



POLSKA AKADEMIA NAUK
Instytut Badań Systemowych

**TECHNOLOGIE INFORMATYCZNE
W ZARZĄDZANIU
SYSTEMY
WSPOMAGANIA DECYZJI**

pod redakcją:
Jana Studzińskiego,
Ludostawa Drelichowskiego,
Olgierda Hryniewicza,
Janusza Kacprzyka



**TECHNOLOGIE INFORMATYCZNE W ZARZĄDZANIU
SYSTEMY WSPOMAGANIA DECYZJI**

Polska Akademia Nauk • Instytut Badań Systemowych

Seria: BADANIA SYSTEMOWE
tom 26

Redaktor naukowy:

Prof. dr hab. Jakub Gutenbaum

Warszawa 2000

**TECHNOLOGIE INFORMATYCZNE
W ZARZĄDZANIU
SYSTEMY WSPOMAGANIA DECYZJI**

pod redakcją

Jana Studzińskiego, Ludosława Drelichowskiego

Olgierda Hryniewicza i Janusza Kacprzyka

Książka zawiera wybór referatów przedstawionych na konferencji "Komputerowe systemy wielodostępne KSW'2000" w Ciechocinku w 2000 r. Konferencja pod patronatem Komitetu Badań Naukowych została zorganizowana przez Akademię Techniczno-Rolniczą w Bydgoszczy, Instytut Badań Systemowych PAN, Komisję Informatyki PAN - Oddział w Gdańsku oraz Bydgoskie Zakłady Elektromechaniczne "BELAM" S.A. w Bydgoszczy.

Komitet Naukowo-Programowy konferencji:

Witold Abramowicz, Ryszard Budziński, Ryszard Choraś, Ludosław Drelichowski (przewodniczący), Grzegorz Głownia, Adam Grzech, Jakub Gutenbaum, Olgierd Hryniewicz, Janusz Kacprzyk, Zbigniew Kierzkowski, Jerzy Kisielnicki, Adam Kopiński, Maciej Krawczak, Henryk Krawczyk, Bernard F. Kubiak, Roman Kulikowski, Marian Kuraś, Ludwik Maciejec, Marek Miłoś, Janusz Stokłosa, Jan Studziński, Zdzisław Szyjewski.

© Instytut Badań Systemowych PAN, Warszawa 2000

ISBN 83-85847-53-7
ISSN 0208-8028

Rozdział 4

**Metody i algorytmy obliczeniowe
w systemach komputerowych**

PARADYGMAT OBLICZEŃ NA SŁOWACH W INTELIGENTNYM WYSZUKIWANIU DANYCH

Janusz Kacprzyk i Sławomir Zadrozny

*Institut Badań Systemowych PAN
E-mail: {kacprzyk,zadrozny}@ibspan.waw.pl*

Zaproponowano zastosowanie tzw. paradygmatu obliczeń na słowach [ang. computing with words – por. Zadeh i Kacprzyk (1999a, b)] do konstrukcji przyjaznego użytkownikowi i efektywnego interfejsu do wyszukiwania informacji w bazie danych. Rozważania zilustrowane są na przykładzie systemu FQUERY for Access zaproponowanego przez autorów, który umożliwia formułowanie zapytań zawierających rozmyte (nieprecyzyjne) terminy i kwantyfikatory lingwistyczne.

Słowa kluczowe: obliczenia na słowach, wyszukiwanie w bazie danych, zapytania rozmyte, logika rozmyta.

1. Wprowadzenie

Systemy zarządzania bazą danych (SZBD) umożliwiają efektywne gromadzenie i przetwarzanie dużych zbiorów danych. Jedną z ich podstawowych funkcji jest zapewnienie szybkiego dostępu do danych żądanych przez użytkownika. Zapytania użytkownika mogą być precyzyjne i rutynowe jak, np., w wypadku podsystemu do sporządzania listy płac. O efektywności systemu decyduje wtedy szybkość odpowiedzi. W praktyce często najważniejsza dla użytkownika jest łatwość formułowania zapytań, najlepiej w języku naturalnym. Szczególnie dotyczy to użytkowników nieinformatyków, jak np. członek zarządu firmy czy klient agencji nieruchomości. Zazwyczaj ich zapytania w formie oryginalnej zawierają będą niejednoznaczne, wysoce zagregowane, terminy, pojęcia i relacje, które raczej trudno przetłumaczyć do postaci zgodnej z tradycyjnymi językami zapytań. Taki proces tłumaczenia, wymagający często pomocy eksperta-informatyka, prowadzić może do niekontrolowanej re-interpretacji oryginalnego zapytania. Można to częściowo przezwyciężyć poprzez wprowadzenie dodatkowej warstwy oprogramowania do przetwarzania elementów lingwistycznych w zapytaniach. Warstwa taka obsługuje słownik terminów lingwistycznych

zapewniając ich definiowanie, gromadzenie i właściwą interpretację w trakcie przetwarzania zapytania, czyli powinna wspierać *obliczenia na słowach*.

W niniejszym artykule pokazujemy jak podstawowe elementy paradygmatu obliczeń na słowach Zadeha [por. Zadeh i Kacprzyk (1999a, b)]: wartości lingwistyczne (np. *niska* cena), relacje lingwistyczne (np. przychody *dużo większe niż* wydatki), modyfikatory lingwistyczne (np. *bardzo* wysoka cena), i kwantyfikatory lingwistyczne (np. *większość*) mogą być wykorzystane przy projektowaniu interfejsu użytkownika do wyszukiwania danych w bazie. Najpierw omawiamy, jak elementy lingwistyczne mogą być uwzględnione w składni tradycyjnego języka zapytań. Następnie podamy, jak definiowanie i przetwarzanie takich terminów lingwistycznych może przebiegać. Na koniec, przedyskutujemy, na przykładzie naszego pakietu FQUERY for Access [por. Kacprzyk i Zadrozny (1994a-1997c)], niektóre szczegóły implementacyjne.

2. Wyrażenia lingwistyczne w zapytaniach rozmytych

Zapytanie do bazy danych określa warunki jakie muszą spełniać poszukiwane dane. Często semantyka tych warunków odwołuje się do interpretacji języka naturalnego. Następnie, zapytanie takie, często nie bez kłopotów, tłumaczone jest na precyzyjne polecenia tradycyjnego języka zapytań. Przykładowo, poszukując domu, klient może preferować *tanie* oferty. W tradycyjnym języku zapytań pojęcie "tani" musi być wyrażone w postaci przedziału cen o granicach trudnych do określenia, a poza tym nie jest to adekwatne do sposobu rozumienia człowieka. Ta sama definicja pojęcia "tani" może być następnie użyta w innych zapytaniach, również wraz z wyrażeniami typu "*bardzo tani*", gdzie "*bardzo*" jest tzw. *modyfikatorem*, dookreślającym znaczenie pojęcia wyjściowego ("tani") niezależnie od kontekstu.

Zazwyczaj zapytanie zawiera więcej niż jeden warunek. W takim wypadku, użytkownik może wymagać spełnienia różnych kombinacji elementarnych warunków:

- wszystkie mają być spełnione (odpowiada to użyciu spójnika I),
- przynajmniej jeden ma być spełniony (odpowiada to użyciu spójnika LUB)

W pierwszym wypadku spełnienie pojedynczego warunku ma znaczenie jedynie, jeśli wszystkie pozostałe są spełnione. W drugim wypadku, spełnienie jakiegokolwiek warunku wystarczy dla spełnienia zapytania. Czasami użyteczny może być bardziej złożony schemat *agregacji* spełnienia

warunków, pomiędzy tymi dwoma ekstremalnymi. Dokładniej, chcielibyśmy traktować każdy z warunków jako określający *częściowe* spełnienie zapytania. Wtedy, tylko spełnienie ustalonej proporcji, np. *większości*, *całkowicie* spełnia zapytania. Co więcej, chcielibyśmy różnicować warunki pod względem ich znaczenia dla spełnienia zapytania. Niektóre z warunków mogą być *ważniejsze* niż inne. Taki elastyczny schemat agregacji może być reprezentowany z użyciem kwantyfikatorów rozmytych (lingwistycznych), umożliwiając uwzględnienie ważności elementów podlegających agregacji.

Skoncentrujemy się na tradycyjnym języku zapytań SQL, stanowiącym standard. W tym kontekście, możliwe jest jeszcze inne zastosowanie kwantyfikatorów rozmytych. Mogą być one użyte w miejsce klasycznych kwantyfikatorów, wyrażonych w SQL za pośrednictwem słów kluczowych ANY (SOME) i ALL. Kwantyfikatory te używane są wraz z podzapytaniami w klauzuli WHERE zapytania SQL-owego.

Podsumowując, wyrażenia lingwistyczne (wraz z ewentualnymi parametrami), jakie mogą być użyteczne w kontekście zapytań można sklasyfikować jako:

- 1) wyrażenia reprezentujące immanentną nieprecyzyjność warunków zapytania, w tym:
 - a) numeryczne wartości rozmyte,
 - b) modyfikatory,
 - c) relacje rozmyte,
 - d) zbiory rozmyte wartości skalarnych,
- 2) wyrażenia odpowiadające niestandardowym operatorom agregacji, w tym:
 - a) kwantyfikatory rozmyte (lingwistyczne),
 - b) współczynniki ważności.

Tradycyjne języki obsługi systemów zarządzania bazą danych nie pozwalają na bezpośrednie reprezentowanie takich wyrażen lingwistycznych. Proponujemy tu więc rozszerzenia języków zapytań, takich jak SQL w duchu paradygmatu *obliczeń na słowach*. W tym punkcie omówimy proponowane rozwinięcie składni języka zapytań. Następnie pokażemy właściwą interpretację poszczególnych konstrukcji lingwistycznych oraz sposób ich pozyskiwania, reprezentacji i przetwarzania. W następnym punkcie dyskutujemy te kwestie na przykładzie naszego pakietu FQUERY for Access.

Uwagę naszą koncentrujemy na poleceniu SELECT...FROM... WHERE języka SQL, a właściwie na rozszerzeniach klauzuli WHERE. Punktem wyjścia jest uproszczona jej wersja z wyłączeniem, np., podzapytań, a następnie rozszerzamy jej tradycyjną składnię uwzględniając bezpośrednio możliwość użycia wyrażeń lingwistycznych:

```

<kaluzula-WHERE > ::= WHERE <warunek>
<warunek> ::= <kwantyfikator rozmyty > <sekwencja
    podwarunków > ;
<sekwencja podwarunków > ::= <podwarunek> | <podwarunek> OR
    <sekwencja podwarunków >
<podwarunek> ::= <współczynnik ważności><kwantyfikator
    rozmyty><sekwencja prostych warunków >
<sekwencja prostych warunków > ::= <prosty warunek > |
    <prosty warunek > AND <sekwencja
    prostych warunków >
<prosty warunek > ::= <atrybut > = <modyfikator>
    <wartość rozmyta > |
    <atrybut > <relacja rozmyta > <atrybut > |
    <atrybut > <relacja rozmyta> <liczba > |
    <atrybut skalarny> IN <zbiór rozmyty > |
    <atrybut wielowartościowy> <operator zgodności > <zbiór
    rozmyty > |
< atrybut > ::= <kolumna liczbowa>
<kwantyfikator rozmyty > ::= <znacznik-OWA > <etykieta
    kwantyfikatora>
<znacznik-OWA > ::= OWA |
<modyfikator > ::= VERY | MORE OR LESS | RATHER | NOT
<operator zgodności > ::= możliwe dopasowanie | pewne
    dopasowanie | wskaźnik Jackarda

```

Poniżej omówimy poszczególne kategorie tych wyrażeń lingwistycznych odwołując się do przykładu bazy danych ofert domów opisanych przez atrybuty: cena, powierzchnia działki, lokalizacja (region), liczba sypialni i łazienek oraz wskaźniki jakości życia, takie jak, ogólna ocena otoczenia, infrastruktura transportowa i handlowa.

Prosty warunek

Prosty warunek jest elementarnym składnikiem warunku zapytania. Składa się on z nazwy atrybutu i ograniczenia nałożonego na wartość, jak np.

Cena <= 200 000

Może również zawierać wyrażenia lingwistyczne jak np.:

- | | |
|---|---|
| 1. <i>Cena = niska</i> | (wartość rozmyta) |
| 2. <i>Powierzchnia działki = bardzo duża</i> | (wartość rozmyta +
modyfikator) |
| 3. <i>Cena nie dużo większa niż 250 000</i> | (relacja rozmyta) |
| 4. <i>Lokalizacja w ulubionym regionie</i> | (zbiór rozmyty) |
| 5. <i>Wskaźniki jakości życia są zgodne ze Wzorcem Wysokiej Jakości Życia</i> | (atrybut wielowartościowy +
zbiór rozmyty) |

Wartości rozmyte mogą być użyte przy określaniu ograniczeń dotyczących kolumn liczbowych, np. kolumny *Cena*. Znaczenie takiego wyrażenia lingwistycznego jest intuicyjnie oczywiste, choć subiektywne. W związku z tym, każdy użytkownik powinien mieć możliwość zdefiniowania swojego rozumienia takich wyrażen jak "niski". Z drugiej strony, pożądana byłaby możliwość wykorzystania raz zdefiniowanego wyrażenia lingwistycznego w kontekście różnych kolumn w poszczególnych tabelach bazy danych. Wartości rozmyte mogą być opatrzone *modyfikatorami*, np., *bardzo*.

Relacje rozmyte umożliwiają uelastycznienie semantyki klasycznych operatorów relacyjnych. W trzecim z powyższych przykładów, prosty warunek zawiera rozmytą relację "*dużo większa niż*" wraz z *negacją traktowaną jako modyfikator*. Dzięki temu, warunek ten będzie spełniony dla ceny nieco wyższej, jak np., 255 000, co wydaje się być dużo praktyczniejsze niż uznanie 250 000 jako nieprzekraczalnego ograniczenia.

Dotychczas omówione wyrażenia lingwistyczne odnoszą się wyłącznie do danych liczbowych. Czwarty przykład wprowadza *zbiór rozmyty* użyty dla danych skalarnych. Stała (*zbiór rozmyty*) *ulubione regiony* reprezentuje preferencje użytkownika co do lokalizacji poszukiwanej nieruchomości. Pojęcie "ulubionych regionów" często będzie miało charakter rozmyty, to znaczy niektóre regiony będą postrzegane przez użytkownika (subiektywnie) jako najlepsza lokalizacja, inne będą całkowicie nie do przyjęcia, a pozostałe będą akceptowalne do pewnego stopnia.

Piąty przykład ilustruje pojęcie *atrybutów wielowartościowych*. Pojęcie to wprowadziliśmy zainspirowani zastosowaniem interfejsu rozmytych zapytań do wyszukiwania w bazie danych zawierającej kwestionariusze opisujące instytucje naukowo-badawcze (por. np. Kacprzyk i Zadrozny, 1997c, Zadrozny i Kacprzyk, 1996). A mianowicie, założmy, że każda instytucja naukowo badawcza opisana jest za pomocą typowych atrybutów takich jak np., nazwa, adres itd. Zazwyczaj instytucje tego typu zajmują się kilkoma dyscyplinami, jednymi w większym innymi w mniejszym stopniu. Tak więc, z punktu widzenia użytkownika, wartość atrybutu opisującego przedmiot *dziedziny działalności*, może stanowić zbiór, a nie pojedynczy

element, jak to jest wymagane przez w relacyjnym modelu danych. Często korzystnie będzie traktować ten zbiór jako rozmyty. Przy tych założeniach, użytkownik poszukujący instytucji naukowo-badawczej, może w swoim zapytaniu podać zbiór (np. rozmyty) dyscyplin naukowych jako ograniczenie dla atrybutu "dziedzina działalności". Powinien przy tym również wskazać *operator zgodności* do ustalenia dopasowania zbiorów (np. rozmytych) występujących w zapytaniu i w bazie danych. Poszczególne operatory zgodności wymienione w proponowanej rozszerzonej składni języka zapytań mogą spełniać różne wymagania użytkownika, jak np.:

- przynajmniej jedną dyscypliną spośród wymienionych w zapytaniu,
- wszystkimi dyscyplinami występującymi w zapytaniu (i być może jeszcze innymi),
- tylko dyscyplinami wymienionymi w zapytaniu, niekoniecznie wszystkimi,
- większością dyscyplin wymienionych w zapytaniu i niewielu innymi.

Oczywiście, zapytania te mogą być skonstruowane bez odwoływania się do atrybutów wielowartościowych i operatorów zgodności, co może być kłopotliwe.

Należy podkreślić różnicę między wprowadzonym pojęciem atrybutów wielowartościowych i podobnym pojęciem używanym w teorii rozmytych baz danych. W wypadku obydwu głównych modeli rozmytych baz danych (z użyciem relacji podobieństwa czy rozkładów możliwości), zakłada się, że kolumna może przyjmować jako wartość zbiór. Jednak, w obydwu modelach, zbiór jest stosowany ze względu na nieprecyzyjną wiedzę co do rzeczywistej *pojedynczej* wartości danej kolumny. Tu atrybuty wielowartościowe z definicji przyjmują zbiory jako wartość. Jak wspomniano, wprowadziliśmy to pojęcie ze względów praktycznych, bo proponowany interfejs zapytań rozmytych jest lepiej dopasowany dla pewnych zastosowań. Choć wydaje się to wykraczać poza relacyjny model danych, to faktycznie tak nie jest. Jak wyjaśniamy w następnym punkcie, atrybut wielowartościowy jest w pewnym sensie pojęciem *wirtualnym* i nie wymaga żadnych zmian w używanym SZBD. Taka wirtualna natura atrybutów wielowartościowych przypomina koncepcję Nomury i in. (1992) atrybutów niejednoznacznych.

Tradycyjne proste warunki, nie zawierające wyrażeń lingwistycznych, mogą być spełnione przez dany wiersz z bazy danych (tabeli) lub niespełnione, czyli klasyczna logika stosuje się do pojęcia spełniania. W wypadku wyrażeń lingwistycznych, prosty warunek może być spełniony do pewnego *stopnia*. Stanowi to istotę podejścia lingwistycznego. Przykłado-

wo, cena może być zdecydowanie *niska* lub zdecydowanie nie być *niska*, ale również pośrednie przypadki są możliwe.

Sekwencja prostych warunków i podwarunek

Sekwencja prostych warunków jest koniunkcją prostych warunków. Ponieważ proste warunki mogą być spełnione do pewnego stopnia, konieczne jest uogólnienie klasycznego spójnika logicznego I, a naturalnym wyborem jest operator *minimum*.

W rozważanym tu kontekście spójnik I może być postrzegany jako operator agregacji stopni spełnienia (prawdziwości) prostych warunków. Schemat agregacji reprezentowany przez I może być czasami niezgodny z intuicją. Rozważmy dwa przypadki. Pierwszy, gdy wszystkie poza jednym prostym warunkiem są spełnione, i drugi, gdy tylko jeden prosty warunek jest spełniony. W obydwu przypadkach, wynikiem agregacji będzie 0 (*falsz*). Operator minimum, umożliwiając zastosowanie schematu działania spójnika I w przypadku logiki wielowartościowej zachowuje jego wskazaną, czasami niezgodną z intuicją, cechę. Dla uzyskania większej elastyczności i zgodności z intuicją można tu zastosować kwantyfikator rozmyty jako operator agregacji. Kwantyfikatory rozmyte (lingwistyczne) mogą być definiowane i interpretowane na dwa sposoby - patrz następny punkt.

Każda sekwencja prostych warunków może być dodatkowo opatrzona współczynnikiem ważności. W ten sposób użytkownik może zmienić stopień w jaki dana sekwencja uczestniczy w spełnieniu całego zapytania.

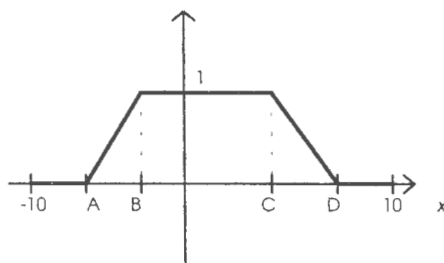
Sekwencja prostych zapytań, z ewentualnie przypisanym kwantyfikatorem rozmytymi współczynnikiem ważności nazywana jest podwarunkiem.

Sekwencja podwarunków i warunek

Sekwencja podwarunków jest dysjunkcją podwarunków. Tak jak w wypadku I, spójnik LUB jest automatycznie zastępowany przez odpowiadający mu w logice rozmytej operator *maximum*. Ponownie, często korzystnym może być użycie innego, bardziej elastycznego schematu agregacji. W takim wypadku, kwantyfikator rozmyty może znaleźć zastosowanie. Jego funkcjonowanie może być sterowane poprzez wybór odpowiednich współczynników ważności, zastosowanych w konkretnym podwarunku.

3. Reprezentacja i określanie wyrażeń lingwistycznych

Wszystkie wyrażenia lingwistyczne reprezentowane są jako zbiory rozmyte w różnych przestrzeniach, zależnie od typu wyrażenia lingwistycznego. W szczególności, do reprezentacji wyrażeń używanych z danymi liczbowymi posługujemy się trapezoidalnymi liczbami rozmytymi. Przykładowo, jeśli dziedziną kolumny *cena* jest przedział $[10\ 000, 100\ 000]$, to wartość rozmyta *niska* w prostym warunku “cena jest niska” jest określona jako liczba rozmyta na tym przedziale. Jednak, stosując się literalnie do tego schematu, musimy zdefiniować wyrażenie *niska* z osobna dla każdego atrybutu z którym wyrażenie to może wystąpić. Proponujemy więc użycie niezależnych od kontekstu wyrażeń lingwistycznych zdefiniowanych na uniwersalnym przedziale, np., $[-10, 10]$. Dodatkowo, dla każdej kolumny liczbowej należy określić zakres jej wartości, czyli przedział $[DolnaGranica, GórnaGranica]$. W trakcie obliczania stopnia spełnienia zapytania liczby rozmyte redefiniowane są z uwzględnieniem zakresu wartości współwystępujących atrybutów. Rysunek 1 pokazuje przykład trapezoidalnej liczby rozmytej (“*średnia*”).



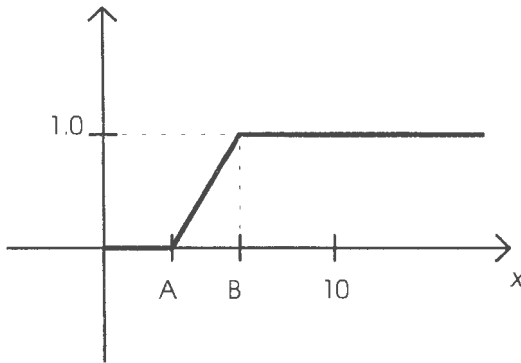
Rys. 1. Trapezoidalna funkcja przynależności wartości rozmytej

Tak więc, konieczne jest podanie czterech liczb, których interpretacja jest oczywista.

To samo podejście znajduje zastosowanie w wypadku *relacji rozmytych*, będących zbiorem rozmytym określonym na iloczynie kartezjańskim dwóch dziedzin. U nas są to dziedziny wartości dwóch atrybutów związanych w prostym warunku daną relacją rozmytą. Definiowanie relacji rozmytej wprost na takim iloczynie kartezjańskim mogłoby być raczej kłopotliwe. Proponujemy więc, aby ją definiować na przestrzeni wartości możliwych do uzyskania jako różnica dwóch atrybutów, przez co ten sam uniwersalny przedział $[-10,10]$ jest stosowany. W trakcie obliczania stopnia spełnienia zapytania jest on przekształcany na rzeczywisty przedział możli-

wych różnic wartości dwóch atrybutów. Oczywiście, są to nadal liczby rozmyte mające trapezoidalne funkcje przynależności.

Koncepcja *uniwersalnej* dziedziny może być również zastosowana w wypadku definicji kwantyfikatorów rozmytych w sensie Zadeha (1983). Kwantyfikator reprezentowany jest jako liczba rozmyta o funkcji przynależności kawałkami liniowej, jak np. na Rys. 2. Kwantyfikator rozmyty może być traktowany jako operator agregacji. W celu przeprowadzenia agregacji musimy znormalizować liczbę wielkości agregowanych, N , tu odpowiadającą liczbie *prostych warunków* lub *podwarunków*. Tak więc liczba N będzie się istotnie zmieniała zależnie od postaci zapytania. W celu uniknięcia konieczności definiowania kwantyfikatorów dla każdego zapytania z osobna posługujemy się ponownie uniwersalnym przedziałem, tym razem postaci $[0,10]$.



Rys. 2. Przykład funkcji przynależności kwantyfikatora rozmytego

Wyrażenia lingwistyczne w postaci *zbiorów rozmytych*, przeznaczone do użycia wraz z atrybutami skalarnymi muszą być zdefiniowane bezpośrednio w odpowiednich przestrzeniach - domenach atrybutów skalarnych. Dokładniej, zbiór wartości, jakie przyjmuje dany atrybut skalarny, stanowi przestrzeń do definiowania zbiorów rozmytych. Tak więc, te wyrażenia lingwistyczne są z definicji zależne od kontekstu. Mogą być one definiowane dla każdej kolumny typu tekstowego jak również dla *atrybutów wielowartościowych*. Wymagają osobnej definicji, jak w w poprzednim punkcie. Definicja atrybutu wielowartościowego polega na wskazaniu listy kolumn, które wspólnie mają być traktowane jako jeden *wirtualny* atrybut.

4. Stopień spełnienia zapytania

W odpowiedzi na zapytanie system zarządzania bazą danych zwraca listę wierszy spełniających warunki zapytania. W podejściu tradycyjnym spełnianie ma charakter binarny - odpowiada mu więc zbiór $\{0,1\}$. Naturalną konsekwencją dopuszczenia w zapytaniach wyrażeń lingwistycznych, jest konieczność uwzględnienia *stopnia* spełnienia. Poniżej krótko omówimy algorytm obliczania stopnia spełnienia dla poszczególnych wyrażeń lingwistycznych omówionych w poprzednich punktach.

Rzeczywiste bazy danych zawierają zwykle setki tysięcy wierszy lub więcej. Zapytania muszą być przetwarzane efektywnie z wykorzystaniem indeksów, statystyk itd. Tu przyjmiemy znacznie bardziej uproszczoną wizję realizacji zapytania. Mianowicie zakładamy, że w trakcie realizacji zapytania cała baza (tabela) jest przeglądana sekwencyjnie i dla każdego wiersza obliczany jest stopień spełnienia. Tak więc, przetwarzając zapytanie, SZBD wykonuje następujące kroki::

- 1) pobiera wiersz z bazy danych,
- 2) wylicza stopnie spełnienia wszystkich (lub tylko wybranych, zależnie od struktury zapytania i stosowanej optymalizacji) prostych warunków zapytania podstawiając wartości z bieżącego wiersza,
- 3) agreguje częściowe stopnie spełnienia uzyskane w poprzednim kroku w celu wyliczenia poszukiwanego całościowego stopnia spełnienia zapytania,
- 4) jeśli stopień spełnienia jest dostatecznie duży wiersz uwzględniany jest w odpowiedzi SZBD,
- 5) jeśli nie ma więcej wierszy, to STOP,
- 6) przechodzi do kroku 1.

Jedynie kroki 2) i 3) wymagają komentarza. Przyjmujemy następującą notację:

AT_1, AT_2	– atrybuty,
$v(AT)$	– wartość atrybutu AT w danym wierszu,
$[LL_{AT}, UL_{AT}]$	– dolna i górna granica wartości przyjmowanych przez atrybut AT,
τ_{AT}	– przekształcenie przedziału $[LL_{AT}, UL_{AT}]$ na <i>uniwersalny przedział</i> $[-10, 10]$,
$\tau_{AT_1 AT_2}$	– przekształcenie przedziału $[LL_{AT_1} - UL_{AT_2}, UL_{AT_1} - LL_{AT_2}]$ na <i>uniwersalny przedział</i>

	[-10,10]
FV, μ_{FV}	– wartość rozmyta i jej funkcja przynależności $\mu_{FV}: [-10, 10] \rightarrow [0,1]$
MOD, η	– modyfikator i reprezentująca go funkcja $\eta: [0,1] \rightarrow [0,1]$
FR, μ_{FR}	– relacja rozmyta i jej funkcja przynależności $\mu_{FR}: [-10, 10] \rightarrow [0,1]$
FS, μ_{FS}	– zbiór rozmyty i jego funkcja przynależności
MVAT	– atrybut wielowartościowy,
μ_{MVAT}	– wartość atrybutu wielowartościowego, tj. funkcja przynależności odpowiedniego zbioru rozmytego
COMPOP	– operator zgodności (etykieta)
$\Psi_{COMPOP}(\cdot, \cdot)$	– funkcja reprezentująca dany operator zgodności (patrz komentarz pod tabelą)
Q, μ_Q	– kwantyfikikator rozmyty i jego funkcja przynależności $\mu_Q: [0, 10] \rightarrow [0,1]$
τ_N	– przekształcenie przedziału $[0, N]$ na <i>uniwersalny przedział</i> $[0,10]$; N jest liczbą prostych warunków w podwarunku lub liczbą podwarunków w zapytaniu
md_i	– częściowy stopień spełnienia (<i>i</i> -tego prostego warunku lub podwarunku)
κ_i	– współczynnik ważności <i>i</i> -tego podwarunku

W Tabeli 1 zestawiliśmy wzory stosowane do obliczenia stopnia spełnienia prostych warunków zawierających poszczególne wyrażenia lingwistyczne

Przekształcenia τ są określone również poza wcześniej wspomnianymi przedziałami. Mianowicie, dla argumentów mniejszych (większych) niż dolna (górną) granica przedziału przyjmują wartość -10 (10). To dodatkowe założenie jest konieczne ponieważ możliwe jest, że np., $v(AT)$ w pewnym wierszu leży poza przedziałem zmienności zadeklarowanym przez użytkownika.

Operatory zgodności mierzą zgodność wartości atrybutu wielowartościowego i zbioru rozmytego. Zgodnie z punktem 2, różne operatory zgodności znajdują zastosowanie w zależności od semantyki zapytania. W Tabeli 2 podajemy wzory służące wyliczaniu Ψ_{COMPOP} dla czterech operatorów.

Operator zgodności oparty na pojęciu pewności dopasowania (drugi w tabeli) jest niesymetryczny. W związku z tym jego druga wersja (podana

jako trzecia w tabeli) ma inną semantykę która może być odpowiednia dla pewnych zapytań.

Tabela 1

Typ wyrażenia lingwistycznego	Postać prostego warunku	Wzór na stopień spełnienia
Wartość rozmyta	AT = FV	$\mu_{FV}(\tau_{AT}(v(AT)))$
Wartość rozmyta z modyfikatorem	AT = MOD FV	$\eta(\mu_{FV}(\tau_{AT}(v(AT))))$
Relacja rozmyta	AT ₁ FR AT ₂	$\mu_{FR}(\tau_{AT_1 AT_2}(v(AT_1) - v(AT_2)))$
Relacja rozmyta	AT FR liczba	$\mu_{FR}(\tau_{AT AT}(v(AT) - number))$
Zbiór rozmyty	AT in FS	$\mu_{FS}(v(AT))$
Zbiór rozmyty z atrybutem wielowartościowym	MVAT COMPOP FS	$\psi_{COMPOP}(\mu_{MVAT}, \mu_{FS})$

Wreszcie, częściowe stopnie spełnienia obliczone dla prostych warunków (podwarunków) są agregowane następujący sposób:

$$md = \mu_Q(\tau_N(\frac{1}{N} \sum_i md_i))$$

gdzie md oznacza stopień spełnienia podwarunku (całego zapytania).

Wzór na obliczenie stopnia spełnienia całego zapytania może być rozszerzony w celu uwzględnienia współczynników ważności w następujący sposób (patrz Zadeh, 1983):

$$md = \mu_Q(\tau_N(\frac{\sum_i (md_i \wedge \kappa_i)}{\sum_i \kappa_i}))$$

Tabela 2

Operator zgodności	Wzór na $\Psi_{\text{COMPOP}}(\mu_{\text{MVAT}}, \mu_{\text{FS}})$
Możliwe dopasowanie	$\Pi(\text{FS} \text{MVAT}) = \sup_{u \in U} \min(\mu_{\text{FS}}(u), \mu_{\text{MVAT}}(u))$
Pewne dopasowanie	$N(\text{FS} \text{MVAT}) = \inf_{u \in U} \max(1 - \mu_{\text{FS}}(u), \mu_{\text{MVAT}}(u))$
Pewne dopasowanie 2	$N(\text{MVAT} \text{FS}) = \inf_{u \in U} \max(1 - \mu_{\text{MVAT}}(u), \mu_{\text{FS}}(u))$
Wskaźnik Jaccarda	$\frac{ FS \cap D }{ FS \cup D }$ $\stackrel{\text{def}}{=} \frac{\sum_{u \in U} (\mu_{\text{FS}}(u) \wedge \mu_{\text{MVAT}}(u))}{\sum_{u \in U} (\mu_{\text{FS}}(u) \vee \mu_{\text{MVAT}}(u))}$

5. Uwagi końcowe

Przedstawiliśmy zastosowanie tzw. paradygmatu obliczeń na słowach do wyszukiwania informacji w bazie danych. Tradycyjne narzędzia SZBD nie są do tego przystosowane, ponieważ nie pozwalają na użycie nieprecyzyjnych elementów i agregacji. Logika rozmyta może być tu bardzo przydatna. Zaproponowano rozszerzenie klasycznej składni języka SQL, zaimplementowane w systemie FQUERY for Access [por. Kacprzyk i Zadrożny (1994a – 1997c)].

Literatura

- Bordogna G., P. Carrara i G. Pasi (1995) Fuzzy approaches to extend Boolean information retrieval. In: P. Bosc i J. Kacprzyk (Eds.) *Fuzziness in Database Management Systems*, Physica-Verlag, Heidelberg, pp. 231 - 274.
- Bosc P., M. Galibourg i G. Hamon (1988) Fuzzy querying with SQL: extensions and implementations aspects. *Fuzzy Sets and Systems* 28, 333 - 349.
- Bosc P. i J. Kacprzyk, Eds. (1995) *Fuzziness in Database Management Systems*. Physica-Verlag, Heidelberg,
- Bosc P. i O. Pivert (1992) Fuzzy querying in conventional databases. In L.A. Zadeh i J. Kacprzyk (Eds.): *Fuzzy Logic for the Management of Uncertainty*. Wiley, New York, pp. 645-671.

- Bosc P. i O. Pivert (1994) SGL^f: a relational database language for fuzzy querying. IEEE Trans. on Fuzzy Systems (to appear).
- Bosc P., L. Lietard i O. Pivert (1995) Quantified statements and database fuzzy querying. In P. Bosc i J. Kacprzyk (Eds.): Fuzziness in Database Management Systems. Physica-Verlag, Heidelberg, pp. 275-308.
- Chang S.K. i Ke J.S. (1978) Database skeleton and its application to fuzzy query translation. IEEE Trans. on Software Eng. SE-4, 31-43.
- Chang S.K. i Ke J.S. (1979) Translation of fuzzy queries for relational database systems. IEEE Trans. on Pattern Anal. and Machine. Intel. PAMI-1, 281-294.
- Kacprzyk J. i S. Zadrożny (1994a) Fuzzy querying for Microsoft Access. Proceedings of the Third IEEE Conference on Fuzzy Systems (Orlando, USA), Vol. 1, pp. 167-171.
- Kacprzyk J. i S. Zadrożny (1994b) Fuzzy queries in Microsoft Access: toward a 'more intelligent' use of Microsoft Windows based DBMSs, Proceedings of the 1994 Second Australian i New Zealand Conference on Intelligent Information Systems - ANZIIS'94 (Brisbane, Australia), pp. 492 - 496.
- Kacprzyk J. and S. Zadrożny (1995a) FQUERY for Access: fuzzy querying for a Windows-based DBMS. In: P. Bosc i J. Kacprzyk (Eds.) Fuzziness in Database Management Systems, Physica-Verlag, Heidelberg, pp. 415 - 433.
- Kacprzyk J. i S. Zadrożny (1995b) Fuzzy queries in Microsoft Access v. 2, Proceedings of 6th International Fuzzy Systems Association World Congress (Sao Paulo, Brazil), Vol. II, pp. 341 - 344.
- Kacprzyk J. i S. Zadrożny (1997a) Fuzzy queries in Microsoft Access v. 2, in D. Dubois, H. Prade i R.R. Yager (Eds.): Fuzzy Information Engineering - A Guided Tour of Applications, Wiley, New York, 1997, pp. 223 - 232.
- Kacprzyk J. i S. Zadrożny (1997b) Implementation of OWA operators in fuzzy querying for Microsoft Access. In: R.R. Yager i J. Kacprzyk (Eds.) The Ordered Weighted Averaging Operators: Theory and Applications, Kluwer, Boston 1997, pp. 293 - 306.
- Kacprzyk i S. Zadrożny (1997c) Flexible querying using fuzzy logic: An implementation for Microsoft Access, in T. Andreassen, H. Christiansen i H.L. Larsen (eds.): Flexible Query Answering Systems, Kluwer, Boston, 1997, pp. 247-275.
- Kacprzyk J., Zadrożny S. i Ziółkowski A. (1989) FQUERY III+: a 'human consistent' database querying system based on fuzzy logic with linguistic quantifiers. Information Systems 6, 443 - 453.
- Kacprzyk J. i Ziółkowski A. (1986b) Database queries with fuzzy linguistic quantifiers. IEEE Transactions on Systems, Man and Cybernetics SMC - 16, 474 - 479.

- Larsen H.L. i Yager R.R. (1993) The use of fuzzy relational thesauri for classificatory problem solving in information retrieval and expert systems. *IEEE Trans. On Syst., Man and Cybern.* SMC-23, 31 - 41.
- Miyamoto S. (1990) *Fuzzy Sets in Information Retrieval and Cluster Analysis*. Kluwer Academic Publishers. Dordrecht, Boston, London.
- Nomura T., Odaka T., Ohki N., Yokoyama T. i Matsuhita Y. (1992) Generating ambiguous attributes for fuzzy queries. *Proceedings of IEEE International Conference on Fuzzy Systems (FUZZ-IEEE '92)*, 1992, pp. 753-760.
- Petry F.E. (1996) *Fuzzy Databases: Principles and Applications*. Kluwer, Boston.
- Tahani V. (1977) A conceptual framework for fuzzy query processing: a step toward very intelligent data systems. *Inf. Proc and Management* 13, 289 - 303.
- Yazici A., R. George, B.P. Buckles i F.E. Petry (1992) A survey of conceptual and logical data models for uncertainty management. In L.A. Zadeh i J. Kacprzyk (Eds.): *Fuzzy Logic for the Management of Uncertainty*. Wiley, New York, pp. 607-643.
- Zadeh L.A. (1983) A computational approach to fuzzy quantifiers in natural languages. *Computers and Maths. with Appls.* 9, 149 - 184.
- Zadrożny S. i J. Kacprzyk (1995) Fuzzy querying using the 'query-by-example' option in a Windows-based DBMS", *Proceedings of Third European Congress on Intelligent Techniques and Soft Computing - EUFIT'95 (Aachen, Germany)*, vol. 2, pp. 733-736.
- Zadrożny S. i J. Kacprzyk (1996) Multi-valued fields and values in fuzzy querying via FQUERY for Access, *Proceedings of FUZZ-IEEE'96 - Fifth International Conference on Fuzzy Systems (New Orleans, USA)*, vol. 2, pp. 1351 - 1357.
- Zemankova M. i J. Kacprzyk (1993) The roles of fuzzy logic and management of uncertainty in building intelligent information systems, *Journal of Intelligent Information Systems* 2, 311-317.

ISSN 0208-8029
ISBN 83-85847-53-7

**W celu uzyskania bliższych informacji i zakupu dodatkowych egzemplarzy
prosimy o kontakt z Instytutem Badań Systemowych PAN
ul. Newelska 6, 01-447 Warszawa
tel. 837-35-78 w. 241 e-mail: bibliote@ibspan.waw.pl**