ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1994

Seria: INFORMATYKA z. 27 Nr kol. 1259

Alicja BORUTA
Jerzy GRZYWOCZ
Stanisław KOZIELSKI

WYKORZYSTANIE ELEMENTÓW JĘZYKA NATURALNEGO W SYSTEMIE WYSZUKIWANIA OPARTYM NA MODELU RELACJI UNIWERSALNEJ

Streszczenie. W pracy przedstawiono system umożliwiający formułowanie pytań do bazy danych, wykorzystujący definiowany zestaw fraz języka naturalnego. W procesie interpretacji pytań zastosowano model relacji uniwersalnej.

USING NATURAL LANGUAGE ELEMENTS IN QUERY SYSTEM BASED ON THE UNIVERSAL RELATION MODEL

Summary. Query system using set of defined natural language phrases is presented in the paper. Universal relation model is applied in the query interpretation model.

DIE AUSNUTZUNG DER ELEMENTE DER NATURSPRACHE IN AUF DEM MODELL DER UNIVERSALRELATION BASIERTEM AUSSUCHSYSTEM

Zusammenfassung. Im Artikel wurde ein System dargestellt, das auf Basis definierter Satzzusammenstellung der Natursprache die Datenbankabfragenformulierung ermöglicht. Im Abfrageninterpretationsprozeß wurde das Modell der Universalrelation ausgenutzt.

Pracę wykonano w ramach grantu KBN nr 8 0269 91 01

1. Cel i zakres pracy

W niniejszym artykule przedstawiono ogólną strukturę system wyszukiwania danych umożliwiającego formułowanie pytań przy użyciu języka zbliżonego do języka naturalnego. System ten może stanowić odpowiednie narzędzie dla użytkownika, któremu obce są formalizmy związane z językami zapytań.

We wstępnym etapie pracy użytkownik ma możliwość definiowania fraz języka naturalnego, opisujących wykorzystywaną bazę danych. W trakcie zwykłej eksploatacji systemu zdefinowany uprzednio zbiór fraz jest wykorzystywany do formułowania pytań.

Zakres analizy językowej ograniczony został do sprawdzania, czy zadane pytanie - przy spełnieniu ogólnych reguł języka - zbudowane zostało z fraz wcześniej przez użytkownika zdefiniowanych.

Zrealizowany system wyszukiwania bazuje na modelu relacji uniwersalnej. Dlatego w kolejnym rozdziale przedstawiona zostanie krótka charakterystyka tego modelu oraz opartego na nim preprocesora pytań.

2. Preprocesor pytań oparty na modelu relacji uniwersalnej

Koncepcja relacji uniwersalnej bazuje na relacyjnym modelu danych. Podstawowe pojęcia tego modelu (np. relacja, schemat relacji, atrybut, operatory algebry relacji) zdefiniowane są w wielu pracach, np. [4, 11, 12]. Relacja uniwersalna jest abstrakcyjnym opisem bazy danych, w którym cała baza widziana jest jako jedna relacja obejmująca wszystkie atrybuty istniejących relacji. Dla pewnej bazy danych $r = \{ r_1, r_2, ..., r_n \}$ o schemacie $R = \{R_1, R_2, ..., R_n\}$ relacja uniwersalna jest więc hipotetyczną relacją u utworzoną nad zbiorem atrybutów $U = R_1 \cup R_2 \cup ... \cup R_n$ z zawartości relacji $r_1, r_2, ..., r_n$. Interpretacja zawartości takiej relacji zależy od przyjęcia szeregu założeń, dyskutowanych np. w pracach [3, 5, 7, 10, 11]. W niniejszym artykule skupimy się jedynie na prezentacji elementarnych własności języka zapytań, który można zdefiniować dla takiego opisu bazy. Najważniejszą z tego punktu widzenia własnością modelu relacji uniwersalnej jest automatyczne ustalanie logicznych powiązań między relacjami tworzącymi ten model, według identycznych nazw atrybutów w poszczególnych relacjach. W celu zilustrowania tej możliwości wygodne jest przedstawienie bazy danych w postaci hipergrafu. Krawędzie hipergrafu utożsamia się wtedy

z tzw. obiektami [7] lub wprost z relacjami [5, 6], natomiast wierzchołki z atrybutami. Rozważmy przykładową bazę danych obejmującą trzy relacje:

Studenci(nazwisko, imię, kierunek, semestr, nr_indeksu),

Oceny(nr_indeksu, ident_przedmiotu, ocena),

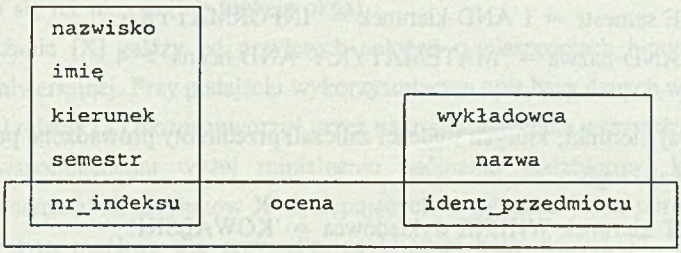
Przedmioty(ident_przedmiotu, nazwa, wykładowca).

Bazę tę można opisać hipergrafem H = (N, E), gdzie N i E są odpowiednio zbiorami wierzchołków i krawędzi o następującej postaci:

N = {nazwisko, imię, kierunek, semestr, nr_indeksu, ident_przedmiotu, ocena, nazwa, wykładowca},

E = {{nazwisko, imię, kierunek, semestr, nr_indeksu}, {nr_indeksu, ident_przedmiotu, ocena}, {ident_przedmiotu, nazwa, wykładowca}}.

Graficzną ilustrację takiego hipergrafu przedstawia rys. 1.



Rys. 1. Opis przykładowej bazy danych w postaci hipergrafu
Fig. 1. Description of the examplary database in the form of hypergraph

Wierzchołkami wiążącymi poszczególne krawędzie tego hipergrafu są wspólne atrybuty. Taka reprezentacja bazy danych umożliwia osiągnięcie podstawowej własności języka zapytań relacji uniwersalnej, jaką jest zwolnienie użytkownika-programisty z konieczności wskazywania relacji wykorzystywanych w pytaniu oraz powiązań między nimi. Relacje te mogą być bowiem ustalone samodzielnie przez system interpretujący pytanie użytkownika poprzez wybór minialnego spójnego podzbioru krawędzi hipergrafu obejmującego zbiór atrybutów występujących w pytaniu.

Formalne podstawy języka dla modelu relacji uniwersalnej stwarza rachunek relacji. Składnia tego języka może więc być wyprowadzona ze składni takich języków jak QUEL czy SQL. Należy przy tym zauważyć, że definicja samej relacji uniwersalnej wprowadza domyślnie do procesu interpretacji pytania listę wszystkich tworzących tę relację plików wraz z warunkami określającymi powiązania między plikami zawierającymi identyczne atrybuty. Zwalnia to użytkownika od konieczności umieszczania tych elementów w pytaniu. Sformułowanie

wanie pytania ogranicza się wtedy do wskazania nazw wyszukiwanych danych (atrybutów) oraz określenia cech, jakie powinny spełniać.

Dla zilustrowania sposobu formułowania pytań w języku opartym na modelu relacji uniwersalnej rozpatrzymy kilka wybranych zadań wyszukiwania dotyczących zdefiniowanej uprzednio bazy danych .

Przykłady formułowania pytań (zadań wyszukiwania):

P1) Znajdź nazwiska studentów semestru 3 kierunku Informatyka.

SELECT nazwisko WHERE semestr = 3 AND kierunek = 'INFORMATYKA'

P2) Znajdź nazwiska studentów semestru 1 kierunku Informatyka, którzy z przedmiotu matematyka otrzymali oceny większe od 4. Znajdź też te oceny.

SELECT nazwisko, ocena
WHERE semestr = 1 AND kierunek = 'INFORMATYKA'
AND nazwa = 'MATEMATYKA' AND ocena > 4

P3) Podaj kierunki, których studenci zaliczali przedmioty prowadzone przez wykładowcę Kowalskiego.

SELECT kierunek WHERE wykładowca = 'KOWALSKI'

Jak łatwo zauważyć, składnia języka stosowanego do zapisu powyższych przykładów została zapożyczona z języka SQL. Zgodnie z poprzednimi uwagami w zapisie pytań nie występuje fraza FROM oraz warunki łączące tablice (pliki) we frazie WHERE.

Model relacji uniwersalnej umożliwia też m. in. formułowanie pytań w przypadku, kiedy nazwa atrybutu występuje w pytaniu więcej niż jeden raz i to w różnym znaczeniu. Atrybuty te są wtedy rozróżniane za pomocą tzw. zmiennych zakresu. Ilustruje to poniższy przykład.

P4) Znajdź nazwy innych przedmiotów, które prowadzi wykładowca przedmiotu elektrotechnika.

SELECT a.nazwa WHERE b.nazwa = 'ELEKTROTECHNIKA'
AND b.wykładowca = a.wykładowca

Interpretacja pytań dla modelu relacji uniwersalnej

Zauważmy, że ogólną postać prezentowanych pytań (zadań wyszukiwania) możemy przedstawić następująco:

SELECT Q WHERE warunki(P)

gdzie Q - lista nazw atrybutów szukanych,

warunki(P) - wyrażenie logiczne wiążące zbiór atrybutów P.

Oznaczmy zbiór wszystkich atrybutów występujących w pytaniu przez X, czyli $X = Q \cup P$.

Do podania odpowiedzi na pytanie skierowane do relacji uniwersalnej trzeba wykorzystać relację określoną na schemacie X, dokonując selekcji jej zawartości według warunków zawartych w pytaniu. Operację tę można zapisać następująco:

$$q(Q) = \pi_Q(\sigma_{warunki(P)} [X]),$$

gdzie π i σ są operatorami projekcji i selekcji ([11]).

Relację [X] nazywa się też oknem (lub funkcją okna).

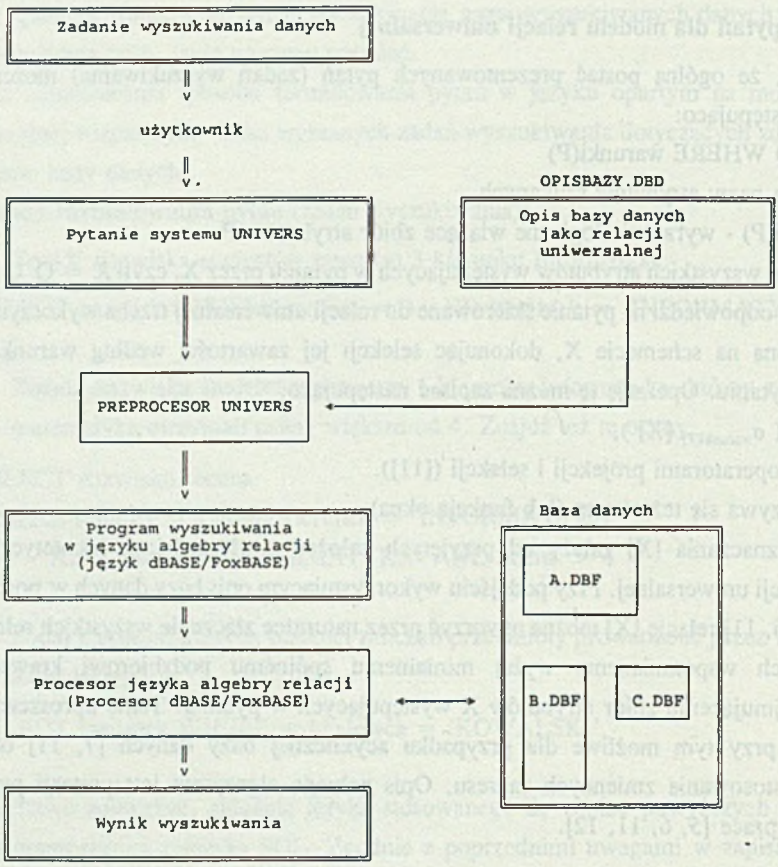
Sposób wyznaczania [X] zależy od przyjętych założeń o własnościach hipotetycznej zawartości relacji uniwersalnej. Przy podejściu wykorzystującym opis bazy danych w postaci hipergrafu [5, 6, 11] relację [X] można utworzyć przez naturalne złączenie wszystkich relacji odpowiadających wspomnianemu wyżej minialnemu spójnemu podzbiorowi krawędzi hipergrafu obejmującemu zbiór atrybutów X występujących w pytaniu. Takie uproszczone podejście jest przy tym możliwe dla przypadku acyklicznej bazy danych [7, 11] oraz rezygnacji ze stosowania zmiennych zakresu. Opis pełnego algorytmu interpretacji pytań prezentują np. prace [5, 6, 11, 12].

2.1. Organizacja systemu UNIVERS bazującego na modelu relacji uniwersalnej

W pracy wykorzystano system wyszukiwania UNIVERS oparty na modelu relacji uniwersalnej [6]. Elementy tego systemu oraz powiązania między nimi przedstawiono na rys. 2. Całość obejmuje dwie części:

1) preprocesor UNIVERS umożliwiający formułowanie pytań w języku opartym na modelu relacji uniwersalnej i generujący program wyszukiwania w języku poziomu algebry relacji (język baz danych typu dBASE),
2) procesor baz danych dBASE III PLUS lub FoxBASE realizujący program wyszukiwania.

Preprocesor UNIVERS został zaprojektowany jako program nie ingerujący w postać bazy danych i pracę innych programów aplikacyjnych współpracujących z daną bazą. W trakcie instalowania systemu UNIVERS tworzony jest plik opisujący bazę danych w kategoriach relacji uniwersalnej. System umożliwia formułowanie pytań do bazy w sposób konwersacyjny,



Rys. 2. Ilustracja kolejnych etapów interpretacji pytań z wykorzystaniem systemu UNIVERS
Fig. 2. Outline view of successive stages of query interpretation using UNIVERS system

maksymalnie przyjazny użytkownikowi. Efektem interpretacji pytania jest program wyszukiwania zapisywany na poziomie algebry relacji w postaci akceptowanej przez procesory baz danych typu dBASE. Program przekazywany jest procesorowi bazy poprzez plik dyskowy.

2.2. Związek języka relacji uniwersalnej z językiem naturalnym; istota prezentowanego systemu

Zwróćmy uwagę na pewną analogię między formułowaniem pytań w języku naturalnym a zapisem tych pytań w języku relacji uniwersalnej. Otóż zakres informacji występujący w zapisie pytania w języku relacji uniwersalnej jest zbieżny z zakresem informacji użytym

w języku naturalnym. Różna jest natomiast składnia formułowanych pytań. Zauważmy jednak, że dla określonej bazy danych można się spodziewać wielu podobnych lub identycznych fraz w zadawanych pytaniach. Na tym opiera się koncepcja prezentowanego systemu. Zakłada ona mianowicie, że system ten jest każdorazowo adaptowany do nowej bazy danych poprzez określenie zbioru typowych fraz występujących w charakterystycznych pytaniach kierowanych do tej bazy. Frazy te są pamiętane w słownikach bazy.

W trakcie interpretacji pytań pierwszym etapem jest próba dopasowania fraz ze słowników bazy oraz nazw atrybutów występujących w bazie do zadanego pytania. Jeśli zakończy się ona pomyślnie, pytanie jest przekształcane do postaci wymaganej przez system wyszukiwania UNIVERS, który generuje na jego podstawie program wyszukiwanie danych. W przeciwnym przypadku użytkownik będzie musiał zmodyfikować sformułowanie pytania, korzystając np. z dostępnych na ekranie zawartości słowników gotowych fraz.

Taki sposób interpretacji pytań jest istotnie różny od analizy stosowanej dla języka naturalnego [2, 8]. Jednak zewnętrzna forma (dopuszczalnych) pytań jest temu językowi bliska. Stąd też proponowany w tej pracy język będziemy dalej nazywali (dla uniknięcia nieporozumień) językiem pseudonaturalnym.

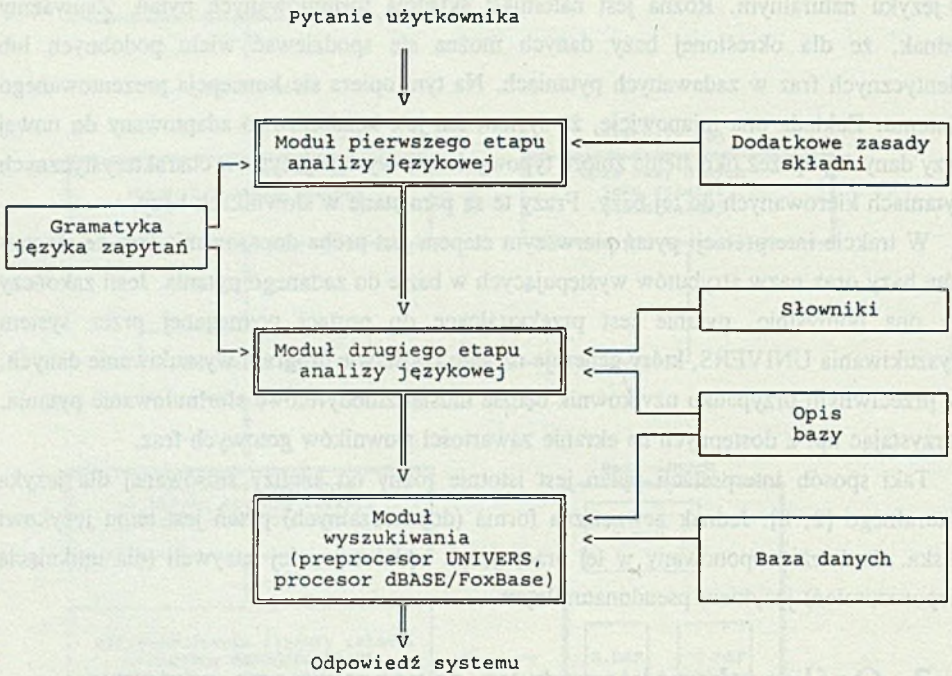
3. Ogólna charakterystyka opracowanego systemu wyszukiwania

System ten umożliwia dostęp do baz danych utworzonych przy wykorzystaniu systemów xBase (dBASE III PLUS, Clipper, FoxBase). Użytkownik może uzyskać dane z bazy, formułując swoje pytanie w zdefiniowanym dalej języku pseudonaturalnym (przez "pytanie" będziemy dalej rozumieli sformułowanie zadania wyszukiwania danych).

Składniki tego systemu oraz przepływ informacji między nimi przedstawiono na rys. 3.

Pytanie użytkownika jest najpierw przetwarzane przez moduł pierwszego etapu analizy językowej. Na tym etapie sprawdza się poprawność sformułowanego pytania, tzn. czy pytanie jest zbudowane z dozwolonych znaków oraz czy zachowane są podstawowe zasady składni przyjętego języka pseudonaturalnego. Wynikiem tej analizy jest lista słów tworzących pytanie.

Moduł drugiego etapu analizy językowej na podstawie listy słów, gramatyki języka, opisu bazy danych oraz zawartości słowników tworzy zdanie w języku pośrednim. Zdanie to jest wynikiem interpretacji pytania użytkownika. Zdanie to jest przekazywane do modułu wyszukiwania, który traktuje je jako sformalizowane zadanie wyszukania informacji. W skład modułu wyszukiwania wchodzą: preprocesor pytań kierowanych do bazy danych, oparty na modelu relacji uniwersalnej (okrojona wersja systemu UNIVERS przedstawionego



Rys. 3. Struktura systemu wyszukiwania danych z dostępem w języku pseudonaturalnym
Fig. 3. General view of the query system using pseudonatural language

w punkcie 2) oraz procesor bazy danych dBase lub FoxBase. Efektem działania preprocesora jest utworzenie w języku dBase programu wyszukania. Program ten jest następnie wykonywany przez procesor bazy danych, co prowadzi do wyszukania żądanych danych.

Do zainstalowania omawianego systemu w określonej bazie danych wymagane jest, aby administrator (użytkownik) bazy jednorazowo zdefiniował opis bazy danych widzianej jako wspomniana relacja uniwersalna. Proces definiowania wymaga wskazania plików tworzących bazę danych, a w przypadkach, kiedy baza jest bazą cykliczną (istnieje wiele dróg rozwiązania zadań wyszukiwania) również określenia obiektów maksymalnych [6, 7] (nazywanych w tym systemie fragmentami). Proces ten w większości przypadków może być realizowany automatycznie przez sam system.

Ponadto wymaga się, aby wykorzystywane w drugim etapie analizy językowej słowniki zawierały określone dalej informacje. Informacje te mogą być wprowadzane do słowników przed rozpoczęciem wyszukiwania danych albo w trakcie formułowania zapytań.

System składa się z kilku programów. Przepływ informacji między tymi programami jest zrealizowany przy użyciu plików buforowych. Taki podział systemu umożliwia rozbudowę

poszczególnych modułów bez konieczności modyfikowania pozostałych. Ponadto w razie potrzeby system może być realizowany na dwóch komputerach.

4. Język formułowania pytań

Definicja języka pseudonaturalnego zostanie podana w dalszej kolejności, a najpierw przeanalizujemy kilka przykładów, prezentujących sposób zapisu zadań wyszukiwania danych w:

- języku naturalnym,
- języku zapytań stosowanym w systemie UNIVERS,
- przyjętym języku pseudonaturalnym.

Rozpatrywane dalej przykłady będą dotyczyły bazy danych *PRACOWNICY NAUKOWI* obejmującej następujące relacje:

INSTYTUTY(instytut, dyrektor, wydział),
PRACOWNICY(nr_prac, nazwisko, data_ur, instytut),
WYKŁADY(nr_prac, przedmiot, liczba_g),
TEMATY(nr_tematu, nazwa, kierownik),
DOCHODY(nr_prac, nr_tematu, kwota).

Dane w tych relacjach dotyczą pracowników uczelni i prowadzonych przez nich zajęć, tematów realizowanych prac badawczych oraz sumarycznych dochodów uzyskanych przez pracowników za realizację określonych tematów.

Przykład 1

(1) Zapis zadania w języku naturalnym.

Znajdź nazwiska tych pracowników instytutu Informatyki, którzy brali udział w realizacji tematów kierowanych przez Miklaszewskiego. Podaj również nazwy tych tematów oraz zarobione kwoty.

(2) Zapis zadania w języku zapytań systemu UNIVERS.

W systemie tym pytanie jest zapisywane w trybie konwersacyjnym w dwóch krokach:

(a) Określenie w oknie *Szukane dane* atrybutów (nazw danych), których wartości są wyszukiwane (odpowiednik frazy SELECT).

Szukane dane: nazwisko, nazwa, kwota

(b) Określenie w oknie *Warunki* zestawu warunków wyszukiwania, domyślnie wiązanych spójnikami AND (odpowiednik frazy WHERE).

Warunki: instytut = 'INFORMATYKA'
kierownik = 'MIKLASZEWSKI'

(3) Zapis w języku pseudonaturalnym.

Zapis ten może mieć następującą formę:

Znajdź nazwiska pracowników, nazwy realizowanych tematów, kwoty zarobione w tych tematach; nazwą instytutu jest 'Informatyka', kierownik tematu nazywa się 'Miklaszewski'.

Przykład 2

(1) Zapis zadania w języku naturalnym.

Podaj nazwiska pracowników urodzonych przed dniem 27 października 1957 roku, którzy pracują nad piątym tematem.

(2) Zapis zadania w języku zapytań systemu UNIVERS.

Szukane dane: nazwisko

Warunki: data_ur < CTOD('1957-10-27')

nr_tematu = 5

(3) Zapis zadania w języku pseudonaturalnym.

Podaj nazwiska pracowników; data urodzenia pracownika przypada przed 1957-10-27, numerem realizowanego tematu jest 5.

Przykład 3

(1) Zapis zadania w języku naturalnym.

Wyszukaj nazwiska wykładowców i nazwy prowadzonych wykładów.

(2) Zapis zadania w języku zapytań systemu UNIVERS.

Szukane dane: nazwisko, przedmiot

(3) Zapis zadania w języku pseudonaturalnym.

Wyszukaj nazwiska wykładowców, nazwy prowadzonych wykładów.

Należy zaznaczyć, że zapis powyższych pytań w języku pseudonaturalnym nie jest jedynym możliwym zapisem. W przykładzie 4. przedstawiono kilka różnych zapisów dla jednego pytania.

Przykład 4

(1) Zapis zadania w języku naturalnym.

Podaj nazwiska tych pracowników oraz nazwy tematów przez nich realizowanych, jeśli numer pracownika odpowiada numerowi realizowanego przez niego tematu.

(2) Zapis zadania w języku zapytań systemu UNIVERS.

Szukane dane: nazwisko, nazwa

Warunki: nr_prac = nr_tematu

(3) Zapis zadania w języku pseudonaturalnym.

(a) *Podaj nazwiska pracowników, nazwy realizowanych tematów; numer pracownika jest równy numerowi realizowanego tematu.*

(b) *Wyszukaj nazwiska wszystkich pracowników, nazwy opracowywanych tematów; numer pracownika jest taki sam jak numer tematu.*

(c) *Znajdź nazwiska pracowników, nazwy wszystkich tematów; numer realizowanego tematu jest równy numerowi pracownika.*

Zauważmy, że każde pytanie kierowane do systemu UNIVERS składa się z jednej lub dwóch części. Część pierwszą, występującą zawsze, stanowi fraza *Szukane dane*. Część drugą stanowi fraza *Warunki*, która występuje tylko wtedy, kiedy poszukiwane dane powinny spełniać jakieś warunki.

Fraza *Szukane dane* zbudowana jest z nazw (atrybutów) poszukiwanych danych, oddzielonych od siebie przecinkami. Natomiast fraza *Warunki* składa się z koniunkcji warunków prostych. W skład każdego warunku prostego wchodzą: nazwa atrybutu, operator porównania, nazwa atrybutu lub wartość atrybutu. Operator porównania opisuje związek, jaki zachodzi między dwoma atrybutami lub między atrybutem a jego wartością.
(W systemie UNIVERS budowa warunku prostego może być bardziej złożona, np. w warunku mogą występować wyrażenia m. in. zawierające funkcje dBASE/FoxBase. W rozpatrywanym języku pseudonaturalnym takich konstrukcji - dla uproszczenia - nie będziemy rozważali.)

Ponieważ pytanie w języku pseudonaturalnym jest przekształcane do postaci akceptowanej przez system UNIVERS, więc przyjęto podobną zasadę jego konstruowania. Zdanie formułujące pytanie może się składać z dwóch lub trzech fraz, zależnie od tego, czy poszukiwane dane muszą spełniać jakieś warunki. Każde zdanie zaczyna się od *frazy polecenia wyszukania danych* (np. hasła: "znajdź", "wyszukaj"), po którym występuje *fraza opisująca poszukiwane dane*. W typowym przypadku wystąpienia warunków, po frazie opisującej poszukiwane dane występuje *fraza opisująca warunki*, oddzielona od niej znakiem średnika. Zdanie kończy się znakiem kropki.

Formalną definicję języka pseudonaturalnego przedstawimy przy użyciu rozszerzonej notacji Backusa-Naura (MBNF).

Gramatykę [13] rozważanego języka pseudonaturalnego stanowi czwórka $G = \langle V, \Sigma, P, \sigma \rangle$, gdzie V jest alfabetem terminalnym języka, Σ - alfabetem pomocniczym, P - listą produkcji, σ - symbolem początkowym (głową) języka.

Alfabet terminalny języka pseudonaturalnego składa się z liter alfabetu języka polskiego, cyfr oraz następujących znaków: kropka, przecinek, myślnik, średnik i apostrof. Małe i duże litery nie są rozróżniane.

Alfabet pomocniczy stanowi zbiór napisów występujących w poniżej przedstawionej liście produkcji w postaci ujętej w nawiasy kątowe < >.

Symbolem początkowym języka jest *zdanie*.

Listę produkcji (opis składni) języka pseudonaturalnego przedstawia następujący ciąg zapisów:

<zdanie> ::= <polecenie> <lista_atrybutów_poszukiwanych>[; <lista_warunków>].
<polecenie> ::= <lista_wyrazów>
<lista_wyrazów> ::= <wyraz>{ <wyraz>}
<wyraz> ::= <ciąg_znaków>
<ciąg_znaków> ::= <znak>{<znak>}
<znak> ::= <litera> | <cyfra>
<litera> ::= A|Ą|B|C|...|z|ź|ż
<lista_atrybutów_poszukiwanych> ::= <opis_atrybutu>{, <opis_atrybutu>}
<opis_atrybutu> ::= <lista_wyrazów>
<lista_warunków> ::= <opis_atrybutu_1> <związek> <opis_atrybutu_2>
{, <opis_atrybutu_1> <związek> <opis_atrybutu_2>}
<opis_atrybutu_1> ::= <opis_atrybutu>
<związek> ::= <lista_wyrazów>
<napis> ::= '<lista_wyrazów>'
<opis_atrybutu_2> ::= <opis_atrybutu> | <liczba> | <data> |<napis>
<liczba> ::= <cyfra>{<cyfra>}
<cyfra> ::= 0|..|9
<data> ::= <rok>-<miesiąc>-<dzień>
<rok> ::= 1900|..|1999
<miesiąc> ::= 01|02|..|12
<dzień> ::= 01|02|..|31

Zdanie może być zapisane w kilku liniach. Linia musi zaczynać się od litery, cyfry, apostrofu lub spacji. *Wyraz, liczba, data* i *napis* muszą być w całości zapisane w jednej linii.

Opis znaczenia niektórych symboli pomocniczych

opis_atrybutu - hasło opisujące atrybut (np. "nazwisko kierownika").
polecenie - hasło opisujące polecenie (np. "wyszukaj").
związek - zapis relacji, jaka zachodzi pomiędzy dwoma atrybutami lub pomiędzy atrybutem a jego wartością.

lista_atrybutów_poszukiwanych - lista haseł opisujących atrybuty poszukiwane w pytaniu.
warunek - opis warunku, jaki muszą spełniać poszukiwane dane.
lista_warunków - ciąg warunków, jakie muszą spełniać poszukiwane dane.
zdanie - zadanie wyszukania w bazie danych.

Hasła opisujące opis_atrybutu, polecenia i związki znajdują się w specjalnych słownikach, których opis zamieszczono w punkcie 5.

5. Opis słowników

Zadania wyszukiwania (zdania) budowane są w języku pseudonaturalnym za pomocą trzech podstawowych jednostek: *polecenie, opis_atrybutu* i *związek*. Prezentowany system ułatwia użytkownikowi formułowanie pytań, umożliwiając mu wybór wzorców tych elementów z plików zwanych słownikami. Z każdą z tych podstawowych jednostek związany jest odrębny plik.

Plik pierwszy, nazywany słownikiem poleceń, zawiera hasła poleceń kierowanych do systemu. Pojedynczym elementem tego pliku jest rekord *Polecenie*, przedstawiony na rys. 4.

TREŚĆ

Rys. 4. Struktura rekordu *Polecenie*
Fig. 4. Form of the record *Command*

Pole *Treść* zawiera hasło opisujące atrybut. Przykładowa zawartość słownika poleceń przedstawiona została na rys. 5.

Treść
Wyszukaj
Wyszukaj dane na temat
Podaj informacje związane z
Znajdź

Rys. 5. Przykładowa zawartość słownika poleceń
Fig. 5. Example of the command dictionary contents

Pojedynczym elementem drugiego pliku, nazywanego słownikiem atrybutów, jest rekord *Atrybut* przedstawiony na rys. 6. Pole *Treść* zawiera hasło opisujące atrybut, a pole

Nazwa_atrybutu nazwę atrybutu (wykorzystywaną później przez procesor dBase) odpowiadającą temu hasłu.

TREŚĆ
NAZWA_ATRYBUTU

Rys. 6. Struktura rekordu *Atrybut*
Fig. 6. Form of the record *Attribute*

Przykładowa zawartość słownika atrybutów dla bazy danych *Pracownicy Naukowi* przedstawiona została na rys. 7.

Treść	Nazwa_atrybutu
nazwisko pracownika	nazwisko
nazwiska wykładowców	nazwisko
nazwy prowadzonych wykładów	przedmiot
nazwy realizbwanych tematów	nazwa
numer realizowanego tematu	nr_tematu
data urodzenia pracownika	data_ur
nazwa instytutu	instytut
kierownik tematu	kierownik

Rys. 7. Przykładowa zawartość słownika atrybutów
Fig. 7. Example of the attribute dictionary contents

Natomiast pojedynczym elementem trzeciego plik, nazywanego słownikiem związków, jest rekord *Związek* o strukturze przedstawionej na rys. 8.

TREŚĆ
SYMBOL_ZWIĄZKU

Rys. 8. Struktura rekordu *Związek*
Fig. 8. Form of the record *Relationship*

Pole *Treść* zawiera hasło opisujące związek, a pole *Symbol_związku* symbol operatora relacyjnego, związanego z tym hasłem. Dopuszczalnymi operatorami relacyjnymi są:

$<>$, $<$, $>$, $=$, $<=$, $>=$.

Przykładowa zawartość słownika związków dla bazy danych *Pracownicy Uczelni* przedstawiona została na rys. 9.

Treść	Symbol_związku
jest	=
nazywa się	=
przypada przed	<
jest równy	=
jest różny od	<>
wynosi więcej niż	>
jest mniejszy od	<

Rys. 9. Przykładowa zawartość słownika związków
Fig. 9. Example of the relationship dictionary contents

Przedstawione słowniki wymagają wstępnego załadowania wzorców. Może się to odbywać we wstępnym etapie przygotowania systemu do pracy bądź też w trakcie formułowania pierwszych pytań. Słowniki można aktualizować w dowolnym momencie pracy systemu w trakcie dialogu z użytkownikiem (poprzez wybór odpowiedniej opcji menu).

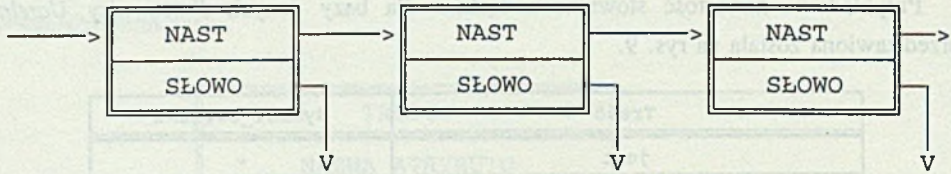
Dzięki zastosowaniu słowników system wyszukiwania z dostępem w języku pseudonaturalnym jest niezależny od dziedziny zastosowań, tzn. umożliwia formułowanie pytań do baz danych związanych z różnymi dziedzinami.

6. Analiza językowa

W prezentowanym systemie analiza językowa składa się z dwóch etapów. Na początku pierwszego etapu konwersacyjnie formułowane jest pytanie użytkownika (zdanie). W trakcie tego etapu jest analizowana poprawność pytania użytkownika (według algorytmu przedstawionego w punkcie 7). Sprawdza się, czy nie zostały naruszone reguły składni. W przypadku wystąpienia jakiegokolwiek błędu dalsza analiza pytania jest zawieszana, a system informuje użytkownika o przyczynie wstrzymania analizy.

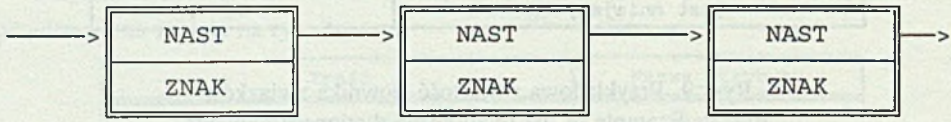
Wynikiem pierwszego etapu analizy językowej jest *lista słów* o strukturze przedstawionej na rys. 10.

W drugim etapie analizy językowej na podstawie *listy słów*, gramatyki języka pseudonaturalnego, opisu bazy danych oraz słowników generowane jest pytanie w języku pośrednim (według algorytmu przedstawionego w punkcie 7). Jako język pośredni przyjęto język



NAST - wskaźnik do następnego elementu *listy słów*,
SŁOWO - wskaźnik do *listy znaków*.

Rys. 10. Struktura *listy słów*
Fig. 10. Form of the *list of words*



NAST - wskaźnik do następnego elementu *listy znaków*,
ZNAK - znak, z którego zbudowane jest słowo.

Rys. 11. Struktura *listy znaków*
Fig. 11. Form of the *list of characters*

zapytań systemu UNIVERS. Wygenerowane pytanie w języku pośrednim zapisywane jest do roboczego pliku tekstowego.

W trakcie analizy sprawdza się, czy pytanie jest zbudowane zgodnie z regułami gramatyki języka pseudonaturalnego. Aby móc stwierdzić, czy zdanie jest poprawne, należy najpierw sprawdzić, czy hasła opisujące polecenie, atrybuty i związki znajdują się w słownikach. Jeżeli w pytaniu istnieje hasło, którego nie można znaleźć w słowniku, to dalsza analiza pytania jest wstrzymana. Użytkownik jest informowany o zaistniałej sytuacji i może wprowadzić brakujące hasła do słowników oraz rozpocząć drugi etap analizy językowej od początku.

Jednakże sam fakt, że wszystkie hasła, z których zbudowane jest pytanie, można znaleźć w słownikach, nie gwarantuje poprawnej interpretacji pytania. Może zajść przypadek, że atrybut nie jest dostępny w aktualnie wykorzystywanej bazie danych. Dlatego też, po każdorazowym odczytaniu hasła opisującego atrybut i znalezieniu go w słowniku atrybutów, odczytywana jest jego nazwa. Następnie, korzystając z opisu bazy, sprawdza się, czy atrybut jest dostępny w aktualnej bazie. Jeżeli okaże się, że atrybut jest niedostępny, to moduł analizy językowej informuje o tym użytkownika.

Końcowym wynikiem drugiego etapu analizy językowej jest pytanie w języku pośrednim, języku zapytań systemu UNIVERS.

W zapisie pytań w języku pseudonaturalnym, podobnie jak w systemie UNIVERS, jednym z podstawowych wymagań jest zgodność typów porównywanych ze sobą członów warunków. Typy te muszą być zgodne z typami danych systemów dBASE/FoxBase, do których należą: *C* — typ tekstowy, *N* — typ numeryczny, *L* — typ logiczny, *D* — typ daty (nie dopuszcza się w zapisie warunków typu *M* — notatnikowego). Następujący przykład pokazuje poprawne konstrukcje warunków pod względem zgodności typów.

Przykład 5

Rozpatrujemy bazę danych *PRACOWNICY NAUKOWI*, przy założeniu że relacja *Pracownicy* zawiera dodatkowe pole *kobieta* typu logicznego, określające, czy pracownik jest kobietą. Natomiast relacja *Tematy* zawiera dodatkowo pole *data_ur_k* typu data, określające datę urodzenia kierownika. W celu zwiększenia czytelności hasło opisujące związek jest wyróżnione podkreśleniem.

Poprawnymi warunkami są:

(1) dla typu *C*

pracownik nazywa się 'Kowalski',

pracownik nazywa się jak kierownik tematu;

(2) dla typu *D*

data urodzenia pracownika przypada na 1968-03-01,

data urodzenia pracownika jest taka sama jak data urodzenia kierownika tematu;

(3) dla typu *N*

numer tematu jest 5,

numer tematu odpowiada numerowi pracownika;

(4) dla typu *L*

kobieta jest pracownikiem,

kobieta nie jest pracownikiem.

Podsumowując, podstawowym celem modułu analizy językowej jest sprawdzenie poprawności sformułowanego pytania. Analizując pytanie korzysta się z opisu bazy, gramatyki języka pseudonaturalnego, reguł składni oraz specjalnych słowników. Następnie przetworzone pytanie (pytanie w języku pośrednim) jest kierowane do modułu wyszukiwania.

7. Opis algorytmów

Przedstawiony w poprzednim punkcie opis analizy językowej pytania zapisanego w języku pseudonaturalnym pozwala na sformułowanie dwóch algorytmów.

Pierwszy z algorytmów, wykorzystywany jest podczas pierwszego etapu analizy językowej, natomiast algorytm drugi, wykorzystywany jest podczas drugiego etapu analizy językowej.

Algorytm 1. Na podstawie pytania w języku pseudonaturalnym generuje *listę słów*

(1) Odczytaj pytanie.

(2) Pobierz znak.

(3) Sprawdź, czy znak należy do alfabetu języka pseudonaturalnego. Jeśli tak, to przejdź do punktu (4), w przeciwnym przypadku wstrzymaj dalszą analizę i wyślij komunikat o błędzie.

(4) Sprawdź, czy są zachowane reguły składni. Jeśli tak, to przejdź do punktu (5), w przeciwnym przypadku wstrzymaj dalszą analizę i wyślij komunikat o błędzie.

(5) Dla danego znaku wybierz jedną z następujących akcji:

(5.1) Znak jest litera.
Umieść znak na *liście znaków* i przejdź do punktu (2).

(5.2) Znak jest cyfra.
Umieść znak na *liście znaków* i przejdź do punktu (2).

(5.3) Znak jest przecinkiem.
Umieść znak na *liście znaków* i przejdź do punktu (6).

(5.4) Znak jest średnikiem.
Umieść znak na *liście znaków* i przejdź do punktu (6).

(5.5) Znak jest myślnikiem.
Umieść znak na *liście znaków* i przejdź do punktu (6).

(5.6) Znak jest apostrofem otwierającym.
Umieść znak na *liście znaków* i przejdź do punktu (2).

(5.7) Znak jest apostrofem zamykającym.
Umieść znak na *liście znaków* i przejdź do punktu (2).

(5.8) Znak jest kropka.
Umieść znak na *liście znaków*, dołącz listę znaków do listy słów i zakończ analizę pytania.

(5.9) Znak jest znakiem odstępu i wchodzi w skład napisu ujętego w apostrofy.
Umieść znak na *liście znaków* i przejdź do punktu (2).

(5.10) Znak jest znakiem odstępu i nie wchodzi w skład napisu ujętego w apostrofy.
Umieść słowo na *liście znaków* i przejdź do punktu (6).

(6) Dołącz listę znaków do listy słów i przejdź do punktu (2).

Algorytm 2. Na podstawie *listy słów* generuje plik z pytaniem w języku pośrednim

(1) Analiza frazy polecenia.

(1.1) Odczytaj słowa z listy słów i utwórz z nich hasło opisujące polecenie. Sprawdź, czy to hasło znajduje się w słowniku poleceń. Jeśli tak, to przejdź do punktu (2), w przeciwnym przypadku wstrzymaj dalszą analizę i wyślij komunikat o błędzie.

(2) Analiza frazy atrybutów poszukiwanych.

(2.1) Odczytaj słowa z listy słów i utwórz z nich hasło opisujące atrybut. Sprawdź, czy to hasło znajduje się w słowniku atrybutów. Jeśli tak, to przejdź do punktu (2.2), w przeciwnym przypadku wstrzymaj dalszą analizę i wyślij komunikat o błędzie.

(2.2) Sprawdź, czy atrybut jest dostępny w aktualnie wykorzystywanej bazie danych. Jeśli tak, to przejdź do punktu (2.2), w przeciwnym przypadku wstrzymaj dalszą analizę i wyślij komunikat o błędzie.

(2.3) Wpisz nazwę atrybutu poszukiwanego do pliku wynikowego. Sprawdź, czy koniec frazy atrybutów poszukiwanych. Jeśli tak, to przejdź do punktu (3), w przeciwnym przypadku przejdź do punktu (2.1).

(3) Sprawdź, czy koniec pytania. Jeśli tak, to zakończ analizę, w przeciwnym przypadku przejdź do punktu (4).

(4) Analiza frazy warunków.

(4.1) Odczytaj słowa z listy słów i utwórz z nich hasło opisujące pierwszy atrybut. Sprawdź, czy to hasło znajduje się w słowniku atrybutów. Jeśli tak, to przejdź do punktu (4.2), w przeciwnym przypadku wstrzymaj dalszą analizę i wyślij komunikat o błędzie.

(4.2) Sprawdź, czy atrybut jest dostępny w aktualnie wykorzystywanej bazie danych. Jeśli tak, to przejdź do punktu (4.3), w przeciwnym przypadku wstrzymaj dalszą analizę i wyślij komunikat o błędzie.

(4.3) Sprawdź, czy atrybut jest typu notatnikowego. Jeśli nie, to przejdź do punktu (4.4), w przeciwnym przypadku wstrzymaj dalszą analizę i wyślij komunikat o błędzie.

(4.4) Odczytaj słowa z listy słów i utwórz z nich hasło opisujące związek. Sprawdź, czy to hasło znajduje się w słowniku związków. Jeśli tak, to odczytaj operator relacyjny i przejdź to punktu (4.5), w przeciwnym przypadku wstrzymaj dalszą analizę i wyślij komunikat o błędzie.

(4.5) Odczytaj słowa z listy słów i utwórz z nich hasło opisujące drugi atrybut. Jeśli hasło to jest datą, liczbą lub napisem ujętym w apostrofy, to przejdź do punktu (4.6), w przeciwnym przypadku przejdź do punktu (4.7).

(4.6) Sprawdź, czy atrybut pierwszy i jego wartość (hasło opisujące atrybut drugi) są zgodnych typów. Jeśli tak, to umieść w pliku wynikowym nazwę atrybutu, operator

relacyjny i wartość atrybutu i przejdź do punktu (5), w przeciwnym przypadku wstrzymaj dalszą analizę i wyślij komunikat o błędzie.

(4.7) Sprawdź, czy atrybut pierwszy jest typu logicznego. Jeśli nie, to przejdź do punktu (4.8), w przeciwnym przypadku umieść nazwę atrybutu pierwszego i symbol relacyjny w pliku wynikowym i przejdź do punktu (5).

(4.8) Odczytaj słowa z listy słów i utwórz z nich hasło opisujące drugi atrybut. Sprawdź, czy to hasło znajduje się w słowniku atrybutów. Jeśli tak, to przejdź to punktu (4.9), w przeciwnym przypadku wstrzymaj dalszą analizę i wyślij komunikat o błędzie.

(4.9) Sprawdź, czy atrybut jest dostępny w aktualnie wykorzystywanej bazie danych. Jeśli tak, to przejdź do punktu (4.10), w przeciwnym przypadku wstrzymaj dalszą analizę i wyślij komunikat o błędzie.

(4.10) Sprawdź, czy atrybut pierwszy i atrybut drugi są tych samych typów. Jeśli tak, to umieść w pliku wynikowym nazwę atrybutu pierwszego, operator relacyjny i nazwę atrybutu drugiego i przejdź do punktu (5), w przeciwnym przypadku wstrzymaj dalszą analizę i wyślij komunikat o błędzie.

(5) Sprawdź, czy koniec pytania. Jeśli tak, to zakończ analizę, w przeciwnym przypadku przejdź do punktu (4.1).

8. Ograniczenia języka pseudonaturalnego

Przykład 6

Rozpatrzmy następujące zadanie wyszukania danych w bazie *Pracownicy Naukowi*:

(1) Zapis zadania w języku naturalnym.

Wyszukaj nazwiska pracowników, którzy pracują w tym samym instytucie co Kowalski.

(2) Zapis zadania w języku zapytań systemu UNIVERS.

Szukane dane: a.nazwisko

Warunki: b.nazwisko = 'KOWALSKI'

b.instytut = a.instytut

Według wcześniej przyjętych założeń pytanie z powyższego przykładu nie może być poprawnie zapisane w języku pseudonaturalnym, ponieważ występują w nim atrybuty należące do dwóch różnych kopii relacji uniwersalnej. Zapisując to zadanie w języku zapytań systemu UNIVERS, użytkownik musi dokładnie określić, do jakiej kopii (tzn. *a* lub *b*) należy każdy atrybut. Wykorzystuje w tym celu zmienne krotkowe. Natomiast system analizujący pytanie w języku pseudonaturalnym nie potrafi określić, z jakiej kopii pochodzi atrybut.

Dlatego przyjęto, że w obecnej wersji system interpretuje pytania korzystające tylko z jednej kopii relacji uniwersalnej.

9. Zakończenie

Przedstawiony system ma charakter eksperymentalny. Opracowany i wykonany moduł programowy [1] stanowi preprocesor dla wcześniej opracowanego systemu wyszukiwania wykorzystującego model relacji uniwersalnej. Pozwala on użytkownikowi między innymi na:

- definiowanie fraz języka naturalnego odnoszących się do wykorzystywanych baz danych,
- formułowanie pytań zbudowanych na podstawie zdefiniowanego zbioru fraz.

Dzięki zastosowaniu języka pseudonaturalnego porozumiewanie się z systemem wyszukiwania informacji uległo, z punktu widzenia użytkownika, znacznemu uproszczeniu. W ramach dalszych prac nad przedstawionym systemem rozpatrywana jest możliwość uściślania pytań w trakcie dialogu z użytkownikiem, a także możliwość automatycznej generacji fraz przy wykorzystaniu opisu bazy za pomocą diagramów obiektów, atrybutów i związków [11, 12].

LITERATURA

[1] Boruta A.: Dla systemu relacji uniwersalnej opracować i wykonać moduł komunikacji z użytkownikiem umożliwiający wykorzystanie w procesie formułowania pytań elementów języka naturalnego. Praca dyplomowa. Instytut Informatyki Politechniki Śl., Gliwice 1992.

[2] Bolc L., Cichy M., Różańska L.: Przetwarzanie języka naturalnego, WNT, Warszawa 1982.

[3] Grzywocz J.: Własności języka zapytań a opis bazy danych. Zeszyty Naukowe Politechniki Śląskiej, seria Informatyka, z. 25, Gliwice 1994.

[4] Delobel C., Adiba M.: Relacyjne bazy danych, WNT, Warszawa 1989.

[5] Korth H.F., Kuper G.M., Feigenbaum J., van Gelder A., Ullman J.D.: System/U: A Database System Based on the Universal Relation Assumption. ACM TODS, 9, 3, 1984.

[6] Kozielski S., Piec J., Grzywocz J.: System wyszukiwania danych oparty na modelu relacji uniwersalnej, Archiwum Informatyki Teoretycznej i Stosowanej, 1993 (w druku).

[7] Maier D., Ullman J.D.: Maximal Objects and the Semantics of Universal Relation Databases. ACM TODS, 8, 1, 1983.

[8] Mykowiecka A.: Podstawy przetwarzania języka naturalnego. Metody generowania tekstów. RM, Warszawa 1992.

[9] Piec J.: System wyszukiwania danych z plików dBASE. Zeszyty Naukowe Politechniki Śląskiej, seria Informatyka, z. 16, Gliwice 1993.

[10] Sagiv Y.: A Characterization of Globally Consistent Databases and Their Correct Access Paths. ACM TODS, 8, 2, 1983.

[11] Ullman J.D.: Principles of Database Systems. Computer Science Press, Rockville, 1983. Polskie wydanie: Systemy baz danych. WNT, Warszawa 1988.

[12] Ullman J.D.: Database and Knowledge-Base Systems. Computer Science Press, Rockville 1989.

[13] Węgrzyn S.: Podstawy informatyki. PWN, Warszawa 1982.

Recenzent: Prof. dr inż. Alicja Wakulicz-Deja

Wpłynęło do Redakcji 19.02.1994 r.

Abstract

Query system using set of defined natural language phrases is presented in the paper. The typical phrases, related to the utilized databased, may be defined during system installation or during query session even. Applying universal relation model makes possible interpretation of queries formulated with these phrases. Queries are translated in two steps: at first to the form required by universal relation model and then to the notation based on the relational algebra. The program may be afterwards executed by database processors realizing algebraic operations. In the data retrieving system presented in the paper dBASE and FoxBASE preprocessors are applied together with the original program of fast files joining.