POLSKA AKADEMIA NAUK
Instytut Badań Systemowych

# ROZMYTOŚĆ I BIPOLARNOŚĆ W INTELIGENTNYM WYSZUKIWANIU INFORMACJI

Sławomir Zadrożny

Warszawa 2013


**POLSKA AKADEMIA NAUK**
**INSTYTUT BADAŃ SYSTEMOWYCH**

**Seria: BADANIA SYSTEMOWE**
**Tom 73**

**Redaktor naukowy:**
**Prof. dr hab. inż. Jakub Gutenbaum**

**Warszawa 2013**

Rada redakcyjna serii: BADANIA SYSTEMOWE


**POLSKA AKADEMIA NAUK**
**INSTYTUT BADAŃ SYSTEMOWYCH**

**Sławomir Zadrożny**

# ROZMYTOŚĆ I BIPOLARNOŚĆ W INTELIGENTNYM WYSZUKIWANIU INFORMACJI

**Warszawa 2013**

## Autorzy:

**Dr hab. Sławomir Zadrożny**
Instytut Badań Systemowych Polskiej Akademii Nauk
ul. Newelska 6, 01-447 Warszawa
*Slawomir.Zadrozny@ibspan.waw.pl*

## Recenzenci:

**dr hab. inż. Maciej Krawczak**
**dr Marek Reformat**

**Skład:** Aneta M. Pielak

**Wydawca:**

**Instytut Badań Systemowych**
**Polskiej Akademii Nauk**
Newelska 6, 01-447 Warszawa
www.ibspan.waw.pl

**ISSN 0208-8029**
**ISBN 83-894-7551-0**

# Rozdział 4

# Niestandardowe zapytania do baz danych

W literaturze zaproponowano wiele rozszerzeń klasycznego modelu relacyjnego i jego języków zapytań. W rozdziale 5 omawiamy nowe podejście, tak zwane *zapytanie bipolarne*. W niniejszym rozdziale omówimy kilka innych wybranych sposród nich, zwracając szczególną uwagę na zastosowanie w nich elementów logiki rozmytej.

Zapytanie skierowane przez użytkownika do systemu zarządzania bazą danych wyraża jego *preferencje* co do poszukiwanych danych. Na użytek naszych rozważań, w pewnym uproszczeniu, utożsamiamy zapytanie z warunkiem (formułą logiczną). W klasycznym modelu relacyjnym warunek $C$ określa praporządek $\preceq$ na zbiorze krotek $T$, taki że:

$$t \preceq s \longleftrightarrow C(t) \leq C(s) \tag{4.1}$$

przy czym $C(t)$ oznacza wartość dla krotki $t \in T$ funkcji charakterystycznej zbioru krotek należących do $T$ i spełniających warunek $C$. Relacja $\preceq$ wyraża preferencje użytkownika dla krotek spełniających warunek $C$.

Preferencje użytkowników są jednak często dużo bardziej złożone. Znaczny postęp w możliwościach ich reprezentowania uzyskujemy stosując *zapytania nieprecyzyjne*, omawiane w p. 4.3. Ich istota polega na użyciu *predykatów rozmytych* pozwalających reprezentować nieprecyzyjność i intensywność preferencji użytkownika, które przekładają się na *stopnie spełnienia* przez krotki warunków zapytania. Takie podejście jest w pewnym stopniu zapożyczone z systemów wyszukiwania informacji tekstowej (rozdział 6), gdzie na przykład w modelu wektorowym pojęcia *stopni ważności* słów kluczowych przy reprezentacji dokumentów i zapytań czy też pojęcie *relewantności* dokumentu względem zapytania mają

charakter stopniowalny. Warto zauważyć, że również w przypadku rozmytych predykatów wzór (4.1) zachowuje ważność i definiuje poprawnie uzyskiwane z użyciem zapytań nieprecyzyjnych uporządkowanie krotek.

Niektórzy autorzy [181] używają terminu "preferencje" w odniesieniu do zapytań wyłącznie wtedy, gdy używane są predykaty rozmyte. Inni [63] proponują zastosowanie bardziej złożonych i jawnie specyfikowanych nierozmytych relacji do określania preferencji użytkownika. W punkcie 4.2 opiszemy jedno z takich podejść.

Prezentację wybranych niestandardowych podejść do wyszukiwania informacji w bazach danych rozpoczniemy od krótkiego omówienia rozmytych wariantów klasycznych relacyjnych języków zapytań.

## 4.1 Rozmyte wersje klasycznych relacyjnych języków zapytań

W literaturze zaproponowano rozszerzenia *klasycznych języków zapytań* przyjętych w relacyjnym modelu danych (por. p. 3.1.1). Prace te dotyczą:

- *algebry relacji rozmytych*, która stanowi naturalne rozszerzenie klasycznej algebry relacji,
- pewnych prób rozszerzenia *rachunku relacyjnego*.

**Algebra relacji rozmytych.** Koncepcja algebry relacji rozmytych polega przede wszystkim na wprowadzeniu *terminów lingwistycznych* do warunków *operacji wyboru* (3.6). Warunek ten możemy utożsamić z wyrażeniem lingwistycznym "$X$ jest $F$" (2.119)[1]. Krotki spełniają taki warunek *w pewnym stopniu* określonym przez wartości funkcji przynależności $\mu_F(\cdot)$ zbioru rozmytego stanowiącego reprezentację terminu lingwistycznego $F$. W związku z tym wynikiem operacji wyboru staje się *relacja rozmyta*. Wymaga to określenia wyniku działania wszystkich operacji algebry w przypadku, gdy ich argumentami są relacje rozmyte.

Postać operacji takich jak: *suma*, *przecięcie*, *różnica* czy *iloczyn kartezjański*, wynika wprost z definicji tych operacji dla zbiorów rozmytych określonych wzorami (2.11)-(2.12), (2.14) oraz działań w klasycznej algebrze relacji określonych wzorami (3.2)-(3.4) i (3.8). Przyjmując oznaczenia z p. 3.1 i utożsamiając relację rozmytą $R$ z jej funkcją przynależności $\mu_R$, można te operacje zdefiniować następująco:

[1] Używamy tu symbolu $F$ na oznaczenie terminu lingwistycznego zamiast symbolu $A$ używanego oryginalnie w (2.119) dla uniknięcia konfliktu z oznczeniami przyjętymi dla atrybutów relacji.

**suma** $(R \cup S)$

$$\mu_{R\cup S}(t) = \max(\mu_R(t), \mu_S(t)) \tag{4.2}$$

**różnica** $(R \setminus S)$

$$\mu_{R\setminus S}(t) = \min(\mu_R(t), \mu_{\neg S}(t)) \tag{4.3}$$

**iloczyn kartezjański** $(R \times S)$

$$\begin{aligned}\mu_{R\times S}(\{A_1 : d_1, \ldots, A_n : d_n, B_1 : e_1, \ldots, B_m : e_m\}) = \\ \min(\mu_R(\{A_1 : d_1, \ldots, A_n : d_n\}), \mu_S(\{B_1 : e_1, \ldots, B_m : e_m\}))\end{aligned} \tag{4.4}$$

**przecięcie** $(R \cap S)$

$$\mu_{R\cap S}(t) = \min(\mu_R(t), \mu_S(t)) \tag{4.5}$$

Działania specjalne, takie jak *rzut*, *wybór* i *złączenie* definiuje się następująco:

**rzut** $(\pi_{A_1,\ldots,A_k}(R))$

$$\begin{aligned}\mu_{\pi_{A_1,\ldots,A_k}(R)}(\{A_1 : d_1, \ldots, A_k : d_k\}) = \\ \sup_{\{A_{k+1}:d_{k+1},\ldots,A_n:d_n\}} \mu_R(\{A_1 : d_1, \ldots, A_k : d_k, A_{k+1} : d_{k+1}, \ldots, A_n : d_n\})\end{aligned} \tag{4.6}$$

**wybór** $(\sigma_\varphi(R))$

$$\mu_{\sigma_\varphi(R)}(\{A_1 : d_1, \ldots, A_n : d_n\}) = \tag{4.7}$$

$$\min(\mu_R(\{A_1 : d_1, \ldots, A_n : d_n\}), \mu_\varphi(\{A_1 : d_1, \ldots, A_n : d_n\})) \tag{4.8}$$

**złączenie** $(R \bowtie_W S)$

$$\begin{aligned}\mu_{R\bowtie_W S}(\{A_1 : d_1, \ldots, A_n : d_n, B_1 : e_1, \ldots, B_m : e_m\}) = \\ \min(\mu_R(\{A_1 : d_1, \ldots, A_n : d_n\}), \mu_S(\{B_1 : e_1, \ldots, B_m : e_m\}), \\ \mu_W(\{A_1 : d_1, \ldots, A_n : d_n, B_1 : e_1, \ldots, B_m : e_m\}))\end{aligned} \tag{4.9}$$

**dzielenie** $(R \div S)$

$$\begin{aligned}\mu_{R\div S}(\{A_1 : d_1, \ldots, A_k : d_k\}) = \\ \min_{\{A_{k+1}:d_{k+1},\ldots,A_n:d_n\}}(\mu_S(\{A_{k+1} : d_{k+1}, \ldots, A_n : d_n\}) \rightarrow \\ \mu_R(\{A_1 : d_1, \ldots, A_k : d_k, A_{k+1} : d_{k+1}, \ldots, A_n : d_n\}))\end{aligned} \tag{4.10}$$

gdzie $R$ i $S$ są relacjami o schematach określonych zbiorami atrybutów, odpowiednio[2], $\{A_1, \ldots, A_n\}$ i $\{B_1, \ldots, B_m\}$; $\varphi$ jest rozmytym warunkiem wyboru[3] a $W$ oznacza rozmytą relację reprezentującą warunek złączenia.

Można zauważyć, że w przypadku relacji rozmytych operację złączenia można wyrazić jako złożenie operacji iloczynu kartezjańskiego i wyboru: $\mu_{R \rhd\lhd_W S}(\cdot) = \mu_{\sigma_W(R \times S)}(\cdot)$, podobnie jak w przypadku algebry relacji nierozmytych (p. 3.1.1).

Dla operacji dzielenia relacji rozmytych nie ma jednej powszechnie przyjętej definicji. Wzór (4.10) jest jednym z możliwych. Faktycznie opisuje on pewną klasę operacji, gdyż operator implikacji rozmytej $\rightarrow$ może w nim przyjąć jedną z wielu postaci – por. p. 2.2.2. Jednocześnie w literaturze zaproponowano również inne postacie operacji dzielenia. Szczegóły można znaleźć w pracach [47, 46, 90, 32, 31, 112, 227].

Na przykład w celu wybrania z tabeli `NIERUCHOMOSCI` (tabl. 3.1) ofert dotyczących nieruchomości o *niskiej* cenie zastosować można następującą operację wyboru

$$\sigma_{\texttt{cena}=niska}(\texttt{NIERUCHOMOSCI})$$

zakładając, że termin "niska" jest reprezentowany przez zbiór rozmyty o tej samej nazwie.

**Rozmyty rachunek relacyjny.** Rozszerzeniu drugiego, obok algebry relacji, standardowego języka zapytań w relacyjnym modelu danych, czyli *rachunku relacyjnego*, poświęcono w literaturze znacznie mniej uwagi. Wynika to z jednej strony z mniejszego zainteresowania nim w praktycznych zastosowaniach, a z drugiej strony z istotnych trudności z takim rozszerzeniem związanych. Źródłem tych trudności jest w znacznym stopniu istnienie wielu sposobów pojmowania *logiki rozmytej*, która stanowić tu musi punkt wyjścia.

Wśród autorów nielicznych prac w tym zakresie wymienić należy Takahashiego [207]. Podejście to inspirowane jest językiem PRUF, wprowadzonym przez Zadeha [240], stanowiącym *formalizm reprezentacji semantyki języka naturalnego*. Takahashi zaproponował oparty na nim język o nazwie FQL, który ma stanowić rozszerzenie *rachunku relacyjnego dziedzin*. Wadą tego podejścia jest nadmierna złożoność składni języka i brak przekonującej interpretacji wyrażeń języka PRUF w terminach zapytań do bazy danych. Poza tym autor w żaden sposób nie odnosi się

[2]W przypadku operacji dzielenia przyjmuje się, że schemat relacji $S$ jest podzbiorem schematu relacji $R$.

[3]Zawierającym *terminy lingwistyczne* modelowane z użyciem zbiorów rozmytych.

do zagadnienia *bezpiecznych* formuł (p. 3.1.1) w proponowanym języku. Inną, bardziej kompletną adaptację relacyjnego rachunku dziedzin na gruncie logiki rozmytej zaproponowali Buckles, Petry i Sachar [53]. Dotyczy ona zapytań względem *rozmytej bazy danych* (por. p. 3.2.2). Podobnie Galindo i in. [111] zaproponowali rozmytą wersję relacyjnego rachunku dziedzin dla rozwijanego przez siebie modelu rozmytej relacyjnej bazy danych GEFRED. Zadrożny i Kacprzyk [249] zaproponowali rozwinięcie relacyjnego rachunku dziedzin na podstawie jednego z wariantów *logiki rozmytej w węższym sensie* [177, 170]. Zaletą proponowanego podejścia jest możliwość formalnego wyrażenia w takim rachunku *stopni ważności* warunków, składających się na zapytanie.

## 4.2 Operacja *winnow* i zapytania z preferencjami

Chomicki [63] proponuje rozszerzenie języka zapytań (algebry relacji) o nową operację nazwaną *winnow*, co można przetłumaczyć jako "odsiewanie". Jest to operacja jednoargumentowa[4], której dodatkowy parametr stanowi relacja $G$. Relacja ta jest określona na zbiorze krotek stanowiącym argument operacji *winnow* i wyraża ona preferencje użytkownika. Wynikiem działania operacji *winnow* jest podzbiór zbioru krotek podanego jako argument, który zawiera jedynie krotki *niezdominowane* w sensie relacji preferencji $G$. Sformalizujemy teraz pojęcie operacji *winnow* następująco.

**Definicja 4.1.** *Relacją preferencji* na zbiorze krotek $T$ o schemacie postaci $\{A_1 : D_1, \ldots, A_n : D_n\}$ nazywa się dowolną relację binarną $G$:

$$G \subseteq (D_1 \times D_2 \times \ldots \times D_n) \times (D_1 \times D_2 \times \ldots \times D_n)$$

Jeśli dla dwóch krotek $t$ i $s$ zachodzi $G(t, s)$, to mówi się, że krotka $t$ *dominuje* nad krotką $s$.

Operację *winnow* definiuje się jako operację algebry relacji, która ze zbioru krotek wybiera krotki *niezdominowane* w sensie relacji $G$.

[4] Argumentem operacji *winnow* jest relacja, tak jak wszystkich innych klasycznych operacji algebry relacji. Występuje tu jednak pewna terminologiczna trudność związana z tym, że dodatkowym parametrem operacji *winnow* jest również relacja, tym razem określona na zbiorze krotek należących do relacji będącej argumentem *winnow* . W związku z tym, dla uniknięcia nieporozumień, relację podaną jako argument operacji *winnow* nazywać będziemy *zbiorem krotek*.

**Definicja 4.2** ([63])**.** Niech $T$ będzie zbiorem krotek, a $G$ określoną na nim, relacją preferencji. Wtedy *operację winnow* $\omega_G$ określa się następująco

$$\omega_G(T) = \{t \in T : \neg\exists_{s \in T}\ G(s,t)\} \tag{4.11}$$

*Zapytaniem z preferencjami* nazywa się zapytanie algebry relacji zawierające przynajmniej jedno wystąpienie operacji *winnow*. Chomicki [63] pokazuje, że operacja *winnow* może być wyrażona z użyciem klasycznej algebry relacji. Wskazuje również jednak, że operacja *winnow* ma wyraźnie określoną semantykę i jej wyróżnienie ułatwia badanie jej działania w zależności od własności przyjętej *relacji preferencji*. Pozwala to również na opracowanie specjalizowanych metod realizacji tej operacji i optymalizacji wykonania zapytań ją zawierających.

Zilustrujmy koncepcję operacji *winnow* na przykładach.

**Przykład 4.1.** Rozważmy tabelę `NIERUCHOMOSCI`, opisaną w tabl. 3.1. Załóżmy, że chcemy wybrać najtańsze domy oferowane w każdej z miejscowości. Określmy relację preferencji następująco:

$$G(t,s) \Leftrightarrow (t.\texttt{adres} = s.\texttt{adres}) \wedge (t.\texttt{cena} < s.\texttt{cena})$$

Wtedy operacja *winnow* $\omega_G(\texttt{NIERUCHOMOSCI})$ wybiera interesujące nas nieruchomości.

Ogólna definicja operacji *winnow* nie określa żadnych własności relacji preferencji $G$ – jest to dowolna relacja binarna na zbiorze krotek. W praktycznych zastosowaniach jednak takie założenia należy przyjąć, gdyż umozliwia to efektywniejszą realizację tej operacji. Z drugiej strony preferencje użytkownika zwykle również w naturalny sposób posiadają pewne cechy, które powodują że reprezentująca je relacja preferencji ma określone właściwości, zazwyczaj typowe dla *relacji porządku*.

**Rozmyta operacja *winnow*.** Operacja *winnow* może zostać również zaadoptowana na potrzeby algebry relacji rozmytych; por. p. 4.1, s. 82. Pojęcie rozmytej relacji preferencji i dominacji w sensie takiej relacji są przedmiotem zainteresowania wielu badaczy i doczekały się licznych opracowań (por. np. [107, 173]). Modelowanie preferencji to jedna z najważniejszych interpretacji przypisywanych wartościom funkcji przynależności w teorii zbiorów rozmytych [91].

Zadrożny i Kacprzyk zaproponowali rozmytą wersję operacji *winnow* w pracy [250]. Punktem wyjścia jest przyjęcie, że $G$ jest *rozmytą relacją*

*preferencji*. Przyjmuje się następnie definicję pojęcia dominacji względem takiej relacji, która ma charakter stopniowalny. W konsekwencji wynikiem zastosowania rozmytej operacji *winnow* jest rozmyty zbiór krotek – krotka może być do pewnego stopnia zdominowana i w związku z tym może należeć do zbioru krotek niezdominowanych, będącego wynikiem działania operacji, do pewnego stopnia.

Za [205] dogodnie jest przyjąć następujący zapis zbioru krotek niezdominowanych w sensie nierozmytej relacji prefrencji $G$, który następnie będzie można zaadoptować dla przypadku relacji rozmytych:

$$N(T,G) = T \cap \bigcap_{s \in T} \overline{G^{-}(s)} \tag{4.12}$$

przy czym

$$G^{-}(s) = \{u : G(s,u)\} \tag{4.13}$$

gdzie $N(T,G)$ oznacza podzbiór elementów niezdominowanych zbioru $T$ w sensie relacji preferencji $G$, natomiast $G^{-}(s)$ jest zbiorem elementów *zdominowanych przez element* $s$ w sensie relacji $G$, zaś $\overline{A}$ oznacza dopełnienie zbioru $A$.

Wzór (4.12) określa zbiór elementów niezdominowanych dla relacji $G$ jako przecięcie dopełnień wszystkich zbiorów elementów zdominowanych przez poszczególne elementy zbioru $T$ dla relacji $G$. Stosując ten wzór, operację *winnow* (4.11) definiuje się następująco

$$\omega_G(T) = N(T,G) \tag{4.14}$$

Rozmytą relację preferencji definiuje się analogicznie do przypadku nierozmytego (por. def. 4.1).

**Definicja 4.3.** *Rozmytą relacją preferencji* na *nierozmytym* zbiorze krotek $T$, o schemacie $\{A_1 : D_1, \ldots, A_n : D_n\}$, nazywa się dowolną relację binarną $\tilde{G}$:

$$\tilde{G} \in \mathcal{F}((D_1 \times D_2 \times \ldots \times D_n) \times (D_1 \times D_2 \times \ldots \times D_n))$$

którą utożsamia się z funkcją przynależności $\mu_{\tilde{G}}$.

Dla rozmytej relacji preferencji $\tilde{G}$ przyjmuje się, że zbiór elementów *niezdominowanych* jest zbiorem rozmytym. Do określenia tego zbioru stosuje się wzór (4.12), przyjmując operacje *przecięcia* (2.12) i *dopełnienia* (2.10) dla zbiorów rozmytych oraz następujące uogólnienie wzoru (4.13):

$$\mu_{\tilde{G}^{-}(s)}(u) = \mu_{\tilde{G}}(s,u) \tag{4.15}$$

określające funkcję przynależności *zbioru rozmytego* $\tilde{G}^{-}(s)$ elementów *zdominowanych* przez element $s$. Pozostaje określić rozszerzenie wzoru (4.12) na przypadek, gdy również zbiór $T$ jest *zbiorem rozmytym*. Wzór (4.12) można zapisać w postaci równoważnej formuły rachunku predykatów

$$N(T,\tilde{G})(t) \Leftrightarrow T(t) \wedge \forall_{s\in T} \neg \tilde{G}^{-}(s)(t), \tag{4.16}$$

gdzie poszczególne predykaty rozmyte oznacza się tymi samymi symbolami co odpowiadające im zbiory rozmyte. Stosując wzór (4.15) można wzór (4.16) wyrazić następująco:

$$N(T,\tilde{G})(t) \Leftrightarrow T(t) \wedge \forall_{s\in T} \neg \tilde{G}(s,t) \tag{4.17}$$

Określenie wzoru (4.17) dla rozmytego predykatu $T$, który będziemy oznaczać jako $\tilde{T}$, wymaga jedynie rozpisania wystąpienia kwantyfikatora ogólnego *o ograniczonym zakresie* $\forall_{s\in T}$, co prowadzi do następującej formuły:

$$N(\tilde{T},\tilde{G})(t) \Leftrightarrow \tilde{T}(t) \wedge \forall_s \ (\tilde{T}(s) \rightarrow \neg \tilde{G}(s,t)) \tag{4.18}$$

Ostatecznie formułuje sie następującą definicję.

**Definicja 4.4.** Niech $\tilde{T}$ będzie rozmytym zbiorem krotek, a $\tilde{G}$ określoną na nim, rozmytą relacją preferencji. Wtedy *rozmytą operację winnow* $\omega_{\tilde{G}}$ definiuje się następująco:

$$\omega_{\tilde{G}}(\tilde{T})(t) = N(\tilde{T},\tilde{G})(t), \tag{4.19}$$

gdzie predykat rozmyty $N(\tilde{T},\tilde{G})$ jest określony wzorem (4.18).

## 4.3 Zapytania nieprecyzyjne

Omawiając algebrę relacji rozmytych (por. p. 4.1) czy rozmytą wersję operatora *winnow* (por. s. 86) używaliśmy już terminów i relacji lingwistycznych do wyrażania preferencji użytkownika. W niniejszym punkcie przedstawimy bardziej formalnie i szczegółowo koncepcję *zapytań nieprecyzyjnych*, których istota polega na zastosowaniu tych konstrukcji.

W procesie wyszukiwania informacji należy określić zakres i warunki, które powinny spełniać poszukiwane dane. Służą temu *języki zapytań*, omawiane w rozdziale 3, wśród których najszersze zastosowanie znajduje obecnie język SQL. Wspólną cechą klasycznych języków zapytań jest wymóg *precyzyjnego* określania warunków, które poszukiwane dane mają spełniać. Na przykład klient agencji nieruchomości poszukujący

*taniego* domu zmuszony jest precyzyjnie podać interesujący go *przedział* cen. Można jednak zauważyć, że jakkolwiekby nie określić granic tego przedziału, to zawsze dom droższy choćby o złotówkę nie będzie spełniał tak sformułowanego zapytania. Przykład ten wskazuje, że źródłem tych trudności jest konieczność *precyzyjnego* określenia warunków pierwotnie wyrażonych w języku naturalnym z użyciem wysoce *nieprecyzyjnych* pojęć. Czynimy tu analogiczne obserwacje, do tych które poczyniliśmy w p. 2.1 argumentując nieadekwatność klasycznego pojęcia zbioru do reprezentacji stopniowalnych terminów jezyka naturalnego. Będziemy więc postulować włączenie zbiorów rozmytych do repertuaru języka zapytań do bazy danych. W tym celu wprowadzimy następującą definicję *zapytań nieprecyzyjnych*.

**Definicja 4.5.** *Zapytaniem nieprecyzyjnym* do relacyjnej bazy danych nazywamy zapytanie zawierające jawnie użyte *wyrażenia języka naturalnego*, zwane *terminami lingwistycznymi*, określające:

- *nieprecyzyjne* wartości, na przykład "*niskie* wynagrodzenie",
- *nieprecyzyjne* porównania (relacje) między różnymi wartościami, na przykład "wynagrodzenie *znacznie większe niż* 2 000 PLN",
- *niestandardowe* sposoby agregacji stopni spełnienia cząstkowych warunków zapytania, w tym z zastosowaniem *kwantyfikatorów lingwistycznych*, na przykład "*większość ważnych* spośród wymienionych warunków ma być spełniona"

przy czym przyjmuje się, że terminy lingwistyczne występujące w zapytaniu modeluje się z użyciem *logiki rozmytej*.

Przyjmuje się, że poszczególne wiersze *spełniają* zapytanie w pewnym *stopniu*, wyrażonym liczbą z przedziału [0,1], przy czym 1 oznacza *całkowite spełnienie*, a 0 *całkowite niespełnienie* zapytania. Wynikiem zapytania nieprecyzyjnego jest zbiór wierszy *uporządkowany* według stopnia spełnienia zapytania.

Wprowadzone pojęcie *stopnia spełnienia zapytania* pozwala lepiej odzwierciedlić pojmowanie przez człowieka dopasowania danych do zapytania. W przypadku złożonych zapytań, zawierających nieprecyzyjne terminy lingwistyczne oraz niestandardowe schematy agregacji, *spełnianie* ma w sposób naturalny charakter *stopniowy*. Człowiek ocenia w takiej sytuacji dane jako *lepiej* bądź *gorzej* spełniające jego wymagania, a nie tylko dzieli je na *spełniające* i *niespełniające* te wymagania. Pojęcie *stopnia spełnienia* pozwala to formalnie przedstawić. Co więcej, zapewnia

ono naturalne uporządkowanie wyników zapytania. Ułatwia to znacznie analizę wyników w przypadku, kiedy brak jest danych całkowicie spełniających warunki zapytania, ale istnieje wiele spełniających je *do pewnego stopnia*.

Obliczanie stopnia spełniania prostego *zapytania nieprecyzyjnego* można w najogólniejszym zarysie przedstawić na następującym przykładzie.

**Przykład 4.2.** Załóżmy, że poszukuje się *taniego* domu w ofercie agencji nieruchomości, zawartej w tabeli `NIERUCHOMOŚCI`, opisanej w tabl. 3.1. Przyjmijmy, że $U$ oznacza przedział liczbowy, w którym określa się *ceny* nieruchomości (w tysiącach złotych) oraz, że termin *tani* modeluje się z użyciem zbioru rozmytego $F \in \mathcal{F}(U)$ o następującej funkcji przynależności:

$$\mu_F(u) = \begin{cases} 1 & \text{dla } u \leq 200 \\ -0.005u + 2 & \text{dla } 200 < u \leq 400 \\ 0 & \text{dla } u > 400 \end{cases} \tag{4.20}$$

Wtedy dla nieruchomości, reprezentowanej przez krotkę $t$ o cenie $t.cena$, stopień spełnienia $md$ powyższego zapytania oblicza się w następująco:

$$md = \mu_A(\texttt{cena}(t)) \tag{4.21}$$

gdzie $\texttt{cena}(t)$ oznacza wartość atrybutu `cena` dla krotki $t$.

Przykład powyższy ilustruje ogólną zasadę interpretacji zapytań nieprecyzyjnych: *stopień spełnienia zapytania* utożsamia się z *wartością funkcji przynależności* odpowiedniego zbioru rozmytego. W przypadku złożonych zapytań tak obliczone *cząstkowe stopnie spełnienia*, odpowiadające poszczególnym warunkom występującym w zapytaniu, *agreguje się* z użyciem wybranych operatorów reprezentujących poszczególne rozmyte spójniki logiczne (por. p. 2.2.2) i operatorów agregacji (por. p. 2.3.2).

Warto zwrócić uwagę, że nawet wówczas gdy wymagania użytkownika są precyzyjnie określone, zastosowanie zapytań nieprecyzyjnych może być uzasadnione. Dotyczy to sytuacji, w której brak jest danych spełniających wymagania. Wtedy zapytanie z klasycznie, precyzyjnie określonymi warunkami daje w odpowiedzi *pusty* zbiór danych. Natomiast zapytanie o nieprecyzyjnie, nieostro określonych warunkach, może dać niepustą odpowiedź. Co więcej, może się okazać, że niektóre spośród uzyskanych wyników o najwyższym stopniu spełnienia są akceptowane przez użytkownika.

Reasumując, zapytania nieprecyzyjne umożliwiają:

– lepszą reprezentację wymagań użytkownika poprzez możliwość bezpośredniego ich wyrażenia z użyciem *terminów lingwistycznych* i złożonych sposobów *agregacji* warunków cząstkowych,

– uporządkowanie wyników według *stopnia spełnienia*, co znacznie ułatwia ich analizę,

– zmniejszenie ryzyka *pustej odpowiedzi* poprzez rozszerzającą interpretację warunków zapytania.

Należy podkreślić, że badania dotyczące tematyki tak rozumianych *zapytań nieprecyzyjnych* mają już długą historię. Tahani, student Zadeha, twórcy teorii zbiorów rozmytych, już w 1977 roku opublikował pracę [206], w której zaproponował zastosowanie pojęć teorii zbiorów rozmytych dla uelastycznienia zapytań do klasycznych baz danych. W szczególności wysunął postulat użycia terminów nieprecyzyjnych w zapytaniach i ich modelowania za pomocą zbiorów rozmytych. Kontynuację i rozwinięcie tego pomysłu stanowią języki zapytań opracowane w projektach FQUERY for Access ([136, 137, 138, 131, 132, 133]) i SQL*f* [34, 35, 44, 38, 40]. Wśród pionierów zastosowania logiki rozmytej w dziedzinie baz danych wymienić należy, obok twórców języka SQL*f* i systemu FQUERY for Access, również Zemankovą [261, 260, 262], Changa i Ke [59, 60], Bucklesa i Petry'ego [52, 53, 54, 178], Krafta i Buella [145], DeCaluwe [218], Chena i Kerrego [61]. Warto zaznaczyć, że większość z wymienionych autorów zajmowała się tematyką pokrewną do zapytań nieprecyzyjnych, w szczególności *rozmytymi bazami danych* lub zastosowaniem logiki rozmytej do *wyszukiwania informacji tekstowej*.

Warto również wspomnieć o wczesnych pracach poświęconych uelastycznieniu zapytań bez jawnego użycia terminów lingwistycznych. Należy tu wymienić przede wszystkim systemy ARES [122] i VAGUE [167], w których wprowadzono możliwość użycia operatora porównania przybliżonego "$\approx$" w miejsce klasycznego operatora równości. Operator taki można traktować jako odpowiednik terminu lingwistycznego *mniej więcej* czy *około*. Stopień spełnienia warunku oblicza się jako pewną funkcję odległości pomiędzy argumentami. Koncepcja operatora "$\approx$" bliska jest więc zapytaniom nieprecyzyjnym, ale nie wydaje się łatwe jej uogólnienie na przypadek bardziej złożony. Podejście to cieszy się nadal pewnym zainteresowaniem [16].

**System FQUERY for Access** System FQUERY for Access, autorstwa Kacprzyka i Zadrożnego [131], stanowi implementację omawianej tu

koncpecji *zapytań nieprecyzyjnych*. Funkcje systemu FQUERY for Access można określić następująco:

- *wspomaganie tworzenia i modyfikacji słownika terminów lingwistycznych* używanych przy formułowaniu zapytań;
- *wspomaganie konstruowania zapytań nieprecyzyjnych* w ramach interfejsu użytkownika standardowego relacyjnego systemu zarządzania bazą danych (MS Access™) przez udostępnianie słownika terminów lingwistycznych i ich kodowanie w zapytaniach;
- *realizacja zapytań nieprecyzyjnych* przez ich przetworzenie, zapewniające właściwą interpretację występujących w nich terminów lingwistycznych przez standardowy mechanizm wykonywania zapytań systemu zarządzania bazą danych.

Możliwości oferowane przez system FQUERY for Access można więc podsumować jako *modelowanie* (*reprezentację*), *pozyskiwanie* i *przetwarzanie* terminów lingwistycznych do realizacji zapytań nieprecyzyjnych do bazy danych.

**SQL*f*** Język SQL*f* [40, 38] został opracowany przez zespół Patricka Bosca. Celem było opracowanie rozszerzonej składni języka SQL uwzględniającej możliwość użycia terminów lingwistycznych, relacji rozmytych i innych operatorów rozmytych wszędzie tam, gdzie stwarza to możliwości bardziej intuicyjnego wyszukiwania danych. Jednocześnie ważnym założeniem konstrukcyjnym tego rozszerzenia było zachowanie równoważności zapytań występujące w klasycznej wersji języka SQL. Utrzymanie takich równoważności jest ważne, choćby ze względu na ich możliwe użycie przez algorytmy optymalizacji realizacji zapytań.

Od samego początku twórcy języka SQL*f* poświęcali wiele uwagi efektywnej realizacji zapytań tak rozszerzonego języka. Tak więc obok samej koncepcji języka opracowano również koncepcję tak zwanych *zapytań pochodnych* (ang. *derived queries*) oraz specjalizowanych struktur danych (indeksów), które mogłyby przyspieszyć realizacje zapytań sformułowanych z użyciem języka SQL*f* [35]. Próby pilotowej implementacji opisano w [35]. Elementy języka SQL*f* odnaleźć można też w implementacji innego systemu opisanego w [44].

ISSN 0208-8029
ISBN 83-894-7551-0

INSTYTUT BADAŃ SYSTEMOWYCH
POLSKIEJ AKADEMII NAUK
tel.: (+48) 22 3810246 / 22 3810277 / 22 3810241 / 22 3810273
e-mail: biblioteka@ibspan.waw.pl