

Zbiory, logika i regulatory rozmyte

Treść zadania - najważniejsze!

- Zadanie trzeba wykonać **SAMODZIELNIE!!!**
- W przypadku pracy niesamodzielnej dostaje się nie tylko zero z obecnego projektu, ale też automatycznie zeruje się całe ćwiczenia i nie zalicza się przedmiotu

Zbiór

- Pojęcie pierwotne z teorii mnogości oznaczające zestaw (grupę) jakiś elementów (przy czym muszą to być elementy unikatowe)
- Przykłady zbiorów:
 - $\{1, 2, 3, 4\}$
 - Zbiór liczb naturalnych = $\{0, 1, 2, 3, \dots\}$
 - Zbiór liczb rzeczywistych
 - Zbiór dni tygodnia

Funkcja charakterystyczna

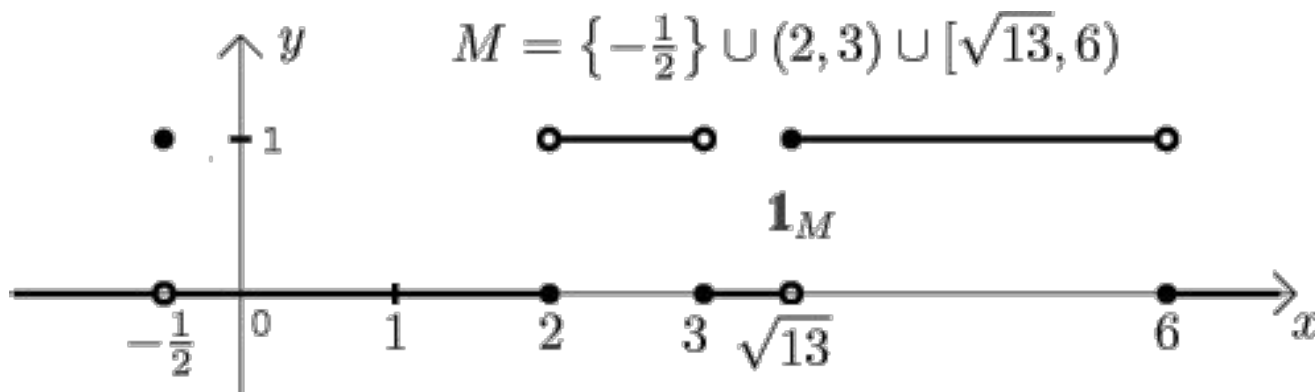
- W zwykłej teorii mnogości każdy element może tylko należeć lub nie należeć do jakiegoś zbioru
- W związku z tym każdy zbiór posiada swoją funkcję charakterystyczną (lub indykator) zdefiniowaną poniżej:

$$I_A(x) = \begin{cases} 1, & x \in A \\ 0, & x \notin A. \end{cases}$$

- Inne częste oznaczenia to $\mathbf{1}_A(\mathbf{x})$ lub $\chi_A(\mathbf{x})$ (to ostatnie to grecka litera chi, a nie “iks”)

Funkcja charakterystyczna

- Przykład:



Funkcja charakterystyczna

- Własności:
 - $1_{A \cap B}(x) = \min\{1_A(x), 1_B(x)\} = 1_A(x) \cdot 1_B(x)$
 - $1_{A \cup B}(x) = \max\{1_A(x), 1_B(x)\} = 1_A(x) + 1_B(x) - 1_A(x) \cdot 1_B(x)$
 - $1_{A^c}(x) = 1 - 1_A(x)$
- Gdzie A^c to dopełnienie zbioru A

Zbiory rozmyte

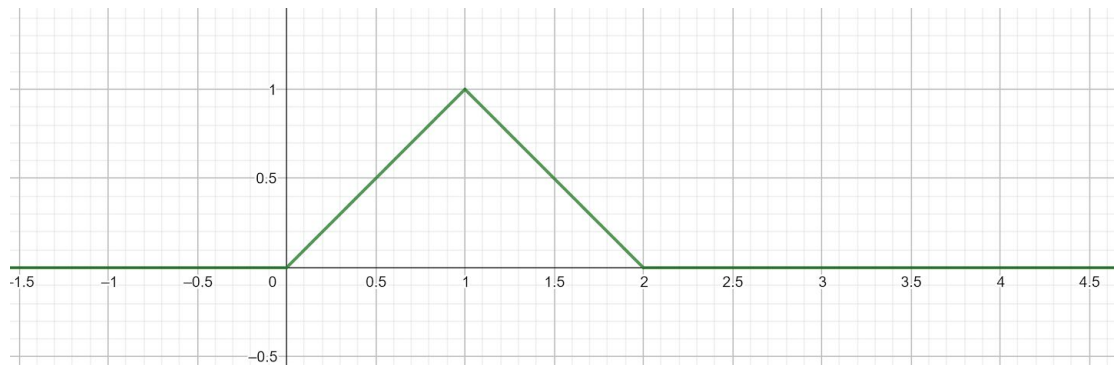
- Zwykłe zbiory (“ostre”) często nie oddają sposobu myślenia człowieka na różne tematy
- Na przykład trudno podzielić ludzi na dwa rozłączne podzbiory ludzi wysokich i niskich (trudno znaleźć obiektywne, jednoznaczne kryterium podziału)
- Stąd zrodził się pomysł, żeby stworzyć zbiory, do których elementy mogą należeć tylko w jakimś stopniu (a nie tylko należeć lub nie należeć jak dotychczas)
- Są one definiowane przez funkcję przynależności (to uogólnienie funkcji charakterystycznej)

Funkcja przynależności

- Opisuje jak bardzo dany element należy do tego zbioru:
 - 1 - oznacza, że całkowicie należy
 - 0 - całkowicie należy
 - 0.5 - należy w połowie
 - 0.1 - tylko w niewielkim stopniu należy
- Może to być dowolna funkcja, której wartości mieszczą się w przedziale $[0,1]$, ale jest kilka najczęściej stosowanych
- Najczęściej oznaczana przez $\mu_A(x)$

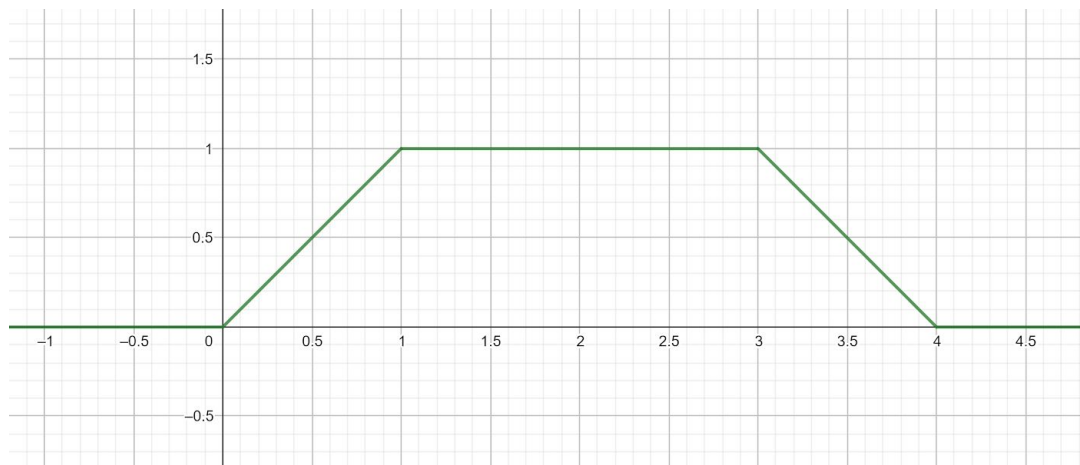
Funkcja przynależności

- Trójkątna
- Trapezowa
- Gaussowska
- Sigmoidalna



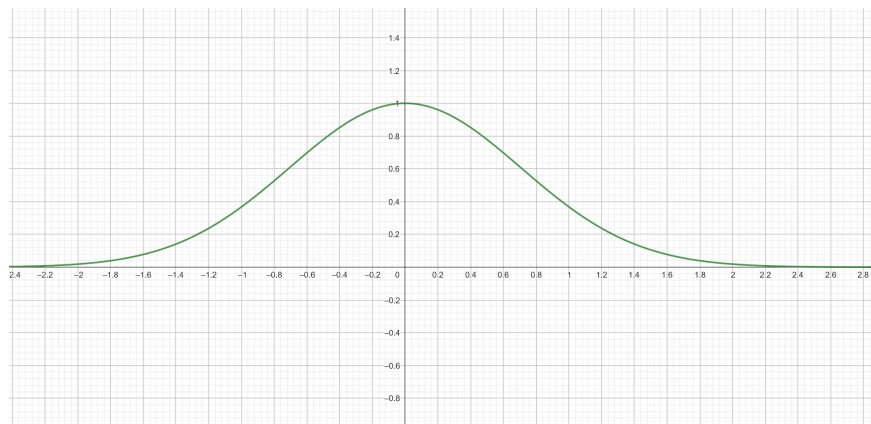
Funkcja przynależności

- Trójkątna
- **Trapezowa**
- Gaussowska
- Sigmoidalna



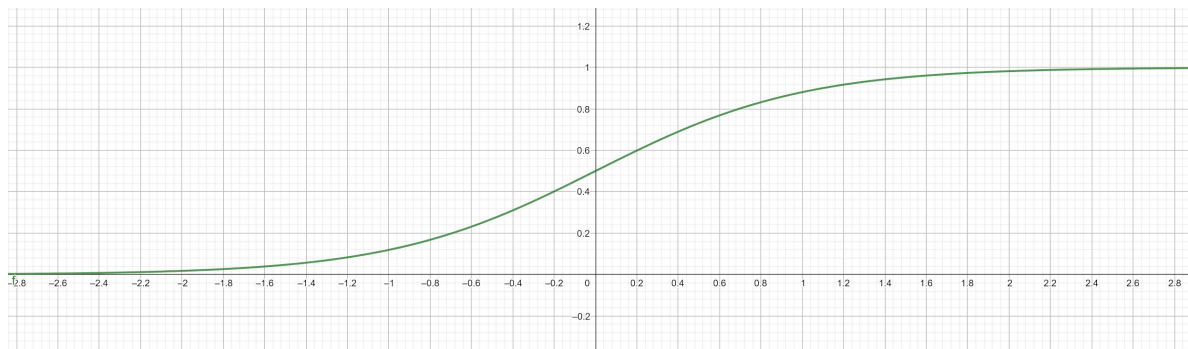
Funkcja przynależności

- Trójkątna
- Trapezowa
- Gaussowska
- Sigmoidalna



Funkcja przynależności

- Trójkątna
- Trapezowa
- Gaussowska
- Sigmoidalna



Zbiory rozmyte - działania

- Zestaw działań nr 1:
 - Suma: $\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}$
 - Przecięcie: $\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}$
 - Dopełnienie: $\mu_{A^c}(x) = 1 - \mu_A(x)$
- Zestaw działań nr 2:
 - Suma: $\mu_{A \cup B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x)$
 - Przecięcie: $\mu_{A \cap B}(x) = \mu_A(x) \cdot \mu_B(x)$
 - Dopełnienie: $\mu_{A^c}(x) = 1 - \mu_A(x)$
- Obydwa zestawy spełniają własności, których po nich oczekujemy (np. prawa De Morgana)
- Dla funkcji charakterystycznej są tożsame
- W kontekście działań na zbiorach rozmytych widziałem zawsze zestaw 1 (ale nie widzę szczególnego powodu, dlaczego akurat ten, bo drugi oraz wiele innych też posiada oczekiwane własności)

Logika rozmyta

- W pewnym sensie to to samo co slajd wcześniej (bo jeśli $x \in A \cup B$, to oznacza, że $x \in A$ **lub** $x \in B$, itd.)
- Żeby mieć cały system logiczny trzeba zdefiniować jak obliczać funkcje “**nie**”, “**i**” oraz “**lub**” (lub jakiś inny zestaw, który tworzy system funkcjonalnie pełny)
- “**nie**” będę oznaczać przez \neg i będzie zawsze obliczana przez:
 - $\neg \mu_A(x) = 1 - \mu_A(x)$
- “**i**” będę oznaczać przez \wedge i będzie definiowana przez T-normę
- “**lub**” będę oznaczać przez \vee i będzie definiowana na podstawie “**nie**” oraz “**i**” tak, żeby zgadzały się prawa De Morgana (czyli będzie to T-konorma)

T-norma

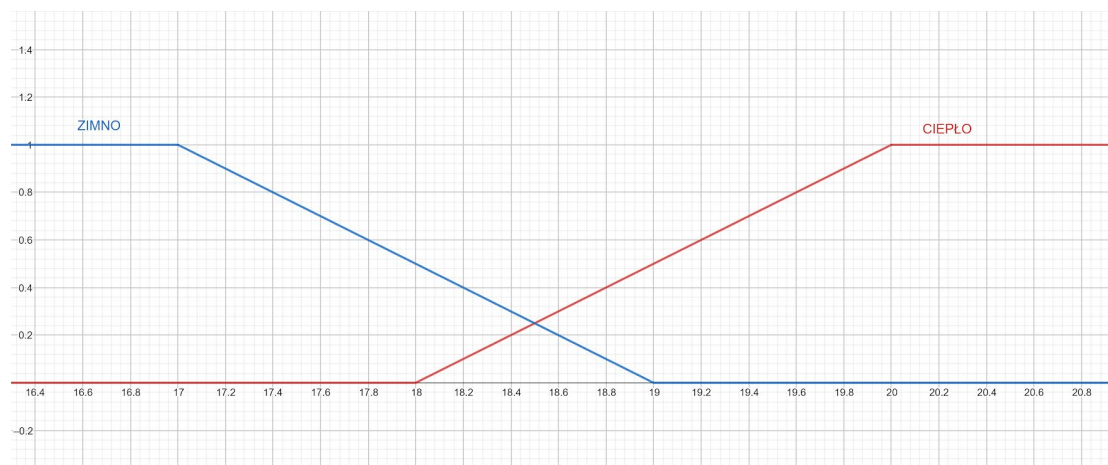
- Funkcja $T: [0, 1] \times [0, 1] \rightarrow [0, 1]$, spełniająca:
 - przemienność: $T(a, b) = T(b, a)$
 - łączność: $T(a, T(b, c)) = T(T(a, b), c)$
 - monotoniczność: $T(a, b) \leq T(c, d)$, gdy $a \leq c$ oraz $b \leq d$
 - tożsamość jedyнки: $T(a, 1) = a$
- Przykłady:
 - $T_{\min}(a, b) = \min\{a, b\}$
 - $T_{\text{prod}}(a, b) = a \cdot b$
 - $T_{\text{Łuk}}(a, b) = \max\{0, a+b-1\}$
 - wiele innych
- Ostatnia wymyślona przez polskiego matematyka Łukasiewicza

Regulacja rozmyta

- Kroki tworzenia systemu regulacji rozmytej;
 - Rozmywanie sygnałów wejściowych i wyjściowych
 - Opracowanie bazy reguł
 - Wyostrzenie wyjścia regulatora

Rozmywanie

- Rozmywanie polega na przetworzeniu każdego sygnału (wejściowego i wyjściowego na funkcję przynależności)
- Np.



Baza reguł

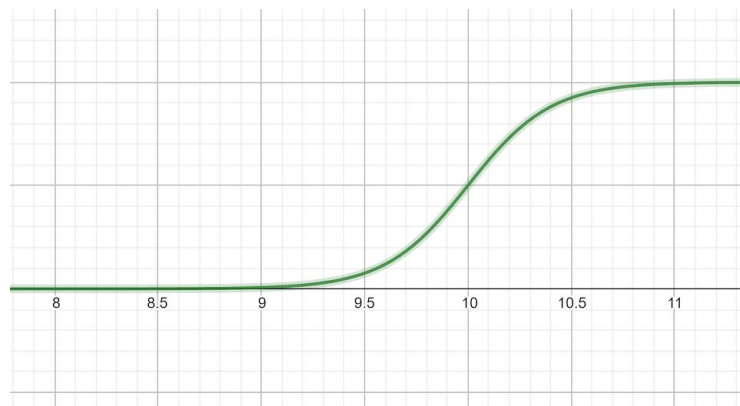
- Opracowanie reguł przy pomocy zmiennych lingwistycznych (czyli tych, które opisują zbiory rozmyte np. wysoki, niski, ciepły, zimny)
- Reguły oddają sposób myślenia człowieka
- Na przykład reguła dla sterowania klimatyzacją (wiatrakiem) w pokoju:
 - **Jeśli** jest gorąco i bardzo wilgotno **ustaw** moc wiatraka na bardzo wysoką
- Na podstawie każdej reguły (na podstawie T-norm, itd.) jest obliczana wartość danego wyjścia

Baza reguł - przykład

- Oznaczenia:
 - $\mu_G(x)$ - oznacza funkcję przynależności dla zbioru gorąco i $\mu_G(25) = 0.9$
 - $\mu_{BW}(x)$ - oznacza funkcję przynależności dla zbioru bardzo wilgotno i $\mu_{BW}(70\%) = 0.8$
 - reguła z poprzedniej strony
 - chcemy obliczyć μ_{DM1} - stopień w jakim moc wiatraka ma być bardzo wysoka
 - Jako T-normę bierzemy np. $T_{\min}(a, b)$
- Wtedy $\mu_{DM1} = \mu_G(25) \wedge \mu_{BW}(70\%) = \min\{\mu_G(25), \mu_{BW}(70\%)\} = \min(0.9, 0.8) = 0.8$

Wyostrezanie wyjścia

- Dla każdego sygnału wyjściowego, jego funkcja przynależności jest ucinana do poziomu jaki wyszedł z bazy reguł (jeśli dany sygnał wyjściowy jest w kilku regułach to jest brana wartość “**lub**” tych wyjść)
- Np. z pierwszej reguły $\mu_{DM1} = 0.7$, a z drugiej $\mu_{DM2} = 0.5$, a funkcja przynależności wygląda tak:



Wyostrowanie wyjścia

- Wtedy najpierw obliczamy $\mu_{DM3} = \mu_{DM1} \vee \mu_{DM2} = \max\{0.7, 0.5\} = 0.7$ (to mogłaby być też inna funkcja lub)
- A potem wyznaczamy uciętą funkcję $\mu_{DM}(x) = \min\{\mu_{DM3}, \mu_{DM}(x)\}$, więc funkcja wygląda następująco:



Wyostrzanie wyjścia

- Wszystkie krzywe z wyjścia (dla jego wszystkich zbiorów rozmytych) są łączone przy pomocy funkcji **lub**
- Znajdowany jest środek ciężkości tak otrzymanej figury (zamiast środka ciężkości powierzchni pod wykresem można brać środek ciężkości samego wykresu)
- Środek ciężkości jest rzutowany na oś poziomą i jest odczytywana liczbowa wartość wyjścia

Uwagi!

- Na kilku etapach obliczana jest wartość jakiegoś wyrażenia logicznego (na etapie bazy reguł, na etapie gdy dany sygnał wyjściowy występuje w kilku regułach i pod koniec na etapie łączenia wszystkich krzywych danego wyjścia)
- Za każdym razem musi być ona wyznaczona przy pomocy jakiejś T-normy (bo negacja zawsze jest tak samo, a **lub** wynika z wyboru T-normy)
- W obrębie danego etapu musi być to zawsze ta sama T-norma
- Ale nie musi to być ta sama T-norma w różnych etapach

Źródła

- Znowu na angielskiej wikipedii większość jest opisana bardzo dobrze (na polskiej jest dość ubogo):
 - https://en.wikipedia.org/wiki/Fuzzy_logic#Process
 - https://en.wikipedia.org/wiki/Fuzzy_set
 - https://en.wikipedia.org/wiki/Fuzzy_control_system
 - <https://en.wikipedia.org/wiki/T-norm>
 - <https://en.wikipedia.org/wiki/Defuzzification>
- Znalazłem też takie prezentacje dotyczące tematu:
 - <https://www.mimuw.edu.pl/~szczuka/ls/Lecture5.pdf>
 - https://web.archive.org/web/20140327122410/http://zsi.tech.us.edu.pl/~nowak/psiise/psiise_3.pdf
- Dość adekwatny filmik:
 - <https://www.youtube.com/watch?v=kxc-iA8jsso>

Treść zadania

- Zaimplementować regulator rozmyty dla wahadła odwróconego na wózku, na przygotowanym przeze mnie szablonie
- Regulator wykorzystuje informacje o położeniu i prędkości wózka oraz o położeniu kątowym i prędkości kątowej wahadła (x , dx , θ , $d\theta$), a na wyjściu daje siłę przykładaną do wózka
- Zero osi X jest w połowie i rośnie w prawo
- θ wynosi zero, jak wahadło jest w pionie i rośnie przeciwnie do ruchu wskazówek zegara
- Trzy etapy regulacji mają być wyraźnie oddzielone w programie
- Funkcje negacji, koniunkcji i alternatywy mają być osobnymi funkcjami

Treść zadania

- Regulator ma utrzymywać wahadło w pionie, a wózek możliwie blisko współrzędnej zerowej
- Na wózek będą działały różne zakłócenia (przekazywane przez ciąg liczb w pliku, ale część z wczytywaniem pliku jest już zaimplementowana)
- Uznaje się, że regulator zadziałał, jeśli dla wszystkich przygotowanych przeze mnie plików utrzymuje wahadło w pionie, a wózek przy zerze przez czas “nieskończony”
- Uznaje się, że regulator nie zadziałał, jeśli wózek odjechał za bardzo w prawo lub w lewo lub wahadło za bardzo opadło

Treść zadania - co wolno, a czego nie

- **Nie wolno** korzystać z:
 - bibliotek do obsługi zbiorów rozmytych np. fuzzy
 - gotowych fragmentów kodu z internetu
 - gotowych fragmentów kodu innych osób
- **Wolno** korzystać z:
 - swojej głowy
 - materiałów przygotowanych przeze mnie
 - materiałów podanych w źródłach
 - internetu oczywiście też, ale w rozsądny i uczciwy sposób
 - pseudokodu
 - struktur takich jak map (w C++), dict (w Pythonie)
- **W razie wątpliwości, czy coś wolno, czy nie, proszę od razu pytać!**

Treść zadania - termin i punktacja

- Termin oddania zadania to 22.12.2024
- Jeśli ktoś odda po tym terminie, ale odda do 29.12.2024 to dostaje 75% punktów, które by dostał normalnie
- Po powyższych terminach automatyczne zero
- Punktacja:
 - 0 pkt - nie działa i zupełnie nie ma sensu lub oddane po terminie
 - 5 pkt - regulator utrzymuje wahadło w pionie, ale nic nie robi z prędkością i położeniem wózka
 - 10 pkt - regulator utrzymuje wahadło w pionie i dojeżdża do zera i się tam zatrzymuje (może być przeregulowanie, ale ostatecznie ma się zatrzymać pionowo w zerze)
 - Jak są jakieś drobne błędy to obcinam jakieś drobne punkty

Treść zadania - uwagi

- Cotygodniowe postępy proszę mi przesyłać przez czat na Teamsach
- Ostateczną wersję proszę przesłać przez Moodla przez zadanie, które będzie tam utworzone
- Trzeba wysłać sam zmodyfikowany szablon
- Nie trzeba robić żadnego sprawozdania
- Proszę pisać możliwie dużo sensownych komentarzy w kodzie
- Jeśli kod będzie czytelny i zrozumiały to po prostu będę wstawiał punkty, a jeśli nie to będę z takimi osobami umawiał się na omawianie jego kodu

Treść zadania - najważniejsze!

- Zadanie trzeba wykonać **SAMODZIELNIE!!!**
- W przypadku pracy niesamodzielnej dostaje się nie tylko zero z obecnego projektu, ale też automatycznie zeruje się całe ćwiczenia i nie zalicza się przedmiotu

Treść zadania - dodatkowe punkty

- Można dostać do 2 bonusowych punktów za wygenerowanie jakiś ciekawych wykresów związanych z tym zadaniem
- Niekoniecznie jeden wykres to jeden punkt, być może więcej (to zależy od jego stopnia bycia ciekawym)
- Jako, że to punkty bonusowe to będę subiektywnie oceniał, czy wykresy są ciekawe, czy nie