & \dots \\
x_n & \mu_R(x_n, y_1) & \mu_R(x_n, y_2) & \dots & \mu_R(x_n, y_m)
\end{array}$$

$$x_{1,} x_{2,} \dots, x_n \in X, y_{1,} y_{2,} \dots, y_m \in Y$$

Uogólniona rozmyta reguła *modus ponens*:

A' ← **zbiór rozmyty w *X***

JEŚLI A, TO B ← **implikacja rozmyta reprezentowana przez relację rozmytą *R* w *X* x *Y***

B'=? ← **zbiór rozmyty w *Y***

$$B' = A' \circ R$$

$$\mu_{B'} = \max_{x \in X} \left(T\left(\mu_{A'}(x), \mu_R(x, y) \right) \right)$$

Wnioskowanie rozmyte

Przykład

$X=\{1,2,3\}, \quad Y=\{1,2,3,4\}$

$$NISKIE=\frac{1}{1}+\frac{0.7}{2}+\frac{0.3}{3} \qquad WYSOKIE=\frac{0.2}{1}+\frac{0.5}{2}+\frac{0.8}{3}+\frac{1}{4}$$

$$R=NISKIE\times WYSOKIE$$

$X \quad Y$	1	2	3	4
1	0.2	0.5	0.8	1
2	0.2	0.5	0.7	0.7
3	0.2	0.3	0.3	0.3

Wnioskowanie rozmyte

Przykład

$$\acute{S}REDNIE = \frac{0.5}{1} + \frac{1}{2} + \frac{0.5}{3}$$

$$\acute{S}REDNIE \circ R = \max_{x \in \{1,2,3\}} \left(T\left(\mu_{\acute{S}REDNIE}(x), \mu_R(x,y) \right) \right) =$$

$$= \frac{0.2}{1} + \frac{0.5}{2} + \frac{0.7}{3} + \frac{0.7}{4}$$

Przyjmując jako *t*-normę minimum.