

Jerzy GRZYWOCZ

METODY OPISU BAZ DANYCH JAKO PODSTAWA AUTOMATYZACJI PROCESU WYSZUKIWANIA

Streszczenie. Artykuł przedstawia opracowaną przez Autora metodę wspomagającą wyszukiwanie w relacyjnej bazie danych widzianej jako relacja uniwersalna. Metoda ta umożliwia jednoznaczną odpowiedź na zapytanie do baz danych o cyklicznym lub niespójnym hipergrafie. Ideą metody jest tworzenie wirtualnego, acyklicznego schematu bazy danych przez przemianowanie atrybutów.

THE METHODS OF DATABASES DESCRIBING AS THE BASE OF THE SEARCHING PROCESS AUTOMATION

Summary. This paper presents the Author's method facilitating a searching process in the relational database viewed as universal relation. This method enables the univocal answer to the query to the databases having the cyclic or disconnected hypergraphs. Presented method consist in creating the virtual, acyclic database schema through changing the attribute names.

LES MÉTHODES DE LA DESCRIPTION DE BASES DE DONNÉES COMME BASE DE AUTOMATION DE PROCÉDÉ DE RECHERCHE

Resumé. L'article présent une méthode, proposé par l'auteur, d'assistance de recherche dans une base de données vue comme une relation universelle. Cette méthode rends possible une réponse à une question à base de donnés avec une hypergraphe cyclique et non-cohérente. L'idée de la méthode est une création de virtuel acyclique schéma de base de données par rédénomination des attributs.

1. Wstęp

Użytkowanie każdej bazy danych wiąże się z wyszukiwaniem przechowywanych w niej informacji. Wyszukiwanie inicjowane jest zadaniem pytania. Pytanie musi być wyrażone w jakimś języku, zwanym językiem zapytań.

Zdarza się, że model danych nie zawiera pełnej informacji zapewniającej poprawność wyszukiwania. W procesie wyszukiwania potrzebne są również informacje co prawda "zaszyte" w strukturze bazy, ale często w sposób rozproszony lub niejawny. Istnieją także więzy, których nie da się wbudować w struktury bazy. Zbiór tych informacji tworzy *opis bazy danych*.

Im mniej informacji o bazie danych musi umieścić w pytaniu użytkownik, tym bardziej złożony musi być opis tej bazy i tym bardziej złożone muszą być algorytmy interpretujące pytanie w aspekcie istniejącego opisu bazy. Opis bazy danych nadaje bowiem kontekst pytaniu. Dopiero suma informacji z pytania i z opisu bazy pozwala zinterpretować pytanie w sposób najbliższy intencjom pytającego.

Analizując dokładniej problem zawartości pytania i jego kontekstu dla relacyjnych baz danych można podjąć próbę formalizacji tego zagadnienia. Autor proponuje wyróżnienie zbioru elementów, które trzeba określić systemowi wyszukiwania (w pytaniu czy w opisie bazy) lub zakodować w jego modułach, aby system ten mógł odpowiedzieć na zapytanie. Elementy te można podzielić na elementy celu i elementy kontekstu. Elementy celu to informacja, co ma zawierać odpowiedź. Elementy kontekstu to opis warunków nałożonych na rozwiązanie zadania wyszukiwania, struktur danych, w których przebiega wyszukiwanie oraz algorytmów uzyskania odpowiedzi na zapytanie.

Przez *automatyzację procesu wyszukiwania* Autor rozumie przeniesienie ze sfery działań użytkownika w sferę działań systemu jak największego podzbioru operacji związanych z wyszukiwaniem. Należy przy tym zaznaczyć, że jeżeli operacje te podzielić na:

- operacje związane z formułowaniem zapytania,
- operacje związane z wyznaczeniem odpowiedzi,

to prace Autora koncentrowały się na pierwszym z wymienionych rodzajów operacji.

Automatyzacja procesu wyszukiwania jest powiązana z poziomem języka zapytań w danym systemie: im mniej elementów kontekstu w pytaniu, tym wyższy poziom języka zapytań w sensie jego przyjazności dla użytkownika.

Autora interesowały koncepcje systemów wyszukiwania potrzebujące minimalnej liczby elementów kontekstu w pytaniu. W niniejszej pracy pokazano, że relacja uniwer-

salna jest narzędziem spełniającym ten warunek. W uproszczeniu koncepcja relacji uniwersalnej polega na takim widzeniu danych całej bazy, jak gdyby były one zgromadzone w jednej tablicy, zwanej *relacją uniwersalną* [4, 8, 9, 11, 13, 14]. Istnieje kilka koncepcji wyszukiwania w relacji uniwersalnej. Każda z nich wymaga odpowiednio dobranej metody opisu bazy. Najtrudniejszym etapem wyszukiwania w relacji uniwersalnej jest wyznaczenie tzw. okna. Jest to hipotetyczna relacja, której schematem jest zbiór atrybutów użytych w pytaniu (oznaczany X), natomiast zawartość jest wyznaczana jako wartość wyrażenia, którego argumentami są relacje bazy danych tworzących relację uniwersalną. Okno jest oznaczane przez $[X]$.

Koncepcje wyszukiwania w relacji uniwersalnej różnią się przede wszystkim postacią wyrażenia wyznaczającego $[X]$. Autor zajmował się tworzeniem opisu bazy danych dla przedstawionych w dalszej części pracy koncepcji fragmentacji bazy i koncepcji złączeń rozszerzonych. Dla obu tych koncepcji istnieją bazy danych, dla których z powodu niespełnienia założenia o jednoznaczności nazw atrybutów można wskazać kilka różnych dróg szukania odpowiedzi (co może prowadzić do odpowiedzi niejednoznacznych). Brak spełnienia tego założenia Autor nazwał *niejednoznacznością atrybutów*.

Artykuł opisuje wyniki rozważań Autora nad możliwościami automatyzacji procesu wyszukiwania dla różnych poziomów języka zapytań ([2]). Największe możliwości automatyzacji wyszukiwania Autor widzi na poziomie relacji uniwersalnej. Składnia zadań wyszukiwania (czyli pytań kierowanych do bazy danych) na tym poziomie abstrahuje bowiem od schematu bazy danych i algorytmów interpretacji zapytań. Pytanie obejmuje tylko nazwy atrybutów szukanych i opis warunków, a wyrażenie wyznaczające rozwiązanie jest tworzone algorytmicznie na podstawie zbioru atrybutów wzmiankowanych w pytaniu oraz informacji znajdujących się w opisie bazy. Należy tu nadmienić, że wspomniane własności języka tego poziomu sprawiają, iż język relacji uniwersalnej może być wygodnym językiem pośrednim pomiędzy procesorami języków zapytań bazujących na grafice lub składni bardzo zbliżonej do języka naturalnego a systemem wyszukiwania wykorzystującym język zapytań niższego poziomu (np. rachunku predykatów lub algebry relacji).

2. Podstawowe pojęcia relacyjnych baz danych

2.1. Struktury modelu relacyjnego

Artykuł dotyczy relacyjnego modelu danych ([5, 10, 12, 13, 14]). W modelu tym dane gromadzone są w dwuwymiarowych tablicach, zwanych *relacjami*. Cechy obiektów świata rzeczywistego odpowiadają w modelu relacyjnym tzw. *atrybutom*. Wartości atrybutów zapamiętywane są w kolumnach tablic, natomiast nazwy atrybutów tworzą nagłówki kolumn. Zbiór atrybutów danej relacji tworzy *schemat relacji*. Relację r o schemacie R oznacza się $r(R)$. Wiersz tablicy nosi nazwę *krotka*. Zapis $t[A]$ oznacza wartość atrybutu A w krotce t .

Zawartość krotki relacji o schemacie $A_1 A_2 \dots A_n$ oznacza się (a_1, a_2, \dots, a_n) , gdzie a_i ($i \in \{1, \dots, n\}$) jest wartością atrybutu A_i w tej krotce.

Schemat bazy danych R to zbiór schematów relacji tworzących bazę danych r .

2.2. Podstawowe operacje na relacjach

W modelu relacyjnym operacje na relacjach mogą być realizowane m. in. za pomocą operatorów algebry relacji. Najczęściej używane są: operator selekcji, projekcji i złączenia naturalnego, które dla danych relacji $r(R)$, $s(S)$ oraz atrybutu $A \in R$ i zbioru atrybutów $X \subseteq R$ są zdefiniowane następująco:

- selekcja: $\sigma_{A=a}(r) = r'(R) = \{t \in r \mid t[A] = a\}$
- projekcja: $\pi_X(r) = r'(X) = \{t[X] \mid t \in r\}$
- złączenie: $r(R) \bowtie s(S) = q(R \cup S) =$
 $\{t \mid t_r \in r \wedge t_s \in s \wedge t[R] = t_r \wedge t[S] = t_s \wedge t_r[R \cap S] = t_s[R \cap S]\}.$

2.3. Podstawowe zależności wiążące atrybuty relacji

Dla przedstawienia tezy pracy istotne są niżej zdefiniowane zależności.

Zależność funkcyjna $f : X \rightarrow Y$ jest spełniona w relacji $r(R)$, gdzie X, Y należą do R , wtedy i tylko wtedy, gdy dla każdych dwóch krotek $t_1, t_2 \in r$, dla których $t_1[X] = t_2[X]$, zachodzi również $t_1[Y] = t_2[Y]$.

Zależność złączeniowa $\bowtie(R_1, R_2, \dots, R_n)$ występuje w $r(R)$, gdzie $R_1 \cup R_2 \cup \dots \cup R_n = R$, wtedy i tylko wtedy, gdy

$$r = \pi_{R_1}(r) \bowtie \pi_{R_2}(r) \bowtie \dots \bowtie \pi_{R_n}(r).$$

Jeżeli powyższa równość jest prawdziwa, to złączenie $\pi_{R_1}(r) \bowtie \pi_{R_2}(r) \bowtie \dots \bowtie \pi_{R_n}(r)$ określane jest jako *złączenie odwracalne* lub *złączenie bez strat*.

Klucz w relacji $r(R)$ jest to taki podzbiór atrybutów $K \subseteq R$, że dla każdych dwóch krotek $t_1, t_2 \in r$ zachodzi $t_1[K] \neq t_2[K]$ i żaden podzbiór $K' \subset K$ nie posiada tej własności.

3. Wyszukiwanie w relacyjnej bazie danych. Rola opisu bazy w automatyzacji wyszukiwania

W bazach danych można wyróżnić m.in. następujące operacje:

- aktualizację,
- wyszukiwanie,
- przetwarzanie (np. z użyciem operacji arytmetycznych),
- raportowanie.

Najczęściej przyjmuje się, że język zapytań obejmuje wyszukiwanie i raportowanie danych. Wyszukiwanie inicjowane jest postawieniem zadania wyszukiwania. Terminowi "zadanie wyszukiwania" równoważne są terminy: pytanie, zapytanie, kwerenda.

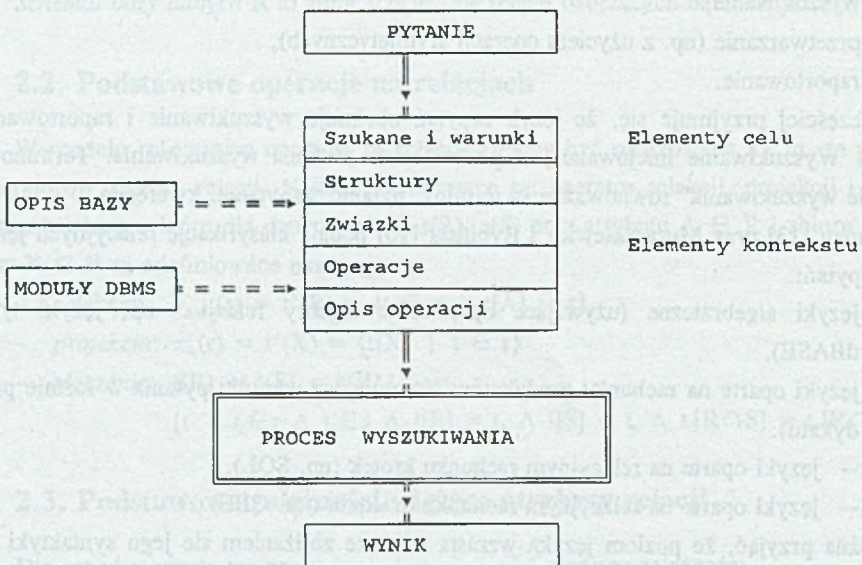
Ullman [13] oraz Muraszkwicz i Rybiński [10] podają klasyfikację relacyjnych języków zapytań:

- języki algebraiczne (używające operatorów algebry relacji - np. języki typu dBASE),
- języki oparte na rachunku predykatów (formułujące warunki pytania w formie predykatu):
 - języki oparte na relacyjnym rachunku krotek (np. SQL),
 - języki oparte na relacyjnym rachunku dziedzin (np. QBE).

Można przyjąć, że poziom języka wzrasta wraz ze zbliżaniem się jego syntaktyki do języka naturalnego. Nie chodzi tu oczywiście o język potoczny. Już dawno zauważono, że język naturalny musi być poddany pewnej formalizacji, aby mógł służyć do ścisłego wyrażania potrzebnych sformułowań (np. zadań wyszukiwania czy twierdzeń matematycznych). Raczej należałoby powiedzieć, że poziom języka zapytań jest tym wyższy, im bardziej pojęcia, którymi on operuje, są zbliżone do pojęć, jakimi posługuje się człowiek, formułując zadanie wyszukiwania.

Autor proponuje wyróżnienie zbioru elementów, które trzeba określić systemowi wyszukiwania (w pytaniu, w opisie bazy lub w procedurach systemu), aby system ten mógł odpowiedzieć na zapytanie:

- E1) **szukane i warunki**, czyli atrybuty, których wartości są poszukiwane oraz warunki nałożone na rozwiązanie,
- E2) **struktury**, czyli relacje, których dotyczy zadanie oraz, jeśli to konieczne, pliki indeksowe związane z tymi relacjami,
- E3) **związki**, czyli powiązania między relacjami wymienionymi w zadaniu wyszukiwania,
- E4) **operacje**, czyli wskazanie operacji na relacjach, które trzeba wykonać, aby otrzymać rozwiązanie,
- E5) **opis operacji**, czyli opis operacji na relacjach z dokładnością do przetwarzania pojedynczych krotek.



Rys. 1. Elementy konieczne w procesie wyszukiwania

Fig. 1. The Elements which are Necessary in Searching Process

Element E1 okreśmy jako *element celu*, natomiast każdy z elementów E2, E3, E4, E5 jako *element kontekstu*. Im wyższy poziom języka zapytań, tym mniej elementów kontekstu trzeba określać w pytaniu. Stąd, im wyższy poziom języka zapytań, tym więcej tych elementów musi być zrealizowanych przez system zarządzania bazą danych automatycznie, czyli bez udziału użytkownika.

Biorąc pod uwagę powyższe rozważania, można w ślad za [2, 6] wyróżnić 4 poziomy języków zapytań (w nawiasach elementy, które musi zawierać pytanie):

- poziom operowania krotkami (elementy $E1, \dots, E5$),
- poziom algebry relacji (elementy $E1, \dots, E4$),
- poziom rachunku predykatów (elementy $E1, \dots, E3$),
- poziom relacji uniwersalnej (element $E1$).

4. Relacja uniwersalna jako podstawa prostego języka zapytań

4.1. Pojęcie relacji uniwersalnej

Relację uniwersalną można zdefiniować dla relacyjnej bazy danych r składającej się z relacji r_1, r_2, \dots, r_n o schematach R_1, R_2, \dots, R_n . Przyjmijmy, że $U = R_1 \cup R_2 \cup \dots \cup R_n$. *Relacja uniwersalna* jest to hipotetyczna relacja u o schemacie U , utworzona z relacji r_1, r_2, \dots, r_n .

Mechanizm relacji uniwersalnej ma na celu między innymi uproszczenie zapisu zadania wyszukiwania oraz uwolnienie użytkownika od konieczności analizy struktury bazy danych w celu wyboru relacji potrzebnych w rozwiązaniu. Zapis zadania wyszukiwania dla relacji uniwersalnej zawiera jedynie element celu, czyli nazwy atrybutów szukanych oraz warunki nałożone na rozwiązanie. Mechanizm relacji uniwersalnej jest więc podstawą języka zapytań jeszcze bardziej przyjaznego dla użytkownika niż języki oparte na rachunku predykatów, takie jak np. SQL. Treść pytania do relacji uniwersalnej jest bowiem niezależna od schematu bazy. Użytkownik nie musi znać ani nazw relacji ani atrybutów, których wartości wiążą krotki z różnych relacji.

Zapis zadania wyszukiwania w języku zapytań na poziomie relacji uniwersalnej ma postać:

`select Q where Warunki(P),` (1)

gdzie:

- Q - ciąg nazw atrybutów, których wartości są szukane,
- $Warunki(P)$ - wyrażenie logiczne, operujące na zbiorze atrybutów P .

Zapis ten oznacza: wyszukaj wartości atrybutów Q zawarte w krotkach, które zawierają wartości atrybutów ze zbioru Q i które spełniają warunki nałożone na atrybuty P . Zbiór wszystkich atrybutów wymienionych w pytaniu oznacza się przez X , czyli $X = Q \cup P$.

Dla potrzeb relacji uniwersalnej schemat bazy wygodnie jest opisywać za pomocą hipergrafu ([8, 9, 13, 14]). Przykłady hipergrafów baz danych przedstawia rys. 2.

4.2. Wyznaczenie okna podstawą wyszukiwania w relacji uniwersalnej

Ullman [14] proponuje algorytm rozwiązywania zadania wyszukiwania dla relacji uniwersalnej, bazujący na pojęciu okna.

Okno [X] dla zbioru atrybutów X wymienionych w pytaniu jest to relacja o schemacie X i zawartości równej wartości wyrażenia, którego argumentami są relacje bazy danych, tworzące relację uniwersalną.

Algorytm 1. Wyznaczanie rozwiązania $q(Q)$ zadania wyszukiwania postaci (1) ze str. 7:

select Q where Warunki(P).

1. Określ zbiór atrybutów $X = Q \cup P$ wymienionych w pytaniu.
2. Wyznacz wartość [X].
3. $q(Q) = \pi_Q(\sigma_{\text{Warunki}(P)}[X])$

Wyznaczanie rozwiązania zadania wyszukiwania polega więc na znalezieniu odpowiedzi na zapytanie w relacji [X] (krok 3 algorytmu). Zasadniczym zagadnieniem całego procesu wyszukiwania jest wyznaczenie [X]. Podstawowe metody wyznaczania [X] bazują na pojęciu złączeń rozszerzonych (Sagiv [11]) lub na pojęciu obiektów (Ullman [13, 14]).

4.2.1. Metoda złączeń rozszerzonych

Metoda rozszerzonych złączeń (zwana dalej ZR) opisana przez Sagiva ([11]) bazuje na pojęciu tzw. *egzemplarza charakterystycznego* relacji uniwersalnej. Sagiv podał algorytm tworzenia takiego egzemplarza i pokazał, że w egzemplarzu charakterystycznym spełnione są wszystkie zależności funkcyjne zdefiniowane w bazie r wtedy i tylko wtedy, gdy $(\forall R_i \in R) R_i$ spełnia warunek jednoznaczności.

Przyjmijmy, że F_i oznacza zbiór wszystkich zależności funkcyjnych zachodzących w relacji r_i ($i \in \{1, \dots, n\}$), natomiast $F = F_1 \cup F_2 \cup \dots \cup F_n$. Schemat R_i spełnia *warunek jednoznaczności*, jeśli dla każdego $j \neq i$ ($i, j = 1 \dots n$) domknięcie $(R_i)_{F-F_j}^+$ nie zawiera żadnego nadklucza schematu R_j . Jeżeli domknięcie R_i zawiera nadklucze R_j , to atrybuty niekluczowe w tych nadkluczach określane są dalej jako *atrybuty niejednoznacznie zależne* od R_j . Zbiór Y jest nadkluczem schematu R_j , jeśli Y zawiera K oraz A takie, że K jest kluczem dla R_j oraz $A \in R_j - K$. *Domknięcie zbioru atrybutów X względem zbioru zależności F*, oznaczane X_F^+ , jest to zbiór wszystkich atrybutów $A_i \in U$, dla których

zależność $X \rightarrow A_j$ może być wyprowadzona z F za pomocą aksjomatów Armstronga (Sagiv [11], Ullman [14], Kozielski [5]).

Do wyznaczenia $[X]$ na podstawie charakterystycznego egzemplarza relacji uniwersalnej wykorzystuje się pojęcie rozszerzonego złączenia (ang. extension join).

Wyrażenie $R_{i_1} \bowtie R_{i_2} \bowtie \dots \bowtie R_{i_m}$ jest *rozszerzonym złączeniem* schematu R_{i_1} , jeśli dla każdego $1 \leq j < m$ zbiór $R_{i_1} \cup R_{i_2} \cup \dots \cup R_{i_j}$ zawiera klucz schematu $R_{i_{j-1}}$. Dla wyznaczenia $[X]$ interesujące są rozszerzone złączenia pokrywające (zawierające) występujący w pytaniu podzbiór atrybutów X . Minimalnym rozszerzonym złączeniem pokrywającym X jest takie rozszerzone złączenie $E = R_{i_1} \bowtie R_{i_2} \bowtie \dots \bowtie R_{i_m}$, że nie istnieje żadne inne rozszerzone złączenie schematu R_{i_1} pokrywające X , którego argumenty byłyby właściwym podzbiorem zbioru schematów $\{R_{i_1}, R_{i_2}, \dots, R_{i_m}\}$. Jeśli wszystkie schematy relacji spełniają warunek jednoznaczności, to wyrażenie wyznaczające $[X]$ jest sumą projekcji na X wszystkich minimalnych złączeń rozszerzonych pokrywających X :

$$[X] = \pi_X(E_1) \cup \pi_X(E_2) \cup \dots \cup \pi_X(E_p). \quad (2)$$

4.2.2. Metoda obiektów

Ullman ([13, 14]) koncepcję relacji uniwersalnej opiera na pojęciu obiektu. *Obiekt* jest to minimalny zbiór atrybutów powiązany elementarnym (nierozkładalnym) związkim. Relację uniwersalną opisuje wyrażenie:

$$u = \{a_1, a_2, \dots, a_m \mid P_1(O_1) \wedge P_2(O_2) \wedge \dots \wedge P_p(O_p)\}, \quad (3)$$

gdzie:

a_1, a_2, \dots, a_m – wartości atrybutów bazy,

$P_i(O_i)$ – predykat opisujący związek między atrybutami i -tego obiektu.

Wyrażenie (3) opisuje relację uniwersalną pod warunkiem, że wszystkie obiekty spełniają zależność złączeniową opisaną w punkcie 2.3. Możliwe jest ([7, 4, 8]) przyjęcie upraszczającego założenia, że obiektami są relacje.

Algorytm 2. Wyznaczenie odpowiedzi na zapytanie metodą obiektów.

1. Zredukuj zbiór obiektów bazy danych do najmniejszego, spójnego zbioru obiektów $\{O_{i_1}, O_{i_2}, \dots, O_{i_m}\}$, w którym są zachowane wszystkie atrybuty z X i zastąp go złączeniem

$$J = O_{i_1} \bowtie O_{i_2} \bowtie \dots \bowtie O_{i_m} \text{ tych obiektów.}$$

2. Wyznacz $[X]$ ze wzoru: $[X] = \pi_X(J)$.

5. Wyszukiwanie w relacji uniwersalnej – problemy

Przy wyszukiwaniu w relacji uniwersalnej mogą powstać następujące problemy:

- brak spełnienia warunku jednoznaczności przez schematy relacji w bazie danych.
- niejednoznaczność atrybutów,
- cykliczność bazy danych,
- niespójność bazy danych.

Problemy wynikające z wystąpienia każdego z tych elementów można określić jako *problemy niejednoznaczności*.

Problemy może również stwarzać syntaktyka języka zapytań.

5.1. Problem braku spełnienia warunku jednoznaczności

Brak spełnienia przez któryś ze schematów bazy danych warunku jednoznaczności (p. 4.2.1) może w relacji uniwersalnej prowadzić do naruszenia zależności funkcyjnych zdefiniowanych w relacjach bazy danych, a tym samym do trudnych w interpretacji wyników wyszukiwania.

Przykład. Rozważmy bazę danych r : $r_{AB}(\underline{AB}) = \{(a_1, b_1)\}$, $r_{BC}(\underline{BC}) = \{(b_1, c_1)\}$, $r_{AD}(\underline{AD}) = \{(a_1, d_1)\}$, $r_{DC}(\underline{DC}) = \{(d_1, c_2)\}$. Istniejące klucze relacji implikują m. in. zależność: $A \rightarrow C$. Warunku jednoznaczności nie spełnia m. in. relacja r_{AB} , ponieważ zakładając, że F to zbiór wszystkich zależności funkcyjnych bazy r , F_{CD} , to zbiór zależności w relacji r_{CD} , mamy

$(AB)_{F-F_{CD}}^+ = ABCD$, a CD jest nadkluczem relacji r_{CD} . W relacji uniwersalnej (konstruowanej np. metodą opisaną przez Sagiva [11]) wartość a_1 atrybutu A jest skojarzona z dwoma wartościami atrybutu C , tzn. c_1 i c_2 , co narusza zależność $A \rightarrow C$. \square

5.2. Problem niejednoznaczności nazw atrybutów

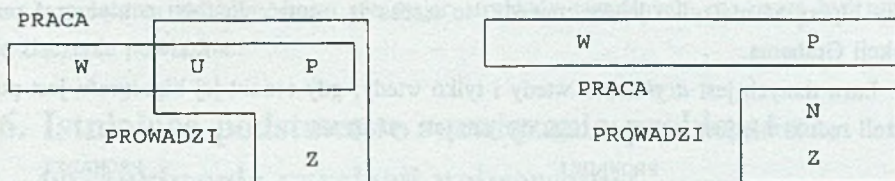
Jednym z podstawowych założeń dotyczących relacji uniwersalnej jest założenie jednoznaczności nazw atrybutów. Mówi ono, że atrybuty oznaczające to samo powinny mieć te same nazwy, natomiast atrybuty oznaczające co innego muszą różnić się nazwami (zgodność syntaktyczna musi pociągać za sobą zgodność semantyczną). Założenie to jest potrzebne do automatycznego znajdowania powiązań między relacjami przez system zarządzania bazą danych. Istotę tego założenia można zilustrować na przykładzie bazy danych, składającej się z relacji $PRACA(\underline{Pracownik}, \underline{Wydział})$ i $PROWADZI(\underline{Pracownik}, \underline{Zajęcia})$. Przyjmijmy, że P oznacza atrybut Pracownik, W – Wydział, Z – Zajęcia. Atry-

but P (Pracownik) oznacza zarówno w relacji PRACA, jak i w relacji PROWADZI identyfikator pracownika, dlatego też nazwa tego atrybutu w obu relacjach jest taka sama. Jeżeli pytanie dotyczy atrybutów $X \subseteq R_1, R_2, \dots, R_m$, to przy wyszukiwaniu w relacji uniwersalnej przyjmuje się, że powiązanie danych z relacji $r_1(R_1), \dots, r_m(R_m)$ następuje poprzez złączenie naturalne $r_1 \bowtie r_2 \bowtie \dots \bowtie r_m$. Jest to złączenie według równych wartości atrybutów mających takie same nazwy w rozważanych relacjach. Spełnienie założenia jednoznaczności nazw atrybutów daje więc semantyczną poprawność wyniku złączenia. Na przykład, dla pytania:

`select Zajęcia where Wydział = 'Mechaniczny'`

konieczne jest powiązanie danych z relacji PRACA i PROWADZI. Nastąpi to po dokonaniu złączenia PRACA \bowtie PROWADZI. Wynik tego złączenia jest sensowny, ponieważ zarówno w relacji PRACA, jak i w relacji PROWADZI atrybut P ma takie samo znaczenie. Brak spełnienia założenia jednoznaczności nazw atrybutów Autor proponuje określać jako *niejednoznaczność nazw atrybutów* lub krótko *niejednoznaczność atrybutów*. Niejednoznaczność nazw atrybutów można podzielić na dwa przypadki:

- dwa atrybuty oznaczają co innego, lecz tak samo się nazywają,
- dwa atrybuty znaczą to samo, ale inaczej się nazywają.



Rys. 2. a) Atrybut niejednoznaczny: U. b) Atrybuty tożsame: P, N
Fig. 2. a) The Ambiguous Attribute: U. b) The Equivalent Attributes: P, N

Przykładem przypadku a) (rys. 2a) jest dodanie do schematów relacji PRACA i PROWADZI atrybutu o nazwie Uwagi (U), przy czym w relacji PRACA atrybut ten oznacza uwagi o pracowniku, natomiast w relacji PROWADZI oznacza uwagi o przedmiotach. Pytanie

`select Zajęcia where Wydział = 'Mechaniczny'`

zadane do bazy z rys. 2a prowadzi do złączenia $r_{PUW}(PUW) \bowtie r_{PUZ}(PUZ)$. Wynik tego złączenia jest semantycznie trudny do interpretacji, ponieważ złączane będą te krotki z relacji PRACA i PROWADZI, które mają taką samą wartość nie tylko atrybutu P, ale i atrybutu U. Oznacza to, że do wyniku przejdą tylko te krotki relacji PRACA i PROWADZI, w których uwagi o pracowniku i o przedmiocie są identyczne! Atrybuty, które nazywają się tak samo, ale posiadają różne znaczenie (takie jak Uwagi, które mają

znaczenie: "uwagi o pracowniku" i "uwagi o przedmiocie"), określane są w pracy jako *atrybuty niejednoznaczne*.

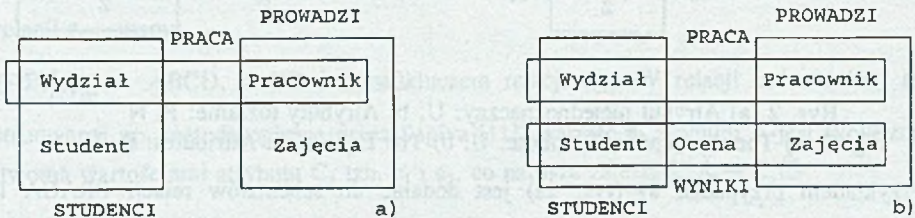
Przypadek b) można zilustrować, przyjmując w relacji PROWADZI (wbrew założeniu jednoznaczności nazw atrybutów), że pracownik określony jest atrybutem Nauczyciel_akademicki (N) a nie Pracownik (rys. 2b). Wówczas w systemie nie byłoby możliwe automatyczne powiązanie ze sobą danych z relacji $r_{PW}(PW)$ oraz $r_{NZ}(NZ)$. Atrybuty znaczące to samo, ale różnie nazywające się (takie jak Nauczyciel_akademicki i Pracownik), określane są w pracy jako *atrybuty tożsame*.

5.3. Problem cykliczności bazy danych

Ze względu na strukturę hipergrafu bazy danych można podzielić na cykliczne i acykliczne. Istnieje wiele definicji cykliczności ([5, 13]). Ullman ([13, 14]) definiuje cykliczność, korzystając z pojęcia redukcji Grahama.

Redukcja Grahama jest to proces kolejnego usuwania z hipergrafu krawędzi odpowiadających takim schematom R_i , że dla każdego $R_{j \neq i} \in R$ każdy atrybut z $R_i - R_j$ należy tylko do R_i . Proces kończy się, gdy w hipergrafie nie ma krawędzi do usunięcia. Gdy hipergraf ma tylko jedną krawędź, to trzeba ją usunąć. *Redukt* to hipergraf po redukcji Grahama.

Baza danych jest *acykliczna* wtedy i tylko wtedy, gdy redukt jej hipergrafu jest pusty. Jeżeli redukt nie jest pusty, to baza danych jest *cykliczna*.



Rys. 3. Hipergraf baz danych: a) acyklicznej i b) cyklicznej

Fig. 3. The Hipergraph of the Databases: a) Acyclic, and b) Cyclic

Cykliczność bazy powoduje, że pytanie select Q where Warunki(P) może być rozwiązane na kilka sposobów, co narusza założenie o jednoznacznej interpretacji zadania wyszukiwania. Rozważmy cykliczną bazę danych z rys. 3b. Pytanie:

select Zajęcia where Wydział = 'Mechaniczny'

zadane do bazy z rys. 3b może być rozumiane na dwa sposoby.

A. Podać nazwy zajęć prowadzonych dla studentów Wydziału Mechanicznego.

B. Podać nazwy zajęć prowadzonych przez pracowników Wydziału Mechanicznego.

Każda z tych interpretacji wymaga złączenia innych relacji dla udzielenia odpowiedzi. Interpretacja pytania zgodna z życzeniem pytającego jest możliwa dopiero po uzupełnieniu pytania select Q where Warunki(P) dodatkowymi informacjami lub założeniami.

5.4. Problem niespójności bazy danych

Niespójność bazy danych oznacza w pracy istnienie takich zbiorów relacji P_1, P_2, \dots, P_m w bazie danych $r(\mathbf{R})$, że dla każdego z nich, jeżeli przez $\text{Atr}(P)$ oznaczyć sumę atrybutów wszystkich relacji ze zbioru relacji P , zachodzi:

$$\text{Atr}(\mathbf{R} - P_i) \cap \text{Atr}(P_i) = \emptyset, \text{ gdzie } i \in \{1, \dots, m\}.$$

Hipergraf bazy danych składa się wówczas z kilku rozłącznych części. Zbiory relacji P_1, P_2, \dots, P_m takie, jak opisane powyżej, określane są w pracy jako *rozłączne podzbiory bazy*.

Jeżeli baza danych jest niespójna i nie istnieje spójny podzbiór bazy zawierający całkowicie zbiór X atrybutów użytych w pytaniu (jest on podzielony między kilka rozłącznych podzbiorów bazy), to odpowiedź na to pytanie nie jest możliwa. Jest tak dlatego, ponieważ w relacji uniwersalnej dane z różnych relacji wiązane są przez złączenie naturalne, a rozłączne podzbiory bazy, nie mając atrybutów wspólnych, nie mogą być przez takie złączenie powiązane.

6. Istniejące podstawowe rozwiązania problemów wyszukiwania w relacji uniwersalnej

Problemy niejednoznaczności mogą być rozstrzygane przez omówione metody wyznaczania okna lub ich rozwinięcia.

6.1. Metoda złączeń rozszerzonych

Metoda złączeń rozszerzonych próbuje rozstrzygnąć problem cykliczności poprzez sumowanie wszystkich częściowych odpowiedzi na zapytanie. Odpowiedzi częściowe są określane jako projekcja na X minimalnego złączenia rozszerzonego pokrywającego X . Metoda uniemożliwia złączenie dwóch relacji (a więc wpływ jednej z nich na rozwiązanie częściowe), jeżeli to złączenie nie jest złączeniem rozszerzonym, nawet gdy jest ono poprawne według innych kryteriów. Ogranicza to z jednej strony wpływ administratora systemu na kształtowanie rozwiązań częściowych, z drugiej zaś jednak - gwarantuje pytającemu formalną poprawność wszystkich złączeń.

Metoda wymaga spełnienia warunku jednoznaczności przez wszystkie schematy bazy danych, a przynajmniej przez wszystkie schematy relacji tworzących odpowiedź na zapytanie. Wymaga to albo odpowiedniego projektu bazy danych, albo wprowadzenia takich więzów integralności, które zapewnią, że w egzemplarzu charakterystycznym relacji uniwersalnej zależności funkcyjne bazy danych nie zostaną naruszone.

Problem niejednoznaczności nazw atrybutów metoda złączeń rozszerzonych rozwiązuje pośrednio albo uniemożliwiając wyszukiwanie (jeżeli skutkiem niejednoznaczności nazw atrybutów istnieją schematy relacji nie spełniające warunku jednoznaczności), albo wyszukując w bazie cyklicznej (jeżeli wynikiem niejednoznaczności nazw atrybutów jest cykliczność bazy).

6.2. Metoda obiektów maksymalnych jako rozwinięcie metody obiektów

Metoda obiektów maksymalnych (zwana dalej metodą OM) przedstawiona jest przez Ullmana w [13] i [14]. Bazuje ona na przedstawionym już w p. 4.2.2 pojęciu obiektu.

Obiekty maksymalne są to acykliczne podzbiory M_1, M_2, \dots, M_m zbioru obiektów określonych w relacyjnej bazie danych $r(R)$. Podzbiory te muszą spełniać kilka własności, z których najistotniejsza polega na tym, że w każdym obiekcie maksymalnym wszystkie obiekty muszą spełniać zależność złączeniową.

Metoda OM umożliwia interpretację zadania wyszukiwania w bazach cyklicznych. Ogranicza ona zakres pytania w tego typu bazach do acyklicznego zbioru obiektów tworzących obiekt maksymalny. Zakłada się, że odpowiedź na zapytanie jest możliwa tylko wtedy, gdy zbiór X atrybutów użytych w pytaniu zawiera się całkowicie w którymś z obiektów maksymalnych bazy (takich obiektów maksymalnych może być kilka).

Autor przyjął wspomniane w p. 4.2.2 upraszczające założenie, że obiektami są relacje. Przy tym założeniu obiekty maksymalne nazywane są fragmentami bazy, a metoda obiektów maksymalnych - metodą fragmentacji bazy (nazywaną dalej FB).

Algorytm 3. Wyznaczenie odpowiedzi na zapytanie metodą FB.

1. Znajdź te fragmenty $F_{i_1}, F_{i_2}, \dots, F_{i_x}$ należące do bazy r , które zawierają X .
2. Zredukuj każdy fragment F_{i_j} ($j = 1 \dots x$) do najmniejszego spójnego zbioru relacji, w którym są zachowane wszystkie atrybuty z X i zastąp go złączeniem J_{i_j} tych relacji.
3. Wyznacz $[X]$ ze wzoru:

$$[X] = \pi_X(J_{i_1}) \cup \pi_X(J_{i_2}) \cup \dots \cup \pi_X(J_{i_x}).$$

Metoda fragmentacji bazy (obiektów maksymalnych) rozstrzyga problemy wyboru drogi wyszukiwania odpowiedzi poprzez podział bazy danych na fragmenty (obiekty maksymalne) i generację dla każdego z nich rozwiązania częściowego. Zakres wyszukiwania inicjowanego pytaniem jest więc ograniczony do pojedynczych fragmentów (obiektów maksymalnych). W zależności od przyjętych założeń odpowiedzi częściowe można sumować lub podawać oddzielnie. W odróżnieniu od metody złączeń rozszerzonych nie narzuca się konkretnego sposobu dokonywania złączeń przy doborze relacji do obiektów maksymalnych. Jedynym kryterium jest odwracalność złączeń. Metoda proponuje automatyczną generację obiektów maksymalnych, ale dopuszcza również dowolne ich kształtowanie przez administratora bazy. Daje to większą elastyczność przy tworzeniu odpowiedzi, ale zwiększa ryzyko błędnej nawigacji w bazie.

Metoda pomija problem niejednoznaczności schematów relacji w bazie danych.

Problem niejednoznaczności nazw atrybutów jest rozwiązywany pośrednio. Jeżeli z powodu niejednoznaczności nazw atrybutów baza danych jest cykliczna, to jest dzielona na obiekty maksymalne i wyszukiwanie jest realizowane według algorytmu 3 ze str. 14.

7. Rozwiązanie problemów wyszukiwania w relacji uniwersalnej proponowane przez Autora

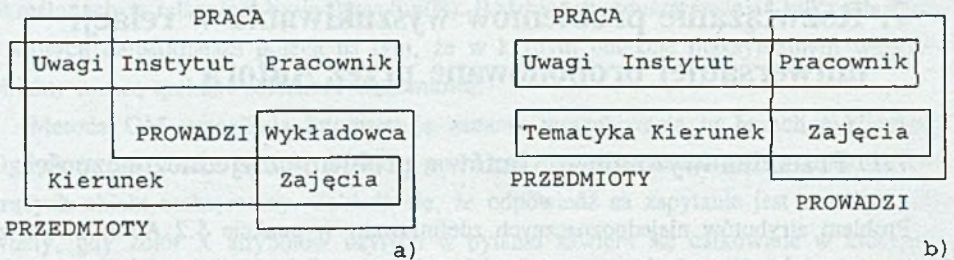
7.1. Przemianowywanie atrybutów a problemy niejednoznaczności

Problem atrybutów niejednoznacznych zdefiniowany w punkcie 5.2 Autor uważa za kluczowy, ponieważ - zdaniem Autora - większość przypadków braku spełnienia warunku jednoznaczności przez schematy relacji w bazie danych, cykliczności bazy oraz jej niespójności ma swoje źródło w niejednoznaczności atrybutów. Literatura dotycząca relacji uniwersalnej zawiera propozycje rozwiązania przede wszystkim wspomnianych wyżej skutków niejednoznaczności atrybutów. Jednakże usuwanie skutków istniejących już zjawisk, takich jak: brak spełnienia warunku jednoznaczności przez schematy relacji w bazie danych, cykliczność bazy lub jej niespójność powoduje komplikację procesu tworzenia opisu bazy danych, zwiększenie złożoności procesu wyszukiwania i najczęściej wzrost złożoności zapytań.

Usunięcie niejednoznaczności nazw atrybutów często doprowadza do usunięcia cykliczności bazy danych lub do spełnienia warunku jednoznaczności przez schematy bazy danych. Niejednoznaczność nazw atrybutów można usunąć, przemianowując atrybuty w schematach relacji tak, aby atrybuty znaczące to samo tak samo się nazywały, natomiast

atrybuty mające różne znaczenia różniły się również nazwami. Jednakże rzeczywiste przemianowanie atrybutów można zrealizować jedynie na etapie projektowania bazy danych, ponieważ manipulowanie schematem już funkcjonującej bazy może prowadzić do błędnego działania istniejących programów użytkowych napisanych dla obsługi tej bazy. Dlatego Autor proponuje przemianowywanie atrybutów poprzez definiowanie wirtualnego schematu bazy. Polega to na zdefiniowaniu tzw. *synonimów*, czyli nowych, zastępczych nazw atrybutów. Użytkownik relacji uniwersalnej operuje wówczas w procesie wyszukiwania jej schematem wirtualnym (z przemianowanymi atrybutami), a wyszukiwanie realizowane jest w rzeczywistych relacjach. *Wirtualny schemat bazy* jest to definiowany dla potrzeb systemu wyszukiwania (poprzez wprowadzenie synonimów do schematów relacji) schemat bazy danych, na którym operuje jedynie system wyszukiwania. Rzeczywisty schemat bazy danych pozostaje bez zmian.

Ponieważ atrybuty niejednoznacznie zależne (p. 4.2.1) oraz atrybuty zadeklarowane kolizyjnie (tzn. mające w różnych relacjach te same nazwy, lecz inne typy wartości) często są atrybutami wymagającymi przemianowania, więc Autor zaproponował algorytmy wspomagające użytkownika w odszukiwaniu tych atrybutów w schematach relacji bazy danych.



Rys. 4. a) Atrybut niejednoznaczny Uwagi oraz atrybuty tożsame: Pracownik i Wykładowca. b) Wirtualny schemat bazy po wstawieniu synonimów
Fig. 4. The Example of Synonyms

Przykład usunięcia niejednoznaczności atrybutów przez wprowadzenie synonimów obrazuje rys. 4. Po lewej stronie rysunku znajduje się hipergraf bazy danych z atrybutem niejednoznacznym Uwagi (ta sama nazwa w różnych relacjach, lecz inne znaczenie) oraz atrybutami tożsamymi: Pracownik i Wykładowca (różne nazwy, lecz to samo znaczenie). Prawa strona pokazuje hipergraf tej samej bazy danych po wprowadzeniu synonimów. W relacji PRZEDMIOTY nazwę Uwagi zastąpiono nazwą Tematyka, a w relacji PROWADZI nazwę Wykładowca zastąpiono nazwą Pracownik.

Przemianowanie atrybutów jest jednym z etapów definiowania opisu bazy. Cały proces tego definiowania obejmuje dokonanie przez użytkownika wyboru relacji (wraz z ich kluczami) tworzących relację uniwersalną oraz ewentualnego (wspomagane go przez system

wyszukiwania) przemianowania atrybutów, a następnie automatyczne wyznaczenie fragmentów (obiektów maksymalnych) według zaproponowanych przez Autora algorytmów.

Utworzony w ten sposób opis bazy jest podstawą procesu wyszukiwania w relacji uniwersalnej.

7.2. Graficzny tryb tworzenia opisu bazy i zadawania pytań

Graficzna komunikacja systemu z użytkownikiem jest ze swej natury bardziej przyjazna dla człowieka niż komunikacja poprzez tekst. Stąd wielu autorów zajmuje się tym zagadnieniem również w aspekcie relacji uniwersalnej. Strukturę bazy można reprezentować na sposób syntaktyczny lub semantyczny. Syntaktyczna reprezentacja struktury relacyjnej bazy danych polega na pokazaniu użytkownikowi schematów relacji wraz z kluczami i takimi atrybutami, które mogą mieć dla użytkownika szczególne znaczenie (np. atrybuty, wg których będą dokonywane złączenia przy wyszukiwaniu w relacji uniwersalnej). Semantyczna reprezentacja relacyjnej bazy danych (np. diagramy ER [12, 13]) pokazuje związki znaczeniowe między danymi.

Dlatego też Autor do graficznej komunikacji systemu z użytkownikiem przyjął reprezentowanie bazy danych w formie jej hipergrafu. Podejście to jest stosowane zarówno przy zadawaniu pytań, jak i przy definiowaniu opisu bazy danych. Na etapie tworzenia opisu bazy schematy relacji są bowiem potrzebne do zaznaczenia w nich kluczy i ewentualnego przemianowania atrybutów. W czasie zadawania pytań schematy relacji wizualizują związki pomiędzy atrybutami. Użytkownik operuje jedynie na atrybutach (schematy relacji mają znaczenie pomocnicze), wskazując kursorem atrybuty szukane lub atrybuty, dla których (po ich wskazaniu) precyzuje warunki wyszukiwania.

7.3. Równoważność metod złączeń rozszerzonych i obiektów maksymalnych – możliwość optymalnego wyboru metody

Autor pokazał, że metoda złączeń rozszerzonych daje odpowiedzi na zapytania równoważne odpowiedziom generowanym przez jeden z wariantów metody obiektów maksymalnych. W wariacie tym obiekty maksymalne (fragmenty) wyznacza się, biorąc pod uwagę tylko zależności funkcyjne.

Oznaczając przez $[X]_{ZR}$ okno wyznaczone metodą złączeń rozszerzonych, natomiast przez $[X]_{FB, ZF}$ okno wyznaczone metodą obiektów maksymalnych (fragmentów bazy) tworzonych przy użyciu zależności funkcyjnych, Autor sformułował i wykazał następujące twierdzenie.

Twierdzenie. Dla metod złączeń rozszerzonych i fragmentacji bazy prawdziwa jest równość:

$$[X]_{ZR} = [X]_{FB,ZP} \quad (4)$$

Z twierdzenia tego można wyciągnąć dwa wnioski.

Wniosek 1. W istniejącym systemie wyszukiwania opartym na koncepcji fragmentacji bazy można otrzymywać wyniki wyszukiwania identyczne z wynikami metody złączeń rozszerzonych (bez implementacji złożonych algorytmów tej metody).

Wniosek 2. Zarówno system oparty na koncepcji złączeń rozszerzonych, jak i oparty na koncepcji obiektów maksymalnych mogą generować równoważne wyniki wyszukiwania. □

Użytkownik, rozważywszy uwagi o metodach wyszukiwania w relacji uniwersalnej przedstawione w punktach 6.1 i 6.2, może dokonać optymalnego dla siebie wyboru. Może więc np. wybrać system oparty na koncepcji złączeń rozszerzonych (nie wymagający jawnego operowania dodatkowymi strukturami takimi, jak obiekty maksymalne i zapewniający z definicji metody odwracalność złączeń w rozwiązaniu zadania wyszukiwania), otrzymując wyniki wyszukiwania nie różniące się od wyników uzyskanych metodą obiektów maksymalnych.

8. Implementacja wyników pracy

8.1. UNIVERS – system wyszukiwania w relacji uniwersalnej

Rezultaty badań przedstawione w artykule zaimplementowano w systemie UNIVERS. Jest to system zapytań oparty na koncepcji relacji uniwersalnej stworzony w Instytucie Informatyki Politechniki Śląskiej.

Wersja podstawowa systemu (UNIVERS 1.0), której głównymi elementami są moduł wyszukiwania i moduł definiowania opisu bazy (tworzący i aktualizujący plik opisu bazy), ma trzy warianty: wariant oparty na koncepcji fragmentacji bazy (opisany w [7, 8]), wariant oparty na koncepcji złączeń rozszerzonych i wariant będący próbą użycia uproszczonego języka naturalnego jako języka zapytań (opisany w [1]).

Rozwinięciem wersji podstawowej jest UNIVERS 2.0, o rozbudowanych możliwościach modułu wyszukiwania i modułu definiowania opisu bazy. Moduł definiowania bazy umożliwia tworzenie wirtualnego schematu bazy danych poprzez przemianowywanie atrybutów. Współpraca z użytkownikiem oparta na systemie okien i menu jest poszerzona w

stosunku do wersji podstawowej o możliwość zadawania pytań i tworzenia opisu bazy przez operowanie na hipergrafie bazy.

Idea współpracy z modulem wyszukiwania jest identyczna we wszystkich wersjach systemu UNIVERS. Ogólna postać pytania jest typowa dla relacji uniwersalnej:

```
select Szukane where Warunki.
```

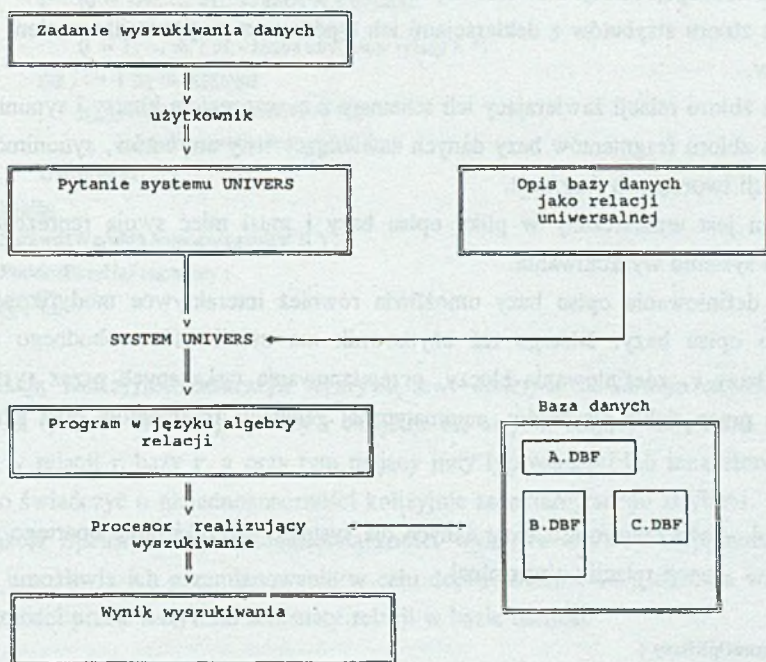
"Szukane" to lista atrybutów tworzących schemat relacji wynikowej. "Warunki" to zbiór warunków, które muszą spełniać krotki tej relacji.

Ogólna idea definiowania opisu bazy jest również podobna dla wszystkich wariantów systemu. Definiowanie polega przede wszystkim na wskazaniu plików z danymi, dla których system UNIVERS tworzy relację uniwersalną. Następnie (w zależności od wariantu systemu i schematu bazy) użytkownik wykonuje dalsze czynności (np. wskazanie kluczy relacji).

Autor zaprojektował i zaimplementował (lub bezpośrednio kierował implementacją) moduł definiowania bazy dla wszystkich wersji systemu UNIVERS.

Specyfikacja zewnętrzna systemu UNIVERS została opisana w [8].

Rysunek 5 ilustruje kolejne etapy procesu wyszukiwania w systemie UNIVERS.



Rys. 5. Ilustracja kolejnych etapów interpretacji pytań w systemie UNIVERS
Fig. 5. The Stages of the Query Interpretation in the System UNIVERS

8.2. Tworzenie opisu bazy danych w systemie UNIVERS

Opis bazy danych zapewnia optymalny dla użytkownika (w warunkach stwarzanych przez relację uniwersalną) tryb współpracy z systemem wyszukiwania.

Dane wejściowe dla modułu definiowania opisu bazy są następujące:

- zbiór relacji $\{r_1(R_1), \dots, r_n(R_n)\}$ tworzących relację uniwersalną,
- dla każdej relacji r_i ($i \in \{1, \dots, n\}$) zbiór kluczy K_i zawierający m_i elementów, gdzie $K_i = \{K_{i1}, \dots, K_{im_i}\}$, a $K_{ij} \subseteq R_i$ jest j -tym kluczem relacji r_i ($j \in \{1, \dots, m_i\}$),
- dla każdej relacji r_i ($i \in \{1, \dots, n\}$) zbiór atrybutów przemianowanych $P_i \subseteq R_i$ wraz z przypisanym każdemu atrybutowi ze zbioru P_i synonimem,
- przyjęte kryterium odwracalności złączeń.

Moduł definiowania opisu bazy, korzystając z powyższych danych wejściowych, generuje opis bazy umieszczany w wewnętrznych strukturach danych systemu wyszukiwania z możliwością zapisania go w pliku. Generację opisu bazy opisuje algorytm 4.

Według koncepcji przyjętej przez Autora opis bazy danych dla systemu wyszukiwania opartego na koncepcji relacji uniwersalnej musi zawierać:

- opis zbioru atrybutów z deklaracjami ich typów oraz przyporządkowaniem synonimów,
- opis zbioru relacji zawierający ich schematy z zaznaczeniem kluczy i synonimów,
- opis zbioru fragmentów bazy danych zawierający listy atrybutów, synonimów oraz relacji tworzących fragment.

Opis ten jest umieszczany w pliku opisu bazy i musi mieć swoją reprezentację w strukturach systemu wyszukiwania.

Moduł definiowania opisu bazy umożliwia również interaktywne modyfikowanie otrzymanego opisu bazy. Dlatego też użytkownik ma możliwość swobodnego dodania relacji do bazy r , zdefiniowania kluczy, przemianowania wskazanych przez system lub wybranych przez siebie atrybutów, automatycznej generacji fragmentów oraz ich modyfikacji.

Algorytm 4. Tworzenie opisu bazy danych dla systemu wyszukiwania opartego na koncepcji relacji uniwersalnej.

procedura TworzOpisBazy ;

(*

dane wprowadzane przez użytkownika

- zbiór relacji $\{r_1(R_1), \dots, r_n(R_n)\}$ tworzących relację uniwersalną,

- dla każdej relacji r_i ($i \in \{1, \dots, n\}$) zbiór kluczy K_i zawierający m_i elementów, gdzie $K_i = \{K_{i1}, \dots, K_{im_i}\}$, a $K_{ij} \subseteq R_i$ jest j -tym kluczem relacji r_i ($j \in \{1, \dots, m_i\}$),
- dla każdej relacji r_i ($i \in \{1, \dots, n\}$) zbiór atrybutów przemianowanych $P_i \subseteq R_i$ wraz z przypisanym każdemu atrybutowi ze zbioru P_i synonimem,
- symbol przyjętego kryterium odwracalności złączeń.

struktury globalne

ZbiórKryteriówOdwracalności = { ZależnościFunkcyjne, ZależnościWielowart,
ZależnościWielowartIFunkcyjne } ;

Fragmenty : ZbiórFragmentów ; (* zbiór fragmentów bazy danych*)

R : ZbiórSchematówRelacji ; (* schemat bazy danych tworzącej relację uniwersalną*)

KryteriumOdwracalności : ZbiórKryteriówOdwracalności ;

*)

początek

dla $i := 1$ do n wykonuj

początek

ZnajdźPlikZRelacją(r_i) ;

OdczytajSchemat(R_i) ;

(* $R_i = \{A_1, \dots, A_{m_i}\}$, gdzie A_j - atrybut,

($j = 1, \dots, m_i$; m_i - liczba atrybutów relacji r_i *)

dla $j := 1$ do m_i wykonuj

jeżeli KolizyjnaDeklaracja(A_j) to

jeżeli OdrzucićRelację to idź do E ;

UstalKlucze(r_i) ;

E: koniec

SprawdźWarunekJednoznaczności(R) ;

PodzielBazęNaFragmenty ;

koniec ; □

Funkcja **KolizyjnaDeklaracja** wykrywa tzw. *kolizyjną deklarację atrybutów*, która polega na tym, że w relacji r_i bazy r znajduje się atrybut mający taką samą nazwę, jak atrybut w relacji r_j bazy r , a przy tym mający inny typ wartości lub inną szerokość pola. Może to świadczyć o niejednoznaczności kolizyjnie zadeklarowanego atrybutu.

Funkcja **SprawdźWarunekJednoznaczności** wykrywa atrybuty niejednoznacznie zależne i umożliwia ich przemianowanie w celu doprowadzenia do spełnienia warunku jednoznaczności przez wszystkie schematy relacji w bazie danych.

Algorytm 5. Sprawdzanie warunku jednoznaczności dla schematów relacji ze zbioru S i wykrywanie atrybutów niejednoznacznie zależnych w tym zbiorze relacji.

procedura SprawdźWarunekJednoznaczności(S : ZbiórSchematówRelacji);

(* Schemat R_i spełnia warunek jednoznaczności, jeśli dla każdego $j \neq i$ ($i, j \in \{1, \dots, n\}$, gdzie n - liczba rozważanych relacji) domknięcie $(R_i)^+$ nie zawiera żadnego nadklucza schematu R_j .

*)

struktury

D : ZbiórAtrybutów ;

A : Atrybut ;

użyte podprogramy

funkcja Atr (M : ZbiórSchematówRelacji) : ZbiórAtrybutów ;

(*zwraca zbiór atrybutów zawartych w schematach relacji zbioru M*)

procedura UsuńTymczasowoZaznaczenieKluczy(P : SchematRelacji) ;

(* Usuwa tymczasowo zaznaczenie kluczy schematu P *)

procedura PrzywróćZaznaczenieKluczy(P) ;

(* Przywraca zaznaczenie kluczy usunięte procedurą UsuńTymczasowoZaznaczenieKluczy *)

funkcja Domknięcie (Y : ZbiórAtrybutów) : ZbiórAtrybutów ;

(* Wyznacza domknięcie Y^+ zbioru atrybutów Y (parz str. ?), gdzie F jest zbiorem zależności funkcyjnych bazy R określonych przez zdefiniowanie kluczy relacji.

Zastosowanie: jeżeli atrybut $A \in Y^+$, to zależność $Y \rightarrow A$ jest wyprowadzalna z F, a więc jest prawdziwa w bazie R *)

procedura PrzemianujAtrybut(A : Atrybut) ;

(* Umożliwia nadanie synonimu atrybutowi A *)

początek (* SprawdźWarunekJednoznaczności *)

dla każdego $\{R\} \in S$ wykonuj

dla każdego $P \neq R$ wykonuj

początek

UsuńTymczasowoZaznaczenieKluczy(P) ;

D := Domknięcie(R) ;

PrzywróćZaznaczenieKluczy(P) ;

dla każdego klucza $K \in P$ wykonuj

jeżeli $((D \cap \text{Atr}(P)) - K) \neq \emptyset$ i $(K \subseteq D)$ to

początek

Pisz('Schemat ', R, ' nie spełnia warunku jednoznaczności');

Pisz('W schemacie ', P, ' atrybuty ', $D \cap (\text{Atr}(P) - K)$,

' są niejednoznacznie zależne od ', R) ;

dla każdego $A \in (D \cap (\text{Atr}(P) - K))$ wykonuj

PrzemianujAtrybut(A) ;

koniec

koniec

koniec

Procedura **PodzielBazęNaFragmenty** umożliwia automatyczny podział bazy danych na fragmenty na podstawie istniejących w bazie zależności funkcyjnych i wielowartościowych. Zależności te wynikają z podanych przez użytkownika kluczy relacji.

Algorytm 6. Automatyczny podział bazy danych na fragmenty.

procedura PodzielBazęNaFragmenty ;

(*

Wejście:

- zbiór wirtualnych schematów relacji R tworzących relację uniwersalną,
- klucze każdej relacji $r(R)$, gdzie $R \in R$,
- identyfikator kryterium odwracalności złączeń (opisane w rozdz. ?).

W wyjście:

- zbiór fragmentów bazy; opis każdego fragmentu zawiera zbiór identyfikatorów relacji tworzących fragment oraz zbiór identyfikatorów atrybutów tych relacji.

struktury globalne

ZbiórKryteriówOdwracalności = { ZależnościFunkcyjne, ZależnościWielowart,
ZależnościWielowartIFunkcyjne } ;

Fragmenty : ZbiórFragmentów ; (*zbiór fragmentów bazy danych*)

R : ZbiórSchematówRelacji ; (*schemat bazy danych tworzącej relację uniwersalną*)

KryteriumOdwracalności : ZbiórKryteriówOdwracalności

*)

struktury lokalne

Fragment : ZbiórSchematówRelacji ; (*fragment bazy danych*)

P, R : SchematRelacji ; (*schematy relacji bazy danych*)

użyte podprogramy

funkcja Atr(M : ZbiórSchematówRelacji) : ZbiórAtrybutów ;

(*zwraca zbiór atrybutów zawartych w schematach relacji zbioru M*)

procedura UsuńTymczasowoZeSchematów(X : ZbiórRelacji) ;

(* Usuwa chwilowo ze schematów relacji bazy r zbiór atrybutów X *)

procedura PrzywróćWSchematach(X : ZbiórAtrybutów) ;

(* Przywraca w schematach relacji atrybuty tymczasowo usunięte procedurą
UsuńTymczasowoZeSchematów *)

procedura UsuńFragmentyBędącePodzbióramiInnych ;

początek (* PodzielBazęNaFragmenty *)

Fragmenty := Φ ;

dla każdego $R \in \mathbf{R}$ wykonuj

początek

Fragment := $\{R\}$;

dońki Fragment zmienia się wykonuj

dla każdego $P \in \mathbf{R}$ wykonuj

jeżeli

$\neg (\{P\} \in \text{Fragment}) \wedge$

MożnaDołączyć(Fragment, P, KryteriumOdwracalności)

to

Fragment := Fragment \cup $\{P\}$;

Fragmenty := Fragmenty \cup $\{\text{Fragment}\}$;

koniec ;

UsuńFragmentyBędącePodzbioramiInnych ;

koniec (* PodzielBazęNaFragmenty *) ;

funkcja MożnaDołączyć(S, Q : ZbiórSchematówRelacji,

KryteriumOdwracalności: KryteriaOdwracalności) : Logiczna ;

początek

MożnaDołączyć := falsz ;

jeżeli \neg ZłączenieOdwracalne(S, Q, KryteriumOdwracalności) to powrót ;

jeżeli Cykliczny(S \cup Q) to powrót ;

MożnaDołączyć := prawda ;

koniec ;

Funkcja MożnaDołączyć sprawdza, czy zbiory schematów relacji S i Q mogą się znaleźć w jednym fragmencie bazy danych. Przyjęto, że S i Q mogą być w jednym fragmencie, jeżeli złączenie $S \bowtie Q$ jest odwracalne i (co jest nowością w stosunku do algorytmu Ullmana [13, 14]) jeżeli hipergraf złożony z S i Q jest acykliczny.

Funkcja ZłączenieOdwracalne umożliwia badanie odwracalności złączenia według jednego z trzech kryteriów:

- kryterium zależności funkcyjnych: złączenie $S \bowtie Q$ jest odwracalne, jeżeli $(\text{Atr}(S) \cap \text{Atr}(Q)) \rightarrow \text{Atr}(S)$ lub $(\text{Atr}(S) \cap \text{Atr}(Q)) \rightarrow \text{Atr}(Q)$;
- kryterium zależności wielowartościowych: złączenie $S \bowtie Q$ jest odwracalne, jeżeli $(\text{Atr}(S) \cap \text{Atr}(Q)) \twoheadrightarrow \text{Atr}(S) - \text{Atr}(Q)$ lub $(\text{Atr}(S) \cap \text{Atr}(Q)) \twoheadrightarrow \text{Atr}(Q) - \text{Atr}(S)$.

(za Ullmanem [13] przyjmuje się, że zależności te zachodzą, gdy S i Q są rozłączne w hipergrafie po usunięciu z hipergrafu ich części wspólnej);

- kryterium zależności funkcyjnych i wielowartościowych: złączenie $S \bowtie Q$ jest odwracalne, jeżeli przy oznaczeniach

$$P = \text{Atr}(S) \cap \text{Atr}(Q),$$

$$R_{S-Q} = \text{Atr}(S) - \text{Atr}(Q),$$

$$R_{Q-S} = \text{Atr}(Q) - \text{Atr}(S)$$

zachodzi

$$((P \rightarrow \text{Atr}(S)) \text{ lub } (P \rightarrow \text{Atr}(Q)))$$

lub

$$(((P \rightarrow R_{S-Q}) \text{ lub } (P \rightarrow R_{Q-S})) \text{ i } (\neg((R_{S-Q} \rightarrow P) \text{ i } (R_{Q-S} \rightarrow P))))$$

9. Podsumowanie

Często przyjmuje się, że poziom języka zapytań wzrasta wraz ze zbliżaniem się jego syntaktyki do języka formalnego o semantyce i składni bliskiej językowi naturalnemu. Poziom języka zapytań jest więc tym wyższy, im bardziej pojęcia, którymi on operuje, są zbliżone do pojęć, jakimi posługuje się człowiek, formułując zadanie wyszukiwania.

Analizując dokładniej problem struktury pytania i jego kontekstu, Autor podjął próbę formalizacji tego zagadnienia w aspekcie możliwości automatyzacji procesu wyszukiwania dla różnych poziomów języków zapytań relacyjnych baz danych. Wynikiem jego rozważań jest sugestia, że najwyższy poziom automatyzacji wyszukiwania można osiągnąć, stosując model relacji uniwersalnej (rozdz. 4)

Na podstawie badania nad relacją uniwersalną prowadzone w Instytucie Informatyki został zrealizowany system wyszukiwania UNIVERS, bazujący na koncepcji relacji uniwersalnej. Autor, tworząc moduł definiowania opisu bazy w podstawowej wersji tego systemu, dokonał implementacji ogólnikowo opisanych w literaturze algorytmów ([4, 9, 13, 14]) tworzenia opisu bazy dla koncepcji obiektów maksymalnych (fragmentacji). Opis tej implementacji znajduje się w rozdz. 8.

Prowadząc prace badawcze i implementacyjne, Autor zauważył niedogodności istniejących metod opisywania bazy danych jako relacji uniwersalnej. Autor wziął pod uwagę trzy metody wyznaczania rozwiązania zadań wyszukiwania w relacji uniwersalnej: metodę listy wyrażeń relacyjnych ([14]), złączeń rozszerzonych ([11]) oraz obiektów maksymalnych ([13]) wraz jej modyfikacją określaną jako metoda fragmentacji bazy ([8]). Metody te opracowane w celu eliminacji problemów związanych z wyszukiwaniem w relacji uniwersalnej (przede wszystkim cykliczności) nie rozwiązują opisanego w rozdz. 5.2 problemu niejednoznaczności atrybutów. Problem ten Autor uważa za kluczowy, ponieważ -

zdaniem Autora – większość przypadków braku spełnienia warunku jednoznaczności przez schematy relacji w bazie danych, cykliczność bazy oraz jej niespójność ma swoje źródło w niejednoznaczności atrybutów. Autor zaproponował własną metodę wspomagającą wyszukiwanie w relacji uniwersalnej. Ideą tej metody jest przemianowywanie atrybutów w schematach relacji. Polega ono na zmianie nazwy wybranych atrybutów jedynie dla potrzeb systemu wyszukiwania. Prowadzi to do stworzenia wirtualnego schematu bazy danych. Przemianowywanie atrybutów umożliwia przede wszystkim usunięcie niejednoznaczności atrybutów, co może doprowadzić do spełnienia warunku jednoznaczności przez wszystkie schematy relacji w bazie danych oraz zlikwidować cykliczność i niespójność bazy. Koncepcję wyszukiwania w relacji uniwersalnej Autor proponuje oprócz na metodzie fragmentacji bazy. Algorytm tworzenia opisu bazy dla tej metody zamieszczony jest w rozdz. 8.2. Algorytm ten m. in. umożliwia wskazanie użytkownikowi zbioru atrybutów, które należy przemianować, aby doprowadzić do spełnienia warunku jednoznaczności przez wszystkie schematy bazy danych oraz opisuje automatyczną generację fragmentów bazy. Dzięki temu możliwe jest wyszukiwanie w bazach zawierających relacje, których schematy nie spełniają warunku jednoznaczności lub w bazach cyklicznych bez jakichkolwiek modyfikacji rzeczywistego schematu bazy lub jej zawartości. Przemianowywanie atrybutów w bazie cyklicznej prowadzące do usunięcia cykliczności może w niektórych przypadkach uprościć zadawanie pytań do relacji uniwersalnej.

Przedstawione w artykule prace Autora miały na celu wykonanie kolejnego kroku w kierunku ułatwienia użytkownikom korzystania z baz danych. Uzyskane rezultaty mogą stać się podstawą do dalszych prac nad stworzeniem nowych generacji systemów zapytań, na przykład wiążących własności modelu relacji uniwersalnej z własnościami innych modeli danych (np. obiektowego ([3]).

LITERATURA

- [1] Boruta A., Grzywocz J., Kozielski S.: Wykorzystanie elementów języka naturalnego w systemie wyszukiwania opartym na modelu relacji uniwersalnej, ZN Pol. Śl. s. Informatyka, z. 27, Gliwice 1994.
- [2] Grzywocz J. : Własności języka zapytań a opis bazy danych. ZN Pol. Śl. s. Informatyka, z. 25, Gliwice 1994.
- [3] Jarvelin, K.; Niemi, T.: An entity-based approach to query processing in relational databases. I. Entity type representation. Data & Knowledge Engineering, Vol: 10, Iss: 2, March 1993, s. 117-50.

- [4] Korth H. F. i inni: System/U: A Database System Based on the Universal Relation Assumption, ACM Transactions on Database Systems, Vol. 9, No. 3, September 1984, s. 331-347.
- [5] Kozielski S.: Projektowanie struktury logicznej relacyjnych baz danych metodą syntezy przy uwzględnieniu zadań wyszukiwania danych. ZN Pol. Śl, s. Informatyka, z. 11, Gliwice 1988.
- [6] Kozielski S.: Języki zapytań w mikrokomputerowych bazach danych. Referat na seminarium Instytutu Informatyki Pol. Śl. (materiały nieopublikowane), marzec 1992.
- [7] Kozielski S., Piec J., Grzywocz J.: System wyszukiwania oparty na modelu relacji uniwersalnej. Archiwum Informatyki Teoretycznej i Stosowanej, 1993 (w druku).
- [8] Bach M., Grzywocz J., Kozielski S., Piec J., Skowronek M., Starzewska-Karwan E.: System wyszukiwania i system analizy statystycznej w relacyjnych bazach danych. Skrypt Pol. Śl. nr 1814, red.: Kozielski S., Skowronek M., Gliwice 1993.
- [9] Maier D., Ullman J.: Maximal Object and the Semantics of Universal Relation databases. ACM Transactions on Database Systems, Vol. 8, No. 1, March 1983, s. 1-14.
- [10] Muraszekiewicz M., Rybiński H.: Bazy danych. Akademicka Oficyna Wydawnicza RM, Warszawa 1993.
- [11] Sagiv Y.: A Characterization of Globally Consistent Databases and Their Correct Access Paths, ACM Transactions on Database Systems, Vol.8, No.2, June 1983, s. 266 - 286.
- [12] Tsichritzis D. C., Lochovsky F. H.: Modele danych. WNT, Warszawa 1990.
- [13] Ullman J. D.: Systemy baz danych. WNT, Warszawa 1988.
- [14] Ullman J. D.: Principles of Database and Knowledge-Base Systems, Computer Science Press, New York, 1989.

Recenzent: Prof. dr hab. inż. Mieczysław Muraszekiewicz

Wpłynęło do Redakcji 14 lipca 1995 r

Abstract

This paper presents the Author's method facilitating a searching process in the relational database. The description of this method follow after the analysis of the query structure, and of the query context. This analysis discovers the possibility of the query processing automation for all query language levels in the relational databases. The results of this analysis suggest that the universal relation model gives the highest level of query processing automation. For this model the Author proposes the method enabling a univocal answer to the query to databases having a cyclic or disconnected hipergraph. This method consists in creating a virtual, acyclic database schema through renaming the attributes.

The Author presented two methods of query processing in the universal relation: the extension joins method, and the maximal object method with its modification which is referred to the database fragmentation method.

The Author founded his method of changing the attribute names upon the database fragmentation method, and he suggested the algorithm of the creation of the database description for his method.

LITERATURA

- [1] J. Grzywocz, J. Kozłowski, S. Kozłowski: *Systemy Bazy Danych*, Wydawnictwo Naukowe PWN, Warszawa, 1987.
- [2] J. Grzywocz, J. Kozłowski, S. Kozłowski: *Systemy Bazy Danych*, Wydawnictwo Naukowe PWN, Warszawa, 1987.
- [3] J. Grzywocz, J. Kozłowski, S. Kozłowski: *Systemy Bazy Danych*, Wydawnictwo Naukowe PWN, Warszawa, 1987.