ZESZYTY
NAUKOWE
POLITECHNIKI
ŚLĄSKIEJ

STANISŁAW KOZIELSKI

PROJEKTOWANIE STRUKTURY LOGICZNEJ RELACYJNYCH BAZ DANYCH METODĄ SYNTEZY PRZY UWZGLĘDNIENIU ZADAŃ WYSZUKIWANIA DANYCH

P. 4193/88

INFORMATYKA

Z. 11
GLIWICE
1988

POLITECHNIKA ŚLĄSKA

ZESZYTY NAUKOWE

Nr 936

STANISŁAW KOZIELSKI

P. 4193/88

PROJEKTOWANIE STRUKTURY LOGICZNEJ RELACYJNYCH BAZ DANYCH METODĄ SYNTEZY PRZY UWZGLĘDNIENIU ZADAŃ WYSZUKIWANIA DANYCH

GLIWICE 1988

OPINIODAWCY
Prof. dr hab. inż. Ryszard Jakubowski
Prof. dr hab. Zdzisław Pawlak

KOLEGIUM REDAKCYJNE
REDAKTOR NACZELNY — Prof. dr hab. inż. Wiesław Gabzdyl
REDAKTOR DZIAŁU — Dr inż. Anna Skrzywan-Kosek
SEKRETARZ REDAKCJI — Mgr Elżbieta Stinzing
CZŁONKOWIE KOLEGIUM — Prof. dr hab. inż. Adolf Maciejny
— Prof. dr inż. Stanisław Malzacher
— Prof. dr hab. inż. Bronisław Skinderowicz

OPRACOWANIE REDAKCYJNE
Mgr Roma Łoś

Wydano za zgodą
Rektora Politechniki Śląskiej

PL ISSN 0208-7286

Dział Wydawnictw Politechniki Śląskiej
ul. Kujawska 3, 44-100 Gliwice

Nakł. 160+85 Ark. wyd. 10,9 Ark. druk. 9,125 Papier offset. kl.III 70x100, 70 g
Oddano do druku 3.11.87 Podpis. do druku 2.03.88 Druk ukończ. w kwietniu 1988
Zam. 932/87 L-24 Cena zł 218.—

Skład, fotokopie, druk i oprawę
wykonano w Zakładzie Graficznym Politechniki Śląskiej w Gliwicach

P 173/88

SPIS TREŚCI

Str.

1. WSTĘP 9

2. POJĘCIA PODSTAWOWE DOTYCZĄCE RELACYJNYCH BAZ DANYCH I PROJEKTOWANIA ICH STRUKTURY LOGICZNEJ 12

2.1. Relacyjne bazy danych 12
2.2. Normalizacja - proces projektowania struktury logicznej relacyjnych baz danych 18
2.3. Prezentacja pojęć podstawowych dotyczących procesu syntezy 21

3. ALGORYTM SYNTEZY SCHEMATU RELACYJNEJ BAZY DANYCH PRZY ZMODYFIKOWANEJ METODZIE WYZNACZANIA POKRYCIA ZBIORU ZALEŻNOŚCI FUNKCJONALNYCH 30

3.1. Wpływ obecności w pokryciu bijekcji między kluczami równoważnymi na własności procesu syntezy 30
3.2. Modyfikacja koncepcji pokrycia zbioru zależności funkcjonalnych - pokrycie dla potrzeb syntezy 37
3.2.1. Klucze pozorne 37
3.2.2. Możliwości wyznaczania dodatkowych kluczy relacji wśród kluczy pozornych 40
3.2.3. Organizacja procesu usuwania zależności redundancyjnych przy wymuszeniu obecności bijekcji w pokryciu 41
3.3. Algorytm syntezy przy zmienionej koncepcji pokrycia 46
3.4. Uwagi na temat realizacji zmodyfikowanego algorytmu syntezy i jego złożoności obliczeniowej 52
3.4.1. Programowa realizacja algorytmu 3.3 52
3.4.2. Uwagi na temat złożoności obliczeniowej nowego algorytmu syntezy 52

4. PROCES SYNTEZY PRZY UWZGLĘDNIENIU ZADAŃ WYSZUKIWANIA DANYCH 56

4.1. Możliwości uzyskania wielu rozwiązań w procesie syntezy. Kryterium wyboru rozwiązania najlepszego 56
4.2. Alternatywnie redundancyjne zależności funkcjonalne i ich własności 61
4.3. Alternatywnie redundancyjne zbiory zależności funkcjonalnych i ich własności 68
4.4. Wyznaczanie zależności i zbiorów alternatywnie redundancyjnych. Synteza wszystkich postaci schematów relacji 78
4.4.1. Wyznaczanie alternatywnie redundancyjnych zbiorów zależności wiążących klucze z różnych klas równoważności 78

Str.

4.4.2. Wykrywanie zbiorów alternatywnie redundancyjnych metodą powtarzania operacji wyznaczania pokrycia 82

4.4.3. Ocena złożoności obliczeniowej algorytmu wyznaczającego zbiory alternatywnie redundacyjne 88

4.5. Ocena schematu relacyjnej bazy danych w procesie automatycznego rozwiązywania zadań wyszukiwania danych 90

4.5.1. Wykorzystanie hipergrafów do opisu schematu relacyjnej bazy danych 91

4.5.2. Hipergrafy acykliczne 93

4.5.3. Wyznaczanie relacji wiążących atrybuty zbioru X dla γ-acyklicznego hipergrafu relacyjnej bazy danych 97

4.5.4. Algorytm wyznaczania zbioru relacji wiążących atrybuty zbioru X 102

4.5.5. Końcowa postać algorytmu syntezy uwzględniającego zadania wyszukiwania danych 107

5. ZAKOŃCZENIE 109

LITERATURA 111

DODATEK. PRZYKŁAD PROJEKTOWANIA STRUKTURY LOGICZNEJ RELACYJNEJ BAZY DANYCH PRZY WYKORZYSTANIU PROGRAMU SYNTEZY 115

STRESZCZENIA 141

СОДЕРЖАНИЕ

Стр.

1. ВСТУПЛЕНИЕ 9

2. ОСНОВНЫЕ ПОНЯТИЯ КАСАЮЩИЕСЯ РЕЛЯЦИОННЫХ БАЗ ДАННЫХ И ПРОЕКТИРОВАНИЯ ИХ ЛОГИЧЕСКОЙ СТРУКТУРЫ 12

2.1. Реляционные базы данных 12

2.2. Стандартизация - процесс проектирования логической структуры реляционных баз данных 18

2.3. Основные понятия касающиеся процесса синтеза 21

3. АЛГОРИТМ СИНТЕЗА СХЕМЫ РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ МОДИФИЦИРОВАННОГО МЕТОДА ОПРЕДЕЛЕНИЯ ПОКРЫТИЯ МНОЖЕСТВА ФУНКЦИОНАЛЬНЫХ ЗАВИСИМОСТЕЙ 30

3.1. Влияние наличия биекции в покрытии между эквивалентными ключами на свойства процесса синтеза 30

3.2. Модификация концепции покрытия множества функциональных зависимостей - покрытие для потребностей синтеза 37

3.2.1. Мнимые ключи 37

3.2.2. Возможности определения дополнительных ключей зависимостей среди мнимых ключей 40

3.2.3. Организация процесса удаления редундантных зависимостей в случае вынужденного присутствия биекции в покрытии 41

3.3. Алгоритм синтеза в изменённой концепции покрытия 46

3.4. Замечания по реализации модифицированного алгоритма синтеза и его вычислительная сложность 52

3.4.1. Программная реализация алгоритма 3.3 52

3.4.2. Замечания по вычислительной сложности нового алгоритма синтеза 52

4. ПРОЦЕСС СИНТЕЗА С УЧЁТОМ ЗАДАЧ ПОИСКА ДАННЫХ 56

4.1. Возможности получения многих решений в процессе синтеза. Критерий выбора наилучшего решения 56

4.2. Альтернативно редундантные функциональные зависимости и их свойства 61

4.3. Альтернативно редундантные множества функциональных зависимостей и их свойства 68

4.4. Определение зависимостей и альтернативно редундантных множеств. Синтез всех видов реляционных схем 78

4.4.1. Определение альтернативно редундантных множеств зависимостей связывающих ключи с разными классами эквивалентности 78

Стр.

4.4.2. Обнаружение альтернативно редундантных множеств методом повторения операций определения покрытий 82

4.4.3. Оценка вычислительной сложности алгоритма определяющего альтернативно редундантные множества 88

4.5. Оценка схемы реляционной базы данных в процессе автоматического решения задач поиска данных 90

4.5.1. Использование гиперграфов для описания схемы реляционной базы данных 91

4.5.2. Ациклические гиперграфы 93

4.5.3. Определение зависимостей связывающих атрибуты множества X для γ - ациклического гиперграфа реляционной базы данных 97

4.5.4. Алгоритм определения множества зависимостей, связывающих атрибуты множества X 102

4.5.5. Окончательный вид алгоритма синтеза, учитывающий задачи поиска данных 107

5. ОКОНЧАНИЕ 109

ЛИТЕРАТУРА 111

ДОПОЛНЕНИЕ. ПРИМЕР ПРОЕКТИРОВАНИЯ ЛОГИЧЕСКОЙ СТРУКТУРЫ РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ С ИСПОЛЬЗОВАНИЕМ ПРОГРАММЫ СИНТЕЗА 115

РЕЗЮМЕ 141

CONTENTS

Page

1. INTRODUCTION 9

2. BASIC CONCEPTS RELATED TO RELATIONAL DATABASES AND THEIR LOGICAL STRUCTURE DESIGN 12

2.1. Relational databases 12

2.2. Normalization - relational database logical structure designing process 18

2.3. Basic concepts related to synthesis process 21

3. RELATIONAL DATABASE SCHEMA SYNTHESIS ALGORITHM WITH MODIFICATED METHOD OF FORMING COVER OF FUNCTIONAL DEPENDENCIES' SET 30

3.1. Influence of presence of bijections in cover for properties of synthesis process 30

3.2. Modification of concept of cover functional dependencies'set - definition of cover for synthesis purposes 37

3.2.1. Apparent keys 37

3.2.2. Possibility of finding of additional keys among apparent keys 40

3.2.3. Organization of redundant dependencies elimination process with forced presence of bijections in cover 41

3.3. Synthesis algorithm with modificated conception of cover .. 46

3.4. Remarks on realization of modificated synthesis algorithm and its computational complexity 52

3.4.1. Program - realization of algorithm 3.3 52

3.4.2. Remarks on computational complexity of new synthesis algorithm 52

4. SYNTHESIS ALGORITHM TAKING QUERY PROCESSING INTO CONSIDERATION 56

4.1. Possibility of obtaining many results in synthesis process. Criterion of choise of the best result 56

4.2. Alternatively redundant functional dependencies and their properties 61

4.3. Alternatively redundant sets of fuctional dependencies and their properties 68

4.4. Determination of alternatively redundant dependencies and sets. Synthesis of all forms of relational schemata 78

4.4.1. Obtaining of alternatively redundant sets of dependencies binding keys from different classes of equivalence 78

Page

4.4.2. Finding of alternatively redundant sets by repeating of cover obtaining operation 82

4.4.3. Estimation of computational complexity of algorithm determining alternatively redundant sets 88

4.5. Estimation of relational database schema in automatic process of query processing 90

4.5.1. Use of hypergraphs to relational database schema description 91

4.5.2. Acyclic hypergraphs 93

4.5.3. Determination of relations binding attributes of set X for γ-acyclic relational database hypergraph 97

4.5.4. Algorithm for finding a set of relations binding attributes of set X 102

4.5.5. Final form of synthesis algorithm taking query processing into consideration 107

5. CONCLUSIONS 109

BIBLIOGRAPHY 111

APPENDIX. EXAMPLE OF DESIGNING OF RELATIONAL DATABASE LOGICAL STRUCTURE WITH THE USE OF THE SYNTHESIS PROGRAM 115

SUMMARY 141

1. WSTĘP

Projektowanie struktury logicznej (np. [22, 5, 74, 76, 81, 28, 88]) stanowi jeden z głównych kierunków badań w dziedzinie relacyjnych baz danych. Punktem wyjścia w tych badaniach jest zbiór atrybutów, których wartości mają być przechowywane w bazie danych oraz zbiór powiązań między atrybutami. Celem natomiast jest zaprojektowanie takiej struktury bazy danych, aby nie wystąpiły w niej dane nadmiarowe (redundancyjne), a operacje wstawiania i usuwania danych były realizowane w sposób możliwie prosty [23, 24, 25, 26, 70, 86]. Własności te zapewnia osiągnięcie tak zwanej trzeciej lub czwartej postaci normalnej [23, 24, 35] w strukturze logicznej, inaczej schematach poszczególnych relacji.

Znane są dwie zasadnicze metody projektowania struktury logicznej. Pierwsza, przedstawiona przez E.F. Codda razem z założeniami modelu relacyjnego, polega na dekompozycji wstępnie przyjętej struktury relacji. Druga zapewnia osiągnięcie trzeciej postaci normalnej na drodze syntezy schematów relacji z zadanych zależności funkcjonalnych bez przyjmowania początkowej struktury relacji. Zalążki tej metody określili Delobel i Casey [29,30] oraz Wang i Wedekind [87, 18], natomiast pełne sformułowanie zasad syntezy nastąpiło w pracach Bernsteina [19, 13], a później Maiera [69, 70].

Postęp prac nad obiema metodami projektowania związany był ściśle z rozwojem badań nad opisem związków między atrybutami. Podstawowym modelem tych związków jest - wprowadzona przez Codda - zależność funkcjonalna. Początkowo w pracach nad normalizacją wykorzystywano analogię między zależnościami funkcjonalnymi a funkcjami logicznymi [36, 80]. Ważniejszym kierunkiem było jednak sformułowanie w pracach Delobela i Caseya [29], a przede wszystkim Armstronga [6] reguł wnioskowania, pozwalających eliminować zależności redundancyjne i wybierać pokrycie zbioru zależności funkcjonalnych, które ogólnie można określić jako reprezentację zbioru zależności o własnościach pożądanych w procesie syntezy. Spełnienie w projektowanej bazie wszystkich zadanych zależności funkcjonalnych (reprezentujących w modelu relacyjnym powiązania między relacjami) stanowi przy tym podstawowe założenie procesu syntezy. Pierwszy kompletny i poprawny algorytm syntezy sformułowany został przez Bernsteina [19].

Pogłębienie badań nad różnymi postaciami pokrycia i ich wykorzystaniem w procesie syntezy przyniosły prace Maiera [69, 70].

Dalsze prace (np. [27]) nad powiązaniami między atrybutami doprowadziły do sformułowania przez Fagina [35] zależności wielowartościowych. Badania te wzbogaciły metodę dekompozycji umożliwiając uzyskanie tak zwanej czwartej postaci normalnej schematu relacji.

Prace Rissanena [78] oraz Aho, Beeriego i Ullmana [1] przyniosły definicję i analizę własności zależności połączeniowej. Badania te pozwoliły na sformułowanie ważnej własności tzw. połączenia bez strat, traktowanego następnie jako warunek reprezentowania w różnych strukturach baz danych tej samej informacji [14].

Lawina dalszych prac i definicji nowych zależności między artybutami (np. [39, 40, 42]) związana była głównie z kolejnymi próbami uogólnienia poprzednich definicji i ich własności, a także próbami stworzenia pełnej aksjomatyki w tej dziedzinie.

W ostatnich latach proces projektowania przysłonięty został koncepcją relacji uniwersalnej [43, 55, 50, 68, 72, 73, 85], pozwalającę użytkownikowi widzieć całą bazę danych jako jednę ogólnę relację. Pierwszoplanowy w tym modelu problem automatycznego rozwiązywania zadań wyszukiwania danych postawił przy tym dodatkowe kryteria przed zadaniem projektowania struktury relacji, związane z cyklicznością bądź acyklicznością hipergrafów reprezentujących schematy relacyjnych baz danych [15, 16, 21, 41, 44, 47, 90].

Główne cele niniejszej pracy obejmują z jednej strony rozszerzenie badań teoretycznych nad własnościami zbioru zależności funkcjonalnych oraz jego pokrycia z punktu widzenia procesu syntezy, drugim zaś zamierzeniem było stworzenie algorytmu pozwalającego uwzględnić w procesie syntezy dodatkowe kryterium jego oceny.

Punktem wyjścia tych badań jest analiza własności nieredundancyjnego pokrycia zbioru zależności funkcjonalnych, tworzonego w procesie syntezy. W algorytmach syntezy Bernsteina [19] oraz Maiera [69, 70] po utworzeniu pokrycia istnieje konieczność dodatkowej eliminacji zależności redundancyjnych, ujawniających się po scaleniu w jednym schemacie relacji zależności o lewych stronach będących tak zwanymi kluczami równoważnymi. W pracach autora [58, 59, 60, 61] wykazano, że powyższa eliminacja jest zbędna, jeśli do nieredundancyjnego pokrycia należę wszystkie zależności wiążące klucze równoważne. Wymuszenie obecności takich zależności w pokryciu pozwala więc uprościć proces syntezy.

W rozdziale 3 niniejszej pracy rozwinięto powyższe tezy i przedstawiono algorytm syntezy bazujący na zmodyfikowanej koncepcji wyznaczania pokrycia.

W niniejszej pracy drugim etapem badań własności pokrycia jest analiza możliwości uzyskania - dla tego samego zbioru zależności funkcjonalnych - różnych postaci pokrycia oraz różnych schematów relacji. Własności takich różnych rozwiązań są identyczne przy uwzględnieniu dotychczasowych kryteriów oceny procesu syntezy. Proste przykłady (rozdział 4.1) pokazują jednak, ze dla takich schematów różna może być złożoność zadań wyszukiwania danych realizowanych w zaprojektowanych bazach.

Tak więc złożoność typowych - w danej bazie - zadań wyszukiwania danych, sformułowanych już na etapie projektowania struktury logicznej (schematu) bazy danych, przyjęta została jako dodatkowe kryterium oceny efektów procesu syntezy.

Wprowadzenie takiego kryterium wymagało w pierwszej kolejności opracowania metody tworzenia wszystkich możliwych postaci pokrycia.

Przedstawione w rozdziałach 4.2 i 4.3 badania zależności usuwanych z pokrycia pozwoliły zdefiniować zależności (i zbiory zależności), które zamiennie mogę występować w pokryciu. Zbadano szereg cech tych zależności, nazywanych alternatywnie redundancyjnymi, wykazując między innymi, że własność alternatywnej redundancyjności jest relacją równoważności. Pozwala to wyodrębnić klasy abstrakcji takich zależności, co istotnie upraszcza problem ich wyznaczania, a tym samym tworzenia - dla tych samych danych - różnych postaci schematu relacyjnej bazy danych.

Ocena tworzonych schematów według wspomnianego wyżej kryterium jest złożonym zagadnieniem algorytmicznym. Przyjęto, że podstawę tej oceny jest liczba połączeń relacji wykorzystywanych w procesie rozwiązania każdego z zadań wyszukiwania danych. Wyznaczanie tej liczby jest praktycznie możliwe jedynie przy automatycznym rozwiązywaniu zadań. Metody takie analizowane są w literaturze [16, 41, 72, 73, 85] dla wspomnianej uprzednio relacji uniwersalnej. Ich prezentecja wymagała wprowadzenia do pracy pojęcia hipergrafów jako powszechnie stosowanych narzędzi w tej dziedzinie. Opracowany w rozdziale 4.5 algorytm umożliwia wyznaczanie liczby łączonych relacji dla zbiorów atrybutów reprezentujących zadania wyszukiwania danych. Ograniczono się przy tym do obszernej klasy baz danych reprezentowanych hipergrafami acyklicznymi.

Przedstawiony w pracy algorytm syntezy pozwala na utworzenie wszystkich możliwych - dla zadanego zbioru zależności funkcjonalnych - schematów relacyjnej bazy danych oraz wybór schematu najlepszego poprzez minimalizację liczby relacji potrzebnych do rozwiązania przyjętych w danej bazie zadań wyszukiwania danych.

W tym miejscu chciałbym serdecznie podziękować Profesorowi dr inż. Stefanowi WĘGRZYNOWI za umożliwienie mi przeprowadzenia części niniejszych badań w ramach problemu węzłowego 06.4, którego był koordynatorem.

Dziękuję również prof. dr hab. inż. Janowi Węglarzowi, prof. dr hab. Zdzisławowi Pawlakowi oraz doc. dr hab. inż. Zbigniewowi Czechowi za uwagi, które pozwoliły mi ulepszyć tekst pracy.

2. POJĘCIA PODSTAWOWE DOTYCZĄCE RELACYJNYCH BAZ DANYCH I PROJEKTOWANIA ICH STRUKTURY LOGICZNEJ

2.1. RELACYJNE BAZY DANYCH

W literaturze współistnieje kilka definicji relacji [23,25,26,70,83,86]*):

(1) Punktem wyjścia pierwszej jest schemat relacji R definiowany jako skończony zbiór nazw atrybutów (lub krótko atrybutów): $R = \{A_1,\dots,A_k\}$. Z każdym atrybutem $A_i \in R$ związany jest niepusty zbiór wartości D_i, nazywany dziedziną atrybutu A_i, oznaczany też przez dom (A_i). Niech $\mathbb{D} = D_1 \cup D_2 \cup \dots \cup D_k$. Relacją r o schemacie R nazywać będziemy skończony zbiór odwzorowań $\{t_1,\dots,t_p\}$ z R do $\mathbb{D}$, przy czym wartość odpowiadająca atrybutowi A_i w każdym odwzorowaniu t musi należeć do D_i, tzn. $\forall\, t \in r \quad t[A_i] \in D_i$, $1 \leq i \leq k$. Odwzorowania te nazywane są krotkami.

(2) Przyjmując poprzednie określenia schematu, atrybutów i ich dziedzin, relację r o schemacie $R = \{A_1,\dots,A_k\}$ nazywamy dowolny podzbiór iloczynu kartezjańskiego dziedzin $D_1 \times D_2 \times \dots \times D_k$.

(3) W trzecim ujęciu relacja prezentowana jest zwykle jako tablica, której zbiór kolumn odpowiada zbiorowi atrybutów tworzącemu schemat, a zbiór wierszy odpowiada zbiorowi krotek. Wartości przechowywane w danej kolumnie muszą przy tym pochodzić z dziedziny atrybutu przypisanego tej kolumnie.

Schematem relacyjnej bazy danych nazywać będziemy zbiór schematów relacji $\mathbb{R} = \{R_1,\dots,R_n\}$ utworzony nad określonym zbiorem atrybutów U, przy czym $\bigcup_{i=1}^{n} R_i = U$.

Relacyjną bazą danych r o schemacie $\mathbb{R}$ jest zbiór relacji $r(\mathbb{R}) = \{r_1(R_1), r_2(R_2),\dots,r_n(R_n)\}$ o schematach z $\mathbb{R}$.

Kluczem w relacji $r(R)$ jest taki podzbiór $K \subseteq R$, 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 niektórych przypadkach używane jest pojęcie nadklucza [19, 13] (ang. superkey), określanego jako taki podzbiór $K_s \subseteq R$, który zawiera klucz.

*) Przedstawiane tu określenia relacji wprowadzone zostały w literaturze na użytek baz danych. Drugie z tych określeń koresponduje z klasyczną definicją relacji stosowaną w matematyce [77, 62].

W tym miejscu wyjaśnimy kilka reguł zapisu stosowanych w literaturze i przyjętych też w niniejszej pracy:

- duże początkowe litery alfabetu: A,B,C... oznaczają pojedyncze atrybuty (atrybuty proste),
- duże końcowe litery alfabetu: ... X,Y,Z oznaczają atrybuty złożone (podzbiory atrybutów prostych),
- zapis XY oznacza sumę zbiorów $X \cup Y$,
- obok zapisu schematu relacji w postaci $R = \{A_1,\dots,A_k\}$ będzie też stosowany zapis $R(A_1,\dots,A_k)$ lub $R = A_1,\dots,A_k$.

Dla relacyjnych baz danych definiuje się szereg operatorów tworzących tzw. algebrę relacji [26, 70, 86].

W niniejszej pracy wykorzystywać będziemy następujące spośród nich:

Operator selekcji. Operator ten tworzy - dla danej relacji r o schemacie R - podzbiór $r' \subseteq r$ takich krotek, dla których spełniony jest określony warunek selekcji, np. wartość atrybutu $A \in R$ wynosi a, gdzie $a \in$ dom (A). Operację selekcji zapiszemy w tym przypadku następująco:

$$\sigma_{A=a}(r) = r'(R) = \{t \in r \mid t[A] = a\}. \tag{2.1.1}$$

Operator rzutowania. Niech X będzie podzbiorem schematu R pewnej relacji r. Posługując się tablicową prezentacją relacji, rzutem r na X nazwiemy relację r' powstałą z r przez odrzucenie kolumn odpowiadających atrybutom ze zbioru $R-X$ i usunięcie powtarzających się krotek. Operację tę zapiszemy następująco:

$$\pi_X(r) = r'(X) = \{t[X] \mid t \in r\}. \tag{2.1.2}$$

Inną formą oznaczenia rzutowania jest zapis

$$r[X] \stackrel{df}{=} \pi_X(r). \tag{2.1.3}$$

Operacja łączenia (naturalnego). Niech dane będą dwie relacje $r(R)$ i $s(S)$. Oznaczmy: $T = R \cup S$. Połączeniem relacji r i s jest relacja $q(T)$, której każda krotka t została utworzona z 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 $t_r = t[R]$, $t_s = t[S]$.

$$r(R) \bowtie s(S) = q(T) = \{t \mid \exists t_r \in r \wedge \exists\, t_s \in s \wedge t[R] = t_r \wedge t[S] = t_s \wedge t_r[R \cap S] = t_s[R \cap S]\}. \tag{2.1.4}$$

Podstawowe zależności między atrybutami

Bazą procesu projektowania struktury logicznej (schematu) relacyjnych baz danych jest analiza powiązań między atrybutami. W punkcie tym przedstawimy najczęściej wykorzystywane zależności między danymi [70, 86, 19, 35].

Zależności funkcjonalne. Powszechnie stosowana jest następująca definicja: Zależność funkcjonalna $f: X \longrightarrow Y$ (co czytamy: Y jest zależne funkcjonalnie od X) jest spełniona w relacji r(R), gdzie X i Y są podzbiorami R, jeśli dla każdych dwóch krotek t_1 i t_2 należących do r, dla których $t_1[X] = t_2[Y]$, zachodzi również $t_1[Y] = t_2[Y]$.

Alternatywna definicja mówi, że przy poprzednich założeniach zależność funkcjonalna jest spełniona w r, jeśli dla dowolnej wartości x atrybutu X (gdzie $x \in dom(X)$) relacja utworzona w wyniku operacji

$$\pi_Y(\sigma_{X=x}(r)) \qquad (2.1.5)$$

zawiera nie więcej niż jedną krotkę.

Mankamentem powyższych definicji jest powiązanie istnienia zależności funkcjonalnej z istnieniem określonej relacji. Tymczasem w rozważanym w niniejszej pracy procesie projektowania schematu relacyjnej bazy danych metodę syntezy, danymi wejściowymi są jedynie zbiór atrybutów i zbiór powiązań między nimi, bez określania jakiejkolwiek wstępnej struktury relacji.

Dlatego też celowe jest przytoczenie dodatkowej definicji zależności funkcjonalnej wprowadzonej przez P.A. Bernsteina [19]: Niech A i B będą atrybutami o dziedzinach dom (A) i dom (B). Zależnością funkcjonalną nazywana jest "funkcja zmienna w czasie" $f: dom(A) \longrightarrow dom(B)$ interpretowana jako zbiór par uporządkowanych $\{(a,b) \mid a \in dom(A) \wedge b \in dom(B)\}$ takich, że w każdej chwili czasu dla danej wartości $a \in dom(A)$ istnieje co najwyżej jedna wartość $b \in dom(B)$ występująca w parze z b w powyższym zbiorze.

Zmienność w czasie funkcji f odpowiada zmienności w czasie danych przechowywanych w relacji i tłumaczy przyczynę użycia dla f nazwy "zależność funkcjonalna".

Dla wygody zapisu opuszczane są oznaczenia dziedzin i stosuje się notację $f: A \longrightarrow B$. Dla atrybutów złożonych $X = \{A_1, A_2, \ldots, A_i\}$, $Y = \{B_1, \ldots, B_j\}$ stosowane jest następujące uogólnienie pojęcia zależności funkcjonalnej: $f: X \longrightarrow Y$ oznacza $f: dom(A_1) \times dom(A_2) \times \ldots \times dom(A_i) \longrightarrow dom(B_1) \times \ldots \times dom(B_j)$. Używa się też zapisu $f: A_1, A_2, \ldots, A_i \longrightarrow B_1, \ldots, B_j$, opuszczając zwykle przecinki. □

Jeśli dla danej zależności funkcjonalnej $X \longrightarrow Y$ istnieje podzbiór $X' \subset X$ taki, że spełniona jest również zależność $X' \longrightarrow Y$, to mówimy, że Y jest częściowo zależne od X. Atrybuty z podzbioru X-X' nazywane są wtedy atrybutami obcymi.

Załóżmy, że dla danego schematu relacji R spełniona jest zależność funkcjonalna $X \longrightarrow A$, gdzie X jest podzbiorem R, A zaś atrybutem należącym do R. Zależność ta jest zależnością tranzytywną, jeśli w R istnieje taki podzbiór Y, że zachodzi

$$X \longrightarrow Y,\ Y \not\longrightarrow X,\ Y \longrightarrow A \text{ oraz } A \notin XY. \qquad (2.1.6)$$

Przykład 2.1.1. Rozpatrzmy następujący zbiór atrybutów: {student, kierunek, wydział, przedmiot, ocena}. Przyjmiemy następujące założenia:

- każda wartość atrybutu "student" jednoznacznie identyfikuje studenta (może to być np. numer indeksu),
- każda wartość atrybutu "przedmiot" jednoznacznie identyfikuje przedmiot kończący się egzaminem,
- student może studiować tylko na jednym kierunku (rezygnacja z tego założenia jest możliwa, ale wymaga wprowadzenia nowego atrybutu),
- dany kierunek prowadzony jest tylko na jednym wydziale,
- student może otrzymać z danego przedmiotu tylko jedną ocenę końcową.

Przy takich założeniach spełnione będą między innymi następujące zależności funkcjonalne:

student ⟶ kierunek,
kierunek ⟶ wydział,
student, przedmiot ⟶ ocena.

Zauważmy, że założenia te pozwalają też wskazać inne zależności funkcjonalne, na przykład:

- student, kierunek, przedmiot ⟶ ocena,
  ale zależność ta jest zależnością częściową wobec istnienia zależności
  student, przedmiot ⟶ ocena;
- student ⟶ wydział,
  przy czym zależność ta jest zależnością tranzytywną wobec istnienia zależności
  student ⟶ kierunek oraz kierunek ⟶ wydział.

Na koniec zwróćmy jeszcze uwagę na wyraźne powiązanie w przedstawionym przykładzie procesu definiowania zależności funkcjonalnych z przyjmowaniem odpowiednich założeń dotyczących jednoznaczności powiązań między określonymi atrybutami. W takim ujęciu zależności funkcjonalne utożsamiane są z predykatami [43] formułowanymi dla dziedziny reprezentowanej w określonej bazie danych.

Inne zależności między atrybutami omówione zostaną skrótowo ze względu na sporadyczne tylko ich wykorzystanie w niniejszej pracy.

Zależność wielowartościowa: Niech R oznacza schemat relacji, a X,Y,Z - rozłączne podzbiory R, dla których zachodzi $Z = R - (X \cup Y)$. Zależność wielowartościowa $X \longrightarrow\!\!\!\!\rightarrow Y$ spełniona jest w relacji r(R), jeśli wystę-

pienie w tej relacji każdych dwóch krotek t_1 i t_2 takich, że $t_1[X] = t_2[X]$ pociąga za sobą istnienie w r trzeciej krotki t_3 o własnościach:

$$t_3[X] = t_1[X], \quad t_3[Y] = t_1[Y], \quad t_3[Z] = t_2[Z].$$

Symetria tej definicji względem t_1 i t_2 implikuje występienie w r także czwartej krotki t_4, dla której zachodzi:

$$t_4[X] = t_1[X], \quad t_4[Y] = t_2[Y], \quad t_4[Z] = t_1[Z].$$

Według alternatywnej definicji zależność $X \twoheadrightarrow Y$ jest spełniona w r(R), jeśli dla każdej pary wartości (x,z) należącej do r (tzn. takiej, że istnieje w r krotka t, dla której zachodzi $t[X] = x$, $t[Z] = z$) spełniona jest zależność:

$$\pi_Y(\sigma_{X=x,Z=z}(r)) = \pi_Y(\sigma_{X=x}(r)). \qquad (2.1.7)$$

Oznacza to, że zbiór wartości atrybutu Y związany w relacji r z daną wartością atrybutu X nie zależy od wartości atrybutu Z. □

Przykład 2.1.2. Rozpatrzmy następujący zbiór atrybutów: R = {student, przedmiot, ocena, konkurencja, rekord}.

Oprócz założeń z przykładu 2.1.1 przyjmiemy ponadto, że atrybut "rekord" oznacza rekordowy wynik uzyskany przez danego studenta w danej konkurencji sportowej. Załóżmy, że w przykładowej relacji r o schemacie R znalazły się następujące dane (przyjmiemy, że nazwiska studentów identyfikują ich jednoznacznie):

r (student, przedmiot, ocena, konkurencja, rekord)

Kowalski	fizyka	4	sprint	12,4
Kowalski	fizyka	4	kula	18,75
Kowalski	matematyka	3	sprint	12,4
Kowalski	matematyka	3	kula	18,75
Nowak	fizyka	3	oszczep	64

W powyższej relacji spełnione są następujące zależności wielowartościowe

student $\twoheadrightarrow$ przedmiot, ocena
student $\twoheadrightarrow$ konkurencja, rekord.

Gdyby jednak w prezentowanej relacji występowała dodatkowo jedna krotka o postaci

(Nowak, matematyka, 5, sprint, 13,3),

to powyższe zależności nie byłyby spełnione. □

Zależność połączeniowa: Relacja r(R) spełnia zależność połączeniową $\bowtie(R_1, R_2, \ldots, R_p)$, gdzie $R_1 \cup R_2 \cup \ldots \cup R_p = R$, wtedy i tylko wtedy, jeśli zachodzi

$$r = \pi_{R_1}(r) \bowtie \pi_{R_2}(r) \bowtie \ldots \bowtie \pi_{R_p}(r). \quad \square \qquad (2.1.8)$$

Przykład 2.1.3. Relacja r z przykładu 2.1.2 spełnia następującą zależność połączeniową:

$\bowtie(\{\text{student, przedmiot, ocena}\}, \{\text{student, konkurencja, rekord}\})$.

Przy uwzględnieniu prezentowanej w tym przykładzie dodatkowej krotki zależność ta nie będzie jednak spełniona. W takim przypadku prawdziwa będzie natomiast zależność:

$\bowtie(\{\text{student, przedmiot, ocena}\}, \{\text{student, konkurencja, rekord}\}, \{\text{student, przedmiot, konkurencja}\})$. □

Spełnienie zależności połączeniowej dla danej relacji utożsamić można ze zdefiniowaną w następnym rozdziale własnością dekompozycji bez strat.

Komentując zasugerowany w powyższym przykładzie związek między zależnościami wielowartościowymi a zależnością połączeniową można wspomnieć, że zachodzi równoważność między tymi dwoma typami zależności w bazach danych reprezentowanych hipergrafami acyklicznymi [16].

Najważniejsze postacie normalne schematów relacji

Schemat relacji R jest w pierwszej postaci normalnej (1NF), jeśli wartości należące do dziedziny atrybutu A - dla każdego A z R - są wartościami atomowymi. □

Oznacza to, że wartości należące do dziedziny atrybutu A nie są na przykład listami lub zbiorami innych wartości.

Przykład 2.1.4. Rozpatrzmy przykładową relację dla zbioru atrybutów z przykładu 2.1.1 (przyjmiemy, że nazwiska studentów identyfikują ich jednoznacznie):

r_1 (student, kierunek, wydział, przedmiot, ocena)

Kowalski	Informatyka	AEI	fizyka	3
Kowalski	Informatyka	AEI	matematyka	5
Kowalski	Informatyka	AEI	programowanie	4
Nowak	Informatyka	AEI	fizyka	3,5
Zieliński	Elektronika	AEI	matematyka	4

Schemat relacji r_1 jest w pierwszej postaci normalnej. □

Definicje kolejnych postaci normalnych schematów relacji R będziemy prezentować w kontekście zbioru zależności funkcjonalnych F wiążących atrybuty ze zbioru R.

Schemat relacji R jest w drugiej postaci normalnej (2NF) z uwagi na zbiór zależności funkcjonalnych F, jeśli jest w 1NF i każdy niekluczowy atrybut jest w pełni zależny funkcjonalnie od każdego klucza schematu R. □

Przykład 2.1.5. Schemat relacji r_1 z przykładu 2.1.4 nie jest w 2NF, ponieważ każdy z atrybutów "kierunek" i "wydział" jest częściowo zależny od atrybutu złożonego $\{$student, przedmiot$\}$, który jest kluczem relacji r_1. Natomiast każdy ze schematów $R_2 = \{$student, kierunek, wydział$\}$, $R_3 = \{$student, przedmiot, ocena$\}$ jest w 2NF. □

Schemat relacji R jest w trzeciej postaci normalnej (3NF) z uwagi na zbiór zależności funkcjonalnych F, jeśli jest w 1NF i żaden niekluczowy atrybut w R nie jest tranzytywnie zależny od żadnego klucza w R. □

Przykład 2.1.6. Schemat relacji R_2 z przykładu 2.1.5 nie jest w trzeciej postaci normalnej, bowiem atrybut "wydział" jest tranzytywnie (poprzez atrybut "kierunek") zależny od klucza "student" tego schematu. Schematy: $R_3 = \{$student, przedmiot, ocena$\}$, $R_4 = \{$student, kierunek$\}$ oraz $R_5 = \{$kierunek, wydział$\}$ są w 3NF. □

Schemat relacji R jest w czwartej postaci normalnej (4NF) z uwagi na zbiór zależności funkcjonalnych i wielowartościowych F, jeśli dla każdej zależności $X \twoheadrightarrow Y \in F$ albo ta zależność jest zależnością trywialną (co odpowiada warunkowi $XY = R$), albo też X jest nadkluczem schematu R. □

Przykład 2.1.7. Schemat relacji R z przykładu 2.1.2 nie jest w 4NF, natomiast następujące schematy:

$R_1 = \{$student, przedmiot, ocena$\}$,
$R_2 = \{$student, konkurencja, rekord$\}$

są schematami w czwarej postaci normalnej. □

W literaturze znane są również inne postacie normalne relacji np. BCNF [70, 86], PJNF [70], JNF [10], LTKNF [65], DK/NF [38], 5NF [1, 51]. Postaci tych nie będziemy jednak definiowali, ponieważ nie są one związane z celami procesu syntezy.

2.2. NORMALIZACJA - PROCES PROJEKTOWANIA STRUKTURY LOGICZNEJ RELACYJNYCH BAZ DANYCH

Cele projektowania schematu (struktury logicznej) relacyjnej bazy danych były formułowane i uściślane w wielu pracach [23, 26, 70, 86, 14].

Ogólnie można określić dwa główne cele:

(1) Minimalizacja redundancji, a więc utworzenie tylko takich schematów relacji, i o takiej postaci, aby żaden ze schematów nie był zbędnym, a przy tym relacje o tych schematach nie zawierały informacji zbędnej. W przypadkach koniecznych redundancja informacji winna być minimalna

(2) Separacja odrębnych związków między atrybutami w oddzielnych relacjach dla uniezależnienia procesu wprowadzania, usuwania bądź aktualizacji danych w jednych relacjach od obecności danych w innych relacjach.

Cele powyżeze osiągane są [23, 26, 14] przez uzyskanie odpowiednich postaci normalnych schematów relacji. Stąd też proces projektowania nazywany jest normalizacją.

Główną zasadą, na której winien się opierać proces normalizacji, jest zasada reprezentacji [14] mówięca, że w utworzonym schemacie bazy danych winna być reprezentowana ta sama informacja co w danych wejściowych do procesu normalizacji. Pojęcie "reprezentowania informacji" jest interpretowane różnie dla dwóch zasadniczych metod normalizacji:

Metoda dekompozycji. Punktem wyjścia w tej metodzie jest założenie istnienia relacji uniwersalnej [14, 9, 43, 84], to znaczy przyjęcie, że wszystkie relacje $r_1(R_1), r_2(R_2), \ldots, r_n(R_n)$ tworzone w danej bazie danych są równe wynikowi rzutowania pewnej (hipotetycznej) relacji $u(U)$ na schematy $R_1, \ldots, R_n$, tzn.

$$r_i(R_i) = \Pi_{R_i}(u),$$

gdzie U jest zadanym zbiorem atrybutów, przy czym $\bigcup_{i=1}^{n} R_i = U$. Zbiór schematów $\rho = \{R_1, R_2, \ldots, R_n\}$ nazywany jest wtedy dekompozycją schematu U.

Dane wejściowe dla tej metody określa zbiór zależności funkcjonalnych i (ewentualnie) wielowartościowych spełnianych w przyjętej relacji uniwersalnej.

Wspomniana zasada reprezentacji interpretowana jest w tej metodzie jako zgodność danych zawartych w relacji uniwersalnej u z danymi zawartymi w relacjach $r_1, \ldots, r_n$, wyrażona równością

$$u = r_1 \bowtie r_2 \bowtie \ldots \bowtie r_n \qquad (2.2.1)$$

Własność ta nazywana jest własnością połączenia bez strat (ang. lossless join) [1, 14, 79, 86].

Proces dekompozycji opiera się na dwóch twierdzeniach [70, 57]:

TWIERDZENIE 2.1. Załóżmy, że w relacji r o schemacie R i kluczu $K \subset R$, atrybut $A \in R$ jest tranzytywnie zależny od klucza K, tzn. istnieje taki podzbiór $X \subset R$, że

$$K \rightarrow X,\ X \not\rightarrow K,\ X \rightarrow A,\ A \notin XK.$$

Wtedy dekompozycja $\rho = \{XA, R-A\}$ ma własność połączenia bez strat.

D o w ó d. Patrz pozycja [70].

TWIERDZENIE 2.2. Niech r będzie relacją o schemacie R i niech X, Y i Z będą podzbiorami R o własności $Z = R - (X \cup Y)$. Relacja r spełnia zależność wielowartościową $X \twoheadrightarrow Y$ wtedy i tylko wtedy, jeśli dekompozycja schematu R w postaci $\rho = \{XY, XZ\}$ ma własność połączenia bez strat.

D o w ó d. Patrz prace [70, 86].

Sformułowanie zasad dekompozycji jest czasem [14] opierane tylko na twierdzeniu 2.2 poprzez wykorzystanie faktu, że każda zależność funkcjonalna jest szczególnym przypadkiem zależności wielowartościowej.

Usunięcie na drodze dekompozycji wszystkich zależności tranzytywnych zapewnia uzyskanie wszystkich schematów w trzeciej postaci normalnej. Wykorzystanie zależności wielowartościowych [7, 31, 12, 63, 46, 49] zapewnia uzyskanie czwartej postaci normalnej.

Wskazać można [14, 37, 70] szereg wad metody dekompozycji. Do najważniejszych należą:

- Możliwość uzyskania większej liczby schematów relacji, aniżeli jest to konieczne.
  Wiąże się to z ogólnym faktem, że metoda dekompozycji prowadzi tylko "w dół", dzieląc otrzymane w poprzednim kroku relacje.
- Zależność wyników końcowych od kolejności dekompozycji.
  Istnieje w ten sposób możliwość uzyskania wyników formalnie poprawnych, ale praktycznie nieużytecznych.
- Trudności w praktycznym wskazaniu istniejących zależności wielowartościowych.
- Duża złożoność obliczeniowa algorytmu dekompozycji.

Niektóre zagadnienia rozwiązywane w procesie dekompozycji, jak np. skazanie atrybutów niekluczowych [13, 70] lub sprawdzanie, czy relacja jest w 4NF [70], należą do problemów NP-zupełnych.

Metoda syntezy. Dane wejściowe do procesu syntezy tworzy zbiór F zależności funkcjonalnych wiążących wszystkie atrybuty z zadanego zbioru U. Celem syntezy jest utworzenie schematu relacyjnej bazy danych $\mathbb{R} = \{R_1, \ldots, R_n\}$ takiego, że

(1) każdy schemat $R_i \in \mathbb{R}$ jest w trzeciej postaci normalnej,

(2) liczba schematów w $\mathbb{R}$ jest minimalna,

(3) w relacyjnej bazie danych o schemacie $\mathbb{R}$ spełnione będą wszystkie zależności funkcjonalne ze zbioru F.

Trzeci z powyższych warunków jest utożsamiany w metodzie syntezy ze sformułowaną uprzednio zasadą reprezentacji.

Koncepcja procesu syntezy wywodzi się z następującej interpretacji definicji trzeciej postaci normalnej schematu relacji:

Schemat relacji w 3NF jest zbiorem atrybutów, z których część tworzy klucz, pozostałe zaś są funkcjonalnie (i nietranzytywnie) zależne od klucza.

Najprostszy algorytm syntezy mógłby więc być sformułowany następująco [19]:

(a) podziel zbiór zależności funkcjonalnych F na grupy zależności o identycznych lewych stronach,

(b) utwórz schemat relacji dla każdej grupy, obejmujący wszystkie atrybuty występujące w zależnościach tej grupy. Lewa strona tych zależności jest kluczem relacji.

Algorytm taki spełnia jednak tylko pierwszą z postulowanych własności procesu syntezy; utworzony schemat zapewnia bowiem spełnienie wszystkich zależności funkcjonalnych ze zbioru F. Natomiast w powstałych schematach mogą istnieć zależności tranzytywne, a ponadto liczba schematów relacji może nie być minimalna.

Dowolny zbiór zależności funkcjonalnych F nie może więc być bezpośrednio podstawą syntezy relacji. Przedstawienie sposobu modyfikacji tego zbioru wymaga jednak szerszej analizy własności zależności funkcjonalnych. Analizę tę, wraz z prezentacją klasycznego algorytmu syntezy Bernsteina [19] zamieszczono w następnym punkcie tego rozdziału.

Uprzedzając szczegółową dyskusję algorytmu syntezy podkreślimy dwie różnice między metodą dekompozycji a metodą syntezy:

(1) Inna jest dla obu metod zasada reprezentacji [14] informacji wejściowych w uzyskiwanym wyniku. Dla metody dekompozycji jest nią własność połączenia bez strat dotycząca danych zawartych we wstępnie rozważanej relacji. Natomiast dla metody syntezy takiej relacji się nie rozpatruje, a zasada reprezentacji dotyczy spełnienia wszystkich zadanych zależności funkcjonalnych w utworzonym schemacie bazy danych. Proces dekompozycji może w pewnych przypadkach [13, 70] nie zachowywać wszystkich zależności.

(2) W metodzie dekompozycji możliwe jest osiągnięcie schematów relacji w 4NF, natomiast w metodzie syntezy - w 3NF. Należy jednak zauważyć, że dla osiągnięcia 4NF muszą być wykorzystane zależności wielowartościowe, których określenie nastręcza wiele problemów [70].

2.3. PREZENTACJA POJĘĆ PODSTAWOWYCH DOTYCZĄCYCH PROCESU SYNTEZY

Na wstępie tego pod rozdziału przedstawimy zasady wyboru - dla zadanego zbioru zależności funkcjonalnych - takiej reprezentacji tego zbioru, która posiada własności pożądane w procesie syntezy.

Jeżeli dany jest pewien zbiór zależności funkcjonalnych spełnianych w pewnej bazie danych, to możliwe jest istnienie również innych zależności funkcjonalnych, które jawnie nie zostały określone, ale w rozpatrywanej bazie danych będą spełnione (np. [54, 71]).

Reguły wnioskowania pozwalające szukać takich zależności przedstawione zostały w pracy [29]. Armstrong wykazał w pracy [6] kompletność i poprawność tych reguł, ujmowanych w literaturze w następującej postaci:

Aksjomaty Armstronga (W, X, Y, Z - podzbiory atrybutów)

A1. (Zwrotność): $(Y \subseteq X) \models X \rightarrow Y$,

A2. (Wzrost): $(X \rightarrow Y) \wedge (Z \subseteq W) \models XW \rightarrow YZ$,

A3. (Pseudoprzechodniość): $(X \rightarrow Y) \wedge (YW \rightarrow Z) \models XW \rightarrow Z$.

Przedstawione trzy zasady tworzę zbiór kompletny, tzn. wszelkie inne reguły wnioskowania można z nich wyprowadzić. Wygodne jest jednak wprowadzenie dwóch dodatkowych zasad:

A4. (Łączność): $(X \longrightarrow Y_1) \wedge (X \longrightarrow Y_2) \models X \longrightarrow Y_1Y_2$,

A5. (Dekompozycja): $(X \longrightarrow Y) \wedge (Y' \subset Y) \models X \longrightarrow Y'$.

Z zasad A4 i A5 wynika, że zależność funkcjonalna o postaci $X \longrightarrow A_1A_2 \ldots A_n$ jest równoważna zbiorowi zależności funkcjonalnych $X \longrightarrow A_1$, $X \longrightarrow A_2, \ldots, X \longrightarrow A_n$. Dlatego wiele rozważań nad własnościami zależności funkcjonalnych można sprowadzić do przypadku, kiedy prawa strona zależności jest pojedynczym atrybutem.

Proces tworzenia nowych zależności na podstawie danego zbioru zależności formalizowany jest następująco:
Dla danego zbioru F zależności funkcjonalnych wyprowadzeniem (albo wywodem) zależności f z F jest ciąg $(f_1, f_2, \ldots, f_n)$ taki, że $f_n = f$, zaś każda zależność f_i albo należy do F, albo też jest wynikiem zastosowania jednej z zasad A1, A2, A3 do jednej lub dwóch zależności ze zbioru $f_1, f_2, \ldots, f_{i-1}$.

Zbiór wszystkich zależności funkcjonalnych, które są wyprowadzalne ze zbioru F, nazywamy domknięciem F i oznaczamy przez F^+. Własność wyprowadzalności danej zależności f ze zbioru F możemy wobec tego utożsamiać z przynależnością f do domknięcia F^+. Pokryciem danego zbioru F zależności funkcjonalnych nazywamy taki zbiór G zależności funkcjonalnych, który posiada takie samo domknięcie jak zbiór dany, to znaczy $G^+ = F^+$. Zależność funkcjonalna $f \in F$ jest redundancyjna w F, jeśli zależność ta jest wyprowadzalna ze zbioru F po usunięciu z niego f, to znaczy $f \in (F - \{f\})^+$. Własność tę możemy interpretować jako spełnienie równości

$$(F - \{f\})^+ = F^+. \qquad (2.3.1)$$

Pokrycie, które nie zawiera żadnej zależności redundancyjnej, nazywane jest pokryciem nieredundancyjnym.

Z przedstawionym wyżej pojęciem wyprowadzenia wiąże się następująca własność, która będzie wykorzystywana w dalszej części pracy:

Lemat 2.1. [19, 13] Załóżmy, że zależność funkcjonalna $g: X \longrightarrow Y$ należy do zbioru zależności funkcjonalnych F. Jeśli zależność $h: V \longrightarrow W$ należy do domknięcia F^+, a zależność g użyta została do wywodu h z F, to wtedy do domknięcia F^+ należy też zależność $V \longrightarrow X$.

D o w ó d przedstawiony został w pracach [19, 13]. □

Wracając do postawionego w punkcie 2.2 zagadnienia modyfikacji zbioru zależności funkcjonalnych dla uzyskania postaci pożądanej w procesie syntezy przypomnijmy, że zależnościami niepożądanymi z punktu widzenia normalizacji są zależności tranzytywne oraz częściowe. Zauważmy, że z definicji zależności tranzytywnej wynika, iż jest ona zależnością redundancyjną. A więc pokrycie nieredundancyjne nie będzie zawierało zależności tranzytywnych. Dla wyeliminowania zależności częściowych należy natomiast usunąć obce atrybuty z lewych stron takich zależności. Wymaga to sprawdzenia dla każdej zależności $f : X \longrightarrow A$ z danego zbioru F i każdego atrybutu $B \in X$, czy zależność $(X - \{B\}) \longrightarrow A$ należy do domknięcia F^+. Jeśli tak, to atrybut B jest usuwany z X.

Tak więc w obu przypadkach, tzn. usuwania zależności redundancyjnych oraz obcych atrybutów, kryterium rozstrzygającym o usuwaniu jest przynależność badanej zależności $f: X \longrightarrow A$ do domknięcia zadanego zbioru zależności funkcjonalnych F.

Dwie wersje algorytmu badania przynależności do domknięcia przedstawione zostały w pracy [13]. Główna idea realizowana w tych algorytmach polega na poszukiwaniu zbioru atrybutów funkcjonalnie zależnych od atrybutu X tworzącego lewą stronę badanej zależności f. Zbiór taki nazywany jest domknięciem X i oznaczany przez X^+. Pytanie czy $f \in F^+$ sprowadza się wtedy do pytania, czy $A \in X^+$. Domknięcie X^+ możemy wyznaczyć, korzystając z zasad Armstronga, w następujący sposób (przez D oznaczymy tworzony zbiór atrybutów zależnych funkcjonalnie od X):

Algorytm 2.1

Dane wejściowe: zbiór F zależności funkcjonalnych oraz atrybut X (być może złożony).

Dane wyjściowe: domknięcie X^+ atrybutu X.

(1) Przyjmij początkowo, że $D := X$ (na mocy A1).

(2) Jeśli w zbiorze F istnieje taka zależność $V \longrightarrow Z$, że $V \subseteq D$ oraz $Z \not\subseteq D$, to włącz Z do D, to znaczy $D := D \cup Z$.

(3) Powtarzaj krok (2) tak długo, aż żadna z prawych stron zależności tworzących F nie będzie mogła być dołączona do D. Wtedy $X^+ = D$.

Poprawność kroku (2) można wykazać tworząc następujący wywód:

- Po kroku (1) : $(D := X) \models X \longrightarrow D$ (na mocy A1)
- $(V \subseteq D) \models D \longrightarrow V$ (na mocy A1),
- $(X \longrightarrow D) \wedge (D \longrightarrow V) \models X \longrightarrow V$ (na mocy A3).
- $(X \longrightarrow V) \wedge (V \longrightarrow Z) \models X \longrightarrow Z$ (na mocy A3),
- $(X \longrightarrow D) \wedge (X \longrightarrow Z) \models X \longrightarrow DZ$ (na mocy A4).

Pesymistyczna złożoność obliczeniowa [2] algorytmu zbudowanego w wyżej opisany sposób wynosi (według [13]): $O(a|F|)$, gdzie a jest liczbą różnych atrybutów w zbiorze F, natomiast $|F|$ oznacza sumaryczną liczbę wystąpień atrybutów w zapisie zbioru F.

W pracy [13] przedstawiono też ulepszoną wersję tego algorytmu o złożoności obliczeniowej $O(|F|)$.

Algorytm syntezy schematu relacyjnej bazy danych

Bazując na algorytmie badania przynależności do domknięcia można we wskazany uprzednio sposób usunąć obce atrybuty z lewych stron zależności zbioru F oraz wyznaczyć nieredundancyjne pokrycie tego zbioru (usuwając zależności redundancyjne). Operacje te stanowią dwa pierwsze etapy algorytmu syntezy schematu relacyjnej bazy danych.

Wstępną wersję algorytmu syntezy można więc przedstawić następująco:

Algorytm 2.2

Dane wejściowe: zbiór F zależności funkcjonalnych.

Dane wyjściowe: schemat relacyjnej bazy danych

$$R = \{R_1, R_2, \ldots, R_q\}.$$

(1) Usuń obce atrybuty z danego zbioru F zależności funkcjonalnych.

(2) Utwórz nieredundancyjne pokrycie H zbioru F.

(3) Podziel pokrycie H na grupy zależności H_i o identycznych lewych stronach.

(4) Dla każdej grupy H_i utwórz schemat relacji R_i z atrybutów występujących w danej grupie.

Algorytm o takiej postaci rozpatrywano w pracach [29, 87, 19]. W pracy [19] wykazano, że każdy schemat relacji utworzony według powyższego algorytmu jest w trzeciej postaci normalnej. Oznacza to, że każdy atrybut niekluczowy jest nietranzytywnie zależny od klucza relacji. Ustalonym w procesie syntezy kluczem jest w każdej relacji lewa strona zależności funkcjonalnych wspólna dla tej grupy, dla której utworzono schemat rozważanej relacji. Klucz taki nazywany jest też kluczem syntetycznym lub wyznaczonym [19, 13] (w relacji mogą istnieć ponadto inne klucze [45, 32, 33]).

Wadą powyższego algorytmu jest możliwość uzyskania w wyniku syntezy liczby relacji większej od minimalnej. Nastąpi to wtedy, gdy klucze dwóch różnych utworzonych schematów relacji (lewe strony dwóch różnych grup zależności) będą wzajemnie od siebie zależne funkcjonalnie, np. $X_i \longrightarrow X_j$, $X_j \longrightarrow X_i$. Klucze takie nazywane są <u>kluczami równoważnymi</u> i razem z zależnymi od nich atrybutami powinny się znaleźć w jednej relacji. Wobec tego jeden z dwóch schematów relacji o kluczach X_i, X_j jest zbędny, czyli liczba utworzonych schematów relacji nie będzie minimalna.

Para zależności funkcjonalnych wiążących wspomniane klucze $X_i \longrightarrow X_j$, $X_j \longrightarrow X_i$ nazywana jest <u>bijekcją</u>.

Zauważmy, że bijekcja jest relacją równoważności [77] w zbiorze potęgowym $\mathcal{A}$ zbioru strybutów $\mathbb{A}$ ($\mathcal{A}$ jest zbiorem wszystkich podzbiorów zbioru $\mathbb{A}$). Oznaczmy bijekcję między złożonymi atrybutami $X, Y \in \mathcal{A}$ przez $X \longleftrightarrow Y$. Bijekcja spełnia następujące warunki:

(1) $X \longleftrightarrow X$ dla każdego $X \in \mathcal{A}$,

(2) $X \longleftrightarrow Y \models Y \longleftrightarrow X$ dla każdego $X, Y \in \mathcal{A}$,

(3) $(X \longleftrightarrow Y) \wedge (Y \longleftrightarrow Z) \models X \longleftrightarrow Z$ dla każdego $X, Y, Z \in \mathcal{A}$.

Zwrotność (1) wynika wprost z aksjomatu A1 Armstronga. Symetria (2) wynika z definicji bijekcji. Przechodność (3) wynika z aksjomatu A3 Armstronga.

Wobec tego z każdym podzbiorem $X \in \mathcal{A}$ związana jest w $\mathcal{A}$ <u>klasa równoważności</u> (klasa abstrakcji) $\|X\|$ dla bijekcji, definiowana jako zbiór takich $Y \in \mathcal{A}$, które związane są bijekcją z X.

W dalszych rozważaniach interesujące będzie wykorzystanie zasady abstrakcji stwierdzającej, że każda relacja równoważności ustala podział zbioru $\mathcal{A}$ na klasy równoważności.

Dla celów syntezy relacji zawężimy zbiór $\mathcal{A}$ do podzbioru X, którego elementami są syntetyczne klucze relacji, tzn. podzbiory atrybutów tworzących lewe strony zależności funkcjonalnych należących do pokrycia.

Uwzględnienie bijekcji w zbiorze X prowadzi - zgodnie z zasadą abstrakcji - do podziału tego zbioru na klasy równoważności względem bijekcji, to znaczy na zbiory kluczy równoważnych.

Połączenie grup zależności o lewych stronach będących kluczami równoważnymi zapewni utworzenie wspólnych schematów relacji dla tych grup.

Proces scalania może jednak doprowadzić do ujawnienia się w takich schematach zależności tranzytywnych. Przypadek ten zbadał P.A. Bernstein w pracy [19]: Załóżmy, że pewien schemat relacji R powstał przez scalenie dwóch grup zależności H_i oraz H_j, których klucze (lewe strony) X_i oraz X_j były kluczami równoważnymi. Zauważmy, że zależności $X_i \longrightarrow X_j$ oraz $X_j \longrightarrow X_i$ nie musiały należeć do F, a tylko do F^+. Bernstein wskazał, że jeśli pewna zależność $X_i \longrightarrow A \in H_i$ potrzebna była do wywodu zależności $X_i \longrightarrow X_j$, to po scaleniu grup zależności H_i oraz H_j, zależność $X_i \longrightarrow A$ może się okazać tranzytywną, a więc redundancyjną.

<u>Przykład 2.3.1</u>. Rozpatrzmy następujący zbiór zależności funkcjonalnych:

$$F = \{XY \longrightarrow AD,\ CD \longrightarrow EY,\ XA \longrightarrow B,\ YB \longrightarrow C,\ C \longrightarrow A,\ CE \longrightarrow H,$$
$$DH \longrightarrow X,\ X \longrightarrow E\}.$$

Nieredundancyjne pokrycie H jest identyczne ze zbiorem F. Lewe strony dwóch pierwszych zależności XY i CD są kluczami równoważnymi. Scalenie tych zależności prowadzi do następujących schematów relacji: R_1 = = <u>XY CD</u> AE, R_2 = <u>XA</u> B, R_3 = <u>YB</u>C, R_4 = <u>C</u>A, R_5 = <u>CE</u>H, R_6 = <u>DH</u>X, R_7 = <u>X</u>E (klucze poszczególnych relacji zostały podkreślone). Po scaleniu w schemacie R_1 ujawniają się dwie zależności częściowe (a tym samym tranzytywne): A jest częściowo zależne od klucza CD, natomiast E - od klucza XY. Dla uzyskania schematu R_1 w 3NF należy obie te zależności usunąć, co daje wynik końcowy $R_1' = $ <u>XY CD</u>. □

Rozważania powyższa wskazują na konieczność powtórnego usuwania zależności redundancyjnych po scaleniu zależności o kluczach równoważnych.

Operacja ta uwzględniona została w następującym algorytmie przedstawionym przez Bernsteina w pracy [19]:

Algorytm 2.3

Dane wejściowe: Zbiór F zależności funkcjonalnych.

Dane wyjściowe: Schemat relacyjnej bazy danych.

(1) Usuń obce atrybuty z danego zbioru F zależności funkcjonalnych.

(2) Utwórz nieredundancyjne pokrycie H zbioru F.

(3) Podziel pokrycie H na grupy zależności H_i o identycznych lewych stronach.

(4) Utwórz zbiór J początkowo pusty. Podziel zbiór kluczy na klasy równoważności względem bijekcji. Scal każdą parę grup H_i oraz H_j o lewych stronach X i Y związanych bijekcją w H^+. Obie zależności tworzące bijekcję $X \longrightarrow Y$ i $Y \longrightarrow X$ włącz do zbioru J. Usuń z pokrycia H każdą zależność o postaci $X \longrightarrow A$, jeśli $A \in Y$ oraz zależność $Y \longrightarrow B$, jeśli $B \in X$.

(5) Usuń zależności tranzytywne znajdując najmniejszy podzbiór $H' \subseteq H$ taki, że $(H' + J)^+ = (H + J)^+$. Dodaj każdą zależność ze zbioru J do odpowiadającej jej grupy w H'.

(6) Dla każdej grupy zależności utwórz schemat relacji składający się ze wszystkich atrybutów występujących w danej grupie. Każdy zbiór atrybutów, który tworzy lewą stronę każdej z zależności funkcjonalnych w danej grupie, jest kluczem relacji.

Uwaga 2.1. W pracy [19] wykazano, że schemat $\mathbb{R}$ utworzony za pomocą algorytmu 2.3 dla zbioru zależności funkcjonalnych F posiada następujące własności:

(a) domknięcie zbioru zależności funkcjonalnych spełnionych w $\mathbb{R}$ równe jest F^+,

(b) każdy ze schematów relacji $R_i \in \mathbb{R}$ jest w trzeciej postaci normalnej,

(c) dla każdego nieredundancyjnego pokrycia zbioru F liczba schematów relacji tworzących $\mathbb{R}$ jest taka sama. Dla żadnego innego zbioru zależności funkcjonalnych G będącego pokryciem zbioru F nie można uzyskać za pomocą algorytmu 2.3 mniejszej liczby schematów relacji.

Dodatkowe uwagi o procesie syntezy

Nieco inne ujęcie procesu wyznaczania pokrycia przedstawił D. Maier w pracach [69, 70].

Celem rozważań była w pierwszym etapie minimalizacja liczby zależności tworzących nieredundancyjne pokrycie danego zbioru zależności funkcjonalnych. Jednakże wykorzystanie takiego pokrycia w procesie syntezy nie eliminuje, wskazanego przez Bernsteina, efektu ujawniania się zależności

tranzytywnych po scaleniu zależności o lewych stronach będących kluczami równoważnymi.

Proces wyznaczania kluczy równoważnych i powtórnej eliminacji zależności redundancyjnych przedstawiony został w pracy [70] jako wyznaczenie tzw. złożonych zależności funkcjonalnych (obejmujących wszystkie zależności o lewych stronach będądych kluczami równoważnymi), a następnie redukcja zbędnych atrybutów z tych zależności dla uzyskania tzw. zredukowanego, minimalnego pokrycia pierścieniowego.

Podsumowując te uwagi należy podkreślić, że zgodnie z uwagą 2.1 minimalizacja liczby zależności w pokryciu nie wpływa na liczbę schematów uzyskaną w procesie syntezy.

Innym problemem poruszanym w niektórych pracach [70, 86] jest doprowadzenie do tego, aby rozwiązanie uzyskane metodą syntezy posiadało zdefiniowaną uprzednio własność połączenia bez strat. Podejście to opiera się na koncepcji klucza uniwersalnego.

Dla schematu relacyjnej bazy danych $\mathbb{R}$ utworzonego dla zbioru atrybutów $\mathbb{A}$ na podstawie danego zbioru zależności funkcjonalnych F, kluczem uniwersalnym X nazywany jest taki podzbiór $\mathbb{A}$, dla którego wyprowadzalna jest ze zbioru F zależność $X \longrightarrow A$.

Dowodzi się [70], że jeśli któryś ze schematów tworzących $\mathbb{R}$ zawiera X, to $\mathbb{R}$ posiada własność połączenia bez strat. Wystąpienie X w $\mathbb{R}$ można wymusić (o ile potrzeba) dołączając do zadanego zbioru F zależność funkcjonalną o postaci $\mathbb{A} \longrightarrow \Theta$, gdzie Θ jest atrybutem nie występującym w zbiorze $\mathbb{A}$. Po zakończeniu syntezy atrybut Θ winien zostać usunięty. Należy ponadto zauważyć, że bardzo prawdopodobne jest usunięcie części atrybutów z lewej strony tej zależności na etapie usuwania atrybutów obcych.

Omówione postępowanie może jednak doprowadzić do utworzenia sztucznej struktury relacji, zawierającej dane bez użytecznego znaczenia. Będzie tak w przypadku istnienia - obok zależności funkcjonalnych - również pewnych zależności wielowartościowych.

Przykład 2.3.2. Rozpatrzmy zbiór atrybutów

{student, przedmiot, ocena, konkurencja, rekord}

z przykładu 2.1.2.

Atrybuty powyższe związane są następującymi zależnościami funkcjonalnymi

student, przedmiot $\longrightarrow$ ocena,

student, konkurencja $\longrightarrow$ rekord.

Proces syntezy z wymuszonym wstawianiem klucza uniwersalnego prowadzi do następujących schematów relacji

R_1 = {student, przedmiot, ocena},

R_2 = {student, konkurencja, rekord},

R_3 = {student, przedmiot, konkurencja}.

Schemat R_3 jest równocześnie kluczem uniwersalnym. Utworzenie tego sztucznego schematu zapewnia uzyskanie własności połączenia bez strat dla przedstawionych zależności funkcjonalnych. Jednakże przyjęcie oczywistych w tym przykładzie zależności wielowartościowych

student $\longrightarrow\!\!\!\!\rightarrow$ przedmiot, ocena,
student $\longrightarrow\!\!\!\!\rightarrow$ konkurencja, rekord,

prowadzi do wniosku, że własność połączenia bez strat posiada już schemat R obejmujący tylko dwa schematy, tzn. R_1 i R_2. □

Powyższy przykład sugeruje wykorzystanie do procesu syntezy zależności wielowartościowych. Próba taka [70, 91] w ogólnym przypadku nie przyniosła jednak satysfakcjonującego rozwiązania.

Pewne problemy związane z wykorzystanie zależności funkcjonalnych w procesie syntezy

Kilku autorów zwróciło w swych pracach [19, 37, 14, 48, 9] uwagę na możliwość wyprowadzenia nieprawdziwych zależności przy zastosowaniu przedstawionych uprzednio reguł wnioskowania do pewnych zależności funkcjonalnych.

Przykład 2.3.3. Rozpatrzmy dwie zależności funkcjonalne f_1: projekt $\longrightarrow$ kierownik oraz f_2: kierownik, rok $\longrightarrow$ fundusze, odpowiadające następującym stwierdzeniom:

- każdy projekt ma jednego kierownika, oraz
- każdy kierownik ma do dyspozycji w danym roku określone fundusze na realizację projektu (lub projektów).

Stosując aksjomat A3 możemy z f_1 i f_2 wyprowadzić zależność f_3: projekt, rok $\longrightarrow$ fundusze, określającą fundusze kierownika danego projektu w danym roku. Zauważmy, że możliwe jest zdefiniowanie zależności g: projekt, rok $\longrightarrow$ fundusze, syntaktycznie identycznej z f_3, a oznaczającej fundusze na realizację danego projektu w danym roku.

Zależności f_3 i g są zależnościami różnymi, jeśli dany kierownik kieruje kilkoma projektami. Zauważmy też, że powyższej sprzeczności możemy uniknąć zmieniając (uściślając) nazwy atrybutów, np.

f_2: kierownik, rok $\longrightarrow$ fundusze-kierownika.

g: projekt, rok $\longrightarrow$ fundusze-projektu. □

Możliwość wystąpienia wskazanej dwuznaczności powoduje konieczność przyjęcia założenia jednoznaczności powiązań, formułowanego na przykład następująco:

- Między dwoma atrybutami może istnieć co najwyżej jedna zależność [9].
- Jeśli w F^+ istnieją dwie zależności f: X$\longrightarrow$Y i g: X$\longrightarrow$Y, to f musi być identyczne z g [19].
- Syntaktyczna identyczność zależności musi pociągać za sobą równoważność semantyczną [14].

Podkreślmy, że jednoznaczność powiązań można zawsze uzyskać w danym zbiorze F zmieniając w razie potrzeby nazwy atrybutów.

Założenie jednoznaczności będzie stale przyjmowane w dalszym ciągu niniejszej pracy.

Na koniec warto podkreślić, że spełnienie tego założenia pozwala uniknąć niejednoznaczności - poza procesem syntezy - również w procesie wyszukiwania danych.

3. ALGORYTM SYNTEZY SCHEMATU RELACYJNEJ BAZY DANYCH PRZY ZMODYFIKOWANEJ METODZIE WYZNACZANIA POKRYCIA ZBIORU ZALEŻNOŚCI FUNKCJONALNYCH

W rozdziale niniejszym zaprezentowany zostanie algorytm syntezy schematu relacyjnej bazy danych o postaci uproszczonej w stosunku do algorytmu Bernsteina [19]. Podstawą modyfikacji jest zmiana koncepcji wyznaczania pokrycia zbioru zależności funkcjonalnych, uwzględniająca przeznaczenie pokrycia dla potrzeb syntezy.

Punktem wyjścia tych rozważań jest analiza możliwości uniknięcia występienia zależności tranzytywnych w algorytmie Bernsteina na etapie scalania zależności, których lewe strony są kluczami równoważnymi.

3.1. WPŁYW OBECNOŚCI W POKRYCIU BIJEKCJI MIĘDZY KLUCZAMI RÓWNOWAŻNYMI NA WŁASNOŚCI PROCESU SYNTEZY

W tym punkcie pracy wykażemy, że obecność w nieredundancyjnym pokryciu H bijekcji wiążących klucze równoważne danego zbioru F stanowi warunek wystarczający, który zapewnia, że utworzone w procesie syntezy relacje będą w trzeciej postaci normalnej.

Na wstępie sprecyzujemy przyjmowane oznaczenia.

Niech H oznacza jak poprzednio nieredundancyjne pokrycie danego zbioru F zależności funkcjonalnych. Załóżmy, że zależności tworzące H zapiszemy w postaci:

$$H = \{X_1 \longrightarrow Y_1,\ X_2 \longrightarrow Y_2, \ldots, X_n \longrightarrow Y_n\}. \tag{3.1.1}$$

Rozpatrując wzajemne powiązania między lewymi stronami (kluczami wyznaczonymi) tych zależności zauważmy, że, zgodnie z zasadą abstrakcji, bijekcje (jako relacja równoważności) ustala podział zbioru kluczy $\mathcal{K} = \{X_1, X_2, \ldots, X_n\}$ na klasy równoważności.

Rozważmy na początku pewną klasę równoważności zawierającą dokładnie dwa klucze

$$K_d = \{X_1^d,\ X_2^d\}. \tag{3.1.2}$$

gdzie $X_1^d, X_2^d \in \mathcal{K}$ są lewymi stronami zależności

$$X_1^d \longrightarrow Y_1^d,\ X_2^d \longrightarrow Y_2^d \tag{3.1.3}$$

należących do H.

TWIERDZENIE 3.1. Załóżmy, że dla nieredundancyjnego pokrycia H o postaci (3.1.1) wyróżniono zdefiniowaną wyżej klasę równoważności kluczy $K_d = \{X_1^d, X_2^d\}$. Jeśli do pokrycia H należę zależności $X_1^d \longrightarrow X_2^d$ oraz $X_2^d \longrightarrow X_1^d$, to schemat relacji R_d utworzony dla klasy K_d z atrybutów zależności (3.1.3) będzie w trzeciej postaci normalnej.

D o w ó d. Rozważany schemat relacji będzie miał postać $R_d = X_1^d X_2^d Y_1^d Y_2^d$. Przeprowadzimy dowód nie wprost, zakładając, że pewien atrybut $A \in Y_1^d$ okazał się tranzytywnie zależny od pewnego klucza K schematu R_d. Ogólnie rzecz biorąc, klucz K może być kluczem spoza zbioru kluczy wyznaczonych, tzn. zbioru $\mathcal{K} = \{X_1, \ldots, X_n\}$, a więc może być różny od X_1^d i X_2^d. Z definicji zależności tranzytywnej (2.1.6) wynika wtedy, że w schemacie R_d istnieje taki podzbiór $T \subset R_d$, że zachodzi

$$K \longrightarrow T,\ T \not\longrightarrow K,\ T \longrightarrow A \in F^+,\ A \notin T \cup K. \tag{3.1.4}$$

Zależność $K \longrightarrow A$ implikuje występienie tranzytywnej zależności atrybutu A również od klucza X_1^d. Jeśli bowiem X_1^d jest kluczem, to $X_1^d \longrightarrow T \in F^+$.

Ponadto $T \not\longrightarrow X_1^d$, bo cięg zależności

$$T \longrightarrow X_1^d,\ X_1^d \longrightarrow K \tag{3.1.5}$$

implikowałby, na mocy aksjomatu A3 Armstronga, zależność $T \longrightarrow K$, co byłoby sprzeczne z (3.1.4).

Wykażemy obecnie, że jeśli $X_1^d \longrightarrow A$ jest zależnością tranzytywną po scaleniu zależności (3.1.3), to musiałaby ona być redundancyjna w H.

Oznaczmy

$$R_d' = R_d - \{A\} = X_1^d \cup X_2^d \cup (Y_1^d - \{A\}) \cup Y_2^d. \tag{3.1.6}$$

Ponieważ $A \notin T$, więc $T \subseteq R_d'$, a stąd na mocy aksjomatu A1 mamy

$$R_d' \longrightarrow T. \tag{3.1.7}$$

Wykażemy teraz, że z $H - \{X_1^d \longrightarrow A\}$ wyprowadzalna jest zależność $X_1^d \longrightarrow R_d'$.

Zauważmy najpierw, że każda z zależności

$$X_1^d \longrightarrow (Y_1^d - \{A\}),\ X_1^d \longrightarrow X_2^d,\ X_2^d \longrightarrow Y_2^d$$

należy - zgodnie z założeniami - do $H - \{X_1^d \longrightarrow A\}$.
Na mocy aksjomatów Armstronga można więc ze zbioru $H - \{X_1^d \longrightarrow A\}$ wyprowadzić zależność

$$X_1^d \longrightarrow X_1^d \cup X_2^d \cup (Y_1^d - \{A\}) \cup Y_2^d. \qquad (3.1.8)$$

Na koniec wykażemy, że zależność $T \longrightarrow A$ też jest wyprowadzalna z $H - \{X_1^d \longrightarrow A\}$. Gdyby tak nie było, to do wywodu $T \longrightarrow A$ z tego zbioru byłaby potrzebna zależność $X_1^d \longrightarrow A$. Z lematu 2.1 wynikałoby wtedy istnienie w H^+ zależności $T \longrightarrow X_1^d$, co przy (3.1.5) dawałoby sprzeczność z (3.1.4).

Cięg analizowanych zależności

$$X_1^d \longrightarrow R_d',\ R_d' \longrightarrow T,\ T \longrightarrow A$$

pozwala wyprowadzić ze zbioru $H - \{X_1^d \longrightarrow A\}$ zależność $X_1^d \longrightarrow A$. Zależność ta jest więc redundancyjna w H, co jest sprzeczne z założeniem, że pokrycie H jest nieredundancyjne.

Przy zmianie poprzedniego założenia i przjęciu, że $A \in Y_2$, przedstawić można analogiczny dowód zamieniając role atrybutów X_1^d i X_2^d oraz Y_1^d i Y_2^d. □

Zauważmy, że sformułowany w powyższym twierdzeniu warunek obecności bijekcji w pokryciu jest warunkiem dostatecznym, ale nie koniecznym. Pokazuje to następujący przykład.

Przykład 3.1.1. Rozpatrzmy następujący zbiór zależności funkcjonalnych: $F = \{XY \longrightarrow AD,\ CD \longrightarrow XY,\ X \longrightarrow B,\ BY \longrightarrow C\}$. Nieredundacyjne pokrycie tego zbioru jest z nim identyczne, tzn. H = F. Lewe strony pierwszej i drugiej zależności są kluczami równoważnymi, tzn. bijekcja $XY \longleftrightarrow CD$ należy do domknięcia F^+, chociaż w pokryciu H brak zależności $XY \longrightarrow C$. Scalając pierwszą i drugą zależność otrzymamy następujące schematy relacji: $R_1 = XYCDA$, $R_2 = XB$, $R_3 = BYC$ (klucze pokreślone). Każdy z tych schematów jest w 3NF, mimo że dla R_1 nie są spełnione założenia twierdzenia 3.1. □

TWIERDZENIE 3.1 można uogólnić w przypadku, kiedy dana klasa tównoważności kluczy K_m zawiera więcej niż dwa klucze: $K_m = \{X_1^m, X_2^m, \ldots, X_r^m\}$, przy czym $\{X_1^m, X_2^m, \ldots, X_r^m\}$ są lewymi stronami zależności $g_i : X_i^m \longrightarrow Y_i^m$, $i = 1 \ldots r$ należących do nieredundancyjnego pokrycia H.

Oznaczmy lewą i prawą stronę zależności g_i następująco:

$$L(g_i) = X_i^m,\quad P(g_i) = Y_i^m. \qquad (3.1.9)$$

Warunek OB (obecności bijekcji w pokryciu). Dla klasy równoważności kluczy K_m spełniony jest warunek OB, jeśli spośród zależności funkcjonalnych obecnych w nieredundancyjnym pokryciu H utworzyć można co najmniej jeden skończony cięg niekoniecznie różnych zależności

$$c_1 = (g_{1_1}, g_{1_2}, \ldots, g_{1_s}) \qquad (3.1.10)$$

taki, że

$$L(g_{1_i}) \subseteq P(g_{1_{i-1}}),\ i = 2,3,\ldots,s. \qquad (3.1.11)$$

$$L(g_{1_1}) \subseteq P(g_{1_s}), \qquad (3.1.12)$$

$$L(g_{1_i}) \in K_m \quad i = 1,\ldots,s. \qquad (3.1.13)$$

Ponadto cięg c_1 obejmuje wszystkie klucze tworzące klasę równoważności K_m, to znaczy

$$\forall X_j^m \in K_m\ \exists\ g_{1_k} \in c_1 : L(g_{1_k}) = X_j^m. \qquad (3.1.14)$$

Warunek OB możemy zilustrować grafowym modelem powiązań między kluczami równoważnymi. Klucze będziemy wtedy interpretować jako wierzchołki grafu, a zależności funkcjonalne między kluczami jako łuki tego grafu. Rozważmy graf G_m utworzony dla obecnych w nieredundancyjnym pokryciu H zależności funkcjonalnych między kluczami równoważnymi tworzącymi daną klasę równoważności K_m. Spełnienie warunku obecności w pokryciu OB dla tej klasy oznacza istnienie w grafie G_m drogi cyklicznej przechodzącej przez wszystkie wierzchołki grafu.

TWIERDZENIE 3.2. Załóżmy, że w zbiorze lewnych stron zależności funkcjonalnych tworzących nieredundancyjne pokrycie H zbioru F wyodrębniono klasę równoważności kluczy ze względu na bijekcję w postaci $K_m = \{X_1^m, X_2^m, \ldots, X_r^m\}$. Jeśli dla tej klasy spełniony jest warunek OB obecności w pokryciu, to w schemacie relacji utworzonym ze wszytkich atrybutów zależności $X_1^m \longrightarrow Y_1^m, \ldots, X_r^m \longrightarrow Y_r^m$ nie wystąpią żadne zależności tranzytywne, a więc schemat ten będzie w trzeciej postaci normalnej.

D o w ó d. Zauważmy, że schemat relacji utworzony z zależności

$$X_1^m \longrightarrow Y_1^m, \ldots, X_r^m \longrightarrow Y_r^m \qquad (3.1.15)$$

będzie miał postać

$$R_m = X_1^m \cup \ldots \cup X_r^m \cup Y_1^m \cup \ldots \cup Y_r^m. \qquad (3.1.16)$$

Załóżmy dla dowodu nie wprost analogicznie do dowodu twierdzenia 3.1, że pewien niekluczowy atrybut $A \in Y_i^m$ okazał się tranzytywnie zależny od pewnego klucza K (niekoniecznie wyznaczonego) schematu R_m. Oznacza to, że istnieje taki podzbiór $T \subset R_m$, dla którego zachodzi

$$K \longrightarrow T,\ T \not\longrightarrow K,\ T \longrightarrow A \in F^+,\ A \notin T \cup K. \tag{3.1.17}$$

Z tranzytywnej zależności $K \longrightarrow A$ wynika też tranzytywna zależność A od klucza $X_i^m \in K_m$. Jeśli bowiem X_i^m jest kluczem, to $X_i^m \longrightarrow T \in F^+$. Ponadto zachodzi $T \not\longrightarrow X_i^m$, ponieważ ciąg zależności

$$T \longrightarrow X_i^m,\ X_i^m \longrightarrow K \tag{3.1.18}$$

implikowałby, na mocy aksjomatu A3, zależność $T \longrightarrow K$, co byłoby sprzeczne z (3.1.17).

Wykażemy teraz, że jeśli $X_i^m \longrightarrow A$ jest zależnością tranzytywną po scaleniu zależności (3.1.15), to musiała ona być redundancyjna w pokryciu H. Oznaczmy

$$R_m' = R_m - \{A\}. \tag{3.1.19}$$

Ponieważ $A \notin T$, więc $T \subseteq R_m'$. Z aksjomatu A1 Armstronga mamy wtedy

$$R_m' \longrightarrow T. \tag{3.1.20}$$

Wykażemy teraz, że ze zbioru $H - \{X_i^m \longrightarrow A\}$ jest wyprowadzalna zależność $X_i^m \longrightarrow R_m'$.

Zauważmy, że zastosowanie aksjomatów A1 i A3 do związków (3.1.11) i (3.1.12) prowadzi do zależności

$$L(g_{l_{i-1}}) \longrightarrow L(g_{l_i}),\quad i = 2,\ldots,s \tag{3.1.21}$$

$$L(g_{l_s}) \longrightarrow L(g_{l_1}). \tag{3.1.22}$$

Ponieważ $L(g_k) = X_k^m$, więc stosując potrzebną liczbę razy aksjomat A3 możemy uzyskać zbiór zależności

$$X_i^m \longrightarrow X_1^m,\ldots,X_i^m \longrightarrow X_{i-1}^m,\ X_i^m \longrightarrow X_{i+1}^m,\ldots,X_i^m \longrightarrow X_r^m. \tag{3.1.23}$$

Zauważmy, że wynik ten uzyskaliśmy nie wykorzystując zależności $X_i^m \longrightarrow A$. Załóżmy bowiem, że zależność ta wchodzi do pewnej zależności złożonej g_1 należącej do ciągu c_1. Rezygnacja z zależności $X_i^m \longrightarrow A$ oznacza usunię-

cie atrybutu A z $P(g_1)$. Ponieważ A jest atrybutem niekluczowym, więc dalej zachodzi $L(g_{l_{j+1}}) \subseteq P(g_{l_j})$, co zapewnia spełnienie zależności (3.1.21) i (3.1.22).

Z zależności (3.1.23) oraz następujących związków

$$X_1^m \longrightarrow Y_1^m,\ldots,X_i^m \longrightarrow (Y_i^m - \{A\}),\ldots,X_r^m \longrightarrow Y_r^m \tag{3.1.24}$$

na mocy aksjomatów Armstronga wyprowadzamy zależność

$$X_i^m \longrightarrow X_1^m \cup \ldots \cup X_r^m \cup Y_1^m \cup \ldots \cup (Y_i^m - \{A\}) \cup \ldots \cup Y_r^m, \tag{3.1.25}$$

to znaczy

$$X_i^m \longrightarrow R_m'. \tag{3.1.26}$$

Wykorzystując (3.1.26), (3.1.20) oraz aksjomat A3 możemy wyprowadzić zależność

$$X_i^m \longrightarrow T. \tag{3.1.27}$$

Zauważmy przy tym, że wynik ten uzyskaliśmy bez wykorzystania zależności $X_i^m \longrightarrow A$.

Na koniec możemy wykazać, w sposób analogiczny do dowodu twierdzenia 3.1, że zależność $T \longrightarrow A$ musi być wyprowadzalna ze zbioru $H - \{X_i^m \longrightarrow A\}$. Tak więc analizowane związki $X_i^m \longrightarrow R_m'$, $R_m' \longrightarrow T$, $T \longrightarrow A$ dowodzą możliwości wywodu zależności $X_i^m \longrightarrow A$ ze zbioru $H - \{X_i^m \longrightarrow A\}$. Jest to sprzeczne z założeniem, że H jest pokryciem nieredundancyjnym. □

Zauważmy, że w przypadku spełnienia założeń twierdzenia 3.2 dla wszystkich klas równoważności kluczy nieredundancyjnego pokrycia H danego zbioru F nie jest konieczna powtórna eliminacja zależności tranzytywnych wykonywana w kroku 5 algorytmu syntezy Bernsteina. Bazujący na tym spostrzeżeniu pomysł modyfikacji tego algorytmu dla klas równoważności liczących dwa klucze (twierdzenie 2.1) przedstawiono w pracy [58]. Sprawdzanie warunku OB dla klas równoważności o większej liczbie kluczy jest jednak zbyt skomplikowane i nieopłacalne.

Interesujące jest natomiast rozpatrzenie pewnej modyfikacji procesu wyznaczania pokrycia bazującej na twierdzeniach 3.1 i 3.2. Rozważmy przypadek, kiedy w zbiorze F obecne były wszystkie zależności tworzące bijekcje między wszystkimi kluczami należącymi do pewnej klasy równoważności liczącej więcej niż dwa klucze. Załóżmy, że w procesie usuwania zależności redundancyjnych żadne z tych zależności nie zostały usunięte. Własności takiego zbioru opisuje następujące twierdzenie:

TWIERDZENIE 3.3. Oznaczmy przez H_s takie pokrycie danego zbioru F, z którego usunięto wszystkie zależności redundancyjne z wyjątkiem zależności należących do bijekcji między kluczami równoważnymi. Załóżmy, że lewe strony następujących zależności funkcjonalnych:

$$X_1^m \rightarrow Y_1^m, \ldots, X_r^m \rightarrow Y_r^m \qquad (3.1.28)$$

należących do pokrycia H_s tworzą odrębną klasę równoważności K_m.

Oznaczmy przez S_m zbiór atrybutów kluczowych w klasie K_m, tzn.

$$S_m = \bigcup_{i=1}^{r} X_i^m \ . \qquad (3.1.29)$$

Jeżeli zachodzi

$$S_m \subseteq Y_i^m, \quad i = 1,\ldots,r, \qquad (3.1.30)$$

czyli dla każdego klucza $X_1^m \in K_m$ wszystkie zależności o lewej stronie równej X_i^m, wiążące ten klucz z pozostałymi kluczami klasy K_m, należą do pokrycia H_s, to schemat relacji utworzonej przez scalenie zależności (3.1.28) będzie w trzeciej postaci normalnej.

D o w ó d. Zwróćmy uwagę na to, że zgodnie z założeniami, jedynymi zależnościami redundancyjnymi mogą być w pokryciu H_s tylko zależności o prawych stronach złożonych z atrybutów kluczowych. Zależności takie nie naruszają własności trzeciej postaci normalnej tworzonego z nich schematu. Zauważmy następnie, że przy spełnieniu warunku (3.1.30) zależności (3.1.28) tworzą ciąg c_1 (3.1.10), słuszne są bowiem następujące związki

$$X_2^m \subseteq Y_1^m, \ X_3^m \subseteq Y_2^m, \ldots, X_r^m \subseteq Y_{r-1}^m, \ X_1^m \subseteq Y_r^m,$$

odpowiadające warunkom (3.1.11)-(3.1.14).

Tak więc w schemacie

$$R_m = X_1^m \cup \ldots \cup X_r^m \cup Y_1^m \cup \ldots \cup Y_r^m$$

żaden z atrybutów niekluczowych nie może być - zgodnie z twierdzeniem 3.2 - tranzytywnie zależny od żadnego z kluczy. Schemat R_m będzie więc w trzeciej postaci normalnej. □

3.2. MODYFIKACJA KONCEPCJI POKRYCIA ZBIORU ZALEŻNOŚCI FUNKCJONALNYCH - POKRYCIE DLA POTRZEB SYNTEZY

Zauważmy, że doprowadzenie do spełnienia założeń twierdzenia 3.3 nie jest operacją złożoną. Wymaga to jedynie uzupełnienia w zbiorze F - o ile potrzeba - takich zależności o prawych stronach będących atrybutami kluczowymi, aby w F były obecne wszystkie zależności funkcjonalne tworzące bijekcje między wszystkimi kluczami równoważnymi.

Praktycznie zaś należy dołączyć do prawych stron zależności (3.1.28) takie atrybuty kluczowe, aby był spełniony warunek (3.1.30). Następnie w procesie usuwania zależności redundancyjnych powinny zostać zachowane wszystkie bijekcje między kluczami równoważnymi.

Podejście to zmienia jednak dotychczasową koncepcję wyznaczania pokrycia w procesie syntezy. Pokrycie takie przestaje bowiem być nieredundancyjne, gdyż może zawierać niektóre redundancyjne zależności wiążące klucze równoważne. Pokrycie tego rodzaju, to znaczy zawierające wszystkie bijekcje między kluczami równoważnymi i nie zawierające żadnych zależności redundancyjnych poza bijekcjami, nazywać będziemy <u>pokryciem dla potrzeb syntezy</u>.

Podkreślmy, że celem wprowadzenia takiej koncepcji pokrycia jest uproszczenie algorytmu syntezy poprzez uniknięcie konieczności powtórnej eliminacji zależności tranzytywnych po scaleniu zależności o kluczach równoważnych.

Sformułowanie tak zmienionego algorytmu syntezy wymaga jednak rozpatrzenia kilku zagadnień szczegółowych.

3.2.1. <u>Klucze pozorne</u>

Zauważmy, że wprowadzenie do algorytmu syntezy koncepcji pokrycie dla potrzeb syntezy wymaga wyznaczenia kluczy równoważnych przed etapem eliminacji zależności redundancyjnych. W nowym algorytmie operacja podziału danego zbioru F na grupy zależności F_i o identycznych lewych stronach oraz operacja wyznaczania klas równoważności kluczy powinny więc poprzedzać etap usuwania zależności redundancyjnych. Na wstępie rozpatrzmy więc konsekwencje wykonania wymienionych operacji bez uprzedniej eliminacji zależności redundancyjnych.

<u>Uwaga 3.2.1</u>. Zauważmy, że podział zbioru F na grupy zależności o identycznych lewych stronach, prowadzony na podstawie aksjomatu A4 Armstronga, nie narzuca żadnych wymagań łączonym zależnościom, oprócz równości lewych stron. Stąd ewentualna redundancyjność łączonych zależności jest dla tego etapu obojętna.

<u>Uwaga 3.2.2</u>. W zbiorze lewostronnie zredukowanych (to znaczy nie posiadających obcych atrybutów) zależności funkcjonalnych proces wyznaczania pokrycia poprzez eliminację zależności redundancyjnych może prowadzic

do usunięcia pewnej liczby kluczy, ale nie narusza postaci kluczy pozostałych.

Spostrzeżenie to wynika bezpośrednio z definicji zależności redundancyjnej. Załóżmy, że zależność $f: X_i \longrightarrow Y_i \in F$ reprezentuje wszystkie zależności o lewej stronie równej X_i istniejące w zbiorze F.

Klucz X_i jest usuwany, jeśli dla każdego atrybutu $A \in Y_i$ zależność funkcjonalna $X_i \longrightarrow A$ jest redundancyjna. W przeciwnym przypadku w pokryciu pozostanie zależność $f': X_i \longrightarrow Y_i'$, gdzie $Y_i' \subseteq Y_i$, o kluczu (lewej stronie) identycznym z kluczem zależności f.

Nazwijmy klucze, które zostaną usunięte w procesie wyznaczania pokrycia kluczami pozornymi.

Analizując możliwość istnienia kluczy pozornych w trakcie wyznaczania kluczy równoważnych, a dalej dołączania brakujących bijekcji, rozpatrzmy dwa przypadki:

(a) Klucz pozorny tworzy samodzielnie odrębną klasę równoważności.

Jeśli pewna klasa równoważności zawiera tylko jeden klucz (klasy takie będziemy nazywali trywialnymi), to z klasę tę nie są związane żadne bijekcje. Wobec tego fakt, czy klucz ten jest kluczem pozornym, czy nie, nie ma wpływu na proces wyznaczania pokrycia.

(b) Klucz pozorny należy do nietrywialnej klasy równoważności, czyli klasy obejmującej dwa lub więcej kluczy.

W takim przypadku powstaje szereg pytań, które przeanalizujemy kolejno:

(1) Czy klucze tworzące pewną klasę równoważności względem bijekcji pozostaną dalej kluczami równoważnymi, jeśli jeden z nich okaże się kluczem pozornym i zostanie usunięty?

Odpowiedź na to pytanie daje następujący lemat:

Lemat 3.1. Niech będzie dana pewna klasa równoważności kluczy względem bijekcji $K_m = \{X_1^m, \ldots, X_r^m\}$ wyznaczona dla zbioru zależności funkcjonalnych F. Jeśli jeden z kluczy należących do K_m okaże się kluczem pozornym i zostanie usunięty, to pozostałe klucze będą w dalszym ciągu tworzyły klasę równoważności.

D o w ó d. Załóżmy, że usuniętym kluczem pozornym był klucz $X_i^m \in K_m$. Oznaczmy zbiór pozostałych kluczy przez $K_m' = \{X_1^m, \ldots, X_{i-1}^m, X_{i+1}^m, \ldots, X_r^m\}$. Zauważmy, że usunięcie klucza X_i^m jest konsekwencją usunięcia ze zbioru F wszystkich zależności o lewej stronie równej X_i^m jako zależności redundancyjnych. Oznaczmy przez F_1 zbiór pozostałych zależności. Z równości (2.3.1) wynika równość domknięć $F_1^+ = F^+$. Wobec tego każda bijekcja $X_j^m \longleftrightarrow X_k^m$ (gdzie $X_j^m, X_k^m \in K_m$), która należała do F^+, należy też do F_1^+. Oznacza to, że wszystkie klucze należące do K_m' są nadal kluczami równoważnymi. □

Drugą wątpliwość dotyczącą przypadku (b) ujmuje następujące pytanie:

(2) Czy wprowadzenie do zbioru F zależności tworzących bijekcje między kluczem pozornym a innymi kluczami równoważnymi nie zakłóci procesu wyznaczania pokrycia?

W tym przypadku można wskazać przykład potwierdzający te obawy.

Przykład 3.2.1. Rozpatrzmy następujący zbiór zależności funkcjonalnych $\{A \longrightarrow BF, BC \longrightarrow DE, DE \longrightarrow FC, AC \longrightarrow DE, FD \longrightarrow G, GE \longrightarrow A\}$. W zbiorze tym występują trzy klucze równoważne BC, DE oraz AC, przy czym klucz AC jest kluczem pozornym, to znaczy zależność $AC \longrightarrow DE$ jest zależnością redundancyjną. Jeżeli zależność ta zostanie usunięta z powyższego zbioru, to żadna z pozostałych zależności nie będzie redundancyjna. Jeśli jednak przyjmiemy wstępnie, że AC jest kluczem równoważnym i wprowadzimy do rozważanego zbioru zależności $B \longrightarrow A$ oraz $DE \longrightarrow AB$ dla uzupełnienia wszystkich bijekcji między wymienionymi kluczami, to przy pewnej kolejności szukania zależności redundancyjnych jako pierwsza może zostać usunięta zależność $DE \longrightarrow F$. Następnie po wykryciu faktu, że klucz AC jest kluczem pozornym i usunięciu związanych z nim bijekcji, ze zbioru pozostałych zależności $\{A \longrightarrow BF, BC \longrightarrow DE, DE \longrightarrow C, FD \longrightarrow G, GE \longrightarrow A\}$ nie można wyprowadzić zależności $DE \longrightarrow A$ i $DE \longrightarrow B$ poprzednio wyprowadzalnych. □

Z powyższego przykładu wynika, że rozpoznanie i usunięcie kluczy pozornych musi nastąpić przed włączeniem do zbioru F zależności uzupełniających w tym zbiorze komplet bijekcji łączących klucze równoważne. Z kolei lemat 3.1 pokazuje, że operacja ta może mieć miejsce po wyznaczeniu klas równoważności. Jest to o tyle istotne, że proces szukania kluczy pozornych można ograniczyć - zgodnie z wnioskiem z analizy przypadku (a) - do nietrywialnych klas równoważności.

Rozpatrzmy jeszcze sposób sprawdzania, czy dany klucz jest kluczem pozornym.

Lemat 3.2. Zależność $X \longrightarrow Y \in F$, gdzie $Y = \{A_1, \ldots, A_t\}$ jest redundacyjna w F wtedy i tylko wtedy, jeśli $\forall A_i \in Y: X \longrightarrow A_i \in (F - \{X \longrightarrow Y\})^+$.

D o w ó d (1) Wykażemy najpierw, że jeśli zależność $X \longrightarrow Y$ jest redundacyjna w F, to $\forall A_i \in Y: X \longrightarrow A_i \in (F - \{X \longrightarrow Y\})^+$.

Redundancyjność złożonej zależności $X \longrightarrow Y$ w zbiorze F oznacza, że wskazać można pewien ciąg operacji usuwania ze zbioru F zależności o postaci $X \longrightarrow A_i$, $A_i \in Y$ jako zależności redundancyjnych. Dla prostoty załóżmy, że ciąg ten ma następującą postać $X \longrightarrow A_1, X \longrightarrow A_2, \ldots, X \longrightarrow A_t$. Jeśli zależność $X \longrightarrow A_1$ jest redundancyjna w F, to $X \longrightarrow A_1 \in (F - \{X \longrightarrow A_1\})^+$. Oznaczmy: $F_1 = F - \{X \longrightarrow A_1\}$. Z własności (2.3.1) wynika, że $F_1^+ = F^+$. Oznaczmy kolejno $F_2 = F_1 - \{X \longrightarrow A_2\}, \ldots, F_t = F_{t-1} - \{X \longrightarrow A_t\}$. Jeśli kolejne zależności są redundancyjne, to zachodzi $X \longrightarrow A_2 \in F_2^+, \ldots,$ $X \longrightarrow A_t \in F_t^+$ oraz $F_t^+ = F_{t-1}^+ = \ldots = F_1^+ = F^+$.

Zauważmy, że $F_t = F - \{X \longrightarrow A_1\} - \ldots - \{X \longrightarrow A_t\} = F - \{X \longrightarrow Y\}$, co kończy ten etap dowodu.

(2) W drugim kroku wykażemy, że jeśli $\forall A_i \in Y: X \longrightarrow A_i \in (F - \{X \longrightarrow Y\})^+$, to $X \longrightarrow A_i \in (F - \{X \longrightarrow A_i\})^+$.

Dowód jest oczywisty, ponieważ

$$F - \{X \longrightarrow Y\} \subseteq F - \{X \longrightarrow A_i\}. \square$$

Zauważmy, że przedstawiony lemat można byłoby potraktować jako definicję redundancyjności zależności funkcjonalnej o złożonej prawej stronie.

Lemat ten pozwala wykonać jedną operację wyznaczania domknięcia w celu ustalenia, czy dany klucz X jest kluczem pozornym. Posługując się definicją zależności redundancyjnej dla przypadku, kiedy prawa strona jest atrybutem prostym, należałoby wykonać w rozważanym przykładzie t takich operacji.

3.2.2. Możliwość wyznaczania dodatkowych kluczy relacji wśród kluczy pozornych

Badanie kluczy pozornych pozwala w pewnych przypadkach na wyznaczenie, a właściwie ujawnienie dodatkowych kluczy relacji poza kluczami wyznaczonymi w procesie syntezy. Uzasadnia to następujący lemat.

Lemat 3.3. Załóżmy, że w zbiorze F wyznaczona została następująca klasa równoważności kluczy względem bijekcji $K_m = \{X_1^m, \ldots, X_r^m\}$. Jeśli jeden z kluczy $X_i^m \in K_m$ jest kluczem pozornym i został usunięty z klasy K_m, ale atrybuty tworzące klucz X_i^m znalazły się w schemacie R_m' utworzonym dla klasy $K_m' = \{X_1^m, \ldots, X_{i-1}^m, X_{i+1}^m, \ldots, X_r^m\}$, to X_i^m będzie również kluczem relacji o schemacie R_m'.

D o w ó d. Istnienie klucza pozornego X_i^m w zbiorze F oznacza, że każda zależność f taka, że $L(f) = X_i^m$ jest redundancyjna w F. Oznaczmy przez F_1 zbiór, który powstał z F po usunięciu tych zależności. Z (2.3.1) wynika, że $F_1^+ = F^+$. Wobec tego dla każdego klucza $X_k^m \in K_m'$ zachodzi $X_k^m \longleftrightarrow X_i^m \in F_1^+$. A więc jeśli $X_i^m \subseteq R_m'$, to każdy niekluczowy atrybut $A \in R_m'$ zależny funkcjonalnie od klucza X_k^m jest też funkcjonalnie zależny od atrybutu (być może złożonego) X_i^m. Atrybut X_i^m jest więc kluczem w schemacie R_m'. $\square$

Przykład 3.2.2. W zbiorze zależności funkcjonalnych $F = \{A \longrightarrow B, BC \longrightarrow D, AC \longrightarrow D, D \longrightarrow ACE\}$ wyodrębnić możemy dwie klasy równoważności kluczy $K_1 = \{A\}$ oraz $K_2 = \{BC, AC, D\}$. Klucz AC jest jednak kluczem pozornym, ponieważ zależność $AC \longrightarrow D$ jest wyprowadzalna z zależności $A \longrightarrow B$, $BC \longrightarrow D$. Usunięcie zależności $AC \longrightarrow D$ prowadzi do modyfikacji klasy K_2 do postaci $K_2' = \{BC, D\}$. Zbiór F nie zawiera dalszych zależności redundancyjnych, wobec czego klasom równoważności K_1 i K_2' odpowiadają następujące schematy relacji: $R_1 = AB$ z kluczem A oraz $R_2' = ABCDE$ z kluczami BC i D. Ponieważ schemat R_2' zawiera atrybuty AC (klucz pozorny),

więc zgodnie z lematem 3.3 atrybuty te tworzą dodatkowy klucz relacji o tym schemacie. $\square$

Na zakończenie tego punktu zauważmy, że klucze zdefiniowane w lemacie 3.3 nie są ujawniane w algorytmie Bernsteina. Wskazanie takich kluczy może być natomiast istotne dla użytkownika, koresponduje to bowiem ze zdefiniowaną przez niego początkową postacią zbioru F.

3.2.3. Organizacja procesu usuwania zależności redundancyjnych przy wymuszeniu obecności bijekcji w pokryciu

Rozważania przedstawione w poprzednich punktach tego rozdziału pozwalają na następującą ogólną prezentację procesu tworzenia pokrycia dla potrzeb syntezy:

Algorytm 3.1. (Tworzenie pokrycia dla potrzeb syntezy)

Dane wejściowe: Zbiór zależności funkcjonalnych F.

Dane wyjściowe: Pokrycie dla potrzeb syntezy H_s.

(1) Usuń obce atrybuty z danego zbioru F zależności funkcjonalnych.

(2) Podziel zbiór F na grupy zależności F_i o identycznych lewych stronach.

(3) Wyznacz klasy równoważności dla bijekcji w zbiorze kluczy (lewych stron zależności F_i). Usuń klucze pozorne spośród kluczy równoważnych.

(4) Uzupełnij zbiór F o brakujące zależności tworzące bijekcje między kluczami równoważnymi. Usuń zależności redundancyjne przy zachowaniu wszystkich bijekcji.

W niniejszym podrozdziale rozpatrzymy szczegółowo krok (4) powyższego algorytmu. Celem naszym będzie doprowadzenie do spełnienia założeń twierdzenia 3.3 (gwarantuje ono trzecią postać normalną schematów relacji). Dla każdej klasy równoważności kluczy, np. $K_m = \{X_1^m, \ldots, X_r^m\}$, należy wobec tego na wstępie uzupełnić zbiór F o takie zależności, aby każda bijekcja $X_i^m \longleftrightarrow X_j^m$; $X_i^m, X_j^m \in K_m$ należała do F.

Przykład 3.2.3. Zbiór zależności funkcjonalnych z przykładu 2.3.1 w postaci $F = \{XY \longrightarrow AD, CD \longrightarrow EY, XA \longrightarrow B, YB \longrightarrow C, C \longrightarrow A, CE \longrightarrow H, DH \longrightarrow X, X \longrightarrow E\}$, w którym występują dwa klucze równoważne XY i CD, należałoby według powyższych wskazówek uzupełnić zależnościami $XY \longrightarrow C$ oraz $CD \longrightarrow X$. $\square$

Załóżmy, że dla zbioru F wyznaczone zostały następujące klasy równoważności kluczy: $K_1, K_2, \ldots, K_p$. Załóżmy dalej, że dołączane do F zależności funkcjonalne (będące bijekcjami lub ich składowymi) tworzą podzbiory $B_1, B_2, \ldots, B_p$. Przynależność danej zależności $f: X \longrightarrow A$ do podzbioru B_i oznacza, że $X \in K_i$. Dla ogólności zakładać będziemy, że niektóre spośród tych podzbiorów mogą być puste. Zbiór wszystkich dołączanych zależności oznaczymy przez B:

$$B = B_1 \cup B_2 \cup \ldots \cup B_p. \qquad (3.2.1)$$

Przypomnijmy, że celem dołączania zależności B jest usunięcie tych zależności redundancyjnych, które w algorytmie Bernsteina ujawniały się po scaleniu zależności o kluczach równoważnych.

Wykażemy, że proces dołączania zależności B i usuwania zależności redundancyjnych można zdekomponować do grup zależności odpowiadających poszczególnym klasom równoważności kluczy. Pozwala to na łatwiejszą implementację przedstawionego algorytmu.

Lemat 3.4. Zależność funkcjonalna $f: X \longrightarrow A \in F$, $X \in K_i$ jest redundancyjna w zbiorze $F \cup B$ wtedy i tylko wtedy, jeśli jest redundancyjna w zbiorze $F \cup B_i$.

D o w ó d. W dowodzie wykorzystamy dwa lematy (3.5 i 3.6) przedstawiona bezpośrednio po zakończeniu dowodu.

(1) W pierwszym etapie dowodu zaprzeczymy tezie zakładając, że f jest redundancyjna w zbiorze $F \cup B$, ale nie jest redundancyjna w $F \cup B_i$. Wynika stąd, że zależność f staje się redundancyjna po dołączeniu do $F \cup B_i$ jednej lub większej liczby zależności należących do zbioru $B_1 \cup \ldots \cup B_{i-1} \cup B_{i+1} \cup \ldots \cup B_p$. Oznaczmy przez F_1 (gdzie $F_1 \supseteq F \cup B_i$) taki zbiór zależności, w którym f jeszcze nie jest redundancyjna, natomiast przez $g: Y \longrightarrow D$ taką zależność, dla której f jest redundancyjna w zbiorze $F_1 \cup \{g\}$. Zakładamy przy tym, że zachodzi

$$F_1 \subseteq F \cup B,\ g \in B_1 \cup B_2 \cup \ldots \cup B_{i-1} \cup B_{i+1} \cup \ldots \cup B_p,\ g \notin B_i. \qquad (3.2.2)$$

Z lematu 3.5 wynika, że $F_1^+ = F^+$. Tak postawiony problem spełnia założenia lematu 3.6. Wynika z niego, że do domknięcia F_1^+ należy bijekcja $X \longleftrightarrow Y$, co oznacza, że X i Y są kluczami równoważnymi. Jednakże zgodnie z (3.2.2) $g \notin B_i$, a więc $Y \notin K_i$, co jest sprzeczne z powyższym wnioskiem i założeniem, że $X \notin K_i$.

(2) Prowadząc dowód w drugim kierunku mamy wykazać, że jeśli f jest redundancyjna w $F \cup B_i$, to jest też redundancyjna w $F \cup B$.

Redundancyjność w $F \cup B_i$ oznacza, że $f \in (F \cup B_i - \{f\})^+$. Wobec tego zależność f musi też być wyprowadzalna ze zbioru $F \cup B_1 \cup \ldots \cup B_i \cup \ldots \cup B_p - \{f\}$, co oznacza redundancyjność w zbiorze $F \cup B$. □

Lemat 3.5. Niech będzie dany zbiór zależności funkcjonalnych F i zależność funkcjonalna f o własnościach: $f \in F^+$, $f \notin F$. Zachodzi wtedy $(F + \{f\})^+ = F^+$. □

D o w ó d. Przyjmijmy oznaczenie $G = F \cup \{f\}$.
Mamy: $f \in G$ i f jest redundancyjna w G, ponieważ $f \in F^+$; $F^+ = (G - \{f\})^+$. Wobec tego z własności (2.3.1) mamy $(G - \{f\})^+ = G^+$, co daje $F^+ = (F + \{f\})^+$. □

Lemat 3.6. Niech będzie dany zbiór zależności funkcjonalnych F oraz dwie zależności $f: X \longrightarrow A$ oraz $g: Y \longrightarrow B$ o własnościach:

$$f \in F,\ f \notin (F - \{f\})^+,\ g \notin F,\ g \in F^+. \qquad (3.2.3)$$

Jeżeli $f \in (F \cup \{g\} - \{f\})^+$, to $X \longleftrightarrow Y \in F^+$.

D o w ó d. Wykażemy najpierw, że zależność g nie może być redundancyjna w zbiorze $F \cup \{g\} - \{f\}$.

Musiałoby bowiem wtedy zachodzić:

$g \in (F - \{f\})^+$ albo zgodnie z (2.3.1)

$(F \cup \{g\} - \{f\})^+ = (F - \{f\})^+$.

Oznaczałoby to sprzeczność z założeniami, bowiem własności

$$f \notin (F - \{f\})^+,\ f \in (F \cup \{g\} - \{f\})^+ \qquad (3.2.4)$$

implikują związek

$$(F - \{f\})^+ \neq (F \cup \{g\} - \{f\})^+. \qquad (3.2.5)$$

Jeśli więc g nie jest redundancyjna w zbiorze $F \cup \{g\} - \{f\}$, to $g \notin (F - \{f\})^+$.

Ponieważ według założeń $g \in F^+$, więc zależność f musi zostać użyta do wywodu zależności g z F.

Z lematu 2.1 wynika wtedy, że spełniona jest zależność

$$Y \longrightarrow X \in F^+. \qquad (3.2.6)$$

Ponadto z własności (3.2.4) wynika, że zależność g służy do wywodu f ze zbioru $F - \{f\}$. Tak więc według lematu 2.1 spełniona jest zależność

$$X \longrightarrow Y \in F^+. \qquad (3.2.7)$$

Zestawienie zależności (3.2.6) i (3.2.7) kończy dowód. □

Z lematu 3.4 wynika, że dla procesu usuwania ze zbioru F zależności redundancyjnych o lewych stronach należących do pewnej klasy równoważności K_m (w obecności dołączonych bijekcji B_m) jest obojętne, czy dodatkowe zależności tworzące bijekcje dla innych klas zostały wprowadzone do zbioru F, czy też nie.

Przedstawione rozważania pozwalają na sformułowanie następującego algorytmu usuwania zależności redundancyjnych przy wymuszonej obecności bijekcji, prezentującego szczegółowo sposób wykonania kroku (4) algorytmu 3.1.

Algorytm 3.2. (Usuwanie zależności redundancyjnych przy wymuszeniu obecności bijekcji)

Dane wejściowe: (1) Zbiór klas równoważności kluczy

$$\{K_1, K_2, \ldots, K_p\}.$$

(2) Zbiór F zależności funkcjonalnych, pogrupowanych według przynależności lewych stron do klas równoważności kluczy; np. klasie równoważności $K = \{X_1^m, \ldots, X_r^m\}$ odpowiadają zależności $f_1^m: X_1^m \longrightarrow Y_1^m, \ldots, f_r^m: X_r^m \longrightarrow Y_r^m$.

Dane wyjściowe: zmodyfikowany zbiór F zawierający między innymi wszystkie bijekcje między kluczami równoważnymi i nie zawierający żadnych zależności redundancyjnych nie należących do bijekcji.

Treść algorytmu:

Dla każdej z klas równoważności kluczy K_m, $m = 1,\ldots,r$, wykonaj następujące operacje:

(a) Jeśli K_m jest trywialną klasą równoważności (liczącą jeden klucz), wtedy dla każdej zależności f_i^m, $i = 1,\ldots,r$, i dla każdego atrybutu $A \in Y_i^m$ wykonaj procedurę usuwania.

(b) Jeśli K_m jest nietrywialną klasą równoważności, wykonaj trzy grupy operacji

(1) Wyznacz zbiór atrybutów kluczowych S_m dla klasy K_m

$$S_m = \bigcup_{j=1}^{r} X_j^m.$$

(2) Przekształć prawe strony Y_i^m wszystkich zależności f_i^m, $i = 1\ldots r$, w następujący sposób:

(A) Oddziel atrybuty kluczowe od niekluczowych

$$Y_i^m = Y_i'^m \cup Y_i''^m$$

gdzie $Y_i'^m \cap S_m = \{\emptyset\}$ oraz $Y_i''^m \subseteq S_m$,

to znaczy podzbiór $Y_i'^m$ tworzą atrybuty kluczowe, $Y_i''^m$ – atrybuty niekluczowe.

(B) Uzupełnij atrybuty kluczowe w prawej stronie Y_i^m (dołączenie zależności tworzących bijekcje):

$$Y_i^m := Y_i^m - Y_i''^m \cup S_m.$$

(3) Dla każdej zależności f_i^m, $i = 1,\ldots,r$, i dla każdego niekluczowego atrybutu $A \in Y_i'^m$ wykonaj procedurę usuwania.

Procedura usuwania:

(1) Zmodyfikuj prawą stronę Y_i^m zależności f_i^m

$$Y_i^m := Y_i^m - \{A\}.$$

(2) Wyznacz domknięcie $(X_i^m)^+$ w zbiorze F.

(3) Sprawdź, czy $A \in (X_i^m)^+$. Jeśli tak – usuń atrybut A, to znaczy pozostaw bez zmian zmodyfikowaną prawą stronę Y_i^m. Jeśli nie – dołącz ponownie atrybut A

$$Y_i^m := Y_i^m \cup \{A\}. \qquad \square$$

Komentując przedstawiony algorytm zauważmy, że:

- zastąpienie $Y_i''^m$ przez S_m dla wszystkich f_i^m, $i = 1,\ldots,r$, odpowiada dołączeniu do zbioru F zależności B_m, a więc wymuszeniu obecności wszystkich bijekcji,
- ewentualnemu usuwaniu z prawych stron podlegają jedynie atrybuty niekluczowe $A \in Y_i'^m$, co umożliwia zachowanie w pokryciu wszystkich bijekcji, a więc zapewnia spełnienie założeń twierdzenia 3.3,
- algorytm wyznaczania domknięcia przedstawiony został w rozdziale 2 (algorytm 2.1).

Rozważania dotyczące wyznaczania pokrycia dla potrzeb syntezy podsumujemy następującym twierdzeniem.

TWIERDZENIE 3.4. Ze zbioru H_s tworzącego pokrycie dla potrzeb syntezy wyprowadzalne są wszystkie zależności funkcjonalne tworzące początkowy zbiór F, to znaczy $H_s^+ = F^+$.

D o w ó d. Rozpatrzmy te kroki algorytmów 3.1 i 3.2, w których pewne zależności są eliminowane bądź dołączane do zbioru F.

- W trakcie eliminacji obcych atrybutów dana zależność funkcjonalna $X \longrightarrow A \in F$ może zostać zastąpiona inną zależnością $X' \longrightarrow A$, gdzie $X' \subset X$, o ile $X' \longrightarrow A \in F^+$. Ponieważ stosując aksjomat A2 Armstronga, z zależności $X' \longrightarrow A$ możemy zawsze wyprowadzić zależność $X \longrightarrow A$, więc zbiór F_1 utworzony ze zbioru F na etapie eliminacji obcych atrybutów posiada to samo domknięcie co zbiór F:

$$F_1^+ = F^+. \tag{3.2.8}$$

- Usunięcie kluczy pozornych jest związane z usunięciem pewnych zależności redundancyjnych, co nie zmienia domknięcia zgodnie z własnością (2.3.1). Oznaczając przez F_2 zbiór powstały po eliminacji kluczy pozornych otrzymamy:

$$F_2^+ = F_1^+ = F^+. \tag{3.2.9}$$

- Każda zależność wprowadzana do zbioru F_2 w celu utworzenia wszystkich bijekcji między kluczami równoważnymi jest wyprowadzalna ze zbioru F. Zgodnie z lematem 3.5 wprowadzenie takiej zależności do zbioru nie zmienia jego domknięcia.

Oznaczając przez B wprowadzane zależności otrzymamy więc

$$(F_2 \cup B)^+ = F_2^+ = F^+. \qquad (3.2.10)$$

- Eliminacja zależności redundancyjnych ze zbioru $F_2 \cup B$ dla uzyskania pokrycia dla potrzeb syntezy H_s nie zmienia domknięcia zbioru wyjściowego, zgodnie z własnością (2.3.1).

Ostatecznie mamy więc

$$H_s^+ = (F_2 \cup B)^+ = F^+. \ \square \qquad (3.2.11)$$

3.3. ALGORYTM SYNTEZY PRZY ZMIENIONEJ KONCEPCJI POKRYCIA

Prosta rozbudowa przedstawionego uprzednio algorytmu tworzenia pokrycia dla potrzeb syntezy pozwala uzyskać następujący algorytm syntezy:

Algorytm 3.3. (Synteza schematu relacyjnej bazy danych)

Dane wejściowe: Zbiór zależności funkcjonalnych F.

Dane wyjściowe: Zbiór schematów relacji $\mathbb{R} = \{R_1,\ldots,R_n\}$ tworzących schemat relacyjnej bazy danych.

(1) Usuń obce atrybuty z danego zbioru F zależności funkcjonalnych.

(2) Podziel zbiór F na grupy zależności F_i o identycznych lewych stronach.

(3) Wyznacz klasy równoważności dla bijekcji w zbiorze kluczy (lewych stron zależności F_i). Usuń klucze pozorne spośród kluczy równoważnych.

(4) Uzupełnij zbiór F o brakujące zależności tworzące bijekcje między kluczami równoważnymi. Utwórz pokrycie dla potrzeb syntezy H_s, usuwając zależności redundancyjne przy zachowaniu wszystkich bijekcji.

(5) Dla każdej klasy równoważności kluczy z pokrycia H_s utwórz schemat relacji jako sumę logiczną atrybutów tworzących zależności o kluczach z tej klasy.

TWIERDZENIE 3.5. Niech $\mathbb{R} = \{R_1,\ldots,R_n\}$ będzie schematem relacyjnej bazy danych utworzonym za pomocą algorytmu 3.3 dla zadanego zbioru F zależności funkcjonalnych. Wtedy każdy ze schematów $R_i \in \mathbb{R}$ jest w trzeciej postaci normalnej, a w relacyjnej bazie danych o schemacie $\mathbb{R}$ spełnione są wszystkie zależności ze zbioru F.

D o w ó d. Zauważmy, że w pokryciu dla potrzeb syntezy H_s, utworzonym w kroku (4) algorytmu 3.3, spełnione są założenie twierdzenia 3.3 dla

każdej z klas równoważności kluczy. Z twierdzenia tego wynika więc, że każdy ze schematów $R_i \in \mathbb{R}$ jest w trzeciej postaci normalnej.

Zauważmy dalej, że w rozważanej relacyjnej bazie danych spełnione są wszystkie zależności reprezentowane w pokryciu dla potrzeb syntezy H_s, a także wyprowadzalne z H_s. Z twierdzenia 3.4 wynika zaś równość domknięć $H_s^+ = F^+$, co kończy dowód. $\square$

Wykazanie ostatniej z pożądanych cech schematu $\mathbb{R}$, a mianowicie tego, że liczba schematów relacji w $\mathbb{R}$ jest minimalna, wymaga szerszych rozważań. Sformułowanie końcowego twierdzenia poprzedzimy analizą tych kroków algorytmu syntezy, które mogą doprowadzić do różnych postaci pokrycia dla potrzeb syntezy. Przyczyną różnic może być różne uporządkowanie zależności funkcjonalnych w wejściowym zbiorze F.

Przykład 3.3.1. Rozpatrzmy następujący zbiór zależności funkcjonalnych: $F = \{A \to BCD,\ B \to AG,\ CD \to G,\ G \to CD\}$. W zbiorze tym występują dwie klasy równoważności kluczy $K_1 = \{A,B\}$, $K_2 = \{CD,G\}$. Pokrycie dla potrzeb syntezy zbioru F może mieć jedną z dwóch postaci:

$$H_s^1 = \{A \to B,\ B \to AG,\ CD \to G,\ G \to CD\}$$

lub

$$H_s^2 = \{A \to BC,\ B \to A,\ CD \to G,\ G \to CD\},$$

zależnie od tego, która z dwóch zależności $A \to CD$ lub $B \to G$ badana była pierwsza przy usuwaniu zależności redundancyjnych. $\square$

Lemat 3.7. Usuwanie obcych atrybutów w zbiorze F zależności funkcjonalnych daje wynik niezależny od kolejności operacji usuwania.

D o w ó d. Operację usuwania obcych atrybutów możemy zapisać następująco:

jeżeli $X \to A \in F$ oraz $X' \to A \in F^+$, gdzie $X' \subset X$,

to

$$F := F - \{X \to A\} \cup \{X' \to A\}. \qquad (3.3.1)$$

Załóżmy, że po usunięciu wszystkich obcych atrybutów otrzymano dwa różne zbiory zależności F_1 i F_2, przy czym musi zachodzić $F_1^+ = F_2^+ = F^+$. Oznacza to, że w co najmniej jednym przypadku istnieją dwie takie zależności $f_1 \in F_1$ i $f_2 \in F_2$, że

$$P(f_1) = P(f_2) \quad \text{oraz} \quad L(f_1) \subset L(f_2)$$

lub

$$L(f_1) \supset L(f_2), \qquad (3.3.2)$$

gdzie $L(f)$ i $P(f)$ oznaczają lewą i prawą stronę zależności f.

Rozważmy najpierw przypadek $L(f_1) \subset L(f_2)$. Ponieważ $f_1 \in F_1^+$, zaś $F_1^+ = F_2^+$, a więc $f_1 \in F_2^+$. Atrybuty tworzące podzbiór $L(f_2) - L(f_1)$ są atrybutami obcymi w lewej stronie zależności f_2, co jest sprzeczne z założeniami. Dla $L(f_2) \supset L(f_1)$ tok dowodu jest analogiczny. □

Zauważmy, że również wyniki dwóch następnych operacji algorytmu syntezy, to znaczy podziału zbioru F na grupy zależności F_i oraz wyznaczania klas równoważności kluczy, nie mogą zależeć od uporządkowania zbioru F. Nie można natomiast przesądzić możliwości usunięcia różnych kluczy pozornych z danej klasy równoważności kluczy K_m, zależnie od kolejności wykrywania takich kluczy. Załóżmy, że prowadzi to do klas K_m^1 i K_m^2. Klasy te nazwiemy <u>alternatywnymi klasami równoważności kluczy</u>. Z lematu 3.1 wynika, że klucze klas K_m^1 i K_m^2 są związane bijekcjami w domknięciu F^+. Powyższe rozważania prowadzę do następującego wniosku:

<u>Uwaga 3.3.1</u>. Niech F_1 i F_2 będą dwoma zbiorami zależności funkcjonalnych otrzymanymi z wejściowego zbioru F po krokach 1,2,3 algorytmu syntezy 3.3. Jeśli $F_1 \neq F_2$, to każdej klasie równoważności kluczy utworzonej dla F_1 odpowiada identyczna lub alternatywna klasa dla F_2.

<u>Lemat 3.8</u>. Niech H_s^1 i H_s^2 będą dwoma różnymi pokryciami dla potrzeb syntezy utworzonymi dla zbioru F zależności funkcjonalnych. Wtedy dla każdej zależności $f: X \longrightarrow A$ należącej do H_s^1 istnieje w H_s^2 taka zależność $g: Y \longrightarrow B$, że lewe strony X i Y związane są bijekcją w domknięciu zbioru F, tzn.

$$L(f) \longleftrightarrow L(g) \in F^+.$$

D o w ó d. Wyeliminujmy na początku z naszych rozważań zbiór zależności należących równocześnie do obu pokryć (spełniających tezę w sposób oczywisty). Zauważmy następnie że jeśli pewna zależność $f \in H_s^1$ będzie redundancyjna w H_s^1, to ze sposobu wyznaczania pokrycia H_s wynika, że zależność ta tworzy jedną z bijekcji między kluczami równoważnymi. A więc $L(f)$ należy do nietrywialnej, a ściślej alternatywnej klasy równoważności kluczy dla bijekcji. Ponieważ po kroku 3 algorytmu syntezy klucze należące do takich klas nie mogą być kluczami pozornymi, więc zgodnie z uwagą 3.2.2 proces eliminacji zależności redundancyjnych w kroku (4) nie naruszy postaci żadnych z tych kluczy. Wobec tego zgodnie z uwagą 3.3.1 klasie równoważności dla H_s^1 zawierającej $L(f)$ odpowiadać musi alternatywna klasa równoważności dla pokrycia H_s^2.

Ponieważ zgodnie z lematem 3.1 klucze takich klas związane są bijekcjami w F^+, więc dla rozważanej dowolnej redundancyjnej zależności $f \in H_s^1$ teza jest spełniona.

Zasadniczą część dowodu przeprowadzimy dla takich zależności f należących do H_s^1, które nie należą do H_s^2 i nie są redundancyjna w H_s^1, tzn.:

$$f \in H_s^1,\ f \notin H_s^2,\ f \notin (H_s^1 - \{f\})^+. \tag{3.3.3}$$

Zauważmy na wstępie, że zgodnie z twierdzeniem 3.4 zachodzi

$$H_s^{1+} = H_s^{2+} = F^+. \tag{3.3.4}$$

Ponieważ $f \notin H_s^2$, ale $f \in H_s^{2+}$, więc istnieje zależność $g \in H_s^2$ taka, że

$$f \in (H_s^1 - \{f\} \cup \{g\})^+ \tag{3.3.5}$$

lub też istnieje podzbiór $G \subseteq H_s^2$ taki, że

$$f \in (H_s^1 - \{f\} \cup G)^+. \tag{3.3.6}$$

Rozpatrzmy najpierw pierwszy przypadek.

Z zależności (3.3.5) oraz lematu 2.1 wynika, że spełniona jest zależność

$$L(f) \rightarrow L(g) \in (H_s^1 - \{f\} \cup \{g\})^+ = H_s^{1+} = F^+. \tag{3.3.7}$$

Załóżmy chwilowo, że $g \notin H_s^1$ i rozpatrzmy problem, czy zależność g można wyprowadzić ze zbioru $H_s^1 - \{f\}$.
Jeśli tak, to $g \in (H_s^1 - \{f\})^+$, a więc z lematu 3.5 wynika, że

$$(H_s^1 - \{f\})^+ = (H_s^1 - \{f\} \cup \{g\})^+. \tag{3.3.8}$$

Biorąc pod uwagę zależność (3.3.5) i lemat 3.5 otrzymamy dalej

$$(H_s^1 - \{f\} \cup \{g\})^+ = (H_s^1 \cup \{g\})^+ = H_s^{1+}. \tag{3.3.9}$$

Z porównania (3.3.8) i (3.3.9) wynika, że

$$(H_s^1 - \{f\})^+ = H_s^{1+} \tag{3.3.10}$$

co oznaczałoby, że zależność f jest redundancyjna w H_s^1. Otrzymaliśmy więc sprzeczność z założeniami. Zauważmy też, że podobną sprzeczność otrzymamy, jeśli przyjmiemy, że $g \in H_s^1$.

Jeżeli natomiast zależności g nie można wyprowadzić ze zbioru $H_s^1 - \{f\}$, a przy tym $g \in H_s^{1+}$, to znaczy, że f jest potrzebne do wywodu g z H_s^1.

Z lematu 2.1 wynika więc, że spełniona jest zależność

$$L(g) \rightarrow L(f) \in H_s^{1+} = F^+. \tag{3.3.11}$$

Zależności (3.3.7) i (3.3.11) kończą dowód dla przypadku określonego zależnością (3.3.5).

Rozpatrzmy z kolei przypadek określony zależnością (3.3.6).

Ponieważ wszystkie zależności g_i tworzące podzbiór G są niezbędne do wywodu zależności f ze zbioru $H_s^1 - \{f\}$, więc z lematu 2.1 wynika, że dla każdej $g_i \in G$ spełniona jest zależność

$$L(f) \longrightarrow L(g_i) \in (H_s^1 - \{f\} \cup G)^+ = H_s^{1+} = F^+. \qquad (3.3.12)$$

Postawmy jak poprzednio pytanie, czy każdą zależność $g_i \in G$ można wyprowadzić ze zbioru $H_s^1 - \{f\}$.

Rozumując podobnie jak w poprzednim przypadku otrzymujemy sprzeczność (f redundancyjna w H_s^1) dla odpowiedzi twierdzącej.

Odpowiedź przecząca oznacza, że istnieje taka zależność $g_j \in G$, dla której zachodzi

$$g_j \notin (H_s^1 - \{f\})^+. \qquad (3.3.13)$$

Ponieważ zaś każda zależność $g_i \in G$ musi być wyprowadzalna z H_s^1, więc zależność f musi służyć do wywodu g_j ze zbioru H_s^1. Wobec tego z lematu 2.1 wynika, że spełniona jest zależność

$$L(g_j) \longrightarrow L(f) \in H_s^{1+} = F^+. \qquad (3.3.14)$$

Zestawienie zależności (3.3.12) i (3.3.14) kończy dowód. □

Lemat 3.9. W każdych dwóch pokryciach dla potrzeb syntezy H_s^1 i H_s^2 danego zbioru F zależności funkcjonalnych istnieje taka sama liczba klas równoważności kluczy dla bijekcji.

D o w ó d. Z lematu 3.8 wynika, że dla każdej zależności $f \in H_s^1$ istnieje taka zależność $g \in H_s^2$, że $L(f) \longleftrightarrow L(g) \in F^+$. Wynika stąd, że klasie równoważności wyznaczonej dla H_s^1 i zawierającej klucz L(f) odpowiada identyczna bądź alternatywna klasa równoważności wyznaczona dla H_s^2 i zawierająca L(g). Z symetrii lematu 3.8 względem pokryć H_s^1 i H_s^2 wynika odpowiedniość klas równoważności w przeciwnym kierunku. □

TWIERDZENIE 3.6. Dla każdych dwóch pokryć dla potrzeb syntezy H_s^1 i H_s^2 danego zbioru F tworzona jest za pomocą algorytmu syntezy 3.3 ta sama liczba schematów relacji.

D o w ó d. Prawdziwość twierdzenia wynika z lematu 3.9 oraz sformułowania kroku (5) algorytmu syntezy 3.3. □

TWIERDZENIE 3.7. Niech H oznacza pokrycie nieredundancyjne, H_s zaś pokrycie dla potrzeb syntezy danego zbioru F zależności funkcjonalnych. Liczba klas równoważności kluczy dla pokryć H i H_s jest identyczna.

D o w ó d. Należy dowieść, że spełnione są dwie własności:

(a) dla każdej zależności $f \in H$ istnieje taka zależność $g \in H_s$, że $L(f) \longleftrightarrow L(g) \in F^+$ oraz

(b) dla każdej zależności $e \in H_s$ istnieje taka zależność $h \in H$, że

$$L(e) \longleftrightarrow L(h) \in F^+.$$

ad a) Podobnie jak dla lematu 3.8 interesujące są tylko takie zależności $f \in H$, które nie należą do H_s. Ponieważ pokrycie H jest nieredundacyjne, więc musi zachodzić $f \in (H - \{f\})^+$. Daleza część dowodu przebiega podobnie jak dla lematu 3.8.

ad b) Interesujące są tylko takie zależności $e \in H_s$, które nie należą do H. Jeśli e nie jest redundancyjne w H_s, wtedy dowód przebiega jak w lemacie 3.8.

Jeśli e jest redundancyjne w H_s, to z algorytmu tworzenia pokrycia H_s wynika, że e jest jedną z zależności tworzących bijekcje między kluczami równoważnymi a jej lewa strona L(e) nie jest kluczem pozornym. Oznacza to (według lematu 3.1), że zależność $e: X \longrightarrow A_i$ należy do wspólnej grupy zależności $X \rightarrow Z$, $A_i \in Z$, przy czym w grupie tej istnieje co najmniej jedna zależność $e_j: X \longrightarrow A_j$, $A_j \in Z$, nie będąca zależnością redundancyjną.

Dla takiej zależności e_j wykazać można w sposób podobny jak w lemacie 3.8 istnienie w pokryciu H takiej zależności h, że

$$L(h) \longleftrightarrow L(e_j) \in F^+.$$

Ponieważ $L(e) = L(e_j)$, więc z aksjomatu A3 Armstronga wynika, że

$$L(e) \longleftrightarrow L(h) \in F^+,$$

co kończy dowód. □

W pracy [19] przedstawiono twierdzenie, z którego wynika, że liczba schematów relacji utworzonych za pomocą algorytmu 2.3 dla nieredundancyjnego pokrycia H jest minimalna. Wobec tego, zgodnie z twierdzeniem 3.7, algorytm syntezy 3.3 daje w wyniku schemat relacyjnej bazy danych również o minimalnej liczbie relacji.

3.4. UWAGI NA TEMAT REALIZACJI ZMODYFIKOWANEGO ALGORYTMU SYNTEZY I JEGO ZŁOŻONOŚCI OBLICZENIOWEJ

3.4.1. Programowa realizacja algorytmu 3.3

Szczegóły realizacji omawianego algorytmu syntezy przedstawiono w pracach [58, 59, 61]. Podstawową procedurę wykorzystywaną w algorytmie jest procedura badania przynależności do domknięcia zbioru zależności funkcjonalnych.

Prostszą wersję algorytmu realizowanego przez taką procedurę przedstawiono w rozdziale 2.3 (algorytm 2.1). Usuwanie obcych atrybutów, wykorzystujące procedurę badania przynależności do domknięcia, realizowane jest w sposób opisany w rozdziale 2.3. Wyznaczanie kluczy równoważnych opiera się również na wykorzystaniu tej procedury. Dla zmniejszenia nakładu obliczeń domknięcia wszystkich kluczy wyznaczane są przy tym na początku kroku (3) algorytmu 3.3.

Dla oznaczenia klas równoważności kluczy tworzona jest w kroku (3) macierz M powiązań między kluczami. Wiersze i kolumny tej macierzy przyporządkowuje się tylko tym kluczom, które należą do nietrywialnych klas równoważności. Przyporządkowanie to określa wektor W. Kolumna o numerze $\emptyset$ macierzy M podaje zwrotnie numery grup F_i o kluczach odnotowywanych w tej macierzy. Jeżeli danemu kluczowi X_k (z grupy F_k) przyporządkowany jest wiersz "i" macierzy M, to $M[i,0] = k$, a innym kluczom związanym bijekcjami z X_k przyporządkowane są wiersze macierzy M o takich numerach j, dla których $M[i,j] = 1$.

Każdy wiersz macierzy M określa więc klasę równoważności, do której należy klucz związany z danym wierszem.

Szczegółową realizację kroku (4) algorytmu syntezy przedstawiono w algorytmie 3.2.

Do wyznaczania zbioru atrybutów kluczowych dla każdej klasy równoważności wykorzystuje się macierz M.

Końcowy etap tworzenia schematów relacji (krok (5) algorytmu) staje się przy wykorzystaniu macierzy M bardzo prosty. Obejmuje on scalenie grup zależności o kluczach równoważnych.

3.4.2. Uwagi na temat złożoności obliczeniowej nowego algorytmu syntezy

Złożoność obliczeniowa algorytmu Bernsteina (algorytm 2.3) oceniona została w pracy [13] dla poszczególnych kroków następująco:

(1) Usuwanie obcych atrybutów: $O(|F|^2)$.
(2) Tworzenie pokrycia H: $O(n|F|)$.
(3) Podział H na grupy H_i : $O(|F|^2)$.
(4) Wyszukiwanie kluczy równoważnych: $O(n|F|)$.
(5) Powtórne usuwanie zależności redundancyjnych: $O(n|F|)$.
(6) Tworzenie schematów relacji: $O(|F|)$.

Przyjęto przy tym oznaczenia: n - liczba zależności funkcjonalnych w zbiorze F; $|F|$ - sumaryczna liczba występień atrybutów w zapisie zbioru F.

Jako operację dominującą w ocenie złożoności przyjmuje się badanie przynależności do zbiorów bądź inkluzji zbiorów atrybutów.

Pesymistyczna złożoność obliczeniowa całego algorytmu Bernsteina określona jest przez przypadek najgorszy (kroki (1) i (3)), a więc wynosi ona $O(|F|^2)$.

Pesymistyczna złożoność obliczeniowa nowego algorytmu syntezy nie zmieni się w stosunku do algorytmu Bernsteina, bowiem usuwanie obcych atrybutów i podział na grupy zależności powtarzają się w obu algorytmach. Wprowadzone modyfikacje zmniejszają natomiast średni nakład obliczeń potrzebnych do syntezy schematu relacyjnej bazy danych.

Zasadnicza różnica w złożoności obliczeniowej obu algorytmów wynika z faktu, że w nowym algorytmie 3.3 ma miejsce tylko jednokrotne usuwanie zależności redundancyjnych (w kroku (4)), natomiast w algorytmie 2.3 wykonywane jest ono dwukrotnie (krok (2) i (5)).

Trzeba jednak zaznaczyć, że w algorytmie 3.3 wykonuje się dodatkową operację tego typu, a mianowicie usuwanie kluczy pozornych. Jest ono jednak wykonywane tylko dla kluczy z nietrywialnych klas równoważności, a ponadto jest rekompensowane przez pomijanie - w trakcie szukania zależności redundancyjnych - atrybutów kluczowych.

Dla zbilansowania nakładu obliczeń na wymienione operacje przyjmiemy następujące założenia:

- zbiór F liczy n zależności,
- w zbiorze lewych stron (kluczy) wyróżniono p klas równoważności ($p \leq n$),
- liczba nietrywialnych klas równoważności wynosi q ($q \leq p$).

Usuwanie zależności redundancyjnych bazuje na badaniu przynależności do domknięcia danego zbioru zależności funkcjonalnych F. Oznaczmy literą D średnią złożoność obliczeniową operacji wyznaczenia domknięcia. Pesymistyczna złożoność obliczeniowa tej operacji wynosi $O(|F|)$. Złożoność badanych operacji będziemy oceniali krotnością liczby D.

Usuwanie kluczy pozornych

Rozpatrzmy klasę równoważności $K_k = \{X_1^k, X_2^k, \ldots, X_{r_k}^k\}$, gdzie $X_i^k \in K_k$ stanowią lewe strony zależności

$$X_1^k \rightarrow Y_1^k, \ldots, X_{r_k}^k \rightarrow Y_{r_k}^k. \qquad (3.4.1)$$

Wykrycie kluczy pozornych wymaga obliczenia dla każdej zależności domknięcia $(X_i^k)^+$, a następnie sprawdzenie, zgodnie z lematem 3.2, czy $Y_i^k \subseteq (X_i^k)^+$.

Wobec tego zbadanie kluczy pozornych dla wszystkich q nietrywialnych klas równoważności wymaga wykonania $\sum_{k=1}^{q} r_k$ operacji wyznaczania domknięcia, co zapiszemy następująco

$$P = D \sum_{k=1}^{q} r_k. \tag{3.4.2}$$

Usuwanie zależności redundancyjnych

Na wstępie zwróćmy uwagę na dwa przypadki:

(1) Dla zależności o lewych stronach tworzących trywialne klasy równoważności proces usuwania zależności redundancyjnych nie ulega zmianie.

(2) Dla zależności, których lewe strony należą do nietrywialnych klas równoważności usuwanie zależności redundancyjnych odbywa się przy zachowaniu wszelkich zależności tworzących bijekcje. Praktycznie sprowadza się to do usuwania zbędnych atrybutów niekluczowych.

Szczegółowe porównania przeprowadzimy w związku z tym dla przypadku (2). Oznaczmy przez n_j^k liczbę atrybutów tworzących prawą stronę zależności $X_j^k \longrightarrow Y_j^k$, gdzie X_j^k należy do klasy równoważności K_k:

$$n_j^k = \left|Y_j^k\right|. \tag{3.4.3}$$

Oznaczmy również przez m_j^k liczbę atrybutów kluczowych w prawej stronie tej zależności (według oznaczeń z algorytmu 3.2):

$$m_j^k = \left|Y_j^{*k}\right|. \tag{3.4.4}$$

Możliwość wystąpienia atrybutów kluczowych w prawych stronach zależności (3.4.1) wynika z faktu, że jeśli $x_1^k, x_2^k, \ldots, x_{r_k}^k$ są kluczami równoważnymi, to każda z zależności (3.4.1) zawierać musi między innymi powiązania między tymi kluczami. Liczba operacji wyznaczania domknięcia dla przypadku (2) bez rozróżniania atrybutów kluczowych i niekluczowych (co odpowiada krokowi (2) algorytmu Bernsteina) wynosi

$$Q_1 = \sum_{k=1}^{p} \sum_{j=1}^{r_k} n_j^k.$$

W nowym algorytmie przy analizie tylko atrybutów niekluczowych (krok (3) algorytmu 3.2) liczba ta wyniesie

$$Q_2 = \sum_{k=1}^{p} \sum_{j=1}^{r_k} (n_j^k - m_j^k).$$

A więc w nowym algorytmie złożoność operacji usuwania zależności redundancyjnych ulega zmniejszeniu o wartość

$$Q = D \sum_{k=1}^{p} \sum_{j=1}^{r_k} m_j^k. \tag{3.4.5}$$

Zauważmy, że wielkość ta jest porównywalna ze złożonością usuwania kluczy pozornych (3.4.2). Przyjmując bowiem, że w każdej prawej stronie zależności (3.4.1) wystąpi tylko 1 atrybut kluczowy, otrzymamy

$$m_j^k = 1, \quad j = 1, \ldots, r_k$$

$$Q = D \sum_{k=1}^{q} \sum_{j=1}^{r_k} m_j^k = D \sum_{k=1}^{q} r_k. \tag{3.4.6}$$

A więc Q = P, co oznacza, że już w takim przypadku nakład obliczeń dla wykrycia kluczy pozornych został zrekompensowany przez uproszczenie procesu usuwania zależności redundancyjnych.

Podsumowując przedstawione rozważania podkreślmy ponownie, że podstawową oszczędnością, jaką daje nowy algorytm, jest konieczność wykonania tylko jednokrotnej operacji usuwania zależności redundancyjnych. Zmniejsza to średnią złożoność obliczeń o wielkość

$$R = n \cdot D.$$

Wyniki obliczeń dla przykładowych danych programu realizującego opracowany algorytm 3.3 zamieszczono w dodatku.

Czas realizacji programu dla dość dużej liczby danych (48 atrybutów, 64 zależności funkcjonalne) skrócił się dla nowego algorytmu o ok. 1/3 w porównaniu z algorytmem Bernsteina.

4. PROCES SYNTEZY PRZY UWZGLĘDNIENIU ZADAŃ WYSZUKIWANIA DANYCH

Treść niniejszego rozdziału koncentruje się wokół następujących problemów: W jakich przypadkach proces syntezy może prowadzić do różnych rozwiązań dla tych samych danych wejściowych? Jakie kryterium można przyjąć do oceny uzyskiwanych rozwiązań? W jaki sposób winien być rozbudowany algorytm syntezy, aby generował wszystkie możliwe rozwiązania i uwzględniał przyjęte kryterium?

W literaturze brak analizy takich problemów.

4.1. MOŻLIWOŚĆ UZYSKANIA WIELU ROZWIĄZAŃ W PROCESIE SYNTEZY. KRYTERIUM WYBORU ROZWIĄZANIA NAJLEPSZEGO

Możliwość uzyskania różnych postaci schematu relacyjnej bazy danych dla tych samych danych wejściowych jest wynikiem uzależnienia rezultatów tworzenia pokrycia od uporządkowania zbioru zależności funkcjonalnych.

Przykład 4.1.1. W zbiorze zależności funkcjonalnych $F = \{A \to B,\ B \to A,\ AC \to D,\ BC \to D\}$ jedna z zależności $AC \to D$ lub $BC \to D$ jest redundancyjna. Zależnie od kolejności badania tych zależności pokrycie nieredundancyjne (a także pokrycie dla potrzeb syntezy) będzie miało postać $H_1 = \{A \to B,\ B \to A,\ BC \to D\}$ lub $H_2 = \{A \to B,\ B \to A,\ AC \to D\}$. Utworzone zostaną schematy relacji $R_1 = \underline{AB}$, $R_2' = \underline{BC}D$ lub $R_2'' = \underline{AC}D$. □

Przykład 4.1.2. Rozpatrzmy powiązania w następującym zbiorze atrybutów dotyczącym fragmentu struktury organizacyjnej uczelni: $U = \{$wydział, dziekan, instytut, dyrektor, zakład, kierownik$\}$.

Będziemy przyjmowali, że:

- dana osoba nie może kierować równocześnie dwoma wydziałami, instytutami lub zakładami,
- w ramach uczelni nie powtarzają się nazwy wydziałów, instytutów i zakładów,
- wartościami atrybutów dziekan, dyrektor i kierownik są nazwiska, imiona i tytuły, przy czym wartości takich trójek nie powtarzają się wśród kadry kierowniczej.

Możliwe jest osłabienie powyższych założeń przy niewielkiej rozbudowie bazy danych. Nie zmieniłoby to toku poniższych rozważań.

Wszystkie możliwe zależności funkcjonalne wiążące atrybuty ze zbioru U tworzą następujący zbiór:

$F = \{$wydział $\to$ dziekan,
dziekan $\to$ wydział,
instytut $\to$ wydział,
instytut $\to$ dziekan,
instytut $\to$ dyrektor,
dyrektor $\to$ instytut,
dyrektor $\to$ wydział,
dyrektor $\to$ dziekan,
zakład $\to$ wydział,
zakład $\to$ dziekan,
zakład $\to$ instytut,
zakład $\to$ dyrektor,
zakład $\to$ kierownik,
kierownik $\to$ zakład,
kierownik $\to$ wydział,
kierownik $\to$ dziekan,
kierownik $\to$ instytut,
kierownik $\to$ dyrektor$\}$.

Większość spośród tych zależności to oczywiście zależności redundancyjne. Dla zbioru F można utworzyć 16 różnych pokryć nieredundancyjnych (lub pokryć dla potrzeb syntezy). We wszystkich tych pokryciach występują zależności:

wydział $\to$ dziekan,
dziekan $\to$ wydział,
instytut $\to$ dyrektor,
dyrektor $\to$ instytut,
zakład $\to$ kierownik,
kierownik $\to$ zakład.

Do zależności tych dodać należy jedną zależność ze zbioru

instytut $\to$ wydział,
instytut $\to$ dziekan,
dyrektor $\to$ wydział,
dyrektor $\to$ dziekan,

oraz jedną zależność ze zbioru

zakład $\to$ instytut,
zakład $\to$ dyrektor,
kierownik $\to$ instytut,
kierownik $\to$ dyrektor.

Z uwagi na istnienie w zbiorze F następujących klas równoważności kluczy

$K_1 = \{$wydział, dziekan$\}$,
$K_2 = \{$instytut, dyrektor$\}$,
$K_3 = \{$zakład, kierownik$\}$

pokrycia te dają następujące warianty schematów relacji:

$R_1 = \{$wydział, dziekan$\}$,
$R'_2 = \{$instytut, dyrektor, wydział$\}$,
$R''_2 = \{$instytut, dyrektor, dziekan$\}$,
$R'_3 = \{$zakład, kierownik, instytut$\}$,
$R''_3 = \{$zakład, kierownik, dyrektor$\}$.

Odpowiednie kombinacje schematów relacji prowadzą do czterech różnych schematów relacyjnej bazy danych:

$$\mathbb{R}_1 = \{R_1, R'_2, R'_3\},\ \mathbb{R}_2 = \{R_1, R''_2, R''_3\},\ \mathbb{R}_3 = \{R_1, R''_2, R'_3\},$$
$$\mathbb{R}_4 = \{R_1, R'_2, R''_3\}.\ \square$$

Zauważmy, że każdy z przedstawionych schematów relacji jest w trzeciej postaci normalnej. Ponadto liczba schematów oraz liczba atrybutów w schematach są we wszystkich wariantach schematu relacyjnej bazy danych takie same. Wobec tego dla każdego schematu osiągnięty został w jednakowy sposób cel procesu syntezy.

Postawmy teraz pytanie: Czy każdy z przedstawionych wariantów schematu będzie jednakowo przydatny dla użytkownika? Odpowiedź będzie zależna od zadań, które użytkownik zamierza realizować w rozważanej bazie danych.

Przykład 4.1.3. Załóżmy, że opierając się na schematach $\mathbb{R}_1 = \{R_1, R'_2, R'_3\}$ i $\mathbb{R}_2 = \{R_1, R''_2, R''_3\}$ z przykładu 4.1.2 utworzone zostały dwie bazy danych $r_1(\mathbb{R}_1)$ i $r_2(\mathbb{R}_2)$ zawierające relacje odpowiednio $r_1(R_1)$, $r'_2(R'_2)$, $r'_3(R'_3)$, $r''_2(R''_2)$, $r''_3(R''_3)$. Dla obu baz danych należy rozwiązać następujące zadania wyszukiwania danych:

1. Podaj wykaz instytutów na wydziale W.
2. Podaj wykaz zakładów na wydziale W.
3. Wskaż wydział i instytut, w skład których wchodzi zakład Y.
4. Podaj nazwisko dyrektora takiego instytutu, w skład którego wchodzi zakład Y.
5. Podaj wykaz par osób związanych zależnością służbową: (dziekan, dyrektor) i (dyrektor, kierownik) na wydziale, którego dziekanem jest N.

R o z w i ą z a n i a. Rozwiązania przedstawione zostaną w kategoriach algebry relacji. Wynik prezentuje zawartość utworzonej relacji z.

Warianty "a" i "b" rozwiązań będą dotyczyły odpowiednio baz danych $r_1(\mathbb{R}_1)$ i $r_2(\mathbb{R}_2)$.

$$z_1^a = \pi_{\text{instytut}}(\sigma_{\text{wydział=W}}(r'_2))$$
$$z_1^b = \pi_{\text{instytut}}((\sigma_{\text{wydział=W}}(r_1)) \bowtie r''_2)$$
$$z_2^a = \pi_{\text{zakład}}((\sigma_{\text{wydział=W}}(r'_2)) \bowtie r'_3)$$
$$z_2^b = \pi_{\text{zakład}}(((\sigma_{\text{wydział=W}}(r_1) \bowtie r''_2) \bowtie r''_3)$$
$$z_3^a = \pi_{\text{wydział,instytut}}((\sigma_{\text{zakład=Y}}(r'_3)) \bowtie r'_2)$$
$$z_3^b = \pi_{\text{wydział,instytut}}(((\sigma_{\text{zakład=Y}}(r''_3)) \bowtie r''_2) \bowtie r_1)$$
$$z_4^a = \pi_{\text{dyrektor}}((\sigma_{\text{zakład=Y}}(r'_3)) \bowtie r'_2)$$
$$z_4^b = \pi_{\text{dyrektor}}(\sigma_{\text{zakład=Y}}(r''_3))$$
$$z_5^a = \pi_{\text{dziekan,dyrektor}}((\sigma_{\text{dziekan=N}}(r_1)) \bowtie r'_2) \cup \pi_{\text{dyrektor,kierownik}}(((\sigma_{\text{dziekan=N}}(r_1)) \bowtie r'_2) \bowtie r'_3)$$
$$z_5^b = \pi_{\text{dziekan,dyrektor}}(\sigma_{\text{dziekan=N}}(r''_2)) \cup \pi_{\text{dyrektor,kierownik}}(((\sigma_{\text{dziekan=N}}(r''_2)) \bowtie r''_3).\ \square$$

Porównując poszczególne rozwiązania zwróćmy uwagę na to, że różnią się one przede wszystkim liczbą połączeń. W zadaniach 1, 2, 3 liczba ta jest mniejsza dla bazy danych o schemacie $\mathbb{R}_1$, w pozostałych - dla $\mathbb{R}_2$. Pozostałe różnice dotyczą sposobu wykonywania selekcji i ich ocena zależy od aktualnych rozmiarów relacji. Można oczywiście przytoczyć szereg dalszych zadań, dla których jedna z baz daje korzystniejsze rozwiązania. Wiarygodną ocenę obu schematów mógłby przeprowadzić administrator (użytkownik) bazy danych, który winien uwzględnić w tym celu:

- wykaz typowych zadań $Z = Z_1, Z_2, \ldots, Z_n$ realizowanych (a właściwie przewidywanych do realizacji) w bazie,
- częstotliwość realizacji poszczególnych zadań.

Dla dalszych rozważań w tym rozdziale przyjmiemy, że sformułowanie powyższych kryteriów będzie możliwe na etapie projektowania schematu relacyjnej bazy danych.

Jako podstawową wielkość służącą do oceny własności schematu przyjmiemy liczbę połączeń relacji wykonywanych przy rozwiązywaniu przedstawionych zadań.

Uzasadniają to dwa fakty:

(1) Przedstawione przykłady pokazują, że wykorzystanie alternatywnych schematów relacji prowadzi przede wszystkim do różnej liczby połączeń relacji.

(2) Złożoność operacji połączenia jest istotnie większa od złożoności pozostałych operacji algebry relacji.

Złożoność połączenia dwóch relacji liczących m i n krotek wynosi bowiem w ogólnym przypadku O(m·n), natomiast złożoność selekcji i rzutowania O(m) lub O(n).

Uwzględnienie częstotliwości realizacji poszczególnych zadań możliwe jest na przykład przez przypisanie każdemu z zadań indywidualnej wagi γ (z przedziału $[\gamma_{min}, \gamma_{max}]$) wprost proporcjonalnej do częstotliwości. Końcową wielkość kryterialną będzie wtedy określało wyrażenie:

$$k = \sum_{i=1}^{n_z} \gamma_i p_i \qquad (4.1.1)$$

gdzie:

n_z - liczba zadań,

p_i - liczba połączeń wykonywanych w trakcie realizacji zadania i,

γ_i - waga przypisana zadaniu i.

Najlepszy schemat relacyjnej bazy danych będzie się charakteryzował najmniejszą wartością wskaźnika k w danym zbiorze rozwiązań.

Analizując sposób rozwiązywania zadań z przykładu 4.1.3 zwróćmy uwagę na to, że wybór relacji potrzebnych do rozwiązania wynikał - przy określonej strukturze schematu - ze zbioru atrybutów, które występowały w sformułowaniu zadania.

Wniosek ten jest istotny przy analizie automatycznego procesu rozwiązywania zadań, przedstawionego w podrozdziale 4.5.

Tak więc przyjmować będziemy, że każde z zadań scharakteryzowane jest przez zbiór atrybutów X występujących w sformułowaniu zadania. Dla przykładu 4.1.3 mielibyśmy:

$X_1 = \{$instytut, wydział$\}$,
$X_2 = \{$zakład, wydział$\}$,
$X_3 = \{$wydział, instytut, zakład$\}$,
$X_4 = \{$dyrektor, zakład$\}$,
$X'_5 = \{$dziekan, dyrektor$\}$, $X''_5 = \{$dyrektor, kierownik$\}$.

Na zakończenie niniejszego podrozdziału zwrócimy jeszcze uwagę na specyfikę zadania 5 w przykładzie 4.1.3, w którym rozwiązanie jest sumą dwóch zbiorów. W dalszych rozważaniach (w podrozdziale 4.5) ograniczymy się do takiej klasy zadań, dla której - w przypadku podania kilku warunków $W_1, W_2, \ldots, W_n$, które spełniać winien wynik - warunki te powiązane są operatorem iloczynu logicznego $W_1 \cap W_2 \cap \ldots \cap W_n$ (ang. conjunctive queries [70, 86, 3, 4]). Zadania nie należące do tej klasy zdekomponować można do sumy

zadań spełniających podane ograniczenia. Na przykład zadanie 5 przedstawić można jako sumę dwóch zadań:

5a. Podaj wykaz par (dziekan, dyrektor) dla wydziału, którego dziekanem jest N.

5b. Podaj wykaz par (dyrektor, kierownik) dla wydziału, którego dziekanem jest N.

Ograniczenie powyższe jest typowym założeniem upraszczającym analizę zadań wyszukiwania danych, zwłaszcza w przypadku automatycznego rozwiązywania zadań.

Przedstawiona w tym podrozdziale analiza pozwala rozszerzyć założenia procesu syntezy przedstawione w rozdziale 2.2. Przed algorytmem syntezy postawimy mianowicie takie wymaganie, aby w przypadku istnienia wielu rozwiązań dotychczas równorzędnych umożliwiał on wyszukanie rozwiązania najlepszego według kryterium 4.1.1.

Nowe założenia wymagają opracowania w trakcie rozbudowy algorytmu syntezy dwóch głównych problemów

(1) metody utworzenia wszystkich możliwych schematów relacyjnej bazy danych,

(2) metody automatycznego rozwiązywania wszystkich zadań wyszukiwania danych dla każdego schematu relacji.

Analizując wstępnie pierwszy problem zauważmy, że wyniki przedstawionego w rozdziale 3 procesu tworzenia pokrycia zależą od uporządkowania zbioru zależności funkcjonalnych. Jednakże bezpośrednie wykorzystanie tego spostrzeżenia w prostym algorytmie wyznaczającym pokrycia dla wszystkich permutacji zależności funkcjonalnych w wejściowym zbiorze F prowadziłoby do niedopuszczalnej złożoności obliczeniowej algorytmu. Tak więc istnieje konieczność poszukiwania takich własności zależności funkcjonalnych, które uprościłyby proces syntezy wszystkich postaci schematów. Badania te przedstawiono w podrozdziałach 4.2, 4.3 i 4.4.

Drugi problem - dotyczący automatycznego rozwiązywania zadań wyszukiwania danych - wymaga znacznego rozszerzenia stosowanych dotąd w tej pracy metod opisu badanej dziedziny. Zagadnienia te - bazujące na pojęciu hipergrafów i ich własnościach - przedstawiono w podrozdziale 4.5.

4.2. ALTERNATYWNIE REDUNDANCYJNE ZALEŻNOŚCI FUNKCJONALNE I ICH WŁASNOŚCI

Wskazana w podrozdziale 4.1 możliwość uzyskania w procesie syntezy różnych postaci pokrycia związana jest ze specyficznymi własnościami niektórych zależności funkcjonalnych należących do wejściowego zbioru F. Zależnościom tym - zdefiniowanym poniżej - nadano nazwę alternatywnie redundancyjnych zależności funkcjonalnych. W niniejszym podrozdziale przed-

stawiono analizę cech tych zależności. Pozwoli to na właściwą organizację procesu wyznaczania wszystkich postaci pokrycia.

Niech H będzie nieredundancyjnym pokryciem danego zbioru F zależności funkcjonalnych.

Zależność g: Y⟶B taką, że g∈F, g∉H, będziemy nazywali <u>alternatywnie redundancyjną zależnością funkcjonalną (ARZF)</u> do pewnej zależności f: X⟶A∈H, jeśli dołączenie zależności g do H spowoduje, że zależność f stanie się redundancyjna w H∪{g}, natomiast zbiór H - {f}∪{g} będzie nowym nieredundancyjnym pokryciem zbioru F.

<u>Przykład 4.2.1</u>. Rozpatrzmy następujący zbiór zależności funkcjonalnych:

F = {wydział ⟶ dziekan, dziekan ⟶ wydział,
instytut ⟶ dyrektor, dyrektor ⟶ instytut,
instytut ⟶ wydział, dyrektor ⟶ wydział,
instytut ⟶ dziekan, dyrektor ⟶ dziekan}.

Jedno z nieredundancyjnych pokryć zbioru F ma postać:

H = {wydział ⟶ dziekan, dziekan ⟶ wydział,
instytut ⟶ dyrektor, dyrektor ⟶ instytut,
instytut ⟶ wydział}.

Każda z następujących zależności

dyrektor ⟶ wydział,
instytut ⟶ dziekan,
dyrektor ⟶ dziekan

jest alternatywnie redundancyjna do zależności

instytut ⟶ wydział.

TWIERDZENIE 4.1. Własność alternatywnej redundancyjności zależności funkcjonalnych jest relacją równoważności.

D o w ó d (1) <u>Zwrotność</u>. Dowód jest oczywisty, bowiem sprawdzenie zwrotności sprowadza się do zastąpienia zależności f w nieredundancyjnym pokryciu przez nią samą.

(2) <u>Symetria</u>. Zakładając jak w definicji, że g jest ARZF do f mamy:

$$g \in F,\ g \notin H,\ f \in H,\ f \notin (H - \{f\})^+,\ f \in (H - \{f\} \cup \{g\})^+. \qquad (4.2.1)$$

Oprócz tego zbiór G = H - {f}∪{g} jest pokryciem nieredundancyjnym i zachodzi $G^+ = H^+$.

Zauważmy, że f∉G i f∈F. Ponadto dołączenie f do G daje zbiór G∪{f} = H∪{g}, w którym zależność g jest redundancyjna, natomiast zbiór G∪{f} - {g} równy H jest pokryciem nieredundancyjnym zbioru F. Spełnione są więc wszystkie warunki określające zależność f jako ARZF do zależności g.

(3) <u>Przechodniość</u>. Sprecyzujemy najpierw szczegółowe założenia i tezę dla dowodu tej własności.

Załóżmy, że do zbioru F należę między innymi trzy zależności funkcjonalne f: X⟶A, g: Y⟶B oraz h: Z⟶C.

Przyjmijmy, że zbiór H_1 = E∪{f} jest nieredundancyjnym pokryciem zbioru F. Załóżmy, że zależność g∈F, g∉H_1 jest ARZF do f. Zastępując f przez g otrzymamy więc inne nieredundancyjne pokrycie $H_2 = H_1 - \{f\} \cup \{g\}$ = E∪{g} zbioru F. Załóżmy dalej, że zależność h∈F, h∉E, różna od f i g, jest ARZF do g. Zamiana g na h daje więc nowe nieredundancyjne pokrycie

$$H_3 = H_2 - \{g\} \cup \{h\} = E \cup \{h\} \text{ zbioru F.}$$

Należy wykazać, że h jest ARZF do f, to znaczy dołączenie do H_1 zależności h spowoduje, że f stanie się redundancyjna w $H_1 \cup \{h\}$, a zbiór $H_1 - \{f\} \cup \{h\}$ będzie pokryciem nieredundancyjnym zbioru F.

Zauważmy, że $H_1 - \{f\} \cup \{h\} = E \cup \{h\} = H_3$, a H_3 z założenia jest nieredundancyjnym pokryciem F.

Ponadto f∈H_3^+ (ponieważ $H_3^+ = H_1^+ = F^+$), to znaczy $f \in (H_1 - \{f\} \cup \{g\})^+$, co dowodzi redundancyjności zależności f w zbiorze $H_1 \cup \{h\}$. □

<u>Wniosek 4.2.1</u>. Rozpatrzmy pewien zbiór zależności funkcjonalnych F i jego nieredundancyjne pokrycie H. Oznaczmy przez RED zbiór zależności redundancyjnych w H: RED = F - H. Oznaczmy przez RED_1 podzbiór zbioru RED zawierający wszystkie zależności alternatywnie redundancyjne do zależności należących do pokrycia H.

Zauważmy, że każda zależność ze zbioru $F_1 = H \cup RED_1$ ($F_1 \subseteq F$) może zostać wybrana do nieredundancyjnego pokrycia zbioru F.

<u>Własność alternatywnej redundancyjności dzieli</u> - jako relacja równoważności - <u>zbiór</u> F_1 na <u>klasy równoważności</u> (abstrakcji).

Podział ten możemy oznaczyć następująco

$$F_1 = AR_1 \cup AR_2 \cup \ldots \cup AR_k, \qquad (4.4.2)$$

gdzie wszystkie podzbiory AR_i, tworzące odrębne klasy równoważności, są rozłączne.

Klasy równoważności zawierające tylko jedną zależność będziemy nazywali trywialnymi. W dalszym ciągu będą nas interesować przede wszystkim <u>nietrywialne klasy równoważności</u> zależności alternatywnie redundancyjnych. Dla uproszczenia będziemy więc pomijać przymiotnik "nietrywialne".

<u>Przykład 4.2.2</u>. W zbiorze zależności funkcjonalnych F opisujących w przykładzie 4.1.2 fragment struktury organizacyjnej uczelni występują dwie (nietrywialne) klasy równoważności zależności alternatywnie redundancyjnych:

$AR_1 = \{$instytut $\longrightarrow$ wydział,
instytut $\longrightarrow$ dziekan,
dyrektor $\longrightarrow$ wydział,
dyrektor $\longrightarrow$ dziekan$\}$,

$AR_2 = \{$zakład $\longrightarrow$ instytut,
zakład $\longrightarrow$ dyrektor,
kierownik $\longrightarrow$ instytut,
kierownik $\longrightarrow$ dyrektor$\}$. □

Lemat 4.1. Oznaczmy jak we wniosku 4.2.1, przez

$$AR_m = \{f_1^m, f_2^m, \ldots, f_p^m\}$$

jedną z klas równoważności alternatywnie redundancyjnych zależności funkcjonalnych dla zbioru F i jego pokrycia H. Jedna z zależności $f_1^m, \ldots, f_p^m$ musi, zgodnie z definicją, należeć do pokrycia; niech np. $f_i^m \in H$. Przyjmijmy oznaczenie $H_1 = H - \{f_i^m\}$.

Postać zbioru H_1 nie zależy od tego, która z zależności $f_i^m \in AR_m$ została wybrana do nieredundancyjnego pokrycia.

D o w ó d. Załóżmy dla dowodu nie wprost, że pewna zależność $g \in F$, $g \notin H$, jest redundancyjna w zbiorze $H \cup \{g\}$, nie jest natomiast redundancyjna w zbiorze $H - \{f_i^m\} \cup \{f_j^m\} \cup \{g\}$, gdzie $f_j^m \in AR_m$. Jednakże z definicji ARZF wynika, że zbiór $H - \{f_i^m\} \cup \{f_j^m\}$ jest nieredundancyjnym pokryciem zbioru F. Zachodzi więc

$$(H - \{f_i^m\} \cup \{f_j^m\})^+ = F^+ = H^+.$$

Ponieważ $g \in F$, więc $g \in H^+$, a dalej $g \in (H - \{f_i^m\} \cup \{f_j^m\})^+$, co daje sprzeczność z poprzednim założeniem. □

Wniosek 4.2.2. Przyjmijmy jak we wniosku 4.2.1, że F_1 oznacza zbiór zależności alternatywnie redundancyjnych zbioru F, podzielonych na klasy równoważności następująco

$$F_1 = AR_1 \cup AR_2 \cup \ldots \cup AR_k .$$

Z każdej klasy AR_i, $i = 1, \ldots, k$, w pokryciu H powinna wystąpić jedna zależność f_j^i. Jak wynika z lematu 4.1, zależność ta może być dowolnie wybrana z AR_i. Wobec tego liczba wszystkich możliwych postaci pokrycia wynosi

$$L_p = \prod_{i=1}^{k} |AR_i| , \qquad (4.2.3)$$

gdzie $|AR_i|$ oznacza liczbę zależności w klasie AR_i. □

Przykład 4.2.3. W przykładach 4.1.2 i 4.2.2 występują dwie klasy równoważności ARZF takie, że $|AR_1| = |AR_2| = 4$. Liczba możliwych postaci pokrycia wynosi więc $L_p = 16$.

TWIERDZENIE 4.2. Alternatywnie redundancyjne zależności funkcjonalne w zbiorze F, tworzące wspólną klasę równoważności, mają lewe strony związane bijakcją w F^+. To znaczy, jeśli $g: Y \longrightarrow B \in F$ jest ARZF do $f: X \longrightarrow A \in F$, to $L(f) \longleftrightarrow L(g) \in F^+$.

D o w ó d. Załóżmy, że H_1 jest nieredundacyjnym pokryciem zbioru F takim, że $f \in H_1$, $g \notin H_1$. Z definicji ARZF wynika, że $f \notin (H_1 - \{f\})^+$, ale $f \in (H_1 - \{f\} \cup \{g\})^+$.

Oznacza to, że do wyprowadzenia zależności f ze zbioru $H_1 - \{f\} \cup \{g\}$ musi zostać użyta zależność g.

Z lematu 2.1 wynika wtedy, że

$$X \longrightarrow Y \in H_1^+ . \qquad (4.2.4)$$

Rozpatrzmy następnie inne nieredundancyjne pokrycie zbioru F w postaci $H_2 = H_1 - \{f\} \cup \{g\}$, utworzone zgodnie z definicją ARZF. Obecnie zachodzi $f \notin H_2$, $g \in H_2$.

Powtarzając poprzednie rozumowanie otrzymamy

$$Y \longrightarrow X \in H_2^+ . \qquad (4.2.5)$$

Ponieważ $H_1^+ = H_2^+ = F^+$, więc zestawienie zależności (4.2.4) i (4.2.5) kończy dowód. □

Zauważmy, że rozpatrywane poprzednio różne postacie pokrycia prowadziły w większości przypadków do tych samych postaci schematów relacji. Przedstawione twierdzenie wyjaśnia ten fakt, wynika z niego bowiem, że zastąpienie zależności $f: X \longrightarrow A$ alternatywnie redundancyjną zależnością $g: Y \longrightarrow A$ daje wprawdzie różne pokrycia, ale prowadzi do tych samych schematów relacji (ponieważ X i Y są kluczami równoważnymi).

Wniosek 4.2.3. Załóżmy, że w danym zbiorze F istnieje k różnych klas równoważności ARZF: $AR_1, \ldots, AR_k$, a w klasie AR_i (liczącej $|AR_i|$ zależności) występuje l_i różnych atrybutów ($l_i \leq |AR_i|$) wśród prawych stron zależności tworzących AR_i. Wtedy liczba wszystkich różnych postaci schematów relacyjnej bazy danych, jakie utworzyć można dla zbioru F, wynosi

$$L_s = \prod_{i=1}^{k} l_i . \quad \square \qquad (4.2.6)$$

Przykład 4.2.4. Dla klas równoważności ARZF z przykładu 4.2.2 (patrz też przykład 4.2.3) mamy $l_1 = l_2 = 2$. Wobec tego liczba różnych schematów relacyjnej bazy danych, jakie utworzyć można dla zbioru F z przykładu 4.1.2, wynosi $L_s = 4$. □

Rozpatrzmy możliwość powiązania koncepcji pokrycia dla potrzeb syntezy z pojęciem alternatywnie redundancyjnych zależności funkcjonalnych. Zauważmy, że w procesie wyznaczania pokrycia H_s, po usunięciu kluczy pozornych, zależności redundancyjnych szuka się tylko wśród zależności nie tworzących bijekcji między kluczami równoważnymi.

Przyjmijmy w związku z tym następujący rozłączny podział pokrycia dla potrzeb syntezy:

$$H_s = H_b \cup H_p$$

gdzie:

H_b - obejmuje zależności tworzące bijekcje,

H_p - obejmuje zależności pozostałe.

Wymiana zależności alternatywnie redundancyjnych może więc dotyczyć tylko podzbioru H_p. Zbiór zależności alternatywnie redundancyjnych może więc ulec zmniejszaniu o zależności wiążące klucze z danej klasy równoważności. Poza tym pojęcie pokrycia dla potrzeb syntezy w niczym nie ogranicza przedstawionych własności ARZF.

TWIERDZENIE 4.3. Załóżmy, że w zbiorze F zależności funkcjonalnych wystąpią dwie zależności $f: X \longrightarrow A$ oraz $g: Y \longrightarrow B$ takie, że $X \longleftrightarrow Y \in F^+$, $A \longleftrightarrow B \in F^+$, ale pary atrybutów (X,Y) i (A,B) należą do różnych klas równoważności kluczy, np. $X,Y \in K_i$; $A,B \in K_j$, $K_i \neq K_j$. Jeśli jedna z tych zależności znalazła się w pokryciu dla potrzeb syntezy H_s zbioru F, to f i g są ARZF.

D o w ó d. Załóżmy, że $f \in H_s$. Ponieważ X i A są kluczami należącymi do różnych klas równoważności kluczy, więc dodatkowo wiadomo, że f musi należeć do H_p (gdzie $H_s = H_b \cup H_p$).

Zauważmy, że zależność g nie może należeć do H_s razem z f, ponieważ z obecnych w H_s zależności $Y \longrightarrow X$, $X \longrightarrow A$, $A \longrightarrow B$ można wyprowadzić g.

Należy dowieść, że:

(1) $f \in (H_s - \{f\} \cup \{g\})^+$, a więc f jest redundancyjna w $H_s \cup \{g\}$ oraz

(2) $(H_s - \{f\} \cup \{g\})^+ = H_s^+$, a przy tym w zbiorze $H_p - \{f\} \cup \{g\}$ brak zależności redundancyjnych.

ad (1). Zauważmy, że do zbioru $H_s - \{f\} \cup \{g\}$ należą zależności: $X \longrightarrow Y$ i $B \longrightarrow A$ (są to bowiem zależności łączące klucze równoważne i ich obecność jest wymuszana w H_b) oraz $g: Y \longrightarrow B$. Stosując aksjomat A3

możemy więc wyprowadzić z tego zbioru zależność $f: X \longrightarrow A$, co kończy dowód własności (1).

ad (2). Przy spełnieniu własności (1) z lematu 3.5 wynika, że

$$(H_s - \{f\} \cup \{g\})^+ = (H_s \cup \{g\})^+. \qquad (4.2.7)$$

Ponieważ $g \in F$, więc $g \in H_s^+$. Wobec tego z lematu 3.5 wynika, że

$$H_s^+ = (H_s \cup \{g\})^+. \qquad (4.2.8)$$

Zestawienie zależności (4.2.7) i (4.2.8) daje poszukiwaną równość

$$(H_s - \{f\} \cup \{g\})^+ = H_s^+.$$

Załóżmy następnie dla dowodu nie wprost, że w pokryciu H_s po zastąpieniu zależności f zależnością g ujawniła się zależność redundancyjna h różna od g i f. Zależność h musiałaby przy tym należeć do zbioru $H_p = \{f\} \cup \{g\}$, ponieważ zbiór H_b zawiera tylko bijekcje. Powyższe założenie redundancyjności możemy zapisać w postaci dwóch warunków

$$h \notin (H_s - \{h\})^+, \quad h \in (H_s - \{f\} \cup \{g\} - \{h\})^+. \qquad (4.2.9)$$

Zauważmy, że ze zbioru $H_s - \{h\}$ wyprowadzić można zależność g, ponieważ do zbioru tego należą zależności $Y \longrightarrow X$, $f: X \longrightarrow A$, $A \longrightarrow B$. Na mocy lematu 3.5 możemy więc zapisać

$$(H_s - \{h\})^+ = (H_s - \{h\} \cup \{g\})^+. \qquad (4.2.10)$$

Podobnie wykażemy, że f jest wyprowadzalna ze zbioru $H_s - \{f\} \cup \{g\} - \{h\}$. Do zbioru tego należą bowiem zależności $X \longrightarrow Y$, $g: Y \longrightarrow B$, $B \longrightarrow A$.

Na mocy lematu 3.5 mamy więc

$$(H_s - \{f\} \cup \{g\} - \{h\})^+ = (H_s \cup \{g\} - \{h\})^+. \qquad (4.2.11)$$

Związki (4.2.10) i (4.2.11) pokazują równość domknięć

$$(H_s - \{h\})^+ = (H_s - \{f\} \cup \{g\} - \{h\})^+.$$

co daje sprzeczność w warunkach (4.2.9). □

We wszystkich dotychczasowych przykładach prawe strony alternatywnie redundancyjnych zależności funkcjonalnych były kluczami równoważnymi. Były to więc zależności tego rodzaju, jakie przedstawione zostały w twierdzeniu 4.3. Zależności takie łatwo można wyszukiwać w zbiorze F.

Niestety poniższy przykład pokazuje, że mogą istnieć zależności alternatywnie redundancyjne, których prawe strony nie są kluczami równoważnymi.

Przykład 4.2.5. Rozpatrzmy następujący zbiór zależności funkcjonalnych: $F = \{X \to ATY,\ Y \to BRX,\ BR \to A,\ AT \to B\}$. W zbiorze tym redundancyjna jest zależność $X \to A$ lub $Y \to B$, obie zależności są więc względem siebie ARZF. Ze zbioru F nie można jednak wyprowadzić związku $A \to B$ ani $B \to A$. □

Twierdzenia 4.1, 4.2 i 4.3 pozwalają w istotny sposób uprościć proces poszukiwania zależności alternatywnie redundancyjnych. Rozwiązanie tego zagadnienia przedstawione zostenie w rozdziale 3.3.

4.3. ALTERNATYWNIE REDUNDANCYJNE ZBIORY ZALEŻNOŚCI FUNKCJONALNYCH I ICH WŁASNOŚCI

Rozpatrzmy pewien zbiór zależności funkcjonalnych F i dwa - niekoniecznie rozłączne - podzbiory $F: E = \{e_1, e_2, \ldots, e_n\}$, gdzie $e_i : X_i \to A_i$, $i = 1,\ldots,n$ oraz $G = \{g_1, g_2, \ldots, g_m\}$, gdzie $g_i : Y_i \to B_i$, $i = 1,\ldots,m$. Niech H będzie nieredundancyjnym pokryciem zbioru F. Zbiór $G \subset F$, $G \notin H$, będziemy nazywali alternatywnie redundancyjnym zbiorem zależności funkcjonalnych (ARZZF) do zbioru $E \subseteq H$, jeśli zbiór $H - E \cup G$ jest nieredundancyjnym pokryciem zbioru F, to znaczy każda z zależności $e_i \in (E-G)$ jest redundancyjna w zbiorze $H \cup G$, a przy tym żaden podzbiór $G' \subset (G-E)$ nie posiada przedstawionej własności zbioru G w stosunku do żadnego podzbioru $E' \subseteq (E-G)$, czyli cały podzbiór E-G może być zastąpiony w pokryciu H tylko całym podzbiorem G-E. □

Przykład 4.3.1. Rozpatrzmy następujący zbiór zależności funkcjonalnych

$$F = \{X \to ABCDPQ,\ ABC \to PQ,\ PQ \to ABD,\ ABD \to C\}.$$

Można utworzyć następujące pokrycia tego zbioru

$H_1 = \{X \to ABC,\ ABC \to PQ,\ PQ \to ABD,\ ABD \to C\}$,

$H_2 = \{X \to PQ,\ ABC \to PQ,\ PQ \to ABD,\ ABD \to C\}$,

$H_3 = \{X \to ABD,\ ABC \to PQ,\ PQ \to ABD,\ ABD \to C\}$.

Wynika stąd, że zbiór $G = \{X \to P,\ X \to Q\}$ jest ARZZF do zbioru $E = \{X \to A,\ X \to C\}$.

Podobnie zbiór $I = \{X \to A,\ X \to B,\ X \to D\}$ jest ARZZF do zbioru E, jak również do zbioru G.

Zwróćmy na marginesie uwagę na to, że alternatywnie redundancyjne zbiory E oraz I nie są rozłączne. □

Wniosek 4.3.1. Alternatywnie redundancyjne zależności funkcjonalne stanowią szczególny przypadek alternatywnie redundancyjnych zbiorów zależności funkcjonalnych.

TWIERDZENIE 4.4. Własność alternatywnej redundancyjności zbiorów zależności funkcjonalnych jest relacją równoważności.

D o w ó d (1) Zwrotność. Sprawdzenie zwrotności sprowadza się do zastąpienia zbioru E w pokryciu tym samym zbiorem.

(2) Symetria. Zakładając jak w definicji, że zbiór $G = \{g_1, g_2, \ldots, g_m\}$ jest ARZZF do zbioru $E = \{e_1, \ldots, e_n\}$, przy czym E zawiera się w nieredundancyjnym pokryciu H_1 zbioru F, mamy dowieść, że również E jest ARZZF do G.

Rozpatrzmy zbiór $H_2 = H_1 - E \cup G$. Zgodnie z definicją ARZZF zbiór ten jest nieredundancyjnym pokryciem zbioru F. Zauważmy, że dołączenie do pokrycia H_2 zbioru E daje zbiór $H_2 \cup E = H_1 \cup G$, w którym każda zależność $g_i \in (G-E)$ jest redundancyjna. Natomiast zbiór $H_2 - G \cup E$ jest pokryciem nieredundancyjnym zbioru F, ponieważ $H_2 - G \cup E = H_1$.

Rozpatrzmy następnie pytanie, czy dołączenie do zbioru H_2 pewnego podzbioru $E' \subset (E-G)$ może spowodować, że pewna zależność $g_j \in (G-E)$ stanie się redundancyjna w zbiorze $H_2 \cup E'$, a zbiór $H_2 - \{g_j\} \cup E'$ będzie pokryciem nieredundancyjnym.

Przyjmijmy dla dowodu nie wprost, że zbiór $H_2 - \{g_j\} \cup E'$ jest pokryciem nieredundancyjnym.

Zauważmy, że $H_2 - \{g_j\} \cup E' = H_1 - (E - E') \cup (G - \{g_j\})$. Związek ten można zinterpretować następująco: dołączenie do nieredundancyjnego pokrycia H_1 (zawierającego zbiór E) zbioru $G' = G - \{g_j\}$ spowoduje redundancyjność zależności tworzących zbiór $E-E'$. Jest to sprzeczne z definicją ARZZF.

(3) Przechodniość. Załóżmy, że G jest alternatywnie redundancyjnym zbiorem zależności funkcjonalnych do zbioru E, natomiast B jest ARZZF do G. Chcemy dowieść, że B jest ARZZF do E.

Oznaczmy przez H_1 takie nieredundancyjne pokrycie zbioru F, które zawiera zbiór E. Z założeń wynika, że zbiór $H_2 = H_1 - E \cup G$ też jest nieredundancyjnym pokryciem zbioru F. Podobnie takim pokryciem jest zbiór $H_3 = H_2 - G \cup B$.

Wykażemy najpierw, że $H_1 - E \cup B$ jest nieredundancyjnym pokryciem zbioru F. Wynika to z następujących równości:

$$H_1 - E \cup B = H_1 - E \cup G - G \cup B = H_2 - G \cup B = H_3.$$

Wobec tego każda z zależności $e_i \in (E-B)$ jest redundancyjna w zbiorze $H_1 \cup B$.

Rozważmy następnie pytanie, czy istnieje taki podzbiór $B' \subset (B-E)$, że dołączenie B' do H_1 spowoduje redundancyjność pewnej zależności $e_i \in (E-B)$

w zbiorze $H_1 \cup B'$. Załóżmy dla dowodu nie wprost, że tak jest, a więc zbiór $H_1 - \{e_i\} \cup B'$ jest nieredundancyjnym pokryciem zbioru F. Powyższe wyrażenie możemy przekształcić następująco:

$$H_1 - \{e_i\} \cup B' = H_1 - E \cup G \cup E - G - \{e_i\} \cup B' = H_2 - G \cup E' \cup B',$$

gdzie $E' = E - \{e_i\}$. Jednakże zbiór G może być zastąpiony w pokryciu H_2 - zgodnie z definicją ARZZF - tylko przez cały zbiór E lub B. Ponieważ $E' \cup B'$ jest różne od E oraz od B, więc zbiór $H_1 - \{e_i\} \cup B'$ nie jest nieredundacyjnym pokryciem zbioru F. □

Wniosek 4.3.2. Własność alternatywnej redundancyjności zbiorów zależności funkcjonalnych dzieli - jako relacja równoważności - rodzinę alternatywnie redundancyjnych podzbiorów zbioru F na klasy równoważności.

Przykład 4.3.2. Rozpatrzmy następujący zbiór zależności funkcjonalnych:

$$F = \{X \longrightarrow ABCDYP,\ Y \longrightarrow DMNPX,\ AB \longrightarrow D,\ D \longrightarrow AC,\ AC \longrightarrow B,\ MN \longrightarrow P,\ P \longrightarrow MN,\ U \longrightarrow QRV,\ V \longrightarrow QRU\}.$$

Jedną z wielu możliwych postaci nieredundancyjnego pokrycia zbioru F przedstawia zbiór H:

$$H = \{X \longrightarrow ABY,\ Y \longrightarrow MNX,\ AB \longrightarrow D,\ D \longrightarrow AC,\ AC \longrightarrow B,\ MN \longrightarrow P,\ P \longrightarrow MN,\ U \longrightarrow QRV,\ V \longrightarrow U\}.$$

W zbiorze F wyróżnić można następujące trzy klasy równoważności alternatywnie redundancyjnych zbiorów zależności funkcjonalnych:

$$ARZ_1 = \{\{X \longrightarrow AB\},\ \{X \longrightarrow AC\},\ \{X \longrightarrow D\},\ \{Y \longrightarrow D\}\},$$
$$ARZ_2 = \{\{Y \longrightarrow MN\},\ \{Y \longrightarrow P\},\ \{X \longrightarrow P\}\},$$
$$ARZ_3 = \{\{U \longrightarrow QR\},\ \{V \longrightarrow QR\},\ \{U \longrightarrow Q,\ V \longrightarrow R\},\ \{U \longrightarrow R,\ V \longrightarrow Q\}\}.$$

Zauważmy, że w przedstawionym pokryciu znajduje się pierwszy ze zbiorów każdej klasy. □

Zauważmy, że definicja alternatywnie redundancyjnych zbiorów zależności funkcjonalnych dopuszcza możliwość wystąpienia niepustej części wspólnej dwóch lub więcej zbiorów. W klasie równoważności ARZZF mogą na przykład wystąpić zbiory Z_1, Z_2, Z_3 o własnościach:

$$ARZ = \{Z_1, Z_2, Z_3\},\ Z_1 \cap Z_2 = \{\phi\},\ Z_1 \cap Z_3 = \{\emptyset\},\ Z_2 \cap Z_3 \neq \{\emptyset\}.$$

Zbiory o tych własnościach prezentują przykłady 4.3.1. i 4.3.2.

Z definicji ARZZF wynika ponadto możliwość wystąpienia niepustej części wspólnej dla wszystkich zbiorów tworzących daną klasę równoważności. Fakt ten nie ma jednak znaczenia dla analizy możliwości tworzenia wszystkich postaci pokrycia, która jest głównym celem tego rozdziału. Taka część wspólna będzie bowiem obecna we wszystkich postaciach pokrycia.

Zauważmy dalej, że jeśli E i G są ARZZF, a przy tym $E \cap G = W \neq \{\emptyset\}$, to zbiory $E' = E - W$ oraz $G' = G - W$ są również ARZZF.

Tak więc przyjmiemy jako regułę, że dla każdej rozpatrywanej klasy równoważności ARZZF w postaci $ARZ = \{Z_1, Z_2, \ldots, Z_n\}$ zachodzi

$$\bigcap_{i=1}^{n} Z_i = \{\emptyset\}. \qquad (4.3.1)$$

TWIERDZENIE 4.5. Niech $E = \{e_1, e_2, \ldots, e_n\}$ i $G = \{g_1, g_2, \ldots, g_m\}$ będą alternatywnie redundancyjnymi zbiorami zależności funkcjonalnych w danym zbiorze F. Spełnione są wtedy własności:

$$(1)\quad \forall_{g_i \in G}\ \forall_{e_j \in E}\quad L(e_j) \longleftrightarrow L(g_i) \in F^+, \qquad (4.3.2)$$

$$(2)\quad \forall_{g_i, g_k \in G}\quad L(g_i) \longleftrightarrow L(g_k) \in F^+, \qquad (4.3.3)$$

$$(3)\quad \forall\quad L(e_j) \longleftrightarrow L(e_1) \in F^+. \qquad (4.3.4)$$

D o w ó d ad (1). Załóżmy, że H_1 jest nieredundancyjnym pokryciem zbioru F i $E \subseteq H_1$. Z definicji ARZZF wynika, że dowolna zależność $e_j \in E$ staje się redundancyjna dopiero po dołączeniu do H_1 całego zbioru G. To znaczy, że w wywodzie zależności e_j ze zbioru $H_1 - E \cup G$ wykorzystana jest każda zależność g_i ze zbioru G. Z lematu 2.1 wynika wtedy, że do domknięcia $(H_1 - E \cup G)^+ = F^+$ należeć musi zależność $L(e_j) \longrightarrow L(g_i)$ dla każdego $g_i \in G$.

Rozpatrzmy następnie inne pokrycie H_2 zbioru G takie, że $G \subseteq H_2$. Ponieważ własność alternatywnej redundancyjności zbiorów jest symetryczna, więc dowolna zależność $g_i \in G$ stanie się redundancyjna dopiero po dołączeniu do H_2 całego zbioru E. Powtarzając poprzednie rozumowanie otrzymamy zależność $L(g_i) \longrightarrow L(e_j)$. Obie zależności tworzą bijekcję (4.3.2).

ad (2). Z własności (4.3.2) wynika, że dla każdej trójki zależności g_i, g_k, e_j zachodzi

$$L(g_i) \longleftrightarrow L(e_j) \in F^+,$$

$$L(g_k) \longleftrightarrow L(e_j) \in F^+.$$

Na podstawie aksjomatu A3 Armstronga z powyższych związków wyprowadzić można zależności $L(g_i) \longleftrightarrow L(g_k)$ oraz $L(g_k) \longleftrightarrow L(g_j)$.

Dowód własności (3) przebiega identycznie przy rozpatrzeniu trójki zależności e_j, e_l, g_i. □

Przykład 4.3.3. Zauważmy, że w zbiorze zależności funkcjonalnych F przedstawionym w przykładzie 4.3.2 lewe strony zależności tworzących alternatywnie redundancyjne zbiory klas ARZ_1, ARZ_2 i ARZ_3 należą do dwóch klas równoważności kluczy $K_1 = \{X,Y\}$ oraz $K_2 = \{U,V\}$. Spełnione są więc własności przedstawione w twierdzeniu 4.5. Zauważmy przy tym, że lewe strony zależności tworzących ARZ_1 i ARZ_2, tzn. dwie różne klasy równoważności ARZZF, należą do tej samej klasy równoważności kluczy. □

Wniosek 4.3.3. Alternatywnie redundancyjne zbiory zależności funkcjonalnych różniące się jedynie lewymi stronami zależności dają te same schematy relacji. Bardziej precyzyjnie możemy sformułować tę własność następująco:

Jeśli $E = \{e_1,\ldots,e_n\}$, $G = \{g_1,\ldots,g_m\}$ są dwoma ARZZF takimi, że $\bigcup_{i=1}^{n} P(e_i) = \bigcup_{i=1}^{m} P(g_i)$, to zbiory E i G prowadzą do tych samych schematów relacji.

Prawdziwość wniosku wynika z faktu, że wszystkie lewe strony zależności $e_1,\ldots,e_n$, a także $g_1,\ldots,g_m$ są kluczami równoważnymi. Wszystkie atrybuty tworzące zależności E lub G wystąpią więc w tym samym schemacie relacji. □

Specyficzny problem wśród ARZZF tworzą alternatywnie redundancyjne zbiory zależności wiążących klucze równoważne. Takie zbiory mogą istnieć w danym zbiorze F, jeśli któraś z klas równoważności kluczy w F liczy więcej niż dwa klucze.

Przykład 4.3.4. Rozpatrzmy następujący zbiór zależności funkcjonalnych:

$$F = \{A \longrightarrow BCDW,\ B \longrightarrow AC,\ C \longrightarrow A,\ D \longrightarrow A\}.$$

W zbiorze tym istnieje jedna klasa równoważności kluczy $K = \{A,B,C,D\}$. Można utworzyć dwa nieredundancyjne pokrycia dla zbioru F:

$$H_1 = \{A \longrightarrow BDW,\ B \longrightarrow C,\ C \longrightarrow A,\ D \longrightarrow A\},$$

$$H_2 = \{A \longrightarrow BCDW,\ B \longrightarrow A,\ C \longrightarrow A,\ D \longrightarrow A\}.$$

Wynika stąd, że w zbiorze F istnieją dwa ARZZF:

$$E = \{A \longrightarrow C,\ B \longrightarrow A\} \quad \text{oraz} \quad G = \{B \longrightarrow C\}.$$

Zauważmy przy tym, że obie postacie pokrycia prowadzą do identycznego schematu jedynej w tym przykładzie relacji

$$R = \underline{ABCDW}.$$

Pokrycie dla potrzeb syntezy H_s dla danego zbioru F będzie miało postać

$$H_s = \{A \longrightarrow BCDW,\ B \longrightarrow ACD,\ C \longrightarrow ABD,\ D \longrightarrow ABC\}.$$

W tym przypadku nie występuje żadna zależność redundancyjna. Pokrycie to daje taki sam jak poprzednio schemat relacji. □

Zauważmy, że zależności tworzące zbiory E i G wiążą klucze należące do tej samej klasy równoważności. Zastąpienie w pokryciu zbioru E zbiorem G lub odwrotnie nie zmieni więc schematu relacji, ponieważ wszystkie klucze tworzące daną klasę równoważności muszą zawsze wystąpić w tym samym schemacie.

Zwróćmy dalej uwagę, że przyjęcie do rozważań pokrycia dla potrzeb syntezy automatycznie eliminuje możliwość wystąpienia alternatywnie redundancyjnych zbiorów zależności wiążących klucze z tej samej klasy równoważności. Wszystkie takie zależności, tworzące bijekcję, będą bowiem należeć do pokrycia H_s.

Podobnie jak dla zależności alternatywnie redundancyjnych możliwe jest rozpatrywanie zbiorów alternatywnie redundancyjnych na podstawie definicji pokrycia dla potrzeb syntezy. Jak pokazuje powyższy przykład, alternatywnie redundancyjna zbiory zależności wiążących klucze z tej samej kla-

sy równoważności nie będę wtedy wyodrębniane. Spowoduje to jednak tylko uproszczenie procesu syntezy.

TWIERDZENIE 4.6. Rozpatrzmy zbiór zależności funkcjonalnych F, w którym wyróżniono między innymi dwie klasy równoważności kluczy $K_1 = \{X_1, X_2, \ldots, X_r\}$, $K_2 = \{V_1, V_2, \ldots, V_q\}$. Oznaczmy zbiór atrybutów kluczoczowych klasy K_1 następująco $S_x = \bigcup_{i=1}^{r} X_i$. Załóżmy, że w zbiorze F istnieją między innymi dwie zależności złożone o postaci

$$E: X_i \longrightarrow (V_k - S_x) \quad \text{oraz} \quad G: X_j \longrightarrow (V_l - S_x),$$

gdzie $X_i, X_j \in K_1$, a $V_k, V_l \in K_2$.

Jeśli jedna z zależności E lub G wystąpi w pokryciu dla potrzeb syntezy H_s zbioru F, to zależności te (będące zbiorami zależności prostych) są względem siebie alternatywnie redundancyjnymi zbiorami zależności funkcjonalnych.

D o w ó d. Przyjmijmy jak poprzednio podział pokrycia H_s na podzbiór H_b zawierający bijekcje i zależności pozostałe H_p, tzn. $H_s = H_b \cup H_p$. Załóżmy, że $E \subseteq H_s$, a dokładniej $E \subseteq H_p$.

Należy wykazać, że

(1) każda zależność $e_i \in E$ jest redundancyjna w zbiorze $H_s \cup G$, tzn.

$$\forall_{e_i \in E}: e_i \in (H_s - E \cup G)^+,$$

(2) $(H_s - E \cup G)^+ = H_s^+$, a przy tym w zbiorze $H_p - E \cup G$ brak zależności redundancyjnych,

(3) żaden podzbiór $G' \subset G$ nie posiada własności zbioru G w stosunku do dowolnego podzbioru $E' \subseteq E$.

ad (1). Z definicji pokrycia dla potrzeb syntezy wynika, że do zbioru H_b muszą należeć wszystkie zależności postaci $X_n \longrightarrow X_m$, $n,m = 1,\ldots,r$ oraz $V_t \longrightarrow V_w$, $t,w = 1,\ldots,q$. Zależności te nie są naruszane przy zamianie zbioru E na zbiór G. Do zbioru $H_s - E \cup G$ należę więc następujące zależności

$$X_i \longrightarrow X_j, \; G: X_j \longrightarrow (V_l - S_x), \; X_j \longrightarrow X_m, \; m = 1,\ldots,r, \; V_l \longrightarrow V_k \qquad (4.3.5)$$

Stosując aksjomat A4 do zależności $X_j \longrightarrow X_m$, $m = 1,\ldots,r$, otrzymamy zależność $X_j \longrightarrow S_x$, scalenie której z $X_j \longrightarrow (V_l - S_x)$ daje zależność $X_j \longrightarrow V_l$. Stosując następnie aksjomat A3 do zależności $X_i \longrightarrow X_j$, $X_j \longrightarrow V_l$, $V_l \longrightarrow V_k$ otrzymamy zależność (zbiór zależności prostych) $X_i \longrightarrow V_k$, zawierającą zależność E.

Wyprowadzanie E z $H_s - E \cup G$ oznacza, że

$$\forall_{e_i \in E}: e_i \in (H_s - E \cup G)^+.$$

ad (2). Wykorzystując własność (1) dla każdej e_i i stosując lemat 3.5 otrzymamy

$$(H_s - E \cup G)^+ = (H_s \cup G)^+.$$

Ponieważ $G \subseteq F$ oraz $H_s^+ = F^+$, więc $\forall_{g_i \in G}: g_i \in H_s^+$. Z lematu 3.5 mamy więc

$$H_s^+ = (H_s \cup G)^+,$$

co daje równość

$$(H_s - E \cup G)^+ = H_s^+.$$

Załóżmy następnie dla dowodu nie wprost, że pewna zależność $h \in H_p$ stała się redundancyjną po zamianie zbioru E na zbiór G, co zapiszemy:

$$h \notin (H_s - \{h\})^+, \quad h \in (H_s - E \cup G - \{h\})^+. \qquad (4.3.6)$$

Przyjmijmy, że $h \notin G$ (przypadek $h \in G$ rozważymy oddzielnie). Do zbioru $H_s = E \cup G - \{h\}$ należę wtedy wszystkie zależności (4.3.5). Na mocy lematu 3.5 otrzymamy więc

$$(H_s - E \cup G - \{h\})^+ = (H_s \cup G - \{h\})^+. \qquad (4.3.7)$$

Zauważmy, że z definicji $h \notin E$. Wobec tego w zbiorze $H_s - \{h\}$ obecne są zależności:

$$X_j \longrightarrow X_i, \; E: X_i \longrightarrow (V_k - S_x), \; X_i \longrightarrow X_m, \; m = 1,\ldots,r, \; V_k \longrightarrow V_l.$$

Powtarzając rozumowanie jak dla części (1) tego dowodu wyprowadzić można zależności zbioru G z $H_s - \{h\}$, co prowadzi do rezultatu

$$(H_s - \{h\})^+ = (H_s \cup G - \{h\})^+. \qquad (4.3.8)$$

Z zależności (4.3.7) i (4.3.8) otrzymamy

$$(H_s - \{h\})^+ = (H_s \cup G - \{h\})^+,$$

co daje sprzeczność warunków (4.3.6).

ad (3). Załóżmy dla dowodu nie wprost, że dołączenie do pokrycia H_s podzbioru $G' \subset G$ w postaci

$$G' : X_j \longrightarrow (V_j - S_x)', \tag{4.3.9}$$

gdzie $(V_1 - S_x)' \subset (V_1 - S_x)$ eliminuje z H_s pewną zależność $X_i \longrightarrow A$, gdzie $A \in (V_k - S_x)$, a zbiór

$$H_1 = H_s - \{X_i \longrightarrow A\} \cup \{X_j \longrightarrow (V_1 - S_x)'\} \tag{4.3.10}$$

jest pokryciem dla potrzeb syntezy zbioru F.

Rozpatrzmy problem wywodu zależności $X_i \longrightarrow A$ ze zbioru H_1. Zauważmy, że w wywodzie tym muszą być wykorzystane wszystkie zależności tworzące zależność złożoną $X_j \longrightarrow (V_1 - S_x)'$, ponieważ inaczej zależność $X_j \longrightarrow A$ byłaby wyprowadzalna w $H_s - \{X_j \longrightarrow A\}$ (tzn. redundancyjna w H_s) bądź też zbiór H_1 nie byłby pokryciem dla potrzeb syntezy. Do H_1 należą zależności:

$$X_i \longrightarrow X_j,\ G' : X_j \longrightarrow (V_1 - S_x)',\ X_j \longrightarrow X_m,\ m = 1,\ldots,r,\ V_1 \longrightarrow A.$$

Postępując jak w trakcie dowodu własności (1) otrzymamy $X_i \longrightarrow V_1'$, $V_1 \longrightarrow A$, gdzie $V_1' \subset V_1$.

Wobec tego wyprowadzenie stąd zależności $X_i \longrightarrow A$ byłoby możliwe w dwóch przypadkach:

(a) przy spełnieniu warunku $(V_1 - V_1') \subset S_x$, co jednak jest sprzeczne z (4.3.9) lub

(b) przy obecności w H_1 zależności $V_1' \longrightarrow A$. Oznaczałoby to jednak częściową zależność atrybutu $A \in V_k$ od klucza V_1, co jest sprzeczne z założeniem, że klucze V_1 i V_k są kluczami równoważnymi. □

Przykład 4.3.5. Dla zbioru zależności funkcjonalnych F z przykładu 4.3.2 wyróżnić można następujące klasy równoważności kluczy: $K_1 = \{X,Y\}$, $K_2 = \{AB,AC,D\}$, $K_3 = \{MN,P\}$, $K_4 = \{U,V\}$.

Założenia twierdzenia 4.6 spełniają następujące grupy zależności (każdą zależność złożoną można przy tym traktować jako zbiór zależności prostych): $\{X \rightarrow AB,\ X \rightarrow AC,\ X \rightarrow D,\ Y \rightarrow D\}$, $\{Y \rightarrow MN,\ Y \rightarrow P,\ X \rightarrow P\}$. Tak więc alternatywnie redundancyjne zbiory należące do klas równoważności ARZ_1 i ARZ_2 mogą być w zbiorze F z przykładu 4.3.2 wyznaczone przy wykorzystaniu twierdzenia 4.6.

Zauważmy przy tym, że pary klas równoważności K_1 i K_2 oraz K_1 i K_3 nie mają wspólnych atrybutów, stąd też każda z prawych stron przedstawionych zależności jest pełnym kluczem. □

Przykład 4.3.6. Rozpatrzmy następujący zbiór zależności funkcjonalnych i dwa różne pokrycia dla potrzeb syntezy tego zbioru:

$$F = \{XA \longrightarrow PRD,\ PRA \longrightarrow QB,\ QB \longrightarrow PRAC,\ XA \longrightarrow QB\},$$

$$H_s^1 = \{XA \longrightarrow PRD,\ PRA \longrightarrow QB,\ QB \longrightarrow PRAC\},$$

$$H_s^2 = \{XA \longrightarrow QBD,\ PRA \longrightarrow QB,\ QB \longrightarrow PRAC\}.$$

W zbiorze F występują dwa alternatywnie redundancyjne zbiory zależności funkcjonalnych $E = \{XA \rightarrow PR\}$ i $G = \{XA \rightarrow QB\}$ wiążących klucze z dwóch różnych klas równoważności względem bijekcji $K_1 = \{XA\}$ i $K_2 = \{PRA,QB\}$.

Przykład ten prezentuje przypadek analizowanych w twierdzeniu 4.6 zależności wiążących klucze z klas K_1 i K_2 posiadających wspólny atrybut A. Atrybut ten nie występuje więc w prawej stronie zależności $XA \rightarrow PRD$. □

TWIERDZENIE 4.7. Rozszerzmy założenia twierdzenia 4.6 przyjmując, że do zbioru F należy dodatkowo podzbiór zależności funkcjonalnych B o postaci:

$$B = \{X_{i_1} \longrightarrow A_{i_1},\ X_{i_2} \longrightarrow A_{i_2},\ \ldots,X_{i_t} \longrightarrow A_{i_t}\},$$

przy czym $X_{i_1} \in K_1,\ldots,X_{i_t} \in K_i$ oraz

$$\bigcup_{j=1}^{t} A_{i_j} = V_p - S_x,\ V_p \in K_2,$$

Wtedy zbiór B jest ARZZF do zbiorów E i G.

D o w ó d. Rozpatrzmy możliwość wyprowadzenia ze zbioru $H_2 = H_s - E \cup B$ zależności tworzących zbiór E. Do H_2 należą między innymi następujące zależności:

$$X_i \longrightarrow X_m, m = 1,\ldots,r,\ X_{i_1} \longrightarrow A_{i_1},\ldots,X_{i_t} \longrightarrow A_{i_t},\ V_p \longrightarrow V_k.$$

Ponieważ dla każdego klucza $X_{i_1},\ldots,X_{i_t}$ można wskazać równy mu klucz X_m, $m = 1,\ldots,r$, więc na podstawie aksjomatów A3 i A4 możliwe jest wyprowadzenie zależności $E: X_i \longrightarrow V_k$.

Wykorzystując to rozumowanie można skonstruować całość dowodu analogicznie do twierdzenia 4.6. □

Przykład 4.3.7. W zbiorze zależności funkcjonalnych $F = \{X \to DYA, YA \to BZ, Z \to CX, ABC \to D, D \to ABC\}$ istnieją dwie klasy równoważności kluczy $K_1 = \{X, YA, Z\}$ i $K_2 = \{ABC, D\}$. Można utworzyć dwa różne pokrycia dla potrzeb syntezy zbioru F:

$H_s^1 = \{X \to DYAZ, YA \to ZX, Z \to XYA, ABC \to D, D \to ABC\}$,

$H_s^2 = \{X \to YAZ, YA \to BZX, Z \to CXYA, ABC \to D, D \to ABC\}$.

W zbiorze F istnieją więc dwa zbiory alternatywnie redundancyjne $Z_1 = \{X \to D\}$, $Z_2 = \{YA \to B, Z \to C\}$.

Struktura zbioru Z_2 odpowiada strukturze zbioru B z twierdzenia 4.7. □

4.4. WYZNACZANIE ZALEŻNOŚCI I ZBIORÓW ALTERNATYWNIE REDUNDANCYJNYCH. SYNTEZA WSZYSTKICH POSTACI SCHEMATÓW RELACJI

W niniejszym podrozdziale jest prezentowana i dyskutowana organizacja procesu wyznaczania zależności i zbiorów alternatywnie redundancyjnych. W omawianych algorytmach nie rozróżnia się przy tym ARZF i ARZZF, traktując pojedyncze zależności jako szczególny przypadek zbiorów alternatywnie redundancyjnych. Wyznaczanie zbiorów alternatywnie redundancyjnych jest tylko pośrednim zadaniem przedstawianych algorytmów, końcowym zaś - synteza wszystkich możliwych postaci schematów relacji.

Pierwsza część rozważań (algorytmy 4.1 i 4.2) dotyczy najczęściej spotykanych przypadków ARZZF, przedstawionych w twierdzeniach 4.6 i 4.7. Są to więc alternatywnie redundancyjne zbiory zależności wiążących klucze z różnych klas równoważności.

Drugę część tego podrozdziału poświęcono przypadkom zilustrowanym przykładem 4.2.5.

Rozważania kończy prezentacja pełnego algorytmu syntezy.

4.4.1. Wyznaczenie alternatywnie redundancyjnych zbiorów zależności wiążących klucze z różnych klas równoważności

Na wstępie przyjmiemy szereg założeń i oznaczeń wykorzystywanych w dalszej treści.

Załóżmy, że w wyniku działania algorytmu 3.3 utworzone zostało dla danego zbioru F zależności funkcjonalnych pokrycie dla potrzeb syntezy H_s składające się z dwóch rozłącznych podzbiorów $H_s = H_b \cup H_p$, gdzie H_b zawiera wszystkie zależności tworzące bijekcje między kluczami równoważnymi, a H_p - zależności pozostałe. Zbiór zależności redundancyjnych, które nie weszły do pokrycia H_s, oznaczać będziemy przez RED.

Przyjmijmy, że w pokryciu H_s istnieją następujące klasy równoważności kluczy:

$$\begin{aligned} K_1 &= \{x_1^1, x_2^1, \ldots, x_q^1\}, \\ &\vdots \\ K_q &= \{x_1^q, x_2^q, \ldots, x_{r_q}^q\}. \end{aligned} \tag{4.4.1}$$

Oznaczmy przez S_k, $k = 1, \ldots, q$ zbiory atrybutów kluczowych poszczególnych klas:

$$S_k = \bigcup_{j=1}^{r_k} x_j^k. \tag{4.4.2}$$

Rozpatrzmy zbiór takich zależności funkcjonalnych należących do H_s, których lewe strony należą do K_m:

$$\{x_1^m \to Y_1^m, \ldots, x_{r_m}^m \to Y_{r_m}^m\}. \tag{4.4.3}$$

Przypomnijmy, że schemat relacji R_m, odpowiadający klasie równoważności kluczy K_m, tworzony jest z atrybutów należących do lewych i prawych stron zależności (4.4.3).

Utworzenie innej postaci schematu R_m dla zbioru F polega na zastąpieniu w pokryciu pewnego podzbioru zależności (4.4.3) zbiorem alternatywnie redundancyjnym. Z twierdzenia 4.5 wynika, że lewe strony zależności tworzących taki zbiór muszą należeć do K_m. Pozwala to zdekomponować proces poszukiwania ARZZF, a dalej nowych postaci schematu R_m do oddzielnych klas równoważności kluczy $K_1, \ldots, K_q$ oraz do oddzielnych podzbiorów zależności z F o lewnych stronach należących do tych klas.

Dalsze rozważania będą więc prowadzone dla wybranej dowolnie klasy równoważności kluczy K_m, $1 \leq m \leq q$.

Z twierdzenia 4.7 wynika, że jeśli w podzbiorze atrybutów tworzących R_m będzie się zawierał dowolny klucz należący do klasy K_p różnej od K_m, tzn.

$$\exists\, x_i^p \in K_p,\ x_i^p \subseteq R_m, \tag{4.4.4}$$

to w zbiorze F mogą istnieć alternatywnie redundancyjne zbiory zależności funkcjonalnych wiążących klucze tych klas. Zależności tych należy przy tym szukać w podzbiorze RED^m zbioru RED:

$$RED^m = \{x_1^m \to Z_1^m, \ldots, x_{r_m}^m \to Z_{r_m}^m\}, \tag{4.4.5}$$

obejmującym te zależności ze zbioru RED, których lewe strony należą do K_m.

Oznaczmy

$$P_m^z = \bigcup_{i=1}^{r_m} Z_i^m. \qquad (4.4.6)$$

Jeśli pewien klucz z klasy K_p (bez ewentualnych atrybutów kluczowych klasy K_m) należy do P_m^z:

$$\exists X_j^p \in K_p : (X_j^p - S_m) \subseteq P_m^z, \quad X_j^p \neq X_i^p, \qquad (4.4.7)$$

to zgodnie z twierdzeniem 4.7 w RED^m istnieje alternatywnie redundancyjny zbiór zależności funkcjonalnych do podzbioru zależności (4.4.3), których prawe strony tworzą klucz X_i^p.

Nową postać schematu R_m utworzymy wtedy zastępując w R_m podzbiór atrybutów $X_i^p - S_m$ podzbiorem $X_j^p - S_m$.

Zauważmy, że zbiór P_m^z może zawierać większą liczbę kluczy należących do K_p. Dalsze postacie schematu R_m można wtedy tworzyć analogicznie do poprzednich.

Zwróćmy w tym miejscu uwagę na to, że w powyższych rozważaniach szeroko korzystamy z faktu, że własność alternatywnej redundancyjności jest relacją równoważności. Jeśli bowiem znajdziemy rodzinę zbiorów $Z_1, Z_2, \ldots, Z_m$ alternatywnie redundancyjnych do zbioru Z, to z własności powyższej wynika, że każde dwa zbiory Z_i, Z_j, $1 \leq i, j \leq m$, $i \neq j$ są względem siebie alternatywnie redundancyjne.

Przedstawione rozważania można ująć w postać następującego algorytmu:

Algorytm 4.1. (Tworzenie nowych postaci schematu R_m dla zbiorów alternatywnie redundancyjnych obecnych w F)

Dane wejściowe: Pokrycie dla potrzeb syntezy H_s z wyróżnionymi klasami równoważności kluczy; podzbiór RED^m zbioru zależności redundancyjnych, schemat R_m dla klasy K_m.

Dane wyjściowe: zbiór $\mathfrak{R}_m$ zawierający wszystkie postacie schematu R_m utworzone dla ARZZF wykrytych w RED^m.

(1) Wyznacz $S_m = \bigcup_{j=1}^{r_m} X_j^m$, $P_m^z = \bigcup_{j=1}^{r_m} Z_j^m$.
Utwórz zbiór $\mathfrak{R}_m$ początkowo pusty i wprowadź do niego schemat R_m.

(2) Dla każdej klasy równoważności kluczy K_p różnej od K_m sprawdź, czy istnieje taki klucz $X_i^p \in K_p$, że zachodzi $(X_i^p - S_m) \subseteq (R_m - S_m)$ oraz $X_i^p - S_m \neq \{\emptyset\}$. Jeśli tak, oznacz $X := X_i^p$ i wykonaj krok (3).

(3) Dla każdego z kluczy $X_j^p \in K_p$, różnego od X, sprawdź, czy zachodzi $(X_j^p - S_m) \subseteq P_m^z$. Jeśli tak, to dla każdej wersji R_m należącej do $\mathfrak{R}_m$ i zawierającej X utwórz nową postać $R_m' := R_m - \{X - S_m\} \cup \{X_j^p - S_m\}$, a następnie dołącz ją do $\mathfrak{R}_m$. Wróć do kroku (2). □

Twierdzenia 4.6 i 4.7 pozwalają na rozważenie ciekawej modyfikacji przedstawionego algorytmu. Z twierdzeń tych wynika mianowicie, że jeśli w pokryciu H_s istnieje zależność wiążąca klucze dwóch różnych klas równoważności K_m i K_p (tzn. jest spełniony warunek (4.4.4)), to teoretycznie istnieje możliwość utworzenia r_p różnych postaci schematu R_m. Każdy z tych schematów zawierałby wtedy inny klucz z klasy K_p. Utworzenie tych schematów za pomocą algorytmu 4.1 będzie zależało od tego, czy w zbiorze F będą obecne ARZZF o prawych stronach równych wszystkim kluczom z klasy K_p. Będzie to więc zależne od spostrzegawczości bądź zamiarów projektanta definiującego zależności funkcjonalne w zbiorze F. W przedstawionym dalej wariancie algorytmu 4.1 proponuje się wygenerowanie wszystkich możliwych postaci schematu biorąc pod uwagę powiązania między klasami równoważności kluczy wykryte w pokryciu H_s. W ten sposób algorytm "wyręcza" projektanta z konieczności deklarowania wszystkich związków między dwiema klasami kluczy albo też ujawnia projektantowi możliwe połączenia, których nie dostrzegł.

Algorytm 4.2. (Tworzenie wszystkich teoretycznie możliwych postaci schematu R_m)

Dane wejściowe: Pokrycie dla potrzeb syntezy H_s z wyodrębnionymi klasami równoważności kluczy; schemat R_m dla klasy K_m.

Dane wyjściowe: Zbiór $\mathfrak{R}_m$ zawierający wszystkie postacie schematu R_m dla wszystkich klas równoważności kluczy związanych z klasą K_m.

(1) Wyznacz $S_m = \bigcup_{j=1}^{r_m} X_j^m$. Utwórz zbiór $\mathfrak{R}_m$ początkowo pusty i wprowadź do niego schemat R_m.

(2) Dla każdej klasy równoważności kluczy K_p różnej od K_m sprawdź, czy istnieje taki klucz $X_i^p \in K_p$, że zachodzi $(X_i^p - S_m) \subseteq (R_m - S_m)$ oraz $X_i^p - S_m \neq \{\emptyset\}$. Jeśli tak, oznacz $X := X_i^p$ i wykonaj krok (3).

(3) Dla każdego schematu $R_m' \in \mathfrak{R}_m$ takiego, że $X \subseteq R_m'$ wyznacz $R_m^1 := R_m' - (X - S_m)$, a następnie dla każdego $X_j^p \in K_p$, $X_j^p \neq X$, wyznacz $R_m^2 := R_m^1 \cup (X_j^p - S_m)$ i dołącz R_m^2 do $\mathfrak{R}_m$, tzn. $\mathfrak{R}_m := \mathfrak{R}_m \cup \{R_m^2\}$. Wróć do kroku (2). □

4.4.2. Wykrywanie zbiorów alternatywnie redundancyjnych metodą powtarzania operacji wyznaczania pokrycia

Przedstawione algorytmy pozwalają wyznaczać ARZZF wiążące dwie klasy kluczy równoważnych. Przykład 4.2.5. pokazuje jednak możliwość istnienia takich zależności alternatywnie redundancyjnych, których prawe strony nie są kluczami równoważnymi.

Ogólna koncepcja proponowanej metody wykrywania tego typu ARZF lub ARZZF będzie polegała na powtórzeniu (jedno- lub wielokrotnym) operacji wyznaczania pokrycia dla tego samego zbioru zależności, ale różnie uporządkowanych. Kolejny zbiór danych wejściowych będzie połączeniem pokrycia i zbioru zależności redundancyjnych z poprzedniego kroku obliczeń. Wykrycie cyklu w uzyskiwanych wynikach pozwoli zakończyć obliczenia.

Na wstępie zwrócimy uwagę na możliwość znacznego ograniczenia zbioru zależności, wśród których poszukiwane będą ARZZF. Przyjmując podział pokrycia na podzbiory H_b i H_p możemy zgodnie z definicją H_s ograniczyć zakres badań do podzbioru H_p. Załóżmy dalej, że omawiany etap poszukiwań poprzedzony zostanie wykryciem ARZZF wiążących klucze z różnych klas (algorytm 4.1 lub 4.2). Przyjmijmy, że wykryte w tym etapie zależności wyodrębnimy w podzbiorze $H_k \subseteq H_p$.

W dalszych badaniach zależności tworzących pokrycie H_s możemy się więc ograniczyć do zależności należących do podzbioru $H_r = H_p - H_k$.

Odrębnym zagadnieniem jest przedstawiona na wstępie tego podrozdziału możliwość zdekomponowania procesu poszukiwań do podzbiorów zależności, których lewe strony należą do oddzielnych klas równoważności kluczy. Rozpatrując pewną klasę równoważności kluczy K_m założymy w związku z tym, że zależności o lewych stronach należących do K_m tworzą w H_r i RED podzbiory $H_r^m \subseteq H_r$ oraz $RED^m \subseteq RED$. Zgodnie z definicją ARZZF i twierdzeniem 4.5, jeśli w podzbiorze H_r^m jakiś podzbiór zależności $E \subseteq H_r^m$ zastąpiony zostanie alternatywnie redundancyjnym podzbiorem G, to G musi pochodzić z RED^m.

Przyjmijmy dalej dwa założenia dotyczące sposobu prezentacji zależności tworzących H_r^m i RED^m.

(a) Prawe strony wszystkich zależności tworzących H_r^m i RED^m są atrybutami prostymi (przekształcenie każdej zależności złożonej do tej postaci jest możliwe na podstawie aksjomatu A5).

(b) Zależności tworzące H_r^m i RED^m zostały uporządkowane; kolejność wystąpienia w zbiorach określają ich indeksy.

Proces powtarzania operacji wyznaczania pokrycia i syntezy nowych postaci schematów relacji, zdekomponowany do klasy równoważności kluczy K_m, przedstawia następujący algorytm:

Algorytm 4.3. (Powtarzanie operacji wyznaczania pokrycia)

Dane wejściowe: Zbiór H_s z wyróżnionym podzbiorem $H_r^m = \{h_1, h_2, \ldots, h_d\}$, zbiór RED z wyróżnionym podzbiorem $RED^m = \{r_1, r_2, \ldots, r_e\}$, schemat R_m dla klasy K_m.

Dane wyjściowe: Dodatkowe postacie schematu relacji R_m rozszerzające rodzinę $\mathcal{R}_m$.

(1) Oznacz $i := 0$, $H_r^{m,0} := H_r^m$, $RED_0^m := RED^m$, $H_s^0 := H_s$. Utwórz zbiór $\mathcal{R}'_m$ początkowo pusty i włącz do niego schemat R_m, utworzony w pokryciu H_s dla klasy równoważności kluczy K_m.

(2) Dla zależności $h_1, h_2, \ldots, h_d$, $r_1, r_2, \ldots, r_e$ tworzących podzbiory $H_r^{m,i}$ oraz RED_i^m sprawdź w przedstawionej kolejności, czy $h_j \in (H_s^i \cup RED_i^m - \{h_j\})^+$, $j = 1, \ldots, d$ lub $r_k \in (H_s^i \cup RED_i^m - \{r_k\})^+$, $k = 1, \ldots, e$. Jeśli tak, usuń h_j ze zbioru $H_r^{m,i}$ lub r_k za zbioru RED_i^m i wprowadź h_j lub r_k do zbioru RED_{i+1}^m zachowując uporządkowanie tego zbioru zgodne z kolejnością wprowadzania. Jeśli nie, wprowadź badaną zależność h_j lub r_k do zbioru $H_r^{m,i+1}$ zachowując kolejność.

(3) Oznacz pokrycie dla potrzeb syntezy zawierające $H_r^{m,i+1}$ przez H_s^{i+1} i przyjmij $i := i+1$.

(4) Utwórz w pokryciu H_s^i schemat R_m dla klasy równoważności kluczy K_m. Jeśli $R_m \notin \mathcal{R}'_m$, dołącz schemat R_m do $\mathcal{R}'_m$ i wróć do kroku (2).

(5) Sprawdź, czy w rodzinie zbiorów $\{H_r^{m,0}, RED_0^m\}, \ldots, \{H_r^{m,i-1}, RED_{i-1}^m\}$ istnieje para zbiorów $\{H_r^{m,j}, RED_j^m\}$ identyczna z parą $\{H_r^{m,i}, RED_i^m\}$ pod względem zawieranych zależności i ich uporządkowania. Jeśli tak, dołącz $\mathcal{R}'_m$ do $\mathcal{R}_m$, tzn. $\mathcal{R}_m := \mathcal{R}_m \cup \mathcal{R}'_m$ i zakończ realizację algorytmu. W przeciwnym przypadku wróć do kroku (2). □

Rozpatrzmy kilka własności przedstawionego algorytmu.

Wniosek 4.4.1. Liczba powtórzeń operacji wyznaczania pokrycia jest skończona.

Wynika to z faktu, że zarówno liczba, jak też postać zależności funkcjonalnych tworzących zbiory $H_r^{m,i}$ i RED_i^m nie zmieniają się. Charakteryzując więc najogólniej algorytm 4.3 można powiedzieć, że zmienia on jedynie uporządkowanie zależności w łącznym zbiorze $\{H_r^{m,i}, RED_i^m\}$.

Tak więc w najgorszym przypadku - bez analizy metody poszukiwań i własności ARZZF - można stwierdzić, że liczba powtórzeń nie przekroczy liczby permutacji zależności tworzących zbiory H_r^m i RED^m.

Pokażemy teraz związki między liczbą powtórzeń operacji wyznaczania pokrycia a liczbą ARZZF tworzących daną klasę równoważności.

Lemat 4.2. Jeśli w zbiorze F istnieje jedna klasa równoważności alternatywnie redundancyjnych zależności funkcjonalnych $AR_m = \{f_1, f_2, \ldots, f_t\}$, przy czym jedna z tych zależności $f_j \in AR_m$ należy do H_r, tzn. $f_j \in H_r \subseteq H_s$, to po t-1-krotnym powtórzeniu operacji wyznaczania pokrycia według algorytmu 4.3 każda z zależności $f_1, \ldots, f_t$ wystąpi dokładnie jeden raz w pokryciu dla potrzeb syntezy H_s.

D o w ó d. Zgodnie z twierdzeniem 4.2 lewe strony zależności $f_1, \ldots, f_t$ muszą należeć do jednej klasy równoważności kluczy. Załóżmy, że jest to klasa K_m. Zależność $f_j \in H_r$ musi wobec tego należeć do H_r^m, pozostałe zaś zależności z klasy AR_m - do RED^m. Załóżmy, że tworzą one w RED^m ciąg $(f_{i_1}, f_{i_2}, \ldots, f_{i_{t-1}})$. Przy pierwszej iteracji algorytmu 4.3 zależność $f_j \in H_r^{m,0}$, badana jako pierwsza spośród zależności tworzących AR_m, zostanie zakwalifikowana do zależności redundancyjnych. Zostanie ona umieszczona jako pierwsza w zbiorze RED_1^m. Przy tej operacji zmienimy jej oznaczenie na f_{i_o}. Do zależności redundancyjnych zaliczone zostaną również $f_{i_1}, \ldots, f_{i_{t-2}}$. Natomiast badana jako ostatnia zależność $f_{i_{t-1}}$ znajdzie się w podzbiorze $H_r^{m,1}$ pokrycia H_s^1. W zbiorze RED_1^m zależności klasy AR_m będą uporządkowane względem siebie następująco: $f_{i_o}, f_{i_1}, \ldots, f_{i_{t-2}}$, przy czym f_{i_o} będzie w ogóle pierwszą zależnością w tym zbiorze.

Ponieważ operacja wyznaczania pokrycia nie zmienia, zgodnie z założeniem, uporządkowania względem siebie zależności pozostających w zbiorze RED^m, więc w kolejnych iteracjach do pokrycia będą wybierane zależności $f_{i_{t-2}}, f_{i_{t-3}}, \ldots$. Po t-1 iteracjach w podzbiorze $H_r^{m,t-1}$ znajdzie się zależność f_{i_1}. Po następnej iteracji w podzbiorze $H_r^{m,t}$ znajdzie się ponownie zależność f_{i_o}, natomiast zależności $f_{i_1}, \ldots, f_{i_{t-1}}$ zajmą t-1 pierwszych pozycji w zbiorze RED_t^m.

Jeśli - zgodnie z założeniami - brak w zbiorze F innych klas równoważności ARZF i ARZZF o lewych stronach należących do K_m, to żadne inne zależności - poza $f_1, \ldots, f_t$ - nie mogły zostać wprowadzone do podzbioru H_r^m pokrycia H_s. Wobec tego podzbiory $H_r^{m,0}$ i $H_r^{m,t}$ oraz RED_o^m i RED_t^m zawierają te same zależności. Mogą się one różnić jedynie ich uporządkowaniem. Dla przyjętych w lemacie założeń warunkiem zakończenia algorytmu mogłaby być wskazana zgodność zbiorów $H_r^{m,0}$ i $H_r^{m,t}$. Jednakże w pewnych przypadkach omówionych dalej (przykład 4.4.1) taki warunek zakończenia nie byłby poprawny. Ponieważ pierwszych t iteracji doprowadziło do ustawienia zależności klasy AR_m na początku zbioru RED_t^m, więc powtórne t iteracji (w najgorszym przypadku) doprowadzi do wymaganej zgodności uporządkowania badanych zbiorów. □

Wniosek 4.4.2. Jeśli w zbiorze F występią dwie klasy równoważności ARZF w postaci $AR_m^a = \{f_1^a, \ldots, f_t^a\}$ i $AR_m^b = \{f_1^b, \ldots, f_s^b\}$, i jeśli para zależności (f_i^a, f_j^b), gdzie $f_i^a \in AR_m^a$, $f_j^b \in AR_m^b$, należy do pokrycia $H_r^{m,0}$, to para ta wystąpi ponownie w pokryciu po w interacjach algorytmu 4.3, przy czym w jest najmniejszą wspólną wielokrotnością liczb t i s.

Słuszność tego wniosku wynika z analizy dowodu lematu 4.2. □

Lemat 4.3. Załóżmy, że w zbiorze F zależności funkcjonalnych istnieje jedna klasa równoważności alternatywnie redundancyjnych zbiorów zależności funkcjonalnych $ARZ_m = \{Z_1, \ldots, Z_t\}$ o następujących własnościach:

(1) Klasa ARZ_m reprezentowana jest w podzbiorze H_r pokrycia dla potrzeb syntezy H_s zbioru F, tzn. $\exists Z_i \in ARZ_m$, $Z_i \subseteq H_r$.

(2) Zbiory tworzące ARZ_m są rozłączne, tzn. $Z_i \cap Z_j = \{\emptyset\}$, $1 \leq i,j \leq t$, $i \neq j$.

Przez K_m oznaczymy klasę równoważności kluczy, do której należą lewe strony wszystkich zależności tworzących zbiory ARZ_m. Zakładać będziemy, że w F nie występuje żadna inna (nietrywialna) klasa równoważności zależności lub zbiorów alternatywnie redundancyjnych o lewych stronach należących do K_m.

Przy powyższych założeniach, po t-1-krotnym powtórzeniu operacji wyznaczania pokrycia w algorytmie 4.3, każdy ze zbiorów $Z_1, \ldots, Z_t$ wystąpi w pokryciu H_s.

D o w ó d. Nazwijmy badaniem krytycznym dla zbioru Z_i wykonywaną po raz pierwszy (w danej iteracji kroku (2) algorytmu 4.3) operację sprawdzania, czy pewna zależność funkcjonalna należąca do Z_i jest redundancyjna. Sens powyższej nazwy tkwi w tym, że jeśli badanie krytyczne doprowadzi do usunięcia badanej zależności, czyli zdekompletowania zbioru Z_i, to wszystkie pozostałe zależności należące do zbioru Z_i też zostaną usunięte (wynika to z definicji ARZZF).

Oznaczmy zbiór $Z_i \subseteq H_r^{m,0}$ przez Z_{i_1}. Zauważmy, że w wyniku powtórzenia operacji wyznaczania pokrycia (krok (2) algorytmu 4.3) do nowej postaci pokrycia (a ściślej $H_r^{m,1}$) wybrany zostanie taki zbiór $Z_{i_2} \in ARZ_m$ różny od Z_{i_1}, na którym jako ostatnim wykonane zostało badanie krytyczne. Zależności zbioru Z_{i_1} znajdą się natomiast na początku zbioru RED_1^m. W kolejnej iteracji algorytmu 4.3 do podzbioru $H_r^{m,2}$ wprowadzony zostanie zbiór Z_{i_3} różny od Z_{i_1} i Z_{i_2}.

Ponieważ zależności zbiorów usuwanych z H_r^m ustawiane są kolejno od początku zbioru RED, więc po t-1-krotnym powtórzeniu algorytmu 4.3 utworzony zostanie ciąg zbiorów $Z_{i_t}, Z_{i_{t-1}}, \ldots, Z_{i_1}$, z których każdy należy do ARZ_m, przy czym zbiór Z_{i_t} będzie należał do $H_r^{m,t-1}$, a pozostałe zbiory wystąpią w przedstawionym porządku w zbiorze RED_{t-1}^m. Oznacza to, że każdy ze zbiorów tworzących ARZ_m wystąpił w pokryciu dla potrzeb syntezy H_s. □

Problem zakończenia realizacji algorytmu przedstawia się podobnie jak w lemacie 4.2. W pierwszych t iteracjach są wyodrębniane i ustawiane w zbiorze RED^m w pewnym porządku wszystkie zbiory klasy ARZ_m. Ponieważ dalsze powtarzanie operacji wyznaczania pokrycia nie zmienia porządku zależności w tych zbiorach, więc po 2t iteracjach (w najgorszym przypadku) powtórzy się identyczny układ zależności w zbiorach H_r^m i RED^m.

W założeniach tego jak i poprzedniego lematu można byłoby rozpatrywać prostszy warunek zakończenia realizacji algorytmu, tzn. równość podzbiorów $H_r^{m,0}$ i $H_r^{m,1}$. Jednakże w przypadku, kiedy zbiory tworzące ARZ_m posiadają niepuste części wspólne, proces wymiany zbiorów alternatywnie redundancyjnych pomiędzy H_r^m i RED^m może nie przebiegać tak regularnie jak w prezentowanych lematach.

Przykład 4.4.1. Przyjmując wszystkie oznaczenia z lematu 4.3 załóżmy, że klasa równoważności ARZ_m składa się z trzech zbiorów $ARZ_m = \{Z_1, Z_2, Z_3\}$ takich, że $Z_1 \cap Z_2 = \{\emptyset\}$, $Z_1 \cap Z_3 = \{\emptyset\}$, $Z_2 \cap Z_3 = W \neq \{\emptyset\}$. Przyjmijmy, że $Z_1 \subseteq H_r^{m,0}$. Wtedy w RED^m pozostanie ten ze zbiorów Z_2 lub Z_3, z którego usunięta została pierwsza zależność redundancyjna. Załóżmy, że $Z_2 \subseteq H_r^{m,1}$.

Ponieważ część wspólna W należy wraz z Z_2 do $H_r^{m,1}$, więc kolejne powtórzenie operacji wyznaczania pokrycia usunie W z $H_r^{m,1}$, a tym samym do $H_r^{m,2}$ wejdzie nie zbiór Z_3, ale ponownie zbiór Z_1. Tak więc kolejność występowania w pokryciu zbiorów tworzących ARZ_m będzie następująca:

$$Z_1, Z_2, Z_1, Z_3, Z_1, Z_2, \ldots \text{ itd.}$$

Aby przeanalizować warunki zakończenia realizacji algorytmu 4.3, w tym przypadku załóżmy następujące początkowe uporządkowanie zbiorów: $(Z_1 | Z_3, Z_2)$, gdzie Z_1 należy do $H_r^{m,0}$, natomiast Z_3 i Z_2 do RED_0^m. Porządek ten będzie się zmieniał po kolejnych iteracjach następująco:

$$(Z_1|Z_3,Z_2),\ (Z_2|Z_1,Z_3),\ (Z_1|Z_2,Z_3),\ (Z_3|Z_1,Z_2),\ (Z_1|Z_3,Z_2),\ \ldots .$$

Ponieważ po czterech iteracjach powtórzył się początkowy układ zbiorów, więc w najgorszym przypadku po ośmiu iteracjach powtórzy się uporządkowanie zależności i działanie algorytmu zostanie zakończone. □

Prezentowany przykład wyjaśnia przyczynę przyjęcia w kroku (4) algorytmu 4.3 tak złożonego warunku zakończenia obliczeń.

Możliwa byłaby dalsza analiza warunków cykliczności procesu powtarzania operacji wyznaczania pokrycia dla zbiorów alternatywnie redundancyjnych o niepustych częściach wspólnych. Zrezygnujemy jednak z prezentacji tych badań podkreślając, że celem stosowania algorytmu 4.3 jest wyszukiwanie tych nielicznych i sporadycznie mogących się pojawić zależności i zbiorów alternatywnie redundancyjnych, które nie zostały wykryte w algorytmach 4.1 lub 4.2.

Na zakończenie tego podrozdziału przedstawimy kompletny algorytm syntezy, generujący alternatywne schematy relacyjnej bazy danych według koncepcji ujętej w algorytmach 4.2 i 4.3.

Algorytm 4.4. (Końcowy algorytm syntezy tworzący zbiór alternatywnych schematów relacyjnej bazy danych)

Dane wejściowe: zbiór zależności funkcjonalnych F.
Dane wyjściowe: zbiór schematów relacyjnej bazy danych $\mathfrak{R}$.

(1) Usuń obce atrybuty z lewych stron zależności zbioru F.

(2) Podziel zbiór F na grupy zależności F_i o identycznych lewych stronach.

(3) Wyznacz klasy równoważności dla bijekcji w zbiorze kluczy (lewych stron zależności F_i). Usuń klucze pozorne spośród kluczy równoważnych.

(4) Uzupełnij zbiór F o brakujące zależności tworzące bijekcje między kluczami równoważnymi. Wyodrębnij zbiór wszystkich bijekcji H_b. Utwórz pokrycie dla potrzeb sytnezy H_s usuwając do zbioru RED zależności redundancyjne przy zachowaniu wszystkich bijekcji zbioru H_b. Oznacz $H_p = H_s - H_b$.

(5) Dla każdej klasy równoważności kluczy $K_m = \{x_1^m, \ldots, x_{r_m}^m\}$, $m=1,\ldots,q$ wykonaj następujące kroki:

(5.1) Utwórz zbiór $\mathfrak{R}_m$ początkowo pusty. Utwórz schemat relacji R_m dla klasy K_m i umieść go w zbiorze $\mathfrak{R}_m$. Oznacz przez H_r^m zbiór zależności wchodzących do H_p o lewych stronach należących do K_m. Utwórz zbiór atrybutów kluczowych klasy K_m w postaci

$$S_m := \bigcup_{i=1}^{r_m} X_i^m .$$

(5.2) Dla każdej klasy równoważności kluczy $K_p = \{x_1^p, \ldots, x_{r_p}^p\}$ różnej od K_m sprawdź, czy istnieje taki klucz $X_j^p \in K_p$, że $X_j^p \subseteq R_m$. Jeśli tak, to dla każdego schematu $R_m' \in \mathfrak{R}_m$ utwórz $r_p - 1$ nowych postaci schematu R_m', zastępując w nim atrybuty $X_j^p - S_m$ atrybutami $X_k^p - S_m$ każdego z pozostałych kluczy X_k^p klasy K_p. Włącz te schematy do $\mathfrak{R}_m$. Wyłącz ze zbioru H_r^m zależności, których prawe strony tworzą zbiór $X_j^p - S_m$.

(5.3) Zależności tworzące zbiór H_r^m przenieść do zbioru $H_r^{m,0}$ rozdzielając je na zależności proste. Wyodrębnij w zbiorze RED podzbiór RED_0^m zależności o lewych stronach należących do K_m. Ustal i:=0 i przyjmij $H_s^0 := H_s$. Utwórz zbiór $\mathfrak{R}_m'$ początkowo pusty i włącz do niego schemat R_m.

(5.4) Dla każdej zależności $h_j \in H_r^{m,i}$ oraz dla każdej zależności $r_k \in RED_i^m$ sprawdź, czy $h_j \in (H_s^i \cup RED_i^m - \{h_j\})^+$ lub $r_k \in (H_s^i \cup RED_i^m - \{r_k\})^+$. Zależności redundancyjne wprowadź do zbioru RED_{i+1}^m. Zależności, które nie są redundancyjne, wprowadź do zbioru $H_r^{m,i+1}$. Oznacz pokrycie dla potrzeb syntezy zawierające $H_r^{m,i+1}$ przez H_s^{i+1}. Przyjmij i:=i+1.

(5.5) Utwórz schemat relacji R_m dla klasy K_m w pokryciu H_s^i. Jeśli $R_m \notin \mathcal{R}'_m$, dołącz R_m do $\mathcal{R}'_m$ i wróć do kroku (5.4).

(5.6) Sprawdź, czy w rodzinie zbiorów $\{H_r^{m,0}, RED_0^m\},\ldots,\{H_r^{m,i-1}, RED_{i-1}^m\}$ istnieje para zbiorów $\{H_r^{m,j}, RED_j^m\}$ identyczna pod względem zawieranych zależności i ich uporządkowania z parą zbiorów $\{H_r^{m,i}, RED_i^m\}$. Jeśli tak, dołącz $\mathcal{R}'_m$ do $\mathcal{R}_m$, tzn. $\mathcal{R}_m := \mathcal{R}_m \cup \mathcal{R}'_m$ i zakończ realizację kroku (5) dla klasy K_m. W przeciwnym przypadku wróć do kroku (5.4).

(6) Utwórz rodzinę wszystkich postaci schematu relacyjnej bazy danych dla zbioru F wyznaczając następujący iloczyn kartezjański

$$\mathcal{R} = \mathcal{R}_1 \times \mathcal{R}_2 \times \ldots \times \mathcal{R}_q . \quad \square$$

4.4.3. Ocena złożoności obliczeniowej algorytmu wyznaczającego zbiory alternatywnie redundancyjne

Na wstępie zauważmy, że pesymistyczna złożoność obliczeniowa czterech pierwszych kroków algorytmu 4.4 wynosi $O(|F|^2)$, gdzie $|F|$ jest liczbą wystąpień atrybutów w zapisie zbioru F. Wynika to z rozważań przedstawionych w rozdziale 3. Złożoność kroku (5.1) można pominąć, natomiast złożoność kroku (5.2) określa - dla klasy K_m - liczba operacji badania zawierania się kluczy innych klas w schemacie R_m. Liczba ta wynosi $\sum_{\substack{i=1 \\ i\neq m}}^{q} r_i$.

Powtórzenie tej operacji dla wszystkich klas równoważności kluczy daje wynik

$$L_1 = \sum_{m=1}^{q} \sum_{\substack{i=1 \\ i\neq m}}^{q} r_i = (q-1) \sum_{i=1}^{q} r_i .$$

Ponieważ liczba kluczy r_i w poszczególnych klasach, jak również liczba q klas równoważności nie może przekroczyć n, tzn. liczby zależności w zbiorze F, więc

$$L_1 \leqslant n^2(n-1)$$

Tak więc złożoność obliczeniowa (pesymistyczna) kroku (5.2) wynosi $O(n^3)$.

Złożoność kroku (5.3) określa liczba porównań lewych stron zależności redundancyjnych z kluczami odpowiednich klas równoważności. Ponieważ liczba zależności w zbiorze RED, jak również sumaryczna liczba kluczy nie może przekroczyć n, więc złożoność tego kroku wynosi $O(n^2)$.

Przechodząc do analizy złożoności kroków (5.4), (5.5) i (5.6) założymy, że do zbioru $H_r^m \cup RED_0^m$ należy l_m zależności prostych. Złożoność operacji wyznaczania domknięcia X^+ atrybutu X oznaczymy literą D (w rozdziale 3 wskazano, że pesymistyczna złożoność tej operacji wynosi $O(|F|)$). Złożoność wyszukiwania zależności redundancyjnych w każdej iteracji kroku (5.4) określa wtedy liczba $l_m \cdot D$. W i-tej iteracji kroków (5.5) i (5.6) wykonywanych jest odpowiednio i oraz $i \cdot l_m$ operacji porównywania zbiorów. Oznaczając przez i_m całkowitą liczbę iteracji wykonanych dla klasy K_m w krokach (5.4), (5.5) i (5.6), złożoność tych operacji określimy wyrażeniem

$$\sum_{i=1}^{i_m} (l_m \cdot D + i + i \cdot l_m) = i_m \cdot l_m \cdot D + (l_m + 1) \frac{i_m(i_m+1)}{2} .$$

Ola wszystkich klas równoważności kluczy otrzymamy

$$L_2 = \sum_{m=1}^{q} \left[i_m \cdot l_m \cdot D + (l_m + 1) \frac{i_m(i_m+1)}{2} \right] .$$

Oznaczmy przez c maksymalną wartość, jaką może osiągnąć liczba iteracji we wszystkich klasach równoważności, tzn. $i_m \leqslant c$, $m = 1,\ldots,q$.

Przypomnijmy przy tym poprzednie uwagi (punkt 4.4.2) wskazujące na to, że wystąpienie takich zależności lub zbiorów alternatywnie redundancyjnych, które muszą być wykrywane dopiero w krokach (5.4)-(5.6), jest przypadkiem bardzo rzadkim. W konsekwencji również mało prawdopodobne jest, aby liczba c przyjąć mogła dużą wartość.

Przyjmując $i_m \leqslant c$ otrzymamy

$$L_2 \leqslant c \cdot D \cdot \sum_{m=1}^{q} l_m + \frac{1}{2} c(c+1) \sum_{m=1}^{q} (l_m+1)$$

Ponieważ $\sum_{m=1}^{q} l_m \leqslant n$ oraz $q \leqslant n$, więc

$$L_2 \leqslant c \cdot n \cdot D + c(c+1)n.$$

Dla porównawczej oceny tego wyniku ze złożonością poprzednich kroków możemy rozważyć dwa przypadki:

(1) Mała wartość c w porównaniu z n . D. Wtedy złożoność rozważanych kroków (5.4)-(5.6) określa wyrażenie $O(n|F|)$.

(2) Duża wartość c, np. porównywalna z D. Wtedy analizowana złożoność określona może być wyrażeniem $O(n|F|^2)$.

Dla orientacyjnego porównania przedstawionych wyników można przyjąć - za pracą [70] - oszacowanie wielkości $|F|$ jako $O(a.n)$, gdzie a jest liczbą atrybutów występujących w zapisie n zależności tworzących zbiór F.

Złożoność poszczególnych kroków algorytmu 4.4 możemy wtedy wyrazić następująco:

(1)-(4): $O(a^2n^2)$,

(5.2)-(5.3): $O(n^3)$,

(5.4)-(5.6): $O(an^2)$ lub $O(a^2n^3)$ - dla wybranych przypadków.

Wielkości a i n nie są ze sobą jawnie związane, chociaż we wszystkich rozważanych przykładach są one podobnego rzędu.

4.5. OCENA SCHEMATU RELACYJNEJ BAZY DANYCH W PROCESIE AUTOMATYCZNEGO ROZWIĄZYWANIA ZADAŃ WYSZUKIWANIA DANYCH

W rozdziale 4.1 sformułowane zostało zagadnienie oceny schematu relacyjnej bazy danych poprzez wyznaczenie liczby łączeń relacji potrzebnych do rozwiązania określonych zadań wyszukiwania danych. Rozwiązywanie takich zadań oparte jest w większości języków manipulowania danymi [26, 52] (ang. Query Languages) na wskazaniu w rozpatrywanej bazie danych wszystkich relacji, które mają brać udział w rozwiązaniu. Podejście to prezentowane jest w językach prostszych opartych na algebrze relacji (np. ISBL [70, 86]), a także językach wyższego poziomu bazujących na rachunku krotek relacji (np. ALPHA [26, 70], QUEL [34, 82, 89]), rachunku dziedzin relacji (np. QBE [26, 92]) lub innych (np. SEQUEL [8, 26, 53]).

Tak więc ograniczenie się do metod związanych z wymienionymi językami musiałoby się sprowadzić do indywidualnego rozwiązania przez projektanta każdego zadania dla każdego ocenianego schematu relacyjnej bazy danych (a przynajmniej do wskazania relacji biorących udział w rozwiązaniu). Chociaż to podejście zapewniłoby najbardziej adekwatną ocenę badanych schematów, to pracochłonność tego procesu kwestionowałaby jego użyteczność.

W rozdziale tym rozpatrzmy więc problem automatycznego wyboru relacji koniecznych do rozwiązania zadania.

Załóżmy w tym celu, że dla bazy danych r o schemacie $\mathbb{R} = \{R_1,\dots,R_n\}$ określone jest zadanie (pytanie) Z dotyczące zbioru atrybutów X, gdzie $X \subseteq \bigcup_{R_k \in \mathbb{R}} R_k$.

Należy wskazać takie relacje $r_{i_1}, r_{i_2}, \dots, r_{i_k}$ należące do $\tau(\mathbb{R})$, których połączenie konieczne jest do uzyskania odpowiedzi, tzn. relacji

$\pi_X(r_{i_1} \bowtie r_{i_2} \bowtie \dots \bowtie r_{i_k})$ zapisywanej też w postaci

$$(r_{i_1} \bowtie r_{i_2} \bowtie \dots \bowtie r_{i_k})[X]. \qquad (4.5.1)$$

Tak postawione zadanie analizowane było w wielu pracach (np. [9, 41, 72, 73, 84, 90]), przy czym wyrażenie (4.5.1) było różnie nazywane (ang. relationship [9], connection [72, 73]). W niniejszej pracy przyjmiemy określenie: zbiór relacji wiążących atrybuty zbioru X.

Należy dodać, że relacja (4.5.1) nie reprezentuje jeszcze, w ogólnym przypadku, pełnej odpowiedzi na pytanie postawione w zadaniu Z. W wyrażeniu (4.5.1) pominięto bowiem ewentualną operację selekcji krotek. Nie ma to jednak wpływu na możliwość wyznaczenia przyjętego kryterium oceny, którym jest liczba łączonych relacji. Wynika stąd, że zamiast zbioru zadań wyszukiwania danych $Z = \{z_1,\dots,z_n\}$ możemy rozważać rodzinę zbiorów atrybutów $\mathcal{X} = \{X_1,\dots,X_{n_z}\}$, których dotyczą poszczególne zadania.

Zagadnienie wyboru zbioru relacji wiążących atrybuty zbioru X było początkowo jednym z problemów rozpatrywanych w dziedzinie tzw. acyklicznych schematów baz danych, później zaś stało się centralnym zagadnieniom relacji uniwersalnej.

Rozmiar zagadnień dotyczących acyklicznych baz danych i relacji uniwersalnej (np. analiza założeń tej relacji [73, 50, 68, 9, 43] lub problem operowania na pustych wartościach atrybutów [25, 20, 66] nie pozwala na ich szczegółową prezentację w niniejszej pracy.

Przedstawimy zatem tylko te pojęcia, założenia i wnioski, które pozwolą na wykorzystanie tego aparatu do oceny schematów relacyjnych baz danych tworzonych drogą syntezy.

4.5.1. Wykorzystanie hipergrafów do opisu schematu relacyjnej bazy danych

Podstawowym modelem stosowanym w omawianym podejściu do prezentacji schematu relacyjnej bazy danych jest hipergraf [17, 16, 41, 11].

Hipergrafem $\mathcal{H}$ jest para $(\mathcal{N}, \mathcal{E})$, gdzie $\mathcal{N}$ jest zbiorem wierzchołków, a $\mathcal{E}$ jest zbiorem hiperkrawędzi (krawędzi). Hiperkrawędzie E_i są arbitralnie ustalonymi, niepustymi podzbiorami $\mathcal{N}$, zachodzi przy tym równość

$$\bigcup_{E_i \in \mathcal{E}} E_i = \mathcal{N}.$$

Zwykły nieskierowany graf bez pętli jest hipergrafem. Każda krawędź tego grafu jest wtedy hiperkrawędzią o dwóch wierzchołkach.

Pojęcie hipergrafu posłużyć może bezpośrednio do ilustracji schematu relacyjnej bazy danych. Załóżmy, że schemat taki określony jest następująco: $\mathbb{R} = \{R_1, R_2, \dots, R_n\}$. Schematowi $\mathbb{R}$ możemy przyporządkować hipergraf, którego zbiorem wierzchołków jest zbiór atrybutów tworzących sche-

maty relacji, tzn. $\mathcal{R} = \bigcup_{i=1}^{n} R_i$. Zbiorem hiperkrawędzi jest natomiast zbiór schematów relacji $\{R_1, R_2, \ldots, R_n\}$.

Oznaczając taki hipergraf również przez $\mathbb{R}$ będziemy często utożsamiali w tym rozdziale schemat relacyjnej bazy danych z jego hipergrafem.

Przykład 4.5.1. Rozpatrzmy bazę danych obejmującą zbiór atrybutów (patrz przykład 2.1.1) {student, kierunek, wydział, przedmiot, ocena}. Przyjmując istnienie następujących zależności funkcjonalnych między tymi atrybutami:

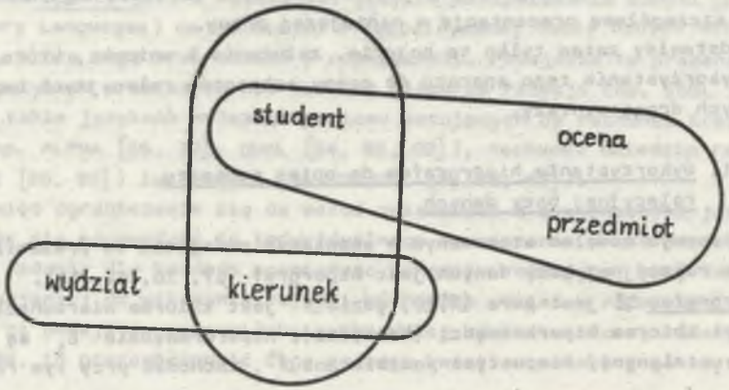
student ⟶ kierunek,
student, przedmiot ⟶ ocena,
kierunek ⟶ wydział,

możemy utworzyć następujące schematy relacji

R_1 = {student, kierunek},
R_2 = {kierunek, wydział},
R_3 = {student, przedmiot, ocena}.

Tworzą one schemat relacyjnej bazy danych $\mathbb{R} = \{R_1, R_2, R_3\}$.

Graficzną ilustrację hipergrafu odpowiadającego schematowi $\mathbb{R}$ przedstawia rys. 4.5.1. □



Rys. 4.5.1. Postać hipergrafu odpowiadającego schematowi $\mathbb{R}$
Fig. 4.5.1. Form of hypergraph for the schema $\mathbb{R}$

Zdefiniujemy jeszcze dwa pojęcia dotyczące hipergrafów:

Drogą od wierzchołka s do wierzchołka t nazywamy [41] ciąg hiperkrawędzi $E_1, E_2, \ldots, E_k$ taki, że

$$s \in E_1,\ t \in E_k,\ E_i \cap E_{i+1} \neq \{\emptyset\} \text{ dla } 1 \leq i \leq k. \qquad (4.5.2)$$

Powyższy ciąg jest też nazywany drogą od krawędzi E_1 do krawędzi E_k.

Dwa wierzchołki lub dwie hiperkrawędzie są spójne, jeśli istnieje droga między nimi.

Zbiór wierzchołków lub krawędzi jest spójny, jeśli każda jego para jest spójna.

Zakładać będziemy, że rozpatrywane dalej hipergrafy są spójne.

Nawiązując do zadania określonego zależnością (4.5.1) zauważmy, że każdym dwóm łączonym relacjom odpowiadają hiperkrawędzie sąsiednie (posiadające niepustą część wspólną) należące do pewnej drogi w hipergrafie reprezentującym określoną bazę danych.

Problem wyboru drogi w hipergrafie prowadzi w ogólnym przypadku do konieczności rozpatrzenia zagadnień cykliczności hipergrafów.

4.5.2. Hipergrafy acykliczne

Przykład 4.5.2. Rozpatrzmy bazę danych $\mathbf{r} = \{r_i(R_i)\}$, $i = 1, \ldots, 4$, o schemacie $\mathbb{R} = \{R_1, R_2, R_3, R_4\} = \{\{\text{pracownik, wydział}\}, \{\text{pracownik, przedmiot}\}, \{\text{student, wydział}\}, \{\text{student, przedmiot}\}\}$.

Rozważmy następujące zadanie wyszukiwania sformułowane dla tej bazy: "Podaj wykaz przedmiotów dotyczących wydziału W". Zbiór atrybutów X charakteryzujących to zadanie ma postać $X = \{\text{przedmiot, wydział}\}$.

Przeanalizujemy trzy różne rozwiązania postawionego zadania:

(1) $\Pi_{\text{przedmiot}}\,((\sigma_{\text{wydział}=W}\,(r_1)) \bowtie r_2)$.

(2) $\Pi_{\text{przedmiot}}\,((\sigma_{\text{wydział}=W}\,(r_3)) \bowtie r_4)$.

(3) $\Pi_{\text{przedmiot}}\,((\sigma_{\text{wydział}=W}\,(r_1)) \bowtie r_2 \bowtie r_4 \bowtie (\sigma_{\text{wydział}=W}\,(r_3)))$.

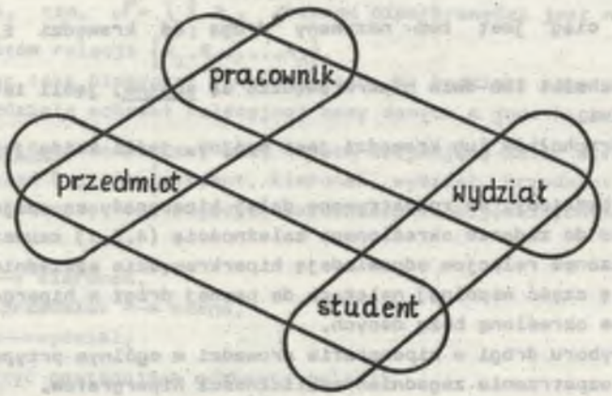
Zauważmy, że rozwiązania te podają:

(1) wykaz przedmiotów prowadzonych przez pracowników wydziału W,
(2) wykaz przedmiotów, na które uczęszczają studenci wydziału W,
(3) wykaz przedmiotów prowadzonych przez pracowników wydziału W, na które uczęszczają tylko studenci wydziału W. □

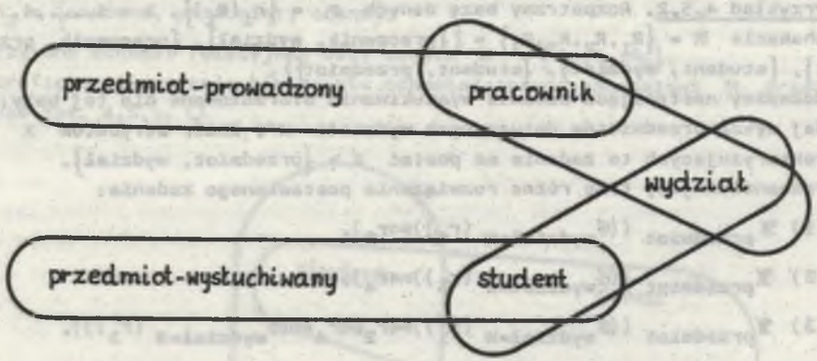
Możliwość uzyskania wielu różnych rozwiązań powyższego zadania wynika z faktu, że hipergraf utworzony dla schematu przedstawionej bazy danych (rys. 4.5.2) jest hipergrafem cyklicznym.

Zdefiniowanie własności cykliczności i acykliczności oraz sformułowanie warunków ich wystąpienia wymaga wprowadzenia szeregu pojęć.

Przed prezentacją tych pojęć zauważmy jeszcze, że przedstawionych w przykładzie 4.5.2 problemów można uniknąć zmieniając (a właściwie "rozszczepiając") nazwę atrybutu "przedmiot" na dwie nazwy, na przykład "przedmiot prowadzony" i "przedmiot wysłuchiwany". Zmieni to hipergraf tej bazy do postaci przedstawionej na rys. 4.5.3.



Rys. 4.5.2. Hipergraf utworzony dla schematu bazy danych z przykładu 4.5.2
Fig. 4.5.2. Hypergraph created for database schema from example 4.5.2



Rys. 4.5.3. Acykliczny hipergraf dla zmodyfikowanego schematu bazy danych z przykładu 4.5.2
Fig. 4.5.3. Acyclic hypergraph for modificated database schema from example 4.5.2

Konieczne będzie również przeformułowanie postawionego zadania poprzez uściślenie atrybutu "przedmiot", co pozwoli wybrać jednoznaczne rozwiązanie.

W literaturze wprowadzono kilka różnych definicji cykliczności i acykliczności (np. [16, 17, 41]). Klasyczną definicją jest definicja Berge'a [17].

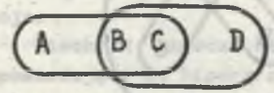
Cyklem Berge'a w hipergrafie $\mathcal{H}$ jest ciąg $(S_1,x_1,S_2,x_2,\ldots,S_m,x_m,S_{m+1})$ taki, że

(1) $x_1,x_2,\ldots,x_m$ są różnymi wierzchołkami w $\mathcal{H}$,

(2) $S_1,S_2,\ldots,S_m$ są różnymi krawędziami w $\mathcal{H}$ oraz $S_{m+1} = S_m$,

(3) $m \geq 2$, tzn. do cyklu wchodzą co najmniej dwie hiperkrawędzie,

(4) x_i należy do S_i oraz do S_{i+1} dla $1 \leq i \leq m$.



Rys. 4.5.4. Hipergraf cykliczny w sensie Berge'a
Fig. 4.5.4. Berge - cyclic hypergraph

Hipergraf jest cykliczny w sensie Berge'a, jeśli posiada cykl Berge'a, inaczej jest acykliczny.

Przydatność definicji Berge'a w prowadzonych rozważaniach jest problematyczna z uwagi na to, że już w hipergrafie przedstawionym na rys.4.5.4 wskazać można następujący cykl

(ABC,C,BCD,B,ABC),

gdzie podkreślone grupy symboli oznaczają krawędzie hipergrafu.

A więc jeśli w hipergrafie istnieje jakaś para krawędzi taka, że ich przecięcie zawiera co najmniej 2 atrybuty, to hipergraf jest cykliczny w sensie Berge'a.

Spośród innych definicji acykliczności wymienić można np. α-acykliczność [16, 70] czy β- acykliczność [41].

W niniejszej pracy wykorzystamy definicję γ-acykliczności [41], przydatną z uwagi na założenia przyjmowane w procesie wyznaczania zbioru relacji wiążących zbiór atrybutów X.

W danym hipergrafie $\mathcal{H}$ γ-cyklem [41] nazywany jest ciąg

$$(S_1,x_1,S_2,x_2,\ldots,S_m,x_m,S_{m+1})$$

taki, że

(1) $x_1,\ldots,x_m$ są różnymi wierzchołkami hipergrafu $\mathcal{H}$,

(2) $S_1,\ldots,S_m$ są różnymi krawędziami oraz $S_{m+1} = S_1$,

(3) $m \geq 3$, tzn. w cięgu występują co najmniej 3 krawędzie,

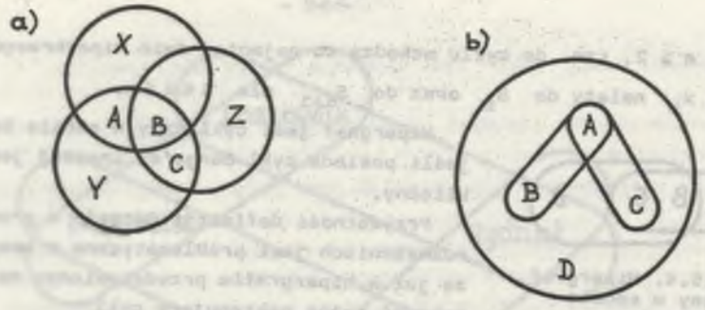
(4) $x_i \in S_i \wedge x_i \in S_{i+1}$, $1 \leq i \leq m$ oraz

(5) dla $1 \leq i \leq m$ każde x_i nie należy do żadnego S_j różnego od S_i oraz S_{i+1}.

Schemat bazy danych $\mathbb{R}$ jest γ-acykliczny, jeśli odpowiadający mu hipergraf jest też γ-acykliczny, tzn. nie można w nim utworzyć żadnego γ-cyklu.

Przykład 4.5.3. Przedstawiony na rys. 4.5.4 hipergraf cykliczny w sensie Berge'a jest hipergrafem γ-acyklicznym (ponieważ zawiera tylko dwie krawędzie).

Przykłady hipergrafów γ-cyklicznych przedstawia rysunek 4.5.5.



Rys. 4.5.5. Przykłady hipergrafów γ-cyklicznych
Fig. 4.5.5. Examples of γ-cyclic hypergraphs

W przypadku (a) cyklem jest ciąg

(XAB, A, ABCY, C, BCZ, B, XAB),

natomiast w przypadku (b) cykl ma postać

(AC, C, ABCD, B, AB, A, AC),

przy czym krawędziami w powyższych zapisach są podkreślone zbiory atrybutów. □

W pracy [41] przedstawiony został następujący algorytm testowania γ-acykliczności hipergrafu:

Algorytm 4.5. (Testowanie γ-acykliczności hipergrafu)

Dane wejściowe: hipergraf $\mathcal{H}$.
Dane wyjściowe: wynik testu.

Wykonuj na hipergrafie $\mathcal{H}$ w dowolnym porządku następujące operacje tak długo, aż żadna z nich nie będzie już mogła być wykonana:

(1) Usuń wierzchołek izolowany (należący tylko do jednej krawędzi).
(2) Usuń krawędź zawierającą tylko jeden wierzchołek (nie usuwając tego wierzchołka, jeśli należy też do innej krawędzi).
(3) Usuń krawędź pustą.
(4) Jeśli dwie krawędzie zawierają dokładnie te same wierzchołki, wtedy usuń jedną z nich.
(5) Jeśli para wierzchołków występuje w tych samych krawędziach, to wtedy usuń jeden z tych wierzchołków.

Hipergraf $\mathcal{H}$ jest hipergrafem γ-acyklicznym, jeśli wynikiem powyższych operacji jest pusty zbiór krawędzi. □

Złożoność powyższego algorytmu oceniono w pracy [41] jako złożoność wielomianową.

4.5.3. Wyznaczanie relacji wiążących atrybutu zbioru X dla γ-acyklicznego hipergrafu relacyjnej bazy danych

Teoretyczne podstawy prezentowanej metody zostały sformułowane w pracach [41, 90]. Ich omówienie rozpoczniemy od wprowadzenia szeregu definicji.

Niech $\mathbb{R}$ oznacza równocześnie schemat relacyjnej bazy danych oraz odpowiadający mu hipergraf. Hipergrafem Bachmana [41] dla $\mathbb{R}$ jest hipergraf, którego krawędziami są wszystkie krawędzie $\mathbb{R}$, a ponadto wszystkie podzbiory wierzchołków $\mathbb{R}$ będące przecięciami dwóch lub więcej krawędzi hipergrafu $\mathbb{R}$. Diagramem Bachmana [41, 64] dla $\mathbb{R}$ jest graf nieskierowany, którego wierzchołki utożsamiane są z krawędziami hipergrafu Bachmana. Dwa wierzchołki S i T tego diagramu łączone są krawędzią wtedy i tylko wtedy, jeśli $S \subsetneq T$, a przy tym w hipergrafie Bachmana nie istnieje taka krawędź W, że $S \subsetneq W \subsetneq T$. (Diagram Bachmana może być również definiowany jako graf skierowany [90]; wtedy dla powyższych określeń łuk łączący S i T skierowany jest od S do T). W pracy [41] wykazano, że jeśli $\mathbb{R}$ jest hipergrafem γ-acyklicznym, to jego diagram Bachmana jest grafem acyklicznym (lub drzewem dla drugiej definicji).

Niech X będzie podzbiorem wierzchołków hipergrafu $\mathbb{R}$ (tzn. podzbiorem atrybutów schematu $\mathbb{R}$). Oznaczmy przez $\mathbf{V}$ spójny zbiór k różnych krawędzi hipergrafu Bachmana dla $\mathbb{R}$:

$$\mathbf{V} = \{V_1, V_2, \ldots, V_k\}.$$

$\mathbf{V}$ jest nazywany [41] minimalnym zbiorem krawędzi wiążących wierzchołki podzbioru X, jeśli:

(1) $X \subseteq V_1 \cup V_2 \cup \ldots \cup V_k$ oraz

(2) w każdym spójnym podzbiorze $\mathbf{W} = \{W_1, \ldots, W_p\}$ krawędzi hipergrafu Bachmana takim, że $X \subseteq W_1 \cup \ldots \cup W_p$, istnieje k różnych krawędzi $W_{i_1}, W_{i_2}, \ldots, W_{i_k}$ (należących do $\mathbf{W}$) takich, że $V_j \subseteq W_{i_j}$ dla $1 \leqslant j \leqslant k$.

W pracy [41] wykazano, że jeśli diagram Bachmana dla $\mathbb{R}$ jest grafem acyklicznym (tzn. jeśli $\mathbb{R}$ jest γ-acykliczny), to $\mathbf{V}$ może zostać utworzone w następujący sposób:

Niech $\mathbf{V}' = \{R_{i_1}, R_{i_2}, \ldots, R_{i_q}\}$ oznacza zbiór wierzchołków najmniejszego spójnego podgrafu Bachmana dla $\mathbb{R}$, zawierającego X, tzn. $X \subseteq R_{i_1} \cup \ldots \cup R_{i_q}$. Wtedy $\mathbf{V}$ jest zbiorem takich $R_{i_j} \in \mathbf{V}'$, dla których nie istnieje żadne p $(1 \leqslant p \leqslant q)$ takie, że $R_{i_j} \subsetneq R_{i_p}$.

Elementy tworzące $\mathbf{V}$ są nazywane maksymalnymi elementami podgrafu określonego przez $\mathbf{V}'$.

W pracy [41] wskazano na jednoznaczność wyboru $\mathbf{V}$, jeśli $\mathbb{R}$ jest hipergrafem γ-acyklicznym.

Zauważmy, że podzbiory atrybutów tworzące $\mathbb{V}$ mogą być:

(a) schematami relacji $R_i \in \mathbb{R}$ lub

(b) częściami wspólnymi dwóch lub więcej schematów należących do $\mathbb{R}$.

Poszukiwany zbiór relacji wiążących atrybuty X będzie więc odpowiednio obejmował:

(a) relacje $r_i(R_i)$ należące do bazy danych o schematach R_i wchodzących do $\mathbb{V}$,

(b) nowo utworzone relacje o schematach $S \in \mathbb{V}$, nie należących do $\mathbb{R}$.

Każdą taką relację s tworzymy [41, 90] jako sumę rzutów na S wszystkich relacji z $\mathbf{r}(\mathbb{R})$, których schematy zawierają S, tzn.

$$s(S) = \bigcup \{ r(R)[S] : r \in \mathbf{r}(\mathbb{R}) \wedge S \subseteq R \}.$$

Ostatecznie zdefiniowane na wstępie zadanie (4.5.1) zapiszemy teraz następująco:

$$(\bowtie \mathbb{V})[X],$$

gdzie $\mathbb{V}$ jest zbiorem relacji wiążących atrybuty X, to znaczy obejmującym relacje o schematach z $\mathbb{V}$, określone w punktach (a) i (b).

Zauważmy na koniec, że poszukiwana liczba połączeń relacji w zadaniu (4.5.1) wynosi

$$p = |\mathbb{V}| - 1,$$

gdzie $|\mathbb{V}|$ jest liczbą elementów zbioru $\mathbb{V}$.

Przykład 4.5.5. Rozpatrzmy bazę danych obejmującą następujące atrybuty:

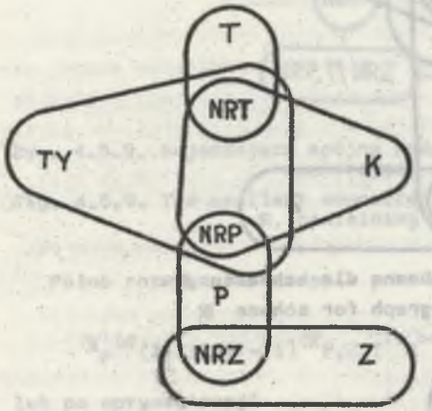
P - nazwisko i imię pracownika,
NRP - numer pracownika,
Z - nazwa zespołu,
NRZ - numer zespołu,
T - temat projektu,
NRT - numer projektu,
K - kwota zarobiona przez określonego pracownika przy realizacji określonego projektu,
TY - tytuł publikacji dotyczącej określonego projektu, której autorem (bądź współautorem) jest określony pracownik.

Będziemy zakładali, że:

- dany pracownik może brać udział w realizacji kilku projektów,
- efektem udziału pracownika w realizacji projektu może być określone honorarium (kwota zarobiona) i/lub określona publikacja.

Przyjmijmy, że w rozważanej bazie istnieją relacje o następujących schematach:

$R_1 = \{P, NRP, NRZ\}$,
$R_2 = \{Z, NRZ\}$,
$R_3 = \{T, NRT\}$,
$R_4 = \{NRP, NRT, K\}$,
$R_5 = \{NRP, NRT, TY\}$.



Rys. 4.5.6. Hipergraf $\mathbb{R}$ dla bazy danych z przykładu 4.5.5

Fig. 4.5.6. Hypergraph $\mathbb{R}$ for database from example 4.5.5

Schemat bazy danych przedstawiony został w postaci hipergrafu na rys. 4.5.6. Łatwo sprawdzić, że hipergraf ten jest hipergrafem γ-acyklicznym.

Załóżmy, że dla tak zdefiniowanej bazy określimy następujące zadanie wyszukiwania danych:
Podać nazwiska tych pracowników zespołu o nazwie Z1, którzy brali udział w realizacji projektu o nazwie T1.
Zbiór X obejmuje następujące atrytuty $X = \{P, Z, T\}$.

Postępując według przedstawionej metody utworzymy dla hipergrafu $\mathbb{R}$ odpowiadający mu hipergraf Bachmana (rys. 4.5.7) oraz diagram Bachmana (rys. 4.5.8).

Najmniejszy spójny podgraf diagramu Bachmana zawierający X przedstawia rys. 4.5.9.

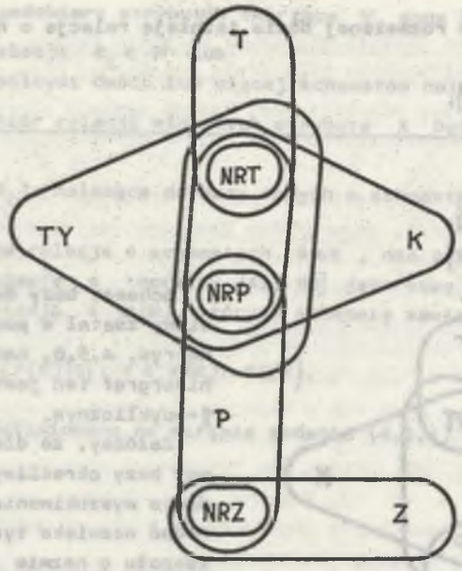
Zbiór $\mathbb{V}$ obejmuje "największe" wierzchołki tego podgrafu, a więc będzie się składał z następujących podzbiorów atrybutów:

$$\mathbb{V} = \{\{T, NRT\}, \{NRT, NRP\}, \{NRP, P, NRZ\}, \{NRZ, Z\}\}.$$

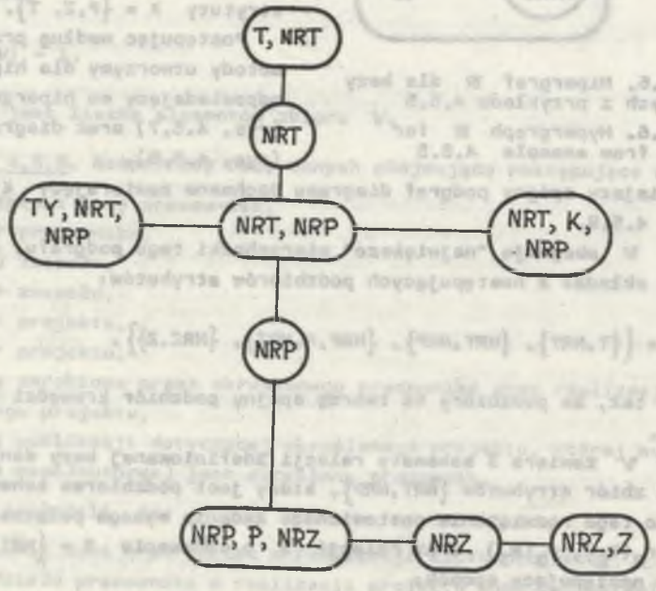
Zauważmy też, że podzbiory te tworzą spójny podzbiór krawędzi hipergrafu Bachmana.

Zbiór $\mathbb{V}$ zawiera 3 schematy relacji zdefiniowanej bazy danych: R_3, R_1, R_2 oraz zbiór atrybutów $\{NRT, NRP\}$, który jest podzbiorem schematów R_4 i R_5. Wobec tego rozwiązanie postawionego zadania wymaga połączenia relacji $r_1(R_1)$, $r_2(R_2)$, $r_3(R_3)$ oraz relacji s o schemacie $S = \{NRT, NRP\}$ utworzonej w następujący sposób:

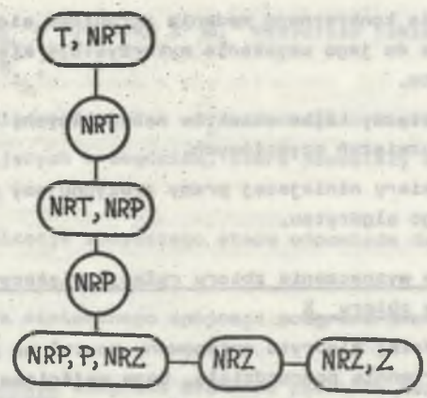
$$s(S) = r_4(R_4)\,[NRT, NRP] \cup r_5(R_5)\,[NRT, NRP].$$



Rys. 4.5.7. Hipergraf Bachmana dla schematu $\mathbb{R}$
Fig. 4.5.7. Bachman hypergraph for schema $\mathbb{R}$



Rys. 4.5.8. Diagram Bachmana dla schematu $\mathbb{R}$
Fig. 4.5.8. Bachman diagram for schema $\mathbb{R}$



Rys. 4.5.9. Najmniejszy spójny podgraf diagramu Bachmana dla schematu $\mathbb{R}$ zawierający podzbiór atrybutów X
Fig. 4.5.9. The smallest connected subgraph of Bachman diagram for schema $\mathbb{R}$, containing subset of attributes X

Pełne rozwiązanie zadania przedstawić można następująco

$$\pi_P(\sigma_{(Z=Z1)\wedge(T=T1)}(\pi_{P,Z,T}(r_1 \bowtie r_2 \bowtie r_3 \bowtie s)))$$

lub po optymalizacji

$$\pi_P((\pi_{NRZ}(\sigma_{Z=Z1}(r_2))) \bowtie r_1 \bowtie s \bowtie (\pi_{NRT}(\sigma_{T=T1}(r_3)))).$$

W obu przypadkach liczba połączeń relacji - stanowiąca kryterium oceny - wynosi 3. □

Przedstawiona metoda wyznaczania połączenia pomiędzy zbiorem atrybutów X ograniczona jest do acyklicznych baz danych. Taka struktura baz danych wydaje się być najbardziej zalecana z uwagi na jednoznaczność powiązań między atrybutami. Założenie acykliczności bądź też dążenie do acykliczności było też krytykowane (np. [9]), przy czym argumentem był fakt, że w bardzo dużych bazach danych liczne powiązania między danymi w naturalny sposób prowadzą do powstania cykli.

W przypadku cyklicznych baz danych postawione na wstępie zadanie wyznaczania związku dla zbioru X znalazło swoje rozwiązanie w ramach relacji uniwersalnej.

Charakteryzując to rozwiązanie w dużym skrócie i uproszczeniu można wyróżnić w nim następujące etapy

- W hipergrafie reprezentującym schemat bazy danych wyróżnione są α-acykliczne fragmenty (tzw. obiekty maksymalne) interesujące dla rozwiązania określonych klas problemów.

- Jeśli rozwiązanie konkretnego zadania ogranicza się do jednego obiektu maksymalnego, to do jego uzyskania wykorzystuje się relacje wchodzące do danego obiektu.
- Jeśli zadanie dotyczy kilku obiektów maksymalnych, to końcowe rozwiązanie jest sumą rozwiązań częściowych.

Z uwagi na rozmiary niniejszej pracy zrezygnujemy ze szczegółowej prezentacji powyższego algorytmu.

4.5.4. Algorytm wyznaczania zbioru relacji wiążących atrybuty zbioru X

Przedstawiony dalej algorytm opracowany został na podstawie metody naszkicowanej w poprzednim podrozdziale. Dane wejściowe dla tego algorytmu stanowią: schemat relacyjnej bazy przedstawiany jako hipergraf $\mathbb{R}$ oraz zadany zbiór atrybutów X. Będziemy zakładali, że $\mathbb{R}$ jest hipergrafem spójnym, γ-acyklicznym. Końcowym wynikiem algorytmu będzie zbiór relacji o schematach w zbiorze V wiążących atrybuty X.

Algorytm przedstawiony zostanie w kilku oddzielnie komentowanych etapach.

Algorytm 4.6. (Wyznaczanie zbioru relacji wiążących atrybuty X)

Etap 1. (Tworzenie hipergrafu Bachmana)

Dane wejściowe: hipergraf $\mathbb{R} = \{R_1,\dots,R_q\}$.
Dane wyjściowe: hipergraf Bachmana $\mathbb{B} = \{B_1,\dots,B_m\}$.

(1) Włącz do zbioru krawędzi hipergrafu Bachmana wszystkie krawędzie hipergrafu $\mathbb{R}$.

(2) Dla każdej krawędzi hipergrafu $\mathbb{R}$ i dla każdej krawędzi tworzonego hipergrafu Bachmana sprawdź, czy istnieje niepusta część wspólna tych krawędzi. Jeśli tak, włącz tę część wspólną jako nową krawędź do hipergrafu Bachmana, o ile w hipergrafie tym nie ma jeszcze takiej krawędzi. □

Zauważmy, że szukanie części wspólnych krawędzi R_i oraz poprzednio utworzonych krawędzi B_j pozwala utworzyć w $\mathbb{B}$ krawędzie będące częściami wspólnymi dowolnej liczby hiperkrawędzi z $\mathbb{R}$.

Etap 2. (Tworzenie diagramu Bachmana)

Dane wejściowe: hipergraf Bachmana $\mathbb{B} = \{B_1,\dots,B_m\}$.
Dane wyjściowe: diagram Bachamana D.

(1) Utwórz zbiór wierzchołków $D = \{D_1,\dots,D_m\}$ przypisując każdemu D_i zbiór atrybutów należących do krawędzi B_i $(1 \leq i \leq m)$.

(2) Dla każdego $B_i \in \mathbb{B}$ wykonaj następujące operacje

(2.1) Utwórz taki podzbiór $\mathbb{B}_i = \{B_{i_1},\dots,B_{i_l}\}$, $B_{i_j} \in \mathbb{B}$, $j = 1,\dots,l$, że dla każdego $B_{i_j} \in \mathbb{B}_i$ zachodzi $B_k \subset B_{i_j}$.

(2.2) Dla każdego $B_{i_j} \in \mathbb{B}_i$ usuń z $\mathbb{B}_i$ wszystkie takie $B_{i_n} \in \mathbb{B}_i$, dla których $B_{i_j} \subset B_{i_n}$.

(2.3) Połącz krawędzią wierzchołek D_i z każdym z wierzchołków $D_{i_1},\dots,D_{i_l}$ odpowiadających krawędziom, które pozostały w zbiorze $\mathbb{B}_i$ po kroku (2.2). □

Zauważmy, że realizacja powyższego etapu odpowiada definicji diagramu Bachmana.

Etap 3. (Tworzenie minimalnego spójnego podgrafu Bachmana zawierającego X)

Wierzchołkiem końcowym diagramu Bachmana będziemy nazywali każdy wierzchołek, który jest incydentny z tylko jedną krawędzią tego diagramu.

Dane wejściowe: diagram Bachmana o wierzchołkach $D = \{D_1,\dots,D_m\}$ i związane z tymi wierzchołkami krawędzie $B_1,\dots,B_m$ hipergrafu Bachmana.

Dane wyjściowe: minimalny spójny podgraf D_X diagramu Bachmana, zawierający X.

(1) Usuń wierzchołek końcowy D_i, jeśli $B_i \cap X = \{\emptyset\}$ lub $B_i \cap X \subseteq B_{sąsiad[i]}$, gdzie $B_{sąsiad[i]}$ jest krawędzią związaną z (jedynym) wierzchołkiem diagramu D połączonym krawędzią z wierzchołkiem D_i.

(2) Powtarzaj krok (1) tak długo, aż żaden z wierzchołków końcowych nie będzie już mógł być usunięty. □

Lemat 4.4. Załóżmy, że dla γ-acyklicznego hipergrafu utworzony został hipergraf Bachmana $\mathbb{B}$ oraz diagram Bachmana D. Jeśli dla dwóch wierzchołków D_{i_1} i D_{i_n} diagramu D odpowiadające im hiperkrawędzie B_{i_1} i B_{i_n} hipergrafu mają niepustą część wspólną $B_{i_1} \cap B_{i_n} = W \neq \{\emptyset\}$, to na drodze łączącej D_{i_1} i D_{i_n} istnieje taki wierzchołek D_{i_k}, $1 \leq k \leq n$, że zachodzi

$$B_{i_1} \supset \dots \supset B_{i_k} \subset \dots \subset B_{i_n}. \tag{4.5.3}$$

D o w ó d. Z zasad konstrukcji hipergrafu Bachmana wynika, że jeśli dwie hiperkrawędzie B_{i_1} oraz B_{i_n} mają niepustą część wspólną $B_{i_1} \cap B_{i_n} = W$, to w hipergrafie $\mathbb{B}$ musi też istnieć krawędź $B_{i_k} = W$ (nie wykluczamy przy tym możliwości, że $B_{i_k} = B_{i_1}$ lub $B_{i_k} = B_{i_n}$). Ponieważ dla γ-acyklicznego hipergrafu $\mathbb{R}$ diagram Bachmana jest drzewem, więc z zasad konstrukcji tego diagramu wynika, że jedyne w takim przypadku połączenie wierzchołka D_{i_k} z wierzchołkami D_{i_1} oraz D_{i_n} odpowiada ciągowi inkluzji (4.5.3). □

Nazwijmy wierzchołkiem zbędnym diagramu Bachmana taki wierzchołek końcowy D_i, dla którego spełniony jest jeden z następujących warunków:

(1) atrybuty $B_i \cap X$ zawierają się w innym zbiorze (krawędzi) B_j (związanym z wierzchołkiem D_j) lub

(2) atrybuty $B_i \cap X$ rozproszone są w zbiorach związanych z kilkoma wierzchołkami.

Lemat 4.5. W etapie 3 algorytmu 4.6 tworzony jest spójny podgraf zawierający X, w którym żaden wierzchołek nie jest wierzchołkiem zbędnym.

D o w ó d. Zauważmy najpierw, że jeśli hipergraf $\mathbb{R}$ jest hipergrafem spójnym, to spójny jest też diagram Bachmana utworzony dla tego hipergrafu. Ponieważ w powyższym algorytmie usuwane są tylko wierzchołki końcowe, więc podgraf D_X też będzie grafem spójnym. Wynika stąd również, że zbędnym wierzchołkiem mógłby być w podgrafie D_X jedynie wierzchołek końcowy, usunięcie innego naruszyłoby bowiem spójność D_X.

W dowodzie rozpatrzmy odrębnie oba przypadki wyróżnione w definicji wierzchołka zbędnego.

Ad (1). Jeśli $B_i \cap X \subseteq B_j$, to $B_i \cap B_j \neq \{\emptyset\}$. Z lematu 4.4 wynika wtedy, że w podgrafie D_X musi - obok wierzchołka D_j - istnieć taki wierzchołek D_k, dla którego zachodzi

$$B_i \supset B_s \supset \ldots \supset B_k \subset \ldots \subset B_j. \qquad (4.5.4)$$

gdzie B_s jest krawędzią związaną z wierzchołkiem D_s sąsiadującym (połączonym krawędzią) z wierzchołkiem D_i, natomiast $B_k = B_i \cap B_j$. W szczególności wierzchołek D_k może być równy wierzchołkowi D_i lub też D_j, co prowadzi do zależności:

$$B_i \subset B_s \subset \ldots \subset B_j \qquad (4.5.5)$$

lub

$$B_i \supset B_s \supset \ldots \supset B_j. \qquad (4.5.6)$$

Wnioski wynikające z analizy powyższych przypadków zapiszemy w skrócie następująco:

Ad (4.5.4) $(B_i \cap X \subseteq B_j) \wedge (B_k = B_i \cap B_j) \models B_i \cap X \subseteq B_k$,

$$(B_i \cap X \subseteq B_k) \wedge (B_k \subset B_s) \models B_i \cap X \subset B_s.$$

Ad (4.5.5) $(B_i \subset B_s) \models B_i \cap X \subset B_s$.

Ad (4.5.6) $(B_i \cap X \subseteq B_j) \wedge (B_j \subset B_s) \models B_i \cap X \subset B_s$.

Ponieważ wierzchołek D_s jest wierzchołkiem sąsiednim wierzchołka końcowego D_i, więc $B_s = B_{\text{sąsiad}[i]}$. Wynika stąd, że w każdym z powyższych przypadków spełniony jest warunek $B_i \cap X \subset B_{\text{sąsiad}[i]}$ określony w kroku (1) rozważanego etapu 3 algorytmu 4.6. Oznacza to, że wierzchołek D_i powinien był zostać usunięty w trakcie realizacji etapu 3 algorytmu 4.6.

Ad (2). Załóżmy, że atrybuty tworzące podzbiór $B_i \cap X$ rozproszone są ponadto w dwóch zbiorach B_l i B_j związanych z wierzchołkami D_l i D_j, należącymi do podgrafu D_X. Słuszne są przy tym następujące zależności:

$$B_i \cap X = X_1 \cup X_2,\ X_1 \subseteq B_l,\ X_2 \subseteq B_j. \qquad (4.5.7)$$

Zachodzi ponadto

$$X_1 \subseteq B_i \cap B_l,\quad X_2 \subseteq B_i \cap B_j. \qquad (4.5.8)$$

Z lematu 4.4 wynika wtedy, że na drogach łączących wierzchołek D_i z wierzchołkami D_l i D_j muszą leżeć takie wierzchołki odpowiednio D_p i D_r, że związane z nimi zbiory B_p i B_r spełniają następujące związki:

$$B_i \supset B_s \supset \ldots \supset B_p \subset \ldots \subset B_l,\ B_p = B_i \cap B_l,$$

$$B_i \supset B_s \supset \ldots \supset B_r \subset \ldots \subset B_j,\ B_r = B_i \cap B_j,$$

gdzie D_s jest jedynym wierzchołkiem sąsiednim wierzchołka końcowego D_i.

Ze związków (4.5.8) wynika, że $X_1 \subseteq B_p$ oraz $X_2 \subseteq B_r$, więc $X_1 \subset B_s$ i $X_2 \subset B_s$, czyli $X_1 \cup X_2 \subseteq B_s$. Ostatecznie mamy $B_i \cap X \subseteq B_s$, co znaczy, że wierzchołek D_i musiałby zostać usunięty w kroku (1) rozważanego algorytmu. Przyjmując, że wierzchołek D_p pokrywa się z jednym z wierzchołków D_i lub D_l, a D_r z D_i lub D_j, otrzymamy związki analogiczne do (4.5.5) i (4.5.6). Powtórzenie przedstawionych uprzednio rozważań doprowadzi do analogicznego wniosku, że wierzchołek D_i nie może istnieć w rozpatrywanym podgrafie D_X. Rozpatrzenie większej liczby zbiorów, w których mogłyby być rozproszone atrybuty $B_i \cap X$, nie zmieni końcowego wniosku.

Pozostaje do wykazania, że zbiory B_k związane ze wszystkimi wierzchołkami podgrafu D_X zawierają wszystkie atrybuty tworzące podzbiór X.

Jeżeli atrybuty te należały do krawędzi hipergrafu $\mathbb{R}$, to należały też do krawędzi $B_1, \ldots, B_m$ związanych z wierzchołkami diagramu Bachmana. Krok (1) etapu 3 rozważanego algorytmu gwarantuje zaś zachowanie w zbiorach B_j związanych z podgrafem D_X wszystkich atrybutów tworzących X. □

Etap 4. (Wyznaczanie maksymalnych elementów podgrafu D_X)

Dane wejściowe: podgraf D_X i związane z jego wierzchołkami krawędzie hipergrafu Bachmana.

Dane wyjściowe: minimalny zbiór krawędzi (schematów relacji) wiążących atrybuty zbioru X.

(1) Utwórz zbiór $\mathbf{V}$ początkowo pusty.

(2) Dla każdego wierzchołka $D_i \in D_X$ sprawdź, czy istnieje taki wierzchołek $D_j \in D_X$, $D_j \neq D_i$, że $B_i \subseteq B_j$. Jeśli nie, włącz B_i do $\mathbf{V}$. □

Zauważmy, że warunki sprawdzane w kroku (2) odpowiadają dokładnie warunkom występującym w definicji maksymalnych elementów podgrafu D_X w podrozdziale 4.5.3.

Etapy (1)-(4) prezentowanego algorytmu wystarczają do wyznaczenia poszukiwanej liczby połączeń relacji w procesie rozwiązywania zadania określonego zbiorem atrybutów X. Wynosi ona $|\mathbf{V}| - 1$, gdzie $|\mathbf{V}|$ jest liczbą elementów zbioru $\mathbf{V}$.

Dołączenie dodatkowego etapu pozwoli ostatecznie wyznaczyć zbiór relacji wiążących atrybuty X, dając możność wykorzystania przezentowanego algorytmu również w systemach automatycznego rozwiązywania zadań wyszukiwania danych.

Etap 5. (Wybór relacji wiążących atrybuty zbioru X)

Dane wejściowe: minimalny zbiór schematów relacji wiążących atrybuty zbioru X oraz schemat relacyjnej bazy danych.

Dane wyjściowe: zbiór relacji wiążących atrybuty zbioru X.

Utwórz zbiór oznaczeń relacji $\mathbf{v}$ początkowo pusty. Dla każdego $V_k \in \mathbf{V}$ wykonaj jeden z dwóch następujących kroków:

(1) Jeśli istnieje takie $R_i \in \mathbf{R}$, że $V_k = R_i$, wprowadź $r_i(R_i)$ do zbioru $\mathbf{v}$.

(2) Jeśli nie istnieje takie $R_i \in \mathbf{R}$, że $V_k = R_i$, wyszukaj wszystkie takie schematy $R_{j_1}, R_{j_2}, \ldots, R_{j_k}$, dla których zachodzi $V_k \subseteq R_{j_1}, \ldots, V_k \subseteq R_{j_k}$. Relacja $s(V_k)$ wprowadzana w tym przypadku do zbioru winna być utworzona następująco:

$$s(V_k) = \bigcup_{i=1}^{k} r_{j_i}(R_{j_i})\,[V_k]. \quad \square$$

Ocena złożoności obliczeniowej algorytmu 4.6

Podstawowym parametrem służącym do oceny złożoności algorytmu 4.6 jest liczba krawędzi m hipergrafu Bachmana utworzonego dla danego hipergrafu $\mathbf{R}$. W pracy [90] pokazano, że dla γ-acyklicznego hipergrafu $\mathbf{R}$ liczba m spełnia nierówność

$$m \leq |\mathbf{R}| + 2|U|,$$

gdzie $|\mathbf{R}|$ jest liczbą krawędzi hipergrafu $\mathbf{R}$, a $|U|$ jest liczbą wszystkich wierzchołków $\mathbf{R}$. Używając poprzednio wprowadzonych oznaczeń możemy też zapisać

$$m \leq q + 2a,$$

gdzie q jest liczbą schematów relacyjnej bazy danych, a - liczbą atrybutów tej bazy.

Złożoność algorytmu 4.6 ocenimy odrębnie dla każdego etapu:

(1) Utworzenie hipergrafu Bachmana wymaga wykonaie $m \cdot q$ operacji badania zawierania się bądź równości zbiorów.

(2) Utworzenie diagramu Bachmana wymaga wykonania dla każdego $B_i \in B$ $m-1$ operacji badania zawierania się zbiorów w kroku (2.1) oraz $1(1-1)$ takich operacji w kroku (2.2). Sumaryczna liczba tych operacji nie przekracza wartości m^3.

(3) Liczba operacji usuwania wierzchołków końcowych nie przekracza liczby wszystkich wierzchołków, to znaczy m.

(4) Wyszukiwanie wierzchołków maksymalnych wymaga wykonania - w najgorszym przypadku - $m(m-1)$ badań zawierania się zbiorów.

(5) Sprawdzenie istnienia takiego zbioru R_i, dla którego zachodzi $V_k = R_i$, wymaga q badań. Wyszukanie wszystkich zbiorów R_i, w których zawiera się V_k, również wymaga q badań. Całkowita liczba badań równości i zawierania się zbiorów nie przekracza więc $2mq$.

W ocenie tego etapu nie uwzględniamy złożoności procesu tworzenia nowych relacji. Ta operacja wykonywana będzie bowiem przy wykorzystaniu translatora (interpretatora) języka manipulowania danymi.

Ponieważ $m > q$, więc pesymistyczna złożoność całego algorytmu wynosi $O(m^3)$.

Uwzględniając związki $m \leq q + 2a$ oraz $q \leq n$, gdzie n jest liczbą zależności funkcjonalnych tworzących zbiór F, oszacowanie złożoności algorytmu 4.6 można też wyrazić zależnością $O((n+2a)^3)$.

4.5.5. Końcowa postać algorytmu syntezy uwzględniającego zadania wyszukiwania danych

W ostatnim punkcie tego rozdziału przedstawimy w skróconym ujęciu końcową postać algorytmu syntezy. Algorytm ten - będący zestawieniem poprzednio prezentowanych algorytmów - tworzy wszystkie postacie schematu relacyjnej bazy danych oraz dokonuje wyboru schematu najlepszego przy uwzględnieniu zadań wyszukiwania danych. Zwarta forma algorytmu wynika z faktu, że większość jego kroków została już poprzednio omówiona.

Algorytm 4.7. (Algorytm syntezy uwzględniający zadania wyszukiwania danych)

Dane wejściowe: zbiór F zależności funkcjonalnych oraz rodzina podzbiorów atrybutów $\mathbf{X} = \{X_1, \ldots, X_{n_z}\}$ reprezentujących zadania wyszukiwania danych $\mathbf{Z} = \{Z_1, \ldots, Z_{n_z}\}$, scharakteryzowane ponadto zbiorem wag $\{\gamma_1, \gamma_2, \ldots, \gamma_{n_z}\}$.

Dane wyjściowe: schemat relacyjnej bazy danych spełniający rozszerzone kryteria procesu syntezy.

(1) Wyodrębnij klasy równoważności kluczy $K_1,\ldots,K_q$ dla lewostronnie zredukowanego zbioru F.

(2) Utwórz pokrycie dla potrzeb syntezy H_s zbioru F.

(3) Dla każdej klasy równoważności kluczy K_m utwórz rodzinę $\mathfrak{R}_m$ alternatywnych schematów relacji.

(4) Utwórz zbiór wszystkich postaci schematu relacyjnej bazy danych

$$\mathfrak{R} = \mathfrak{R}_1 \times \mathfrak{R}_2 \times \ldots \times \mathfrak{R}_q .$$

(5) Dla każdego schematu $\mathbb{R} \in \mathfrak{R}$ utwórz jego hipergraf. Jeśli hipergraf $\mathbb{R}$ jest γ-acykliczny, utwórz dla niego hipergraf Bachmana i diagram Bachmana, a potem wykonaj następujące kroki:

(5.1) Dla każdego podzbioru $X \in \mathbb{X}$ wyznacz spójny podgraf D_X diagramu Bachmana, a następnie zbiór $\mathbb{V}$ schematów relacji wiążących atrybuty podzbioru X.

(5.2) Wyznacz sumaryczną ważną liczbę połączeń relacji dla wszystkich $X \in \mathbb{X}$

(6) Wybierz jako rozwiązanie końcowe taki schemat $\mathbb{R} \in \mathfrak{R}$, któremu odowiada najmniejsza sumaryczna ważona liczba połączeń. □

Komentując przedstawiony algorytm zauważmy, że kroki (1) i (2) tego algorytmu zostały szczegółowo przedstawione w algorytmie 3.3, kroki (3) i (4) w algorytmie 4.4, natomiast kroki (5) i (5.1) - w algorytmach 4.5 i 4.6.

Bardziej szczegółowego objaśnienia wymaga sposób realizacji kroków (5.2) i (6).

Zauważmy, że liczba połączeń relacji p dla danego X jest równa $|\mathbb{V}| - 1$, gdzie $|\mathbb{V}|$ jest liczbą maksymalnych elementów podgrafu D_X. Zbiór $\mathbb{V}$ wyznaczany jest dla każdego $X \in \mathbb{X} = \{X_1,\ldots,X_{n_z}\}$. W rezultacie otrzymamy zbiór liczb połączeń $\{p_1,p_2,\ldots,p_{n_z}\}$. Wykorzystując wagi $\gamma_1,\ldots,\gamma_{n_z}$ przypisane poszczególnym zadaniom ze zbioru $\mathbb{Z} = \{Z_1,\ldots,Z_{n_z}\}$ wyliczymy wtedy sumaryczną ważoną liczbę połączeń zgodnie z zależnością (4.1.1)

$$k = \sum_{i=1}^{n_z} \gamma_i \cdot p_i .$$

Kryterialna wartość k wyznaczona zostanie w ten sposób dla każdego schematu $\mathbb{R} \in \mathfrak{R}$.

W kroku (6) algorytmu 4.7 wybierana jest najmniejsza liczba k i odpowiadający jej schemat $\mathbb{R}$. Stanowi on poszukiwane rozwiązanie końcowe.

5. ZAKOŃCZENIE

Przedmiotem rozważań przedstawionej pracy są problemy procesu syntezy schematu relacyjnej bazy danych.

Treść pracy skupia się zarówno na zagadnieniach teoretycznych, do których należą sformułowanie koncepcji pokrycia dla potrzeb syntezy oraz zdefiniowanie i szeroka analiza zależności i zbiorów alternatywnie redundancyjnych, jak też na praktycznej organizacji procesu syntezy w przedstawionych algorytmach.

Wskazując na możliwość uzyskania dla dotychczasowych metod syntezy pewnych rozwiązań równorzędnych, zaproponowano dodatkowe kryterium oceny wyników syntezy. Jest nim złożoność zadań wyszukiwania danych przewidywanych do realizacji w projektowanej bazie. Uwzględnienie tego kryterium wymagało istotnej rozbudowy algorytmu syntezy w celu:

- utworzenia wszystkich możliwych schematów relacyjnej bazy danych oraz
- wyznaczenia zbioru relacji potrzebnych do rozwiązania każdego z zadań wyszukiwania danych.

Końcowym efektem pracy jest algorytm syntezy realizujący zamierzone cele.

W treści pracy można jednak wyróżnić kilka problemów, których rozwiązanie wydaje się interesujące samo w sobie i może zostać wykorzystane w odrębnych badaniach.

Pierwszym z nich jest modyfikacja klasycznego algorytmu syntezy Bernsteina.

Wprowadzenie koncepcji pokrycia dla potrzeb syntezy, zawierającego wszystkie zależności wiążące klucze równoważne, pozwoliło uprościć proces syntezy zarówno w zakresie pojęciowym, jak też w zakresie złożoności obliczeniowej (ale tylko średniej).

Jako drugi odrębny problem wskazać można teoretyczną analizę tych własności zależności funkcjonalnych, które prowadzą do utworzenia wielu pokryć, a dalej wielu schematów relacyjnej bazy danych dla tego samego wejściowego zbioru zależności. Wprowadzone zostały pojęcia zależności i zbiorów alternatywnie redundancyjnych. Wykazano, że własność alternatywnej redundancyjności jest relacją równoważności. Zbadane własności zależności i zbiorów alternatywnie redundancyjnych pozwalają w istotny sposób uprościć proces syntezy wszystkich postaci schematu relacyjnej bazy danych. Spełnienie pewnych własności uwarunkowane było przy tym zastosowaniem koncepcji pokrycia dla potrzeb syntezy.

Trzecim wreszcie oddzielnym zagadnieniem jest opracowanie algorytmu wyznaczania zbioru relacji łączonych w procesie rozwiązywania zadań w bazie danych. Algorytm ten wykorzystany został tylko do oceny złożoności tych zadań, może on jednak stanowić bazę pełnego systemu automatycznego rozwiązywania zadań wyszukiwania danych.

LITERATURA

1. Aho A.V., Beeri C., and Ullman J.D.: The Theory of Joins in Relational Databases. ACM TODS 4:3, September 1979, 297-314.
2. Aho A.V., Hopcroft J.E., i Ullman J.D.: Projektowanie i analiza algorytmów komputerowych. PWN, Warszawa 1983.
3. Aho A.V., Sagiv Y., and Ullman J.D.: Efficient Optimization of a Class of Relational Expressions. ACM TODS 4:4, December 1979, 435-454.
4. Aho A.V., Sagiv Y., and Ullman J.D.: Equivalences Among Relational Expressions. SIAM J. on Computing 8:2, May 1979, 218-246.
5. Albano A., De Antonellis V., Di Leva A.: Computer-Aided Database Design. The DATAID Project. Elsevier Science Publishers, 1985.
6. Armstrong W.W.: Dependency Structures of Data Base Relationships. 1974 IFIP Cong. Geneva, Switzerland, 580-583.
7. Armstrong W.W., and Delobel C.: Decompositons and Functional Dependencies in Relations. ACM TODS 5:4, December 1980, 404-430.
8. Astrahan M.M., Blasgen M.W., Chamberlin D.D., Eswaran K.P., Gray J.N., Griffiths P.P., King W.F., Lorie R.A., McJones P.R., Mehl J.W., Putzolu G.R., Traiger I.I., Wade B.W., Watson V.: System R: A Relational Approach to Database Management. ACM TODS 1:2, June 1976, 97-137.
9. Atzeni P., and Parker D.S.: Assumptions in relational database theory. In Proceedings ACM Symposium on Principles of Database Systems (Los Angeles, Mar. 29-31, 1982), ACM, New York, 1-9.
10. Babb B.: Joined normal form: A storage encoding for relational databases. ACM Trans. Database Syst. 7 (1982), 588-614.
11. Batini C. and D'Atri A.: Relational Data Base Desing Using Refinement Rules. R.A.I.R.O. Theoretical Informatics, 17:2, 1983, 97-119.
12. Beeri C.: On the Membership Problem for Functional and Multivalued Dependencies in Relational Databases. ACM TODS 5:3, September 1980, 241-259.
13. Beeri C., and Bernstein P.A.: Computational Problems Related to the Design of Normal Form Relational Schemas. ACM TODS 4:1, March 1979, 30-59.
14. Beeri C., Bernsterin P.A., and Goodman N.: A Sophisticate's Introduction to Database Normalization Theory. VLDB IV, West Berlin, Germany, ACM, IEEE 113-124.
15. Beeri A., Fagin R., Maier D., Mendelzon A., Ullman J., Yannakakis M.: Properties of Acyclic Database Schemes. ACM Symp. on Theory of Computing 1981, 355-362.
16. Beeri C., Fagin R., Maier D., Yannakakis M.: On the Desirability of Acyclic Database Schemes. J.ACM, 30:3, July 1983, 479-513.
17. Berge C.: Graphs and Hypergraphs. North-Holland, Amsterdam 1973.
18. Bernstein P.A.: Comment on "Segment Synthesis in Logical Database Design". IBM J. of Research and Development 20:4, Armonk, NY, July 1976, 412.
19. Bernstein P.A.: Synthesizing Third Normal Form Relations from Functional Dependencies. ACM TODS 1:4, December 1976, 277-298.

20. Biskup J.: A Foundation of Codd's Relational Maybe-operations. ACM TODS 8:4, December 1983, 608-636.

21. Chase K.: Join Graphs and Acyclic Database Schemes. VLDB VII, Cannes, 1982, France. ACM, IEEE 95-100.

22. Chen P.P.S.: The Entity-Relationship Model Toward a Unified View of Data. ACM TODS 1:1, March 1976, 9-36.

23. Codd E.F.: A Relational Model of Data for Large Shared Data Banks. CACM 13:6, June 1970, 377-387.

24. Codd E.F.: Further Normalization of the Database Relational Model. In Rustin, Englewood Cliffs, NJ: Prentice-Hall, 33-64.

25. Codd E.F.: Extending the Database Relational Model to Capture More Meaning. ACM TODS 4:4, December 1979, 397-343.

26. Date C.J.: Wprowadzenie do baz danych. WNT, Warszawa 1981.

27. Delobel C.: Normalization and Hierarchical Dependencies in the Relational Data Model. ACM TODS 3:3, September 1978, 201-222.

28. Delobel C.: An Overview of the Relational Data Theory. Information Processing 80, IFIP, 1980, 413-426.

29. Delobel C., and Casey R.G.: Decompositon of a Data Base and the Theory of Boolean Switching Functions. IBM J. of Research and Development 17:5, Armonk, NY, September 1973, 374-386.

30. Delobel C., Casey R.G., and Bernstein P.A.: Comment on "Decomposition of a Data Base and the Theory of Boolean Switching Functions". IBM J. of Research and Development 21:5, Armonk, NY, September 1977, 484-485.

31. Delobel C. and Pichat E.: The Dosing of Relational Information System According to Different Kind of Dependencies. Universite de Grenoble, France.

32. Demetrovics J.: On the equivalence of candidate keys with Sperner systems. Acta Cybernetica, Tom 4, Fasc. 3, Szeged, 1979, 247-252.

33. Demetrovics J.: Candidate keys and antichains. SIAM J. Alg. Disc. Math., Vol. 1, No. 1, 1980, 92.

34. Epstein R.: A Tutorial on INGRES. Memorandum No. ERL - M77-25, 1977, University of California.

35. Fagin R.: Mutlivalued Dependencies and a New Normal Form for Relational Databases. ACM TODS 2:3, September 1977, 262-278.

36. Fagin R.: Functional Dependencies in a Relational Database and Propositional Logic. IBM J. of Research and Development 21:6, Armonk, NY, November 1977, 534-544.

37. Fagin R.: The Decomposition Versus the Synthetic Approach to Relational Database Desing. VLDB III, Tokyo, Japan, ACM, IEEE 1978, 441-446.

38. Fagin R.: A Normal Form for Relational Databases that is Based on Domains and Keys. ACM TODS 6:3 September 1981, 387-415.

39. Fagin R.: Horn Clauses and Database Dependencies. JACM, 29:4, October 1982, 952-985.

40. Fagin R.: Armstrong Databases. 7th IBM Symph. on Math. Found. of Computer Science, Kangawa, Japan, 1982.

41. Fagin R.: Degrees of Acyclicity for Hypergraphs and Relational Database Schemes. J. ACM 30,3, July 1983, 514-550.

42. Fagin R., Maier D., Ullman J.D., Yannakakis M.: Tools for Template Dependencies SIAM J. Comp. 12:1, Feb. 1983, 36-59.

43. Fagin R., Mendelzon A.O., and Ullman J.D.: A Simplified Universal Relation Assumption and its Properties. ACM TODS, 7:3, September 1982 343-360.

44. Fagin R.: Acyclic database schemes (of various degrees): A Painless Introduction. CAAP83 8th Colloq. on Trees in Algebra and Program, Italy, March 1983.

45. Fadous R., and Forsyth J.: Keys for synthesized relations. Technical Report 82-004, Michigan State University.

46. Forsyth J. and Charnock L.: Relational schema graphs from multivalued dependencies with lossless joins. Technical Report 82-003, Michigan State University.

47. Goodman N. and Shmueli O.: Tree queries: A simple class of relational queries. ACM TODS, 7:4, December 1982, 653-677.

48. Graham M.H.: Functions in Databases. ACM TODS, 8:1, March 1983, 81-109.

49. Hagihara K., Ito M., Taniguchi K., Kasami T.: Decision Problems for Mutlivalued Dependencies in Relational Databases. SIAM J. on Computing 8:2 May 1979, 247-264.

50. Kent W.: Consequences of assuming a universal relation. ACM Trans. Database Syst. 6 (1981), 539-556.

51. Kent W.: A Simple Guide to Five Normal Forme in Relational Database Theory, CACM 26:2, February 1983, 120-125.

52. Kim W.: Relational Database Systems. ACM Computing Surveys 11:3, September 1979, 185-212.

53. King W.F.: Relational Database Systems: Where we stand today. Information Processing 80, IFIP, 1980, 369-381.

54. Klug A.: Calculating Constraints on Relational Expressions. ACM TODS 5:3, September 1980, 260-290.

55. Korth H.F., Kuper G.M., Feigenbaum J., Van Gelder A., and Ullman J.D.: System/U: A database system based on the universal relation assumption. ACM TODS 9:3, September 1984, 331-347.

56. Kozielski St.: A computer system for choosing the evacuation routes in the case of fire in a coal mine. In Proc. 3rd IFAC Symposium on Automation in mining, mineral and metal processing, Montreal, August 1980.

57. Kozielski St.: Projektowanie struktury relacji w relacyjnych bazach danych w zastosowaniu do wybranego systemu zbierania i selekcji danych. Podstawy Sterowania, t. 15, z. 1-2, 1985, 41-64.

58. Kozielski St.: Synteza schematów relacji przy uwzględnieniu wybranych własności kluczy. Podstawy Sterowania, t. 15, z. 4, 1985, 261-282.

59. Kozielski St.: Nowa koncepcja wyznaczania pokrycia i syntezy schematu relacyjnej bazy danych. Program wspomagania w projektowaniu schematu relacyjnej bazy danych. Sprawozdanie z realizacji tematu 01/04.7/ 1986, w ramach problemu węzłowego 06.4.

60. Kozielski St.: Pokrycie zbioru zależności funkcjonalnych w procesie syntezy schematu relacyjnej bazy danych. Podstawy Sterowania, w druku.

61. Kozielski St.: Realizacja algorytmu syntezy przy zmodyfikowanej metodzie wyznaczania pokrycia zbioru zależności funkcjonalnych. Podstawy Sterowania, w druku.

62. Kuratowski K.: Wstęp do teorii mnogości i topologii. PWN, Warszawa 1980.

63. Lien Y.E.: Hierarchical Schemata for Relational Databases. ACM TODS 6:1, March 1981, 48-69.

64. Lien Y.E.: On the equivalence of database models. J.ACM 29:2, April 1982, 333-363.

65. Ling T.W., Tompa F.W., and Kameda T.: An Improved Third Normal Form for Relational Databases. ACM TODS 6:2, June 1981, 329-346.

66. Lipski W.: On Semantic Issues Connected With Incomplete Information Databases. ACM TODS 4:3, September 1979, 262-296.

67. Lipski W.: Kombinatoryka dla programistów. WNT, Warszawa 1982.

68. Maier D.: Discarding the Universal Instance Assumption: Preliminary Results. Technical Report 80/008, March 1980.

69. Maier D.: Minimum Covers in the Relational Database Model. JACM 27:4, October 1980, 664-674.

70. Maier D.: The Theory of Relational Databases. Computer Science Press, Rockville, Md., 1983.

71. Maier D., Mendelzon A.O., and Sagiv Y.: Testing Implications of Data Dependencies. ACM TODS 4:4 December 1979, 455-469.

72. Maier D., and Ullman J.D.: Maximal Objects and the Semantics of Universal Relation Databases. ACM TODS 8:1, March 1983, 1-14.

73. Maier D., Ullman J., Vardi M.: On the Foundations of the Universal Relation Model. ACM TODS 9:2, June 1984, 283-308.

74. Oleński J., Staniszkis W.: Projektowanie bazy danych. PWE, Warszawa 1984.

75. Pawlak Z.: Systemy informacyjne. Podstawy teoretyczne. WNT, Warszawa 1983.

76. Perkinson R.C.: Data Analysis: The Key to Data Base Desing. QED Information Science, Wellesley, Masachusetts 1984.

77. Rasiowa H.: Wstęp do matematyki współczesnej. PWN, Warszawa 1977.

78. Rissanen J.: Independent Components of Relations. ACM TODS 2:4, December 1977, 317-325.

79. Rissanen J.: Theory of relations for databases - A tutorial survey. In Proc. 7th Symp. on Math. Found. of Computer Science, Springer-Verlag, 1978, 537-551.

80. Sagiv Y., Delobel C., Parker D.S., Fagin R.: An Equivalence Between Relational Database Dependencies and a Fragment of Propositional Logic. JACM 28:3, July 1981, 435-453.

81. Smith H.: Database Desing: Composing Fully Normalized Tables from a Rigorous Dependency Diagram. CACM 28:8, August 1985, 826-838.

82. Stonebraker M., Wong E., Kreps P., Held G.: The Desing and Implementation of INGERS. ACM TODS 1:3, September 1976, 189-222.

83. Tschritzis D.C., and Lochovsky F.H.: Data Base Mangement Systems. New York: Academic Press, Inc.

84. Ullman J.D.: The U.R. strikes back. In Proceedings ACM Symposium on Principles of Database Systems (Los Angeles, Mar. 29-31, 1982), ACM, New York, 10-22.

85. Ullman J.D.: Universal Relation Interface For Database Systems. Information Processing 83, IFIP, Paris 1983, 243-252.

86. Ullman J.D.: Principles of Database Systems. Rockville, MD: Computer Science Press, 1983.

87. Wang C.P., and Wedekind H.H.: Segment Synthesis in Logical Database Desing. IBM J. of Research and Development, Armonk, NY, January 1975, 71-77.

88. Wiederhold G.: Database Desing. New York: McGraw-Hill, 1977.

89. Wong E., and Youssefi K.: Decomposition - A Strategy for Query Precessing ACM TODS 1:3, September 1976, 223-241.

90. Yannakakis M.: Algorithms for Acyclic Database Schemes. VLDB VII, 1982, Cannes, France. ACM, IEEE 82-94.

91. Zaniolo C., and Melkanoff M.A.: On the Desing of Relational Database Schemata. ACM TODS 6:1, March 1981, 1-47.

92. Zloof M.M.: Query-by-Example: A Database Language. IBM Systems J. 16:4. Armonk, NY, December 1977, 324-343.

D O D A T E K

PRZYKŁAD PROJEKTOWANIA STRUKTURY LOGICZNEJ RELACYJNEJ BAZY DANYCH PRZY WYKORZYSTANIU PROGRAMU SYNTEZY

Niniejszy dodatek prezentuje wyniki procesu projektowania przykładowej bazy danych przy wykorzystaniu programu realizującego algorytm syntezy.

Przedmiotem projektu jest baza danych dotyczących procesu nauczania na przykładowym wydziale szkoły wyższej, jak też samej organizacji tego wydziału.

Przyjmiemy na wstępie, że liczba rozważanych atrybutów a także wskazane powiązania między nimi powinny zapewnić między innymi możliwość wyszukania takich informacji, jak:

- struktura organizacyjna wydziału (instytuty, zakłady, pracownie, ich dyrektorzy i kierownicy),
- skład osobowy instytutów,
- kierunki i specjalności kształcenia,
- skład grup dziekańskich na poszczególnych kierunkach i specjalnościach; wykaz starostów i opiekunów grup,
- rozkłady zajęć dla wszystkich grup dziekańskich,
- rozkłady zajęć dla poszczególnych sal dydaktycznych,
- literatura zalecana do poszczególnych przedmiotów,
- programy studiów,
- wyniki zaliczeń i egzaminów,
- dane osobowe dotyczące studentów,
- rozkłady zajęć pracowników,
- informacje o kołach naukowych.

Wykaz atrybutów tworzących omawianą bazę przedstawia się następująco (przyjęte symbole używane są w prezentowanych dalej wynikach programu):

Dane dotyczące organizacji wydziału:

W - wydział,
I - instytut,
Z - zakład,
PR - pracownia,
DZ - dziekan,
DY - dyrektor instytutu,
KZ - kierownik zakładu,
KP - kierownik pracowni,
PK - pracownik,
NT - numer telefonu,
PO - numer pokoju.

Dane dotyczące organizacji studiów:

K - kierunek studiów,
SP - specjalność,
R - rok studiów,
SE - semestr studiów,
GR - numer grupy dziekańskiej,
ST - starosta grupy,
OR - opiekun roku,
OG - opiekun grupy,
P - przedmiot,
WY - wykładowca,
LG - liczba godzin (w tygodniu),
D - dzień tygodnia,
SA - numer sali,
GO - godzina (przedział od - do).

Dane dotyczące wyników studiów:

OZ - ocena zaliczenia,
OE - ocena egzaminu,
DA - data zaliczenia,
DE - data egzaminu,
TE - termin egzaminu.

Dane dotyczące studenta:

S - student,
N - nazwisko,
IM - imiona,
DU - data urodzenia,
NI - numer indeksu,
ND - numer dowodu osobistego,
NL - numer legitymacji studenckiej.

Dane dotyczące podręczników:

TY - tytuł,
AU - autor,
NW - numer wydania,
IS - ISBN
WW - wskaźnik ważności.

Dane dotyczące studenckich kół naukowych:

KO - nazwa koła,
CZ - członek koła,
PZ - prezes koła.

Przy określaniu zależności funkcjonalnych między atrybutami przyjmowano następujące ważniejsze założenia:

- Baza danych organizowana jest dla jednego określonego wydziału.
- Przechowywane są tylko aktualne dane dotyczące struktury organizacyjnej wydziału.
- Nazwisko, imię i tytuł jednoznacznie identyfikują każdego pracownika (również na stanowisku kierowniczym).
- Dane dotyczące składu grup studenckich, opiekunów grup, rozkładów zajęć itp. przechowuje się tylko dla bieżącego semestru.
- Każdego studenta jednoznacznie identyfikuje wartość atrybutu "student" stanowiąca łańcuch znaków tworzących nazwisko i imię (imiona) danego studenta. Przyjmować będziemy przy tym, że atrybuty "nazwisko" i "imiona" istnieć będą niezależnie od atrybutu "student".
- Student może studiować na więcej niż jednym kierunku (lub specjalności).
- Na danym kierunku może nie być odrębnych specjalności.
- W przypadku zajęć prowadzonych przez kilka osób atrybut "wykładowca" dotyczy pracownika zaliczającego dany przedmiot.
- W rozkładach zajęć wartością atrybutu "godzina" jest para liczb określających początek i koniec danych zajęć.
- W wykazie literatury do danego przedmiotu atrybut "wskaźnik ważności" określa wagę danej pozycji.
- Jedyne związki między niektórymi atrybutami są zależnościami wielowartościowymi, np.:

koło naukowe $\longrightarrow\!\!\!\!\rightarrow$ członek koła,
przedmiot $\longrightarrow\!\!\!\!\rightarrow$ podręcznik.

Dla uwzględnienia tych zależności - istotnych w procesie syntezy - można wprowadzić (wykorzystując sugestie przedstawione w pracy [19]) argumenty puste. Prowadzi to do przekształcenia powyższych zależności do postaci

koło naukowe, członek koła $\longrightarrow \Theta_1$,
przedmiot, podręcznik $\longrightarrow \Theta_2$.

Po zakończeniu procesu syntezy argumenty puste Θ_1, Θ_2 należy usunąć.

Zamieszczone dalej wydruki prezentują wyniki działania programu realizującego algorytm 3.3 dla zadanego zbioru zależności funkcjonalnych. Dla zilustrowania działania programu przedstawiono wyniki pośrednie uzyskane w procesie sytnezy.

```
WYNIKI PROGRAMU SYNTEZA :

DANE WPROWADZONE :

 1. W  -> DZ
 2. DZ -> W
 3. I  -> W
 4. Z  -> I
 5. Z  -> W
 6. PR -> I
 7. PR -> Z
 8. PR -> W
 9. DY -> I
10. I  -> DY
11. KZ -> Z
12. Z  -> KZ
13. PR -> KP
14. KP -> PR
15. I  PK -> W
16. PK -> PR
17. PK -> Z
18. PK -> I
19. PK -> PO
20. NT -> PO
21. DY -> W
22. DY -> DZ
23. PK GO D  -> W
24. PK GO D  -> SA
25. SP K  -> W
26. SP K  S  -> SE GR
27. SP K  S  -> NI
28. NI -> SP K  S
29. SP K  SE -> OR
30. SP K  SE GR -> OG
31. SP K  SE GR -> ST
32. SE -> R
33. GO D  SP K  SE GR -> SA P  WY
34. GO D  SA -> SP K  SE GR P
35. W  SP K  SE P  -> WY LG
36. SP K  S  SE P  -> DA OZ
37. SP K  S  SE P  TE -> DE OE
38. TY AU NW -> IS
39. SP K  SE P  TY AU NW WW -> T1
40. IS -> TY AU NW
41. S  -> N  IM DU NL ND
42. N  IM -> S
43. NL -> S
44. ND -> S
45. KO -> W  I  SA K  PZ
46. KO CZ -> T2

 KONIEC DANYCH

 PO REDUKCJI LEWOSTRONNEJ :

 1. W  -> DZ
```



```
 2. DZ -> W
 3. I  -> W
 4. Z  -> I
 5. Z  -> W
 6. PR -> I
 7. PR -> Z
 8. PR -> W
 9. DY -> I
10. I  -> DY
11. KZ -> Z
12. Z  -> KZ
13. PR -> KP
14. KP -> PR
15. PK -> W
16. PK -> PR
17. PK -> Z
18. PK -> I
19. PK -> PO
20. NT -> PO
21. DY -> W
22. DY -> DZ
23. PK -> W
24. PK GO D  -> SA
25. SP K  -> W
26. SP K  S  -> SE GR
27. SP K  S  -> NI
28. NI -> SP K  S
29. SP K  SE -> OR
30. SP K  SE GR -> OG
31. SP K  SE GR -> ST
32. SE -> R
33. GO D  SP K  SE GR -> SA P  WY
34. GO D  SA -> SP K  SE GR P
35. SP K  SE P  -> WY LG
36. SP K  S  P  -> DA OZ
37. SP K  S  P  TE -> DE OE
38. TY AU NW -> IS
39. SP K  SE P  TY AU NW WW -> T1
40. IS -> TY AU NW
41. S  -> N  IM DU NL ND
42. N  IM -> S
43. NL -> S
44. ND -> S
45. KO -> W  I  SA K  PZ
46. KO CZ -> T2

 PO PODZIALE NA GRUPY

 1. W  -> DZ
 2. DZ -> W
 3. I  -> W  DY
 4. Z  -> W  I  KZ
 5. PR -> W  I  Z  KP
 6. DY -> W  DZ I
 7. KZ -> Z
 8. KP -> PR
 9. PK -> W  I  Z  PR PO
10. NT -> PO
11. PK GO D  -> SA
12. SP K  -> W
13. SP K  S  -> SE GR NI
14. NI -> SP K  S
15. SP K  SE -> OR
16. SP K  SE GR -> OG ST
17. SE -> R
18. GO D  SP K  SE GR -> SA P  WY
```

```
19. GO D  SA -> SP K  SE GR P
20. SP K  SE P  -> WY LG
21. SP K  S  P  -> DA OZ
22. SP K  S  P  TE -> DE OE
23. TY AU NW -> IS
24. SP K  SE P  TY AU NW WW -> T1
25. IS -> TY AU NW
26. S  -> N  IM DU NL ND
27. N  IM -> S
28. NL -> S
29. ND -> S
30. KO -> W  I  SA K  PZ
31. KO CZ -> T2
```

WEKTOR W :

```
WC 1]= 1  WC 2]= 2  WC 3]= 3  WC 4]= 5  WC 5]= 7  WC 6]= 4
WC 7]= 6  WC 8]= 8  WC 9]= 0  WC10]= 0  WC11]= 0  WC12]= 0  WC13]= 9
WC14]=10  WC15]= 0  WC16]= 0  WC17]= 0  WC18]=11  WC19]=12  WC20]= 0
WC21]= 0  WC22]= 0  WC23]=13  WC24]= 0  WC25]=14  WC26]=15  WC27]=16
WC28]=17  WC29]=18  WC30]= 0  WC31]= 0
```

MACIERZ M :

```
    0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18
 1 !  1  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 2 !  2  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 3 !  3  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 4 !  6  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 5 !  4  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0
 6 !  7  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0
 7 !  5  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0
 8 !  8  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0
 9 ! 13  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0
10 ! 14  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0
11 ! 18  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0
12 ! 19  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0
13 ! 23  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0
14 ! 25  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0
15 ! 26  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1
16 ! 27  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  1  1
17 ! 28  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  0  1
18 ! 29  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  0
```

POSTAC POKRYCIA DLA POTRZEB SYNTEZY:

```
 1. W  -> W  DZ
 2. DZ -> W  DZ
 3. I  -> I  DY
 4. Z  -> I  Z  KZ
 5. PK -> Z  PR KP
 6. DY -> DZ I  DY
 7. KZ -> Z  KZ
 8. KP -> PR KP
 9. PK -> PR PO
10. NT -> PO
11. PK GO D  -> SA
12. SP K  -> W
13. SP K  S  -> SP K  S  SE GR NI
14. NI -> SP K  S  NI
15. SP K  SE -> OR
16. SP K  SE GR -> OG ST
17. SE -> R
18. GO D  SP K  SE GR -> GO D  SA SP K  SE GR
```



```
19. GO D  SA -> GO D  SA SP K  SE GR P
20. SP K  SE P  -> WY LG
21. SP K  S  P  -> DA OZ
22. SP K  S  P  TE -> DE OE
23. TY AU NW -> TY AU NW IS
24. SP K  SE P  TY AU NW WW -> T1
25. IS -> TY AU NW IS
26. S  -> S  N  IM DU NL ND
27. N  IM -> S  N  IM NL ND
28. NL -> S  N  IM NL ND
29. ND -> S  N  IM NL ND
30. KO -> I  SA K  PZ
31. KO CZ -> T2
```

ZALEZNOSCI REDUNDANCYJNE:

```
1. I  -> W
2. Z  -> W
3. PR -> W  I
4. DY -> W
5. PK -> W  I  Z
6. GO D  SP K  SE GR -> P  WY
7. KO -> W
```

```
----------RELACJA NR   1
  KLUCZE:
KLUCZ  1 =  W
KLUCZ  2 =  DZ
  SCHEMAT:
 W  DZ
----------RELACJA NR   2
  KLUCZE:
KLUCZ  1 =  I
KLUCZ  2 =  DY
  SCHEMAT:
 DZ I  DY
----------RELACJA NR   3
  KLUCZE:
KLUCZ  1 =  Z
KLUCZ  2 =  KZ
  SCHEMAT:
 I  Z  KZ
----------RELACJA NR   4
  KLUCZE:
KLUCZ  1 =  PR
KLUCZ  2 =  KP
  SCHEMAT:
 Z  PR KP
----------RELACJA NR   5
  KLUCZE:
KLUCZ  1 =  PK
  SCHEMAT:
 PR PK PO
----------RELACJA NR   6
  KLUCZE:
KLUCZ  1 =  NT
  SCHEMAT:
 PO NT
----------RELACJA NR   7
  KLUCZE:
KLUCZ  1 =  PK GO D
  SCHEMAT:
 PK GO D  SA
----------RELACJA NR   8
```

```
  KLUCZE:
KLUCZ  1 =  SP K
  SCHEMAT:
 W  SP K
-----------RELACJA NR  9
  KLUCZE:
KLUCZ  1 =  SP K  S
KLUCZ  2 =  NI
  SCHEMAT:
 SP K  S  SE GR NI
-----------RELACJA NR  10
  KLUCZE:
KLUCZ  1 =  SP K  SE
  SCHEMAT:
 SP K  SE OR
-----------RELACJA NR  11
  KLUCZE:
KLUCZ  1 =  SP K  SE GR
  SCHEMAT:
 SP K  SE GR OG ST
-----------RELACJA NR  12
  KLUCZE:
KLUCZ  1 =  SE
  SCHEMAT:
 SE R
-----------RELACJA NR  13
  KLUCZE:
KLUCZ  1 =  GO D  SP K  SE GR
KLUCZ  2 =  GO D  SA
  SCHEMAT:
 GO D  SA SP K  SE GR P
-----------RELACJA NR  14
  KLUCZE:
KLUCZ  1 =  SP K  SE P
  SCHEMAT:
 SP K  SE P  WY LG
-----------RELACJA NR  15
  KLUCZE:
KLUCZ  1 =  SP K  S  P
  SCHEMAT:
 SP K  S  P  DA OZ
-----------RELACJA NR  16
  KLUCZE:
KLUCZ  1 =  SP K  S  P  TE
  SCHEMAT:
 SP K  S  P  TE DE OE
-----------RELACJA NR  17
  KLUCZE:
KLUCZ  1 =  TY AU NW
KLUCZ  2 =  IS
  SCHEMAT:
 TY AU NW IS
-----------RELACJA NR  18
  KLUCZE:
KLUCZ  1 =  SP K  SE P  TY AU NW WW
  SCHEMAT:
 SP K  SE P  TY AU NW WW T1
-----------RELACJA NR  19
  KLUCZE:
KLUCZ  1 =  S
KLUCZ  2 =  N  IM
KLUCZ  3 =  NL
KLUCZ  4 =  ND
  SCHEMAT:
 S  N  IM DU NL ND
-----------RELACJA NR  20
```



```
  KLUCZE:
KLUCZ  1 =  KO
  SCHEMAT:
 I  SA K  KO PZ
-----------RELACJA NR  21
  KLUCZE:
KLUCZ  1 =  KO CZ
  SCHEMAT:
 KO CZ T2

  KONIEC PROGRAMU
 CZAS REALIZACJI :     5343
```

^**********

Analizując otrzymane wyniki zwrócimy uwagę na to, że efekty lewostronnej redukcji (usuwania obcych atrybutów) możemy ocenić porównując odpowiednie zbiory zależności. Atrybutem obcym jest więc np. atrybut W w zależności W,SP,K,SE,P ⟶ WY,LG lub też atrybut SE w zależności SP,K,S,SE,P ⟶ DA,OZ.

Komentując zawartość wektora W zwrócimy uwagę na to, że zerowe wartości elementów tego wektora odpowiadają tym zależnościom funkcjonalnym, których lewe strony tworzą trywialne klasy równoważności kluczy. Niezerowe wartości tych elementów przyporządkowują natomiast danemu kluczowi wskazany wiersz macierzy M. Jedynki na określonych pozycjach tego wiersza pokazują, które klucze tworzą wraz z kluczem danym wspólną klasę równoważności (nietrywialną).

Wyznaczone pokrycie jest pokryciem dla potrzeb syntezy, więc każda zależność, której lewa strona należy do nietrywialnej klasy równoważności kluczy, zawiera w prawej stronie wszystkie klucze równoważne z danej klasy.

Poszczególne relacje, których schematy były celem syntezy, będą zawierały następujące informacje:

1 - nazwę wydziału, dla którego tworzona jest baza oraz nazwisko i imię dziekana (znaczenie tej relacji jest raczej symboliczne),
2 - wykaz instytutów i ich dyrektorów,
3 - wykaz zakładów (w ramach instytutów) i kierowników tych zakładów,
4 - wykaz pracowni (w ramach zakładów) i kierowników pracowni,
5 - skład osobwy pracowni wraz z numerami zajmowanych pokoi,
6 - wykaz numerów telefonów i ich rozmieszczenie,
7 - rozkład zajęć pracowników,
8 - wykaz kierunków i specjalności prowadzonych na wydziale,
9 - skład osobowy grup dziekańskich,
10 - wykaz opiekunów lat,
11 - wykaz starostów i opiekunów grup,
12 - zbiór odwzorowań semestr - rok studiów (relację tę można "zastąpić" procedurę wyznaczającą takie odwzorowanie),

13 - rozkład zajęć studentów,
14 - program studiów i wykaz prowadzących zajęcia,
15 - wyniki zaliczeń,
16 - wyniki egzaminów,
17 - dane o podręcznikach,
18 - wykaz podręczników do poszczególnych przedmiotów, (atrybut T1 jest atrybutem pustym i nie należy go uwzględniać przy zakładaniu relacji),
19 - dane osobowe studenta,
20 - dane o kołach naukowych,
21 - wykaz członków poszczególnych kół naukowych, (atrybut T2 jest atrybutem pustym).

W dalszej części niniejszego dodatku zaprezentowane zostaną wyniki procesu projektowania schematu relacyjnej bazy danych przy uwzględnieniu zadań wyszukiwania danych.

Wykorzystano do tego celu program realizujący algorytm 4.7 (zrezygnowano przy tym w kroku (3) z operacji przedstawionych w algorytmie 4.3, praktycznie nie wpływających na wyniki).

Przedstawione badania dotyczą w dalszym ciągu rozpatrywanej bazy danych dla wyższej uczelni.

Zaprojektowany uprzednio schemat okazał się schematem γ- cyklicznym, onieczne więc było rozdzielenie nazw następujących atrybutów:

- obok atrybutu "przedmiot" wprowadzono atrybut "przedmiot planowany" (PP), wykorzystywany w zależnościach dotyczących rozkładu zajęć,
- obok atrybutów "pracownik" i "wykładowca" wprowadzono atrybut "prowadzący zajęcia" (PZ), uwzględniany również w zależnościach dotyczących rozkładu zajęć,
- obok atrybutu "student" wprowadzono atrybuty "student zaliczający" (SZ) i "student egzaminowany" (SG), wykorzystane w zależnościach dotyczących wyników nauczania.

Z uwagi na ograniczoną liczbę atrybutów akceptowanych przez program zrezygnowano z niektórych atrybutów i zależności (koła naukowe, rok studiów).

Interpretacja utworzonych relacji jest analogiczna do przedstawionej dla poprzedniego rozwiązania (przy nieco zmienionej kolejności).

Dla oceny tworzonych kolejno wersji schematu relacyjnej bazy danych wybrano następujące zadania wyszukiwania danych, przypisując im arbitalnie przyjęte współczynniki wagi:

1. Podaj wykaz pracowników wskazanego instytutu:

$$X_1 = \{PK,I\}, \quad \gamma_1 = 5.$$

2. Podaj wykaz pracowni w poszczególnych instytutach:

$$X_2 = \{PR,I\}, \quad \gamma_2 = 1.$$

3. Podaj wykaz przedmiotów wykładanych przez pracowników wskazanego instytutu na wskazanym semestrze, specjalności i kierunku:

$$X_3 = \{P,I,PK,WY,SE,SP,K\}, \quad \gamma_3 = 5.$$

4. Podaj rozkład zajęć dla wskazanej grupy (określenie grupy wymaga podania numeru grupy, specjalności, kierunku i semestru):

$$X_4 = \{GR,SP,K,SE,GO,D,SA,PP\}, \quad \gamma_4 = 5.$$

5. Podaj wykaz opiekunów grup, numery ich telefonów oraz nazwy instytutów, w których pracują:

$$X_5 = \{OG,SP,K,SE,GR,I,NT\}, \quad \gamma_5 = 4.$$

6. Podaj skład poszczególnych grup dziekańskich:

$$X_6 = \{SP,K,SE,GR,S\}, \quad \gamma_6 = 5.$$

7. Podaj wyniki egzaminu wskazanej grupy ze wskazanego przedmiotu:

$$X_7 = \{SP,K,SE,GR,S,SG,TE,OE\}, \quad \gamma_7 = 4.$$

8. Podaj rozkład zajęć dydaktycznych pracowników wskazanego instytutu:

$$X_8 = \{PK,PZ,I,SP,K,SE,GR,PP,GO,D,SA\}, \quad \gamma_8 = 3.$$

Przyjęte dane i wyniki działania programu prezentują zamieszczone dalej wydruki.

Dla celów ilustracyjnych zamieszczono pełne wyniki oceny każdego schematu dla wszystkich zadań wyszukiwania danych. Ograniczając objętość wydruków przedstawiono rezultaty oceny tylko piętnastu pierwszych wersji schematu relacyjnej bazy danych. W tym zbiorze rozwiązeń najlepsze wyniki uzyskano dla wersji nr 4. Wersja ta stanowi jedno z kilku równorzędnych rozwiązań (o identycznej minimalnej wartości ważonej sumy połączeń) w zbiorze wszystkich możliwych postaci schematu projektowanej bazy danych.

```
        WYNIKI PROGRAMU SYNTEZA I OCENA :

     DANE WPROWADZONE :

 1. W  -) DZ
 2. DZ -) W
 3. S  SP K  -) SE GR NI
 4. NI -) S  SP K
 5. I  -) W
 6. S  -) N  IM DU NL ND
 7. NL -) S
 8. ND -) S
 9. Z  -) W
10. Z  -) I
11. PR -) I
12. PR -) Z
13. PR -) W
14. I  -) DY
15. DY -) I
16. Z  -) KZ
17. KZ -) Z
18. PR -) KP
19. KP -) PR
20. PK -) PR
21. PK -) Z
22. PK -) I
23. DY -) W
24. DY -) DZ
25. PK -) PO
26. NT -) PO
27. SP K  -) W
28. SP K  SE -) OR
29. SP K  SE GR -) OG
30. SP K  SE GR -) ST
31. SP K  SE GR PP -) PZ
32. SP K  SE GR GO D  -) PP PZ SA
33. GO D  SA -) SP K  SE GR PP PZ
34. SP K  SE P  -) LG WY
35. SP K  SE P  SZ -) DA OZ
36. SP K  SE P  SG TE -) DE OE
37. TY AU NW -) IS
38. IS -) TY AU NW
39. SP K  SE P  TY AU NW WW -) T1

ZBIORY ATRYBUTOW X DLA ZADAN WYSZUKIWANIA DANYCH
 WRAZ ZE WSPOLCZYNNIKAMI WAG
 1.  I  PK   -      5
 2.  I  PR   -      1
 3.  SP K  SE I  PK P  WY   -      5
 4.  SP K  SE GR PP GO D  SA   -      5
 5.  SP K  SE GR I  NT OG   -      4
 6.  S  SP K  SE GR   -      5
 7.  S  SP K  SE GR P  SG TE OE   -      4
 8.  SP K  SE GR I  PK PP PZ GO D  SA   -      1

LICZBA ATRYBUTOW WYSTEPUJACYCH W DANYCH = 46
```



```
 KONIEC DANYCH

  POSTAC POKRYCIA DLA POTRZEB SYNTEZY:

 1. W  -) W  DZ
 2. DZ -) W  DZ
 3. S  SP K  -) S  SP K  SE GR NI
 4. NI -) S  SP K  NI
 5. I  -) I  DY
 6. S  -) S  N  IM DU NL ND
 7. NL -) S  NL ND
 8. ND -) S  NL ND
 9. Z  -) I  Z  KZ
10. PR -) Z  PR KP
11. DY -) DZ I  DY
12. KZ -) Z  KZ
13. KP -) PR KP
14. PK -) PR PO
15. NT -) PO
16. SP K  -) W
17. SP K  SE -) OR
18. SP K  SE GR -) OG ST
19. SP K  SE GR PP -) PZ
20. SP K  SE GR GO D  -) SP K  SE GR GO D  SA
21. GO D  SA -) SP K  SE GR PP GO D  SA
22. SP K  SE P  -) LG WY
23. SP K  SE P  SZ -) DA OZ
24. SP K  SE P  SG TE -) DE OE
25. TY AU NW -) TY AU NW IS
26. IS -) TY AU NW IS
27. SP K  SE P  TY AU NW WW -) T1

  ZALEZNOSCI REDUNDANCYJNE:

 1. I  -) W
 2. Z  -) W
 3. PR -) W  I
 4. DY -) W
 5. PK -) I  Z
 6. SP K  SE GR GO D  -) PP PZ
 7. GO D  SA -) PZ

------------------RELACJA NR   1
  KLUCZE:
KLUCZ  1 =  W
KLUCZ  2 =  DZ
  SCHEMAT R 1 =  W  DZ
WSZYSTKIE MOZLIWE SCHEMATY:
 W  DZ
------------------RELACJA NR   2
  KLUCZE:
KLUCZ  1 =  S  SP K
KLUCZ  2 =  NI
  SCHEMAT R 2 =  S  SP K  SE GR NI
WSZYSTKIE MOZLIWE SCHEMATY:
 S  SP K  SE GR NI ,   SP K  SE GR NI NL ,
 SP K  SE GR NI ND
------------------RELACJA NR   3
  KLUCZE:
KLUCZ  1 =  I
KLUCZ  2 =  DY
  SCHEMAT R 3 =  DZ I  DY
WSZYSTKIE MOZLIWE SCHEMATY:
 DZ I  DY ,   W  I  DY
```

```
------------------RELACJA NR   4
  KLUCZE:
KLUCZ  1 =  S
KLUCZ  2 =  NL
KLUCZ  3 =  ND
  SCHEMAT R 4 =  S  N  IM DU NL ND
WSZYSTKIE MOZLIWE SCHEMATY:
 S  N  IM DU NL ND
------------------RELACJA NR   5
  KLUCZE:
KLUCZ  1 =  Z
KLUCZ  2 =  KZ
  SCHEMAT R 5 =  I  Z  KZ
WSZYSTKIE MOZLIWE SCHEMATY:
 I  Z  KZ ,   Z  DY KZ
------------------RELACJA NR   6
  KLUCZE:
KLUCZ  1 =  PR
KLUCZ  2 =  KP
  SCHEMAT R 6 =  Z  PR KP
WSZYSTKIE MOZLIWE SCHEMATY:
 Z  PR KP ,   PR KZ KP
------------------RELACJA NR   7
  KLUCZE:
KLUCZ  1 =  PK
  SCHEMAT R 7 =  PR PK PO
WSZYSTKIE MOZLIWE SCHEMATY:
 PR PK PO ,   KP PK PO
------------------RELACJA NR   8
  KLUCZE:
KLUCZ  1 =  NT
  SCHEMAT R 8 =  PO NT
WSZYSTKIE MOZLIWE SCHEMATY:
 PO NT
------------------RELACJA NR   9
  KLUCZE:
KLUCZ  1 =  SP K
  SCHEMAT R 9 =  W  SP K
WSZYSTKIE MOZLIWE SCHEMATY:
 W  SP K  ,   DZ SP K
------------------RELACJA NR  10
  KLUCZE:
KLUCZ  1 =  SP K  SE
  SCHEMAT R10 =  SP K  SE OR
WSZYSTKIE MOZLIWE SCHEMATY:
 SP K  SE OR
------------------RELACJA NR  11
  KLUCZE:
KLUCZ  1 =  SP K  SE GR
  SCHEMAT R11 =  SP K  SE GR OG ST
WSZYSTKIE MOZLIWE SCHEMATY:
 SP K  SE GR OG ST
------------------RELACJA NR  12
  KLUCZE:
KLUCZ  1 =  SP K  SE GR PP
  SCHEMAT R12 =  SP K  SE GR PP PZ
WSZYSTKIE MOZLIWE SCHEMATY:
 SP K  SE GR PP PZ
------------------RELACJA NR  13
  KLUCZE:
KLUCZ  1 =  SP K  SE GR GO D
KLUCZ  2 =  GO D  SA
  SCHEMAT R13 =  SP K  SE GR PP GO D  SA
WSZYSTKIE MOZLIWE SCHEMATY:
 SP K  SE GR PP GO D  SA
------------------RELACJA NR  14
```



```
  KLUCZE:
KLUCZ  1 =  SP K  SE P
  SCHEMAT R14 =  SP K  SE P  LG WY
WSZYSTKIE MOZLIWE SCHEMATY:
 SP K  SE P  LG WY
------------------RELACJA NR  15
  KLUCZE:
KLUCZ  1 =  SP K  SE P  SZ
  SCHEMAT R15 =  SP K  SE P  SZ DA OZ
WSZYSTKIE MOZLIWE SCHEMATY:
 SP K  SE P  SZ DA OZ
------------------RELACJA NR  16
  KLUCZE:
KLUCZ  1 =  SP K  SE P  SG TE
  SCHEMAT R16 =  SP K  SE P  SG TE DE OE
WSZYSTKIE MOZLIWE SCHEMATY:
 SP K  SE P  SG TE DE OE
------------------RELACJA NR  17
  KLUCZE:
KLUCZ  1 =  TY AU NW
KLUCZ  2 =  IS
  SCHEMAT R17 =  TY AU NW IS
WSZYSTKIE MOZLIWE SCHEMATY:
 TY AU NW IS
------------------RELACJA NR  18
  KLUCZE:
KLUCZ  1 =  SP K  SE P  TY AU NW WW
  SCHEMAT R18 =  SP K  SE P  TY AU NW WW TI
WSZYSTKIE MOZLIWE SCHEMATY:
 SP K  SE P  TY AU NW WW TI ,   SP K  SE P  IS WW TI


 OCENA WSZYSTKICH WERSJI SCHEMATU  RELACYJNEJ BAZY DANYCH

 1. BADANY SCHEMAT:
R 1( 1)= W  DZ                    R 2( 1)= S  SP K  SE GR NI
R 3( 1)= DZ I  DY                 R 4( 1)= S  N  IM DU NL ND
R 5( 1)= I  Z  KZ                 R 6( 1)= Z  PR KP
R 7( 1)= PR PK PO                 R 8( 1)= PO NT
R 9( 1)= W  SP K                  R10( 1)= SP K  SE OR
R11( 1)= SP K  SE GR OG ST        R12( 1)= SP K  SE GR PP PZ
R13( 1)= SP K  SE GR PP GO D  SA  R14( 1)= SP K  SE P  LG WY
R15( 1)= SP K  SE P  SZ DA OZ     R16( 1)= SP K  SE P  SG TE DE OE
R17( 1)= TY AU NW IS              R18( 1)= SP K  SE P  TY AU NW WW TI

    SCHEMATY RELACJI WIAZACYCH DLA X 1 =  I  PK
     I  Z  KZ                     Z  PR KP
     PR PK PO
    LICZBA POLACZEN =  2
    SCHEMATY RELACJI WIAZACYCH DLA X 2 =  I  PR
     I  Z  KZ                     Z  PR KP
    LICZBA POLACZEN =  1
    SCHEMATY RELACJI WIAZACYCH DLA X 3 =  SP K  SE I  PK P  WY
     W  DZ                        DZ I  DY
     I  Z  KZ                     Z  PR KP
     PR PK PO                     W  SP K
     SP K  SE P  LG WY
    LICZBA POLACZEN =  6
    SCHEMATY RELACJI WIAZACYCH DLA X 4 =  SP K  SE GR PP GO D  SA
     SP K  SE GR PP GO D  SA
    LICZBA POLACZEN =  0
    SCHEMATY RELACJI WIAZACYCH DLA X 5 =  SP K  SE GR I  NT OG
     W  DZ                        DZ I  DY
     I  Z  KZ                     Z  PR KP
     PR PK PO                     PO NT
```

```
 W  SP K                          SP K  SE GR OG ST
LICZBA POLACZEN =   7
SCHEMATY RELACJI WIAZACYCH DLA X 6 =  S  SP K  SE GR
 S  SP K  SE GR NI
LICZBA POLACZEN =   0
SCHEMATY RELACJI WIAZACYCH DLA X 7 =  S  SP K  SE GR P  SG TE DE
 S  SP K  SE GR NI                SP K  SE P  SG TE DE OE
LICZBA POLACZEN =   1
SCHEMATY RELACJI WIAZACYCH DLA X 8 =  SP K  SE GR I  PK PP PZ GO D  SA
 W  DZ                            DZ I  DY
 I  Z  KZ                         Z  PR KP
 PR PK PO                         W  SP K
 SP K  SE GR PP PZ                SP K  SE GR PP GO D  SA
LICZBA POLACZEN =   7

WAZONA SUMA POLACZEN DLA BADANEGO SCHEMATU =   80
```

```
 2. BADANY SCHEMAT:
R 1( 1)= W  DZ                    R 2( 2)= SP K  SE GR NI NL
R 3( 1)= DZ I  DY                 R 4( 1)= S  N  IM DU NL ND
R 5( 1)= I  Z  KZ                 R 6( 1)= Z  PR KP
R 7( 1)= PR PK PO                 R 8( 1)= PO NT
R 9( 1)= W  SP K                  R10( 1)= SP K  SE OR
R11( 1)= SP K  SE GR OG ST        R12( 1)= SP K  SE GR PP PZ
R13( 1)= SP K  SE GR PP GO D  SA  R14( 1)= SP K  SE P  LG WY
R15( 1)= SP K  SE P  SZ DA OZ     R16( 1)= SP K  SE P  SG TE DE OE
R17( 1)= TY AU NW IS              R18( 1)= SP K  SE P  TY AU NW WW T1

    SCHEMATY RELACJI WIAZACYCH DLA X 1 =  I  PK
     I  Z  KZ                     Z  PR KP
     PR PK PO
    LICZBA POLACZEN =   2
    SCHEMATY RELACJI WIAZACYCH DLA X 2 =  I  PR
     I  Z  KZ                     Z  PR KP
    LICZBA POLACZEN =   1
    SCHEMATY RELACJI WIAZACYCH DLA X 3 =  SP K  SE I  PK P  WY
     W  DZ                        DZ I  DY
     I  Z  KZ                     Z  PR KP
     PR PK PO                     W  SP K
     SP K  SE P  LG WY
    LICZBA POLACZEN =   6
    SCHEMATY RELACJI WIAZACYCH DLA X 4 =  SP K  SE GR PP GO D  SA
     SP K  SE GR PP GO D  SA
    LICZBA POLACZEN =   0
    SCHEMATY RELACJI WIAZACYCH DLA X 5 =  SP K  SE GR I  NT OG
     W  DZ                        DZ I  DY
     I  Z  KZ                     Z  PR KP
     PR PK PO                     PO NT
     W  SP K                      SP K  SE GR OG ST
    LICZBA POLACZEN =   7
    SCHEMATY RELACJI WIAZACYCH DLA X 6 =  S  SP K  SE GR
     SP K  SE GR NI NL            S  N  IM DU NL ND
    LICZBA POLACZEN =   1
    SCHEMATY RELACJI WIAZACYCH DLA X 7 =  S  SP K  SE GR P  SG TE OE
     SP K  SE GR NI NL            S  N  IM DU NL ND
     SP K  SE P  SG TE DE OE
    LICZBA POLACZEN =   2
    SCHEMATY RELACJI WIAZACYCH DLA X 8 =  SP K  SE GR I  PK PP PZ GO D  SA
     W  DZ                        DZ I  DY
     I  Z  KZ                     Z  PR KP
     PR PK PO                     W  SP K
     SP K  SE GR PP PZ            SP K  SE GR PP GO D  SA
    LICZBA POLACZEN =   7

    WAZONA SUMA POLACZEN DLA BADANEGO SCHEMATU =   89
```

```
 3. BADANY SCHEMAT:
R 1( 1)= W  DZ                    R 2( 3)= SP K  SE GR NI ND
R 3( 1)= DZ I  DY                 R 4( 1)= S  N  IM DU NL ND
R 5( 1)= I  Z  KZ                 R 6( 1)= Z  PR KP
R 7( 1)= PR PK PO                 R 8( 1)= PO NT
R 9( 1)= W  SP K                  R10( 1)= SP K  SE OR
R11( 1)= SP K  SE GR OG ST        R12( 1)= SP K  SE GR PP PZ
R13( 1)= SP K  SE GR PP GO D  SA  R14( 1)= SP K  SE P  LG WY
R15( 1)= SP K  SE P  SZ DA OZ     R16( 1)= SP K  SE P  SG TE DE OE
R17( 1)= TY AU NW IS              R18( 1)= SP K  SE P  TY AU NW WW T1

    SCHEMATY RELACJI WIAZACYCH DLA X 1 =  I  PK
     I  Z  KZ                     Z  PR KP
     PR PK PO
    LICZBA POLACZEN =   2
    SCHEMATY RELACJI WIAZACYCH DLA X 2 =  I  PR
     I  Z  KZ                     Z  PR KP
    LICZBA POLACZEN =   1
    SCHEMATY RELACJI WIAZACYCH DLA X 3 =  SP K  SE I  PK P  WY
     W  DZ                        DZ I  DY
     I  Z  KZ                     Z  PR KP
     PR PK PO                     W  SP K
     SP K  SE P  LG WY
    LICZBA POLACZEN =   6
    SCHEMATY RELACJI WIAZACYCH DLA X 4 =  SP K  SE GR PP GO D  SA
     SP K  SE GR PP GO D  SA
    LICZBA POLACZEN =   0
    SCHEMATY RELACJI WIAZACYCH DLA X 5 =  SP K  SE GR I  NT OG
     W  DZ                        DZ I  DY
     I  Z  KZ                     Z  PR KP
     PR PK PO                     PO NT
     W  SP K                      SP K  SE GR OG ST
    LICZBA POLACZEN =   7
    SCHEMATY RELACJI WIAZACYCH DLA X 6 =  S  SP K  SE GR
     SP K  SE GR NI ND            S  N  IM DU NL ND
    LICZBA POLACZEN =   1
    SCHEMATY RELACJI WIAZACYCH DLA X 7 =  S  SP K  SE GR P  SG TE OE
     SP K  SE GR NI ND            S  N  IM DU NL ND
     SP K  SE P  SG TE DE OE
    LICZBA POLACZEN =   2
    SCHEMATY RELACJI WIAZACYCH DLA X 8 =  SP K  SE GR I  PK PP PZ GO D  SA
     W  DZ                        DZ I  DY
     I  Z  KZ                     Z  PR KP
     PR PK PO                     W  SP K
     SP K  SE GR PP PZ            SP K  SE GR PP GO D  SA
    LICZBA POLACZEN =   7

    WAZONA SUMA POLACZEN DLA BADANEGO SCHEMATU =   89
```

```
 4. BADANY SCHEMAT:
R 1( 1)= W  DZ                    R 2( 1)= S  SP K  SE GR NI
R 3( 2)= W  I  DY                 R 4( 1)= S  N  IM DU NL ND
R 5( 1)= I  Z  KZ                 R 6( 1)= Z  PR KP
R 7( 1)= PR PK PO                 R 8( 1)= PO NT
R 9( 1)= W  SP K                  R10( 1)= SP K  SE OR
R11( 1)= SP K  SE GR OG ST        R12( 1)= SP K  SE GR PP PZ
R13( 1)= SP K  SE GR PP GO D  SA  R14( 1)= SP K  SE P  LG WY
R15( 1)= SP K  SE P  SZ DA OZ     R16( 1)= SP K  SE P  SG TE DE OE
R17( 1)= TY AU NW IS              R18( 1)= SP K  SE P  TY AU NW WW T1

    SCHEMATY RELACJI WIAZACYCH DLA X 1 =  I  PK
     I  Z  KZ                     Z  PR KP
     PR PK PO
```

```
 LICZBA POLACZEN =   2
 SCHEMATY RELACJI WIAZACYCH DLA X 2 =  I  PR
  I  Z  KZ                             Z  PR KP
 LICZBA POLACZEN =   1
 SCHEMATY RELACJI WIAZACYCH DLA X 3 =  SP K  SE I  PK P  WY
  W  I  DY                             I  Z  KZ
  Z  PR KP                             PR PK PO
  W  SP K                              SP K  SE P  LG WY
 LICZBA POLACZEN =   5
 SCHEMATY RELACJI WIAZACYCH DLA X 4 =  SP K  SE GR PP GO D  SA
  SP K  SE GR PP GO D  SA
 LICZBA POLACZEN =   0
 SCHEMATY RELACJI WIAZACYCH DLA X 5 =  SP K  SE GR I  NT OG
  W  I  DY                             I  Z  KZ
  Z  PR KP                             PR PK PO
  PO NT                                W  SP K
  SP K  SE GR OG ST
 LICZBA POLACZEN =   6
 SCHEMATY RELACJI WIAZACYCH DLA X 6 =  S  SP K  SE GR
  S  SP K  SE GR NI
 LICZBA POLACZEN =   0
 SCHEMATY RELACJI WIAZACYCH DLA X 7 =  S  SP K  SE GR P  SG TE OE
  S  SP K  SE GR NI                    SP K  SE P  SG TE DE OE
 LICZBA POLACZEN =   1
 SCHEMATY RELACJI WIAZACYCH DLA X 8 =  SP K  SE GR I  PK PP PZ GO D  SA
  W  I  DY                             I  Z  KZ
  Z  PR KP                             PR PK PO
  W  SP K                              SP K  SE GR PP PZ
  SP K  SE GR PP GO D  SA
 LICZBA POLACZEN =   6

 WAZONA SUMA POLACZEN DLA BADANEGO SCHEMATU =   70
 ------------------------------------
```

5. BADANY SCHEMAT:

```
R 1( 1)= W  DZ                        R 2( 2)= SP K  SE GR NI NL
R 3( 2)= W  I  DY                     R 4( 1)= S  N  IM DU NL ND
R 5( 1)= I  Z  KZ                     R 6( 1)= Z  PR KP
R 7( 1)= PR PK PO                     R 8( 1)= PO NT
R 9( 1)= W  SP K                      R10( 1)= SP K  SE OR
R11( 1)= SP K  SE GR OG ST            R12( 1)= SP K  SE GR PP PZ
R13( 1)= SP K  SE GR PP GO D  SA      R14( 1)= SP K  SE P  LG WY
R15( 1)= SP K  SE P  SZ DA OZ         R16( 1)= SP K  SE P  SG TE DE OE
R17( 1)= TY AU NW IS                  R18( 1)= SP K  SE P  TY AU NW WW T1
```

```
 SCHEMATY RELACJI WIAZACYCH DLA X 1 =  I  PK
  I  Z  KZ                             Z  PR KP
  PR PK PO
 LICZBA POLACZEN =   2
 SCHEMATY RELACJI WIAZACYCH DLA X 2 =  I  PR
  I  Z  KZ                             Z  PR KP
 LICZBA POLACZEN =   1
 SCHEMATY RELACJI WIAZACYCH DLA X 3 =  SP K  SE I  PK P  WY
  W  I  DY                             I  Z  KZ
  Z  PR KP                             PR PK PO
  W  SP K                              SP K  SE P  LG WY
 LICZBA POLACZEN =   5
 SCHEMATY RELACJI WIAZACYCH DLA X 4 =  SP K  SE GR PP GO D  SA
  SP K  SE GR PP GO D  SA
 LICZBA POLACZEN =   0
 SCHEMATY RELACJI WIAZACYCH DLA X 5 =  SP K  SE GR I  NT OG
  W  I  DY                             I  Z  KZ
  Z  PR KP                             PR PK PO
  PO NT                                W  SP K
  SP K  SE GR OG ST
 LICZBA POLACZEN =   6
```



```
 SCHEMATY RELACJI WIAZACYCH DLA X 6 =  S  SP K  SE GR
  SP K  SE GR NI NL                    S  N  IM DU NL ND
 LICZBA POLACZEN =   1
 SCHEMATY RELACJI WIAZACYCH DLA X 7 =  S  SP K  SE GR P  SG TE OE
  SP K  SE GR NI NL                    S  N  IM DU NL ND
  SP K  SE P  SG TE DE OE
 LICZBA POLACZEN =   2
 SCHEMATY RELACJI WIAZACYCH DLA X 8 =  SP K  SE GR I  PK PP PZ GO D  SA
  W  I  DY                             I  Z  KZ
  Z  PR KP                             PR PK PO
  W  SP K                              SP K  SE GR PP PZ
  SP K  SE GR PP GO D  SA
 LICZBA POLACZEN =   6

 WAZONA SUMA POLACZEN DLA BADANEGO SCHEMATU =   79
 ------------------------------------
```

6. BADANY SCHEMAT:

```
R 1( 1)= W  DZ                        R 2( 3)= SP K  SE GR NI ND
R 3( 2)= W  I  DY                     R 4( 1)= S  N  IM DU NL ND
R 5( 1)= I  Z  KZ                     R 6( 1)= Z  PR KP
R 7( 1)= PR PK PO                     R 8( 1)= PO NT
R 9( 1)= W  SP K                      R10( 1)= SP K  SE OR
R11( 1)= SP K  SE GR OG ST            R12( 1)= SP K  SE GR PP PZ
R13( 1)= SP K  SE GR PP GO D  SA      R14( 1)= SP K  SE P  LG WY
R15( 1)= SP K  SE P  SZ DA OZ         R16( 1)= SP K  SE P  SG TE DE OE
R17( 1)= TY AU NW IS                  R18( 1)= SP K  SE P  TY AU NW WW T1
```

```
 SCHEMATY RELACJI WIAZACYCH DLA X 1 =  I  PK
  I  Z  KZ                             Z  PR KP
  PR PK PO
 LICZBA POLACZEN =   2
 SCHEMATY RELACJI WIAZACYCH DLA X 2 =  I  PR
  I  Z  KZ                             Z  PR KP
 LICZBA POLACZEN =   1
 SCHEMATY RELACJI WIAZACYCH DLA X 3 =  SP K  SE I  PK P  WY
  W  I  DY                             I  Z  KZ
  Z  PR KP                             PR PK PO
  W  SP K                              SP K  SE P  LG WY
 LICZBA POLACZEN =   5
 SCHEMATY RELACJI WIAZACYCH DLA X 4 =  SP K  SE GR PP GO D  SA
  SP K  SE GR PP GO D  SA
 LICZBA POLACZEN =   0
 SCHEMATY RELACJI WIAZACYCH DLA X 5 =  SP K  SE GR I  NT OG
  W  I  DY                             I  Z  KZ
  Z  PR KP                             PR PK PO
  PO NT                                W  SP K
  SP K  SE GR OG ST
 LICZBA POLACZEN =   6
 SCHEMATY RELACJI WIAZACYCH DLA X 6 =  S  SP K  SE GR
  SP K  SE GR NI ND                    S  N  IM DU NL ND
 LICZBA POLACZEN =   1
 SCHEMATY RELACJI WIAZACYCH DLA X 7 =  S  SP K  SE GR P  SG TE OE
  SP K  SE GR NI ND                    S  N  IM DU NL ND
  SP K  SE P  SG TE DE OE
 LICZBA POLACZEN =   2
 SCHEMATY RELACJI WIAZACYCH DLA X 8 =  SP K  SE GR I  PK PP PZ GO D  SA
  W  I  DY                             I  Z  KZ
  Z  PR KP                             PR PK PO
  W  SP K                              SP K  SE GR PP PZ
  SP K  SE GR PP GO D  SA
 LICZBA POLACZEN =   6

 WAZONA SUMA POLACZEN DLA BADANEGO SCHEMATU =   79
 ------------------------------------
```

```
 7. BADANY SCHEMAT:
R 1( 1)= W  DZ                    R 2( 1)= S  SP K  SE GR NI
R 3( 1)= DZ I  DY                 R 4( 1)= S  N  IM DU NL ND
R 5( 2)= Z  DY KZ                 R 6( 1)= Z  PR KP
R 7( 1)= PR PK PO                 R 8( 1)= PO NT
R 9( 1)= W  SP K                  R10( 1)= SP K  SE OR
R11( 1)= SP K  SE GR OG ST        R12( 1)= SP K  SE GR PP PZ
R13( 1)= SP K  SE GR PP GO D  SA  R14( 1)= SP K  SE P  LG WY
R15( 1)= SP K  SE P  SZ DA OZ     R16( 1)= SP K  SE P  SG TE DE OE
R17( 1)= TY AU NW IS              R18( 1)= SP K  SE P  TY AU NW WW T1

     SCHEMATY RELACJI WIAZACYCH DLA X 1 =  I  PK
      DZ I  DY                    Z  DY KZ
      Z  PR KP                    PR PK PO
     LICZBA POLACZEN =   3
     SCHEMATY RELACJI WIAZACYCH DLA X 2 =  I  PR
      DZ I  DY                    Z  DY KZ
      Z  PR KP
     LICZBA POLACZEN =   2
     SCHEMATY RELACJI WIAZACYCH DLA X 3 =  SP K  SE I  PK P  WY
      W  DZ                       DZ I  DY
      Z  DY KZ                    Z  PR KP
      PR PK PO                    W  SP K
      SP K  SE P  LG WY
     LICZBA POLACZEN =   6
     SCHEMATY RELACJI WIAZACYCH DLA X 4 =  SP K  SE GR PP GO D  SA
      SP K  SE GR PP GO D  SA
     LICZBA POLACZEN =   0
     SCHEMATY RELACJI WIAZACYCH DLA X 5 =  SP K  SE GR I  NT OG
      W  DZ                       DZ I  DY
      Z  DY KZ                    Z  PR KP
      PR PK PO                    PO NT
      W  SP K                     SP K  SE GR OG ST
     LICZBA POLACZEN =   7
     SCHEMATY RELACJI WIAZACYCH DLA X 6 =  S  SP K  SE GR
      S  SP K  SE GR NI
     LICZBA POLACZEN =   0
     SCHEMATY RELACJI WIAZACYCH DLA X 7 =  S  SP K  SE GR P  SG TE OE
      S  SP K  SE GR NI           SP K  SE P  SG TE DE OE
     LICZBA POLACZEN =   1
     SCHEMATY RELACJI WIAZACYCH DLA X 8 =  SP K  SE GR I  PK PP PZ GO D  SA
      W  DZ                       DZ I  DY
      Z  DY KZ                    Z  PR KP
      PR PK PO                    W  SP K
      SP K  SE GR PP PZ           SP K  SE GR PP GO D  SA
     LICZBA POLACZEN =   7

     WAZONA SUMA POLACZEN DLA BADANEGO SCHEMATU =   86
     ----------------------------------

 8. BADANY SCHEMAT:
R 1( 1)= W  DZ                    R 2( 2)= SP K  SE GR NI NL
R 3( 1)= DZ I  DY                 R 4( 1)= S  N  IM DU NL ND
R 5( 2)= Z  DY KZ                 R 6( 1)= Z  PR KP
R 7( 1)= PR PK PO                 R 8( 1)= PO NT
R 9( 1)= W  SP K                  R10( 1)= SP K  SE OR
R11( 1)= SP K  SE GR OG ST        R12( 1)= SP K  SE GR PP PZ
R13( 1)= SP K  SE GR PP GO D  SA  R14( 1)= SP K  SE P  LG WY
```



```
R15( 1)= SP K  SE P  SZ DA OZ     R16( 1)= SP K  SE P  SG TE DE OE
R17( 1)= TY AU NW IS              R18( 1)= SP K  SE P  TY AU NW WW T1

     SCHEMATY RELACJI WIAZACYCH DLA X 1 =  I  PK
      DZ I  DY                    Z  DY KZ
      Z  PR KP                    PR PK PO
     LICZBA POLACZEN =   3
     SCHEMATY RELACJI WIAZACYCH DLA X 2 =  I  PR
      DZ I  DY                    Z  DY KZ
      Z  PR KP
     LICZBA POLACZEN =   2
     SCHEMATY RELACJI WIAZACYCH DLA X 3 =  SP K  SE I  PK P  WY
      W  DZ                       DZ I  DY
      Z  DY KZ                    Z  PR KP
      PR PK PO                    W  SP K
      SP K  SE P  LG WY
     LICZBA POLACZEN =   6
     SCHEMATY RELACJI WIAZACYCH DLA X 4 =  SP K  SE GR PP GO D  SA
      SP K  SE GR PP GO D  SA
     LICZBA POLACZEN =   0
     SCHEMATY RELACJI WIAZACYCH DLA X 5 =  SP K  SE GR I  NT OG
      W  DZ                       DZ I  DY
      Z  DY KZ                    Z  PR KP
      PR PK PO                    PO NT
      W  SP K                     SP K  SE GR OG ST
     LICZBA POLACZEN =   7
     SCHEMATY RELACJI WIAZACYCH DLA X 6 =  S  SP K  SE GR
      S  SP K  SE GR NI NL        S  N  IM DU NL ND
     LICZBA POLACZEN =   1
     SCHEMATY RELACJI WIAZACYCH DLA X 7 =  S  SP K  SE GR P  SG TE OE
      SP K  SE GR NI NL           S  N  IM DU NL ND
      SP K  SE P  SG TE DE OE
     LICZBA POLACZEN =   2
     SCHEMATY RELACJI WIAZACYCH DLA X 8 =  SP K  SE GR I  PK PP PZ GO D  SA
      W  DZ                       DZ I  DY
      Z  DY KZ                    Z  PR KP
      PR PK PO                    W  SP K
      SP K  SE GR PP PZ           SP K  SE GR PP GO D  SA
     LICZBA POLACZEN =   7

     WAZONA SUMA POLACZEN DLA BADANEGO SCHEMATU =   95
     ----------------------------------

 9. BADANY SCHEMAT:
R 1( 1)= W  DZ                    R 2( 3)= SP K  SE GR NI ND
R 3( 1)= DZ I  DY                 R 4( 1)= S  N  IM DU NL ND
R 5( 2)= Z  DY KZ                 R 6( 1)= Z  PR KP
R 7( 1)= PR PK PO                 R 8( 1)= PO NT
R 9( 1)= W  SP K                  R10( 1)= SP K  SE OR
R11( 1)= SP K  SE GR OG ST        R12( 1)= SP K  SE GR PP PZ
R13( 1)= SP K  SE GR PP GO D  SA  R14( 1)= SP K  SE P  LG WY
R15( 1)= SP K  SE P  SZ DA OZ     R16( 1)= SP K  SE P  SG TE DE OE
R17( 1)= TY AU NW IS              R18( 1)= SP K  SE P  TY AU NW WW T1

     SCHEMATY RELACJI WIAZACYCH DLA X 1 =  I  PK
      DZ I  DY                    Z  DY KZ
      Z  PR KP                    PR PK PO
     LICZBA POLACZEN =   3
     SCHEMATY RELACJI WIAZACYCH DLA X 2 =  I  PR
      DZ I  DY                    Z  DY KZ
      Z  PR KP
     LICZBA POLACZEN =   2
     SCHEMATY RELACJI WIAZACYCH DLA X 3 =  SP K  SE I  PK P  WY
      W  DZ                       DZ I  DY
      Z  DY KZ                    Z  PR KP
      PR PK PO                    W  SP K
```

```
 SP K  SE P  LG WY
LICZBA POLACZEN =   6
SCHEMATY RELACJI WIAZACYCH DLA X 4 =  SP K  SE GR PP GO D  SA
 SP K  SE GR PP GO D  SA
LICZBA POLACZEN =   0
SCHEMATY RELACJI WIAZACYCH DLA X 5 =  SP K  SE GR I  NT OG
 W  DZ                        DZ I  DY
 Z  DY KZ                     Z  PR KP
 PR PK PO                     PO NT
 W  SP K                      SP K  SE GR OG ST
LICZBA POLACZEN =   7
SCHEMATY RELACJI WIAZACYCH DLA X 6 =  S  SP K  SE GR
 SP K  SE GR NI ND            S  N  IM DU NL ND
LICZBA POLACZEN =   1
SCHEMATY RELACJI WIAZACYCH DLA X 7 =  S  SP K  SE GR P  SG TE OE
 SP K  SE GR NI ND            S  N  IM DU NL ND
 SP K  SE P  SG TE DE OE
LICZBA POLACZEN =   2
SCHEMATY RELACJI WIAZACYCH DLA X 8 =  SP K  SE GR I  PK PP PZ GO D  SA
 W  DZ                        DZ I  DY
 Z  DY KZ                     Z  PR KP
 PR PK PO                     W  SP K
 SP K  SE GR PP PZ            SP K  SE GR PP GO D  SA
LICZBA POLACZEN =   7

WAZONA SUMA POLACZEN DLA BADANEGO SCHEMATU =   95
```

10. BADANY SCHEMAT:

```
R 1( 1)= W  DZ                    R 2( 1)= S  SP K  SE GR NI
R 3( 2)= W  I  DY                 R 4( 1)= S  N  IM DU NL ND
R 5( 2)= Z  DY KZ                 R 6( 1)= Z  PR KP
R 7( 1)= PR PK PO                 R 8( 1)= PO NT
R 9( 1)= W  SP K                  R10( 1)= SP K  SE OR
R11( 1)= SP K  SE GR OG ST        R12( 1)= SP K  SE GR PP PZ
R13( 1)= SP K  SE GR PP GO D  SA  R14( 1)= SP K  SE P  LG WY
R15( 1)= SP K  SE P  SZ DA OZ     R16( 1)= SP K  SE P  SG TE DE OE
R17( 1)= TY AU NW IS              R18( 1)= SP K  SE P  TY AU NW WW TI

     SCHEMATY RELACJI WIAZACYCH DLA X 1 =  I  PK
      W  I  DY                    Z  DY KZ
      Z  PR KP                    PR PK PO
     LICZBA POLACZEN =   3
     SCHEMATY RELACJI WIAZACYCH DLA X 2 =  I  PR
      W  I  DY                    Z  DY KZ
      Z  PR KP
     LICZBA POLACZEN =   2
     SCHEMATY RELACJI WIAZACYCH DLA X 3 =  SP K  SE I  PK P  WY
      W  I  DY                    Z  DY KZ
      Z  PR KP                    PR PK PO
      W  SP K                     SP K  SE P  LG WY
     LICZBA POLACZEN =   5
     SCHEMATY RELACJI WIAZACYCH DLA X 4 =  SP K  SE GR PP GO D  SA
      SP K  SE GR PP GO D  SA
     LICZBA POLACZEN =   0
     SCHEMATY RELACJI WIAZACYCH DLA X 5 =  SP K  SE GR I  NT OG
      W  I  DY                    Z  DY KZ
      Z  PR KP                    PR PK PO
      PO NT                       W  SP K
      SP K  SE GR OG ST
     LICZBA POLACZEN =   6
     SCHEMATY RELACJI WIAZACYCH DLA X 6 =  S  SP K  SE GR
      S  SP K  SE GR NI
     LICZBA POLACZEN =   0
     SCHEMATY RELACJI WIAZACYCH DLA X 7 =  S  SP K  SE GR P  SG TE OE
      S  SP K  SE GR NI           SP K  SE P  SG TE DE OE
```



```
     LICZBA POLACZEN =   1
     SCHEMATY RELACJI WIAZACYCH DLA X 8 =  SP K  SE GR I  PK PP PZ GO D  SA
      W  I  DY                    Z  DY KZ
      Z  PR KP                    PR PK PO
      W  SP K                     SP K  SE GR PP PZ
      SP K  SE GR PP GO D  SA
     LICZBA POLACZEN =   6

     WAZONA SUMA POLACZEN DLA BADANEGO SCHEMATU =   76
```

11. BADANY SCHEMAT:

```
R 1( 1)= W  DZ                    R 2( 2)= SP K  SE GR NI NL
R 3( 2)= W  I  DY                 R 4( 1)= S  N  IM DU NL ND
R 5( 2)= Z  DY KZ                 R 6( 1)= Z  PR KP
R 7( 1)= PR PK PO                 R 8( 1)= PO NT
R 9( 1)= W  SP K                  R10( 1)= SP K  SE OR
R11( 1)= SP K  SE GR OG ST        R12( 1)= SP K  SE GR PP PZ
R13( 1)= SP K  SE GR PP GO D  SA  R14( 1)= SP K  SE P  LG WY
R15( 1)= SP K  SE P  SZ DA OZ     R16( 1)= SP K  SE P  SG TE DE OE
R17( 1)= TY AU NW IS              R18( 1)= SP K  SE P  TY AU NW WW TI

     SCHEMATY RELACJI WIAZACYCH DLA X 1 =  I  PK
      W  I  DY                    Z  DY KZ
      Z  PR KP                    PR PK PO
     LICZBA POLACZEN =   3
     SCHEMATY RELACJI WIAZACYCH DLA X 2 =  I  PR
      W  I  DY                    Z  DY KZ
      Z  PR KP
     LICZBA POLACZEN =   2
     SCHEMATY RELACJI WIAZACYCH DLA X 3 =  SP K  SE I  PK P  WY
      W  I  DY                    Z  DY KZ
      Z  PR KP                    PR PK PO
      W  SP K                     SP K  SE P  LG WY
     LICZBA POLACZEN =   5
     SCHEMATY RELACJI WIAZACYCH DLA X 4 =  SP K  SE GR PP GO D  SA
      SP K  SE GR PP GO D  SA
     LICZBA POLACZEN =   0
     SCHEMATY RELACJI WIAZACYCH DLA X 5 =  SP K  SE GR I  NT OG
      W  I  DY                    Z  DY KZ
      Z  PR KP                    PR PK PO
      PO NT                       W  SP K
      SP K  SE GR OG ST
     LICZBA POLACZEN =   6
     SCHEMATY RELACJI WIAZACYCH DLA X 6 =  S  SP K  SE GR
      SP K  SE GR NI NL           S  N  IM DU NL ND
     LICZBA POLACZEN =   1
     SCHEMATY RELACJI WIAZACYCH DLA X 7 =  S  SP K  SE GR P  SG TE OE
      SP K  SE GR NI NL           S  N  IM DU NL ND
      SP K  SE P  SG TE DE OE
     LICZBA POLACZEN =   2
     SCHEMATY RELACJI WIAZACYCH DLA X 8 =  SP K  SE GR I  PK PP PZ GO D  SA
      W  I  DY                    Z  DY KZ
      Z  PR KP                    PR PK PO
      W  SP K                     SP K  SE GR PP PZ
      SP K  SE GR PP GO D  SA
     LICZBA POLACZEN =   6

     WAZONA SUMA POLACZEN DLA BADANEGO SCHEMATU =   85
```

12. BADANY SCHEMAT:

```
R 1( 1)= W  DZ                    R 2( 3)= SP K  SE GR NI ND
R 3( 2)= W  I  DY                 R 4( 1)= S  N  IM DU NL ND
R 5( 2)= Z  DY KZ                 R 6( 1)= Z  PR KP
R 7( 1)= PR PK PO                 R 8( 1)= PO NT
```

```
R 9( 1)= W  SP K                    R10( 1)= SP K  SE OR
R11( 1)= SP K  SE GR OG ST          R12( 1)= SP K  SE GR PP PZ
R13( 1)= SP K  SE GR PP GO D  SA    R14( 1)= SP K  SE P  LG WY
R15( 1)= SP K  SE P  SZ DA OZ       R16( 1)= SP K  SE P  SG TE DE OE
R17( 1)= TY AU NW IS                R18( 1)= SP K  SE P  TY AU NW WW T1

   SCHEMATY RELACJI WIAZACYCH DLA X 1 =  I  PK
    W  I  DY                        Z  DY KZ
    Z  PR KP                        PR PK PO
   LICZBA POLACZEN =   3
   SCHEMATY RELACJI WIAZACYCH DLA X 2 =  I  PR
    W  I  DY                        Z  DY KZ
    Z  PR KP
   LICZBA POLACZEN =   2
   SCHEMATY RELACJI WIAZACYCH DLA X 3 =  SP K  SE I  PK P  WY
    W  I  DY                        Z  DY KZ
    Z  PR KP                        PR PK PO
    W  SP K                         SP K  SE P  LG WY
   LICZBA POLACZEN =   5
   SCHEMATY RELACJI WIAZACYCH DLA X 4 =  SP K  SE GR PP GO D  SA
    SP K  SE GR PP GO D  SA
   LICZBA POLACZEN =   0
   SCHEMATY RELACJI WIAZACYCH DLA X 5 =  SP K  SE GR I  NT OG
    W  I  DY                        Z  DY KZ
    Z  PR KP                        PR PK PO
    PO NT                           W  SP K
    SP K  SE GR OG ST
   LICZBA POLACZEN =   6
   SCHEMATY RELACJI WIAZACYCH DLA X 6 =  S  SP K  SE GR
    SP K  SE GR NI ND               S  N  IM DU NL ND
   LICZBA POLACZEN =   1
   SCHEMATY RELACJI WIAZACYCH DLA X 7 =  S  SP K  SE GR P  SG TE OE
    SP K  SE GR NI ND               S  N  IM DU NL ND
    SP K  SE P  SG TE DE OE
   LICZBA POLACZEN =   2
   SCHEMATY RELACJI WIAZACYCH DLA X 8 =  SP K  SE GR I  PK PP PZ GO D  SA
    W  I  DY                        Z  DY KZ
    Z  PR KP                        PR PK PO
    W  SP K                         SP K  SE GR PP PZ
    SP K  SE GR PP GO D  SA
   LICZBA POLACZEN =   6

   WAZONA SUMA POLACZEN DLA BADANEGO SCHEMATU =   85
   ----------------------------------

13. BADANY SCHEMAT:
R 1( 1)= W  DZ                      R 2( 1)= S  SP K  SE GR NI
R 3( 1)= DZ I  DY                   R 4( 1)= S  N  IM DU NL ND
R 5( 1)= I  Z  KZ                   R 6( 2)= PR KZ KP
R 7( 1)= PR PK PO                   R 8( 1)= PO NT
R 9( 1)= W  SP K                    R10( 1)= SP K  SE OR
R11( 1)= SP K  SE GR OG ST          R12( 1)= SP K  SE GR PP PZ
R13( 1)= SP K  SE GR PP GO D  SA    R14( 1)= SP K  SE P  LG WY
R15( 1)= SP K  SE P  SZ DA OZ       R16( 1)= SP K  SE P  SG TE DE OE
R17( 1)= TY AU NW IS                R18( 1)= SP K  SE P  TY AU NW WW T1

   SCHEMATY RELACJI WIAZACYCH DLA X 1 =  I  PK
    I  Z  KZ                        PR KZ KP
    PR PK PO
   LICZBA POLACZEN =   2
   SCHEMATY RELACJI WIAZACYCH DLA X 2 =  I  PR
    I  Z  KZ                        PR KZ KP
   LICZBA POLACZEN =   1
   SCHEMATY RELACJI WIAZACYCH DLA X 3 =  SP K  SE I  PK P  WY
    W  DZ                           DZ I  DY
    I  Z  KZ                        PR KZ KP
```



```
    PR PK PO                        W  SP K
    SP K  SE P  LG WY
   LICZBA POLACZEN =   6
   SCHEMATY RELACJI WIAZACYCH DLA X 4 =  SP K  SE GR PP GO D  SA
    SP K  SE GR PP GO D  SA
   LICZBA POLACZEN =   0
   SCHEMATY RELACJI WIAZACYCH DLA X 5 =  SP K  SE GR I  NT OG
    W  DZ                           DZ I  DY
    I  Z  KZ                        PR KZ KP
    PR PK PO                        PO NT
    W  SP K                         SP K  SE GR OG ST
   LICZBA POLACZEN =   7
   SCHEMATY RELACJI WIAZACYCH DLA X 6 =  S  SP K  SE GR
    S  SP K  SE GR NI
   LICZBA POLACZEN =   0
   SCHEMATY RELACJI WIAZACYCH DLA X 7 =  S  SP K  SE GR P  SG TE OE
    S  SP K  SE GR NI               SP K  SE P  SG TE DE OE
   LICZBA POLACZEN =   1
   SCHEMATY RELACJI WIAZACYCH DLA X 8 =  SP K  SE GR I  PK PP PZ GO D  SA
    W  DZ                           DZ I  DY
    I  Z  KZ                        PR KZ KP
    PR PK PO                        W  SP K
    SP K  SE GR PP PZ               SP K  SE GR PP GO D  SA
   LICZBA POLACZEN =   7

   WAZONA SUMA POLACZEN DLA BADANEGO SCHEMATU =   80
   ----------------------------------

14. BADANY SCHEMAT:
R 1( 1)= W  DZ                      R 2( 2)= SP K  SE GR NI NL
R 3( 1)= DZ I  DY                   R 4( 1)= S  N  IM DU NL ND
R 5( 1)= I  Z  KZ                   R 6( 2)= PR KZ KP
R 7( 1)= PR PK PO                   R 8( 1)= PO NT
R 9( 1)= W  SP K                    R10( 1)= SP K  SE OR
R11( 1)= SP K  SE GR OG ST          R12( 1)= SP K  SE GR PP PZ
R13( 1)= SP K  SE GR PP GO D  SA    R14( 1)= SP K  SE P  LG WY
R15( 1)= SP K  SE P  SZ DA OZ       R16( 1)= SP K  SE P  SG TE DE OE
R17( 1)= TY AU NW IS                R18( 1)= SP K  SE P  TY AU NW WW T1

   SCHEMATY RELACJI WIAZACYCH DLA X 1 =  I  PK
    I  Z  KZ                        PR KZ KP
    PR PK PO
   LICZBA POLACZEN =   2
   SCHEMATY RELACJI WIAZACYCH DLA X 2 =  I  PR
    I  Z  KZ                        PR KZ KP
   LICZBA POLACZEN =   1
   SCHEMATY RELACJI WIAZACYCH DLA X 3 =  SP K  SE I  PK P  WY
    W  DZ                           DZ I  DY
    I  Z  KZ                        PR KZ KP
    PR PK PO                        W  SP K
    SP K  SE P  LG WY
   LICZBA POLACZEN =   6
   SCHEMATY RELACJI WIAZACYCH DLA X 4 =  SP K  SE GR PP GO D  SA
    SP K  SE GR PP GO D  SA
   LICZBA POLACZEN =   0
   SCHEMATY RELACJI WIAZACYCH DLA X 5 =  SP K  SE GR I  NT OG
    W  DZ                           DZ I  DY
    I  Z  KZ                        PR KZ KP
    PR PK PO                        PO NT
    W  SP K                         SP K  SE GR OG ST
   LICZBA POLACZEN =   7
   SCHEMATY RELACJI WIAZACYCH DLA X 6 =  S  SP K  SE GR
    SP K  SE GR NI NL               S  N  IM DU NL ND
   LICZBA POLACZEN =   1
   SCHEMATY RELACJI WIAZACYCH DLA X 7 =  S  SP K  SE GR P  SG TE OE
    SP K  SE GR NI NL               S  N  IM DU NL ND
```

```
 SP K  SE P  SG TE DE OE
LICZBA POLACZEN =   2
SCHEMATY RELACJI WIAZACYCH DLA X 8 =  SP K  SE GR I  PK PP PZ GO D  SA
 W  DZ                              DZ I  DY
 I  Z  KZ                           PR KZ KP
 PR PK PO                           W  SP K
 SP K  SE GR PP PZ                  SP K  SE GR PP GO D  SA
LICZBA POLACZEN =   7

WAZONA SUMA POLACZEN DLA BADANEGO SCHEMATU =   89
-----------------------------------

15. BADANY SCHEMAT:
R 1( 1)= W  DZ                     R 2( 3)= SP K  SE GR NI ND
R 3( 1)= DZ I  DY                  R 4( 1)= S  N  IM DU NL ND
R 5( 1)= I  Z  KZ                  R 6( 2)= PR KZ KP
R 7( 1)= PR PK PO                  R 8( 1)= PO NT
R 9( 1)= W  SP K                   R10( 1)= SP K  SE OR
R11( 1)= SP K  SE GR OG ST         R12( 1)= SP K  SE GR PP PZ
R13( 1)= SP K  SE GR PP GO D  SA   R14( 1)= SP K  SE P  LG WY
R15( 1)= SP K  SE P  SZ DA OZ      R16( 1)= SP K  SE P  SG TE DE OE
R17( 1)= TY AU NW IS               R18( 1)= SP K  SE P  TY AU NW WW T1

SCHEMATY RELACJI WIAZACYCH DLA X 1 =  I  PK
 I  Z  KZ                          PR KZ KP
 PR PK PO
LICZBA POLACZEN =   2
SCHEMATY RELACJI WIAZACYCH DLA X 2 =  I  PR
 I  Z  KZ                          PR KZ KP
LICZBA POLACZEN =   1
SCHEMATY RELACJI WIAZACYCH DLA X 3 =  SP K  SE I  PK P  WY
 W  DZ                              DZ I  DY
 I  Z  KZ                           PR KZ KP
 PR PK PO                           W  SP K
 SP K  SE P  LG WY
LICZBA POLACZEN =   6
SCHEMATY RELACJI WIAZACYCH DLA X 4 =  SP K  SE GR PP GO D  SA
 SP K  SE GR PP GO D  SA
LICZBA POLACZEN =   0
SCHEMATY RELACJI WIAZACYCH DLA X 5 =  SP K  SE GR I  NT OG
 W  DZ                              DZ I  DY
 I  Z  KZ                           PR KZ KP
 PR PK PO                           PO NT
 W  SP K                            SP K  SE GR OG ST
LICZBA POLACZEN =   7
SCHEMATY RELACJI WIAZACYCH DLA X 6 =  S  SP K  SE GR
 SP K  SE GR NI ND                  S  N  IM DU NL ND
LICZBA POLACZEN =   1
SCHEMATY RELACJI WIAZACYCH DLA X 7 =  S  SP K  SE GR P  SG TE OE
 SP K  SE GR NI ND                  S  N  IM DU NL ND
 SP K  SE P  SG TE DE OE
LICZBA POLACZEN =   2
SCHEMATY RELACJI WIAZACYCH DLA X 8 =  SP K  SE GR I  PK PP PZ GO D  SA
 W  DZ                              DZ I  DY
 I  Z  KZ                           PR KZ KP
 PR PK PO                           W  SP K
 SP K  SE GR PP PZ                  SP K  SE GR PP GO D  SA
LICZBA POLACZEN =   7

WAZONA SUMA POLACZEN DLA BADANEGO SCHEMATU =   89
-----------------------------------
```

^**********

PROJEKTOWANIE STRUKTURY LOGICZNEJ
RELACYJNYCH BAZ DANYCH METODĄ SYNTEZY
PRZY UWZGLĘDNIENIU ZADAŃ WYSZUKIWANIA DANYCH

S t r e s z c z e n i e

W prezentowanej pracy podjęto próbę powiązania procesu syntezy schematu relacyjnej bazy danych z oceną przydatności struktury projektowanej bazy do rozwiązywania przewidywanych zadań wyszukiwania danych.

Punktem wyjścia przedstawionych badań jest modyfikacja znanego z literatury algorytmu syntezy Bernsteina. Wprowadzenie nowej koncepcji pokrycia zbioru zależności funkcjonalnych zawierającego wszystkie bijekcje między kluczami równoważnymi pozwoliło na uniknięcie w procesie syntezy powtarzania operacji usuwania zależności redundancyjnych. Opracowano nowy algorytm syntezy wykorzystujący tę koncepcję nazwaną pokryciem dla potrzeb syntezy.

Analizując proces syntezy zwrócono uwagę na możliwość uzyskania różnych rozwiązań dla tych samych danych wejściowych. Pokazano przy tym, że różne schematy mogą wymagać odmiennego rozwiązywania zadań wyszukiwania danych. Zaproponowano kryterium oceny różnych struktur tej samej bazy uwzględniające złożoność zadań wyszukiwania danych. Wprowadzenie nowego kryterium wymagało rozwiązania dwóch problemów:

- tworzenia wszystkich możliwych postaci schematu relacyjnej bazy danych dla określonych danych wejściowych,
- wyznaczanie zbiorów relacji potrzebnych do rozwiązania zdefiniowanych zadań wyszukiwania danych.

Pierwsze zagadnienie wymagało rozwinięcia badań własności zależności funkcjonalnych, które mogą zostać wybrane do nieredundancyjnego pokrycia danego zbioru tych zależności. Zdefiniowano tak zwane alternatywnie redundancyjne zależności funkcjonalne i ich zbiory. Wykazano, że własność alternatywnej redundancyjności jest relacją równoważności.

Udowodniono też między innymi, że lewe strony zależności wzajemnie alternatywnie redundancyjnych należą do wspólnej klasy równoważności kluczy względem bijekcji. Zbadane własności pozwoliły zdekomponować i uprościć proces tworzenia wszystkich postaci pokrycia i schematu relacyjnej bazy danych.

Dla wyznaczenia relacji potrzebnych do rozwiązania zadania wyszukiwania danych wykorzystano metody stosowane w dziedzinie acyklicznych sche-

matów relacyjnych baz danych i częściowo relacji uniwersalnej. Opracowano algorytm przekształcający acykliczny hipergraf danej bazy w graf, który jest następnie redukowany do zbioru węzłów odpowiadających poszukiwanym relacjom. Metoda ta pozwala ocenić złożoność zadań wyszukiwania danych w rozważanej bazie i w konsekwencji wybrać schemat najlepszy spośród wszystkich schematów utworzonych dla przyjętych danych wejściowych.

Dodatek do pracy prezentuje proces projektowania przykładowej bazy danych przy wykorzystaniu zmodyfikowanego algorytmu syntezy.

ПРОЕКТИРОВАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ РЕЛЯЦИОННЫХ БАЗ ДАННЫХ МЕТОДОМ СИНТЕЗА С УЧЁТОМ ЗАДАЧ ПОИСКА ДАННЫХ

Р е з ю м е

В работе предпринята попытка указания связей процесса синтеза схемы реляционной базы данных с оценкой пригодности структуры проектированной базы для решения намечаемых задач поиска данных.

Исходной точкой представленных исследований является модификация известного по литературным источникам алгоритма синтеза Бернштейна. Введение новой концепции покрытия множества функциональных зависимостей содержащей все биекции между равнозначными ключами, позволило избежать в процессе синтеза повторения операций удаления редундантных зависимостей. Разработан новый алгоритм синтеза, использующий эту концепцию, называемой покрытием для нужд синтеза.

Анализируя процесс синтеза, обращено внимание на возможность получения различных решений для тех самых исходных данных. Показано подчас, что различные схемы могут требовать иного решения задач поиска данных. Предложен критерий оценки разных структур той-же базы, учитывающий сложность задач поиска данных. Введение нового критерия требовало решения двух проблем:

- получения всех возможных видов схем реляционных баз данных для определённых входных данных,
- получения множеств зависимостей нужных для решения определённых задач поиска данных.

Решение первого вопроса требовало расширения исследований свойств функциональных зависимостей, которые могут быть выбраны для нередундантного покрытия данного множества этих зависимостей. Определены т.н. альтернативно редундантные функциональные зависимости и их множества. Показано, что свойство альтернативной редундантности является зависимостью эквивалентности. Доказано также, что левые части взаимно альтернативно редундантных зависимостей пренадлежат к общему классу эквивалентности ключей по отношению к биекциям. Исследованные зависимости позволили декомпонировать и упростить процесс создавания всех видов покрытия и схемы реляционной базы данных.

С целью определения зависимостей нужных для решения задачи поиска данных, использованы методы применяемые в отрасли ациклических схем реляционных баз данных и частично - универсальных зависимостей. Разработан алгоритм преобразующий ациклический гиперграф данной базы в граф, который в последствии редуцируется к множеству связей соответствующим поисковым

зависимостям. Метод этот даёт возможность оценить сложность задач поиска данных в рассматриваемой базе и в результате выбрать наилучшую схему среди всех схем созданных по принятым исходным данным.

В дополнении к работе показывается процесс проектирования примерной базы данных с использованием модифицированного алгоритма синтеза.

DESIGNING OF RELATIONAL DATABASE LOGICAL STRUCTURE
BY SYNTHESIS METHOD WITH TAKING QUERY PROCESSING INTO CONSIDERATION

S u m m a r y

The trial of connection of the relational database schema synthesis process with estimation of designed schema usefulness to foreseen query processing is presented in this paper. The starting point of presented investigations was modification of the known in literature Bernstein's synthesis algorithm. New conception of cover of functional dependencies' set was introduced. This cover - called cover for synthesis purposes - contains all bijections binding equivalent keys. Single process of elimination of redundant functional dependencies allows to obtain 3NF relation schemes in new algorithm basing on the idea of this cover.

In analysis of the synthesis process there was paid attention to possibility of obtaining different results for the same input set of functional dependencies.

It was shown in examples, that different schemata may lead to different query processing. There was introduced a criterion of estimation of different schemata of just the same database, taking computational complexity of foreseen query processing into consideration.

Solution of two problems is required to determine the new criterion:

- creating all possible forms database schema for input data,
- finding sets of relations needed to processing of foreseen queries.

Research of properties of functional dependencies forming a nonredundant cover were developed within the framework of the above first problem.

Alternatively redundant functional dependencies were defined. It was shown, that alternative redundancy is an equivalence relation. It was proved also, that the left sides of alternatively redundant dependencies belong to the same class of equivalence with regard to bijection. These properties allowed to decompose the process of obtaining all forms of relational database schema.

Methods used in the area of acyclic database schemata were applied to find sets of relations needed to query processing. In this work an algorithm is presented determining these sets of relations in acyclic hypergraph representing given database.

Cardinality of these sets is a measure of queries processing complexity and makes possible choise the best database schema from among all schemata created for given input set of functional dependencies.

In the appendix there is an example of database designing process with the use of a program based on the new synthesis algorithm.

BIBLIOTEKA GŁÓWNA
Politechniki Śląskiej
P. 4193/88/11

WYDAWNICTWA NAUKOWE I DYDAKTYCZNE POLITECHNIKI ŚLĄSKIEJ MOŻNA NABYĆ W NASTĘPUJĄCYCH PLACÓWKACH:

44-100 Gliwice — Księgarnia nr 096, ul. Konstytucji 14 b
44-100 Gliwice — Spółdzielnia Studencka, ul. Wrocławska 4 a
40-950 Katowice — Księgarnia nr 015, ul. Żwirki i Wigury 33
40-096 Katowice — Księgarnia nr 005, ul. 3 Maja 12
41-900 Bytom — Księgarnia nr 048, Pl. Kościuszki 10
41-500 Chorzów — Księgarnia nr 063, ul. Wolności 22
41-300 Dąbrowa Górnicza — Księgarnia nr 081, ul. ZBoWiD-u 2
47-400 Racibórz — Księgarnia nr 148, ul. Odrzańska 1
44-200 Rybnik — Księgarnia nr 162, Rynek 1
41-200 Sosnowiec — Księgarnia nr 181, ul. Zwycięstwa 7
41-800 Zabrze — Księgarnia nr 230, ul. Wolności 288
00-901 Warszawa — Ośrodek Rozpowszechniania Wydawnictw Naukowych PAN — Pałac Kultury i Nauki

Wszystkie wydawnictwa naukowe i dydaktyczne zamawiać można poprzez Składnicę Księgarską w Warszawie, ul. Mazowiecka 9.