mgr inż. Marcin Mrukowicz

Systemy rozmyte

1. Informacja nieprecyzyjna i niepewna

W życiu codziennym często posługujemy się wyrażeniami typu: niedaleko, daleko, duży, mały, zimny, gorący. Słowa te mają z jednej strony konkretne znaczenie, jednakże ciężko precyzyjnie określić, **co to właściwie oznacza być daleko, a co blisko.** Jeżeli popatrzymy z punktu widzenia pieszego, to blisko będzie oznaczać raczej odległość kilkudziesięciu metrów (chociaż ktoś może uważać, że jest to odległość większa, powiedzmy do 1-2 km, którą można bez problemu pokonać piechotą), natomiast z punktu widzenia posiadacza samochodu za bliską odległość możemy uznać do kilku-kilkunastu kilometrów, gdyż przemieszczanie się samochodem jest szybsze i jednocześnie mniej męczące. **Jak widać, ciężko w sposób jednoznaczny wyrazić tego typu pojęcia, również z uwagi na ich względność w zależności od sytuacji, czy nawet osobistych przekonań danej osoby.**

Tradycyjna matematyka skupiała się na rozważaniu bardzo ściśle ustalonych pojęć, które były bardzo precyzyjnie zdefiniowane. Jednakże ten aparat matematyczny okazał się niewystarczający do opisania informacji nieprecyzyjnej, czyli właśnie takiej, której nie można jednoznacznie opisać. **Informacja nieprecyzyjna odróżnia się od informacji niepewnej.** W przypadku informacji niepewnej mamy do czynienia z sytuacją, gdzie albo występuje jakiś brak w wiedzy i dane zjawisko posiada więcej niż jeden opis, ale tylko jeden z nich jest poprawny (np. nie potrafimy dokładnie zmierzyć jakiejś wielkości, pojawiają się niepewności pomiarowe i błędy pomiarowe, które powodują, że nie wiemy, ile konkretnie wynosi mierzona wielkość, ale wiemy, że zawiera się w jakimś przedziale, zatem mamy potencjalnie wiele odpowiedzi, z których jedna jest poprawna, ale nie wiemy która) albo istnieje sytuacja, że dane zjawisko posiada wiele opisów i każdy z nich jest poprawny, co za tym idzie nie można wybrać jednego z nich a pozostałe odrzucić (np. odpowiedź na pytanie ile waży butelka 1,5 l wody jest zależna od aktualnej temperatury otoczenia, gdyż gęstość wody zmienia się w zależności od temperatury; zwykle dla celów praktycznych podaje się gęstość wody dla jednej, ustalonej temperatury, jednak nie można powiedzieć tutaj o tym, że jedna z tych odpowiedzi jest poprawna, a pozostałe niepoprawne).

Informacja precyzyjna	**Informacja nieprecyzyjna**
Jan ma 1,92 metrów wzrostu.	Jan jest wysoki.
Do najbliższego sklepu spożywczego jest 67,4 metra.	Do najbliższego sklepu spożywczego jest blisko.
Do ciasta proszę dodać 2 gramy soli.	Do ciasta proszę dodać odrobinę soli.
Sieć autostrad w Chinach liczy 161 000 km.	Chiny posiadają bardzo dużą sieć autostrad.

Tabela pokazuje różnice pomiędzy wyrażaniem informacji w sposób precyzyjny i nieprecyzyjny

Informacja niepewna na przykładzie: wyobraźmy sobie że mamy w domu dwa zegary, które wskazują różny czas (ale zbliżony, jak na zdjęciach obok); ustalenie jaka naprawdę jest godzina, bez skorzystania z jakiegoś zewnętrznego źródła wiedzy (np. odczytanie czasu z Internetu, telewizji, itp.), jest właściwie niemożliwe.

Gdyby nie było możliwości skorzystania z tej dodatkowej wiedzy, to właściwie problem ten pozostałby niemożliwy do rozstrzygnięcia (pomijamy tutaj przypadek, w którym zegary wskazywałyby inną godzinę niż wynika z pory dnia i można by uznać, że ich wskazania są definitywnie fałszywe). Nie wiemy, czy różnica we wskazaniu wynika np. z uszkodzenia jednego z zegarów i niepoprawnego odmierzania przez niego czasu? Czy może jeden z zegarów nastawiono niepoprawnie? Może jeszcze jakiś inny czynnik wpłynął na wskazania zegarów? Na te pytania nie ma odpowiedzi i dlatego informacja przez nie przekazywana jest niepewna.

Ilustracja pokazuje obrazowo, że ta sama temperatura otoczenia, może być przez różnych ludzi odczuwana jako zupełnie inna. Gdyby zapytać mieszkańców jednej strefy klimatycznej o określenie tego, jak określą temperaturę otoczenia równą 20 st C, to otrzymamy zapewne różne odpowiedzi, choć będą one w jakiś sposób bardziej zbliżone, niż gdyby porównywać je z odpowiedziami mieszkańców innego strefy klimatycznej. Stąd określenia temperatury są dodatkowo nieprecyzyjne z uwagi na różnice kontekstów: w pustynnym Meksyku temperatura 20 st C jest obiektywnie niewielka, natomiast w realiach rosyjskiej Syberii ta sama temperatura będzie już uchodzić za upalną.

Czasami wykorzystuje się również zamianę informacji precyzyjnej, uważanej za zbyt szczegółową, w informację nieprecyzyjną, która okazuje się łatwiejsza do przyswojenia. Przykładowo wyobraźmy sobie, że chcemy dowiedzieć się, jakie miasta występują w Kanadzie. Można przedstawić tę informację w postaci szczegółowej tabeli (jak poniżej, ale zakładając, że wymienimy w ten sposób **wszystkie miasta**).

L.p.	Miasto	Prowincja	Liczba ludności
1.	Toronto	Ontario	2 731 579
2.	Montreal	Quebec	1 704 694
3.	Calgary	Alberta	1 239 220
4.	Ottawa	Ontario	934 243
5.	Edmonton	Alberta	932 546
6.	Mississauga	Ontario	721 599
7.	Winnipeg	Manitoba	705 224
8.	Vancouver	Kolumbia Brytyjska	631 486
9.	Brampton	Ontario	593 638
10.	Hamilton	Ontario	536 917
11.	Québec	Quebec	531 902
12.	Surrey	Kolumbia Brytyjska	517 887

Zamiast tego, możemy napisać krótkie podsumowanie. **Kanada posiada 310 miast przekraczających 7 tysięcy mieszkańców. Trzy miasta: Toronto, Montreal i Calgary, przekraczają liczbę 1 mln mieszkańców. Stolica kraju Ottawa posiada około 900 tysięcy mieszkańców. Miast przekraczających 100 tysięcy mieszkańców a poniżej jednego miliona mieszkańców jest 94. Małych miast, liczących pomiędzy 25 tysięcy a 50 tysięcy mieszkańców jest 63, pozostałe miasta nie przekraczają limitu 25 tysięcy mieszkańców.** Jak widać, w ten sposób zatracamy wiedzę o wielu szczegółach: nazwach miast, dokładnej liczbie mieszkańców i ich położeniu w poszczególnych prowincjach, jednak uzyskujemy „skondensowaną" wiedzę, z której wynika, że Kanada posiada tylko kilka metropolii i relatywnie mało dużych ośrodków miejskich, dominują natomiast miasta małe. Przykład ten jest obrazowy i oczywiście będą występować sytuację, w której ktoś chciałby dokładniejszego opisu sytuacji, ale wydaje się bezdyskusyjne, że podsumowanie zawierające kilka zdań, ogólnie (niekoniecznie bardzo precyzyjnie) opisujące stan faktyczny, jest niejednokrotnie bardziej praktyczne, niż bardzo szczegółowe opisy, które trzeba dopiero analizować, żeby wyciągnąć jakieś ogólne wnioski.

2. Logika rozmyta

Logika rozmyta jest rozwinięciem logiki klasycznej (binarnej, dwuwartościowej). Logika klasyczna była rozwijana od czasów starożytnych i początkowo zajmowała się przede wszystkim problemem właściwego rozumowania (wnioskowania), tj. uzasadnieniem że z pewnych zdań, nazywanych przesłankami wynika pewne zdanie: wniosek (konkluzja) i jest ono prawdziwe. Dopiero z czasem logika matematyczna wykształciła powszechnie stosowany język formalny, służący do przeprowadzania precyzyjnych rozumowań, powszechnie nazywany rachunkiem zdań. Dalsze badania doprowadziły do powstania nowych dziedzin logiki takich jak teoria dowodu, czy teoria modeli.

Logika klasyczna zakładała, że dane zdanie może być jedynie prawdziwe albo fałszywe (zasada wyłączonego środka). Z czasem zaczęto rozważać różne rozwinięcia logiki klasycznej, takie jak logika temporalna, logika modalna, logika trójwartościowa, które zaczęły podważać w jakiś sposób tę zasadę.

Logika rozmyta (ang. fuzzy logic) według twórcy tego pojęcia Loftiego Zadeha to „precyzyjna logika nieprecyzyjnego i przybliżonego rozumowania." Logika rozmyta znajduje zastosowanie w przetwarzaniu informacji, która jest nieprecyzyjna, niepewna, niekompletna, jednym słowem niedoskonała. Zadeh stwierdza także, że „siłą logiki rozmytej jest duża zdolność do precyzowania pojęć nieprecyzyjnych".

Podstawowym pojęciem w logice rozmytej jest zbiór rozmyty (ang. fuzzy set). Tradycyjnie rozumiane zbiory posiadały elementy, co do których możemy powiedzieć, że należą, albo nie do tego zbioru. Zatem relacja należenia do zbioru jest relacją binarną. Każdy zbiór można przedstawić za pomocą jego funkcji charakterystycznej.

Niech A będzie dowolnym zbiorem, zaś B jego podzbiorem, $B \subseteq A$.

Funkcją charakterystyczną zbioru B nazywamy funkcję rzeczywistą $f\colon A \longrightarrow \{0,1\}$ określoną następującym wzorem:

$$f(x) := \begin{cases} 1, & \text{gdy } x \in B, \\ 0, & \text{gdy } x \notin B. \end{cases}$$

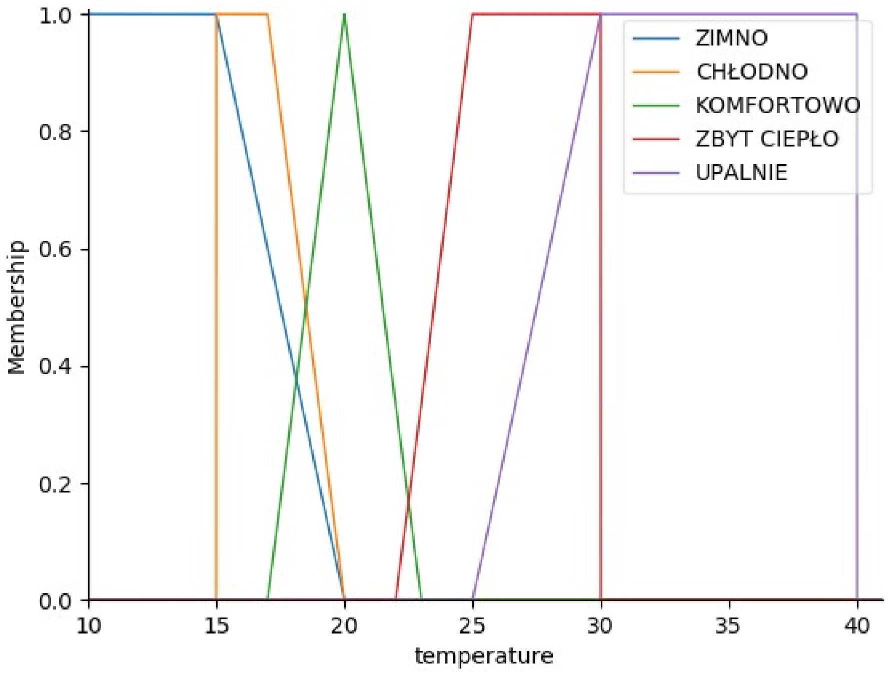
Oznaczeniem funkcji charakterystycznej zbioru $B \subseteq A$ jest $\mathbf{1}_{B,A},\ \chi_{B,A},\ \mathbf{1}_B$, bądź χ_B.

Widać, że relację należenia/nienależenia do zbioru można zamodelować jako 0 i 1, gdzie 0 oznacza nienależenie do zbioru, natomiast 1 oznacza należenie do zbioru.

Lofti Zadeh postanowił uogólnić tę relację w ten sposób, że dopuścił pojęcie „częściowego należenia do zbioru". W ten sposób o danym elemencie możemy powiedzieć, że należy do zbioru np. w stopniu 0,7. Bardziej formalnie dla dowolnego, niepustego klasycznego zbioru X, można zdefiniować zbiór rozmyty $A \subseteq X$ jako zbiór par $A = \{(x, \mu_A(x)); x \in X\}$, gdzie $\mu_A : X \rightarrow [0,1]$.

μ_A nazywamy funkcją przynależności zbioru rozmytego. **Oznacza to mniej formalnie, że elementom pewnego tradycyjnego zbioru (zbioru ostrego, ang. crisp set) przypisano wartość przynależenia do zbioru rozmytego, która zawiera się w przedziale [0, 1]. Zbiór X nazywamy często uniwersum. Warto zaznaczyć, że w jednym uniwersum, można zdefiniować wiele zbiorów rozmytych.**

W zastosowaniach praktycznych logiki rozmytej wprowadza się pojęcie **zmiennej lingwistycznej**, która nieformalnie będzie dla nas oznaczać zmienną, której można przypisać określenie w języku naturalnym, takie jak: wysoki, niski, małe, średnie, wysokie, nazywane **wartością lingwistyczną**, które posiada pewne znaczenie nieprecyzyjne, zależne od kontekstu. Wartość lingwistyczna może być zamodelowana jako zbiór rozmyty, określony w uniwersum, które jest tutaj zbiorem wartości, jakie są sensowne dla danej wielkości. Jeśli przykładowo chciałbym zamodelować zmienną lingwistyczną **temperatura pokojowa**, to zakładam, że temperatura w tym kontekście wynosi minimalnie 10 stopni Celsjusza, natomiast maksymalnie 40 stopni Celsjusza. Zatem moje uniwersum będzie przedziałem w zakresie [10, 40]. Mogę teraz określić arbitralnie takie wartości lingwistyczne: {ZIMNO, CHŁODNO, KOMFORTOWO, ZBYT CIEPŁO, UPALNIE} i następnie zamodelować je jako zbiory rozmyte:

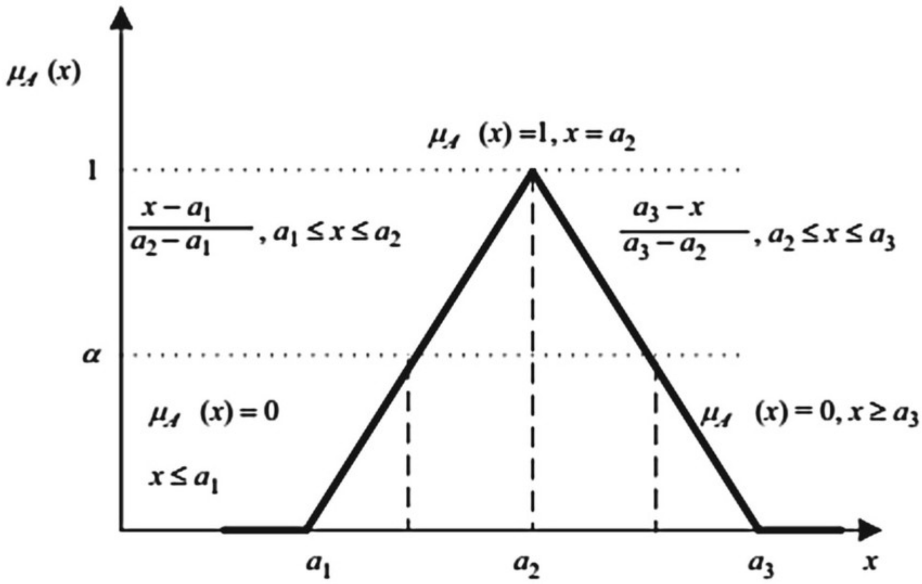


Należy zwrócić uwagę, że każda wartość lingwistyczna jest zbiorem rozmytym i na osi y posiada wartość funkcji przynależności do zbioru rozmytego, zaś na osi x elementy z uniwersum.

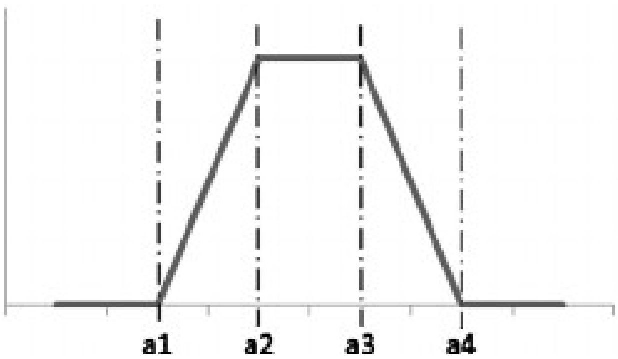
Tak zdefiniowane zbiory rozmyte modelują informację nieprecyzyjną. Warto zwrócić uwagę, że ustalone tutaj funkcje przynależności nie są bezsprzeczne i różne osoby, mogą w różny sposób zamodelować powyższe pojęcia. Jednakże bezsprzeczny pozostaje fakt, że pewne temperatury są bardziej i mniej pasujące do danej wartości lingwistycznej i metoda modelowania zaproponowana przez zbiory rozmyte pozwala wyrazić więcej niż gdyby użyć zwykłych zbiorów.

Funkcją przynależności może być dowolna funkcja, której przeciwdziedzina mieści się pomiędzy 0 a 1, natomiast kilka funkcji przynależności jest najbardziej popularnych i najczęściej stosowanych. Należą do nich m.in.:

1. funkcja trójkątna (triangularna):



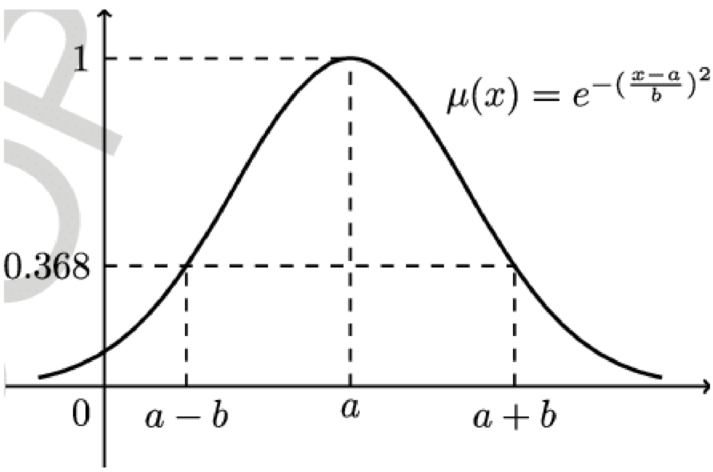
2. funkcja trapezoidalna



$$\mathrm{trapezoid}(x; a, b, c, d) = \begin{cases} 0, & x \le a. \\ \frac{x-a}{b-a}, & a \le x \le b. \\ 1, & b \le x \le c. \\ \frac{d-x}{d-c}, & c \le x \le d. \\ 0, & d \le x. \end{cases}$$

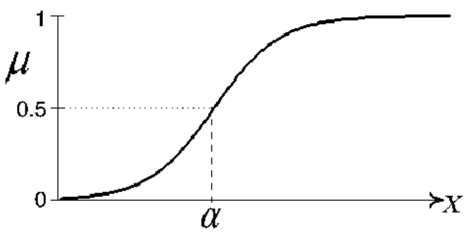
With $a1 \le x \le a2$ and $a3 \le x \le a4$

3. funkcja gaussowska



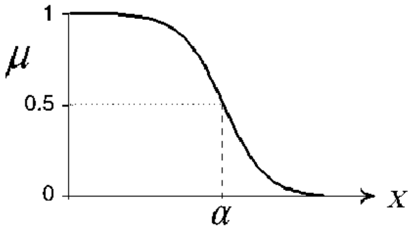
4. funkcja sigmoidalna

Right shoulder sigmoidal function



$$\mu = \frac{1}{1 + e^{-\beta(x - \alpha)}}$$

Left shoulder sigmoidal function



$$\mu = \frac{1}{1 + e^{\beta(x - \alpha)}}$$

Literatura przedmiotu wyróżnia jeszcze kilka innych klasycznych funkcji przynależności, jednak nie będą one szczegółowo omówione w tym miejscu.

3. Wnioskowanie rozmyte

Klasyczny modus ponens

Logika klasyczna wprowadza pojęcie schematu (reguł) wnioskowania, gdzie o prawdziwości pewnych zdań wnioskujemy na podstawie prawdziwości innych zdań (to znaczy wiemy że te zdania są prawdziwe). Zwykle stosujemy specjalną notację: zdania na podstawie których wnioskujemy (przesłanki) zapisujemy przed kreską poziomą, natomiast wniosek zapisujemy poniżej kreski.

$$\begin{array}{l} A_1 \\ A_2 \\ \ldots \\ A_k \\ \hline B \end{array}$$

lub

$$\frac{A_1, A_2, \ldots, A_k}{B}$$

Notacje stosowanie w schematach wnioskowania.

Jednym z najbardziej znanych schematów wnioskowania jest **modus ponens:**

p
p → q

---

q

Powyższy zapis należy odczytać tak: wiem, że zdanie **p** jest prawdziwe i wiem, że zachodzi implikacja z **p** do **q, z czego wynika że zdanie q jest prawdziwe**. Mając wiedzę na temat zdania p i tego że zachodzi implikacja z p do q, możemy odczytać z definicji implikacji, że jedyny przypadek w którym **p ma wartość logiczną prawda** i **jednocześnie implikacja z p do q jest prawdziwa**, to taki, w którym jednocześnie **q jest prawdziwe. Można zatem interpretować tę regułę wnioskowania w ten sposób, że jeżeli ustali się, że pewne zdanie p jest prawdziwe i prawdziwe jest, że to zdanie implikuje zdanie q, to możemy wyciągnąć wniosek, że zdanie q jest prawdziwe. Przykładowo ze zdania p: Jan jest kierowcą i implikacji Jeżeli Jan jest kierowcą to posiada prawo jazdy, wynika że Jan posiada prawo jazdy. Zasadę te można udowodnić korzystając z tablicy prawdy (widać wyraźnie że jeżeli implikacja jest prawdziwa i p jest prawdziwe to q też musi być prawdziwe).**

p	q	$p \Rightarrow q$
1	1	1
1	0	0

Rozmyty modus ponens

Logika rozmyta pozwala uogólnić klasyczną regułę wnioskowania modus ponens, w ten sposób:

p jest A'
p jest A' → q jest B

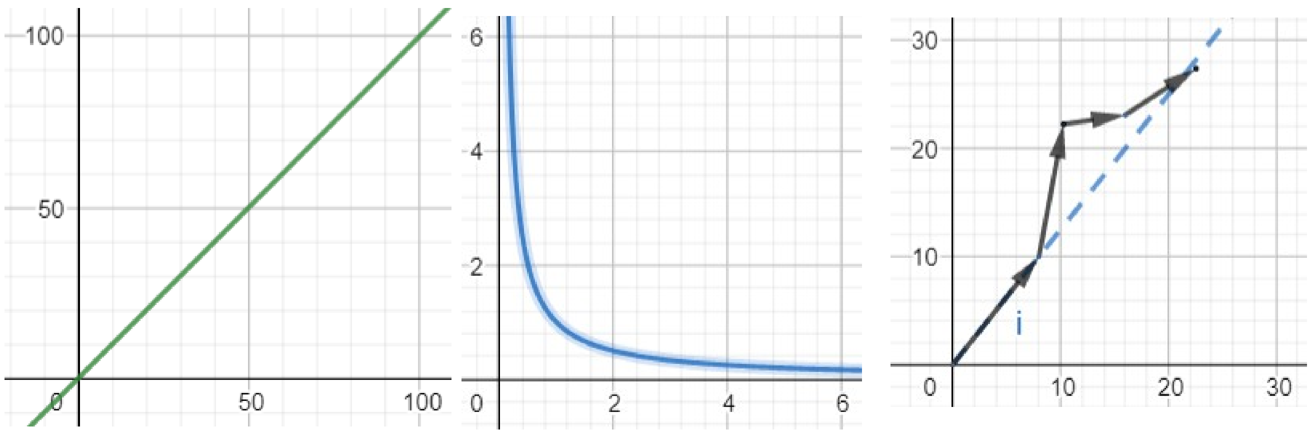
---

q jest B'

gdzie A,A',B,B' są **zbiorami rozmytymi**, natomiast **p, q** są **zmiennymi lingwistycznymi.** Warto zaznaczyć, że występuje tutaj **implikacja rozmyta, a nie klasyczna, gdyż p i q nie są już zdaniami logicznymi w znaczeniu klasycznym, tylko są to rozmyte zdania logiczne. Rozmyte zdanie logiczne to zdanie w którym elementowi uniwersum, w którym zdefiniowano zbiór rozmyty przypisujemy wartość funkcji przynależności do tego zbioru rozmytego.** Załóżmy że x = 10 stopni Celsjusza jest elementem uniwersum X = [0, 40], przynależy do wartości zmiennej lingwistycznej ZIMNO (która jest zbiorem rozmytym) w stopniu 0,4. Zatem zdaniu x jest ZIMNO mogę przypisać stopień prawdziwości 0,4.
Zbiór B' jest obliczany jako złożenie zbioru A' i rozmytej implikacji (A → B). Mówiąc obrazowo, istnieje tutaj zdolność do uogólniania wiedzy. Przykładowo wiem, że zachodzi taki fakt: Jeżeli prędkość samochodu jest bardzo duża, to poziom emitowanego przez pojazd hałasu jest wysoki, to mogę wysnuć wniosek, że jeżeli prędkość samochodu jest duża, to poziom emitowanego przez pojazd hałasu jest średni. Przy czym warto zaznaczyć, że nigdzie nie zdefiniowano co to znaczy średni; wartość ta zostanie w pewien sposób obliczona, na podstawie tego, że wiemy co to jest wysoki.

Relacja	Przesłanka x jest A'	Wniosek y jest B'
1	x jest A	y jest B
2a	x jest „bardzo A”	y jest „bardzo B”
2b	x jest „bardzo A”	y jest B
3a	x jest „mniej więcej A”	y jest „mniej więcej B”
3b	x jest „mniej więcej A”	y jest B
4a	x jest „nie A”	y jest nieokreślone
4b	x jest „nie A”	y jest „nie B”

W tabeli powyżej przedstawiono możliwe relacje pomiędzy A i B; jak widać są one znacznie bardziej rozbudowane niż gdyby to było możliwe w przypadku logiki klasycznej. **Ogólnie chodzi tutaj o fakt, że wiedza o jakimś konkretnym x nie musi uogólniać się w nieskończoność.** Tylko **w przypadku trywialnych zależności, takich jak proporcjonalność prosta albo odwrotna, jest możliwe wnioskowanie przez analogię w nieskończoność.** Przykładowo dla wielkości wprost proporcjonalnych, jeżeli wiem, że małe x implikuje małe y, to mogę stąd zakładać, że średnie x implikuje średnie y i dalej że duże x implikuje duże y. Bardzo często w zastosowaniach praktycznych zależność między x a y jest bardziej złożona i tego typu wnioskowanie nie jest możliwe. Jednak wciąż rozmyty modus ponens jest bardziej ogólny niż klasyczny, gdzie należałoby podać implikacje dla każdej możliwej pary (x, y); rozmyty modus ponens pozwala zatem w pewnym sensie na lokalną aproksymację[1] wartości y na podstawie x.



Na rysunkach przedstawiono od lewej zależność wprost proporcjonalną, odwrotnie proporcjonalną i zależność bardziej złożoną, dla której zachodzą dość różnorodne wzorce i np. wiedza na temat relacji małych x i y, jest nieadekwatna do relacji dużych x do y. Jednakże można za pomocą kilku implikacji rozmytych wyrazić przedstawioną tutaj relację, natomiast klasyczny modus ponens, jak już wspomniano, wymagałby podania tych zależności dla każdej możliwej pary (x, y).

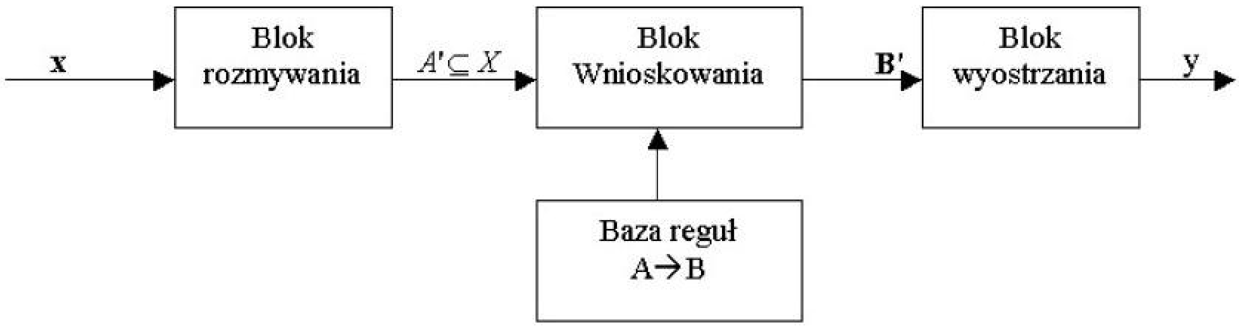
Niekiedy występują też sytuację, że istnieje zależność między x a y, ale jest ona nieproporcjonalna, czyli niewielkie wzrosty na x, powodują duże wzrosty na y, bądź bardzo duże wzrosty x powodują niewielkie wzrosty na y. Zatem charakter tego związku łatwo jest wyrazić za pomocą określeń typu jeżeli x jest bardzo A to y jest B.

4. Sterowanie rozmyte

1 Aproksymacją funkcji nazywamy proces jej przybliżenia za pomocą innej funkcji. Zwykle oryginalna funkcja jest albo wprost nieznana (znamy jedynie jej wartości dla wybranych x), bądź posiada bardzo skomplikowany wzór i chcemy znaleźć inną funkcję, zwykle dużo łatwiejszą do obliczania, która posiada zbliżone wartości na osi y.

Przedstawione we wcześniejszym punkcie metody wnioskowania rozmytego można stosować do dowolnego typu rozumowania, jednak najpopularniejsze jest stosowania wnioskowania rozmytego w **sterowaniu**. **W problemie sterowania zwykle chcemy na podstawie zmiennych wejściowych uzyskiwać pożądane wartości zmiennych wyjściowych.** Klasyczne sterowanie wymaga zastosowania skomplikowanych modeli matematycznych. Sterowanie rozmyte pozwala uprościć zdecydowanie to zagadnienie.

W kolejnych akapitach zakładamy, że omawiany jest jeden z rodzajów wnioskowania rozmytego, to jest model wnioskowania Mamdaniego.



W problemie rozmytego sterowania traktujemy odczyty z czujników jako **przesłanki (zakładamy zatem, że odczyt z czujnika jest faktem, który jest dany i zakładamy jednocześnie prawdziwość tego faktu)**, przy czym zwykle musimy zamienić ostre wartości, jakie z nich otrzymujemy na zbiory rozmyte (zwykle czujniki podają wynik pomiaru jako liczbę a nie zbiór rozmyty). Proces ten nazywamy rozmywaniem (ang. fuzzyfication). Od strony teoretycznej możliwe jest również zaprojektowanie dedykowanych układów scalonych, które byłyby wysoko zoptymalizowane do wykonywania tego zadania (np. istnieją dedykowane układy scalone do uczenia sieci neuronowych).

Następnie definiuje się **bazę wiedzy**, innymi słowy **zbiór rozmytych reguł sterowania (odpowiednik implikacji). W szeroko pojętej sztucznej inteligencji istnieje wiele różnych typów reguł. Do najważniejszych należą reguły decyzyjne, klasyfikacyjne i asocjacyjne. Drzewa decyzyjne można również sprowadzić do zbioru reguł klasyfikacyjnych. Warto zaznaczyć że istnieją dwa główne, zasadnicze typy reguł: reguły odkrywane z danych, zatem powstałe w wyniku procesu uczenia oraz reguły sformułowane przecz człowieka (ludzką inteligencję), które są traktowane jako prawdziwe. Reguły zasadniczo są reprezentacją pewnej wiedzy,** przy czym w przypadku reguł odkrywanych z danych, zakładamy, że dane są również reprezentacją tej wiedzy, natomiast w przypadku reguł sformułowanych przez człowieka, zakłada się, że człowiek ten wyraził za ich pomocą wiedzę, którą sam posiadał. Klasycznym przykładem wykorzystywania reguł zdefiniowanych przez człowieka były historycznie (i są nadal) tzw. systemy ekspertowe (eksperckie). Systemy te są wykorzystywane w celu wspomagania podejmowania decyzji, względnie (w skrajnych przypadkach) podejmowania decyzji. Zwykle zakłada się, że ekspert (eksperci), posiadający rozległą wiedzę dziedzinową, wyrażają ją w postaci reguł, jednocześnie zakłada się, że wiedza ta jest pewna, gdyż jest sformułowana przez niezwykle doświadczone i wykształcone osoby. System ekspertowy zwykle wyposażony jest również w silnik wnioskowania (ang. inference engine), który pozwala na podstawie bazy reguł i podawanych mu faktów przeprowadzić wnioskowanie, które doprowadzi zwykle do zasugerowania jakiejś decyzji (np. w medycynie wspomaganie diagnozy pacjenta). Typowo polega to na tym, że wprowadza się dane wejściowe i system może zadawać użytkownikowi dodatkowe pytania (pytania te w zasadzie powstają na skutek przeprowadzenia wnioskowania), które pozwolą ostatecznie systemowi na zasugerowanie podjęcia jakiejś decyzji. **Innym popularnym zastosowaniem reguł sformułowanych przez człowieka jest tzw. programowanie w logice, gdzie również występuje pojęcie bazy wiedzy (faktów) i gdzie znając dodatkowo relację między obiektami można przeprowadzać automatyczne wnioskowania.**

<u>Warto zaznaczyć, że zwykle sama postać reguł nie różni się pomiędzy regułami odkrywanymi z danych a tymi zadanymi przez człowieka.</u>

Sterowanie rozmyte zasadniczo korzysta z reguł sformułowanych przez człowieka, na wzór reguł w systemach ekspertowych; choć warto zaznaczyć, że teoretycznie możliwe jest także rozwiązanie hybrydowe, gdzie takie reguły odkrywa się za pomocą uczenia maszynowego i traktuje na równi z takimi, jakie podaje ekspert.

Reguły są właściwie rodzajem implikacji, przy czym zwykle stosuje się specjalny zapis, jako tzw. **if then rules, „reguły jeżeli to", w postaci if <warunek> then <konkluzja>. Warunek zwykle nazywa się poprzednikiem reguły (ang. antecedent), natomiast konkluzję następnikiem (ang. consequent) reguły. Zarówno warunek, jak i konkluzja są pewną formułą logiczną, przy czym składają się z selektorów, tj. wyrażeń typu** $a_i=v_j, a_i \in \{v_{j1}, \ldots, v_{jk}\}, a_i < v_j, a_i \in [v_{j1}, v_{j2}]$. **Selektory łączone są spójnikami logicznymi and i or; możliwe jest również niekiedy stosowanie negacji.** Przykładowa reguła klasyfikacyjna może mieć postać: if *ból_głowy = tak* and *gorączka > 39* then *choroba = grypa*. Dla zwiększenia czytelności reguły, zdarza się, że zapisujemy słowa kluczowe z wielkich i/lub pogrubionych liter, tj. **IF** … **THEN** …. **Warto zwrócić uwagę, że wiedza wyrażona w postaci reguły jest zwykle bardzo łatwa do zrozumienia i zastosowania, jest to rodzaj tzw. modelu białoskrzynkowego (niektóre inne metody sztucznej inteligencji podejmują decyzję w sposób trudny do zrozumienia dla człowieka (nie wiemy dlaczego podjęto daną decyzję), są to tzw. modele czarnoskrzynkowe).**

Rozmyte reguły sterowania, będą miały specyficzną postać, gdyż będą operować na zbiorach rozmytych. Zatem przykładowa reguła może wyglądać następująco**: IF jakość_jedzenia is niska THEN napiwek is niski, przy czym określenia niska i niski są zbiorami rozmytymi, które muszą zostać zdefiniowane. Zwykle podczas procesu rozmywania definiujemy te zbiory i w momencie formułowania reguł sterowania wykorzystujemy już gotowe definicje.**

Możliwe jest zarówno posiadanie wielu wejść, które wpływają na jedno wyjście, jak i posiadanie jednego wejścia, które wpływa na wiele wyjść. Oczywiście w danym problemie sterowania może też być wiele wejść i wiele wyjść, wszystko zależy od rozpatrywanego problemu.

Zwykle powinno się zdefiniować reguły dla każdej możliwej kombinacji wartości zmiennych lingwistycznych wejść. Istnieją przypadki szczególne, gdzie nie jest to konieczne i akceptowalne jest, że sterownik w określonych okolicznościach nie podejmie żadnej akcji, jednak takie założenie albo musi być oparte o posiadanie rozległej wiedzy domenowej, albo musi powstać w wyniku eksperymentalnego ustalenia minimalnego zbioru reguł. Możliwe jest też najpierw zdefiniowane większej liczby reguł, po czym w wyniku ich analizy i względnie wykonywania symulacji działania, pomniejszenie liczby i/lub uproszczenie reguł z tego zbioru.

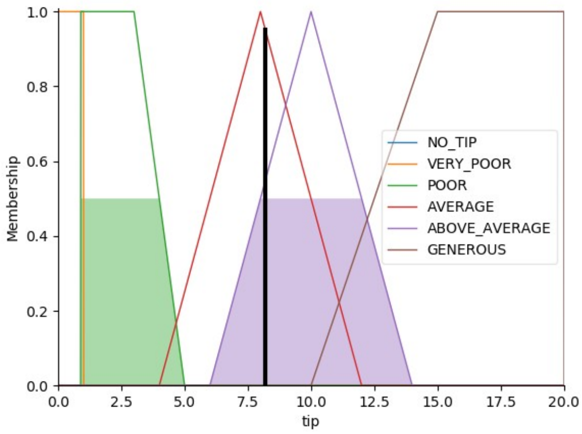
Z uwagi na fakt, że sterowanie rozmyte oparte jest o wnioskowanie rozmyte (m.in. **o rozmyty modus ponens**), **to nie ma potrzeby budowania dedykowanego silnika wnioskowania** (jak byłoby w przypadku systemu ekspertowego), zatem po zdefiniowaniu funkcji przynależności do zbiorów rozmytych dla wejść/wyjść i sformułowaniu reguł, sterownik rozmyty będzie w stanie pracować.

Dla każdego wejścia podczas fuzzyfikacji obliczamy jego stopień przynależności do każdej zdefiniowanej wartości zmiennej lingwistycznej, po czym badamy, czy nie jest spełniona któraś z reguł. Co ważne, sama reguła może być **spełniona w jakimś stopniu**, a nie binarnie (reguła jest spełniona albo nie). **Reguła jest spełniona (odpalona)**, **jeżeli stopień przynależenia do zbioru/ów rozmytych w poprzedniku reguły jest niezerowy. W przypadku spełnienia więcej niż jednej reguły następuje agregacja tych reguł: obliczony zostanie nowy zbiór rozmyty, będący w uproszczeniu złożeniem konkluzji (również zbiorów rozmytych) płynących z tych reguł. Złożenie to jest w istocie wartością zmiennej wyjściowej. O ile dla inteligencji ludzkiej, nieprecyzyjna informacja wyrażona przez zbiór rozmyty jest intuicyjnie zrozumiała i może pozostać w takiej postaci (np. że powinno się dać niski napiwek), to zwykle w sterowaniu trzeba wtórnie uzyskać wartość ostrą, to jest wykonać wyostrzanie (ang. defuzzyfikację) zmiennej wyjściowej (tj. zamienić pojęcie niski napiwek na konkretną kwotę w złotych).**

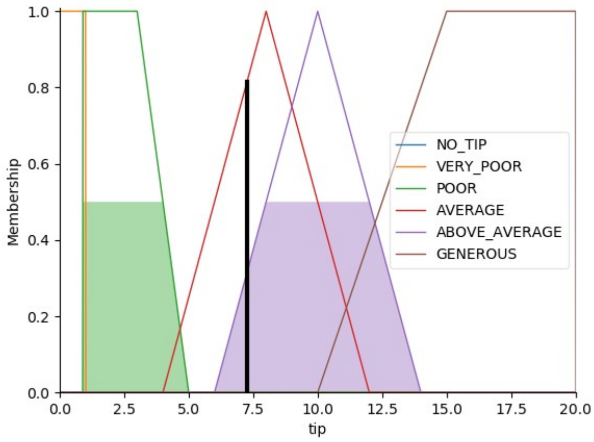
Warto przy tym zaznaczyć, że wynik agregacji reguł zależy od tego, jak zdefiniujemy rozmytą implikację i rozmyte spójniki logiczne.

Literatura podaje wiele metod wyostrzania zbiorów rozmytych. Dobór metody defuzzyfikacji zależy od konkretnego problemu i powinien być do niego dostosowany. Poniżej pokazano kilka najbardziej znanych metod defuzzyfikacji. Są to odpowiednio metoda bisektora, centroidu, smallest of maximum (som), mean of maximum (mom) i largest of maximum (lom). Jak widać (na wykresach na kolejnej stronie) wynik końcowy jest dla każdej z tych metod zupełnie różny.

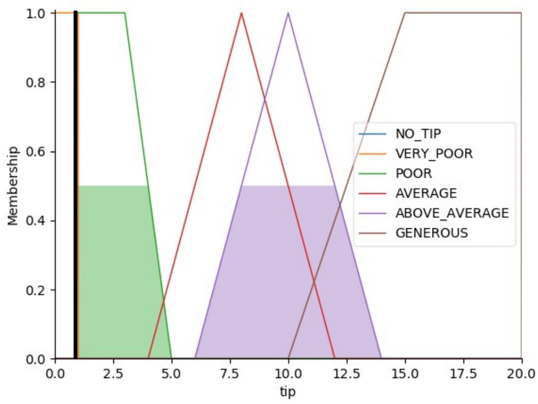
Na koniec proszę zwrócić uwagę na fakt, że jakość działania sterownika rozmytego jest zwykle możliwa do eksperymentalnego ustalenia, to znaczy można przekonać się, czy sterownik rozmyty pracuje poprawnie, czyli zgodnie z założonym wcześniej celem (innymi słowy poprawnie oblicza wyjście na podstawie wejścia). Niekiedy możliwe jest również wykonanie symulacji komputerowej, która może wykazać poprawność/niepoprawność sterownika bez konieczności wykonywania drogiego prototypu rzeczywistego automatu/maszyny/robota.



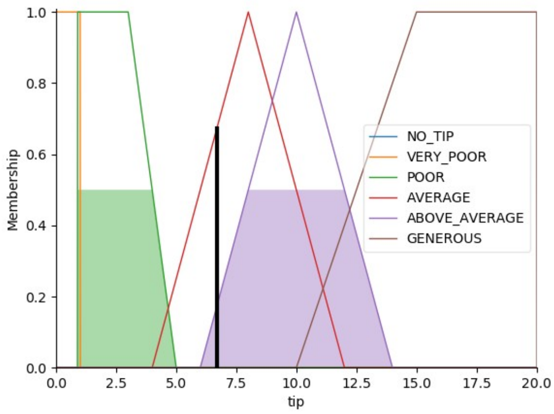
bisector



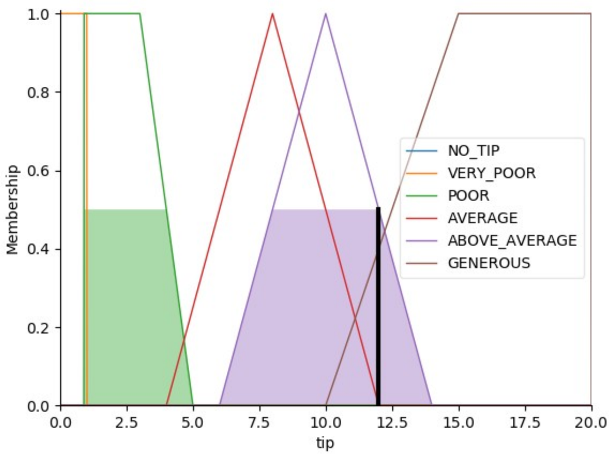
centroid



som



mom



lom

Materiały uzupełniające

1. Przykład – przeprowadzono pełny proces wnioskowania rozmytego za pomocą pakietu scikit-fuzzy (fuzzyfikacja, ustalenie reguł, agregacja reguł, defuzzyfikacja) i pokazano cały proces krok po kroku (w tym pokazano jak to zrobić w języku Python)
https://pythonhosted.org/scikit-fuzzy/auto_examples/plot_tipping_problem.html

2. Przykład – ten sam przykład w uproszczonym api w pakiecie scikit-fuzzy
https://pythonhosted.org/scikit-fuzzy/auto_examples/plot_tipping_problem_newapi.html

3. Jak można zmienić metody agregacji reguł i defuzzyfikacji w pakiecie scikit-fuzzy
https://github.com/scikit-fuzzy/scikit-fuzzy/issues/100

Zadania

1. W języku Python napisz implementację funkcji przynależności do zbioru rozmytego:
a) trójkątną
b) trapezową

Sprawdź poprawność funkcji, podstawiając wybrane wartości brzegowe (znajdujące się pomiędzy hiperparametrami); powinny one spełniać warunki zadane we wzorach tych funkcji (wartość funkcji przynależności zawsze musi mieścić się pomiędzy zerem i jedynką, w tym powinna ona odpowiednio rosnąć/maleć).

2. Przeanalizuj działanie sterownika rozmytego w pliku *tipper.py* i zapoznaj się z tym jak działa pakiet scikit-fuzzy. Przykład ten jest wariantem znanego w literaturze przedmiotu tzw. tipping problem, gdzie na podstawie oceny jakości obsługi w restauracji decydujemy, jaki pozostawimy napiwek dla obsługi. Zauważ, że w przykładzie występuje jedna zmienna wejściowa *jakość_obsługi* i jedna zmienna wyjściowa *wysokość_napiwku*. Dla zmiennych określono uniwersum (zbiór ostry), w jakim są zdefiniowane oraz utworzono wartości zmiennej lingwistycznej jako zbiory rozmyte, zdefiniowane w tym uniwersum. Wykorzystano jedynie funkcje trójkątne i trapezowe. Zdefiniowano również dość trywialne, oczywiste reguły. Na koniec zaprezentowano symulacje działania sterownika, gdzie na postawie konkretnego ostrego wejścia uzyskuje się konkretne ostre wyjście. Program rysuje także wykresy, na których graficznie zaprezentowano funkcje przynależności dla zmiennej wejściowej, jak i obliczone wyjście. Spróbuj podstawić różne oceny jakości obsługi i obserwuj zmianę wartości zmiennej wyjściowej.

3. Do sterownika zdefiniowanego w poprzednim zadaniu dodaj nową zmienną wejściową *jakość jedzenia*, która niech ma 3 wartości zmiennej lingwistycznej, następnie uaktualnij zbiór reguł, aby druga zmienna również wpływała na decyzję. W regułach wykorzystaj spójnik logiczny and (w pakiecie scikit-fuzzy oznaczany jako &). Spróbuj podstawić różne oceny jakości obsługi i jakości jedzenia i obserwuj zmianę wartości zmiennej wyjściowej. Zastanów się, czy uzyskany w ten sposób sterownik dobrze rozwiązuje problem i w razie konieczności zmodyfikuj podane zbiory rozmyte, i/lub reguły, tak, aby sterownik działał intuicyjnie poprawnie.

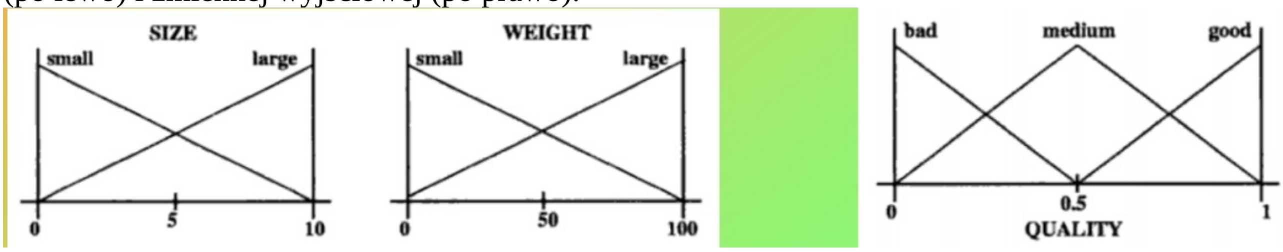
4. Wyobraź sobie, że tworzysz sterownik rozmyty dla autonomicznego pojazdu.
Załóżmy, że autonomiczny pojazd potrafi zmierzyć odległość od najbliższego pojazdu w metrach i jednocześnie jego hamulce są sterowane za pomocą poziomów od 0 (brak hamowania) do 6 (bardzo mocne hamowanie), przy czym poziomy są ciągłe (to znaczy możliwa jest dowolna wartość w przedziale [0, 6]). Za pomocą pakietu scikit-fuzzy zamodeluj zmienną lingwistyczną odległość od najbliższego pojazdu, która będzie zmienną wejściową i zmienną lingwistyczną hamowanie, która

będzie zmienną wyjściową; następnie zaproponuj rozmyte reguły wnioskowania, które będą na postawie odległości od najbliższego pojazdu proponować hamowanie w celu uniknięcia kolizji z tym pojazdem.

5. Rozwiń model w poprzedniego zadania o nową zmienną wejściową, jaką jest wilgotność nawierzchni drogowej (ponownie zakładamy, że pojazd autonomiczny potrafi ocenić za pomocą czujnika tę wielkość). Pamiętaj, że droga efektywnego hamownia zależy m.in. właśnie od poziomu wilgotności powierzchni.

6. Do modelu z poprzedniego zadania dodaj jeszcze zmienną lingwistyczną oblodzenie nawierzchni drogowej.

7. Utworzyć sterownik rozmyty, który ocenia jakość owoców na podstawie pomiaru ich wagi i określenia ich rozmiaru i posiada następujące funkcje przynależności do zmiennych wejściowych (po lewo) i zmiennej wyjściowej (po prawo).



Ponadto zdefiniowano następującą bazę reguł:

R_1 : *IF size is small and weight is small THEN quality is bad,*
also
R_2 : *IF size is small and weight is large THEN quality is medium,*
also
R_3 : *IF size is large and weight is small THEN quality is medium,*
also
R_4 : *IF size is large and weight is large THEN quality is good*