ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1994

Seria: INFORMATYKA z. 25 Nr kol. 1236

Jerzy GRZYWOCZ

WŁASNOŚCI JĘZYKA ZAPYTAŃ A OPIS BAZY DANYCH

Streszczenie. Artykuł jest przeglądem współzależności poziomu języka zapytań i złożoności opisów baz danych. Wspomnianą współzależność opisano dla następujących poziomów języka zapytań: operowania krotkami, algebry relacji, rachunku predykatów oraz relacji uniwersalnej. Dla relacji uniwersalnej przedstawiono wpływ sposobu opisywania bazy na rozwiązanie zadania wyszukiwania.

QUERY LANGUAGE CHARACTERISTICS AND A DATABASE DESCRIPTION

Summary. This paper is a review of an interdependece of a query language level and of a complexity of a database description. This interdependence was presented for the following query language levels: a tuple manipulation level, a relational algebra level, a domain and tuple relational calculus, and an universal relation. For an universal relation, an influence of a method of a database description on a query results was presented.

LES PROPRIÉTÉS DE LANGAGE DE QUESTIONS ET DESCRIPTION DE BASES DE DONNÉS

Résumé. L'article est une présentation d'une corrélation entre le niveau de langage de questions et la complexité des descriptions de bases de données. Cette corrélation est décrite pour les suivants niveaux de langage de questions:

Pracę wykonano w ramach grantu KBN nr 8 0269 91 01.

manipulation des tuples, algèbre de relations, calcul de prédicats et aussi pour relation universelle. Pour cette dernière il est présenté l'influence de la manière de description de bases de données sur la résolution de problème de recherche.

1. Wstęp

Celem artykułu jest przegląd poziomów języka zapytań w relacyjnych bazach danych pod kątem współzależności osiąganego poziomu języka zapytań i złożoności opisów struktur w wybranych systemach relacyjnych baz danych.

Im wyższy jest poziom języka zapytań, tym prostsze jest dla użytkownika sformułowanie zadania wyszukiwania. Idealny byłby język zapytań, w którym pytanie sformułowane w *języku naturalnym* zawierałoby jedynie niezbędne informacje o tym, *co* ma zawierać odpowiedź i *jakie warunki* powinna spełniać. Wszystkie potrzebne informacje o bazie danych, które da się ustalić przed zadaniem pytania (tj. nazwy plików z danymi, format danych, powiązania pomiędzy danymi) powinny być dostępne w formie opisu bazy programowi odpowiadającemu na pytanie.

Obecnie prowadzone są liczne prace mające na celu zbudowanie systemu zbliżonego do tego ideału. Do pokonania są jednak liczne trudności. Im mniej bowiem informacji o bazie danych będzie w pytaniu użytkownika, 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. Na przykład pytanie "Na jakim wydziale pracuje Kowalski?" może mieć różne odpowiedzi w zależności od kontekstu, w jakim zostało zadane. Pytającym mógł być np. urzędnik, któremu chodziło o precyzyjne określenie miejsca, w którym Kowalski formalnie jest zatrudniony. Równie dobrze mogło jednak pytającemu chodzić o wydziały, na których Kowalski ma zajęcia ze studentami. Problemem jest również utworzenie opisu bazy danych w sposób minimalnie obciążający użytkownika.

Artykuł prezentuje różne poziomy języka zapytań i przedstawia relację uniwersalną jako jedną z dróg ewolucji systemów baz danych w kierunku idealnego języka zapytań.

2. Relacyjna baza danych. Pojęcia podstawowe

Relacyjna baza danych może być zdefiniowana jako zbiór relacji, czyli dwuwymiarowych tablic. *Nazwy atrybutów* tworzą nagłówki kolumn. Określenie "nazwa atrybutu" skraca się zwykle do określenia "*atrybut*". *Krotka* to zbiór wartości atrybutów tworzących wiersz tablicy. Każda relacja ma swoją nazwę.

ANGAŻE

Pracownik	Wydział
Kowalski	Mech.
Wilk	MatFiz

ZAJĘCIA

Pracownik	Przedmiot
Kowalski	Mechanika

PLAN

Wydział	Przedmiot	Godziny
Mech.	Fizyka	200
MatFiz	Algebra	180

Rys. 1. Baza danych UCZELNIA
Fig. 1. Database UCZELNIA

Rys. 1 przedstawia przykładową bazę danych. Każda krotka relacji ANGAŻE informuje, że pracownik identyfikowany atrybutem Pracownik jest zatrudniony na wydziale identyfikowanym przez atrybut Wydział. Krotki relacji ZAJĘCIA zawierają informację o tym, że pracownik identyfikowany atrybutem Pracownik ma zajęcia z przedmiotu określonego przez atrybut Przedmiot. Krotki relacji PLAN podają liczbę godzin zajęć (atrybut Godziny) z danego przedmiotu dla studentów danego wydziału.

Zbiór atrybutów relacji tworzy *schemat relacji*. Zapis r(R) oznacza relację r o schemacie R.

Zapis t[A] oznacza wartość atrybutu A w krotce t.

Schemat bazy danych to zbiór schematów relacji tworzących bazę danych.

Pomiędzy atrybutami zachodzą zależności funkcyjne. *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 $t_1[Y] = t_2[Y]$. Zapis $X \rightarrow Y$ czytamy: Y jest zależne funkcyjnie od X.

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.

W artykule przyjęto, że zależności funkcyjne w bazie są dane poprzez określenie kluczy relacji. W schemacie relacji klucze będą wskazywane przez ich podkreślenie.

Dalej rozważane będą dwie bazy danych składające się z relacji przedstawionych na rys. 1:

– baza danych UCZELNIA:

ANGAŻE(Pracownik, Wydział),

ZAJĘCIA(Pracownik, Przedmiot),

PLAN(Wydział, Przedmiot, Godziny);

– baza danych KADRY:

ANGAŻE(Pracownik, Wydział),

ZAJĘCIA(Pracownik, Przedmiot),

W artykule niekiedy stosowane są nazwy atrybutów skrócone do pierwszej litery, ewentualnie uzupełnionej znacznikiem (np. Pracownik - P, Przedmiot - P_1). Suma atrybutów A_1, A_2, ..., A_m będzie wówczas zapisywana jako $A_1A_2...A_m$ (np. Pracownik $\cup$ Wydział $\cup$ Godziny może być zapisane jako PWG).

Właściwości zależności funkcyjnych określone są przez *aksjomaty Armstronga.*

Przyjmijmy, że F jest zbiorem zależności funkcyjnych spełnionych w zbiorze atrybutów U oraz że X, Y, Z $\subseteq$ U. Postać trzech podstawowych aksjomatów wg Ullmana [6] jest następująca:

A1) (zwrotność): jeżeli Y $\subseteq$ X $\subseteq$ U, to X –> Y

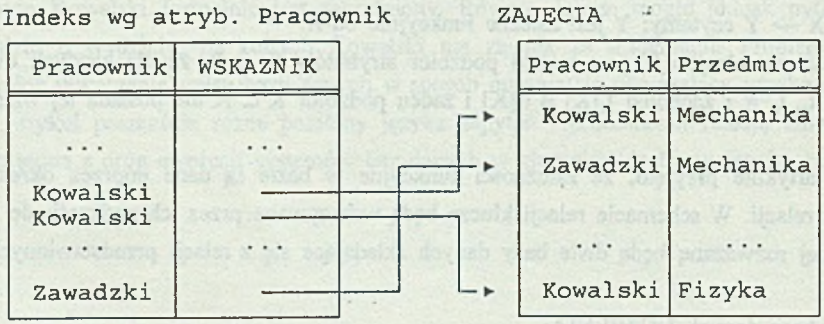
A2) (rozszerzenie): jeżeli X –> Y, Z $\subseteq$ U to XZ –> YZ

A3) (przechodniość): jeżeli X –> Y, Y –> Z to X –> Z

Domknięcie zbioru atrybutów X względem zbioru zależności F, oznaczane X_F^+ jest to zbiór wszystkich atrybutów $A_i \in$ U, dla których zależność X -> A_i może być wywnioskowana z F za pomocą aksjomatów Armstronga.

System zarządzania bazą danych (DBMS), to zestaw środków programowych umożliwiających użytkownikom operowanie na danych i pozwalających administratorowi bazy danych na centralne sterowanie danymi.

Dostęp do danych w relacji może przyśpieszyć indeks. *Indeks* jest to tablica, która na podstawie wartości *wyrażenia indeksującego* (jest nim najczęściej pojedyńczy atrybut) określa fizyczne położenie krotki w relacji. Przykład indeksu przedstawia rys. 2. Wyrażeniem indeksującym jest atrybut Pracownik.



Rys. 2. Przykład indeksu dla relacji ZAJĘCIA w bazie danych UCZELNIA

Fig. 2. The Example of an Index for a Relation ZAJĘCIA in the Database UCZELNIA

3. Pojęcie języków zapytań w relacyjnych bazach danych. Poziomy języków zapytań

W relacyjnej bazie danych można wyróżnić następujące operacje na danych:
- aktualizacja,
- wyszukiwanie,
- przetwarzanie (np. z użyciem operacji arytmetycznych),
- zapamiętywanie i raportowanie.

Język zapytań obejmuje wyszukiwanie oraz zapamiętywanie i raportowanie danych. Według Kozielskiego [3], można wyróżnić 4 poziomy języków zapytań:
- poziom operowania krotkami,
- poziom algebry relacji,
- poziom rachunku relacji i rachunku dziedzin, zwany poziomem rachunku predykatów,
- poziom relacji uniwersalnej.

Poziomy te opisane są w tabeli 1.

Wraz z poziomem języka zapytań rośnie zwięzłość zapisu pytania. Rośnie również rola opisu bazy danych i podprogramów interpretujących ten opis.

Przyjęto równoważność terminów: zadanie wyszukiwania, pytanie, zapytanie.

4. Przegląd poziomów języka zapytań. Powiązania między danymi. Opisy baz danych

Aby odpowiedzieć na zapytanie do bazy danych DBMS musi mieć określone (w pytaniu lub w opisie bazy) następujące elementy:

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 związki pomiędzy relacjami wymienionymi w zadaniu wyszukiwania,

E4) **operacje**, czyli wskazanie operacji na relacjach, które trzeba wykonać, aby otrzymać rozwiązanie,

Tabela 1

Poziomy języków zapytań

Nazwa	Własności	Elementy pytania	Systemy mające język zapytań danego poziomu
Operowanie krotkami	Wyszukiwanie krotek (np. o zadanych wartościach atrybutów), najczęściej z użyciem pętli programowych.	Szukane i warunki, struktury, związki, operacje, opis operacji.	BTRIEVE, dBASE, Clipper, FoxBase, FoxPro, PROGRESS.
Algebra relacji	Działania na całych relacjach przy użyciu operatorów algebry relacji.	Szukane i warunki, struktury, związki, operacje.	dBASE, Clipper, FoxBase, FoxPro, Informix, PROGRESS.
Rachunek relacji	Opis warunków nałożonych na rozwiązanie. Opis zawiera m. in. nazwy atrybutów, nazwy relacji, z których te atrybuty pochodzą oraz zmienne typu krotka.	Szukane i warunki, struktury, związki.	dBASE IV, FoxPro 2.0, Paradox 3.5, Informix, PROGRESS.
Rachunek dziedzin	Opis warunków nałożonych na rozwiązanie. Graficzna forma opisu. Opis zawiera m. in. nazwy atrybutów, nazwy relacji, z których te atrybuty pochodzą oraz zmienne typu dziedzina.	Szukane i warunki, struktury, związki.	dBASEIV, FoxPro 2.0, Paradox.
Relacja uniwersalna	Opis warunków nałożonych na rozwiązanie. Operuje się jedynie nazwami atrybutów, abstrahując od ich rozmieszczenia w relacjach.	Szukane i warunki.	Systemy eksperymentalne

E5) **opis operacji**, czyli opis operacji na relacjach z dokładnością do przetwarzania pojedynczych krotek.

Element E1) określmy 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 programie. Stąd im wyższy poziom języka zapytań, tym więcej tych elementów musi być w opisie bazy.

Element "szukane i warunki" zawsze musi znaleźć się w programie wyszukiwania. Jest to zapis sensu pytania. Opis bazy może dostarczyć jedynie:

- pewnych stałych fragmentów składni pytania (np. słów kluczowych),
- biblioteki gotowych pytań.

Element "struktury" nie występuje na poziomie relacji uniwersalnej, a nie musi wystąpić na poziomie rachunku predykatów. Aby nie podawać w pytaniu nazw relacji, muszą być spełnione następujące warunki:

- należy zapamiętać w opisie bazy schemat bazy danych lub jego część, której dotyczy pytanie,
- system musi na podstawie opisu bazy i tekstu pytania wywnioskować, o które relacje chodzi pytającemu (możliwe niejednoznaczności i drogi ich rozwiązania będą omówione wraz z relacją uniwersalną).

Podawanie indeksów nie jest konieczne na jakimkolwiek poziomie, jeżeli w opisie bazy pliki indeksowe zostaną przypisane relacjom. Zastosowanie indeksów przyśpiesza uzyskanie odpowiedzi, ale nie zmienia poziomu języka zapytań.

Element "związki" oznacza zapisanie sposobu powiązania danych z różnych relacji. Związki pomiędzy danymi zawartymi w różnych relacjach odzwierciedlane są poprzez wartości wybranych do tego celu atrybutów. Na przykład w bazie danych UCZELNIA związek pomiędzy danymi w relacjach ANGAŻE i ZAJĘCIA opisuje atrybut Pracownik. Zalecane jest, aby w bazie danych atrybuty oznaczające to samo w różnych relacjach nazywały się tak samo. Z kolei atrybuty mające różne znaczenie powinny różnić się nazwami. Taka interpretacja powiązań na ogół nie jest przyjmowana automatycznie. Systemowi zarządzania bazą danych należy podać atrybuty, poprzez które powiązane są dane z różnych relacji. Na każdym poziomie języka zapytań związki pomiędzy relacjami są inaczej wyrażane i można je zapamiętać w opisie bazy.

Element "operacje" oznacza zapisanie algorytmu uzyskania wyniku za pomocą operowania całymi relacjami. Istnieją metody pozwalające dla dowolnego zadania wyszukiwania uzyskać element "operacje" mając dane elementy "szukane i warunki", "struktury", "związki".

Element "opis operacji" oznacza zdefiniowanie operacji na relacjach za pomocą operacji na krotkach.

W kolejnych podrozdziałach dokonany jest przegląd języków zapytań na różnych poziomach, dla wybranych systemów baz danych. Przegląd ten jest podstawą do rozważań na temat relacji uniwersalnej umieszczonych w rozdziale 5.

W dalszej części artykułu będzie rozważane następujące zadanie przykładowe.

Zadanie 1

Dla bazy danych KADRY podaj przedmioty, z których mają zajęcia pracownicy wydziału Mechanicznego ('Mech.').

4.1. Poziom operowania krotkami

Jest to poziom najbardziej elementarny. Zapis zadania wyszukiwania zawiera zarazem szczegółowy algorytm rozwiązania. Języki zapytań na poziomie operowania krotkami są zaimplementowane w systemach BTRIEVE, dBASE, Clipper, FoxBase, FoxPro oraz PROGRESS.

System BTRIEVE oferuje najniższy poziom języka zapytań. Program musi zawierać wszystkie elementy celu i kontekstu oraz definicję schematu relacji, ponieważ system, zapamiętując dane w pliku, nie zapamiętuje logicznej struktury krotek. Struktura ta jest definiowana w programie operującym na pliku z danymi poprzez deklarację rekordu stanowiącego bufor zawierający aktualnie przetwarzaną krotkę. Związki między danymi są określane w programie poprzez algorytm wyszukiwania.

W systemach dBASE, Clipper, FoxBase, FoxPro nazwy i typy atrybutów pamiętane są wraz z danymi. Program wyszukiwania na poziomie operowania krotkami musi zawierać następujące elementy: "szukane i warunki", struktury", "związki", "operacje", "opis operacji". Programy wyszukiwania można uprościć, opisując związki pomiędzy relacjami przed zapisem operacji wyszukiwania. Związki te można opisać albo definiując je w programie przed operacjami wyszukiwania, albo tworząc uprzednio plik opisu bazy (view) zawierający opis tych związków.

W systemie PROGRESS informacje o schematach relacji w bazie danych zapamiętywane są w tzw. słowniku bazy danych. Pamiętane są m. in.; nazwa i typ atrybutu, jego wartość początkowa przy wprowadzaniu nowej krotki oraz dopuszczalny zakres wartości atrybutu. W programach wyszukiwania można korzystać z zapamiętanych w słowniku bazy powiązań pomiędzy relacjami. System ma możliwość automatycznego zapamiętania tych powiązań dla tak samo nazywających się atrybutów w różnych relacjach.

Tabela 2 przedstawia rozwiązanie zadania 1 w języku systemu dBASE na poziomie operowania krotkami przy wykorzystaniu indeksów. Prawa kolumna tablicy zawiera program, w którym posłużono się następującym opisem bazy:

select 1

```
use ZAJĘCIA alias ZAJ
index on Pracownik to ZAJ_P
select 2
use ANGAŻE alias ANG
index on Wydział to ANG_W
&& Zapamiętanie środowiska w pliku opisu bazy KADRY.VUE:
create view Kadry from environment
```

Tabela 2

Zapis zadania 1 na poziomie operowania krotkami w języku dBASE

Program bez użycia opisu bazy	Program używający opisu bazy
set talk off select 1 use ZAJĘCIA alias ZAJ index on Pracownik to ZAJ_P select 2 use ANGAŻE alias ANG index on Wydział to ANG_W seek 'Mech.' if found() do while Wydział = 'Mech.' .and. .not. eof() select ZAJ seek ANG->Pracownik if found() do while Pracownik = ANG->Pracownik .and. .not. eof() ? Przedmiot skip enddo select ANG skip enddo set talk on	set talk off set view to Kadry seek 'Mech.' if found() do while Wydział = 'Mech.' .and. .not. eof() select ZAJ seek ANG->Pracownik if found() do while Pracownik = ANG->Pracownik .and. .not. eof() ? Przedmiot skip enddo select ANG skip enddo set talk on

4.2. Poziom algebry relacji

Program wyszukiwania na tym poziomie zawiera elementy "szukane i warunki", "struktury", "związki" i "operacje" opisane na początku rozdz. 4. Wyszukiwanie na tym poziomie bazuje na operatorach algebry relacji. Podstawowe operatory algebry relacji definiuje tabela 3.

Tabela 3

Podstawowe operatory algebry relacji

N	Operator	Zn.	Definicja
1	Suma	$\cup$	$r(R) \cup s(S) = q(Q) = \{t \mid t[R] \in r \vee t[S] \in s \wedge (R=S=Q)\}$
2	Różnica	–	$r(R) - s(S) = q(Q) = \{t \mid t[R] \in r \wedge (\neg(t[S] \in s)) \wedge (R=S=Q)\}$
3	Selekcja	σ	$\sigma_{A=a}(r) = r'(R) = \{t \in r \mid t[A] = a\}$
4	Projekcja	π	$\pi_X(r) = r'(X) = \{t[X] \mid t \in r\}$
5	I. kartez.	$\times$	$r \times s = q(R \square S) = \{t \mid \forall t_r \in r \wedge \forall t_s \in s \wedge t = t_r \square t_s\}$

Oznaczenia :

I. kart.	-	iloczyn kartezjański,
r, s, q, r'	-	relacje,
R, S, Q	-	schematy relacji,
X	-	podzbiór schematu relacji,
A	-	pojedynczy atrybut,
a	-	wartość atrybutu A,
$\square$	-	operator konkatenacji.

Za pomocą operatorów zdefiniowanych w tabeli 3 definiuje się operatory pochodne. Najczęściej używanym z nich jest operator złączenia. *Operator złączenia* działa na dwóch relacjach r(R) i s(S). Istnieje złączenie naturalne i θ-złączenie.

Złączenie naturalne powoduje utworzenie relacji wynikowej q(Q), gdzie $Q = R \cup S$, w której każda krotka t została utworzona z wartości atrybutów należących do takich krotek $t_r \in r$ i $t_s \in s$, dla których zachodzi $t_r[R \cap S] = t_s[R \cap S]$ oraz $t[R] = t_r$ i $t[S] = t_s$. Formalna definicja jest następująca:

$$r(R) \bowtie s(S) = q(R \cup S) = \{t \mid t_r \in r \wedge t_s \in s \wedge t[R] = t_r \wedge t[S] = t_s \wedge t_r[R \cap S] = t_s[R \cap S]\}.$$

θ-złączenie powoduje utworzenie relacji wynikowej q(Q) zawierającej te wszystkie konkatenacje krotek z relacji r i s, dla których spełniony jest warunek nałożony na wartości wybranych atrybutów. Zapis formalny operacji złączenia jest następujący:

$$q(Q) = r \underset{A\,\theta\,B}{\bowtie} s = \sigma_{A\,\theta\,B}(r \times s)$$

gdzie

A - atrybut relacji r

B - atrybut relacji s

θ - operator = , < , > , <= , >= , ...

x - iloczyn kartezjański relacji.

W praktycznych zastosowaniach algebry relacji najczęściej używana jest operacja tzw. równozłączenia, tzn. w warunku złączenia jest użyty operator = .

Powiązania między danymi na poziomie algebry relacji określane są przy realizacji łączeń.

Przy złączeniu naturalnym dwie krotki (każda z jednej z łączonych relacji) łączone są razem, gdy w obu krotkach są równe wartości atrybutów będących częścią wspólną schematów obu łączonych relacji.

Przy θ-złączeniu dwie krotki (każda z jednej z łączonych relacji) sklejane są razem, gdy dla tych krotek spełniony jest warunek złączenia (najczęściej równość wartości wskazanych atrybutów - jednego z jednej, a drugiego z drugiej relacji).

W językach klasy dBASE (dBASE, Clipper, FoxBase, FoxPro) powiązania między danymi z różnych relacji można definiować w programie wyszukiwania na poziomie algebry relacji w instrukcji JOIN (jako warunek złączenia) lub w instrukcji SET RELATION (jako powiązanie krotek mających równą wartość ustalonego wyrażenia).

Zadawanie pytań na poziomie algebry relacji możliwe jest również w języku SQL opisanym m. in. przez Ullmana [6], [7].

Tabela 4

Zapis zadania 1 na poziomie algebry relacji w języku dBASE

Program z użyciem JOIN	Program z użyciem opisu bazy
select 1 use ZAJĘCIA alias ZAJ select 2 use ANGAŻE alias ANG set filter to Wydzial = 'Mech.' join with ZAJ to WYN ; for Pracownik = ZAJ->Pracownik ; fields Przedmiot use WYN list	set view to Kadry list Przedmiot for ANG->Wydział = 'Mech.'

Zapis zadania 1 w symbolice algebry relacji jest następujący:

$$WYNIK = \pi_{Przedmiot}(\sigma_{Wydział='Mech.'}(ANGAŻE) \bowtie ZAJĘCIA)$$

Tabela 4 przedstawia rozwiązanie zadania przykładowego w języku systemu dBASE na poziomie algebry relacji. Pierwsza kolumna przedstawia rozwiązanie z użyciem instrucji JOIN, druga - z użyciem powiązań między relacjami zapamiętanych w następującym opisie bazy:

```
select 1
use ANGAŻE alias ANG
index on Pracownik to ANG_P
select 2
use ZAJĘCIA alias ZAJ
set relation to Pracownik into ANG
create view Kadry from environment
```

4.3. Poziom rachunku predykatów

Rachunek predykatów obejmuje rachunek relacji i rachunek dziedzin.

4.3.1. Poziom rachunku relacji

Ogólna postać wyrażeń rachunku relacji jest następująca:

$$\{ t \mid \Phi(t) \} \qquad (1)$$

Wyrażenie (1) oznacza ogół takich t, że $\Phi(t)$ jest prawdziwe.

W wyrażeniu tym

t - oznacza zmienną typu krotka,

Φ - oznacza formułę zwaną predykatem.

Formuła zbudowana jest z atomów oraz operatorów.

Atomy mogą mieć postać:

a) R(s) - co oznacza relację o nazwie R, dla której s jest zmienną typu krotka o strukturze odpowiadającej strukturze krotek relacji R,

b) s[A] θ u[B] - co oznacza porównanie, zdefiniowane operatorem θ, między wartością atrybutu A zmiennej krotkowej s i wartością atrybutu B zmiennej krotkowej u; $\theta \in \{ =, <, >, \ldots \}$,

c) s[A] θ a - co oznacza porównanie między wartością atrybutu A zmiennej krotkowej s i wartością stałą a należącą do zbioru wartości A.

Formuła zdefiniowana jest następująco:

1. Atom jest formułą.
2. Jeżeli Φ_1, Φ_2 są formułami, to $\Phi_1 \vee \Phi_2$, $\Phi_1 \wedge \Phi_2$, $\neg\Phi_1$, $\neg\Phi_2$ też są formułami.
3. Jeżeli Φ jest formułą, to $(\exists s) \Phi$ oraz $(\forall s) \Phi$ też są formułami.

Przykładem języka zapytań umożliwiającym formułowanie zadań wyszukiwania m. in. na poziomie rachunku relacji jest SQL. Związki pomiędzy relacjami definiowane są w zadaniu wyszukiwania atomami typu b). W języku SQL związki te można umieścić bezpośrednio w pytaniu lub w opisie bazy danych w formie perspektywy.

Perspektywa bazy r(R) jest to wirtualna baza r'(R'), gdzie R' jest odwzorowaniem R, w której zachowane są więzy (czyli ograniczenia zapewniające poprawność danych) bazy r, a operacje na bazie r' są odwzorowywane na bazę r (Tsichritzis, Lochovsky [5]).

W przeciwieństwie do dwu niższych poziomów języka zapytań, zapisując pytanie do bazy danych, podajemy jedynie *co* chcemy otrzymać. Program wyszukiwania zawiera elementy "szukane i warunki", "struktury", "związki" zdefiniowane w rozdziale 3, a jeśli zdefiniować uprzednio perspektywę, to tylko elementy "szukane i warunki" i "struktury".

Spośród opisywanych systemów językiem SQL dysponują dBASE IV, FoxPro 2.0, INFORMIX SQL i PROGRESS.

Zapis zadania przykładowego w postaci wyrażenia rachunku relacji jest następujący:

$$\{ x[\text{Przedmiot}] \mid (\exists a)(\text{ANGAŻE}(a)) \wedge (\exists z)(\text{ZAJĘCIA}(z)) \wedge a[\text{Wydział}] = \text{'Mech.'} \wedge a[\text{Pracownik}] = z[\text{Pracownik}] \wedge x[\text{Przedmiot}] = z[\text{Przedmiot}] \}$$

Tabela 5 pokazuje zapis zadania przykładowego w języku SQL. Prawa kolumna zawiera zapis zadania po uprzednim rozszerzeniu opisu bazy o następującą perspektywę:

```
create view Kadry
  as select Wydział, Pracownik, Przedmiot
     from ANGAŻE a, ZAJĘCIA z
     where a.Pracownik = z.Pracownik
```

Tabela 5

Zapis zadania 1 na poziomie rachunku relacji w języku SQL

Program bez użycia perspektywy	Użycie perspektywy Kadry
select Przedmiot from ANGAŻE a, ZAJĘCIA z where a.Wydział = 'Mech.' and a.Pracownik = z.Pracownik	select Przedmiot from Kadry where Wydział = 'Mech.'

4.3.2. Poziom rachunku dziedzin

Ogólna postać wyrażeń jest następująca:

$$\{ x_1, x_2, ..., x_k \mid \Phi(x_1, x_2, ..., x_k) \} \qquad (2)$$

Wyrażenie (2) oznacza ogół takich zestawień $x_1, x_2, ..., x_k$, że $\Phi(x_1, x_2, ..., x_k)$ jest prawdziwe. W wyrażeniu (2):

x_i - zmienna typu dziedzina ; $i \in \{ 1, ..., k \}$,

Φ - formuła (predykat).

Podobnie jak w rachunku relacji, formuła może być zbudowana z atomów oraz z operatorów.

Atomy mogą mieć postać:

a) $R(x_1, x_2, ..., x_k)$, gdzie R jest relacją zawierającą k atrybutów, a każde x_i ($i \in \{1, ..., k\}$) jest stałą lub zmienną dziedzinową,

b) $x \theta y$; gdzie x, y są zmiennymi typu dziedzina lub wartościami stałymi, natomiast $\theta \in \{ =, <, >, ...\}$.

Użycie $R(x_1, x_2, ..., x_k)$ oznacza, że wartości x_i muszą być tak dobrane, aby $x_1, x_2, ..., x_k$ było krotką należącą do R. Użycie $x \theta y$ wymusza taką wartość x, y, aby $x \theta y$ było prawdziwe. Reguły konstruowania formuły są takie same, jak w rachunku krotek.

Przykładem języka zapytań na poziomie rachunku relacji jest język Query By Example (QBE) opisany m. in. przez Ullmana [6], [7].

Związki pomiędzy relacjami definiowane są w zadaniu wyszukiwania poprzez umieszczenie w tych atomach typu a), które odpowiadają powiązanym relacjom, tej samej zmiennej. Zmienna musi być umieszczona w miejscu przypisanym tym atrybutom, które wiążą relacje. W języku QBE związki te można umieścić bezpośrednio w pytaniu lub w formie perspektywy w opisie bazy danych.

Podobnie jak dla rachunku relacji, program wyszukiwania zawiera elementy "szukane i warunki", "struktury", "związki" zdefiniowane w rozdziale 3, a jeśli istnieje uprzednio zdefiniowana perspektywa, to tylko elementy "szukane i warunki" i "struktury".

Rozwiązanie przykładowego zadania 1 na poziomie rachunku dziedzin jest następujące:

{ Przedmiot | (∃Pracownik)
(ANGAŻE(Pracownik, 'Mech.') ∧
ZAJĘCIA(Pracownik, Przedmiot))}

5. Relacja uniwersalna - narzędzie wyszukiwania

5.1. Koncepcja relacji uniwersalnej

W rozdz. 4 pokazano, że poziom języka wzrasta, gdy przybywa elementów kontekstu przenoszonych z pytania do opisu bazy. Na poziomie rachunku predykatów pytanie zawiera co najwyżej dwa elementy kontekstu: "struktury" oraz "związki". Próba przeniesienia tych elementów z pytania do opisu bazy prowadzi do koncepcji relacji uniwersalnej opisanej we wszystkich pozycjach podanej literatury.

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

Mechanizm relacji uniwersalnej ma na celu między innymi uwolnienie użytkownika od konieczności pamiętania schematów relacji oraz uproszczenie zapisu zadania wyszukiwania. Zapis zadania wyszukiwania dla relacji uniwersalnej zawiera jedynie element celu E1) opisany na początku rozdz. 4, czyli nazwy atrybutów szukanych oraz warunki nałożone na rozwiązanie. Nazwy atrybutów używane są bez kwalifikatorów określających relacje, z których te atrybuty pochodzą.

Podobnie jak w pracy Ullmana [6] przyjmijmy, że zadanie wyszukiwania w języku zapytań na poziomie relacji uniwersalnej ma postać:

select Q where Warunki(P) (3)

gdzie

Q - ciąg nazw atrybutów szukanych,

Warunki(P) - wyrażenie logiczne, operujące na zbiorze atrybutów P (być może pustym).

Zapis (3) oznacza: podaj wszystkie krotki o schemacie Q, które spełniają warunki nałożone na atrybuty P. Zbiór wszystkich atrybutów wymienionych w pytaniu oznacza się przez X, czyli $X = Q \cup P$.

Zadanie 1 kierowane do relacji uniwersalnej dla bazy danych KADRY ma postać:

select Przedmiot where Wydział = 'Mech.' (4)

Wyrażenie logiczne Warunki(P) zawiera zmienne zakresu. Jeżeli jest tylko jedna zmienna zakresu, to jest to zmienna pusta, tak jak w rozwiązaniu zadania przykładowego. Jeżeli trzeba ich użyć więcej, to należy nadać im unikalne nazwy, a nazwy atrybutów należy poprzedzać nazwami odpowiednich zmiennych.

W idealnym przypadku zdefiniowanie relacji uniwersalnej wymaga przyjęcia kilku założeń:

Z1) założenie zachowania zależności funkcyjnych,

Z2) założenie jednoznaczności nazw atrybutów,

Z3) założenie jednoznacznej interpretacji zadania wyszukiwania,

Z4) założenie zachowania zależności złączeniowej.

Założenie Z1 dotyczy zbioru F zależności funkcyjnych spełnionych w bazie **r**. W relacji uniwersalnej u, zdefiniowanej dla bazy **r**, zależności ze zbioru F powinny być również spełnione.

Założenie Z2, należące do najważniejszych, przyjmuje, że atrybuty oznaczające to samo powinny mieć te same nazwy, natomiast atrybuty oznaczające co innego muszą różnić się nazwami. Założenie to jest potrzebne do automatycznego znajdowania powiązań między relacjami przez system zarządzania bazą danych.

Założenie Z3 dotyczy sposobu wyznaczania rozwiązań zadań wyszukiwania. Powiązania pomiędzy atrybutami ze zbioru X powinny umożliwić jednoznaczne rozwiązanie zadania. Jeżeli rozwiązań jest więcej niż jedno, to system musi otrzymać dodatkową informację (z opisu bazy lub od zadającego pytanie).

Założenie Z4 wymaga spełnienia przez relację uniwersalną u o schemacie $U = R_1 \cup R_2 \cup ... \cup R_n$ utworzoną nad bazą danych $\mathbf{r} = \{r_1(R_1), r_2(R_2), ..., r_n(R_n)\}$ równań

$$(\forall r_i \in \mathbf{r})\ \pi_{R_i}(u) = r_i\ ,\ i = 1, 2, ..., n. \qquad (5)$$

Założenie Z4 można rozpatrywać w kontekście aktualizacji i wyszukiwania.

Aktualizacja może doprowadzić do stanu bazy **r**, w którym równania (5) nie są spełnione. Piszą o tym Ullman[6] oraz Tsichritzis i Lochowski[5]. Brak spełnienia warunku (5) wpływa na wynik wyszukiwania. Wpływ ten zostanie dokładniej omówiony poniżej.

Warunek (5) może nie być spełniony z dwóch powodów:

a) schemat bazy danych został źle zaprojektowany,

b) w bazie danych znajdują się tzw. krotki zwisające.

W artykule zostanie rozważony problem krotek zwisających.

Rozważmy przykładową bazę danych UCZELNIA z rys. 1. Druga krotka relacji ANGAŻE nie ma odpowiednika w relacji ZAJĘCIA. Jest to krotka zwisająca, bowiem wynik złączenia ANGAŻE $\bowtie$ ZAJĘCIA nie zawiera tej krotki.

Ogólnie, jeżeli $s = \bowtie_{i=1}^{n} r_i(R_i)$, a przy tym $\exists i \in \{1..n\}\ \pi_{R_i}(s) \diamond r_i$, to *krotki zwisające* (ang. dangling tuples) w relacji r_i są to krotki należące do $r_i - \pi_{R_i}(s)$.

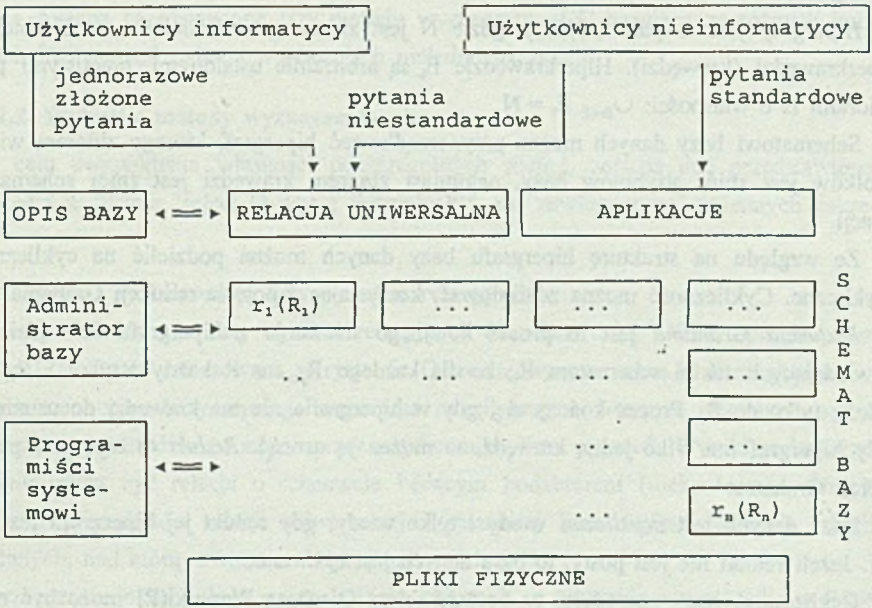
Sens założenia Z4 w kontekście wyszukiwania polega na tym, że w bazie danych nie powinno być krotek zwisających, ponieważ krotki te nie przechodzą do wyników złączeń. Warunek Z4 jest trudny zarówno do wypełnienia (w rzeczywistej bazie danych

występują krotki zwisające), jak i do sprawdzenia (sprawdzenie należy do problemów NP-zupełnych).

Aby utworzyć relację uniwersalną dla bazy danych z krotkami zwisającymi można (przynajmniej hipotetycznie) uzupełnić brakujące dane wartościami pustymi. Wartości puste muszą być wstawiane tak, aby zachować zależności funkcyjne bazy r, nad którą jest tworzona relacja uniwersalna.

Wartość pusta (ang. null value) jest wprowadzana zamiast konkretnej wartości atrybutu, gdy wartość ta nie jest znana. Wartości puste rozróżnia się między sobą i oznacza δ_i, gdzie i jest liczbą całkowitą identyfikującą symbol pusty. Uważa się, że $\delta_i = \delta_i$ (np. przy równozłączeniu), oraz że $\delta_i <> \delta_j$ dla $i <> j$.

System oparty na koncepcji relacji uniwersalnej można traktować jako kompletny system zarządzania relacyjną bazą danych. Ullman [6], [7] pokazuje jednak, że baza danych o takim modelu nastręcza wiele problemów przy aktualizacji (zmiana zawartości, wstawianie, a zwłaszcza usuwanie krotek). W związku z tym sugeruje on zastosowanie relacji uniwersalnej jedynie jako narzędzia umożliwiającego użycie języka zapytań, przy założeniu, że pozostałe manipulacje danymi można realizować na poszczególnych relacjach bazy innymi narzędziami (rys. 3).



Rys. 3. Zastosowanie relacji uniwersalnej jako narzędzia wyszukiwania
Fig. 3. The application of an Universal Relation as a Searching Tool

5.2. Wyszukiwanie w relacji uniwersalnej

Ullman [7] 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.

Algorytm 1. Wyznaczanie rozwiązania q(Q) zadania wyszukiwania:

select Q where Warunki(P).

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

Zasadniczym problemem jest wyznaczenie [X].

5.2.1. Wpływ schematu bazy danych na metodę wyznaczania okna

Przy wyznaczaniu okna uwzględnia się schemat bazy danych i zależności funkcyjne między jej atrybutami. W praktyce, pod uwagę bierze się zależności wynikające ze wskazania kluczy w relacjach bazy. Schemat bazy wygodnie jest opisywać za pomocą hipergrafu.

Hipergraf jest to para (N,E), gdzie N jest zbiorem wierzchołków, a E zbiorem hiperkrawędzi (krawędzi). Hiperkrawędzie E_i są arbitralnie ustalonymi, niepustymi podzbiorami N o własności: $\cup_{E_i \in E} E_i = N$.

Schematowi bazy danych można przyporządkować hipergraf, którego zbiorem wierzchołków jest zbiór atrybutów bazy, natomiast zbiorem krawędzi jest zbiór schematów relacji.

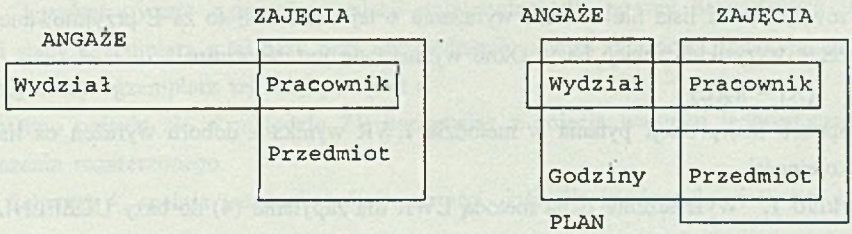
Ze względu na strukturę hipergrafu bazy danych można podzielić na cykliczne i acykliczne. Cykliczność można zdefiniować, korzystając z pojęcia redukcji Grahama.

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 można 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*.

Cykliczność bazy powoduje, że pytanie select Q where Warunki(P) może być rozumiane wieloznacznie, co narusza założenie o jednoznacznej interpretacji zadania wyszukiwania. Rozważmy cykliczną bazę UCZELNIA z rys. 4. Pytanie (4):

select Przedmiot where Wydział = 'Mech.'



Rys. 4. Hipergraf baz danych: acyklicznej KADRY i cyklicznej UCZELNIA
Fig. 4. The Hipergraph of an Acyklic Database KADRY and of a Cyklic Database UCZELNIA

zadane do bazy UCZELNIA może być rozumiane na kilka sposobów.

A. Podać przedmioty prowadzone dla studentów Wydziału Mechanicznego.

B. Podać przedmioty prowadzone przez pracowników Wydziału Mechanicznego.

C. Podać sumę odpowiedzi na pytania A i B.

D. Podać iloczyn odpowiedzi na pytania A i B.

Każda z tych interpretacji wymaga innego okna [X] dla udzielenia odpowiedzi. Interpretacja pytania zgodna z życzeniem pytającego jest możliwa dopiero po uzupełnieniu pytania "select Q where Warunki(P)" dodatkowymi informacjami lub założeniami. W artykule zostaną przedstawione trzy metody wyznaczania [X] bazujące na różnych koncepcjach interpretacji pytania w sytuacjach niejednoznacznych.

5.2.2. Wybrane metody wyznaczania okna

W celu uwypuklenia własności poszczególnych metod, zostaną one przedstawione dla pytania w formie "select Q where Warunki(P)", nie zawierającego zmiennych zakresu.

5.2.2.1. Metoda definiowania listy wyrażeń relacyjnych

Metoda definiowania listy wyrażeń relacyjnych (nazwana dalej LWR), opisana przez Ullmana [7], została zastosowana w systemie Q zrealizowanym przez B. Kernighama.

Tworzenie opisu bazy wymaga wskazania relacji bazy **r**. Istotnym elementem opisu bazy jest w tej metodzie tworzona przez użytkownika lista wyrażeń. Wynikiem każdego wyrażenia musi być relacja o schemacie będącym podzbiorem (niekoniecznie właściwym) zbioru atrybutów relacji uniwersalnej. Najczęściej lista zawiera złączenia relacji bazy danych, nad którą utworzono relację uniwersalną.

Wyznaczenie okna polega na znalezieniu zbioru X, a następnie wyszukaniu w liście wyrażeń pierwszego z wyrażeń E, w którym użyty zbiór atrybutów zawiera X lub jest

mu równy. Jeżeli lista nie zawiera wyrażenia o tej własności, to za E przyjmowane jest złączenie wszystkich relacji bazy. Okno wyznaczane jest ze wzoru:

$[X] = \pi_X(E)$

Sposób interpretacji pytania w metodzie LWR wynika z doboru wyrażeń na liście i ich kolejności.

Przykład 1. Wyznaczenie okna metodą LWR dla zapytania (4) do bazy UCZELNIA:

select Przedmiot where Wydział = 'Mech.'

Załóżmy, że lista wyrażeń jest następująca:

ANGAŻE

ZAJĘCIA

PLAN

ANGAŻE ⋈ ZAJĘCIA

ANGAŻE ⋈ ZAJĘCIA ⋈ PLAN

Atrybuty użyte w pytaniu: $X = P_tW$

Okno: $[X] = [P_tW] = \pi_{P_tW}(PLAN)$

Interpretacja pytania (4): zależna od kolejności wyrażeń na liście; dla podanej powyżej listy jest to interpretacja B.

5.2.2.2. Metoda rozszerzonych złączeń

Metoda rozszerzonych złączeń (zwana dalej ZR) opisana przez Sagiva [4] i Ullmana [7] bazuje na pojęciu egzemplarza charakterystycznego relacji uniwersalnej (Sagiv [4], Ullman [7]).

Przed zdefiniowaniem tego pojęcia należy wprowadzić kilka definicji pomocniczych.

Jeżeli u jest relacją zdefiniowaną nad zbiorem atrybutów U i $Y \subseteq U$, to *ogólna projekcja U na Y* jest następującym zbiorem krotek:

$\pi\downarrow_Y(u) = \{ t[Y] \mid (t \in u) \wedge (t[Y]$ nie zawiera wartości pustych$) \}$

Jeżeli baza danych **r** składa się z relacji $r_1(R_1), r_2(R_2), ..., r_n(R_n)$, to *słaby egzemplarz* relacji uniwersalnej dla bazy **r** jest to taka relacja u(U), gdzie $U = R_1 \cup R_2 \cup ... \cup R_n$, że

$\forall i \in \{1, ..., n\}\ r_i \subseteq \pi\downarrow_{R_i}(u)$

Krotka *t pokrywa* (podciąga) krotkę *v*, jeśli istnieje takie odwzorowanie h, które jest identycznością dla symboli niepustych oraz h(v) = t. Odwzorowanie h nazywane *pokrywającym*, zmniejsza liczbę symboli pustych w krotkach, przekształcając je w symbole niepuste. Ogólnie zbiór krotek S pokrywa zbiór krotek T, jeśli istnieje odwzorowanie pokrywające h, które odwzorowuje każdą krotkę zbioru T w krotkę zbioru S.

Charakterystyczny egzemplarz relacji uniwersalnej dla pewnej bazy danych jest to taki słaby egzemplarz s tej bazy przy uwzględnieniu zbioru zależności funkcyjnych F, że każdy słaby egzemplarz tej bazy pokrywa s.

Okno opisuje się w metodzie ZR korzystając z pojęcia warunku jednoznaczności i złączenia rozszerzonego.

Schemat R_i spełnia *warunek jednoznaczności*, jeśli dla każdego $j \neq i$ $(i, j = 1 \ldots n)$ domknięcie $(R_i)^+_{F-F_j}$ nie zawiera żadnego nadklucza schematu R_j. Zbiór Y jest nadkluczem schematu R_j, jeśli Y zawiera K oraz A takie, że $K \in K_j$ (tzn. K jest kluczem dla R_j) oraz $A \in R_j - K$.

Sagiv wykazał, że jeżeli relacje $r_1, r_2, \ldots, r_n$ nie zawierają wartości pustych, to przy każdej zawartości tych relacji charakterystyczny egzemplarz relacji uniwersalnej dla bazy danych r spełniający zależności F istnieje wtedy i tylko wtedy, jeśli każdy ze schematów R_i spełnia warunek jednoznaczności.

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

Wyrażenie $R_{i_1} \bowtie R_{i_2} \bowtie \ldots \bowtie R_{i_m}$ jest *rozszerzonym złączeniem* schematu R_{i_1}, jeśli dla każdego $1 \leq j < m$ zbiór $R_{i_1} \cup R_{i_2} \cup \ldots \cup R_{i_j}$ zawiera klucz schematu $R_{i_{j+1}}$. Przy wyznaczeniu [X] interesujące są rozszerzone złączenia pokrywające (zawierające) występujący w pytaniu podzbiór atrybutów X. Minimalnym rozszerzonym złączeniem pokrywającym X jest takie rozszerzone złączenie $E = R_{i_1} \bowtie R_{i_2} \bowtie \ldots \bowtie R_{i_m}$, że nie istnieje żadne inne rozszerzone złączenie schematu R_{i_1} pokrywające X, którego argumenty byłyby właściwym podzbiorem zbioru schematów $\{R_{i_1}, R_{i_2}, \ldots, R_{i_m}\}$.

Sagiv [4] wykazał, że jeśli dla pewnej bazy danych istnieje charakterystyczny egzemplarz spełniający zbiór zależności F (tzn. jeśli wszystkie schematy relacji spełniają warunek jednoznaczności), to poszukiwana funkcja okna [X] jest sumą projekcji na X wszystkich minimalnych złączeń pokrywających X, czyli

$$[X] = \pi_X(E_1) \cup \pi_X(E_2) \cup \ldots \cup \pi_X(E_p).$$

Algorytm wyznaczania [X] według powyższej zależności został przedstawiony przez Sagiva [4].

Odpowiedź na pytanie zawiera więc zawsze sumę odpowiedzi dla wszystkich możliwych interpretacji pytania.

Tworzenie opisu bazy dla tej metody wymaga wskazania relacji bazy r, zdefiniowania kluczy tych relacji i zbadania, czy wszystkie schematy relacji bazy r spełniają warunek jednoznaczności.

Przykład 2. Wyznaczenie okna metodą ZR dla zapytania (4) do bazy UCZELNIA:

select Przedmiot where Wydział = 'Mech.'

Wszystkie schematy bazy spełniają warunek jednoznaczności.

Atrybuty użyte w pytaniu: $X = P_tW$

Złączenia rozszerzone pokrywające P_tW:

PLAN

ANGAŻE ⋈ ZAJĘCIA

ANGAŻE ⋈ ZAJĘCIA ⋈ PLAN

Minimalne złączenia rozszerzone pokrywające P_tW:

E_1 = PLAN

E_2 = ANGAŻE ⋈ ZAJĘCIA

Okno: $[X] = [P_tW] = \pi_{P_tW}(\text{PLAN}) \cup \pi_{P_tW}(\text{ANGAŻE} \bowtie \text{ZAJĘCIA})$

Interpretacja pytania (4): interpretacja C.

5.2.2.3. Metoda obiektów maksymalnych

Metoda obiektów maksymalnych (zwana dalej metodą OM) przedstawiona jest przez Ullmana w [6 i 7]. Bazuje ona na pojęciu obiektu.

Obiekt jest to minimalny zbiór atrybutów powiązany elementarnym (nierozkładalnym) związkiem. Relację uniwersalną opisuje wyrażenie:

$\{ a_1, a_2, ..., a_m \mid P_1(O_1) \wedge P_2(O_2) ... \wedge P_p(O_p) \}$

gdzie

$a_1, a_2, ..., a_m$ - wartości atrybutów bazy,

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

Relacja uniwersalna dla bazy danych UCZELNIA jest zdefiniowana następująco:

{ P, W, P_t, G | 'Pracownik P jest zatrudniony na wydziale W' ∧ 'Pracownik P prowadzi zajęcia z przedmiotu P_t' ∧ 'Przedmiot P_t jest prowadzony dla studentów wydziału W w wymiarze G godzin' }

Obiekty maksymalne są to acykliczne zbiory $M_1, M_2, ..., M_m$ obiektów określonych w relacyjnej bazie danych r(R). Zbiory te muszą spełniać trzy własności:

W1. W każdym obiekcie maksymalnym M_i musi być zapewniona zależność złączeniowa dla wszystkich możliwych złączeń w M_i.

W2. Każdy obiekt maksymalny $M_i \in r$ jest największą możliwą, spójną i acykliczną sumą obiektów, która spełnia własność W1.

W3. Suma wszystkich obiektów maksymalnych jest równa bazie danych.

Metoda OM umożliwia jednoznaczną interpretację zadania wyszukiwania w bazach cyklicznych. Ogranicza ona w bazach cyklicznych zakres pytania. Zakłada się, że pytanie ma sens tylko wtedy, gdy zbiór X użytych w nim atrybutów zawiera się całkowicie

w którymś z obiektów maksymalnych bazy (takich obiektów maksymalnych może być kilka). Przyjmuje się z definicji, że każda baza acykliczna składa się z jednego fragmentu. Ullman [6, 7] podaje algorytm tworzenia "kandydata" na obiekt maksymalny. "Kandydat" spełnia własność W1, a nie musi spełniać własności W2.

Algorytm 2. Tworzenie "kandydata" na obiekt maksymalny M'.

Użyta funkcja: ATTR(M') - atrybuty obiektów należących do M'.

Q, P - obiekty

<u>początek</u>

M' := {Q} ;

<u>dopóki</u> zachodzą zmiany w M' <u>wykonuj</u>

<u>dla</u> każdego obiektu P, takiego że

<u>albo</u> ((P ∩ ATTR(M')) -> P <u>lub</u> (P ∩ ATTR(M')) -> ATTR(M'))

<u>albo</u> P - ATTR(M') jest rozłączne z ATTR(M') - P,

gdy P ∩ ATTR(M') jest usunięte z hipergrafu

<u>wykonuj</u>

M' := M' ∪ {P}

<u>koniec</u>

Algorytm 2 należy wykonać tyle razy, ile jest obiektów w bazie, biorąc za każdym razem jako Q inny obiekt. Spośród powstałych "kandydatów" na obiekt maksymalny należy usunąć cykliczne oraz będące podzbiorami innych. Każdy "kandydat", który pozostał, uważany jest za obiekt maksymalny.

Korth [2] opisał implementację algorytmu tworzenia obiektów maksymalnych bazującą na idei algorytmu 2. Kozielski, opierając się na pracy Kortha [2], zaproponował w [1] przyjęcie następujących uproszczeń:

U1. Obiektami są relacje.

U2. Pod uwagę brane są zależności funkcyjne wynikające z podania klucza relacji.

U3. Odpowiednikami obiektów maksymalnych są *fragmenty*, czyli zbiory relacji bazy danych r. Własności fragmentów są analogiczne do własności obiektów maksymalnych.

Ponieważ założenie U1 nie jest spełnione jedynie w bardzo specyficznych przypadkach, a założenie U2 jest ogólnie przyjęte jako wystarczający sposób definiowania zależności funkcyjnych, więc w dalszych rozważaniach metoda OM będzie omawiana po zastosowaniu uproszczeń U1, U2, U3 i oznaczana OMF.

Tworzenie opisu bazy dla metody OMF wymaga wskazania relacji bazy r oraz ich kluczy. Opis bazy musi również zawierać opis podziału bazy na fragmenty. Podziału tego należy dokonać na etapie tworzenia opisu (można automatycznie - stosując algorytm 2). Odpowiedź na pytanie jest generowana według algorytmu 3.

Algorytm 3. Wyznaczenie odpowiedzi na zapytanie metodą OMF.

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

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

Jednym ze sposobów redukcji fragmentu w kroku 2 algorytmu 3 jest zmodyfikowana redukcja Grahama. Modyfikacja redukcji Grahama polega na tym, że z hipergrafu nie wolno usuwać schematów relacji, które zawierają atrybuty ze zbioru X nie należące do innej relacji.

Sposób interpretacji pytania zależy od tego, czy istnieją fragmenty zawierające zbiór X. Jeżeli nie istnieją, to odpowiedź jest pusta. Jeżeli istnieją, to wynik jest sumą odpowiedzi na pytanie w poszczególnych fragmentach. Jeżeli w danym fragmencie możliwe są różne interpretacje pytania, to wybierana jest taka, która wynika z przyjętego algorytmu redukcji fragmentu.

Przykład 3. Wyznaczenie okna metodą OMF dla zapytania (4) do bazy UCZELNIA:

select Przedmiot where Wydział = 'Mech.'

Fragmenty bazy otrzymane algorytmem 2:

Fragment 1: ANGAŻE, ZAJĘCIA

Fragment 2: PLAN

Atrybuty użyte w pytaniu: $X = P_tW$

Fragmenty zawierające P_tW:

Fragment 1,

Fragment 2.

Złączenia relacji w reduktach fragmentów:

ANGAŻE $\bowtie$ ZAJĘCIA

PLAN

Okno: $[X] = [PW] = \pi_{P_tW}(\text{ANGAŻE} \bowtie \text{ZAJĘCIA}) \cup \pi_{P_tW}(\text{PLAN})$

Interpretacja pytania (4): interpretacja C.

6. Podsumowanie

Wzrost poziomu języka zapytań oznacza przenoszenie elementów kontekstu z pytania do opisu bazy. Badając dla relacyjnego modelu danych współzależność poziomu języka zapytań i złożoności opisów baz danych można zauważyć, że relacja uniwersalna stanowi najwyższy poziom języka zapytań w sensie najmniejszej liczby elementów kontekstu w pytaniu.

W relacji uniwersalnej opis bazy w zasadniczy sposób decyduje o interpretacji pytania. W metodzie LWR okno [X] zależy od kolejności wyrażeń w liście wyrażeń relacyjnych. Gdyby w przykładzie 1 pierwszym elementem listy było wyrażenie ANGAŻE ⋈ ZAJĘCIA, to [X] miałoby wartość π_{P_tW}(ANGAŻE ⋈ ZAJĘCIA). W metodzie ZR okno zależy od tego, jakie atrybuty tworzą klucze w relacjach. W metodzie OMF wartość [X] zależy od przynależności relacji do fragmentów. Jeżeli w przykładzie 3 użytkownik relacji uniwersalnej "poprawi" (wolno mu !) podział na fragmenty w taki sposób, że Fragment 1 = {ANGAŻE}, Fragment 2 = {ZAJĘCIA}, Fragment 3 = {PLAN}, to [X] = PLAN, ponieważ X = P_tW zawiera się jedynie we Fragmencie 3.

Celem "oczyszczenia" pytania z elementów kontekstu było przybliżenie się do idealnego języka zapytań wspomnianego w rozdz. 1.

Opisany język relacji uniwersalnej nie jest jedynym możliwym środkiem przybliżającym do tego celu. Jest kolejnym szczeblem rozwoju języków operujących na strukturach relacji i ewentualnie niejawnie uwzględniających semantyczne zależności między danymi. Przedstawiony język relacji uniwersalnej jest daleki od ideału języka zapytań z powodu wielu niedogodności. Dwie z nich są najistotniejsze.

A. Spełnienie założenia jednoznacznej interpretacji zadania wyszukiwania jest mało zadowalające, zwłaszcza dla baz cyklicznych. Interpretacja pytania jest albo narzucona (jak w metodach LWR i ZR), albo do opisu bazy wprowadzane są dodatkowe pojęcia (jak w metodzie OM). Aby uzyskać wymaganą interpretację pytania, należy poszerzyć zbiór X do takiego zbioru atrybutów, który wymusza potrzebną interpretację pytania. Można to zrobić albo poszerzając zbiór atrybutów szukanych, albo dodając nowe warunki (co na ogół wymaga użycia zmiennych zakresu).

B. Formułowanie w języku zapytań relacji uniwersalnej pytań operujących związkami nie odzwierciedlonymi wprost w bazie danych wymaga użycia zmiennych zakresu i stanowi dużą trudność pojęciową dla niewprawnego użytkownika. Przy-

kładem jest pytanie: "Podaj pracowników, którzy mają zajęcia na wydziałach innych niż te, na których znajdują się zatrudniające ich instytuty".

Można szukać rozwiązania tych problemów, rozwijając opisane w artykule metody wyznaczania okna.

Na przykład Ullman ([6, 7]), aby uniknąć problemów związanych z cyklicznością bazy, proponuje obok wprowadzenia obiektów maksymalnych również tzw. powielanie atrybutów, czyli nadanie atrybutom zamykającym cykle różnych nazw w różnych relacjach, tak aby cykliczność bazy została usunięta. Użytkownik relacji uniwersalnej byłby wówczas uwolniony od konieczności manipulowania obiektami maksymalnymi (fragmentami) w metodzie OM (OMF), ponieważ cała relacja uniwersalna składa się z dokładnie jednego obiektu maksymalnego (fragmentu), gdy baza jest acykliczna. Na przykład w bazie danych UCZELNIA można w schemacie relacji ANGAŻE zamiast nazwy atrybutu Wydział użyć nazwy WydziałA, a w schemacie relacji PLAN użyć nazwy WydziałP. Schemat bazy danych UCZELNIA byłby wówczas acykliczny. Jednakże rzeczywiste powielanie atrybutów można zrealizować jedynie na etapie projektowania bazy danych, ponieważ manipulowanie schematem już funkcjonującej bazy stwarza liczne problemy. Dlatego Autor artykułu proponuje w swoich pracach przejście od cyklicznej bazy danych do acyklicznej, definiując wirtualny schemat bazy poprzez nadanie tzw. atrybutom niejednoznacznym (czyli mającym takie same nazwy, ale różne znaczenie w różnych relacjach) różnych nazw dla różnych relacji. Użytkownik relacji uniwersalnej operuje wówczas w procesie wyszukiwania jej schematem wirtualnym (z przemianowanymi atrybutami), a wyszukiwanie realizowane jest w rzeczywistych relacjach. System oparty na tym założeniu jest obecnie realizowany jako rozwinięcie systemu UNIVERS opisanego w pracy [1].

Obiecujące wydaje się również uzupełnienie pytania o opis związków między danymi. Powołując się na związek między danymi można wskazać sposób interpretacji pytania lub ominąć konieczność użycia zmiennych zakresu. Oczywiście, związki te muszą być zdefiniowane w czasie tworzenia opisu bazy i zapamiętane w nim. Problemem staje się wówczas sposób zdefiniowania tych związków.

Im wyższy więc poziom języka zapytań i większe jego podobieństwo do języka naturalnego, tym mniej elementów kontekstu znajduje się w pytaniu i tym bardziej rozbudowane są środki opisu bazy.

LITERATURA

[1] Grzywocz J., Kozielski S., Piec J., Skowronek M.: System programowy wspomagający procesy analizy, tworzenia i weryfikacji hipotez o związkach i zależnościach między elementami relacyjnych baz danych. Raport CPBR nr 8.6, Gliwice 1988.

[2] 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.

[3] Kozielski S.: Języki zapytań w mikrokomputerowych bazach danych. Referat na seminarium Instytutu Informatyki Politechniki Śl. (matriały nieopublikowane), marzec 1992.

[4] 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.

[5] Tsichritzis D. C., Lochovsky F. H.: Modele danych. WNT, Warszawa 1990.

[6] Ullman J. D.: Systemy baz danych. WNT, Warszawa 1988.

[7] Ullman J. D.: Principles of Database and Knowledge-Base Systems. Computer Science Press, New York 1989.

Recenzent: Doc. dr hab. inż. Mieczysław Muraszkiewicz

Wpłynęło do Redakcji 16 lipca 1993 r.

Abstract

This papier is a review of an interdependence of a query language level and of a complexity of a database description. This interdependence was presented for the following query language levels: a tuple manipulation level, relational algebra level, a domain and tuple relational calculus, and for an universal relation. The basis notions for databases were defined in the chapter 3. In the first part of the chapter 4 elements, which must be determined so that the system may answer a question, (the target elements and context elements), were presented. Next, the division of these elements between a query and database description for the different query language levels was discussed. The

examples of database management systems for all query language levels, in which they are implemented, were given. It was shown that the query language level increases in accordance with the number of context elements transfered from the query to the databases description. It was shown that the universal relation constitutes the highest level of the query language in the sense of the smallest number of context elements in the query. For the universal relation the influence of the database description on the solution of searching problem was presented in the chapter 5.